



**TRIBHUVAN UNIVERSITY  
INSTITUTE OF ENGINEERING  
PULCHOWK CAMPUS**

**A  
MAJOR PROJECT REPORT  
ON  
IOE PORTALS:  
A INTEGRATED EDUCATIONAL TOOL**

**SUBMITTED BY**

NIRAJ SHRESHTA (PUL075BCT054)  
RUPAK RAJ PANTHA (PUL075BCT070)  
SITAL NAGARKOTI (PUL075BCT085)  
SMARAN DHUNGANA (PUL075BCT086)

**SUBMITTED TO**

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING  
LALITPUR, NEPAL

May 3, 2023

# Page of Approval

TRIBHUVAN UNIVERSITY  
INSTITUTE OF ENGINEERING  
PULCHOWK CAMPUS  
DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

The undersigned certifies that they have read and recommended to the Institute of Engineering for acceptance of a project report entitled "IOE Portals: A Integrated Educational Tools" submitted by **Niraj Shrestha, Rupak Raj Pantha, Sital Nagarkoti, Smaran Dhungana** in partial fulfillment of the requirements for the Bachelor's degree in Electronics & Computer Engineering.

.....

Supervisor

**Dr. Ram Krishna Maharjan**

Professor

Department of Electronics and Computer

Engineering,

Pulchowk Campus, IOE, TU.

.....

Internal examiner

.....

External examiner

Date of approval:

# Copyright

The author has agreed that the Library, Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering may make this report freely available for inspection. Moreover, the author has agreed that permission for extensive copying of this project report for scholarly purposes may be granted by the supervisors who supervised the project work recorded herein or, in their absence, by the Head of the Department wherein the project report was done. It is understood that the recognition will be given to the author of this report and to the Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering in any use of the material of this project report. Copying or publication or the other use of this report for financial gain without approval of to the Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering and author's written permission is prohibited.

Request for permission to copy or to make any other use of the material in this report in whole or in part should be addressed to:

Head

Department of Electronics and Computer Engineering

Pulchowk Campus, Institute of Engineering, TU

Lalitpur, Nepal.

# Acknowledgement

We feel immensely blessed and grateful for the invaluable guidance and assistance we received from numerous individuals throughout the completion of our Major Project. Firstly, we express our sincerest appreciation to the Institute of Engineering for incorporating the major project as an integral component of our curriculum. We extend our heartfelt thanks to the Department of Electronics and Computer Engineering for initiating and facilitating this project.

Our profound gratitude goes to our college for providing us with the necessary resources and support to work on this academic project. We are particularly indebted to our esteemed supervisor, Prof. Dr. Ram Krishna Maharjan, for his constructive and motivating suggestions regarding this project.

Finally, we would also like to express our sincere appreciation to our classmates, families, and friends for their encouragement, love, support, suggestions, and constructive criticism. Their unwavering support and motivation have kept us going throughout the project.

Thank you all for your invaluable support and contribution toward the successful completion of this project.

Sincerely,

Niraj Shrestha

Rupak Raj Pantha

Sital Nagarkoti

Smaran Dhungana

# Abstract

This report presents a mobile application project, titled "IOE Portals: An Integrated Educational Tools," aimed at addressing the digitalization gap in educational sectors, particularly in academic settings such as universities. The project is in response to the challenges faced by students and teachers in accessing academic resources and keeping up with the ongoing happenings at the institution. The mobile application includes features such as routine, syllabus, library, notification, personalized content, and probable exam question. Additionally, a web app for admin is created to manage the application's backend. The project's scope includes utilizing the application as a platform for sharing and accessing digital educational resources, organizing schedules, and providing informative resources. The report includes the project's problem statement, objectives, methodology, and results, including a screenshots of the application's interface. Overall, the project aims to provide digital assistance for the welfare of educational institutions and improve academic performance by enhancing accessibility to educational resources.

**KEYWORDS:** mobile application, digitalization, learning materials, resource acquisition, engagement, student accessibility, educational resources.

# Contents

Page of Approval	i
Copyright	ii
Acknowledgement . . . . .	iii
List of Figures . . . . .	viii
<b>1 Introduction</b>	<b>1</b>
1.1 Problem Statement . . . . .	1
1.2 Objectives . . . . .	2
1.3 Scope and Applications . . . . .	2
<b>2 Literature Review</b>	<b>3</b>
2.1 Mobile Applications in Higher Education . . . . .	3
2.2 Mobile Learning and it's impact . . . . .	3
2.3 Hybrid Recommendation System . . . . .	4
<b>3 Related Theory</b>	<b>5</b>
3.1 Clean Architecture . . . . .	5
3.2 Role-based access control . . . . .	6
3.2.1 Design . . . . .	6
3.2.2 Standardized levels . . . . .	7
3.2.3 Attribute-based access control . . . . .	7
3.3 Bidirectional Sync and TimeStamp Based Conflict Resolution . . . . .	8
3.4 Collaborative Filtering . . . . .	8
3.5 Content Based Filtering . . . . .	9
3.6 TF-IDF Vectorizer . . . . .	10
3.7 Cosine Similarity . . . . .	11
3.8 CMS . . . . .	12
<b>4 Methodology</b>	<b>13</b>
4.1 Preliminary . . . . .	13
4.2 Question Prediction System . . . . .	13

4.2.1	Data Collection . . . . .	13
4.2.2	Data Preprocessing . . . . .	14
4.2.3	OCR . . . . .	14
4.2.4	Topic Extraction . . . . .	15
4.2.5	Model Training . . . . .	15
4.2.6	Deployment . . . . .	16
4.3	Hybrid Recommendation System . . . . .	17
4.3.1	Data Collection . . . . .	17
4.3.2	Data preprocessing . . . . .	17
4.3.3	Model selection and training . . . . .	17
4.3.4	Evaluation . . . . .	17
4.3.5	Deployment . . . . .	18
4.4	Role-Based Access Control . . . . .	18
4.5	Notification System . . . . .	18
4.6	Test Driven Development . . . . .	18
4.7	AWS Deployment . . . . .	19
4.8	Development . . . . .	21
4.8.1	GitHub . . . . .	21
4.8.2	CircleCI . . . . .	21
4.8.3	AWS CodePipeline . . . . .	22
4.8.4	AWS CodeBuild . . . . .	22
4.8.5	AWS CodeDeploy . . . . .	22
4.8.6	Testing the pipeline . . . . .	22
4.8.7	Monitoring and debugging . . . . .	22
4.9	Tools and Technology Used . . . . .	22
<b>5</b>	<b>System Design</b>	<b>24</b>
5.1	Project Architecture . . . . .	24
5.1.1	System Design . . . . .	24
5.1.2	Usecase Diagram . . . . .	26
5.2	Database . . . . .	27
5.3	Project Development . . . . .	30
5.4	Backend . . . . .	32
5.4.1	JWT Token Authentication . . . . .	32
<b>6</b>	<b>Result</b>	<b>37</b>
6.1	Mobile Features . . . . .	38

6.1.1	Login . . . . .	38
6.1.2	Sign Up . . . . .	39
6.1.3	Library . . . . .	40
6.1.4	Syllabus and Routine . . . . .	41
6.1.5	Document Viewer . . . . .	42
6.1.6	Notification . . . . .	44
6.1.7	Probable Exam Question . . . . .	45
6.2	Web App . . . . .	46
<b>7</b>	<b>Conclusion</b>	<b>47</b>
<b>8</b>	<b>Limitations and Future Enhancement</b>	<b>48</b>
	References . . . . .	50



# List of Figures

3.1	Clean Architecture Diagram . . . . .	6
4.1	Question Prediction Pipeline . . . . .	13
4.2	OCR Flowchart . . . . .	14
4.3	TFIDF and Cosine Similarity Flowchart . . . . .	16
4.4	AWS Architecture . . . . .	20
4.5	CI/CD Workflow . . . . .	21
5.1	System Architecture Diagram . . . . .	24
5.2	System Diagram . . . . .	25
5.3	System Usecase Diagram . . . . .	26
5.4	User System ER Diagram . . . . .	28
5.5	Content System ER Diagram . . . . .	29
5.6	Miscellaneous System ER Diagram . . . . .	30
5.7	Sample UI Flow Diagram . . . . .	31
5.8	No Auth Token Sequence Diagram . . . . .	33
5.9	Logged In: Expired Access Token Sequence Diagram . . . . .	34
5.10	Logged In:Valid Access Token Sequence Diagram . . . . .	35
5.11	Logged In: Both Token Expired Sequence Diagram . . . . .	36
6.1	Login Page . . . . .	38
6.2	Sign Up Page . . . . .	39
6.3	Library Page . . . . .	40
6.4	Syllabus and Routine . . . . .	41
6.5	Document Viewer . . . . .	42
6.6	Document Contents . . . . .	43
6.7	Notification Page . . . . .	44
6.8	Probable Exam Question . . . . .	45
6.9	Web App . . . . .	46

# 1. Introduction

Over the past decade, the use of digital technology has seen an exponential increase. The rapid advancement of digital technology in the last decade has resulted in exponential growth in its usage. The widespread adoption of smartphones has played a crucial role in the popularity of mobile applications. These applications have become essential tools for a variety of tasks, ranging from basic functions such as setting alarms to complex ones such as accessing vast amounts of information. In recent years, the digitalization of education has been a tremendous boon to both teachers and students.

In 2022, Statista reported that the education category was the third most popular category of apps downloaded from the Apple App Store in the first quarter of the year 2022[1]. This explosion in popularity of mobile applications presents a significant opportunity for the development of educational applications, especially in academic settings like universities. The potential for using mobile technology to improve and enhance the learning experience for students is vast, and educators are continually exploring ways to utilize this technology to facilitate learning in innovative ways.

Educational applications of mobile technology have tremendous potential to revolutionize the way students learn, making learning more engaging, accessible, and interactive. Mobile applications offer a wide range of benefits, such as personalized learning experiences, and the ability to access learning materials from anywhere at any time. They also have the potential to reduce the cost of education by making educational resources more accessible and affordable to students. The rapid pace of technological advancement indicates that the potential for digital education is limitless, and it is up to educators to tap into its potential to benefit students.

## 1.1 Problem Statement

The usage of digital technology in educational sectors has witnessed an unprecedented growth in recent years, with the widespread adoption of smartphones and other mobile devices. However, mobile applications have been overlooked despite their potential in enhancing the learning experience for students and teachers alike.

One of the primary challenges faced by students is the lack of easy access to essential academic resources such as the semester's schedule and materials. This can cause inconvenience and

difficulty in keeping. Teachers, too, encounter obstacles in managing study resources and providing informative resources to students. This can lead to frustration and dissatisfaction among both students and teachers, ultimately leading to a decline in academic performance.

Therefore, to address these pressing issues, we propose the development of a mobile application that provides digital assistance to students and teachers. This application aims to simplify various academic tasks, such as accessing schedules, materials. By improving the accessibility of academic resources, the application will enhance the learning experience for students and streamline academic activities for teachers.

## **1.2 Objectives**

IOE Portal intends to accomplish the following purposes-

1. To create a mobile application for educational institutes.
2. To ease the accessibility of different educational resources for teachers, students and administrators through an integrated application.
3. To ease the sharing of digital educational resources for teachers and publications.

## **1.3 Scope and Applications**

The scope of this application is enumerated in the points listed below:

1. The application can be utilized by teachers and student as a platform to share and access digital educational resources.
2. Students can use the routine and syllabus tools to organize their schedules.
3. The application will also provide a feature for teachers to manage and update the course syllabus, including course outlines
4. The application will be designed to be user-friendly and accessible to a wide range of users, including those who may not be particularly tech-savvy.

## 2. Literature Review

### 2.1 Mobile Applications in Higher Education

As of the summer of 2014, education was the third most popular category among the more than 75 billion mobile programs (apps) downloaded from the Apple App Store [1]. Researchers note that mobile technology is quickly being employed as a learning aid, thanks to technological developments such as wireless network development and device capacity increases [2][3]. Currently, mobile apps are considered a key emerging technology in higher education. Students nowadays have been labelled as "Digital Natives," who are heavily impacted by technological advancements[4]. Bomhold (2013) highlighted that the ease of acquiring information is vital for today's digital natives, due to their expectation that information be available "anytime, anywhere"[5].

Previous research on mobile apps in higher education has mostly examined the usage rate of various kinds of apps in an attempt to predict students' behaviour when using them for learning. One of the key findings from Bomhold (2013) shows that 76 % of college students use smartphone applications for academic instruction. Additionally, other studies contrast the use of applications among various student groups (distinguished by their academic disciplines). For example, Kim et al. (2013) analysed how students from different majors (Education and Engineering) used smartphones for learning through their regularly used apps in a Korean university[3].

### 2.2 Mobile Learning and its impact

Due to the growing use of mobile devices, the possibility of spatial problems has been significantly decreased and people may now be reached and connected anytime, anywhere. Since the phrase "mobile learning" was first used in 2005, several academics and professionals have tried to define it. The first definitions were either techno-centric or only focused on gadgets and technology. The most often used definition of mobile learning is "learning in different contexts, through social and content exchanges, using personal electronic devices. [6]" It has been noted that mobile technology and communication have impacted many disciplines, including education, and are now undergoing change known as the "mobile age." [7]. One can learn no matter where they are or what time it is since knowledge is accessible anywhere and at any time.[8] Learning may therefore take place everywhere. It can take many forms,

and when a person is ready to learn, access to education need not always occur in a formal classroom setting. This led UNESCO to believe that M-Learning broadens the reach and promotes equality in education.

Despite prior designs that have been established, there are still no transferrable frameworks for mobile learning. Practitioners, instructional designers, and trainers have a higher need for design theories so they may more effectively incorporate mobile devices into learning settings and increase the usability of these technologies.[9] Therefore, based on these details, we are developing a student-specific mobile application that will facilitate communication and ease the usage of mobile devices for students and teachers.

## 2.3 Hybrid Recommendation System

Several studies have shown that hybrid recommendation systems can improve the accuracy and diversity of recommendations compared to single-technique approaches. In a study by Adomavicius and Tuzhilin (2005) found that a weighted hybrid approach combining content-based and collaborative filtering techniques outperformed both techniques individually in terms of recommendation accuracy.[10]

Another study by Burke (2007) compared several hybrid approaches and found that a cascade approach combining content-based and collaborative filtering techniques achieved the best performance in terms of recommendation accuracy and diversity.[11]

Similarly in another study by Cheng et al. (2018) proposed a hybrid sequential recommendation approach, which combines sequential recommendation with matrix factorization and content-based recommendation.[12] The study used a recurrent neural network to model the sequential patterns and a deep neural network to learn the weights for the hybrid approach. The study showed that the hybrid approach outperformed both individual techniques in terms of recommendation accuracy and diversity.

Also in a study done by Melville et al. (2002) showed that combining content-based and collaborative filtering techniques can improve recommendation accuracy.[13] They used a neural network to learn the weights for the hybrid approach, and found that the hybrid approach outperformed both individual techniques.

## 3. Related Theory

### 3.1 Clean Architecture

Clean Architecture is the blueprint for a modular system that carefully adheres to the design concept known as separation of concerns. This architecture approach focuses on splitting software into layers to facilitate system development and maintenance. Individual layers can be utilized, developed, and updated separately when layers are well-separated. Clean Architecture is one of the most powerful solutions to build clean apps with independent data layers that multiple teams can work on. The resulting app would also be scalable, readable, testable, and can be easily maintained at any time. This architecture produce systems that are:

- Independent of Frameworks: The availability of a library of feature-rich software is not necessary for the architecture. This enables you to utilize such frameworks as tools rather than forcing your system to fit within their restrictive limits.
- Testable: The business rules can be tested independent of the user interface, database, web server, or any other external aspect.
- Independent of UI: The user interface can be readily changed without affecting the rest of the system. A Web UI, for example, could be replaced with a console UI without impacting the business logic.
- Independent of Database: This allows to replace Oracle or SQL Server with Mongo, BigTable, CouchDB, or another database. Business rules are not tied to a database.

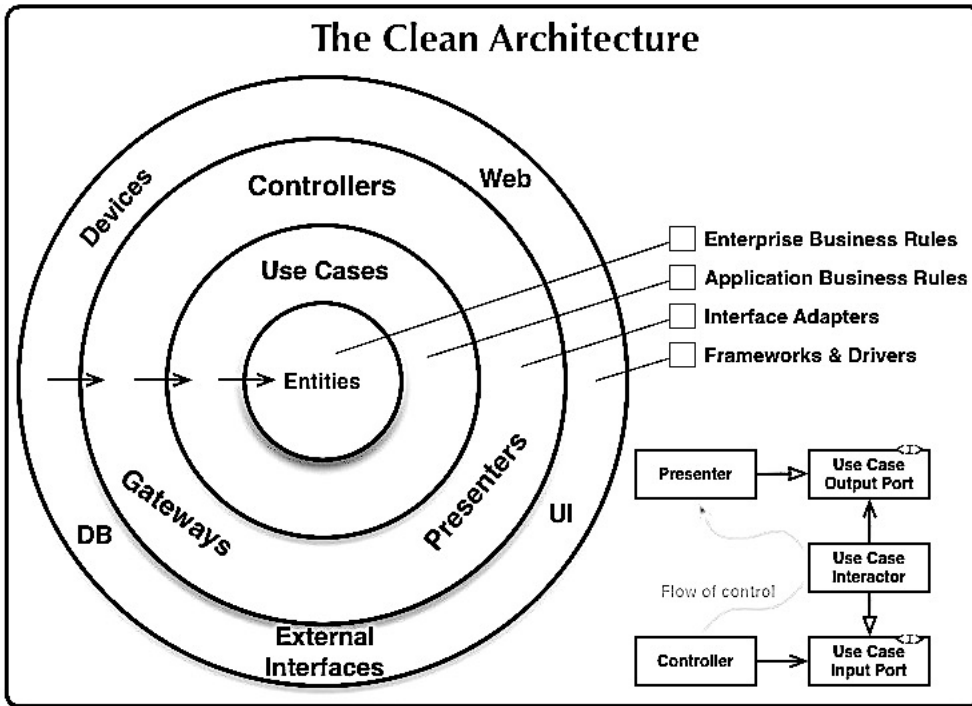


Figure 3.1: Clean Architecture Diagram

## 3.2 Role-based access control

In computer systems security, role-based access control (RBAC)[14][15] or role-based security[16] is an approach to restricting system access to authorized users. It is an approach to implement mandatory access control (MAC) or discretionary access control (DAC).

Role-based access control is a policy-neutral access-control mechanism defined around roles and privileges. The components of RBAC such as role-permissions, user-role and role-role relationships make it simple to perform user assignments. A study by NIST has demonstrated that RBAC addresses many needs of commercial and government organizations.[17] RBAC can be used to facilitate administration of security in large organizations with hundreds of users and thousands of permissions. Although RBAC is different from MAC and DAC access control frameworks, it can enforce these policies without any complication.

### 3.2.1 Design

Within an organization, roles are created for various job functions. The permissions to perform certain operations are assigned to specific roles. Members or staff (or other system users) are assigned particular roles, and through those role assignments acquire the permissions needed to perform particular system functions. Since users are not assigned permissions

directly, but only acquire them through their role (or roles), management of individual user rights becomes a matter of simply assigning appropriate roles to the user's account; this simplifies common operations, such as adding a user, or changing a user's department.

Role based access control interference is a relatively new issue in security applications, where multiple user accounts with dynamic access levels may lead to encryption key instability, allowing an outside user to exploit the weakness for unauthorized access. Key sharing applications within dynamic virtualized environments have shown some success in addressing this problem.[5]

Three primary rules are defined for RBAC:

- Role assignment: A subject can exercise a permission only if the subject has selected or been assigned a role.
- Role authorization: A subject's active role must be authorized for the subject. With rule 1 above, this rule ensures that users can take on only roles for which they are authorized.
- Permission authorization: A subject can exercise a permission only if the permission is authorized for the subject's active role. With rules 1 and 2, this rule ensures that users can exercise only permissions for which they are authorized.

### **3.2.2 Standardized levels**

The NIST/ANSI/INCITS RBAC standard (2004) recognizes three levels of RBAC: core RBAC hierarchical RBAC, which adds support for inheritance between roles constrained RBAC, which adds separation of duties

### **3.2.3 Attribute-based access control**

Attribute-based access control or ABAC is a model which evolves from RBAC to consider additional attributes in addition to roles and groups. In ABAC, it is possible to use attributes of:

the user e.g. citizenship, clearance, the resource e.g. classification, department, owner, the action, and the context e.g. time, location, IP.

ABAC is policy-based in the sense that it uses policies rather than static permissions to define what is allowed or what is not allowed.



### **3.3 Bidirectional Sync and TimeStamp Based Conflict Resolution**

Bidirectional sync and timestamp-based conflict resolution are closely related concepts in the context of data synchronization between multiple devices or systems.

Maintaining two or more devices or systems in sync with one another in order for any changes made in one system to be reflected in the other system and vice versa is known as "bidirectional sync." This can be done using a variety of techniques, including data replication, file syncing, and cloud syncing.

However, conflicts can occur when numerous devices or systems are simultaneously making changes to the same data, which could result in data loss or corruption. Timestamp-based conflict resolution can be used in this situation.

The approach of timestamp-based conflict resolution is used to settle disagreements that could occur during a bidirectional sync operation. It entails giving each device or system a special timestamp and using those timestamps to figure out which change was made first. The device with the earlier timestamp receives priority when making changes, and those changes are applied to the other device or system.

For example, if you modify a document while it is saved on both your computer and phone without synchronizing them, a conflict will occur when you attempt to sync the devices. The synchronization process will prioritize the changes made on the device with the earlier timestamp after using timestamps to establish which version of the document was created first.

Overall, bidirectional synchronization and timestamp-based conflict resolution are crucial techniques for guaranteeing that data is accurate and current across many systems or devices. These methods can be used to avert conflicts and stop data loss or corruption.

### **3.4 Collaborative Filtering**

collaborative filtering is a popular approach used in recommendation systems that focuses on finding similarities between users in order to make personalized recommendations. By analyzing past behavior and preferences of a user and comparing it to other users, collaborative filtering can provide accurate and relevant recommendations. However, it also has its limitations, such as the cold-start problem when new users join the system, and the potential for recommendations to be biased towards popular items. Another challenge is the popularity bias, which occurs when the system recommends only popular items and overlooks niche

or lesser-known items that the user might also enjoy. Despite these challenges, collaborative filtering remains a powerful tool for providing personalized recommendations to users in a variety of industries. There are two main types of collaborative filtering: user-based and item-based.

- In user-based collaborative filtering, the system looks for users who have rated items similarly to the current user and then recommends items that those similar users have liked. For example, if user A and user B have both rated several movies highly, the system might recommend other movies that user B has liked but user A hasn't seen yet.
- In item-based collaborative filtering, the system looks for items that are similar to the ones the user has already rated highly and recommends other similar items. For example, if a user has rated several romantic comedies highly, the system might recommend other romantic comedies that have similar themes or actors.

Overall, collaborative filtering is a powerful tool for providing personalized recommendations in a variety of industries, but it requires careful consideration of its strengths and limitations in order to be used effectively.

## 3.5 Content Based Filtering

Content-based filtering is another popular approach used in recommendation systems. Unlike collaborative filtering, which focuses on similarities between users, content-based filtering focuses on similarities between items. In other words, it recommends items that are similar to items the user has liked in the past.

To do this, content-based filtering looks at the attributes of the items themselves. For example, in a movie recommendation system, content-based filtering might analyze factors such as the genre, director, actors, plot, and language of the movies the user has previously enjoyed. It can then recommend other movies that have similar attributes. For example, if a user has enjoyed several action movies starring Tom Cruise, the system might recommend other action movies starring Tom Cruise.

Content-based filtering has several advantages over collaborative filtering. For one thing, it doesn't suffer from the cold-start problem that collaborative filtering can experience with new users. Additionally, it can recommend niche or lesser-known items that might not be as popular as the items recommended by collaborative filtering.

However, content-based filtering also has some limitations. One challenge is that it can only recommend items that are similar to items the user has already liked, which can result

in a narrower range of recommendations. Additionally, it can be more difficult to define and quantify the attributes of items in some domains than in others, making content-based filtering less effective in those domains.

Overall, content-based filtering is a powerful tool for providing personalized recommendations based on the attributes of items themselves. By analyzing the attributes of items that users have enjoyed in the past, content-based filtering can provide highly relevant and personalized recommendations. However, it also has its limitations and must be carefully considered in the context of the particular domain and user base it is being applied to.

### 3.6 TF-IDF Vectorizer

TF-IDF vectorizer is a common feature extraction technique used in natural language processing and information retrieval tasks. It stands for "Term Frequency-Inverse Document Frequency". It is a numerical statistic that reflects how important a word is to a document in a collection of documents or corpus. The goal of TF-IDF is to highlight words that are important and distinct in a document, while downplaying words that are common across many documents.

TF-IDF can be calculated as

$$TF - IDF = TF * IDF \tag{3.1}$$

where; At first TF is calculated as :

$$TF = \text{number of times term appears in the document} / \text{number of terms in the document} \tag{3.2}$$

Similarly IDF is calculated as:

$$IDF = \text{number of documents in the corpus} / \text{number of documents in the corpus contain the term} \tag{3.3}$$

Once the TF-IDF vectorizer has computed the TF-IDF scores for each word in each document, it creates a vector for each document that represents the relative importance of each word. These vectors can be used in various natural language processing tasks, such as document classification, topic modeling, and recommendation systems. Overall, the TF-IDF vectorizer is a powerful tool for extracting features from text data and highlighting important words in a document.

The TF-IDF vectorizer is also used in recommendation systems, where the goal is to suggest items (e.g. books, movies, products) to users based on their preferences. In this case, the

TF-IDF vectors for each item can be used to calculate the similarity between items, based on the words that they have in common. This can help to make recommendations that are tailored to the specific interests of each user.

Overall, the TF-IDF vectorizer is a powerful and versatile tool for natural language processing and information retrieval. It is widely used in both research and industry, and has proven to be effective in a wide range of applications.

### 3.7 Cosine Similarity

Cosine similarity is a measure of similarity between two vectors that is commonly used in information retrieval and machine learning tasks. It measures the cosine of the angle between the two vectors, where a value of 1 indicates that the two vectors are identical, and a value of 0 indicates that they are orthogonal (i.e. not similar at all). Cosine similarity is widely used in text analysis tasks, where the goal is to measure the similarity between two documents or pieces of text.

we can calculate the cosine similarity using the following formula:

$$\text{cosine similarity}(x, y) = (x \cdot y) / (|x| * |y|) \tag{3.4}$$

where  $x$  and  $y$  are the two vectors being compared, " $\cdot$ " represents the dot product of the two vectors, and " $|x|$ " and " $|y|$ " represent the magnitudes (i.e. lengths) of the two vectors.

The resulting value ranges from -1 to 1, where a value of 1 indicates that the two vectors are identical, a value of 0 indicates that they are completely dissimilar, and a value of -1 indicates that they are completely dissimilar but in opposite directions.

In natural language processing, cosine similarity is often used to compare the similarity of TF-IDF vectors between documents. The cosine similarity between two vectors is calculated as the dot product of the two vectors, divided by the product of their magnitudes. The resulting value is a measure of how similar the two vectors are in terms of the words they contain and their relative importance.

Cosine similarity has several advantages over other similarity measures. One advantage is that it is scale-invariant, meaning that it is not affected by the length or magnitude of the vectors being compared. Another advantage is that it is well-suited to high-dimensional spaces, such as those encountered in natural language processing, where the number of dimensions (i.e. words) can be very large.

Overall, cosine similarity is a powerful and widely-used tool in natural language processing and machine learning. It provides a simple and intuitive way to measure the similarity between two vectors, and has proven to be effective in a wide range of applications.

## **3.8 CMS**

A Content Management System (CMS) is a software application that allows users to create, manage, and publish digital content, typically for websites. CMSs are designed to provide a user-friendly interface that enables non-technical users to easily create and manage digital content without requiring specialized technical knowledge.

A CMS typically provides a wide range of features, including content creation tools, editing tools, media management tools, and user management tools. Users can create and edit content using a WYSIWYG (What You See Is What You Get) editor, which allows them to easily format text, add images and videos, and insert other multimedia content. CMSs also typically include built-in SEO tools, which help users optimize their content for search engines.

One of the key advantages of a CMS is that it allows multiple users to collaborate on creating and managing content. Users can be assigned different roles and permissions, which enables them to access only the parts of the CMS that are relevant to their role. This can be particularly useful for larger organizations, where multiple teams or departments may be responsible for creating and managing different types of content.

Overall, CMSs are a powerful tool for creating, managing, and publishing digital content. They can help organizations streamline their content creation and management processes, while also providing a user-friendly interface that makes it easy for non-technical users to create and manage content. With a wide range of features and capabilities, CMSs are an essential tool for any organization that wants to effectively manage their digital content.

# 4. Methodology

## 4.1 Preliminary

In this section, the project conception and architecture development is discussed.

## 4.2 Question Prediction System

The question prediction system uses OCR and topic extraction to extract questions from PDF files and predict similar questions using a combination of TF-IDF vectorization and cosine similarity.

The pipeline for Question Prediction System is:

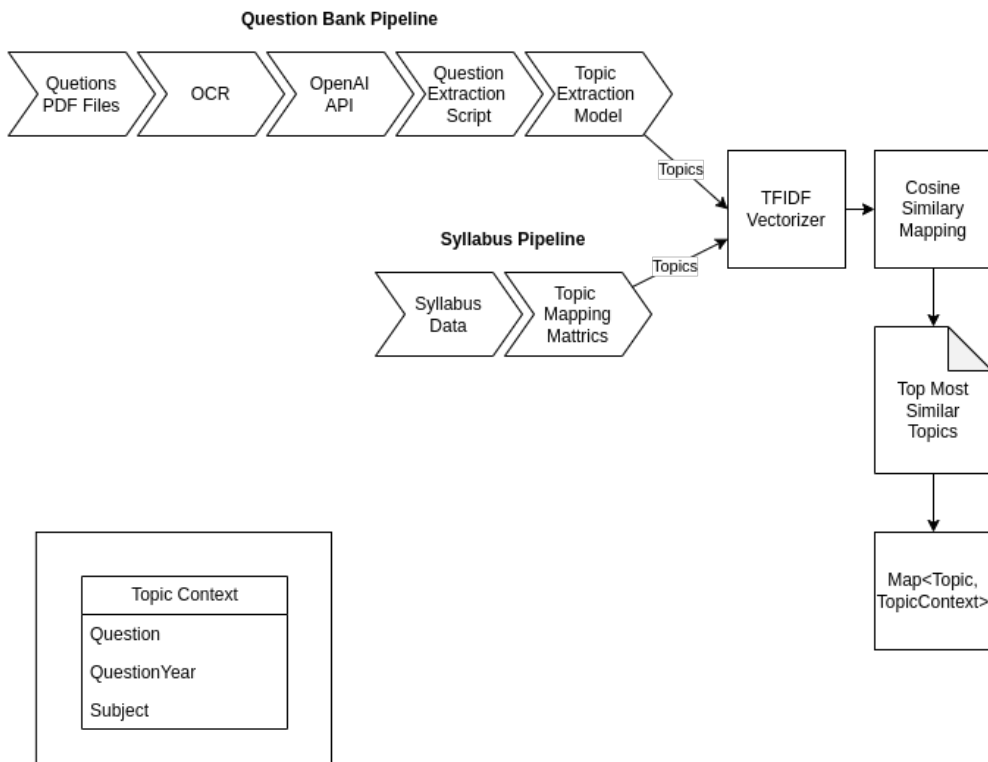


Figure 4.1: Question Prediction Pipeline

### 4.2.1 Data Collection

To build a reliable question prediction system, it is essential to have a comprehensive and diverse set of questions in the dataset. The question bank PDF used in this project was

obtained from various public domain sources. It is important to note that the quality of the PDF files can have a significant impact on the accuracy of the OCR software used to extract the text data. Therefore, the PDF files used in this project were carefully selected to ensure that they are of high quality, with legible text for the OCR software.

### 4.2.2 Data Preprocessing

It is important to preprocess the textual data to remove any inconsistencies, errors, or irrelevant information that may affect the quality of the data. During the preprocessing stage, various techniques were applied to ensure that all textual discrepancies were removed. By removing unnecessary elements from the text data, the preprocessing stage helped to improve the accuracy and efficiency of the question prediction system, ultimately leading to more accurate and reliable results.

### 4.2.3 OCR

The process of OCR using Pytesseract involved several steps. First, the PDF files containing the questions were converted into images using a PDF-to-image converter. Then, the images were passed through the Pytesseract OCR engine, which scanned the images and extracted the text data. The resulting text data was then saved in a machine-readable format that could be further processed and analyzed by the question prediction system.

#### OCR Process Flow

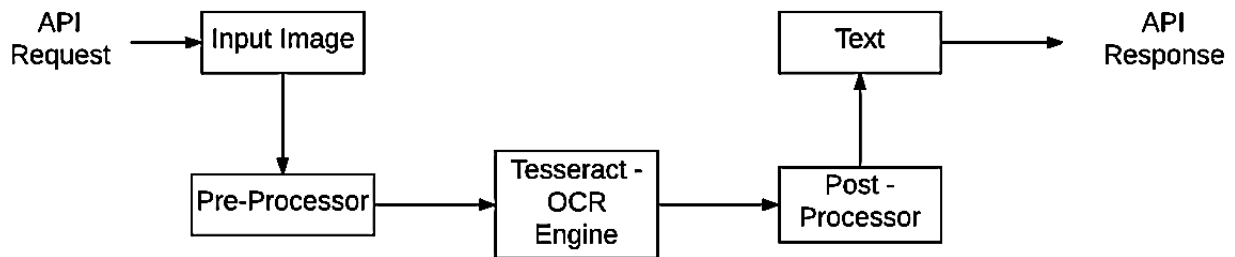


Figure 4.2: OCR Flowchart

#### 4.2.4 Topic Extraction

The topic extraction algorithm is trained on a corpus of documents to learn the underlying themes or topics. The topics has been extracted from the syllabus provided by the IOE.

#### 4.2.5 Model Training

TF-IDF is used to convert the preprocessed text data into a numerical vector representation that could be used to calculate the similarity between questions. Once the text data has been vectorized using TF-IDF, cosine similarity was used to calculate the similarity between questions. Cosine similarity is a metric used to measure the similarity between two vectors by calculating the cosine of the angle between them. In the context of the question prediction system, cosine similarity was used to calculate the similarity between the vector representations of two questions.

To calculate the cosine similarity between two questions, the TF-IDF vectors for each question were calculated and then the cosine similarity between the vectors was calculated. This provided a measure of the similarity between the two questions, based on the underlying themes or concepts present in the questions. the use of TF-IDF and cosine similarity in the question prediction system helped to identify the most similar questions to a user's input based on the content of the questions.



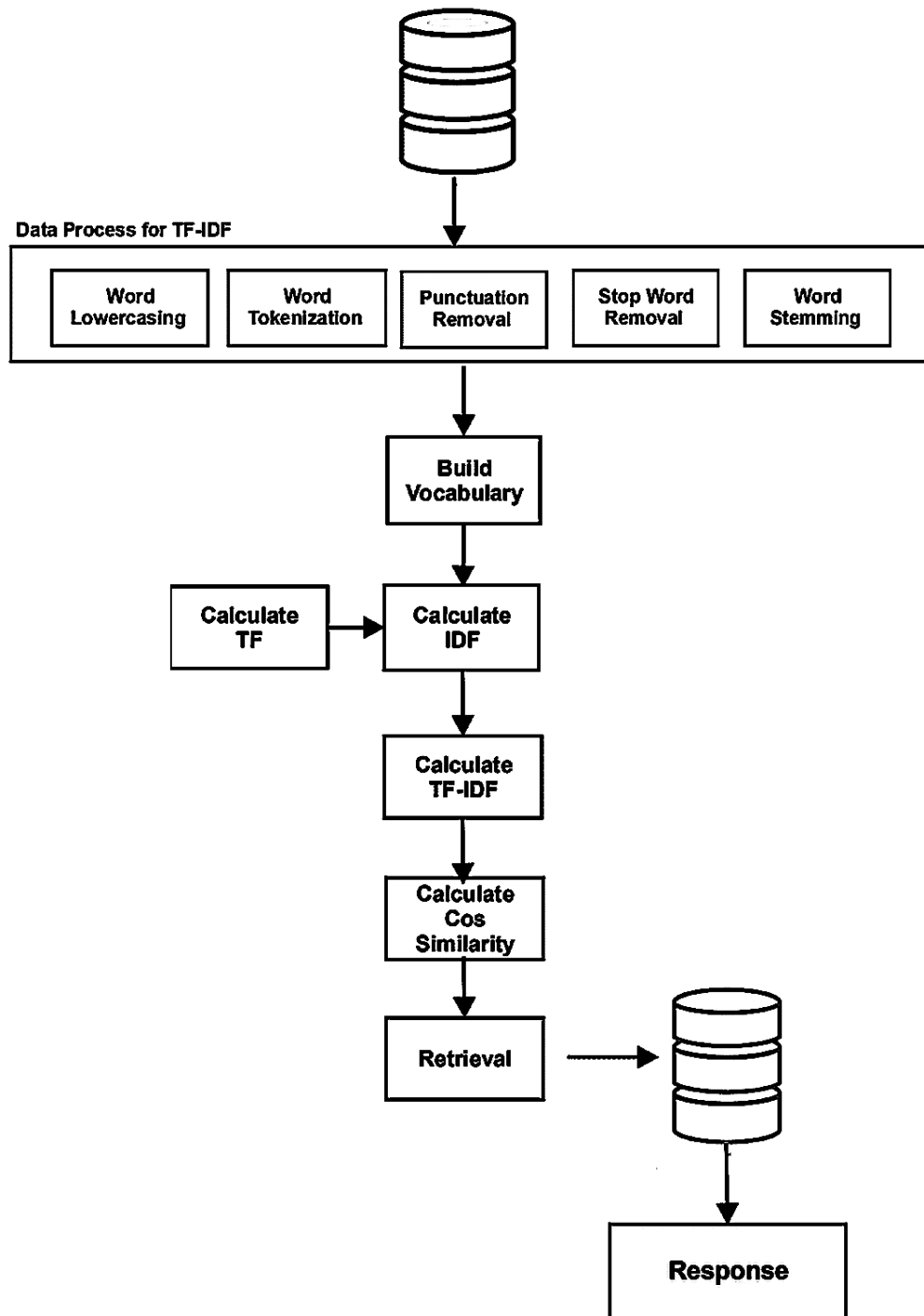


Figure 4.3: TFIDF and Cosine Similarity Flowchart

### 4.2.6 Deployment

After the Question Prediction System was evaluated and fine-tuned, it was deployed in the production environment.

## 4.3 Hybrid Recommendation System

A hybrid recommendation system is a type of recommendation system that combines multiple algorithms to generate personalized recommendations for users. Hybrid systems are designed to overcome the limitations of individual recommendation algorithms by combining their strengths and compensating for their weaknesses. There are several approaches to building hybrid recommendation systems, but the most common approach is to combine collaborative filtering and content-based filtering algorithms. There is a more detailed explanation of each step in the methodology for building a hybrid recommendation system below:

### 4.3.1 Data Collection

The first step in building a hybrid recommendation system is to collect data on user preferences and behavior so we collected data based on preference and behavior of user. The data were diverse and representative of the user population to ensure that the recommendations are useful to a wide range of users.

### 4.3.2 Data preprocessing

After collecting the data, it was preprocessed to prepare it for modeling. This includes cleaning and filtering the data, normalizing the data, and feature engineering. Feature engineering involves extracting relevant features from the raw data, such as keywords or tags for content-based filtering or user demographics for demographic filtering.

### 4.3.3 Model selection and training

Once the data was preprocessed, appropriate recommendation algorithms need to be selected for the hybrid system. The choice of algorithms will depend on the nature of the data and the problem being solved. Collaborative filtering is a popular approach for hybrid systems, but content-based filtering, demographic filtering, and other approaches can also be used so both were chosen and trained on preprocessed data.

### 4.3.4 Evaluation

Once the hybrid recommendation system has been developed, it needs to be evaluated to determine its effectiveness. Common evaluation metrics include Cumulative Heat Rate and Mean Absolute Error were used. During the training of model we used Mean Absolute Error while after deployment we used Cumulative Heat Rate for accuracy of model.

### 4.3.5 Deployment

After the hybrid recommendation system was evaluated and fine-tuned, it was deployed in a production environment.

## 4.4 Role-Based Access Control

With the help of 'djangoestframework-role-filters' module, we build two different types of roles where different access on actions like 'create', 'list', 'retrieve', 'update', and 'partial\_update' will be provided based on the types of roles. We basically used two types of roles in our system i.e. creator role and registered user role. The creator Role can be used by teachers to update attendance on the server which will then be displayed on the mobile app as a notification to the registered users i.e. students. Also through Web CMS content can be provided by teachers which will be notified to the registered users.

## 4.5 Notification System

Firebase Cloud Messaging (FCM) is a cloud-based messaging service provided by Firebase that allows us to send notifications and messages to your app users on various platforms, including Android, iOS, and the web. With Firebase Cloud Messaging, we can send customized notifications with a variety of options, including sound, vibration, and icons to specific devices, user segments, or entire user bases.

We also used Firebase Cloud Messaging for push notifications where we first register users for notifications. This involves requesting permission from the user to receive notifications and then registering their device token with FCM. Once they were registered for notifications, messages were sent using the Firebase Cloud Messaging API. Depending on the requirements, notifications could be sent to individual devices or to groups of devices. Finally, for those notifications received on a user's device, the necessary code was written on our app to handle the notification and display it to the user.

Overall, Firebase Cloud Messaging is a powerful and reliable tool for sending notifications to your app users. With its scalability, customization options, and analytics tools, Firebase Cloud Messaging can help you build a robust notification system that keeps your users engaged and informed.

## 4.6 Test Driven Development

Test-driven development (TDD) is a software development approach where developers write automated test cases before they write the actual code. This approach is designed to ensure that the code meets the requirements of the application and behaves as expected.

The TDD process typically involves three steps:

- Write a failing test: In this step, the developer writes a test case that checks for the desired behavior of the code. Since there is no code yet, the test case will fail.
- Write the minimum code required to pass the test: In this step, the developer writes the minimum amount of code required to make the test case pass.
- Refactor the code: In this step, the developer improves the code's design and structure without changing its behavior. The developer can then run the test cases again to ensure that the code still passes.

By following this process, we ensure that the code we write meets the requirements of the application and that any changes we make to the code do not introduce new errors. So, we used Pytest as our testing framework. Pytest is a testing framework for Python. It is a popular tool for writing and running tests for Python applications, libraries, and frameworks. It allows writing tests in a variety of styles, including the traditional "assert" statements and the newer "expect" statements. It also provides a number of advanced features, such as fixtures, parameterization, and test discovery, which make it easy to write and run complex tests.

We used it in our backend since we used Django for our backend. We used to write tests before writing the actual code so that code will be error-free. We used various features of pytest as mentioned above like fixtures, mock, etc which made it easy for writing test cases. Then we used coverage.py on those test cases to measure the coverage of those test cases on the application code. Based on the result of coverage.py necessary changes were made to the application code to produce 100% coverage report. Finally, this code was pushed to a remote server on GitHub where again CircleCI checks the test cases.

Overall, Pytest is a powerful and flexible tool for testing Python applications. Its ease of use, advanced features, and flexibility make it a popular choice among Python developers for testing their code.

## 4.7 AWS Deployment

AWS (Amazon Web Services) has been used as a cloud computing platform to deploy and host the project. With AWS, the project can be deployed in a scalable, reliable, and cost-effective way, providing a secure environment for the project's data and services. AWS offers a wide range of services and tools to help deploy, manage, and scale applications and infrastructure. By using AWS, the project can take advantage of features like auto-scaling, load balancing, and managed databases, which help ensure high availability and performance.

The AWS architecture of the project is given as:

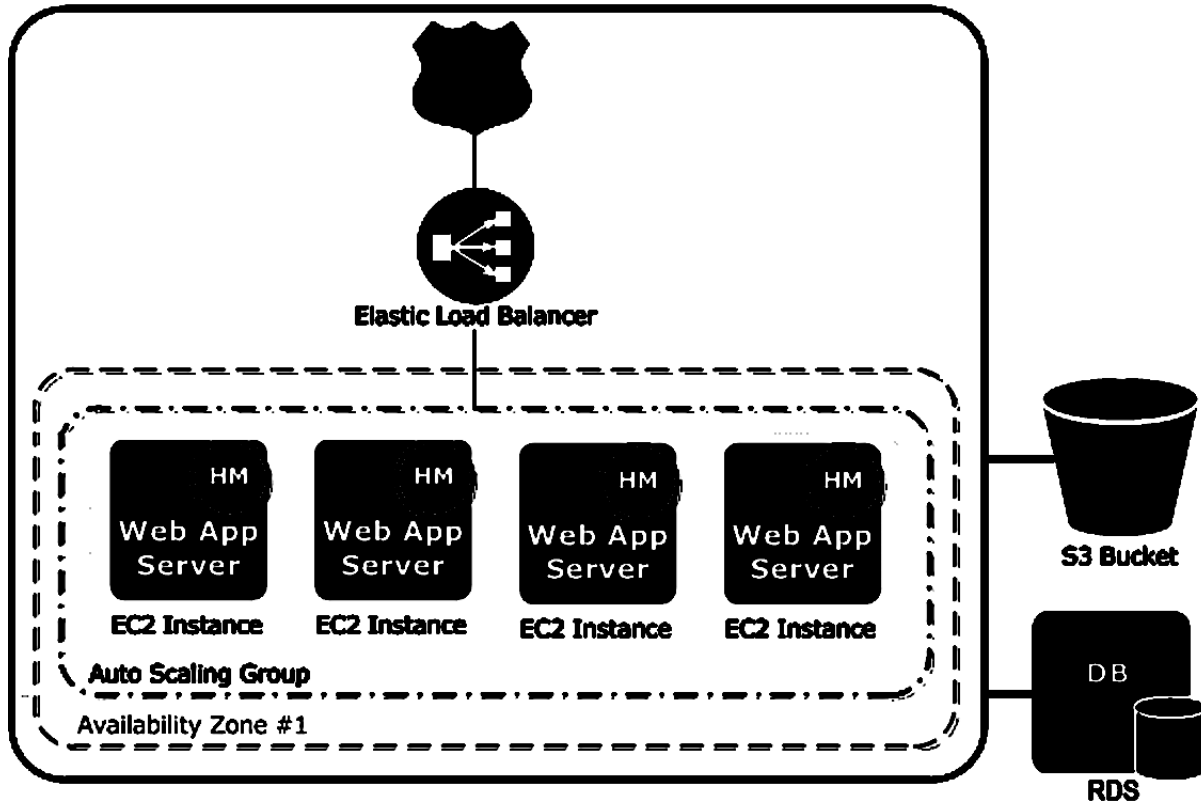


Figure 4.4: AWS Architecture

The following steps were taken to set up the AWS architecture with the components mentioned above:

- **EC2 Instance Configuration:** The EC2 instances host the application and can access the RDS database and S3 bucket for data storage and retrieval. The instances are launched and managed by Elastic Beanstalk, which automatically scales the number of instances based on traffic.
- **RDS Configuration:** The RDS database instance stores application data and is accessed by the application running on the Elastic Beanstalk environment. RDS provides a managed database service, including automated backups, monitoring, and security features.
- **S3 Bucket Configuration:** An S3 bucket was created to store the static content like images, videos, and other media files. The content was uploaded to the bucket and attached to the EC2 instance.
- **ELB Configuration:** An Elastic Load Balancer was created and configured to distribute

incoming traffic across the EC2 instances. The security group rules were configured to allow traffic only from the internet.

- **Route 53 Configuration:** A Route 53 hosted zone was created for the domain name, and DNS records were created to route traffic to the Elastic Load Balancer.
- **Elastic Beanstalk Configuration:** AWS Elastic Beanstalk was used to deploy the web application code and manage the underlying infrastructure.

## 4.8 Development

The development of this project has been done with the CI/CD pipeline that has been established using GitHub, CircleCI, AWS CodePipeline, CodeBuild, and CodeDeploy. The pipeline is designed to automate the software delivery process, reduce the time required to deploy new features and bug fixes, and improve overall software quality.

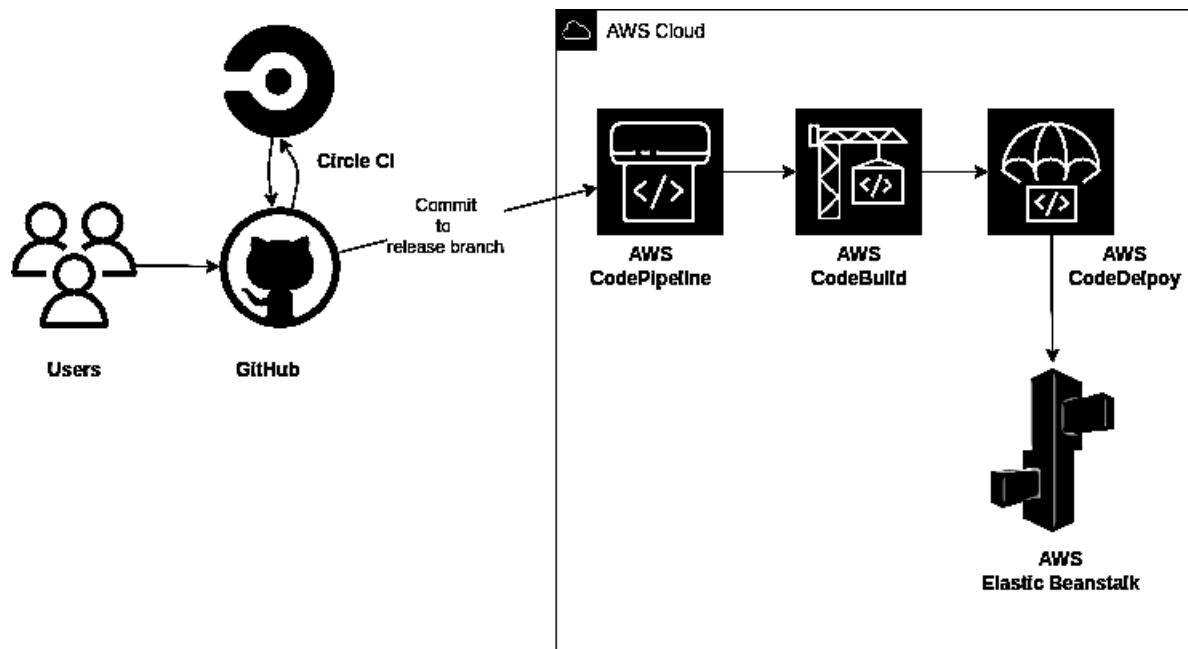


Figure 4.5: CI/CD Workflow

### 4.8.1 GitHub

A GitHub repository has been set up to store the source code. The code is the foundation of the pipeline, and it is regularly updated to reflect new changes.

### 4.8.2 CircleCI

CircleCI is linked to the GitHub account. The project has been configured to run appropriate tests and build commands. CircleCI is triggered automatically each time changes are made

to the repository, ensuring that the code is thoroughly tested and built before deployment.

### **4.8.3 AWS CodePipeline**

A pipeline has been created in AWS CodePipeline. GitHub has been designated as the source provider, with the desired branch selected for building. CircleCI has been selected as the build provider, and CodeDeploy as the deployment provider.

### **4.8.4 AWS CodeBuild**

A project has been created in AWS CodeBuild. The project has been configured to use the appropriate build image and build commands. CodeBuild is utilized by AWS CodePipeline to build the code.

### **4.8.5 AWS CodeDeploy**

A deployment group has been created in AWS CodeDeploy. It has been configured to use the appropriate deployment settings. CodeDeploy is utilized by AWS CodePipeline to deploy the code.

### **4.8.6 Testing the pipeline**

The pipeline has been rigorously tested by pushing changes to the GitHub repository. Upon trigger, CircleCI runs the appropriate tests and builds the code. AWS CodePipeline subsequently triggers CodeBuild to compile the artifact and CodeDeploy to deploy it. The pipeline's progress has been monitored through the AWS Console.

### **4.8.7 Monitoring and debugging**

The pipeline's progress has been monitored using logs and reports provided by CircleCI, CodePipeline, CodeBuild, and CodeDeploy. The analysis tools facilitate the prompt identification and resolution of any issues that may arise in the pipeline.

## **4.9 Tools and Technology Used**

The application will be developed in client-server architecture and main focus in client side will be in mobile application. Having said that tools we will be using will be:

- Front-end Framework: Flutter , React
- Back-end Framework: Django Rest Framework
- Database: PostgreSQL
- IDE: Android Studio, XCode, VSCode

- CI Tool: GitHub Action, Circle CI
- Version Control: GitHub
- Prototyping: Figma, Draw.io
- Project Management: GitHub Project
- Deployment: AWS
- Testing Framework: Pytest, Coverage.py



# 5. System Design

## 5.1 Project Architecture

Project Architecture consist of simple three-tier architecture as show in Figure 5.1. The project will contain two major sections; backend and frontend. The database for the system will be stored in server side while client side will have caching functionality for single client related data. The backend storage will be MongoDB and in frontend will be using SQLite storage.

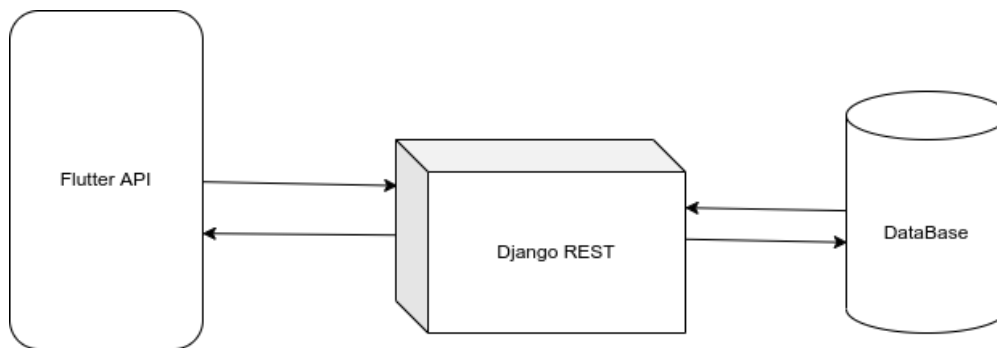


Figure 5.1: System Architecture Diagram

### 5.1.1 System Design

The project is broken down into multiple subsystem, which provides individual features independent of others. Of all the portal creation system being the center of all. All system included in this project are:

- Portal Creation System
- Content System
- Routine System
- Syllabus System
- Search System
- Library System
- Personalization System

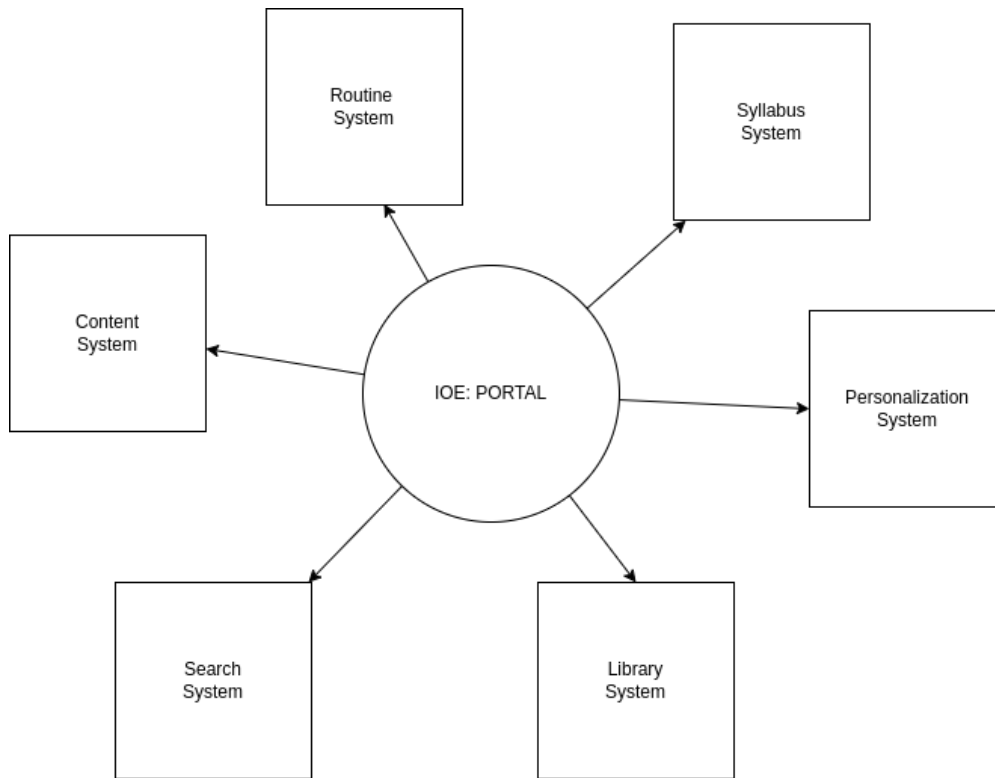


Figure 5.2: System Diagram

Among all system, the portal system is the glue as it is used to standardized the API provided by other systems. It provides three standard interface,

- User Interface For Create Mode
- User Interface For Edit Mode
- User Interface For View Mode

The visibility of these user interfaces in the given portal is again controlled by the privilege a member of that portal has. There are four type of privilege for a member subscribed to a portal, i.e. [CREATE — EDIT — VIEW — DENY]. And these privilege level dictates what interaction a user can have from a given portal.

5.1.2 Usecase Diagram

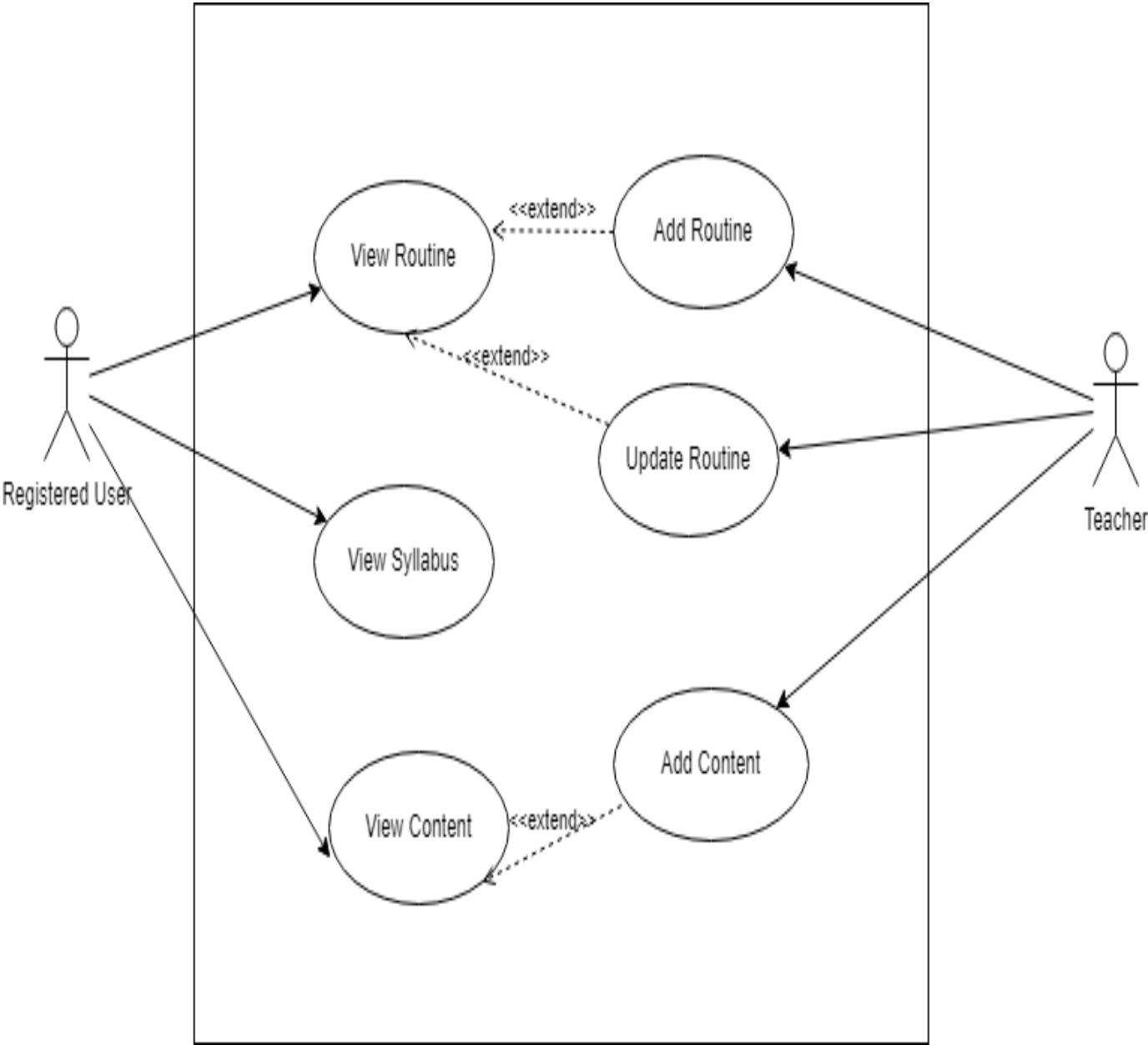


Figure 5.3: System Usecase Diagram

## 5.2 Database

We are using Amazon RDS for PostgreSQL as our database which is a relational database service in Microsoft cloud. Amazon RDS (Relational Database Service) for PostgreSQL is a managed database service offered by Amazon Web Services (AWS) that makes it easy to set up, operate, and scale a PostgreSQL database in the cloud. With Amazon RDS for PostgreSQL, you can run your PostgreSQL database instances in a highly available and scalable environment with automatic backups, software patching, and monitoring. Some of the key features of Amazon RDS for PostgreSQL include:

- **Automatic backups:** Amazon RDS automatically takes daily backups of your database, which can be easily restored with a few clicks.
- **Scalability:** You can easily scale up or down your database instance based on your application's needs.
- **High availability:** Amazon RDS for PostgreSQL provides a highly available environment with automatic failover to a standby instance in case of any issues with the primary instance.
- **Security:** Amazon RDS for PostgreSQL includes multiple security features such as encryption at rest, network isolation, and secure access control.
- **Monitoring and metrics:** Amazon RDS provides detailed monitoring and metrics for your PostgreSQL database instance, allowing you to identify and troubleshoot issues quickly.

Overall, it provides services like built-in high availability with no additional setup, configuration or extra cost along with many other features that suits our application.

ER diagram of our system is shown below which illustrates different entities which we will make throughout the project. Here the ER diagrams for User System, Content System and Miscellaneous Systems is shown

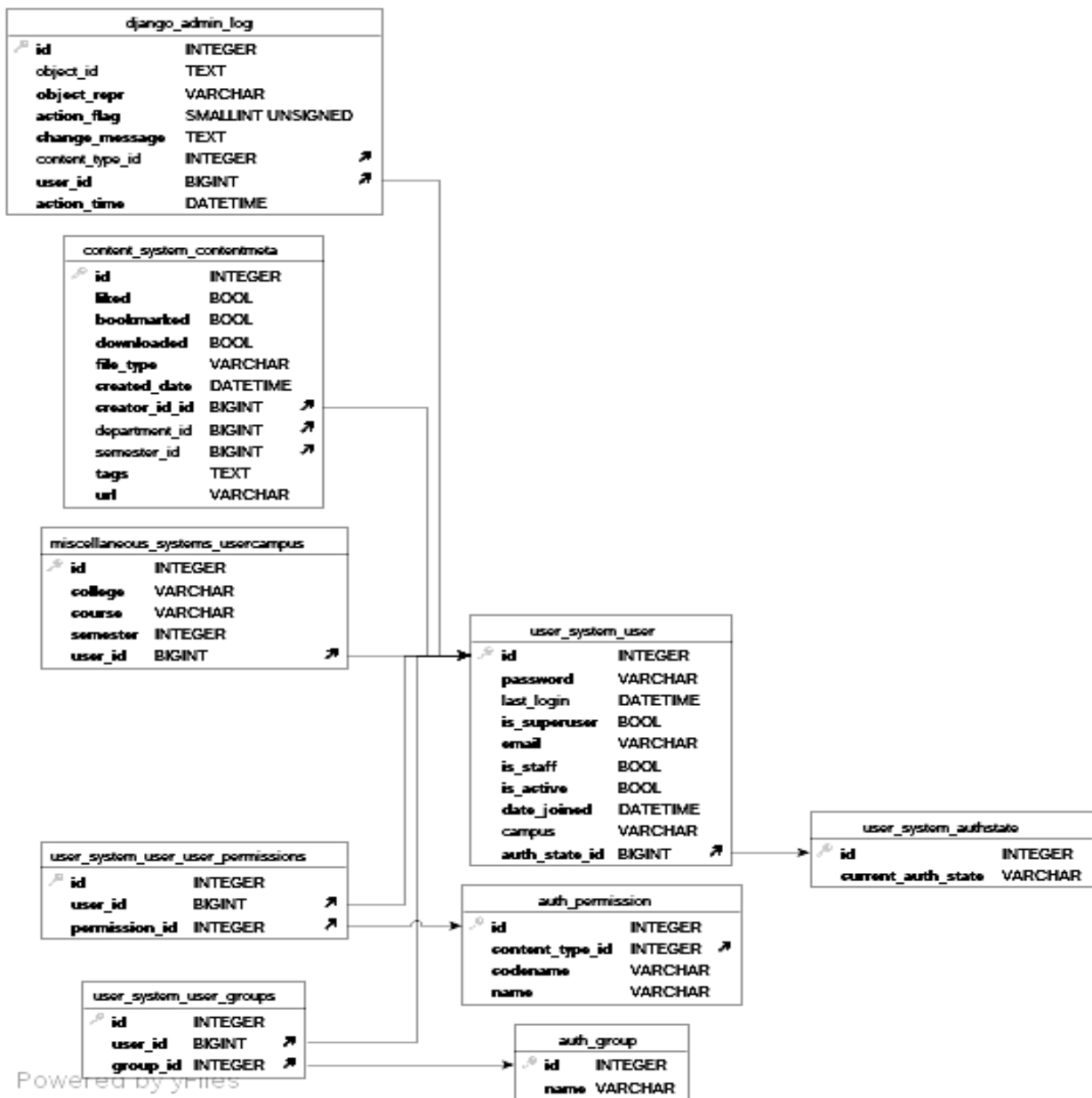


Figure 5.4: User System ER Diagram

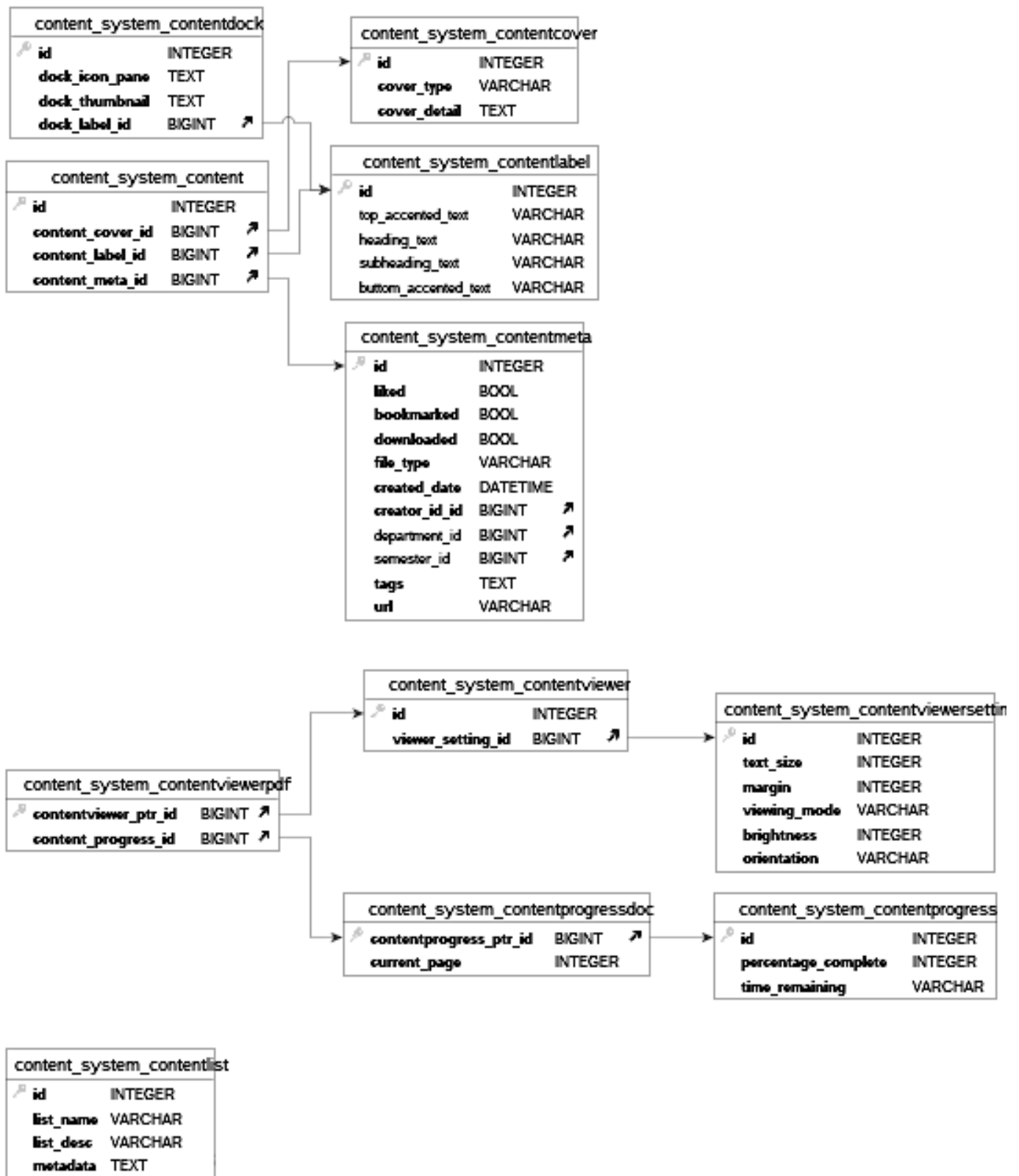


Figure 5.5: Content System ER Diagram

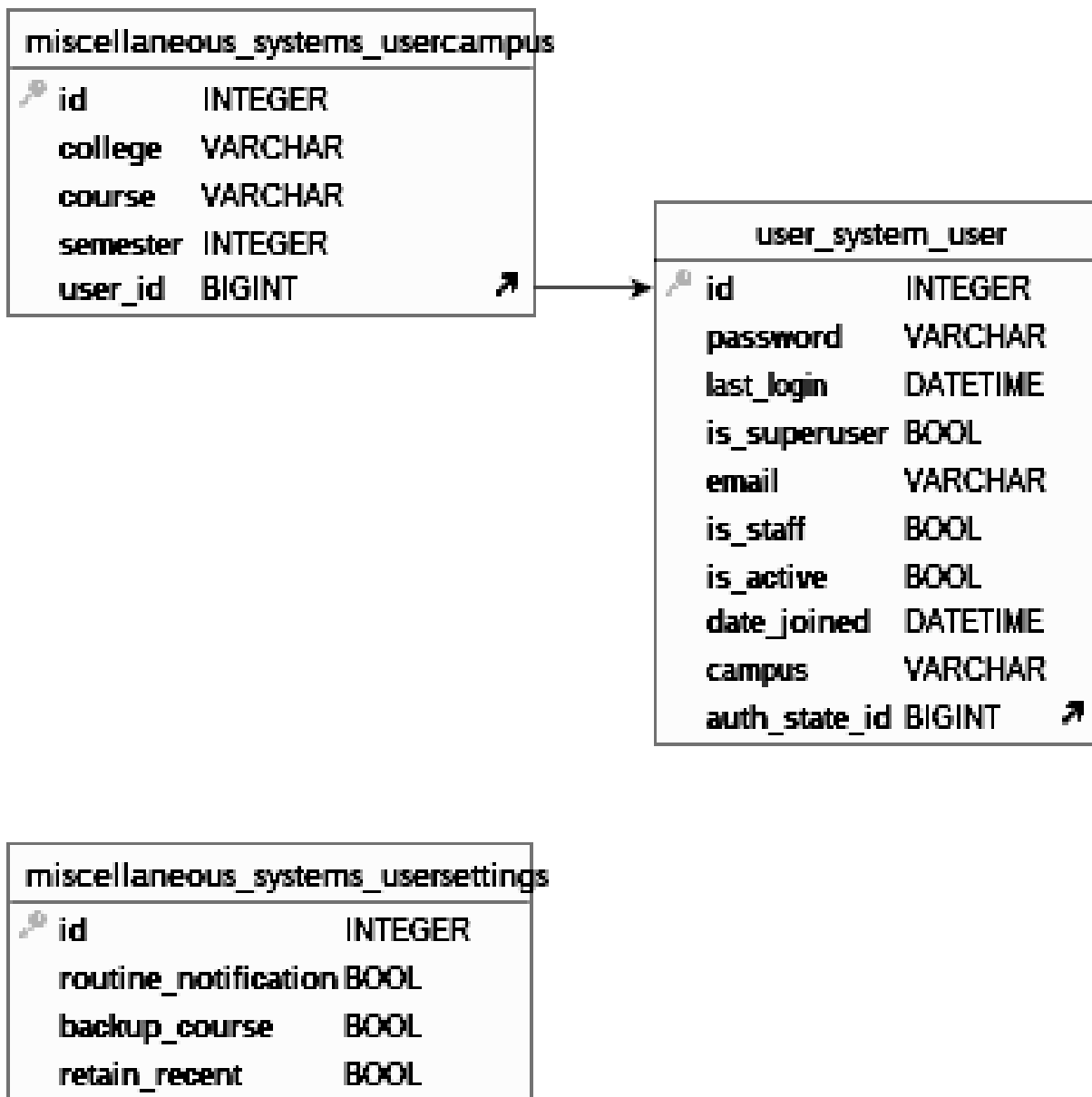


Figure 5.6: Miscellaneous System ER Diagram

### 5.3 Project Development

The project is developed in sprints, where one use case is dealt at a time. In one sprint, backlog stacking, object oriented analysis and design, and development was completed. Before developing the system UI an elaborate UI flow diagram was created as shown in Figure 5.7

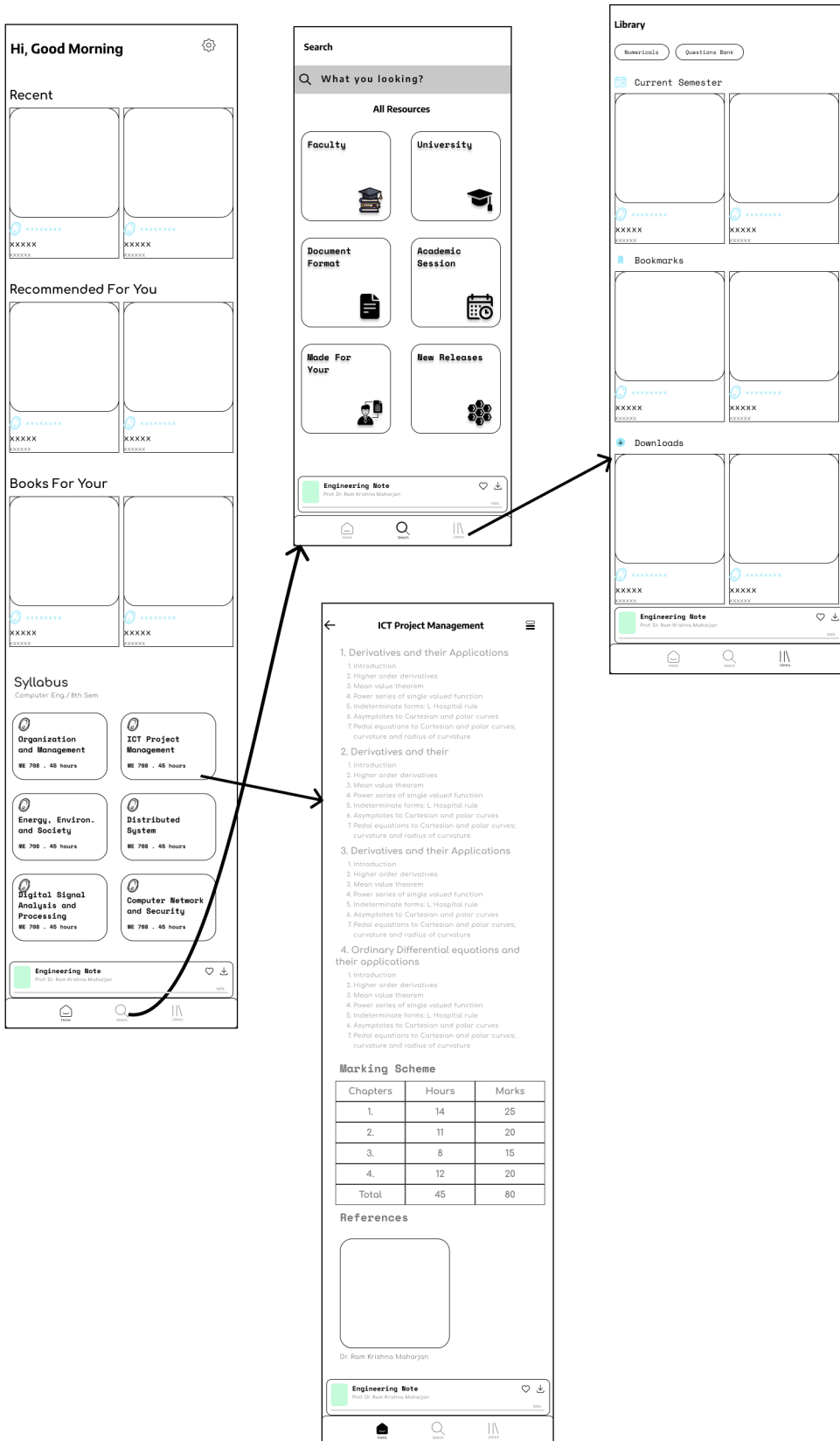


Figure 5.7: Sample UI Flow Diagram



## 5.4 Backend

### 5.4.1 JWT Token Authentication

The initial step in using the application is to register and then log in. The safety of the user is prioritised as a primary concern in this mobile application. Therefore, JWT Token Authentication has been implemented within the application. It allows the server to authenticate the client by exchanging a compact and digitally signed token, which contains user information and access privileges, between the client and the server. Here's how it works:

- The client sends a request to the server to log in.
- The server authenticates the user and creates a JWT token containing user information and access privileges.
- The server sends the JWT token back to the client in the response.
- The client stores the JWT token in local storage or a cookie.
- For subsequent requests to protected resources, the client includes the JWT token in the request header.
- The server validates the JWT token by verifying its signature and decoding its contents.
- If the token is valid, the server grants access to the requested resource; otherwise, it returns an error.

Here some of the Sequence Flow Diagram of the Authentication System are shown:

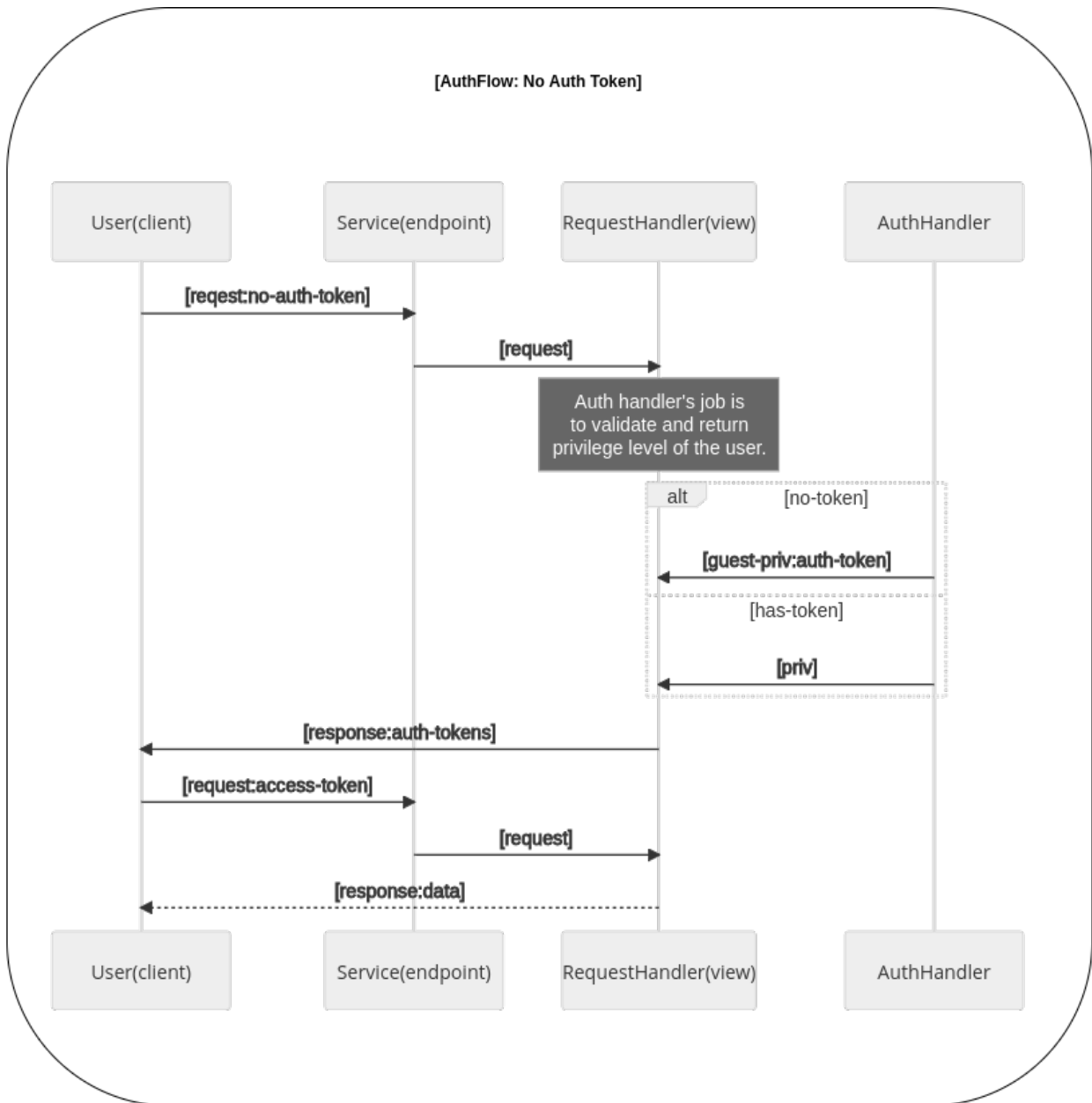


Figure 5.8: No Auth Token Sequence Diagram

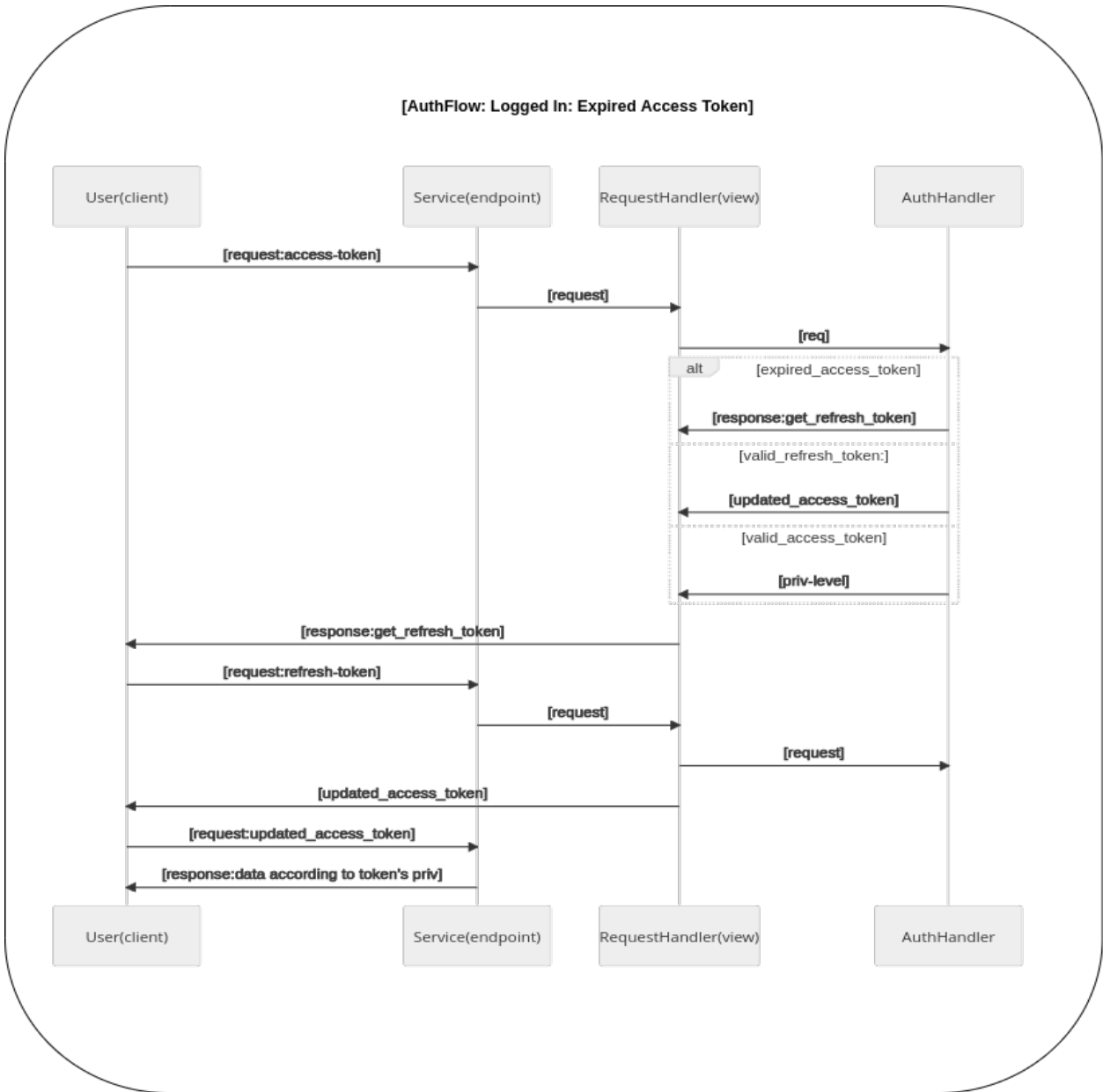


Figure 5.9: Logged In: Expired Access Token Sequence Diagram

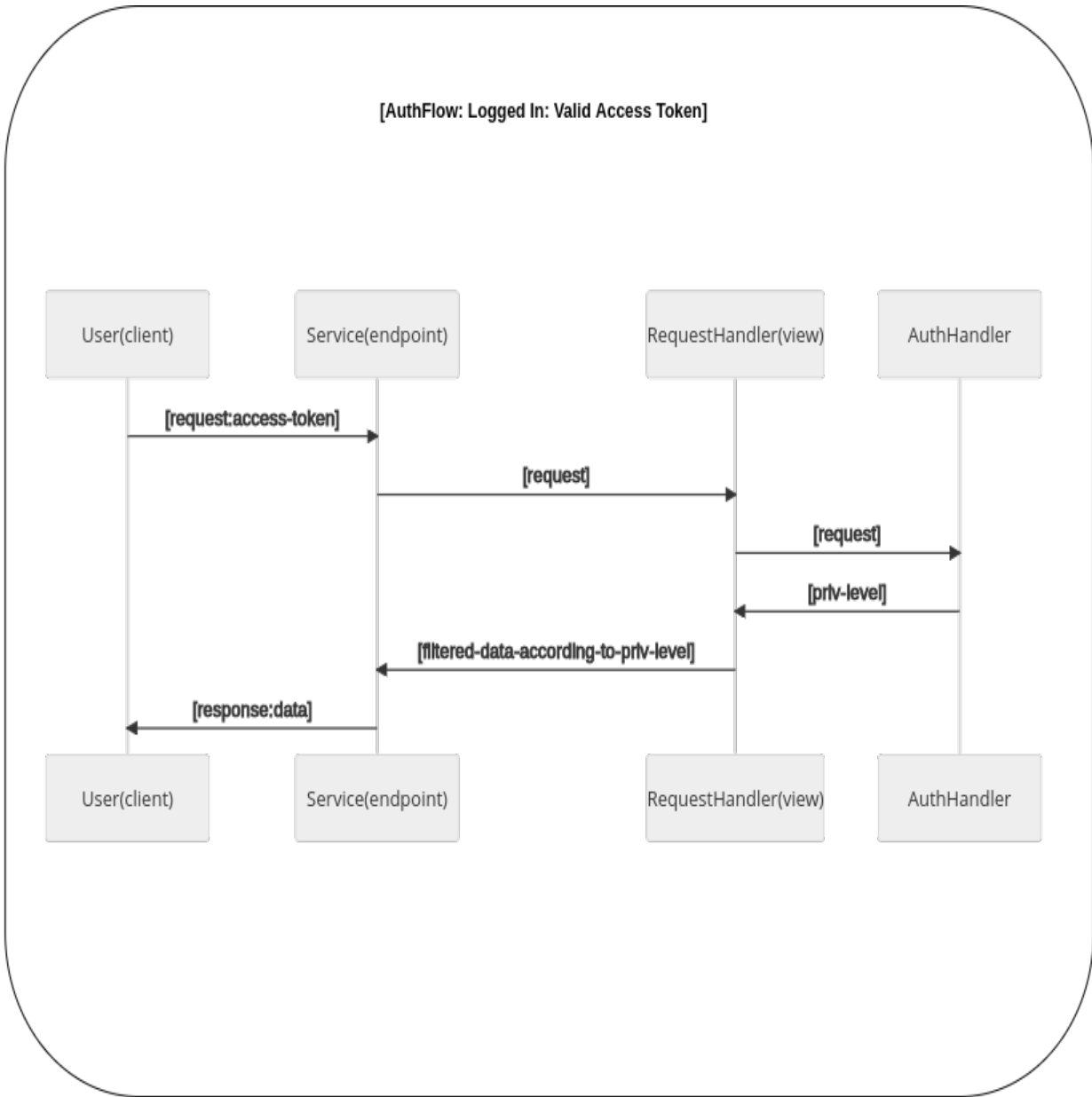


Figure 5.10: Logged In:Valid Access Token Sequence Diagram

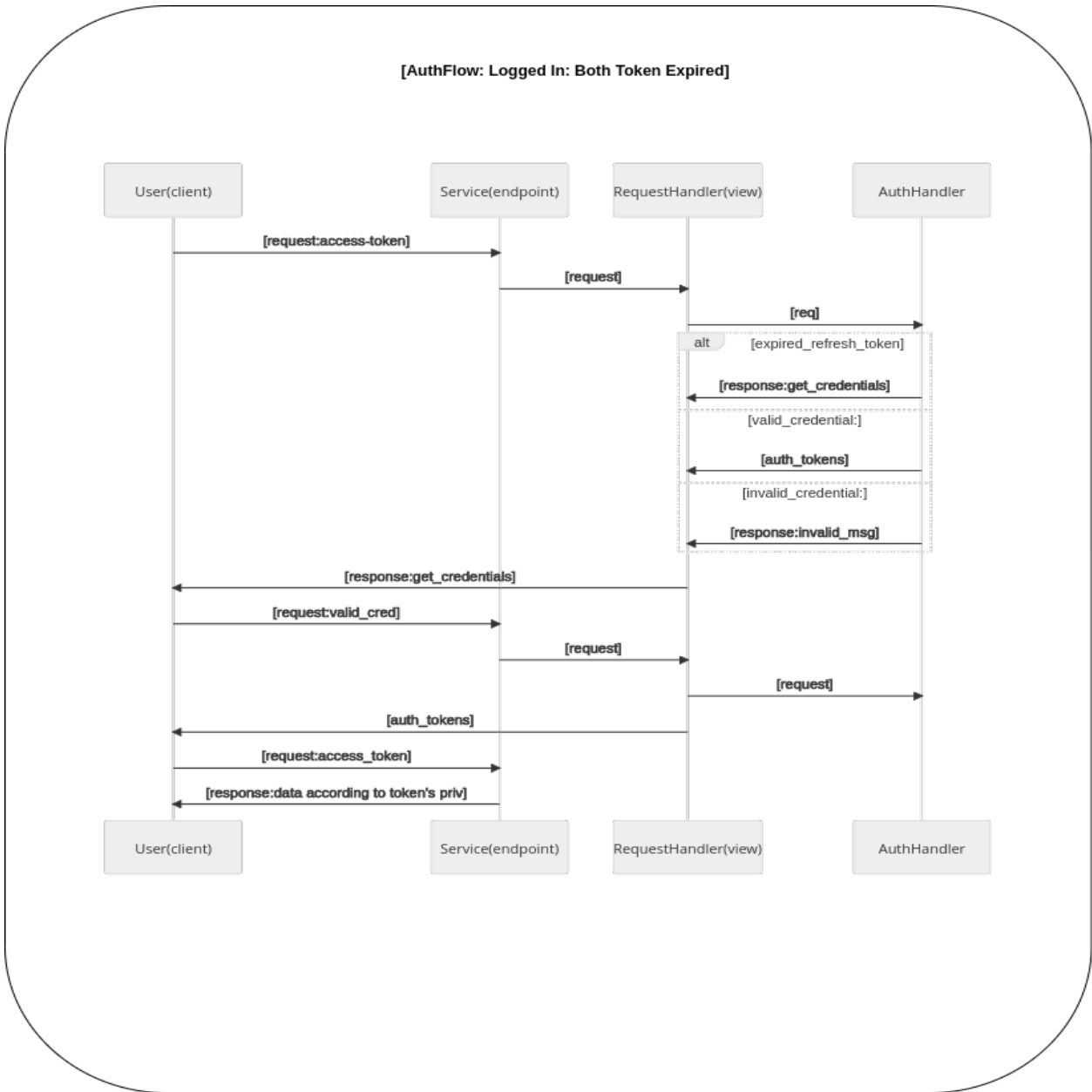


Figure 5.11: Logged In: Both Token Expired Sequence Diagram

## 6. Result

The output of the project is a mobile application that features routine management, syllabus organization, a library for accessing digital educational resources, notification alerts, personalized content tailored to individual users, and probable exam questions to enhance examination preparation. The application also includes a web-based administration portal to manage and oversee the various functionalities of the platform.

## 6.1 Mobile Features

### 6.1.1 Login

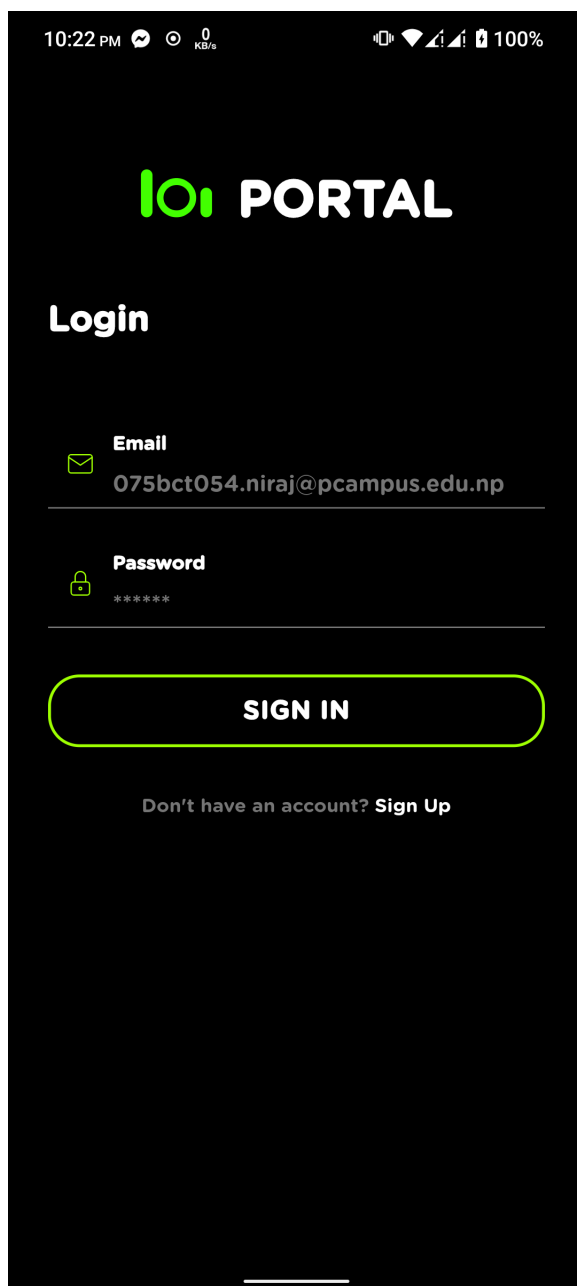


Figure 6.1: Login Page

## 6.1.2 Sign Up

10:22 PM 18 KB/s 100%

**ioi PORTAL**

Hustle free university,  
Join us now

1 — 2 — 3 — 4

**Email**  
075bct054.niraj@pcampu...

**Full Name**  
Niraj Shrestha

**Password**  
Password

**NEXT**

Already have an account? [Sign In](#)

Figure 6.2: Sign Up Page



6.1.3 Library

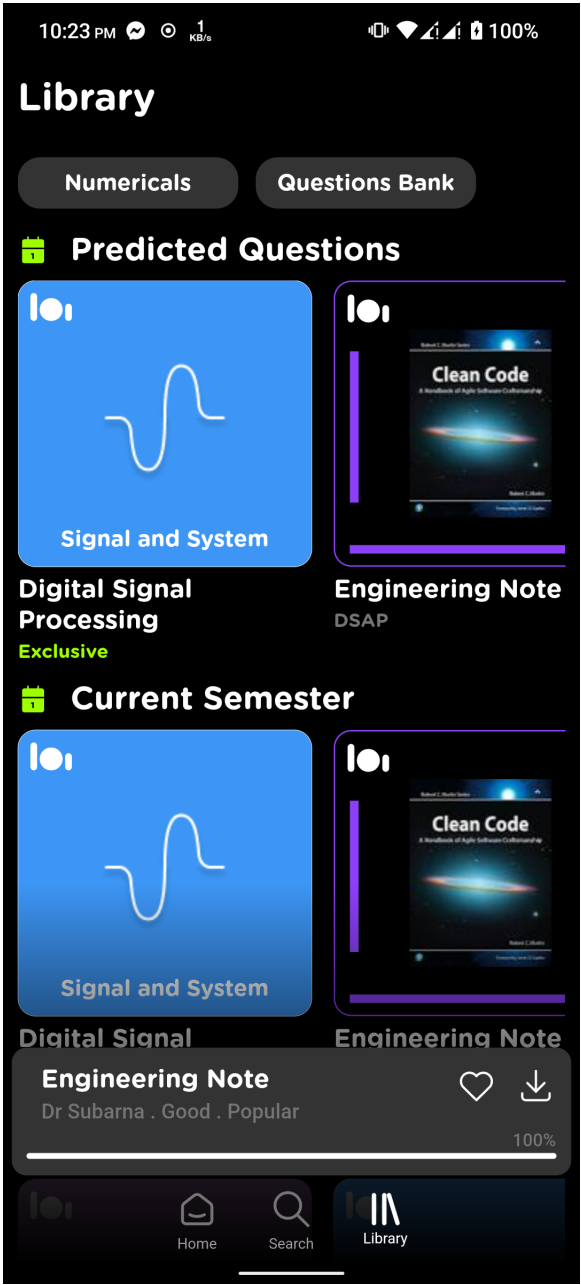


Figure 6.3: Library Page

## 6.1.4 Syllabus and Routine

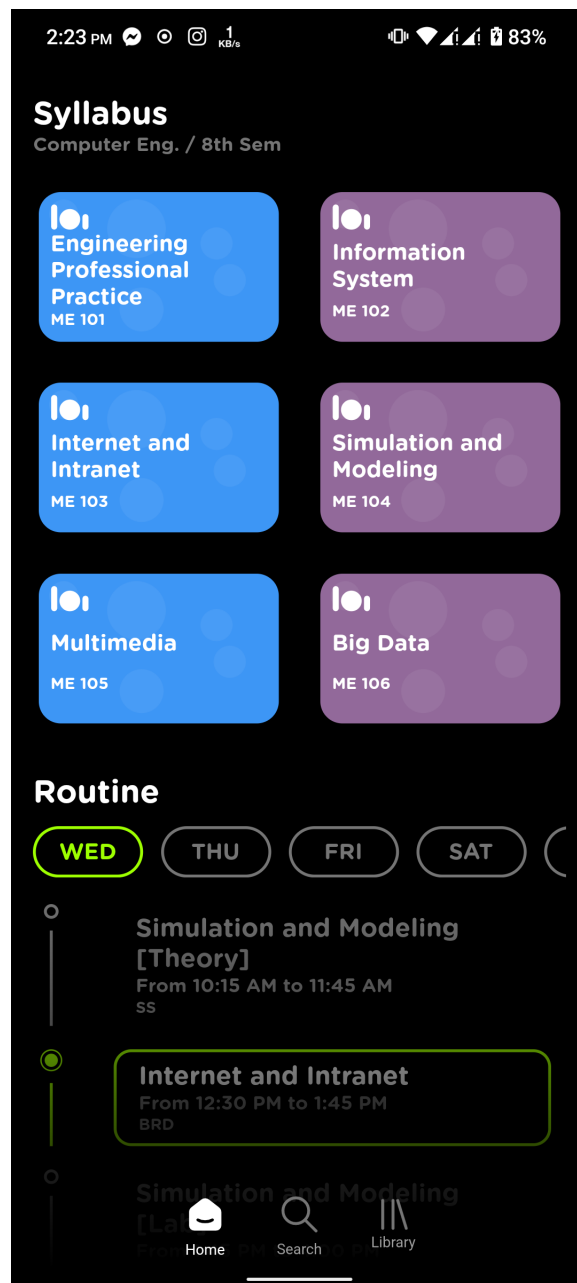


Figure 6.4: Syllabus and Routine

# 6.1.5 Document Viewer

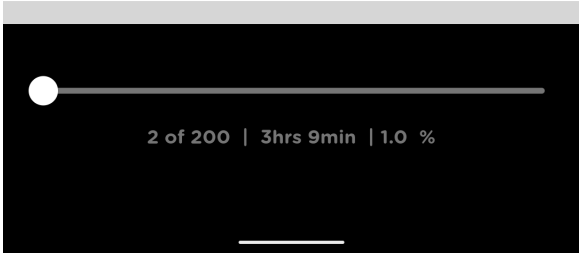
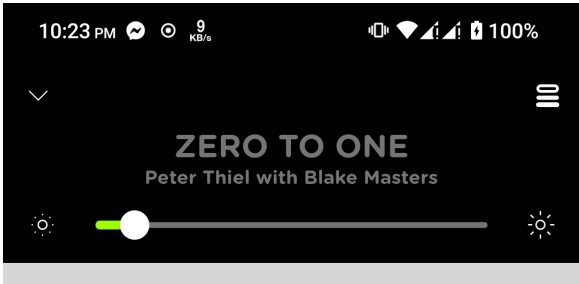


Figure 6.5: Document Viewer

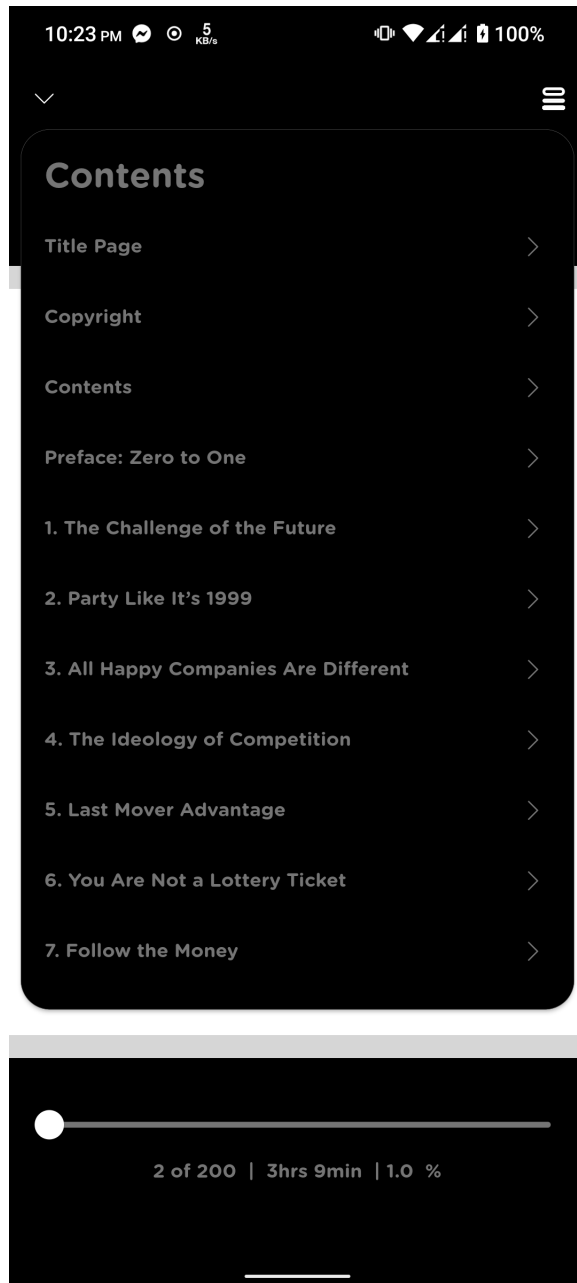


Figure 6.6: Document Contents

# 6.1.6 Notification

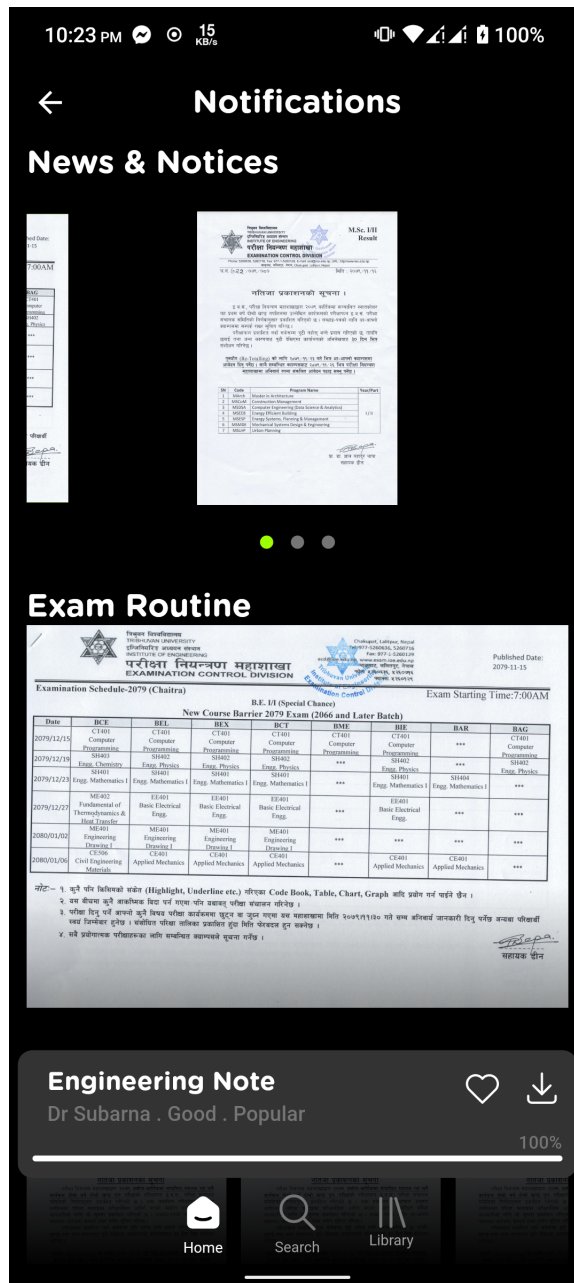


Figure 6.7: Notification Page

## 6.1.7 Probable Exam Question



Figure 6.8: Probable Exam Question

# 6.2 Web App



Figure 6.9: Web App

## 7. Conclusion

In conclusion, our report has demonstrated that our mobile app is a powerful and sophisticated tool for students seeking to improve their academic performance. By offering a comprehensive range of features, including a CMS for teachers, machine learning-powered recommendation systems, and a user-friendly interface available on both web and mobile platforms, our app represents the cutting-edge of modern education technology. The app is built on a robust and scalable infrastructure, including Postgres for its database, AWS beanstalk for deployment, and Circle CI and AWS Codebuild for streamlined development.

Our app's CMS, Creator Studio, allows teachers to upload resource material and attendance data, streamlining the learning experience and providing valuable insights into student progress. Additionally, the machine learning-powered recommendation system and topic importance prediction feature predict questions for the next exam, making studying more efficient and effective.

The app's accessibility on both web and mobile platforms means that students can access its features from anywhere, at any time. This level of accessibility, combined with its sophisticated feature set, makes our app the perfect solution for students seeking to maximize their academic performance.

Overall, our app represents a significant step forward in the field of education technology, offering students and teachers a powerful suite of tools and features designed to enhance the learning experience. By leveraging the latest in machine learning and cloud infrastructure technology, our app represents a powerful tool for students seeking to achieve their full academic potential.



## 8. Limitations and Future Enhancement

During the implementation of our project, we have identified several shortcomings that could enhance the system's performance and capabilities if resolved. These limitations with their future enhancement include:

- The document viewer of the app currently only supports PDF files. In the future, the app can be enhanced to support other popular document formats such as docx, epub etc.
- Currently, the content of the app can only be managed through the web app. This limitation can be addressed by developing a mobile app for content management, allowing for more convenient and efficient content updates and modifications.
- The app does not provide a feature for students to ask questions or interact with their peers and instructors. In the future, this limitation can be addressed by integrating a discussion forum or chat feature within the app, enabling students to ask questions and engage in discussions with their peers and instructors.

# Bibliography

- [1] L. Ceci, *Apple: Most popular app store categories*, May 2022. [Online]. Available: <https://www.statista.com/statistics/270291/popular-categories-in-the-app-store/>.
- [2] B. Hall, “Five trends in the LMS market,” *Chief Learning Officer*, vol. 7, 2008.
- [3] J. Kim, L. Ilon, and J. Altmann, “Adapting smartphones as learning technology in a korean university,” *J. Integr. Des. Process Sci.*, vol. 17, pp. 5–16, 2013.
- [4] M. Prensky, “Digital natives, digital immigrants part 1,” en, *On horiz.*, vol. 9, no. 5, pp. 1–6, Sep. 2001.
- [5] C. R. Bomhold, “Educational use of smart phone technology: A survey of mobile phone application use by undergraduate university students,” *Electronic Library and Information Systems*, vol. 47, pp. 424–436, 2013.
- [6] H. Crompton, “A historic view of m-learning toward learner centered education,” in 2013, pp. 3–15.
- [7] J. Traxler, “Learning in a mobile age,” *International Journal of Mobile and Blended Learning*, vol. 1, no. 1, pp. 1–12, 2009.
- [8] V. Sharples MTaylor, *Towards a theory of mobile learning*, <http://www.iamlearn.org/mlearn/>, 2021.
- [9] T. Cochrane, “M-learning as a subfield of open and distant education,” in 2013, pp. 15–24.
- [10] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.
- [11] R. Burke, “Hybrid recommender systems: Survey and experiments,” in *Proceedings of the 2007 ACM conference on Recommender systems*, ACM, 2007, pp. 91–98.
- [12] J. Cheng, L. Wu, Y. Liu, J. Xu, and K. Lu, “Sequential recommender system based on hierarchical attention network,” in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, ACM, 2018, pp. 305–314.
- [13] P. Melville, R. J. Mooney, and R. Nagarajan, “Content-boosted collaborative filtering for improved recommendations,” in *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, AAAI Press, 2002, pp. 187–192.

- [14] D. Ferraiolo D.F. Kuhn, "Role-based access control," *15th National Computer Security Conference*; no. 554, Oct. 1992.
- [15] Y. C. Sandhu R. Coyne E.J. Feinstein H.L., "Role-based access control models," *IEEE Computer*, vol. 29, no. 38, Aug. 1996.
- [16] S. A. O. V. E. K. S. M. ABREU VILMAR, "A multi-domain role activation model," *ICC 2017 2017 IEEE International Conference on Communication*, no. 1, 2017.
- [17] F. F. Gilbert MD Lynch N, "An examination of federal and commercial access control policy needs," *National Computer Security Conference, 1993 (16th) Proceedings: Information Systems Security: User Choices*, no. 107, 1995.