

National College of Ireland
BSc in Computing: Software Development Stream
2016/2017

John Deegan
X13327936
X13327936@student.ncirl.ie

GoalSpring
Technical Report



Declaration Cover Sheet for Project Submission

SECTION 1 *Student to complete*

Name: John Deegan
Student ID: x13327936
Supervisor: Lisa Murphy

SECTION 2 **Confirmation of Authorship**

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature: John Deegan Date: 08/05/2017

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College's Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to your being suspended or expelled from college.

Complete the sections above and attach it to the front of one of the copies of your assignment

What constitutes plagiarism or cheating?

The following is extracted from the college's formal statement on plagiarism as quoted in the Student Handbooks. References to "assignments" should be taken to include any piece of work submitted for assessment.

Paraphrasing refers to taking the ideas, words or work of another, putting it into your own words and crediting the source. This is acceptable academic practice provided you ensure that credit is given to the author. Plagiarism refers to copying the ideas and work of another and misrepresenting it as your own. This is completely unacceptable and is prohibited in all academic institutions. It is a serious offence and may result in a fail grade and/or disciplinary action. All sources that you use in your writing must be acknowledged and included in the reference or bibliography section. If a particular piece of writing proves difficult to paraphrase, or you want to include it in its original form, it must be enclosed in quotation marks

and credit given to the author.

When referring to the work of another author within the text of your project you must give the author's surname and the date the work was published. Full details for each source must then be given in the bibliography at the end of the project

Penalties for Plagiarism

If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the college's Disciplinary Committee. Where the Disciplinary Committee makes a finding that there has been plagiarism, the Disciplinary Committee may recommend

- that a student's marks shall be reduced
- that the student be deemed not to have passed the assignment
- that other forms of assessment undertaken in that academic year by the same student be declared void
- that other examinations sat by the same student at the same sitting be declared void

Further penalties are also possible including

- suspending a student college for a specified time,
- expelling a student from college,
- prohibiting a student from sitting any examination or assessment.,
- the imposition of a fine and
- the requirement that a student to attend additional or other lectures or courses or undertake additional academic work.

Contents

Executive Summary	6
Introduction	7
Background	7
Aims	7
Technologies	8
HTML5 & CSS3	8
JavaScript	8
MySQL	8
NodeJS	8
jQuery	9
ExpressJS	9
System	10
Requirements	10
Functional Requirements	10
Data Requirements	10
User Requirements	10
Environmental Requirements	10
Usability Requirements	11
Performance/Response Time Requirement	11
Availability Requirements	11
Recover Requirements	11
Robustness Requirements	12
Security Requirements	12
Reliability Requirements	12
Maintainability Requirements	12
Portability Requirements	12
Reusability Requirements	12
Design and Architecture	13
Use Case Diagram	13
Implementation	13
The Beginning & Setup	14
User Authentication	14

Task Creation.....	14
Responsive Calendar.....	15
Data Visualization.....	16
Artificial Intelligence	16
API Endpoints.....	18
Testing	29
Responsiveness	29
Usability Testing.....	33
Conclusion and Further Development.....	36
Appendix One: Project Proposal	38
Objectives	38
Develop a System for tracking and learning habitual behaviours.....	38
Increase Productivity and Learning.....	38
Background.....	38
Technical Approach	39
Front-End.....	39
Back-End	40
Project Plan.....	40
Proposal Conclusion	40
Appendix Two: Monthly Journals	41
Bibliography	50

Executive Summary

Allowing the youth in Ireland to maximize their grades and productivity is vital to creating a more talented workforce in Ireland. However, there has never been another time in human history where young people have faced as much expectation as the present day, children are almost expected to go to college once their school days are over, what was unthinkable for most families thirty years ago is now a social norm. Thus, the competitiveness for college places and the rapidly increasing nature of CAO points can place a huge amount of stress on young people, and that's before the usual stresses of teenage life. If they're fortunate enough to make it to college and participate their dream course, then graduate programs are also expecting at least 2:1 degrees, thus the pressure is constant to perform well academically for Ireland's young generation.

My app is a solution for the young people of today to better organise all aspects of their life to ensure increased productivity and mental health. Users can input their daily tasks and receive feedback for what they have completed over time. Artificial Intelligence can assist students in creating healthy schedules.

I used HTML5, CSS3 and JavaScript to code the front-end of the application, ensuring that all design was mobile first and the front-end code kept the UI fast and responsive. I implemented NodeJS as the backend framework for the application, as this allowed me to focus on JavaScript for front and back end and greatly improved efficiency during development.

Introduction

Background

Young people of the present day are weighed down by expectation more than at any other time in human history. College places have become more and more competitive (O'Brien, 2016) yet it's almost expected of young people to go to college, it's almost become a social norm. Young people of the present day are also the first generation in history to have grown up with the internet being ever present in their lives, and thus they also have more social pressures than their young predecessors.

My decision to make an application for scheduling was influenced by these pressures. Often apps that offer scheduling such as Google Calendar, "Cloud Cal" and "The Rest" don't offer recommended schedules or assistance in creating calendars, they have more of a "Insert as you go" feel. It is my considered opinion that if someone already doesn't know how to schedule their lives, how will apps like these help them? How does the current market cater to those of us who are at a loss as to what to do next? My research into calendar apps offered very little answers to this question. Most apps offer a "re-occurring feature" which allows you to set one task at set intervals, such as daily or weekly occurrence. The only app I found that offered similar functionality in terms of analytics is Esper (<https://esper.com>), but I felt the UI isn't friendly especially towards young people, instead it seemed to be geared towards the working professional 30-something market. Thus, I felt there was a sufficient gap in the current market for an app which offered a combination of generation and analytical functionality.

During my third-year work placement, a colleague introduced me to the NodeJS framework and more in depth knowledge of the JavaScript language. It was during my placement when I realised that Web Development was the career I wished to pursue and thus I felt it was imperative to use my Final Year project as a platform for improving my Web Development portfolio and skills. Per a Stack Overflow survey in 2016, JavaScript has surpassed Java and PHP as the most popular language used amongst full stack developers, and this has been recognised by universities such as Stanford University in the United States, who now offer their introductory programming course CS 106A in JavaScript rather than Java (Hughes, 2017). In addition, this project allowed me to hone my Relational Database skills using MySQL and cloud based development using Heroku and ClearDB services.

Aims

The goal of the app as stated was to assist young people to find a suitable work/life balance in a world which demands so much of young people. It was important that this app was developed with mobile first design in mind, as the assumption is made that most users of the website will use it via smart devices such as a phone or tablet.

This app is intended to be a progressive web application, defined by google as feeling like "a natural app on the device, with an immersive user experience" (developers.google.com, 2017). This also includes

being responsive to all pixel sizes and quick to respond for modern internet connections. It is required to be fit for its intended purpose. A user can create a profile and login smoothly and securely, with maximum security for user data via the bcrypt encryption library and the PassportJS middleware framework. The user can be greeted by a calendar screen which is simple and self-explanatory to use. Once the user has clicked on a day on the calendar, they will be greeted with a webpage with the tasks they have currently set. The user can edit and delete these tasks which will be displayed in a clean and readable manner. The user can also generate a schedule which will use AI to ensure a user has a balanced schedule. The user can also analyse a schedule which will show a webpage of data visualization to showcase user data. Long term usage of the app is ensured via the user also being able to view analysis for their weekly and monthly task data.

Technologies

HTML5 & CSS3

HTML5 and CSS3 are now the standard markup languages for the front end of web applications and websites. In 2017 there is no reason to use earlier versions of either, especially now responsive web design is so prevalent. HTML5's meta tag, and CSS3's media queries are key aspects of responsive design, and as this project is "mobile-first", using them was an easy decision.

JavaScript

JavaScript was chosen as the primary language for this project due to its rapidly increasing popularity in the world of full stack web development. The language is extremely well supported and has an extremely helpful community of users. In recent years, JavaScript has developed from being a functional programming language to incorporating many features of more traditional languages such as Java and C# to allow it to extend to the back end with frameworks such as Node JS.

MySQL

MySQL was the primary relational database management system taught to this year's 4th Year NCI computing students during their time at college. Thus, it made sense to use MySQL for this project rather than having to learn an entire new database system. I had contemplated using MongoDB as it would fit the "JavaScript theme" of the project, but I thought that the familiarity of the MySQL system would improve the efficiency of the project.

NodeJS

NodeJS is an open-source server side JavaScript environment that is built on Google Chrome's JavaScript engine that allows for server-side and networking applications. Node also provides a rich library of JS modules which allow for more efficiency and simplicity in developing web applications. It's quick, asynchronous, highly scalable and no buffering.

JQuery

JQuery is a front-end JavaScript framework that is extremely fast and rich in features. It allows for more simple and efficient traversal of the DOM. JQuery allowed me to use Ajax calls much easier, including get and post requests to Endpoints specified in my ExpressJS API. It also allows me to include animations and event handling for more reactive front end user interaction.

ExpressJS

Express is an extremely simplified web application framework that allows backend JavaScript to use MVC architecture. ExpressJS is to JavaScript what Rails or Sinatra is to Ruby. The most important aspect of ExpressJS in this project is an HTTP module, which makes abstraction for making a web server and RESTful services easy.

System

Requirements

The requirements were among the first things to be considered at the outset of the GoalSpring project. To produce these requirements, I met with young people and asked them varying questions as to what they would like to see in a scheduling application. Not only did I ask children below the age of 18 at Leaving Cert level, I was also interested to see how such an app would also benefit young adults attending third level institutions in Ireland. I also consulted with my academic supervisor as what she considers this app should contain as she is a lecturer and may have gained insights from college students during her work.

Functional Requirements

- The system must allow for the creation/deletion/updating of user account information.
- The system must be responsive to all devices and screen/pixel widths and heights.
- The system must allow for CRUD operations to be performed on user's set tasks.
- The system must be able to read the user's inputs and conduct an analysis on tasks completed/inserted.
- This analysis must be conducted for different time spans set by the user. This may include daily, weekly, or annually.
- The system will store user's data within a database safely and securely using encryption services.
- The system must allow users to collaborate and "share" tasks.

Data Requirements

User data is stored within a MySQL database. This data will be secure and encrypted using the bcrypt library. As this app is catered for young people, including children, all data that is received from the user will not entail any personal information apart from the user's e-mail address.

User Requirements

The user must have access to a modern broadband internet connection to access the webpage. Thus, they must also be in possession of a device that can download and/or install an internet browser such as Microsoft Edge or Google Chrome that can showcase the webpage.

Environmental Requirements

As this application is usable on devices which can run an internet browser and an internet connection, the environment that the application is used in is not a major issue. However, the application's interface was designed with mobile-first design in mind, which means it has been optimized for users with touch screen devices such as smart phones or tablets. Thus, a user must ensure that their hands are not in unstable environments such public transport or using gloves in cold weather as this may affect their experiences trying to use the application.

Usability Requirements

The Usability requirements were an area of great research. Techniques such as the five second test and the think out loud test were used on the website to ensure perfection in the interface. Details of these can be seen in the Testing section of this document.

As mentioned previously, this application was designed with mobile-first in mind, thus the challenge was to make a "control panel" like user interface friendly on a smaller screen. Had I designed this project with desktop users as a priority, this would have been a much easier task as the greater pixel width would allow for a more "dashboard" feel for the interface.

Efficiency was highly prioritised as this application is catering towards younger people. If the app is difficult to use, it will not keep the attention of the young user. It needed to be straightforward and to the point. Buttons and actions must be self-explanatory and users must not be confused as to what an action does.

Performance/Response Time Requirement

The application will run in real time live on a web page. Once a day has been completed, the system will tally up the totals and analyse the user's schedule, returning the results on request.

An optional E-Mail will be sent to the user in regards to the availability of this analysis.

Availability Requirements

This application will be available to all devices that use an internet browser and an internet connection.

Recover Requirements

Once a user has created an account, their data will be stored on a live database. Therefore, it is immune to any hardware malfunctions the user may have with their personal device.

The user may however, delete their data from the database by deleting their account.

Robustness Requirements

This application must work on the user's chosen device unless the user has deleted their account or is disconnected from the internet.

Security Requirements

All personal data stored in our database must only be accessed by the user via sign in authentication. The system must collect the user's schedule for no other reason than to conduct its analysis of it.

Reliability Requirements

The application must work and conduct itself in a manner as advertised.

Maintainability Requirements

At sporadic intervals, the application can be maintained and updated if bugs are found or if an efficient new feature is thought of.

Portability Requirements

The application and website must be accessible from any device which can use an internet browser. The application must work as smooth as possible across all modern browsers such as Edge, Safari, Chrome and Firefox.

Reusability Requirements

The system is marketed towards students but may be beneficial to any person that seeks an analysis of their schedule or management of tasks

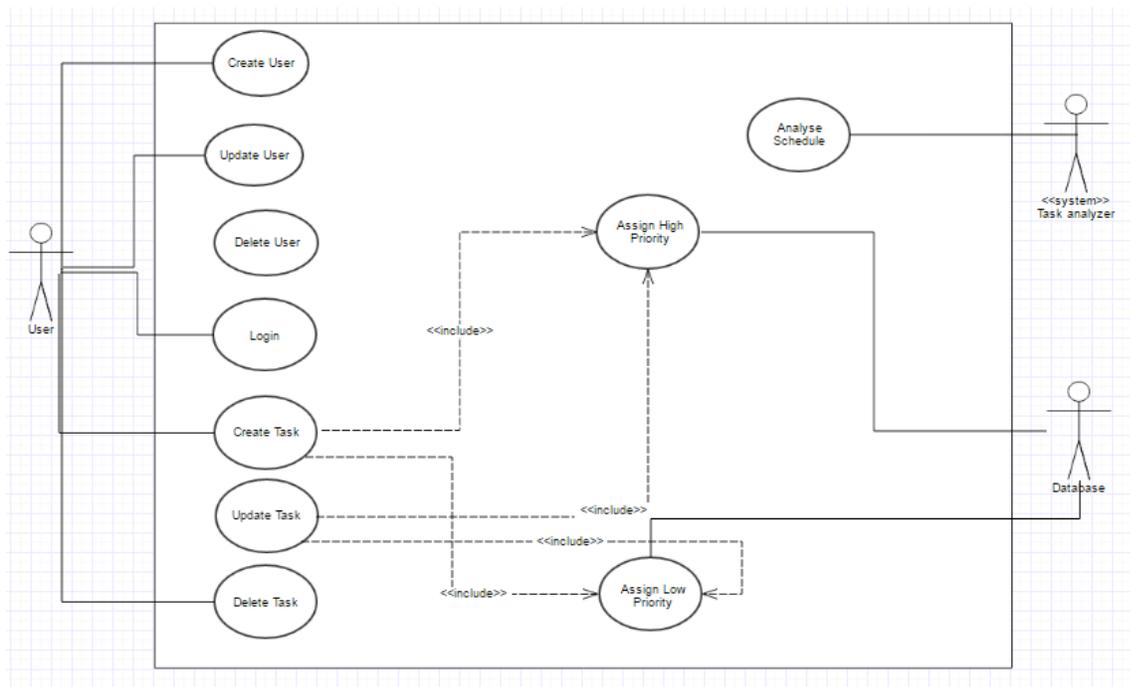
Design and Architecture

This website was primarily built using EJS (Embedded JavaScript) and CSS. EJS uses a virtually identical syntax to HTML, but allows me to embed Database data that ExpressJS/NodeJS has rendered along with the webpage. EJS is the JavaScript equivalent to Ruby on Rails' ERB files.

The front-end of the application was created using the technologies mentioned above in addition to JQuery and Ajax. I used the JQuery .get function which sends an Ajax call to the endpoint specified in my API routes file. Within this routes file I use the ExpressJS framework to create endpoints that send MySQL queries to my ClearDB hosted database.

I used bootstrap and most importantly the bootstrap grid system to create a responsive layout for the application.

Use Case Diagram



Implementation

I felt like this project has been a great learning curve for me in the realms of web development, no less because of the challenges that I faced whilst developing. There were times where I was in great difficulty for many hours on a single issue or bug, but I felt this was of great benefit for my problem-solving skills. The challenge of the project was my favourite aspect of the implementation.

The Beginning & Setup

I began by creating a new git repository for my project. Git is important for version control and keeping backups of my code in case there is a major issue or my computer's hard drive is impaired. The name GoalSpring derives from the theme of the project being young people aspiring to be the best they can be, thus they are "springing" to their goals. Two syllable names for applications are proven to be more successful, some of the most widely used apps such as "Face-book", "Twit-ter" and "Insta-gram" are proof of this.

From here, I set up my NodeJS project in my Atom text editor. I prefer Atom as my text editor due to its flexible customization and especially its minimap feature. Since some files of this project have hundreds of lines of JavaScript code, this minimap has been of great benefit to me. Using NodeJS, I installed the ExpressJS npm package via my command line. This allows me to use the ExpressJS framework anywhere on my project locally. I created a server.js file, which contained such variables as the port number I will use and setting up the paths within my project so that they may work when the website is hosted on Heroku.

User Authentication

From here I set about creating basic user authentication. To do this of course, there needs to be a live website first. This early chapter of my project consisted of figuring out colour schemes and what I wanted to general "feel" of the website to be. I wanted green to be a primary colour on the website as green tends to symbolize an action or a "go" movement. I wanted the user to always feel motivated for such a productivity application and thus I felt this was a valid choice. I created wireframes and took screenshots from websites of which I liked certain aspects of their design and placed them in a folder for later inspiration.

The user authentication was completed using the PassportJS library and npm package. PassportJS is an open source popular middleware for NodeJS/ExpressJS applications that uses powerful features such as OAuth, OpenID and persistent sessions. At this point I began to design my schema as user authentication requires such. I created a simple user table within my ClearDB hosted MySQL schema that held the user's e-mail and password. To enable PassportJS to work with MySQL posed the first big challenge of this project as Passport is mainly supported for MongoDB users. I spent a couple of days looking under the hood of the Passport framework figuring out every intricate detail of the middleware, which allowed me to "translate" the code and change it to work with my MySQL database. To finish user authentication, I imported the JavaScript version of the bcrypt library to encrypt my user's passwords. Bcrypt is a password hashing function that was created by Niels Provos and David Mazieres in 1999 based from the Blowfish cipher and has implementations for most popular programming languages.

Task Creation

Now I could register users, I set about tackling the focus of the application, which was task creation. I created another table in the schema, the task's table, to hold user activities. This schema contains a

foreign key, the user_id column, which links each task to a user from the user table. Within this table with have the ID of the task, the name, category and the start and end datetimes. Following the creation of the schema, I could begin my development of my API endpoints within the routes.js file of my project. This file uses the ExpressJS framework and allows me to use HTTP requests at a specified endpoint, and code what this endpoint does. I could also run MySQL queries within these endpoints to return the data requested. Finally, I used the EJS templating engine to render this data to the view. This follows a very “Ruby on Rails-esque” technique and allows for more efficient development by sticking to JavaScript rather than using another language such as PHP.

Responsive Calendar

I created endpoints for my CRUD task operation. I felt I learned a lot about how to use ExpressJS efficiently from this experience, as I had to combine JQuery Ajax requests from my client side JavaScript with these endpoints to successfully retrieve and display the data that wasn’t already rendered by the EJS templating engine.

It was at this point where upon meeting with my academic supervisor I decided to first create a “calendar” like interface when the user first logs in, rather than automatically be greeted with the CRUD task functionality. This would allow the user to have a greater control over which day they wished to create their tasks for, rather than having to specify the date each time they wished to create a task. This prompted me to research again, looking for inspiration from similar scheduling applications. However, I found these interfaces to be very difficult to design for smaller screen sizes such as mobile and tablet screens, and since this project is designed mobile-first, this created the second major problem with the implementation. I spent a while figuring out how I could create both an intuitive calendar design whilst maintaining responsiveness and managed to figure out a solution.



As you can see, this is a screenshot during development from a mobile device. I've managed to create a calendar "feel" whilst also maintaining responsiveness.

Data Visualization

Following this, I fixed my endpoints so that clicking on each "Day Div" would render a webpage with all the tasks assigned to the day that the user clicked on. I had to use some client side JavaScript to retrieve the value of the day the user has clicked on, including working with the built-in JS date object to retrieve the current date and time. Using JQuery Ajax get function, I could insert the current date into a string which I could send to my server to run the endpoint and query the database for the data for that day.

I considered Data Visualization for JavaScript and found two main popular libraries for this. D3 is the most popular followed by ChartJS. D3 tends to be for more complex Data projects than what I would require, therefore I felt the more streamlined ChartJS to be a good choice for this project. ChartJS allows me to implement various charts such as Bar, Chart and Line graphs simply by using an object and telling the library what labels and data I want inside the app. I found ChartJS library to be very efficient and exactly what I was looking for. I managed to use JQuery Ajax get requests on the analysis webpage to get the data required and then format this data into an object which can then be inserted into the chart object and displayed to the user.

Task Type



This is an example of a bar chart as it appears in the application.

Artificial Intelligence

Finally, the artificial intelligence for schedule generation was considered. For the AI, I studied extensively JavaScript artificial intelligence, as the only exposure I had to AI up to this point was creating a chess game in Java during one of my fourth-year modules. I found that the principles were the same across programming languages and that they used similar algorithms.

Firstly, I had to give evaluation to the categories available to assign to tasks. My categories are Social, Exercise, Productive, School and Positivity. These categories are assigned weekly “quotas” to achieve, like assigning values to pieces in chess. These quotas were decided upon after extensive research into recommended exercise, studying and socializing hours per week. The AI agent will examine the previous seven days of data and will tally up the data. The agent will then use this data to assign a recommended schedule based upon this data for the user, who may decide to use or modify it as they so wish.

I researched various resources for finding the “recommended” or “right” volume of time to do things. (Insert resources here). Using these resources, I came up with numbers for the number of hours expected to do these categories in a day.

Task	Weekday Hours	Weekend Hours
Study	3	6
Socializing	1	6
Exercise	1	1
School	User defined.	0
Positivity	1	1

These are all block tasks that will be inserted into the user’s schedule. From here, the user can edit and delete the tasks as they so wish.

We first retrieve the last seven days of user data from the database. Then the total hours of each category are tallied and matched with the data recommended weekly defined above. The AI is now posed with a choice of possible actions and tasks to create. For example, if the user’s school hours are set and the user has set a task for the evening already but their study hours are particularly low that week in comparison to the other categories, the AI will attempt to slot in hours that evening. This position evaluation is an aspect of AI I learned from my college module this semester and I was happy that I was able to implement it within the scope of this project.

API Endpoints

Following my fourth-year module in Web Services and API Development, I found that implementing RESTful web services using Node and Express wasn't too different from what I had learned in Java.

Endpoint 1: Login

URL	/login
Method	GET
URL Params	N/A
Data Params	N/A
Success Response	Code: 200 Content: Renders Login.ejs
Error Response	Code: 404 Content: CANNOT GET /login
Sample Call	N/A

URL	/login
Method	POST
URL Params	N/A
Data Params	N/A
Success Response	Code: 302 Content: Renders Profile.ejs
Error Response	Code: 404 Content: CANNOT GET /login
Sample Call	N/A

Endpoint 2: Signup

URL	/signup
Method	GET
URL Params	N/A
Data Params	N/A
Success Response	Code: 200 Content: Renders Signup.ejs
Error Response	Code: 404 Content: CANNOT GET /signup
Sample Call	N/A

URL	/signup
Method	POST

URL Params	N/A
Data Params	N/A
Success Response	Code: 302 Content: Renders Profile.ejs
Error Response	Code: 404 Content: CANNOT GET /signup
Sample Call	N/A

Endpoint 3: Profile

URL	/profile
Method	GET
URL Params	N/A
Data Params	N/A
Success Response	Code: 302 Content: Renders Profile.ejs
Error Response	Code: 404 Content: CANNOT GET /profile
Sample Call	N/A

Endpoint 4: Get One Day Data

URL	/tasks/day/:id
Method	GET
URL Params	id = Date of box clicked. e.g: Friday 5 th May = 05052017 /tasks/day/05052017 returns tasks for that date.
Data Params	N/A
Success Response	Code: 304 Content: [RowDataPacket { id: 172, name: 'Study', category: 'Productive', start: 2017-05-05T02:00:00.000Z, end: 2017-05-05T05:00:00.000Z, user_id: 12 }, RowDataPacket { id: 262, name: 'Football', category: 'Excercise', start: 2017-05-05T07:00:00.000Z, end: 2017-05-05T09:00:00.000Z, user_id: 12 }, RowDataPacket { id: 212,

	<pre>name: 'Running', category: 'Excercise', start: 2017-05-05T09:00:00.000Z, end: 2017-05-05T11:00:00.000Z, user_id: 12 }]</pre> <p>Renders 'tasks.ejs'.</p>
Error Response	<p>Code: 404 Content: CANNOT GET /tasks/day/:id</p>
Sample Call	<pre>var date = 05052017; \$.get('/tasks/day/'+date+", function(data, status) { });</pre>

Endpoint 5: Create Task

URL	/tasks/day/single/create
Method	POST
URL Params	N/A
Data Params	N/A
Success Response	<p>Code: 302 Content:</p> <pre>RowDataPacket { id: 272, name: 'Homework', category: 'Productive', start: 2017-05-05T14:00:00.000Z, end: 2017-05-05T16:00:00.000Z, user_id: 12 }]</pre> <p>Refreshes 'tasks.ejs' on POST.</p>
Error Response	<p>Code: 404 Content: CANNOT GET /tasks/day/:id</p>
Sample Call	<pre><form class="taskForm" id="showForm" method="post" action="/tasks/day/single/create"> <button type="submit" id="submitTaskBtn" class="btn btn- success">Submit</button> </form></pre>

Endpoint 6: Edit Task

URL	/tasks/day/single/edit/:id
Method	GET
URL Params	:id
	<p>When task.ejs is rendered it outputs daily task data from the server. Amongst rendered is the id of the task which is inserted into the href of the</p>

	<a> HTML tag to make the Edit button call this API on click and thus this ID becomes the :id param of the API call.
Data Params	N/A
Success Response	Code: 200 Content: Renders 'edit.ejs'
Error Response	Code: 404 Content: CANNOT GET /tasks/day/single/edit/:id
Sample Call	Where dbdata[i] is daily database data sent from the ExpressJS server. <a href="/tasks/day/single/edit/<%= dbdata[i].id %>"><button class="btn btn-success">Edit</button>

URL	/tasks/days/single/edit/save/:id
Method	POST
URL Params	:id Where :id is the ID (primary key) of the task as stored within the MySQL database. When edit.ejs is rendered we send the ID (primary key) of the task as it currently it within the MySQL database and render this to the view. Using this, we can append this ID to the form action so that when the edit form is submitted, we are submitting it to the correct id parameter.
Data Params	N/A
Success Response	Code: 302 Content: Returns the modified data. { name: 'Study', category: 'Productive', start: '2017:05:05 21:00:00', end: '2017:05:05 22:30:00', user_id: 12 }
Error Response	Code: 404 Content: CANNOT POST /tasks/days/single/edit/save/:id
Sample Call	<form class="taskForm" id="showForm" method="post" action="/tasks/days/single/edit/save/<%=selectedData[0].id%>"> <button type="submit" id="submitTaskBtn" class="btn btn-success">Submit</button> </form>

Endpoint 7: Delete Task

URL	/tasks/days/single/edit/save/:id
Method	POST
URL Params	:id Where :id is the ID (primary key) of the task as stored within the MySQL database. When edit.ejs is rendered we send the ID (primary key) of the task as it currently it within the MySQL database and render this to the view. Using this, we can append this ID to the form action so that when the edit form is submitted, we are submitting it to the correct id parameter.
Data Params	N/A
Success Response	Code: 302 Content: Returns the modified data. { name: 'Study', category: 'Productive', start: '2017:05:05 21:00:00', end: '2017:05:05 22:30:00', user_id: 12 }
Error Response	Code: 404 Content: CANNOT POST /tasks/days/single/edit/save/:id
Sample Call	<form class="taskForm" id="showForm" method="post" action="/tasks/days/single/edit/save/<%=selectedData[0].id%>"> <button type="submit" id="submitTaskBtn" class="btn btn-success">Submit</button> </form>

Endpoint 8: Count Tasks

URL	/tasks/days/count
Method	GET
URL Params	N/A
Data Params	N/A
Success Response	Code: 302 Content: Returns the number of tasks a User has on a certain day and displays this to our 'profile.ejs' page on our calendar.

Error Response	Code: 404 Content: CANNOT POST /tasks/days/count
Sample Call	\$.get('/tasks/days/count', function(data, status) { });

Endpoint 9: Analyse Daily Tasks

URL	/tasks/days/analyze/:id
Method	GET
URL Params	:id Where id is equal to the day that the user is requesting to analyse.
Data Params	N/A
Success Response	Code: 200 Content: Renders the 'analyze.ejs' file.
Error Response	Code: 404 Content: CANNOT GET /tasks/days/analyze/:id
Sample Call	N/A

URL	/tasks/days/analyze/get/:id
Method	GET
URL Params	:id Where id is equal to the day that the user is requesting to analyse.
Data Params	N/A
Success Response	Code: 200 Content: Gathers all the daily tasks of the user for analysis within the client side JavaScript file 'analyze.js' [RowDataPacket { id: 262, name: 'Football', category: 'Exercise', start: 2017-05-05T07:00:00.000Z, end: 2017-05-05T09:00:00.000Z, user_id: 12 }, RowDataPacket { id: 212, name: 'Running', category: 'Exercise', start: 2017-05-05T09:00:00.000Z, end: 2017-05-05T11:00:00.000Z,

	<pre> user_id: 12 }, RowDataPacket { id: 272, name: 'Homework', category: 'Productive', start: 2017-05-05T14:00:00.000Z, end: 2017-05-05T16:00:00.000Z, user_id: 12 }, RowDataPacket { id: 172, name: 'Study', category: 'Productive', start: 2017-05-05T20:00:00.000Z, end: 2017-05-05T21:30:00.000Z, user_id: 12 }] </pre>
Error Response	<p>Code: 404 Content: CANNOT GET /tasks/days/analyze/get/:id</p>
Sample Call	<pre> var url = window.location.href; //get the id of the day which is in the url var urlId = url.substr(url.length - 8); console.log("This file"); var taskData = []; \$.get('/tasks/days/analyze/get/'+ urlId +", function(data, status) { }); </pre>

Endpoint 10: Analyse Weekly Tasks

URL	/tasks/days/analyze/weekly/:id
Method	GET
URL Params	:id Where id is equal to the first day of the week that the user is requesting to analyse.
Data Params	N/A
Success Response	<p>Code: 200 Content: Renders the 'analyzeWeekly.ejs' file.</p>
Error Response	<p>Code: 404 Content: CANNOT GET /tasks/days/analyze/:id</p>
Sample Call	N/A

URL	/tasks/days/analyze/weekly/get/:id
Method	GET
URL Params	:id Where id is equal to the first day in the week that the user can request their data from (This date – 7)
Data Params	N/A
Success Response	<p>Code: 200</p> <p>Content: Gathers all the weekly tasks of the user for analysis within the client side JavaScript file 'analyzeWeekly.js'</p> <pre>[RowDataPacket { id: 262, name: 'Football', category: 'Excercise', start: 2017-05-05T07:00:00.000Z, end: 2017-05-05T09:00:00.000Z, user_id: 12 }, RowDataPacket { id: 212, name: 'Running', category: 'Excercise', start: 2017-05-05T09:00:00.000Z, end: 2017-05-05T11:00:00.000Z, user_id: 12 }, RowDataPacket { id: 272, name: 'Homework', category: 'Productive', start: 2017-05-05T14:00:00.000Z, end: 2017-05-05T16:00:00.000Z, user_id: 12 }, RowDataPacket { id: 172, name: 'Study', category: 'Productive', start: 2017-05-05T20:00:00.000Z, end: 2017-05-05T21:30:00.000Z, user_id: 12 }, RowDataPacket { id: 222, name: 'WeekTest1', category: 'Social', start: 2017-05-06T10:00:00.000Z, end: 2017-05-06T12:00:00.000Z, user_id: 12 }, RowDataPacket { id: 232, name: 'WeekTest1', category: 'Excercise', start: 2017-05-07T10:00:00.000Z, end: 2017-05-07T13:00:00.000Z, user_id: 12 },</pre>

	<pre> RowDataPacket { id: 242, name: 'WeekTest1', category: 'Social', start: 2017-05-08T10:00:00.000Z, end: 2017-05-08T11:00:00.000Z, user_id: 12 }, RowDataPacket { id: 252, name: 'WeekTest1', category: 'Productive', start: 2017-05-08T14:00:00.000Z, end: 2017-05-08T16:00:00.000Z, user_id: 12 }] </pre>
Error Response	<p>Code: 404 Content: CANNOT GET /tasks/days/analyze/weekly/get/:id</p>
Sample Call	<pre> var url = window.location.href; //get the id of the day which is in the url var urlId = url.substr(url.length - 8); console.log("This file"); var taskData = []; \$.get('/tasks/days/analyze/get/'+ urlId +", function(data, status) { }); </pre>

Endpoint 11: Analyse All-Time Tasks

URL	/tasks/days/analyze/alltime/
Method	GET
URL Params	N/A
Data Params	N/A
Success Response	<p>Code: 200 Content: Renders the 'analyzeAllTime.ejs' file.</p>
Error Response	<p>Code: 404 Content: CANNOT GET /tasks/days/analyze/alltime/:id</p>
Sample Call	N/A

URL	/tasks/days/analyze/alltime/get/
Method	GET
URL Params	N/A
Data Params	N/A
Success Response	<p>Code: 200</p> <p>Content: Gathers every task the user has ever created for analysis within the client side JavaScript file 'analyzeAllTime.js'</p> <pre>[RowDataPacket { id: 262, name: 'Football', category: 'Excercise', start: 2017-05-05T07:00:00.000Z, end: 2017-05-05T09:00:00.000Z, user_id: 12 }, RowDataPacket { id: 212, name: 'Running', category: 'Excercise', start: 2017-05-05T09:00:00.000Z, end: 2017-05-05T11:00:00.000Z, user_id: 12 }, RowDataPacket { id: 272, name: 'Homework', category: 'Productive', start: 2017-05-05T14:00:00.000Z, end: 2017-05-05T16:00:00.000Z, user_id: 12 }, RowDataPacket { id: 172, name: 'Study', category: 'Productive', start: 2017-05-05T20:00:00.000Z, end: 2017-05-05T21:30:00.000Z, user_id: 12 }, RowDataPacket { id: 222, name: 'WeekTest1', category: 'Social', start: 2017-05-06T10:00:00.000Z, end: 2017-05-06T12:00:00.000Z, user_id: 12 }, RowDataPacket { id: 232, name: 'WeekTest1', category: 'Excercise', start: 2017-05-07T10:00:00.000Z, end: 2017-05-07T13:00:00.000Z, user_id: 12 }, RowDataPacket { id: 242, name: 'WeekTest1', category: 'Social', start: 2017-05-08T10:00:00.000Z, end: 2017-05-08T11:00:00.000Z, user_id: 12 }, RowDataPacket {</pre>

	<pre>id: 252, name: 'WeekTest1', category: 'Productive', start: 2017-05-08T14:00:00.000Z, end: 2017-05-08T16:00:00.000Z, user_id: 12 }]</pre>
Error Response	<p>Code: 404 Content: CANNOT GET /tasks/days/analyze/alltime/get</p>
Sample Call	<pre>\$.get('/tasks/days/analyze/alltime/get/', function(data, status) { });</pre>

Endpoint 12: Logout

URL	/logout
Method	GET
URL Params	N/A
Data Params	N/A
Success Response	<p>Code: 200 Content: Uses PassportJS's logout() function to log the user out of session.</p> <p>The 'index.ejs' page is rendered after logout is completed.</p>
Error Response	<p>Code: 404 Content: CANNOT GET /logout</p>
Sample Call	N/A

Testing

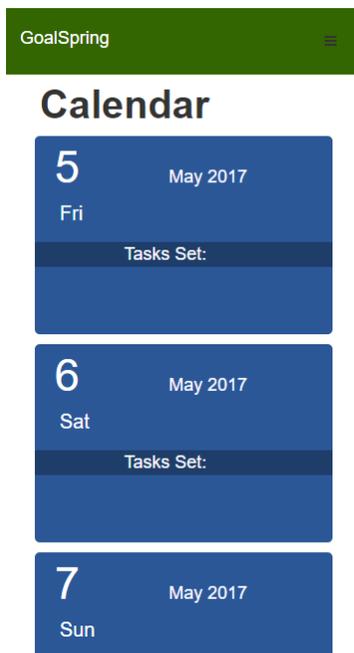
Responsiveness

Responsiveness was a key focus for the project. I wanted my website and application to be optimized for smaller devices first (mobile first). Thus, a lot of testing was carried out in order to reach this goal.

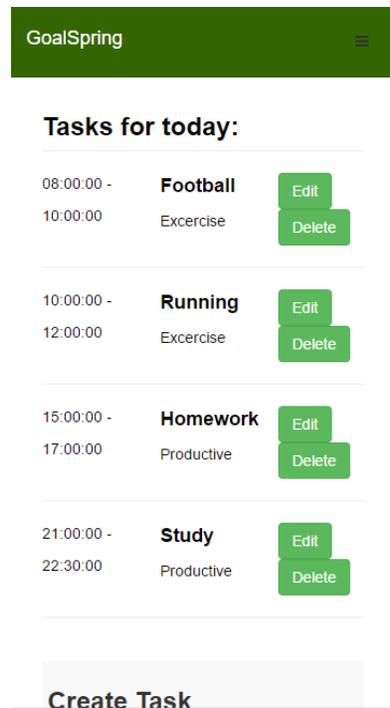
For the purposes of this test I tested the views of the main user interactive pages of this project. Namely the calendar view which I discussed during the implementation section was a struggle to ensure responsiveness, and the task view which the user will be interacting with the most during the project.

First Test: Mobile (iPhone 6s)

The iPhone has a screen width of 414x736 and is perhaps the most popular device at this pixel range. As mobile first is a priority of the project, ensuring that the application looked good on a device that was the most popular smartphone by a “majority” in 2016 (Chang, 2017).



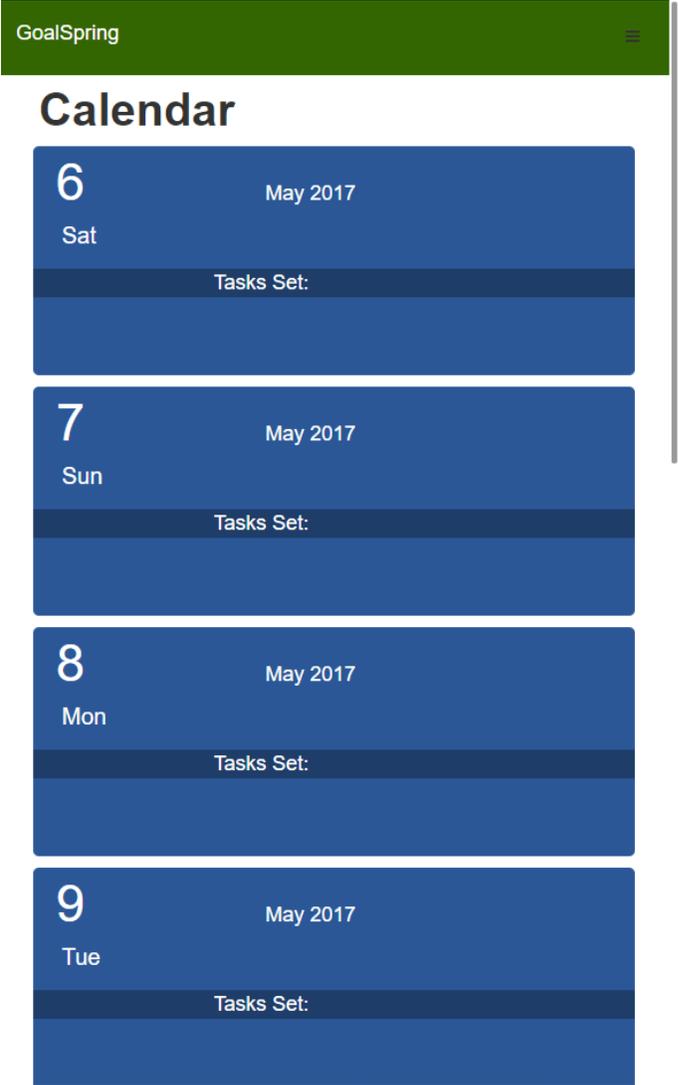
As demonstrated, the iPhone can view the calendar screen without difficulties. The navbar has collapsed into a drop-down menu on click.



Here Is the task view. I was not happy with how close together the buttons for “Edit” and “Delete” were during testing and this is something I will need to fix in the near future.

Second Test: Tablet

For this test, the website’s UI was tested on a Nexus 7.



As displayed above, the calendar UI looks like what you would see on the mobile but slightly enlarged. Overall, I’m pleased with how it looks.

GoalSpring: Technical Report

The screenshot displays the GoalSpring mobile application interface. At the top, there is a green header with the text "GoalSpring" and a hamburger menu icon. Below the header, the section "Tasks for today:" is visible. It contains four task entries, each with a time range, a title, a category, and two action buttons ("Edit" and "Delete").

Time Range	Task Title	Category	Actions
10:00:00 - 12:00:00	Running	Exercise	Edit, Delete
11:00:00 - 13:00:00	WeekTest1	Social	Edit, Delete
21:00:00 - 22:30:00	Study	Productive	Edit, Delete
22:30:00 - 23:30:00	Football	Exercise	Edit, Delete

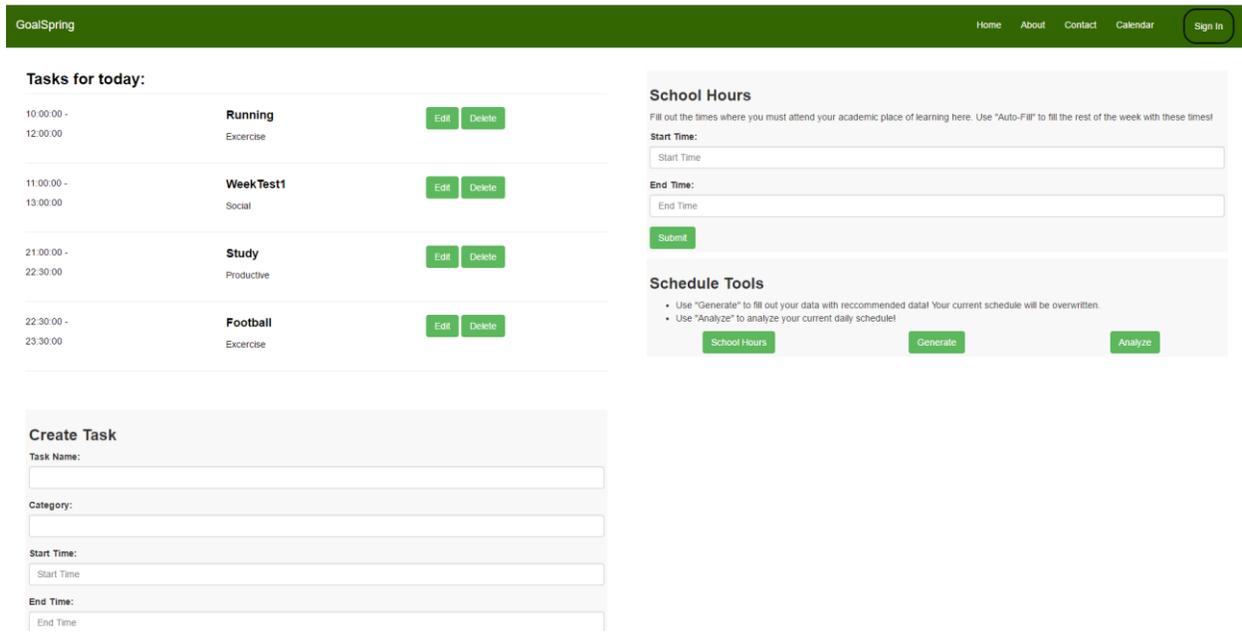
Below the task list is a "Create Task" form with the following fields:

- Task Name:
- Category:
- Start Time:
- End Time:

I feel like the tablet view is somewhat restricted by the collapsing navbar. I don't think it's entirely necessary and I may investigate possible changes with it soon. Other than that, I cannot complain with how the tasks view looks on a tablet screen, I'm pleased with the responsiveness.

Third Test: Desktop

For testing purposes, I used a twenty-three-inch Acer computer monitor.



The view is zoomed out for the purposes of fitting the image in this document. As displayed, the user can see most functions on the page. I'm happy with how this turned out, as I was initially worried that focusing too much on "mobile-first" design would mean that I would struggle to create a good user experience for desktop users. Changes I would make in the future would possibly be to separate the "School Hours" and "Schedule Tools" divs slightly further vertically apart from one another at this screen size, as there is a lot of unused white space below them.

Usability Testing

A focus of my application was ensuring that the design was fluid throughout and that user experiences were always satisfactory. I wished for one of the key selling points of this app to be that it felt "good" for the user to interact with. Thus, I placed emphasis on Usability tests. Using multiple people, I performed five second tests and think out loud testing.

Participants were as follows:

- Oran
- Aisling
- Thomas
- Conor

- Deirdre

Five Second Test:

Upon viewing the website for five seconds, each user was asked to recount their experiences and what they were currently feeling when viewing the webpage.

Oran (Homepage)

“I liked the colour scheme, the green colour appeals to me as a sort of success colour. I found that all buttons and functions were self-explanatory.”

Aisling (Calendar/Profile View)

“The calendar view I looked at was quite self-explanatory, I didn’t really have to think of where I am on the page if that makes sense. I was aware that this was the section of the website that I needed to click on a date for.”

Thomas (Task View)

“I feel that the task view is a bit plain. Usually I’m a fan of minimalistic design, and this is minimalistic and I like that aspect of it. But it just feels bland, just a bunch of information and some buttons.”

Conor (Analysis/Data Visualization View)

“I feel this opinion may be tainted by the fact that I just love looking at data visualization and stats and things like that, but I loved it. All charts were clearly labelled and I knew exactly where these charts were telling me. A++.”

Deirdre (Edit Task View)

“It’s very simple like. The colour scheme is grand, the form is grand, I know exactly where I am and what I’m meant to be doing. Is there supposed to be more to say about this?”

Think Out Loud Test:

For this test, Oran volunteered to participate. We both discussed what I envision would be a usual user visit to the website. Starting off on the homepage and logging in, we’d visit the profile page and click on

the current day. Within this day, we'll both create a task then generate a schedule. After this, we'll return to the Profile screen and request a weekly analysis of our tasks. After which, we'll logout.

The following dialogue is Oran as he visits the website:

“So I've arrived on the homepage and everything looks and feels like I'm browsing a normal webpage, there's nothing that I can comment on, it's not spectacular but I can see my login and signup options pretty clearly. As I scroll down the homepage I see more advertising for the features of the website and what devices they work on. I've clicked on the login button and that's brought me to a new screen, I don't see why the login form can't be on the webpage it just feels a bit unnecessary to have it on another page when there isn't much going on the homepage.

I've logged in and I've been greeted with a calendar screen. Looks good. I'm browsing this on my tablet and it's clear to see what days are which and vice versa. As I scroll down I can see further options to implement my times of education and a weekly analysis tool. These aren't too clunky and are clearly separated. I'm going to click on today...well that just seems a bit bland. When I clicked on it there was no animation or something to assure me that I've clicked on something, it's just brought me to a new page.

I like the colour scheme as I said during my five second test, I'd just like to repeat that statement. I love the green and white, it's simple and there aren't colours all over the place. I'm currently on what looks like my task screen..ah yes it is. I can see there are tasks planned and most importantly they are all in order. So if I click generate...ah yes I've had more tasks inserted in with recommended topics, great! I like this, it feels very intuitive, like I have control over everything kinda if you catch my drift.

I've backed out to the profile screen which was available via the dropdown menu on the navigation bar. Easy to navigate this website also. I've clicked on the analyse weekly button and it's brought me to a page with all sorts of charts. I love little stats like these so this is pretty cool. The colours are bright and I'm able to distinguish easily what the charts are trying to inform me. I really like it!

After this I could logout via the navigation bar once more. Perhaps the user should have a little button at the top of each screen for the logout? I had to kinda search for that. Overall though I like the feel of the website other than the small things I mentioned earlier about the logins and the blandness of the calendar view.”

Conclusion and Further Development

I feel that the world of web development is an extremely active one, and is field that will become increasingly important soon as e-commerce and social media continue to dominate the internet. Google's focus on creating "progressive web applications" rather than native mobile applications signalled their prediction as to how the technology will progress over the next number of years (developers.google.com, 2017). Thus, I felt creating GoalSpring as a Web Application optimized for the mobile device was a good option and I'm content with how it turned out. I feel like the UI especially is quite aesthetically pleasing to look at, and the analytical tool will assist me and others in figuring out where they spend their time.

I felt somewhat hindered throughout the project by distractions in the form of my other college modules and personal life. I had originally intended to develop this application using the React JavaScript framework by Facebook which has been growing huge in popularity in recent years, overtaking Angular as the most popular JavaScript framework in 2016, with a rise in popularity of 300% per a Stack Overflow survey. Thus, I wanted to use this framework in my project. However, I felt like the limited time I had to dedicate to the project did not match well with the time needed to learn the framework, thus I used JQuery which I already had a great knowledge of. Interestingly, the same survey shows that the JavaScript + SQL combination is the most popular two language combination, which means that my decision to use MySQL over MongoDB as my database system may have been somewhat justified.

I feel however that with more time, that I could make the user experience of GoalSpring even better. The React JavaScript framework could make this a single-page application with fast loading times and smooth transitions between page states. I could also further enhance the Artificial Intelligence of this program to somehow make use of the alpha-beta pruning technique to make it even more smarter.

However, despite the time constraints I enjoyed immensely developing this application. I was introduced to many aspects of JavaScript that I had not explored before, including using JavaScript for Data Visualization and Artificial Intelligence which interested me and allowed me to feel passionate about the work I was doing.

National College of Ireland
BSc in Computing: Software Development Stream
2016/2017

John Deegan
X13327936
X13327936@student.ncirl.ie

GoalSpring
Project Proposal



Appendix One: Project Proposal

Objectives

Develop a System for tracking and learning habitual behaviours.

This web application, GoalSpring, will allow user to enter their day by day activities and draw up statistics and conclusions from the statistics and recommendations that the app draws up.

Increase Productivity and Learning

This app is geared towards all people who wish to gain a better understanding of their habits and how they can tweak them to improve their quality of life. Stephen R. Covey writes in his famous 1989 self-help book “The 7 Habits of Highly Effective People” that humans can develop themselves through an eight-step process, “Sow a thought, reap an action, sow an action, reap a habit, reap a character; sow a character, reap a destiny”. What this system aims to do is to help users fulfil this eight-step process easier by giving them a more manageable outlook on their habits and tendencies.

Background

Procrastination is a problem in the modern world. As technology advances, more and more ways of getting quick “fixes” of pleasure, or dopamine rushes in the brain are available to the public. To contrast, in 1978 only 5% of the American population claimed to be procrastinators, and this figure subsequently “rose to 26% by 2007” (Tremayne, 2015) and is on the rise thanks to the invention of smart devices such as the iPhone.

The main thinking behind the Goalspring project is to harness the addictiveness of these technological devices and use it to combat the procrastination problem in society. Attack the root to solve the problem. Goalspring aims to give procrastinators a short term “good feeling” that they normally associate with unproductive tasks, and instead attempt to associate these feeling with productive habits. A study carried out by Fuschia Sirois, a professor at Bishops University in Canada carried out a study on 80 students with the aim of comparing the answers they gave to a hypothetical worrying situation. One of the questions asked related to a hypothetical situation where you have returned from a sun holiday to discover a new mole, but you put off visiting the doctor for many months and now you’ve been diagnosed with a serious illness. Sirois discovered that self-proclaimed procrastinators tended to respond with answers such as “At least the illness was discovered now before it could get any worse” whereas non-procrastinators responded more along the lines of “I really wish I had gone to the doctor sooner.”. From this study, Sirois concluded that procrastinator’s responses are more positive to make them feel better in the short term, and thus they are doomed to repeat their procrastination as they only associate good feelings with the downsides of their procrastination. Whereas the lamenting of the non-procrastinators shows that people who fall into this category are more likely to remember the consequences of their actions, mainly the bad feelings, thus attempt to “fix” their mistakes so it does not happen again. (mytimemanagement.com, 2016). Goalspring wants to take this desire for short term pleasures and change it into a force for

remedying procrastination and rewarding task completion. Users can see their progress and achievements and feel good about them, safe in the knowledge that they are “reaping action” as explained by Convey in his book, all the while being in a traditionally unproductive environment such as the internet.

The reach of the market Goalspring can reach is broad, but is currently aimed at younger people and school goers. Employers can use the system to boost workplace productivity, which despite a technological boom and the backing of trillions of dollars, has only grown 1% in the USA since the “tech boom” began (Harvard Business Review, 2016). By encouraging employees to focus on general well mental well-being habits such as getting enough sleep, eating well and mindfulness, productivity and happiness can rise in the workplace, leading to a more productive environment. The system can even be used on young students, with over 4,000 students failing Leaving Certificate Maths in Ireland in 2016 (Irish Times, 2016), these figures could be reduced by encouraging children to use the system to cultivate healthy habits themselves to boost grades, just Studying could be a habit!

Technical Approach

Front-End

Mobile first. It’s a buzz word in the web development community, but not one without substance or merit. Per marketingland.com, in December 2015, 65% of Digital Media users were accessing the internet using smart devices rather than desktops. As such, working with screens with small pixels is the number one priority for this project. Of course, that’s not to say that desktop users will be ignored, as 35% of Digital Media users is still many people to cater for, it’s just that the mobile users will be the priority. The latest front-end web languages such as HTML5, CSS3 and the ever-expanding open source world of JavaScript allow us to cater to these smart devices.

HTML5 introduced some great meta tags such as viewport, which allows us to define different viewport widths and zoom settings, full screen browsing and home screen icons to easily save your favorite webpages to your smart device’s home screen. This project will aim to use these features to their fullest extent, in addition to HTML5’s cleaner code and accessibility features.

As responsive design, has become a priority in recent years alongside the smart device boom, the world of CSS has expanded to cater for this demand in useful ways. The latest version of CSS introduced media queries, which allow for specific and more detailed control over how our styled elements look on specific screen sizes. The open-source community has built frameworks such as bootstrap and foundation to make it even easier to create elements and components that are designing for mobile viewing. This project aims to use these new features and frameworks to their fullest extent, making sure the user experience looks good on smart devices with varying screen sizes (iPhones have a significantly smaller screen than some iPads for example). I aim to use Bootstrap and their grid system to efficiently plan the layouts of my webpages, in addition to using their built-in classes for more efficient element styling.

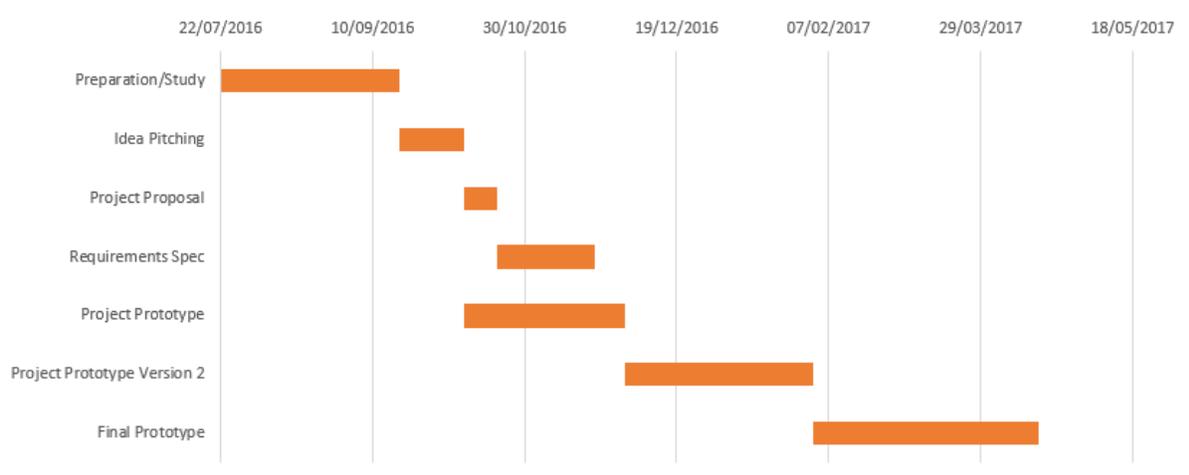
Like CSS, JavaScript is benefitting from a strong open source community and a rapidly expanding collection of frameworks. This will be especially useful in building strong user interactions with my webpages, such as smooth transitions from a user clicking a button to create a new goal and a form appearing. JavaScript will also be used to implement Artificial Intelligence for the generation of user schedules.

Back-End

For the back end of the website I aim to use NodeJS for my web server. This means less “shuffling” between languages as I would be using JavaScript for the Front-End as well. Not only this, nodejs is also extremely quick in code execution as it’s built on Google Chrome’s JS engine. It’s highly scalable and uses a single threaded program to handle a larger amount of requests in comparison to more traditional servers such as Apache.

For Database requirements MySQL will be used. MySQL is a familiar database system that I have used during college which will allow for more efficient work. I chose this over MongoDB for exactly that reason, not having to learn a new system which will take up valuable development time.

Project Plan



Proposal Conclusion

GoalSpring is a well-intentioned project. It aims to better the lives of people all over the world by helping them learn their habitual patterns and attempt to change them. The use of a mobile first strategy in design will allow greater accessibility on all devices that the website is viewed on. NodeJS and MongoDB will handle our scalability concerns in addition to keeping our focus language wise on JavaScript

Appendix Two: Monthly Journals

Name: John Deegan

Student Number: x13327936

Course: BSHC

Stream: Software Development

Month: September 2016

Hi there,

I'll be honest, I don't like journaling or blogging. I've tried a few in my teenage years when I thought my opinion was something people should care about, like Gandhi preaching to the Indians or Graeme Souness anytime a camera is afoot. Truth is, not many people care enough to read your opinions on the current situation in the middle-east, or your latte. Speaking of which, I'm enjoying one right now in a South Dublin Starbucks. This is my outlet for a year in which I'll probably go a little bit crazy, something to look back on in a few years on my yacht in Monaco and cringe at. Overall I think it'll be an entertaining experience to share with you all.

So what am I doing on this momentous Day 1? I bought a JavaScript course on Udemy that's just fantastic. For only a tenner I get 16 hours of lifetime React tutorial content. Considering I'd otherwise spend that tenner on a burrito or craft beer (yes I am one of those people), I feel it's a great investment. I've a solo project to complete for college by next May using a technology that I've self-learnt, So I'm planning a web application due to its popularity in the development world at the moment.

September Week 1

I'm learning how to use git and github properly. I don't know how to use them properly which from what I read online is somewhat of a criminal sin for aspiring developers.

College is starting back in two weeks yet I've accepted some freelance work. I'm the type of person that laments any time wasted not learning or preparing for something, be it college, a date or even a game of FIFA. A good friend of mine runs a radio society in a local university and they recently fell victim to some domain squatters. The website just screams "please come begging on your hands and knees for your domain back", it's full of filler, adsense keywords I can only presume, and it's coded in XHMTL for goodness sake. Thus, they need a new website, and he's asked me to urgently create a basic homepage within two weeks, as that's when the radio show returns for the academic year. It'll be a challenge, but I see it as good practice for the upcoming year, not to mention being paid and getting invaluable experience.

September Week 2

So yeah all hell is breaking loose.

I've a website to build. College to prepare for. Friends to maintain. Girlfriend to date. Bills to pay. Life is hectic right now. I'd been preparing for it for a while, I don't watch it but I believe the TV show Game of Thrones has a saying that goes "Winter is coming".

I've been topping up my Java skills in the little spare time I have. My college course tends to be very Java centric, which I dislike. I just love JavaScript so much more, Java feels very "serious and proper", kind of like the golden child of the programming languages whereas JavaScript feels like the unwanted rebellious child that breaks the conventional rules. I love it.

By this time next week, I'd have started my final year of college and will probably have about 9000 more things to do. That's the test of how prepared I am for it. Bring it.

September Week 3

My schedule is perfect. Monday is dedicated to my big project, more on that later. Tuesday is a full day of labs and lectures; Wednesday and Thursday have a two-hour lecture each which allows me to spend the rest of those days in the library studying and working on my projects and labs. I take Thursday evening off from three o'clock onwards, this is a system I've seen several MMA fighters use, three days working as hard as you can, one day off. I spend that evening with my friends or my girlfriend, and that allows me to come into my full day (yes, no break from 9-5 because of a careers class at lunch) refreshed and ready to go. I also take Saturday and Sunday evenings off. When I'm actually studying any of those days, I study for 45 minutes, take a 15-minute break for three hours until I take an hour break. This schedule works fantastically for me thus far.

September Week 4

It feels like everything is kicking off at the same time. Projects, assignments, grad fairs. It's like a never-ending game of scheduling and managing myself and my time. I was ill this morning with tooth ache, headache and a sore throat, I decided that staying in was probably a wiser decision than going outside considering that I'm going to a tech fair in Dublin this evening, a more generic grad fair tomorrow and presenting my project idea to a panel of lecturer's tomorrow afternoon. I feel like there's a lot of truth to the "make no excuses" cliché, but sometimes you must play it safe.

As for my project presentation, it's a short one. 2 minutes long and no powerpoint, just me and my gab. As mentioned previously, I'm hoping that they don't mind that my idea isn't exactly one that will make a lot of money yet, but it's more of an idea that will allow to me to show off my skills to a potential employment suitor. I'm worried because the college keeps talking about "making something that's marketable". My housemate is a pretty good salesman so I've been asking him for advice on subtleties like body language and the actual language you use whilst pitching. I'm confident.

GoalSpring: Technical Report

As for my studies, it's been tough. I spend all day in the library until around 8pm. I'm thinking of cutting this back to 7 for my own mental state of mind, giving myself only three hours (one of those to commute and cook dinner) to wind down before going to sleep seems to be somewhat of a stumbling block. I'm ahead of time of course work so far and I feel to not burn out I need to halt myself somewhat. It's three weeks until I have a week off college so it's important I don't hit a wall in between then and keep up this momentum that I've created.

Post Presentation

I left my presentation more confused than UFC fighter Jose Aldo when Conor McGregor knocked him out cold in 13 seconds. From what I could gather, I was told that the technologies I was planning to use were fine, but my idea was not complex enough. A habit creating web app is a "6th year schoolboy title" I was told. In addition, what I feared had occurred, I needed a unique selling point for my application that has a purpose. If you're confused about that last sentence, then I don't blame you. The three lecturers were very nice in the way they delivered these verdicts to me, but I'm still very confused at what exactly it was that they were trying to explain to me. I wasn't told either if my idea was accepted or rejected, yet other students were. I know I keep saying It but everything is very confusing now, and I hope by next week it's all cleared up so I can get up and running with creating my project.

Name: John Deegan

Student Number: x13327936

Course: BSHC

Stream: Software Development

Month: October 2016

I'm on a bus somewhere on the N11. Despite it being October, the sun is glaring through the windows at my face, thankfully the air conditioning is great on Bus Eireann. It's slightly more expensive than the local operator, Wexford Bus, but having plug sockets in the modern world is well worth the extra euro or so.

This is the first time this year when I feel the usual creeping up of assignments and midterm exams etching away at me. I have a Java Chess Game to figure out, an essay on Cisco's business model, a Data Mining exam and a Mobile exam to boot. All of this is before I submit my plan for my project idea (which was accepted on review by the way!). It's more important than ever that I keep my mind in the right place now, and block out the negative self-talk. I have been here before and succeeded and that's when I wasn't as prepared as I am now.

I've been spending a lot of time watching MMA fighters and their training camps. I feel it takes a certain degree of mental toughness to walk into a cage where the other guy is trying to incapacitate you that you just don't quite get in other disciplines of life. If I can learn from their preparation, it can help me become a better coder and a better student. Coding isn't a physically taxing task, it's all mental. How long can you look at a screen before your mind tells you enough is enough. With UFC or MMA fighters in general, they must emphasise the mental training as much as the physical, if a fighter walks into the Octagon worried about a fight they had with their girlfriend or forgetting to take out the trash, they're at a huge disadvantage to their opponent, no matter what type of physical condition they're in. To me, during this taxing year there's no better group of people to learn from than MMA fighters. I recommend a good video series on Chris Weidman's preparation to defend his middleweight belt against Vitor Belfort that you can view [here](#).

I have a week off the week after next, so if I can push myself as hard as I can go for these next two weeks I can take a well-deserved break at home after. I need to spend that week getting my productivity app off the ground as I have a prototype due for before Christmas. It's important that this week I practice mindfulness, and ensure that I meditate twice a day to keep my mind focused and off other distractions. I also need to ensure that I don't burn out and listen to my body and my mind. If I allow myself to get burnt out or stressed, I won't accomplish these tasks to the level that I know I can perform at. See you on the other side.

Name: John Deegan

Student Number: x13327936

Course: BSHC

Stream: Software Development

Month: November 2016

This month has been extremely difficult. The workload this year is quite frankly insane and there's been little time to breathe never mind do adequate work on my project. Progress has been too slow for my liking and overall I will need to try and get a suitable prototype ready for December.

I've thus far managed to make the front-end responsive and have begun coding my API and Database integration. The colour scheme for my website is looking good and I'm confident that the finished product will be very polished and user friendly. It's just finding the time to complete and polish it off has been limited, so I feel that focusing on the backend this month has helped the project progress more. It won't look pretty when I present it, but at least the lecturers will be able to see what the back-end is doing and I feel that's what most important right now.

Over Christmas I plan to put in a huge amount of work for my project whilst I study for my exams. Completing the 4 projects that I have due for Christmas will take a huge load off my back, which will open time for GoalSpring so I can make it all I want it to be. My work this month has made me extremely passionate about this project as I very much enjoy developing web projects, especially on the front-end.

By January I hope to have the back-end almost complete. This will allow me to focus solely on making the front-end as user friendly as I can possibly make it. I will aim to have most of the project completed by the time semester two and the inevitable horde of assignments slows down progress again. I feel I am better prepared for the onslaught of assignments now, and that I can plan work for the project around it better now because of my experience from this semester.

Name: John Deegan

Student Number: x13327936

Course: BSHC

Stream: Software Development

Month: December 2016

This month mainly focused around my mid-point presentation. It was stressful as a lot of deadlines were approaching for the end of the semester and there was little time to prepare my project to the standards of which I wanted it to be at this point. I had hoped to have had a simple working prototype with maybe around 70% of the features I wished to include in my final version. However, by the date of the presentation I had maybe 30% of what I wanted to include in my project completed.

The prototype version of the project had the front-end design almost completed. The colour schemes and the general feel of the project have been established. I have implemented some basic JQuery which allowed me to set some conditions for displaying different hints and tips to help the end user create a more balanced schedule, such as if the user wanted to do a lot of social activities during the day, the application would alert the user as to not neglect looking after their health or their studies.

I had hoped to have implemented some Artificial Intelligence by this point to generate a schedule for users, but I decided that having an extremely simplified version to show the lecturers what I'm trying to do would be the best I could achieve at this moment in time. By the time the project is done, I hope to have programmed intelligence to generate schedules for the user. I also hope to have a polished front end full of animation and a good "feel" to the user interaction.

The presentation itself I felt went well. I had created my slides which basically told the story of my development so far. I went through how I started thinking and preparing for this project only four days after my 3rd year work placement finished in July. How I struggled to convince the college to allow me to proceed with my idea that I felt would most show off my full-stack JavaScript skills. How deadlines and continuous assessments hampered my own project schedule. There were little questions posed to me, and they were mostly related to what the final version of my project would look like. I took this as a positive sign, that there were no glaring errors in my project that caught their attention, and that the finished project should receive a good grade if my post-Christmas schedule goes to plan.

After Christmas, my exams take priority, but after that I hope to dedicate the rest of January and February to getting as much of the project completed as possible, and getting to the "70%" milestone I talked about earlier. If I can implement the basic AI for generated a schedule, and gloss my front-end design somewhat, I would be happy!

GoalSpring: Technical Report

Name: John Deegan

Student Number: x13327936

Course: BSHC

Stream: Software Development

Month: January 2017

Happy new year!

I managed to complete a satisfactory amount of work this month. I managed to find a way for my Database to work alongside my API and connect with the front-end's JavaScript via AJAX requests. I had read online that this shouldn't be possible without PHP, but NodeJS tends to possess a lot of magic when it comes to things that you shouldn't be able to do and things that you should.

I had a reshuffle of my front-end design, making the font bigger and more prominent, especially on mobile devices. I've made the mobile design much more friendly. At the time of writing, I'm aiming to make my calendar/schedule page a lot more mobile friendly. I'm looking to include a lot of "block" design, like you would see in a lot of online applications. Reading the book "Don't make me think" by Steve Krug has been a huge help in reshuffling my design from the prototype stage to the final product.

As for the AI, I feel as if leaving this aspect until last is a wise decision. I feel like there is too many variables and aspects that could change as I alter my database and design that could affect any AI Agent's decisions. The generating a schedule aspect of the project however is huge and I must begin work on this shortly. I research some JavaScript AI for a simple chess game and it seems manageable, it expands on what I learned in my AI module from last semester which was in Java, but the concepts remain the same.

Looking forward I feel as if I'll shorten my work on the database, as I know the syntax for the GET, UPDATE, DELETE and PUT methods and it's a matter of just slotting them in after I complete the design and how exactly the information will be relayed to the user.

Speaking of the design, for next month I hope to have the front-end pretty much completed apart from the AI. I also hope to have started my AI.

Name: John Deegan

Student Number: x13327936

Course: BSHC

Stream: Software Development

Month: February 2017

This month I wanted to further enhance my project by learning EJS, or embedded JavaScript, to allow for pure JavaScript communication between my client and server. This would allow me to easily display data from my database onto the webpage easily. I also began to learn the ExpressJS framework with NodeJS fully to utilize all the features of the framework for my back end.

Observations

I felt I made great strides in learning my frameworks quickly. I managed to fully utilize the Express and NodeJS frameworks to retrieve data from my database and display it on my webpage using EJS.

I realised how important a good working environment can be to people working in the software industry. During the span of this project I've noticed the small things such as having good posture and the use of multiple screens can hugely benefit the efficiency of working on a project.

One problem I had during this month was trying to find a date picker for JavaScript. The problem is that even though HTML5 has a type="date" functionality built in, it doesn't work on a lot of browsers including all versions of the popular web browser Firefox. Thus, I managed to download a JQuery plugin called "datepicker" which does what it says, brings up a calendar like interface on click which allows a user to select a date which is then transferred on submit to my server in String format. From here, I use string manipulation to format the string to one which is acceptable by schemas which uses MYSQL DATETIME, which is presented in a different format to JavaScript dates.

I managed to complete some artificial intelligence this month. It's not up to the standard that I want yet, but that will come in time now that I've finalized the date picking functionality

GoalSpring: Technical Report

Name: John Deegan

Student Number: x13327936

Course: BSHC

Stream: Software Development

Month: March 2017

This month saw great strides into completing this project. I've completed the AI for the project for the most part, I'm hoping that between now and the presentation that I can improve on it further. I introduced some interesting data visualization for the project and I fully completed the backend. We're almost there!

This month was another where I felt my work load for this semester affected the development of my project. I have three projects and an exam all to complete for the end of April and this meant that my project took somewhat of a backseat to my modules.

However, I was happy with the work that I did manage to complete. I managed to get the design that I wanted the calendar and managed to manipulate the strings from the MySQL datetime to assign them to the calendar "boxes" on my task pages.

I feel I am very accomplished with using the NodeJS and ExpressJS backend framework. I would very much love to delve deeper into those frameworks even after my project is completed, and perhaps even look for employment using those languages.

This month I also carried out various usability tests with my project, overall I'm happy with how it turned out.

From Here...

I hope to refine my AI before the presentations. I feel like it's very error prone in some cases. I also hope to do a few things with the user interface of the project, as I feel there can be tweaks here and there such as the size of the buttons on the home screen and the tasks view.

Bibliography

1. Carl O'Brien. 2016. CAO offers: Sharp points rise for courses linked to recovery. [ONLINE] Available at: <https://www.irishtimes.com/news/education/cao-offers-sharp-points-rise-for-courses-linked-to-recovery-1.2763747>. [Accessed 22 April 2017].
2. thenextweb. 2017. Universities finally realize that Java is a bad introductory programming language.. [ONLINE] Available at: https://thenextweb.com/dd/2017/04/24/universities-finally-realize-java-bad-introductory-programming-language/#.tnw_uiW49rmT. [Accessed 20 April 2017].
3. Eric Jaffe. 2013. Why Wait? The Science Behind Procrastination. [ONLINE] Available at: <https://www.psychologicalscience.org/publications/observer/2013/april-13/why-wait-the-science-behind-procrastination.html>. [Accessed 16 October 2016].
4. mytimemanagement.com. 2016. Procrastination Statistics. [ONLINE] Available at: <http://www.mytimemanagement.com/procrastination-statistics.html>. [Accessed 16 October 2016].
5. Harvard Business Review. 2016. The Paradox of Workplace Productivity. [ONLINE] Available at: <https://hbr.org/2016/04/the-paradox-of-workplace-productivity>. [Accessed 16 October 2016].
6. The Irish Times. 2016. Leaving Cert: More than 4,000 pupils fail maths . [ONLINE] Available at: <https://www.irishtimes.com/news/education/leaving-cert-more-than-4-000-pupils-fail-maths-1.2758421>. [Accessed 16 October 2016].
7. Marketing Land. 2016. All digital growth now coming from mobile usage -- comScore. [ONLINE] Available at: <http://marketingland.com/digital-growth-now-coming-mobile-usage-comscore-171505>. [Accessed 20 October 2016].
8. Stack Overflow. 2017. Stack Overflow Developer Survey 2016 Results. [ONLINE] Available at: <https://insights.stackoverflow.com/survey/2016>. [Accessed 01 May 2017].
9. Google Developers. 2017. Web | Google Developers. [ONLINE] Available at: <https://developers.google.com/web/progressive-web-apps/>. [Accessed 01 May 2017].
10. The Irish Times. 2017. The key to being a successful student . [ONLINE] Available at: <https://www.irishtimes.com/sponsored/exam-times/study-guide/the-key-to-being-a-successful-student-1.1611997>. [Accessed 12 November 2017].