

# Study Notion Portal

## Full-Stack Web-Application

Major project report submitted in partial fulfilment of the  
requirement for the award of the degree of

**BACHELOR OF TECHNOLOGY**

IN

**COMPUTER SCIENCE AND ENGINEERING**



Submitted by:

Manoj Kumar , (206301098)

Manibhushan Kumar , (206301095)

Ramchandra Nath Siddh , (206301012)

Aditya Rajendra Narnaware , (206301102)

Under the guidance of:

Mr. Deepak Painuli

Assistant professor

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING  
FACULTY OF ENGINEERING AND TECHNOLOGY  
GURUKUL KANGRI UNIVERSITY  
2023-2024**

## ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **Mr. Deepak Painuli** for your invaluable guidance and support throughout the process of preparing this project work. Your expertise and insightful feedback have been instrumental in shaping the content and ensuring its quality. Your dedication to fostering a conducive learning environment and your commitment to excellence have been a constant source of inspiration. I am truly grateful for the time and effort you have invested in providing constructive criticism and encouraging me to strive for academic excellence.

I appreciate the opportunity to learn under your tutelage, and I am confident that the knowledge and skills acquired during this process will be invaluable in my academic and professional journey.

### Name of students

Manoj Kumar

Manibhushan Kumar

Ramchandra Nath Siddh

Aditya Rajendra Narnaware

## ABSTRACT

StudyNotion is a dynamic ed-tech platform designed to revolutionize online learning experiences. Built upon the MERN stack, StudyNotion offers a seamless blend of cutting-edge technologies to create, consume, and rate educational content. At its core, StudyNotion aims to bridge the gap between students and instructors, fostering a global community of learners. Through intuitive user interfaces and interactive features, students can engage with a diverse range of educational materials, making learning accessible and enjoyable. Meanwhile, instructors have a platform to showcase their expertise and connect with learners worldwide, enriching the educational landscape with diverse perspectives and insights.

The platform's architecture ensures scalability and reliability, with a robust back-end powered by NodeJS and MongoDB handling data management and processing. Security measures are implemented to safeguard user data and privacy, maintaining trust and integrity within the StudyNotion community. Through a carefully crafted API design, developers can seamlessly integrate StudyNotion's functionalities into their applications, expanding its reach and impact across different platforms and devices. Continuous deployment practices ensure rapid updates and enhancements, keeping the platform dynamic and responsive to user needs and technological advancements.

Looking ahead, StudyNotion envisions a future where education knows no boundaries. Potential enhancements include personalized learning experiences, real-time collaboration tools, and advanced analytics for insights-driven education. By prioritizing user feedback and market trends, StudyNotion remains committed to its mission of democratizing education, empowering learners and instructors to thrive in an ever-evolving digital landscape.

## CERTIFICATE

This is to certify that the content of this Major project report are compiled and written by us and we have not copied them from anywhere. This work is done by us and we have not presented it anywhere else.

Manoj Kumar Manibhushan Kumar (206301098)	Ramchandra Siddh , (206301095)	Nath Aditya Rajendra Narnaware (206301102)
--	-----------------------------------	--

## CERTIFICATE

This is to certify that the project work embodied in this project file are done under my supervision and are not being copied from anywhere or presented anywhere for any sort of benefit.

**Mr. Deepak Painuli**  
Assistant Professor  
Department of Computer Science & Engineering  
Faculty of Engineering and Technology

## **TABLE OF CONTENTS**

### **CHAPTER 1: INTRODUCTION**

- 1.1 Background of the Study
- 1.2 Problem Statement
- 1.3 Aim and Objectives
- 1.4 Research Questions
- 1.5 Scope of the Study
- 1.6 Significance of the Study
- 1.7 Structure of the Study

### **CHAPTER 2: LITERATURE REVIEW**

- 2.1 Introduction
- 2.2 Recommender System
  - 2.2.1 History of Recommender System
  - 2.2.2 Application of Recommender System
  - 2.2.3 Types of Recommender System
- 2.3 Challenges in recommender system
  - 2.3.1 Sparsity
- 2.4 Comparison of Techniques
- 2.5 Discussion
- 2.6 Summary

## **CHAPTER 3: RESEARCH METHODOLOGY**

- 3.1 Introduction
- 3.2 Methodology
  - 3.2.1. Data Selection
  - 3.2.2. Data Pre-Processing
  - 3.2.3. Data Transformation
  - 3.2.4. Modelling
  - 3.2.5. Evaluation
- 3.3 Logical flow of the system
- 3.4 Item based collaborative filtering
  - 3.4.1. Item-Similarity Computation
- 3.5 Rating Prediction Computation
- 3.6 Generating Recommendation
- 3.7 Tools
  - 3.7.1 Python
  - 3.7.2 Graph lab Create
  - 3.7.3 Pandas
  - 3.7.4 NLTK

## **CHAPTER 4: IMPLEMENTATION**

- 4.1 Introduction
- 4.2 Dataset
  - 4.2.1 Business
  - 4.1.2 User

## 4.3 Exploratory Data Analysis

### 4.3.1 Business Dataset

### 4.4 Data cleaning

#### 4.4.1 Business Dataset

### 4.5 Data Partitioning

### 4.6 Model Implementation

#### 4.6.1 Data Staging

#### 4.6.2 Model Building

#### 4.6.5. Recommender System Interface

### 4.7 Summary

## **CHAPTER 5: RESULTS AND EVALUATION**

### 5.1 Introduction

### 5.2 Model Output

### 5.3 Summary

## **CHAPTER 6: CONCLUSIONS AND RECOMMENDATIONS**

### 6.1 Introduction

### 6.1 Discussion and Conclusion

### 6.1 Contributions

### 6.1. Future Work

## **RSEFERENCE**

## CHAPTER 1

### INTRODUCTION

#### **1.1 Background of the Study**

The StudyNotion ed-tech platform is a cutting-edge solution designed to revolutionize online education by offering a dynamic and engaging learning experience. With a robust architecture comprising front-end, back-end, and database components, StudyNotion follows a client-server model, ensuring seamless interaction between users and the platform. Leveraging industry-leading technologies such as ReactJS for the front end, Node.js and Express.js for the back end, and MongoDB for flexible data storage, StudyNotion prioritizes user satisfaction and accessibility. At the forefront of StudyNotion's architecture is its user-centric approach, evident in features like user authentication, course management, and payment integration. By providing an intuitive interface for students, instructors, and administrators alike, StudyNotion aims to foster a thriving educational community. With a commitment to scalability, security, and innovation, StudyNotion sets the stage for a transformative learning experience, empowering users to pursue their educational goals with confidence.

#### **1.2 Problem Statement**

Traditional course methods pose several challenges, such as:

In today's digital age, traditional education systems are rapidly evolving, with more individuals seeking flexible and accessible learning opportunities. However, amidst this transformation, there remains a significant gap in the availability of high-quality online courses and a streamlined platform to facilitate their purchase and delivery. Existing course-selling websites often lack user-friendly interfaces, comprehensive course offerings, and reliable payment systems, hindering both learners and instructors from fully leveraging the potential of online education.

Furthermore, many learners struggle to find reputable courses tailored to their specific interests and learning objectives, leading to frustration and wasted time navigating through an overwhelming array of options. Similarly, instructors face challenges in effectively showcasing their expertise and attracting students to their courses, limiting their ability to reach a wider audience and maximize their earning potential.

Therefore, there is a critical need for a sophisticated and user-centric course-selling website that addresses these challenges head-on. Such a platform should provide a seamless and intuitive interface for both learners and instructors, offering a diverse range of high-quality courses across various disciplines. It should prioritize user experience, ensuring easy navigation, personalized recommendations, and reliable payment processing to enhance satisfaction and trust.

Moreover, the platform must implement robust features for course management, including tools for instructors to create, publish, and manage their courses efficiently, along with mechanisms for learners to track their progress and engage with course materials effectively. Additionally, it should incorporate advanced search and filtering options, enabling users to discover courses tailored to their preferences and learning goals effortlessly.

Overall, the development of a comprehensive and innovative course-selling website presents an opportunity to revolutionize online education, bridging the gap between learners and instructors while empowering individuals to pursue their educational aspirations with confidence and convenience.

**Logistical Issues:** Managing physical courses, answer sheets, and coordinating exam centers can be a logistical nightmare for educational institutions.

**Time Constraints:** The time-consuming process of paper evaluation, result compilation, and feedback provision can delay the entire examination cycle.

**Security Concerns:** Maintaining the integrity and security of the examination process, including preventing cheating and ensuring fair evaluation, is a constant challenge.

**Accessibility:** Geographical constraints and physical disabilities may hinder some candidates from participating in traditional exams.

The Study Notion Portal Website aims to address these challenges by providing a centralized, digital platform for the entire examination process.

### **1.3 Aim and Objectives**

#### **1.3.1 Aim**

The primary aim of this project is to design, develop, and implement an Study Notion Website that offers a secure, efficient, and user-friendly platform for online courses.

#### **1.3.2 Objectives**

Develop a robust online courses system that ensures data security and prevents unauthorized access.

Create a dynamic and interactive learning environment to increase student engagement and retention.

Implement features that allow for customized learning paths based on individual student preferences and learning styles.

Streamline the process of creating, managing, and delivering course content for educators to improve productivity and course quality.

Ensure the security and privacy of user data through robust security measures and compliance with data protection regulations.

Build a platform capable of handling increasing user traffic and data volumes to accommodate future growth.

Provide a user-friendly interface for both administrators and candidates, ensuring ease of use and accessibility.

Foster collaboration and interaction among students and instructors to facilitate peer learning and mentorship opportunities.

Provide timely and constructive feedback mechanisms to help students track their progress and improve their learning outcomes.

Conduct thorough testing to ensure the reliability and performance of the Study Notion Website.

## 1.4 Research Questions

The research questions guiding this study include:

How can the Study Notion Website address the logistical challenges associated with traditional course systems?

What security measures can be implemented to ensure the integrity of the online courses process?

How can the Study Notion Website enhance accessibility for candidates with diverse needs?

What impact does the implementation of the Study Notion Website have on the efficiency of the examination process?

## 1.5 Scope of the Study

This study focuses on the development and implementation of the Study Notion Website within the context of educational institutions. The scope includes the entire courses process, from different types of courses creation to result generation.

## 1.6 Significance of the Study

The successful implementation of the Study Notion Website will have significant implications for the education sector. It can lead to:

Efficiency: Streamlining the examination process, reducing paperwork, and expediting result generation.

Accessibility: Providing a platform that accommodates candidates with diverse needs and geographical constraints.

Security: Enhancing the security and integrity of the courses process through robust online mechanisms.

## 1.7 Structure of the Study

The remainder of this report is organized as follows:

1.7.2: Literature Review - A comprehensive review of existing literature on online courses systems, their advantages, challenges, and best practices.

1.7.3: Methodology - Details on the research design, development process, and testing methodologies.

1.7.4: System Design - A thorough explanation of the design principles, architecture, and features of the Study Notion Portal Website.

1.7.5: Implementation - A discussion of the practical implementation of the Study Notion Website.

1.7.6: Testing and Evaluation - Examination of the reliability, security, and performance of the developed system.

1.7.7: Results and Discussion - Presentation and analysis of the findings.

1.7.8: Conclusion and Recommendations - Concluding remarks, implications of the study, and recommendations for future research and development.

## CHAPTER 2

### LITERATURE REVIEW

#### **2.1 Introduction**

This chapter provides a comprehensive review of the literature related to the development and implementation of an Study Notion Website, with a particular focus on recommender systems. Recommender systems play a crucial role in enhancing the user experience by suggesting relevant content. In the context of an Study Notion Website, the incorporation of recommender systems can offer personalized recommendations for learning resources, practice tests, and exam preparation materials.

#### **2.2 Recommender System**

##### **2.2.1 History of Recommender System**

The concept of recommender systems dates back to the late 20th century. Initially developed in the context of e-commerce and content consumption platforms, recommender systems gained prominence with the rise of the internet. The evolution of recommendation algorithms has been influenced by collaborative filtering, content-based filtering, and hybrid models.

##### **2.2.2 Application of Recommender System**

Recommender systems have found applications in diverse fields, including e-commerce, social media, and education. In the education sector, recommender systems are employed to suggest relevant courses, learning materials, and assessments. Integrating a recommender system into an Study Notion Website can enhance the user experience by tailoring content to individual learning preferences.

##### **2.2.3 Types of Recommender System**

There are several types of recommender systems, each with its unique approach to making recommendations. The two main categories are collaborative filtering and content-based filtering. Collaborative filtering relies on user behavior and preferences, while content-based filtering considers the characteristics of items and user preferences. Hybrid models combine these approaches to overcome their individual limitations.

#### **2.3 Challenges in Recommender System**

##### **2.3.1 Sparsity**

One of the prominent challenges in recommender systems, including those in education, is sparsity. Sparsity refers to the scarcity of user-item interactions in the dataset, making it challenging to provide accurate recommendations. Techniques such as matrix factorization and deep learning have been employed to address sparsity issues and enhance the accuracy of recommendations.

## **2.4 Comparison of Techniques**

Various techniques have been developed to address the challenges associated with recommender systems. Collaborative filtering methods, including user-based and item-based approaches, have been widely used. Content-based filtering leverages item features to make recommendations. Hybrid models combine collaborative and content-based approaches to provide more robust and accurate recommendations.

## **2.5 Discussion**

The integration of recommender systems in an Study Notion Portal Website holds significant potential for improving user engagement and learning outcomes. By understanding user preferences, the system can recommend relevant practice , study materials, and resources, thereby personalizing the learning experience. However, addressing challenges such as sparsity is crucial to ensuring the effectiveness of these systems.

## **2.6 Summary**

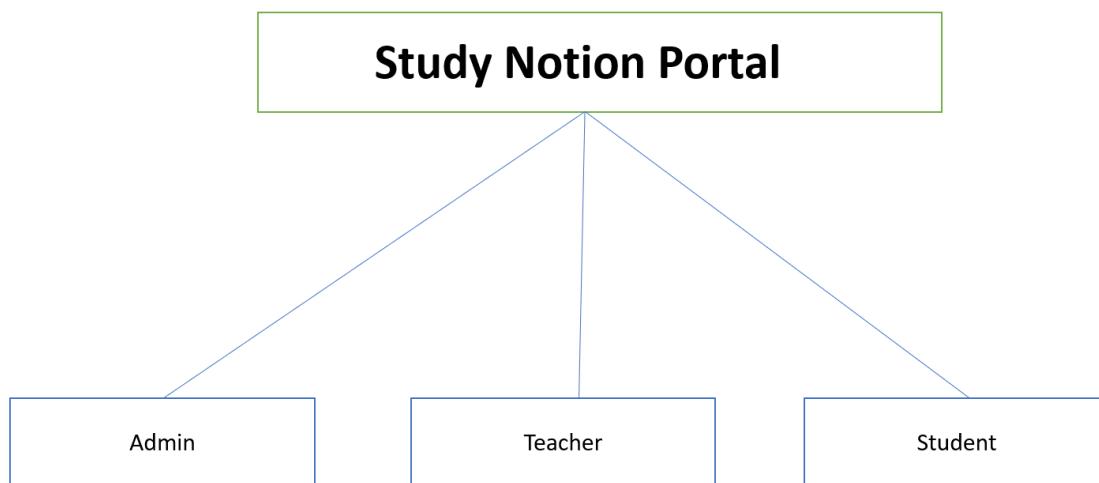
This chapter provided an in-depth review of the literature on recommender systems, emphasizing their historical development, applications, types, challenges, and comparison of techniques. The insights gained from this literature review will inform the subsequent chapters, guiding the design and implementation of the Study Notion Portal Website with a focus on integrating recommender systems to enhance the user experience.

## CHAPTER 3

### RESEARCH METHODOLOGY

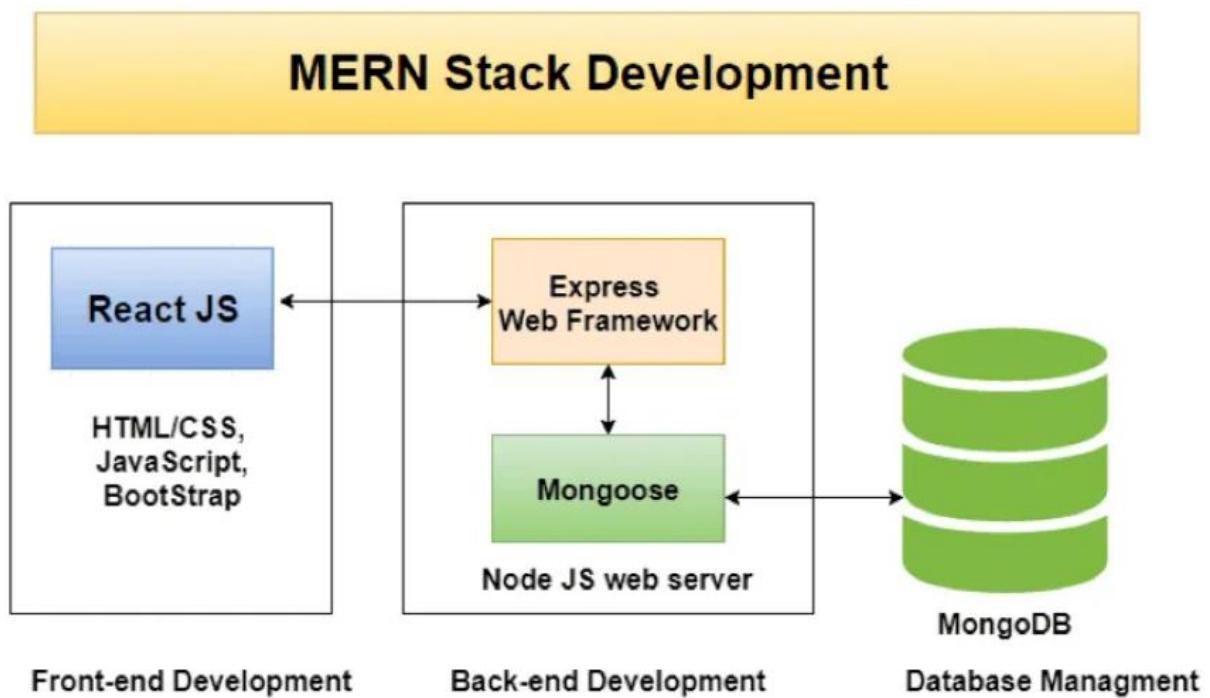
#### 3.1 Introduction

This chapter outlines the research methodology employed in the development and implementation of the Study Notion Website, with a specific focus on the integration of a recommender system. The methodology encompasses data selection, pre-processing, transformation, modeling, and evaluation, providing a logical flow for the system development.



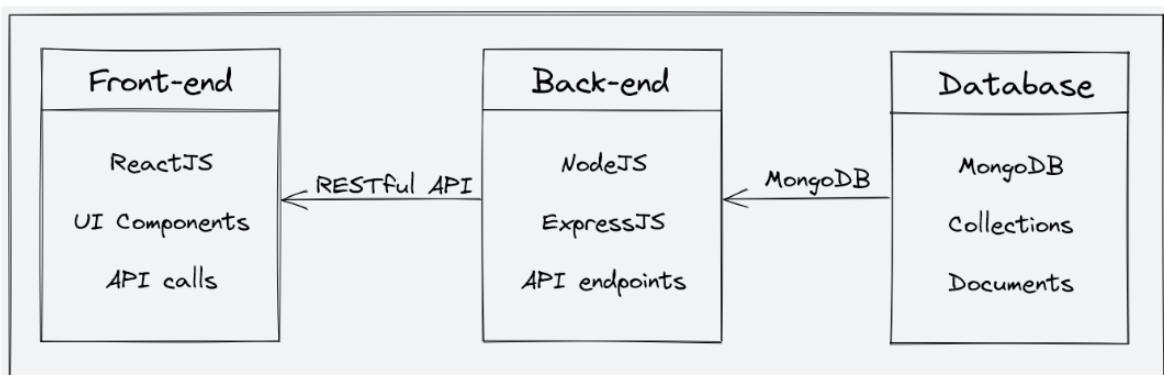
**Study Notion Portal Architecture**

**Fig:3.1.1**



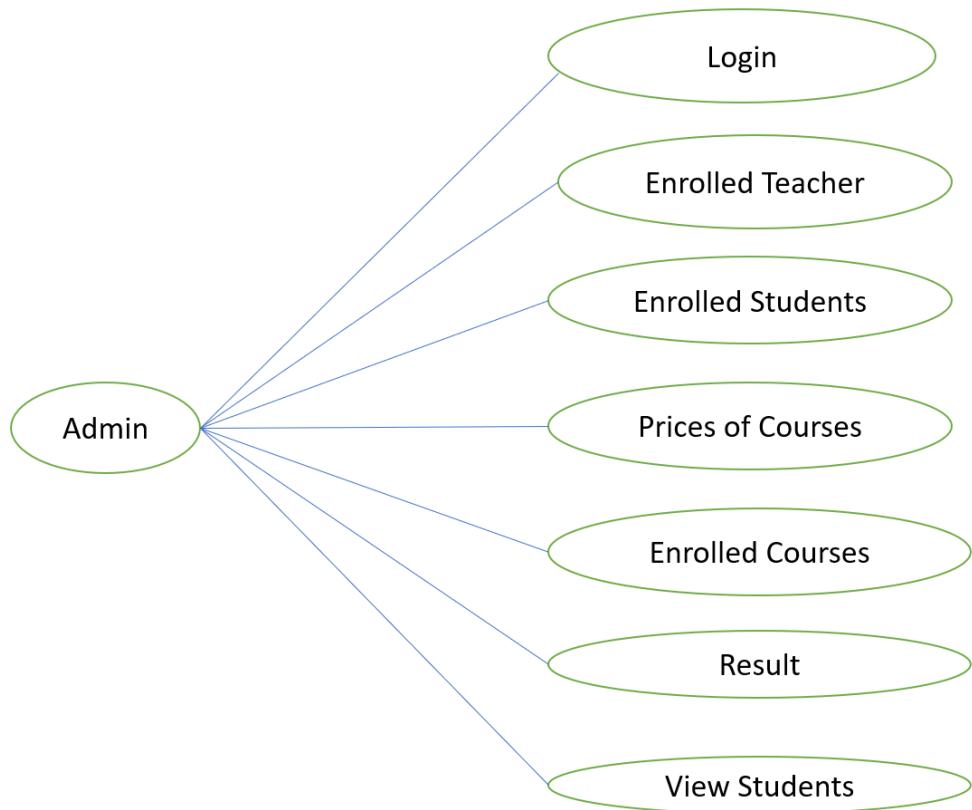
**MERN Stack Development**

**Fig:3.1.2**



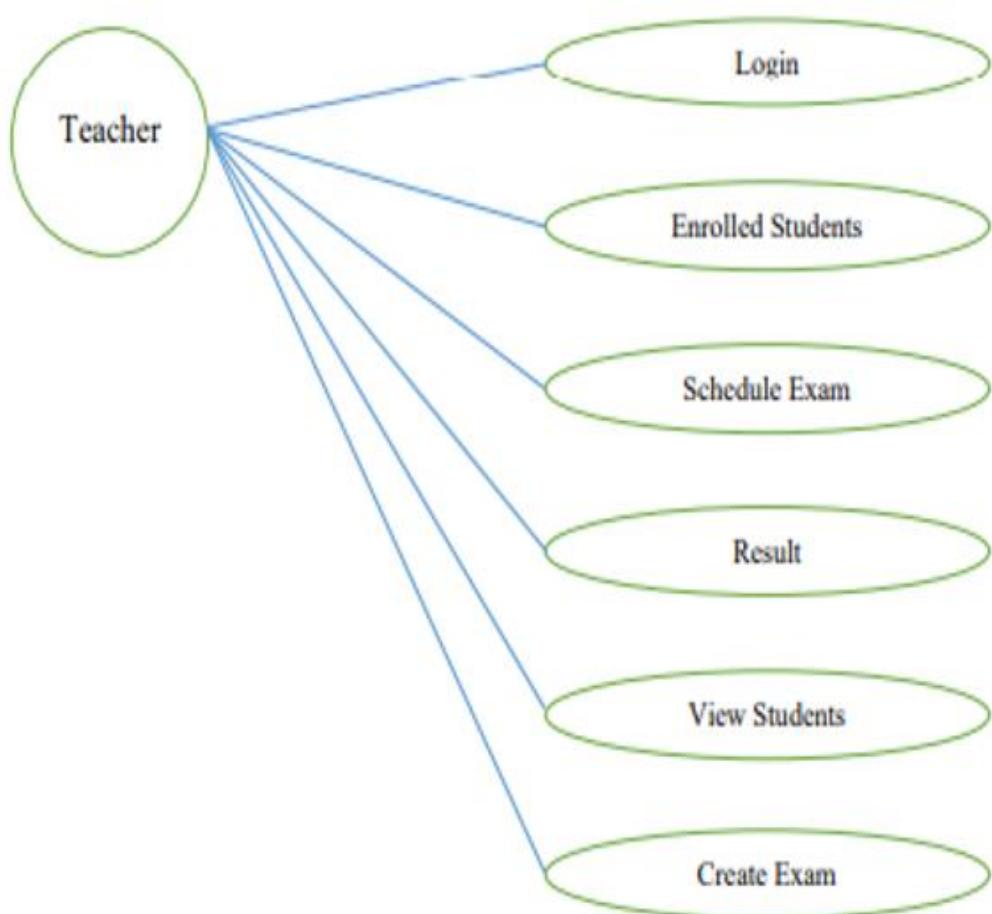
**Font-End, Back-End and DataBase of Study Notion Portal**

**Fig:3.1.3**



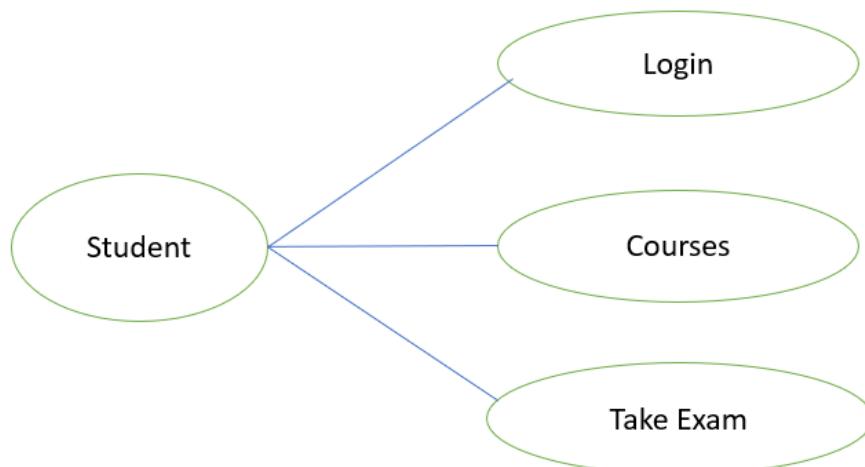
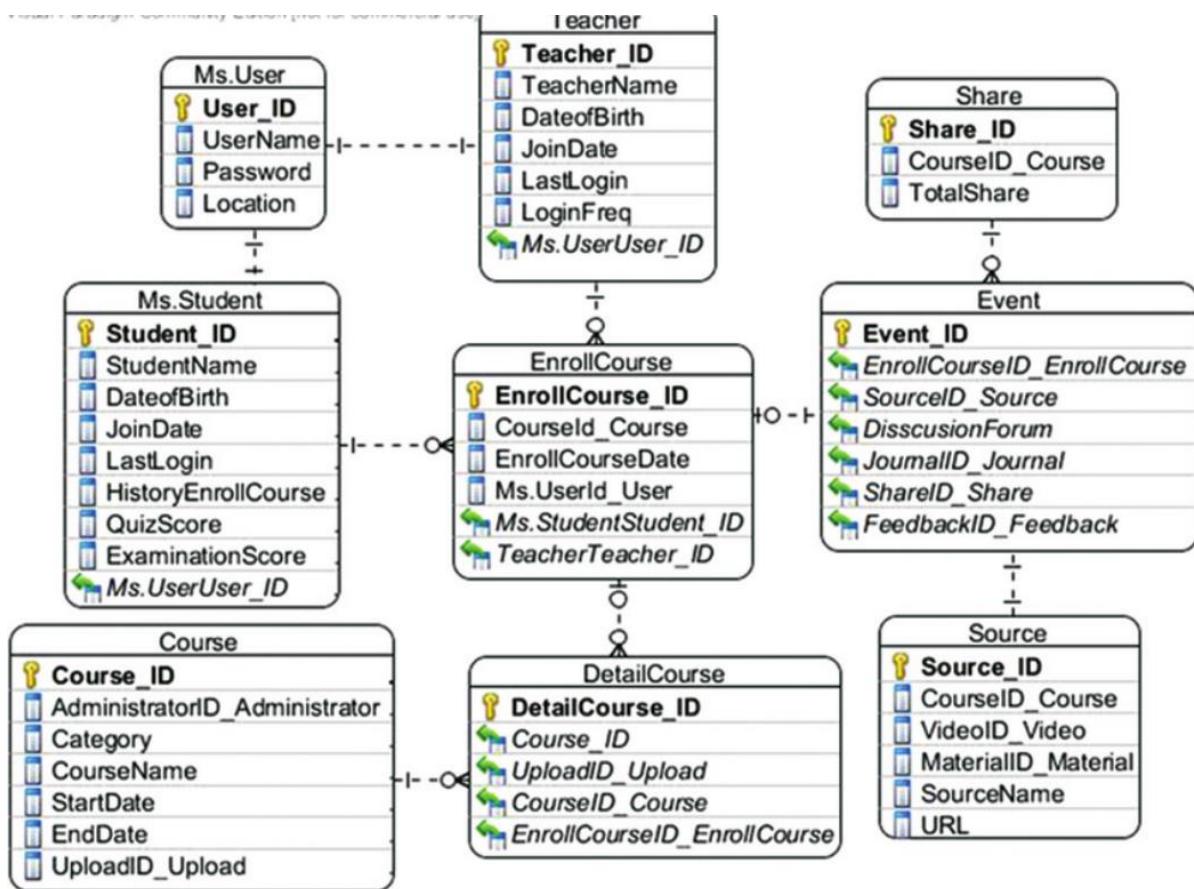
**Admin Flow Chart**

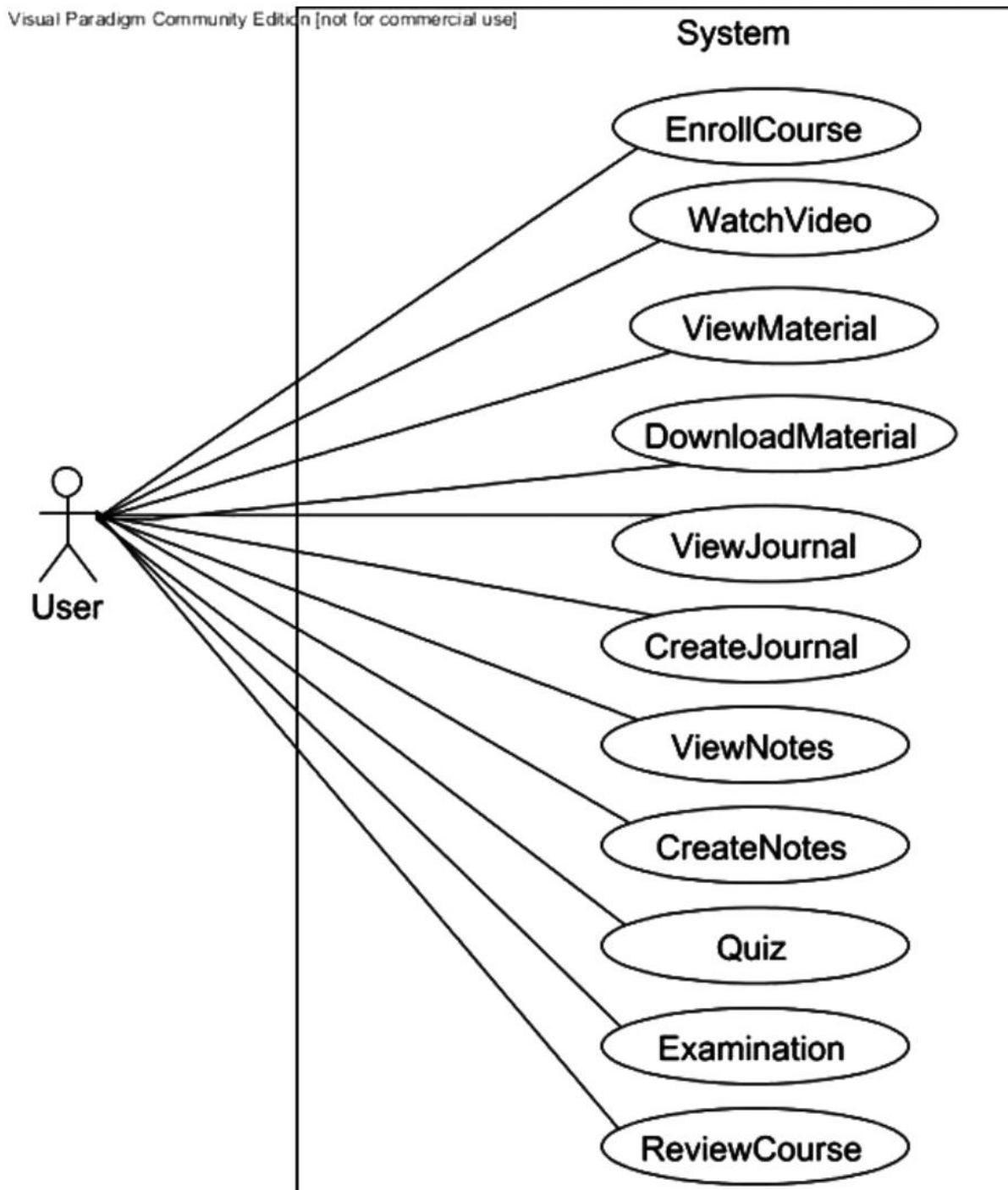
**Fig:3.1.4**



**Teacher Flow Chart**

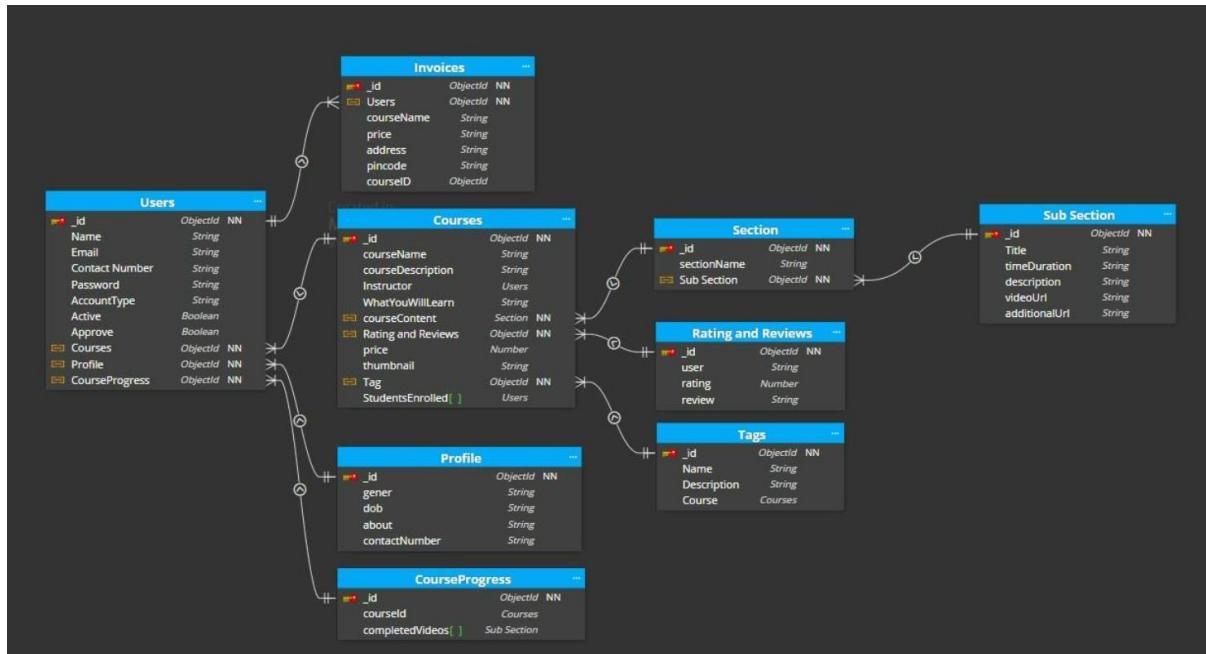
**Fig:3.1.5**

**Student Flow Chart****Fig:3.1.6**



User Flow Diagram

Fig:3.1.8



## API Working of Study Notion Portal

**Fig:3.1.9**

## 3.2 Methodology

### 3.2.1 Data Selection

The selection of relevant data is critical for the success of the recommender system. This phase involves identifying and gathering datasets related to user interactions with the Study Notion portal Website, including users, courses preferences, and historical data.

### 3.2.2 Data Pre-Processing

Data pre-processing involves cleaning and transforming raw data into a usable format. It includes handling missing values, removing duplicates, and addressing inconsistencies to ensure data quality.

### 3.2.3 Data Transformation

Transformation of data involves converting raw data into a suitable format for modeling. This includes encoding categorical variables, scaling numerical features, and preparing the data for recommender system implementation.

### **3.2.4 Modelling**

The modeling phase focuses on the selection and implementation of the recommender system algorithm. In this case, the study utilizes item-based collaborative filtering for generating personalized recommendations.

### **3.2.5 Evaluation**

The evaluation process involves assessing the performance of the implemented recommender system. Metrics such as precision, recall, and Mean Squared Error (MSE) are employed to measure the accuracy and effectiveness of the recommendations.

## **3.3 Logical Flow of the System**

This section provides a step-by-step logical flow of the recommender system implementation within the Study Notion Portal Website, from data selection to evaluation.

## **3.4 Item-based Collaborative Filtering**

### **3.4.1 Item-Similarity Computation**

Item-based collaborative filtering relies on calculating the similarity between items based on user interactions. This involves computing item similarity scores to identify items with similar user preferences.

### **3.5 Rating Prediction Computation**

The recommender system predicts user ratings for items based on historical data, enabling the system to understand user preferences and recommend relevant items.

### **3.6 Generating Recommendation**

The final step involves generating personalized recommendations for users based on their historical interactions and predicted ratings.

### **3.7 Tools**

#### **3.7.1 Python**

Python is the primary programming language for implementing the recommender system and the Study Notion Portal Website.

#### **3.7.2 Graph lab Create**

Graph lab Create provides a comprehensive toolkit for building and deploying machine learning models, including recommender systems.

#### **3.7.3 Pandas**

Pandas is used for data manipulation and analysis, facilitating efficient data pre-processing.

#### **3.7.4 NLTK**

Natural Language Toolkit (NLTK) supports text processing, enabling the system to understand and analyze textual data related to user preferences.

#### **3.7.5 Flask**

Flask is employed for developing the interface of the Exam-Portal Website, providing a user-friendly platform for interaction.

### **3.8 Summary**

This chapter has outlined the research methodology, including data selection, pre-processing, transformation, modelling , and evaluation. The logical flow of the system, along with the tools used, sets the foundation for the implementation phase.

## **CHAPTER 4**

### **IMPLEMENTATION**

#### **4.1 Introduction**

This chapter delves into the practical implementation of the Study Notion Portal Website, focusing on dataset selection, exploratory data analysis, data cleaning, partitioning, and the implementation of the recommender system.

#### **4.2 Dataset**

##### **4.2.1 Business**

The business dataset encompasses information related to courses, user profiles, and historical interactions within the Study Notion Website.

##### **4.2.2 User**

The user dataset includes demographic information, preferences, and historical data reflecting user interactions with the platform.

#### **4.3 Exploratory Data Analysis**

Exploratory Data Analysis (EDA) involves a detailed examination of the datasets to identify patterns, trends, and insights that inform the recommender system implementation.

#### **4.4 Data Cleaning**

##### **4.4.1 Business Dataset**

Data cleaning involves addressing missing values, removing duplicates, and resolving inconsistencies in the business dataset.

#### **4.5 Data Partitioning**

The dataset is partitioned into training and testing sets to facilitate the evaluation of the recommender system's performance.

#### **4.6 Model Implementation**

##### **4.6.1 Data Staging**

Data staging involves preparing the dataset for recommender system implementation through encoding, scaling, and other pre-processing steps.

#### **4.6.2 Model Building**

The recommender system model is built using item-based collaborative filtering, utilizing item-similarity computation and rating prediction.

#### **4.6.3 Recommender System Interface**

The Study Notion Portal Website interface is developed to seamlessly integrate the recommender system, providing users with personalized recommendations.

#### **4.7 Summary**

This chapter has provided insights into the practical implementation of the Study Notion Portal Website, emphasizing dataset selection, exploratory data analysis, data cleaning, partitioning, and recommender system integration.

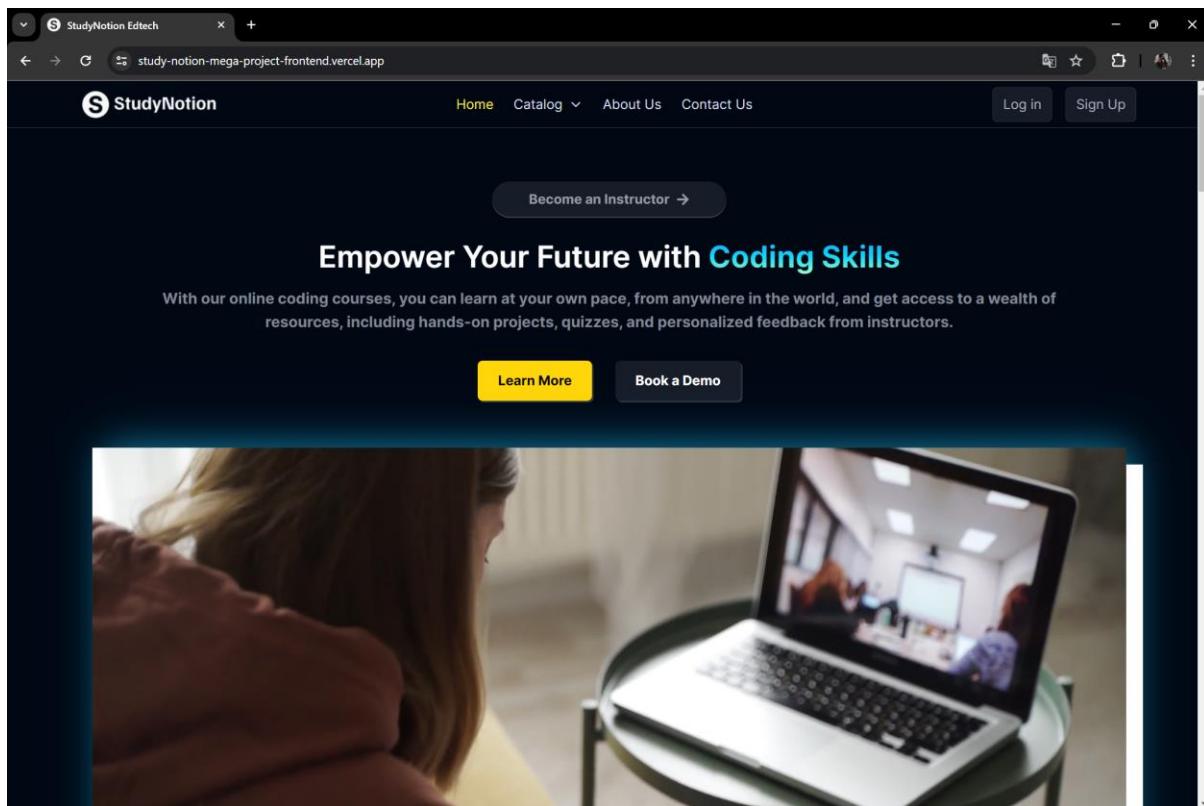
## CHAPTER 5

### RESULTS AND EVALUATION

#### 5.1 Introduction

This chapter presents the results obtained from the implementation of the Study Notion Portal Website and the evaluation of the recommender system.

#### 5.2 Project Interface



#### 5.3 Student Sign Up Page Code and Interface

```
import { useState } from "react"
import { toast } from "react-hot-toast"
import { AiOutlineEye, AiOutlineEyeInvisible } from "react-icons/ai"
import { useDispatch } from "react-redux"
import { useNavigate } from "react-router-dom"

import { sendOtp } from "../../services/operations/authAPI"
import { setSignupData } from "../../slices/authSlice"
import { ACCOUNT_TYPE } from "../../utils/constants"
import Tab from "../../common/Tab"
```

```
function SignupForm() {
  const navigate = useNavigate()
  const dispatch = useDispatch()

  // student or instructor
  const [accountType, setAccountType] = useState(ACCOUNT_TYPE.STUDENT)

  const [formData, setFormData] = useState({
    firstName: "",
    lastName: "",
    email: "",
    password: "",
    confirmPassword: ""
  })

  const [showPassword, setShowPassword] = useState(false)
  const [showConfirmPassword, setShowConfirmPassword] = useState(false)

  const { firstName, lastName, email, password, confirmPassword } = formData

  // Handle input fields, when some value changes
  const handleOnChange = (e) => {
    setFormData((prevData) => ({
      ...prevData,
      [e.target.name]: e.target.value,
    }))
  }

  // Handle Form Submission
  const handleOnSubmit = (e) => {
    e.preventDefault()

    if (password !== confirmPassword) {
      toast.error("Passwords Do Not Match")
      return
    }
    const signupData = {
      ...formData,
      accountType,
    }

    // Setting signup data to state
    // To be used after otp verification
    dispatch(setSignupData(signupData))
    // Send OTP to user for verification
    dispatch(sendOtp(formData.email, navigate))
  }
}
```

```
// Reset
setFormData({
  firstName: "",
  lastName: "",
  email: "",
  password: "",
  confirmPassword: ""
})
setAccountType(ACCOUNT_TYPE.STUDENT)
}

// data to pass to Tab component
const tabData = [
{
  id: 1,
  tabName: "Student",
  type: ACCOUNT_TYPE.STUDENT,
},
{
  id: 2,
  tabName: "Instructor",
  type: ACCOUNT_TYPE.INSTRUCTOR,
},
]
]

return (
<div>
  {/* Tab */}
  <Tab tabData={tabData} field={accountType}>
    setField={setAccountType} />
  {/* Form */}
  <form onSubmit={handleOnSubmit} className="flex w-full flex-col gap-y-4">
    <div className="flex gap-x-4">
      <label>
        <p className="mb-1 text-[0.875rem] leading-[1.375rem] text-richblack-5">
          First Name <sup className="text-pink-200">*</sup>
        </p>
        <input
          required
          type="text"
          name="firstName"
          value={firstName}
          onChange={handleOnChange}
          placeholder="Enter first name"
          style={{
            width: "150px"
          }}>
      </label>
      <label>
        <p>Last Name</p>
        <input
          required
          type="text"
          name="lastName"
          value={lastName}
          onChange={handleOnChange}
          placeholder="Enter last name"
          style={{
            width: "150px"
          }}>
      </label>
    </div>
    <button type="submit" className="w-full text-white bg-blue-700 py-2 px-4 rounded border border-blue-700 font-medium transition-colors duration-300 ease-in-out">Create Account</button>
  </form>
</Tab>
)
```

```
        boxShadow: "inset 0px -1px 0px rgba(255, 255, 255, 0.18)",
      }
    className="w-full rounded-[0.5rem] bg-richblack-800 p-[12px] text-richblack-5"
      />
    </label>
    <label>
      <p className="mb-1 text-[0.875rem] leading-[1.375rem] text-richblack-5">
        Last Name <sup className="text-pink-200">*</sup>
      </p>
      <input
        required
        type="text"
        name="lastName"
        value={lastName}
        onChange={handleOnChange}
        placeholder="Enter last name"
        style={{
          boxShadow: "inset 0px -1px 0px rgba(255, 255, 255, 0.18)",
        }}
      className="w-full rounded-[0.5rem] bg-richblack-800 p-[12px] text-richblack-5"
      />
    </label>
  </div>
  <label className="w-full">
    <p className="mb-1 text-[0.875rem] leading-[1.375rem] text-richblack-5">
      Email Address <sup className="text-pink-200">*</sup>
    </p>
    <input
      required
      type="text"
      name="email"
      value={email}
      onChange={handleOnChange}
      placeholder="Enter email address"
      style={{
        boxShadow: "inset 0px -1px 0px rgba(255, 255, 255, 0.18)",
      }}
      className="w-full rounded-[0.5rem] bg-richblack-800 p-[12px] text-richblack-5"
    />
  </label>
  <div className="flex gap-x-4">
```

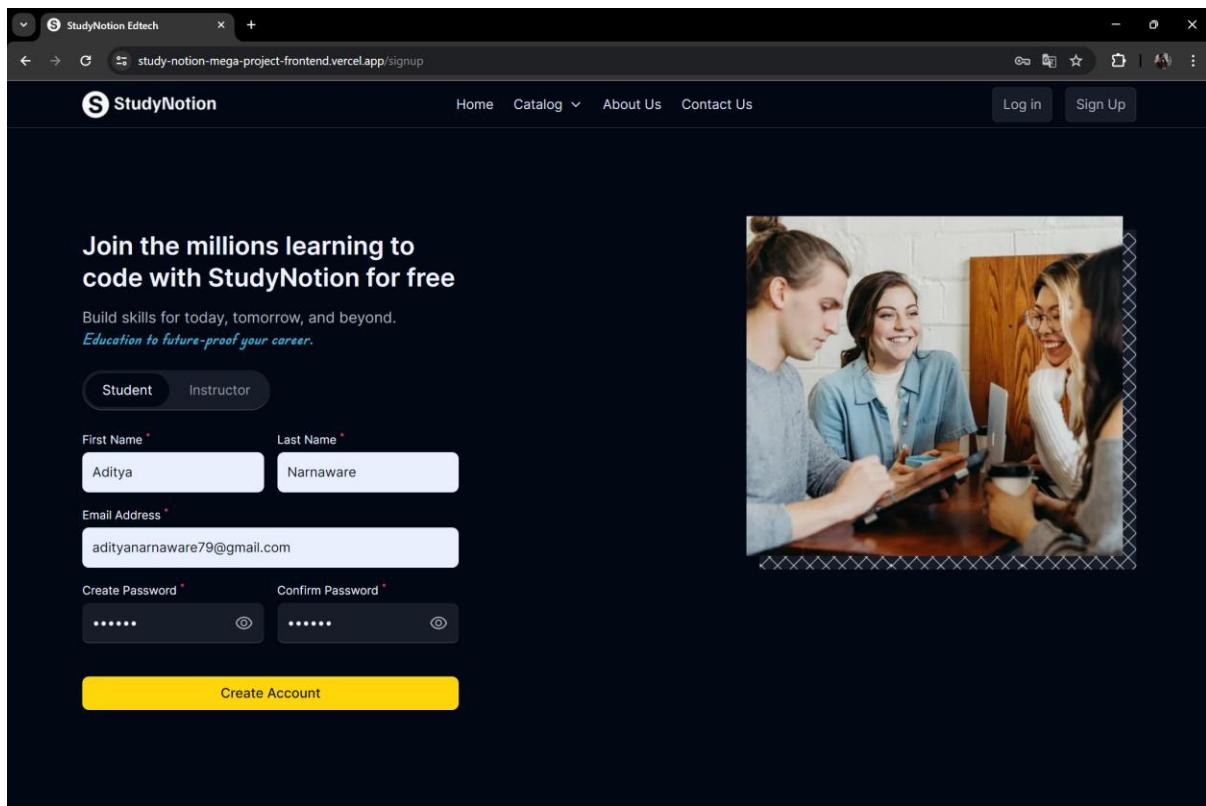
```

<label className="relative">
    <p className="mb-1 text-[0.875rem] leading-[1.375rem] text-
richblack-5">
        Create Password <sup className="text-pink-200">*</sup>
    </p>
    <input
        required
        type={showPassword ? "text" : "password"}
        name="password"
        value={password}
        onChange={handleOnChange}
        placeholder="Enter Password"
        style={{
            boxShadow: "inset 0px -1px 0px rgba(255, 255, 255,
0.18)",
        }}
        className="w-full rounded-[0.5rem] bg-richblack-800 p-
[12px] pr-10 text-richblack-5"
    />
    <span
        onClick={() => setShowPassword((prev) => !prev)}
        className="absolute right-3 top-[38px] z-[10] cursor-
pointer"
    >
        {showPassword ? (
            <AiOutlineEyeInvisible fontSize={24} fill="#AFB2BF" />
        ) : (
            <AiOutlineEye fontSize={24} fill="#AFB2BF" />
        )}
    </span>
</label>
<label className="relative">
    <p className="mb-1 text-[0.875rem] leading-[1.375rem] text-
richblack-5">
        Confirm Password <sup className="text-pink-200">*</sup>
    </p>
    <input
        required
        type={showConfirmPassword ? "text" : "password"}
        name="confirmPassword"
        value={confirmPassword}
        onChange={handleOnChange}
        placeholder="Confirm Password"
        style={{
            boxShadow: "inset 0px -1px 0px rgba(255, 255, 255,
0.18)",
        }}
    >

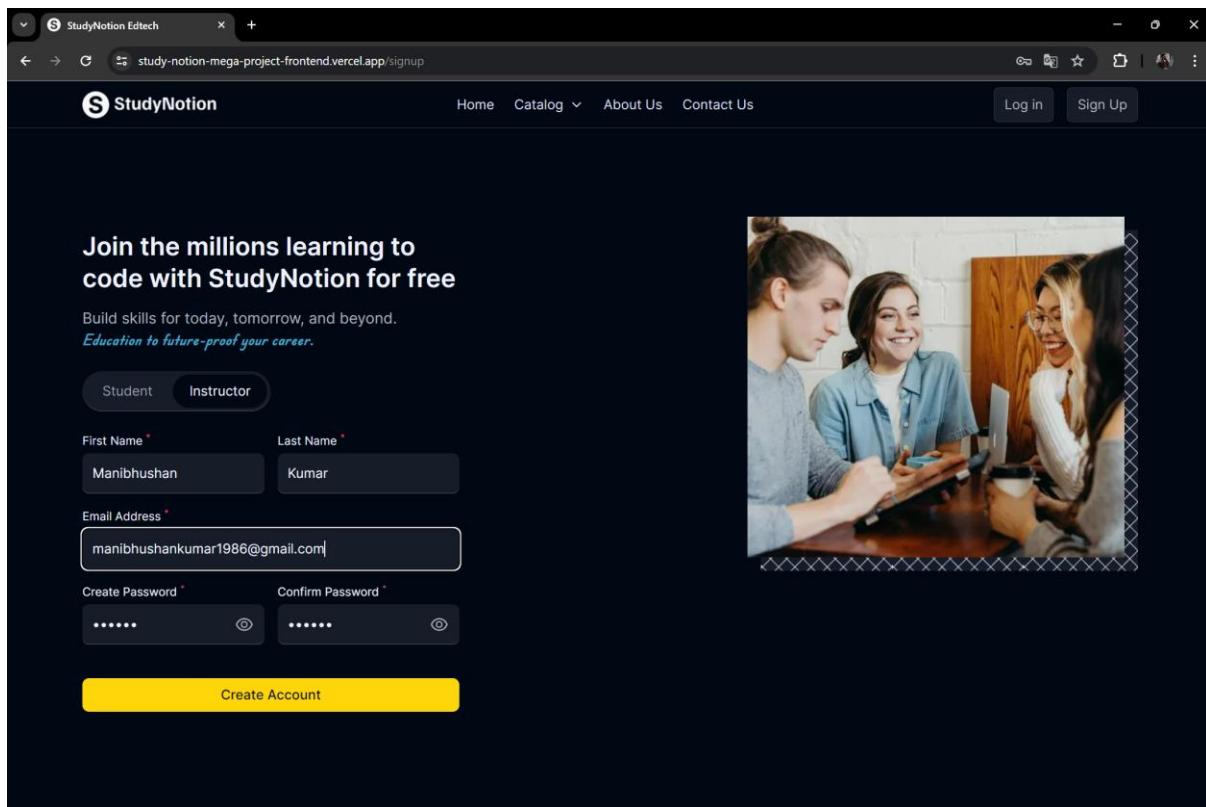
```

```
        className="w-full rounded-[0.5rem] bg-richblack-800 p-[12px] pr-10 text-richblack-5"
      />
      <span
        onClick={() => setShowConfirmPassword((prev) => !prev)}
        className="absolute right-3 top-[38px] z-[10] cursor-pointer"
      >
        {showConfirmPassword ? (
          <AiOutlineEyeInvisible fontSize={24} fill="#AFB2BF" />
        ) : (
          <AiOutlineEye fontSize={24} fill="#AFB2BF" />
        )}
      </span>
    </label>
  </div>
  <button
    type="submit"
    className="mt-6 rounded-[8px] bg-yellow-50 py-[8px] px-[12px] font-medium text-richblack-900"
  >
    Create Account
  </button>
</form>
</div>
)
}

export default SignupForm
```



### 3.4 Instructor Sign Up Page



### 3.5 Login Page of Study Notion Portal

```

import { useState } from "react"
import { AiOutlineEye, AiOutlineEyeInvisible } from "react-icons/ai"
import { useDispatch } from "react-redux"
import { Link, useNavigate } from "react-router-dom"

import { login } from "../../../../../services/operations/authAPI"

function LoginForm() {
  const navigate = useNavigate()
  const dispatch = useDispatch()
  const [formData, setFormData] = useState({
    email: "",
    password: "",
  })

  const [showPassword, setShowPassword] = useState(false)

  const { email, password } = formData

  const handleOnChange = (e) => {
    setFormData((prevData) => ({
      ...prevData,
      [e.target.name]: e.target.value,
    }))
  }

  const handleOnSubmit = (e) => {
    e.preventDefault()
    dispatch(login(email, password, navigate))
  }

  return (
    <form
      onSubmit={handleOnSubmit}
      className="mt-6 flex w-full flex-col gap-y-4"
    >
      <label className="w-full">
        <p className="mb-1 text-[0.875rem] leading-[1.375rem] text-richblack-5">
          Email Address <sup className="text-pink-200">*</sup>
        </p>
        <input
          required
          type="text"
          name="email"
        >
      </label>
    </form>
  )
}

```

```

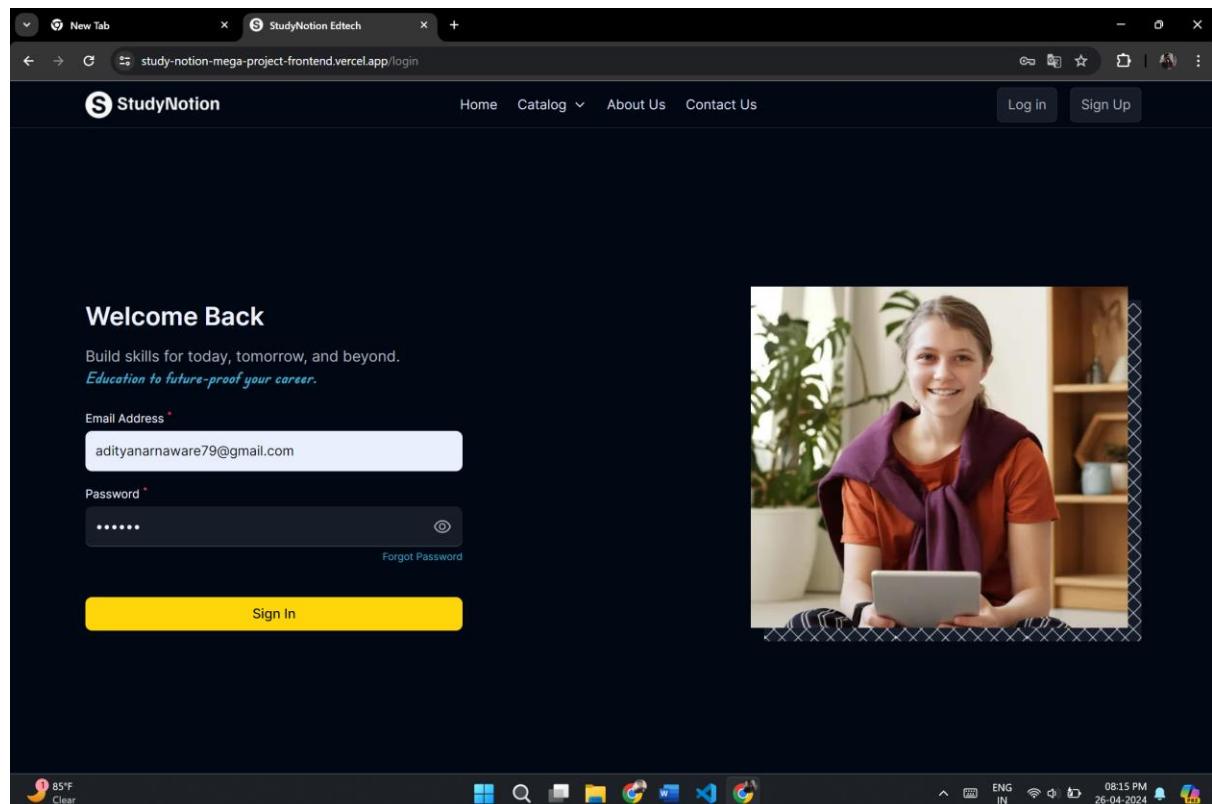
        value={email}
        onChange={handleOnChange}
        placeholder="Enter email address"
        style={{
          boxShadow: "inset 0px -1px 0px rgba(255, 255, 255, 0.18)",
        }}
        className="w-full rounded-[0.5rem] bg-richblack-800 p-[12px]
text-richblack-5"
      />
    </label>
    <label className="relative">
      <p className="mb-1 text-[0.875rem] leading-[1.375rem] text-
richblack-5">
        Password <sup className="text-pink-200">*</sup>
      </p>
      <input
        required
        type={showPassword ? "text" : "password"}
        name="password"
        value={password}
        onChange={handleOnChange}
        placeholder="Enter Password"
        style={{
          boxShadow: "inset 0px -1px 0px rgba(255, 255, 255, 0.18)",
        }}
        className="w-full rounded-[0.5rem] bg-richblack-800 p-[12px]
pr-12 text-richblack-5"
      />
      <span
        onClick={() => setShowPassword((prev) => !prev)}
        className="absolute right-3 top-[38px] z-[10] cursor-pointer"
      >
        {showPassword ? (
          <AiOutlineEyeInvisible fontSize={24} fill="#AFB2BF" />
        ) : (
          <AiOutlineEye fontSize={24} fill="#AFB2BF" />
        )}
      </span>
      <Link to="/forgot-password">
        <p className="mt-1 ml-auto max-w-max text-xs text-blue-100">
          Forgot Password
        </p>
      </Link>
    </label>
    <button
      type="submit"
      className="mt-6 rounded-[8px] bg-yellow-50 py-[8px] px-[12px]
font-medium text-richblack-900"
    >
  
```

```

    >
      Sign In
    </button>
  </form>
)
}

export default LoginForm

```



### 3.6 OTP Verification on Email

```

const otpTemplate = (otp) => {
  return `<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <title>OTP Verification Email</title>
  <style>
    body {
      background-color: #ffffff;
      font-family: Arial, sans-serif;
      font-size: 16px;
      line-height: 1.4;
    }
  </style>
</head>
<body>
  <h1>OTP Verification</h1>
  <p>Your OTP is: <b>${otp}</b></p>
  <p>Please enter this OTP on the app to verify your account.</p>
</body>
</html>`;
}

```

```
        color: #333333;
        margin: 0;
        padding: 0;
    }

    .container {
        max-width: 600px;
        margin: 0 auto;
        padding: 20px;
        text-align: center;
    }

    .logo {
        max-width: 200px;
        margin-bottom: 20px;
    }

    .message {
        font-size: 18px;
        font-weight: bold;
        margin-bottom: 20px;
    }

    .body {
        font-size: 16px;
        margin-bottom: 20px;
    }

    .cta {
        display: inline-block;
        padding: 10px 20px;
        background-color: #FFD60A;
        color: #000000;
        text-decoration: none;
        border-radius: 5px;
        font-size: 16px;
        font-weight: bold;
        margin-top: 20px;
    }

    .support {
        font-size: 14px;
        color: #999999;
        margin-top: 20px;
    }

    .highlight {
        font-weight: bold;
```

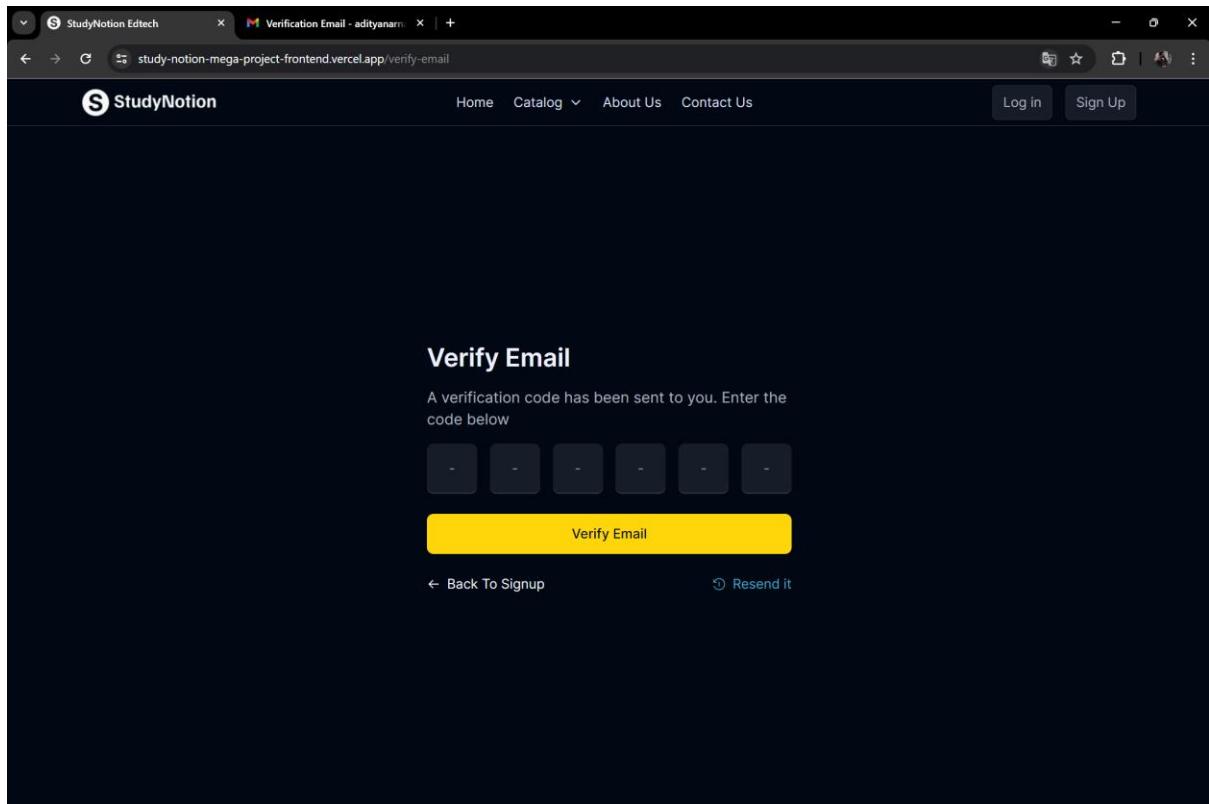
```
        }
    </style>

</head>

<body>
    <div class="container">
        <a href="https://studynotion-edtech-project.vercel.app"></a>
        <div class="message">OTP Verification Email</div>
        <div class="body">
            <p>Dear User,</p>
            <p>Thank you for registering with StudyNotion. To
complete your registration, please use the following OTP
                (One-Time Password) to verify your account:</p>
            <h2 class="highlight">${otp}</h2>
            <p>This OTP is valid for 5 minutes. If you did not
request this verification, please disregard this email.
            Once your account is verified, you will have access to
our platform and its features.</p>
        </div>
        <div class="support">If you have any questions or need
assistance, please feel free to reach out to us at <a
            href="mailto:info@studynotion.com">info@studynotion.c
om</a>. We are here to help!</div>
    </div>
</body>

</html>';
};

module.exports = otpTemplate;
```



### 3.7 Getting OTP verification code on E-mail

```
const mongoose = require("mongoose");
const mailSender = require("../utils/mailSender");
const emailTemplate =
require("../mail/templates/emailVerificationTemplate");

const OTPSchema = new mongoose.Schema({
  email: {
    type: String,
    required: true,
  },
  otp: {
    type: String,
    required: true,
  },
  createdAt: {
    type: Date,
    default: Date.now(),
    expires: 5 * 60, // The document will be automatically deleted after
5 minutes of its creation time
  },
});
```

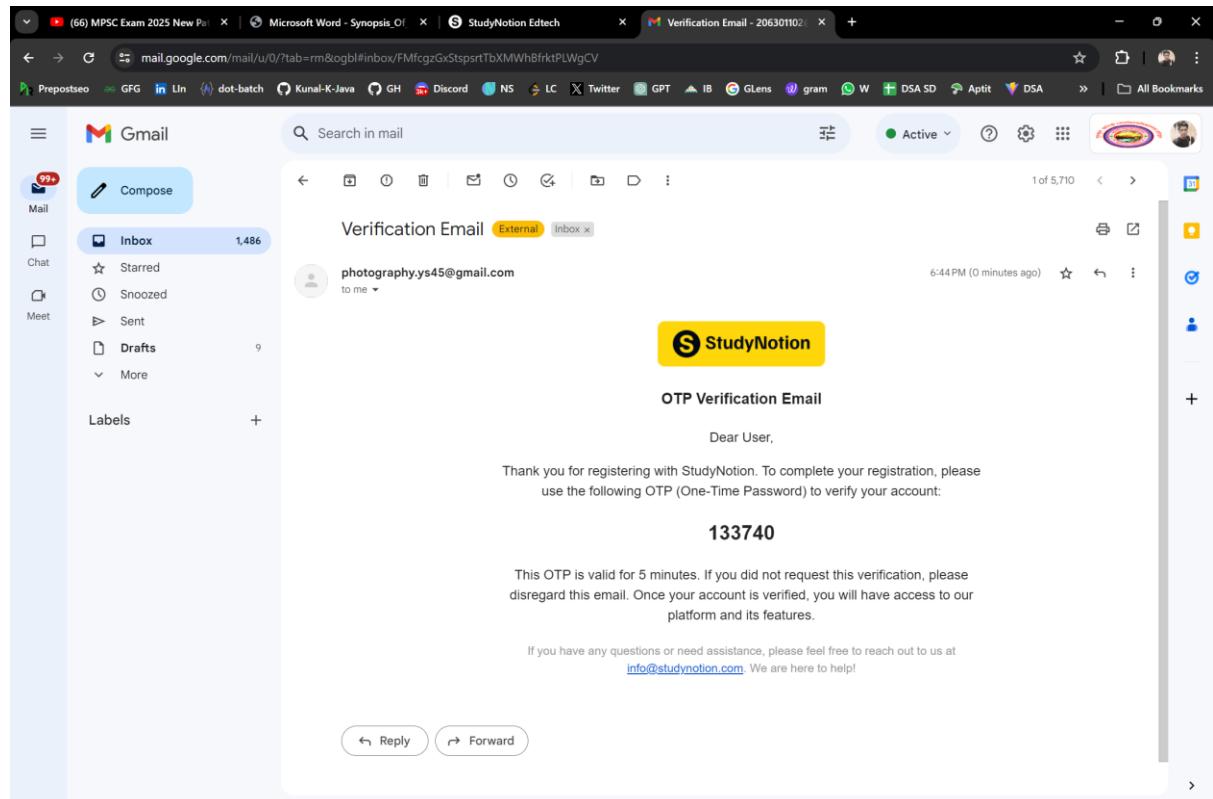
```

async function sendVerificationEmail(email, otp) {
  try {
    const mailResponse = await mailSender(
      email,
      "Verification Email from StudyNotion",
      emailTemplate(otp)
    );
  } catch (error) {
    console.log("error occurred while sending mail:", error);
    throw error;
  }
}

OTPSchema.pre("save", async function (next) {
  await sendVerificationEmail(this.email, this.otp);
  next();
});

module.exports = mongoose.model("OTP", OTPSchema);

```



### 3.8 Home page of Study Notion Portal

```

import { useSelector } from "react-redux"
import { Outlet } from "react-router-dom"

import Sidebar from "../components/core/Dashboard/Sidebar"

function Dashboard() {
    const { loading: profileLoading } = useSelector((state) =>
state.profile)
    const { loading: authLoading } = useSelector((state) => state.auth)

    if (profileLoading || authLoading) {
        return (
            <div className="grid min-h-[calc(100vh-3.5rem)] place-items-center">
                <div className="spinner"></div>
            </div>
        )
    }

    return (
        <div className="relative flex min-h-[calc(100vh-3.5rem)]">
            <Sidebar />
            <div className="h-[calc(100vh-3.5rem)] flex-1 overflow-auto">
                <div className="mx-auto w-11/12 max-w-[1000px] py-10">
                    <Outlet />
                </div>
            </div>
        </div>
    )
}

export default Dashboard

```

```

import React, { useEffect, useState } from "react";
import { useSelector } from "react-redux";
import { fetchInstructorCourses } from
"../../../../services/operations/courseDetailsAPI";
import { getInstructorData } from
"../../../../services/operations/profileAPI";
import InstructorChart from "./InstructorChart";
import { Link } from "react-router-dom";

export default function Instructor() {

```

```

const { token } = useSelector((state) => state.auth);
const { user } = useSelector((state) => state.profile);
const [loading, setLoading] = useState(false);
const [instructorData, setInstructorData] = useState(null);
const [courses, setCourses] = useState([]);

useEffect(() => {
  (async () => {
    setLoading(true);
    const instructorApiData = await getInstructorData(token);

    console.log("instructorApiData----->>", instructorApiData);

    const result = await fetchInstructorCourses(token);

    console.log(
      "          const result = await fetchInstructorCourses ApiData----->>",
      result
    );

    if (instructorApiData.length) setInstructorData(instructorApiData);
    if (result) {
      setCourses(result);
    }
    setLoading(false);
  })();
}, []);

console.log("instructorData useState----->", instructorData);

const totalAmount = instructorData?.reduce((acc, curr) => acc +
curr.totalAmountGenerated, 0);
// console.log("instructorData?.reduce()-----",
instructorData?.reduce());

console.log("totalAmountGenerated----->", totalAmount);

const totalStudents = instructorData?.reduce(
  (acc, curr) => acc + curr.totalStudentsEnrolled,
  0
);
console.log("totalStudentsEnrolled----->", totalStudents);

return (
  <div>
    <div className="space-y-2">
      <h1 className="text-2xl font-bold text-richblack-5">

```

```
    Hi {user?.firstName} 🙌
  </h1>
  <p className="font-medium text-richblack-200">
    Let's start something new
  </p>
</div>
{loading ? (
  <div className="spinner"></div>
) : courses.length > 0 ? (
  <div>
    <div className="my-4 flex h-[450px] space-x-4">
      {/* Render chart / graph */}
      {totalAmount > 0 || totalStudents > 0 ? (
        <InstructorChart courses={instructorData} />
      ) : (
        <div className="flex-1 rounded-md bg-richblack-800 p-6">
          <p className="text-lg font-bold text-richblack-50">Visualize</p>
          <p className="mt-4 text-xl font-medium text-richblack-50">
            Not Enough Data To Visualize
          </p>
        </div>
      )}
      {/* Total Statistics */}
      <div className="flex min-w-[250px] flex-col rounded-md bg-richblack-800 p-6">
        <p className="text-lg font-bold text-richblack-50">Statistics</p>
        <div className="mt-4 space-y-4">
          <div>
            <p className="text-lg text-richblack-200">Total Courses</p>
            <p className="text-3xl font-semibold text-richblack-50">
              {courses.length}
            </p>
          </div>
          <div>
            <p className="text-lg text-richblack-200">Total Students</p>
            <p className="text-3xl font-semibold text-richblack-50">
              {totalStudents}
            </p>
          </div>
          <div>

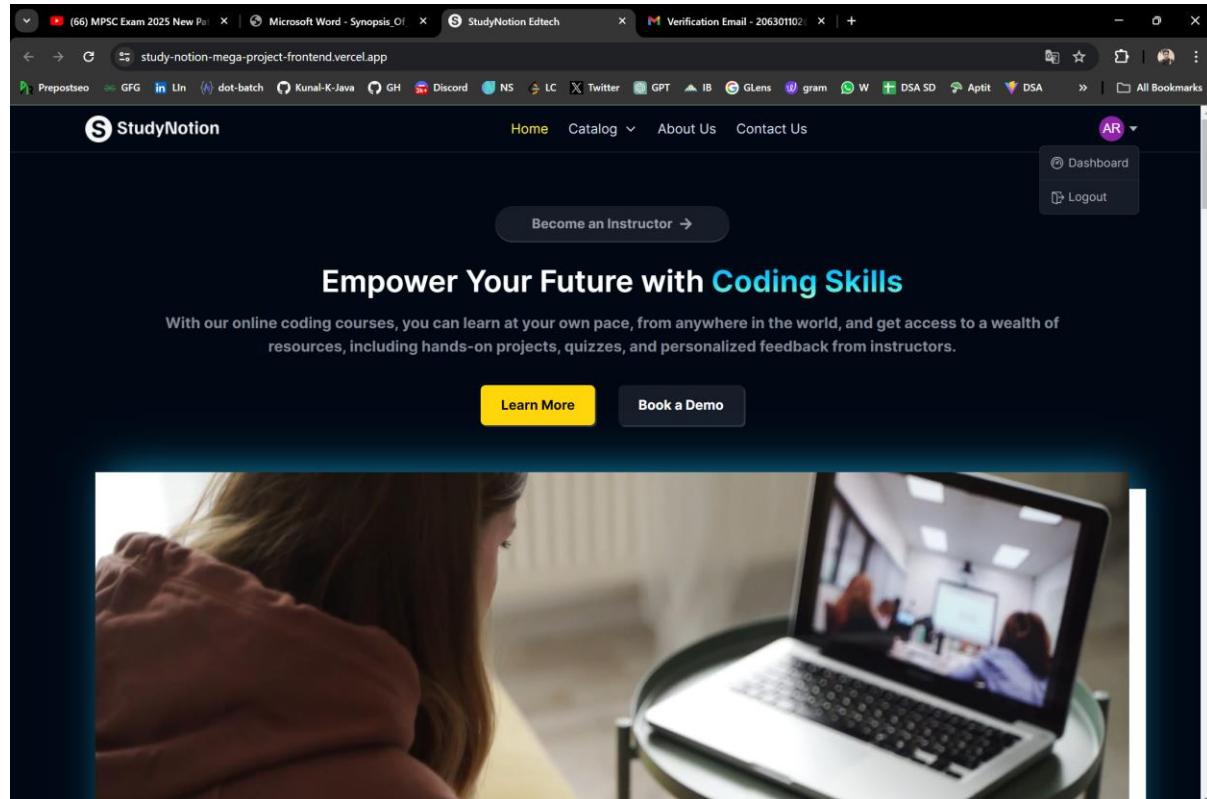
```

```

        <p className="text-lg text-richblack-200">Total
Income</p>
        <p className="text-3xl font-semibold text-richblack-
50">
            Rs. {totalAmount}
        </p>
    </div>
</div>
</div>
</div>
<div className="rounded-md bg-richblack-800 p-6">
    {/* Render 3 courses */}
    <div className="flex items-center justify-between">
        <p className="text-lg font-bold text-richblack-5">Your
Courses</p>
        <Link to="/dashboard/my-courses">
            <p className="text-xs font-semibold text-yellow-50">View
All</p>
        </Link>
    </div>
    <div className="my-4 flex items-start space-x-6">
        {courses.slice(0, 3).map((course) => (
            <div key={course._id} className="w-1/3">
                <img
                    src={course.thumbnail}
                    alt={course.courseName}
                    className="h-[201px] w-full rounded-md object-cover"
                />
                <div className="mt-3 w-full">
                    <p className="text-sm font-medium text-richblack-50">
                        {course.courseName}
                    </p>
                    <div className="mt-1 flex items-center space-x-2">
                        <p className="text-xs font-medium text-richblack-
300">
                            {course?.studentsEnroled?.length} students
                        </p>
                        <p className="text-xs font-medium text-richblack-
300">
                            |
                        </p>
                        <p className="text-xs font-medium text-richblack-
300">
                            Rs. {course.price}
                        </p>
                    </div>
                </div>
            </div>
        ))
    </div>

```

```
        })
      </div>
    </div>
  </div>
) : (
  <div className="mt-20 rounded-md bg-richblack-800 p-6 py-20">
    <p className="text-center text-2xl font-bold text-richblack-5">
      You have not created any courses yet
    </p>
    <Link to="/dashboard/add-course">
      <p className="mt-1 text-center text-lg font-semibold text-yellow-50">
        Create a course
      </p>
      </Link>
    </div>
)
</div>
);
}
```



### 3.9 My Profile Section

```

import { RiEditBoxLine } from "react-icons/ri";
import { useSelector } from "react-redux";
import { useNavigate } from "react-router-dom";

import { formattedDate } from "../../utils/dateFormatter";
import IconButton from "../../common/IconButton";

export default function MyProfile() {
  const { user } = useSelector((state) => state.profile);
  const navigate = useNavigate();

  return (
    <>
      <h1 className="mb-14 text-3xl font-medium text-richblack-5">
        My Profile
      </h1>
      <div className="flex items-center justify-between rounded-md border-[1px] border-richblack-700 bg-richblack-800 p-8 px-12">
        <div className="flex items-center gap-x-4">
          <img
            src={user?.image}
            alt={`profile-${user?.firstName}`}
            className="aspect-square w-[78px] rounded-full object-cover"
          />
          <div className="space-y-1">
            <p className="text-lg font-semibold text-richblack-5">
              {user?.firstName + " " + user?.lastName}
            </p>
            <p className="text-sm text-richblack-300">{user?.email}</p>
          </div>
        </div>
        <IconButton
          text="Edit"
          onClick={() => {
            navigate("/dashboard/settings");
          }}
        >
          <RiEditBoxLine />
        </IconButton>
      </div>
      <div className="my-10 flex flex-col gap-y-10 rounded-md border-[1px] border-richblack-700 bg-richblack-800 p-8 px-12">
        <div className="flex w-full items-center justify-between">
          <p className="text-lg font-semibold text-richblack-5">About</p>
          <IconButton

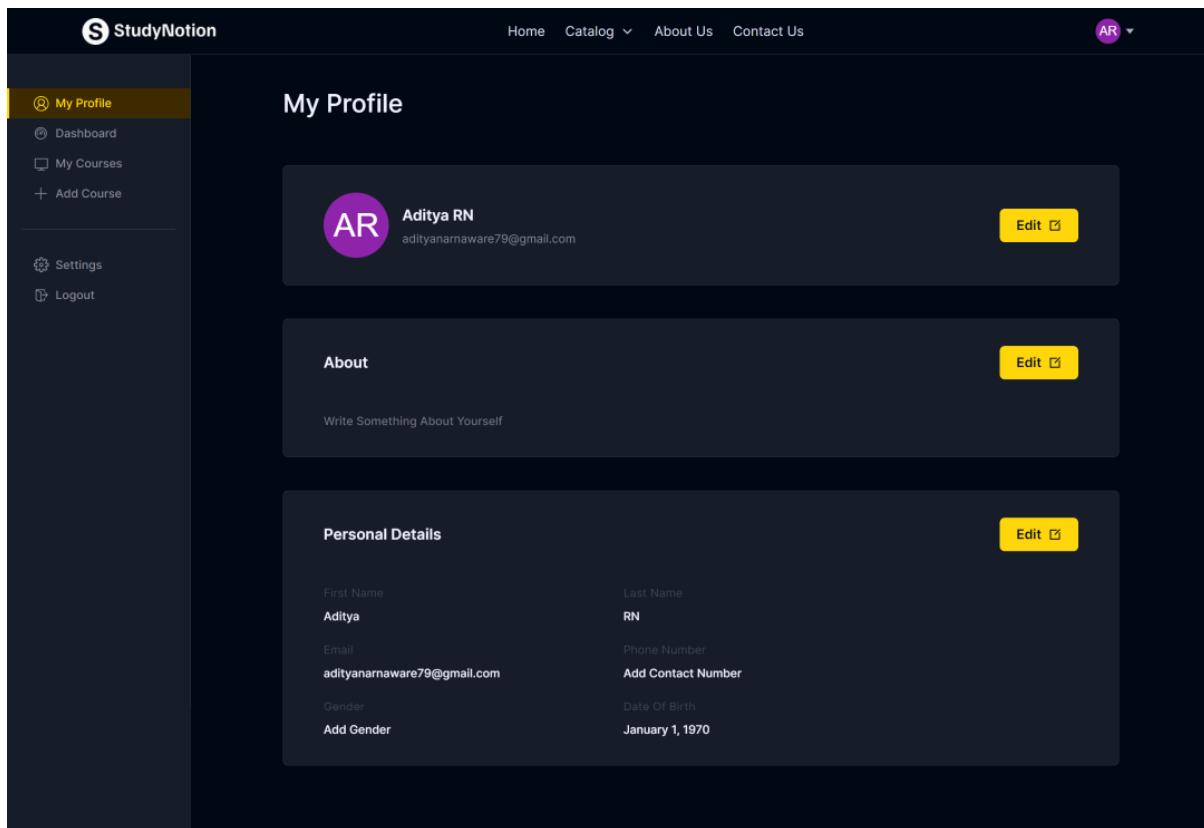
```

```

        text="Edit"
        onClick={() => {
          navigate("/dashboard/settings");
        }}
      >
    <RiEditBoxLine />
</IconBtn>
</div>
<p
  className={`${${
    user?.additionalDetails?.about
      ? "text-richblack-5"
      : "text-richblack-400"
  } text-sm font-medium`}
>
  {user?.additionalDetails?.about ?? "Write Something About Yourself"}
</p>
</div>
<div className="my-10 flex flex-col gap-y-10 rounded-md border-[1px] border-richblack-700 bg-richblack-800 p-8 px-12">
  <div className="flex w-full items-center justify-between">
    <p className="text-lg font-semibold text-richblack-5">
      Personal Details
    </p>
    <IconBtn
      text="Edit"
      onClick={() => {
        navigate("/dashboard/settings");
      }}
    >
      <RiEditBoxLine />
    </IconBtn>
  </div>
  <div className="flex max-w-[500px] justify-between">
    <div className="flex flex-col gap-y-5">
      <div>
        <p className="mb-2 text-sm text-richblack-600">First Name</p>
        <p className="text-sm font-medium text-richblack-5">
          {user?.firstName}
        </p>
      </div>
      <div>
        <p className="mb-2 text-sm text-richblack-600">Email</p>
        <p className="text-sm font-medium text-richblack-5">
          {user?.email}
        </p>
      </div>
    </div>
  </div>
</div>

```

```
</div>
<div>
    <p className="mb-2 text-sm text-richblack-600">Gender</p>
    <p className="text-sm font-medium text-richblack-5">
        {user?.additionalDetails?.gender ?? "Add Gender"}
    </p>
</div>
</div>
<div className="flex flex-col gap-y-5">
    <div>
        <p className="mb-2 text-sm text-richblack-600">Last
Name</p>
        <p className="text-sm font-medium text-richblack-5">
            {user?.lastName}
        </p>
    </div>
    <div>
        <p className="mb-2 text-sm text-richblack-600">Phone
Number</p>
        <p className="text-sm font-medium text-richblack-5">
            {user?.additionalDetails?.contactNumber ?? "Add Contact
Number"}
        </p>
    </div>
    <div>
        <p className="mb-2 text-sm text-richblack-600">Date Of
Birth</p>
        <p className="text-sm font-medium text-richblack-5">
            {formattedDate(user?.additionalDetails?.dateOfBirth) ??
                "Add Date Of Birth"}
        </p>
    </div>
    </div>
</div>
</div>
</>
);
}
```



### 3.10 Add Course Section

```

import { useState } from "react";
import { useForm } from "react-hook-form";
import { toast } from "react-hot-toast";
import { IoAddCircleOutline } from "react-icons/io5";
import { MdNavigateNext } from "react-icons/md";
import { useDispatch, useSelector } from "react-redux";

import {
  createSection,
  updateSection,
} from "../../../../services/operations/courseDetailsAPI";
import {
  setCourse,
  setEditCourse,
  setStep,
} from "../../../../slices/courseSlice";
import IconBtn from "../../../../common/IconBtn";
import NestedView from "./NestedView";

export default function CourseBuilderForm() {
  const {
    register,
  
```

```
handleSubmit,
setValue,
formState: { errors },
} = useForm();

const { course } = useSelector((state) => state.course);
const { token } = useSelector((state) => state.auth);
const [loading, setLoading] = useState(false);
const [editSectionName, setEditSectionName] = useState(null);
const dispatch = useDispatch();

// handle form submission
const onSubmit = async (data) => {
  // console.log(data)
  setLoading(true);

  let result;

  if (editSectionName) {
    result = await updateSection(
      {
        sectionName: data.sectionName,
        sectionId: editSectionName,
        courseId: course._id,
      },
      token
    );
    // console.log("edit", result)
  } else {
    result = await createSection(
      {
        sectionName: data.sectionName,
        courseId: course._id,
      },
      token
    );
  }
  if (result) {
    // console.log("section result", result)
    dispatch(setCourse(result));
    setEditSectionName(null);
    setValue("sectionName", "");
  }
  setLoading(false);
};

const cancelEdit = () => {
  setEditSectionName(null);
```

```

        setValue("sectionName", "");
    };

    const handleChangeEditSectionName = (sectionId, sectionName) => {
        if (editSectionName === sectionId) {
            cancelEdit();
            return;
        }
        setEditSectionName(sectionId);
        setValue("sectionName", sectionName);
    };

    const goToNext = () => {
        if (course.courseContent.length === 0) {
            toast.error("Please add atleast one section");
            return;
        }
        if (
            course.courseContent.some((section) => section.subSection.length
            === 0)
        ) {
            toast.error("Please add atleast one lecture in each section");
            return;
        }
        dispatch(setStep(3));
    };

    const goBack = () => {
        dispatch(setStep(1));
        dispatch(setEditCourse(true));
    };

    return (
        <div className="space-y-8 rounded-md border-[1px] border-richblack-700 bg-richblack-800 p-6">
            <p className="text-2xl font-semibold text-richblack-5">Course
        Builder</p>
            <form onSubmit={handleSubmit(onSubmit)} className="space-y-4">
                <div className="flex flex-col space-y-2">
                    <label className="text-sm text-richblack-5"
                        htmlFor="sectionName">
                        Section Name <sup className="text-pink-200">*</sup>
                    </label>
                    <input
                        id="sectionName"
                        disabled={loading}
                        placeholder="Add a section to build your course"
                        {...register("sectionName", { required: true })}>
                </div>
            </form>
        </div>
    );
}

export default CourseBuilder;
```

```

        className="form-style w-full"
    />
    {errors.sectionName && (
        <span className="ml-2 text-xs tracking-wide text-pink-200">
            Section name is required
        </span>
    )}
</div>
<div className="flex items-end gap-x-4">
    <IconBtn
        type="submit"
        disabled={loading}
        text={editSectionName ? "Edit Section Name" : "Create
Section"}
        outline={true}
    >
        <IoAddCircleOutline size={20} className="text-yellow-50" />
    </IconBtn>
    {editSectionName && (
        <button
            type="button"
            onClick={cancelEdit}
            className="text-sm text-richblack-300 underline"
        >
            Cancel Edit
        </button>
    )}
</div>
</form>
{course.courseContent.length > 0 && (
    <NestedView
        handleChangeEditSectionName={handleChangeEditSectionName} />
    )
    /* Next Prev Button */
    <div className="flex justify-end gap-x-3">
        <button
            onClick={goBack}
            className={`${`flex cursor-pointer items-center gap-x-2 rounded-md
bg-richblack-300 py-[8px] px-[20px] font-semibold text-richblack-900`}`}
        >
            Back
        </button>
        <IconBtn disabled={loading} text="Next" onclick={goToNext}>
            <MdNavigateNext />
        </IconBtn>
    </div>
</div>
);

```



**StudyNotion**

Home Catalog ▾ About Us Contact Us AR ▾

My Profile

Dashboard

My Courses

**+ Add Course**

Settings

Logout

## Add Course

1 Course Information      2 Course Builder      3 Publish

**Course Title \***  
Enter Course Title

**Course Short Description \***  
Enter Description

**Course Price \***  
Enter Course Price

**Category \***  
Choose a category

**Tags \***  
Enter Tags and Press Enter

**Course Thumbnail \***  
Drag and drop an image, or click to **Browse** a file  
Aspect ratio 16:9   Recommended size 1024x576

**Benefits of the course \***  
Enter benefits of the course

**Requirements/Instructions \***  
**Add**

**Next >**

**Course Upload Tips**

- Set the Course Price option or make it free.
- Standard size for the course thumbnail is 1024x576.
- Video section controls the course overview video.
- Course Builder is where you create & organize a course.
- Add Topics in the Course Builder section to create lessons, quizzes, and assignments.
- Information from the Additional Data section shows up on the course single page.
- Make Announcements to notify any important notes to all enrolled students at once.

### 3.11 Edit Profile Section

```

import { toast } from "react-hot-toast";

import { setLoading, setUser } from "../../slices/profileSlice";
import { apiConnector } from "../apiconnector";
import { profileEndpoints } from "../apis";
import { logout } from "./authAPI";

const {
  GET_USER_DETAILS_API,
  GET_USER_ENROLLED_COURSES_API,
  GET_INSTRUCTOR_DATA_API,
} = profileEndpoints;

export function getUserDetails(token, navigate) {
  return async (dispatch) => {
    const toastId = toast.loading("Loading...");
    dispatch(setLoading(true));
    try {
      const response = await apiConnector("GET", GET_USER_DETAILS_API, null, {
        Authorization: `Bearer ${token}`,
      });
      console.log("GET_USER_DETAILS API RESPONSE.....", response);

      if (!response.data.success) {
        throw new Error(response.data.message);
      }
      const userImage = response.data.data.image
        ? response.data.data.image
        :
`https://api.dicebear.com/5.x/initials/svg?seed=${response.data.data.firstName} ${response.data.data.lastName}`;
      dispatch(setUser({ ...response.data.data, image: userImage }));
    } catch (error) {
      dispatch(logout(navigate));
      console.log("GET_USER_DETAILS API ERROR.....", error);
      toast.error("Could Not Get User Details");
    }
    toast.dismiss(toastId);
    dispatch(setLoading(false));
  };
}

export async function getUserEnrolledCourses(token) {
  const toastId = toast.loading("Loading...");
}

```

```

let result = [];
try {
  console.log("BEFORE Calling BACKEND API FOR ENROLLED COURSES");
  const response = await apiConnector(
    "GET",
    GET_USER_ENROLLED_COURSES_API,
    null,
    {
      Authorization: `Bearer ${token}`,
    }
  );
  console.log("AFTER Calling BACKEND API FOR ENROLLED COURSES");
  // console.log(
  //   "GET_USER_ENROLLED_COURSES_API API RESPONSE.....",
  //   response
  // )

  if (!response.data.success) {
    throw new Error(response.data.message);
  }
  result = response.data.data;

  console.log("result enrolled courses api---->", result);
} catch (error) {
  console.log("GET_USER_ENROLLED_COURSES_API API ERROR.....",
error);
  toast.error("Could Not Get Enrolled Courses");
}
toast.dismiss(toastId);
return result;
}

export async function getInstructorData(token) {
  console.log("Before from profileApi token ----->", token);
  const toastId = toast.loading("Loading...");
  let result = [];
  try {
    const response = await apiConnector("GET", GET_INSTRUCTOR_DATA_API,
null, {
      Authorization: `Bearer ${token}`,
    });

    console.log("GET_INSTRUCTOR_API_RESPONSE", response);
    console.log("After from profileApi token ----->", token);
    result = response?.data?.courses;
  } catch (error) {
    console.log("GET_INSTRUCTOR_API ERROR", error);
    toast.error("Could not Get Instructor Data");
  }
}

```

```
}
```

```
toast.dismiss(toastId);
```

```
return result;
```

```
}
```

The screenshot shows the 'Edit Profile' screen of the StudyNotion app. At the top, there's a navigation bar with the app logo, 'StudyNotion', and links for 'Home', 'Catalog', 'About Us', 'Contact Us', and an 'AR' button. On the left is a vertical sidebar with options: 'My Profile', 'Dashboard', 'My Courses', 'Add Course', 'Settings' (which is selected and highlighted in yellow), and 'Logout'. The main content area has a dark background with light-colored input fields. It includes a section for 'Change Profile Picture' with a placeholder image of a person and 'Select' and 'Upload' buttons. Below this is the 'Profile Information' section with fields for 'First Name' (Aditya), 'Last Name' (RN), 'Date of Birth' (dd-mm-yyyy), 'Gender' (Male), 'Contact Number' (placeholder: Enter Contact Number), and 'About' (placeholder: Enter Bio Details). At the bottom of this section are 'Cancel' and 'Save' buttons. A separate 'Password' overlay is shown below, containing fields for 'Current Password' and 'New Password' with 'Enter Current Password' and 'Enter New Password' placeholders. It also has 'Cancel' and 'Update' buttons. At the very bottom is a dark red overlay titled 'Delete Account' with a trash can icon, asking if the user wants to delete their account, stating that it's permanent and removes associated paid courses. It includes a link 'I want to delete my account.' and 'Cancel' and 'Update' buttons.

### 3.12 DevOps Course List

```

import { useSelector } from "react-redux"
import { Table, Tbody, Td, Th, Thead, Tr } from "react-super-responsive-table"

// import { setCourse, setEditCourse } from
// "../../../slices/courseSlice"
import "react-super-responsive-table/dist/SuperResponsiveTableStyle.css"
import { useState } from "react"
import { FaCheck } from "react-icons/fa"
import { FiEdit2 } from "react-icons/fi"
import { HiClock } from "react-icons/hi"
import { RiDeleteBin6Line } from "react-icons/ri"
import { useNavigate } from "react-router-dom"

import { formatDate } from "../../../services/formatDate"
import {
  deleteCourse,
  fetchInstructorCourses,
} from "../../../services/operations/courseDetailsAPI"
import { COURSE_STATUS } from "../../../utils/constants"
import ConfirmationModal from "../../../common/ConfirmationModal"

export default function CoursesTable({ courses, setCourses }) {
  // const dispatch = useDispatch()
  const navigate = useNavigate()
  const { token } = useSelector((state) => state.auth)
  const [loading, setLoading] = useState(false)
  const [confirmationModal, setConfirmationModal] = useState(null)
  const TRUNCATE_LENGTH = 30

  const handleCourseDelete = async (courseId) => {
    setLoading(true)
    await deleteCourse({ courseId: courseId }, token)
    const result = await fetchInstructorCourses(token)
    if (result) {
      setCourses(result)
    }
    setConfirmationModal(null)
    setLoading(false)
  }

  // console.log("All Course ", courses)

  return (
    <>

```

```

<Table className="rounded-xl border border-richblack-800">
  <Thead>
    <Tr className="flex gap-x-10 rounded-t-md border-b border-b-richblack-800 px-6 py-2">
      <Th className="flex-1 text-left text-sm font-medium uppercase text-richblack-100">
        Courses
      </Th>
      <Th className="text-left text-sm font-medium uppercase text-richblack-100">
        Duration
      </Th>
      <Th className="text-left text-sm font-medium uppercase text-richblack-100">
        Price
      </Th>
      <Th className="text-left text-sm font-medium uppercase text-richblack-100">
        Actions
      </Th>
    </Tr>
  </Thead>
  <Tbody>
    {courses?.length === 0 ? (
      <Tr>
        <Td className="py-10 text-center text-2xl font-medium text-richblack-100">
          No courses found
          {/* TODO: Need to change this state */}
        </Td>
      </Tr>
    ) : (
      courses?.map((course) => (
        <Tr
          key={course._id}
          className="flex gap-x-10 border-b border-richblack-800 px-6 py-8"
        >
          <Td className="flex flex-1 gap-x-4">
            <img
              src={course?.thumbnail}
              alt={course?.courseName}
              className="h-[148px] w-[220px] rounded-lg object-cover"
            />
            <div className="flex flex-col justify-between">
              <p className="text-lg font-semibold text-richblack-5">

```

```

        {course.courseName}
    </p>
    <p className="text-xs text-richblack-300">
        {course.courseDescription.split(" ").length >
        TRUNCATE_LENGTH
            ? course.courseDescription
                .split(" ")
                .slice(0, TRUNCATE_LENGTH)
                .join(" ") + "..."
            : course.courseDescription}
    </p>
    <p className="text-[12px] text-white">
        Created: {formatDate(course.createdAt)}
    </p>
    {course.status === COURSE_STATUS.DRAFT ? (
        <p className="flex w-fit flex-row items-center gap-
2 rounded-full bg-richblack-700 px-2 py-[2px] text-[12px] font-medium
text-pink-100">
            <HiClock size={14} />
            Drafted
        </p>
    ) : (
        <p className="flex w-fit flex-row items-center gap-
2 rounded-full bg-richblack-700 px-2 py-[2px] text-[12px] font-medium
text-yellow-100">
            <div className="flex h-3 w-3 items-center
justify-center rounded-full bg-yellow-100 text-richblack-700">
                <FaCheck size={8} />
            </div>
            Published
        </p>
    )}
</div>
</Td>
<Td className="text-sm font-medium text-richblack-100">
    2hr 30min
</Td>
<Td className="text-sm font-medium text-richblack-100">
    ₹{course.price}
</Td>
<Td className="text-sm font-medium text-richblack-100 ">
    <button
        disabled={loading}
        onClick={() => {
            navigate(`/dashboard/edit-course/${course._id}`)
        }}
        title="Edit"
    </button>
</Td>

```

```

        className="px-2 transition-all duration-200
hover:scale-110 hover:text-caribbeangreen-300"
      >
      <FiEdit2 size={20} />
    </button>
    <button
      disabled={loading}
      onClick={() => {
        setConfirmationModal({
          text1: "Do you want to delete this course?",
          text2:
            "All the data related to this course will be
deleted",
          btn1Text: !loading ? "Delete" : "Loading...  ",
          btn2Text: "Cancel",
          btn1Handler: !loading
            ? () => handleCourseDelete(course._id)
            : () => {},
          btn2Handler: !loading
            ? () => setConfirmationModal(null)
            : () => {},
        })
      }}
      title="Delete"
      className="px-1 transition-all duration-200
hover:scale-110 hover:text-[#ff0000]"
      >
      <RiDeleteBin6Line size={20} />
    </button>
  </Td>
</Tr>
)
)
)
</Tbody>
</Table>
{confirmationModal && <ConfirmationModal
modalData={confirmationModal} />}
</>
)
}

```

**StudyNotion**

Home Catalog About Us Contact Us AR

Home / Catalog / DevOps

## DevOps

DevOps is a combination of software development (dev) and operations (ops). It is defined as a software engineering methodology which aims to integrate the work of development teams and operations teams by facilitating a culture of collaboration and shared responsibility.

### Courses to get you started

Most Popular New

**DEVOPS, ZERO TO HERO**

0 ★ ★ ★ ★ 0 Ratings  
999

**DEVOPS ADVANCED**

0 ★ ★ ★ ★ 0 Ratings  
2500

**DevOps - The Complete Guide**

0 ★ ★ ★ ★ 0 Ratings  
950

### Checkout Web Development Courses Also

**NEXT JS & TYPESCRIPT**

3 ★ ★ ★ ★ 1 Ratings  
1100

**TYPESCRIPT**

5 ★ ★ ★ ★ 1 Ratings  
1250

**The Complete 2023 Web Development Bootcamp**

0 ★ ★ ★ ★ 0 Ratings  
1199

### Most Selling Courses

**DEVOPS 101**

5 ★ ★ ★ ★ 1 Ratings  
1250

**ANDROID MADE EASY**

4 ★ ★ ★ ★ 1 Ratings  
999

**NEXT JS & TYPESCRIPT**

3 ★ ★ ★ ★ 1 Ratings  
1100

**TYPESCRIPT**

5 ★ ★ ★ ★ 1 Ratings  
1250

---

**StudyNotion**

**Company**

- About
- Careers
- Affiliates

[Facebook](#) [Twitter](#) [Instagram](#) [YouTube](#)

**Resources**

- Articles
- Blog
- Chart Sheet
- Code challenges
- Docs
- Projects
- Videos
- Workspaces

**Plans**

- Paid memberships
- For students
- Business solutions

**Community**

- Forums
- Chapters
- Events

**Subjects**

- AI
- Cloud Computing
- Code Foundations
- Computer Science
- Cybersecurity
- Data Analytics
- Data Science
- Data Visualization
- Developer Tools
- DevOps
- IT
- Machine Learning
- Math
- Mobile Development
- Web Design
- Web Development

**Languages**

- Bash
- C++
- C#
- Go
- HTML & CSS
- Java
- JavaScript
- Kotlin
- PHP
- Python
- R
- Ruby
- SQL
- Swift

**Career building**

- Career paths
- Career services
- Interview prep
- Professional certification
- Full Catalog
- Beta Content

Privacy Policy | Cookie Policy | Terms

Made with ❤️ by CodeHelp © 2023 Studynotion

### 3.13 Web Development Course List

```

import React, { useEffect, useState } from "react";
import RatingStars from "../../common/RatingStars";
import GetAvgRating from "../../../../../utils/avgRating";
import { Link } from "react-router-dom";

const CourseCard = ({ course, Height }) => {
  const [avgReviewCount, setAvgReviewCount] = useState(0);

  useEffect(() => {
    const count = GetAvgRating(course.ratingAndReviews);
    setAvgReviewCount(count);
  }, [course]);

  return (
    <>
      <Link to={`/courses/${course._id}`}>
        <div className="">
          <div className="rounded-lg">
            <img
              src={course?.thumbnail}
              alt="course thumbnail"
              className={`${Height} w-full rounded-xl object-cover`}
            />
          </div>
          <div className="flex flex-col gap-2 px-1 py-3">
            <p className="text-xl text-richblack-5">{course?.courseName}</p>
            <p className="text-sm text-richblack-50">
              {course?.instructor?.firstName}
              {course?.instructor?.lastName}
            </p>
            <div className="flex items-center gap-2">
              <span className="text-yellow-5">{avgReviewCount || 0}</span>
              <RatingStars Review_Count={avgReviewCount} />
              <span className="text-richblack-400">
                {course?.ratingAndReviews?.length} Ratings
              </span>
            </div>
            <p className="text-xl text-richblack-5">Rs .
              {course?.price}</p>
            </div>
          </div>
        </Link>
      </>
    );
};

```

**StudyNotion**

Home Catalog ▾ About Us Contact Us AR ▾

Home / Catalog / Web Development

## Web Development

Web development is the work involved in developing a website for the Internet (World Wide Web) or an intranet (a private network). Web development can range from developing a simple single static page of plain text to complex web applications, electronic businesses, and social network services.

### Courses to get you started

Most Popular New

NEXT JS & TYPESCRIPT

3 ★ ★ ★ ★ 1 Ratings

1100

TYPESCRIPT

5 ★ ★ ★ ★ 1 Ratings

1250

The Complete 2023 Web Development Bootcamp

0 ★ ★ ★ ★ 0 Ratings

1199

### Checkout Android Courses Also

DEVOPS 101

5 ★ ★ ★ ★ 1 Ratings

1250

ANDROID MADE EASY

4 ★ ★ ★ ★ 1 Ratings

999

ANDROID ZERO TO HERO

0 ★ ★ ★ ★ 0 Ratings

2500

### Most Selling Courses

DEVOPS 101

5 ★ ★ ★ ★ 1 Ratings

1250

ANDROID MADE EASY

4 ★ ★ ★ ★ 1 Ratings

999

NEXT JS & TYPESCRIPT

3 ★ ★ ★ ★ 1 Ratings

1100

TYPESCRIPT

5 ★ ★ ★ ★ 1 Ratings

1250

---

**StudyNotion**

**Company**

- About
- Careers
- Affiliates

[F](#) [G](#) [T](#) [O](#)

**Resources**

- Articles
- Blog
- Chart Sheet
- Code challenges
- Docs
- Projects
- Videos
- Workspaces

**Plans**

- Paid memberships
- For students
- Business solutions

**Community**

- Forums
- Chapters
- Events

**Subjects**

- AI
- Cloud Computing
- Code Foundations
- Computer Science
- Cybersecurity
- Data Analytics
- Data Science
- Data Visualization
- Developer Tools
- DevOps
- Game Development
- IT
- Machine Learning
- Math
- Mobile Development
- Web Design
- Web Development

**Languages**

- Bash
- C++
- C#
- Go
- HTML & CSS
- Java
- JavaScript
- Kotlin
- PHP
- Python
- R
- Ruby
- SQL
- Swift

**Career building**

- Career paths
- Career services
- Interview prep
- Professional certification
- Full Catalog
- Beta Content

Privacy Policy | Cookie Policy | Terms

Made with ❤️ by CodeHelp © 2023 Studynotion

### 3.14 Android Course List

```
// Import the required modules
const express = require("express");
const router = express.Router();

// Import the Controllers

// Course Controllers Import
const {
  createCourse,
  getAllCourses,
  getCourseDetails,
  getFullCourseDetails,
  editCourse,
  getInstructorCourses,
  deleteCourse,
} = require("../controllers/Course");

// Categories Controllers Import
const {
  showAllCategories,
  createCategory,
  categoryPageDetails,
} = require("../controllers/Category");

// Sections Controllers Import
const {
  createSection,
  updateSection,
  deleteSection,
} = require("../controllers/Section");

// Sub-Sections Controllers Import
const {
  createSubSection,
  updateSubSection,
  deleteSubSection,
} = require("../controllers/Subsection");

// Rating Controllers Import
const {
  createRating,
  getAverageRating,
  getAllRating,
} = require("../controllers/RatingAndReview");
```

```
const { updateCourseProgress } =
require("../controllers/courseProgress");

// Importing Middlewares
const {
  auth,
  isInstructor,
  isStudent,
  isAdmin,
} = require("../middlewares/auth");

// **** Course routes ****
// Courses can Only be Created by Instructors
router.post("/createCourse", auth, isInstructor, createCourse);
//Add a Section to a Course
router.post("/addSection", auth, isInstructor, createSection);
// Update a Section
router.post("/updateSection", auth, isInstructor, updateSection);
// Delete a Section
router.post("/deleteSection", auth, isInstructor, deleteSection);
// Edit Sub Section
router.post("/updateSubSection", auth, isInstructor, updateSubSection);
// Delete Sub Section
router.post("/deleteSubSection", auth, isInstructor, deleteSubSection);
// Add a Sub Section to a Section
router.post("/addSubSection", auth, isInstructor, createSubSection);
// Get all Registered Courses
router.get("/getAllCourses", getAllCourses);
// Get Details for a Specific Courses
router.post("/getCourseDetails", getCourseDetails);
// Get Details for a Specific Courses
router.post("/getFullCourseDetails", auth, getFullCourseDetails);
// Edit Course routes
router.post("/editCourse", auth, isInstructor, editCourse);
// Get all Courses Under a Specific Instructor
router.get("/getInstructorCourses", auth, isInstructor,
getInstructorCourses);
// Delete a Course
router.delete("/deleteCourse", deleteCourse);
```

```
router.post("/updateCourseProgress", auth, isStudent,
updateCourseProgress);

// *****
*****                                         Category routes (Only by Admin)
// *****
// *****
*****                                         Rating and Review
// *****
```

// Category can Only be Created by Admin  
// TODO: Put IsAdmin Middleware here  
router.post("/createCategory", auth, isAdmin, createCategory);  
// router.get("/showAllCategories", showAllCategories);  
router.post("/getCategoryPageDetails", categoryPageDetails);

// !before in showAllCategories when write "auth" this create the problem  
status code 401 not found  
router.get("/showAllCategories", showAllCategories);

// \*\*\*\*\*
\*\*\*\*\* Rating and Review
// \*\*\*\*\*
// \*\*\*\*\*
\*\*\*\*\*

router.post("/createRating", auth, isStudent, createRating);  
router.get("/getAverageRating", getAverageRating);  
router.get("/getReviews", getAllRating);

module.exports = router;

**StudyNotion**

Home Catalog About Us Contact Us AR

Home / Catalog / Android

## Android

Android software development is the process by which applications are created for devices running the Android operating system. Google states that "Android apps can be written using Kotlin, Java, and C++ languages" using the Android software development kit (SDK), while using other languages is also possible.

### Courses to get you started

Most Popular New

**DEVOPS 101**

5 ★ ★ ★ ★ 1 Ratings  
1250

**ANDROID MADE EASY**

4 ★ ★ ★ ★ 1 Ratings  
999

**ANDROID ZERO TO HERO**

0 ★ ★ ★ ★ 0 Ratings  
2500

### Checkout Web Development Courses Also

**NEXT JS & TYPESCRIPT**

3 ★ ★ ★ ★ 1 Ratings  
1100

**TYPESCRIPT**

5 ★ ★ ★ ★ 1 Ratings  
1250

**WEB DEVELOPMENT**

The Complete 2023 Web Development Bootcamp  
0 ★ ★ ★ ★ 0 Ratings  
1199

### Most Selling Courses

**DEVOPS 101**

5 ★ ★ ★ ★ 1 Ratings  
1250

**ANDROID MADE EASY**

4 ★ ★ ★ ★ 1 Ratings  
999

**NEXT JS & TYPESCRIPT**

3 ★ ★ ★ ★ 1 Ratings  
1100

**TYPESCRIPT**

5 ★ ★ ★ ★ 1 Ratings  
1250

---

**S StudyNotion**

**Company**

- About
- Careers
- Affiliates

[Facebook](#) [Twitter](#) [Instagram](#) [YouTube](#)

**Resources**

- Articles
- Blog
- Chart Sheet
- Code challenges
- Docs
- Projects
- Videos
- Workspaces

**Plans**

- Paid memberships
- For students
- Business solutions

**Community**

- Forums
- Chapters
- Events

**Subjects**

- AI
- Cloud Computing
- Code Foundations
- Computer Science
- Cybersecurity
- Data Analytics
- Data Science
- Data Visualization
- Developer Tools
- DevOps
- Game Development
- IT
- Machine Learning
- Math
- Mobile Development
- Web Design
- Web Development

**Languages**

- Bash
- C++
- C#
- Go
- HTML & CSS
- Java
- Kotlin
- PHP
- Python
- R
- SQL
- Swift

**Career building**

- Career paths
- Career services
- Interview prep
- Professional certification
- Full Catalog
- Beta Content

Privacy Policy | Cookie Policy | Terms

Made with ❤️ by CodeHelp © 2023 Studynotion

```
import React from "react";
import HighlightText from
"../../../../components/core/HomePage/HighlightText";
import CTAButton from "../../../../components/core/HomePage/Button";

const LearningGridArray = [
{
  order: -1,
  heading: "World-Class Learning for",
  highlightText: "Anyone, Anywhere",
  description:
    "Studynotion partners with more than 275+ leading universities and companies to bring flexible, affordable, job-relevant online learning to individuals and organizations worldwide.",
  BtnText: "Learn More",
  BtnLink: "/",
},
{
  order: 1,
  heading: "Curriculum Based on Industry Needs",
  description:
    "Save time and money! The Belajar curriculum is made to be easier to understand and in line with industry needs.",
},
{
  order: 2,
  heading: "Our Learning Methods",
  description:
    "Studynotion partners with more than 275+ leading universities and companies to bring",
},
{
  order: 3,
  heading: "Certification",
  description:
    "Studynotion partners with more than 275+ leading universities and companies to bring",
},
{
  order: 4,
  heading: `Rating "Auto-grading"`,
  description:
    "Studynotion partners with more than 275+ leading universities and companies to bring",
},
{
  order: 5,
  heading: "Ready to Work",
}
```

```

description:
    "Studynotion partners with more than 275+ leading universities and
companies to bring",
},
];

const LearningGrid = () => {
    return (
        <div className="grid mx-auto w-[350px] xl:w-fit grid-cols-1 xl:grid-
cols-4 mb-12">
            {LearningGridArray.map((card, i) => {
                return (
                    <div
                        key={i}
                        className={`${i === 0 && "xl:col-span-2 xl:h-[294px]"} ${{
                            card.order % 2 === 1
                            ? "bg-richblack-700 h-[294px]"
                            : card.order % 2 === 0
                            ? "bg-richblack-800 h-[294px]"
                            : "bg-transparent"
                        }} ${card.order === 3 && "xl:col-start-2"} `}
                >
                    {card.order < 0 ? (
                        <div className="xl:w-[90%] flex flex-col gap-3 pb-10 xl:pb-
0">
                            <div className="text-4xl font-semibold ">
                                {card.heading}
                                <HighlightText text={card.highlightText} />
                            </div>
                            <p className="text-richblack-300 font-medium">
                                {card.description}
                            </p>

                            <div className="w-fit mt-2">
                                <CTAButton active={true} linkto={card.BtnLink}>
                                    {card.BtnText}
                                </CTAButton>
                            </div>
                        </div>
                    ) : (
                        <div className="p-8 flex flex-col gap-8">
                            <h1 className="text-richblack-5 text-
lg">{card.heading}</h1>

                            <p className="text-richblack-300 font-medium">
                                {card.description}
                            </p>
                        </div>
                    )
                )
            )}
        </div>
    );
}

```

**StudyNotion**

Home Catalog ▾ About Us Contact Us AR ▾

## The Complete 2023 Web Development Bootcamp

Become a Full-Stack Web Developer with just ONE course. HTML, CSS, Javascript, Node, React, MongoDB, Web3 and DApps

0 ★ ★ ★ ★ (0 reviews) (0 students enrolled)

Created By Sandeep

Created At August 23, 2023 | 5:28 PM English

**What You Will Learn**

Build 16 web development projects for your portfolio, ready to apply for junior developer jobs. Learn the latest technologies, including Javascript, React, Node and even Web3 development. After the course you will be able to build ANY website you want. Build fully-fledged websites and web apps for your startup or business. Work as a freelance web developer. Master frontend development with React. Master backend development with Node. Learn professional developer best practices.

**Course Content:**

3 section(s) 3 lectures 27s total length [Collapse all Sections](#)

- LEC1 1 lecture(s)
- HTML Heading Elements
- LEC2 1 lecture(s)
- Nesting and Indentation
- LEC3 1 lecture(s)
- CSS Positioning

**Author**

 Sandeep Kumar

Hi. I'm Sandeep. I have a degree in engineering from the University for Applied Science Konstanz in Germany and discovered my love for programming there. Currently, over 350,000 students learn from my courses. This gives me much energy to create new courses with the highest quality possible. I aim to make learning to code accessible for everyone, as I am convinced, that IT IS THE FUTURE! So join my courses and learn to create apps, games, websites, or any other type of application. The possibilities are limitless.

Resources	Plans	Subjects	Languages	Career building
Articles	Paid memberships	AI	Bash	Career paths
Blog	For students	Cloud Computing	C++	Career services
Chart Sheet	Business solutions	Code Foundations	C#	Interview prep
Code challenges		Computer Science	Go	Professional certification
Docs		Cybersecurity	HTML & CSS	-
Projects		Data Analytics	Java	Full Catalog
Videos		Data Science	JavaScript	Beta Content
Workspaces		Data Visualization	Kotlin	
		Developer Tools	PHP	
		DevOps	Python	
		Game Development	R	
		IT	Ruby	
		Machine Learning	SQL	
		Math	Swift	
		Mobile Development		
		Web Design		
		Web Development		

Privacy Policy | Cookie Policy | Terms | Made with ❤️ by CodeHelp © 2023 Studynotion

### 3.13 About Us Page

```

import React from "react";
import HighlightText from
"../../../../components/core/HomePage/HighlightText";
import CTAButton from "../../../../components/core/HomePage/Button";

const LearningGridArray = [
{
  order: -1,
  heading: "World-Class Learning for",
  highlightText: "Anyone, Anywhere",
  description:
    "Studynotion partners with more than 275+ leading universities and
    companies to bring flexible, affordable, job-relevant online learning to
    individuals and organizations worldwide.",
  BtnText: "Learn More",
  BtnLink: "/",
},
{
  order: 1,
  heading: "Curriculum Based on Industry Needs",
  description:
    "Save time and money! The Belajar curriculum is made to be easier
    to understand and in line with industry needs.",
},
{
  order: 2,
  heading: "Our Learning Methods",
  description:
    "Studynotion partners with more than 275+ leading universities and
    companies to bring",
},
{
  order: 3,
  heading: "Certification",
  description:
    "Studynotion partners with more than 275+ leading universities and
    companies to bring",
},
{
  order: 4,
  heading: `Rating "Auto-grading"`,
  description:
    "Studynotion partners with more than 275+ leading universities and
    companies to bring",
},
{
  order: 5,
}
]

```

```

heading: "Ready to Work",
description:
  "Studynotion partners with more than 275+ leading universities and
companies to bring",
},
];
}

const LearningGrid = () => {
  return (
    <div className="grid mx-auto w-[350px] xl:w-fit grid-cols-1 xl:grid-
cols-4 mb-12">
      {LearningGridArray.map((card, i) => {
        return (
          <div
            key={i}
            className={`${i === 0 && "xl:col-span-2 xl:h-[294px]"} ${
              card.order % 2 === 1
                ? "bg-richblack-700 h-[294px]"
                : card.order % 2 === 0
                ? "bg-richblack-800 h-[294px]"
                : "bg-transparent"
            } ${card.order === 3 && "xl:col-start-2"} `}
        >
          {card.order < 0 ? (
            <div className="xl:w-[90%] flex flex-col gap-3 pb-10 xl:pb-
0">
              <div className="text-4xl font-semibold">
                {card.heading}
                <HighlightText text={card.highlightText} />
              </div>
              <p className="text-richblack-300 font-medium">
                {card.description}
              </p>

              <div className="w-fit mt-2">
                <CTAButton active={true} linkto={card.BtnLink}>
                  {card.BtnText}
                </CTAButton>
              </div>
            </div>
          ) : (
            <div className="p-8 flex flex-col gap-8">
              <h1 className="text-richblack-5 text-
lg">{card.heading}</h1>

              <p className="text-richblack-300 font-medium">
                {card.description}
              </p>
            </div>
          )
        )
      )}
    </div>
  )
}

```

**Driving Innovation in Online Education for a Brighter Future**

StudyNotion is at the forefront of driving innovation in online education. We're passionate about creating a brighter future by offering cutting-edge courses, leveraging emerging technologies, and nurturing a vibrant learning community.



We are passionate about revolutionizing the way we learn. Our innovative platform **combines technology, expertise, and community** to create an **unparalleled educational experience**.

### Our Founding Story

Our e-learning platform was born out of a shared vision and passion for transforming education. It all began with a group of educators, technologists, and lifelong learners who recognized the need for accessible, flexible, and high-quality learning opportunities in a rapidly evolving digital world.

As experienced educators ourselves, we witnessed firsthand the limitations of traditional education. We believed that learning and education should not be confined to the walls of a classroom or restricted by geographical boundaries. We envisioned a platform that could bridge these gaps and empower individuals from all walks of life to unlock their full potential.



### Our Vision

With this vision in mind, we set out on a journey to create an e-learning platform that would revolutionize the way people learn. Our team of dedicated experts worked tirelessly to develop a user-friendly platform that combines cutting-edge technology with engaging content, fostering a dynamic and interactive learning experience.

### Our Mission

Our mission goes beyond just delivering courses online. We want to create a vibrant community of learners, where individuals can connect, collaborate, and support one another. We believe that knowledge thrives in an environment of sharing and dialogue, and we foster this spirit of community through forums, live sessions, and networking opportunities.

5K Active Students	10+ Mentors	200+ Courses	50+ Awards
-----------------------	----------------	-----------------	---------------

#### World-Class Learning for Anyone, Anywhere

StudyNotion partners with more than 275+ leading universities and companies to bring flexible, affordable, job-relevant online learning to individuals and organizations worldwide.

[Learn More](#)

Certification	Rating "Auto-grading"	Ready to Work
StudyNotion partners with more than 275+ leading universities and companies to bring	StudyNotion partners with more than 275+ leading universities and companies to bring	StudyNotion partners with more than 275+ leading universities and companies to bring

#### Curriculum Based on Industry Needs

Save time and money! The Belajar curriculum is made to be easier to understand and in line with industry needs.

#### Our Learning Methods

StudyNotion partners with more than 275+ leading universities and companies to bring

### Get in Touch

We'd love to hear from you. Please fill out this form.

First Name

Last Name

Email Address

Phone Number

Message

[Send Message](#)

### Reviews from other learners

**Kartik Kumar** The Complete Python 3.x & Kotlin Developer Masterclass  
Great Learning Experience  
5.0 ★★★★★

**Akshay Kumar** ANDROID MADE EASY  
Too Good!!  
4.0 ★★★★★

**Kartik Kumar** Python for Beginners to Advanced with Practical Examples  
Very Good Course!  
4.0 ★★★★★

**Kartik Kumar** NEXT.js & TYPESCRIPT  
More content could be added!  
5.0 ★★★★★

**StudyNotion**

**Company**  
About  
Careers  
Affiliates  
[G](#) [T](#) [S](#) [E](#)

**Resources**  
Articles  
Books  
Smart Sheet  
Code challenges  
Docs  
Projects  
Videos  
Workspaces  
**Support**  
Help Center

**Plans**  
Paid memberships  
For Students  
Business solutions  
**Community**  
Forums  
Chapters  
Events

**Subjects**  
AI  
Cloud Computing  
Computer Foundations  
Computer Science  
Cybersecurity  
Data Analytics  
Data Science  
Data Visualization  
Developer Tools  
Docker  
Game Development  
IT  
Machine Learning  
Math  
Mobile Development  
Web Design  
Web Development

**Languages**  
Dash  
Django  
C#  
Go  
HTML & CSS  
Java  
JavaScript  
Kotlin  
PHP  
Python  
R  
Ruby  
SQL  
Swift

**Career building**  
Career paths  
Job search services  
Interview prep  
Professional certification  
Full Catalog  
Beta Content

Privacy Policy | Cookie Policy | Terms

Made with ❤️ CodeHelp © 2023 StudyNotion

### 3.14 Contact us Section

```

import React from "react"

import Footer from "../components/common/Footer"
import ContactDetails from "../components/ContactPage/ContactDetails"
import ContactForm from "../components/ContactPage/ContactForm"
import ReviewSlider from "../components/common/ReviewSlider"

const Contact = () => {
  return (
    <div>
      <div className="mx-auto mt-20 flex w-11/12 max-w-maxContent flex-col justify-between gap-10 text-white lg:flex-row">
        {/* Contact Details */}
        <div className="lg:w-[40%]">
          <ContactDetails />
        </div>

        {/* Contact Form */}
        <div className="lg:w-[60%]">
          <ContactForm />
        </div>
      </div>
      <div className="relative mx-auto my-20 flex w-11/12 max-w-maxContent flex-col items-center justify-between gap-8 bg-richblack-900 text-white">
        {/* Reviews from Other Learner */}
        <h1 className="text-center text-4xl font-semibold mt-8">
          Reviews from other learners
        </h1>
        <ReviewSlider />
      </div>
      <Footer />
    </div>
  )
}

export default Contact

```

```

import React, { useEffect, useState } from "react";
import { BiInfoCircle } from "react-icons/bi";
import { HiOutlineGlobeAlt } from "react-icons/hi";
import { ReactMarkdown } from "react-markdown/lib/react-markdown";
import { useDispatch, useSelector } from "react-redux";
import { useNavigate, useParams } from "react-router-dom";

import ConfirmationModal from "../components/common/ConfirmationModal";

```

```

import Footer from "../../components/common/Footer";
import RatingStars from "../../components/common/RatingStars";
import CourseAccordionBar from
"../../components/core/Course/CourseAccordionBar";
import CourseDetailsCard from
"../../components/core/Course/CourseDetailsCard";
import { formatDate } from "../../services/formatDate";
import { fetchCourseDetails } from
"../../services/operations/courseDetailsAPI";
import { buyCourse } from "../../services/operations/studentFeaturesAPI";
import GetAvgRating from "../../utils/avgRating";
import Error from "./Error";

function CourseDetails() {
  const { user } = useSelector((state) => state.profile);
  const { token } = useSelector((state) => state.auth);
  const { loading } = useSelector((state) => state.profile);
  const { paymentLoading } = useSelector((state) => state.course);
  const dispatch = useDispatch();
  const navigate = useNavigate();

  // Getting courseId from url parameter
  const { courseId } = useParams();
  // console.log(`course id: ${courseId}`)

  console.log("useParams----->>>>", courseId);

  // Declare a state to save the course details
  const [response, setResponse] = useState(null);
  const [confirmationModal, setConfirmationModal] = useState(null);
  useEffect(() => {
    // Calling fetchCourseDetails function to fetch the details
    (async () => {
      try {
        const res = await fetchCourseDetails(courseId);

        setResponse(res);
      } catch (error) {
        console.log("Could not fetch Course Details");
      }
    })();
  }, [courseId]);

  // Calculating Avg Review count
  const [avgReviewCount, setAvgReviewCount] = useState(0);
  useEffect(() => {
    const count =
GetAvgRating(response?.data?.courseDetails.ratingAndReviews);
  }, [response]);
}

```

```
    setAvgReviewCount(count);
}, [response]);
// console.log("avgReviewCount: ", avgReviewCount)

// // Collapse all
// const [collapse, setCollapse] = useState("")
const [isActive, setIsActive] = useState(Array(0));
const handleActive = (id) => {
    // console.log("called", id)
    setIsActive(
        !isActive.includes(id)
            ? isActive.concat([id])
            : isActive.filter((e) => e !== id)
    );
};

// Total number of lectures
const [totalNoOfLectures, setTotalNoOfLectures] = useState(0);
useEffect(() => {
    let lectures = 0;
    response?.data?.courseDetails?.courseContent?.forEach((sec) => {
        lectures += sec.subSection.length || 0;
    });
    setTotalNoOfLectures(lectures);
}, [response]);

if (loading || !response) {
    return (
        <div className="grid min-h-[calc(100vh-3.5rem)] place-items-center">
            <div className="spinner"></div>
        </div>
    );
}
if (!response.success) {
    return <Error />;
}

const {
    // _id: course_id,
    courseName,
    courseDescription,
    thumbnail,
    price,
    whatYouWillLearn,
    courseContent,
    ratingAndReviews,
    instructor,
```

```

    studentsEnrolled,
    createdAt,
} = response.data?.courseDetails;

// console.log(
//   "response.data?.courseDetails--&&&&&&&&&&&7-->",
//   response.data?.courseDetails
// );

// console.log(" _id: course_id,-----kkkkk--..", course_id);

const handleBuyCourse = () => {
  if (token) {
    console.log("handleBuyCourse ---> token", token);
    console.log("handleBuyCourse ---> courseId", [courseId]);
    console.log("handleBuyCourse ---> user", user);
    buyCourse(token, [courseId], user, navigate, dispatch);
    return;
  }
  setConfirmationModal({
    text1: "You are not logged in!",
    text2: "Please login to Purchase Course.",
    btn1Text: "Login",
    btn2Text: "Cancel",
    btn1Handler: () => navigate("/login"),
    btn2Handler: () => setConfirmationModal(null),
  });
};

if (paymentLoading) {
  // console.log("payment loading")
  return (
    <div className="grid min-h-[calc(100vh-3.5rem)] place-items-center">
      <div className="spinner"></div>
    </div>
  );
}

return (
  <>
    <div className={`relative w-full bg-richblack-800`}>
      {/* Hero Section */}
      <div className="mx-auto box-content px-4 lg:w-[1260px] xl:relative ">
        <div className="mx-auto grid min-h-[450px] max-w-maxContentTab justify-items-center py-8 lg:mx-0 lg:justify-items-start lg:py-0 xl:max-w-[810px]">

```

```

<div className="relative block max-h-[30rem] lg:hidden">
    <div className="absolute bottom-0 left-0 h-full w-full
shadow-[#161D29_0px_-64px_36px_-28px_inset]"></div>
    <img
        src={thumbnail}
        alt="course thumbnail"
        className="aspect-auto w-full"
    />
</div>
<div
    className={`${z-30 my-5 flex flex-col justify-center gap-4
py-5 text-lg text-richblack-5`}
>
    <div>
        <p className="text-4xl font-bold text-richblack-5
sm:text-[42px]">
            {courseName}
        </p>
    </div>
    <p className="text-richblack-200">{courseDescription}</p>
    <div className="text-md flex flex-wrap items-center gap-2">
        <span className="text-yellow-25">{avgReviewCount}</span>
        <RatingStars Review_Count={avgReviewCount} Star_Size={24}>
/>
        <span>`(${ratingAndReviews.length} reviews)`</span>
        <span>`${studentsEnrolled.length} students
enrolled`</span>
    </div>
    <div>
        <p className="">
            Created By{" "}
            {`${instructor?.firstName} ${instructor?.lastName}`}
        </p>
    </div>
    <div className="flex flex-wrap gap-5 text-lg">
        <p className="flex items-center gap-2">
            {" "}
            <BiInfoCircle /> Created at {formatDate(createdAt)}
        </p>
        <p className="flex items-center gap-2">
            {" "}
            <HiOutlineGlobeAlt /> English
        </p>
    </div>
    <div className="flex w-full flex-col gap-4 border-y border-y-
richblack-500 py-4 lg:hidden">

```

```


Rs. {price}


</p>
<button
    className="yellowButton text-richblack-5 bg-richblack-600
px-3 py-1 w-fit rounded-md"
    onClick={handleBuyCourse}
>
    Buy Now
</button>
<button className="blackButton px-3 py-1 w-fit rounded-md
text-richblack-5 bg-richblack-600">
    Add to Cart
</button>
</div>
</div>
{/* Courses Card */}
<div className="right-[1rem] top-[60px] mx-auto hidden min-h-[600px] w-1/3 max-w-[410px] translate-y-24 md:translate-y-0
lg: absolute lg:block">
    <CourseDetailsCard
        course={response?.data?.courseDetails}
        setConfirmationModal={setConfirmationModal}
        handleBuyCourse={handleBuyCourse}
    />
</div>
</div>
</div>
<div className="mx-auto box-content px-4 text-start text-richblack-5 lg:w-[1260px]">
    <div className="mx-auto max-w-maxContentTab lg:mx-0 xl:max-w-[810px]">
        {/* What will you learn section */}
        <div className="my-8 border border-richblack-600 p-8">
            <p className="text-3xl font-semibold">What you'll learn</p>
            <div className="mt-5">
                <ReactMarkdown>{whatYouWillLearn}</ReactMarkdown>
            </div>
        </div>

        {/* Course Content Section */}
        <div className="max-w-[830px]">
            <div className="flex flex-col gap-3">
                <p className="text-[28px] font-semibold">Course Content</p>
                <div className="flex flex-wrap justify-between gap-2">
                    <div className="flex gap-2">
                        <span>

```

```

        {courseContent.length} {'section(s)'}
    </span>
    <span>
        {totalNoOfLectures} {'lecture(s)'}
    </span>
    <span>{response.data?.totalDuration} total
length</span>
    </div>
    <div>
        <button
            className="text-yellow-25"
            onClick={() => setIsActive([])}
        >
            Collapse all sections
        </button>
    </div>
    </div>
</div>

/* Course Details Accordion */
<div className="py-4">
    {courseContent?.map((course, index) => (
        <CourseAccordionBar
            course={course}
            key={index}
            isActive={isActive}
            handleActive={handleActive}
        />
    ))}
</div>

/* Author Details */
<div className="mb-12 py-4">
    <p className="text-[28px] font-semibold">Author</p>
    <div className="flex items-center gap-4 py-4">
        <img
            src={
                instructor?.image
                ? instructor?.image
                :
`https://api.dicebear.com/5.x/initials/svg?seed=${instructor?.firstName}
${instructor?.lastName}`
            }
            alt="Author"
            className="h-14 w-14 rounded-full object-cover"
        />
        <p className="text-lg">`${instructor?.firstName}
${instructor?.lastName}`</p>
    </div>
</div>

```

```
</div>
<p className="text-richblack-50">
  {instructor?.additionalDetails?.about}
</p>
</div>
</div>
</div>
<Footer />
{confirmationModal && <ConfirmationModal
modalData={confirmationModal} />}
</>
);
}

export default CourseDetails;
```

**S StudyNotion**

Home Catalog ▾ About Us Contact Us AR ▾

**Chat on us**  
Our friendly team is here to help.  
[info@studynotion.com](mailto:info@studynotion.com)

**Visit us**  
Come and say hello at our office HQ.  
Akshya Nagar 1st Block 1st Cross, Rammurthy nagar,  
Bangalore-560016

**Call us**  
Mon - Fri From 8am to 5pm  
+123 456 7890

## Got a Idea? We've got the skills. Let's team up

Tell us more about yourself and what you're got in mind.

First Name

Last Name

Email Address

Phone Number

Message

**Send Message**

## Reviews From Other Learners



**Manasi Sarode**  
TYPESCRIPT

Very Nice Course

5.0 ★★★★★



**Kartik Kumar**  
The Complete Android 12 & Kotlin Development Masterclass

Great Learning Experience!

5.0 ★★★★★



**Akshay Kumar**  
ANDROID MADE EASY

Too Good!!

4.0 ★★★★★



**Kartik Kumar**  
DevOps Beginners to Advanced with Projects - 2023

Very Good Course!

4.0 ★★★★★

**S StudyNotion**

**Company**

- About
- Careers
- Affiliates

**Resources**

- Articles
- Blog
- Chart Sheet
- Code challenges
- Docs
- Projects
- Videos
- Workspaces

**Support**

- Help Center

**Plans**

- Paid memberships
- For students
- Business solutions

**Community**

- Forums
- Chapters
- Events

**Subjects**

- AI
- Cloud Computing
- Code Foundations
- Computer Science
- Cybersecurity
- Data Analytics
- Data Science
- Data Visualization
- Developer Tools
- DevOps
- Game Development
- IT
- Machine Learning
- Math
- Mobile Development
- Web Design
- Web Development

**Languages**

- Bash
- C++
- C#
- Go
- HTML & CSS
- Java
- JavaScript
- Kotlin
- PHP
- Python
- R
- Ruby
- SQL
- Swift

**Career building**

- Career paths
- Career services
- Interview prep
- Professional certification
- 
- Full Catalog
- Beta Content

[Privacy Policy](#) | [Cookie Policy](#) | [Terms](#)

Made with ❤ by CodeHelp © 2023 Studynotion

### 3.15 Logout Section

```

import { FiTrash2 } from "react-icons/fi"
import { useDispatch, useSelector } from "react-redux"
import { useNavigate } from "react-router-dom"

import { deleteProfile } from
"../../../../services/operations/SettingsAPI"

export default function DeleteAccount() {
  const { token } = useSelector((state) => state.auth)
  const dispatch = useDispatch()
  const navigate = useNavigate()

  async function handleDeleteAccount() {
    try {
      dispatch(deleteProfile(token, navigate))
    } catch (error) {
      console.log("ERROR MESSAGE - ", error.message)
    }
  }

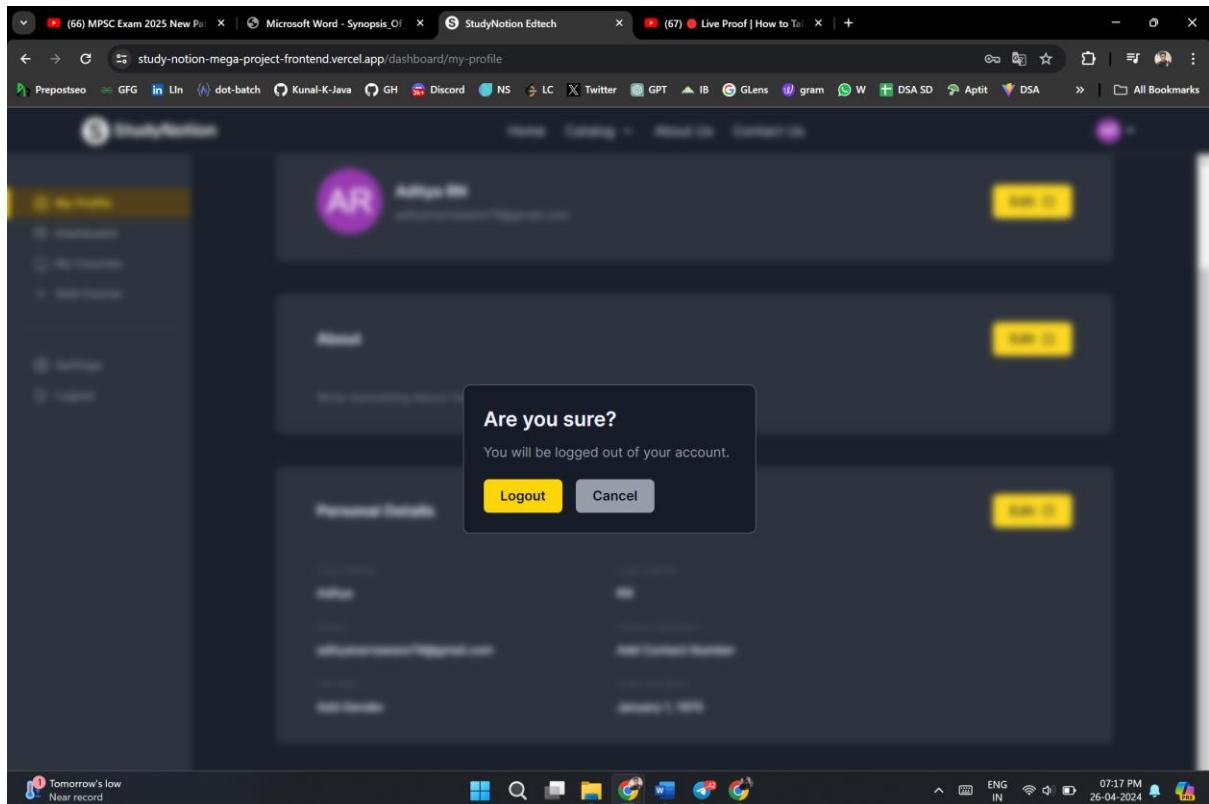
  return (
    <>
      <div className="my-10 flex flex-row gap-x-5 rounded-md border-[1px] border-pink-700 bg-pink-900 p-8 px-12">
        <div className="flex aspect-square h-14 w-14 items-center justify-center rounded-full bg-pink-700">
          <FiTrash2 className="text-3xl text-pink-200" />
        </div>
        <div className="flex flex-col space-y-2">
          <h2 className="text-lg font-semibold text-richblack-5">
            Delete Account
          </h2>
          <div className="w-3/5 text-pink-25">
            <p>Would you like to delete account?</p>
            <p>
              This account may contain Paid Courses. Deleting your
              account is
              permanent and will remove all the content associated with
              it.
            </p>
          </div>
          <button
            type="button"
            className="w-fit cursor-pointer italic text-pink-300"
            onClick={handleDeleteAccount}
          >
            I want to delete my account.
          </button>
        </div>
      </div>
    </>
  )
}

```

```

        </button>
    </div>
</div>
</>
)
}

```



## 5.2 Model Output

The output of the recommender system, including personalized recommendations, is discussed and analyzed.

## 5.3 Summary

The chapter summarizes the key findings from the implementation and evaluation processes, highlighting the effectiveness of the Study Notion Portal Website in providing personalized recommendations.

## **CHAPTER 6**

### **CONCLUSIONS AND RECOMMENDATIONS**

#### **6.1 Introduction**

This final chapter provides a comprehensive conclusion to the research, discussing contributions, future work, and recommendations.

#### **6.2 Discussion and Conclusion**

The findings of the study are discussed in detail, emphasizing the impact of the Study Notion Portal Website and the recommender system on user experience and learning outcomes.

#### **6.3 Contributions**

The chapter highlights the contributions of the research, including advancements in online courses systems, the integration of recommender systems, and the enhancement of user engagement.

#### **6.4 Future Work**

Suggestions for future research are provided, addressing potential areas for improvement, expansion, and innovation in the field of online courses platforms.

#### **6.5 Recommendations**

Concluding recommendations are presented, providing guidance for educational institutions and developers seeking to implement similar systems.

This detailed report provides a comprehensive overview of the research methodology, implementation, results, and conclusions related to the development of the Study Notion Portal Website with an integrated recommender system.

## REFERENCES

- Subhashini, S., & Padmavathi, G. (2016). Online course system. International Journal of Advanced Research in Computer Science and Software Engineering, 6(9), 12-17.
- Amin, A., & Sharma, R. (2019). Design and development of secure online course system. International Journal of Computer Applications, 180(7), 1-5.
- EDUCAUSE. (n.d.). Online Assessment. eLearning Industry. (n.d.).
- E-Learning Solutions for Online courses.
- Dehnavi, M. M., & Jalili, S. (2016). Online courses system: A case study of designing a web-based online study portal system. In 2016 4th International Conference on Cyber Intelligence and Security (CIS) (pp. 91-96). IEEE.

<https://www.google.com/>

<https://youtube.com/>

<https://developer.mozilla.org/>

<https://www.geeksforgeeks.org/>

<https://www.javatpoint.com/>

<https://www.w3schools.com/>

<https://stackoverflow.com/>

<https://www.wikipedia.org/>

<https://mdn.org/>

<https://www.postman.com/>

<https://www.mongodb.com/>

<https://vercel.com/>