



Smart E-Learning Management System

By

Abdul Ghani

(BSCS/F17/0145)

Sagar Abbas

(BSCS/F17/0110)

Sadam Hussain

(BSCS/F17/0129)

A Project Presented to the

Faculty of Engineering Sciences and Technology

Hamdard Institute of Engineering and Technology

In partial fulfilment of the degree requirements

Bachelors of Science

In

Computer Science

Faculty of Engineering Science and Technology

Hamdard Institute of Engineering and Technology

Hamdard University, Karachi, Pakistan

Table of Contentss

Certificate.....	I
Acknowledgment.....	III
Abstract.....	III
List of Figures.....	IV
List of Tables.....	V
List of Abbreviations.....	VI
Chapter-1.....	1
INTRODUCTION.....	1
1.1 Motivation.....	1
1.2 Problem Statement.....	1
1.3 Objective and Contributions..	2
1.4 Project Scope...	3
1.5 Organization of report.....	4
Chapter-2.....	5
LITERATURE REVIEW.....	5
2.1 Introduction.....	6
2.1.1 Flex Website	6
2.1.2 Zoom App.....	7
2.1.3 Google Classroom.....	7
2.1.4 Iqra University Portal.....	7
2.1.5 Slate Portal.....	7
2.2 Feature Comparison Table.....	8
Chapter-3.....	9
SYSTEM MODEL.....	9
3.1 Functional and Non-Functional Requirements.....	9
3.1.1 Functional Requirements.....	9
3.1.2 Non-Functional Requirements.....	10
3.2 System Design	10
3.3 System Architecture.....	12
3.4 Scalability.....	13
3.5 Service.....	13
Chapter-4.....	14
SYSTEM DESIGN.....	14
4.1 UML Diagram.....	14
4.1.1 Use Case Diagram.....	15
4.1.2 Activity Diagram.....	16

4.1.3	Sequence Diagram.....	17
4.1.4	Class Diagram.....	18
4.2	Software Plateform.....	19
4.2.1	Operating System.....	19
4.2.2	Web Technology.....	19
4.3	Schema Design Diagram.....	19
4.4	Schema Table Dictionary.....	20
4.5	Inside Project.....	20
4.5.1	React.js.....	20
4.5.2	MongoDB.....	20
4.5.3	Express.js.....	20
4.54	Node.js.....	21
Chapter-5.....	22	
SYSTEM IMPLEMENTATION.....	22	
5.1	Project Timeline.....	22
5.2	Project Implementation.....	24
5.1.1	Login Module.....	26
5.1.2	Dashboard Module.....	27
5.1.3	Admin Screen Panel.....	31
5.1.4	Teacher Panel	27
5.1.5	Student Panel	29
5.1.6	Student Working Panel.....	28
5.1.7	Teacher Working Panel.....	27
Chapter-6.....	33	
RESULT AND DISCUSSION.....	33	
6.1	Testing.....	33
6.1.1	Black Box Testing.....	33
6.2	Test Case.....	34
6.1.1	Test Case # 1	34
6.1.2	Test Case # 2.....	34
6.1.3	Test Case # 3.....	35
6.1.4	Test Case # 4.....	35

Chapter-7.....	36
CONCLUSION AND FEATURE WORK.....	36
7.1 Conclusion	37
7.2 Future Work.....	38
REFERENCE.....	49

APPENDICES

Appendix-A: App.css.....	40
Appendix-B: App.test.....	41
Appendix-C: Index.htm.....	41
Appendix-D: User_Profile.....	42
Appendix-E: Admin_Login.....	55
Appendix-F: Admin_Dashboard.....	59
Appendix-G: Admin_menu.....	82
Appendix-H: admin_Sidebar.....	83
Appendix-I: Student_Login.....	87
Appendix-J:	
Student_Dashboard.....	88
Appendix-K: Student_Attendance.....	91
Appendix-L: Student_Sidebar.....	93
Appendix-M: Student_Attrmp_Assignment.....	95
Appendix-N: Teacher_Login.....	99
Appendix-O: Teacher_Dashboard.....	102
Appendix-P: ~Middleware.....	109
Appendix-Q: Models.....	110
Appendix-R: Server.....	122

Acknowledgement

All honor and glory **to Al Mighty" ALLAH,"** the most merciful, the most gracious, the source of knowledge and understanding owing to mankind. Respect to the Holy **Prophet "Hazrat MUHAMMAD,"** whose personality will continue to be a source of wisdom for humanity.

Recognize **Hamdard Institute of Engineering and Technology** for their support of our project, which is a significant accomplishment for us at the undergraduate level.

Our supervisor, **Asst. Prof. Dr. Adnan Ahmed Siddiqui**, has served as our primary advisor. It would like to convey our gratitude for their attentive direction, sincere assistance, which encouraged us to succeed in the project and helped it become a reality.

Many people, particularly our students and team members, have provided helpful feedback on this proposal, which has inspired us to develop our idea. Special thanks to **Asst Prof Dr Adnan Ahmed Siddiqui** and **Mohsin Raza Khan**, as well as everyone else, for their direct and indirect contributions to the completion of our project.

ABSTRACT

living in an era of innovation and technology. We have evolved to a great extent where everything is moving towards digitalization. Similarly, we are part of this era and we have developed a web application that inculcates the ability and design to meet the current needs without compromising the future needs. The purpose of this project is to develop an “**online e-learning**”. Online education is one of the most commonly used terms in the pandemic, that is why we have created the Smart E-Learning Management System. The Smart E-Learning Management System is a platform that makes online learning more easier and sociable. The COVID-19 pandemic has prompted innovative approaches to education. Educational institutions all across the world are turning to online learning platforms to continue the process of educating students. With online learning, the Smart E-Learning Management System has changed the concept of online education. Students and schools all across the world now use digital learning as a crucial resource. This is a whole new manner of learning that many educational institutions have had to adapt. Today's newest and most popular kind of distance education is online learning. It has had a significant impact on post-secondary education over the last decade, and the trend is only expected to continue. The internet and distance learning, often known as online e-learning, play an important role in the country's university system as well. It is undeniable that web-based e-learning provides a huge number of benefits to young students. In the end, there are also some disadvantages of online education. The few disadvantages of online e-learning are restricted cooperative learning, schedule expansion, and effortless study which involves physical activities. Our project and research in this field investigate the aftermaths of online e-learning among students, specifically in a private higher education institution, and its effect on them. Overviews, interviews, and supplementary data were used to collect data. According to the findings, there is a number of serious difficulties with regard to online universities and their impact on the nature of the educational system. A few concerns have been identified, and these issues must be addressed in order to maintain the essence of the university

for future generations. Furthermore, we have closely monitored the implementation of online universities, analyzed and surveyed the teaching methods, and promoted the progress to keep up with the nature of online e-learning in private advanced education institutions. Online learning is currently used not only for academic purposes but also for learning extracurricular activities for students. The demand for online learning has increased dramatically in recent months and will continue to do so in the future. Online education is a versatile instructional delivery strategy that includes any type of learning that occurs over the Internet. Online learning allows educators to reach students who would otherwise be unable to enroll in a regular classroom course, as well as students who prefer to work on their own time and at their own speed. Thus, we provide a platform for such prolific students to overcome their difficulties and help make their learning easier.

Keywords: Smart E-learning Management System, Web Application E-learning

List of Figures

Figure No.	Description	Page no.
Fig-1	Scope Of Project.....	3
Fig-2	Comparision Table.....	8
Fig-3	Overall System Flow Diagram.....	11
Fig-4	System Architecture Design.....	12
Fig-5	Use Case Diagram.....	15
Fig-6	Activity Diagram.....	16
Fig-7	Sequence Diagram.....	17
Fig-8	Class Diagram.....	18
Fig-9	Schema Design Diagram.....	19
Fig-10	Login Model.....	26
Fig-11	Admin Dashboard Model.....	27
Fig-12	Student Dashboard.....	28
Fig-13	Teacher Dashboard.....	29
Fig-14	Main Working Panel.....	30
Fig-15	Password Change.....	32

List of Tables

Table no.	Description	Page No.
Table 3.1	PHASES OF PROJECT.....	23
Table 5.1	TEST CASE 1.....	34
Table 5.2	TEST CASE 2.....	34
Table 5.3	TEST CASE 3.....	35
Table 5.4	TEST CASE 4.....	35
Table 5.5	TEST CASE 5.....	36
Table 5.6	TEST CASE 6.....	36

List of Abbreviations

DPR: Daily Performance Report

Mongo DB: Cross-Platform Document Oriented Database

Express: Back-End Framework

React: Front-End Library

Node: JS Runtime Environment

JS: JavaScripts.

AELMS: Advanced E-learning Management System

Chapter 1

Introduction to Smart E-Learning Management System

1.1 Motivation:

Incorporating technology into the educational system does not guarantee motivation of the students. To be honest, internet instruction has made the relationship between the understudy and the instructor less private. Instructors are expected to make the classroom a web-based environment. The question is what instructors are expected to do to excite understudies in a web-based environment. It is critical for educators to comprehend their understudy's motivation. Despite the fact that students enrol in online courses with the intention of successfully completing them, they frequently fail for a variety of reasons..

1. Explain to their understudies how to use the internet-based climate.
2. Encourage their understudy to communicate and work together.
3. Form a review group so that understudies never have to study alone again.
4. Assist understudies in making friends by introducing them to other understudies over the internet.
5. Work with their understudies by evaluating their online presence and offering them with constant feedback..

1.2 Problem Statement

The aim is to combine all online applications features at one platform, means all-in-one web application that will provide the online education. All properties of our project are similar to google classroom, and other applications that are used for online study. That is why we decided to implement it in a single web application that facilitated online platform to the students according to the modern standards.

Nowadays smartphone or laptop users are increasing day by day that is why in this project we will develop a web application in which we combine all systems in one web application. For instace, taking attendance and taking exams and quizzes and students and teachers interact with each other. In this modern period, all things will come to online systems so we decided that we implement one website that facilities online learning for students according to the modern period. Nowadays, covid-19 is increasing instantly. According to this situation, we will need to implement a website that facilities and motivates students towards e-learning.

1.3 Objective and Contributions

Our objective is to combine these systems into an all-in-one web application that will provide the students with almost all student portal features along with digital whiteboard and video lecture functionalities. Doing so, we can probably replace the current learning management systems in multiple universities across Pakistan. Thus, paving the way for a new form of flexible education.

Following are the objectives of our final year project:

- To provide a good and innovative environment.
- Easy to implement and understand.
- Supervised/secured admin rights.
- Privacy and system control throw admin and Teacher.
- Teacher dashboard which manages all the activities of the students.
- Admin which manages all the activities of academic and accounts of teacher and students

1.4 Project Scope

The scope of this project is to create a system which can make the work easy for the students and teachers. In Pakistan, the students face the issues of having less access to their areas of education while having extreme pressure of a transparent method of education. This is paving the way towards enrolling for online education, online quizzes, online assignment, watching online video lectures, online file uploading, online exam uploading, admin, students, teachers interact with each other through this website they will get the notification and take online classes and exams. Hence, the scope of online Smart E-Learning Management System in Pakistan is increasing regularly.

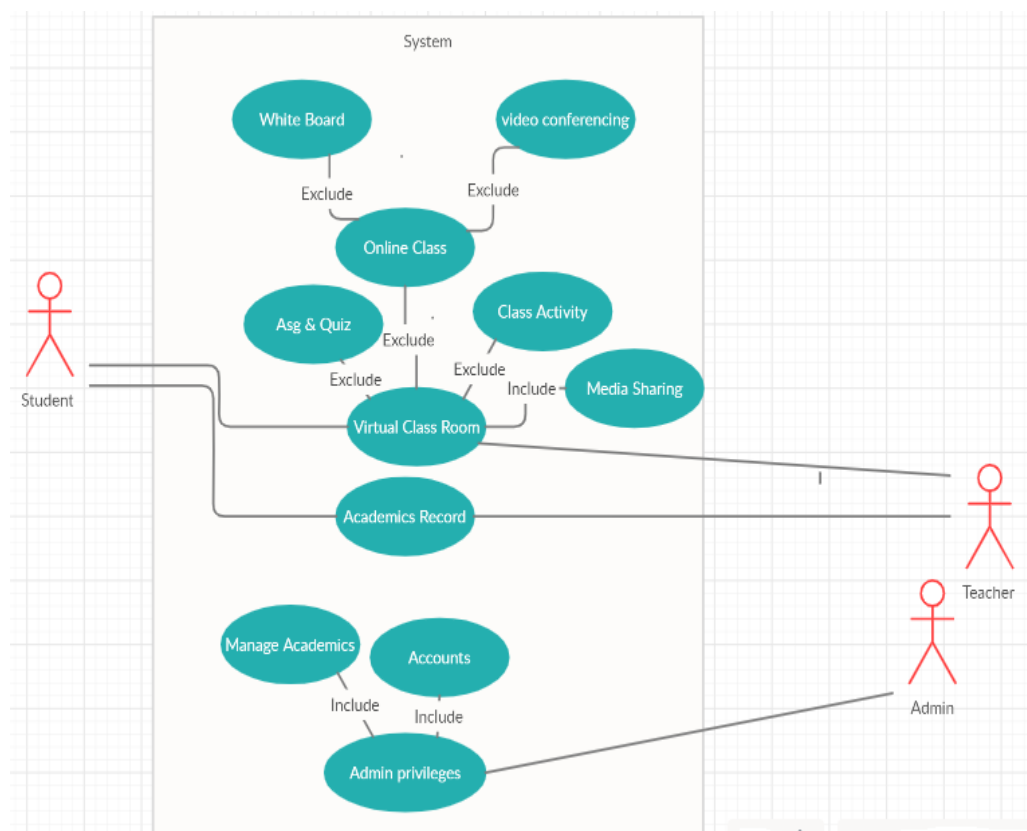


Fig-1: Project Scope

1.5 Organization of Report

The report begins with an explanation of the project's concept, which includes not only the project's foundation pillars but also the overall domain to which the initiative belongs and is intended to target.

Furthermore, the report has been divided into two primary sections, each with its own set of significant details, such as an introduction and an appendix. Software Engineering and project-related details are the two key parts.

1: The software engineering aspects covers all of the elements of the software side requirements for developing this project, such as the software engineering life cycle, software schedule development, and software risk management, among other things.

2: Whereas, the project details include Smart e-learning management system and motivation as well as scope and other factors.

3: By the end of the report, the conclusion of the report displays the project's complete outcomes as well as the project's future work, both of which will go in the same direction.

Chapter 2

Literature Review

2.1 Introduction

We are introducing a web application which is a single web application. Smart E-Learning Management System is a web application that promotes education/learning environments communication tools collaboration and education Management. There is always an emerging need for an increase in the development and utilization of vast technology to make learning more effective This project proposes for Smart E-learning Management System as a feasible product-based all-in-one solution for almost all systems out there. This study will specifically look at perfectly giving through smartphones and Laptops which continue to become more powerful and more popular for navigating the Internet and making online Learning. Literature audit deals with evaluating and summarising a current topic that is disseminated in order to acquire some related continuous subject arrangement in surveying. The main rationale for doing so is to link with the ongoing examination in order to grow that knowledge. The writing audit might be able to settle the dispute and improve the transparency of the momentum research. The writing survey's consistency is to examine the work of more experienced understudies and contrast this framework with another framework. This project team will acquire an idea of what additional features we can incorporate in the Smart E-Learning Management System final year project.

According to ongoing studies, university students who enrolled in e-learning courses outperformed those who took traditional classes. Carnegie Mellon University (CMU) in America is an example of this, where understudy test outcomes have improved as a result of e-learning techniques. As a result, it is critical that a university framework be created that allows for rapid adaptation to mechanical, social, societal, and political changes.

Implementing a learning the executives framework is a fantastic option for a higher education institution. The majority of merchants provide a comprehensive learning the executives framework item. If a single office or even a small college is considering purchasing the learning the executives framework, these costs could be very prohibitive. To overcome these problems,.

2.1.1 Background

In the teaching-learning process, e-learning is an interactive process between students and teachers, as well as study material. In other words, e-learning is a method of assisting students in their academic pursuits. To achieve the optimum result, the media and manner of the learning process transmitting the material would be critical. There are various methods of delivering information to pupils, one of which is the lecture approach. The lecture technique involves the teacher giving a verbal explanation and narrative to the class. The student's responsibility is to listen closely and take notes on what the teacher says. The problem with the lecture technique is that the engagement is centered on the teacher, therefore teachers cannot be confident about how well pupils understand the topic. The lecture method of learning is more controlled by professors as "the transferor" of science, while students are more passive as "recipients" of science, as seen above. As a result, in today's digital world, providing a web application for e-learning is critical. The recent pandemic has shown us that online education yields an excellent result for the effectiveness of students one ideal example is zoom Education at the moment the system that is present provides all but specific functionalities as a learning system.

2.1.2 Materials

The system experimented on a laptop machine. The laptop machine was an AMD machine consisting of the Windows 10 operating system, running over 8 GB of RAM. During the testing of the project, it experimented over intel machines, and thanks to the compatibility level of Mongo DB, React and JavaScripts, Node, and Express the project was running accurately as it was running over the AMD machine.

2.1.3 Flex Website

Flex Portal is the Pakistan Fast University website. Fast University is founded in July 2000. The Fast University Has five campus and this website use overall all campus of Fast. Flex has 6 features on the website 1: Home page 2: Course Registration, 3: Marks uploading, 4: Attendance, 5: Transcript, 6: Fess Detail. This website has access to teachers and admin, Teachers can only upload marks and attendance and the admin can register the course of students. [1]

2.1.4 Zoom App

Zoom Application is based on cloud computing based on communication through virtual video calls and virtual voice calls, live chats, screening share, call recording, and also share call recordings with another person. These are basic features of zoom .[2]

2.1.5 Google classroom

Google Classroom is a free platform that integrates with services, Gmails, Google Docs, and Google Classroom saves time for teachers and students. Teachers can quickly see who has not uploaded assignments and projects and teachers can add grades based on uploaded Assignments. Goggle Classroom has many features including Teachers can Upload Assignments, projects, set grades, and notifications for students. Students can see Assignments and uploaded Assignments , grades. [3]

2.1.6 Iqra university portal

Iqra University Portal is the all-campus of Iqra University. This website has many features. On this website, teachers can upload marks and attendance and students can only see marks and attendance.[4]











2.1.7 Slate portal

Slate Portal has a portal of Fast university. The slate includes 6 features include Assignments upload, Notification, Share Time Table and show all courses of students. On this website has teachers can upload Assignments and notifications about class and students can see notifications and Assignments. [5]

2.1 Comparision of with different Application

This project can be compared to a previously completed project. The table below compares the project to others.

Table 1 Comparision table

FEATURE COMPARISION TABLE										
FEATURES	 Google Classroom	 Zoom Education	 Seesaw	 Adobe Captivate	 Edmodo	 IQRA University Portal	 CBM University Portal	 Slate	 Flex Academic Portal	 EduTech
Virtual Classroom	✓	✓	✓	✓	✓	—	—	—	—	✓
Access Authentication	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Admin Privilege	✓	✓	—	✓	✓	✓	✓	✓	✓	✓
Media Sharing	✓	✓	—	✓	✓	✓	✓	✓	—	✓
Assignments	✓	—	✓	✓	✓	✓	✓	✓	—	✓
Online Quiz	—	—	—	—	✓	✓	✓	✓	—	✓
Video Conferencing	—	✓	—	—	—	✓	—	—	—	✓
Digital Whiteboard	—	✓	✓	—	—	—	—	—	—	✓
Grading	✓	—	✓	✓	✓	✓	—	✓	✓	✓
Academic Reports	—	—	✓	—	—	✓	✓	—	✓	✓
Course Registration	—	—	—	—	—	✓	✓	—	✓	✓
Attendance	—	—	—	—	—	✓	✓	—	✓	✓
Fee Challan	—	—	—	—	—	—	—	—	✓	✓
Calendar System	✓	✓	—	—	—	—	—	✓	—	✓
Push Notifications	✓	✓	—	—	✓	✓	—	—	—	✓

Feature Comparision Table

Chapter 3

System Model

This chapter goes over all of the phases of the system's design and requirements, as well as the functional and non-functional needs, the project's significance, and the tools and technologies employed in this project.

3.1 Functional and a Non-Functional Requirement

3.1.1 Functional Requirement

After gathering valuable information, we reached the critical conclusions are as follows:

1. DPR is able to meet the functional requirements as mentioned in the FYP Scope.
2. To maintain the system as per technology requirements.
3. Make the system easy to understand for the user.

3.1.2 Non-Functional Requirement

The working criteria of the web application are that the web application will launch over localhost access. The user will see the main page from here they can redirect to their desired form.

I. Usability

The usability of the form is manual and easy to understand as all the fields are related to their desired department with the amount of data to be entered in the entry section.

II. Reliability

As far as the reliability is concerned, the application will work as a service therefore no interference from the user would occur until any user interacts with the application. Then the application would become more reliable.

III. Performance

The Performance will depend on the specifications of the machine. The slow the processing speed, the slow will be the response of the interface.

IV. Supportability

A Minimum of a 6 GB Ram and 512 GB of HDD is required for the processing of the web application.

V. Implementation

The implementation of the project is not that hard as the user can easily access the webpage and use it.

VI. Interface

The Interface is very simple and easy to understand

VII. Legal

The Minimum of Windows-7 will be required for the web application to support.

3.2 System Design

This chapter is about the design of the system and figures with their notations that represent the system. It discusses the different types of diagrams that are used in the system development process among which some of us have used in this system. The Smart e-learning Management system is designed using the agile process model to provide the utmost feasibilities to the students and the admin while using it as it provides a quick response to changing demands while mitigating the design allows the students to easily access their preferred features by reducing the cognitive load while navigating .by keeping the user interface minimal the load time pages are reduced this is also done through reusable component in react JS.

3.2.1 Overall System Flow Diagram

In this diagram we can see the whole making process of this system from start to end.

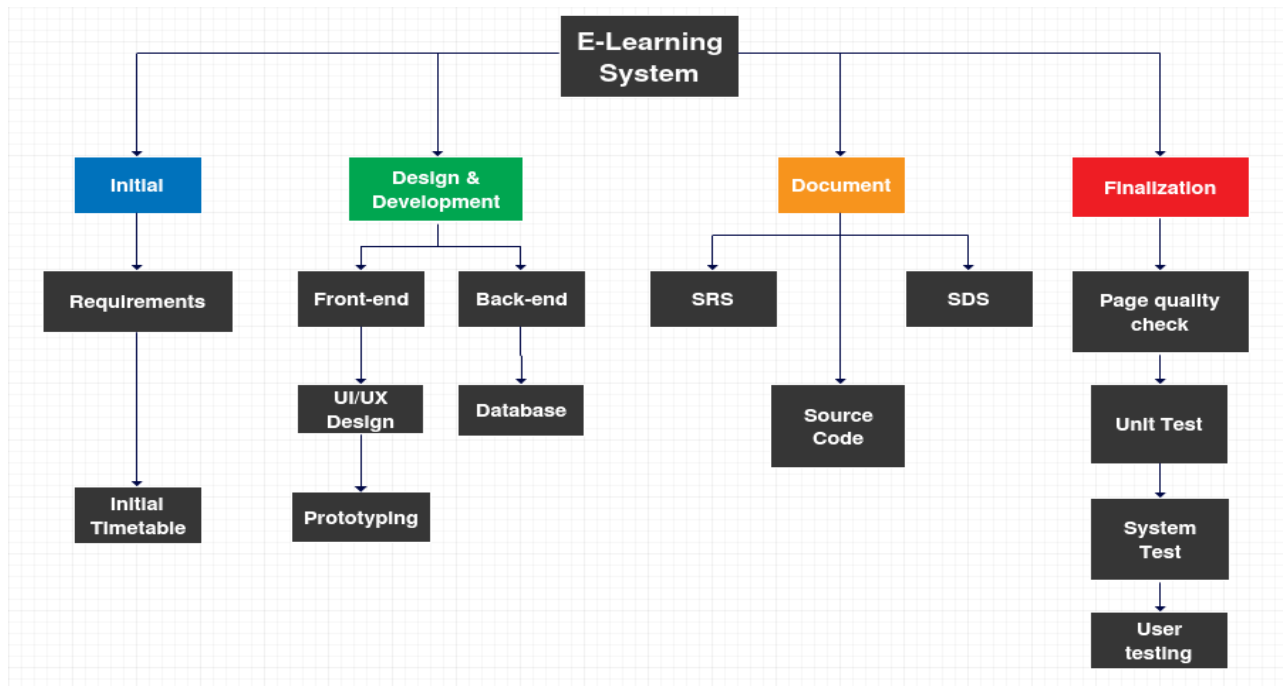


Fig-3: Overall System Flow Diagram

3.3 System Architecture/Design

For the implementation process, we have decided to use the MERN stack framework that consists of Mongo DB (for creating and managing the database as it enables high data availability with automatic, quick, and immediate data recovery through its cloud service), ExpressJS to provide a web framework for NodeJS, ReactJS (a client-side JavaScript framework) for building the frontend and NodeJS for premiering the JavaScript web server. We have made use of the Insomnia tool for handling the API requests at the backend. The trade-offs while using MERN are that it provides The UI has a lot of interactivity, and it saves a significant amount of data transformation code In addition, unlike other stacks, you don't need a template language to create pages.

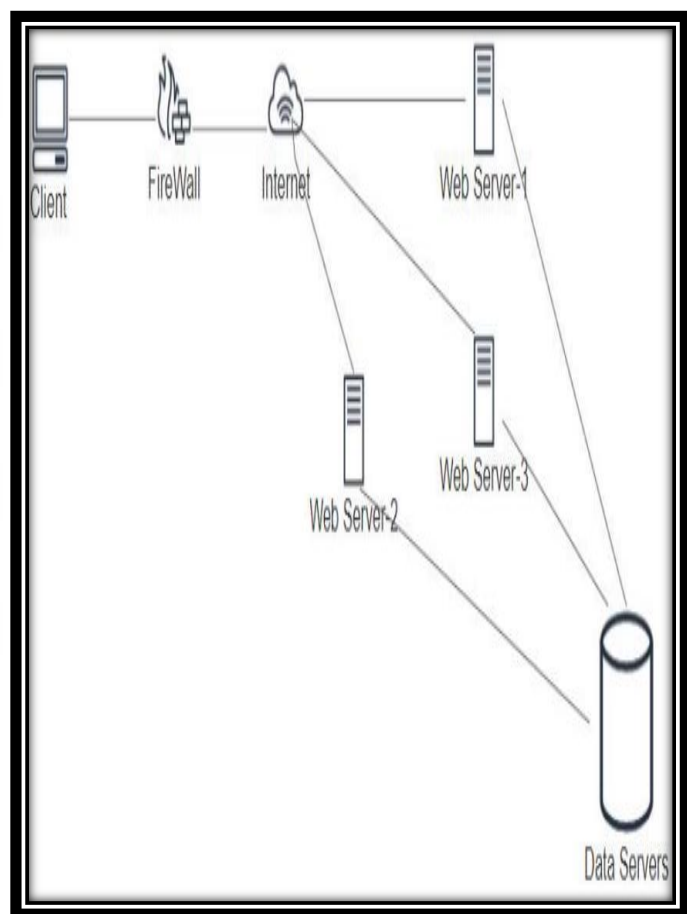
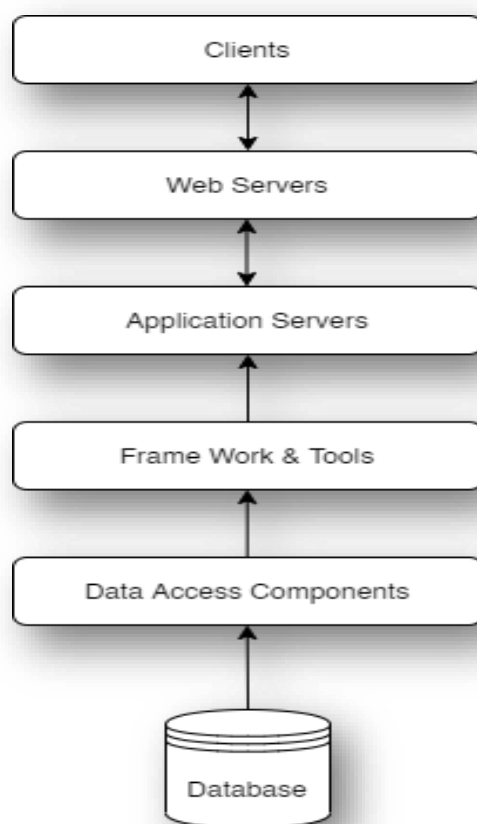


Fig-4: System architecture /Design

3.1 Scalability

An e-learning management system is an e-learning platform that can scalability to suit the changing needs of your users and organization. It's an online training platform that can handle more users, more content, and more requests without slowing down the loading time or introducing errors. Existing learning management systems can be tweaked to allow for scaling, or a new SLMS can be built from the ground up with scalability in mind. Take a look at this case study to see how we helped a client scale their WordPress site to 1 million active users. There are numerous reasons why an organization would require a scalability SLMS. First and foremost, you don't want to build or invest in an SLMS that can't scale with your company or user population. If your SLMS is preventing you from expanding, you should seriously explore some of the options listed below to assist you.

Finally, having a scalable SLMS allows you to:

- 1: Handle a large amount of data and multiple requests at the same time.
- 2: Allow millions of people to use your system (including high levels of concurrent users)
- 3: Easily organize your users and manage their rights and courses.
- 4: High-performance and low-load times should be used to provide an intuitive experience.

3.5 Service

Service qualities are defined in E-learning as the outcome of comparison in the use of E-learning between user expectations for the service and perceptions of how the service is delivered through characterization of services such as intangibility, heterogeneity, inseparability, and perishability.

Chapter 4

System Design

This chapter is about the design of the system and figures with their notations that represent the system. It discusses the different types of diagrams that are used in the system development process among which some of us have used in this system. The Smart e-learning Management system is designed using the agile process model to provide the utmost feasibilities to the students and the admin while using it as it provides a quick response to changing demands while mitigating the design allows the students to easily access their preferred features by reducing the cognitive load while navigating .by keeping the user interface minimal the load time pages are reduced this is also done through reusable component in react JS.

4.1 UML Diagram

This is an activity UML diagram of a Smart e-learning management system that allows for the flow of information between course classes, schedules, students, and teachers. The key actor involved in this UML activity diagram of e-learning is learning.

4.1.1 Use Case Diagram

The interactions among the pieces of the E-Learning Management System are graphically depicted in this Use Case Diagram. It is the process used in system analysis to discover, clarify, and organize E-Learning Management System system requirements. In this Use Case Diagram, the primary actors of the E-Learning Management System are System User and Super Admin. Teachers and students who conduct various use cases such as Manage Courses, Manage Students, Manage Schedules, Manage Fees, Manage Durations, and Manage Training. Manage subjects, users, and the entire E-Learning Management System. The major components of the E-Learning Management System UML use case diagram are depicted in the graphic below.[10]

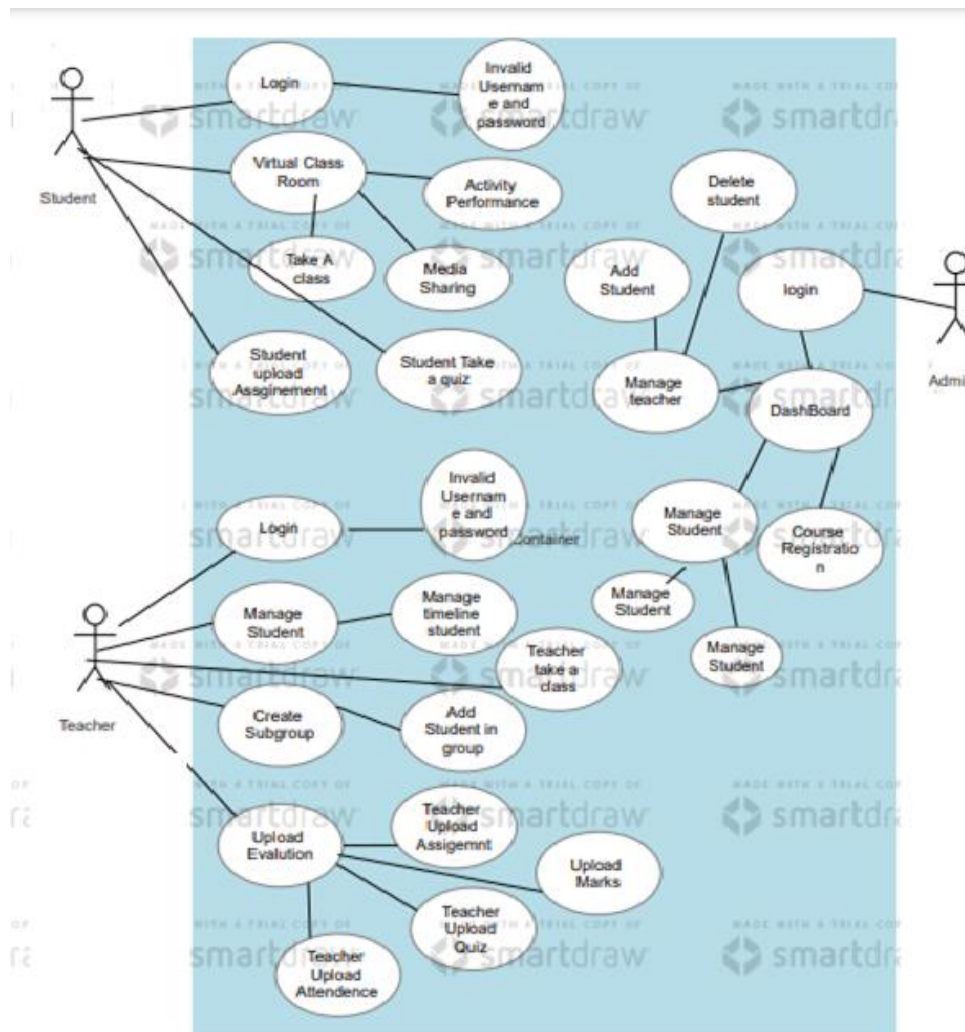


Fig-5: Use Case Daigram [10]

4.1.2 Activity Diagram

This is a diagram of an activity. The Admin User can search for Courses, examine their descriptions, add Courses, amend Courses, and delete Courses. It depicts the editing, adding, and updating of the Durations activity flow. The user will be able to search for training, schedules, and students and generate reports. All items are interconnected, such as courses, durations, and students. It contains a complete explanation and flow of the Course, Schedules, Students, Training, and Duration. [10]

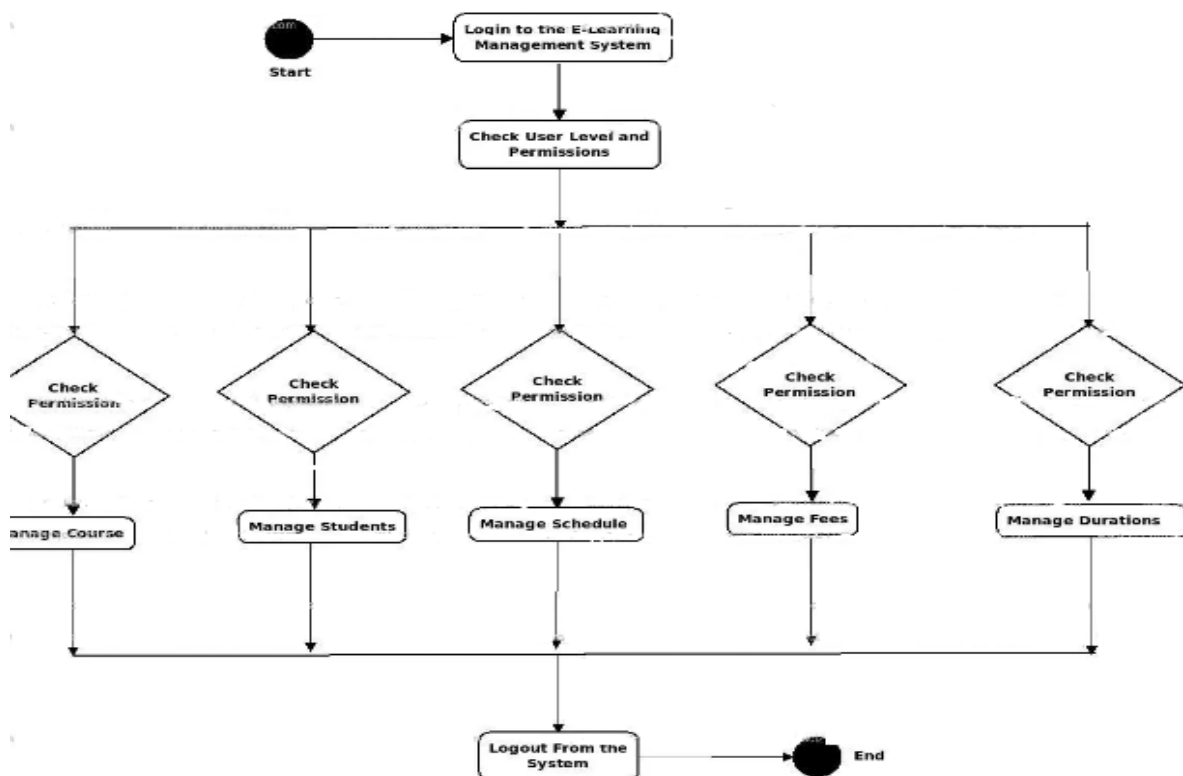


Fig-6: Activity Diagram

4.1.3 Sequence Diagram

This is an E-Learning Management System Sequence Diagram that depicts the sequence, offered and necessary interfaces, ports, and linkages between Courses, Schedules, Fees, Students, and Subjects. This sort of graphic is used to represent systems using Service-Oriented Architecture in Sequence-Based Development (SBD) . The ELearning Management System UML Sequence diagram depicts the physical system's organization and wiring [10]

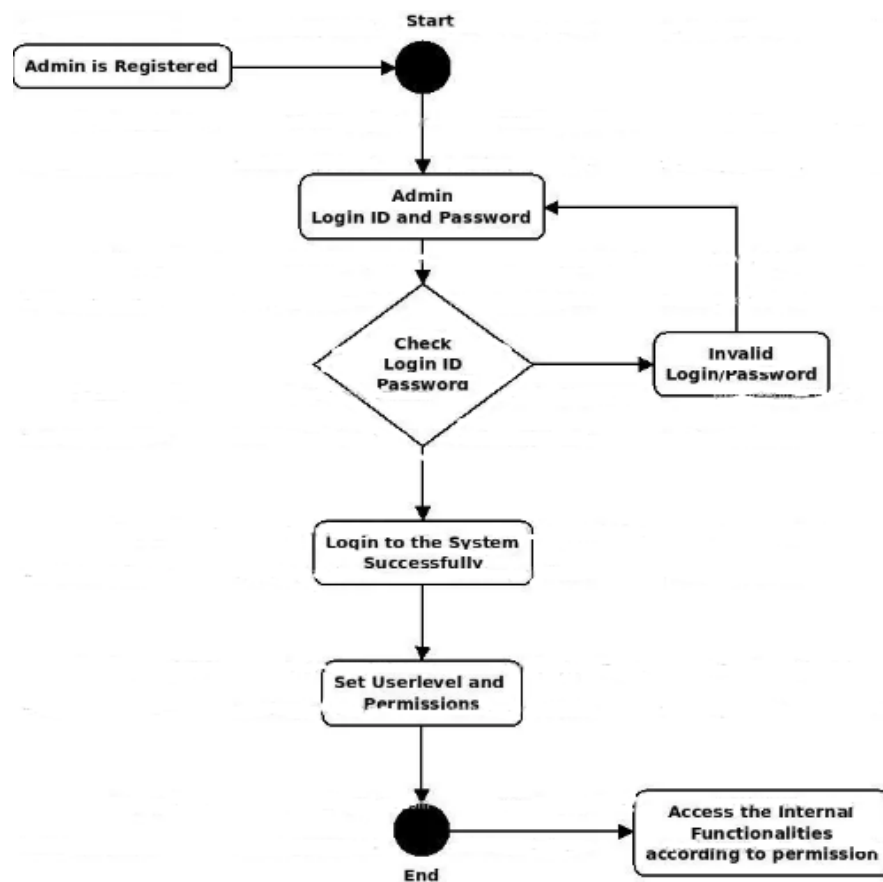


Fig-7: Sequence Diagram

4.1.4 Class Diagram

The E-Learning Management System Class Diagram depicts the structure of E-Learning Management System classes, as well as their properties, actions (or methods), and object relationships. Course, Students, Schedules, and g are the main classes of the E-Learning Management System.[10]

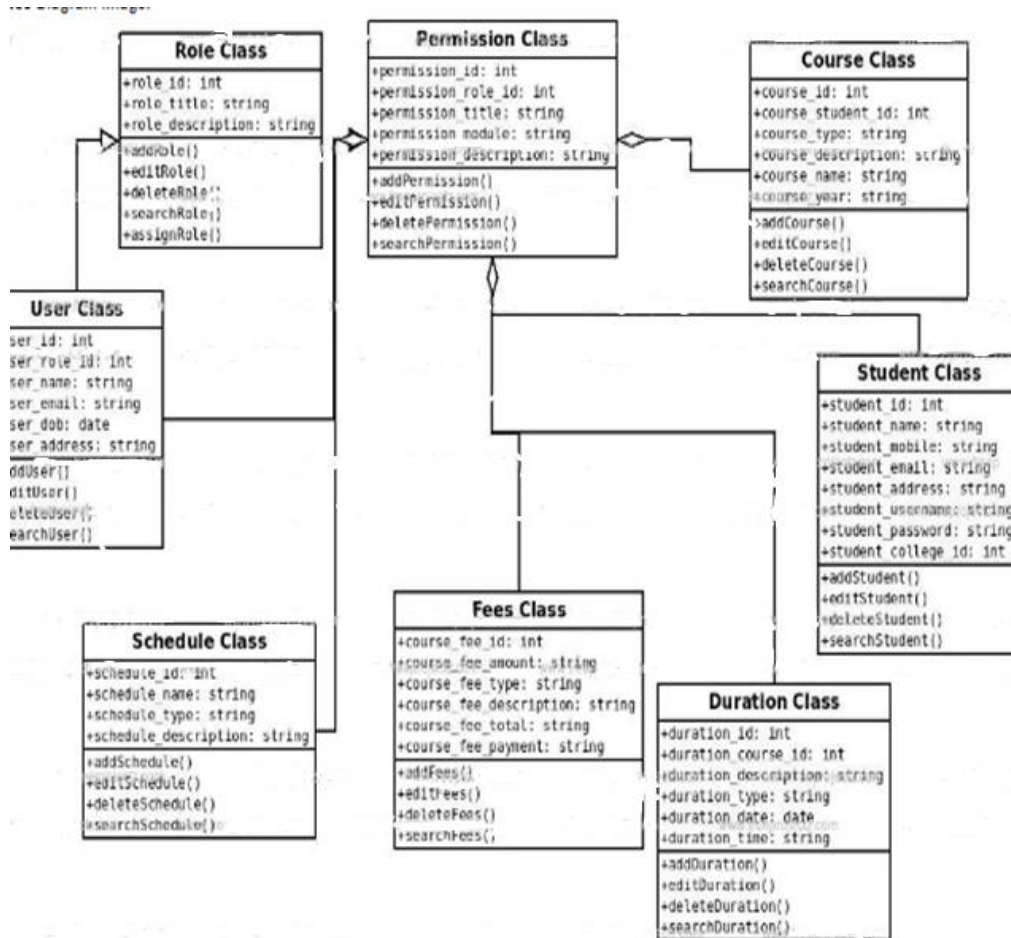


Fig-8: Class Diagram

4.2 software platform

Microsoft Visual Studio Code (VS-Code) is a Microsoft integrated development environment. It's used to make websites, web apps, web services, and mobile apps, among other things. Visual Studio provides a foundation for implementing session handling throughout the project.

4.2.1 Operating System

- System requirements: Microsoft Windows 10, 64-bit x64-based processor.
- CPU type: AMD A6-3410MX APU with Radeon™ HD Graphics 1.6 GHz
- Memory: 8 GB RAM / 1 TB HDD

4.2.2 Firebase:

Backend-as-a-Service provider Firebase (Baas). It offers a number of tools and services to assist developers create high-quality apps, expand their user base, and make money. It is based on Google's technology. Firebase is a NoSQL database application that saves data in JSON-like documents

4.2.3 HP Laptop:

A laptop with internet access

4.1 Schema Design Diagram

Mongo DB is used for creating and managing the database as it provides high availability of data with automatic, fast and instant data recovery.

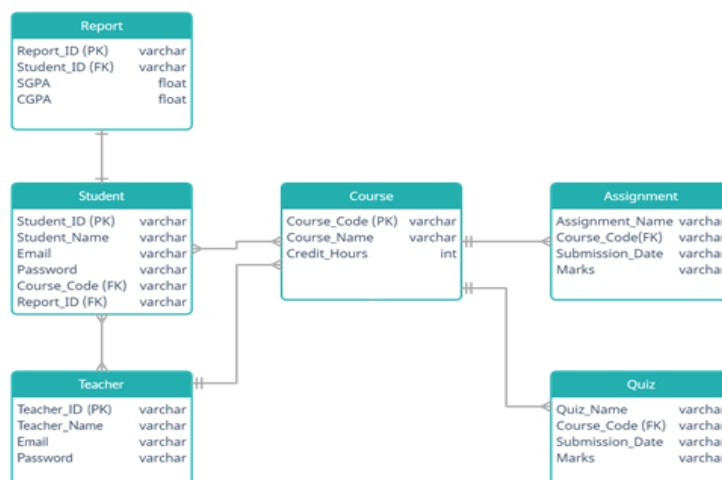


Fig-9: Schema Design Diagram

4.2 Project Inside

This e-Learning is a time-consuming procedure. Online students can text or video call an online instructor, do Online assignments and paper submissions, and other work but most of the time it's just them and the computer. An e-Learning course requires particular attention during the planning phase of the e-Learning Project to guarantee that the learning objectives are met. Even before they begin the design and production process, eLearning teams build the groundwork for success. Here's how to plan your next e-Learning project step by step.

4.2.1 React.js

React.js is an open-source JavaScript package that is used to create single-page applications' user interfaces. For web and mobile apps, it's utilized to manage the view layer. We can also make reusable UI components with React.

4.2.2 MongoDB

MongoDB is a document-oriented database that is open source and cross-platform. MongoDB is a NoSQL database application that uses JSON-like documents with schemas that may be customized. MongoDB Inc. is the company that created the database.

4.2.3 Express.js

The Express.js server-side framework, which runs inside a Node.js server, is the next step down. Express.js describes itself as a "quick, unopinionated, minimalist web framework for Node.js," and it is precisely that. For URL routing (matching an incoming URL with a server function) and handling HTTP requests and answers, Express.js includes strong models. You can connect to Express.js functions that power your application by sending XML HTTP Requests (XHRs), GETs, or POSTs from your React.js front-end. To access and change data in your MongoDB database, those functions use MongoDB's Node.js drivers, either via callbacks or Promises.

4.2.4 Node.js

Node.js is a JavaScript runtime environment that is open-source and cross-platform. It's a versatile tool that can be used for practically any project! Outside of Google Chrome, Node.js runs the V8 JavaScript engine. As a result, Node.js is extremely quick. Without establishing a new thread for each request, a Node.js application operates in a single process. Node.js' standard library contains a set of asynchronous I/O primitives that prevent JavaScript code from blocking, and libraries in Node.js are often designed in non-blocking paradigms, making blocking behavior the exception rather than the rule.

Chapter 5

System Implementation

5.1 Project Timeline

Activities are taking to complete in following given time. Each activity taking different time to complete. Here is the start date and end date with total duration (in days):

ACTIVITY	START DATE	END DATE	DURATION (IN DAYS)
Literature Review	9/11/2020	13/12/2020	20
Research	20/11/2020	30/11/2020	10
Coding	1/12/2020	1/5/2021	150
Front-End	1/12/2020	1/2/2021	50
Back-End	1/2/2021	1/6/2021	60
Finalizing	14/6/2021	14/11/2021	60
Report	1/12/2021	30/12/2021	15

5.2 Project Phases

Following were the periods or Phases of the undertaking also the name and responsibility of a person who performed which phase and he is responsible for that.

No.	Phase	Duration	Action Performed	Responsible Person
1	Gathering Requirements	35 Days	Requirements that are both non-functional and functional	Abdul Ghani Sagar Abbas Sadam Hussain
2	Analysis of Requirements	30 Days	System and functional analysis	Abdul Ghani Sagar Abbas Sadam Hussain
3	Designing	30 Days	User Interface and Front-End	Abdul Ghani Sagar Abbas Sadam Hussain
4	Development	60 Days	Back-end and session handling	Abdul Ghani Sagar Abbas Sadam Hussain
5	Testing	28 Days	AMD and Intel Machine responses	Abdul Ghani Sagar Abbas Sadam Hussain
6	Documentation	25 Days	FYP Document initializing and compiling	Abdul Ghani Sagar Abbas
7	Overall Project	208 Days		

5.2 Project Implementation

For the implementation process, we have decided to use the MERN stack framework that consists of Mongo DB (for creating and managing the database as it provides high availability of data with automatic, fast and instant data recovery through its cloud service), ExpressJS to provide a web framework for NodeJS, ReactJS (a client-side JavaScript framework) for building the frontend and NodeJS for premiering the JavaScript web server. We have made use of the Insomnia tool for handling the API requests at the backend. The trade-offs while using MERN are that it provides a large amount of interactivity built into the frontend and it saves a lot of data transformation code. Plus you don't even need a template language to generate pages like other stacks.

5.2 Modules for DPR

There are 7 modules in DPR

- Main Login Screen Panel
- Admin Screen Panel
- Teacher Screen Panel
- Student Screen Panel
- Saving the entered data
- Logout
- Returning to main screen

Main Screen

The main screen panel is showing the all courses.

Login Screen

The login screen is same in terms of front-end but the login initials of each user.

Data Entry

Data gathered from different resources is manually entered by the user by terms of field.

Saving the entered data

The entered data is saved in the Mongod database using JS.

Logout

Once the data has been saved, the user logout his/her account.

Returning to Main Screen

Once the user is logout, the user is redirected to the main screen of the web application.

5.2.1 Login Model

Here we can see and access the main login screen of this web. where user can easily login to his account using password and id and perform their daily activity work.

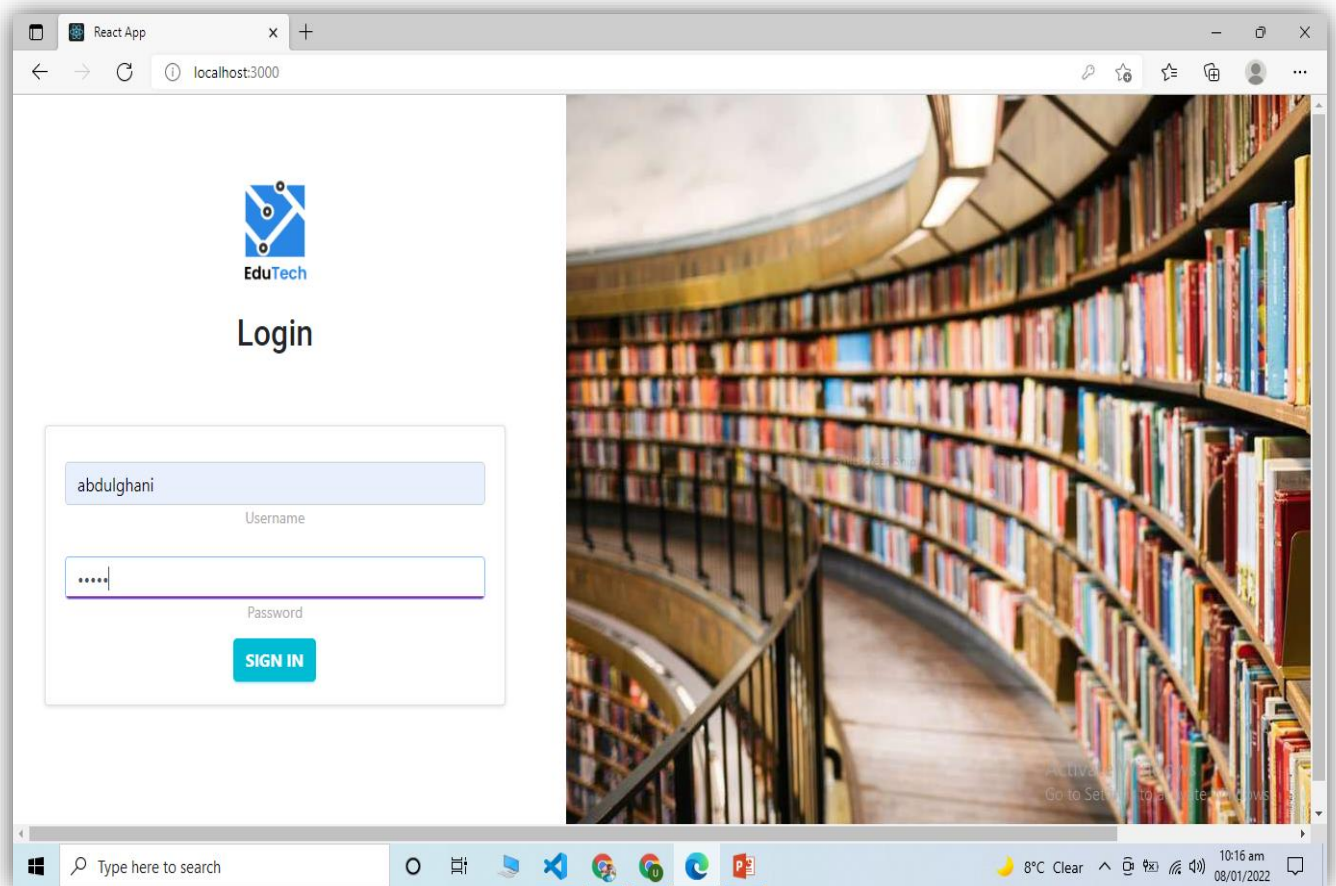


Fig-10: Login Model

5.2.1 Dashboard Model

Here is the teachers dashboard screen where they can perform their daily activity work. they can Assign quizzes and assignment to their students also check their student work.

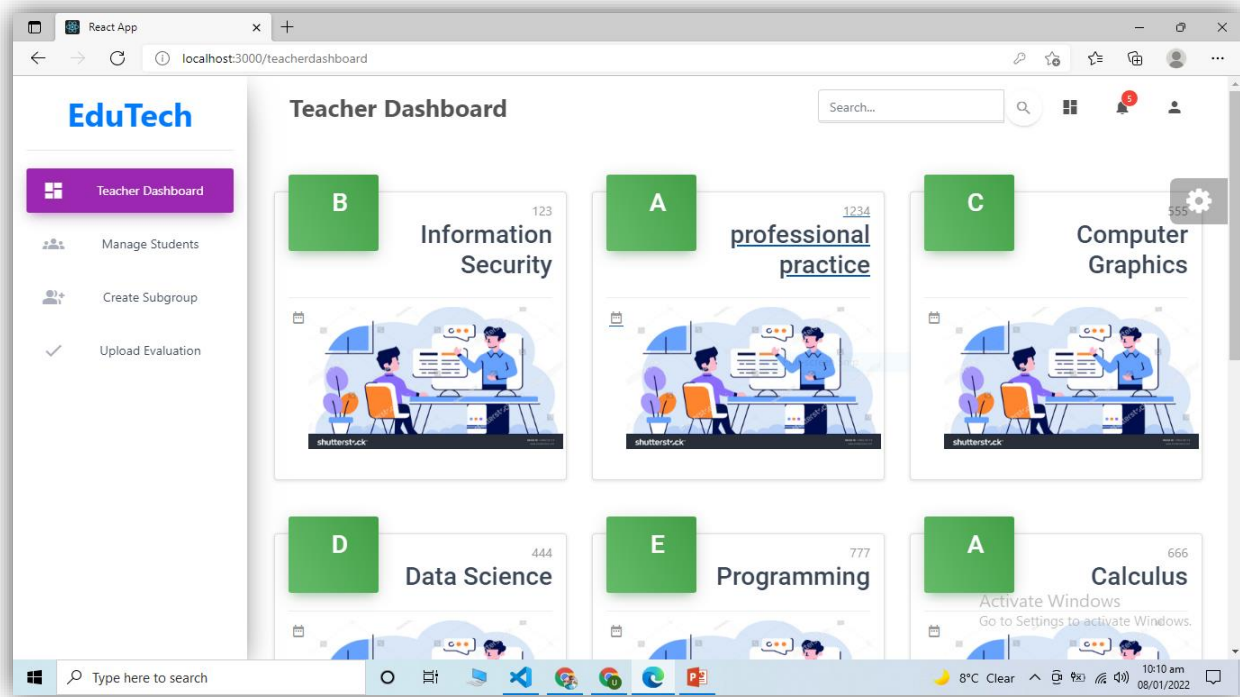


Fig-11: Admin Dashboard Model

5.2.2 Student Dashboard

Here is the students dashboard screen where they can perform their daily work.

They can access their any registered course and complete their work easily.

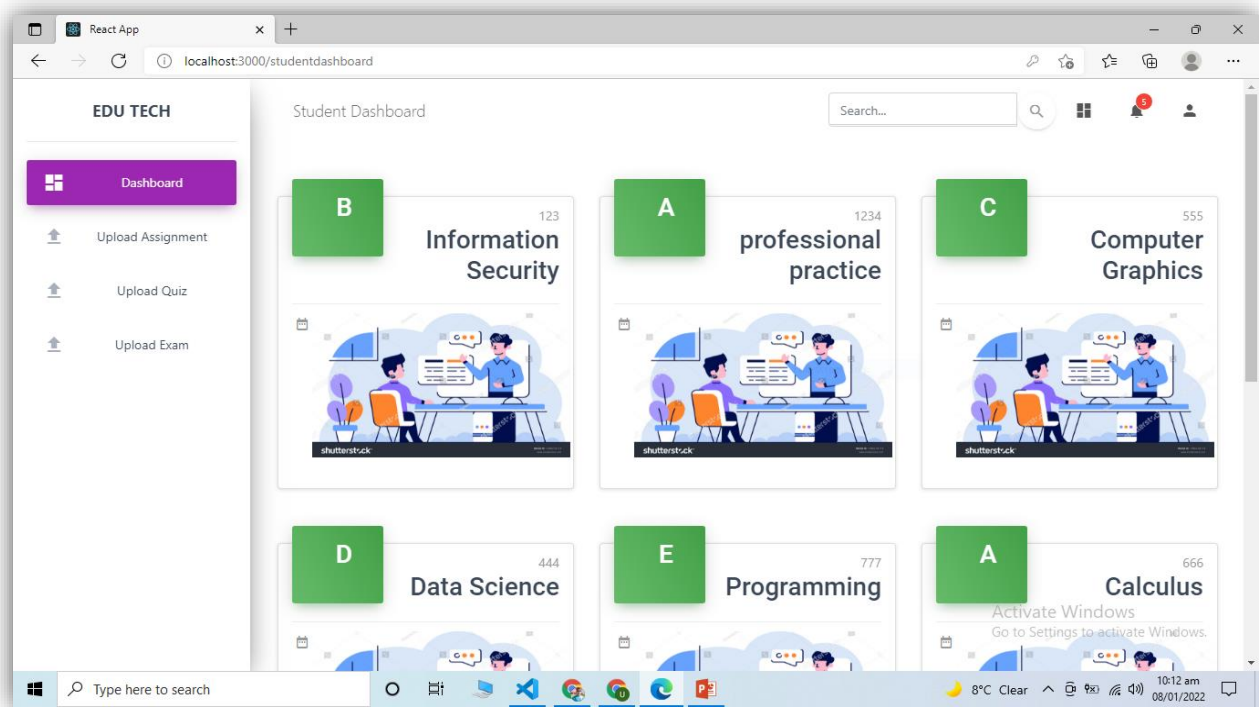


Fig-12: Student Dahshboard Model

5.2.3 Student Profile

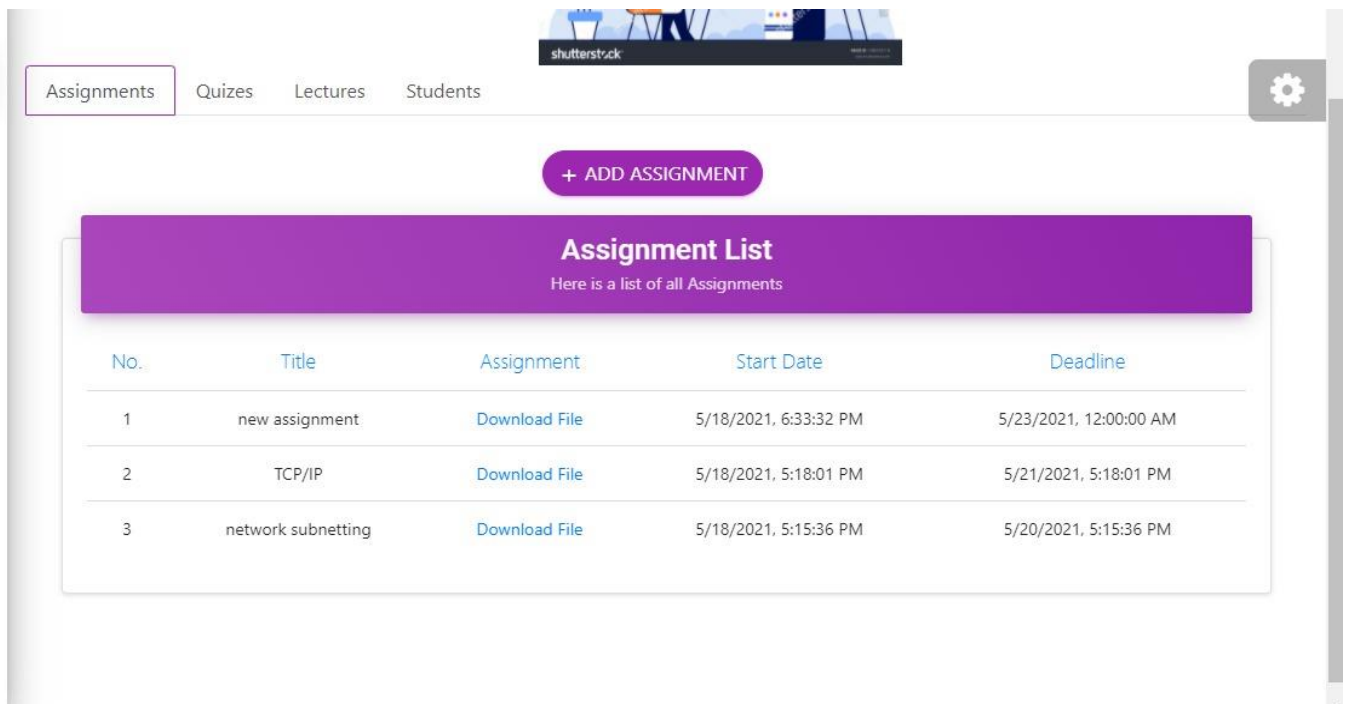
Here is the students profile screen where they can update their profile example student profile pic their also they update their password.

The screenshot displays a web application interface for a student profile. The browser window shows the URL 'localhost:3000/profile'. The page layout includes a sidebar on the left with links for 'Overview' and 'Settings'. The main content area features a profile card for '@ghani' with a 'CHANGE PHOTO' button. Below the profile card is a 'Settings' section with input fields for 'First Name', 'Last Name', and 'Username' (which contains the text 'qhani'). A 'SAVE CHANGES' button is positioned below these fields. Further down is a 'Change Password' section with three input fields: 'Current Password', 'New Password', and 'Confirm Password', each followed by a series of dots for password masking. A 'SAVE PASSWORD' button is located at the bottom of this section. On the right side of the page, there is a 'LOGOUT' button, a 'SUPPORT' section with the text 'Get fast, free help from our friendly assistants.' and a 'CONTACT US' button, and an 'Activate Windows' watermark at the bottom right. The Windows taskbar at the bottom shows the search bar, task view button, and several application icons, along with system information like '19°C Cloudy' and the date '26/01/2022'.

Fig-13: student Profile Model

5.2.4 Main Working Dashboard

Here we can see that students can see their assignment and other works start and end date. So they can manage and complete their work before end date.



The screenshot displays a web application interface for a dashboard. At the top, there is a navigation bar with tabs for 'Assignments', 'Quizes', 'Lectures', and 'Students'. The 'Assignments' tab is currently selected. To the right of the navigation bar is a settings icon (a gear). Below the navigation bar, there is a purple button labeled '+ ADD ASSIGNMENT'. The main content area features a purple header for 'Assignment List' with the subtitle 'Here is a list of all Assignments'. Below this header is a table with five columns: 'No.', 'Title', 'Assignment', 'Start Date', and 'Deadline'. The table contains three rows of data, each with a 'Download File' link in the 'Assignment' column.

No.	Title	Assignment	Start Date	Deadline
1	new assignment	Download File	5/18/2021, 6:33:32 PM	5/23/2021, 12:00:00 AM
2	TCP/IP	Download File	5/18/2021, 5:18:01 PM	5/21/2021, 5:18:01 PM
3	network subnetting	Download File	5/18/2021, 5:15:36 PM	5/20/2021, 5:15:36 PM

Fig-14: Main Working Model panel

5.2.5 Admin Dashboard

Here we can see admin dashboard screen where admin can perform his daily operations.

He can access any course also he can add student to that course. He can also assign a teacher to that Course. Also he can add new teacher and new student. Also he can Manage student and class.

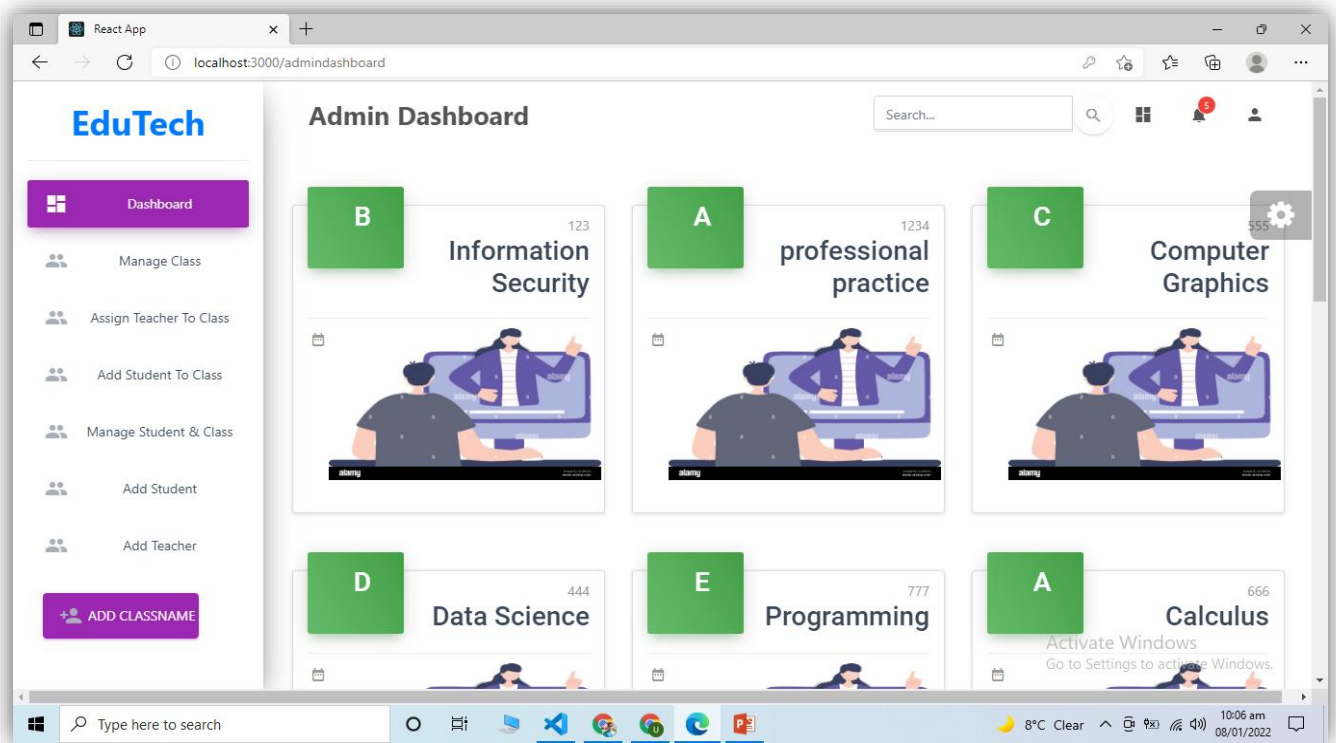


Fig-11: Admin Dashboard

5.2.6 Password change Model

We see the password update screen where admin can update his password but he have enter their current password to update their account.without current password he can not change his password.

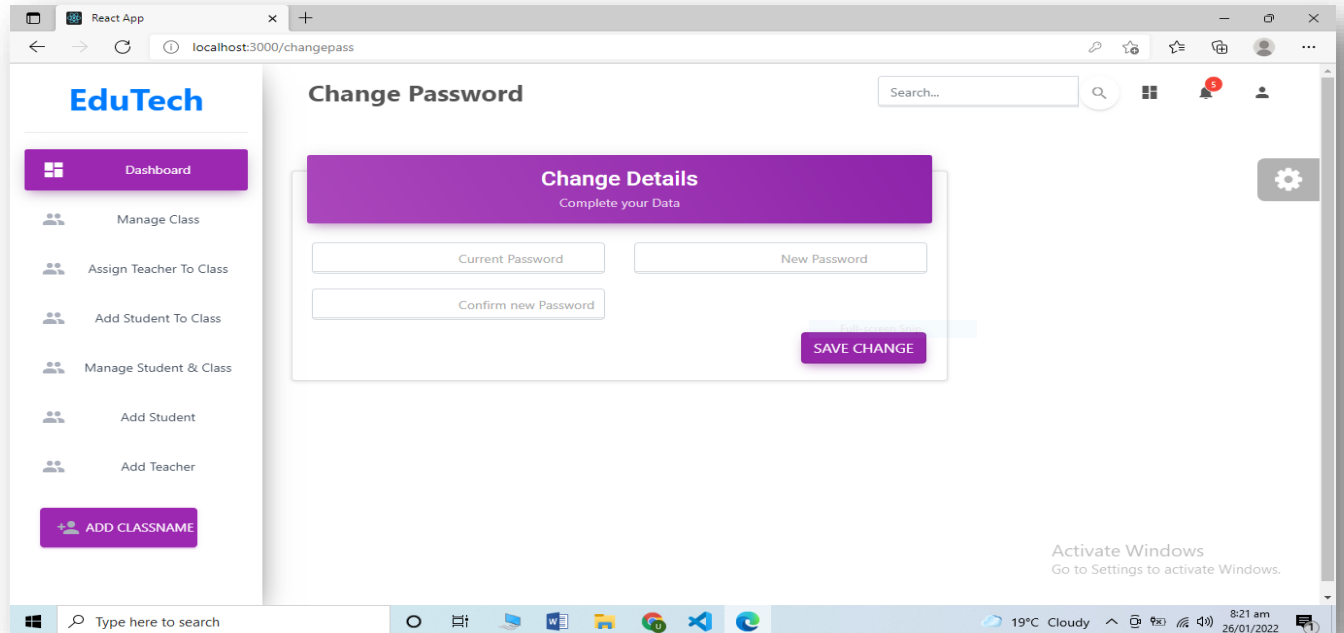


Fig-15: Password Change Model

Chapter 6

Results And Discussion

6.1 Testing

Software testing is not only error detection. The functionality of testing is a web application is following the conditions and also confirming what the user really wanted or what web application was proposed.

1. Verification is testing or you're checking items, such as web applications, by assessing the results for conformance and consistency. Verification: Are we correctly designing the system?
2. Detection of Errors To see if things happen when things go wrong, testing should purposefully strive to make things go wrong. things don't happen when they should or they shouldn't.
3. Validation examines the correctness of the system - i.e. the procedure of creating what is provided and what is required. Validation: Is our system organized and designed properly? To put it another way, validation checks to see if we're developing the right application, verification checks to make sure we're constructing the right system, and what the client wants/needs. Verification and validation are both necessary but distinct aspects of any testing operation.

6.1.1 The Goal of the Test

Testing is done for two reasons: to monitor hazards and to double-check acquirement decisions. The first step of testing ensures that what is specified is delivered; it ensures that the system meets the operational, performance, design, and execution requirements outlined in the procurement specifications. For the developer/vendor, testing is all about risk management.

6.2 Test Cases

6.2.1 Test Case 1

Title: course registration.

Test Case Description	Admin should be able to assign specific courses to students through backend
Test Procedure	1. Get the authentication token of the logged in student 2. Fill out the course details 3. Send the API request
Test Data	Course name, code, credit hours
Expected Result	Courses are registered to specific students
Actual Result	Courses are registered to specific students
Status	Pass
Priority	Major
Remarks	This is a unit test for a feature of course registration

Table 5.1

6.2.2 Test Case 2

Title: Course Addition.

Test Case Description	Teacher should be able to add courses to be displayed to the students
Test Procedure	1. Fill out the course details in the form provided 2. Submit the form
Test Data	Course name, code, credit hours
Expected Result	Courses are added to be displayed or registered
Actual Result	Courses are added to be displayed or registered
Status	Pass
Priority	Major
Remarks	This is a unit test for a feature of course addition

Table 5.2

6.2.3 Test Case 3

Title: : Browser Compatibility

Test Case Description	The site should launch properly on all the major browsers
Test Procedure	Run the application on the local host on the specific browser
Test Data	Browser
Expected Result	The site launches and functions properly on the browser
Actual Result	The site launches and functions properly on the browser
Status	Pass
Priority	Major
Remarks	This is a unit test for checking browser compatibility

Table 5.3

6.2.4 Test Case 4

Title: 404 Page

Test Case Description	The site displays an error 404 page if an unidentified URL is opened
Test Procedure	Redirect the application to an unidentified URL
Test Data	URL
Expected Result	The site displays an error 404 page
Actual Result	The site displays an error 404 page
Status	Pass
Priority	Minor
Remarks	This is a unit test for checking existence of 404 page

Table 5.4

Chapter 7

Conclusion and Feature Work

7.1 The Project's Strength

This project has a few important strengths that set it apart from other activities or improvements; let's take a look at a couple of them.

7.1.1 Real Time Data Saving

As the user saves data in the firebase database, the saved data can be seen in the database as it is uniquely identified by its key.

7.1.2 Recent Technology

We used MERN Stack: MERN Stack is a Javascript Stack that is used for sending full-stack web applications in a more straightforward and faster manner. MongoDB, Express, React, and Node.js are the four innovations that make up the MERN Stack. Its goal is to make the improvement process easier and more efficient. Each of these four significant developments provides a complete system for engineers to work in, and each of these advancements has a significant impact on the development of online applications. Getting everything rolling: First, make another venture envelope. Then, at that point, go to the undertaking organizer in order brief/terminal and type the beneath order to introduce a package.json record. (Ensure npm is introduced).

7.1.3 Enhanced Security

As firebase have the firewall and security of Google, the same proxy is used for firebase to protect user data

7.2 Problems Encountered

When we started working on the project, we faced many delays in gathering information as it was to come from e-learning as per their requirements. As it is a web-based application so the designing was also a critical task as the organization wanted the design as per their requirement and due to covid-19 it was hard to interact and gather the required information.

7.3 Conclusion

To conclude, We have made up a system that has been the need of the hour. We have been having this thought of promoting digitalization. We have promoted and contributed to this era of technology by proving ourselves the best in our work. The project has been implemented accordingly with respect to the requirement of an institution. The Project team has designed and enhanced the daily performance reporting Form for daily data entry. The aim of this project is to create Smart E-Learning Management Systems (SLMS) is a critical programming stage for delivering university and instructional classes, as well as other business processes, via the internet. They facilitate the creation, organization, and distribution of educational content, making it easier to handle educational content. A framework like LMS should also be simple to use and effective to employ.

7.4 Future Work

It is a notion that wherever we reach, there is always room and space for more learning. We have implemented adequate work and knowledge in this project as per the current needs. However, there is always room for enhancement. We have also enhanced and modified to a great extent an already existing project in the market. Our project fulfills the criteria for better work and improved quality. Nevertheless, the room for enhancing our project is always open in the air. We have saved the credentials to ourselves for any copyrights. We more than welcome everyone who wishes to contribute in our project by making the world a better place. For further information to the project, the webapplication was created on React, Express, and JS; it can be further enhanced through the following:

1. Create mobile application

The web application can be converted into a mobile application through which it can be easy for the user to access his/her webpage.

2. More Compatible

Make the system more compatible related to operating systems so that there is no technology lacking in the project.

Reference

- [1] “Flex Website” [Online] Available: <https://flexstudent.nu.edu.pk/Login>. [Accessed 22 Feb 2022].
- [2] “Zoom App” [Online] Available:
https://scholar.google.com.pk/scholar?q=zoom+app+research&hl=en&as_sdt=0&as_vis=1&oi=scholar. [Accessed 22 Feb 2022].
- [3] “Google Classroom” [Online] Available:
https://edu.google.com/products/classroom/?gclid=Cj0KCQjw0umSBhDrARIsAH7FCocKuF51wcS1ygZZt4rbLO9ACJBDP-pxo0Eath7nL2qIqhMfDR0dfnYaAoZ3EALw_wcB.
[Accessed 22 Feb 2022].
- [4] “Iqra University Portail” [Online] Available: <https://iulms.edu.pk/login/index.php>.
[Accessed 22 Feb 2022].
- [5] “Slate Website” [Online] Available: <http://slate.nu.edu.pk/portal/relogin>. [Accessed 22 Feb 2022].
- [6] “Helwan University”, [Online] Available: <http://www.helwan.edu.eg/english/>. [Accessed 22 Feb 2022].
- [7] “Schoology learning management system (LMS)” [Online] Available:
<https://www.monashcollege.edu.au/virtual-classroom>.
[Accessed 22 Feb 2022].
- [8] “Globalized e-learning cultural challenges”, [Online] Available:
<https://www.igi-global.com/chapter/globalized-learning-cultural-challenges/19303>.
[Accessed 22 Feb 2022].
- [9] Weaver, D., Spratt, C., & Nair, C. S. (2008). Academic and student use of a learning management system: Implications for quality. Australasian Journal of Educational Technology, 24(1), 30-41[Online] Available:
<https://research.monash.edu/en/publications/academic-and-student-use-of-a-learning-management-system-implicat>. [Accessed 22 Feb 2022].
- [10] “UML-Digram”, [Online] Available:
<https://cloud.smartdraw.com/share.aspx/?pubDocShare=25D86011BA4A79F6EBBA8DE27293B9472BA>.
[Accessed 22 Feb 2022].

Appendices:

Appendix-A: App.Css

```
import './App.css';

import { BrowserRouter as Router, Switch, Route } from 'react-router-dom';

import Menu from './Admin/Menu.js';

import Studentlogin from './Student/Studentlogin.js';
import StuProfile from './Student/StuProfile.js';
import Studentsignup from './Student/Studentsignup.js';
import StudentDashboard from './Student/StudentDashboard.js';
import JoinClassByStudent from './Student/JoinClassByStudent';
import ChangepassStu from './Student/ChangepassStu';
import AttemptAssignmentByStu from './Student/AttemptAssignmentByStu';
import AttemptExamByStu from './Student/AttemptExamByStu';
import AttemptQuizByStu from './Student/AttemptQuizByStu';
import JoinAudioMeetingByStu from './Student/JoinAudioMeetingByStu';
import JoinVideoMeetingByStu from './Student/JoinVideoMeetingByStu';
import JoinChatMeetingByStu from './Student/JoinChatMeetingByStu';
import NotificationsStu from './Student/NotificationsStu';

import Teacherlogin from './Teacher/Teacherlogin.js';
import TeacherProfile from './Teacher/TeacherProfile.js';
import Teachersignup from './Teacher/Teachersignup.js';
import TeacherDashboard from './Teacher/TeacherDashboard.js';
import Changepasstea from './Teacher/Changepasstea.js';
import CclassByTeacher from './Teacher/CclassByTeacher';
import ListAllClassesByTea from './Teacher/ListAllClassesByTea';
import ManageStuByTeacher from './Teacher/ManageStuByTeacher';
import AddstuByTeacher from './Teacher/AddstuByTeacher';
```

```

import DeletestuByTeacher from "./Teacher/DeletestuByTeacher";
import CreatAssignmentByTeacher from "./Teacher/CreatAssignmentByTeacher";
import CreatExamByTeacher from "./Teacher/CreatExamByTeacher";
import CreatQuizByTeacher from "./Teacher/CreatQuizByTeacher";
import CreatSubgroupByTeacher from "./Teacher/CreatSubgroupByTeacher";
import UploadAssessmentByTeacher from "./Teacher/UploadAssessmentByTeacher";
import EvaluteStuByTeacher from "./Teacher/EvaluteStuByTeacher";
import AddMarks from "./Teacher/AddMarks";
import ListstubyTeacher from "./Teacher/ListstubyTeacher";
import ClassDetails from "./Teacher/ClassDetails";

import Login from "./Admin/AdminLogin.js";
import Signup from "./Admin/Signup.js";
import Dashboard from "./Admin/Dashboard.js";
import ManageuserByAdmin from "./Admin/ManageuserByAdmin";
import ManageStuByAdmin from "./Admin/ManageStuByAdmin";
import ManageTeaByAdmin from "./Admin/ManageTeaByAdmin";
import ManageClassbyAdmin from "./Admin/ManageClassbyAdmin";
import AddStuByAdmin from "./Admin/AddStuByAdmin";
import DeleteStuByAdmin from "./Admin/DeleteStuByAdmin";
import AddteaByAdmin from "./Admin/AddteaByAdmin";
import DeleteTeaByAdmin from "./Admin/DeleteTeaByAdmin";
import CclassByAdmin from "./Admin/CclassByAdmin";
import ListAllclassesByAdmin from "./Admin/ListAllclassesByAdmin";
import Changepass from "./Admin/Changepass";
import Notifications from "./Admin/Notifications";
import AddStudentToClass from "./Admin/AddStudentToClass";
import AssignTeacherToClass from "./Admin/AssignTeacherToClass";
import ManageStudentAndClass from "./Admin/ManageStudentAndClass";
import ClassDetailsForStudent from "./Student/ClassDetailsForStudent";

```

```

import ShowStudentSubmission from './Teacher/ShowStudentSubmission';
import UserProfile from './UserProfile';

function App() {
  return (
    <Router>
      <div className="App">
        <Route path="/" exact component={Login} />

        <Route path="/stuprofile" exact component={StuProfile} />
        <Route path="/studentsignup" exact component={Studentsignup} />
        <Route path="/studentdashboard" exact component={StudentDashboard} />
        <Route path="/joinclassbystu" exact component={JoinClassByStudent} />
        <Route
          path="/attemptassignmentbystu"
          exact
          component={AttemptAssignmentByStu}
        />
        <Route path="/attemptexambystu" exact component={AttemptExamByStu} />
        <Route path="/attemptquizbystu" exact component={AttemptQuizByStu} />
        <Route
          path="/joinaudiomeetingbystu"
          exact
          component={JoinAudioMeetingByStu}
        />
        <Route
          path="/joinvideomeetingbystu"
          exact
          component={JoinVideoMeetingByStu}
        />
      </div>
    </Router>
  );
}

```

```

<Route
  path="/joinchatmeetingbystu"
  exact
  component={JoinChatMeetingByStu}
/>
<Route path="/changepassstu" exact component={ChangepassStu} />
<Route path="/notificationsstu" exact component={NotificationsStu} />

<Route path="/teachersignup" exact component={Teachersignup} />
<Route path="/teaprofile" exact component={TeacherProfile} />
<Route path="/teacherdashboard" exact component={TeacherDashboard} />
<Route path="/changepasstea" exact component={Changepasstea} />
<Route path="/createclassbyteacher" exact component={CclassByTeacher} />
<Route
  path="/listallclassesbyteacher"
  exact
  component={ListAllClassesByTea}
/>
<Route
  path="/managestubyteacher"
  exact
  component={ManageStuByTeacher}
/>
<Route path="/addstubyteacher" exact component={AddstuByTeacher} />
<Route
  path="/deletestubyteacher"
  exact
  component={DeletestuByTeacher}
/>
<Route

```

```

    path="/creatassignmentbyteacher"
    exact
    component={CreatAssignmentByTeacher}
  />
  <Route
    path="/createxambyteacher"
    exact
    component={CreatExamByTeacher}
  />
  <Route
    path="/creatquizbyteacher"
    exact
    component={CreatQuizByTeacher}
  />
  <Route
    path="/createsubgroupbyteacher"
    exact
    component={CreatSubgroupByTeacher}
  />
  <Route
    path="/uploadassessmentbyteacher"
    exact
    component={UploadAssessmentByTeacher}
  />
  <Route
    path="/evalutestudentbyteacher"
    exact
    component={EvaluteStuByTeacher}
  />
  <Route path="/addmarks" exact component={AddMarks} />

```

```

<Route path="/liststubyteacher" exact component={ListstubyTeacher} />
<Route path="/classdetails/:id" exact component={ClassDetails} />
<Route
  path="/classdetailsforstudents/:id"
  exact
  component={ClassDetailsForStudent}
/>
<Route path="/profile" component={UserProfile} />
<Route
  path="/show-student-submission/:id"
  component={ShowStudentSubmission}
/>
<Route path="/adminsingup" exact component={Signup} />
<Route path="/admindashboard" exact component={Dashboard} />
<Route path="/manageuserbyadmin" exact component={ManageuserByAdmin} />
<Route
  path="/add-student-by-admin"
  exact
  component={ManageStuByAdmin}
/>
<Route
  path="/add-teacher-by-admin"
  exact
  component={ManageTeaByAdmin}
/>
<Route
  path="/manage-student-by-admin"
  exact
  component={ManageStudentAndClass}
/>

```

```

<Route
  path="/manageclassbyadmin"
  exact
  component={ManageClassbyAdmin}
/>
<Route
  path="/assign-teacher-to-class"
  exact
  component={AssignTeacherToClass}
/>
<Route
  path="/add-student-to-class"
  exact
  component={AddStudentToClass}
/>
<Route path="/addstubyadmin" exact component={AddStuByAdmin} />
<Route path="/deletestubyadmin" exact component={DeleteStuByAdmin} />
<Route path="/addteaabyadmin" exact component={AddteaByAdmin} />
<Route path="/deleteteabyadmin" exact component={DeleteTeaByAdmin} />
<Route path="/createclassbyadmin" exact component={CclassByAdmin} />
<Route
  path="/listallclassesbyadmin"
  exact
  component={ListAllclassesByAdmin}
/>
<Route path="/changepass" exact component={Changepass} />
<Route path="/notifications" exact component={Notifications} />
</div>
</Router>
);

```

```
}
```

```
export default App;
```

Appendix-B: App.Test

```
import { render, screen } from '@testing-library/react';
```

```
import App from './App';
```

```
test('renders learn react link', () => {  
  render(<App />);  
  const linkElement = screen.getByText(/learn react/i);  
  expect(linkElement).toBeInTheDocument();  
});
```

Appendix-C: Index

```
import React from "react";  
import ReactDOM from "react-dom";  
import Axios from "axios";  
import "./index.css";  
import App from "./App";  
import reportWebVitals from "./reportWebVitals";  
import "bootstrap/dist/css/bootstrap.min.css";  
Axios.defaults.baseURL = "http://localhost:5000";  
ReactDOM.render(  
  <React.StrictMode>  
    <App />  
  </React.StrictMode>,  
  document.getElementById("root")  
)  
;  
reportWebVitals();
```


Appendix-D: UserProfile:

```
import axios from "axios";
import { useEffect, useState } from "react";
import { useHistory } from "react-router";

const UserProfile = () => {
  const history = useHistory();

  const [imagePreview, setImagePreview] = useState(null);
  const [file, setFile] = useState(null);
  const [username, setUsername] = useState("");
  const [firstname, setFirstname] = useState("");
  const [lastname, setLastname] = useState("");
  const [password, setPassword] = useState("");
  const [confirmPassword, setConfirmPassword] = useState("");
  const [currentPassword, setCurrentPassword] = useState("");

  useEffect(() => {
    if (!localStorage.getItem("user")) history.goBack();
    return () => {
      setFirstname("");
      setLastname("");
      setUsername("");
      setImagePreview(null);
      setFile(null);
      setCurrentPassword("");
      setPassword("");
      setConfirmPassword("");
    };
  }, []);
```

```

const handleUpdateMe = async (e) => {
  e.preventDefault();
  try {
    if ((firstname && !lastname) || (!firstname && lastname))
      return alert("Please enter first name and last name.");
    const body = new FormData();
    if (firstname && lastname) {
      body.append("firstname", firstname);
      body.append("lastname", lastname);
    }
    if (username) {
      body.append("username", username);
    }
    body.append("role", JSON.parse(localStorage.getItem("user")).role);
    if (file) {
      body.append("avatar", file);
    }
    const {
      data: { data },
    } = await axios.put(
      "http://localhost:5000/users/update-me/" +
      JSON.parse(localStorage.getItem("user")).id,
      body
    );
    setFirstname("");
    setLastname("");
    setUsername("");
    setImagePreview(null);
    setFile(null);
  }
}

```

```

localStorage.setItem("user", JSON.stringify(data));
alert("Your data has been updated.");
window.location.reload();
} catch (error) {
  error?.response?.data?.msg
    ? alert(error.response.data.msg)
    : alert(error.message);
}
};

const handleUpdatePassword = async (e) => {
  e.preventDefault();
  try {
    if (!password && !currentPassword && !confirmPassword)
      return alert("all fields are required");
    if (password !== confirmPassword)
      return alert("password and confirm-password should be same.");
    const body = {};

    body["role"] = JSON.parse(localStorage.getItem("user")).role;
    body["currentPassword"] = currentPassword;
    body["password"] = password;
    const {
      data: { data },
    } = await axios.put(
      "http://localhost:5000/users/update-password/" +
      JSON.parse(localStorage.getItem("user")).id,
      body
    );
    setCurrentPassword("");
    setPassword("");
  }

```

```

setConfirmPassword("");
alert("Your password has been updated.");
} catch (error) {
    error?.response?.data?.msg
    ? alert(error.response.data.msg)
    : alert(error.message);
}
};
return (
    <
    {/*      <link      href="https://maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-
awesome.min.css" rel="stylesheet"> */}
    <div className="container">
        <div className="row flex-lg-nowrap">
            <div className="col-12 col-lg-auto mb-3" style={{ width: "200px" }}>
                <div className="card p-3">
                    <div className="e-navlist e-navlist--active-bg">
                        <ul className="nav">
                            <li className="nav-item">
                                <a className="nav-link px-2 active" href="/overview.html">
                                    <i className="fa fa-fw fa-bar-chart mr-1"></i>
                                    <span>Overview</span>
                                </a>
                            </li>

                            <li className="nav-item">
                                <a className="nav-link px-2" href="#">
                                    <i className="fa fa-fw fa-cog mr-1"></i>
                                    <span>Settings</span>
                                </a>
                            </li>

```

```

    </ul>
  </div>
</div>
</div>

<div className="col">
  <div className="row">
    <div className="col mb-3">
      <div className="card">
        <div className="card-body">
          <div className="e-profile">
            <div className="row">
              <div className="col-12 col-sm-auto mb-3">
                <div className="mx-auto" style={{ width: "140px" }}>
                  <div
                    className="d-flex justify-content-center align-items-center rounded"
                    style={{
                      height: "140px",
                      backgroundColor: "rgb(233, 236, 239)",
                    }}
                  >
                    <img
                      style={{
                        height: "140px",
                        width: "140px",
                      }}
                      alt="user-avatar"
                      src={
                        imagePreview
                          ? imagePreview

```

```

        : JSON.parse(localStorage.getItem('user'))
        .avatar
    ? `http://localhost:5000${
        JSON.parse(localStorage.getItem('user'))
        .avatar
    }

    }
    />
</div>
</div>
</div>
<div className="col d-flex flex-column flex-sm-row justify-content-between mb-
<div className="text-center text-sm-left mb-2 mb-sm-0">
<h4 className="pt-sm-2 pb-1 mb-0 text-nowrap">
    {
        JSON.parse(localStorage.getItem('user'))
        .firstname
    }{" "}
    {
        JSON.parse(localStorage.getItem('user'))
        .lastname
    }
</h4>
<p className="mb-0">
    @
    {
        JSON.parse(localStorage.getItem('user'))
        .username
    }

```

```

</p>
<input
  id="imageFile"
  type="file"
  hidden
  onChange={(e) => {
    setImagePreview(
      URL.createObjectURL(e.target.files[0])
    );
    setFile(e.target.files[0]);
  }}
/>
<div className="mt-2">
  <button
    onClick={() =>
      document.getElementById("imageFile").click()
    }
    className="btn btn-primary"
    type="button"
  >
    <i className="fa fa-fw fa-camera"></i>
    <span>Change Photo</span>
  </button>
</div>
</div>
<div className="text-center text-sm-right">
  <span className="badge badge-secondary">
    {JSON.parse(localStorage.getItem("user")).role}
  </span>
  <div className="text-muted">

```

```

    <small>
      {JSON.parse(localStorage.getItem("user"))
        ?.createdAt
        ? new Date(
          JSON.parse(
            localStorage.getItem("user")
          ).createdAt
        ).toLocaleDateString()
        : "Joined 09 Dec 2017"}
    </small>
  </div>
</div>
</div>
</div>
<div>
  <ul className="nav nav-tabs">
    <li className="nav-item">Settings</li>
  </ul>
  <div className="tab-content pt-3">
    <div className="tab-pane active">
      <form onSubmit={handleUpdateMe} className="form">
        <div className="row">
          <div className="col">
            <div className="row">
              <div className="col">
                <div className="form-group">
                  <label>First Name</label>
                  <input
                    className="form-control"
                    type="text"
                    name="name"

```



```

        onChange={e =>
            setFirstname(e.target.value)
        }
        placeholder={
            JSON.parse(
                localStorage.getItem("user")
            ).firstname
        }
    />
</div>
</div>
<div className="col">
    <div className="form-group">
        <label>Last Name</label>
        <input
            className="form-control"
            onChange={e =>
                setLastname(e.target.value)
            }
            type="text"
            name="name"
            placeholder={
                JSON.parse(
                    localStorage.getItem("user")
                ).lastname
            }
        />
    </div>
</div>
<div className="col">

```

```

<div className="form-group">
  <label>Username</label>
  <input
    onChange={(e) =>
      setUsername(e.target.value)
    }
    className="form-control"
    type="text"
    name="username"
    placeholder={
      JSON.parse(
        localStorage.getItem("user")
      ).username
    }
  />
</div>
</div>
</div>
</div>
</div>
</div>
<div className="row">
  <div className="col d-flex justify-content-center mb-3">
    <button className="btn btn-primary mb-5">
      Save Changes
    </button>
  </div>
</div>
</form>
<form onSubmit={handleUpdadePassword}>
  <div className="row">

```

```

<div className="col-12 mb-3">
  <div className="mb-2">
    <b>Change Password</b>
  </div>
  <div className="row">
    <div className="col">
      <div className="form-group">
        <label>Current Password</label>
        <input
          className="form-control"
          onChange={e =>
            setCurrentPassword(e.target.value)
          }
          type="password"
          placeholder="•••••"
        />
      </div>
    </div>
  </div>
  <div className="row">
    <div className="col">
      <div className="form-group">
        <label>New Password</label>
        <input
          className="form-control"
          onChange={e =>
            setPassword(e.target.value)
          }
          type="password"
          placeholder="•••••"
        />
      </div>
    </div>
  </div>

```

```

    />
  </div>
</div>
</div>
<div className="row">
  <div className="col">
    <div className="form-group">
      <label>
        Confirm{" "}
        <span className="d-none d-xl-inline">
          Password
        </span>
      </label>
      <input
        className="form-control"
        type="password"
        placeholder="....."
        onChange={(e) =>
          setConfirmPassword(e.target.value)
        }
      />
    </div>
  </div>
</div>
</div>
<div className="row">
  <div className="col d-flex justify-content-center">
    <button className="btn btn-primary">
      Save Password

```

```

        </button>
      </div>
    </div>
  </form>
</div>
</div>
</div>
</div>
</div>
</div>
</div>

<div className="col-12 col-md-3 mb-3">
  <div className="card mb-3">
    <div className="card-body">
      <div className="px-xl-3">
        <button
          onClick={() => {
            localStorage.clear();
            history.push("/");
          }}
          className="btn btn-block btn-secondary"
        >
          <i className="fa fa-sign-out"></i>
          <span>Logout</span>
        </button>
      </div>
    </div>
  </div>
</div>
<div className="card">
  <div className="card-body">

```



```

    if (localStorage.getItem("admin_token")) {
      history.push("admindashboard");
    } else if (localStorage.getItem("student_token")) {
      history.push("studentdashboard");
    } else if (localStorage.getItem("teacher_token")) {
      history.push("teacherdashboard");
    }
  }
}
return () => {
  // on unmount do clean states
  setUsername("");
  setPassword("");
};
}, []);

const submitForm = async (e) => {
  e.preventDefault();
  if (!username || !password)
    return alert("username and password is required!");
  try {
    const { data } = await Axios.post("http://localhost:5000/users/login", {
      username,
      password,
    });
    if (data.success) {
      switch (data.data.role) {
        case "admin":
          localStorage.setItem("admin_token", JSON.stringify(data.token));
          localStorage.setItem("user", JSON.stringify(data.data));
          return history.push("/admindashboard");

```

```

    case "student":
        localStorage.setItem("student_token", JSON.stringify(data.token));
        localStorage.setItem("user", JSON.stringify(data.data));
        return history.push("/studentdashboard");
    case "teacher":
        localStorage.setItem("teacher_token", JSON.stringify(data.token));
        localStorage.setItem("user", JSON.stringify(data.data));
        return history.push("/teacherdashboard");
    default:
        return alert("Something went wrong, please try again.");
    }
}
} catch (error) {
    if (error?.response?.data?.msg) alert(error.response.data.msg);
    else alert(error.message);
}
};

```

```

return (
    <div>
        <div className="row">
            <div className="col-md-5 p-5">
                

                <h2 className="name animated bounceIn"> Login </h2>
                <br></br>
            </div>
        </div>
    </div>
)

```



```

<div className="card text-center">
  <div className="card-body ">
    <form
      onSubmit={handleSubmit}
      className="text-center"
      style={{ color: "#757575" }}
    >
      <div className="md-form mt-3">
        <input
          type="text"
          value={username}
          id="username"
          className="form-control"
          onChange={(e) => setUsername(e.target.value)}
        />
        <label htmlFor="username">Username</label>
      </div>

      <div className="md-form mt-3">
        <input
          type="password"
          value={password}
          id="password"
          className="form-control"
          onChange={(e) => setPassword(e.target.value)}
        />
        <label htmlFor="password">Password</label>
      </div>

      <button
        className="btn btn-info btn-rounded"

```

```

        style={{ fontWeight: "bold" }}
        // type="submit"
    >
        Sign in
    </button>
</form>
</div>
</div>
</div>

<div className="col-md-7">
    
</div>
</div>
</div>
);
}

```

Appendix-F: Dashboard_of_Admin

```

import React, { Component } from "react";
import Sidebar from "./Sidebar";
import Navbar from "./Navbar";
import Content from "./Content";
import Footer from "./Footer";

```

```

import { Link } from "react-router-dom";
import axios from "axios";

export default class Dashboard extends Component {
  constructor(props) {
    super(props);
    this.state = {
      classes: [],
      images: [
        "https://img.freepik.com/free-vector/students-classroom-flat-vector-illustration_74855-6663.jpg?size=626&ext=jpg",
        "https://image.shutterstock.com/z/stock-vector-vector-illustration-business-man-study-in-online-course-the-mentor-explain-in-online-class-computer-1496233118.jpg",
        "https://c8.alamy.com/comp/2C3KEYG/online-training-teacher-explain-a-student-in-computer-video-education-and-courses-learning-digital-vector-illustration-2C3KEYG.jpg",
        "https://cdn5.vectorstock.com/i/1000x1000/90/24/education-online-students-with-laptop-watching-vector-30289024.jpg",
        "https://cdn2.vectorstock.com/i/1000x1000/52/11/online-education-concept-distant-training-courses-vector-30565211.jpg",
      ],
    };
  }

  componentDidMount() {
    const user = localStorage.getItem("user");
    if (user && JSON.parse(user).role !== "admin") {
      this.props.history.goBack();
    }
    axios
      .get("/classrooms/")
      .then(({ data: { data } }) => {
        this.setState({ classes: data });
      })
  }
}

```

```

.catch((error) => {
  error.response.data.msg
    ? alert(error.response.data.msg)
    : alert(error.message);
});
}
render() {
  return (
    <div>
      <div class="wrapper">
        <Sidebar />
        <div class="main-panel">
          <Navbar name="Admin Dashboard" />

          <div class="content">
            <div class="container-fluid">
              <div class="row flex-wrap">
                {this.state.classes.map((currentclass) => {
                  return (
                    <div class="col-lg-4 col-md-4 col-sm-6">
                      <div
                        class="card card-stats"
                        style={{ height: "325px", width: "325px" }}
                      >
                        <div class="card-header card-header-success card-header-icon">
                          <div class="card-icon" style={{ width: "100px" }}>
                            <i class="material-icons">
                              <h3>{currentclass.section}</h3>
                            </i>
                          </div>
                        </div>

```

```

    <p class="card-category">
        {currentclass.coursecode}
    </p>

    <h3 class="card-title">{currentclass.classname}</h3>
</div>

<div class="card-footer">
    <div class="stats">
        <i class="material-icons">date_range</i>
        
    </div>
</div>
</div>
</div>
);
}}
</div>
</div>

<div class="row">
    <div class="row">
        <div class="col-lg-6 col-md-12">
            <div class="card">
                <div class="card-header card-header-tabs card-header-primary">
                    <div class="nav-tabs-navigation">

```

```

<div class="nav-tabs-wrapper">
  <span class="nav-tabs-title">Tasks:</span>
  <ul class="nav nav-tabs" data-tabs="tabs">
    <li class="nav-item">
      <a
        class="nav-link active"
        href="#profile"
        data-toggle="tab"
      >
        <i class="material-icons">bug_report</i> Bugs
        <div class="ripple-container"></div>
      </a>
    </li>
    <li class="nav-item">
      <a
        class="nav-link"
        href="#messages"
        data-toggle="tab"
      >
        <i class="material-icons">code</i> Website
        <div class="ripple-container"></div>
      </a>
    </li>
    <li class="nav-item">
      <a
        class="nav-link"
        href="#settings"
        data-toggle="tab"
      >
        <i class="material-icons">cloud</i> Server

```

```

        <div class="ripple-container"></div>
    </a>
</li>
</ul>
</div>
</div>
</div>
<div class="card-body">
    <div class="tab-content">
        <div class="tab-pane active" id="profile">
            <table class="table">
                <tbody>
                    <tr>
                        <td>
                            <div class="form-check">
                                <label class="form-check-label">
                                    <input
                                        class="form-check-input"
                                        type="checkbox"
                                        value=""
                                        checked
                                    />
                                    <span class="form-check-sign">
                                        <span class="check"></span>
                                    </span>
                                </label>
                            </div>
                        </td>
                        <td>
                            Sign contract for "What are conference

```

```

        organizers afraid of?"
    </td>
    <td class="td-actions text-right">
        <button
            type="button"
            rel="tooltip"
            title="Edit Task"
            class="btn btn-primary btn-link btn-sm"
        >
            <i class="material-icons">edit</i>
        </button>
        <button
            type="button"
            rel="tooltip"
            title="Remove"
            class="btn btn-danger btn-link btn-sm"
        >
            <i class="material-icons">close</i>
        </button>
    </td>
</tr>
<tr>
<td>
    <div class="form-check">
        <label class="form-check-label">
            <input
                class="form-check-input"
                type="checkbox"
                value=""
            />

```



```

        <span class="form-check-sign">
          <span class="check"></span>
        </span>
      </label>
    </div>
  </td>
  <td>
    Lines From Great Russian Literature? Or
    E-mails From My Boss?
  </td>
  <td class="td-actions text-right">
    <button
      type="button"
      rel="tooltip"
      title="Edit Task"
      class="btn btn-primary btn-link btn-sm"
    >
      <i class="material-icons">edit</i>
    </button>
    <button
      type="button"
      rel="tooltip"
      title="Remove"
      class="btn btn-danger btn-link btn-sm"
    >
      <i class="material-icons">close</i>
    </button>
  </td>
</tr>
<tr>

```

```

<td>

<div class="form-check">

  <label class="form-check-label">

    <input

      class="form-check-input"

      type="checkbox"

      value=""

    />

    <span class="form-check-sign">

      <span class="check"></span>

    </span>

  </label>

</div>

</td>

<td>

  Flooded: One year later, assessing what was

  lost and what was found when a ravaging rain

  swept through metro Detroit

</td>

<td class="td-actions text-right">

  <button

    type="button"

    rel="tooltip"

    title="Edit Task"

    class="btn btn-primary btn-link btn-sm"

  >

    <i class="material-icons">edit</i>

  </button>

  <button

    type="button"

```

```

        rel="tooltip"
        title="Remove"
        class="btn btn-danger btn-link btn-sm"
    >
        <i class="material-icons">close</i>
    </button>
</td>
</tr>
<tr>
<td>
    <div class="form-check">
        <label class="form-check-label">
            <input
                class="form-check-input"
                type="checkbox"
                value=""
                checked
            />
            <span class="form-check-sign">
                <span class="check"></span>
            </span>
        </label>
    </div>
</td>
<td>
    Create 4 Invisible User Experiences you
    Never Knew About
</td>
<td class="td-actions text-right">
    <button

```

```

        type="button"
        rel="tooltip"
        title="Edit Task"
        class="btn btn-primary btn-link btn-sm"
    >
        <i class="material-icons">edit</i>
    </button>
    <button
        type="button"
        rel="tooltip"
        title="Remove"
        class="btn btn-danger btn-link btn-sm"
    >
        <i class="material-icons">close</i>
    </button>
</td>
</tr>
</tbody>
</table>
</div>
<div class="tab-pane" id="messages">
    <table class="table">
        <tbody>
            <tr>
                <td>
                    <div class="form-check">
                        <label class="form-check-label">
                            <input
                                class="form-check-input"
                                type="checkbox"

```

```

        value=""
        checked
    />
    <span class="form-check-sign">
        <span class="check"></span>
    </span>
</label>
</div>
</td>
<td>
    Flooded: One year later, assessing what was
lost and what was found when a ravaging rain
swept through metro Detroit
</td>
<td class="td-actions text-right">
    <button
        type="button"
        rel="tooltip"
        title="Edit Task"
        class="btn btn-primary btn-link btn-sm"
    >
        <i class="material-icons">edit</i>
    </button>
    <button
        type="button"
        rel="tooltip"
        title="Remove"
        class="btn btn-danger btn-link btn-sm"
    >
        <i class="material-icons">close</i>

```

```

        </button>
    </td>
</tr>
<tr>
<td>
    <div class="form-check">
        <label class="form-check-label">
            <input
                class="form-check-input"
                type="checkbox"
                value=""
            />
            <span class="form-check-sign">
                <span class="check"></span>
            </span>
        </label>
    </div>
</td>
<td>
    Sign contract for "What are conference
    organizers afraid of?"
</td>
<td class="td-actions text-right">
    <button
        type="button"
        rel="tooltip"
        title="Edit Task"
        class="btn btn-primary btn-link btn-sm"
    >
        <i class="material-icons">edit</i>

```

```

        </button>
        <button
            type="button"
            rel="tooltip"
            title="Remove"
            class="btn btn-danger btn-link btn-sm"
        >
            <i class="material-icons">close</i>
        </button>
    </td>
</tr>
</tbody>
</table>
</div>
<div class="tab-pane" id="settings">
    <table class="table">
        <tbody>
            <tr>
                <td>
                    <div class="form-check">
                        <label class="form-check-label">
                            <input
                                class="form-check-input"
                                type="checkbox"
                                value=""
                            />
                            <span class="form-check-sign">
                                <span class="check"></span>
                            </span>
                        </label>

```

```

    </div>
</td>
<td>
    Lines From Great Russian Literature? Or
    E-mails From My Boss?
</td>
<td class="td-actions text-right">
    <button
        type="button"
        rel="tooltip"
        title="Edit Task"
        class="btn btn-primary btn-link btn-sm"
    >
        <i class="material-icons">edit</i>
    </button>
    <button
        type="button"
        rel="tooltip"
        title="Remove"
        class="btn btn-danger btn-link btn-sm"
    >
        <i class="material-icons">close</i>
    </button>
</td>
</tr>
<tr>
<td>
    <div class="form-check">
        <label class="form-check-label">
            <input

```



```

        class="form-check-input"
        type="checkbox"
        value=""
        checked
    />
    <span class="form-check-sign">
        <span class="check"></span>
    </span>
</label>
</div>
</td>
<td>
    Flooded: One year later, assessing what was
lost and what was found when a ravaging rain
swept through metro Detroit
</td>
<td class="td-actions text-right">
    <button
        type="button"
        rel="tooltip"
        title="Edit Task"
        class="btn btn-primary btn-link btn-sm"
    >
        <i class="material-icons">edit</i>
    </button>
    <button
        type="button"
        rel="tooltip"
        title="Remove"
        class="btn btn-danger btn-link btn-sm"

```

```

>
    <i class="material-icons">close</i>

</button>

</td>

</tr>

<tr>

<td>

    <div class="form-check">

        <label class="form-check-label">

            <input

                class="form-check-input"

                type="checkbox"

                value=""

                checked

            />

            <span class="form-check-sign">

                <span class="check"></span>

            </span>

        </label>

    </div>

</td>

<td>

    Sign contract for "What are conference
    organizers afraid of?"

</td>

<td class="td-actions text-right">

    <button

        type="button"

        rel="tooltip"

        title="Edit Task"

```

```

        class="btn btn-primary btn-link btn-sm"
    >
        <i class="material-icons">edit</i>
    </button>
    <button
        type="button"
        rel="tooltip"
        title="Remove"
        class="btn btn-danger btn-link btn-sm"
    >
        <i class="material-icons">close</i>
    </button>
</td>
</tr>
</tbody>
</table>
</div>
</div>
</div>
</div>
</div>
<div class="col-lg-6 col-md-12">
    <div class="card">
        <div class="card-header card-header-warning">
            <h4 class="card-title">Employees Stats</h4>
            <p class="card-category">
                New employees on 15th September, 2016
            </p>
        </div>
        <div class="card-body table-responsive">

```

```

<table class="table table-hover">
  <thead class="text-warning">
    <th>ID</th>
    <th>Name</th>
    <th>Salary</th>
    <th>Country</th>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Dakota Rice</td>
      <td>$36,738</td>
      <td>Niger</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Minerva Hooper</td>
      <td>$23,789</td>
      <td>Curaçao</td>
    </tr>
    <tr>
      <td>3</td>
      <td>Sage Rodriguez</td>
      <td>$56,142</td>
      <td>Netherlands</td>
    </tr>
    <tr>
      <td>4</td>
      <td>Philip Chaney</td>
      <td>$38,735</td>

```



```

    <span
      class="badge filter badge-azure"
      data-color="azure"
    ></span>

    <span
      class="badge filter badge-green"
      data-color="green"
    ></span>

    <span
      class="badge filter badge-warning"
      data-color="orange"
    ></span>

    <span
      class="badge filter badge-danger"
      data-color="danger"
    ></span>

    <span
      class="badge filter badge-rose active"
      data-color="rose"
    ></span>
  </div>

  <div class="clearfix"></div>

</a>

</li>

<li class="header-title">Images</li>

<li class="active">
  <a class="img-holder switch-trigger" href="javascript:void(0)">
    
  </a>
</li>

```

```

<li>
  <a class="img-holder switch-trigger" href="javascript:void(0)">
    
  </a>

```

```

</li>

```

```

<li>
  <a class="img-holder switch-trigger" href="javascript:void(0)">
    
  </a>

```

```

</li>

```

```

<li>
  <a class="img-holder switch-trigger" href="javascript:void(0)">
    
  </a>

```

```

</li>

```

```

<li class="button-container">
  <a
    href="https://www.creative-tim.com/product/material-dashboard"
    target="_blank"
    class="btn btn-primary btn-block"
  >

```

```

    Free Download

```

```

  </a>

```

```

</li>

```

```

<li class="button-container">

```

```

  <a
    href="https://demos.creative-tim.com/material-dashboard/docs/2.1/getting-
started/introduction.html"
    target="_blank"
    class="btn btn-default btn-block"

```

```

>
  View Documentation
</a>
</li>
<li class="button-container github-star">
  <a
    class="github-button"
    href="https://github.com/creativetimofficial/material-dashboard"
    data-icon="octicon-star"
    data-size="large"
    data-show-count="true"
    aria-label="Star ntkme/github-buttons on GitHub"
  >
    Star
  </a>
</li>
<li class="header-title">Thank you for 95 shares!</li>
<li class="button-container text-center">
  <button id="twitter" class="btn btn-round btn-twitter">
    <i class="fa fa-twitter"></i> &middot; 45
  </button>
  <button id="facebook" class="btn btn-round btn-facebook">
    <i class="fa fa-facebook-f"></i> &middot; 50
  </button>
  <br></br>
  <br></br>
</li>
</ul>
</div>
</div>

```



```

    </div>
  );
}
}

```

Appendix-G: Admin_Menu:

```

import React from 'react'
import { Link } from 'react-router-dom';

export default function Menu() {
  return (
    <div>

      <div class="container-fluid">
        <div class="row m-5">
          <div className="col-md-2"></div>
          <div className="col-md-5 mt-5">
            <h3 style={{fontWeight:'bold', marginTop:'20px'}}>Welcome </h3>
            
          </div>
          <div className="col-md-3 mt-5" style={{marginTop:'50px;'}}>

            <Link class="nav-link" to="/loginadmin">
              <button style={{fontWeight:'bold'}} type="button" class="btn btn-lg btn-
primary" >Login as Admin</button>
            </Link>

            <Link class="nav-link" to="/loginteacher">
              <button style={{fontWeight:'bold'}} type="button" class="btn btn-lg btn-
primary" >Login as Teacher</button>
            </Link>

```

```

        <Link class="nav-link" to="/loginstudent">
            <button style={{fontWeight:'bold'}}type="button" class="btn btn-lg btn-
primary" >Login as Student</button>
        </Link>

    </div>

    <div className="col-md-2"></div>
</div>

</div>
</div>

)

}

```

Appendix-H: Admin_Sidebar

```

import React from "react";
import { Link } from "react-router-dom";
import { Button, Input } from "react-bootstrap";
import Modal from "react-bootstrap/Modal";
import ModalDialog from "react-bootstrap/ModalDialog";
import ModalHeader from "react-bootstrap/ModalHeader";
import ModalBody from "react-bootstrap/ModalBody";
import ModalFooter from "react-bootstrap/ModalFooter";
import ModalofAddClass from "../Components/ModalofAddClass";

export default function Sidebar() {
    const [modalShow, setModalShow] = React.useState(false);

```



```

</li>
<li className="nav-item">
  <Link className="nav-link" to="/assign-teacher-to-class">
    <i className="material-icons">people</i>
    <p>Assign Teacher to Class</p>
  </Link>
</li>
<li className="nav-item">
  <Link className="nav-link" to="/add-student-to-class">
    <i className="material-icons">people</i>
    <p>Add Student to Class</p>
  </Link>
</li>
<li className="nav-item">
  <Link className="nav-link" to="/manage-student-by-admin">
    <i className="material-icons">people</i>
    <p>Manage Student & Class</p>
  </Link>
</li>
<li className="nav-item">
  <Link className="nav-link" to="/add-student-by-admin">
    <i className="material-icons">people</i>
    <p>Add Student </p>
  </Link>
</li>
<li className="nav-item">
  <Link className="nav-link" to="/add-teacher-by-admin">
    <i className="material-icons">people</i>
    <p>Add Teacher</p>
  </Link>

```


<li className="nav-item">

<Link className="nav-link" to="#">

<button

className="nav-link btn btn-primary"

onClick={() => setModalShow(true)}

>

<i

style={{ margin: "auto", paddingRight: "7px" }}

className="material-icons"

>

person_add

</i>

<p style={{ marginRight: "17px" }}>Add ClassName</p>

</button>

</Link>

<ModalofAddClass

show={modalShow}

onHide={() => setModalShow(false)}

/>

</div>

</div>

</div>

);

}

Appendix-I: Student_Login

```
import React , {useState} from 'react';
import { Link } from 'react-router-dom';
```

```
export default function StudentLogin() {
```

```
  return (
    <div>
      <div className="row">
        <div className="col-md-5 p-5">
          

          <h2 style={{fontWeight:'bold', color:'#DA70D6', marginTop:'10px'}}> Student Login </h2>
          <br></br>
          <div class="card text-center">
            <div class="card-body ">
              <form class="text-center" style={{color: '#757575'}} method="POST">
                <div class="md-form mt-3">
                  <div class="form-group">
                    <label class="bmd-label-floating">Email</label>
                    <input type="text" class="form-control" />
                  </div>
                </div>

                <div class="md-form mt-3">
                  <div class="form-group">
                    <label class="bmd-label-floating" >Password</label>
                    <input type="text" class="form-control" />
                  </div>
                </div>
                <Link to="/studentdashboard" >
                  <button type="submit" class="mt-3 btn btn-primary pull-center">Sign In </button>
                </Link>
              </form>

            </div>
          </div>
        </div>
      </div>
    </div>
    <div className="col-md-7 col-sm-2">
      
    </div>
  )
}
```

```

</div>
</div>

    </div>

    )
}

```

Appendix-J: Student_Dashboard

```

import React, { useState } from "react";
import StudentSidebar from "../StudentSidebar";
import StudentNavbar from "../StudentNavbar";
import StudentFooter from "../StudentFooter";
import { useHistory } from "react-router";
import axios from "axios";

export default function StudentDashboard() {
  const [classes, setClasses] = useState([]);

  const history = useHistory();

  React.useEffect(() => {
    const user = localStorage.getItem("user");
    if (user && JSON.parse(user).role !== "student") {
      history.goBack();
    }

    axios
      .get("/students/" + JSON.parse(user).id)
      .then(({ data: { data } }) => {
        setClasses(data.classes);
      })
      .catch((error) =>

```

```

    error.response.data.msg
    ? alert(error.response.data.msg)
    : alert(error.message)
  );
}, []);
return (
  <div>
    <div className="wrapper ">
      <StudentSidebar />
      <div className="main-panel">
        <StudentNavbar name="Student Dashboard" />
        <div className="content">
          <div className="container-fluid">
            {classes.length ? (
              <div className="row flex-wrap">
                {classes.map((currentclass) => {
                  return (
                    <div
                      key={currentclass._id}
                      style={{ cursor: "pointer" }}
                      onClick={() =>
                        history.push(
                          "classdetailsforstudents/" + currentclass._id
                        )
                      }
                      className="col-lg-4 col-md-4 col-sm-6"
                    >
                      <div
                        className="card card-stats"
                        style={{ height: "325px", width: "325px" }}

```



```

>
<div className="card-header card-header-success card-header-icon">
  <div
    className="card-icon"
    style={{ width: "100px" }}
  >
    <i className="material-icons">
      <h3>{currentclass.section}</h3>
    </i>
  </div>
  <p className="card-category">
    {currentclass.coursecode}
  </p>
  <h3 className="card-title">
    {currentclass.classname}
  </h3>
</div>

<div className="card-footer">
  <div className="stats">
    <i className="material-icons">date_range</i>
    
  </div>
</div>
</div>

```

```

        </div>

    );

    }}

    { /* <ManageStudentModal

    show={modalShow}

    classRoom={classRoom}

    onHide={() => setModalShow(false)}

    /> */}

    </div>

    ) : (

        "You are not enrolled in any Course."

    )}

    </div>

</div>

{ /* <StudentFooter /> */}

</div>

</div>

</div>

);

}

```

Appendix-K: Student_Attendance

```

import axios from "axios";
import { useEffect, useState } from "react";
import uniqBy from "lodash.uniqby";

const Attendance = (props) => {
  const [attendanceList, setAttendanceList] = useState(null);
  const [loading, setLoading] = useState(false);
  async function fetchAttendance() {
    try {
      setLoading(true);
      const {
        data: { data },
      } = await axios.get(
        `http://localhost:5000/attendance/${props.course._id}/${props}`
      );
    } catch (error) {
      console.log(error);
    }
  }
  useEffect(() => {
    fetchAttendance();
  }, [props.course._id]);

  return (
    <div>
      <h3>Student Attendance</h3>
      <table>
        <thead>
          <tr>
            <th>Name</th>
            <th>Attendance</th>
          </tr>
        </thead>
        <tbody>
          {attendanceList?.map((student) => (
            <tr>
              <td>{student.name}</td>
              <td>{student.attendance}</td>
            </tr>
          ))}
        </tbody>
      </table>
    </div>
  );
};

```

```

    JSON.parse(localStorage.getItem("user")).id
  }
);
let attendanceList = [];
if (data) {
  data.attendanceRecord.forEach((a) => {
    const d = new Date(a.date).toLocaleDateString();
    attendanceList.push({
      ...a,
      date: d,
    });
  });
}
attendanceList = attendanceList.reverse();
attendanceList = uniqBy(attendanceList, "date");
attendanceList = attendanceList.reverse();
setAttendanceList({ ...data, attendanceRecord: attendanceList });
setLoading(false);
} catch (error) {
  if (error?.response?.data?.msg) alert(error.response.data.msg);
  else alert(error.message);
  setLoading(false);
}
}
useEffect(() => {
  if (props.course) {
    fetchAttendance();
  }
}, []);
return (
  <div className="content">
    <div className="container-fluid">
      <div className="row">
        <div className="col-md-12 col-lg-12 col-sm-12">
          <div className="card">
            <div className="card-header card-header-primary">
              <h4 className="card-title ">Attendance List </h4>
            </div>

            {!loading && attendanceList ? (
              <>
                <p className="card-category">
                  Here is a list of all Your Attendance.
                </p>
                <div className="card-body">
                  <div className="table-responsive">
                    <table className="table">
                      <thead className=" text-primary">

```

```

    <tr>
      <th>#</th>
      <th>Date</th>
      <th>Presence</th>
    </tr>
  </thead>

  <tbody>
    {attendanceList.attendanceRecord
      // .sort(
      //   (a, b) =>
      //     new Date(b.createdAt) - new Date(a.createdAt)
      // )
      .map((att, i) => (
        <tr key={i}>
          <td>{i + 1}</td>
          <td>{new Date(att.date).toDateString()}</td>
          <td>
            {att.attendanceStatus === "present" ? (
              <h6 style={{ color: "green" }}>P</h6>
            ) : (
              <h6 style={{ color: "red" }}>A</h6>
            )}
          </td>
        </tr>
      ))}
    </tbody>
  </table>
</div>
</div>
</>
) : (
  "Attendance Sheet is updated yet."
)
</div>
</div>
</div>
</div>
</div>
);
};

```

```
export default Attendance;
```

Appendix-L: Student_Sidebar

```

import React from "react";
import { Link } from "react-router-dom";

```

```

export default function StudentSidebar() {
  return (
    <div>
      <div
        class="sidebar"
        data-color="purple"
        data-background-color="white"
        data-image="../assets/img/sidebar-1.jpg"
      >
        <div class="logo">
          <Link
            to="#"
            class="simple-text logo-normal"
            style={{ fontWeight: "bold" }}
          >
            Edu Tech
          </Link>
        </div>
        <div class="sidebar-wrapper">
          <ul class="nav">
            <li class="nav-item active ">
              <Link class="nav-link" to="/studentdashboard">
                <i class="material-icons">dashboard</i>
                <p>Dashboard</p>
              </Link>
            </li>

            <li class="nav-item ">
              <Link class="nav-link" to="/studentdashboard">
                <i class="material-icons">upload</i>
                <p>Upload Assignment</p>
              </Link>
            </li>

            <li class="nav-item ">
              <Link class="nav-link" to="/studentdashboard">
                <i class="material-icons">upload</i>
                <p>Upload Quiz</p>
              </Link>
            </li>

            <li class="nav-item ">
              <Link class="nav-link" to="/studentdashboard">
                <i class="material-icons">upload</i>
                <p>Upload Exam</p>
              </Link>
            </li>
          </ul>
        </div>
      </div>
    </div>
  )
}

```

```

    </div>
  </div>
);
}

```

Appendix-M: Attemp_Assignment

```

import React from 'react'
import StudentSidebar from './StudentSidebar';
import StudentNavbar from './StudentNavbar';
import StudentFooter from './StudentFooter';
import { Link } from 'react-router-dom';

export default function AttemptAssignmentByStu() {
  return (
    <div>
      <div class="wrapper ">
        <StudentSidebar />
        <div class="main-panel">
          <StudentNavbar name="Upload Assignment"/>

          <div class="content">
            <div class="container-fluid">
              <div class="row">
                <div class="col-md-8">
                  <div class="card">
                    <div class="card-header card-header-primary">
                      <h4 class="card-title">Add File</h4>
                      <p class="card-category">Complete your Data</p>
                    </div>
                    <div class="card-body">

```

```

    <form>
      <div class="row">
        <div class="col-md-12">

          <div class="form-group form-file-upload form-file-multiple">
            <input type="file" multiple="" class="inputFileHidden" />
            <div class="input-group">
              <input type="text" class="form-control inputFileVisible" placeholder="Single File" />
              <span class="input-group-btn">
                <button type="button" class="btn btn-fab btn-round btn-primary">
                  <i class="material-icons">attach_file</i>
                </button>
              </span>
            </div>
          </div>
        </div>
      </div>
    </div>

```

```

    </div>
  </div>
  <Link to="/studentdashboard">
    <button type="submit" class="btn btn-primary pull-right">Save
Change</button>
  </Link>
  <div class="clearfix"></div>
</form>
</div>
</div>
</div>
</div>
</div>

```

</div>

<StudentFooter />

</div>

</div>

<div class="fixed-plugin">

<div class="dropdown show-dropdown">

<i class="fa fa-cog fa-2x"> </i>

<ul class="dropdown-menu">

<li class="header-title"> Sidebar Filters

<li class="adjustments-line">

<div class="badge-colors ml-auto mr-auto">

</div>

<div class="clearfix"></div>

<li class="header-title">Images

<li class="active">

```

        
    </a>
</li>
<li>
    <a class="img-holder switch-trigger" href="javascript:void(0)">
        
    </a>
</li>
<li>
    <a class="img-holder switch-trigger" href="javascript:void(0)">
        
    </a>
</li>
<li>
    <a class="img-holder switch-trigger" href="javascript:void(0)">
        
    </a>
</li>
<li class="button-container">
    <a href="https://www.creative-tim.com/product/material-dashboard" target="_blank"
class="btn btn-primary btn-block">Free Download</a>
</li>

<li class="button-container">
    <a href="https://demos.creative-tim.com/material-dashboard/docs/2.1/getting-
started/introduction.html" target="_blank" class="btn btn-default btn-block">
        View Documentation
    </a>
</li>
<li class="button-container github-star">

```

```

    <a class="github-button" href="https://github.com/creativetimoofficial/material-
dashboard" data-icon="octicon-star" data-size="large" data-show-count="true" aria-
label="Star ntkme/github-buttons on GitHub">Star</a>

  </li>

  <li class="header-title">Thank you for 95 shares!</li>

  <li class="button-container text-center">

    <button id="twitter" class="btn btn-round btn-twitter"><i class="fa fa-twitter"></i>
&middot; 45</button>

    <button id="facebook" class="btn btn-round btn-facebook"><i class="fa fa-facebook-
f"></i> &middot; 50</button>

    <br></br>

    <br></br>

  </li>

</ul>

</div>

</div>

)
}

```

Appendix-N: Teacher_Login

```

import React , {useState} from 'react';
import { Link } from 'react-router-dom';

```

```

export default function TeacherLogin() {

```

```

  return (
    <div>
      <div className="row">
        <div className="col-md-5 p-5">
          

```

```

    <h2 style={{fontWeight:'bold', color:'#DA70D6', marginTop:'10px'}}> Teacher Login
  </h2>
  <br></br>
  <div class="card text-center">
    <div class="card-body ">
      <form class="text-center" style={{color: '#757575'}} method="POST">
        <div class="md-form mt-3">
          <div class="form-group">
            <label class="bmd-label-floating">Email</label>
            <input type="text" class="form-control" />
          </div>
        </div>

        <div class="md-form mt-3">
          <div class="form-group">
            <label class="bmd-label-floating" >Password</label>
            <input type="text" class="form-control" />
          </div>
        </div>
        <Link to="/teacherdashboard" >
          <button type="submit" class="mt-3 btn btn-primary pull-center">Sign In </button>
        </Link>
      </form>
    </div>
  </div>
</div>
<div>
  <div class=Name="col-md-7 col-sm-2">
    
  </div>
</div>

  </div>

)
}
import React , {useState} from 'react';
import { Link } from 'react-router-dom';

export default function TeacherLogin() {

```

```

    return (
    <div>
    <div className="row">
    <div className="col-md-5 p-5">
    

    <h2 style={{fontWeight:'bold', color:'#DA70D6', marginTop:'10px'}}> Teacher Login
    </h2>
    <br></br>
    <div class="card text-center">
    <div class="card-body ">
    <form class="text-center" style={{color: '#757575'}} method="POST">
    <div class="md-form mt-3">
    <div class="form-group">
    <label class="bmd-label-floating">Email</label>
    <input type="text" class="form-control" />
    </div>
    </div>

    <div class="md-form mt-3">
    <div class="form-group">
    <label class="bmd-label-floating" >Password</label>
    <input type="text" class="form-control" />
    </div>
    </div>
    <Link to="/teacherdashboard" >
    <button type="submit" class="mt-3 btn btn-primary pull-center">Sign In </button>
    </Link>
    </form>

    </div>

    </div>

    </div>

    <div className="col-md-7 col-sm-2">
    
    </div>
    </div>

    </div>

    )
}

```

Appendix-O:Teacher_Dashboard

```
import React, { useState } from "react";  
import TeacherSidebar from "../TeacherSidebar";  
import TeacherNavbar from "../TeacherNavbar";  
import TeacherFooter from "../TeacherFooter";  
import ClassDetails from "../ClassDetails";  
import { Link, useHistory } from "react-router-dom";  
import axios from "axios";
```

```
export default function TeacherDashboard() {  
  const [classes, setClasses] = useState([]);  
  const history = useHistory();  
  React.useEffect(() => {  
    const user = localStorage.getItem("user");  
    if (user && JSON.parse(user).role !== "teacher") {  
      history.goBack();  
    }  
    axios  
      .get("/teachers/" + JSON.parse(user).id)  
      .then(({ data: { data } }) => {  
        setClasses(data.classes);  
      })  
  })  
}
```

```

    })

    .catch((error) =>

        error.response.data.msg

        ? alert(error.response.data.msg)

        : alert(error.message)

    );

}, []);

return (

    <div>

        <div class="wrapper ">

            <TeacherSidebar />

            <div class="main-panel">

                <TeacherNavbar />


                <div class="content">

                    <div class="container-fluid">

                        {classes.length ? (

                            <div class="row">

                                {classes.map((currentclass) => (

                                    <div

                                        key={currentclass._id}

                                        class="col-lg-4 col-md-4 col-sm-6"

                                    >

                                        <Link to={"/classdetails/" + currentclass._id}>

                                            <div

                                                class="card card-stats"

                                                style={{ height: "325px", width: "325px" }}

```

```

>
<div class="card-header card-header-success card-header-icon">

  <div class="card-icon" style={{ width: "100px" }}>

    <i class="material-icons">

      <h3>{currentclass.section}</h3>

    </i>

  </div>

  <p class="card-category">

    {currentclass.coursecode}

  </p>

  <h3 class="card-title">{currentclass.classname}</h3>

</div>


<div class="card-footer">

  <div class="stats">

    <i class="material-icons">date_range</i>

  </div>

</div>

</div>

```

```

        </Link>

    </div>

    )})

</div>

) : (

    "You are not assigned to any Class yet."

    })

</div>

</div>

<TeacherFooter />

</div>

</div>

<div class="fixed-plugin">

    <div class="dropdown show-dropdown">

        <a href="#" data-toggle="dropdown">

            <i class="fa fa-cog fa-2x"> </i>

        </a>

        <ul class="dropdown-menu">

            <li class="header-title"> Sidebar Filters</li>

            <li class="adjustments-line">

                <a href="javascript:void(0)" class="switch-trigger active-color">

                    <div class="badge-colors ml-auto mr-auto">

                        <span

                            class="badge filter badge-purple"

                            data-color="purple"

                        ></span>

                        <span

```



```

        class="badge filter badge-azure"

        data-color="azure"

    ></span>

    <span

        class="badge filter badge-green"

        data-color="green"

    ></span>

    <span

        class="badge filter badge-warning"

        data-color="orange"

    ></span>

    <span

        class="badge filter badge-danger"

        data-color="danger"

    ></span>

    <span

        class="badge filter badge-rose active"

        data-color="rose"

    ></span>

</div>

<div class="clearfix"></div>

</a>

</li>

<li class="header-title">Images</li>

<li class="active">

    <a class="img-holder switch-trigger" href="javascript:void(0)">

```

```

</a>
</li>
<li>
  <a class="img-holder switch-trigger" href="javascript:void(0)">
    
  </a>
</li>
<li>
  <a class="img-holder switch-trigger" href="javascript:void(0)">
    
  </a>
</li>
<li>
  <a class="img-holder switch-trigger" href="javascript:void(0)">
    
  </a>
</li>
<li class="button-container">
  <a
    href="https://www.creative-tim.com/product/material-dashboard"
    target="_blank"
    class="btn btn-primary btn-block"
  >
    Free Download
  </a>
</li>

```

```

<li class="button-container">

  <a

    href="https://demos.creative-tim.com/material-dashboard/docs/2.1/getting-
started/introduction.html"

    target="_blank"

    class="btn btn-default btn-block"

  >

    View Documentation

  </a>

</li>

<li class="button-container github-star">

  <a

    class="github-button"

    href="https://github.com/creativetimofficial/material-dashboard"

    data-icon="octicon-star"

    data-size="large"

    data-show-count="true"

    aria-label="Star ntkme/github-buttons on GitHub"

  >

    Star

  </a>

</li>

<li class="header-title">Thank you for 95 shares!</li>

<li class="button-container text-center">

  <button id="twitter" class="btn btn-round btn-twitter">

    <i class="fa fa-twitter"></i> &middot; 45

  </button>

```

```

    <button id="facebook" class="btn btn-round btn-facebook">
      <i class="fa fa-facebook-f"></i> &middot; 50
    </button>
  <br></br>
  <br></br>
</li>
</ul>
</div>
</div>
</div>
);
}

```

Backend Code:

Appendix-P:Middleware:

User.material:

```

const Course = require("models/classroom.model");
exports.checkCourse = async (req, res, next) => {
  try {
    if (!(await Course.findById(req.params.courseId))) {
      return res
        .status(404)
        .send({ success: false, msg: "course does not exists" });
    }
    next();
  } catch (error) {

```

```

    return res.status(404).send({ success: false, msg: error.message });
  }
};

exports.checkFile = (req, res, next) => {
  if (!req.file)
    return res.status(404).send({
      success: false,
      msg: "File file is required.",
    });

  // check if user hasn't provided a photo then go to next middleware of the stack
  if (!req.file.detectedFileExtension.match(/\.(pdf|pptx|docx)$/)) {
    return res.status(404).send({
      success: false,
      msg: "Please upload only pdf, ppt and word files.",
    }); // 415 is http status code for unsupported media type
  }

  req.dir = req.file.detectedFileExtension.substr(
    1,
    req.file.detectedFileExtension.length
  );

  next();
};

```

Appendix-Q:Models

Account:

```

const mongoose = require("mongoose");
const bcrypt = require("bcryptjs");

```

```

const jwt = require('jsonwebtoken');
const Schema = mongoose.Schema;

const schema = new Schema({
  firstname: { type: String, required: firstNameAndLastNameValidation },
  lastname: { type: String, required: firstNameAndLastNameValidation },
  rollno: { type: String, required: rollNoValidation },
  qualification: { type: String, required: qualificationValidation },
  department: { type: String, required: departmentValidation },
  username: {
    type: String,
    unique: true,
    trim: true,
    lowercase: true,
    required: true,
  },
  avatar: String,
  password: { type: String, required: true },
  role: { type: String, required: true, enum: ['admin', 'student', 'teacher'] },
  createdAt: { type: Date, default: Date.now },
});

schema.set('toJSON', {
  virtuals: true,
  versionKey: false,
  transform: function (doc, ret) {
    delete ret._id;
    delete ret.password;
  },
});

```

```

function firstNameAndLastNameValidation() {
return this.role === "admin" ? false : true;
}

function rollNoValidation() {
return this.role === "student" ? true : false;
}

function qualificationValidation() {
return this.role === "teacher" ? true : false;
}

function departmentValidation() {
return this.role === "teacher" ? true : false;
}

schema.pre("save", async function (next) {
console.log(this)
const user = this;
if (!user.isModified("password")) return next();
user.password = await bcrypt.hash(user.password, 12);
next();
});

schema.methods.confirmPassword = async function (
candidatePassword,
savePassword )
{
return await bcrypt.compare(candidatePassword, savePassword);
};

generating auth token
schema.methods.generateAuthToken = async function () {
const user = this;
const token = jwt.sign(

```

```

{
  id: user._id + "",
},
process.env.JWT_SECRET,
{
  expiresIn: "10 days",
}
);
return token;
};

```

```

schema.statics.findByCredentials = async (username, password) => {
  const user = await AccountModel.findOne({
    username,
  });
  if (!user) {
    throw new Error("Invalid username or password");
  }
  const isMatch = await bcrypt.compare(password, user.password);
  if (!isMatch) {
    throw new Error("Invalid username or password");
  }
  user.password = undefined;
  return user;
};

```

//-----

```

const AccountModel = mongoose.model("Account", schema);
module.exports = AccountModel;
Attendance_Model:

```



```

const mongoose = require('mongoose');

const attendanceSchema = new mongoose.Schema({
  student: {
    type: mongoose.Types.ObjectId,
    ref: 'Student',
    required: true,
    // unique: true,
  },
  course: {
    type: mongoose.Types.ObjectId,
    ref: 'Classroom',
    required: true,
    // unique: true,
  },
  attendanceRecord: [
    {
      attendanceStatus: {
        type: String,
        required: true,
        enum: ['absent', 'present'],
      },
      date: Date,
    },
  ],
});

module.exports = mongoose.model('Attendance', attendanceSchema);
Course.Material.model:
const mongoose = require('mongoose');

```

```
const Schema = mongoose.Schema;
```

```
const courseMaterialSchema = new Schema({  
  course: { type: mongoose.Types.ObjectId, required: true, ref: "Classroom" },  
  lectures: [  
    {  
      title: String,  
      createdAt: { type: Date, default: Date.now },  
      filePath: String,  
    },  
  ],  
  quizes: [  
    {  
      title: String,  
      link: String,  
      deadline: {  
        type: Date,  
        required: [true, "quiz should have a deadline."],  
      },  
      createdAt: { type: Date, default: Date.now },  
    },  
  ],  
  announcements: [  
    {  
      title: String,  
      description: String,  
      createdAt: { type: Date, default: Date.now },  
    },  
  ],
```

```

assignments: [
  {
    title: String,
    deadline: {
      type: Date,
      required: [true, "assignment should have a deadline."],
    },
    totalMarks: {
      type: Number,
      // required: [true, "assignment should have total marks."],
    },
    createdAt: { type: Date, default: Date.now },
    filePath: String,
  },
],
});

```

```

const Course = mongoose.model("CourseMaterial", courseMaterialSchema);

```

```

module.exports = Course;

```

```

Student.Model:

```

```

const mongoose = require("mongoose");

```

```

const Schema = mongoose.Schema;

```

```

const studentSchema = new Schema({
  firstname: { type: String, required: true },
  lastname: { type: String, required: true },
  rollno: { type: String, required: true },
  username: {

```

```

    type: String,
    required: true,
    unique: true,
    trim: true,
    lowercase: true,
  },
  avatar: String,
  password: { type: String, required: true },
  role: { type: String, default: "student" },
});

```

```

studentSchema.set(

```

```

  "toJSON",
  {
    virtuals: true,
    versionKey: false,
    transform: function (doc, ret) {
      delete ret._id;
      delete ret.password;
    },
  },
  { timestamps: true }
);

```

```

//=====
=====
===

```

```

// virtual populate

```

```

studentSchema.virtual("classes", {
  ref: "Classroom",
  foreignField: "students",
  localField: "_id",
});

```

```

// virtual populate
studentSchema.virtual("attendances", {
  ref: "Attendance",
  foreignField: "student",
  localField: "_id",
});

const Student = mongoose.model("Student", studentSchema);
module.exports = Student;
Teacher.Model:
const mongoose = require("mongoose");

const Schema = mongoose.Schema;

const teacherSchema = new Schema(
  {
    firstname: { type: String, required: true },
    lastname: { type: String, required: true },
    username: {
      type: String,
      required: true,
      unique: true,
      trim: true,
      lowercase: true,
    },
    avatar: String,

    qualification: { type: String, required: true },
    department: { type: String, required: true },
    password: { type: String, required: true },
  }
);

```

```

    role: { type: String, default: "teacher" },
  },
  { timestamps: true }
);
teacherSchema.set("toJSON", {
  virtuals: true,
  versionKey: false,
  transform: function (doc, ret) {
    delete ret._id;
    delete ret.password;
  },
});
//=====
=====
===
// virtual populate
teacherSchema.virtual("classes", {
  ref: "Classroom",
  foreignField: "teacher",
  localField: "_id",
});
const Teacher = mongoose.model("Teacher", teacherSchema);

module.exports = Teacher;
Utils.file:
const { promisify } = require("util");
const fs = require("fs");
const pipeline = promisify(require("stream").pipeline);
exports.persistDocFiles = async (file, dir) => {
  const fileName =
    file.originalName.split(".")[0] +

```

```

    Math.floor(Math.random() * 1000) +
    file.detectedFileExtension;
await pipeline(
    file.stream,
    fs.createWriteStream(`${__dirname}/../public/files/${dir}/${fileName}`)
);
return `/files/${dir}/${fileName}`;
};

exports.persistImages = async (file, dir) => {
    const fileName =
        file.originalName.split('.')[0] +
        Math.floor(Math.random() * 1000) +
        file.detectedFileExtension;
    await pipeline(
        file.stream,
        fs.createWriteStream(`${__dirname}/../public/images/${dir}/${fileName}`)
    );
    return `/images/${dir}/${fileName}`;
};

```

Package.json:

```

"name": "fyp",
"version": "0.1.0",
"private": true,
"dependencies": {
  "@testing-library/jest-dom": "^5.16.1",
  "@testing-library/react": "^11.2.7",
  "@testing-library/user-event": "^12.8.3",
  "axios": "^0.21.4",
  "bcryptjs": "^2.4.3",
  "bootstrap": "^4.6.1",

```

```

    "express": "^4.17.2",
    "express.js": "^1.0.0",
    "jsonwebtoken": "^8.5.1",
    "lodash.uniqby": "^4.7.0",
    "node": "^17.3.1",
    "node.js": "^0.0.1-security",
    "nodemon": "^2.0.15",
    "react": "^17.0.2",
    "react-bootstrap": "^1.6.4",
    "react-datepicker": "^3.8.0",
    "react-dom": "^17.0.2",
    "react-modal": "^3.14.4",
    "react-router-dom": "^5.3.0",
    "react-scripts": "^4.0.3",
    "server.js": "^1.0.0",
    "start": "^5.1.0",
    "web-vitals": "^1.1.2",
    "yarn": "^1.22.17"
  },
  "scripts": {
    "start": "react-scripts --openssl-legacy-provider start",
    "build": "react-scripts --openssl-legacy-provider build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  }
}

```



```

},
"browserslist": {
  "production": [
    ">0.2%",
    "not dead",
    "not op_mini all"
  ],
  "development": [
    "last 1 chrome version",
    "last 1 firefox version",
    "last 1 safari version"
  ]
}
}

```

Appendix-R: Server:

```

require("rootpath")();
const express = require("express");
const cors = require("cors");
const mongoose = require("mongoose");
const logger = require("morgan");
require("dotenv").config();

const app = express();
const port = process.env.PORT || 5000;

app.use(logger("dev"));
app.use(cors());

```

```

app.use(express.json());
app.use(express.static("public"));

const uri = process.env.ATLAS_URI;
mongoose.connect(uri, {
  useNewUrlParser: true,
  useCreateIndex: true,
  useUnifiedTopology: true,
  useFindAndModify: false,
});
const connection = mongoose.connection;
connection.once("open", () => {
  console.log("MongoDB database connection established successfully");
});

const studentsRouter = require("./routes/students");
const teachersRouter = require("./routes/teachers");
const courseMaterialRouter = require("./routes/course.material");
const classroomsRouter = require("./routes/classrooms");
const studentSubmitRouter = require("./routes/student.submission");
const usersRouter = require("./routes/users");
const attendanceRouter = require("./routes/attendance");
const gradesRouter = require("./routes/grades");

app.use("/students", studentsRouter);
app.use("/teachers", teachersRouter);
app.use("/course-material", courseMaterialRouter);
app.use("/classrooms", classroomsRouter);
app.use("/student-submit", studentSubmitRouter);
app.use("/users", usersRouter);
app.use("/attendance", attendanceRouter);
app.use("/grades", gradesRouter);

app.listen(port, () => {
  console.log(`Server is running on port: ${port}`);
});

```