FINAL PROJECT TECHNICAL REPORT

National College of Ireland

SUBMISSION DUE DATE:
16/05/2021

CARLOS KOUAKOU BARBIER
X17117852@student.ncirl.ie

Contents

1. Introduction

The objective of the present document is to provide a general overview and technical description of my final year project: Code Avec Moi. Firstly, the background and context of the project will be presented explaining the aims and scope of the project and the high-level deliverables. The technology will be detailed explaining the programming languages, frameworks and methods used in the project. Likewise, a detail breakdown of all the functional and non-functional requirements for the project are specified, giving a blueprint of the main features of the application along with the usability and functionality required for the end-user. Furthermore, the System Design and Architecture is explained using diagrams and figures to illustrate the flow of the application followed by the User interface design, were the site map and wireframes of the application is. In t presented. Finally, the strategy and testing carried out is explained, to conclude with the general findings and further development of this project.

1.1 Background and Context

Education is a powerful agent of change. It improves health and livelihoods, contributes to social stability, and stimulates long-term economic growth. According to UNESCO, 420 million people would be lifted out of poverty if they had a secondary education. As we can notice the education is the key factor of any development in our society. Today, with the ease of access of internet and technologies, people are changing the way they are learning, the skills and knowledge can be transferred easily to anyone and anywhere in the world.

Six million students around the world pursue formal higher education fully online, representing a $30B market. This number do not consider all the E-learning platforms or Online course available which will make the market larger. There are more than 100 million online learners nowadays, and likely who have taken a short online course on Lynda.com, Udemy, Eduonix etc. The entire market is projected to be worth $325 Billion in 2025, and Africa, who that has the youngest population in the world, will reach $2073 million. Looking closer, we can conclude that Africa is the smallest E-Learning market in the world to date, in one word there many to do in the education sector and E-learning is a powerful tool that can help to decrease this gap between Africa and the rest of the world.

1.2 Aims

The aim of this project is to create an online learning platform specialized in Information Technology Courses. This platform will be developed for the French African Market with the intention to provide accessible and affordable resources that help the population to develop the required skills in IT. The application will be design in the way that it helps the instructors to improve the content of their course but also can recommend courses to students.

1.3 Project Scope & Deliverables

The scope and deliverables defined for this project are listed below. These deliverables will be described at a high level as the technology and specific project requirements are explained in detail later in this document.

- For the back end, a full functional API implemented in Node JS to manage the application that interact with MySQL database.
- A front-end web application to consume the actual API developed in NextJs which is a React framework.
- A document to explain how to run locally the application.
- Structured documentation & deliverables at all phases of the project.

2. Acronyms and Abbreviations

The below will help to understand some terms, acronyms and abbreviations used throughout the document.

- **API:** Application Programming Interface, which is a software intermediary that allows two applications to talk to each other.
- **HTML:** Hypertext Markup Language (HTML) is the code used to structure a web page and its content.

- **CSS:** Stands for Cascading Style Sheets and is a language used for describing the style of a document such as page layouts, fonts, or colours, written in a markup language like HTML.
- **JS:** Is the abbreviation of JavaScript.
- **JavaScript:** Is a scripting language used in the client-side where the source code is handled by the browser in the client side instead of the web server.
- **Java:** Is a high-level object-oriented programming language created by Sun Microsystems being one of the most popular languages to create web applications.
- **Back end:** Refers to the part of an application responsible for storing and managing date and is not directly accessed by the final user.
- **Front end:** Is the part of a system or application where the user interacts and receive the back-end functionalities.
- **CodeAvecMoi:** The application itself developed for this final project.
- **PCI Compliance:** Payment Card Industry Data Security Standard (PCI) which is a protocol to ensure that all the companies that process, store, or transmit credit card information must do it in a secure environment.
- **IDE:** Integrated Development Environment
- .**UI:** Stands for user interface design and refers to the area where interactions between humans and machines take place.
- **UX:** Stands for the interaction and experience users have with products and services provided by a company.
- **CD CI:** Continuous Deployment and Continuous Integration.
- **CPU**: Central Processing unit
- **MTBF:** Mean Time Between Failures
- **MTTR**: Mean Time to Repair

- **Bootstrap**: One of the most popular frameworks used to develop responsive and mobile first web projects using HTML, CSS, and JS. The design of both the application front-end and marketing website were completed in Bootstrap.
- **Sass:** Is an extension of CSS and stands for -Syntactically awesome style sheets- which enables to create faster style sheets using elements such as variables and nested rules.
- **Typescript:** TypeScript is an open-source object oriented, compiled languages which builds on JavaScript.
- **IDE:** Stands for an *integrated development environment* and is a software to create applications providing tools to facilitate the development.
- **CI CD:** Stands for Continuous integration and continuous delivery, a coding set of principles applied in development and is part of the agile methodology. This principle basically focuses on develop and deliver code changes more frequently and reliably.

3. Technologies

Several technologies, libraries, tools and services are used in this project.

3.1 Programming Language

For this project, I will use various programming languages:

- **JavaScript:** will be used to build the front-end application.
- **Typescript:** will be used to implement the API (backend).
- **Html and CSS:** for the frontend
- **Sass**: for the frontend

Java: to develop an API with Rest-assured which can be executed to make sure that all the API endpoints are working as expected

3.2 IDE

To develop this application, I will use WebStorm. se the full power of the modern JavaScript ecosystem – WebStorm's got you covered! Enjoy the intelligent code completion, on-the-fly error detection, powerful navigation and refactoring for JavaScript, TypeScript, stylesheet languages, and all the most popular frameworks.

3.3 Hosting

The API, the frontend application and the MySQL database will be hosted on the cloud service provider Digital Ocean that offer affordable package. For the videos and images, they will be stored on Amazon S3.

3.4 CD CI

In order to follow the best practices in the software development industry, the CD CI methodology will be applied in this project. GitLab CI/CD will be used for this purpose.

3.5 Framework

In the following paragraphs I will explain the main frameworks used to accomplish the project including the libraries and folders structures when applicable.

3.5.1. Adonis JS: A node JS web framework

AdonisJs is a Node. js MVC framework considered a full stack tool that can run in different operating systems offering a stable environment to create web applications from the server-side and providing great performance benefits.

Libraries used:

- **Adonisjs/auth:** AdonisJs Auth is the official authentication system to use with the framework AdonisJs. This system supports multiple schemes by default to authenticate your HTTP request.
- **Adonisjs/core:** This library gives you the core functionality of AdonisJS this is Fullstack MVC framework for Node.js.
- **Adonisjs/lucid:** Lucid is a powerful tool that works as interface to communicate between the database in use and the application models allowing to manage and

store object attributes and data in relational databases. The models in Adonis applications can be mapped and use database relationships among tables.

- **Adonisjs/mail:** This is the mail provider used in the Adonis Framework supporting the most common e mail providers and services required to send emails.
- **Adonisjs/redis:** This is the official library to with Redis. Redis is an in-memory data store for use as a database, cache, message broker, and queue.
- **Adonisjs/view:** Is a template engine for Adonis. In this project is used to send emails.
- **Adonis-lucid-filter:** This is an addon to use in Adonis and lucid providing the functionality to filter Lucid Models.
- **Aws-sdk:** AWS SDK stands for the software development kit allows developers to connect to AWS components like such as Amazon S3, Amazon SQS, Amazon SNS and DynamoDB, from JavaScript code running in the browser directly (browser-based).
- **Get-video-duration:** This functionality gets you the duration of video files/streams with ffprobe.
- **Luxon:** Luxon is a library that helps you to work with dates and times in JavaScript in an easier way.
- **Mysql:** This library allow you interact with any Mysql database which is an open-source relational database management system.
- **Node-video-duration:** This library allow you to work with NodeJs module to retrieve the duration of assets in video or audio format and works with ffprobe binary.
- **Pretty-ms:** This module help you to convert milliseconds to a human readable string.
- **Proxy-addr:** This module is used to Return the address of the request, using the given trust parameter.
- **Reflect-metadata:** Reflect-metadata Allows you to do runtime reflection on types.

- **Validator:** This AdonisJs package is a powerful tool that validates all the incoming request to help you keep your application safe and avoid common security attacks.

- **Slugify:** This library is an Addon used to generate unique slugs for your Lucid models.

- **Source-map-support:** This module uses the source-map module to replace the paths and line numbers of source-mapped files with their original paths and line numbers with the goal of making every compile-to-JS language more of a first-class citizen

- **Stripe:** This module allows you to interact with Stripe which is the API payment processing platform selected for this project.

- **Uuid:** Stands for universally unique identifier (UUID), this package allows you to generate a unique random string.

3.5.2. NextJS: A React framework

Next. js is a JavaScript framework used to build applications in React. This framework is well known for the fast refresh, and has other benefits such as an improved performance, development process and SEO. Many companies such as Netflix, GitHub and Uber use it to develop their websites. Due to the nature of this project, NextJs give us the best solution to accomplish our goals.

Libraries used:

- **@ant-design/icons:** Ant Design's icons is used to adhere cross-platform consistency.

- **Antd :** This is a React UI library that contains a set of high-quality components and demos for building interactive user interfaces.

- **Array-move:** This package is used to move an array item to a different position. It clones the given array, moves the item to a new position in the new array, and then returns the new array. The given array is not mutated.

- **Axios:** Axios is a lightweight HTTP used to provide a single API for dealing with XMLHttpRequests and node's http interface.

- **Bootstrap-icons:** This is a free open-source icon library in SVGs SVG sprite or web fonts and can be used with or without the Bootstrap framework.

- **Collect.js:** Collect.js is a wrapper to work with arrays and objects. Is dependency free and contains many functions to help us working easier with data and write more concise JavaScript code.

- **Encrypt-with-password:** Encrypt-with-password simplifies encrypting text and data securely with leading security standards. It uses PBKDF2 for key derivation and AES 256-bit for encryption and decryption.

- **Humanize-duration:** Humanize Duration helps us to turn millisecond durations into human-readable strings.

- **Js-cookie:** This package is a lightweight JavaScript API that helps us to handle cookies. It allows the cookies to be created, retrieved, and deleted through a simple and intuitive interface.

- **Moment:** Moment is a JavaScript library that helps us to create, manipulate, and format dates and can be used in Node.js or in the browser.

- **Next. Js:** Is used to make much faster the page loading times. It only loads the JavaScript and CSS that are needed for any given page.

- **Next-cookies:** This is a simple function for getting cookies on both client & server with next.js rendering the pages that depend on cookies.

- **Next-plugin-antd-less:** This plugin supports both Next.js and CRA-Co since v1.0.

- **Nprogress:** Nprogress is a nanoscopic progress bar that help us to communicate to the users that something is happening using featured animations.

- **Query-string:** The Query String in Node.js provides methods to deal with query string and can be used to convert query string into JSON object and vice-versa.

- **Rc-tween-one:** Tween-one is a component for React, an open-source tool in the npm Package's category of a tech stack.

- **React:** React is a JavaScript library for creating user interfaces. This package contains only the functionality necessary to define React components and is commonly used with a React render like react-dom for the web or react-native for the native environments.

- **React-credit-cards:** A slick credit card component for React that support all credit card issuers available in payment.

- **React-dom:** This package serves as the entry point to the DOM and server renderers for React. It is intended to be paired with the generic React package, which is shipped as react to npm.

- **React-google-login:** This is a Google oAUth Sign-in / Log-in Component to use in React.

- **React-js-pagination:** This component is used to render a pagination. This component does not comes with built-in styles but is compatible with Bootstrap 3 pagination stylesheets.

- **React-player:** This component is used for playing different URLs, including file paths, YouTube, Facebook, Twitch, SoundCloud, Streamable, Vimeo, Wistia, Mixcloud, DailyMotion and Kaltura.

- React-promise-tracker:

- This component is used when we need to track blocking promises (e.g. fetch or axios http calls), and control whether to display a loading spinner indicator not.

- **React-rating:** React-rating gives us the ability to create a component for rating supporting custom symbols both with inline styles and glyphicons found in popular CSS Toolkits like Fontawesome or Bootstrap and was inspired by the jQuery plugin bootstrap-rating.

- **React-redux:** React-Redux is a predictable state container designed to help you write JavaScript apps that behave consistently across client, server, and native environments and are easy to test. Is mainly used as a state management tool with React, but it can be used with any other JavaScript framework or library.

- **React-resizable:** Is a widget that React-resizable is a widget that helps you to wrap your existing components inside. It gives those components resize handles.

- **React-responsive:** React-responsive is an NPM package that allows you to create truly responsive designs in your React projects. It uses the combined power of media queries and breakpoints to define DOM elements the developer wants to show and hide.

- **React-social-login:** React Social Login is a higher-order component (HOC) which provides social login through multiple providers such as Google, Facebook, Amazon, GitHub, LinkedIn and Instagram.
- **React-social-login-buttons:** This package is used to display your social login buttons using React.
- **React-sortable-hoc:** React Sortable provides a simple set of higher-order components to fill those gaps and add sortable functionality to your lists.
- **Redux:** Redux helps you write applications that behave consistently, running in different environments (client, server, and native), and are easy to test, is basically a predictable state container for JavaScript apps.
- **Redux-thunk:** Redux Thunk middleware is used to write action creators that return a function instead of an action. One of their uses is to delay the dispatch of an action, or to dispatch only if a certain condition is met. The inner function receives the store methods dispatch and getState as parameters.
- **Sass:** Sass is a CSS pre-processor. Sass files are executed on the server and sends CSS to the browser.

Folder Structure:

3.5.3. Bootstrap: CSS framework

Bootstrap is a CSS Framework that help us to develop responsive and mobile-first front end websites providing CSS- and JavaScript- templates with different components such as typography, buttons, forms, and other design elements.

3.5.4. Ant Design

Ant Design is an open-source frontend design solution and React UI library that provides high-quality components and development resources for building interactive web applications. Is written in TypeScript and supports a wide range of languages, modern browsers, and Internet explorer 11.

3.6 Content Delivery Network

A content delivery network (CDN) is a group of servers placed in different locations that work together to deliver Internet content faster. The popularity of CND has increased significantly due to the speed it gives you when transferring content like videos, images, html pages, stylesheets, or JavaScript files. Companies that require big traffic of data are using it like Facebook, Amazon and Netflix. Regarding the security CND also gives the required protection against some common attacks. For this project CDN is used in order to optimize the access to the content in our website.

3.7 Monitoring

It is very useful to have in one place all your data come from servers and databases in that way you are aware in real time how your application is behaving. For that reason, for this project, we will be monitoring our application with a cloud monitoring platform call Datadog.

4. Requirements

This section will give a solid picture of the Functional Requirements which will define exactly what the application does, the Non-Functional Requirements specifies how the system should perform the tasks, the Data Requirements will help to know which information the system required, the User Requirement will mention the different components or entities that interact with the system and finally the Usability Requirements to specify how the application is easy to use.

4.1 Functional requirements

The functional requirements for Codeavecmoi are listed as bellow:

4.1.1. For Student, Teacher, Admin

1. Must be able to register an account.

2. Must be able to sign in his account when the account is activated.
3. Must be able to sign in with his Google account when the account is activated.
4. Must be able to sign in with Facebook account when the account is activated.
5. Must be able to sign out when he is authenticated.
6. Must be able to reset his password.
7. Must be able to change his password when he is authenticated.
8. Must be able to update his name and his email when he is authenticated.
9. Must be able to see all the available courses.
10. Must be able to see course details.

4.1.2. For Teacher

11. Should be able to create an account as a Teacher.
12. Must be able to set his payment details.
13. Should be able to create, publish and update his profile.
14. Must be able to see, create, update, and publish courses when he is authenticated.
15. Must be able to set a price for his courses.
16. Must be able to get access to all the comments related to his course when he is authenticated.
17. Must be able to reply to a comment when he is authenticated.
18. Must know how many students are enrolled in his course when he is authenticated.
19. Must know his income when he is authenticated.
20. Must be able to see all the transactions related to his courses.

21. Must be able to access a dashboard where he can see the summary of courses activity.
22. Must be able to get notified when someone leave a comment to his course.
23. Must be able to arrange easily the videos based on his preferences.
24. Must be able to preview his course.
25. Must be able to update, delete or re arrange the videos in his courses.

4.1.3. For Student

26. Must be able to watch and access free courses when is authenticated.
27. Must be able to pay a course when is authenticated.
28. Must be able to access his paid courses when is authenticated.
29. Must be able to cancel his account when he is authenticated.
30. Must be able to rate a course when he is authenticated.
31. Must be able to post questions in a course when he is authenticated.
32. Must be able to comment or answer questions in a thread of the course when in authenticated.
33. Must be able to see his progress in all the courses that is registered when is authenticated.
34. Must be able to see his completed courses when he is authenticated.
35. Must get notified when a new course is released.
36. Must get notified when someone reply to his comment.

4.1.4. For Admin

37. Must be able to access the admin dashboard when is authenticated.

38. Must be able to get access to all the courses content when is authenticated.
39. Must be able to get access to all the users info when is authenticated.
40. Must be able to see a notification for courses pending to approve when is authenticated.
41. Must be able to approve a course when is authenticated.
42. Must be able to see relevant performance indicators such as the bestselling courses, total courses, total sells, total students, new teachers, new students, and pending curses when is authenticated.
43. Must be able to see the transactions and payments when is authenticated.
44. Must be able to get notified, when is Authenticated, about the API heath, to corroborate if the application is up running properly.

4.2 Non-Functional requirements

4.2.1. Compatibility Requirement

The application developed must guarantee that the product is able to work under different environments, settings, and configurations. CodeAvecMoi should be responsive, which means be accessible on Desktop, Tablet and Mobile*.*

4.2.2. Performance Requirement

The performance testing is a broad topic. For this project, some keys aspects should be taken in consideration:

- Make sure that the response time for a request is not more than 4 seconds when multiple users try to access the application simultaneously.
- Verify that the response time to load the application is fair enough when we have a slow connection as the application is built for the African Market where sometimes the internet speed is slow.

- Check how many users the system can handle before the application crashes.
- Check database execution time when 1000 records are read/written simultaneously.
- Check CPU and memory usage of the application and the database server under peak load conditions.
- Verify the response time of the application under low, normal, moderate, and heavy load conditions.

4.2.3. Security Requirement

The platform must provide a handful tools or techniques to keep the websites secure from common web attacks like SQL Injection, Cross-Site Request Forgery (CSRF), Password attack, Cross-site scripting (XSS) attack to increase the user's trust.

4.2.4. Reliability Requirement

Systems always fail due to an internal or external factor. Having that in mind the application should be able to evaluate the Mean Time Between Failures (MTBF) and the Mean Time to Repair (MTTR).

Mean Time Between Failures is nothing than the measurement of time between a system breakdown.

Mean Time to Repair itself is the average time it takes to fix a system.

4.2.5. Recovery Requirement

The system must have enough capability to recover quickly after an issue and become fully functional.

4.2.6. Maintainability Requirement

This requirement is for the system the ability to easily upgrade all the components, libraries and software involved in the project. In another word make thing very simple in the way that if someone was not involved in the implementation can in the future add other configurations and logic to the existing application.

4.2.7. Scalability Requirement

Codeavecmoi should be capable to deal and perform well under an increased scope.

4.2.8. Deployment Requirement

Check if the application is deployed successfully and work with any issues after the deployment.

4.2.9. Data requirements

Aware of the General Data Protection Regulation (GDPR) that mentions that any data collected should not be Racial or Ethnic Origin, Political Opinions, Religious or Philosophical Beliefs, Trade Union Membership, Genetic Data and Biometric, the system must collect only the username, the user email, the user password. Moreover, collecting the name, the email and the password, the system, which is dealing with credit and debit card, must use a service that is PCI Compliance.

4.2.10. User requirements

CodeAvecMoi has three different types of users:

- **Student:** any person desire who wants to learn something on the platform.
- **Teacher:** any person that is an expert in a field and want to provide courses in the platform.
- **Admin:** any person that supervise different activities on the platform including the maintenance, ensuring that the platform is working correctly.

4.2.11. Usability requirements

The application must be user-friendly, for that reason it must use the latest libraries and software available on the market to provide a simple and intuitive user interface.

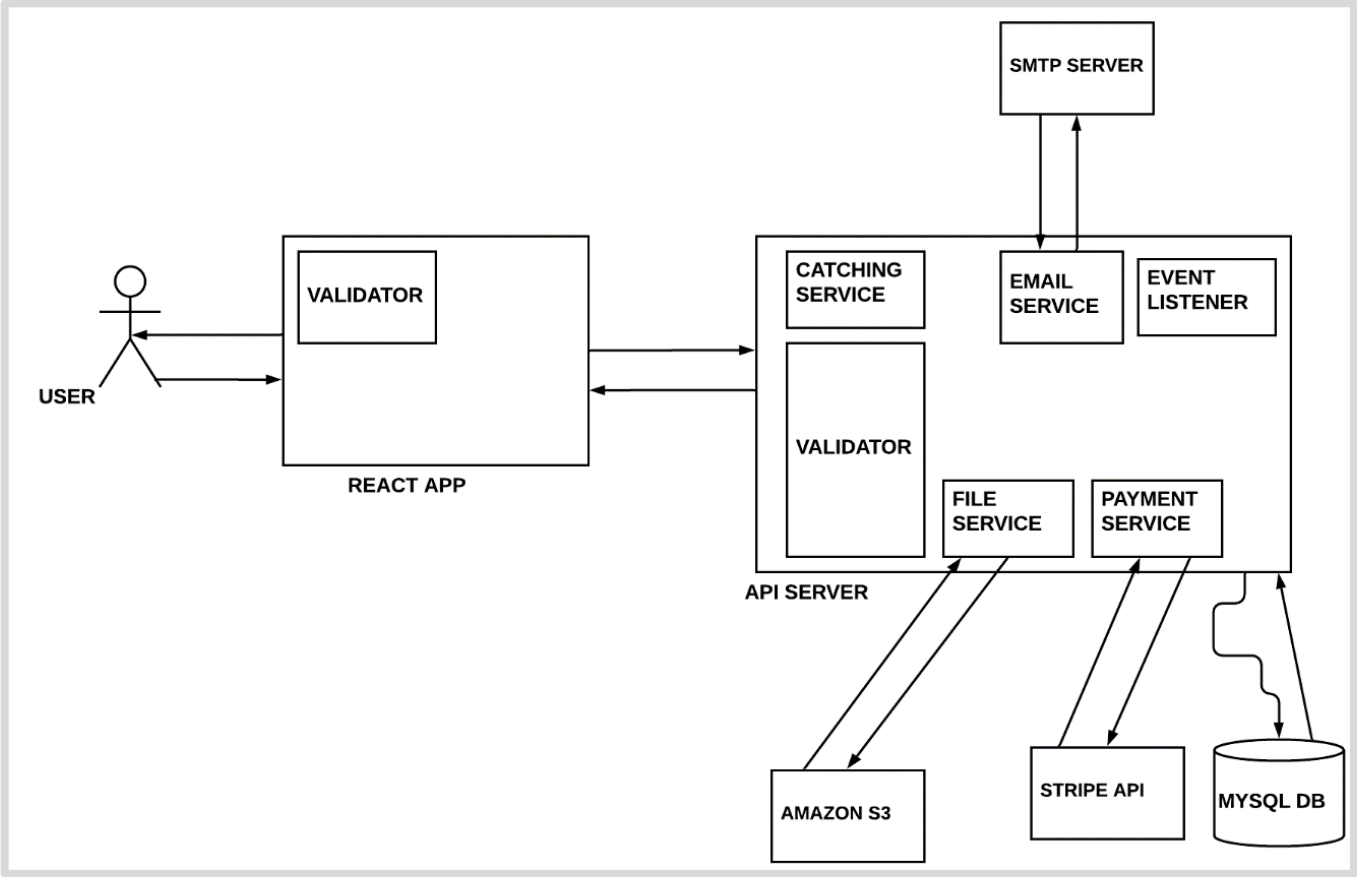
5. System

5.1 Design and Architecture

The below section will describe the architecture and relations between all the components or services of CodeAvecMoi.

5.1.1. Architecture

The following figure represents the architecture of the application.



5.1.2. Architecture description

The user interacts with the frontend application, the React application clean, sanitize and validate the data if the user filled up some forms, then proceed the request. If the front-end application realize that he needs some information from the API it will send an Http request to retrieve that piece of information. The backend service receives the request, make sure that the incoming request is safe. After the validation stage the API will proceed the request and send back the response to the frontend application where it will display the response to the user.

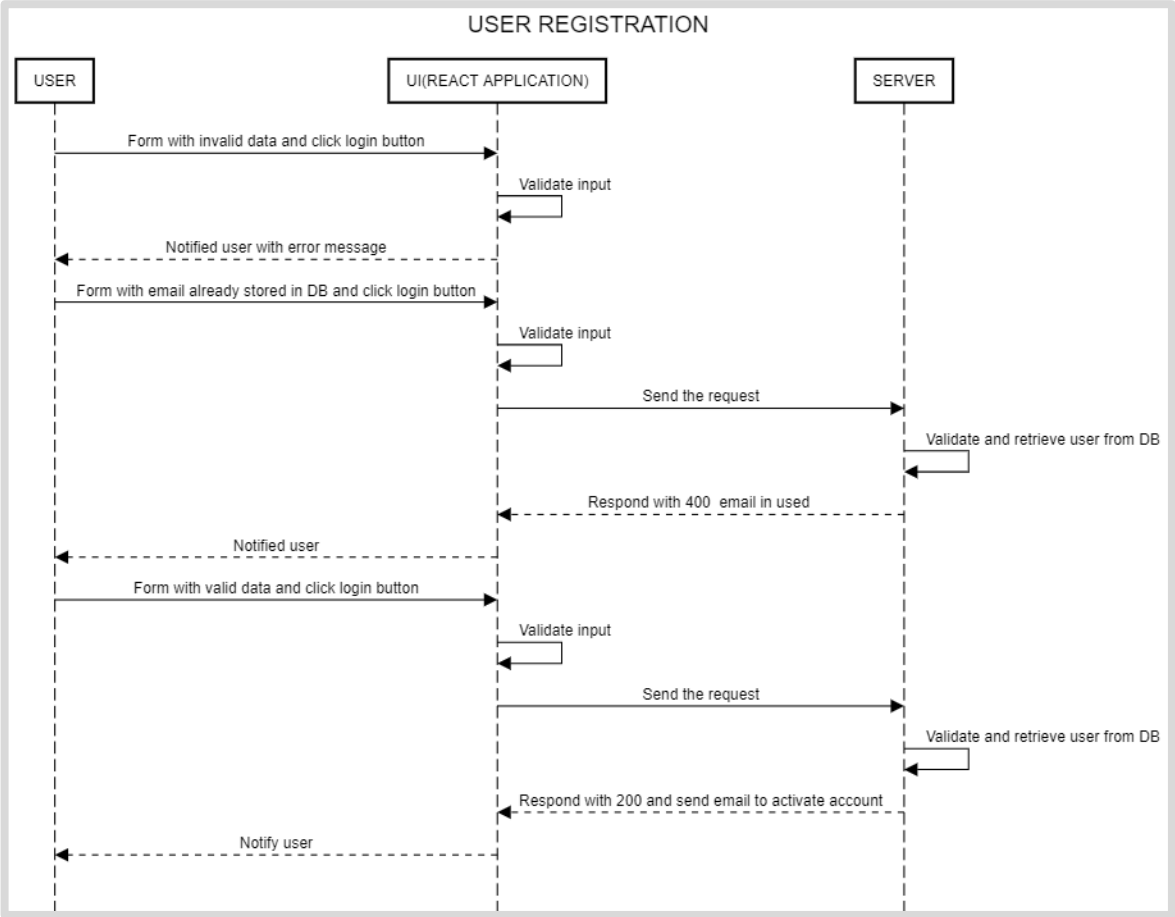
- **Catching Service:** this service use Redis to store some information in memory instead of query the database every time.

- **Validator:** this service act as interceptor, his rule is to sanitize, clean and validate all the incoming request.
- **Email Service:** this service is used to send emails to the users.
- **File Service:** this service is used to upload or download images and videos to Amazon S3.
- **Event Service:** the rule of this service is to notify the backend server if some events occurred. These events can be for instance a new user registration, an email that should be sent, etc.
- **Payment Service:** the payment service function is to manage payment and subscriptions. It will interact with the Stripe API.

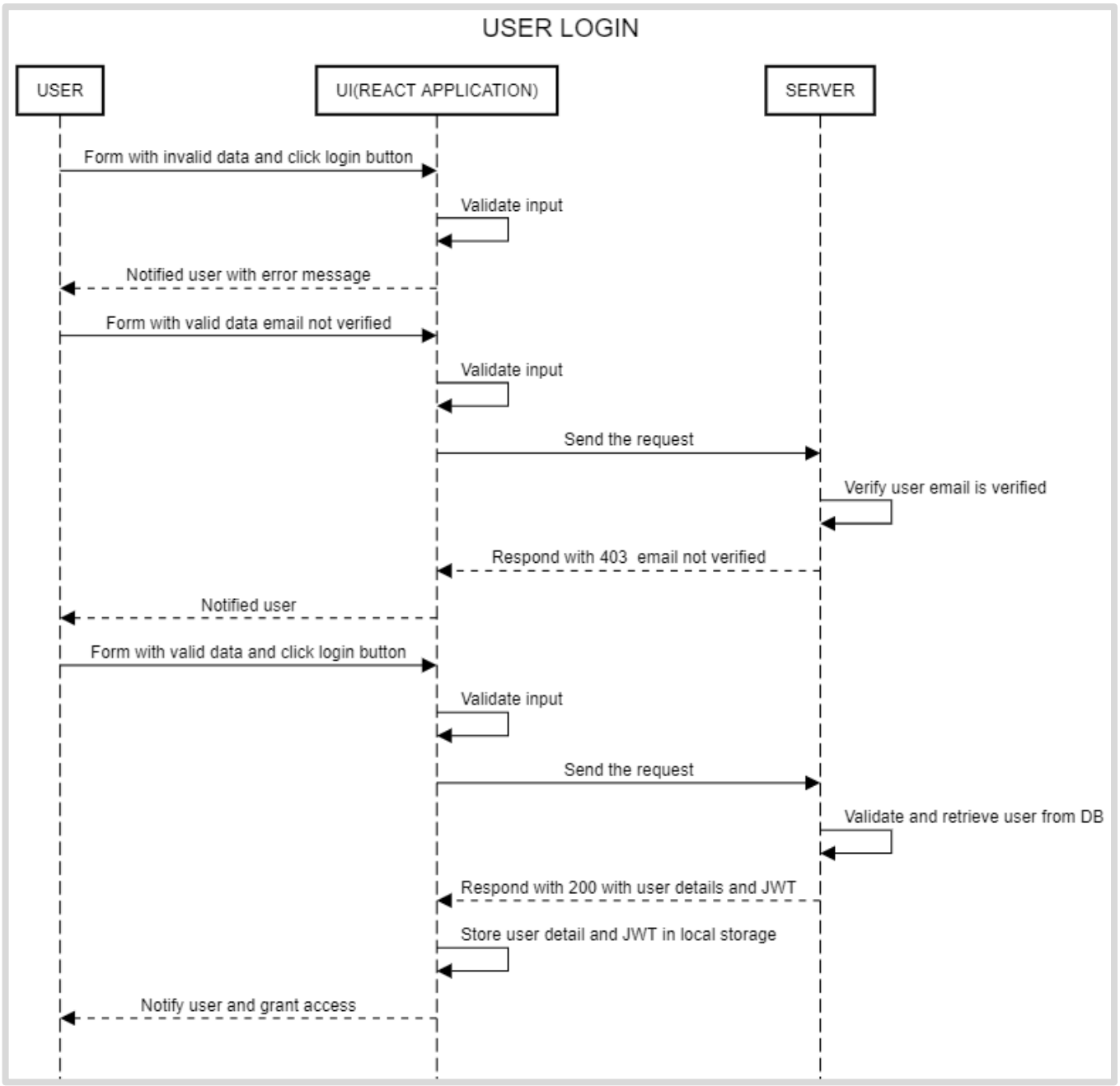
5.1.3. Sequence Diagram

The purpose of this section is to list all the sequence diagram for this application.

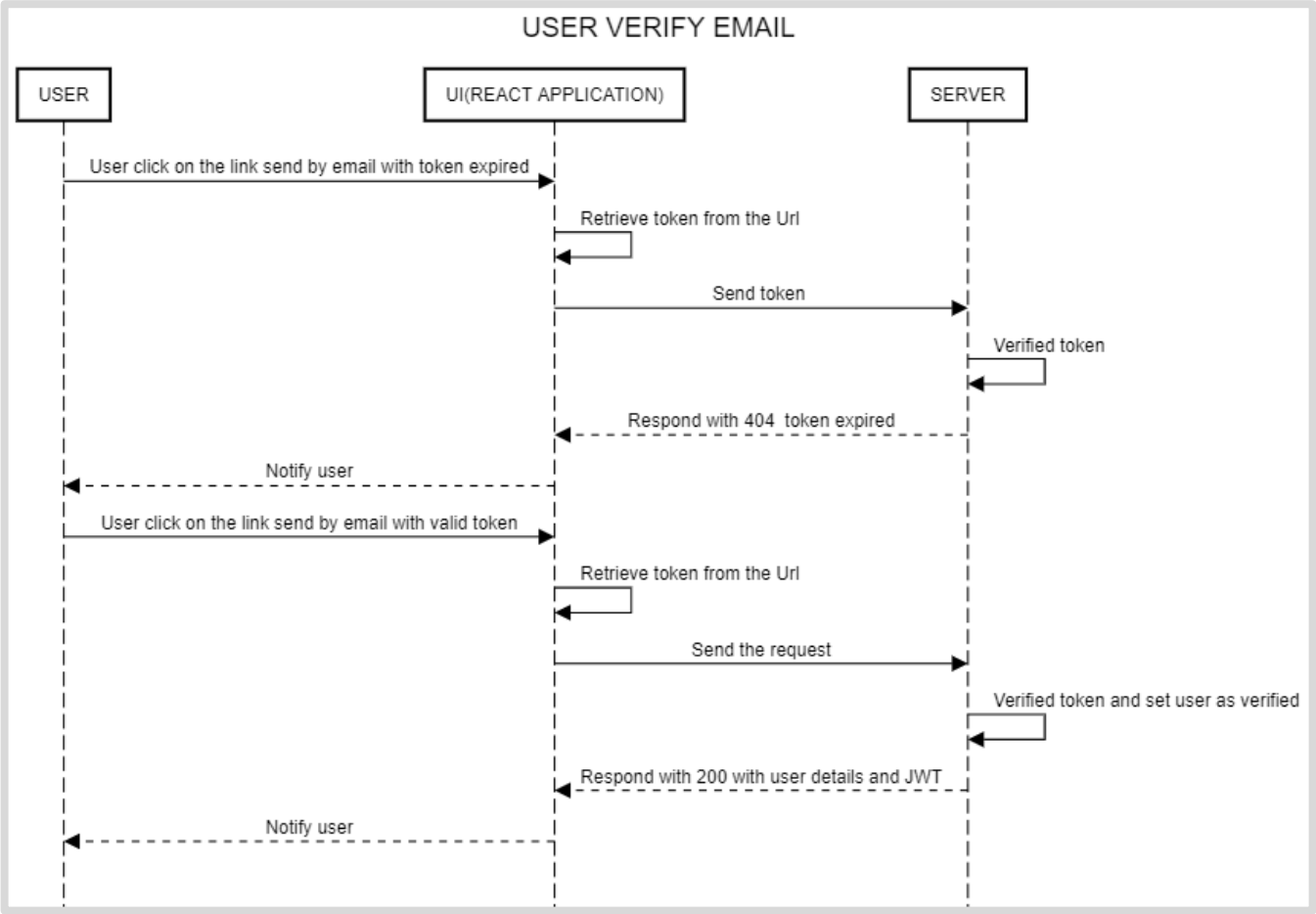
5.1.3.1. User registration sequence diagram



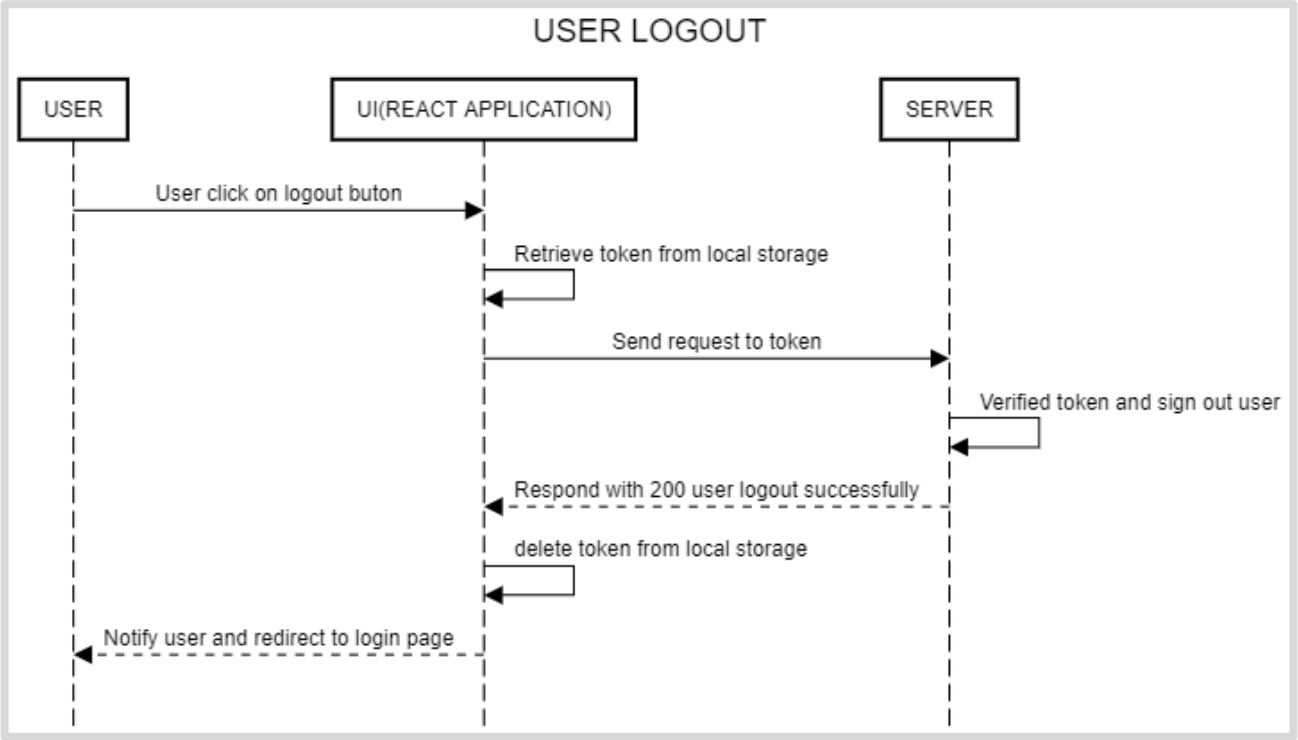
5.1.3.2. User login sequence diagram



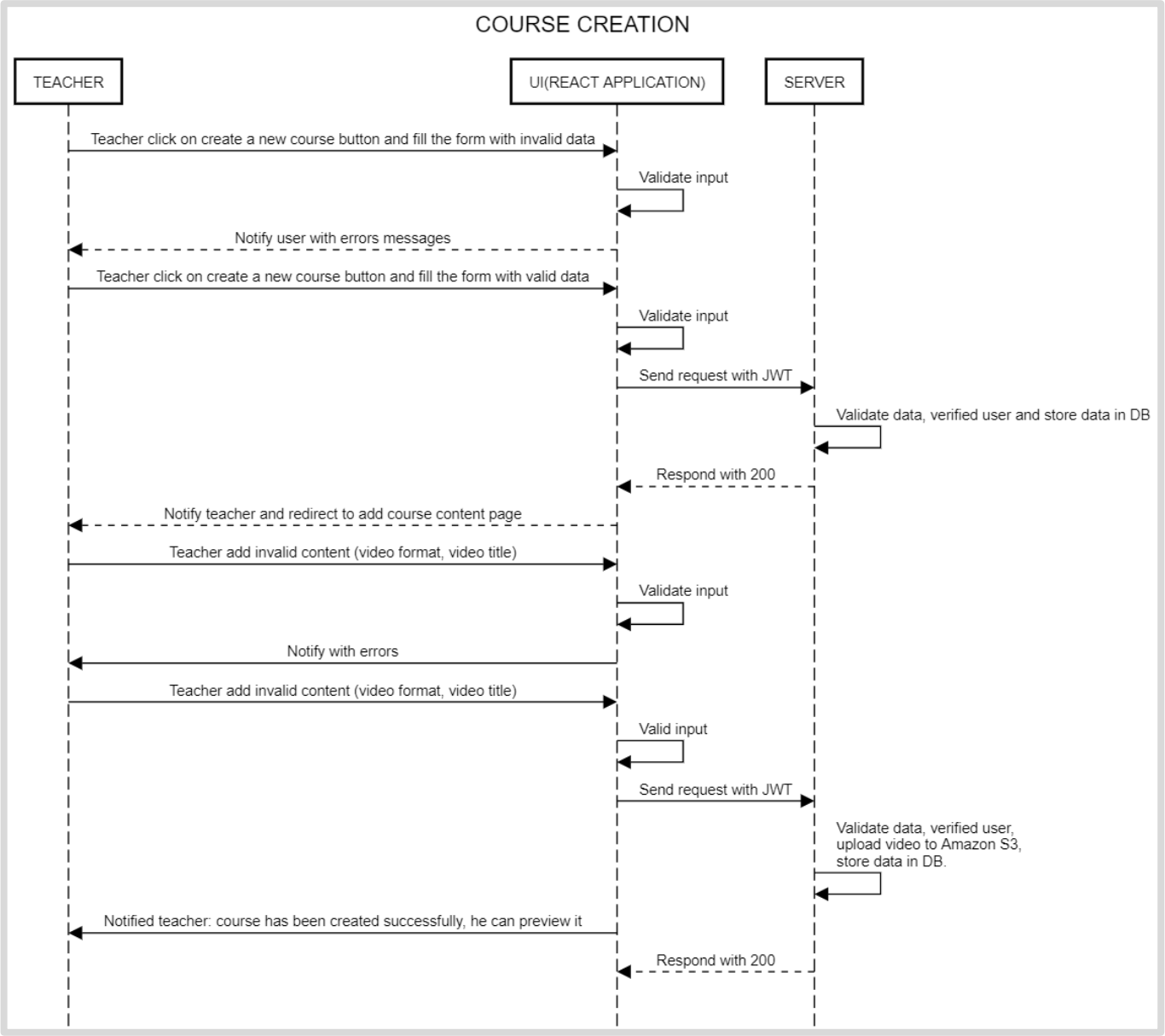
5.1.3.3. Verify user email diagram



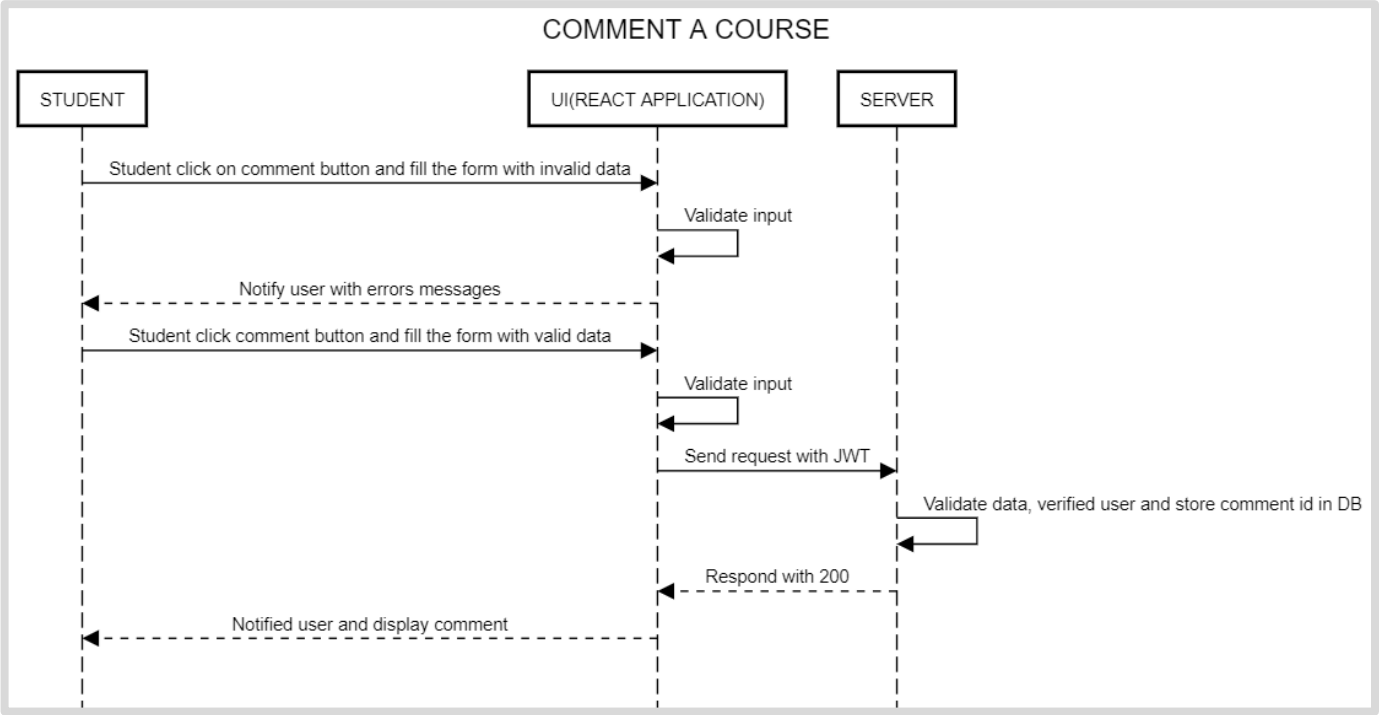
5.1.3.4. User logout diagram



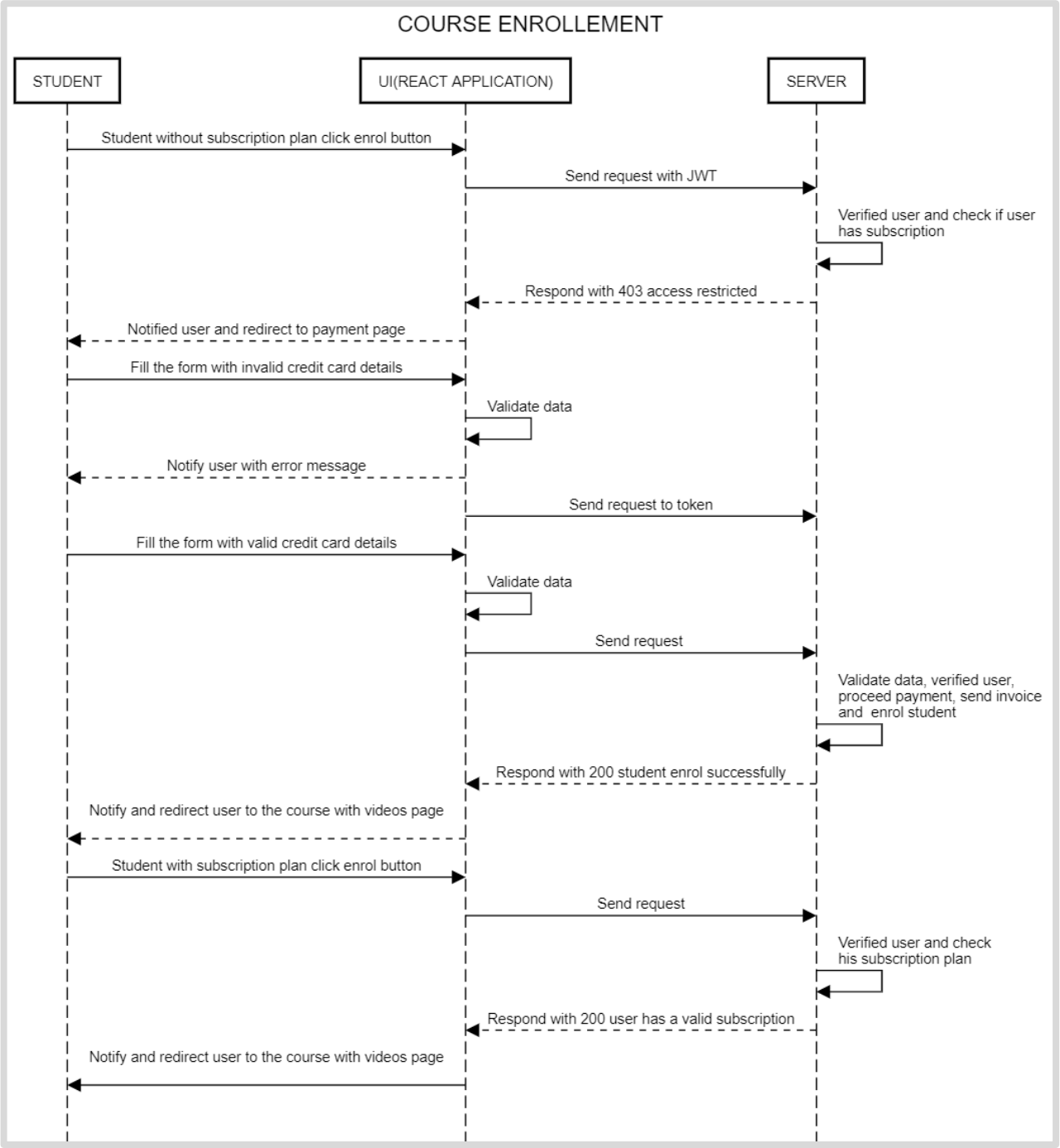
5.1.3.5. Course creation diagram



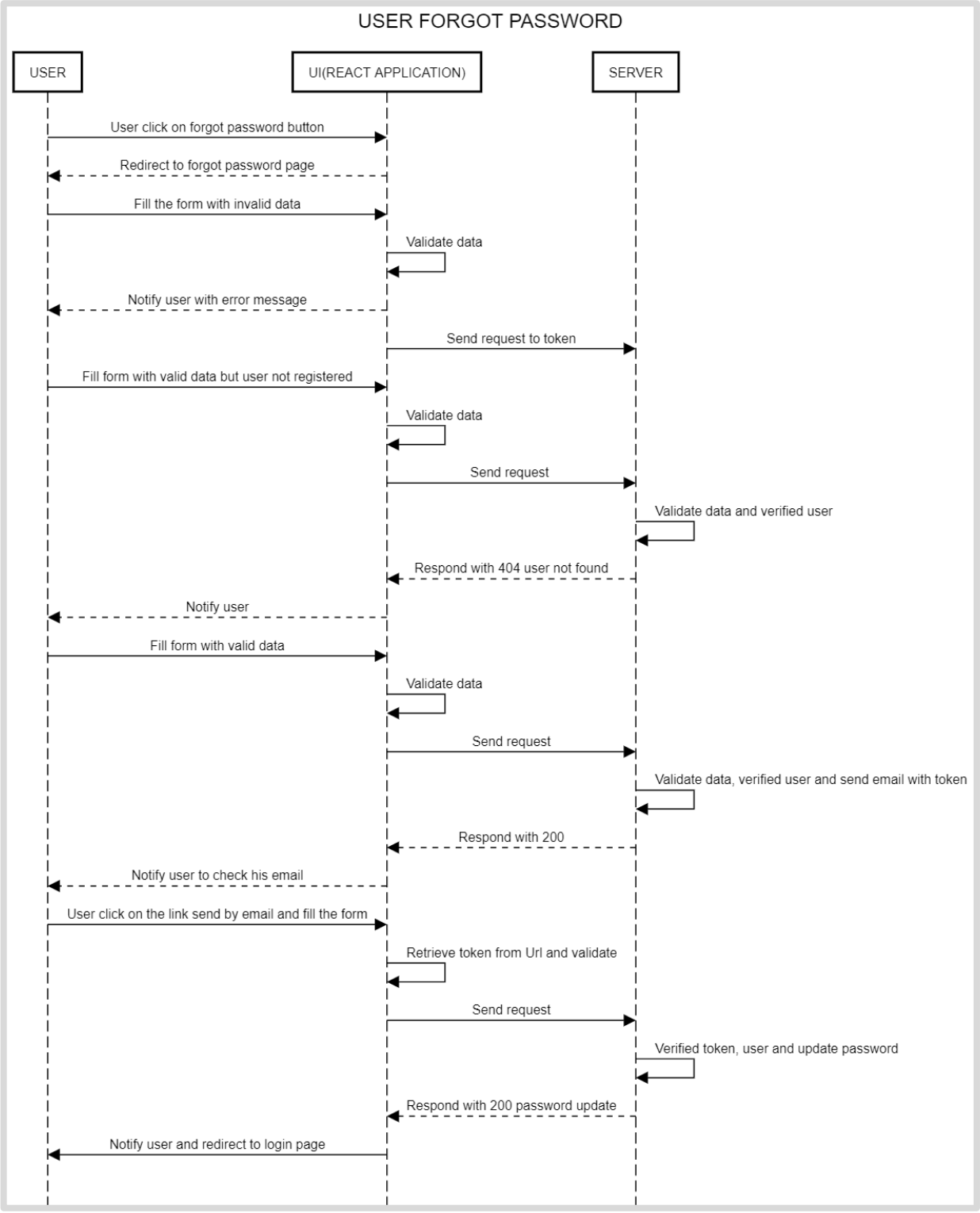
5.1.3.6. Course comment diagram



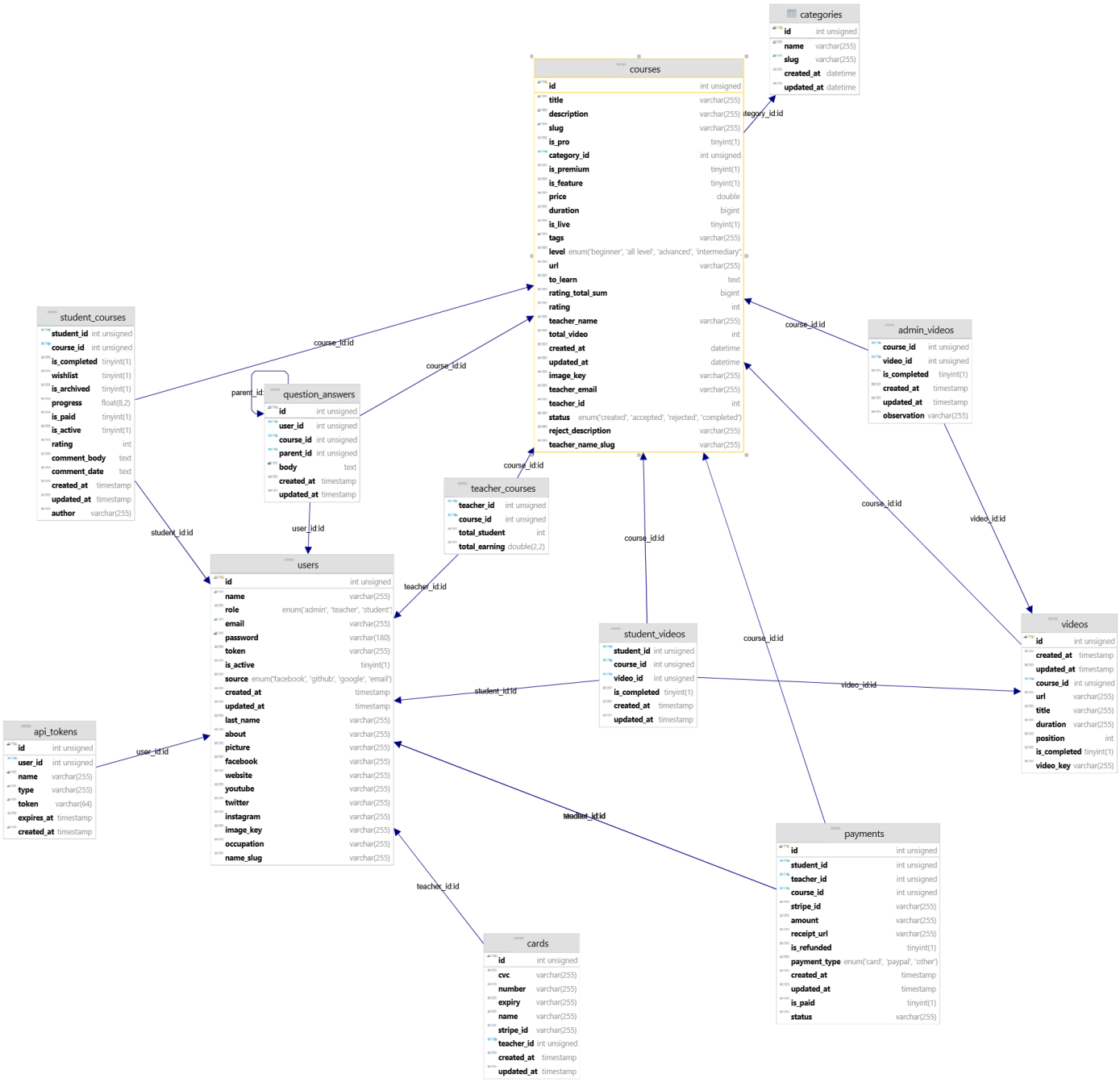
5.1.3.7. Course enrollment diagram

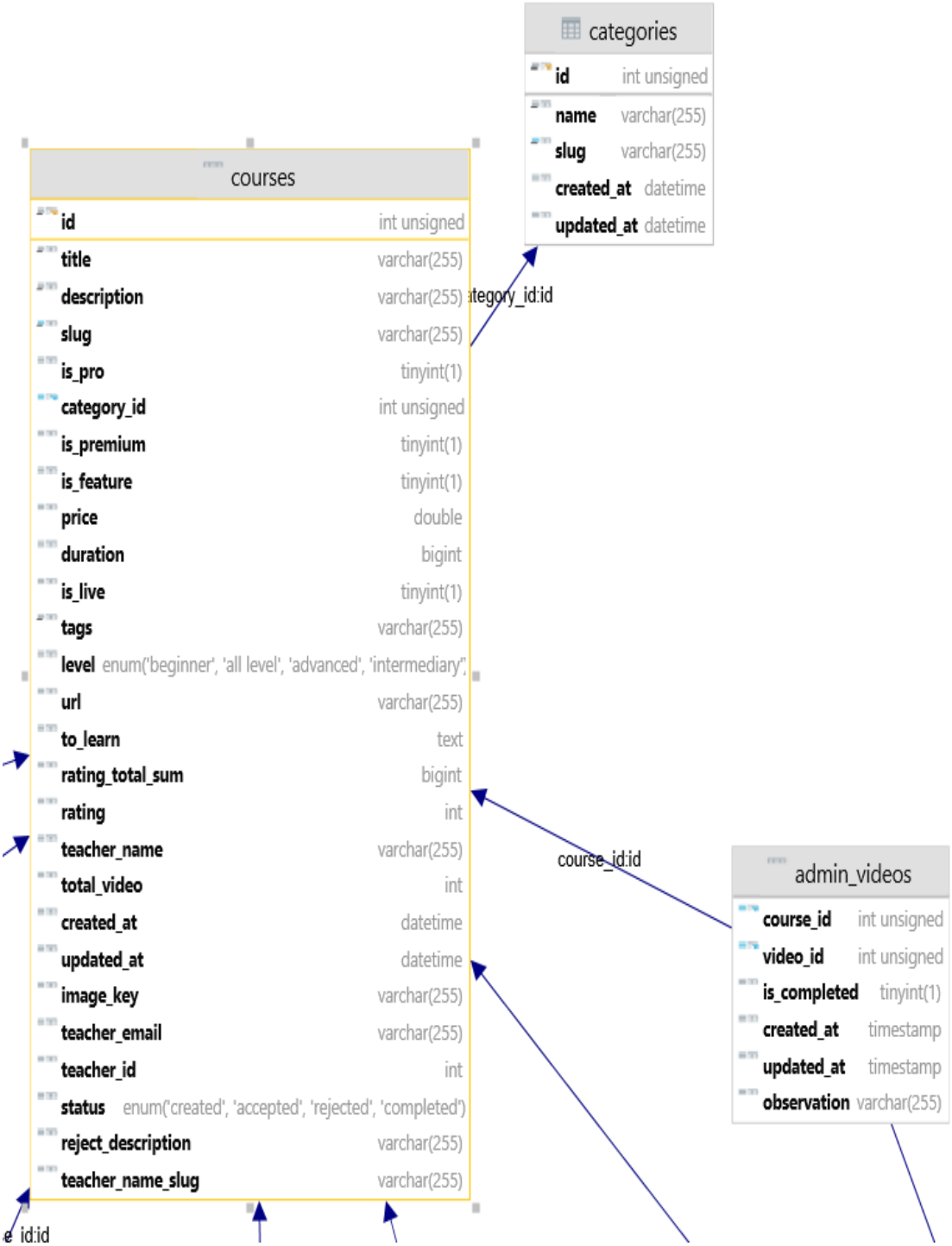


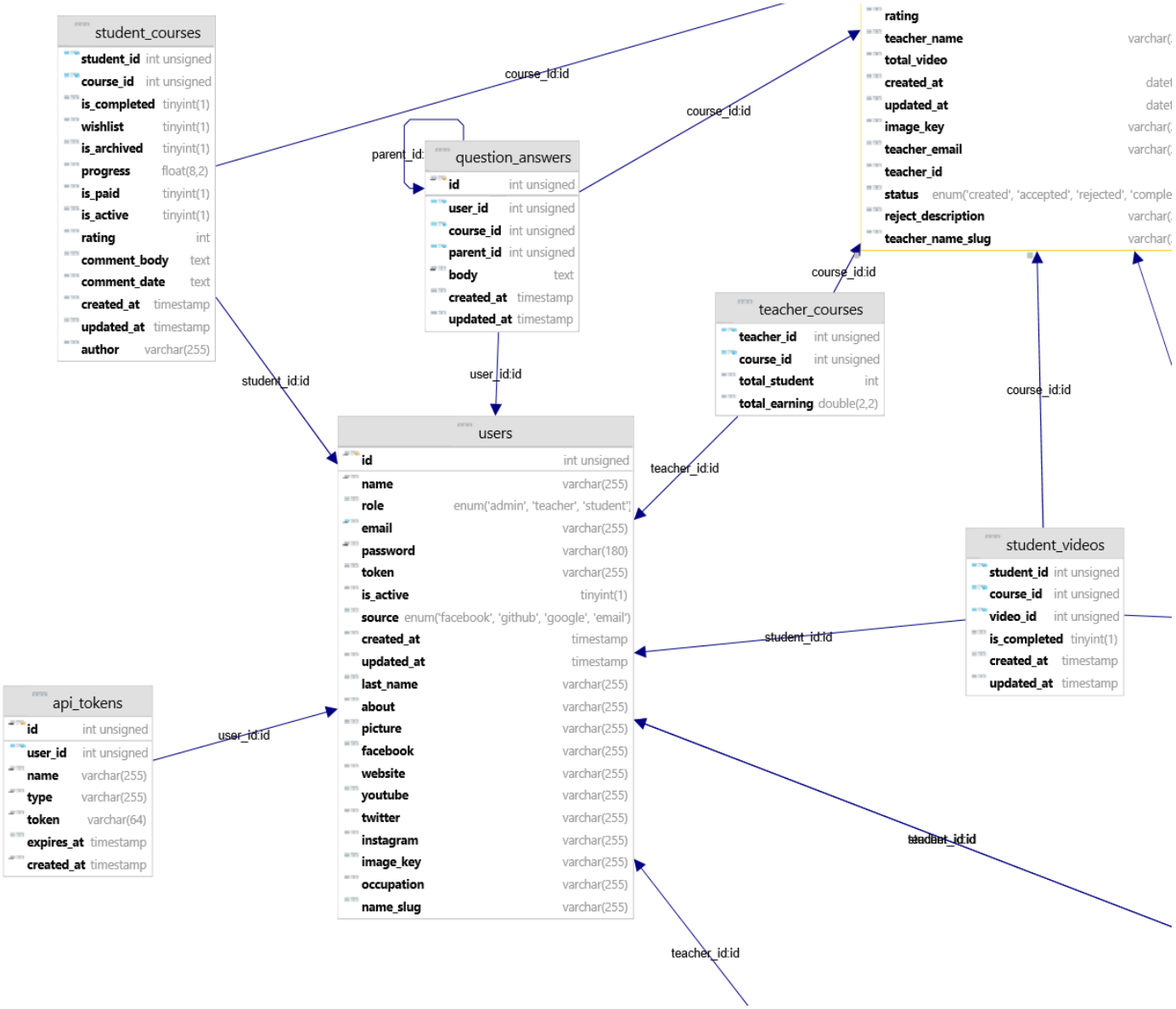
5.1.3.8. User forgot password diagram

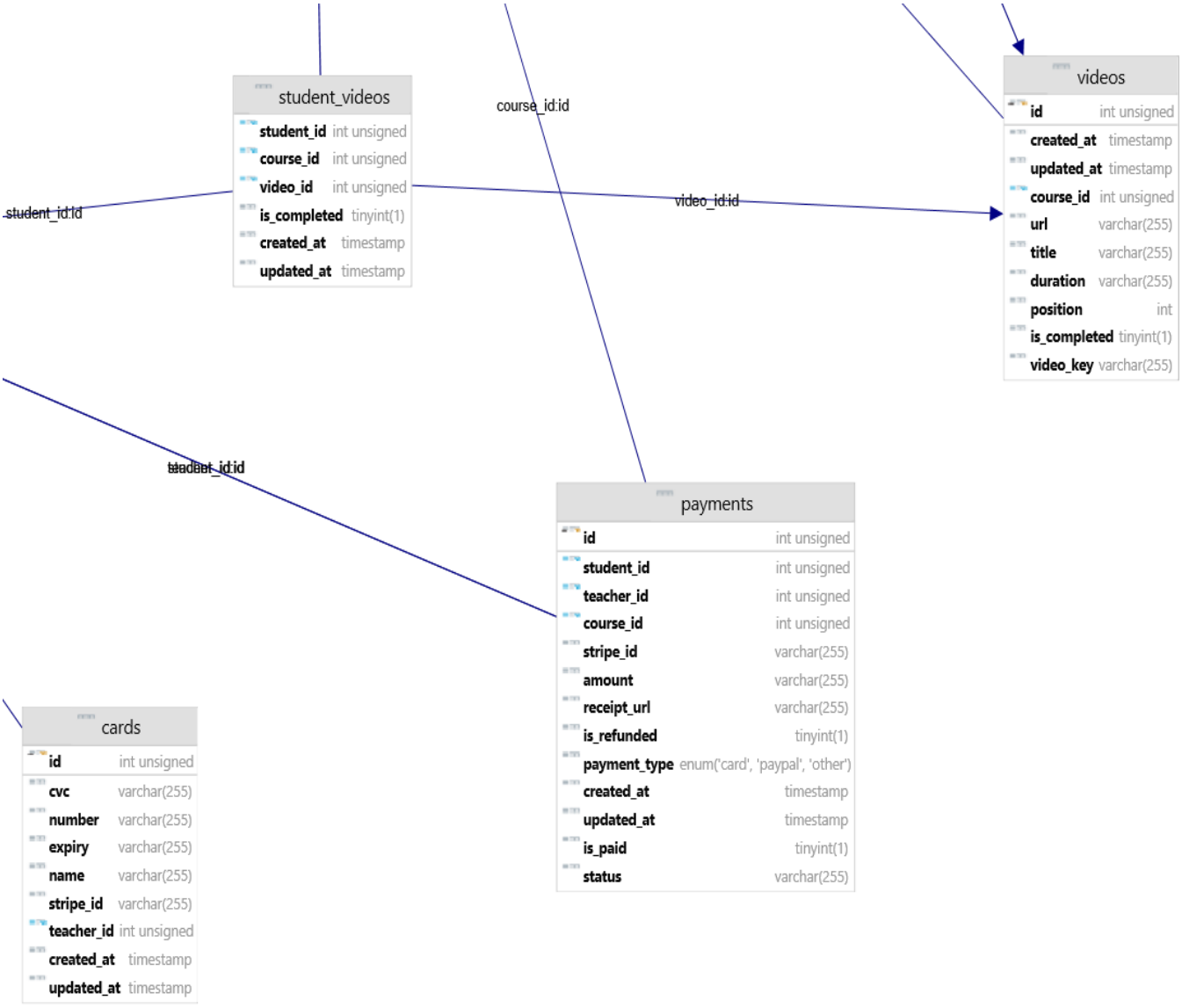


5.1.4. Database ER diagram




categories
id int unsigned
name varchar(255)
slug varchar(255)
created_at datetime
updated_at datetime
courses
id int unsigned
title varchar(255)
description varchar(255)
slug varchar(255)
is_pro tinyint(1)
category_id int unsigned
is_premium tinyint(1)
is_feature tinyint(1)
price double
duration bigint
is_live tinyint(1)
tags varchar(255)
level enum('beginner', 'all level', 'advanced', 'intermediary',
url varchar(255)
to_learn text
rating_total_sum bigint
rating int
teacher_name varchar(255)
total_video int
created_at datetime
updated_at datetime
image_key varchar(255)
teacher_email varchar(255)
teacher_id int
status enum('created', 'accepted', 'rejected', 'completed')
reject_description varchar(255)
teacher_name_slug varchar(255)
tegory_id:id
course_id:id
admin_videos
course_id int unsigned
video_id int unsigned
is_completed tinyint(1)
created_at timestamp
updated_at timestamp
observation varchar(255)
e id:id


student_courses
student_id int unsigned
course_id int unsigned
is_completed tinyint(1)
wishlist tinyint(1)
is_archived tinyint(1)
progress float(8,2)
is_paid tinyint(1)
is_active tinyint(1)
rating int
comment_body text
comment_date text
created_at timestamp
updated_at timestamp
author varchar(255)
question_answers
id int unsigned
user_id int unsigned
course_id int unsigned
parent_id int unsigned
body text
created_at timestamp
updated_at timestamp
rating
teacher_name varchar(
total_video
created_at date
updated_at date
image_key varchar(
teacher_email varchar(
teacher_id
status enum('created', 'accepted', 'rejected', 'comple
reject_description varchar(
teacher_name_slug varchar(
teacher_courses
teacher_id int unsigned
course_id int unsigned
total_student int
total_earning double(2,2)
users
id int unsigned
name varchar(255)
role enum('admin', 'teacher', 'student')
email varchar(255)
password varchar(180)
token varchar(255)
is_active tinyint(1)
source enum('facebook', 'github', 'google', 'email')
created_at timestamp
updated_at timestamp
last_name varchar(255)
about varchar(255)
picture varchar(255)
facebook varchar(255)
website varchar(255)
youtube varchar(255)
twitter varchar(255)
instagram varchar(255)
image_key varchar(255)
occupation varchar(255)
name_slug varchar(255)
student_videos
student_id int unsigned
course_id int unsigned
video_id int unsigned
is_completed tinyint(1)
created_at timestamp
updated_at timestamp
api_tokens
id int unsigned
user_id int unsigned
name varchar(255)
type varchar(255)
token varchar(64)
expires_at timestamp
created_at timestamp
course_id:id
course_id:id
parent_id:
course_id:id
user_id:id
student_id:id
teacher_id:id
course_id:id
student_id:id
user_id:id
teacher_id:id
teacher_id:id



5.2 User Interface and Usability

A good application design is more than aesthetics, therefore is important that beyond the looks, we consider the usability and functionality.

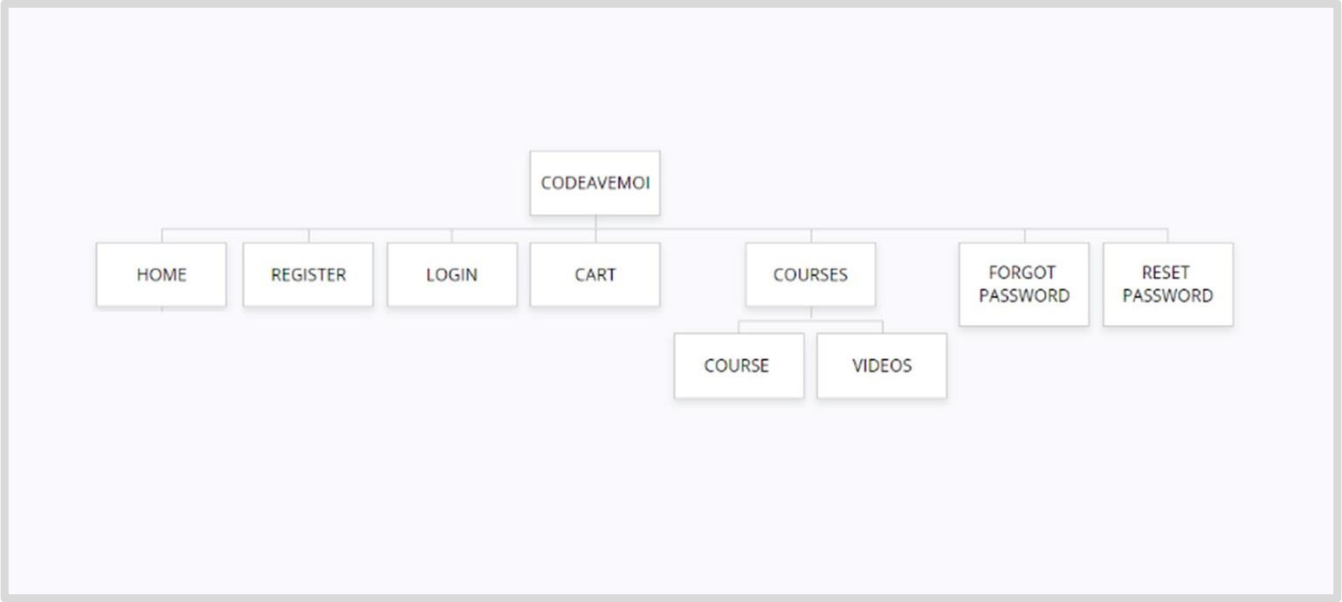
Usability is defined by the ISO 9241 standard as: "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use." Based on this definition, we can see that the concept of usability design goes beyond creating something easy to use, it involves effectiveness, efficiency, engagement, error tolerance and easy to learn.

Therefore, is imperative to adopt a user-centered design approach during all the design process, understanding the user needs and preferences when interacting with our product, and thus improve the user experience and deliver greater value.

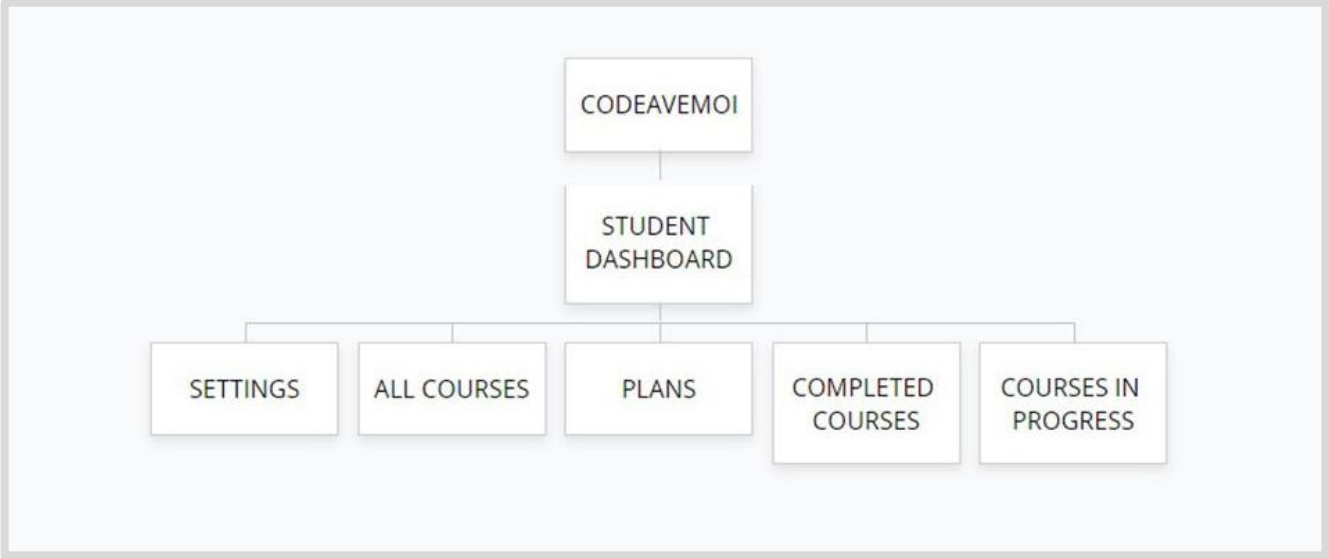
In this section I will present the sitemap which explain how the application is structured and the design process form the low fidelity and the wireframes.

5.2.1. UX Site map

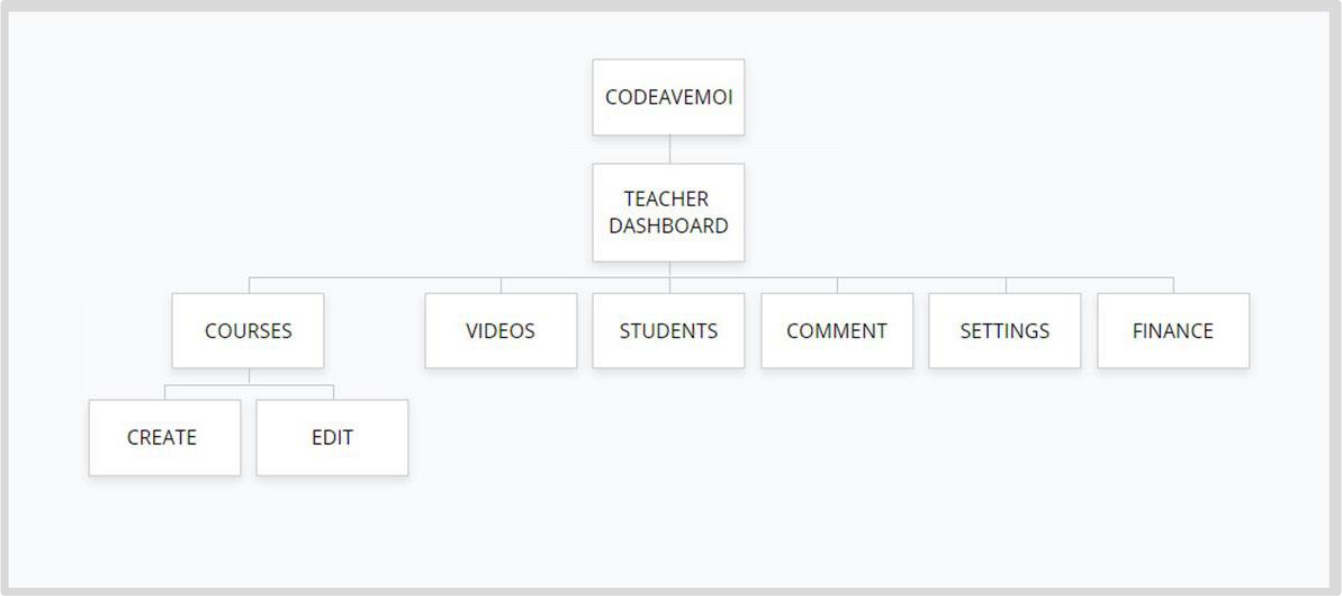
5.2.1.1. General view



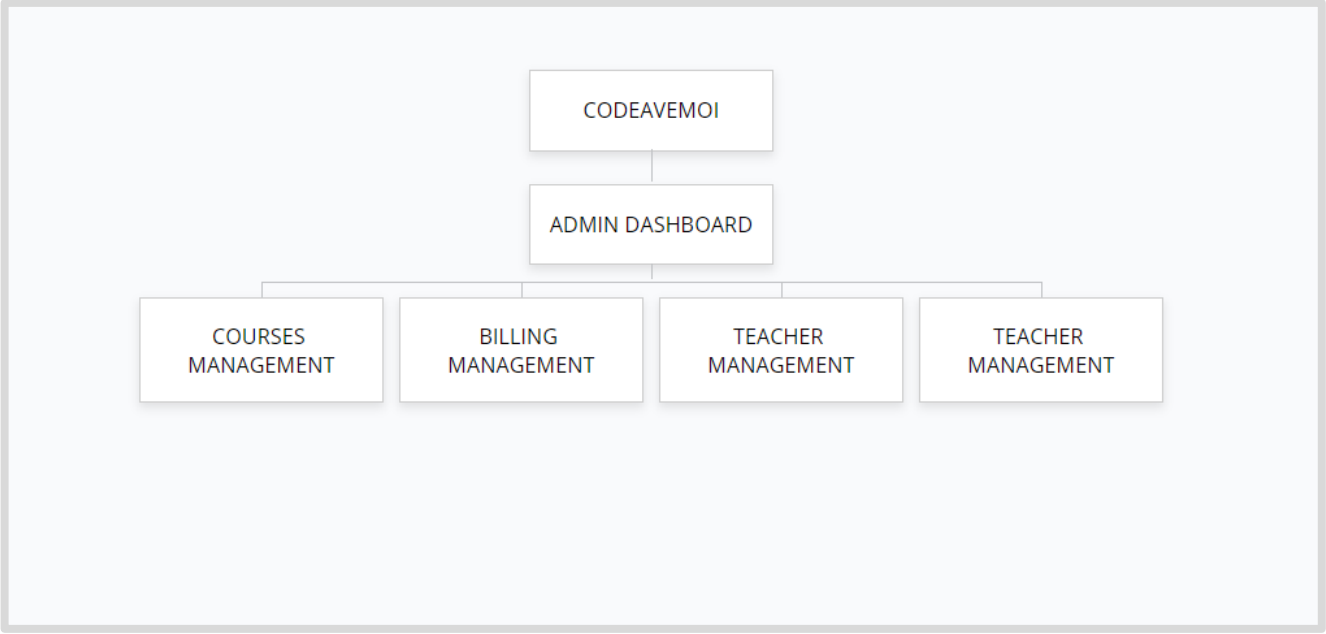
5.2.1.2. Student View



5.2.1.3. Teacher View

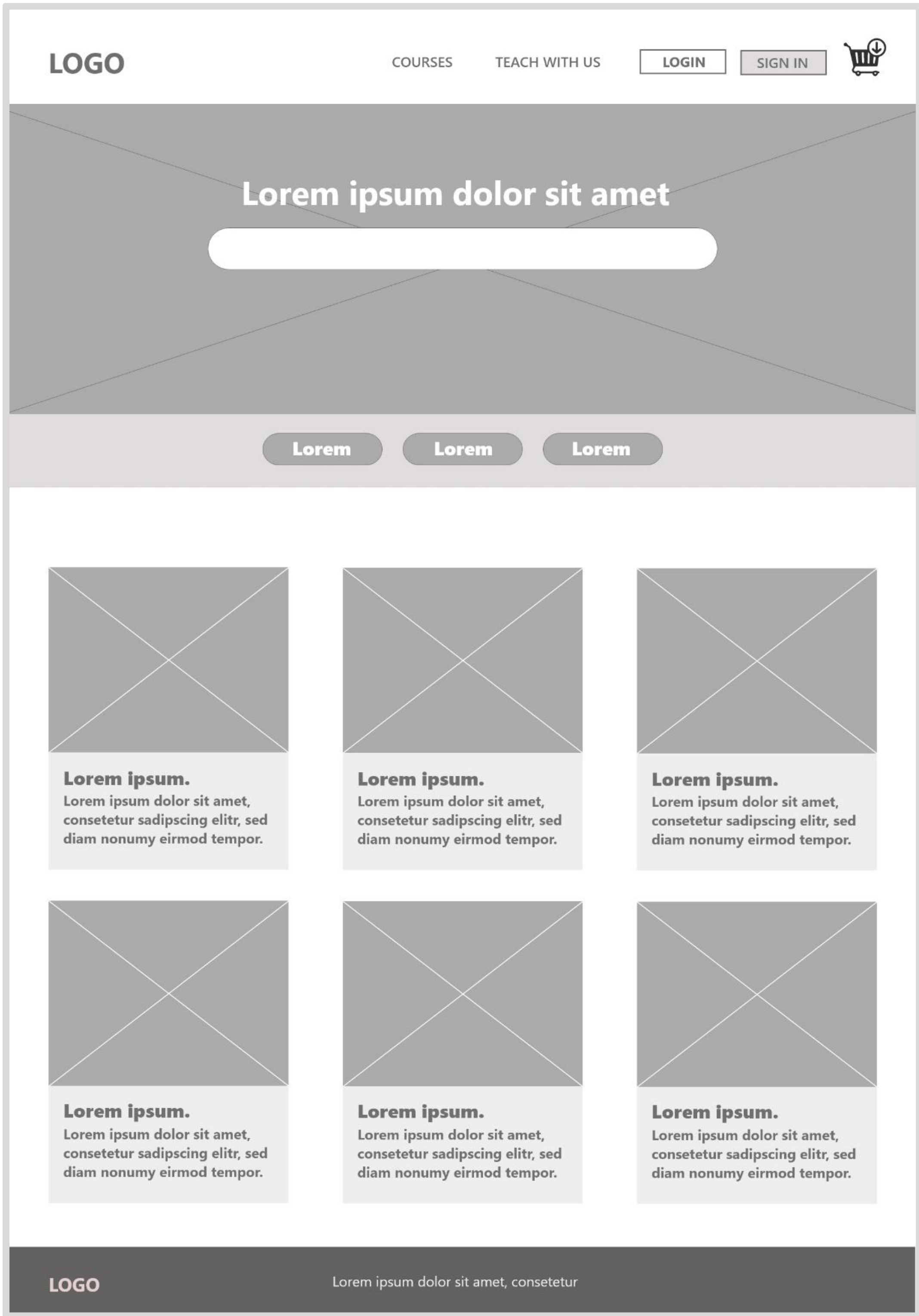


5.2.1.4. Admin View



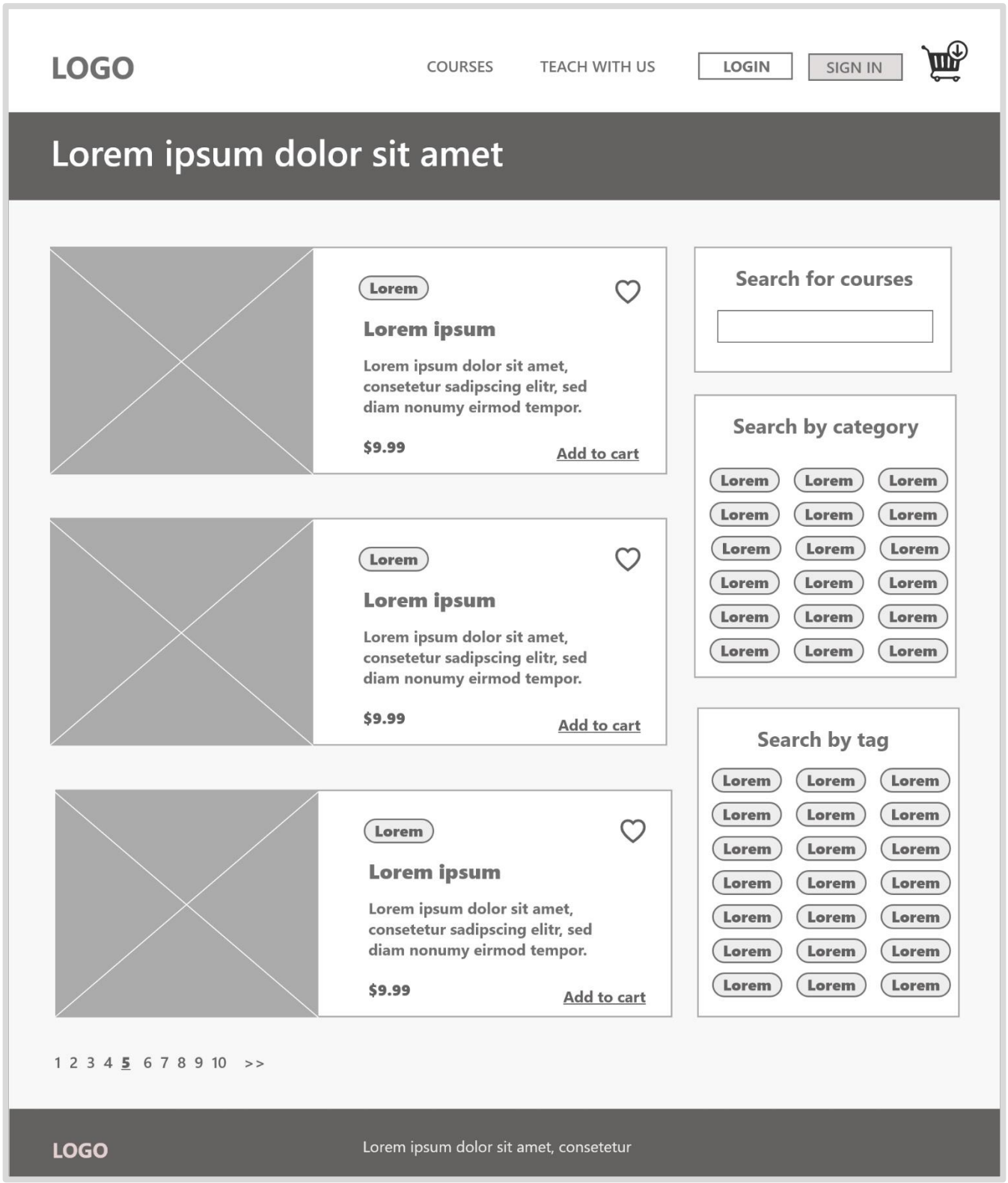
5.2.2. Low Fidelity Wireframes

5.2.2.1. Home Page

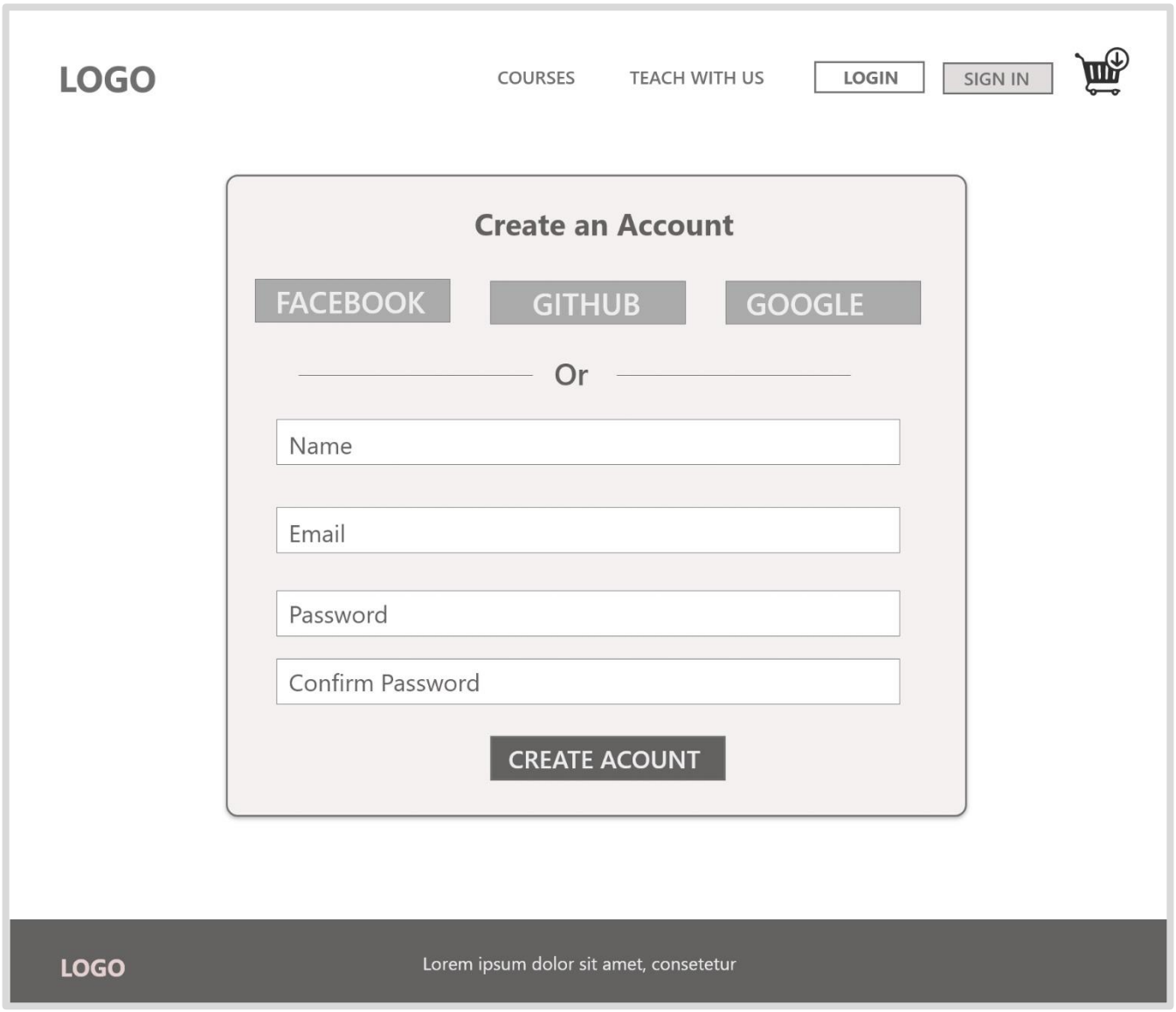


5.2.2.1. Teach with Us

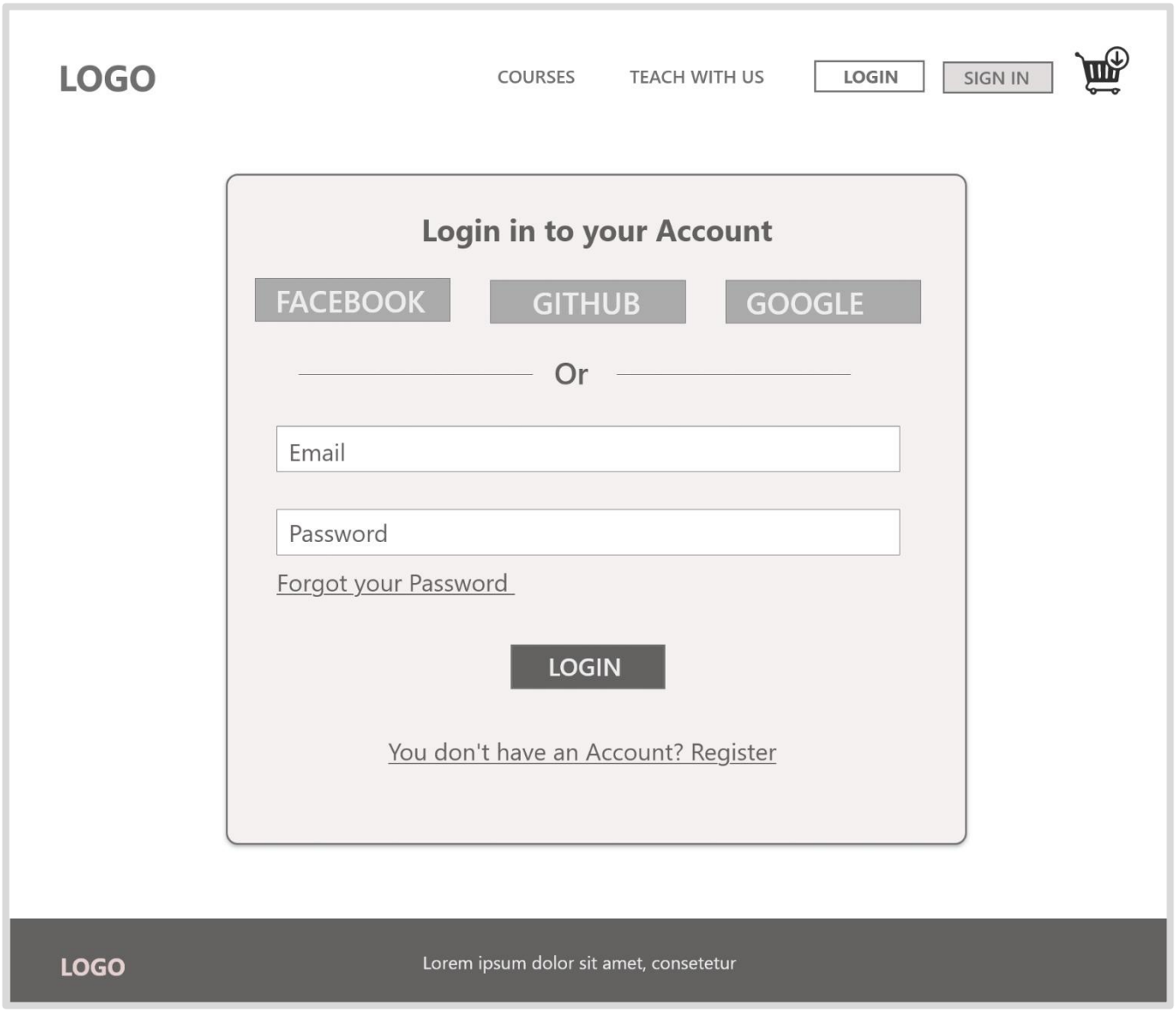
5.2.2.2. Courses



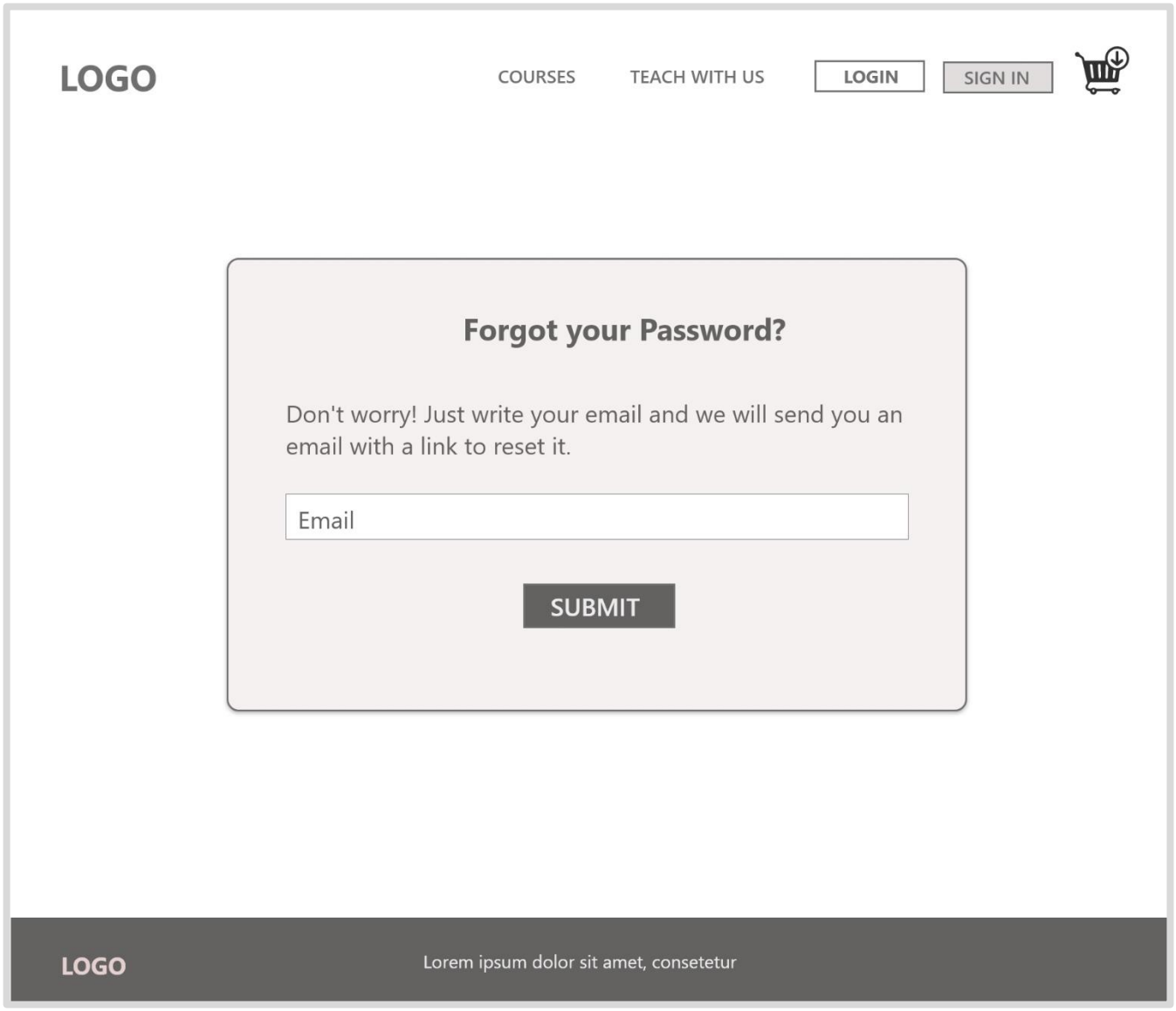
5.2.2.3. Create an Account



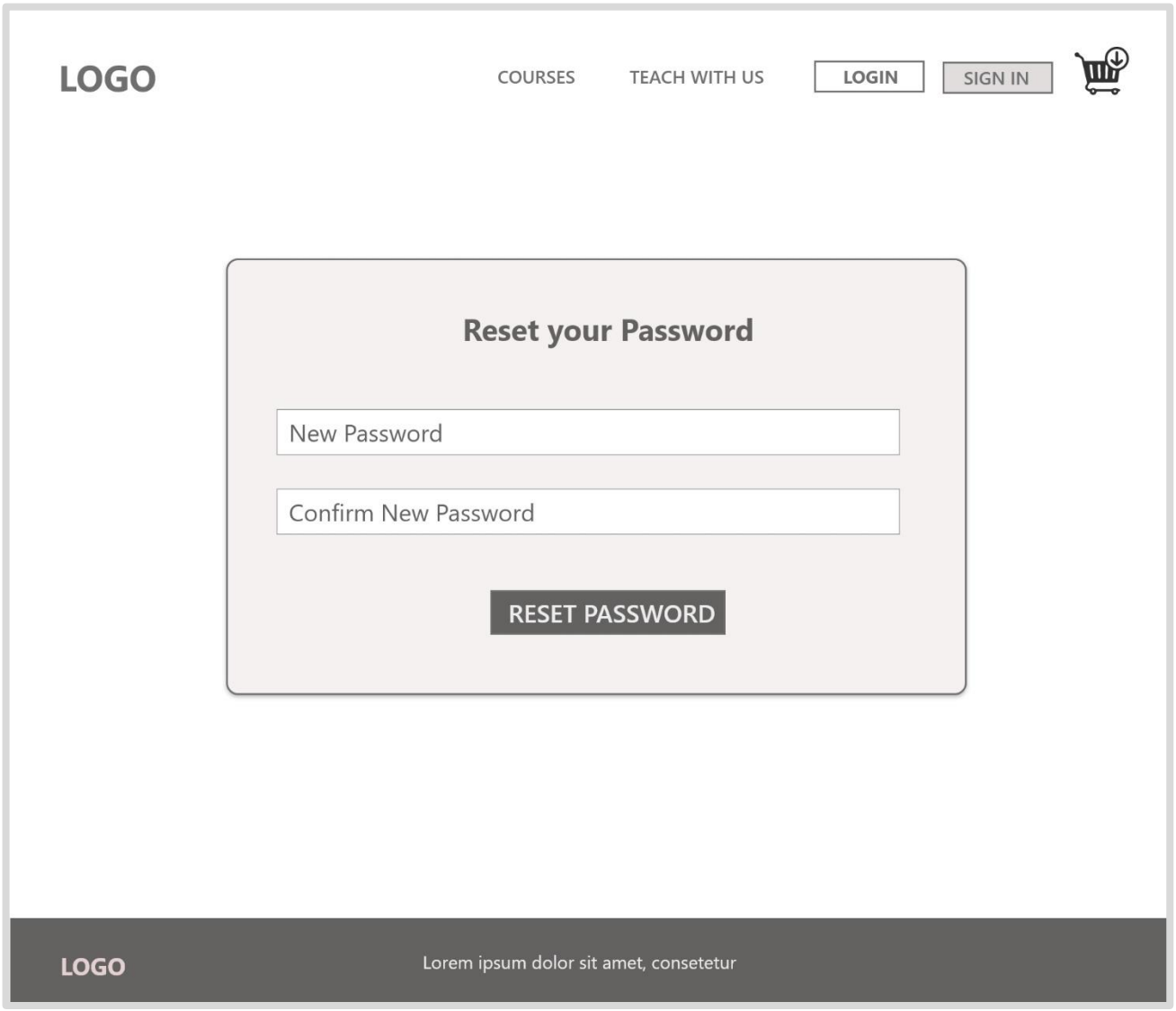
5.2.2.4. Login Your Account



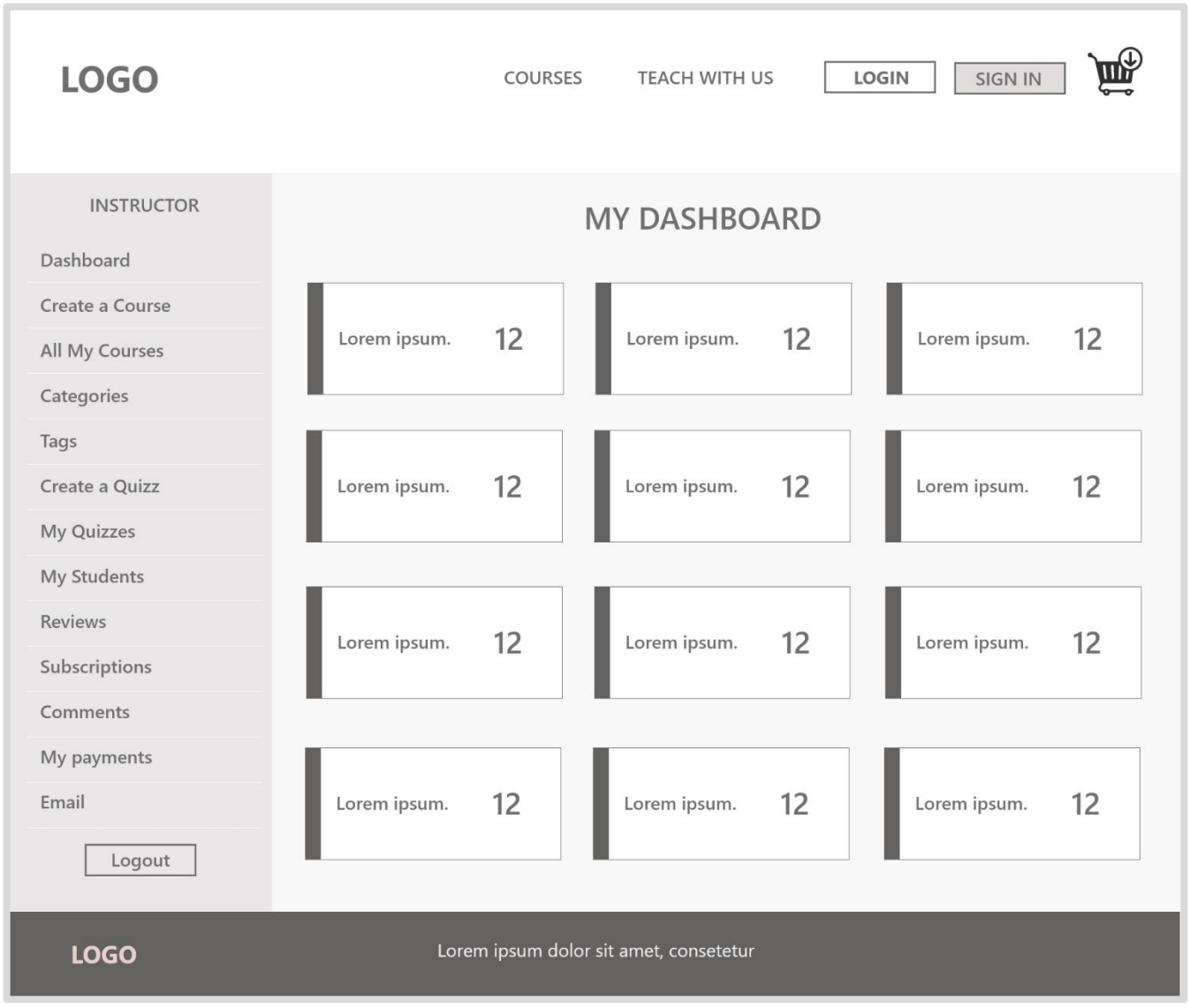
5.2.2.5. Forgot Password



5.2.2.6. Reset Password

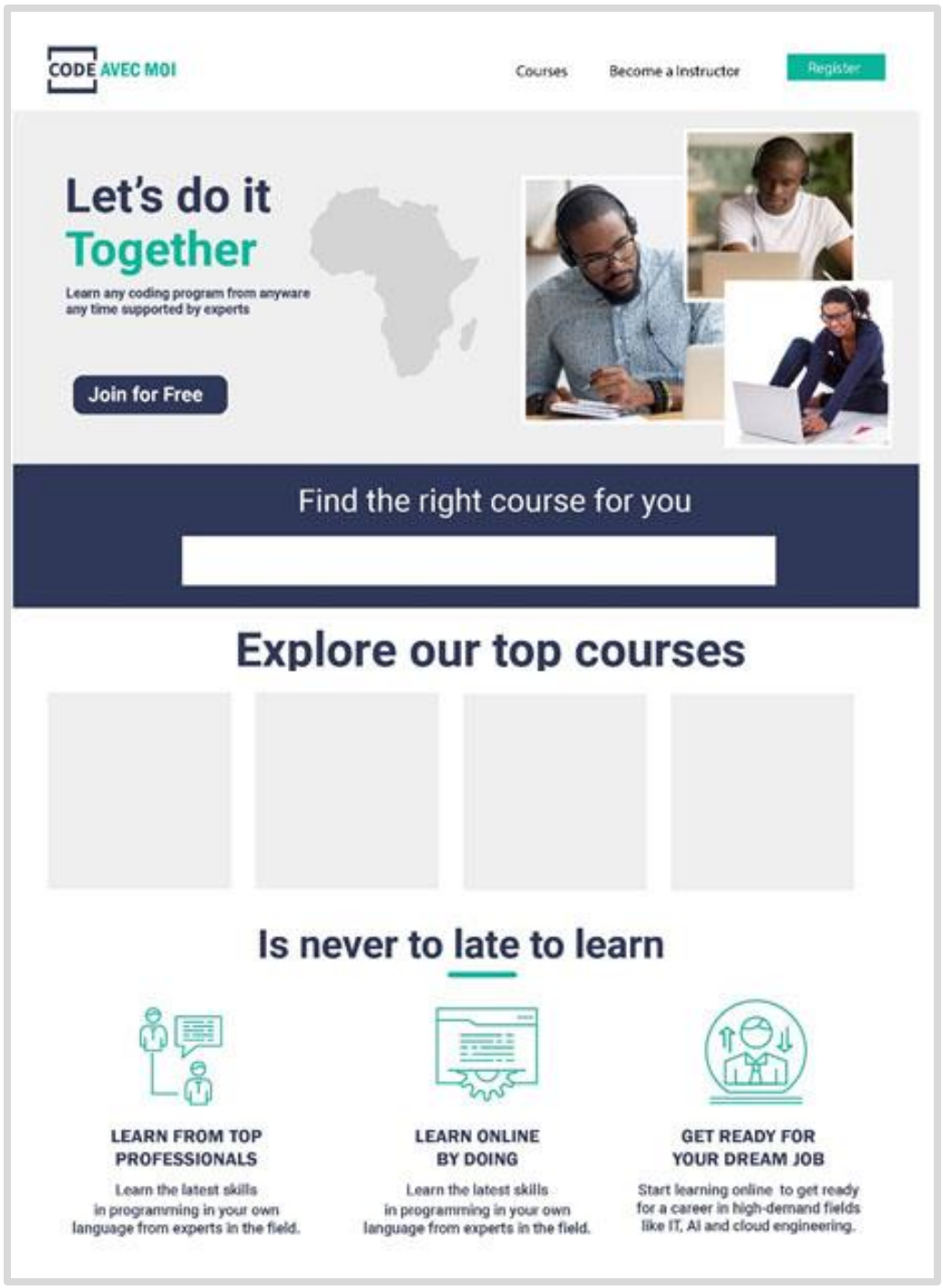


5.2.2.7. Teacher's Dashboard

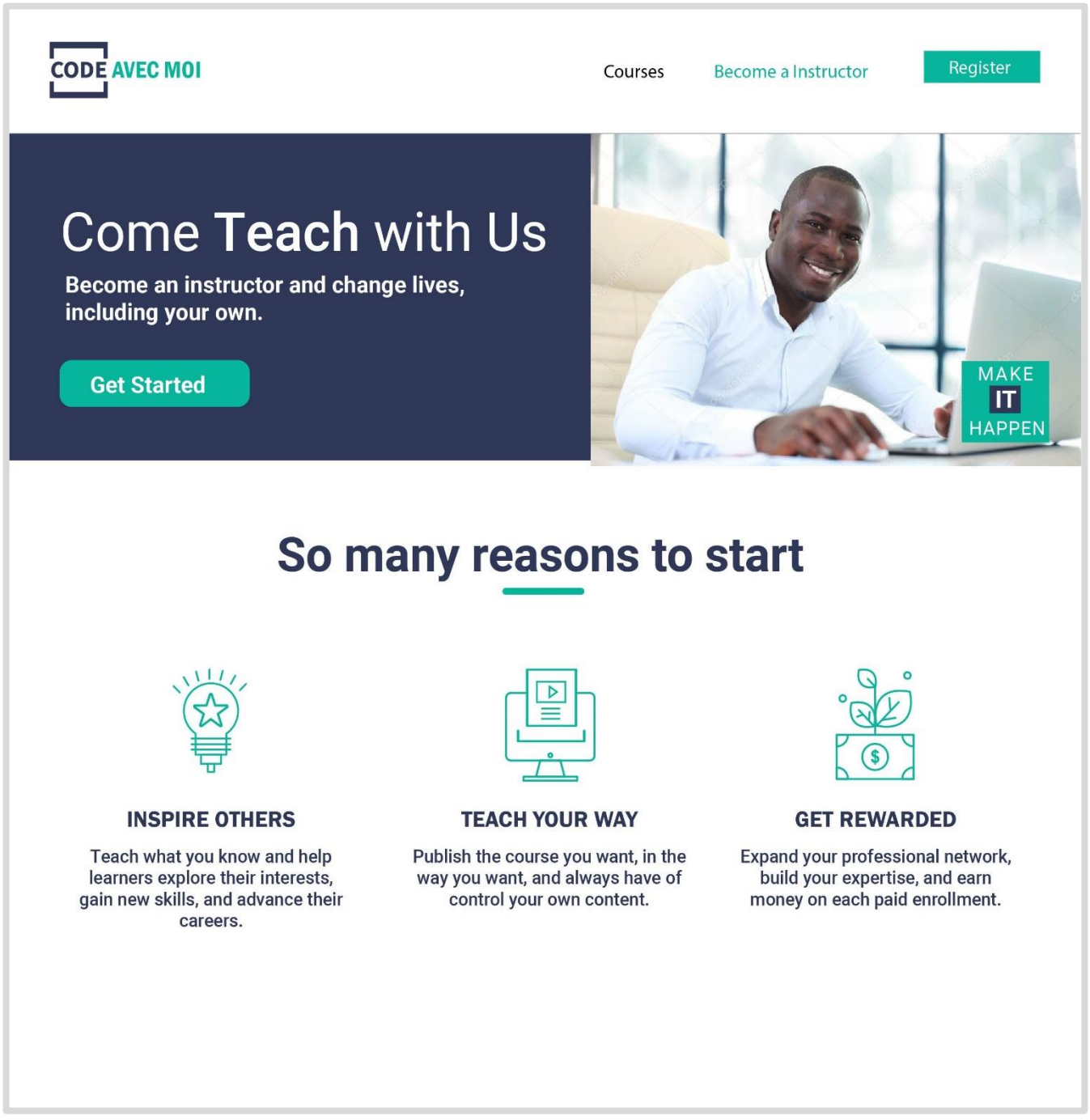


5.2.3. High Fidelity Wireframes

5.2.3.1. Home Page



5.2.3.2. Become an Instructor



6. Implementation and resources

In this stage of the project is where all the plans are putting in to action. One key step is ensuring the required resources to make it possible. As the resources for the project are restrained, was imperative to manage each activity having a strict tracking all the way, this allowed me to react fast when a problem came up.

6.1.1. Resources

To achieve this project the following resources are required:

- A MySQL database to store the data (Digital Ocean)
- A server to host the backend service (Digital Ocean)
- A server to host the front-end application (Vercel)
- A server to host all the videos and images (Amazon S3)
- A Cache server (Redis)
- Application Performance Monitoring (Datadog)
- A domain name (www.codeavecmoi.com)
- An Email server to send emails.

6.1.2. Implementation

To complete the actual project the first thing that I did was to set the environment; ensure that I have the newest version of Node JS Npm installed, Redis and MySQL installed on my machine. After this step I just created the two projects the backend and the front-end application. I started with the backend, for each feature of my application, I implemented the functionality and test that functionality with postman. Once finish with the backend, I jump to the front end to implement that feature.

Let consider for instance the feature of enrolling a student for a free course. For this feature I know that the student must be registered first before being able to enrol in a

course. In the backend I have a middleware as show on this image that check whenever a user is authenticated and have activate his account or not. The frontend of this feature will

```
export default class Student {
  public async handle ({auth}: HttpContextContract, next: () => Promise<void>) {
    const user = await auth.use( guard: 'api').authenticate()
    console.log(user.isActive)
    if (user && user.isActive && user.role === UserRole.STUDENT) {
      await next()
    } else {
      throw new UnauthorizedException('')
    }
  }
}
```

consider most of the case to create a page that the fulfil this requirement. In this example the student can enrol on the page that shows all the courses or the page that shows a single page.

```
const handleEnroll = async (e) => {
    const slug = e.currentTarget.dataset.slug
    setLoading( value: true)
    enrollForFreeCourse(slug).then(response => {
        if (response.errors) {
            console.log(response.errors)
        } else {
            setLoading( value: false);
            Router.push( url: "/student/account")
        }
    })
}
```

This code spinet above takes care of the enrolment of the student. What this code is doing is to get the course slug (unique reference to identify a course) and send the request to the API, enrollForFreeCourse will get the cookie(I am using Jwt token for authentication) and proceed the request. But how the backend deals this request, let take a look. When the request comes, the request goes first to the controller and then the controller itself talk to the service, the service has the responsibility to handle all the business requirements. Here the code snippet that take care of this logical.

```
public static async enrolForFreeCourse(slug: string, user: any) {
  const course = await this.findCourse(slug);
  const payload = {courseId: course.id, studentId: user.id, author: user.name}
  try {
    await StudentCourse.firstOrCreate(payload, payload)
    await this.teacherTotalStudent(course.id, course.teacherId)
  } catch (e) {
    Logger.error(e.message)
    throw new InternalServerException("")
  }
}  Barbier, 10/02/2021 12:28 • course api
```

I get the course slug, I verify if this course exists in the database, if it is, I enrol the student and increment total student in the teacher table. If something goes wrong, I just throw an Exception. That is in one word how I implemented the entire application. Backend, Testing with Postman and Frontend.

The entire API has more than 60 endpoints and the frond end application has more than 34 pages.

7. Testing

7.1 Testing Objectives

The results are evaluated to assess progress of design, performance, supportability, etc. Testing is required to evaluate the quality of our application and help us to identify and correct any defect found during the development. During this stage we compare the components developed with requirements and core functionalities defined for the project. The main objectives to accomplish in the testing stage for this project are:

- Evaluate the requirements, design, and code.
- Verify that the core functionalities and requirements are delivered.
- Ensure that all the requirements are implemented after de product development.
- To find defects in the product

7.2 Testing Scope

7.2.1. Scope

The Scope of the testing is based in the requirements stablished in the section four of the present document. The main features, functional and non-functional requirements of CodeAvecMoi will be tested and are grouped in four modules according to the user's roles and tasks to perform:

1. All users (Students, Teachers, Admin).
2. Students
3. Teachers
4. Admin

7.2.2. Out of Scope

Due to limited resources and time constraints the testing will be perform by myself and do not consider any testing with end users.

7.3 Methodology

For the project CodeAvecMoi we will conduct 2 types of testing:

- Unit Testing
- Manual Testing

Dummy data will be used to run the test.

7.4 Testing Execution Matrix

7.4.1. All users (Students, Teachers, Admin)

Requirement Number	Action	Description	Result	Passed/Failed
#1	Registration/Activation	Ensure the user can register with a valid email and can set a valid password (min 6 characters).	As expected	Passed
#2	Login/password	Ensure the user can sign in his account when the account is activated using his password.	As expected	Passed
#3	Login/Google	Ensure the user can sign in using his Google Account	As expected	Passed
#4	Login/Facebook	Ensure the user can sign in using his Facebook Account	As expected	Passed
#5	Sign out	Ensure the user can sign out when is authenticated in his Account	As expected	Passed

#6	Reset Password	Ensure the user can recover his password	As expected	Passed
#7	Change Password	Ensure the user can change his password	As expected	Passed
#8	Update profile	Ensure the user can update his email and personal info when is Authenticated	As expected	Passed
#9	Courses Visibility	Ensure all the courses available are visible for the user when is authenticated	As expected	Passed
#10	Coursed detail	Ensure the user can see all the courses details	As expected	Passed

7.4.2. User: Teacher

Requirement Number	Action	Description	Result	Passed/Failed
#11	Teacher Account	Ensure the user (Teacher) can register with a valid email and can set a valid password (min 6 characters).	As expected	Passed
#12	Payment settings	Ensure the user can set his payment details using a valid bank account.	As expected	Passed
#13	Profile update	Ensure the user can create, publish, and update his profile when is authenticated.	As expected	Passed
#14	Courses Creation and Management	Ensure the user can create, visualize, publish and update his courses when he is authenticated.	As expected	Passed
#15	Price Setting	Ensure the user can set a price for his courses.	As expected	Passed
#16	Comments Access	Ensure the user can see all the comments related to his courses when he is authenticated.	As expected	Passed
#17	Comments Reply	Ensure the user can reply to a comment when he is authenticated.	As expected	Passed
#18	Students' visibility	Ensure the user can see all the students enrolled in his course when he is authenticated.	As expected	Passed
#19	Income	Ensure the Teacher can see his income when he is authenticated.	As expected	Passed
#20	Transactions	Ensure the Teacher can see all the transactions related to his courses from the Dashboard.	As expected	Passed
#21	Dashboard	Ensure the Teacher can access his dashboard and can see the summary of courses activity.	As expected	Passed
#22	Notifications	Ensure the Teacher get a notification when someone join a course or leave a comment.	As expected	Passed
#23	Videos re arrangement	Ensure the Teacher can arrange easily the videos based on his preferences.	As expected	Passed
#24	Video preview	Must be able to preview his course.	As expected	Passed
#25	Video creation/edition	Must be able to create and edit the course content	As expected	Passed

7.4.3. User: Student

Requirement Number	Action	Description	Result	Passed/Failed
#26	Free Courses Access	Ensure the Student can access free courses when is authenticated.	As expected	Passed

#27	Buy Courses	Ensure the Student can buy courses when is authenticated using a valid payment method.	As expected	Passed
#28	Paid Courses Access	Ensure the Student can access his paid courses when is authenticated.	As expected	Passed
#29	Cancel Account	Ensure the user can cancel his Account when is authenticated.	As expected	Passed
#30	Post Questions	Ensure the user can post questions in a course when he is authenticated.	As expected	Passed
#31	Comments Access	Ensure the user can see all the comments related to his courses when he is authenticated.	As expected	Passed
#32	Comments Reply	Ensure the user can comment or answer questions in a thread of the course when in authenticated.	As expected	Passed
#33	Course Progress	Ensure the user can see his progress in all the courses that is registered when is authenticated.	As expected	Passed
#34	Courses Completed	Ensure the user can see his completed courses when he is authenticated.	As expected	Passed
#35	Notifications/ New courses	Ensure the user is notified when a new course is released.	As expected	Passed
#36	Notifications/ New comment	Ensure the user is notified when someone reply to his comment.	As expected	Passed

7.4.4. User: Admin

Requirement Number	Action	Description	Result	Passed/Failed
#37	Admin Dashboard Access	Ensure the user can access the admin dashboard when is authenticated.	As expected	Passed
#38	Courses Catalog Access	Ensure the user can access all the courses content when is authenticated.	As expected	Passed
#39	Users Info	Ensure the user can access to all the users info when is authenticated.	As expected	Passed
#40	Courses Notification/ Approval	Ensure the user can see a notification for courses pending to approve when is authenticated.	As expected	Passed
#41	Courses Approval	Ensure the user can approve a course when is authenticated.	As expected	Passed
#42	Performance indicators	Ensure the user can visualize performance indicators such as the bestselling courses, total courses, total sells, total students, new teachers, new students, and pending curses when is authenticated.	As expected	Passed
#43	Transactions	Ensure the user can see the transactions and payments when is authenticated.	As expected	Passed
#44	API Health	Ensure the Admin gets notified, when is Authenticated, about the API heath, to corroborate if the application is up running properly.	As expected	Passed

8. References

- W., 2021. What is Node.js?. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/1884724/what-is-node-js/9634307#9634307> [Accessed 6 May 2021].

- Redhat, 2021. [online] Available at: <https://www.redhat.com/en/topics/middleware/what-is-ide#:~:text=An%20integrated%20development%20environment%20(IDE,graphical%20user%20interface%20(GUI).> [Accessed 15 May 2021].

- Ant.design. 2021. Ant Design of React - Ant Design. [online] Available at: <https://ant.design/docs/react/introduce> [Accessed 6 March 2021].

- DEV Community. 2021. Why to choose NextJS for future projects. [online] Available at: <https://dev.to/yuridevat/why-to-choose-nextjs-for-future-projects-4k8o> [Accessed 10 May 2021].

- Developer.mozilla.org. 2021. HTML basics - Learn web development | MDN. [online] Available at: <https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics> [Accessed 13 April 2021].

- GitHub. 2021. AdonisJS Framework. [online] Available at: <https://github.com/adonisjs> [Accessed 23 February 2021].

- Legacy.adonisjs.com. 2021. About AdonisJs. [online] Available at: <https://legacy.adonisjs.com/docs/4.0/about> [Accessed 10 May 2021].

- Medium. 2021. Understanding the Ant Design System — a UI Design for Enterprises. [online] Available at: <https://betterprogramming.pub/understanding-the-ant-design-system-a-ui-design-for-enterprises-39afdb188b06?gi=e2656f992eed> [Accessed 10 May 2021].

- Nielsen Norman Group. 2021. The Definition of User Experience (UX). [online] Available at: <https://www.nngroup.com/articles/definition-user-experience/> [Accessed 11 March 2021].

- Nielsen Norman Group. 2021. 10 Usability Heuristics for User Interface Design. [online] Available at: <https://www.nngroup.com/articles/ten-usability-heuristics/> [Accessed 14 February 2021].

- Node.js. 2021. About | Node.js. [online] Available at: <https://nodejs.org/en/about/> [Accessed 6 April 2021].

- npm. 2021. adonis-auth. [online] Available at: <https://www.npmjs.com/package/adonis-auth> [Accessed 18 January 2021].

- Reactjs.org. 2021. React – A JavaScript library for building user interfaces. [online] Available at: <https://reactjs.org/> [Accessed 22 February 2021].

- Sacolick, I., 2021. What is CI/CD? Continuous integration and continuous delivery explained. [online] InfoWorld. Available at: <https://www.infoworld.com/article/3271126/what-is-cicd-continuous-integration-and-continuous-delivery-explained.html> [Accessed 9 May 2021].

- Techterms.com. 2021. Java Definition. [online] Available at: <https://techterms.com/definition/java> [Accessed 4 May 2021].

- Typescriptlang.org. 2021. [online] Available at: <https://www.typescriptlang.org/> [Accessed 10 May 2021].

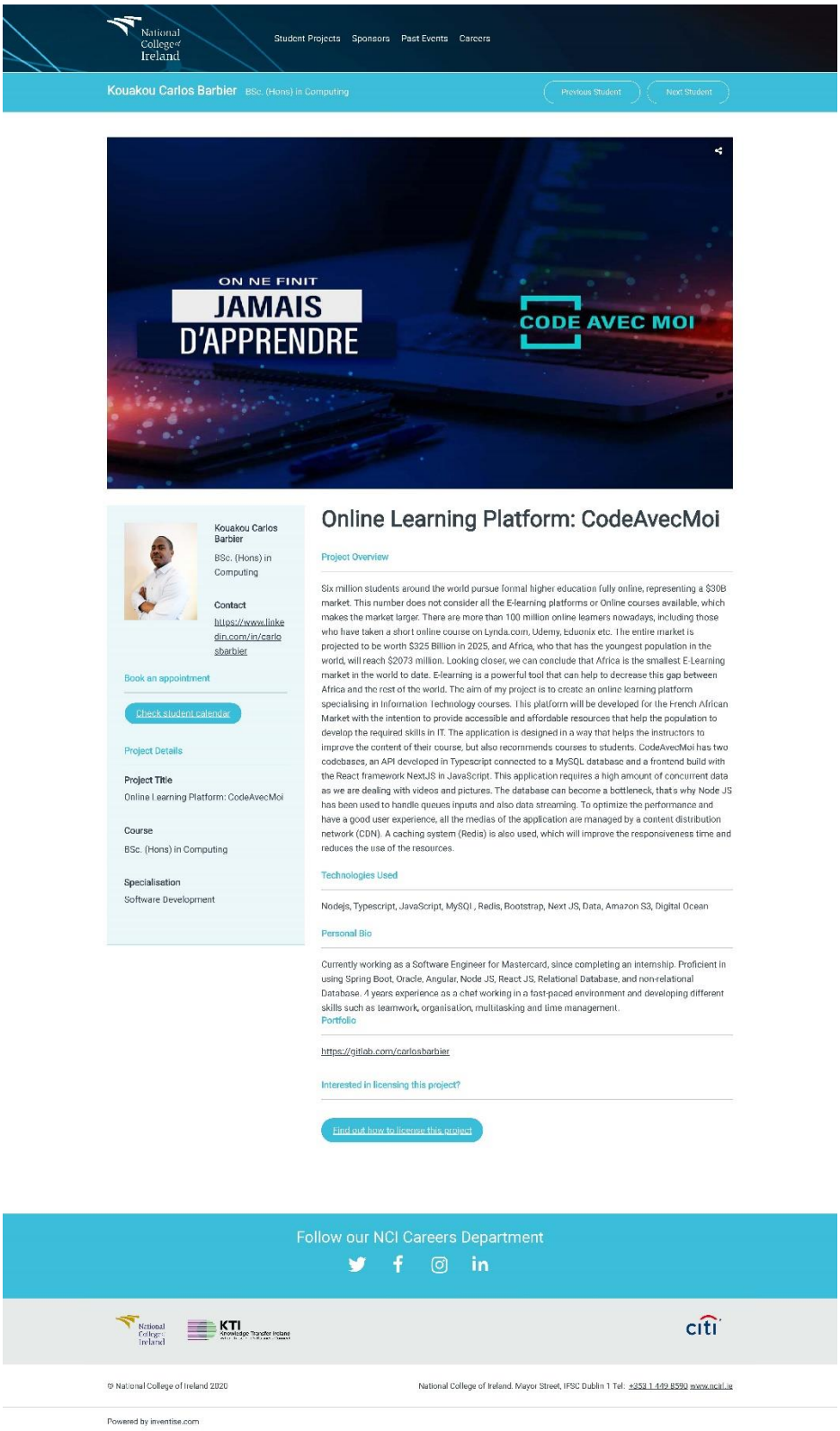
Appendix

A. Gitlab links

https://gitlab.com/carlosbarbier/codeavecmoi-frontend

https://gitlab.com/carlosbarbier/codeavecmoi-backend

B. Showcase Poster



Journal

October Journal

Name: Barbier Kouakou Carlos

Number: 17117852

Week 1: I was very happy to start the college on the 28th of September, meet my teachers and my classmates with a different environment, due to the COVID 19 for the first semester we are take our classes online. The first week was challenging, I did not know how things will work, how I will interact with my teachers and my classmates. At the end of the first week I was able to understand the entire platform, the modules, how the modules will be evaluated in one word have a general overview of the semester.

Week 2: The second week was very busy at work because some of my colleagues were on holidays. I was a little tired, but I attempted all my classes. I got familiar with the modules, we started getting deeper in the courses content. Personally, I was not attracted to mobile Development but when I started, I really enjoyed it and I could say it is so far my favourite module. I failed a little bit under pressure because I needed to decide what will be the final year project.

Week 3: This week I started exploring different ideas that I had in my mind and make some research about my ideas. I was a little indecisive in the beginning of the week but by Friday I had a clear idea of what I want to do for my final year project. As we have classes all Saturday, I dedicated all Sunday morning to structure my idea and write my pitch. In the afternoon I recorded the video and upload on moodle

Week 4: This week I really realised that I must be extremely organize with work and school. My priority was to create a calendar with all my deliverables and organize, define my weekly tasks. I started all my projects and CAs. Regarding the classes I think we are more familiar with the online system; I see many of my classmates interacting with the teachers.

January Journal

Name: Barbier Kouakou Carlos

Number: 17117852

National College of Ireland

Week 1: The first week of January was very busy for me as I had to submit two final assignments. The first one was for Introduction to Artificial Intelligence module and consisted in a Terminal Assignment Based Assessment (TABA), and the second assignment was from the module Strategic Management where I had to do a Value Chain Analysis.

Week 2: During this week I decided to take a break from the school after I submitted all my mid-term assignments. I used my spare time to rest and do some sport after work which was very helpful as this week was more demanding at work.

Week 3: After my short break from school projects, this week I continued with my final year project. I had a call with my supervisor, and he gave me feedback about my Mid-Point Implementation, Documentation & Video Presentation. My supervisor gave me insights about my project which was very helpful to continue improving my application.

Week 4: This week I started again my classes, and I continued working with my project. I could not advance as I expected due to the classes and work, but I was able to do some progress in the UI.

February Journal

Name: Barbier Kouakou Carlos

Number: 17117852

Week 1: I was not feeling very well the first week of February, I was constantly feeling tired all the times. I called my General Practitioner (GP) to let him know about the situation. He recommends me to make the Coronavirus Test, thing that I did, the result was negative. I have appointment with him to seek what was the root cause of the tiredness, I made more test everything was ok. I could not do that much on my projects due to this situation.

Week 2: During this week I make a good progress on my final year project. I could say roughly I have 50% done.

Week 3: This week I was most focus on learning Ruby on Rails for the module Cloud Application Development. Ruby on Rails is a MVC web framework to build website and applications. I am familiar with MCV pattern but not Ruby language. It was straightforward to pick up the structure of Ruby on Rails, the main challenge was Ruby Language itself but comparing to other languages like Java Ruby is very simple. I also have dedicated a good time on Usability Design CA1.

Week 4: I worked on Distributed System Cloud Application Project. Regarding to Distributed System, I submitted the proposal and committed the first codes. For Cloud Application Development, I completed the Authentication feature.

March Report

Name: Barbier Kouakou Carlos

Number: 17117852

Week 1, Week 3, Week 2: During these three weeks of March, I worked on my projects in Usability Design, Distributed System and Cloud application development. I could not work on my final year project.

Week 4: I had some free time to work on my project by doing firstly a refactoring and later work on the rating system. As the time is flying, I am taking 10 days of from work to finish my final year project.

April Report

Name: Barbier Kouakou Carlos

Number: 17117852

Week 1, Week 2, Week 3: During these three weeks of April, I worked on my final assignments from Usability Design, Distributed System and Cloud application development. I work on my spare time on final year project in the admin area.

Week 4: This week I asked for holidays at work to dedicate 100% of the time to my project. While working in the teacher's admin area, I faced a memory leak issue due to a circular dependency. To find and solve this issue took me 4 days, which was a setback in my project but is solved now.

C. Project Proposal

National College of Ireland

BSc Honours in Computing

Software Engineering

Academic Year: 2020/2021

Barbier Kouakou Carlos

Student number:17117852

Email: x17117852@student.ncirl.ie

PROJECT

Online Learning Platform: CodeAvecMoi

Technical Report

1.0 Objectives

The objective of this project is to create an online learning platform specialized in Information Technology Courses. This platform will be developed for the African Market with the intention to provide accessible and affordable resources that help the population to develop the required skills in IT.

The System will be designed applying Artificial Intelligence to integrate a Recommendation Engine for the students based on their skills, knowledge, performance, preferences and usability. Additionally, the artificial intelligence module will give feedback to the content providers helping them to improve their courses.

2.0 Background

Education is a powerful agent of change. It improves health and livelihoods, contributes to social stability and stimulates long-term economic growth. According to UNESCO, 420 million people would be lifted out of poverty if they had a secondary education. As we can notice the education is the key factor of any development in our society. Today, with the ease of access of internet and technologies, people are changing the way they are learning, the skills and knowledge can be transferred easily to anyone and anywhere in the world.

Six million students around the world pursue formal higher education fully online, representing a $30B market. This number do not consider all the E-learning platforms or Online course available which will make the market larger. There are more than 100 million online learners nowadays, and likely who have taken a short online course on Lynda.com, Udemy, Eduonix etc. The entire market is projected to be worth $325 Billion in 2025, and Africa, who that has the youngest population in the world, will reach $2073 million. Looking closer, we can conclude that Africa is the smallest E-Learning

market in the world to date, in one word there many to do in the education sector and E-learning is a powerful tool that can help to decrease this gap between Africa and the rest of the world.

3.0 Technical Approach

The first task for this project is to make a technical research to identify the best practices in E-Learning industry to incorporate the most suitable in my application. I will also research about the latest trends Artificial Intelligence to apply it to my recommendation engine.

For this project, I will be using the Agile methodology. Therefore, I am dividing the system in modules. The first structure proposal considers the following modules:

- Authentication module
- Course creation module
- Admin dashboard module
- Student dashboard module
- Teacher dashboard module
- Course recommendation module
- Subscriptions and Payment module
- Quiz module
- Certification module

4.0 Special Resources Required

To achieve this project the following resources are required:

- A MySQL database to store the data (Digital Ocean)
- A server to host the backend service (Digital Ocean)
- A server to host the front-end application (Vercel)
- A server to host all the videos and images (Amazon S3)

- A Cache server (Redis)
- Application Performance Monitoring (Datadog)
- A domain name
- An Email server to send emails

5.0 Project Plan

The following Gantt chart shows in details the main tasks and milestones for my project. I will be tracking my project using this file that I created in Excel (Appendix 1).

[CodeAvecMoi] Project Schedule

Final Year Project

Project Start Date 10/12/2020 (Monday) Display Week 1

Project Lead Carlos Barbier

WBS	TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS
1	**Deliverables**			-			-
1.1	Video Spech		Mon 10/12/20	Sun 10/18/20	7	100%	5
1.2	Project Proposal		Sun 11/01/20	Sun 11/08/20	8	100%	5
1.3	Mid Point Implementation		Mon 11/09/20	Tue 12/22/20	44	0%	32
1.3.1	Documentation		Mon 11/09/20	Fri 12/18/20	40	0%	30
1.3.2	Video Presentation		Fri 12/18/20	Sun 12/20/20	3	0%	1
1.4	Final Implementation & Documentation		Sun 1/03/21	Tue 4/27/21	115	0%	82
1.5	Video Presentation		Sat 5/01/21	Fri 5/14/21	14	0%	10
1.6	Viva Examination		Thu 5/20/21	Fri 5/21/21	2	0%	2
1.7	Project Showcase		Mon 5/24/21	Sat 5/29/21	6	0%	5
2	**System Design**			-			-
2.1	System Architecture		Mon 11/09/20	Tue 11/10/20	2	0%	2
2.2	UI design		Thu 11/12/20	Sat 11/14/20	3	0%	2
3	**Authentication Module**			-			-
3.1	Login and Registration		Sun 11/15/20	Wed 11/18/20	4	0%	3
3.2	Reset password ,Forgot Password		Thu 11/19/20	Sat 11/21/20	3	0%	2
3.3	Account Validation		Sun 11/22/20	Sun 11/22/20	1	0%	0
4	**Course Creation Module**			-			-
4.1	Course Category		Tue 11/24/20	Wed 11/25/20	2	0%	2
4.2	Crud operation		Thu 11/26/20	Sat 12/05/20	10	0%	7

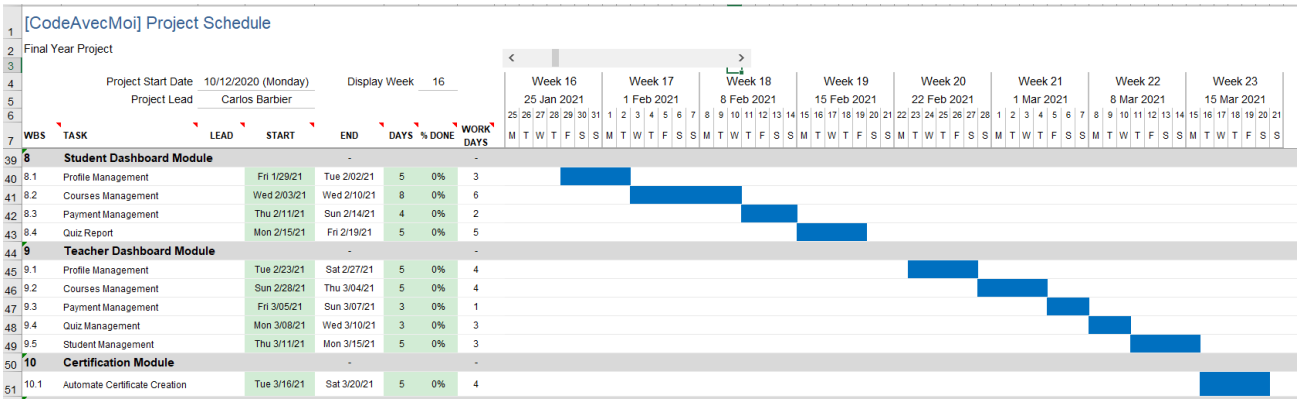
[CodeAvecMoi] Project Schedule

Final Year Project

Project Start Date 10/12/2020 (Monday) Display Week 9

Project Lead Carlos Barbier

WBS	TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS
5	**Subscriptions and Payment Module**			-			-
5.1	Payment Sytem Integration		Mon 12/07/20	Wed 12/09/20	3	0%	3
5.2	Subscriptions Feature		Thu 12/10/20	Sat 12/12/20	3	0%	2
6	**Quiz Module**			-			-
6.1	Crud Operation		Sun 12/13/20	Mon 12/21/20	9	0%	6
7	**Admin Dashboard Module**			-			-
7.1	User Management		Sun 1/03/21	Thu 1/07/21	5	0%	4
7.2	Course Management		Fri 1/08/21	Fri 1/15/21	8	0%	6
7.3	Bill Management		Sat 1/16/21	Mon 1/18/21	3	0%	1
7.4	Quiz management		Tue 1/19/21	Sat 1/23/21	5	0%	4
7.5	Comment and Review Management		Sun 1/24/21	Thu 1/28/21	5	0%	4



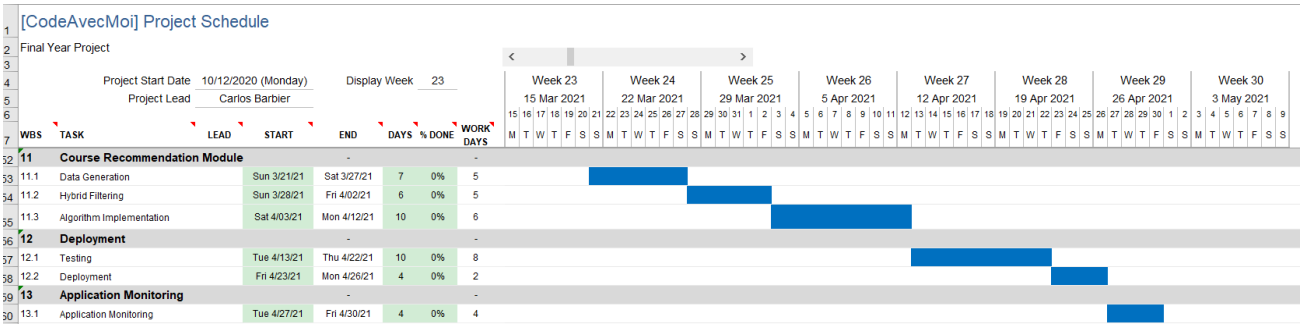
[CodeAvecMoi] Project Schedule

Final Year Project

Project Start Date 10/12/2020 (Monday) Display Week 16

Project Lead Carlos Barbier

WBS	TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS
8	**Student Dashboard Module**			-			-
8.1	Profile Management		Fri 1/29/21	Tue 2/02/21	5	0%	3
8.2	Courses Management		Wed 2/03/21	Wed 2/10/21	8	0%	6
8.3	Payment Management		Thu 2/11/21	Sun 2/14/21	4	0%	2
8.4	Quiz Report		Mon 2/15/21	Fri 2/19/21	5	0%	5
9	**Teacher Dashboard Module**			-			-
9.1	Profile Management		Tue 2/23/21	Sat 2/27/21	5	0%	4
9.2	Courses Management		Sun 2/28/21	Thu 3/04/21	5	0%	4
9.3	Payment Management		Fri 3/05/21	Sun 3/07/21	3	0%	1
9.4	Quiz Management		Mon 3/08/21	Wed 3/10/21	3	0%	3
9.5	Student Management		Thu 3/11/21	Mon 3/15/21	5	0%	3
10	**Certification Module**			-			-
10.1	Automate Certificate Creation		Tue 3/16/21	Sat 3/20/21	5	0%	4



[CodeAvecMoi] Project Schedule

Final Year Project

Project Start Date 10/12/2020 (Monday) Display Week 23

Project Lead Carlos Barbier

WBS	TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS
11	**Course Recommendation Module**			-			-
11.1	Data Generation		Sun 3/21/21	Sat 3/27/21	7	0%	5
11.2	Hybrid Filtering		Sun 3/28/21	Fri 4/02/21	6	0%	5
11.3	Algorithm Implementation		Sat 4/03/21	Mon 4/12/21	10	0%	6
12	**Deployment**			-			-
12.1	Testing		Tue 4/13/21	Thu 4/22/21	10	0%	8
12.2	Deployment		Fri 4/23/21	Mon 4/26/21	4	0%	2
13	**Application Monitoring**			-			-
13.1	Application Monitoring		Tue 4/27/21	Fri 4/30/21	4	0%	4

6.0 Technical Details

The stack that I will use for this project is Node JS and React JS. The choice of this Technology is due to the flexibility of working with the Server Side and Client Side using the Programming Language that in this is JavaScript.

Node JS is has become very popular because it allows you to build a fast, scalable, lightweight and hight performant applications which make it a good fit for my project.

For the recommendation module, I will be using python as it is one the most powerful programming language to manage data and has many libraries for data science.

To store the data, I will be use MySQL and Redis.

7.0 Evaluation

The testing will be done using dummy data, unit tests and integration test. This decision is made by the fact that with the current situation, to the testing with real users is extremely complex. To make sure that the System is working properly, I will use a monitoring tool to check the health and performance of the application.

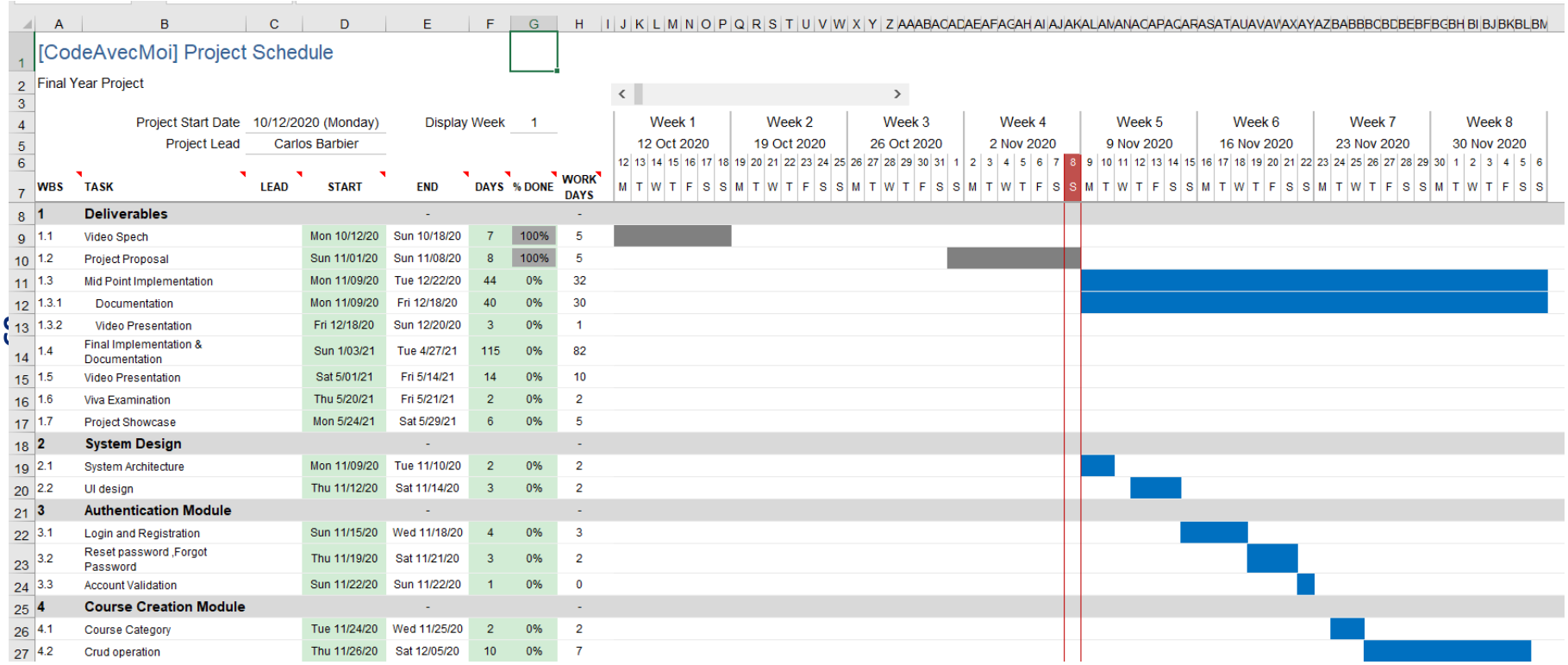
Appendix

[CodeAvecMoi] Project Schedule

Final Year Project

Project Start Date: 10/12/2020 (Monday)
Display Week: 1
Project Lead: Carlos Barbier

WBS	TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS
1	**Deliverables**			-			-
1.1	Video Spech		Mon 10/12/20	Sun 10/18/20	7	100%	5
1.2	Project Proposal		Sun 11/01/20	Sun 11/08/20	8	100%	5
1.3	Mid Point Implementation		Mon 11/09/20	Tue 12/22/20	44	0%	32
1.3.1	Documentation		Mon 11/09/20	Fri 12/18/20	40	0%	30
1.3.2	Video Presentation		Fri 12/18/20	Sun 12/20/20	3	0%	1
1.4	Final Implementation & Documentation		Sun 1/03/21	Tue 4/27/21	115	0%	82
1.5	Video Presentation		Sat 5/01/21	Fri 5/14/21	14	0%	10
1.6	Viva Examination		Thu 5/20/21	Fri 5/21/21	2	0%	2
1.7	Project Showcase		Mon 5/24/21	Sat 5/29/21	6	0%	5
2	**System Design**			-			-
2.1	System Architecture		Mon 11/09/20	Tue 11/10/20	2	0%	2
2.2	UI design		Thu 11/12/20	Sat 11/14/20	3	0%	2
3	**Authentication Module**			-			-
3.1	Login and Registration		Sun 11/15/20	Wed 11/18/20	4	0%	3
3.2	Reset password ,Forgot Password		Thu 11/19/20	Sat 11/21/20	3	0%	2
3.3	Account Validation		Sun 11/22/20	Sun 11/22/20	1	0%	0
4	**Course Creation Module**			-			-
4.1	Course Category		Tue 11/24/20	Wed 11/25/20	2	0%	2
4.2	Crud operation		Thu 11/26/20	Sat 12/05/20	10	0%	7



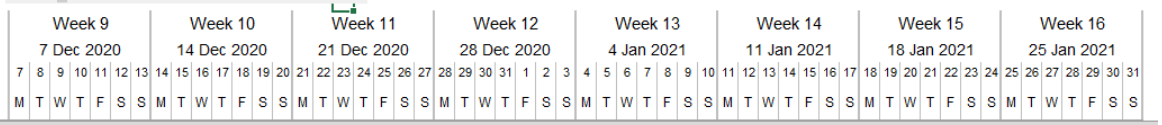
[CodeAvecMoi] Project Schedule

Final Year Project

Project Start Date: 10/12/2020 (Monday) | Display Week: 9

Project Lead: Carlos Barbier

WBS	TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS
5	**Subscriptions and Payment Module**			-			-
5.1	Payment Sytem Integration		Mon 12/07/20	Wed 12/09/20	3	0%	3
5.2	Subscriptions Feature		Thu 12/10/20	Sat 12/12/20	3	0%	2
6	**Quiz Module**			-			-
6.1	Crud Operation		Sun 12/13/20	Mon 12/21/20	9	0%	6
7	**Admin Dashboard Module**			-			-
7.1	User Management		Sun 1/03/21	Thu 1/07/21	5	0%	4
7.2	Course Management		Fri 1/08/21	Fri 1/15/21	8	0%	6
7.3	Bill Management		Sat 1/16/21	Mon 1/18/21	3	0%	1
7.4	Quiz management		Tue 1/19/21	Sat 1/23/21	5	0%	4
7.5	Comment and Review Management		Sun 1/24/21	Thu 1/28/21	5	0%	4



[CodeAvecMoi] Project Schedule

Final Year Project

Project Start Date: 10/12/2020 (Monday) | Display Week: 16

Project Lead: Carlos Barbier

WBS	TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS
8	**Student Dashboard Module**			-			-
8.1	Profile Management		Fri 1/29/21	Tue 2/02/21	5	0%	3
8.2	Courses Management		Wed 2/03/21	Wed 2/10/21	8	0%	6
8.3	Payment Management		Thu 2/11/21	Sun 2/14/21	4	0%	2
8.4	Quiz Report		Mon 2/15/21	Fri 2/19/21	5	0%	5
9	**Teacher Dashboard Module**			-			-
9.1	Profile Management		Tue 2/23/21	Sat 2/27/21	5	0%	4
9.2	Courses Management		Sun 2/28/21	Thu 3/04/21	5	0%	4
9.3	Payment Management		Fri 3/05/21	Sun 3/07/21	3	0%	1
9.4	Quiz Management		Mon 3/08/21	Wed 3/10/21	3	0%	3
9.5	Student Management		Thu 3/11/21	Mon 3/15/21	5	0%	3
10	**Certification Module**			-			-
10.1	Automate Certificate Creation		Tue 3/16/21	Sat 3/20/21	5	0%	4