

National College of Ireland

BSHCSDE4

Software Development

2020

Cian Harrison

X17116970

X17116970@student.ncirl.ie

check-mate

Technical Report

Contents

Executive Summary	Error! Bookmark not defined.
1.0 Introduction	5
1.1. Background	5
1.2. Aims	6
1.2.1. Simplify Emotions for Users	6
1.2.2. Provide an Opportunity to Share	6
1.2.3. Empower Users	6
1.3. Technology	7
1.3.1. Bootstrap, HTML & CSS	7
1.3.2. JavaScript & JSX	7
1.3.3. Node & Express	7
1.3.4. MongoDB & Mongoose	7
1.3.5. React & Redux	7
1.4. Structure	8
2.0 System	9
2.1. Requirements	9
2.1.1. Functional Requirements	9
2.1.1.1. Use Case Diagram	9
2.1.1.2. Requirement 1: Register & Authenticate Users	10
2.1.1.2.1. Description & Priority	10
2.1.1.2.2. Use Case: R01	10
2.1.1.3. Requirement 2: Create & Manage User Profile	12
2.1.1.3.1. Description & Priority	12
2.1.1.3.2. Use Case: R02	12
2.1.1.4. Requirement 3: Create & Manage Checkins	14
2.1.1.4.1. Description & Priority	14
2.1.1.4.2. Use Case: R03	14
2.1.1.5. Requirement 4: Display User Profiles	16
2.1.1.5.1. Description & Priority	16
2.1.1.5.2. Use Case: R04	16
2.1.1.6. Requirement 5: Create & Manage Interactions	18
2.1.1.6.1. Description & Priority	18
2.1.1.6.2. Use Case: R05	18
2.1.1.7. Requirement 6: Create & Manage Real-Time Chat	20
2.1.1.7.1. Description & Priority	20

2.1.1.7.2.	Use Case: R06.....	20
2.1.2.	Data Requirements	22
2.1.3.	User Requirements.....	22
2.1.3.1.	Confidentiality	22
2.1.3.2.	Security.....	22
2.1.3.3.	Data Protection - GDPR.....	22
2.1.4.	Environmental Requirements	23
2.1.4.1.	Connectivity	23
2.1.4.2.	Compute.....	23
2.1.5.	Usability Requirements	23
2.1.5.1.	Accessibility	23
2.1.5.2.	Learnability.....	23
2.1.5.3.	Navigation	23
2.1.5.4.	User Friendliness	23
2.2.	Design & Architecture	24
2.2.1.	Class Diagram	24
2.2.2.	System Architecture Diagram	25
2.3.	Implementation.....	26
2.3.1.	Back-end.....	26
2.3.2.	Front-end.....	30
2.4.	Graphical User Interface (GUI)	34
2.4.1.	Home Component.....	34
2.4.2.	Dashboard Component	35
2.4.3.	Personal Checkin Component	36
2.4.4.	Public Checkins Component.....	37
2.4.5.	Checkin Discussion Component	38
2.4.6.	Profile Component	39
2.4.7.	Profile From Component.....	40
2.4.8.	Real-Time Chat Component	41
2.5.	Testing	42
2.5.1.	Testing Tools, Plans & Specifications	42
2.5.1.1.	Postman	42
2.5.1.2.	Redux Developer Tools.....	42
2.5.1.3.	React Testing Library & Jest	43
2.5.2.	Unit Testing	44
2.5.3.	Integration Testing	46

2.5.4.	Functional Testing	47
2.5.5.	End User Testing.....	49
2.5.5.1.	Five Second Test.....	49
2.5.5.2.	Think Aloud Analysis	51
2.6.	Evaluation	51
3.0	Conclusions	52
4.0	Further Development or Research	52
5.0	References	53
5.1.	Documentation	53
5.2.	Code	53
6.0	Appendices.....	54
6.1.	Project Plan.....	54
6.2.	Reflective Journals.....	62
6.3.	Testing Documentation	83

Executive Summary

Check-mate Final Solution - [<https://check-mate-journal.herokuapp.com>]

The purpose of the following report is to outline the research and development process of the check-mate web application. The subject area is mental health, in particular emotional intelligence and self-reflection. The research conducted provides support to the value and reasoning behind the development efforts.

The aim is to provide people with a modern solution that helps them get back in check with themselves and become more aware of their mental health and wellbeing. The digital journal feature developed as a result of the development, offers users a dedicated space to log and review their daily experiences. This simple tool is designed to enable users to gradually build a consistent routine of self-reflection. Communication is the other fundamental aspect of building a pro-active mentality towards mental health and this drove the requirement for the topic based chat feature. Users are provided with real-time communication in chat rooms that are dedicated to specific topics of importance in the area of mental health. This is a vital service for users who are seeking advice or looking for support in times of distress.

A potential improvement planned for the solution is to develop a counselling system where approved medical professionals can register with *check-mate* and provide counselling services on a 1-to-1 basis with users of the application. This is modern approach to counselling that makes mental health services more accessible to a wider demographic. This would also monetize the project which would provide stimulus for future development in this area.

1.0 Introduction

1.1. Background

I have been involved in awareness and fundraising events for Pieta House for a number of years, which means I have seen first-hand the effects that poor mental health is having on our society. The more vulnerable in society are normally the first to feel the effects but that does not mean that they will be the only ones. We are treating the result of the problem rather than the cause of the problem which is having little effect.

The fast paced world we live in today leaves little time for self-reflection and other mindfulness techniques. Self-reflection is a vital habit for optimizing the understanding and quality of our daily lives. It is a way of observing who we are and what makes us behave the way we do (*Benefits of Reflective Journaling, 2019*). I have used a personal journal for many years now and I genuinely could not imagine my daily routine without that few minutes of reflection each day. This is what led to the birth of the idea for the check-mate digital journaling and mental health application. One of the leading experts in this area for the past two decades, Dr James W. Pennebaker has conducted endless research in the benefits of expressive writing as a mechanism for healing. In his findings, he suggests that *“the implication of this research is that expressive writing could provide a simple and cost-effective therapeutic intervention method for general as well as clinical populations. It could also be used as an adjunct to other psychotherapy, especially for clinical populations.”* (Pennebaker & Susman, 1988)

Modern internet technologies have become ubiquitous in our lives, and instead of fighting against it, we at check-mate believe there is a great opportunity to harness these technologies in a positive way. The application will be easy to use and require no previous knowledge or experience which means it can be slotted seamlessly into most people’s daily routines. Robert Plutchik’s psychoevolutionary theory, particularly the Plutchik Wheel of Emotions, is used as a classification system to help guide user’s through their journal entries and become more conscious of their emotional triggers. Plutchik proposed that these 'basic' emotions are biologically primitive and have evolved in order to increase the reproductive fitness of the animal. Plutchik argues for the primacy of these emotions by showing each to be the trigger of behaviour with high survival value, such as the way fear inspires the fight-or-flight response. (Plutchik, R., 1990)

A major problem for people in hard times is the feeling of isolation, whether that be physical or emotional, and we hope to provide a solution to this problem. We live in a connected world so we should be able to utilize this connectivity to our benefit. The application will provide an environment free from judgement where users have the opportunity to share information and advise each other on things that may be causing them issues in their lives. According to *Sperling, M. et al “Writing has critical connections to speaking... the subvocalization of tracing your written thoughts naturally translates in actual vocalization”*. This supports the value that can be provided by the check-mate solution as it provides users with a method of positive habit-building through self-reflection and written communication.

1.2. Aims

1.2.1. Simplify Emotions for Users

The main aim of the project is to raise awareness around mental health and the importance of maintaining a healthy routine in combatting mental health issues. By providing users with a simple yet effective application for logging and acknowledging their emotional experiences, users are hoped to develop an awareness of their own mental health and the triggers in their daily lives that contribute towards those feelings. This will be facilitated by a digital journal that aims to loosely guide users through a simple self checkin system, enabling users to simplify what can appear to be complex emotional situations and become more comfortable with their feelings through understanding.

1.2.2. Provide an Opportunity to Share

Check-mate aims to provide a platform for people suffering with mental health conditions where they can find support from the *Check-mate* community and have open discussions on how to address and improve certain aspects of their lives. The community aspect of the application is massively important as it offers a communication outlet where users can have meaningful and potentially lifesaving conversations. If check-mate can make an impact in one person's life then the project will be a success.

A dedicated private chat room for different topics around mental health will be designed so that users can find more detailed information on popular topics within the checkmates community and have more specific discussions on topics that relate to their experiences. This feature gives users that opportunity to express how they are feeling and the find consolation through conversation with other *check-mate* users.

1.2.3. Empower Users

A more advanced aim of the project is providing more complex analytics capabilities to users so they can identify and address and concerning patterns or behaviours outlined by their digital journal checkin entries. This can also be used to provide monitoring support for users who vulnerable to self-harm or suicide, as even the smallest intervention could help save someone's life. The insights provided by the analytics and self-reflection approach is aimed at empowering users to take ownership of their mental health in a heuristic fashion.

1.3. Technology

1.3.1. Bootstrap, HTML & CSS

Bootstrap styling is applied in certain sections of the application along with standard HTML and CSS. The component based structure of React meant that each of the components has its own html representation that is embedded into the Root component of the application. This approach allows you to couple CSS stylesheets with specific components rather than applying styles to the entire document in one master CSS file.

1.3.2. JavaScript & JSX

JavaScript is used heavily throughout the application, especially for the server side logic. JSX is a syntax extension of traditional JavaScript that was born out of ES6 syntax. JSX is commonly used in React applications as it is more intuitive when creating and rendering components. It allows developers to produce React elements that can be easily rendered to the DOM. When the code is compiled, JSX expressions become regular JavaScript function calls and objects.

1.3.3. Node & Express

The Back End Server is built using Node.js and the Express API Framework. The reason for this choice is that the Node Package Manager (NPM) library contains a plentiful selection of open source programs that can be downloaded as add-ons to assist with the development of the backend environment. Express for example, is a package with pre-defined handlers, designed to facilitate HTTP requests in Node. These technologies are used to provide network communication between the different services and resources in the application.

1.3.4. MongoDB & Mongoose

The database storage platform being used is MongoDB, this is where application data will be persisted. MongoDB was chosen because it's a non-relational database and this suits the storage requirements for the semi-structured data in the Checkmates application. The Mongoose package is used to create the data models and constraints involved in storing and processing user data. Each resource in the application will be represented through a model in the Checkmates database. Eg. *Checkin, Profile, User*, etc.

1.3.5. React & Redux

React is a component based client side framework that is used for defining and managing application state in the browser. It adopts the single page application (SPA) approach similar to Angular. The reason for this is to prevent unnecessary communication between the server and client, by dynamically rendering components based on the client side logic. These technologies are becoming increasingly popular due to the improved response times and design approach.

Redux is used to make react asynchronous and together these technologies are implemented to facilitate the management of the application state on the client side to improve the aesthetics and user experience.

1.4. Structure

The document consists of several sections, that provide a breakdown of the specification of the application requirements, the background and the aims of the project, the design and architecture strategy and the technical implementation. This will be rounded off with the evaluation of the development project and any conclusions.

System

- **Functional Requirements**
This section outlines what the application must do to achieve desired functionality.
- **Non-functional Requirements**
This section outlines how the desired functionality will be achieved.
- **Architecture & Design**
This section logically defines the Architecture & Design of the project, through diagram and description.
- **Implementation**
This section provides a description of the most significant code implementation techniques for the front and back end.
- **Graphical User Interface**
This section provide provides images and descriptions of the main GUI components of the application.
- **Testing & Evaluation**
This section offers detailed testing procedures and results, along with an overall evaluation.

Conclusions

This section provides the conclusions and findings produced as a result of the project

Further Development & Research

The document closes with potential further development & research ideas as a result of the project.

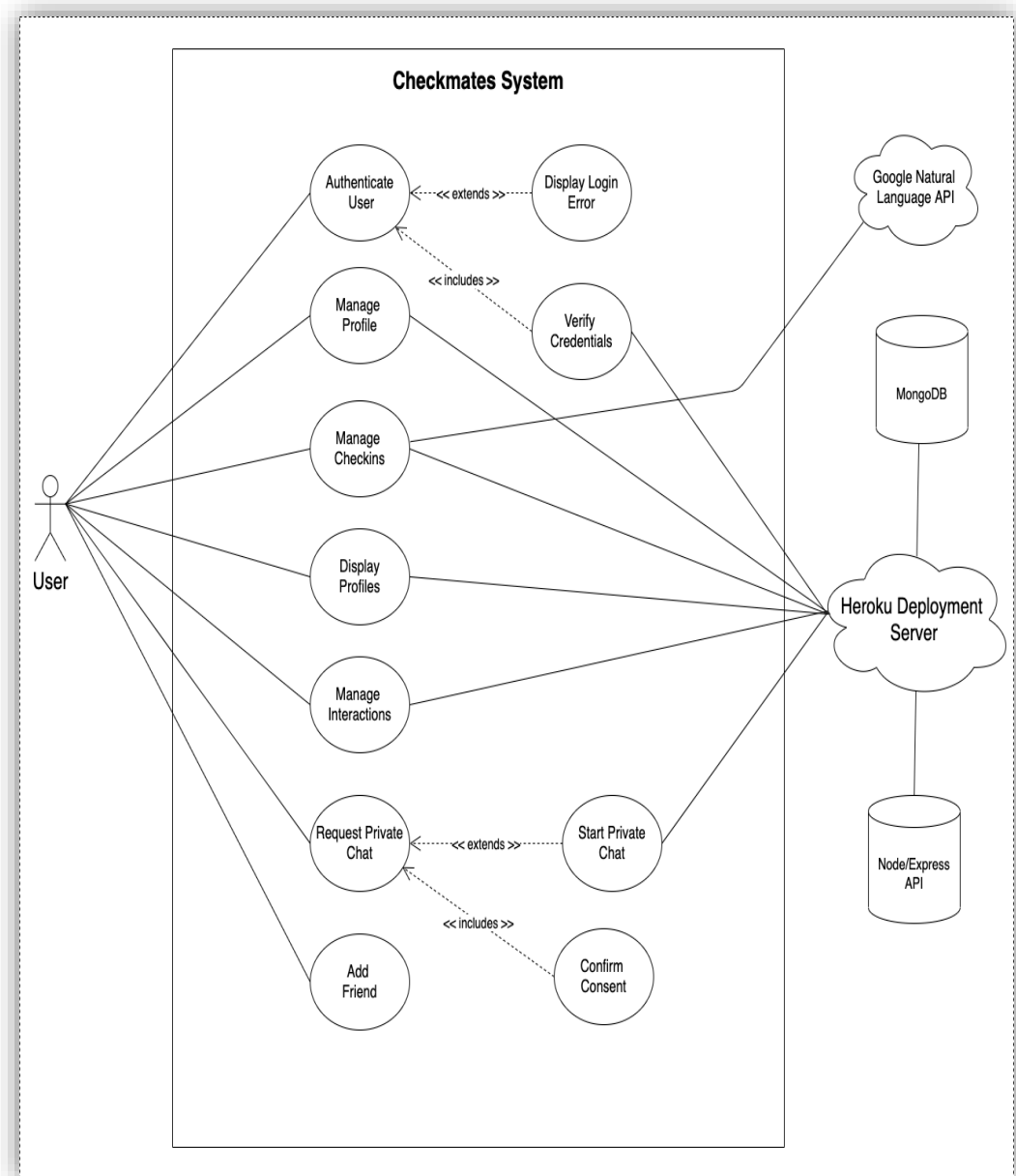
2.0 System

2.1. Requirements

2.1.1. Functional Requirements

Functional requirements describe the possible effects of a software system, in other words, *what* the system must accomplish.

2.1.1.1. Use Case Diagram



2.1.1.2. Requirement 1: Register & Authenticate Users

2.1.1.2.1. Description & Priority

This requirement provides a user the ability to create and register a new account to be able to take advantage of full features and functionality. This is a fundamental aspect of the development as other requirements will be dependent on this feature.

2.1.1.2.2. Use Case: R01

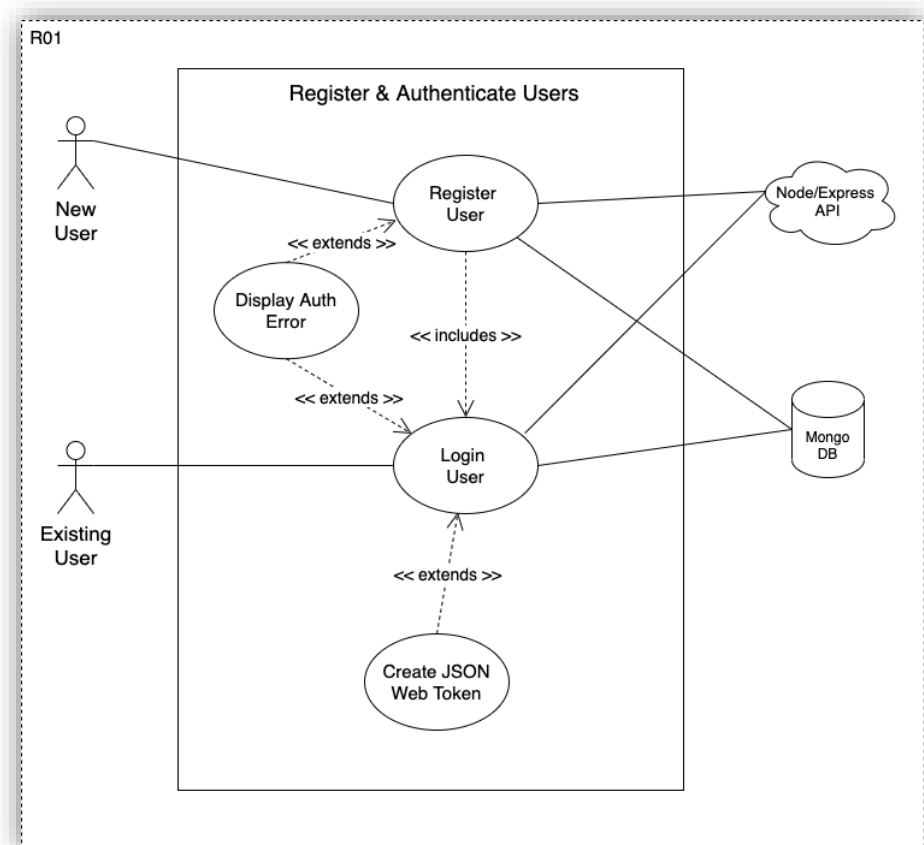
Scope

The scope of this use case is to allow existing users to login to the system securely and to allow new users to register their account information securely.

Description

This use case describes the actors involved in registering/authenticating a new/existing user in the system.

Use Case Diagram



Flow Description

Precondition

The system is in initialisation mode and the user has a stable connection.

Activation

This use case starts when a User selects the option to login

Main flow

1. The system identifies the user is not logged in and directs them to the login page.
2. The User does not have an existing account and selects to register (see A1).
3. The User enters their credentials and attempts login
4. The User enters invalid login credentials (See E1)
5. The User credentials are valid, and the system authenticates the user
6. The User is redirected to the dashboard page.

Alternate flow

A1: User is required to create account

1. The system prompts the user to create and register an account
2. The User creates and submits account details that are validated and saved by the system
3. The use case continues at position 6 of the main flow

Exceptional flow

E1: User re-attempts login

1. The system checks the new credentials entered
2. The User is authenticated
3. The use case continues at position 5 of the main flow

Termination

The system presents the user's dashboard

Post condition

The system goes into a wait state

2.1.1.3. Requirement 2: Create & Manage User Profile

2.1.1.3.1. Description & Priority

This requirement provides authenticated users with the ability to create and manage their personal Checkmates Profile. This is a key part of the application as it provides information that is used throughout the system.

2.1.1.3.2. Use Case: R02

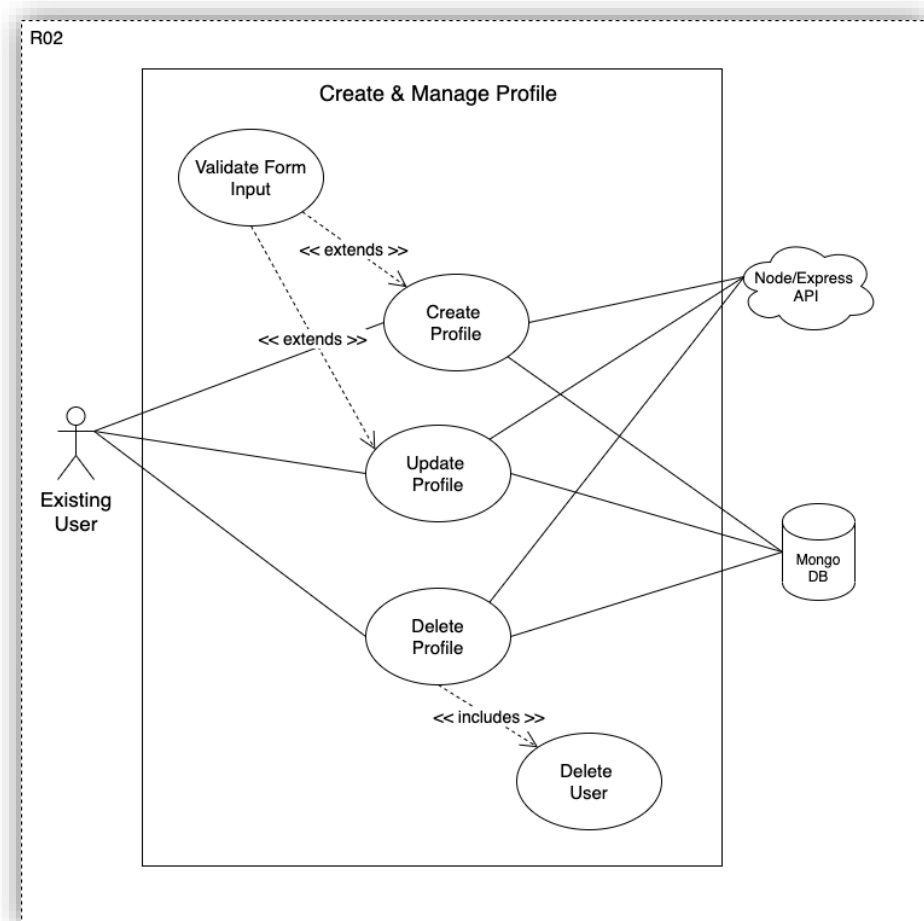
Scope

The scope of this use case is to allow users to create a personal profile that is stored securely within the system.

Description

This use case describes the actors involved in creating and managing a Checkmates profile within the system

Use Case Diagram



Flow Description

Precondition

The system is in wait mode and a user is authenticated.

Activation

This use case starts when a User visits the Checkmates dashboard.

Main flow

1. The system identifies the user does not have an existing profile
2. The <User> has an existing profile (See A1)
3. The system prompts the <User> to create a profile
4. The <User> enters the required information.
5. The system validates the information.
6. The information provided is invalid (See E1)
7. The profile information is saved

Alternate flow

A1: User has an existing profile

1. The system displays profile information.
2. The User selects edit profile.
3. The use case continues at position 4 of the main flow.

Exceptional flow

E1: The information provided is invalid

1. The system notifies the user of the errors.
2. The User corrects and re-submits.
3. The use case continues at position 5 of the main flow

Termination

The system redirects the user to the dashboard page.

Post condition

The system goes into a wait state

2.1.1.4. Requirement 3: Create & Manage Checkins

2.1.1.4.1. Description & Priority

This requirement provides authenticated users with the ability to create and manage the checkins in their digital journal. This is the main functional requirements of the app as this is where user data will be handled.

2.1.1.4.2. Use Case: R03

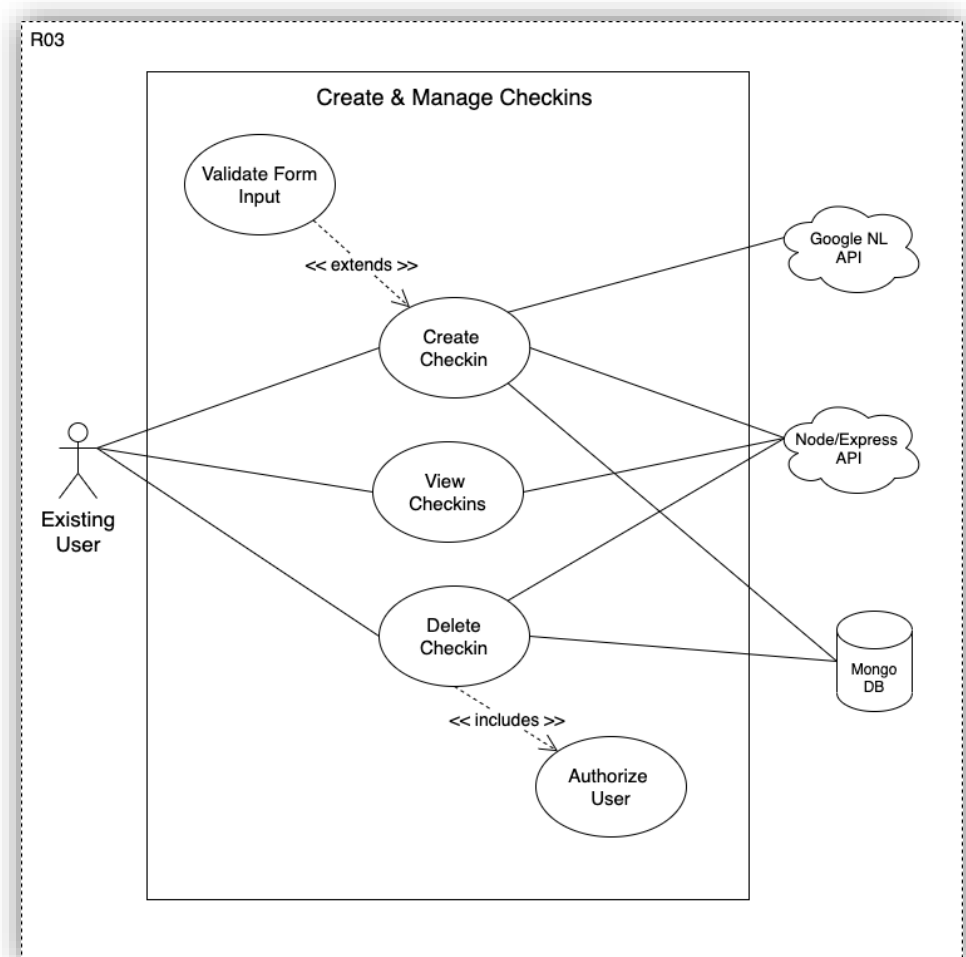
Scope

The scope of this use case is to allow users to create a checkin entry.

Description

This use case describes the primary and secondary actors involved in creating and storing user checkins.

Use Case Diagram



Flow Description

Precondition

The system is in wait mode and a user is authenticated

Activation

This use case starts when a User selects the option to create a new checkin on the checkin dashboard.

Main flow

1. The system presents the user with the checkin form and a list of their previous personal checkins.
2. The User chooses to review previous checkins (See A1).
3. The User enters the required information to create a checkin.
4. The system validates the information provided.
5. The information provided is invalid (See E1).
6. The checkin information is saved
7. The updated checkin list information is presented to the user by the system.

Alternate flow

A1: Review previous checkins

1. The User selects to delete a previous checkin.
2. The system confirms authorization and deletes the resource.
3. The use case continues at position 7 of the main flow.

Exceptional flow

E1: The information provided is invalid

1. The system notifies the user of the errors.
2. The User corrects and re-submits.
3. The use case continues at position 4 of the main flow.

Termination

The system presents the checkin dashboard to the User with updated information.

Post condition

The system goes into a wait state.

2.1.1.5. Requirement 4: Display User Profiles

2.1.1.5.1. Description & Priority

This requirement provides authenticated users with the ability to browse and connect with other checkmates users. This provides a social interaction aspect to create a sense community within the application

2.1.1.5.2. Use Case: R04

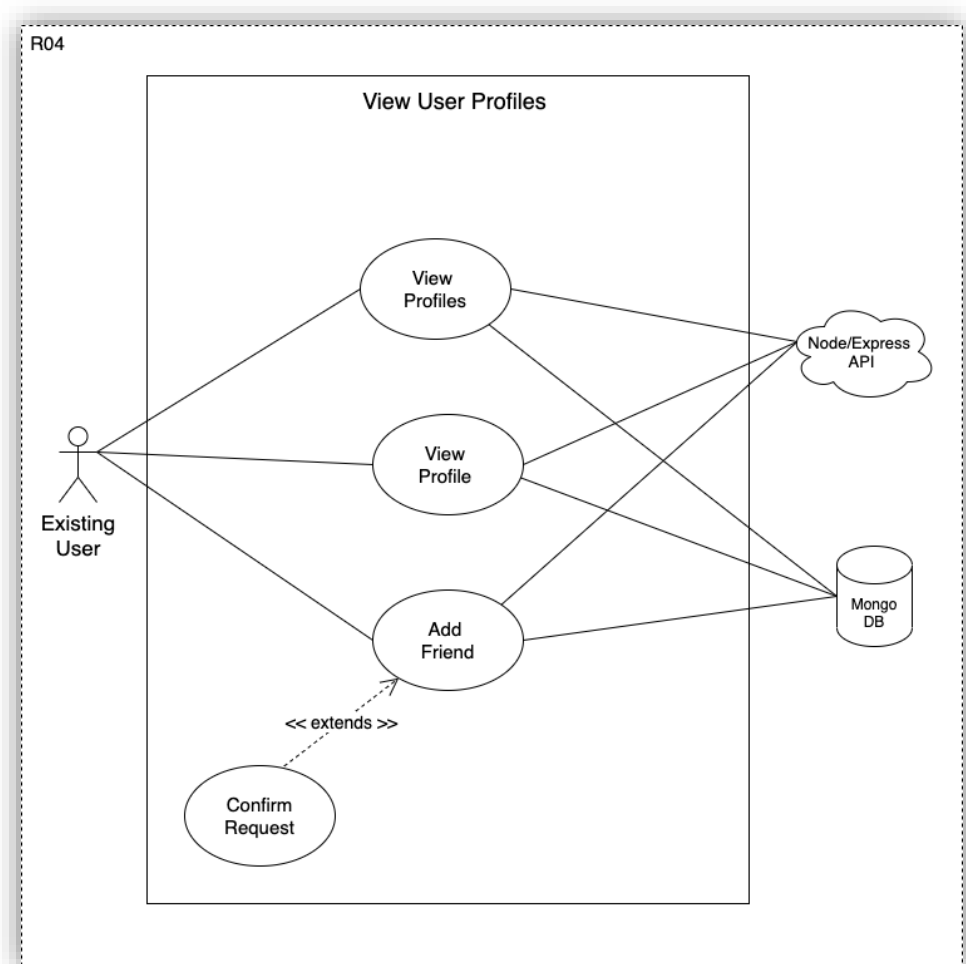
Scope

The scope of this use case is to allow authenticated users to view other user's public profile information and connect with like-minded users.

Description

This use case describes the actors involved in displaying public checkmates profiles to the User.

Use Case Diagram



Flow Description

Precondition

The system is in wait mode and a user is authenticated.

Activation

This use case starts when a User visits the profiles page.

Main flow

1. The system presents a list of public user profiles.
2. The User selects to view an individual Profile
3. The system presents the individual profile information to the User
4. The User chooses to add the Profile to their friends list (See A1)
5. The system redirects the user to the profiles page.

Alternate flow

A1: User adds Profile to friends list

1. The system sends the request to the selected Profile user
2. The User approves or denies the requested connection

Termination

The system redirects the User to the profiles display page.

Post condition

The system goes into a wait state.

2.1.1.6. Requirement 5: Create & Manage Interactions

2.1.1.6.1. Description & Priority

This requirement provides authenticated users with the ability to add support reaction and comments to other users public journal checkin entries. This feature provides a mechanism for discussion and communication on important topics.

2.1.1.6.2. Use Case: R05

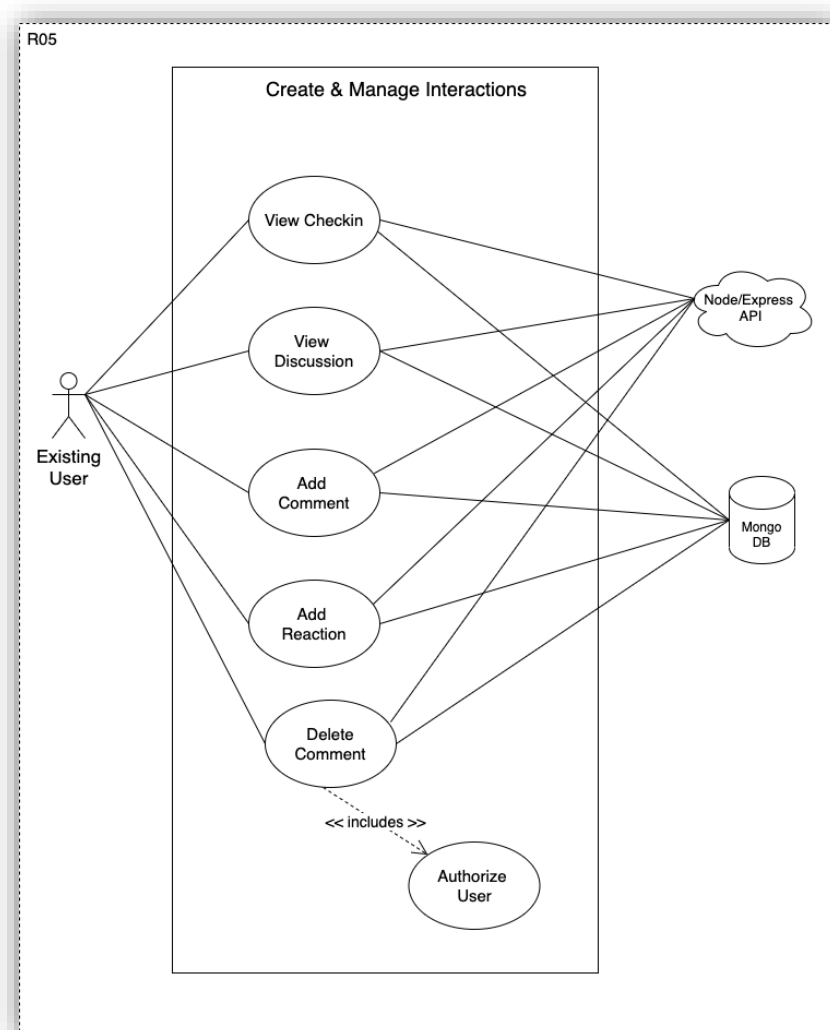
Scope

The scope of this use case is to allow authenticated users to interact with other user's by adding reactions and comments to public journal checkin entries.

Description

This use case describes the actors involved in creating, managing, and displaying public checkin entry comments and reactions.

Use Case Diagram



Flow Description

Precondition

The system is in wait mode and a user is authenticated.

Activation

This use case starts when a User visits the public checkins page.

Main flow

1. The system presents a list of public user checkins to the User.
2. The User selects to add a support reaction to an individual checkin entry.
3. The system updates the checkin entry.
4. The User chooses to view the discussion on an individual checkin (See A1).
5. The User leaves the public checkins page.

Alternate flow

A1: User views checkin discussion

1. The system presents the user with the checkin and all current comments in the discussion.
2. The User adds a comment to the checkin.
3. The system saves the information and updates the comments list.

Termination

The system redirects the User to the desired page.

Post condition

The system goes into a wait state.

2.1.1.7. Requirement 6: Create & Manage Real-Time Chat

2.1.1.7.1. Description & Priority

This requirement provides authenticated users with the ability to create and manage real-time chat room with other authenticated users in the system. This is a lower priority feature intended for future development.

2.1.1.7.2. Use Case: R06

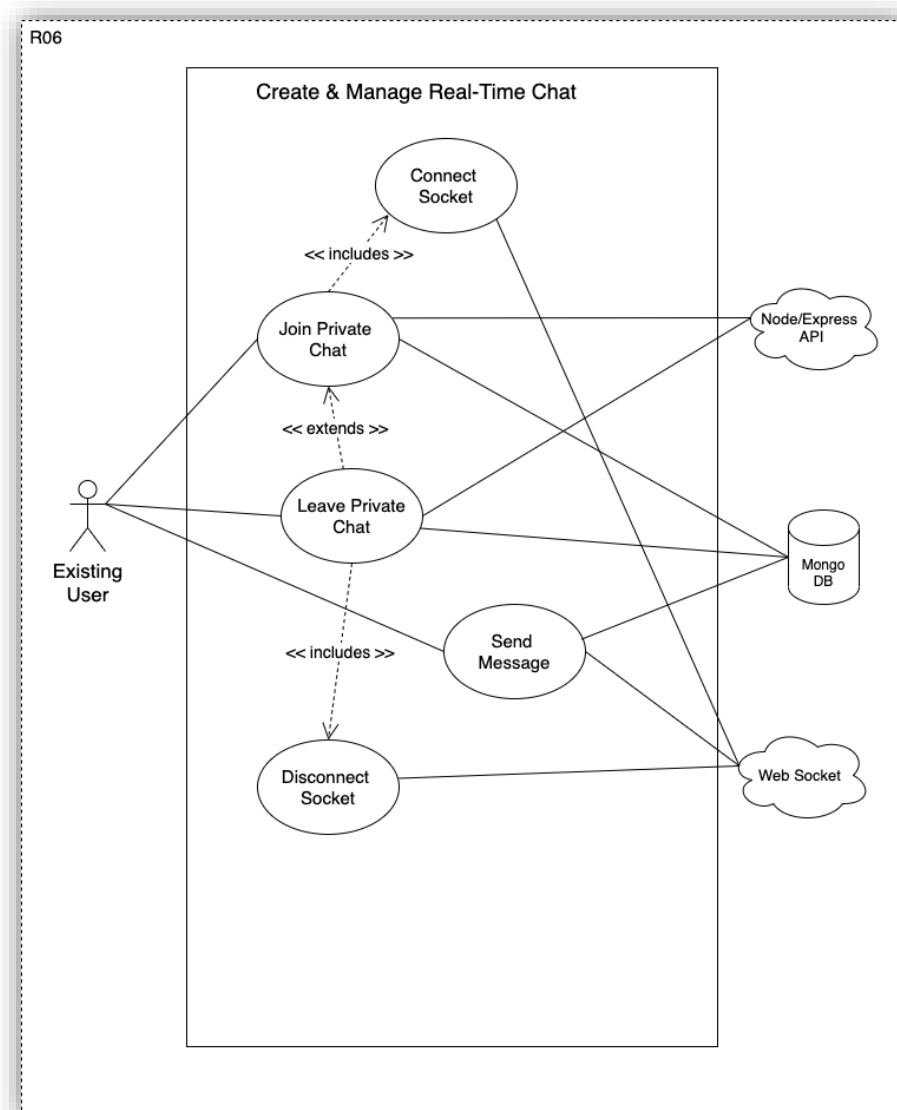
Scope

The scope of this use case is to allow authenticated users to communicate and interact with other user's by joining private chat rooms.

Description

This use case describes the actors involved in creating, managing, and facilitating real time chat communication through web sockets.

Use Case Diagram



Flow Description

Precondition

The system is in wait mode and a user is authenticated.

Activation

This use case starts when a User selects the private chat option on a checkin entry.

Main flow

1. The system initializes web socket connection.
2. Web socket connection fails (See E1)
3. The system presents the private chat room page to the User
4. The secondary User is prompted to accept or deny the chat request.
5. The secondary User denies the request. (See A1)
6. The primary and secondary Users add messages to the chat.
7. The system presents the updated messages the Users in the private chat room.
8. The Users leaves the private chat.
9. The system disconnects web socket connection

Alternate flow

A1: Chat request denied

1. The system notifies the primary user
2. The system redirects the primary User back to the previous page
3. The use case continues at position 9 of the main flow

Exceptional flow

E1: Web socket connection fails

1. The system notifies the user of the errors.
2. The system tries to re-connect
3. The use case continues at position 3 of the main flow.

Termination

The system redirects the User to the dashboard page.

Post condition

The system goes into a wait state.

2.1.2. Data Requirements

The checkmates system was designed to be able to manage and process large amounts of semi- structured data. This includes raw user data input into the application and potentially digital media uploaded by different users. MongoDB is used to manage the storage of user data in the Checkmates application and the mongoose library is incorporated to create the data models.

The programming languages that are used are a combination of JavaScript, JSX, HTML & CSS together with the front end frameworks of React and Redux. These languages are used to process, manipulate, analyze and display the data being managed in the application. To allow the use of third party services such as the Google Natural Language API, standard JavaScript Object Notation is the chosen approach for data representation and transportation. The data is retrieved from Mongo DB Atlas via the Node/Express API using HTTP protocol methods.

2.1.3. User Requirements

2.1.3.1. Confidentiality

User data being entered in the application can be highly sensitive and private to the user. Users will be offered with privacy filters for all of their Journal entries in the application that will ensure that only they can view more sensitive information. The user must feel comfortable when using the application and can share as much or as little to their discretion.

2.1.3.2. Security

Application security is a fundamental requirement for users. Users details will be handled with extreme caution to prevent any cybersecurity issues. It is imperative that no user credentials are compromised and any stage in the application's lifecycle.

2.1.3.3. Data Protection - GDPR

GDPR is a regulatory compliance requirement that must be adhered to by all companies handling user data. Checkm8 is no different and will provide users with access to their data whenever requested. Users will also have the option to update and remove any data that they do not consent to being stored in the application.

2.1.4. Environmental Requirements

2.1.4.1. Connectivity

The environment must have any active connection to the both MongoDB and the Google Natural Language API. These credentials were configured as environment variables in the production environment. A stable connection to the Heroku Server is also a vital part of the connectivity requirement as this is where the application code is hosted.

2.1.4.2. Compute

The cloud platform chosen for Platform as a Service (PaaS) hosting is Heroku. The Heroku deployment is hosted by Amazon and requires a single Virtual Machine (VM) to run the application. The free tier of Heroku's available services currently covers all of the compute and connectivity requirements of the application. There is scope to scale this depending on the future direction of the development efforts.

2.1.5. Usability Requirements

2.1.5.1. Accessibility

Check-mate is designed to be absolutely inclusive so accessibility is an important requirement when design the UI. We will be taking the 'design to the edges' approach which takes into account the full range of human characteristics, not just the average person.

2.1.5.2. Learnability

The application should be intuitive and easy for users to learn and progress quickly. The heuristic nature of the application is designed to facilitate a journey of self-discovery as users begin to consciously log their emotions. Users must be provided with an interface that enhances their experience, not one that distracts or inhibits it. The concepts and features of the application will be laid out clearly and concisely from the beginning of the users' experience and will be re-enforced throughout.

2.1.5.3. Navigation

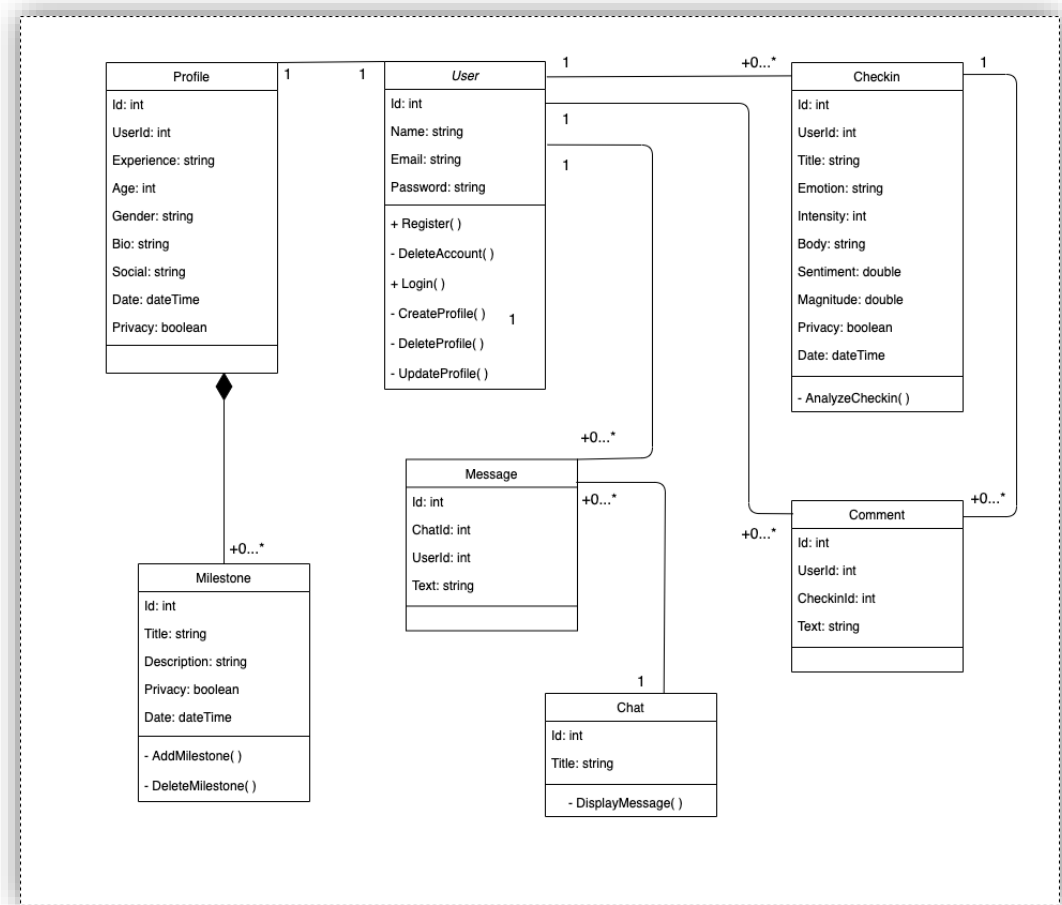
Navigation throughout the website should be seamless and clearly defined for the user. This will be achieved through consistent menus and navigation options. The application loads extremely fast on the front-end due to the React components which improves the navigability of the site overall.

2.1.5.4. User Friendliness

The application must be both visually and practically appealing to the user. This will be provided by a warm, welcoming and carefully designed UI. The design patterns will be minimalistic to avoid any clutter or confusion for user's while using the app.

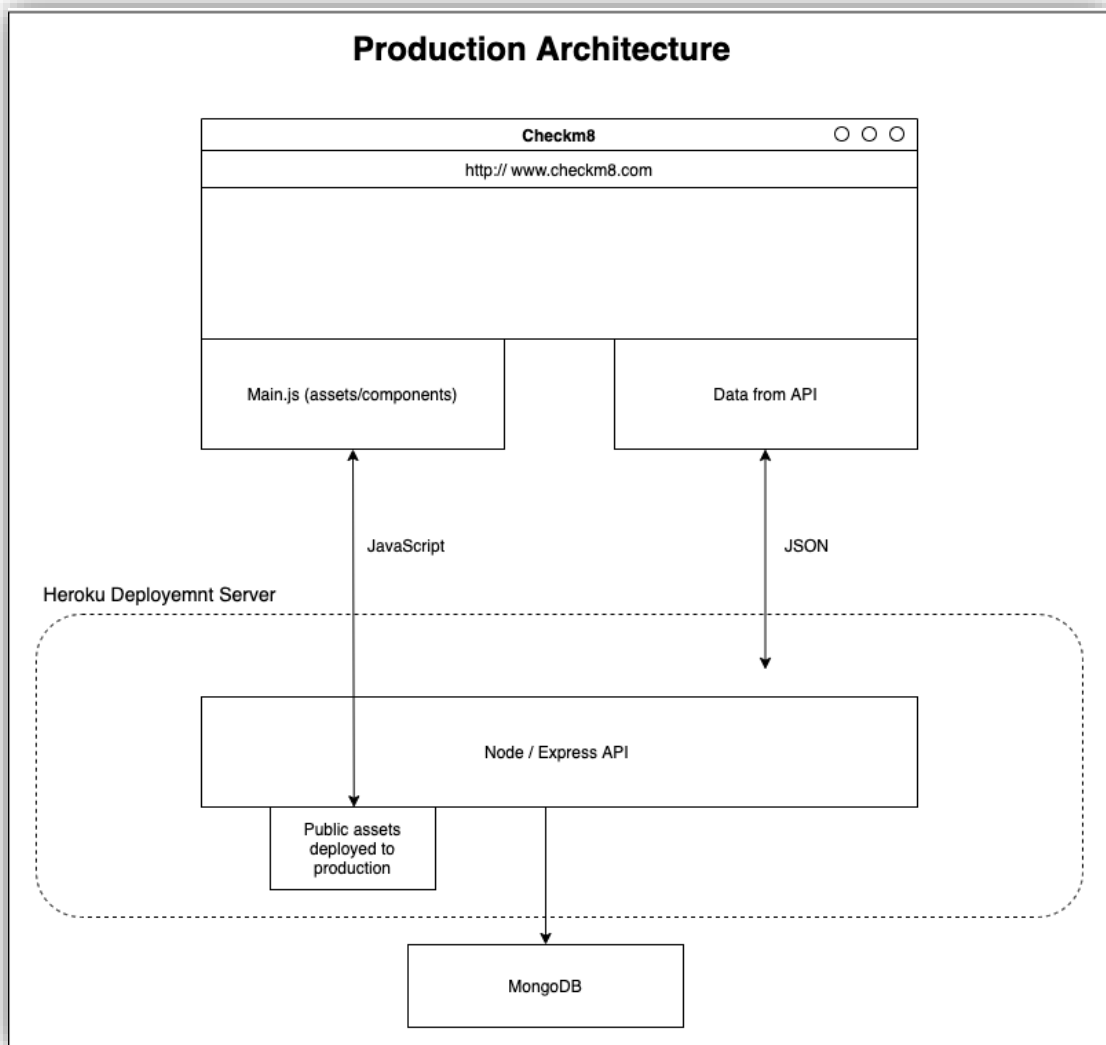
2.2. Design & Architecture

2.2.1. Class Diagram



The class diagram above represents the different classes that the Checkmates system is comprised of. Each class defines the key attributes of the class along with any methods that they use to interact with the system and other classes. The basic relationships are defined along with their relative cardinality constraints. This diagram was used to logically define the application architecture before the development process had begun.

2.2.2. System Architecture Diagram



Before deploying to production, the node command `npm run build` is executed which compiles the project files and current application assets to prepare them for th production configuration. This removes the need for the front end server that can be seen in the development architecture diagram. As the deployed application will contain all of the required files to run the application from the Heroku deployment.

2.3. Implementation

2.3.1. Back-end

[User registration – bcrypt password hashing and JSON web token creation]

Location: /routes/api/users.js

```
// > ROUTE      POST api/users
// > DESC       Registration route for user
// > PERMISSION Public
router.post('/', [
  // implementing express router validation constraints on required attributes
  // Req_Id: R02
  // Test_Id: T004
  check('name', 'Name is a required field')
    .not()
    .isEmpty(),
  check('email', 'Please enter a valid email')
    .isEmail(),
  check(
    'password',
    'Please Enter a password with a minimum of 8 characters'
  ).isLength({ min: 8 })
],
async (req, res) => {
  const errors = validationResult(req);
  if(!errors.isEmpty()) {
    return res.status(400).json({ errors: errors.array() });
  }
  // extracting fields from the body
  const { name, email, password } = req.body;

  try {
    // check if user already exists
    let user = await User.findOne({ email });
    if(user) {
      return res.status(400).json({ errors: [{ msg: 'User already exists' }] });
    }
    // Initializing gravatar module and defining size, rating and default image if user does not have a gravatar
    const avatar = gravatar.url(email, {
      s: "200",
      r: 'pg',
      d: 'mm'
    })
    // create user
    user = new User({
      name,
      email,
      avatar,
      password
    });
    // encrypt password
    // generates the salt to be used for hashing – higher number is greater security but longer loading
    // Test_Id: T005
    const salt = await bcrypt.genSalt(10);

    user.password = await bcrypt.hash(password, salt);

    await user.save();
    // return JWT for immediate login
    // REFERENCE https://www.npmjs.com/package/jsonwebtoken
    // Req_Id: R02
    // Test_Id: T006
    const payload = {
      user: {
        id: user.id
      }
    }
    jwt.sign(
      payload,
      keys.jwtSecret,
      { expiresIn: 3600 },
      (error, token) => {
        if(error) throw error;
        res.json({ token });
      }
    );
  } catch(error) {
    console.log(error.message);
    res.status(500).send('Internal Sever error');
  }
}
);
```

[Login user – bcrypt password hashing and JSON web token creation on success]

Location: /routes/api/auth.js

```
router.post('/', [
  // Req_Id: R01
  // Test_Id: T007
  check('email', 'Please enter a valid email').isEmail(),
  check('password', 'Password is a required field').exists()
],
async (req, res) => {
  const errors = validationResult(req);
  if(!errors.isEmpty()) {
    return res.status(400).json({ errors: errors.array() });
  }
  // extracting fields from the request body
  const { email, password } = req.body;

  try {
    // check if user email exists
    let user = await User.findOne({ email });
    if(!user) {
      return res.status(400).json({ errors: [{ msg: 'Invalid credentials provided' }] });
    }
    // check if the password provided is valid for the user email
    const isValid = await bcrypt.compare(password, user.password);

    if(!isValid) {
      return res.status(400).json({ errors: [{ msg: 'Invalid credentials provided' }] });
    }

    const payload = {
      user: {
        id: user.id
      }
    };
    jwt.sign(
      payload,
      keys.jwtSecret,
      { expiresIn: 360000 },
      (error, token) => {
        if(error) throw error;
        res.json({ token });
      }
    );
  } catch(error) {
    console.log(error.message);
    res.status(500).send('Sever error');
  }
});
```

[Deleting Profile and dependant Checkin and User resources.]

Location: /routes/api/profile.js

```
// ROUTE      DELETE api/profile
// DESC      Route delete user profile and dependant resources (checkins, user)
// PERMISSION Private
// Req_Id:    R02
// Test_Id:   T012
router.delete('/', auth, async (req, res) => {
  try {
    // Remove user checkins on account deletion
    await Checkin.deleteMany({ user: req.user.id });
    // finding the user profile and removing
    await Profile.findOneAndRemove({ user: req.user.id });
    // Remove the user account
    await User.findOneAndRemove({ _id: req.user.id });

    res.json({ msg: 'User account deleted' });
  } catch (error) {
    console.error(error.message);
    res.status(500).send('Internal Server Error');
  }
});
```

[Middleware function used to authenticate user requests to private routes]

Location: /middle/auth.js

```
module.exports = function(req, res, next) {
  // extract token from the request header
  const token = req.header('x-auth-token');
  // check if no token is present
  if(!token) {
    return res.status(401).json({ msg: 'No token present, authorization unsuccessful' });
  }
  // verify token
  // Req_Id: R02
  // Test_Id: T006
  try {
    const extracted = jwt.verify(token, config.get('jwtSecret'));
    req.user = extracted.user;
    next();
  } catch(err) {
    res.status(401).json({ msg: 'Token is invlaid' });
  }
}
```

This middleware function simply checks for the x-auth-token header in any incoming requests.

[Google Natural Language API Call on Checkin Creation]

Location: /routes/api/checkins.js

```
// create the checkin from the user model and the request body
const document = {
  content: req.body.maintext,
  type: 'PLAIN_TEXT'
}

// REFERENCE: https://googleapis.dev/nodejs/language/latest/
// Google Natural Language API call
const [result] = await client.analyzeSentiment({ document: document });
const sentimentRes = result.documentSentiment;
// ReqId: R0
// TestId: T055
console.log('Text: ${req.body.maintext}');
console.log('Sentiment score: ${sentimentRes.score}');
console.log('Sentiment magnitude: ${sentimentRes.magnitude}');

sent = sentimentRes.score.toFixed(3)
mag = sentimentRes.magnitude.toFixed(3)

const newCheckin = new Checkin({
  user: req.user.id,
  title: req.body.title,
  emotion: req.body.emotion,
  intensity: req.body.intensity,
  maintext: req.body.maintext,
  sentiment: sent,
  magnitude: mag,
  privacy: req.body.privacy,
  name: user.name,
  avatar: user.avatar
});
// save the new checkin
const checkin = await newCheckin.save();
```

The Google Natural Language API requires a service account with API credentials for in app authentication. This account was created in the Google Developers console and the JSON credentials file was downloaded to the local machine and exported to the working directory file path.

These credentials were also configured separately in the Heroku environment, as environment variables

[Getting all public user checkins and getting all user's personal checkins]

Location: /routes/api/checkins.js

```
// ROUTE      GET api/checkins
// DESC      Route to get user checkins
// PERMISSION Private
// Req_Id:   R03 - Create & Manage Checkins
// Test_Id:  T016
router.get('/', auth, async (req, res) => {
  try {
    // getting all checkins and sorting them by most recent date
    //const checkins = await Checkin.find().sort({ date: -1 });
    const checkins = await Checkin.find({ 'privacy': false }).sort({ date: -1 });
    res.json(checkins);
  } catch (error) {
    console.error(error.message);
    res.status(500).send('Internal Server Error');
  }
});

// ROUTE      GET api/checkins/user
// DESC      Route to get users personal checkins
// PERMISSION Private
// Req_Id:   R03 - Create & Manage Checkins
// Test_Id:  T059
router.get('/user', auth, async (req, res) => {
  try {
    // getting all user checkins and sorting them by most recent date
    const checkins = await Checkin.find({ user: req.user.id }).sort({ date: -1 });
    res.json(checkins);
  } catch (error) {
    console.error(error.message);
    res.status(500).send('Internal Server Error');
  }
});
});
```

The private checkins are being filtered using the privacy Boolean in the Checkin model. The personal checkin are being filtered based of the user id of request submitter.

[File that sets the application keys depending on the environment]

Location: /config/key.js

```
// keys.js - deciding what set of credentials to use PROD or DEV
if (process.env.NODE_ENV === 'production'){
  // heroku - PROD
  module.exports = require('./production');
}
else{
  // local machine - DEV
  module.exports = require('./dev');
}
```

The key.js file is used to determine which set off application keys are used depending on the detected Node environment. The dev.js file has hard coded credentials whereas the production.js file contains references to the environment variables configured in Heroku.

```
// ----- PROD KEYS -----
// uses the environmet variables declared in the remote Heroku environment
module.exports = {
  mongoURI: process.env.MONGO_URI,
  jwtSecret: process.env.JWT_SECRET
};
```

2.3.2. Front-end

[JSON Web Token - Helper Method to add json web token to the request headers]

Location: /cm-client/src/helpers/setAuthToken.js

```
// helper function that receives the token if present and adds it to the request headers
import axios from 'axios';

const setAuthToken = token => {
  // if there is a token in local storage, add it to global headers as the x-auth-token
  if(token) {
    axios.defaults.headers.common['x-auth-token'] = token;
  }
  else {
    // otherwise remove the x-auth-token from the global headers
    delete axios.defaults.headers.common['x-auth-token'];
  }
};

export default setAuthToken;
```

The setAuthToken() helper method was implemented in the React application to set the global headers for the axios requests that are being called from the front end actions to the Node API endpoints on the server.

[Redux store creation with Thunk middleware]

Location: /cm-client/src/store.js.js

```
import { createStore, applyMiddleware } from 'redux';
import { composeWithDevTools } from 'redux-devtools-extension';
import thunk from 'redux-thunk';
import rootReducer from './reducers';

//initial state
const initState = {};
// thunk middleware
const middle = [thunk];
// creating the redux store
const store = createStore(rootReducer, initState, composeWithDevTools(applyMiddleware(...middle)));

export default store;
```

Usually RESTful services using client and server model communication are stateless. Redux library allows developers to create a global store that is maintained and can be accessed throughout the client side of the application. This store is used to manage the application state on the client side through synchronous actions, removing the need for unnecessary calls to the backend for data that is already present. To create the store, the root application reducer is added along with the empty initial application state and a Thunk middleware object.

The Thunk middleware extends the stores capabilities, allowing asynchronous logic that can interact with and update the store. “thunks are a functional programming technique used to delay computation” (Lebec, 2017)

[Checkins action reducer – managing and updating application state]

Location: /cm-client/src/actions/checkin.js

```
// action to get checkins
export const getCheckins = () => async dispatch => {
  try {
    // getting all checkins from backend api
    const res = await axios.get('/api/checkins')
    // dispatching the checkins reducer action
    dispatch({
      type: GET_CHECKINS,
      payload: res.data
    })
  } catch (error) {
    dispatch({
      type: CHECKIN_ERROR,
      payload: { msg: error.response.statusText, status: error.response.status }
    })
  }
}

// action to get all personal checkins
export const getPersonalCheckins = [] => async dispatch => {
  try {
    // getting all user's checkins from backend api
    const res = await axios.get('/api/checkins/user')
    // dispatching the checkins reducer action
    dispatch({
      type: GET_CHECKINS,
      payload: res.data
    })
  } catch (error) {
    dispatch({
      type: CHECKIN_ERROR,
      payload: { msg: error.response.statusText, status: error.response.status }
    })
  }
}

// action to add support reaction to checkin
export const addReaction = checkinId => async dispatch => {
  try {
    // getting all checking from backend api
    const res = await axios.put(`/api/checkins/reaction/${checkinId}`)
    // dispatching the checkins reducer action
    dispatch({
      type: ADD_REACTION,
      payload: { checkinId, reactions: res.data }
    })
  } catch (error) {
    dispatch({
      type: CHECKIN_ERROR,
      payload: { msg: error.response.statusText, status: error.response.status }
    })
  }
}

// action to delete a user checkin
export const deleteCheckin = checkinId => async dispatch => {
  try {
    await axios.delete(`/api/checkins/${checkinId}`)
    // dispatching the delete checkin reducer action
    dispatch({
      type: DELETE_CHECKIN,
      payload: checkinId
    })
    // notifying the user through the set alert action
    dispatch(setAlert('Checkin has been successfully deleted', 'success'))
  } catch (error) {
    dispatch({
      type: CHECKIN_ERROR,
      payload: { msg: error.response.statusText, status: error.response.status }
    })
  }
}
```

The actions directory contains all of the front end actions that communicate with the backend. Axios is a node module that is used to create the requests to different API endpoints. The dispatch action is the main method that is provided by Redux. This allows specific action types to be dispatched to the relevant reducer with payload data.

Location: /cm-client/src/reducers/checkin.js

```
export default function checkin(state = initState, action) {
  switch(action.type) {
    case GET_CHECKINS:
      return {
        ...state,
        checkins: action.payload,
        loading: false
      }
    case CREATE_CHECKIN:
      // adding the newly created checkin to the existing checkins array
      return {
        ...state,
        checkins: [action.payload, ...state.checkins],
        loading: false
      }
    case GET_CHECKIN:
      return {
        ...state,
        checkin: action.payload,
        loading: false
      }
    case DELETE_CHECKIN:
      return {
        ...state,
        // returning all the checkin except for the one that has been deleted
        checkins: state.checkins.filter(checkin => checkin._id !== action.payload),
        loading: false
      }
    case CHECKIN_ERROR:
      return {
        ...state,
        error: action.payload,
        loading: false,
        checkin: null
      }
    case ADD_COMMENT:
      return {
        ...state,
        checkin: { ...state.checkin, comments: action.payload },
        loading: false
      }
    case DELETE_COMMENT:
      return {
        ...state,
        // filtering out the comment that matches the id
        checkin: { ...state.checkin, comments: state.checkin.comments.filter(comment => comment._id !== action.payload) },
        loading: false
      }
    case ADD_REACTION:
      return {
        ...state,
        // mapping through the checkins to see if it matches the payload id, return the state with reactions
        checkins: state.checkins.map(checkin => checkin._id === action.payload.checkinId ? {
          ...checkin, reactions: action.payload.reactions
        } : checkin),
        loading: false
      }
    default:
      return state
  }
}
```

The reducer function receives the state and the action that has been dispatched. Depending on the action type the reducer will manipulate, update, and return the new state with the payload data that has been received from the action.

[Chat Socket connection using Socket IO module]

Location: /cm-client/src/components/chats/Chats.js

```
let CONNECTION_PORT
if(process.env.NODE_ENV === 'production') {
  CONNECTION_PORT = window.location.hostname
}
else {
  CONNECTION_PORT = 'http://localhost:5000'
}

let socket

const Chats = ({ auth: {user} }) => {

  const [room, setRoom] = useState('')
  const [userName, setUserName] = useState('')
  const [inRoom, setInRoom] = useState(false)

  const [message, setMessage] = useState('')
  const [messageList, setMessageList] = useState([])

  useEffect(() => {
    socket = io(CONNECTION_PORT)
  }, [CONNECTION_PORT])

  useEffect(() => {
    socket.on('receiveMessage', (data) => {
      console.log(data)
      setMessageList([...messageList, data])
    })
  })

  const connectToRoom = () => {
    socket.emit('joinRoom', ({ userName, room }))
    setInRoom(true)
  }

  const leaveRoom = () => {
    socket.emit('leaveRoom', ({ userName, room }))
    setInRoom(false)
  }

  const sendMessage = async () => {

    let chatMessage = {
      room: room,
      content: {
        userName: userName,
        message: message
      }
    }

    await socket.emit('sendMessage', chatMessage )
    setMessageList([...messageList, chatMessage.content])
    setMessage('')
  }
}
```

The socket io connection was set up on the front end using Socket io client node module and a useEffect hook that allows you to set and manipulate the state before the component renders. The connection port is defined dynamically depending on the status of the node environment.

Regular JSX functions were used to trigger socket.emit functions that facilitate the transfer of message between connected sockets. This data is sent to the server and handled depending on the message type that has been sent. The messages are processed and delivered to any relevant sockets that should receive the message, in real time.

```
// socket io connection
// ReqId: R0
// TestId: T058
io.on('connection', (socket) => {
  console.log('Connected: ${socket.id}')
  socket.emit('message', 'Welcome to Checkmates Chat')
  // broadcast when a user joins
  socket.broadcast.emit('message', 'A user has joined the chat')

  socket.on('disconnect', () => {
    console.log('User Disconnected: ')
    io.emit('message', 'A user has left the chat')
  })

  socket.on('joinRoom', ({userName, room}) => {
    socket.join(room)
    console.log('User ${userName} joined chat => ${room}')
  })

  socket.on('leaveRoom', ({userName, room}) => {
    socket.leave(room)
    console.log(`${userName} has left chat room ${room}`)
  })

  socket.on('sendMessage', ({room, content}) => {
    socket.to(room).emit('receiveMessage', content)
    console.log('Message received from ${content.userName}: \n${content.message}')
  })
})
```

2.4. Graphical User Interface (GUI)

2.4.1. Home Component



The home component is the only public page accessible without registering for an account. This section aims to provide users with some introductory information to the subject-matter, concept and function of the website.

Visual media is always more captivating, so that's why some relative videos have been embedded in the home page for user's to build up some understanding of the importance of emotional health & intelligence before they are required to submit any information.

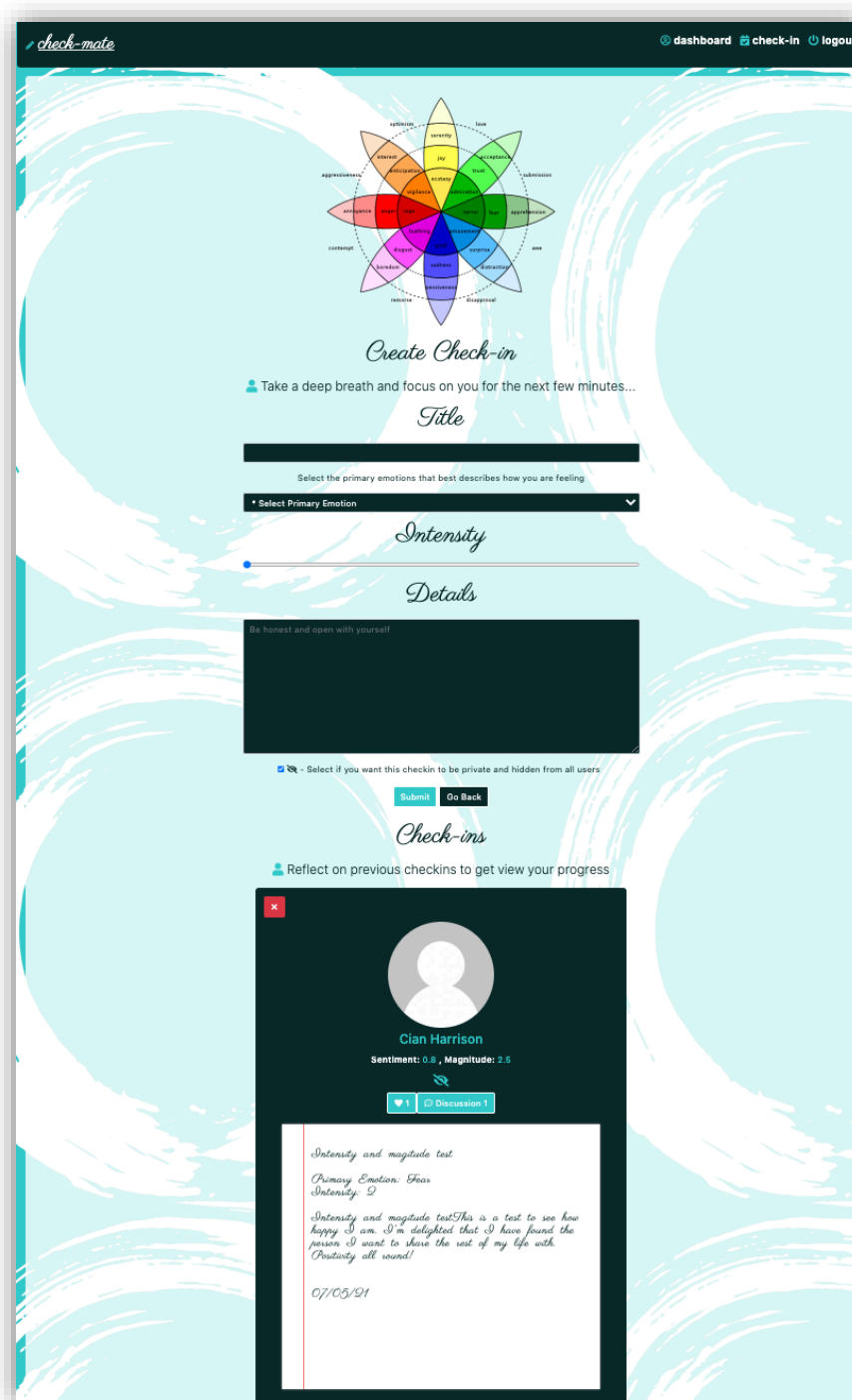
2.4.2. Dashboard Component



The dashboard component is the main area of the application for authenticated profile users. Users can find access all of the different features and functionality of the application through their dashboard. It was important to centralize the information to improve the website's navigability and overall user experience.

Users can also create new milestones to add to their profile here. These are displayed on the users dashboard to celebrate certain achievements or progress they have made along their personal journey.

2.4.3. Personal Checkin Component



This component is comprised of a number of subcomponents to make up the check-in section for users. This is where users take the time to complete their daily check-ins and reflect how they feel that day or in general. Users can also delete any checkins that they wish by selecting the x on the individual checkin

Users check-ins are analyzed using Google's Natural Language API and the results are shown to the individual owner of the checkin only. This was implemented using the Boolean privacy attribute of the checkin model when retrieving from MongoDB

2.4.4. Public Checkins Component



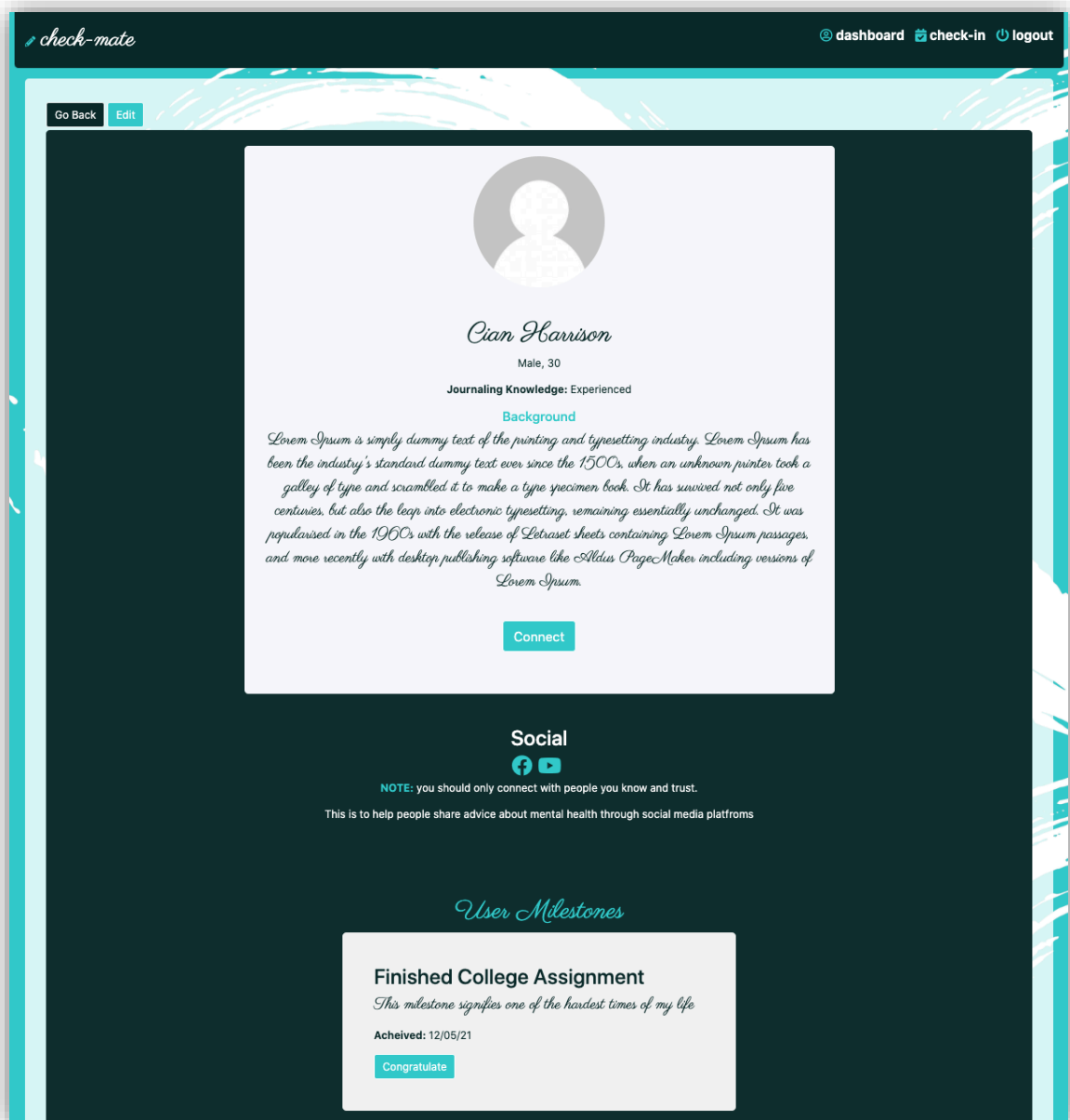
The community check-in component displays all check-mate user's check-ins that are flagged as public. This is where users can check-up on friends and other users who may need support even just a quick chat. Users can add a support heart or join the discussion on a specific topic by interaction with the action buttons on each check-in. The aim of this page is to encourage interaction between users and give people a chance to open up.

2.4.5. Checkin Discussion Component

The screenshot shows the 'check-mate' application interface. At the top, there is a navigation bar with 'check-mate' on the left and 'dashboard', 'check-in', and 'logout' on the right. A 'Go Back' button is located in the top left corner of the main content area. The main content area features a large, stylized background image of a person's face. The central focus is a checkin discussion component for user 'Aofie Reilly'. This component includes a circular profile picture, the user's name, and a text area containing the following text: 'A Tough Day at The Office', 'Primary Emotion: Fear', 'Intensity: 6', and a paragraph of Lorem Ipsum text. Below the text area is a date '28/04/21'. Underneath the checkin is a dark input field labeled 'Add your comment' and a 'Submit' button. At the bottom of the screenshot, another user's profile 'Emily Walsh' is partially visible, showing a circular profile picture and a snippet of Latin text: 'Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa'.

This component is displayed when a user clicks on the discussion link on a user's checkin, whether that be their own checkin or another user's. Any existing comments in the discussion are displayed in a thread below the original checkin entry and users have the option to join the discussion or provide some advice using the input field displayed.

2.4.6. Profile Component



The profile component is used to display information about a user that they make public on their profile. This where users can users can view Milestones they have logged in the application, as a reminder of the progress they have made on their journey.

2.4.7. Profile From Component

check-mate dashboard check-in logout

Edit Your Check-mate Profile

Let's start by getting some basic profile information to help with your experience

Give us an idea of where you are at in your personal journey with mental health

Experienced

Age

30

Biological Sex

Male

This is used solely for analytics purposes and not for classification of any sort

Personal Bio

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived

Add Social Links Optional - do not share social links if you are not comfortable with it or are looking for a more private experience

Twitter URL

https://www.facebook.com/

https://www.youtube.com/

Instagram URL

Submit Go Back

The profile form component provides users with the ability to create a profile and also update existing profile information as required. The information provided here is displayed on the users public profile and can be viewed by other check-mate users who may want to connect.

2.4.8. Real-Time Chat Component



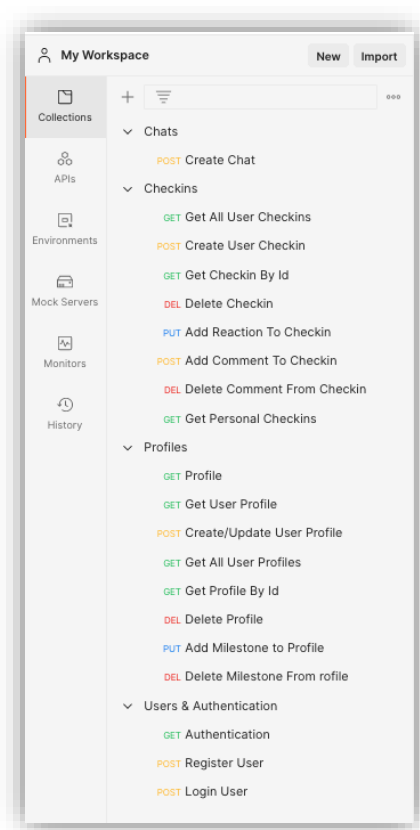
The chat component is where users can join open chat room that are dedicated to specific areas of mental health. The first option presented is for the user to select their desired room. Once connected, multiple users can chat in real-time to share advice with and provide support to each other. This component is facilitated using web socket communication and no messages are stored for confidentiality purposes.

2.5. Testing

2.5.1. Testing Tools, Plans & Specifications

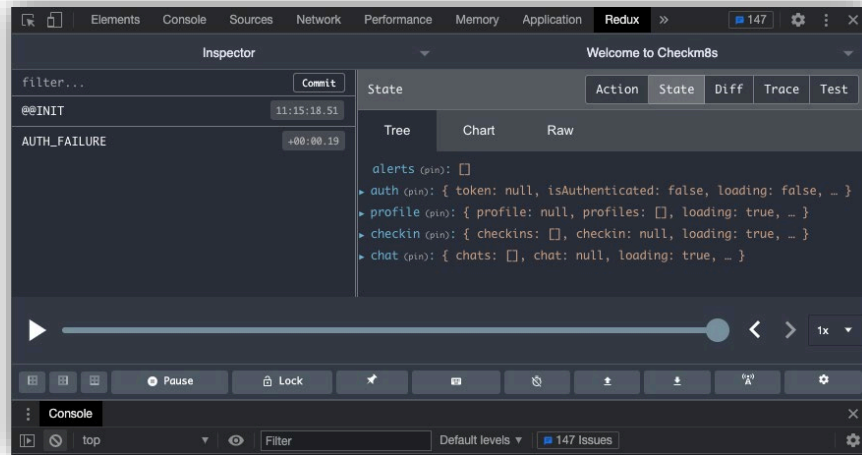
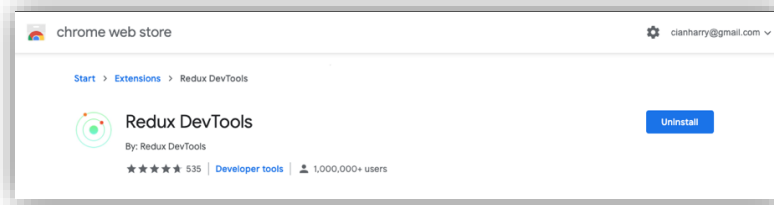
2.5.1.1. Postman

Postman is widely perceived as the industry standard tool for designing and testing APIs in web application development. There are twenty different endpoints that make up the Checkmates API which meant that testing each of these individually was important. Postman provided the perfect solution to test these endpoints especially before the front end of the application was developed. The workspace allows the grouping of different HTTP method requests into collections which was helpful for organization during development.



2.5.1.2. Redux Developer Tools

The Redux module was crucial to the development of the Checkmates application. This framework combined with React, are what facilitate the application's state level management and routing on the client side of the application. Redux Developer Tools browser extension was applied to test the front end state and the different reducer actions that manage it. There are 5 separate reducers implemented throughout the application which provide the fast loading single page application characteristics associated with front end frameworks such as React.



2.5.1.3. React Testing Library & Jest

React Testing Library provides a virtual DOM for testing purposes that allows developers to test component rendering, mouse clicks and other action based events without the need for a browser. Jest is the a test runner that locates and runs tests written in a React application . It also provides the assertions on whether the test has passed or failed. By running the command 'npm test' in the client directory of the project, the react scripts command is triggered and jest locates and executes any tests within the App.test.js file. A simple example can be seen below, that shows the test results of the rendering of the 'Sign Up' and 'Login' links to the virtual DOM.

```
test('should render auth links', () => {
  render(<App/>)
  // screen global object is used to access the virtual dom
  const linkElement = screen.getByRole('link', { name: 'Sign Up'})
  const linkElement1 = screen.getByRole('link', { name: 'Login'})
  expect(linkElement).toBeInTheDocument()
  expect(linkElement1).toBeInTheDocument()
})
```

```
PASS src/App.test.js
  ✓ should render auth links (245 ms)

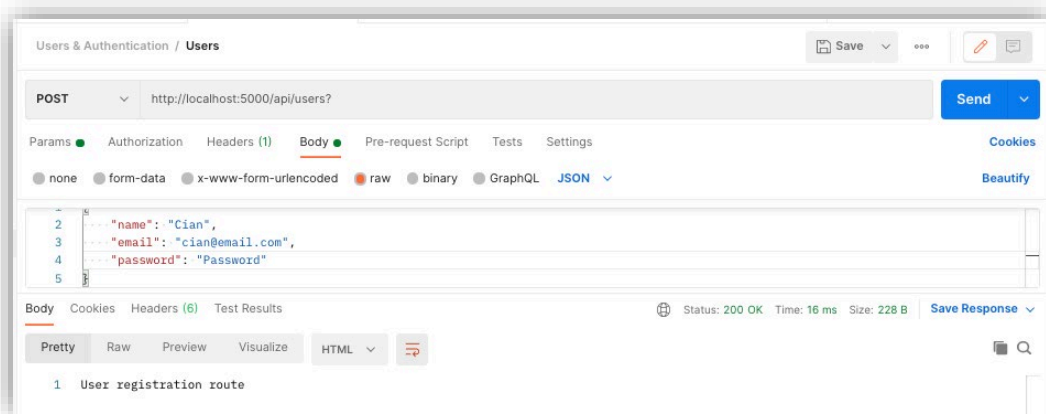
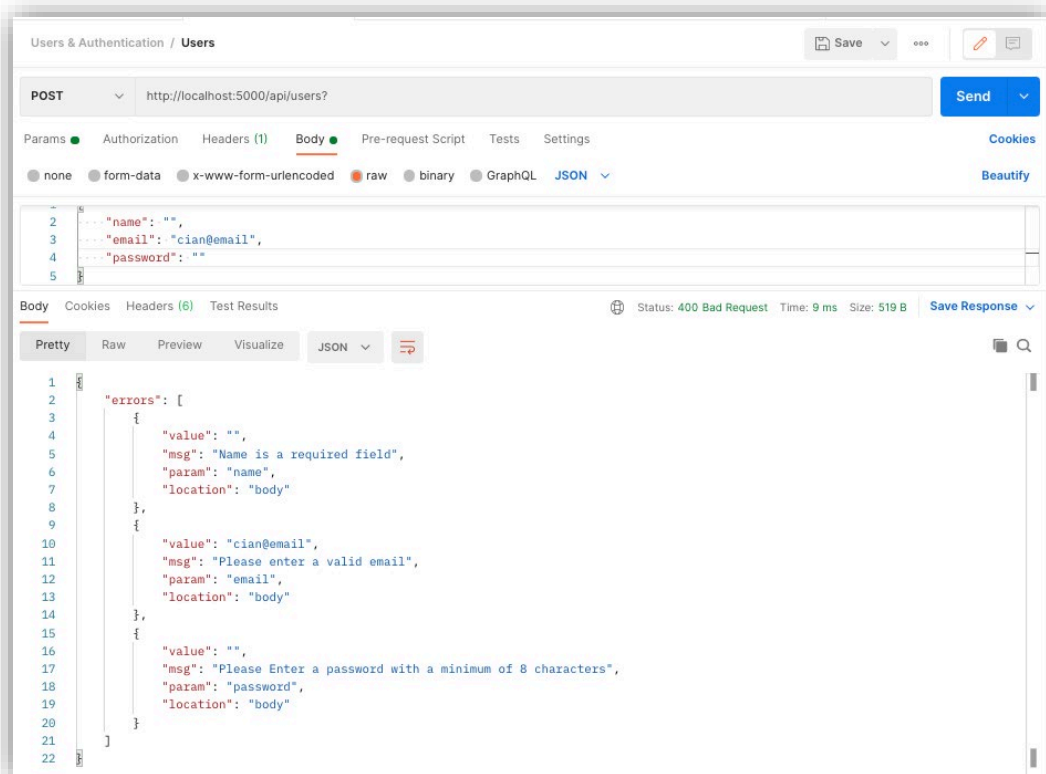
Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        2.83 s
Ran all test suites related to changed files.

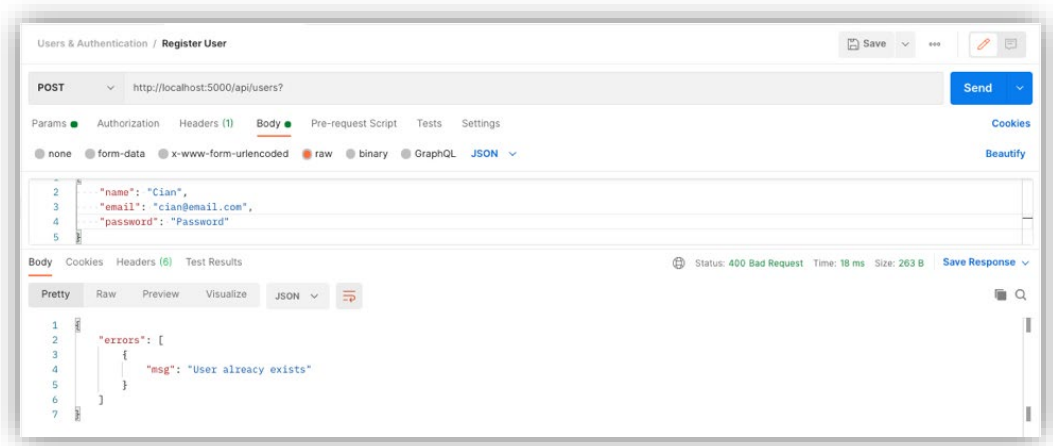
Watch Usage: Press w to show more.[]
```

2.5.2. Unit Testing

A complete set of test results can be found in Appendix 6.3 – Testing Documentation

Unit testing was used to test individual units of code in isolation, particularly in the early stages of developing the API. It is very easy to pinpoint failures which ensures that the individual units of code that that make up the application are error free before any dependencies are introduced. This was helpful when building the data models and testing the implemented constraints. Postman was used for the majority of the early unit testing, the full results for which can be found in Appendix 6.3. Below is an example of a unit test on the validation constraints for user model creation.





2.5.3. Integration Testing

A complete set of test results can be found in Appendix 6.3 – Testing Documentation

Integration testing was used to test how multiple units work together. It is important to cover the interactions between different components of the application to ensure they are functional in real situations. The results helped validate some of the more complex use cases that involved database and web socket connectivity. Below is an example of the Natural API Language call on check-in creation.

```
// Google Natural Language API call
const [result] = await client.analyzeSentiment({ document: document });
const sentiment = result.documentSentiment;
// ReqId: R0
// TestId: T055
console.log(`Text: ${req.body.maintext}`);
console.log(`Sentiment score: ${sentiment.score}`);
console.log(`Sentiment magnitude: ${sentiment.magnitude}`);
```

```
[0] Text: This is a test to see how angry I am. What a bad day, so stressful at work and the weather is terrible. I just want to be alone in bed!
[0] Sentiment score: -0.400000059604645
[0] Sentiment magnitude: 1.600000023841858
[]
```

```
[0] Text: This is a test to see how happy I am. I'm delighted that I have found the person I want to share the rest of my life with. Positivity all round!
[0] Sentiment score: 0.800000011920929
[0] Sentiment magnitude: 2.4000000953674316
[]
```

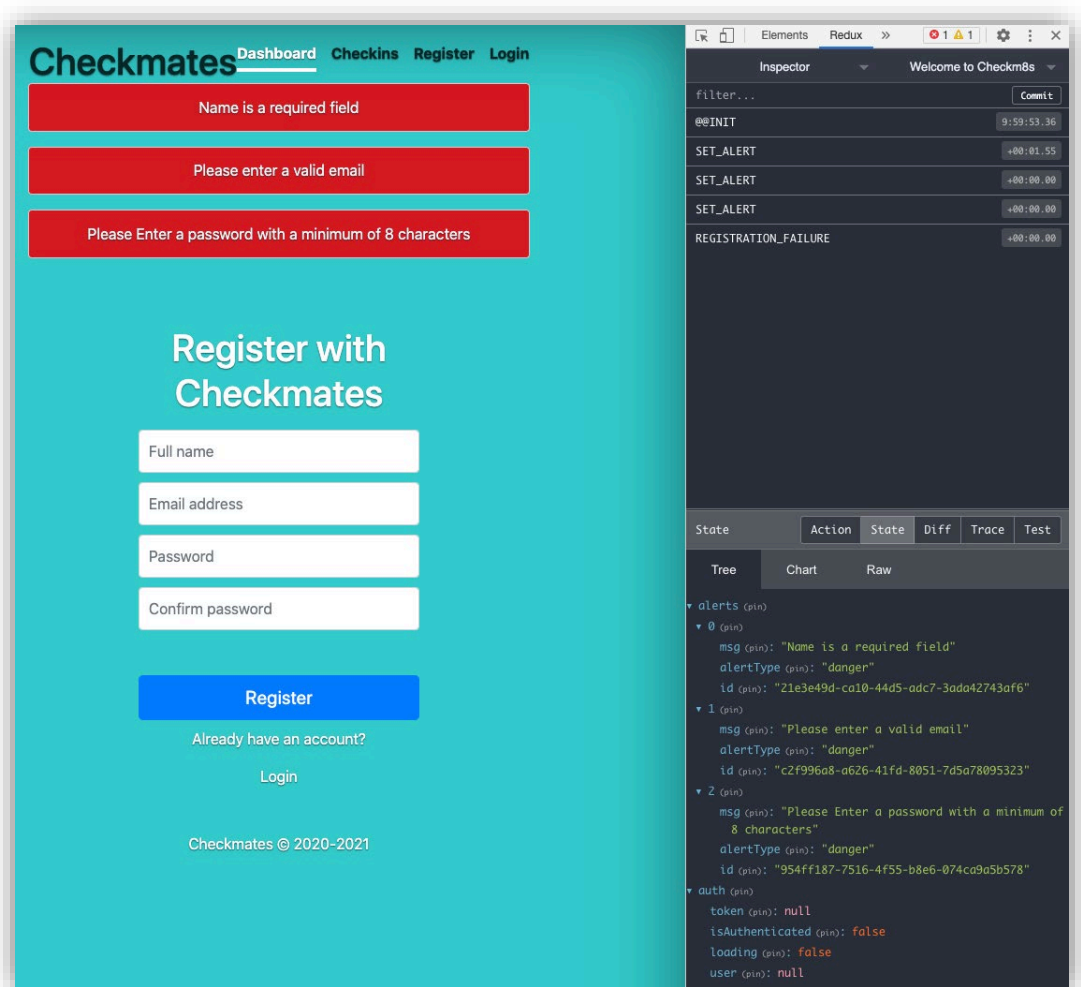
The screenshot shows a web application interface for 'Checkins' with a teal and white color scheme. At the top, there are 'Submit' and 'Go Back' buttons. Below is a 'Welcome to the checkmates community' message. Two check-in cards are displayed, each for a user named 'Cian Harrison' with a profile picture placeholder. The first card is titled 'Natural Language Test' and shows 'Emotion: Joy', 'Intensity: 8', and 'Privacy: Public'. The text of the check-in is 'This is a test to see how happy I am. I'm delighted that I have found the person I want to share the rest of my life with. Positivity all round!'. It includes icons for a heart, discussion, and private chat, and a 'Checkin date: 29/04/21'. The second card is also titled 'Natural Language Test' but shows 'Emotion: Anger', 'Intensity: 6', and 'Privacy: Private'. Its text is 'This is a test to see how angry I am. What a bad day, so stressful at work and the weather is terrible. I just want to be alone in bed!'. It also includes the same interaction icons and date. On the right side, the Redux DevTools sidebar is open, showing a list of actions like '@@INIT', 'USER_PRESENT', 'GET_CHECKINS', 'CREATE_CHECKIN', 'SET_ALERT', 'REMOVE_ALERT', and 'CREATE_CHECKIN'. The state tree on the right shows a 'checkins' array with three elements, each containing 'id', 'text', 'emotion', 'intensity', and 'privacy'.

2.5.4. Functional Testing

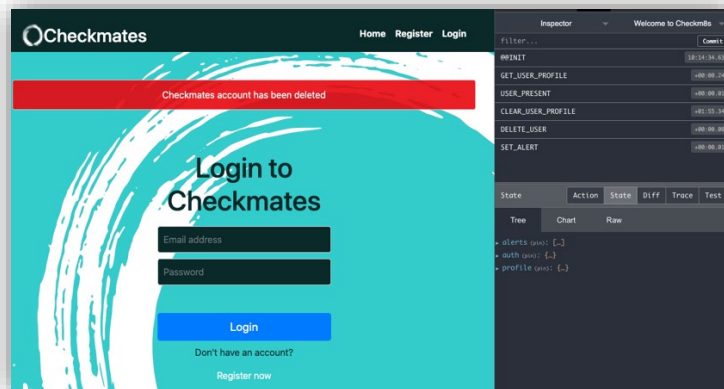
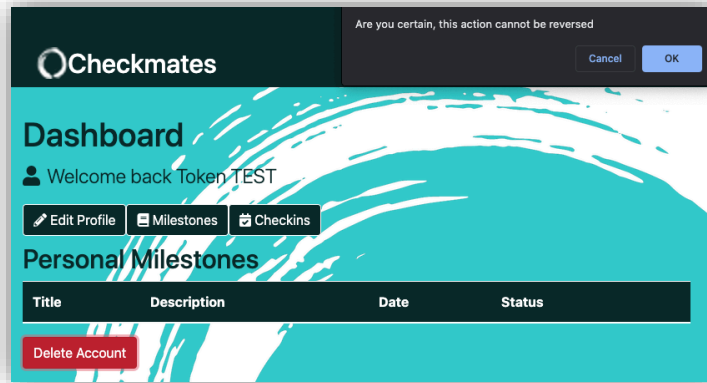
A complete set of test results can be found in Appendix 6.3 – Testing Documentation

Testing behaviour rather than code. Robust tests that replicate how users interact with the software. The only negative aspect is that it can be quite difficult to debug failing tests as the tests are not as tightly coupled with the code.

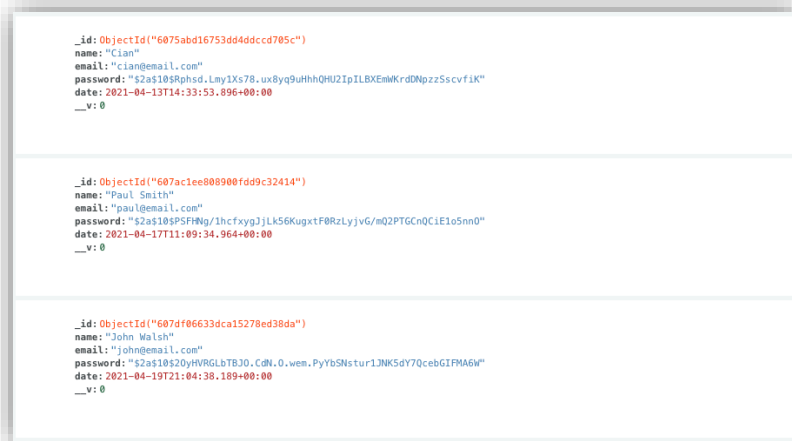
Below is an example of a functional test that was used to observe the user registration function. The results can be seen in the Redux developer tools console where the empty form field alerts are being displayed. This was then used to add the alert banners that are displayed to the user on the page.



Another example of functional testing using Redux Developer Tools can be seen on the next page, that illustrates the delete account function. This is a vital function of the application as it's important that all dependant resources and data relating to a user is deleted if they delete their account.



[User resource deleted]



[Profile resource deleted]



2.5.5. End User Testing

End user testing was conducted throughout the development process in keeping with usability design standards. This helped ensure that the application was designed with the users input. This was approached using 2 recognized techniques.

2.5.5.1. Five Second Test

A five second test was conducted to test the efficiency and overall layout of the landing page to see if user's understood the operating principle. Users were shown a flash image of the homepage for 5 seconds and asked 3 basic questions about what they saw. This provided actionable feedback that ensure the application was visually engaging and that branding was memorable.

Five Second Questions

- What is this website's function?
- What stood out on the homepage?
- What was the name of the company?

End User: 1

Question	Answer
What is this website about?	Health
What stood out on the homepage?	The Diagram
What was the name of the company?	Check-8

End User: 2

Question	Answer
What is this website about?	Emotions
What stood out on the homepage?	Wheel Diagram
What was the name of the company?	It had an 8 in it

End User: 3

Question	Answer
What is this website about?	Mental Health
What stood out on the homepage?	The colour of background was eye catching
What was the name of the company?	Checkm8

End User: 4

Question	Answer
What is this website about?	Mental Health
What stood out on the homepage?	Diagram with colours
What was the name of the company?	Checkm8

End User: 5

Question	Answer
What is this website about?	Emotion?
What stood out on the homepage?	Videos
What was the name of the company?	Check-me?

On evaluation of the findings, the branding was changed from Checkm8 to Check-mate as it seemed to be a little confusing for users to remember the name exactly. This feedback helped refine the application design and improved brand memorability.

2.5.5.2. Think Aloud Analysis

Think aloud is one of the most popular techniques used to test a website's usability. The purpose of this technique is to understand an end user's interaction and understanding of the interface. It also gives an idea of the general sentiment of the user as they are searching the site.

Task Name / Number	01 / Create Checkin
Task Goal	Register and create a checkin
Start & End Times	21:04 – 21:09
Expected / Ideal Behaviour	The test user can register and create a journal entry in under 5 minutes
Actual Behaviour	The task was completed in just over 5 minutes with minor distractions
Notes / Comments	The test user used the navigated directly to the dashboard menu rather than using the header
Anything additional	Dashboard feature section added as a result

This test provided actionable feedback in the form of a feature section for the dashboard that provide users with direct access to the main features of the application. The navbar was cluttered with too many links and users were not necessarily looking there first.

2.6. Evaluation

The testing results were crucial to the iterative evaluation of the development process. By testing the model constraints and data validation we ensured that the data models were appropriate to support the function of the system. This provided a comprehensive foundation to begin the development of the application logic.

On evaluation of the Integration and Functional Tests, it was clear that the React framework was providing considerable advantages in response times and browser loading speeds due to its component based rendering approach. This is of massive importance as users will not stick around if they are waiting around on a website.

The end user testing was the most valuable of all as multiple nuggets of valuable feedback were procured that helped refine the development. This resulted in a user-focused product, which in this area or specialization, is key to success.

3.0 Conclusions

One of the main advantages of the project is the chat component. This helps solve one of the persistent problems many people face with their mental health, and that is communication. The ability to communicate with other users who may understand and have experienced what you are going through provides a sense of consolation that cannot be found anywhere else. The support of other people can help us through the most difficult times and *check-mate* provides a platform for people to do just that.

Another advantage is that the component based architecture of the project allowed for a certain degree of separation of concerns, as components could be designed, styled and programmed in isolation. The different services and features are then logically combined to form the overall structure of the application. This helped to simplify some of the more complex aspects of the project and provided a really positive development experience.

A real strength of the project is the response time due to the client side rendering techniques implemented using React and Redux. The Single Page Application (SPA) provides a smooth and responsive journey for users when navigating through the application, something that is vital for a positive user experience.

A limitation of the project at present is the lack of options to upload additional or supplementary media content with user checkins. This is something that was intended but due to prioritization it was not completed. The creative aspect of music/art and media in general would definitely offer a more comprehensive checkin experience for users.

4.0 Further Development or Research

If there were additional time and resources provided for the development of the *check-mate* application, there would be a preferential areas to address. The first would be a better checkin experience for the user with a more interactive UI that explains each step of the process to the user. The more conducive the environment the more value the users will get from their daily checkins. It is important that users have positive sentiment towards using the application to allow them to build the habit of self-reflecting regularly.

Another addition would be to improve the level of detail in the natural language analytics provided by the *check-mate* application. A feature that tracks key words or recurring words could be utilized to identify triggers for certain emotions or situations for users. By providing detailed analytics like this, users would have more valuable feedback to help understand and adapt to their emotions.

At present there is a dedicated chat room for certain important mental health topics that authenticated users can access. With additional development resources, a private chat component would be implemented to facilitate one to one chats between individual users. A potential expansion of the business operation would be to introduce a counselling system in the application. The private chat component would be a fundamental part of this expansion, that would offer a personalized counselling experience for users who are having more difficulty managing their mental health alone.

5.0 References

5.1. Documentation

Plutchik, R. (1990). Emotions and psychotherapy: A psychoevolutionary perspective. In R. Plutchik & H. Kellerman (Eds.), *Emotion: Theory, research and experience, Emotion, psychopathology and psychotherapy* (Vol. 5, pp. 3–42). New York: Academic Press.

Plutchik, R. (1970). Emotions, evolution, and adaptive processes. In M. B. Arnold (Ed.), *Feelings and emotions* (pp. 3–24). New York: Academic Press.

Wellman Psychology. 2019. *Benefits of Reflective Journaling*. [online] Available at: <<https://wellmanpsychology.com/mindbodyblog/2019/8/22/benefits-of-reflective-journaling>> [Accessed 26 October 2020].

Sperling, M. (1996) 'Revisiting the Writing-Speaking Connection: Challenges for Research on Writing and Writing Instruction', *Review of Educational Research*, 66(1), pp. 53–86. doi: 10.3102/00346543066001053.

Korb, A., 2012. *The Grateful Brain*. [online] Psychology Today. Available at: <<https://www.psychologytoday.com/ie/blog/prefrontal-nudity/201211/the-grateful-brain>> [Accessed 7 November 2020].

Crane, C., Barnhofer, T., Duggan, D. S., Hepburn, S., Fennell, M. V., & Williams, J. M. G. (2008). Mindfulness-Based Cognitive Therapy and Self-Discrepancy in Recovered Depressed Patients with a History of Depression and Suicidality, *Cognitive Therapy Research*, 32, 775–787.

Roberts, S., 2012. *Can mindfulness make you happier?*. [online] The Berkeley Science Review. Available at: <<https://berkeleysciencereview.com/2012/11/can-mindfulness-make-you-happier/>> [Accessed 14 November 2020].

Ivtzan, I., Gardner, H. E., & Smailova, Z., (2011). Mindfulness meditation and curiosity: The contributing factors to wellbeing and the process of closing the self-discrepancy gap. *International Journal of Wellbeing*, 1(3), 316-326.

Lebec, G., 2017. *Thunks in Redux: The Basics*. [online] Medium. Available at: <<https://medium.com/fullstack-academy/thunks-in-redux-the-basics-85e538a3fe60>> [Accessed 3 April 2021].

5.2. Code

(Version 2.0) [Source code]. https://github.com/bradtraversy/devconnector_2.0

(Version 1.0) [Source code]. <https://github.com/machadop1407/Socket-Chat-App>

Blogs Dope. 2021. [online] Available at: <<https://www.codesdope.com/blog/article/getting-notebook-paper-effect-with-css/>> [Accessed 7 May 2021].

6.0 Appendices

6.1. Project Plan



National College of Ireland

Project Proposal

Checkm8

08/11/2020

BSHCE4

Software Development

2020/2021

Cian Harrison

X17116970

X17116970@student.ncirl.ie

Contents

1. Objectives.....	56
2. Background.....	56
3. Technical Approach.....	57
Market Research.....	57
Implementation.....	58
4. Special Resources Required.....	59
5. Project Plan.....	59
6. Technical Details.....	60
Front End Framework.....	60
Back End Framework.....	60
Hosting.....	60
7. Evaluation.....	60
8. Bibliography.....	61

1. Objectives

The objective of developing the Checkm8 web application is to provide a valuable service to people from all walks of life, by offering a secure and comforting environment where users can track and analyze their own emotional experiences whilst interacting with other approved users if desired. The general busy nature of our daily lives can leave us with little time to wind down and take note of how we are feeling, and more importantly why we are feeling that way? *Checkm8* will provide users with a *checkin* system aimed and giving people the chance to put their daily thoughts and emotions into words without any pressure or pre-defined structure. The level of detail or interaction is entirely up to the individual user and any information disclosed will be completely confidential to that specific user. This approach is aimed at encouraging users to take ownership of their own mental health and begin to build a routine that suits their needs.

There will also be a chance for users to interact with other Checkm8 account users on a consensual basis. Certain information can be shared between users if and only if both parties give their consent. The objective here is to give users a chance to open up and make that initial request for help and support in a difficult time. Opening up is usually the hardest part for people, as there is still a prevalent stigma that exists around mental health and the associated illnesses. Acknowledgement, Analysis and Discussion are the three key areas that will be focused on throughout the development of the *Checkm8* application.

By providing this generation with the right tools and techniques, we can start to improve our approach and attitude towards maintaining both our own personal mental health and impacting the lives of others in a positive way.

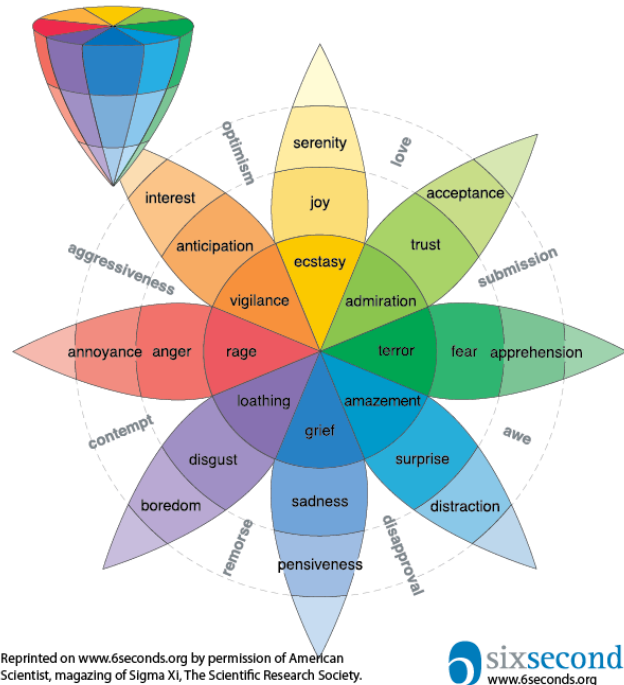
2. Background

In the Global Burden of Disease study published in 2017, the Institute for Health Metrics and Evaluation estimated that approximately 792 million people suffer from some form of mental health disorder throughout the world. (Ritchie and Roser, 2020)

This figure has only increased since that study was conducted, this is a shocking statistic when you consider that is well over 10% of the world's population. Although there has been an increased awareness around the issues we are seeing, there is little action being taken to provide answers as to why this figure is so high, or to develop a possible solution to combat the problem. With the continual growth of social media technologies and the control that they have over what we think and feel, we need to teach ourselves how to adapt and to train our mind to manage all of the new stresses that we face today.

A leading psychologist, Robert Plutchik has created a framework model that defines the primary 8 emotions that humans experience. This is one of the areas that I will use in my application to build the monitoring framework for user's checkins. This is renowned

theoretical model that helps to provide an understanding of the complexity of the spectrum of emotions that we experience on a day to day basis. I am confident that the checkm8 application can adapt this model to help structure the logging and analysis requirements of the project.



(Six Seconds, 2011)

3. Technical Approach

Market Research

The majority of the applications that I researched are based around suicide and self-harm prevention. Although useful, I really believe that these apps are looking at the problem from the wrong perspective. They are trying to provide a solution to the end problem rather than looking at the root cause of the issue. We have failed to realize what the real problem is if we have let someone get to that point without having some sort of intervention or open conversation.

The biggest wellbeing and lifestyle application on the market at the moment appears to be Calm. This app focuses mainly on meditation and sleep to help combat anxiety. I do appreciate the benefit that these apps have to offer but I think they are only treating a subsection of the problem rather than looking at the fundamental issue. It's clear to see that people are becoming more aware and conscious of their mental health from the popularity of mindfulness and meditation apps. This is an encouraging sign and another reason that this market is so appealing. *Checkm8* has a unique concept and I believe the features will differentiate it from the services provided by other competitors in the market.

One of the applications that I found that has some similar features to the product I am intending to design is an app called 'Happyfeed'. The main difference is that this app is designed for recording happy emotions only, which is intended to create a greater sense of gratitude for the good things in your day. I like this approach and I understand the reasoning behind it but we cannot avoid the negative thoughts and emotions, we must be tracking the full spectrum of emotions in order to provide a realistic and effective solution for users. The UI has soft, pastel type colours which is really complimentary to the overall look and feel of the app. This is something that I will incorporate in my design strategy for the *checkmates* user interface.

Implementation

Users will build up their profile with general person information and have the ability to invite friends to join the app. The aim here is for user to be able to procure their own 'trusted circle' of connections with their selected users. This will be private to the user and the person who they are connected with to ensure privacy and confidentiality for both users. The main goal is to provide a *checkin* service for users, almost like their own diary but with more to offer.

Users will be asked to select a routine that they think suits them to get started so that they can receive gentle reminders to complete their intended checkin. For each checkin users will select the emotion that best represent how they feel that way, using the *Plutchik Model* as a guide. The user will associate and intensity score at this point to create another data point for analysis later on. This will be used for in application analysis to provide feedback for the users themselves and also with the option to include an AI component for more severe cases where self-harm is a risk. The idea here is to develop an algorithm that can help provide valuable insight into users personal psychological state and if there are queues to help improve that.

The fundamental feature will be the digital journal/diary in the application. This is where users will be encouraged to enter their general feelings and emotions on a daily basis, whether good or bad, through text input or voice recording. The thought process here is to have users consciously address any troublesome thoughts or emotions as they occur to avoid any build-up of anxiety or depression. This remains completely private to the user to ensure that they feel comfortable and secure when opening up and disclosing sensitive details in the journal. Users will also be provided the option to upload media to attach to and associate with their *checkins*, which is aimed at providing users with more ways to express and acknowledge their emotions through creativity and the arts.

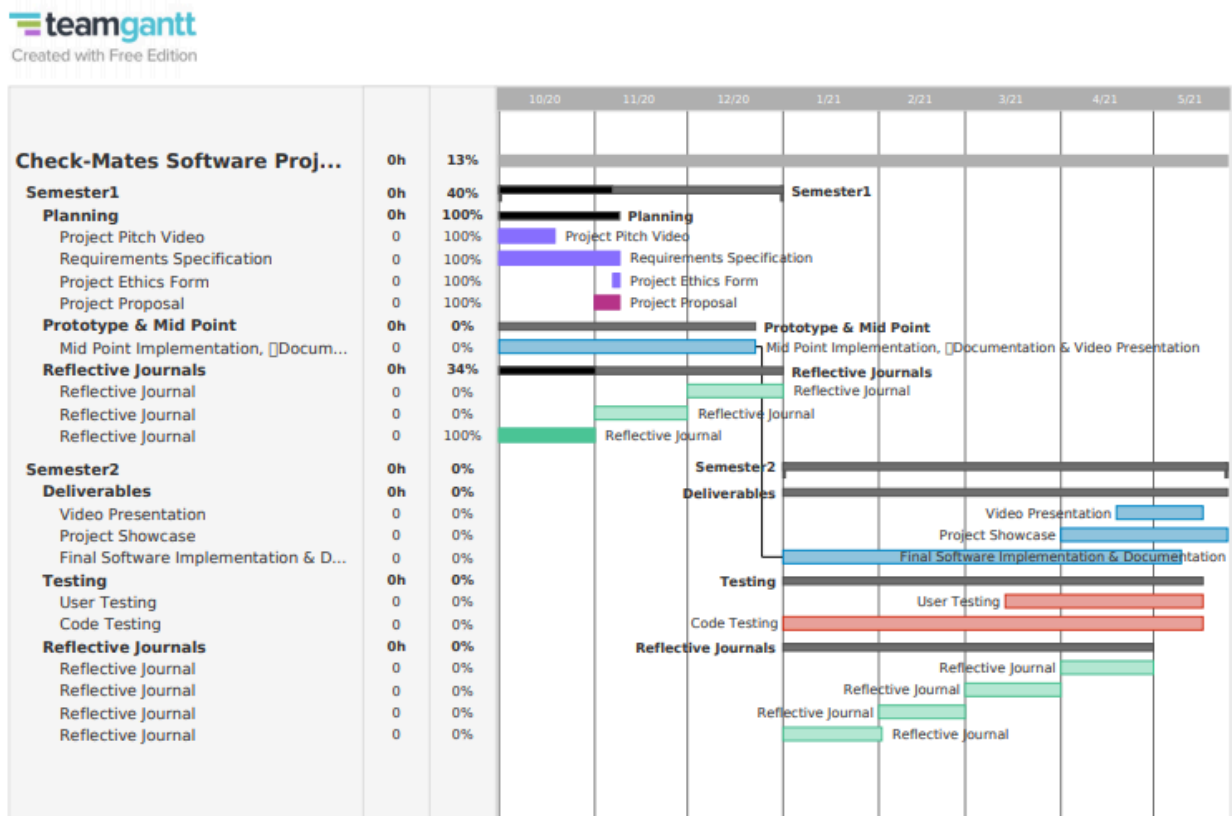
On the home dashboard for a user, they will be able to view the media items that have been selected by users in their 'trusted circle' for recent journal entries. There will be no details provided with the media item here, only an indication of whether the item has a more positive or negative emotive score than the friends previous entry. If a user wants to reach out to a friend having viewed their post, they can request to start a conversation with

user where they can begin to share some more descriptive information about why they are feeling a certain way. Again this must be consented by both parties to proceed. Once the chat is ended by either party the conversation will be terminated and there will be no history of the chat. This is to ensure that confidentiality is maintained.

4. Special Resources Required

There are no specific special software or hardware resources required at present and there is nothing to suggest that that will change moving forward with development.

5. Project Plan



6. Technical Details

Front End Framework

- Javascript
- HTML & CSS
- React.js & Redux

Back End Framework

- JavaScript
- Java/Python
- Node.js & Express API
- MongoDB

Hosting

- Heroku Deployment

7. Evaluation

I have created a Continuous Integration/Continuous Development pipeline to help with the continuous development, testing and deployment of the *Checkm8* application lifecycle. Due to the nature of the schedule, there will not be enough time to rectify any major bugs or issues that could be identified during testing if it is done at the end of the development lifecycle. I will be taking an iterative approach to this development effort in keeping with AGILE principles.

The application will be put through user testing at a number of key phases throughout the development process to ensure that we are meeting the expectations of our intended users and to get valuable feedback on the layout and aesthetics of the User Interface.

Possible migration to AWS where some of the technical approaches will need to be refactored.

8. Bibliography

Ritchie, H. and Roser, M., 2020. *Mental Health*. [online] Our World in Data. Available at: <<https://ourworldindata.org/mental-health>> [Accessed 5 November 2020].

Six Seconds. 2011. *Plutchik's Wheel Of Emotions: Feelings Wheel • Six Seconds*. [online] Available at: <<https://www.6seconds.org/2020/08/11/plutchik-wheel-emotions/>> [Accessed 1 December 2020].



National College of Ireland

Software Project

Reflective Journal - April

01/05/2021

BSHCSDE4

Software Development

2020/2021

Cian Harrison

X17116970

X17116970@student.ncirl.ie

My Achievements

This month has been incomparably productive in relation to any of the previous months throughout both semesters. The extra time and headspace has allowed me to accelerate the development process significantly and I know have the majority of my application functionality complete.

Final API Testing

I was finally able to fully test all the application's endpoints that expose the resources in MongoDB. I used postman to test each endpoint individually and verify the response data that was returned. I compiled these results and added them to the test log documentation for the final technical report.

Cloud Natural Language API

I have successfully implemented the cloud natural language API in the development environment and tested it with some dummy checkin data to observe the results. There may be an issue with using this feature in the production environment as the service account API access key is stored locally on my machine and should not be added to deployed code.

React-Redux Development

I created and implemented the 4 Redux reducers that I intended to this moth which has allowed me to properly complete the front end routing data management. These reducers control the actions that manage the application state on the client side. This library enhances React and provides the single page application structure. The browser does not need to reload if actions are being called within the same reducer which increases response times and improves the user experience.

Deployment

The deployment platform I am using is Heroku as its best suited for the MERN stack application structure (Mongo, Express, React, Node). The first step was to configure the environment variables in the Heroku environment to ensure that the MongoDB connection is maintained and that the appropriate secret is used to hash the JSON web token. The first deployed version of the application is working correctly but with limited functionality.

My Reflection

This month has been one of the toughest months of my life, there are so many deliverable that must be met and the complexity of the application development has continually increased. I have also had an interview for a new role in work which has added additional pressures. On reflection, I am proud of how I have dealt with such circumstances and I am

confident about continuing my professional development with the knowledge and skills I have picked up this year at NCI.

Intended Changes

I have yet to implement the private chat functionality into the application. This is a feature that I would really like to complete before the deadline as it would really boost the value that the application provides to the end user. I would also like to put more time into the user interface design and aesthetics if there is enough time to focus on that area.

Supervisor Meeting

Date: 30th April

Time: 7.30 – 8.00pm

Location: Teams

Attendees: Cian Harrison, Clive Gargan

Meeting Notes:

- Progress checkin
- Tech Report review scheduled for next week
- Advice for Video Presentation

National College of Ireland

Software Project

Reflective Journal - March

01/04/2021

BSHCSDE4

Software Development

2020/2021

Cian Harrison

X17116970

X17116970@student.ncirl.ie

My Achievements

Cloud Natural Language API

One of the main functions of the app is to determine the sentiment of user check-ins to be able to provide valuable feedback on their digital diary entries. I have decided to use Google's Cloud Natural Language API to implement this functionality into the Checkm8 application. This API service offers natural language understanding technologies including, sentiment analysis, entity recognition and other text based annotations.

I have enabled this API for the checkm8 project on the Google developers console and I have set up the API keys both locally and in the deployed Heroku environment. The next step is to implement the functionality into the application and begin testing the API to ensure that the feature is operating as expected and is providing value to the user.

Module Submissions

- Usability Design – Usability Design Report (14/3/21)
- Distributed Systems – gRPC Distributed System Project (25/3/21)
- Cloud Application Development – Rails Information Management Application Project (2/4/21)

My Reflection

This month has been filled with deadlines for the other module that I am taking this semester. For this reason, I have not had as much time to complete development on the software project as I first hoped. I am really happy with the submissions that I have made for the other modules and I am looking forward to applying the knowledge and skills I have learned to my final software project.

Intended Changes

I have a few front-end design ideas that I have picked up from other modules that I will be incorporating into my software project. This will be part of the changes that I will be making in the next week of project work. The front end of the application is not as developed as the back end, so that's what I will be focusing on for the coming weeks.

Supervisor Meeting

Date: 2nd March

Time: 9.00 – 9.30pm

Location: Teams

Attendees: Cian Harrison, Clive Gargan

Meeting Notes:

- Mid-point feedback discussion
- Questions on testing techniques - Advice on testing (Coded UI), Postman to test any exposed APIs
- Have agreed to review all documentation 2 weeks before submission to ensure that all deliverables are covered sufficiently.

National College of Ireland

Software Project

Reflective Journal - February

01/03/2021

BSHCSDE4

Software Development

2020/2021

Cian Harrison

X17116970

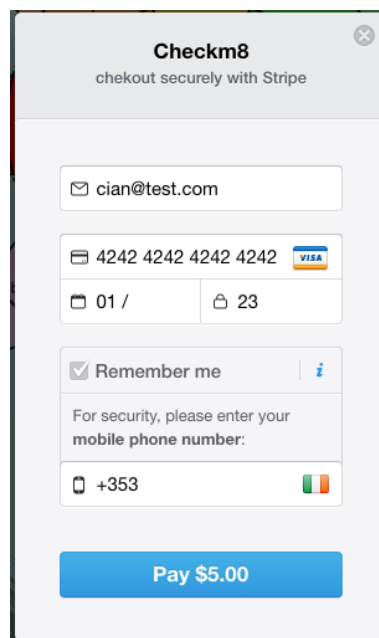
X17116970@student.ncirl.ie

My Achievements

This month I have made considerable progress in the development of the front end and the back end of the Checkmates application. There were a couple of main areas that I focused on this month; one was implementing the Stripe API on the back end and the other was the development of the UI on the front end of the application.

Stripe API

As this service handles user's bank account detail this is a very sensitive area in terms of security and data protection. To facilitate the handling of the environment variables for both the development and production environments, I created separate .env files to store each of the API keys that provide authorized access to the Stripe checkout API. I installed the react-stripe-checkout npm module which provides a template checkout component that can be modified by developers. Below is a screenshot of the initial development:

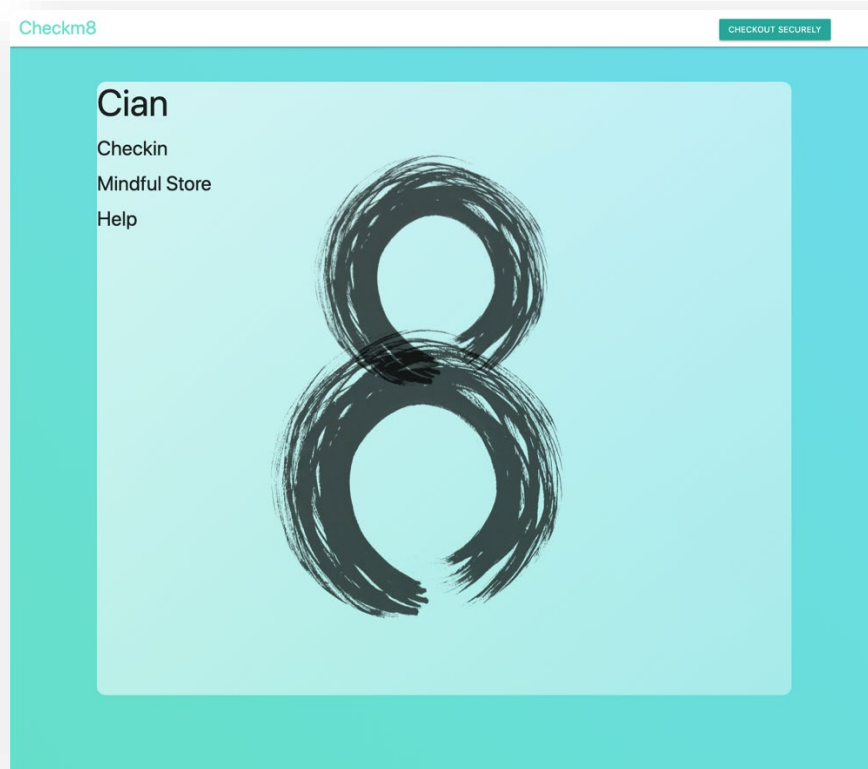


This feature will be used for the store section of the website where users can browse mindfulness and health related products. Stripe is a renowned provider of third-party payment facilities and this is an option that is preferable for smaller companies who do not have the knowledge/resources to facilitate proprietary transaction functionality themselves.

I also had to create the appropriate Stripe secret key environment variable for my Heroku deployment that hosts my production version of the application. This has been completed successfully and tested in both of the environments, there is still some development needed from a design perspective and but I am happy with this feature for now.

Frontend Development

I have started to put some time into the developing the basic design of the applications user interface. I have designed a glass style card component that will be the container that hosts the dashboard, profile and other components. The colours have been carefully selected to provide an ambient and comforting user interface for users to interact with, in keeping the overarching theme of mindfulness and wellbeing in the application.



I created an App.css file that contains most of the styling information for the application so far but moving forward I will be separating out the .css styling into separate files based on the component that they relate to e.g Checkin.js -> Checkin.css.

My Reflection

This month has been a reminder of how much time and effort is required to get through the development phases of a dull stack project. With this in mind, I will be devoting additional time each week to the software project to ensure I am making the required progress with the application's functionality and appearance.

I have had some difficult issues outside of college for the past few months that has been a major distraction from college. I felt unmotivated and have been procrastinating which was not a desirable situation for this semester. This situation has improved recently so I am hoping that this will allow me to be more productive and maximize my time and effort moving towards the end of this crucial semester.

Intended Changes

By the time I complete my next reflective journal I will have completed all of the deliverables for the other modules this semester and will be able to devote my complete focus and effort on the software project. I will be putting further work into the UI of the application over the coming as I build out the application. I want to have a functional and aesthetically pleasing UI ready in the coming weeks so I can test it with users.

Supervisor Meeting

N/A

National College of Ireland

Software Project

Reflective Journal - January

01/02/2021

BSHCSDE4

Software Development

2020/2021

Cian Harrison

X17116970

X17116970@student.ncirl.ie

My Achievements

The early part of January was focused on the Terminal Assignment Based Assessments for both the Strategic Management and Advanced Programming modules, this meant that work on the software project was limited. I have used the 2 weeks of time off in between semester 1 and semester 2 to improve my understanding of React and how I will be utilising its functionality in my application.

TABA Deliverables

The schedule toward the end of semester one was hectic and that leaked into the first week or so of January with the TABAs. This was a stressful period and one that I am glad to be through with. As I have been working full time throughout this busy period, I had to focus solely on the TABA assignments and did not get any work done on the software project during this time. I took a few days off after these deadlines to relax and take a break from intensive studying but started preparing for semester 2 after that.

React Research and Tutorials

I have been using Udemy as a supplementary resource for learning throughout my time at NCI and this year I have been using it to help with the development process of my software project application. One of the courses is helping with the component lifecycle that React uses to render and re-render the Document Object Model (DOM) as updates occur in the browser. Fundamental concepts like this are vital to the development of a component-based web application where re-rendering in the browser is constant. I will continue with this course in conjunction with my studies this semester with the aim of producing a high quality, modern application using React and the complimentary libraries available.

My Reflection

On reflection of Semester 1, I identified that I procrastinated with some of the assignments for other modules which put additional pressure on me towards the end of the semester. I have mapped out all of my module deliverables on a Trello board for this semester and I am adamant that I will stay ahead of schedule this semester to allow me to approach each deliverable with the appropriate time and effort. I am aiming for an honour's degree so I am aware that I need to maximize my efforts to achieve this goal.

Intended Changes

In terms of scheduling work on assignments, I will be allocating a lot more time per week to completing development on the software project as there is one module less this semester. This will benefit all of the student greatly as it provides not only more time but headspace for students with multiple deliverables in the upcoming weeks and months.

I will start to implement the component-based elements and techniques that I am learning about through my research into React. This along with the increased time devoted to the software project will accelerate the development process and begin to create a feature rich web application interface.

National College of Ireland

Software Project

Reflective Journal - December

03/01/2021

BSHCSDE4

Software Development

2020/2021

Cian Harrison

X17116970

X17116970@student.ncirl.ie

My Achievements

The main focus for this month in terms of the software project were the Mid-point presentation deliverables that were due for December 22nd. The components that were covered were the finalized Project Proposal and the first 2 sections of the Technical Report, the code development for the mid-point prototype, and the presentation slides and recorded demo of the working prototype.

Mid-point Presentation

This deliverable required the preparation of a power point presentation in order to structure the video recording. The video itself was quite difficult to do as there many different aspects to cover within the 10-minute time limit. After multiple attempts I was satisfied that I covered all of the requirements of the demo and conveyed the fundamental ideas of my project to the viewer.

Technical Report & Project Proposal

Finalizing the Project Proposal was much easier at this point as I now have a greater understanding of the requirements of my application and I have refined my approach in many different areas. It required me to do more in depth market research and observe the strengths and weaknesses of the individual competitors that are currently providing similar solutions.

The Technical Report and in particular the elicitation and specification of requirements using use case diagram techniques was the most beneficial aspect of this submission. It helped me define the general flow of the application and also identified some incomplete areas of the application that I needed to pay more attention to.

Prototype

This month was the start of the code development for my application in preparation for the mid-point prototype submission. I feel that this was the earliest point I could have started as my ideas and implementation approach has only been clearly defined since the start of December. One of the most aspects I completed this month was the creation of the separate development and production environments for my application. This was a deliberate decision made in order to ensure that I have a continuously integrated and deployed application throughout the course of the development. I also created the back end server using Node.JS and Express framework and implemented basic authentication in the application using Google Oauth20.

My Reflection

Looking back on the achievements this month, I am amazed at the volume of work that my classmates and I have gotten through. Not only did we have an important deliverable due for the software project module, but we also had several important end of semester submissions due for our other modules. The use of task management tools such as Trello in my case, is an absolute must. This has helped me organize my time in respect to upcoming deliverable I have each week. Although it has been incredibly tough, I'm happy that I have gotten through the first semester of the year in the position I am now.

Intended Changes

I have many intended changes from a technical standpoint but also in terms of how I am approaching and planning my project work. Next semester I am looking forward to having 1 less module than semester 1. I will invest that extra time in the development efforts for the software project and I hope that this will help increase productivity at an important time of the year.

From a technical perspective I am looking to develop the front end of the application and incorporate more of the features provided by React JS and Redux. The UI is one of the most important parts of developing a mindfulness and mental health application and that is where my focus will be over the next month.

Supervisor Meeting

Date: Wednesday 9th December

Time: 18:30 – 19:00

Location: Teams Meeting

Attendees: Cian Harrison, Clive Gargan



Reflective Journal – November 2020

CIAN HARRISON – X17116970

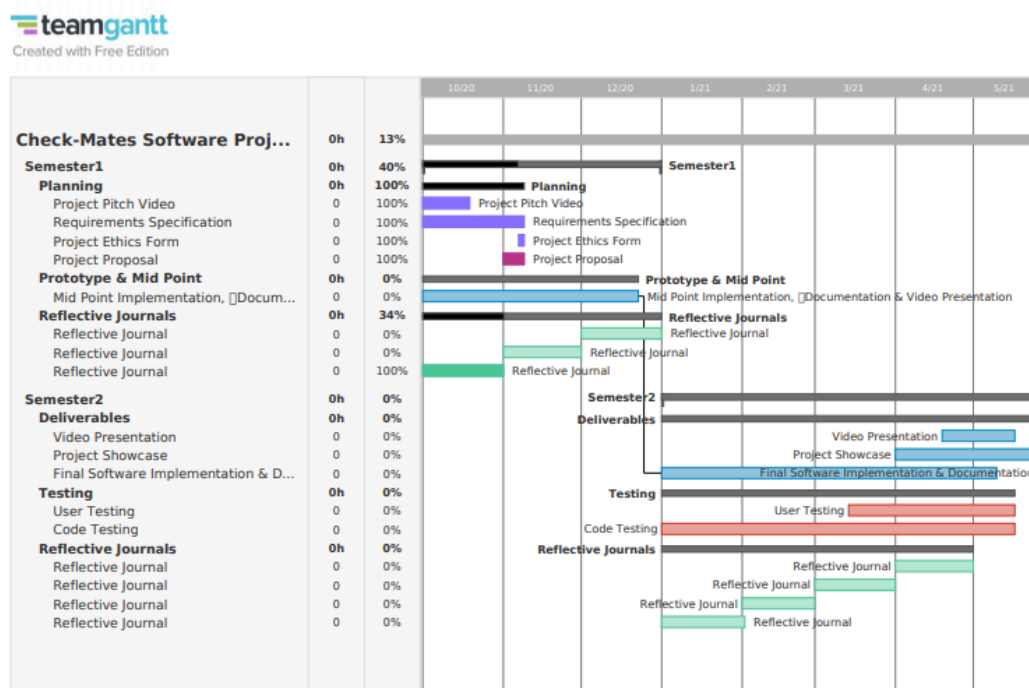
My Achievements

Project Proposal

The main focus this month for the Software Project module was to complete and submit the project proposal deliverable. This was a fundamental part of the planning process, as it pushed me to dig deeper into the subject matter of mental health and begin analysing some of the statistics involved.

Project Plan

This month, as part of the project proposal, I developed a project plan for the many different deliverables involved in the software project module. I used Trello with the teamgantt plugin to build up the diagram below which illustrates the plan.



Time Management

Although I did not invest a lot of time in the software development module this month, I was able to fulfil multiple deliverable deadlines for other modules which will allow me to dedicate more time to project work moving forward. There was a bottle neck that I was able to negotiate through by managing my time efficiently and effectively. Trello is a tool that I started using in 1st year and I feel it has provided great benefits in terms of tracking assignment and current assessment deliverables.

My Reflection

In creating and reviewing the project plan, I realised that there is a large number of deliverables that need to be met throughout both semesters to ensure that the module is completely successfully.

Seeing this visually represented has help re-iterate the importance of managing my time and efforts this year. I like knowing what has to be done and when it has to be done by, that way I can plan appropriately and change things around to suit different modules.

Intended Changes

I will be building out my use case diagrams this month, and I am more than certain that this will result in some changes to the initial structure of the application. I'm hoping that be the time I complete these requirements specifications tasks that I will have comprehensively defined a refined set of requirements to feed the development effort.

From a technical perspective the changes next month will involve setting up the coding platform and application structure to begin the practical development of a prototype. I will be setting up my GitHub repository for version control and will maintain that strategy throughout the project.

Supervisor Meetings

Due to the lack of real progress with practical development of the project I have not met with my supervisor, I am organizing a meeting this week to go over the project proposal and the planned implementation strategy.



Reflective Journal – October 2020

CIAN HARRISON – X17116970

My Achievements

In the first month of this semester I have developed and pitched my project idea for review by my academic supervisor. Filming the pitch was harder than expected but it will serve as good preparation for the other project submissions further down the line.

I've conducted high-level market research on the potential idea that I have. I will proceed with more detailed analysis of the target market and competitors in the market when I have feedback from my supervisor to confirm the idea.

I have developed a Trello board to organise the software development module and the many different deliverables that are due throughout the year. I hope this will help me optimize my time by getting this on to paper and not having it weighing on my mind

The screenshot shows a Trello board for a 'Software Development Project'. The board is organized into three columns: BACKLOG, IN PROGRESS, and COMPLETE. The BACKLOG column contains several cards, including a 'Mid Point Marking Scheme' card with a table of tasks and their completion percentages, and a 'Class Completely Reused' card with a code snippet. The IN PROGRESS column features a 'Project Ethics Form' card with a flowchart diagram and a 'Project Proposal' card. The COMPLETE column includes a 'Project Pitch Video' card. The board also shows a 'Reflective Journal' card in the IN PROGRESS column and another in the COMPLETE column. The background of the board is a colorful, abstract image.

Mid Point Marking Scheme	
Data Analysis	50%
Project Proposal	50%
High Level Analysis	50%
Communication	50%
Implementation & Documentation	40%
Post-mortem Data Analysis	40%
Technology/Tools	20%
Prototype	20%

```
public class NewClass {
    public String newMethod(int param) {
    }
}
```

My Reflection

Pitch Video – The pitch video is a good idea because it forces you to think about the different aspects involved in the project from the get-go. It also allowed me to critically analyse my idea, personally, and begin to improve and refine some of the ideas/features that I intend to include in my application.

Virtual Learning Environment – it's difficult to adapt to as the structure of going to college and sitting in a classroom is gone. There are many distractions at home and it can take from the learning experience. I have removed the distractions from my study space to ensure that I optimize my time spent and I feel this has helped a lot.

Intended Changes

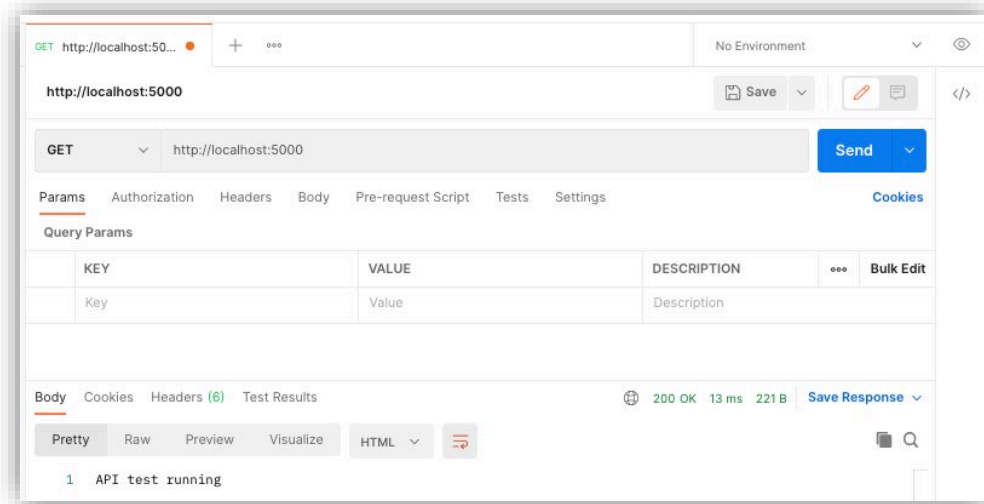
As to be expected this year, change will be ever present. Even if my idea is approved by the academic supervisor, I am sure there will be plenty of suggestions on what to change and how to move forward. I hope to use this feedback to help finalize my idea and build out the Project Proposal document.

Supervisor Meetings

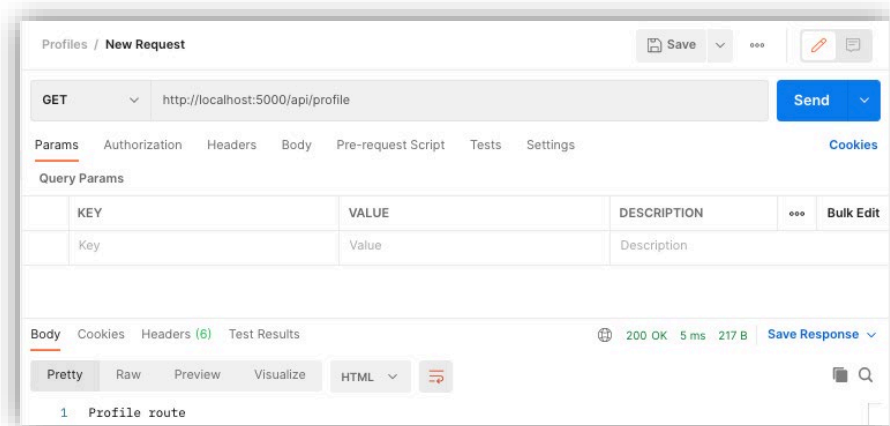
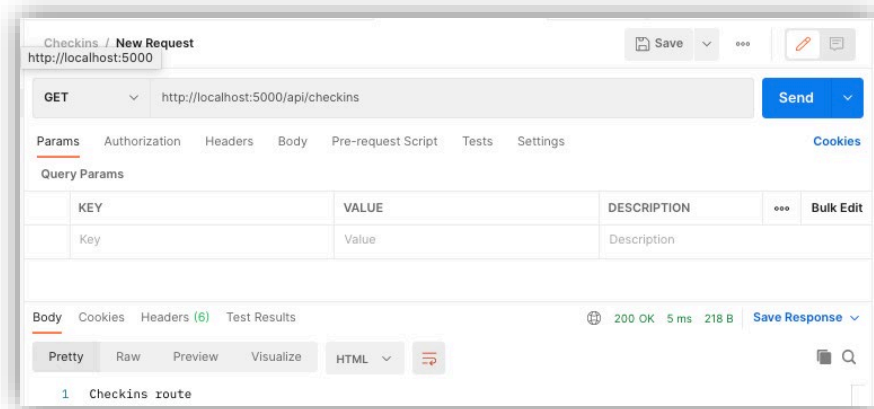
N/A – supervisor has just been assigned.

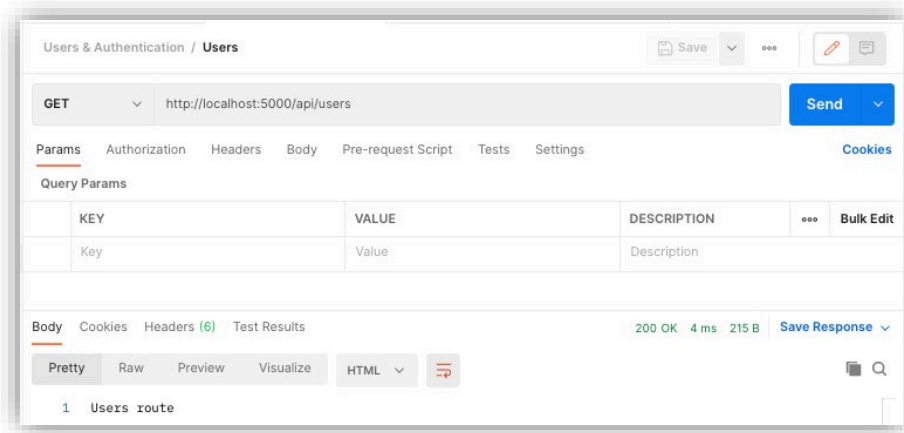
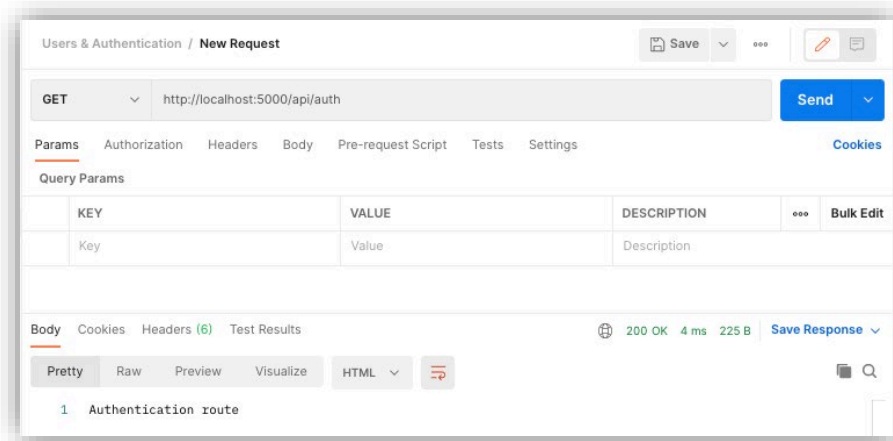
6.3. Testing Documentation

Test_Id	T001
Details	Initial Node & Express API test
Date	20/1/2021

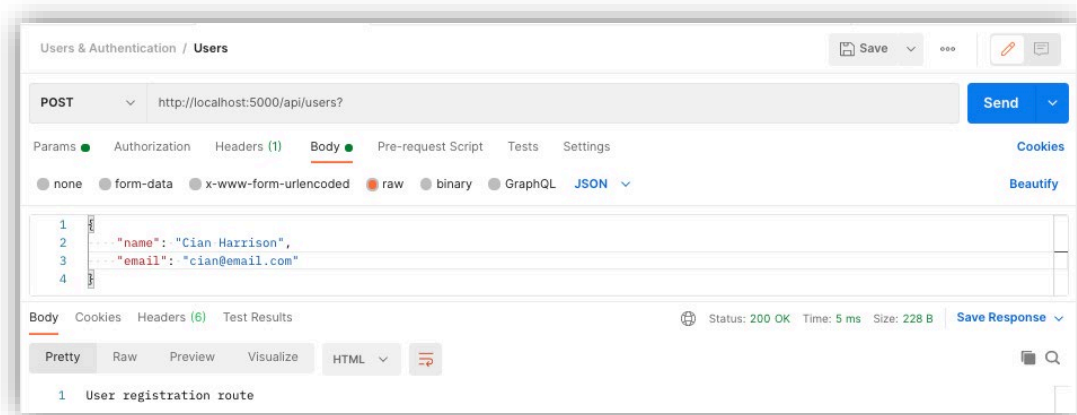


Test_Id	T002
Details	Initial Express API Router Testing
Date	20/1/2021





Test_Id	T003
Req_Id	R01
Details	Initial DB connection & Create User route testing
Date	22/1/2021



```

MongoDB connected
{ name: 'Cian Harrison', email: 'cian@email.com' }

```

Test_Id	T004
Req_Id	R01
Details	Validation constraints for user model creation
Date	22/1/2021

Users & Authentication / Users

POST http://localhost:5000/api/users? Send

Params Authorization Headers (1) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1  {
2  .. "name": "",
3  .. "email": "cian@email",
4  .. "password": ""
5  }

```

Body Cookies Headers (6) Test Results Status: 400 Bad Request Time: 9 ms Size: 519 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2  .. "errors": [
3  ..   {
4  ..     "value": "",
5  ..     "msg": "Name is a required field",
6  ..     "param": "name",
7  ..     "location": "body"
8  ..   },
9  ..   {
10 ..     "value": "cian@email",
11 ..     "msg": "Please enter a valid email",
12 ..     "param": "email",
13 ..     "location": "body"
14 ..   },
15 ..   {
16 ..     "value": "",
17 ..     "msg": "Please Enter a password with a minimum of 8 characters",
18 ..     "param": "password",
19 ..     "location": "body"
20 ..   }
21 .. ]
22 }

```

Users & Authentication / Users

POST http://localhost:5000/api/users? Send

Params Authorization Headers (1) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1  {
2  .. "name": "Cian",
3  .. "email": "cian@email.com",
4  .. "password": "Password"
5  }

```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 16 ms Size: 228 B Save Response

Pretty Raw Preview Visualize HTML

```

1  User registration route

```

Users & Authentication / Register User

POST http://localhost:5000/api/users? Send

Params Authorization Headers (1) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1  {
2  .. "name": "Cian",
3  .. "email": "cian@email.com",
4  .. "password": "Password"
5  }

```

Body Cookies Headers (6) Test Results Status: 400 Bad Request Time: 18 ms Size: 263 B Save Response

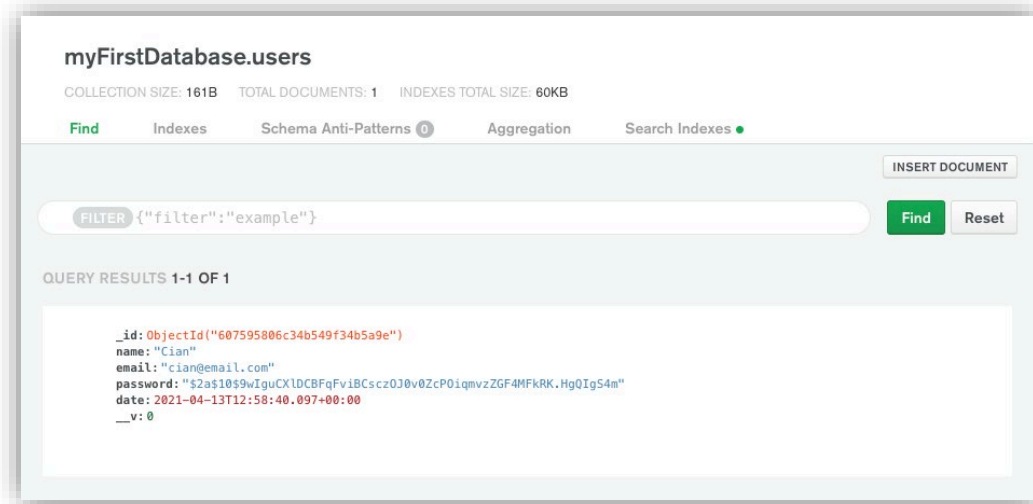
Pretty Raw Preview Visualize JSON

```

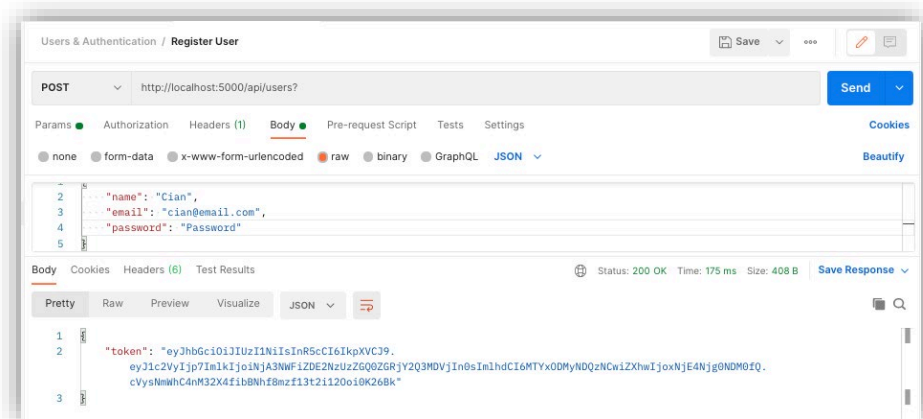
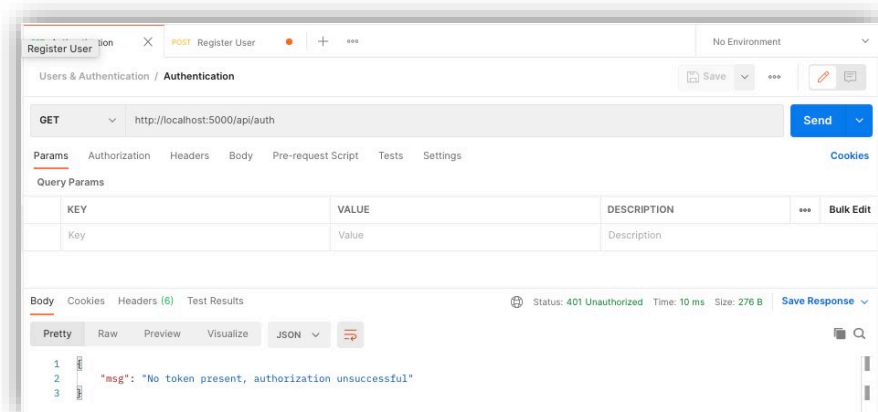
1  {
2  .. "errors": [
3  ..   {
4  ..     "msg": "User already exists"
5  ..   }
6  .. ]
7  }

```

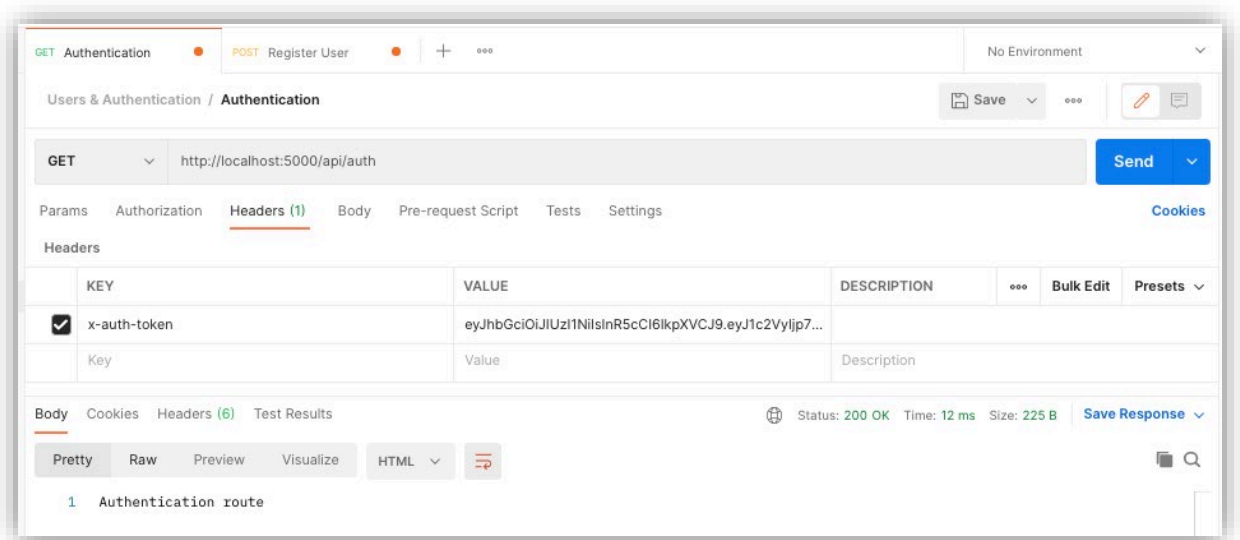
Test_Id	T005
Req_Id	R01
Details	Initial password encryption when user creates account
Date	23/1/2021



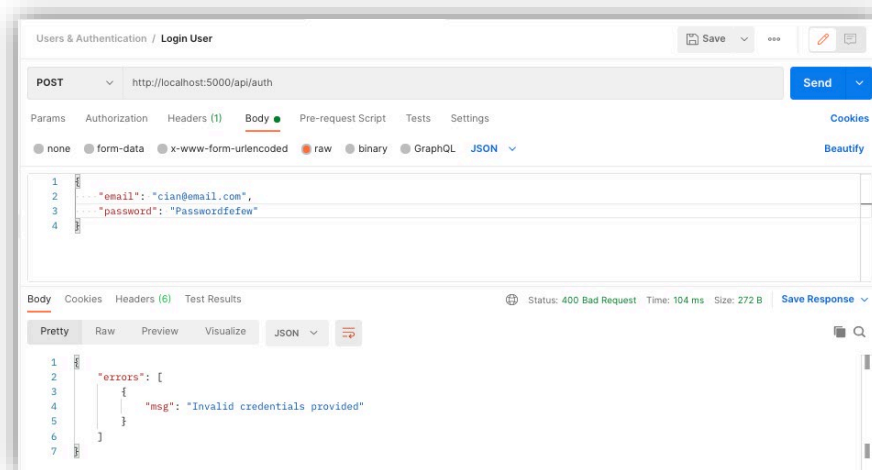
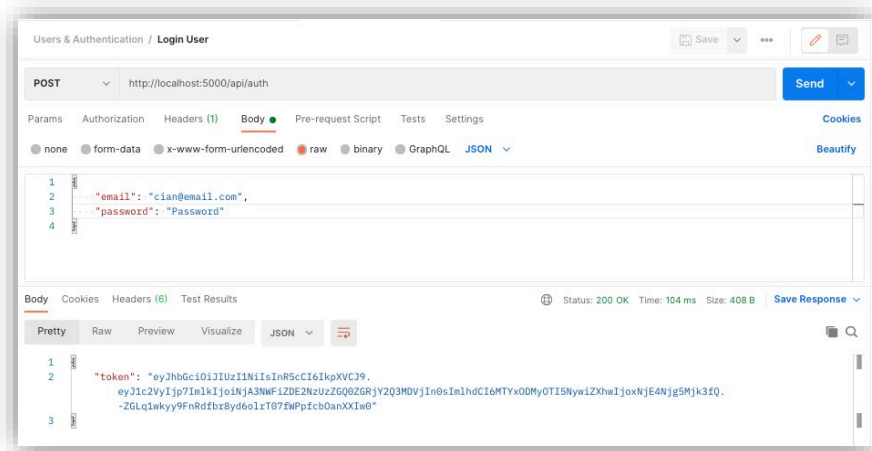
Test_Id	T006
Req_Id	R01
Details	Initial authentication set up for json web token
Date	23/1/2021



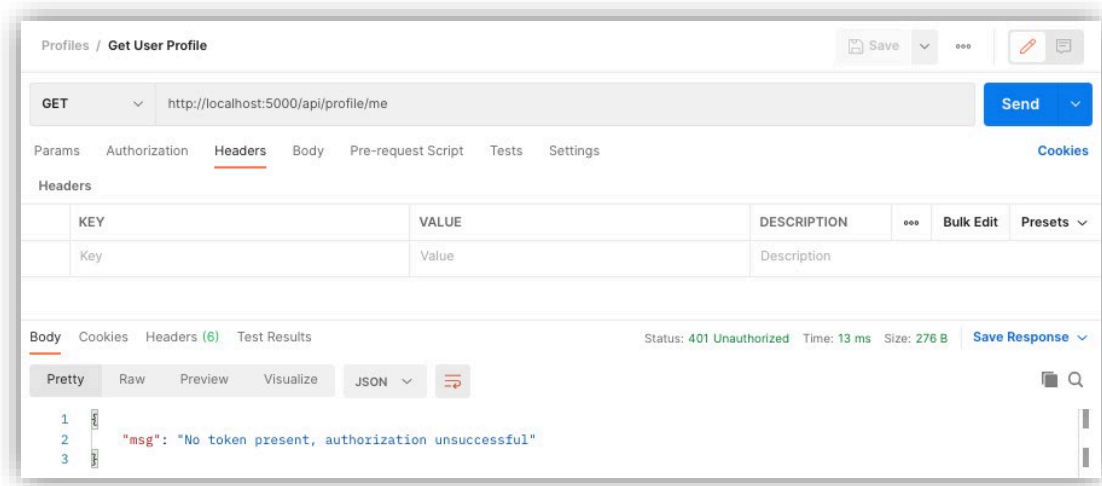
[Token added to the headers of the request and the authentication]



Test_Id	T007
Req_Id	R01
Details	Login existing user with valid credentials
Date	24/1/2021



Test_Id	T008
Req_Id	R02
Details	Getting the current user's profile using JWT in headers
Date	2/2/2021



Profiles / Get User Profile

GET http://localhost:5000/api/profile/me

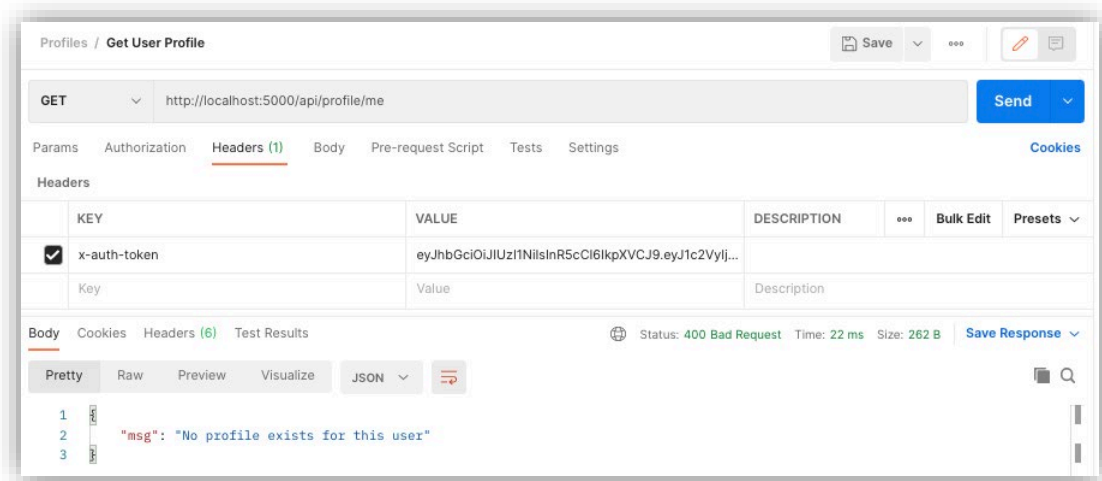
Params Authorization Headers Body Pre-request Script Tests Settings Cookies

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
Key	Value	Description			

Body Cookies Headers (6) Test Results Status: 401 Unauthorized Time: 13 ms Size: 276 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {  
2   "msg": "No token present, authorization unsuccessful"  
3 }
```



Profiles / Get User Profile

GET http://localhost:5000/api/profile/me

Params Authorization Headers (1) Body Pre-request Script Tests Settings Cookies

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vybm9uZCI6ImFkbWUiLCJyb2wiOiJ1b250b290In0.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vybm9uZCI6ImFkbWUiLCJyb2wiOiJ1b250b290In0.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vybm9uZCI6ImFkbWUiLCJyb2wiOiJ1b250b290In0				
Key	Value	Description			

Body Cookies Headers (6) Test Results Status: 400 Bad Request Time: 22 ms Size: 262 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {  
2   "msg": "No profile exists for this user"  
3 }
```

Test_Id	T009
Req_Id	R02
Details	Creating or updating the current user's profile
Date	2/2/2021

http://localhost:5000/api/profile

POST http://localhost:5000/api/profile

Params Authorization Headers (2) **Body** Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1
2  "gender": "Male",
3  "bio": "New to this, looking for any suggestions"
4
5

```

Body Cookies Headers (6) Test Results Status: 400 Bad Request Time: 16 ms Size: 298 B Save Response

Pretty Raw Preview Visualize JSON

```

1
2  "errors": [
3    {
4      "msg": "Age is a required field",
5      "param": "age",
6      "location": "body"
7    }
8  ]
9

```

http://localhost:5000/api/profile

POST http://localhost:5000/api/profile

Params Authorization Headers (2) **Body** Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1
2  "age": 15,
3  "gender": "Male",
4  "bio": "New to this, looking for any suggestions",
5  "youtube": "https://youtube.com",
6  "instagram": "https://instagram.com",
7  "facebook": "https://facebook.com"

```

Body Cookies Headers (6) Test Results Status: 400 Bad Request Time: 127 ms Size: 309 B Save Response

Pretty Raw Preview Visualize JSON

```

1
2  "errors": [
3    {
4      "value": 15,
5      "msg": "Age is a required field",
6      "param": "age",
7      "location": "body"
8    }
9  ]
10

```

http://localhost:5000/api/profile

POST http://localhost:5000/api/profile

Params Authorization Headers (2) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "age": 18,
3   "gender": "Male",
4   "bio": "New to this, looking for any suggestions",
5   "youtube": "https://youtube.com",
6   "instagram": "https://instagram.com",
7   "facebook": "https://facebok.com",
8   "twitter": "https://twitter.com"
9 }
```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 121 ms Size: 558 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "_id": "60760e169b69b75eeba3085b",
3   "user": "6075abd16753dd4ddccd705c",
4   "age": 18,
5   "gender": "Male",
6   "bio": "New to this, looking for any suggestions",
7   "social": {
8     "youtube": "https://youtube.com",
9     "instagram": "https://instagram.com",
10    "facebook": "https://facebok.com",
11    "twitter": "https://twitter.com"
12  },
13   "milestones": [],
14   "date": "2021-04-13T21:33:10.311Z",
15   "__v": 0
16 }
```

Test_Id	T010
Req_Id	R04
Details	Display an array of Checkmate user profiles with limited information
Date	3/2/2021

Profiles / Get All User Profiles

GET http://localhost:5000/api/profile Send

Params Authorization Headers Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (6) Test Results Status: 200 OK Time: 27 ms Size: 1008 B Save Response

Pretty Raw Preview Visualize JSON ...

```

1  {
2    {
3      "social": {
4        "youtube": "https://youtube.com",
5        "instagram": "https://instagram.com",
6        "facebook": "https://facebok.com",
7        "twitter": "https://twitter.com"
8      },
9      "_id": "60760e169b69b75eeba3985b",
10     "user": {
11       "_id": "6075abd16753dd4ddcccd705c",
12       "name": "Cian"
13     },
14     "age": 18,
15     "gender": "Male",
16     "bio": "New to this, looking for any suggestions",
17     "milestones": [],
18     "date": "2021-04-13T21:33:10.311Z",
19     "__v": 0
20   },
21   {
22     "social": {
23       "youtube": "https://youtube.com",
24       "instagram": "https://instagram.com",
25       "facebook": "https://facebok.com",
26       "twitter": "https://twitter.com"
27     },
28     "_id": "6076cb232ff65a780a408ea6",
29     "user": {
30       "_id": "6076ca6c2ff65a780a408ea5",
31       "name": "Sean"
32     },
33     "age": 20,
34     "gender": "Male",
35     "bio": "I have bipolar disorder and I am looking for some help from people have expernced similar emotions",
36     "milestones": [],
37     "date": "2021-04-14T10:59:47.077Z",
38     "__v": 0
39   }
40 }

```

Test_Id	T011
Req_Id	R04
Details	Find a user profile by id
Date	3/2/2021

Profiles / Get Profile By Id

GET `http://localhost:5000/api/profile/user/6075abd16753dd4ddcccd705c` Send

Params Authorization Headers Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (6) Test Results Status: 200 OK Time: 37 ms Size: 580 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "social": {
3      "youtube": "https://youtube.com",
4      "instagram": "https://instagram.com",
5      "facebook": "https://facebok.com",
6      "twitter": "https://twitter.com"
7    },
8    "_id": "60760e169b69b75eeba3085b",
9    "user": {
10     "id": "6075abd16753dd4ddcccd705c",
11     "name": "Cian"
12   },
13   "age": 18,
14   "gender": "Male",
15   "bio": "New to this, looking for any suggestions",
16   "milestones": [],
17   "date": "2021-04-13T21:33:10.311Z",
18   "__v": 0
19 }

```

Test_Id	T012
Req_Id	R02
Details	Delete a user profile and dependent resources
Date	4/2/2021

Profiles / Delete Profile

DELETE `http://localhost:5000/api/profile` Send

Params Authorization Headers (1) Body Pre-request Script Tests Settings Cookies

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp...				
Key	Value	Description			

Body Cookies Headers (6) Test Results Status: 200 OK Time: 38 ms Size: 242 B Save Response

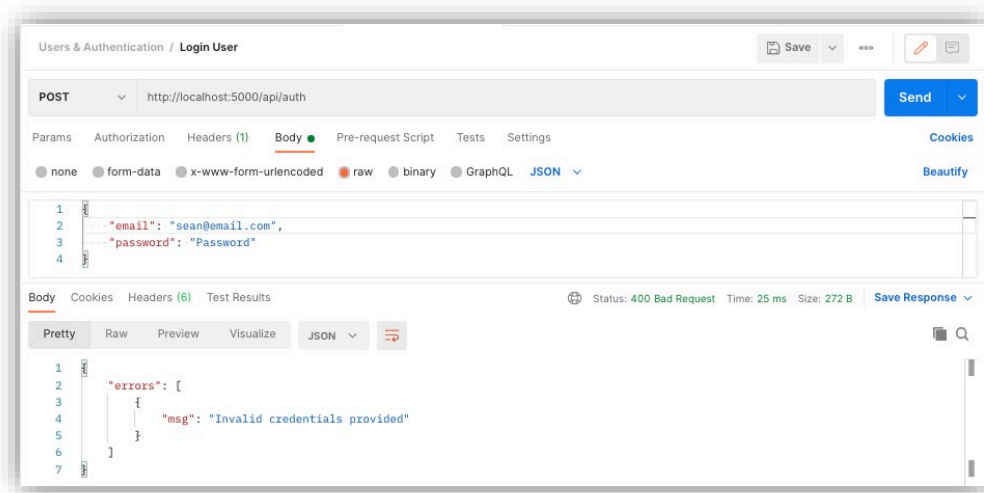
Pretty Raw Preview Visualize JSON

```

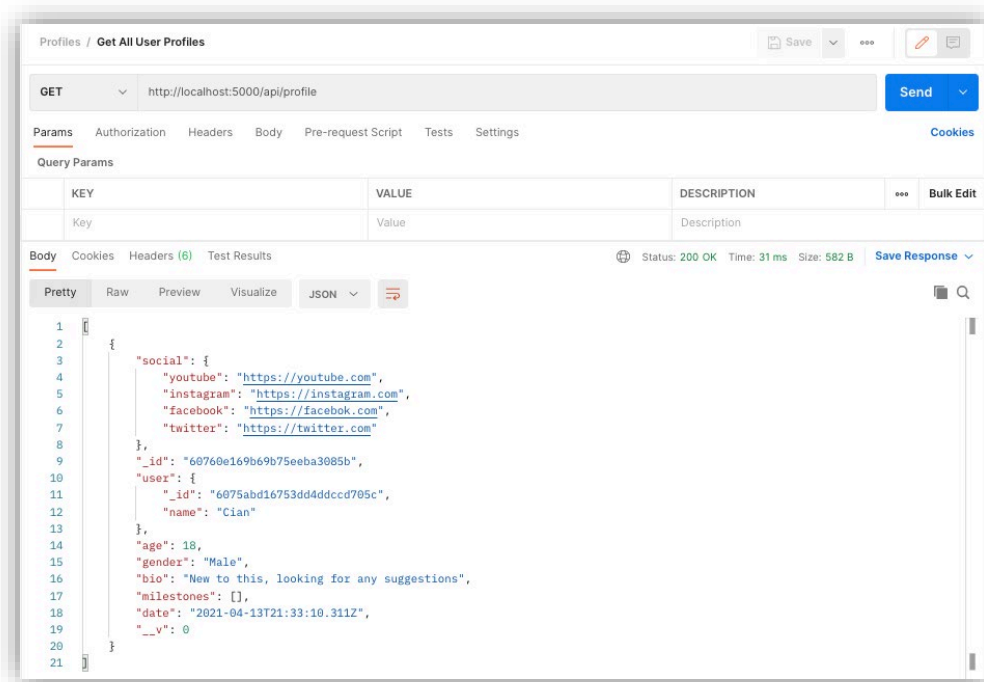
1  {
2    "msg": "User account deleted"
3  }

```

[A User resource has been deleted]



[A Profile resource has been deleted]



Test_Id	T013
Req_Id	R02
Details	Add a milestone to a user profile
Date	6/2/2021

Profiles / Add Milestone to Profile

PUT http://localhost:5000/api/profile/milestone

Params Authorization Headers (2) Body Pre-request Script Tests Settings

Body form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   "title": "My First Checkin",
3   "description": "This milestone marks your first checkin"
4 }

```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 196 ms Size: 717 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "social": {
3     "youtube": "https://youtube.com",
4     "instagram": "https://instagram.com",
5     "facebook": "https://facebok.com",
6     "twitter": "https://twitter.com"
7   },
8   "_id": "60760e169b69b75eeba3085b",
9   "user": "6075abd16753d4ddcc705c",
10  "age": 18,
11  "gender": "Male",
12  "bio": "New to this, looking for any suggestions",
13  "milestones": [
14    {
15      "_id": "60772bfc793a7a83b51820ca",
16      "title": "My First Checkin",
17      "description": "This milestone marks your first checkin",
18      "dateAcheived": "2021-04-14T17:53:00.683Z"
19    }
20  ],
21  "date": "2021-04-13T21:33:10.311Z",
22  "__v": 1
23 }

```

Test_Id	T014
Req_Id	R02
Details	Delete a milestone from a user profile
Date	8/2/2021

Profiles / Delete Milestone From Profile

DELETE http://localhost:5000/api/profile/milestone/60774695793a7a83b51820cb

Params Authorization Headers (1) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
<input type="checkbox"/> ms_id	60774695793a7a83b51820cb		

Body Cookies Headers (6) Test Results Status: 200 OK Time: 142 ms Size: 717 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "social": {
3     "youtube": "https://youtube.com",
4     "instagram": "https://instagram.com",
5     "facebook": "https://facebok.com",
6     "twitter": "https://twitter.com"
7   },
8   "_id": "60760e169b69b75eeba3085b",
9   "user": "6075abd16753d4ddcc705c",
10  "age": 18,
11  "gender": "Male",
12  "bio": "New to this, looking for any suggestions",
13  "milestones": [
14    {
15      "_id": "60772bfc793a7a83b51820ca",
16      "title": "My First Checkin",
17      "description": "This milestone marks your first checkin",
18      "dateAcheived": "2021-04-14T17:53:00.683Z"
19    }
20  ],
21  "date": "2021-04-13T21:33:10.311Z",
22  "__v": 3
23 }

```

Test_Id	T015
Req_Id	R03
Details	Create a user checkin
Date	10/2/2021

Checkins / Create User Checkin

POST http://localhost:5000/api/checkins

Params Authorization Headers (2) Body Pre-request Script Tests Settings

Body

```

1 {
2   "title": "Day 2",
3   "emotion": "Fear",
4   "intensity": 5,
5   "maintext": "No change today but I am determined to build a habit of self reflection through my journal"
6 }

```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 50 ms Size: 505 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "_id": "6077ff107c5f5a9bd2664f32",
3   "user": "6075abd16753d4ddcccd705c",
4   "title": "Day 2",
5   "emotion": "Fear",
6   "intensity": 5,
7   "maintext": "No change today but I am determined to build a habit of self reflection through my journal",
8   "reactions": [],
9   "comments": [],
10  "date": "2021-04-15T08:53:36.587Z",
11  "_v": 0
12 }

```

Test_Id	T016
Req_Id	R03
Details	Get all user checkins
Date	10/2/2021

Checkins / Get All User Checkins

GET http://localhost:5000/api/checkins

Params Authorization Headers (1) Body Pre-request Script Tests Settings

Headers

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vybm9udGVzLnR5cCI6IkpXVCJ9.eyJ1c2Vybm9udGVzLnR5cCI6IkpXVCJ9	

Body Cookies Headers (6) Test Results Status: 200 OK Time: 20 ms Size: 828 B Save Response

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "_id": "6077ff107c5f5a9bd2664f32",
4     "user": "6075abd16753d4ddcccd705c",
5     "title": "Day 2",
6     "emotion": "Fear",
7     "intensity": 5,
8     "maintext": "No change today but I am determined to build a habit of self reflection through my journal",
9     "reactions": [],
10    "comments": [],
11    "date": "2021-04-15T08:53:36.587Z",
12    "_v": 0
13  },
14  {
15    "_id": "607761177dba9e8c07606a58",
16    "user": "6075abd16753d4ddcccd705c",
17    "title": "Fist digital diary checkin",
18    "emotion": "Fear",
19    "intensity": 5,
20    "maintext": "I am feeling a little overwhelmed lately. I am afraid of losing my job due to COVID-19 restrictions",
21    "reactions": [],
22    "comments": [],
23    "date": "2021-04-14T21:39:35.732Z",
24    "_v": 0
25  }
26 ]

```


Test_Id	T017
Req_Id	R03
Details	Get Checkin by Id
Date	10/2/2021

Checkins / Get Checkin By Id

GET http://localhost:5000/api/checkins/60780252a244659cb7490510

Headers (1)

KEY	VALUE	DESCRIPTION
x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vybm9keiI6ImN1b3R5IiwiaWF0IjoiMjAyMS00L15TOD99	

Body

```

1  {
2    "_id": "60780252a244659cb7490510",
3    "user": "6075abd16753dd4ddccd705c",
4    "title": "Day 3",
5    "emotion": "Anticipation",
6    "intensity": 4,
7    "maintext": "Still a little anxious but starting to feel a sense of anticipation about the future.",
8    "name": "Cian",
9    "reactions": [],
10   "comments": [],
11   "date": "2021-04-15T09:07:30.884Z",
12   "__v": 0
13 }

```

Status: 200 OK Time: 17 ms Size: 521 B

[Invalid Object Id]

Checkins / Get Checkin By Id

GET http://localhost:5000/api/checkins/1

Headers (1)

KEY	VALUE	DESCRIPTION
x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vybm9keiI6ImN1b3R5IiwiaWF0IjoiMjAyMS00L15TOD99	

Body

```

1  {
2    "msg": "Checkin not found"
3  }

```

Status: 404 Not Found Time: 5 ms Size: 246 B

```

Server started on port ${PORT} 5000
MongoDB connected
Cast to ObjectId failed for value "60780252a244659cb7490510" at path "_id" for model "checkin"
Cast to ObjectId failed for value "1" at path "_id" for model "checkin"

```

Test_Id	T018
Req_Id	R03
Details	Delete User Checkin
Date	13/2/2021

Checkins / Delete Checkin

DELETE http://localhost:5000/api/checkins/60780252a244659cb7490510

Params Authorization Headers (1) Body Pre-request Script Tests Settings Cookies

Headers

KEY	VALUE	DESCRIPTION	Bulk Edit	Presets
<input checked="" type="checkbox"/> x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7...			
Key	Value	Description		

Body Cookies Headers (6) Test Results Status: 200 OK Time: 50 ms Size: 258 B Save Response

Pretty Raw Preview Visualize JSON

```

1
2  "msg": "Checkin has been pezmanently deleted"
3

```

[Invalid ObjectId]

Checkins / Delete Checkin

DELETE http://localhost:5000/api/checkins/1

Params Authorization Headers (1) Body Pre-request Script Tests Settings Cookies

Headers

KEY	VALUE	DESCRIPTION	Bulk Edit	Presets
<input checked="" type="checkbox"/> x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjp7...			
Key	Value	Description		

Body Cookies Headers (6) Test Results Status: 404 Not Found Time: 6 ms Size: 246 B Save Response

Pretty Raw Preview Visualize JSON

```

1
2  "msg": "Checkin not found"
3

```

```

Server started on port ${PORT} 5000
MongoDB connected
Cast to ObjectId failed for value "1" at path "_id" for model "checkin"

```

Test_Id	T019
Req_Id	R03
Details	Add support reaction to friends checkin
Date	15/2/2021

The screenshot shows a REST client interface for a PUT request to `http://localhost:5000/api/checkins/reaction/6077ff107c5f5a9bd2664f32`. The request includes an `x-auth-token` header. The response is a JSON object with the following structure:

```

1  {
2    "_id": "607823a10ac836a1f5dbb898",
3    "user": "6075abd16753dd4ddccd705c"
4  }
5
6

```

The status is 200 OK, with a response time of 86 ms and a size of 282 B.

[Trying to add a second reaction for a user]

The screenshot shows the same REST client interface, but the request has failed. The status is 400 Bad Request, with a response time of 16 ms and a size of 265 B. The response body contains the following JSON object:

```

1  {
2    "msg": "User can only react to a post once"
3  }

```

Test_Id	T020
Req_Id	R05
Details	Add comment to friends checkin
Date	15/2/2021

Checkins / Add Comment To Checkin

POST http://localhost:5000/api/checkins/comment/6077ff107c5f5a9bd2664f32

Params Authorization Headers (2) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body Cookies Headers (6) Test Results Status: 200 OK Time: 238 ms Size: 392 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   {
3     "_id": "60787ad19bf1dbab86d74601",
4     "user": "6075abd16753dd4ddccd705c",
5     "text": "You are doing great! Reach out if you need anything",
6     "name": "Cian",
7     "date": "2021-04-15T17:41:37.941Z"
8   }
9 }

```

Checkins / Get All User Checkins

GET http://localhost:5000/api/checkins

Params Authorization Headers (1) Body Pre-request Script Tests Settings Cookies

Headers

KEY	VALUE	DESCRIPTION	Bulk Edit	Presets
<input checked="" type="checkbox"/> x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vybm9p7...			
Key	Value	Description		

Body Cookies Headers (6) Test Results Status: 200 OK Time: 37 ms Size: 1.05 KB Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   {
3     "_id": "6077ff107c5f5a9bd2664f32",
4     "user": "6075abd16753dd4ddccd705c",
5     "title": "Day 2",
6     "emotion": "Fear",
7     "intensity": 5,
8     "maintext": "No change today but I am detexmined to build a habbit of self reflection through my journal",
9     "reactions": [
10      {
11        "_id": "607823a10ac836a1f5dbb898",
12        "user": "6075abd16753dd4ddccd705c"
13      }
14    ],
15     "comments": [
16      {
17        "_id": "60787ad19bf1dbab86d74601",
18        "user": "6075abd16753dd4ddccd705c",
19        "text": "You are doing great! Reach out if you need anything",
20        "name": "Cian",
21        "date": "2021-04-15T17:41:37.941Z"
22      }
23    ]
24   }
25 }

```

Test_Id	T021
Req_Id	R05
Details	Remove comment from friends checkin
Date	15/2/2021

Checkins / Add Comment To Checkin

POST http://localhost:5000/api/checkins/comment/6077ff107c5f5a9bd2664f32

Params Authorization Headers (2) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1
2  {
3   "text": "TEST COMMENT"
4 }

```

Body Cookies Headers (6) Test Results Status: 200 OK Time: 166 ms Size: 532 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "comment": {
3     "_id": "60788221caaf65acdfe7f4f4",
4     "user": "6075abd16753dd4ddccd705c",
5     "text": "TEST COMMENT",
6     "name": "Cian",
7     "date": "2021-04-15T18:12:49.568Z"
8   },
9   "comment": {
10    "_id": "60787ad19bf1dbab86d74601",
11    "user": "6075abd16753dd4ddccd705c",
12    "text": "You are doing great! Reach out if you need anything",
13    "name": "Cian",
14    "date": "2021-04-15T17:41:37.941Z"
15  }
16 }

```

Checkins / Delete Comment From Checkin

DELETE http://localhost:5000/api/checkins/comment/6077ff107c5f5a9bd2664f32/60788221caaf65acdfe7f4f4

Params Authorization **Headers (1)** Body Pre-request Script Tests Settings

Headers

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vybm9uZCI6ImNpYW50dW50IiwiaWF0Ijoi				
Key	Value	Description			

Body Cookies Headers (6) Test Results Status: 200 OK Time: 121 ms Size: 392 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "comment": {
3     "_id": "60787ad19bf1dbab86d74601",
4     "user": "6075abd16753dd4ddccd705c",
5     "text": "You are doing great! Reach out if you need anything",
6     "name": "Cian",
7     "date": "2021-04-15T17:41:37.941Z"
8   }
9 }

```

Test_Id	T022
Req_Id	R01
Details	User form validation – front end
Date	18/2/2021

Register with Checkmates

Cian Harrison

cianharry@gmail.com

This site uses Gravatar so if you want a profile image, use a Gravatar email

.....

.....

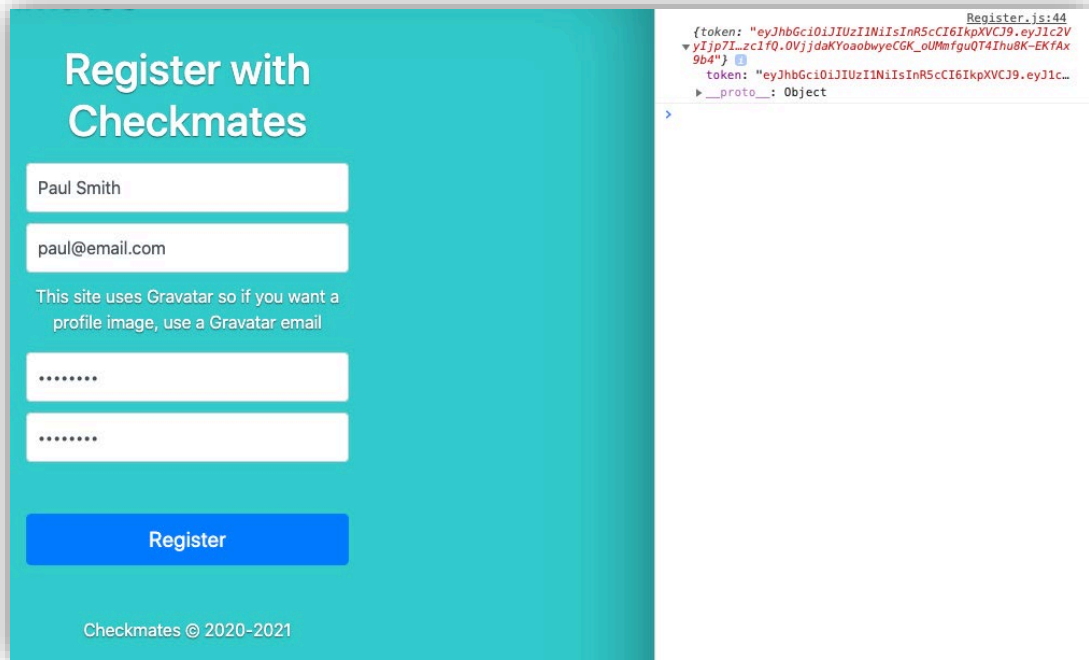
Register

Passwords do not match Register.js:21

```
Register.js:24
{name: "Cian Harrison", email: "cianharry@gmail.com", password: "12345678", password2: "12345678"}
  name: "Cian Harrison"
  password: "12345678"
  password2: "12345678"
  __proto__: Object
```

```
const onSubmit = async e => {
  e.preventDefault();
  // Checking that the passwords match
  if(password !== password2) {
    console.log('Passwords do not match')
  }
  else {
    console.log(formData);
  }
}
```

Test_Id	T023
Req_Id	R01
Details	Token validation on successful registration
Date	18/2/2021



[Test code to check that registration is being handled correctly by the front end and routed successfully to the back-end API]

```
const onSubmit = async e => {
  e.preventDefault();
  // Checking that the passwords match
  if(password !== password2) {
    console.log('Passwords do not match')
  }
  else {
    //console.log(formData);
    const newUser = {
      name,
      email,
      password
    }
    //
    try {
      // configuring the headers for the request
      const config = {
        headers: {
          'Content-Type': 'application/json'
        }
      }
      // configuring the request body to be sent
      const body = JSON.stringify(newUser);
      // await the axios post request to backend api
      const res = await axios.post('/api/users', body, config);
      // log the response data - json web token
      console.log(res.data)
    } catch (error) {
      console.error(error.response.data);
    }
  }
}
```

Test_Id	T024
Req_Id	R01
Details	REDUX alert reducer
Date	21/2/2021

The screenshot shows the registration page of 'Checkmates' with the following details:

- Navigation: Dashboard, Checkins, Register, Login
- Form Fields:
 - Name: Cian TEST
 - Email: test@test.com
 - Message: "This site uses Gravatar so if you want a profile image, use a Gravatar email"
 - Two password fields (both masked with dots)
- Buttons: Register, Already have an account?, Login
- Footer: Checkmates © 2020-2021

On the right, Redux DevTools shows the initial state with an `@@INIT` action at 11:47:27.16. The state is empty, and the message "(states are equal)" is displayed.

The screenshot shows the registration page with the following details:

- Form Fields:
 - Name: Cian Harrison
 - Email: test@test.com
 - Message: "This site uses Gravatar so if you want a profile image, use a Gravatar email"
 - Two password fields (both masked with dots)
- Buttons: Register, Already have an account?, Login
- Footer: Checkmates © 2020-2021

On the right, Redux DevTools shows a sequence of `SET_ALERT` actions:

- 2:48:16.30
- +00:19.75
- +04:56.88
- +00:45.24

The state is expanded to show:

```

alerts (array)
  0 (array) [-]
  1 (array) [-]
  2 (array)
    msg (string): "Passwords do not match"
    alertType (string): "danger"
    id (string): "c3bae11f-7efb-40b3-9dd8-4f8f3a7df3b2"
  
```

The console shows a warning: `src/actions/alerts.js react_devtools_backend.js:2557` with the message: `Line 3:21: 'REMOVE_ALERT' is defined but never used no-unused-vars`. Another warning from `src/reducers/alerts.js` states: `Line 7:1: Unexpected default export of anonymous function import/no-anonymous-default-export`. The message "Passwords do not match" is also visible in the console.

Test_Id	T025
Req_Id	R01
Details	Test of removeAlert timeout
Date	21/2/2021

The screenshot shows the registration page of 'Checkmates' with a red error banner at the top stating 'Passwords do not match'. The registration form contains the following fields: Name (Cian Harrison), Email (cianharry@gmail.com), and two password fields (both masked with dots). A blue 'Register' button is visible below the form. The Redux DevTools interface on the right shows a log with the following entries:

- filter...
- @@INIT (4:26:24.48)
- SET_ALERT (+00:10.86)

The Redux state tree shows an array of alerts:

```

alerts (pin)
  0 (pin)
    msg (pin): "Passwords do not match"
    alertType (pin): "danger"
    id (pin): "405e192d-309a-4b17-84ec-7d8767b9ea02"

```

The screenshot shows the registration page of 'Checkmates' with a URL bar indicating 'https://serato.com'. The registration form contains the following fields: Name (Cian Harrison), Email (cianharry@gmail.com), and two password fields (both masked with dots). A blue 'Register' button is visible below the form. The Redux DevTools interface on the right shows a log with the following entries:

- filter...
- @@INIT (4:52:23.11)
- SET_ALERT (+00:08.76)
- REMOVE_ALERT (+00:05.00)
- SET_ALERT (+00:20.45)
- REMOVE_ALERT (+00:05.00)

The Redux state tree shows an array of alerts:

```

alerts (pin)
  0 (pin): {msg: "Passwords...cbd"}

```

Test_Id	T026
Req_Id	R01
Details	Registration failure reducer alerts
Date	25/2/2021

The screenshot displays a web application interface for registration. The page has a teal background and a navigation bar with links for Dashboard, Checkins, Register, and Login. The main heading is "Register with Checkmates". Below the heading are four input fields: "Full name", "Email address", "Password", and "Confirm password". A blue "Register" button is positioned below the inputs. Underneath the button are links for "Already have an account?" and "Login". At the bottom, it says "Checkmates © 2020-2021".

Three red error messages are displayed above the form:

- "Name is a required field"
- "Please enter a valid email"
- "Please Enter a password with a minimum of 8 characters"

On the right side, the Redux DevTools inspector is open, showing a list of actions: @@INIT, SET_ALERT, SET_ALERT, SET_ALERT, and REGISTRATION_FAILURE. The "Tree" tab is selected, showing the state structure:

```

alerts (pin)
  0 (pin)
    msg (pin): "Name is a required field"
    alertType (pin): "danger"
    id (pin): "21e3e49d-ca10-44d5-adc7-3ada42743af6"
  1 (pin)
    msg (pin): "Please enter a valid email"
    alertType (pin): "danger"
    id (pin): "c2f996a8-a626-41fd-8051-7d5a78095323"
  2 (pin)
    msg (pin): "Please Enter a password with a minimum of 8 characters"
    alertType (pin): "danger"
    id (pin): "954ff187-7516-4f55-b8e6-074ca9a5b578"
auth (pin)
  token (pin): null
  isAuthenticated (pin): false
  loading (pin): false
  user (pin): null
  
```

Test_Id	T027
Req_Id	R01
Details	Registration success reducer alerts
Date	25/2/2021

The screenshot shows a web application interface for 'Checkmates' with a registration form and a Redux DevTools window. The form includes fields for name, email, and password, along with a 'Register' button and links for 'Already have an account?' and 'Login'. The Redux DevTools window displays a 'REGISTRATION_SUCCESS' alert with a timestamp of 10:04:12.27 and a duration of +00:26.06. The state tree shows the 'auth' object with a 'token', 'isAuthenticated: true', 'loading: false', and 'user: null'.

The screenshot shows a MongoDB query results page. The left sidebar displays the database structure with 'users' selected. The main area shows the query results for three users. The filter used is {"filter":"example"}. The query results are as follows:

Document
<pre> { "_id": ObjectId("6075abd16753dd4ddccd705c"), "name": "Cian", "email": "cian@email.com", "password": "\$2a\$10\$Rphsd.Lmy1Xs78.ux8yq9UhhHQU2IpILBXEmwKrdNpzzSscvfik", "date": 2021-04-13T14:33:53.896+00:00, "__v": 0 } </pre>
<pre> { "_id": ObjectId("607ac1ee808900fdd9c32414"), "name": "Paul Smith", "email": "paul@email.com", "password": "\$2a\$10\$SPFHNq/1hcfxygJjLk56KugxtF0RzLyjvG/mQ2PTGCn0CiE1o5nn0", "date": 2021-04-17T11:09:34.964+00:00, "__v": 0 } </pre>
<pre> { "_id": ObjectId("607df06633dca15278ed38da"), "name": "John Walsh", "email": "john@email.com", "password": "\$2a\$10\$20yHVRGLbTB30.CdN.0.wem.PyYbSnstur1JNK5dY7QcebGIFMA6W", "date": 2021-04-19T21:04:38.189+00:00, "__v": 0 } </pre>

Test_Id	T028
Req_Id	R01
Details	Get current user reducer alert
Date	25/2/2021

The image shows a web application interface on the left and a Redux DevTools inspector on the right.

Web Application Interface:

- Header: Checkmates Dashboard Checkins Register Login
- Main heading: Register with Checkmates
- Form fields: Full name, Email address, Password, Confirm password
- Register button
- Links: Already have an account? Login
- Footer: Checkmates © 2020-2021

Redux DevTools Inspector:

- Inspector: Welcome to Checkm8s
- filter... Commit
- @@INIT 12:04:45.60
- USER_PRESENT +00:00.14
- State: Action State Diff Trace Test
- Tree: Chart Raw
- alerts (pin):
- auth (pin)
 - token (pin): "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjpbImkIjoieXNjA3ZTBjNjRmMzIybnTE1NzU2NDM3OTUzIn0sImhhdCI6MTYxODg3MzQ0N0N0cWlZKhwIj09NjE5SMjMzNDQ0fQ.tdAZtL5Fa7t1bfDzty6joSLorK6LZ5fL7YFn0Yz28K8"
 - isAuthenticated (pin): true
 - loading (pin): false
- user (pin)
 - _id (pin): "607e0c64f3225157e6437956"
 - name (pin): "Token TEST"
 - email (pin): "token@test.com"
 - date (pin): "2021-04-19T23:04:04.763Z"
 - __v (pin): 0

Test_Id	T029
Req_Id	R01
Details	Login failure reducer action alerts
Date	30/2/2021

The screenshot displays a web application interface for 'Checkmates' with a teal background. At the top, there are navigation links: 'Dashboard', 'Checkins', 'Register', and 'Login'. A red error banner at the top center reads 'Invalid credentials provided'. Below this, the main heading is 'Login to Checkmates'. The login form contains an email input field with 'cianharry@gmail.com', a password input field with masked characters, and a blue 'Login' button. Below the button are links for 'Don't have an account?' and 'Register now'. At the bottom, it says 'Checkmates © 2020-2021'.

On the right side, a Redux DevTools window is open, showing the Redux state. The state is as follows:

```

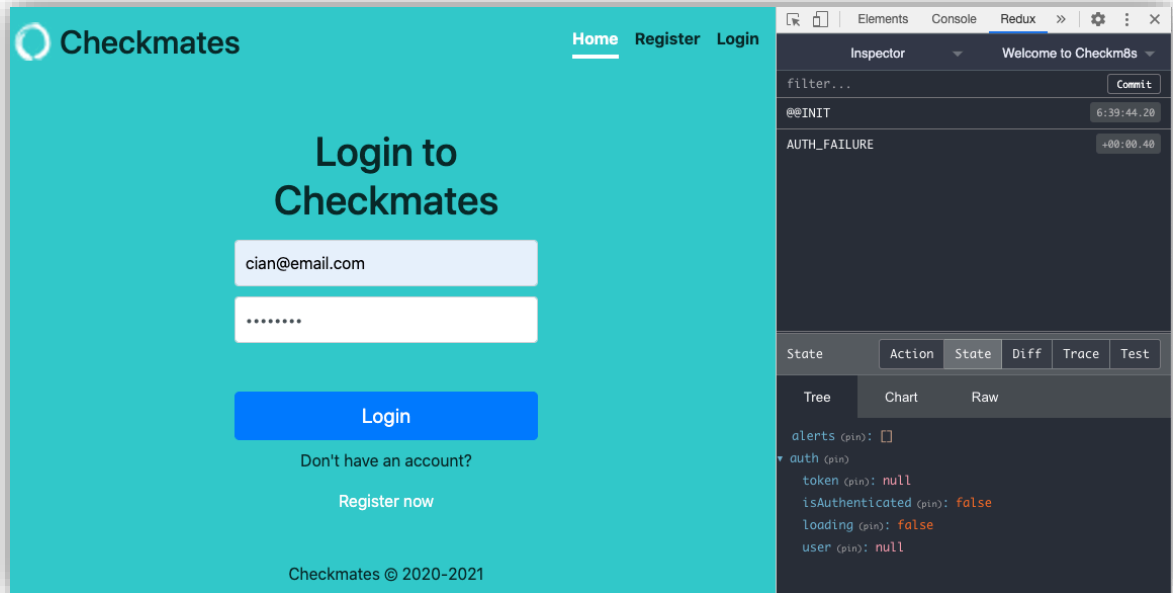
{
  alerts: [
    {
      msg: "Invalid credentials provided",
      alertType: "danger",
      id: "da2777ed-8bf4-4af3-8a12-bf17f12ee730"
    }
  ],
  auth: {
    token: null,
    isAuthenticated: false,
    loading: false,
    user: null
  }
}

```

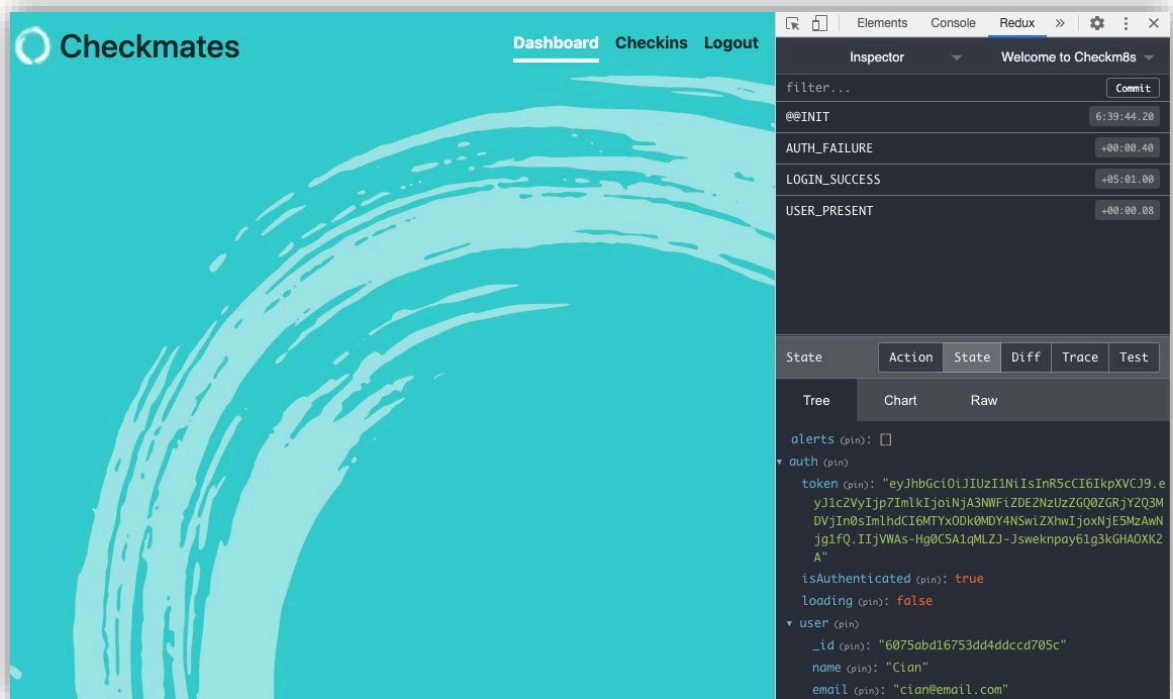
The Redux DevTools window also shows a list of actions in the 'Inspector' tab:

- @@INIT (10:47:06.11)
- AUTH_FAILURE (+00:00.15)
- SET_ALERT (+00:07.59)
- LOGIN_FAILURE (+00:00.00)

Test_Id	T031
Req_Id	R01
Details	Navbar – unauthenticated/guest user
Date	4/3/2021



Test_Id	T032
Req_Id	R01
Details	Navbar – authenticated user
Date	4/3/2021



Test_Id	T033
Req_Id	R01
Details	Navbar – logout auth reducer action
Date	4/3/2021

The screenshot shows the Redux DevTools interface for the Checkmates application. The application is in the 'Home' state. The Redux state tree is visible, showing the following structure:

```

alerts (pin): []
auth (pin): {
  token (pin): null
  isAuthenticated (pin): false
  loading (pin): false
}
user (pin): {
  _id (pin): "6075abd16753dd4ddccd705c"
  name (pin): "Cian"
  email (pin): "cian@email.com"
}

```

Test_Id	T034
Req_Id	R02
Details	Get user profile reducer action
Date	6/3/2021

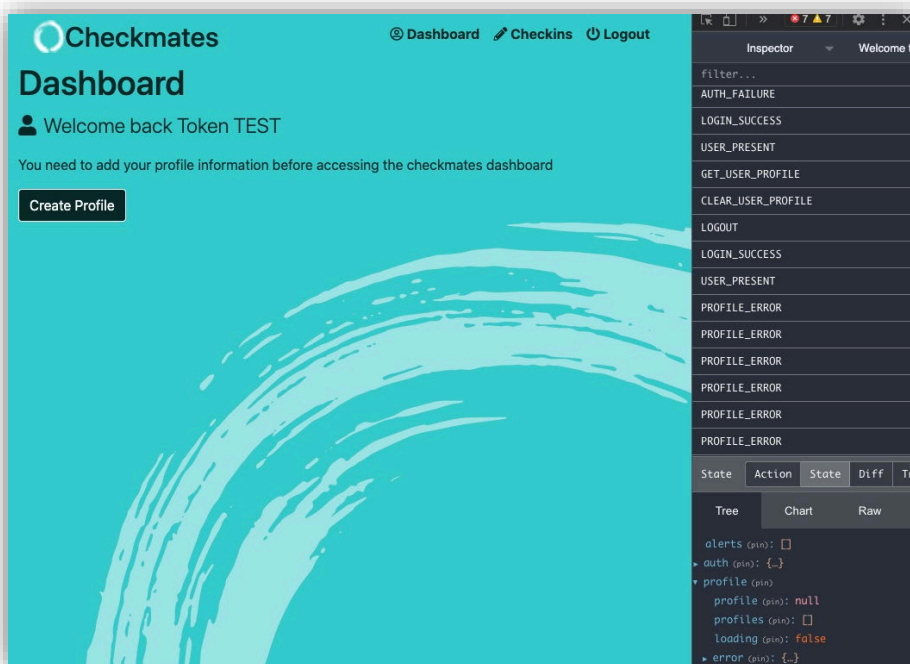
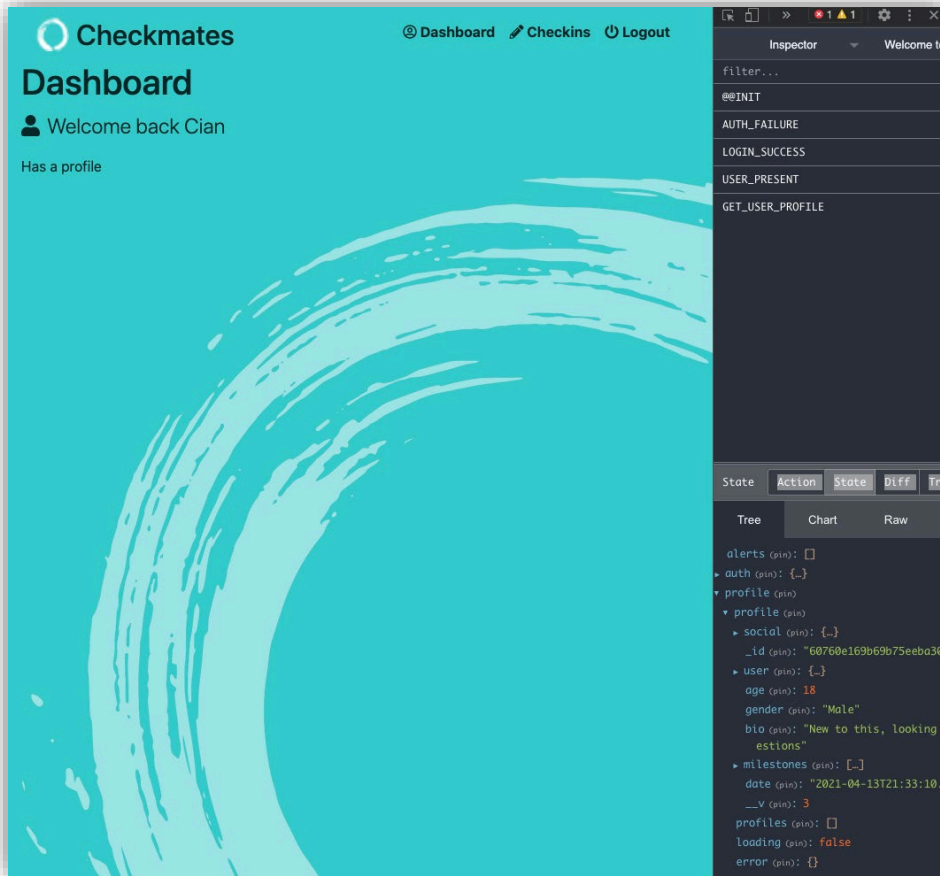
The screenshot shows the Redux DevTools interface for the Checkmates application. The application is in the 'Dashboard' state. The Redux state tree is visible, showing the following structure:

```

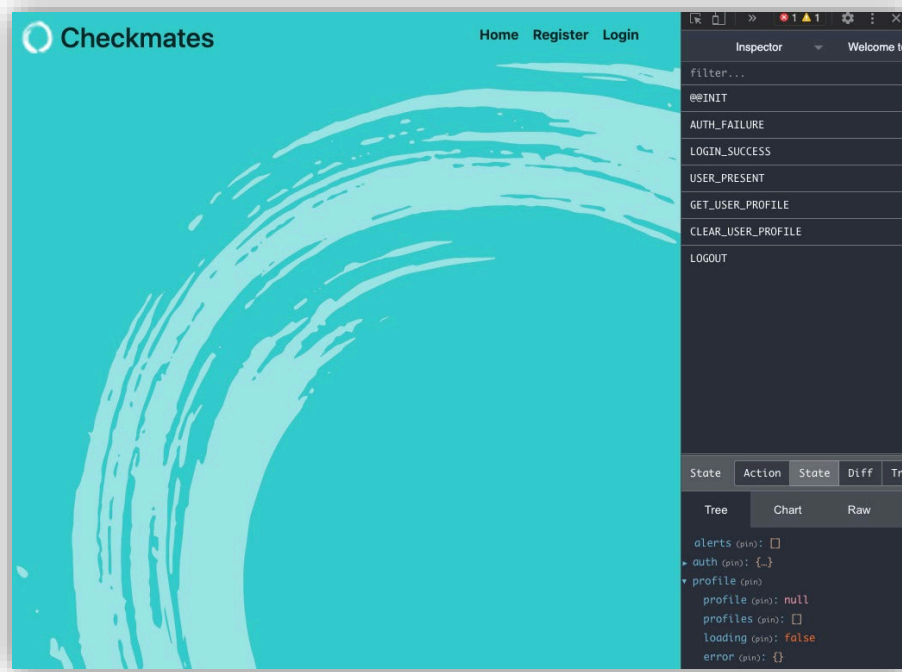
alerts (pin): []
auth (pin): {}
profile (pin): {
  profile (pin): {
    social (pin): {}
    _id (pin): "60760e169b69b75eeba3085b"
  }
  user (pin): {
    age (pin): 18
    gender (pin): "Male"
    bio (pin): "New to this, looking for any suggestions"
  }
  milestones (pin): []
}

```

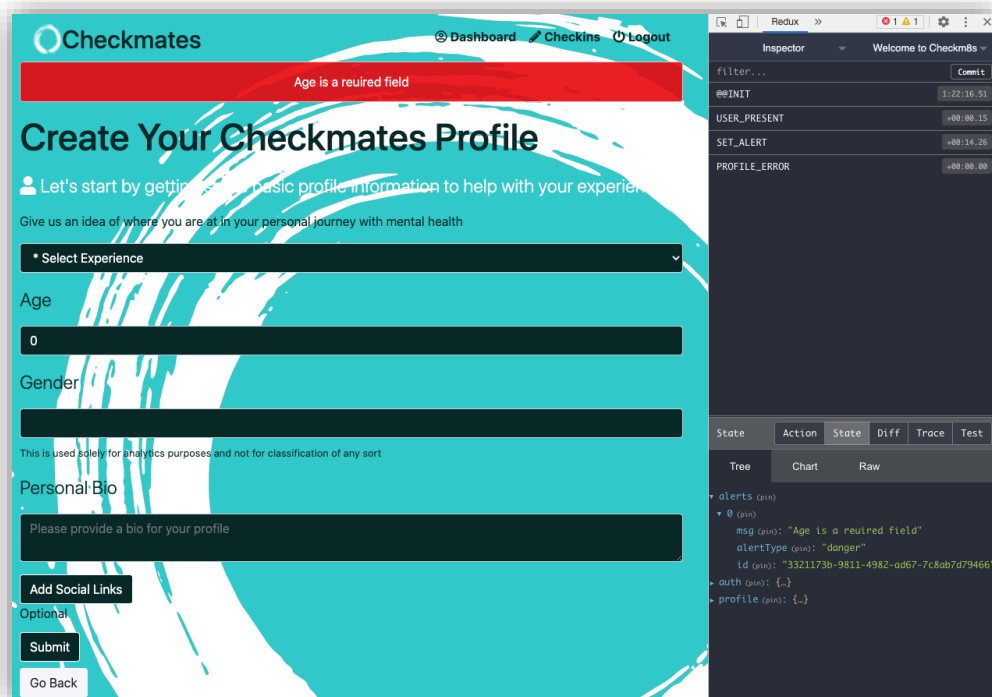

Test_Id	T035
Req_Id	R02
Details	Verifying user profile is present to access dashboard
Date	10/3/2021



Test_Id	T036
Req_Id	R02
Details	Clearing user profile from state on logout
Date	10/3/2021



Test_Id	T037
Req_Id	R02
Details	Profile from validation and error alerts
Date	13/3/2021



Test_Id	T038
Req_Id	R02
Details	Form submission and profile creation
Date	13/3/2021

The screenshot shows the Checkmates application dashboard. At the top, there is a navigation bar with 'Dashboard', 'Checkins', and 'Logout' links. A green notification banner at the top of the main content area displays the message 'Profile has been created'. Below this, the dashboard title 'Dashboard' is followed by a user greeting 'Welcome back Token TEST' and a status indicator 'Has a profile'. The main content area contains a profile creation form with fields for 'Name', 'Age', 'Gender', 'Bio', 'Milestones', 'Date', 'Experience', and 'Social'. The Redux DevTools interface is open on the right side, showing the Redux state. The state object includes 'alerts', 'auth', 'profile', and 'loading'. The 'profile' object is expanded, showing its properties: 'social', '_id', 'user', 'age', 'gender', 'bio', 'milestones', 'date', '_v', 'experience', 'profiles', 'loading', and 'error'.

```

>
  _id: ObjectId("60816cc36846276ca6d64a0f")
  user: ObjectId("607e0c64f3225157e6437956")
  age: 20
  gender: "Male"
  bio: "I have been struggling through the lockdown period and I want to find ..."
  milestones: Array
    date: 2021-04-22T12:32:03.952+00:00
    __v: 0
  social: Object
    experience: "Beginner"

```

Test_Id	T039
Req_Id	R02
Details	Edit profile action and alert
Date	13/3/2021

The screenshot displays the 'Create Your Checkmates Profile' page. At the top, a green notification bar states 'Profile has been updated'. The form includes fields for 'Experience' (set to 'Intermediate'), 'Age' (25), and 'Gender' (Male). A 'Personal Bio' section contains the text: 'I have been struggling through the lockdown period and I want to find a way of understanding my emotions through journaling'. Below this are 'Optional' social media links for Twitter, Facebook, YouTube, and Instagram. A 'Submit' button is at the bottom left, and a 'Go Back' button is at the bottom right.

On the right side, the Redux DevTools inspector shows the state after the profile update. The state object is as follows:

```

{
  alerts: [
    {
      msg: "Profile has been updated",
      alertType: "success",
      id: "ea4cabc9-5eda-47ff-961b-289a88969231"
    }
  ],
  auth: {},
  profile: {
    social: {},
    _id: "60816cc36846276ca6d64a0f",
    user: "607e0c64f3225157e6437956",
    age: 25,
    gender: "Male",
    bio: "I have been struggling through the lockdown period and I want to find a way of understanding my emotions through journaling",
    milestones: [],
    date: "2021-04-22T12:32:03.952Z",
    __v: 0,
    experience: "Intermediate"
  },
  profiles: [],
  loading: false,
  error: {}
}

```

Test_Id	T040
Req_Id	R02
Details	Add milestone to profile action and reducer alerts
Date	18/3/2021

The screenshot displays a web application interface on the left and the Redux DevTools inspector on the right. The application shows a notification 'Milestone added to User Profile', a dashboard with a welcome message 'Welcome back Cian', and navigation buttons for 'Edit Profile', 'Milestones', and 'Checkins'. The Redux DevTools inspector shows the state of the application, including a bio, milestones array, and user profile information.

```

    _id: ObjectId("60760e169b69b75eeba3085b")
  user: ObjectId("6075abd16753dd4ddccd705c")
  age: 18
  gender: "Male"
  bio: "New to this, looking for any suggestions"
  > social: Object
  > milestones: Array
  > 0: Object
  >   _id: ObjectId("6082eba85084839e9b31c804")
  >   title: "1 Month Checkin Streak"
  >   description: "This milestones represents a big change in my life. I have been checki..."
  >   privacy: true
  >   dateAcheived: 2021-04-23T15:45:44.765+00:00
  > 1: Object
  > 2: Object
  date: 2021-04-13T21:33:10.311+00:00
  __v: 5
  experience: "Intermediate"

```

Test_Id	T041
Req_Id	R02
Details	Profile from validation and error alerts
Date	18/3/2021

The screenshot shows the 'Add Milestone to Profile' form in the Checkmates application. Two red error messages are displayed at the top: 'Milestone title is required' and 'Milestone description is required'. The form includes input fields for 'Milestone Title' and 'Milestone Description', a checkbox for 'Select if you want this milestone to be private and not visible on your profile', and 'Submit' and 'Go Back' buttons. To the right, the Redux DevTools log shows two error actions: '@@INIT' and two 'PROFILE_ERROR' actions with messages 'Milestone title is required' and 'Milestone description is required'.

Test_Id	T042
Req_Id	R02
Details	List Profile milestones
Date	18/3/2021

The screenshot shows the 'Dashboard' page in the Checkmates application. The user is logged in as 'Cian'. The 'Personal Milestones' section displays a table with three milestones, each with a 'Remove' button. The Redux DevTools log on the right shows the state after the user profile is fetched, including the user's details and a list of milestones.

Title	Description	Status	Action
1 Month Checkin Streak	This milestones represents a big change in my life. I have been checking in daily for the past month and I have never felt better!	Private	Remove
TEST	TEST	Private	Remove
My First Checkin	This milestone marks your first checkin	Private	Remove

Test_Id	T043
Req_Id	R02
Details	Successfully delete milestone from profile
Date	18/3/2021

The screenshot shows the Checkmates dashboard with a green notification bar at the top stating "Milestone successfully removed from User Profile". The user is logged in as "Cian". The "Personal Milestones" section contains two items:

Title	Description	Status
1 Month Checkin Streak	This milestones represents a big change in my life. I have been checking in daily for the past month and I have never felt better!	Private Remove
My First Checkin	This milestone marks your first checkin	Private Remove

On the right, the Redux DevTools log shows the state after the milestone removal:

```

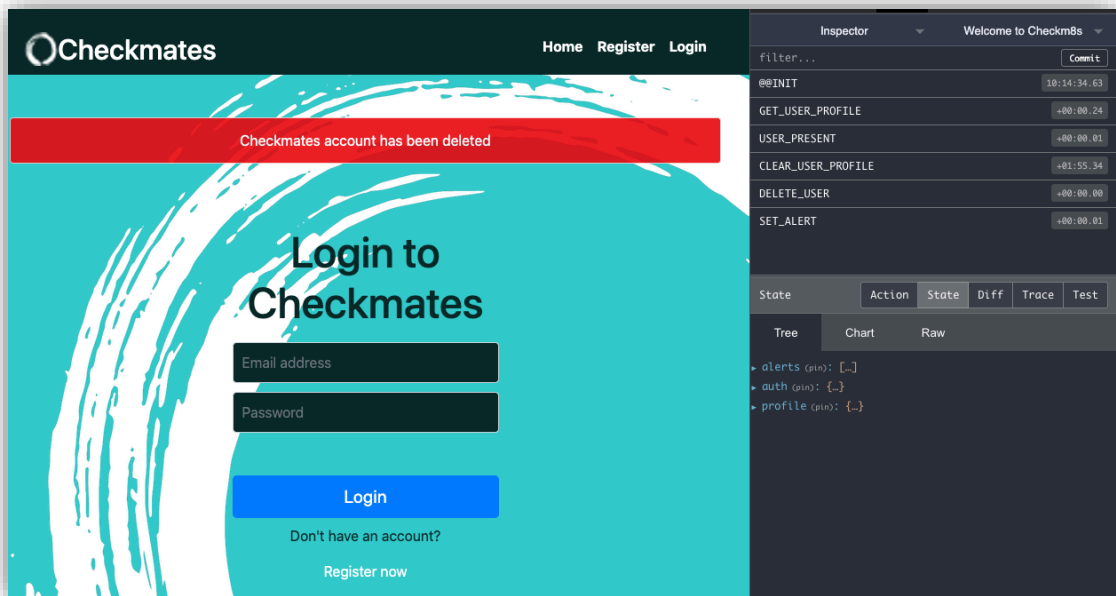
@@INIT 9:41:31.11
USER_PRESENT +00:00.21
GET_USER_PROFILE +00:00.03
UPDATE_USER_PROFILE +00:14.95
SET_ALERT +00:00.01

State Action State Diff Trace Test

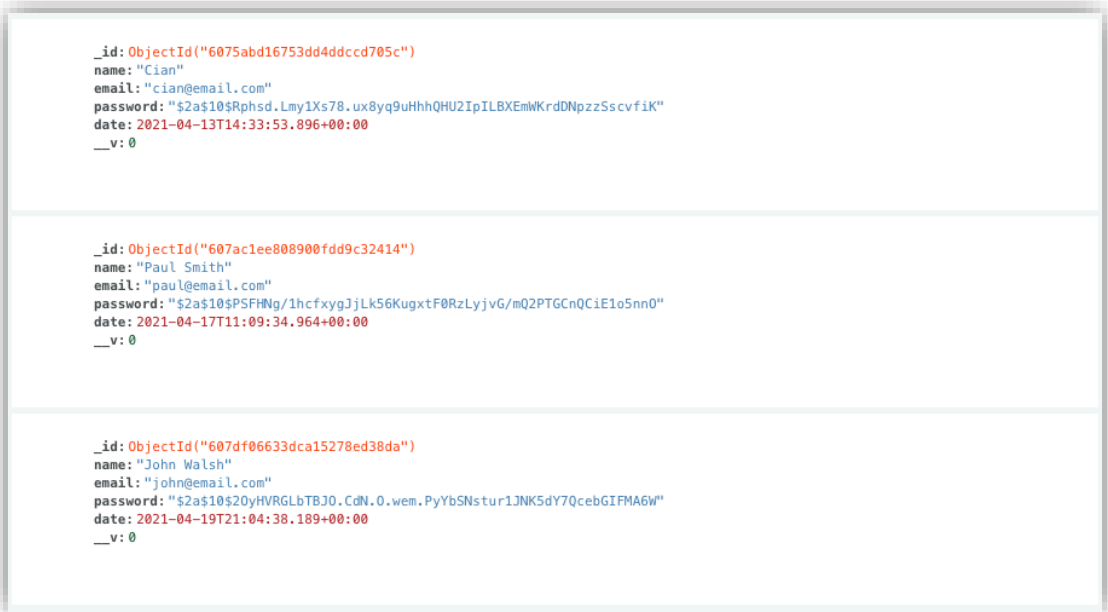
Tree Chart Raw
- alerts (pin): [...]
- auth (pin): {}
- profile (pin)
  - profile (pin)
    - social (pin): {}
      - _id (pin): "60760e169b69b75eeba3085b"
      - user (pin): "6075abd16753dd4dccc705c"
      - age (pin): 18
      - gender (pin): "Male"
      - bio (pin): "New to this, looking for any suggestions"
    - milestones (pin)
      - 0 (pin): [...]
      - 1 (pin): [...]
  
```

Test_Id	T044
Req_Id	R02
Details	Successfully delete account
Date	21/3/2021

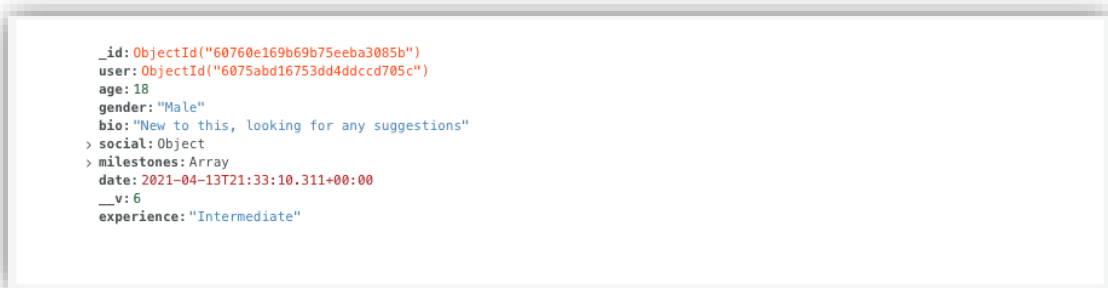
The screenshot shows the Checkmates dashboard with a confirmation dialog overlay. The dialog asks "Are you certain, this action cannot be reversed" and has "Cancel" and "OK" buttons. The user is logged in as "Token TEST". The "Personal Milestones" section is empty, and a "Delete Account" button is visible at the bottom left.



[User resource deleted]



[Profile resource deleted]



Test_Id	T045
Req_Id	R04
Details	List all user profiles
Date	21/3/2021

Test_Id	T046
Req_Id	R03
Details	Get All checkins
Date	25/3/2021

The screenshot shows a web application interface on the left and the Redux DevTools state on the right.

Web Application Interface:

- Header: Checkmates logo, navigation links: Mates, Dashboard, Checkins, Logout.
- Page Title: Get checkins test.
- Content: A large, abstract, teal and white circular graphic.

Redux DevTools State:

```

State
  Action
  State
  Diff
  Trace
  Test

Tree
  alerts (pin): []
  auth (pin): {}
  profile (pin): {}
  checkin (pin)
  checkins (pin)
    0 (pin)
      _id (pin): "6077ff107c5f5a9bd2664f32"
      user (pin): "6075abd16753dd4ddccd705c"
      title (pin): "Day 2"
      emotion (pin): "Fear"
      intensity (pin): 5
      maintext (pin): "No change today but I am determined to build a habit of self reflection through my journal"
      reactions (pin): []
      comments (pin): []
      date (pin): "2021-04-15T08:53:36.587Z"
      __v (pin): 4
    1 (pin)
      _id (pin): "607761177aba9e8c07606a58"
      user (pin): "6075abd16753dd4ddccd705c"
      title (pin): "Fist digital diary checkin"
      emotion (pin): "Fear"
      intensity (pin): 5
      maintext (pin): "I am feeling a little overwhelmed lately. I am afraid of losing my job due to COVID-19 restrictions"
  
```

Test_Id	T047
Req_Id	R03
Details	Checkin Item component display
Date	27/3/2021

Checkmates Mates Dashboard Checkins Logout

Checkins

Welcome to the checkmates community

Day 2
Emotion: Fear
Intensity: 5
 No change today but I am determined to build a habbit of self reflection through my journal
 ❤️ 1 Discussion 1 Private Chat
 Checkin date: 15/04/21

Fist digital diary checkin
Emotion: Fear
Intensity: 5
 I am feeling a little overwhelmed lately. I am afraid of losing my job due to COVID-19 restrictions
 ❤️ Discussion Private Chat
 Checkin date: 14/04/21

```

@@INIT
GET_CHECKINS
USER_PRESENT
  
```

Test_Id	T048
Req_Id	R03
Details	Add support reaction to user checkin
Date	29/3/2021

Checkmates Mates Dashboard Checkins Logout

Checkins

Welcome to the checkmates community

Day 3
Emotion: Anticipation
Intensity: 4
 Still a little anxious but starting to feel a sense of anticipation about the future.
 ❤️ 2 Discussion Private Chat
 Checkin date: 27/04/21

Day 2
Emotion: Fear
Intensity: 5
 No change today but I am determined to build a habbit of self reflection through my journal
 ❤️ 2 Discussion 1 Private Chat
 Checkin date: 15/04/21

```

@@INIT
GET_CHECKINS
USER_PRESENT
ADD_REACTION
  
```

Test_Id	T049
Req_Id	R03
Details	Validation that checkin can only be reacted to once
Date	29/3/2021

The screenshot shows the 'Checkins' page in the Checkmates application. It features two checkin entries for user 'Cian':

- Day 3:** Emotion: Anticipation, Intensity: 4. Description: 'Still a little anxious but starting to feel a sense of anticipation about the future.' Checkin date: 27/04/21. It has 2 likes and options for Discussion and Private Chat.
- Day 2:** Emotion: Fear, Intensity: 5. Description: 'No change today but I am determined to build a habit of self reflection through my journal.' Checkin date: 15/04/21. It has 2 likes and options for Discussion and Private Chat.

On the right, the Redux DevTools Inspector shows a log of actions: @@INIT, GET_CHECKINS, USER_PRESENT, ADD_REACTION, and CHECKIN_ERROR. The error log shows a 'Bad Request' (400) status.

Test_Id	T050
Req_Id	R03
Details	Delete Checkin Reducer Action
Date	1/4/2021

The screenshot shows the 'Checkins' page in the Checkmates application after a successful deletion. A green banner at the top reads 'Checkin has been successfully deleted'. The checkin entries are:

- Day 3:** Emotion: Anticipation, Intensity: 4. Description: 'Still a little anxious but starting to feel a sense of anticipation about the future.' Checkin date: 27/04/21. It has 2 likes and options for Discussion and Private Chat.
- Fist digital diary checkin:** Emotion: Fear, Intensity: 5. Description: 'I am feeling a little overwhelmed lately. I am afraid of losing my job due to COVID-19 restrictions.' Checkin date: 14/04/21. It has 1 like and options for Discussion and Private Chat.

On the right, the Redux DevTools Inspector shows a log of actions: @@INIT, USER_PRESENT, GET_CHECKINS, DELETE_CHECKIN, and SET_ALERT. The state log shows the 'alerts' array containing a success message: 'Checkin has been successfully deleted'.

Test_Id	T051
Req_Id	R03
Details	Checkin creation
Date	1/4/2021

Checkmates Mates Dashboard Checkins Logout

Checkin has been created, keep it up!

Create Checkin

Take a deep breath and focus on you for the next few minutes...

Checkin Title

Select the primary emotions that best describes how you are feeling

* Select Primary Emotion

Intensity

0

Checkin Entry


Be honest and open with yourself

- Select if you want this checkin to be private and hidden from all users

Submit Go Back

Checkins

Welcome to the checkmates community



Cian

Feeling grateful
Emotion: Trust
Intensity: 7

it is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

Discussion Private Chat

Inspector Welcome to Checkm8s

filter... Commit

@@INIT 5:27:58.89

GET_CHECKINS +00:00.15

USER_PRESENT +00:00.06

CREATE_CHECKIN +00:11.77

SET_ALERT +00:00.02

State Action State Diff Trace Test

Tree Chart Raw

```

alerts (pin)
  0 (pin)
    msg (pin): "Checkin has been created, keep it up!"
    alertType (pin): "success"
    id (pin): "eb5071e0-2ddc-45cb-8495-fe7e7e980443"
  auth (pin): {}
  profile (pin): {}
  checkin (pin): {}

```

Test_Id	T052
Req_Id	R03
Details	View Individual Checkin Action
Date	3/4/2021

The screenshot shows a web application interface for 'Checkmates' and a Redux DevTools inspector. The application interface includes a navigation bar with 'Mates', 'Dashboard', 'Checkins', and 'Logout'. The main content area displays a user profile for 'Cian' with a circular profile picture. The profile information includes the title 'Feeling grateful', 'Emotion: Trust', 'Intensity: 7', and 'Privacy: Private'. Below this is a paragraph of Lorem Ipsum text and a 'Private Chat' button. The Redux DevTools inspector shows the state of the application, with the 'checkin' object selected. The state is as follows:

```


{
  alerts: [],
  auth: {},
  profile: {},
  checkin: {
    checkins: [],
    checkin: {
      privacy: true,
      _id: "60883b9290eca648c521e6fd",
      user: "6075abd16753dd4ddccc705c",
      title: "Feeling grateful",
      emotion: "Trust",
      intensity: 7,
      maintext: "it is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like)."
    }
  }
}

```

Test_Id	T053
Req_Id	R05
Details	Add Advice Comment to Checkin
Date	6/4/2021

Comment added

Go Back



Cian

Feeling grateful

Emotion: Trust
Intensity: 7
Privacy: Private

it is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).


Private Chat

Checkin date: 27/04/21

Share some advice or show your support

Add your comment

Submit



Hi Cian, being grateful for small things every day has helped me a lot in becoming a more mindful person

Posted on 28/04/21

✕

Paul Smith

Inspector Welcome to Checkm8s

filter... Commit

@@INIT 10:14:21.49

GET_CHECKIN +00:00.18

USER_PRESENT +00:00.05

GET_USER_PROFILE +00:00.08

GET_CHECKIN +00:04.04

ADD_COMMENT +24:55.37

SET_ALERT +00:00.01

State Action State Diff Trace Test

Tree Chart Raw

alerts (pin)

- 0 (pin)
 - msg (pin): "Comment added"
 - alertType (pin): "success"
 - id (pin): "492c4bd1-b2ae-42a1-b300-9127f7807a58"
- auth (pin): {...}
- profile (pin): {...}
- checkin (pin): {...}

Test_Id	T054
Req_Id	R05
Details	Delete Advice Comment from Checkin
Date	7/4/2021

Checkmates Circle Dashboard Checkins Logout

Comment deleted successfully

Go Back

Cian

Feeling grateful
Emotion: Trust
Intensity: 7
Privacy: Private

it is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

Private Chat

Checkin date: 27/04/21

Share some advice or show your support

Add your comment

Submit

Inspector Welcome to Checkm8s

filter... Commit

@@INIT 10:58:12.57

GET_CHECKIN +00:00.20

USER_PRESENT +00:00.05

DELETE_COMMENT +00:38.04

SET_ALERT +00:00.01

State Action State Diff Trace Test

Tree Chart Raw

```

alerts (pin)
  0 (pin)
    msg (pin): "Comment deleted successfully"
    alertType (pin): "success"
    id (pin): "5f6c496a-5646-4655-ab17-2a923a936fe9"
  auth (pin): {}
  profile (pin): {}
  checkin (pin)
    checkins (pin): []
    checkin (pin): {}
  loading (pin): false
  error (pin): {}

```

Console What's New

top Filter Default len

Test_Id	T055
Req_Id	R06
Details	Analyze Checkin for Sentiment
Date	29/4/2021

```
// Google Natural Language API call
const [result] = await client.analyzeSentiment({ document: document });
const sentiment = result.documentSentiment;
// ReqId: R0
// TestId: T055
console.log(`Text: ${req.body.maintext}`);
console.log(`Sentiment score: ${sentiment.score}`);
console.log(`Sentiment magnitude: ${sentiment.magnitude}`);
```

```
[0] Text: This is a test to see how angry I am. What a bad day, so stressful at work and the weather is terrible. I just want to be alone in bed!
[0] Sentiment score: -0.4000000059604645
[0] Sentiment magnitude: 1.600000023841858
```

```
[0] Text: This is a test to see how happy I am. I'm delighted that I have found the person I want to share the rest of my life with. Positivity all round!
[0] Sentiment score: 0.800000011920929
[0] Sentiment magnitude: 2.4000000953674316
```

The screenshot shows a web application interface with two checkin cards and a Redux DevTools inspector on the right.

Checkins Section:

- Submit** [Go Back](#)
- Checkins**
- Welcome to the checkmates community

Checkin Card 1 (Joy):

- Natural Language Test**
- Emotion: Joy**
- Intensity: 8**
- Privacy: Public**
- Text: This is a test to see how happy I am. I'm delighted that I have found the person I want to share the rest of my life with. Positivity all round!
- Cian Harrison
- Discussion Private Chat
- Checkin date: 29/04/21

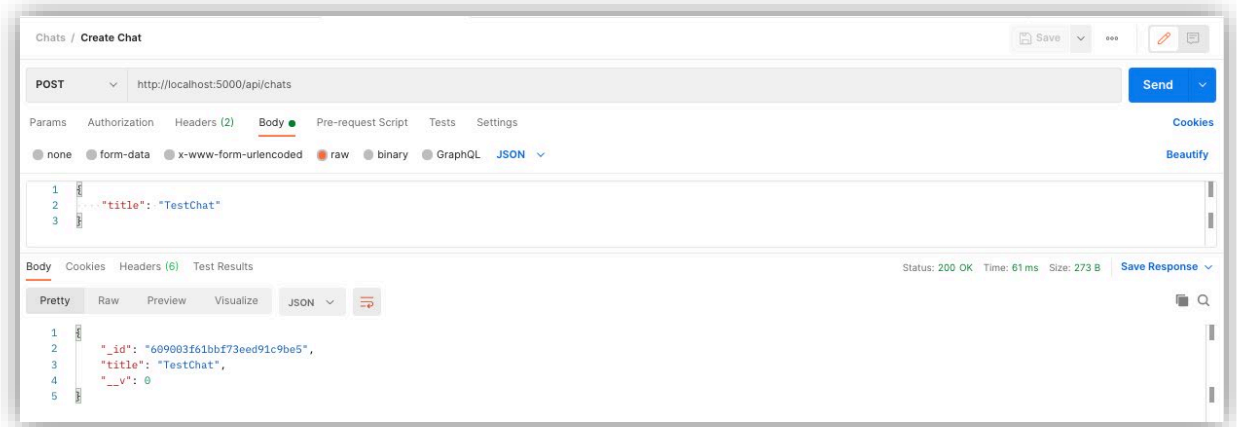
Checkin Card 2 (Anger):

- Natural Language Test**
- Emotion: Anger**
- Intensity: 6**
- Privacy: Private**
- Text: This is a test to see how angry I am. What a bad day, so stressful at work and the weather is terrible. I just want to be alone in bed!
- Cian Harrison
- Discussion Private Chat
- Checkin date: 29/04/21

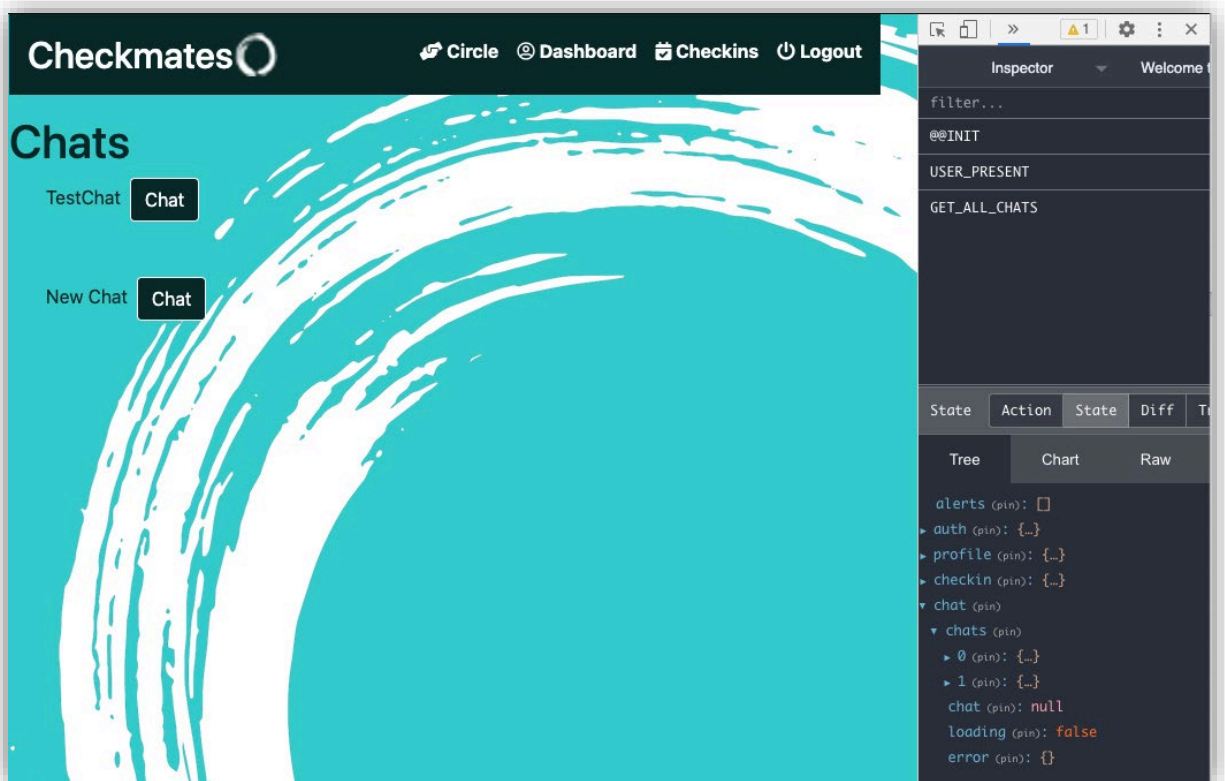
Redux DevTools Inspector:

- Inspector Welcome!
- filter...
- @@INIT
- USER_PRESENT
- GET_CHECKINS
- CREATE_CHECKIN
- SET_ALERT
- REMOVE_ALERT
- CREATE_CHECKIN
- SET_ALERT
- REMOVE_ALERT
- State Action State Diff Tr
- Tree Chart Raw
- alerts (pin): []
- auth (pin): {}
- profile (pin): {}
- checkin (pin)
- checkins (pin)
 - 0 (pin): {}
 - 1 (pin): {}
 - 2 (pin): {}
 - 3 (pin): {}
- checkin (pin): null
- loading (pin): false
- error (pin): {}

Test_Id	T056
Req_Id	R07
Details	Create a chat
Date	1/5/2021



Test_Id	T057
Req_Id	R07
Details	List Existing Chats
Date	1/5/2021



Test_Id	T058
Req_Id	R07
Details	Socket.io connection
Date	2/5/2021

```

56 // socket io connection
57 // ReqId: R0
58 // TestId: T058
59 io.on('connection', (socket) => {
60   console.log('Connected: ')
61
62   socket.on('disconnect', () => {
63     console.log('Disconnected: ')
64   })
65 })

```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

```

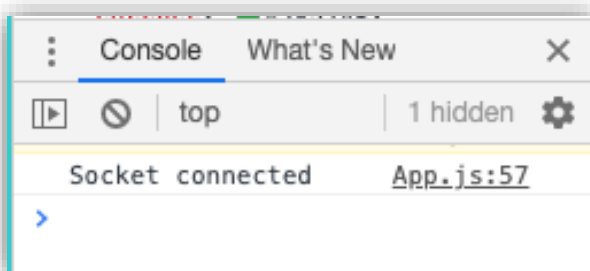
[0] [nodemon] restarting due to changes...
[0] [nodemon] starting `node server.js`
[0] [SERVER] Server started on port: 5000
[0] [DATABASE] MongoDB connected
[0] Connected:
[0] Disconnected:

```

```

45 // socket io connection
46 // ReqId: R0
47 // TestId: T058
48 // socket disconnection
49 newSocket.on('disconnect', () => {
50   setSocket(null)
51   setTimeout(socketSetup, 3000)
52   console.log('Socket Disconnected')
53 })
54 // socket connection
55 newSocket.on('connect', () => {
56   console.log('Socket connected')
57 })
58 //
59 setSocket(newSocket)
60 }

```



Test_Id	T059
Req_Id	R03
Details	Get personal checkins
Date	4/5/2021

Checkins / Get Personal Checkins Save ... 🔍

GET ▼ http://localhost:5000/api/checkins/user/60894b559a7e2823947d7ac0 Send ▼

Params Authorization **Headers (1)** Body Pre-request Script Tests Settings Cookies

Headers

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> x-auth-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2V5Ijpb71...				
Key	Value	Description			

Body Cookies Headers (7) Test Results Status: 200 OK Time: 53 ms Size: 3.91 KB Save Response ▼

Pretty Raw Preview Visualize JSON ▼ 🔍


```

2  {
3    "privacy": false,
4    "_id": "608a9c6815d1de599dfc107c",
5    "user": "60894b559a7e2823947d7ac0",
6    "title": "Natural Language Test",
7    "emotion": "Joy",
8    "intensity": 8,
9    "maintext": "This is a test to see how happy I am. I'm delighted that I have found the person I want to share the rest of
10    my life with. Positivity all round!",
11    "name": "Cian Harrison",
12    "avatar": "https://www.gravatar.com/avatar/b14f326dcd5d6193cf19b8669887c955?s=200&r=pg&d=mm",
13    "reactions": [
14      {
15        "_id": "609131a419ff331961322465",
16        "user": "60894b559a7e2823947d7ac0"
17      }
18    ],
19    "comments": [],
20    "date": "2021-04-29T11:45:44.941Z",
21    "__v": 1
22  },
23  {
24    "privacy": true,
25    "_id": "608a9862b8d7215864060df9",
26    "user": "60894b559a7e2823947d7ac0",
27    "title": "Natural Language Test",
28    "emotion": "Anger",
29    "intensity": 6,
30    "maintext": "This is a test to see how angry I am. What a bad day, so stressful at work and the weather is terrible. I just
31    want to be alone in bed!",
32    "name": "Cian Harrison",
33    "avatar": "https://www.gravatar.com/avatar/b14f326dcd5d6193cf19b8669887c955?s=200&r=pg&d=mm",
34    "reactions": [
35      {
36        "_id": "6090222e8ee72bf7894fe4a5",
37        "user": "60894b559a7e2823947d7ac0"
38      }
39    ],
40    "comments": [],
41    "date": "2021-04-29T11:28:34.434Z",
42    "__v": 3
43  },
44  {
45    "privacy": true,
46    "_id": "60894bc29a7e2823947d7ac2",
47    "user": "60894b559a7e2823947d7ac0",
48    "title": "Feeling grateful",
49    "emotion": "Trust",
50    "intensity": 7,
51    "maintext": "Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's
52    standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type
53    specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining
54    essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum

```

Checkins

Welcome to the checkmates community



Cian Harrison


Natural Language Test

Emotion: Joy
Intensity: 8
Privacy: Public

This is a test to see how happy I am. I'm delighted that I have found the person I want to share the rest of my life with. Positivity all round!

♥ 1 Discussion Private Chat

Checkin date: 29/04/21



Cian Harrison


Natural Language Test

Emotion: Anger
Intensity: 6
Privacy: Private

This is a test to see how angry I am. What a bad day, so stressful at work and the weather is terrible. I just want to be alone in bed!

♥ 1 Discussion Private Chat

Checkin date: 29/04/21



Cian Harrison

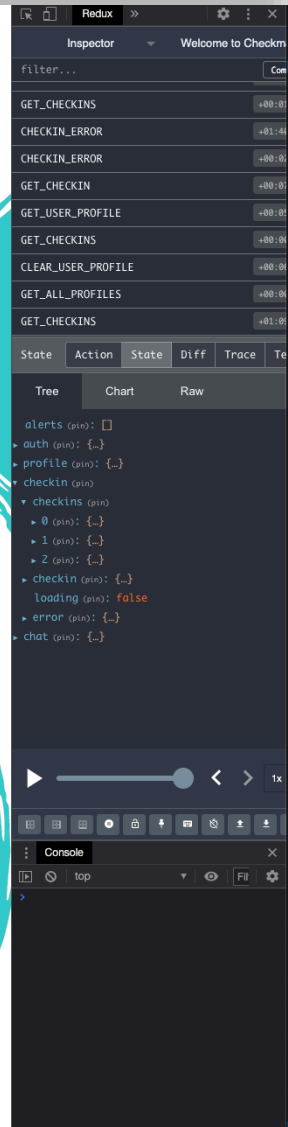
Feeling greatful

Emotion: Trust
Intensity: 7
Privacy: Private

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

♥ 2 Discussion 2 Private Chat

Checkin date: 28/04/21



Inspector Welcome to Checkm

filter...

- GET_CHECKINS +00:00
- CHECKIN_ERROR +01:40
- CHECKIN_ERROR +00:00
- GET_CHECKIN +00:00
- GET_USER_PROFILE +00:00
- GET_CHECKINS +00:00
- CLEAR_USER_PROFILE +00:00
- GET_ALL_PROFILES +00:00
- GET_CHECKINS +01:00

State	Action	State	Diff	Trace	Te
-------	--------	-------	------	-------	----

Tree Chart Raw

```
alerts (pin): []
auth (pin): {}
profile (pin): {}
checkin (pin)
checkins (pin)
  0 (pin): {}
  1 (pin): {}
  2 (pin): {}
  checkin (pin): {}
    loading (pin): false
  error (pin): {}
chat (pin): {}
```

Console

top