Web Based Ed-Tech Platform

By
Ariful Hoque
ID: 201-15-13793

SM. Faisal Ahmed
ID: 201-15-14345

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the **Degree of Bachelor of Science in Computer Science and Engineering**

Supervised by

Dr. Sheak Rashed Haider Noori
Professor & Head
Department of Computer Science and Engineering Daffodil International University

Co-Supervised by

Mr. Showmick Guha Paul

Lecturer
Department of Computer Science and Engineering Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

May 14, 2025

APPROVAL

This Project titled **WEB BASED ED-TECH PLATFORM**, submitted by **Ariful Hoque**, ID No: **201-15-13793** and **SM. Faisal Ahmed**, ID No: **201-15-14345** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **14 May, 2025.**

BOARD OF EXAMINERS

Dr. Arif Mahmud (AM) — **Chairman**
Associate Professor & Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Mr. Md. Ali Hossain (MAH) — **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Mr. Amir Sohel (ARS) — **Internal Examiner**
Sr. Lecturer, Internal Member
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Md. Manowarul Islam — **External Examiner**
Associate Professor, External Member
Department of Computer Science and Engineering
Jagannath University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Dr. Sheak Rashed Haider Noori**, **Professor & Head**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

Dr. Sheak Rashed Haider Noori
Professor & Head
Department of Computer Science and
Engineering Daffodil International University

Co-Supervised by:

Mr. Showmick Guha Paul
Lecturer
Department of Computer Science and
Engineering Daffodil International University

Submitted by:

Ariful Hoque
Student ID: 201-15-13793
Department of Computer Science and
Engineering Daffodil International University

SM. Faisal Ahmed
Student ID: 201-15-14345
Department of Computer Science and
Engineering Daffodil University

ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. We are deeply grateful to everyone who has assisted us in one way or another.

First, we express our heartfelt thanks and gratefulness to the almighty for His divine blessing making it possible for us to complete the **Final Year Design Project(FYDP)** successfully.

We are grateful and wish our profound indebtedness to **Dr. Sheak Rashed Haider Noori**, **Professor & Head**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of **Web Application Development** to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartfelt gratitude to the Head of the Department of Computer Science and Engineering, for his kind help in finishing our project and also to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

We would like to thank our entire course-mates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

This project describes the design and development of a Ed-Tech platform aimed at filling the gap between traditional and digital education systems, as it provides the overall modular and user-oriented learning experience. As the demand for remote education rises, our objective is to develop a web-type application in which the learners can study whenever and wherever they want using an internet connection alone. The platform has a vast number of learning features such as pre-recorded video classes, structured assessment, authentication of users, and future-ready AI-assisted learning. It is built upon Spring Boot (Java 17) for backend programming, PostgreSQL to guarantee secure storage of facts, and Docker for viable and containerized deployment. Keycloak is for secure and flexible identity management; IntelliJ IDEA Ultimate and other industry-standard tools are for the development work. It is designed in a modular manner and it is easily expandable and customizable in future. Even though the existing version is only used in a local environment and has not incorporated certain features such as real time AI integration and cloud deployment because resources are limited, it provides a strong base for an expansion. The corresponding functional testing provided reliability of the core features, while API and performance testing is left for future stages. During the project, we adhered to the engineering standards and the impacts of being ethical, social, and sustainable considered, highlighting inclusive and lifelong learning. All phases of the project are described in this report from problem identification and requirement analysing, through methodology, implementation, testing, and difficulties. Additionally, it gets into budget planning and project management strategies as mapped the work against complex engineering problem-solving criteria. By constantly supporting this platform, it has potential to integrate such advanced technologies as virtual reality, online exams modules, interactive code compilers, etc and even AI personalized teachers so that it redefines education delivery and access in the digital world.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Learning is one of the many aspects of daily life that have been transformed by digital technology. Traditional educational institutions, while still useful, sometimes fall behind the demands of today's students. Students confront hurdles such as strict class schedules, high tuition, limited access to professional lecturers, and long trips to class. These challenges are particularly difficult for students in distant or economically poor places.

In response, we created an EdTech platform to bridge these gaps. Using Spring Boot as a basis, the system was designed to accommodate flexible learning models while preserving robust security and the capacity to grow to meet future demands. Our primary goal is to provide a better learning experience by combining important characteristics of online education—such as live and recorded video sessions, rapid evaluations, and interesting study tools—into a single, seamless system.

What distinguishes this platform is its ability to give a personalized learning experience. Students may study at their own pace, take exams and quizzes to track their progress, and ask questions anytime they want, even when an instructor is not present—thanks to our integrated AI assistant. Teachers benefit from the platform's ability to manage classes, submit files, and engage with students.

In addition to helping to solve the shortcomings of traditional classrooms, this integrated strategy aims to bridge the gap formed by many of the online learning platforms now accessible. Although BLC, Udemy, and Google Classroom are all valuable tools, they typically do not provide real-time interaction, intelligent coaching, or flexibility for students with diverse learning styles.

Whether they are enrolled in a school, college, training facility, or learning on their own, our platform is designed for all types of teachers and learners. We hope to build a more inclusive, entertaining, and productive educational environment for today's students by integrating video courses, dynamic evaluations, AI-powered coaching, and community participation.

This research goes into great detail on the platform's design and development, the problems it aims to solve, and how it differs from previous alternatives. Our ultimate objective is simple: to employ technology to improve the accessibility and flexibility of high-quality education for everyone, wherever on the planet.

1.2 Motivation

This program is motivated by a strong desire to improve human learning via the use of modern technology, namely artificial intelligence (AI). We have direct understanding of the benefits and limitations of current online learning environments because we are both students and developers. Even with their flexibility and convenience, many of these platforms still lack adequate engagement, intelligent support, and an intuitive framework for both teachers and students. This prompted us to consider the possibility of creating a more intelligent, responsive, and connected learning environment.

In today's digital world, education should not be limited by time, money, or geography. Our major objective is to make learning simple, affordable, and accessible to everyone, regardless of whether they prefer to learn during the day or at night or if they live in a booming metropolis or a remote village. With only a smartphone and an internet connection, we want people to be able to broaden their knowledge, learn new skills, and progress professionally or personally.

We were also encouraged from a computational standpoint by the growing application of AI to real-world problems. In the absence of human teachers, AI may assist students, automate tasks, provide instant feedback, and personalize learning. By incorporating AI-driven capabilities into our platform, we intend to give intelligent support that will assist users in better understanding ideas, finding answers more quickly, and maintaining motivation throughout their learning journey.

We benefit from resolving this issue on both professional and intellectual levels. We were able to put our experience of frontend technologies, system design, data management, and backend development (via Spring Boot) to work. It has also challenged us to exercise our ideas and work together to solve real-world problems for users. Above all, it has shown us how technology can truly enhance people's lives when used correctly.

This platform is designed to be a valuable resource for self-learners, institutions, and teachers. We aspire to contribute to a society in which everyone, regardless of background, has the opportunity to learn and prosper by increasing education's interaction and inclusion.

1.3 Objectives

The main objective of this project is to develop a smart, user-friendly EdTech platform to enhance online learning effectiveness and engagement. We would like to create an environment where professors and students can also be serviced with such features as live lessons, quizzes, and AI-powered support. Flexibility is one of our highest priority goals and we would like the learner to not be bound to a specific schedule because they learn at their own pace, at their own convenience, from anywhere in the world. At the same time we want to make things easier for administrators so that we can provide tools for managing users, generate batches, and easily upload course materials.

Another important goal is to merge the benefits of a regular classroom educator with the simplicity of online learning. This would mean the creation of a system for a real time interaction—not only with the student and the instructor, but also with the AI assistants who will answer questions, provide assistance, and maintain student interest. We also desire our platform to be user-friendly when it comes to integrating it into existing systems schools, colleges and training centres make use of to enable them to increase their online services as well. Our ultimate goal is to help make education more accessible, tailored and fun for all of the players using it, either through smart features, scalability, or meaningful human-AI partnership.

1.4 Methodology

We built this EdTech platform through a step-by-step development process that combined technical skill with attention to user needs. Instead of relying on assumptions, we started by exploring real challenges faced by students and teachers using current digital learning systems. Visits to schools and direct user feedback helped us gather concrete input. That shaped our list of functional requirements based on actual problems rather than theoretical ones. Technically, we used microservices-based structure. Spring Boot powers the backend, and the frontend is taken care of with Angular. This technique enabled the development of various components independently while still making sure they do fit in together. Maintenance and lures for future improvements were also less daunting to manage. In order to further support users, we brought in the ai-based versions of these features, such as personalized suggestions and quick help during learning activities. We brought together qualitative and quantitative research methods to guide us on how we developed and enhanced the platform. This included interviews, surveys and technical evaluation. Rather than thinking only for functionality we wanted to develop such a system which could run easily on various devices with security and responsiveness. We stayed close to progress all the way through development. Testing occurred as a habitual activity, and users' feedback was helping us on our continuous improvement with how the system works and feels to use.

1.5 Project Outcome

This is not only about the construction of one more digital education instrument. The aim is to build a learning world that really works for all parties involved: students, teachers, and institutions. With the platform, instead of having just lessons imparted online, one is expected to have a better interactive and personalized environment. Functions such as real-time communications, AI-governed learning, and convenient access, are supposed to support users to evolve not only in terms of academics but also in terms of critical thinking, self-direction and problem-solving. This we view as a way forward to building learners who are well fitted to today's complex world. For the educational institutions, the platform also creates opportunities to expand the scope of their influence, enhance training quality, and simplify their work with the course and student communication. Somewhat grandiloquently, we view this project as an exemplar for future developments in EdTech – a project that puts education at everyone's doorsteps, makes learning a more fun thing, and opens doors for a more inclusive, and global learning experience.

1.6 Organization of the Report

This report is organized in multiple chapters that will cover different phases of each project to provide a complete view of how the Ed-Tech platform developed from start to finish. In the first chapter, the introductory part of the report, we describe the general idea of the project, introduce the issue to solve and mention our objectives and intent. This part mentions how relevant the project is to the modern educational environment. In chapter two, the Background summarizes modern educational technologies and systems. It emphasizes and assesses the drawbacks of existing solutions such as BLC, Udemy, and Google Classroom. This chapter outlines the shortcomings of current educational systems that our platform was designed to improve. It introduces the research strategies we used throughout the planning and development stages. In chapter three, discussions on system architecture, how we selected our technology stack, the analysis of requirements, and the overall development cycle is mentioned. We also describe how we gathered input from users and used their feedback to guide our decisions. We explain in the Implementation and Results section how the platform was developed with Angular and Spring Boot. We highlight critical details about testing results, performance testing, and early users feedback to establish whether the system does what it says. The Engineering Standards and Design Challenges chapter goes in depth into the technical problems we have faced, like linking up individual modules, enhancing security, and ensuring the platform could grow to whatever scale was necessary. We also explain how we adhered to standard engineering practice to make the system dependable and easy to maintain. The concluding part addresses the key achievements, as well as critical lessons learned, and explores what the platform can aim for in the future. Its emphasis is on the larger vision – the use of this technology to bring digital education to a more effective, inclusive and more future ready form of delivery.

Chapter 2
Background

2.1 Introduction

The digital frontier has revolutionarily changed the way people access education. Online study is thriving now, largely due in part to the flexibility and lower cost of electronic study over traditional class settings. But even with this growth, many platforms are struggling to provide effective educational experience. Such features such as live communication, personalised guidance and the responsive support systems are commonly not there.

Though traditional schools and universities have the benefit of a structure and credibility, they aren't always accessible. Overly high tuition, set schedule, and travel requirements can be barriers – especially for areas, remote or under served. Some digital tools, such as Google Classroom, Udemy and Blended Learning Centers, have come a long way, but they tend to use static pre-recorded content too heavily. These setups are not always able to provide the type of real time interaction or individual learning that the learners of today require.

Our project took shape as a reaction to these gaps. The mission was to develop a system that combines the merits of the traditional approaches with the flexibility and availability of the new technologies. By integrating live video courses, AI-driven help systems, and dynamic evaluation tools, we want to create a flexible, user-friendly solution that is ideal for both individual students and educational institutions.

This chapter examines the history of modern Ed-Tech solutions, including a comparison of well-known platforms and the remaining gaps in the online learning industry. To understand why our platform is required and how it differs in the rapidly increasing sector of digital education, we must first address the gaps.

2.2 Literature Review

With the increasing rise of online education, several platforms have evolved that provide a variety of learning methods and technologies. However, the majority of these sites exclusively address one or two components of education, such as video material or assignment sharing. A really effective online learning system should include interactive classrooms, real-time communication, and tailored learning support. In this literature analysis, we look at some of the most popular current platforms, including Google Classroom, Udemy, and Blended Learning Center (BLC), to determine their strengths and drawbacks, as well as the important areas for development.

2.2.1 Similar Applications

We examined a number of existing web and mobile applications, as well as research articles on online learning systems, to gain a better knowledge of the state of educational technology today and how our platform fits within it. These tools helped us shape the foundation of our own design by emphasizing the components of digital education that are now lacking, what needs to be improved, and what works well.

Google Classroom is used by many schools to help teachers share documents, give homework, and check student progress. But it mostly focuses on sharing files and sending messages. It doesn't have live video lessons or smart help tools, which can make it less engaging for students who learn better through interaction.

Udemy has thousands of courses in video format. These courses are good for learning and some have short tests or quizzes. But, students observe videos and cannot ask questions or get it right away, and that is why students feel one-sided learning.

Coursera and edX kind of have more formal listing, college level courses, even full degrees. Apart from a few live classes, most lessons remain pre recorded. These platforms are excellent for deep learning, but they can be too expensive and too difficult to use for some people and smaller schools.

We also investigated the Blended Learning Centre (BLC) which assists with such things as marking and feedback. But it lacks video call capabilities, and most users struggle with it. Also, it works slowly when too many people use it, which also disturbs the lessons.

In order to create our own platform better, we researched these systems and read papers on what works well in online learning. For example, Sun and Chen's 2016 research emphasizes the importance of technology integration and participation in successful online learning. Khan et al. (2019) address the design of a responsive e-learning system for higher education, emphasizing the importance of mobile compatibility and user-friendly interfaces. In a wider regional context, Afroz (2023) investigated the challenges of online education in Bangladesh, such as poor communication, a lack of personalization, and technological constraints—all of which our platform aimed to overcome.

Additionally, we were motivated by mobile programs like Khan Academy and Duolingo. Our decision to include AI assistants was inspired by Duolingo's use of AI to adapt learning pathways based on user progress. Although Khan Academy combines exercises and instructive films, it does not currently offer dynamic, real-time help like the AI-powered assistance we want to offer.

To summarize, while each of these systems has something beneficial to offer, none of them combines efficient batch or course management, personalized AI support, and live interaction into a single solution. Our platform is intended to bridge that gap by providing a balanced solution that combines the freedom of online education with the structure and support of traditional classrooms, all improved by intelligent features.

2.2.1.1 Google Classroom

Google Classroom is a popular platform for managing educational resources. It enables professors to communicate with students, create and distribute assignments, and track progress. However, Google Classroom tends to focus more on document sharing and assignment management, with less emphasis on live, interactive learning. The lack of integrated video conferencing and personalized AI features limits the platform's ability to engage students and offer them greater freedom in their learning journey.

2.2.1.2 The Udemy

Udemy, a worldwide online learning platform, enables educators to market video courses on many subjects. With all due respect to its vast content library, most courses are pre-recorded, so interaction with instructors is very limited. Without the real-time assistance, students cannot immediately get quick answers to their questions or even some personalized guidance, which makes many students get disinterested, particularly students who require immediate assistance.

2.2.1.3 The Blended Learning Center (BLC)

The Blended Learning Center (BLC) facilitates the ability to communicate, track progress and keep the assignments organized. Even though it assists in organizing online classes, many users claim the interface to be hard to use. The platform also has slow server speeds that can cause poor performance. Plus, it couldn't be as engaging without live video or real-time interaction, as it could for fully interactive learning.

2.3 Gap Analysis

An overview of different EdTech platforms and research indicates major differences in how such platforms support teachers and students is explained below in table 2.1. Although numerous platforms excel in tasks such as course management or content delivery, very few do a true interactive, smart learning. For example, Google Classroom and Blended Learning Center (BLC), do not provide live video capabilities which will restrict communication between students and teachers. On its part, however, sites such as Udemy provide a large library of pre-recorded courses, but lack live interaction as well as real time support for students. Lack of AI powered help in most current systems is one of the key problems. Students usually have to wait for teachers' responses or rely on narrow interactions with peers with no instant, smart assistance aimed to help students deal with the problems. Moreover, there is a lot of slow performance, confusing user interfaces and the lack of customization which makes both teachers and students more difficult to get involved entirely. Our solution addresses these issues through a more complete and a more flexible platform. We've created it so that there is live video teaching, interactive assessments, real-time

communication and AI-powered support all in one place. We have set a goal to develop a versatile and easy to use environment that can rely on cutting edge technology to deliver a more engaging, effective and personalized learning experience. Using this approach, our goal is to fill the gaps of the current EdTech solution and provide a more connected, more impactful online learning place.
for fully interactive learning.

Table 2.1 Comparative analysis with previous work

SL No	Educational Platforms	Services	Shortcomings
1.	Google Classroom	Assignment creation and distribution, Classroom communication, Document sharing and collaboration, Grading and feedback, Announcements and class updates, Integration with Google Drive, Class calendars, Student progress tracking, Real-time collaboration, Parental access (Guardians feature)	No live class, Limited Assessment Tools, Simplified Gradebook, Limited Customization Options, Communication Challenges, and Integration with Third-Party Tools
2.	Udemy	Udemy Free Courses, Udemy Certifications, Udemy Live, video lectures and assessments, pre-recorded content	Quality Control, Limited Interaction, and Limited Support for Advanced Topics
3.	BLC	Assignment creation and distribution, Document sharing, Grading and feedback, Student progress tracking,	complexity may pose a learning curve, and does not have video class capabilities

2.4 Summary

In this section the development of online learning platforms were discussed, with the help of a few examples Google Classroom, BLC and Udemy. Although each of these platforms supports online education as its strength, they also have their limitations, such as lack of customisation, no AI-based assistance and no real time interaction. By studying the available research and teaching methods, we realized that there is a massive gap in the EdTech services available. Our platform tries to bridge these gaps by integrating the comprehensive testing, AI-powered help, live classes and user friendly interface in a single solution. This strategy will ensure a more flexible, reachable and interesting study.

Chapter 3

Research Methodology

3.1 Requirement Analysis & Design Specification

In order to create a strong and a beneficial Ed-Tech platform, we followed a structured process of technical planning, and constant development, and research of the needs of the users. Our primary aim was to learn about the true problems being faced by educators and students and to create solutions that would solve these problems. We aimed at developing a user-friendly platform with a capability to be expanded, scalable, and combine the advanced technologies such as AI support, and live video connectivity.

We collected data with interviews, visits to educational institutions and teacher and student feedback. For the purpose of determining the features to adopt or improve, we also compared the existing platforms. When developing, we were keen in handling critical stages like requirement analysis, system design, implementation, testing and evaluation etc..

3.1.1 Overview

This chapter outlines the design requirements, analysis and development process of our platform, Ed-Tech platform. It was our primary objective to develop a system that is effective and accountable to real needs of teachers, students and educational institutions. To achieve this, we used a research based, structured approach. First we identified common challenges of existing platforms: low engagement, bad user experience, the lack of real-time communication. We then carried out fieldwork in real places of education, interacting with students and watching their learning experiences for useful insights.

Such details enabled us to outline the functional and technical requirements for our platform. We made a priority out of working on an easy to use, scalable, and flexible design. The platform uses Angular for the frontend, Spring Boot for the backend and features with AI powered support for personalized service. We embraced a microservices architecture to eliminate lack of scalability and failure to be fault tolerant. By integrating sound technical design and research into the working detail, we've developed a platform that is reliable and easy to use and can offer a much better digital learning experience.

3.1.2 System Design

To develop a trustworthy and user-friendly Ed-Tech platform, we decided to use a thoughtful, modular approach, supported by modern methods of development. We started off by highlighting the biggest functional and technical requirements on the basis of actual user reviews and on the basis of educational institutions. After careful consideration of these requirements, we designed the system architecture with the use of a microservices approach. This design enables each of the application's components to work independently as they work within the same platform, and this enhances the system resistance to failure and scalability and flexibility.

For our frontend we decided to use the powerful TypeScript framework, Angular for an interactive user experience. We improved the visual design through SASS as a result of custom styling and Font Awesome for icons. We chose the backend by using Spring Boot, an extensively used Java based framework which allows for the efficient creation of reliable web applications. With Spring Security and Spring Data JPA, the backend manages data operation and authenticates the user securely.

The platform is hosted in an Apache Tomcat server as we ensure that our Java application based web services run efficiently and securely . Each service is linked to a central database such as PostgreSQL based on what is configured for the deployment.

Thanks to the architecture based on multiple microservices, such functionalities as course uploads, batch creation, user management, and, for example, AI interaction can work separately. This modularity also enables us to add new features such as AI models of personalized learning support without disturbing the entire system. The architecture is designed with the present needs in mind, as well as scalable and inclusive enough for future developments, presenting a long term solution for current education.

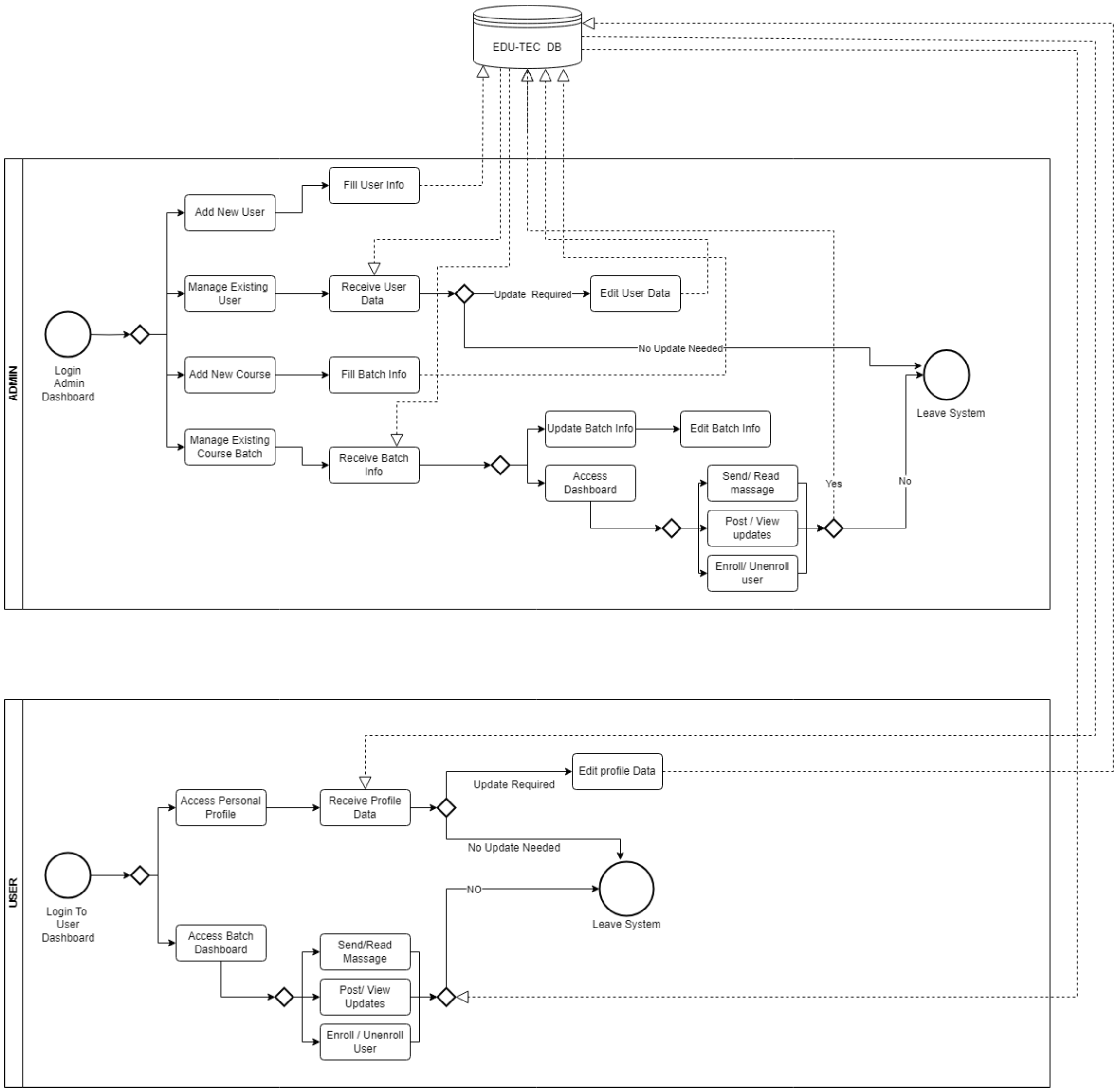

Figure 3.1: Business Model

Figure 3.1 summarizes the essence of the functioning of the educational platform, including the interaction both with the admin-side and with the user-side within the system. The platform combines the mechanisms for handling user data, course content and personal dashboards, and is underpinned by a centralized database known as EDU-TECH DB.

Admin Panel Workflow

Upon logging into the dashboard, admins can perform several key functions:

User Management:

Administrators can either add a new user or update existing user records. These actions trigger interactions with the database to insert or update relevant data entries. In cases where user info is already present, it is fetched and shown for review before updates are committed.

Course Management:

Similarly, the admin interface allows for adding new courses and managing previously created course batches. The system fetches batch info to validate entries before storing them in the database.

Data Processing and Distribution:

Once all inputs are finalized, the system aggregates user and course data, making it ready for review. Admins can access this consolidated information and proceed to:

- Send batch records via email,
- Print detailed views,
- Email filtered user data to instructors or stakeholders.

This structured flow ensures consistency, data integrity, and quick access to essential information.

User Panel Workflow

End users—primarily students—access their interface by logging into the dashboard.

Profile Access:

Users may review their personal information. If profile data requires changes, the system allows an update request to be submitted, which is then reflected in the database.

Batch Enrollment and Messaging:

Users can explore available course batches. Upon selection, they receive related messages and can:

View batch content, Receive notifications, Get direct email communications.

In the event that no update is needed, the system provides a seamless exit path to leave the platform.

3.1.3 Functional and Nonfunctional Requirements

3.1.3.1 Functional requirements

The two primary roles for users on our Ed-Tech platform are Administrator and User, which includes students and teachers. Within the system, each position is capable of performing a certain set of activities.

Administrators may administer the entire This includes the creation and modification of course batches, assigning users to those batches, uploading course content, issuing announcements or changes, adding new users and administrators, and logging into the system. They can further manage the learning environment, monitoring user behavior and ensuring that information is properly structured.

Users, on the other hand, may see and amend their personal profiles, log in, participate in discussions, and access the course content within the batches to which they have been allocated.Students have a possibility to ask such things either from the AI assistant or from their professors besides taking lessons and making examinations. The teachers can touch pupils, provide things, and observe their development. This design enables a smooth teaching and learning process.

3.1.3.2 Non-functional requirements

Apart from its main features, the platform needs to meet several non-functional requirements to ensure both long-term success and effectiveness of the platform. First, it must be compatible with diverse devices including desktops, laptops and smartphones. The system has to be high performance with quick load times and smooth transitions regardless of several users accessing it simultaneously.

Security is another essential aspect. In order to secure user data, the platform should implement secure communication via encryption, and role-based access controls together with a web-safe login system. Spring Security has a central role to play in securing sensitive information such as user profiles and course materials and in ensuring that user information is secure with good security measures.

The platform is planned to be scalable and manageable. Through its modular architecture, it is possible to upgrade or substitute utilization of individual parts without disturbing the whole system. With this method, it is easy for us to add new features in the future e.g. other AI models or language support without altering the underlying framework.

Finally, usability of the platform is its highest priority. The interface should be straightforward with easy to use functionalities, particularly for uses that may not be that tech savvy. A simple layout, straightforward design elements; useful feedback messages; all this ensures that interaction with the platform is easy and smooth for users.

3.1.4 Context Diagram

th.

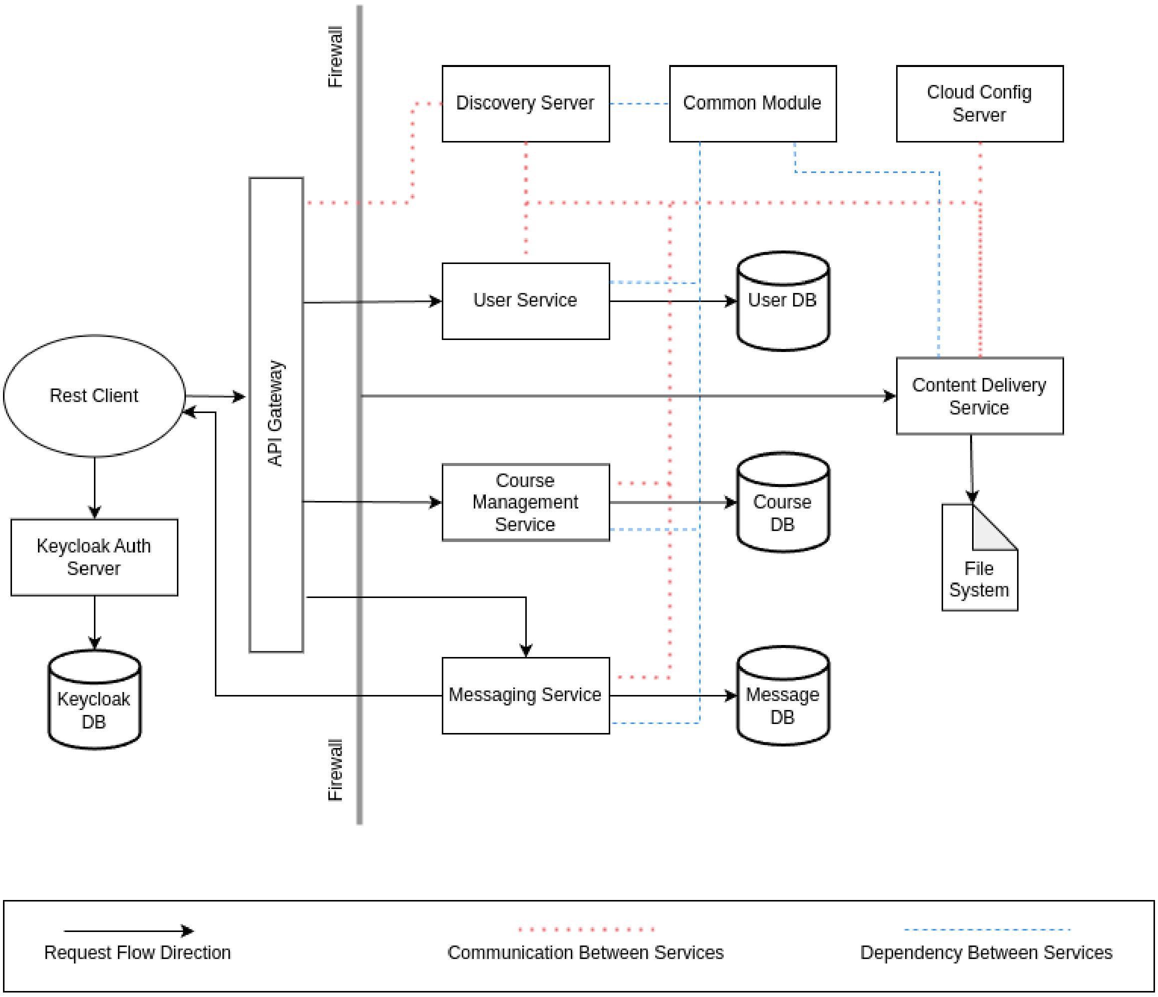

Figure 3.2. Design Model Diagram

The overall context diagram provides a big picture of the Ed-Tech platform in figure 3.2, demonstrating how external factors correlate with the system, and how data is funneling between them. This diagram helps the understanding of how users, external systems and the platform itself are interrelated.

The primary entities outside the program are students, teachers, admins, followed by AI assistants. The admins rely on this to run accounts, place groups and upload course materials. Teachers post their courses and schedule lessons online using the platform and interact with their students online too. Students use the platform by accessing live classes, taking tests, reviewing course materials and reaching the AI assistant for assistance. The AI assistant developed with the assistance of complex algorithms offers individualized guidance on learning and gives responses to students' questions and provides additional material according to students' progress.

The system itself operates as the heart of the platform: all data interactions are being managed by it. It accepts requests, reads or updates information from the database, receives the input from the user and returns relevant information to the users. This design guarantees the platform can handle the communication, content delivery, testing and AI interactions smoothly thus maintaining the entirety of the user experience smoothly.

3.1.5 Data Flow Diagram Level 1

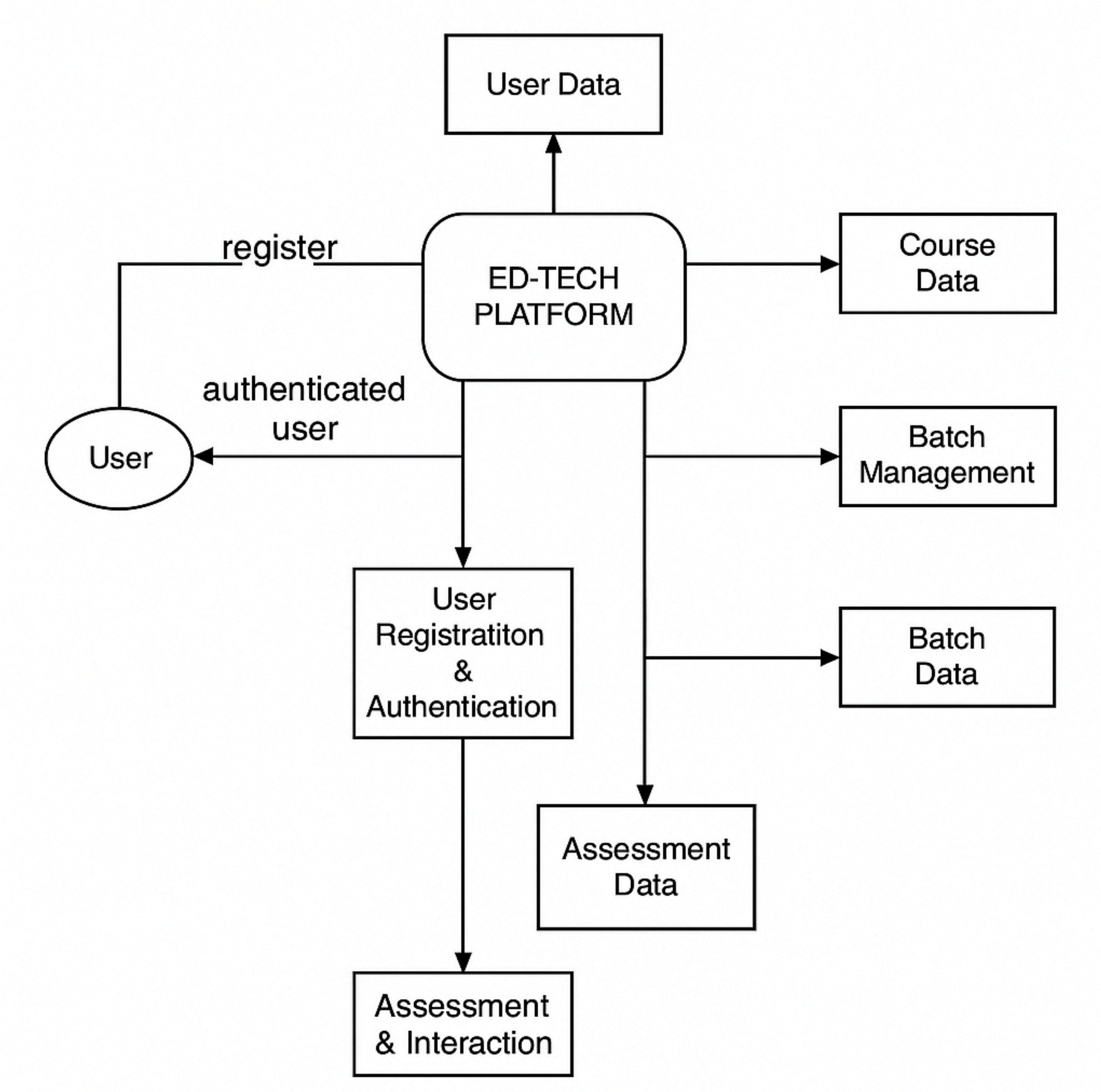

Figure 3.3 Model Data Flow Diagram

Figure 3.3 displays the data flow between users, data storage and core system functions at our Ed-Tech platform (Level 1 DFD) (Level 1 DFD). The ed-tech platform is the bedrock of the system which considers all operations. The process begins when the user is registered or logs into the platform to begin interaction with the platform. In the register and authenticate user stage, the system authenticates to allow safe access. The user is given access, once the authentication is successful, to certain platform features that are tailored to each user based on their role.

3.1.6 UI Design

We have designed the user interface of our Ed-Tech platform, with role specificity, aspect of simplicity and user friendliness in mind. Two major user roles are implemented by the platform: administrator and student with individual specific dashboards corresponding to their duties and tasks. In order to have smooth access from different devices, the design is clear, compatible and will present a uniform experience whether viewed on PCs, tablets or smart phones.

3.1.6.1 Admin Interface

The UI admin (Fig. 3.4-3.10) gives full control over the backend aspects of the platform. After logging into the application, an administrator is greeted by a dashboard that has batch statistics, platform data, and active users. The sidebar provides access to the major items such as teacher assignments, batch creation, user management and course uploads in a very short time. Real-time validation and confirmations process makes adding or editing users and batches easier. More business-like in nature, the design is efficient and clear in the sense that administrators must be capable of managing the system without getting tangled up in technicalities.

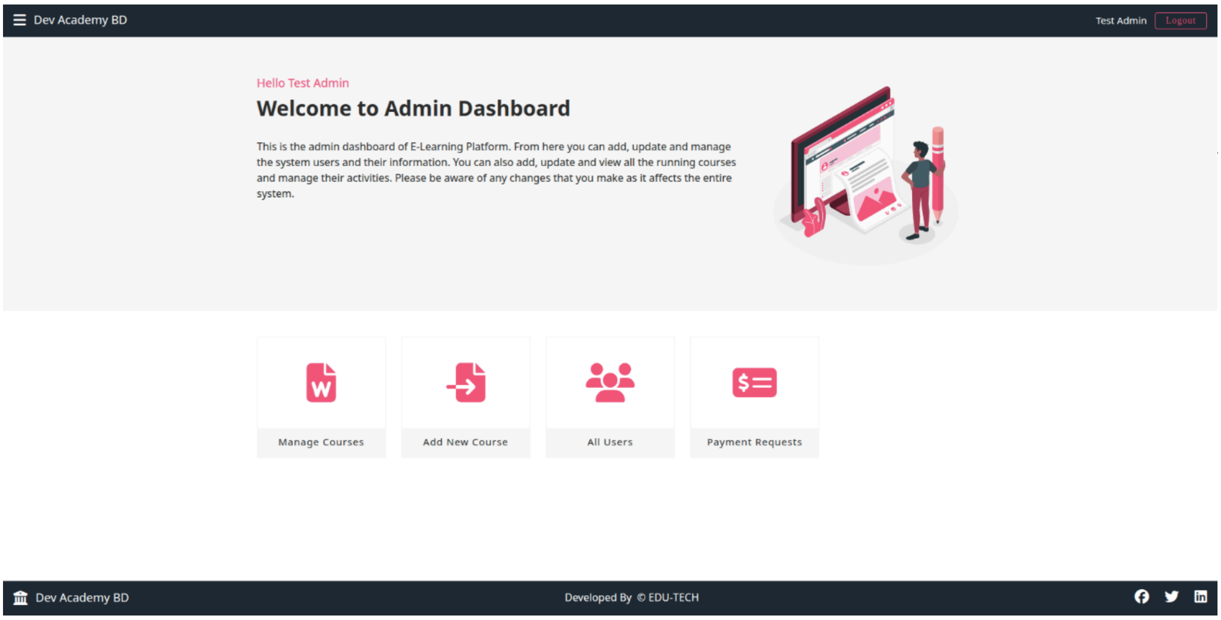

Figure 3.4 Admin Interface

In figure 3.4 we can see the welcome page of the admin dashboard. From here admin can choose any option and take actions accordingly. Here we can see four options such as manage course, add new course, all users, payment requests.

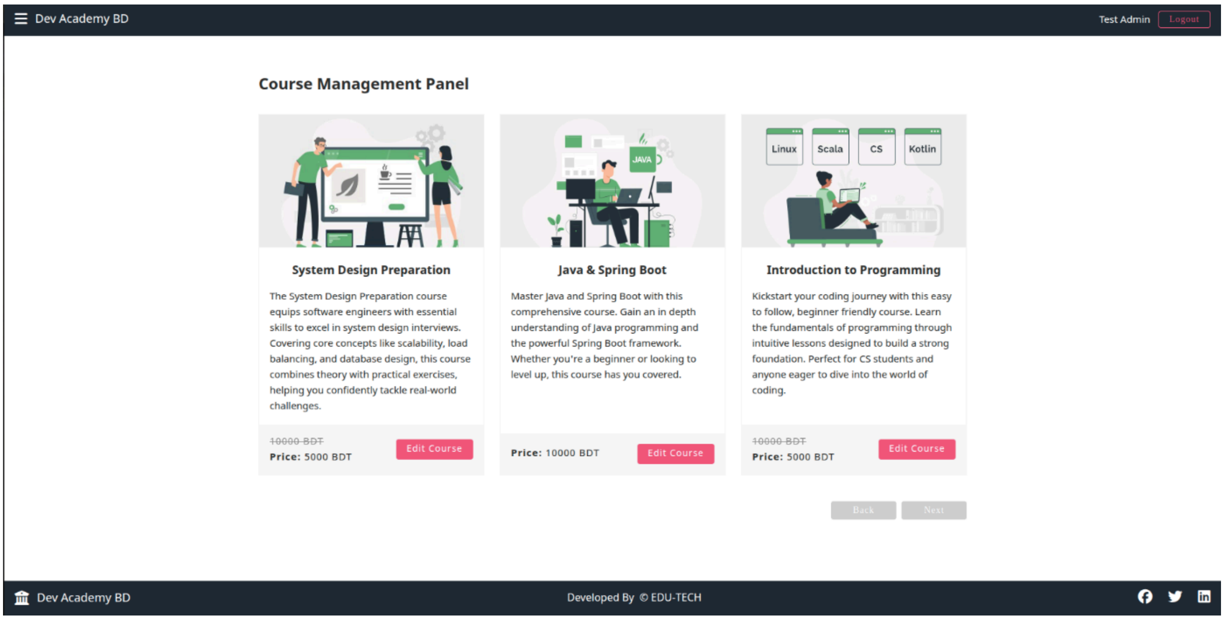

Figure 3.5 Admin Interface

If admin chooses to manage course action as shown in figure 3.4 they'll see a UI like this shown in figure 3.5 and will be able to edit courses accordingly. Here they can choose to edit whatever course that is needed to be updated.

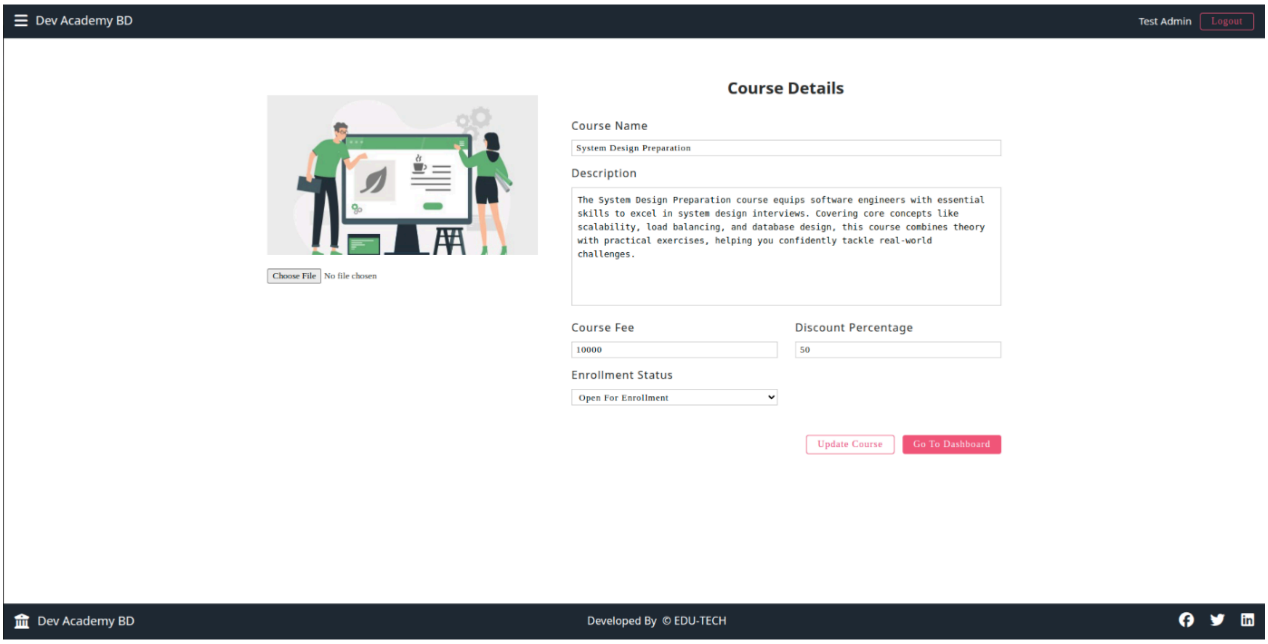

Figure 3.6 Admin Interface

This is the UI in figure 3.6 an admin will see if they choose to edit any course. Here they can easily update the course title, description of the course, update the course fee and apply any discount if there is any. After clicking the update course button the course will be updated.

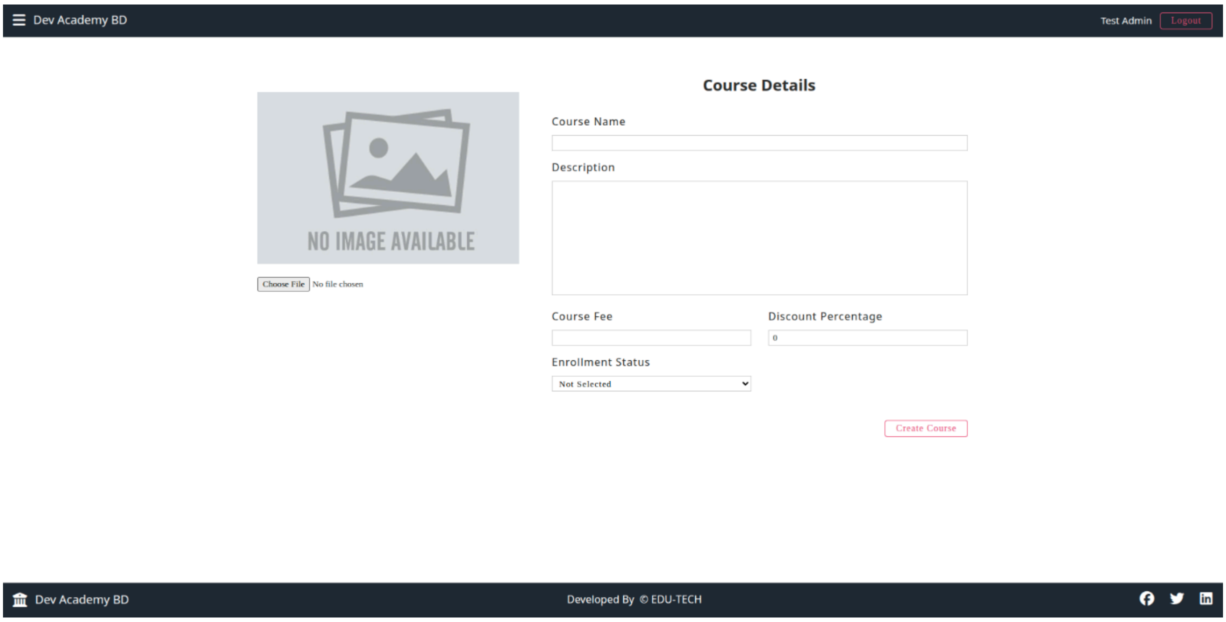

Figure 3.7 Admin Interface

In figure 3.7 we can see the UI of creating a course by an admin. Here an admin has to put an image of the course and write its title and description and update its fee and by clicking the create course button the course will be created.

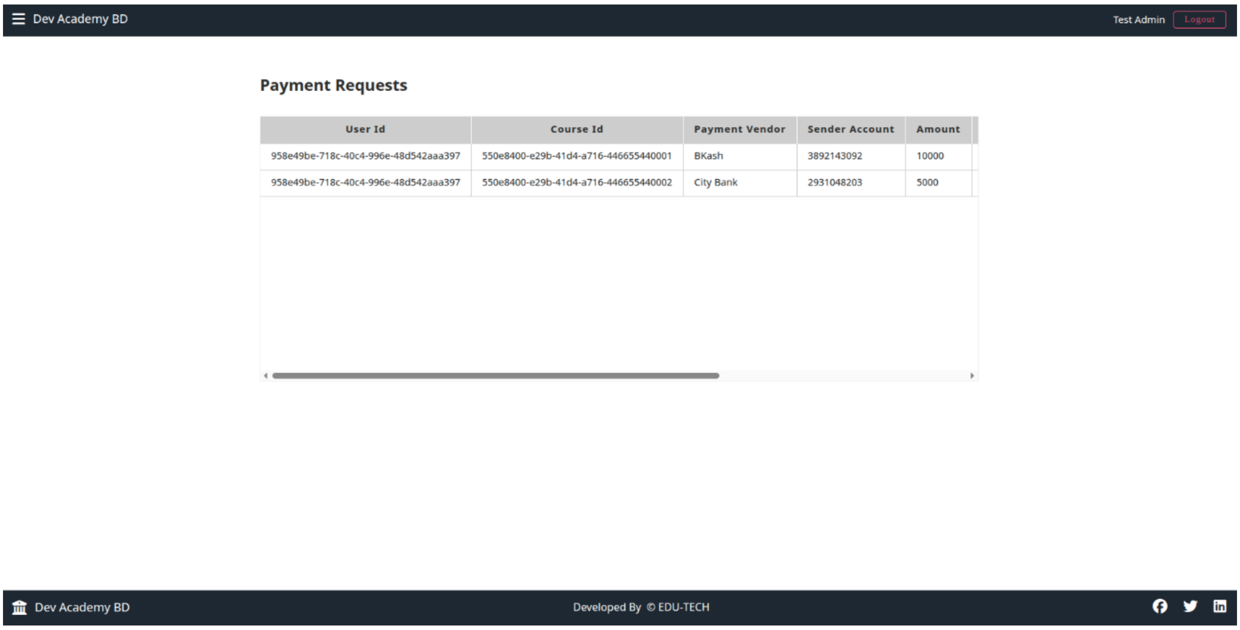

Figure 3.8 Admin Interface

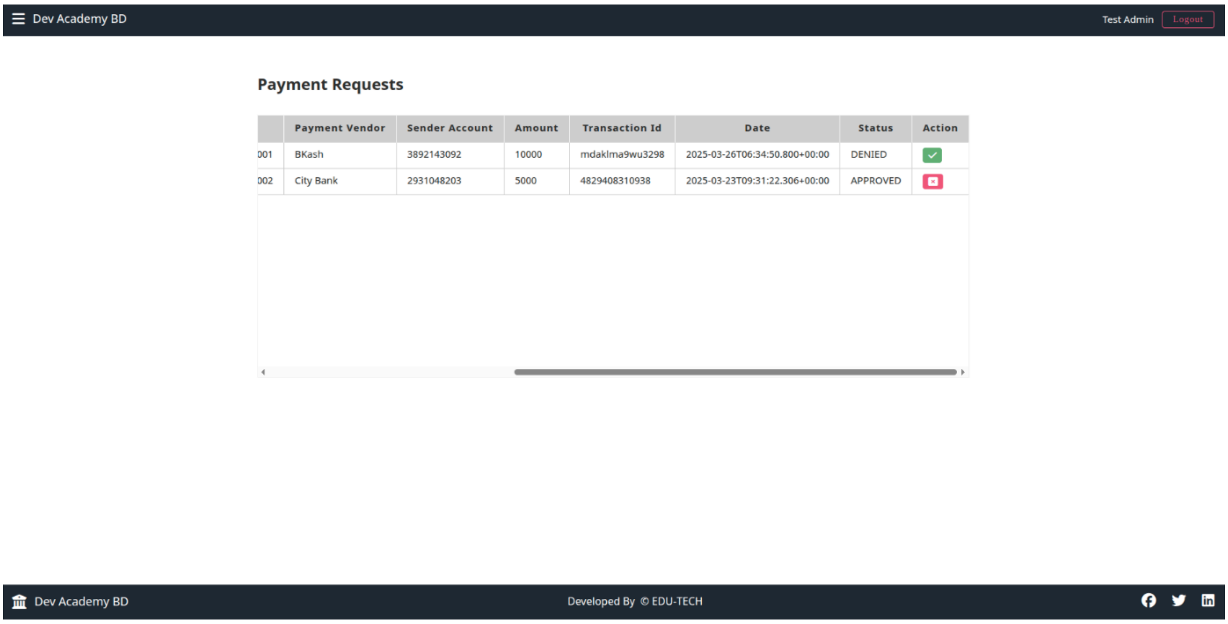

Figure 3.9 Admin Interface

In fig 3.8 and 3.9 we can see when a user pays for a course and the admin has to approve it. After a user pays for one or multiple courses it stays in review. Admin reviews the transaction number and verifies. If a verified admin clicks the green checked button or the cross button for wrong verification.

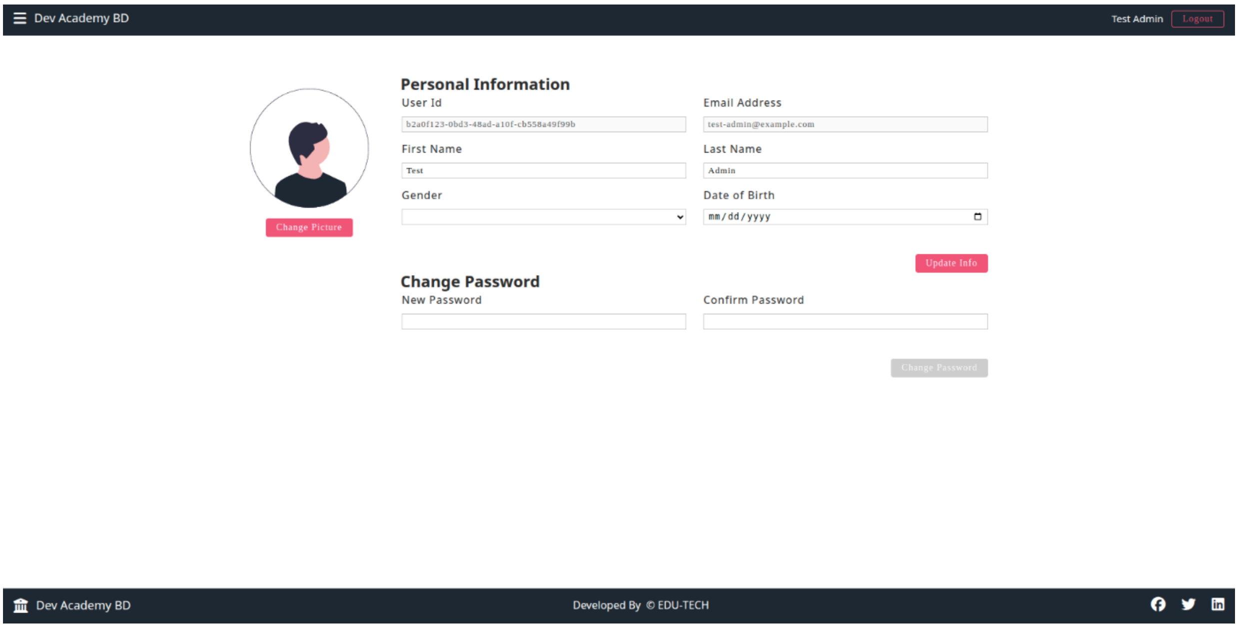

Figure 3.10 Admin Interface

In fig 3.10 we can see the profile page of an admin where they can change their name, gender, date of birth and password.

3.1.6.2 Student Interface

The Student UI (Fig. 3.11-3.17) is centered around connection and accessibility. After logging in, students can view their registered courses, along with upcoming classes and assignments. The dashboard features sections for communication, discussion forums, quizzes, and course materials. Students can engage in real-time chats with instructors and peers, access shared files, and ask questions. The interface uses icons, progress bars, and status indicators to create a user-friendly and engaging learning experience. With features like smartphone compatibility and an adaptive design, students can easily study from anywhere.

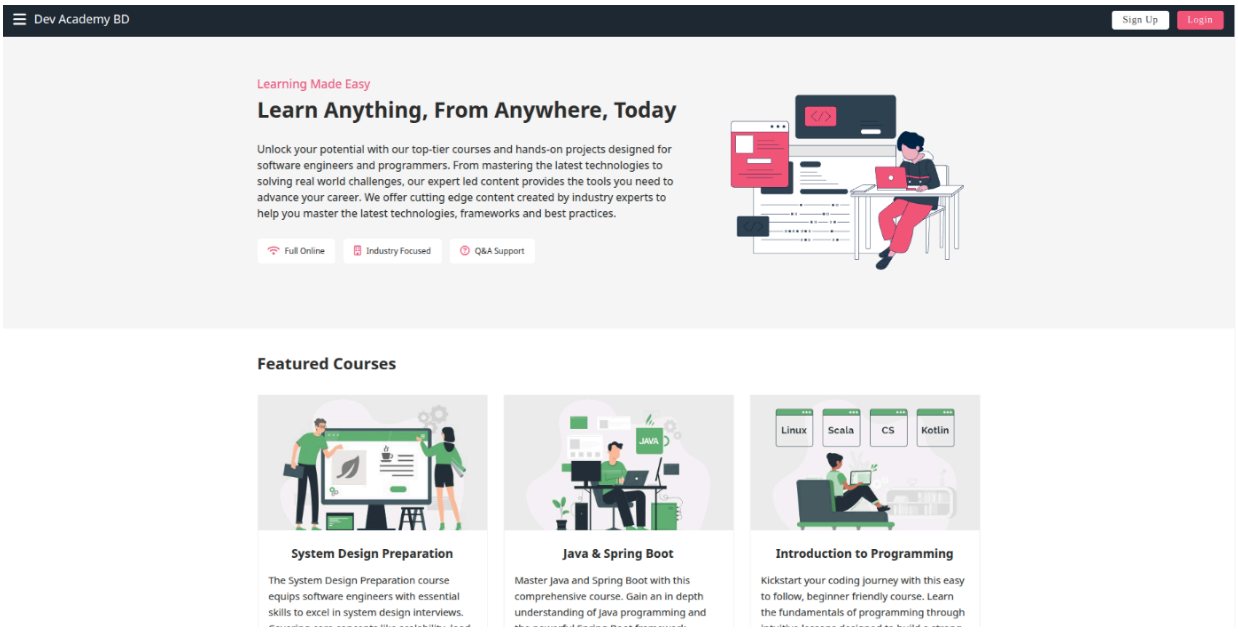

Figure 3.11 Student Interface

In figure 3.11 we can see the landing page of any user who is browsing the website. From here any user can navigate and see courses login or sign-up.

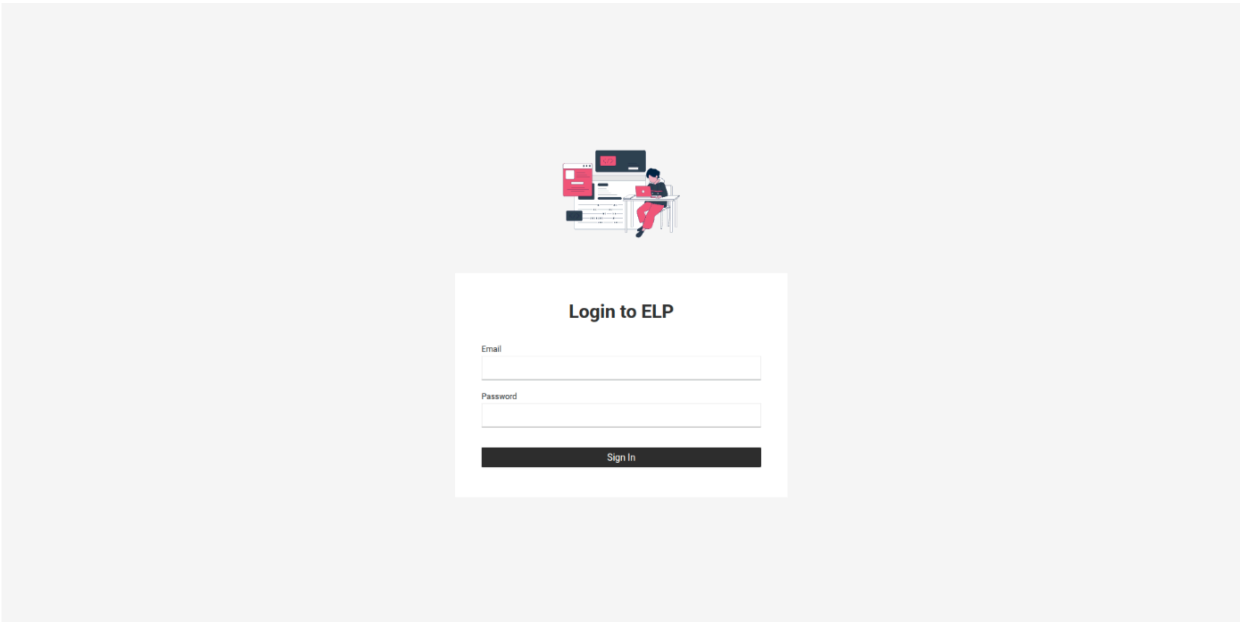

Figure 3.12 Student Interface

In fig 3.12 we can see the login interface of the user. By providing valid email and password a user can login to the website.

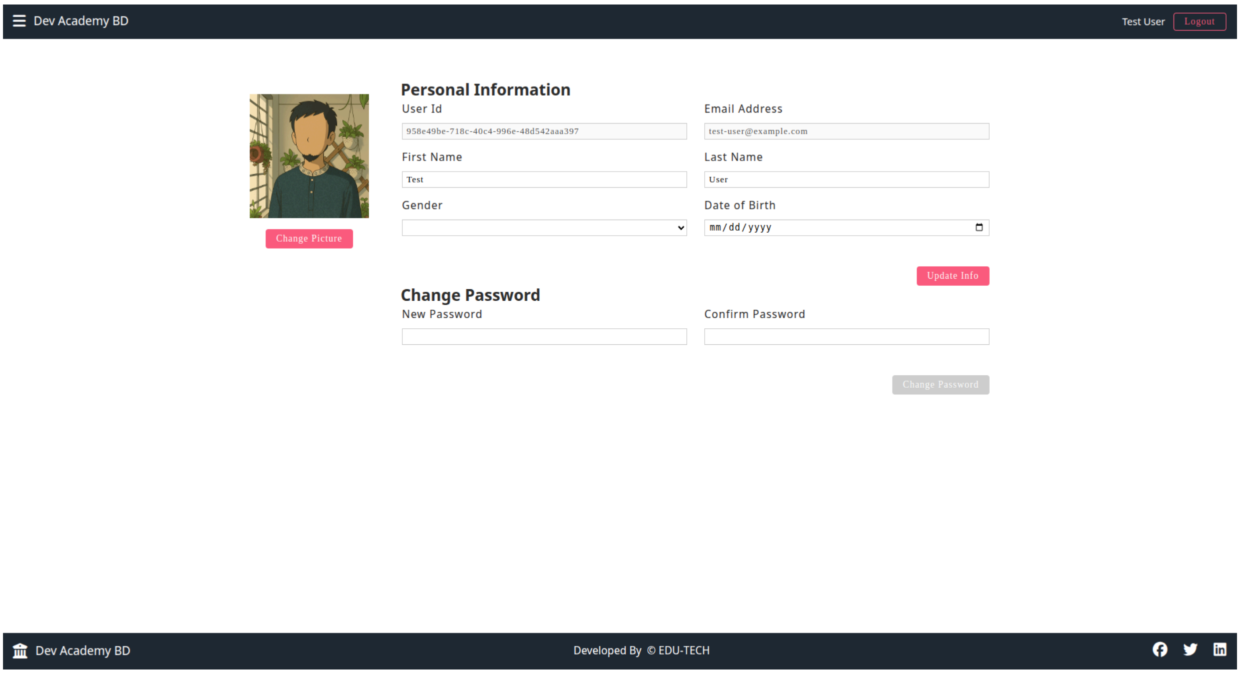

Figure 3.13 Student Interface

In fig 3.13 we can see the profile of the user. Here the user can edit their basic information and update their password. After updating information by clicking the update info button, users can update their information or change their password.

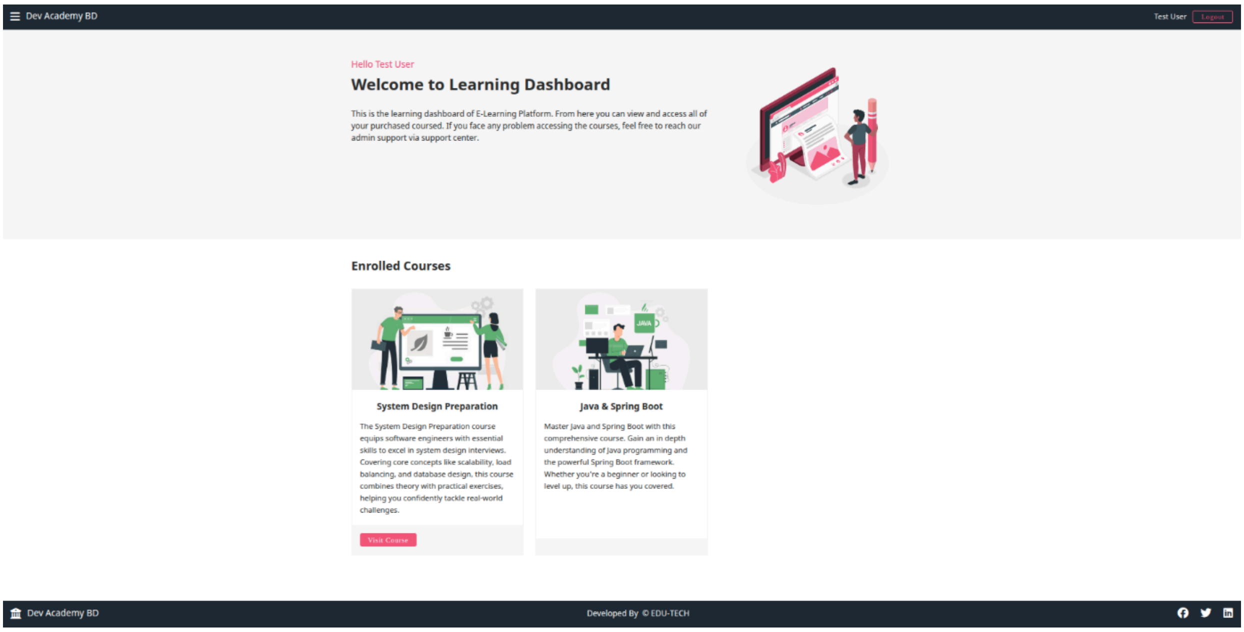

Figure 3.14 Student Interface

This is the user dashboard shown in figure 3.14 where users can access their courses and see if any of their requested courses got rejected or not. Accepted courses can be visited by the user.

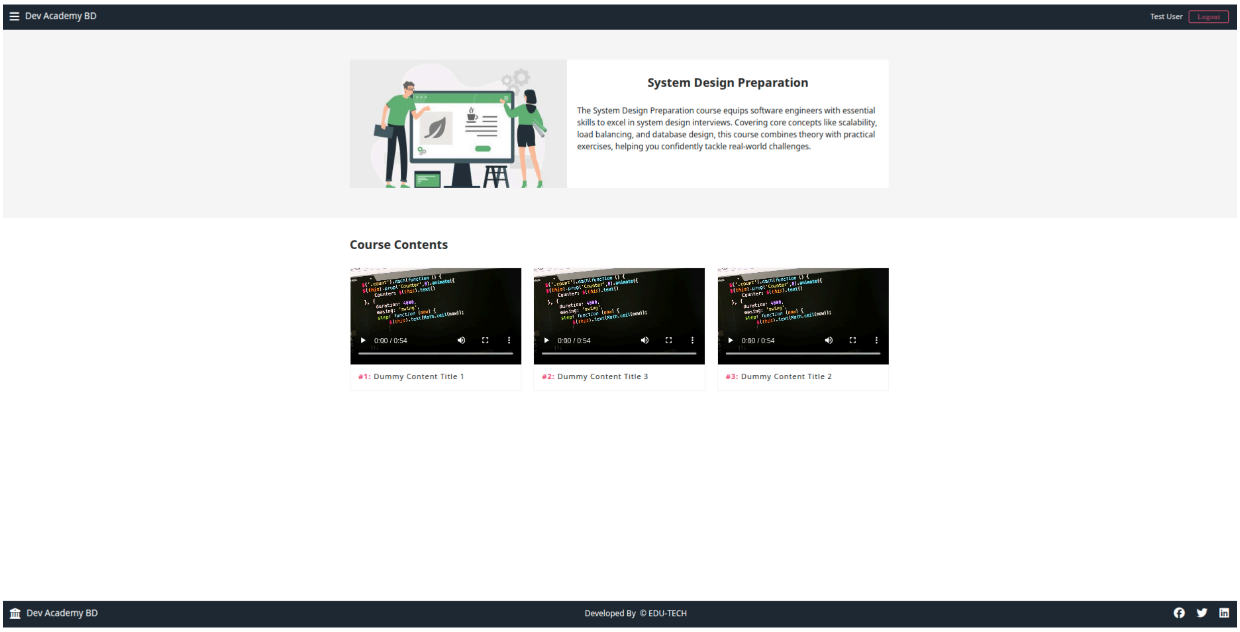

Figure 3.15 Student Interface

After entering a purchased course the user can access its content like shown in figure 3.15. They can view the content of the course.

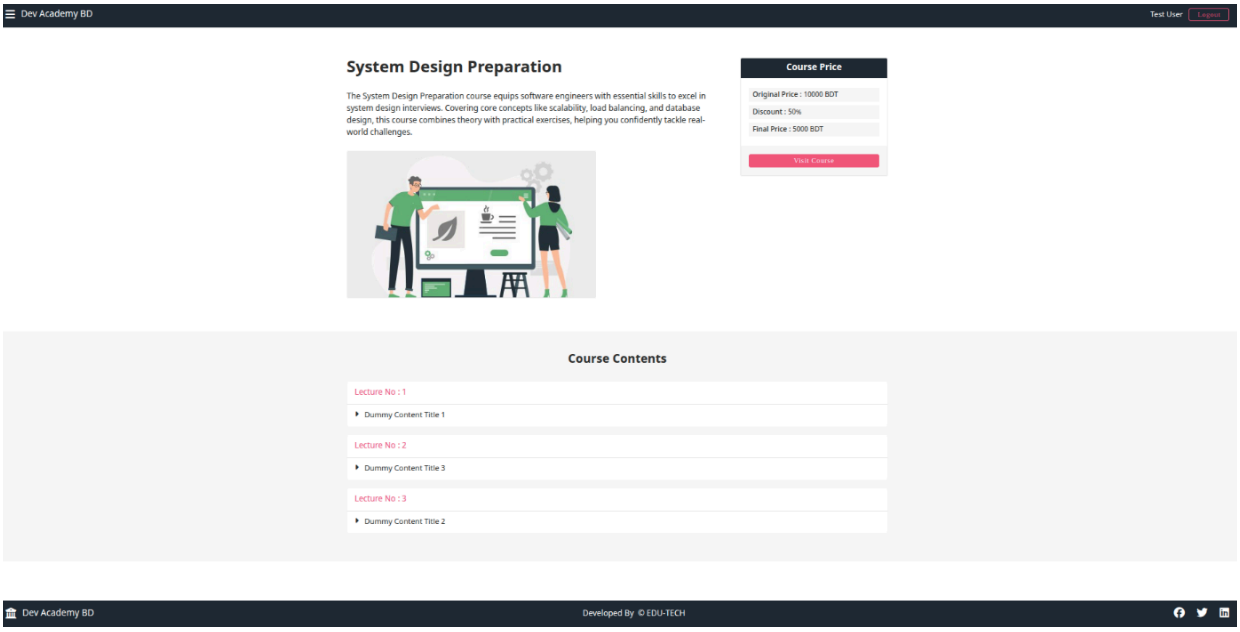

Figure 3.16 Student Interface

In figure 3.16 we can see the view of course when a user tries to see without having access. They can see the topic but cannot access the actual content. To access the content they have to first buy the course.

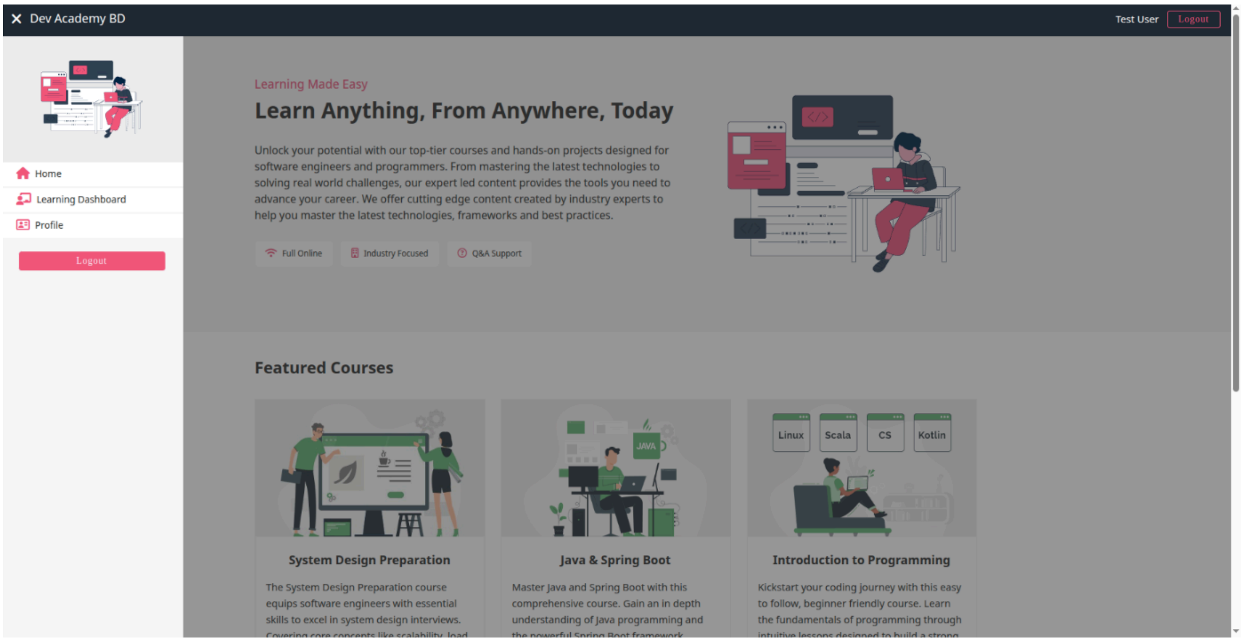

Figure 3.17 Student Interface

In fig 3.17 we can see the side panel a user can see after login. Through the side panel they can access their profile, learning dashboard and go to the home page. And also they can log out through the side panel.

3.2 Detailed Methodology and Design

When creating our Ed-Tech platform, we made sure that it was useful and flexible enough to fulfill a variety of learning needs. We studied a lot of different techniques and technological innovations before settling on the final design. The process of choice was based on three important factors: usability, scalability, and integration simplicity.

We first thought about creating a complete application in PHP or conventional Java. Smaller projects might find this technique easier, but growing and maintaining the platform will become more difficult as it grows. Also, a monolithic structure increases the likelihood of downtime because the entire system may fail if one component fails. Also, adding new features like AI or mobile support may entail considerable software changes.

We looked into combining popular content management tools (CMS) with LMS plugins, including These systems lack the ability to combine real-time video communication and unique AI capabilities, which may have lowered development time. Furthermore, in scenarios with a large user base and a focus on training, their efficacy and security are limited by a lack of extensive customization.

In addition, we investigated the usage of frameworks like Django (Python) and Laravel. Both are dependable and suitable for internet use. However, because our development team was experienced in Java, we chose to use the Spring Boot framework, which has crucial features for developing scalable web applications such as integrated security, data processing, and REST API compatibility.

After important discussion, we chose a microservice-based design with Spring Boot for the backend and Angular for the frontend. This method allows us to break down the system into smaller, distinct services such as testing, AI integration, course management, and user administration, as each component can be designed, tested, and updated separately. As a result, the platform's expansion and maintenance are made easier.

Angular was chosen because it can be used to develop responsive single-page applications (SPAs), which improve the user experience significantly. It allows better interactivity, faster load times, and dynamic page adjustments, all of which are very important for dashboard features and live training.

We select Spring Data JPA for seamless database connection, Apache Tomcat for dependable web server, and Spring Security for strong permissions and authentication. To ensure an easy and clear user interface we opt for Font Awesome and SASS for stylistic and UI enhancement.

Our architecture enables us to meet demands of modern online learning such as live classrooms, real-time chat, AI-driven assistance and the raft of assessment tools. All these features have dynamic and secure infrastructure behind them. More critically, the system is conceived to scale up hence enabling easy growth once more users join in and the institutions augment their digital learning echelon.

.

3.3 Project Plan

To make our EdTech platform a reality we arranged the work on the project into clear phases, with the objectives, the timelines, and the output specified for each phase. This approach helped us to stay on the same line, to assign resources smoothly and track development progress within the development process. The first phase covered the first two to four weeks, and was designed for research and preparation. We reviewed existing platforms, while doing background and field study research to get feedback from the users. This information allowed us to sketch the functional scope of the platform and to identify key problems to solve.

We then moved to the prototyping and design and this took approximately 8-10 weeks. At this time, we used tools such as Figma to make wireframes and UI mockups as well as system architecture diagrams. Important technology decisions were taken with Spring Boot being selected for backend and Angular used for front-end. Development was broken into two stages: frontend and backend. The frontend team used angular, SASS and TypeScript to build responsive dashboards and user interfaces took between 8 to 12 weeks. At the same time, the backend team was working on the development of RESTful APIs and defining the database schema as well as a field such as user authentication, course management, etc., with Spring Boot and related technologies during a 10-14 week period. After development was complete, the platform went for four to six weeks of testing and quality assurance phase. This was both manual and automated, with tools such as Selenium used in automation. Internal testing cycles and user input were indispensable for the platform improvement. The last steps of the project have been deployment, 2-4 weeks and post-launch monitoring. During this time table, the platform ran on a test server to test its real world performance. User and administrator documentation was developed and analytics at a rudimentary level were installed for the purpose of monitoring the performance and usage of the platform. Regular weekly progress assessments kept the team on their feet, allowing timely project completion and good groundwork for future expansion and functionality.

3.4 Task Allocation

The project was conducted as a collaborative venture by two members of the team but with specialization in varying areas. Such allocation of functions helped us with managing all aspects of development, starting from the user interface up to back-end logic and testing. One of the team members was also responsible for the backend development, which involved interfacing with the database by invoking Spring Database Spring Data JPA, creating user authentication and authorization with Spring Security, and organizing projects with Spring Boot. This individual managed major functionalities such as batch creation, integration with AI, and user registration, course management, all done via RESTful APIs. They also monitored the backend deployment on the server, to be sure that the core logic and the data process went smoothly. The other members of the team worked on frontend development, manual testing and documentation. Using Angular, and SASS, they proceeded to build the intuitive and responsive interface which consisted of such components as dashboards, course view, and login & registration pages. This individual also did manual testing to ensure that the platform worked fine and detected and fixed problems that presented themselves. Besides, they were responsible for preparing technical documentation and preparing the final project report and submission materials. Team members worked together hand in hand to complete the project; sharing responsibilities in planning, requirement gathering, and final presentation of the project. With good communication and all working together we managed to unite a well-built backend with a user-friendly interface and managed to complete the platform on time.

3.5 Summary

This chapter explains the detailed process of building our EdTech platform. We began by identifying key functional and non-functional requirements, gathering insights from users, and setting clear expectations. After evaluating various options, we chose a microservice-based architecture with Angular for the frontend and Spring Boot for the backend, prioritizing scalability, flexibility, and ease of maintenance. We then mapped out the system's architecture, selecting appropriate development tools for both the frontend and backend, and designed the user experience. A comprehensive project plan was created, with well-defined tasks for each team member and a clear breakdown of the development stages. By following this organized and collaborative approach, we successfully developed a modern, efficient, and interactive online learning platform that meets the demands of the evolving digital education landscape.

Chapter 4

Implementation and Results

4.1 Environment Setup

We created a well-equipped development environment to effectively build our Ed-Tech platform, leveraging cutting-edge tools and technologies to provide seamless front-end and back-end development, testing, deployment, and collaboration. The environment was also intended to be adaptable, allowing for future changes as the platform progressed. To ensure scalability, security, and flexibility, we carefully chose modern technologies that would function reliably in both development and production contexts. This configuration contributed to a smooth development process and reliable platform functionality.

For the backend, we used Java 17 as the major programming language. Its long-term support, increased security features, expanded language possibilities, and superior performance over previous versions made it the best choice. The backend services were created using the Spring Boot framework, which ensures complete compatibility and robustness.

PostgreSQL was chosen as the relational database management system because of its dependability, data integrity, and ability to efficiently perform complicated queries. As a Spring Data JPA-compliant system, we were able to connect with the database via Java objects, removing the requirement for raw SQL code.

To maintain consistency across many environments, we organized the application using Docker containers. The Docker Engine was used to deploy different instances for the database, frontend, backend, and supporting services. Docker Compose made it easier to handle multi-container systems by needing only one configuration file, allowing the services (database, frontend, backend, and Keycloak) to start and work together.

We managed user authentication and identity management using Keycloak, an open-source identity and access management tool. With little setup, Keycloak enabled us to handle token-based authentication, role-based access control, login, and registration. It was set up as a Docker container and connected to our Spring Boot application via OAuth 2.0 protocols.

The majority of the development work was done using IntelliJ IDEA, which provided substantial support for Spring Boot, database management, and project structure visualization. We picked Visual Studio Code for frontend development because of its lightweight editing features and good support for Angular, TypeScript, and SASS. Postman was used to test APIs. while Git was used to manage versions.

In summary, our development environment included:

- Java 17 – for backend development
- Spring Boot – main backend framework

- PostgreSQL – relational database
- Docker Engine – container management
- Docker Compose – for service orchestration
- Keycloak – user authentication and identity management
- IntelliJ IDEA Ultimate – backend IDE
- Visual Studio Code – frontend editor
- Postman – for API testing
- Git & GitHub – version control and collaboration

This environment setup provided a strong and flexible foundation for implementing a secure, scalable, and maintainable Ed-Tech platform.

4.2 Testing and Evaluation

Because the Ed-Tech platform was developed and tested locally, the assessment focused mostly on functional testing to ensure that major features worked as intended. We ensured that the user interface, navigation flow, and major platform features were properly tested in the local setup, even if full-scale deployment and API-level testing were not part of the current phase.

Functional testing involved ensuring that users could register, log in, navigate their dashboards, examine batch information, and access course materials. To ensure that everything worked as expected, admin-level operations such as content uploading, batch creation, and user management were tested. We also confirmed that role-based access (admin vs. user) and authentication functioned well by testing the login flow integration with Keycloak.

We did manual testing using a variety of user scenarios to simulate typical usage. These tests ensured that all elements, including form validations, error messages, and page navigation, worked as expected. Performance tests, such as load testing and concurrent user access, were not performed at this time since the system had not yet been deployed to a live server.

Despite being envisioned as a key advance, the AI assistant function has yet to be implemented. Its functionality will be evaluated separately upon implementation and will be incorporated into the next stage of development. In future editions, we want to connect the AI module via an API, allowing it to give real-time learning assistance, suggestions, and rapid responses to commonly requested queries.

Even without full deployment, the platform already offers some advantages over current solutions in terms of comparative analysis. Our platform provides a more comprehensive experience than Google Classroom or Udemy by including features such as batch-based content management, real-time communication capabilities, and a framework designed to foster AI-driven learning, all of which are lacking in many existing systems.

Although local testing has generated positive results, the platform is still in its early functioning stages. It validates the design and execution strategy's success and provides a solid foundation for future production, deployment, and inclusion of more advanced features.

4.3 Results and Discussion

Our Ed-Tech platform development resulted in a helpful and straightforward web application that adequately depicts the key components of a modern online learning environment. Despite the fact that the project had not yet been uploaded to a public server or had the AI assistant integration, we were able to complete a number of major milestones that validated our design and development methods.

One of the most significant benefits is the effective implementation of a role-based system that allows administrators and users (teachers and students) to access various functions based on their permissions. Individuals may log in, explore their courses, access resources, and interact with their batch content, while administrators can create and manage batches, assign users, and upload course materials. These features were personally tested, and their functionality in the local environment was confirmed.

Another notable success is the use of Keycloak for authentication. With proper session management and role-based access control, we were able to implement a secure login system. This link shows that, once implemented, the platform can securely scale and manage genuine user accounts in a production environment.

Using Angular and SASS, we were able to design a responsive and clean front-end experience. The interface, navigation, and layout aspects were designed to be basic, user-friendly, and easy to use. Our goal of developing an accessible platform for a varied user base is backed by the simplicity with which users may access all of the primary features.

Nonetheless, a number of critical system components are still in development or are planned for integration in the future. The AI assistant, one of the platform's most imaginative features, has yet to be implemented. This makes it more difficult for us to evaluate how well the platform promotes personalized, real-time learning experiences. Furthermore, we were unable to conduct real-world performance testing, including load testing with many users, because the project was not hosted on a cloud server.

With these shortcomings, the project's overall outcome is positive. The platform's architecture was designed to accommodate future expansion and scalability. Because of its adaptability and flexibility, new features may be easily added and upgraded. Based on the results of local functional testing, the platform performs well in its current configuration and has a high potential for reliable growth and deployment.

This project shows the need of careful design and current technologies such as Docker, Keycloak, Angular, and Spring Boot. By utilizing these resources, we have laid the groundwork for an educational platform capable of providing creative, dynamic, and secure online learning experiences to both individual learners and educational institutions.

4.4 Summary

This chapter described the technical configuration, installation procedure, and testing findings for our Ed-Tech platform. The platform was successfully constructed and tested locally, and it includes crucial features such as user role management, batch processing, Keycloak secure login, and an Angular-powered dynamic user interface. While complete deployment and testing are still in the works, API-level and manual functional testing have validated that the basic functionalities work as planned. The AI helper is envisaged for future integration. Overall, the results indicate that the project is moving forward well, with a solid basis for future expansion, scalability, and deployment.

Chapter 5

Engineering Standards and Design Challenges

5.1 Compliance with the Standards

When developing our Ed-Tech platform, we made sure to follow a number of important software engineering standards and best practices to ensure the system's scalability, maintainability, security, and usability. The rules we followed addressed themes such as software development practices, data security, system architecture, and code quality. The major requirements for our project are stated here, along with prospective substitutions and an explanation of our decision-making process.

5.1.1.1 RESTful API specification (OpenAPI).

We built the backend services using the principles of RESTful API design. These ensure consistent, stateless, and easily maintainable communication between the frontend and backend. By allowing clients to get just the information they need, GraphQL may have reduced the number of requests made to the server. However, GraphQL complicates things and requires more setup and instruction for testers and developers.

- Chosen: RESTful APIs (OpenAPI-compatible)
- Pros: Simplicity, wide community support, easier debugging and documentation
- Cons: Sometimes over-fetches or under-fetches data
- Rationale: Given our team's familiarity and the nature of our platform, REST APIs provided a better balance between ease of use and functionality.

5.1.1.2 OpenID Connect/OAuth 2.0 authentication and authorization standards

We implemented Keycloak, which secures user authentication and permission by conforming to OAuth 2.0 and OpenID Connect specifications. Using third-party services like Firebase Authentication or custom JWT-based login logic would have been an option. Despite its rapid deployment, Firebase is better suited for commercial or mobile-first applications because it offers less control over enterprise-level identity management and user responsibilities.

- Chosen: Keycloak with OAuth 2.0 and OpenID Connect
- Pros: Secure, customizable, supports single sign-on (SSO), scalable for large systems
- Cons: Initial configuration can be complex

- Rationale: Keycloak aligns with our security goals and integrates well with Spring Boot and Docker.

5.1.1.3 Microservices Architecture, a standard in software architecture

As a design standard, we adopted Microservices Architecture. A monolithic design, in which the entire application works as a single unit, is an alternative approach. Monolithic applications are easier to develop initially, but as the system grows, they become more complex to scale and administer.

- Chosen: Microservices
- Pros: Scalability, isolation of components, easier to update and deploy features independently
- Cons: More complex to manage and deploy initially
- Rationale: Our platform is built for growth and future integration of modules like AI, so modular microservices were the best long-term choice.

5.1.1.4 Data Storage Standard (PostgreSQL ACID Compliance)

We picked PostgreSQL because it follows the ACID (Atomicity, Consistency, Isolation, and Durability) principles, which provide accurate and reliable data handling. We also investigated NoSQL databases, such as MongoDB, which provide faster performance for unstructured data and a flexible schema. They are unsuitable for complicated relational data such as user roles, courses, and batches.

- Chosen: PostgreSQL (Relational, ACID-compliant)
- Pros: Strong consistency, ideal for transactional systems
- Cons: Slightly less flexible compared to NoSQL for evolving data structures
- Rationale: Our platform heavily depends on structured and relational data, making PostgreSQL a more reliable choice.

5.1.1.5 Frontend Development Guidelines (Responsive Web and SPA Design)

In order to support multiple screen sizes, our frontend employs Angular in compliance with the Single Page Application (SPA) paradigm and Responsive Web Design (RWD) requirements. While multi-page applications (MPA) may be more SEO-friendly, SPAs provide a smoother, app-like user experience.

- Chosen: SPA with Angular and Responsive Design
- Pros: Faster navigation, seamless user experience, better performance
- Cons: SEO limitations and initial loading time may be higher
- Rationale: The dynamic nature of the platform, with dashboards and real-time interactions, made SPA the ideal choice.

5.1.2 Software Standards

Throughout the development process, we used a set of acknowledged software engineering standards and best practices to ensure the quality, maintainability, and reliability of our EdTech platform. These requirements regulated all aspects of our code authoring, version control, security, and project quality.

5.1.2.1 Code Quality Standards (Robert Martin's Clean Code Principles)

We followed clean coding rules, which included single-responsibility functions, modular techniques, comprehensible naming conventions, and minimal duplication. These recommendations help other developers understand and change the code.

- Tools used include ESLint (for Angular), code formatting tools, and IntelliJ IDEA's integrated code inspection.
- Benefit: Streamlines debugging and cooperation while improving long-term maintainability.

5.1.2.2 GitHub and Git's version control standards

We handled our code repository with GitHub and utilized Git for version control This enabled us to track changes, collaborate productively, and roll back to previous revisions as needed.

- The following methods were used: pull request reviews, commit message standards, and feature branching.
- Benefit: Enables collaborative and parallel development while keeping the codebase organized.

5.1.2.3 The Software Development Life Cycle (SDLC) Model is an iterative method inspired by Agile

Despite being a student project, we followed an Agile-inspired iterative development cycle. Each step (planning, design, development, and testing) was reviewed as needed in response to comments and issues discovered during the process.

- Benefits include more flexibility, faster issue solutions, and a speedier response to design changes.

5.1.2.4 Security Guidelines (OWASP Top 10 Knowledge)

We ensured that our application was created with awareness of the OWASP Top 10 most common online vulnerabilities, even if comprehensive security testing was not performed. We avoided common difficulties such as inadequate authentication, unsafe API access, and ineffective session handling by utilizing Keycloak, secure HTTPS endpoints (planned for deployment), and validated user inputs.

- Benefit: Builds the basis for a secure system that can be widely utilized and implemented.

5.1.2.5 SQL Database Standards Best Practices.

Our PostgreSQL queries followed standard practices, including normalization, indexing when necessary, and avoiding hard-coded queries in the codebase. We used Spring Data JPA to guarantee that the business logic and data access logic remained distinct.

- The benefits include improved performance, the elimination of data duplication, and clean, reusable code.

5.1.2.6 Documentation Standards.

Throughout the project, we kept internal records of the environment settings (Docker, Keycloak, and database), setup instructions, and development notes. We also developed external documents like user manuals and this project report.

- Tools Used: GitHub markdown files, PDF instructions, and inline code comments
- Benefit: Assists users and prospective contributors in comprehending how to efficiently operate, maintain, and use the system.
-

5.1.3 Hardware Standards

Although the majority of our Ed-Tech platform is software-based, performance and accessibility necessitated compatibility with typical hardware configurations. We followed basic hardware standards to guarantee that developers and end users could use and maintain the system successfully without the need for expensive or specialized equipment.

5.1.3.1 Compatibility of user devices

We ensured that the platform could be accessed using readily available devices. This includes:

- Desktops and laptops: Computers with a modern web browser (Chrome, Firefox, or Edge), 4GB of RAM, and a dual-core CPU (1GHz+).
- Smartphones and tablets are devices that have a CPU speed of at least 1GHz, run Android 8+ or iOS 12+, and can access the web.

By adhering to these standards, we ensured that instructors and students from diverse backgrounds could utilize the platform without requiring expensive or powerful equipment.

- An alternative would be to develop a mobile application that makes more use of the device's resources.
- Justification: A web-based responsive platform contributes to the reduction of the digital divide by facilitating increased interoperability across all devices.

5.1.3.2 Hardware Development and Testing

We used standard hardware setups for the development environment that support Angular, Docker, and Spring Boot projects:

Machines for Development:

- Minimum requirements are 8GB of RAM, SSD storage, and an Intel i5 or Ryzen 5 processor.
- 16GB of RAM is recommended for flawless Docker multi-container testing.

5.1.3.3 Machines for testing

To ensure responsiveness and usability across platforms, computers of various configurations (from basic to mid-range) were used.

- Tools used include Visual Studio Code, Postman, Docker Engine, IntelliJ IDEA Ultimate, and Git CLI.
- Virtual computers or cloud-based development environments are viable alternatives.
- Justification: Local development ensures better control, speed, and offline access. AccessCloud settings may be more costly and need stable internet connections.

5.1.3.4 Deployment Readiness and Server

Our Dockerized architecture prepares the system for future hosting on cloud platforms such as AWS, Azure, or DigitalOcean, even if it has not yet been deployed. These servers use popular hardware profiles, such as

- 4GB RAM and 2 vCPUs for light hosting.
- For production with moderate traffic, 4+ vCPU and 8GB+ RAM.

This ensures that the system will run properly in modern container-based environments with industry-standard hardware support.

5.1.4 Communication Standards

Throughout the development of our Ed-Tech project, team members, system components, and the platform need excellent communication with one another. We followed a variety of acknowledged communication standards during the project's technical and cooperation phases to ensure security, clarity, and consistency in our discussions.

5.1.4.1 Communication with the system (RESTful API Standard)

All interactions between the frontend (Angular) and backend (Spring Boot) were implemented via RESTful APIs. This standard defines structured data transmission utilizing JSON, a lightweight and widely accepted format, as well as specific HTTP methods (GET, POST, PUT, and DELETE).

- Alternatives: SOAP protocol or GraphQL

- Chosen: RESTful API with JSON.
- Pros: Simple, easy to implement, and supported by both Spring Boot and Angular.
- Cons: It may be less flexible than GraphQL in certain sophisticated queries.
- REST offered the ideal blend of simplicity and performance for our system's requirements.

5.1.4.2 Communication for Authentication: OpenID Connect/OAuth 2.0

For user authentication and authorization, we used Keycloak, which supports OAuth 2.0 and OpenID Connect. These protocols provide secure token-based communication between the user, the authentication server, and the principal application.

- Firebase Auth and a bespoke JWT system were discussed as options.
- OAuth 2.0 with Keycloak was chosen.
- Advantages: Secure, versatile, and compatible with role-based access and single sign-on (SSO).
- Cons: Additional setup and comprehension are required.
- Justification: The standards-based connection between Keycloak and Spring Security ensured robust and scalable identity management.

5.1.4.3 Collaboration within the Team and Project Communication

We used both informal and tool-based communication standards for work tracking and team communication.

- Code Collaboration and Version Control: GitHub and Git.
- The GitHub flow principles were followed for naming branches, pull requests, and commit statements.
- Trello, Messenger, and Google Docs for Planning and Discussions
- The duties were properly assigned and organized into checklists.
- The documentation offers PDF reports, inline comments, and Markdown (for setup and readme).

By adhering to these standards, the two team members were able to stay on the same page throughout the development process and contribute constructively without any confusion or code disagreements.

5.1.4.4 User-Platform Interaction (UI Communication Patterns)

In-platform communication that followed strict UI/UX principles included messages, discussion threads, and batch-level notifications.

- Feedback and layout are consistent.
- verification of forms and status updates
- Breadcrumbs for navigation and icons on buttons

Users were able to interact with the system with confidence and ease because of the internal communication design.

5.2 Impact on Society, Environment and Sustainability

In addition to its educational impact, our Ed-Tech platform aims to generate good social, environmental, and long-term sustainability. While essentially a digital tool, it has the potential for real-world impact.

The platform improves educational flexibility, inclusion, and accessibility. By reducing financial and geographic constraints, it provides students, particularly those in rural or underserved regions, with access to high-quality education from anywhere with an internet connection. It also opens up professional growth, skill acquisition, and lifelong learning to a larger audience, including working professionals. Furthermore, the platform promotes instructor-student engagement, boosting creativity, digital literacy, and information sharing. The proposed incorporation of AI assistants will help learners with impairments or those experiencing learning difficulties by giving rapid assistance and individualized feedback.

Environmentally, the platform encourages digital learning, reducing the need for paper, transit, and physical resources. This lowers the need to commute and print course materials, resulting in less waste and carbon emissions. Furthermore, the system's architecture, which incorporates containerized technologies such as Docker, encourages efficient server utilization, leading to a reduction in hardware resource consumption and supporting sustainability objectives. This is consistent with current advances in cloud-based deployment and green computing, which lessen the environmental effect of software systems.

The platform is designed for long-term use, scalability, and maintainability in order to be sustainable. It is adaptable to future educational demands and technologies due to its microservice design, which ensures that new features may be incorporated without hurting the system as a whole. Open-source technologies like as Docker, Angular, Keycloak, and Spring Boot promote cost-effective development and reuse, which benefits startups and small institutions aiming to expand digital education at a reasonable cost. Because the system is built on standards-based technology, future developers will find it easier to maintain, modify, or expand the platform without having to start over.

Our Ed-Tech platform promotes UN Sustainable Development Goal 4 (Quality Education) and contributes to the creation of a more fair and environmentally conscious society via the use of online learning and sustainable technology practices.

5.2.1 Impact on Life

The development and launch of our Ed-Tech platform had a huge influence on our academic, professional, and personal lives. As students, completing this project allowed us to apply the theoretical information we had learned in class to a real-world system that may potentially help others. It improved our knowledge of full-stack web development, project management, software architecture, and cooperation.

Throughout the project, we learnt how to use current tools such as Docker, Keycloak, Angular, and Spring Boot. Our hands-on experience with these technologies has prepared us for future roles in software engineering and development, both of which are highly valued in the industry. Furthermore, our soft skills—such as problem-solving, communication, and time management—were enhanced, particularly when we collaborated to solve difficulties and discover answers. Aside from the technological achievements, this initiative has altered our perception of technology and its role in education. It made us more aware of how digital platforms might improve the learning experience, particularly for those struggling in traditional educational environments. As aspiring engineers, we now understand how we can help to create scalable, inclusive, and socially meaningful solutions.

Finally, this project was not just a technical triumph, but also a transforming experience that helped us improve our talents, extend our perspectives, and shape our long-term professional goals.

5.2.2 Impact on Society & Environment

Our Ed-Tech platform aims to have a good and practical influence on both society and the environment. It contributes to the removal of social obstacles to education by offering an adaptable online learning environment. Students from rural, underprivileged, and diverse places may now access high-quality educational resources without having to physically visit regular schools. This encourages digital inclusion and ensures equitable educational opportunities for everybody. Furthermore, the platform promotes upskilling and lifetime learning, allowing people of all ages to remain competitive in a quickly changing environment. One of the proposed future additions is AI-powered learning support, which will aid students who require extra assistance or have specific requirements, hence enhancing educational inclusion and customisation. The platform also promotes ecologically responsible behaviors by lowering the use of paper, printed materials, and physical textbooks. Online learning reduces commuting, saving CO_2 emissions and fuel use. Furthermore, our use of containerization technology (via Docker) improves server resource consumption, allowing for more energy-efficient deployments. By merging digital education, social inclusion, and environmental sustainability, our platform helps to create a more equitable, informed, and sustainable future.

5.2.3 Ethical Aspects

Our Ed-Tech platform's development incorporated ethical considerations as a critical component. We prioritized user privacy and data protection because the platform stores personal information such as names, emails, and learning histories. To ensure that only authorized users could view or handle critical data, we established role-based access and secure login with Keycloak. Another focus was on ensuring equitable educational opportunities. Students from low-income or rural areas may still benefit from online learning without the need for expensive tools because the platform is designed to work on simple devices and sluggish internet rates. We are well aware of the issues of bias in automated learning systems, even if the AI assistant has not yet been implemented. In the future, we want to make sure that our AI helper treats all students equitably, regardless of their background, learning style, or ability. This dedication to diversity will contribute to a helpful and neutral educational environment. To ensure the integrity of course materials, upload access will be limited to authorized individuals such as administrators and instructors. Future versions of the platform will also feature moderation and reporting mechanisms to help prevent the abuse or spread of improper information. Throughout the creation process, we focused on academic integrity and ethical responsibilities. All research sources were correctly mentioned, and any third-party libraries or tools utilized in the project were implemented with the necessary legal licenses. Our objective has been to build a platform that is not only functional and effective but also aligned with ethical standards and responsible digital practices.

5.2.4 Sustainability Plan

Our Ed-Tech platform was thoughtfully designed with long-term sustainability in mind—both from a technological and societal perspective. The aim was to create a system that can evolve over time, scale efficiently, and continue serving learners effectively as demands change. From a technical standpoint, we implemented scalable tools and frameworks such as Docker, Angular, and Spring Boot. These choices make it possible to extend or upgrade the system without requiring a full overhaul. By adopting a microservice-based structure, we ensure that individual components can be modified or improved independently, helping to reduce future maintenance costs. The system is also optimized for efficiency, consuming minimal computing resources. Its lightweight design allows it to operate smoothly on basic hardware and makes it suitable for deployment on cost-effective cloud platforms or local servers. In terms of content and user flexibility, the platform supports institutions in managing and updating their own educational materials, fostering long-term adaptability. It can accommodate various teaching methods, languages, and subject areas, ensuring relevance across different educational contexts. Financially, the use of open-source technologies minimizes licensing expenses, making the platform affordable for schools and organizations. In the future, sustainable funding can be supported through subscription models, course-based payments, or partnerships with academic institutions. By combining a scalable software foundation with efficient resource use and adaptable content delivery, our platform is built not only for current needs but also for a lasting impact in digital education.

5.3 Project Management and Financial Analysis

Building and maintaining a comprehensive EdTech platform requires strategic resource planning to cover all key areas, including infrastructure, development tools, personnel, and future upgrades. To ensure both feasibility and sustainability, we developed a detailed cost projection based on current industry standards and practical needs. The budget is organized into six core categories: platform development, educational content production, AI integration, server and API hosting, personnel expenses, and ongoing system maintenance. Our primary objective was to support scalability and long-term functionality while minimizing total expenditures wherever possible. The financial analysis for this project is displayed in the table below 2.2.

Table 2.2 Financial Analysis

SL No.	Expense Description	Amount (BDT)	Description
01	Domain & SSL	5,000	Purchase of custom domain, SSL certificate (1 year)
02	Hosting & Cloud Infrastructure	80,000	VPS or cloud hosting (AWS/scalable 1-year plan
03	AI Assistant API Integration	70,000	Estimated cost for GPT-based API usage (OpenAI/others)
04	Live Conference API	65,000	Zoom/Agora/Twilio API usage for live classes
05	Course Video Production	40,000	Equipment, editing, and studio rental (basic setup)
06	Developer Salaries (2 members, 6 months)	300,000	Backend + Frontend (Part-time/freelance model)
07	QA & Manual Testing	30,000	Manual testers & feedback collection (user testing phase)
08	UI/UX and Branding	25,000	Design assets, logo, UI wireframes
	Maintenance & Support (1st Year)	40,000	Server updates, bug fixes, system monitoring
	Marketing & Outreach	25,000	Initial online promotions, email campaigns, demo sessions
09	Miscellaneous	10,000	Licenses, plugins, additional third-party tools

10	Contingency (10% of total)	70,000	Buffer for unexpected expenses
	Total	800,000	

5.4 Complex Engineering Problem

Developing our Ed-Tech platform involved overcoming a number of technical challenges. Our goal was to build a secure, scalable, and flexible system that could serve multiple user roles while remaining user-friendly. The complexity lay in unifying different components—such as user dashboards, batch and course management, secure authentication, and provisions for future features like AI tools and live virtual classrooms—under a single, expandable architecture. The platform currently supports two primary user types: administrators and students, each with their own access rights and workflows. Implementing role-based access control using Keycloak, integrated with Spring Boot and Angular, required a solid grasp of session security, token handling, and authentication protocols. Striking the right balance between ease of use and strict data security standards was one of the most demanding aspects of the build.

Another key priority was designing the system to be future-proof. While features like code compilers, automated testing modules, AI-driven support, and even virtual reality-based learning are still on the roadmap, we made sure the underlying architecture could support them when the time comes. To do this, we followed a modular design approach and relied on Docker-based containerization, making it easier to scale and update components without disrupting the core system.

Additionally, the project required rigorous database design to ensure that course, user, and batch data were relational, normalized, and queryable. Working in a local context with limited resources, we needed to maintain responsiveness, speed, and dependability throughout the system's backend, frontend, and data storage layers. Our ability to solve a complex technological problem is evidenced by the functional and expandable prototype we created despite the limits.Through research, cooperation, and continuous learning, we were able to create a working system that satisfies current educational expectations and is ready for further improvement.

5.4.1 Complex Problem Solving

Our Ed-Tech platform addresses a real-world technological challenge: designing and implementing a scalable, secure, and user-focused educational system. In addition to supporting many user roles, data flows, and third-party connections, the system had to be safe, maintainable, and accessible. As seen in the map below (5.1), the problem we addressed closely mimics multiple components of complex engineering issues.

Table 5.1: Mapping with complex problem solving.

EP1 Dept of Knowledge EP2 Range Of Conflicting Requirements	EP1 Dept of Knowledge EP2 Range Of Conflicting Requirements	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Extent of Applicable Codes	EP6 Extent Of Stake-holder Involvement	EP7 Interdependence
✔	✔	✔	✔	✔	✔	

EP 1: Knowledge Depth.

This project required multi-domain experience in front-end (Angular, UI frameworks), back-end (Spring Boot, security, databases), system architecture (microservices, Docker), and authentication protocols (OpenID Connect with Keycloak, OAuth 2.0). The system's layers must operate fluidly while being flexible and secure. To prepare the system for future integration of AI-based APIs, we needed to understand the use cases, performance implications, and data privacy concerns associated with natural language processing.

EP2: Diverse Contradictory Needs.

We encountered conflicting requirements for cost and functionality, real-time interactivity and modular development, user experience and security, and scalability and performance. For example, while Zoom and Agora provided amazing live conferencing features, they also came with additional charges and difficult APIs. It was required to create a balance between long-term maintenance (microservice compatibility, containerization) and real-time support (live classes, AI).

Episode 3: Analytical Depth

To identify feature gaps and determine what our system needs to improve, we did a detailed analysis of the current platforms (Google Classroom, BLC, and Udemy). We compared PostgreSQL to NoSQL competitors like MongoDB and investigated authentication techniques such as Keycloak, Firebase Auth, and custom JWT solutions. To ensure viability and user effect, performance, usability, and development costs were considered at all levels.

EP4: Issue Familiarity.

The project had several innovative components, such as Keycloak integration, Docker Compose orchestration, real-time video API configuration, and OAuth 2.0 standards, even if certain backend development ideas were well-known. We were needed to do self-directed learning, trial-and-error experimentation, and resolve incompatibilities across several software versions and platforms.

EP 5: Scope of Relevant Codes

The system meets a variety of international standards, which include:

- OpenID Connect and OAuth 2.0 (authentication).
- Architecture of a RESTful API (OpenAPI Structure),
- principles of database design (ACID compliance).
- SPA rules for frontend, and
- Understanding of the top ten internet security risks defined by OWASP.

This dedication improves platform security, dependability, and long-term compliance.

EP6: Level of Involvement of Stakeholders

This initiative included three major stakeholder groups: students, educators, and administrators. The UI, backend permissions, and dashboard architecture were built to meet the specific needs of each group, such as resource access, communication tools, and content management. In order to better understand usability needs, we collected feedback from peers and teachers.

Mapping with Knowledge Profile for EP1

This table 5.2 is designed to map the EP1 to the Knowledge Profile.

Table 5.2: Mapping with knowledge Profile.

K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practice	K8 Research Literature
✔	✔	✔		✔

K3: Foundations of Engineering

To develop the framework's core, we used our basic knowledge of software design, object-oriented programming (Java), relational databases (PostgreSQL), and client-server communication. The project was built on features like session management, REST principles, and the MVC architecture.

K4: Specialist Knowledge

Complex technologies and frameworks, including Docker, Keycloak, Angular, OAuth, and Spring Boot, must have advanced technical knowledge. Understanding their setup, compatibility, and seamless integration was vital to finishing the project.

K5: Engineering Design

The project included dashboards, user journeys, data flows, role-based access logic, and modular backend services. We developed database schemas, wireframes, and well-organized system architecture diagrams in accordance with current architectural standards. We used practical software engineering processes such as project planning, issue tracking, manual testing, local development environments (Docker), and version control (Git/GitHub). These activities created an environment helpful to professional growth

K8: Research Literature

We reviewed academic research, case studies, and documentation to identify platform gaps and select the best tools. These included assessments on EdTech best practices and cloud software engineering, as well as evaluations of user experiences and performance.

5.4.2 Engineering Activities

Our Ed-Tech platform was created utilizing a variety of engineering activity levels, which correlate to the components of complex engineering procedures as shown in the table below (5.3). Our project is linked to each relevant engineering work in the accompanying table, which is followed by detailed arguments.

Table 5.3: Mapping with complex engineering activities.

EA1 Range of re- sources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
✔	✔		✔	✔

5.4.2.1 Justifications for Each Mapping

EA1 – Range of Resources

The project utilized a diverse spectrum of technology and human resources. These included:

- Git, Docker, PostgreSQL, Angular, Keycloak, and Spring Boot are some examples of software technologies.
- Infrastructure includes PostgreSQL for the database, Docker Engine and Docker Compose for containerization, and cloud hosting for deployment.
- Human Resources: Two developers were assigned to test, document, and create the frontend and backend. In addition, we used instructor feedback to confirm the design.

The wide range of equipment and skills required to construct this system demonstrates that the project had a substantial and well-managed resource base.

EA2 – Level of Interaction

Our platform enables detailed communications between:

- System components: RESTful APIs are used to communicate between the frontend and backend. While forthcoming AI and video APIs will manage intelligent and real-time communication, Keycloak will handle user roles and authentication.
- Using role-specific features and customizable dashboards, administrators, teachers, and students interact with the system.
- Future modules: To make interactions even more multi-layered, the architecture is designed to support the addition of new services (such as AI, analytics, and mobile apps).

This level of technical and user interaction necessitates strong system coordination and coordinated engineering decisions..

EA4: Consequences for Society and Environment

The platform greatly benefits both the environment and society:

- Accessibility: It provides high-quality online education to students from diverse socioeconomic and geographic backgrounds.
- Inclusivity: The system is designed to accommodate all sorts of learners, and future AI capabilities may give tailored support to people with learning disabilities or issues.
- Environmental Benefit: Encourages a greener learning method by reducing travel, paper use, and the need for classroom facilities.

The platform contributes to global educational sustainability goals by improving access to learning and reducing dependency on physical infrastructure.

EA5: Familiarity

While many components of the project, such as web frontend development, database management, and basic API development, were well-known, others brought new and tough challenges:

- For the first time, I am using Keycloak to authenticate OAuth 2.0 and OpenID Connect requests.
- Dockerized environments are being used for both frontend and backend orchestration.
- Understanding ML/NLP models built on APIs is required to prepare the system for AI integration.

These additional areas necessitated independent research, analysis, and trial-and-error, resulting in the project being classified as a complex engineering undertaking.

5.5 Summary

This chapter investigated the technical standards and design challenges encountered when developing our Ed-Tech platform. Our solution follows basic requirements for database design (ACID compliance), coding techniques (clean code principles), software architecture (RESTful APIs, OAuth 2.0), and authentication (OpenID Connect via Keycloak). To improve platform usability and reliability, we verified compatibility with commonly used hardware, software, and communication protocols. Addressing complex design challenges—such as ensuring modularity, managing service dependencies, integrating future AI features, and implementing secure, role-based access—was central to the development of our Ed-Tech platform. Beyond the technical aspects, we also considered the broader social and environmental impacts of the system. Ethical concerns, especially those related to data privacy and ensuring equal access to education, were prioritized from the outset. Our approach to sustainability focused on creating a system that was both cost-effective and environmentally conscious. By relying on open-source technologies and containerization, we were able to minimize expenses and reduce our ecological footprint. The project was guided by a well-structured financial plan and grounded in practical engineering practices, aiming to strike the right balance between innovation, effectiveness, and long-term feasibility.

Chapter 6

Conclusion

6.1 Summary

The development of our Ed-Tech platform marks a significant step towards bridging the gap between traditional classroom teaching and modern digital education. Our aim was to create a system that is not only accessible and flexible but also scalable, bringing together students and instructors in a unified online environment. By integrating essential features such as batch-based course access, administrative content control, secure user authentication, and real-time discussions with advanced technologies like Spring Boot, Angular, Keycloak, and Docker, we laid the foundation for an engaging and intelligent learning experience. While we've already established a solid base, we have plans to further enhance the platform with features like AI-powered assistance and live conferencing in future versions. These additions will elevate the learning experience, making it even more interactive and personalized. Throughout the development process, we adhered to key software engineering principles, ensuring the platform's security, maintainability, and long-term viability. This project was not only a technical achievement but also an invaluable learning experience for the team. It pushed us to experiment with new tools, solve complex problems, and manage resources effectively. The platform's current capabilities already demonstrate its potential to impact both students and educators positively. Looking ahead, we envision extending the platform with mobile accessibility, deeper integration with emerging technologies, and AI-driven features. By creating a modular, role-based system focused on scalability and user experience, we've built the framework for a robust Ed-Tech platform. Moving forward, we believe with continuous effort and development, this system can evolve into a dynamic educational ecosystem, catering to diverse learning communities across different backgrounds and regions.

6.2 Limitation

Although our Ed-Tech platform clearly demonstrates the basic components of a modern learning management system, it has numerous limitations in its current iteration, all of which are attributable to budgetary and resource constraints rather than the project's technological feasibility or ambition. One of the platform's major shortcomings is that it has yet to be installed on a public server. Because all of the testing was done locally, we were unable to evaluate performance in real-world traffic circumstances. Similar to this, time constraints and the cost of premium API subscriptions kept us from integrating and testing live conferencing APIs or the AI-powered assistant. Furthermore, while our human functional testing focused on proving critical functionality, automated testing—particularly API-level or load testing—was not used. Furthermore, several advanced analytics tools and UI/UX enhancements were postponed until a later stage of development. Despite these disadvantages, our system's modular and future-ready design makes it straightforward to integrate all of these functionalities as soon as resources become available. We regard these as opportunities for future development and advancement, not as setbacks. We are certain that with more funding and assistance, these enhancements can be implemented to create a truly complete, intelligent, and scalable learning platform.

6.3 Future Work

Our Ed-Tech platform's current version provides a solid foundation for developing an intelligent, scalable, and engaging online learning environment. However, with the correct assistance, funding, and technology resources, there is plenty of space to expand and develop the platform even more. In future generations, the platform may be expanded and adjusted in a variety of interesting ways. One of the primary aims is to include a dedicated AI assistant meant to assist students with adaptive suggestions, personalized learning paths, and doubt resolution. As a consequence, intelligent help would be provided during each learning session. Virtual reality (VR) and virtual labs are also being utilized to help scientists, engineers, and physicians. These characteristics, which do not require costly physical equipment, may help students gain practical skills by emulating real-world activities and surroundings. Another objective for the future is to provide a secure exam module. Timed assessments, proctoring assistance, and auto-evaluation will be incorporated to make the platform appropriate for enterprises that administer formal examinations and certificates. With appropriate facilities, people, and financial resources, the possibilities are limitless. Because the platform is designed to be modular and adaptable, adding new features or customizing it to individual educational institutions is both practical and useful. With continued extension and support, we are certain that this program will grow into a robust, comprehensive educational environment that serves the diverse and changing requirements of students all around the worldBecause all of the testing was done locally, we were unable to evaluate performance in real-world traffic circumstances. Similar to this, time constraints and the cost of premium API subscriptions kept us from integrating and testing live conferencing APIs or the AI-powered assistant. Furthermore, while our human functional testing focused on proving critical functionality, automated testing particularly API level or load testing was not used. Furthermore, several advanced analytics tools and UI/UX enhancements were postponed until a later stage of development.

References

[1] Sun, A., & Chen, X. (2016). Online education and its effective practice: A research review. Journal of Information Technology Education: Research, 15, 157-190.

[2] Linda Harasim, Shift happens: online education as a new paradigm in learning, The Internet and Higher Education, Volume 3, Issues 1–2, 2000, 1096-7516

[3] Adarkwah, M. A. (2021). "I'm in a difficult situation": Exploring students' challenges with online remote teaching and learning during the COVID-19 pandemic. Education and Information Technologies, 26(6), 6665-6687.

[4] Elshami, W., Abuzaid, A., Abu এলেinen, M., & Farha, R. A. (2022). Students' and faculty's perspectives on factors supporting online learning engagement in health professions education. BMC Medical Education, 22(1), 489.

[5] Bakhmat, M., Antoniuk, V., Bakhmat, O., Tymchyk, M., & Voitenko, N. (2023). Challenges and prospects of online education in the context of barrier-free access. Educational Challenges, 28(1), 80-91.

[6] Cleveland-Innes, M., Stenbom, S., & Garrison, D. R. (2024). The Community of Inquiry framework: Applications and future directions in online and blended learning. Open Learning: The Journal of Open, Distance and e-Learning, 39(1), 1-17.

[7] Frontiers in Psychology. (2025). What are the influencing factors of online learning engagement? A systematic literature review.

[8] Hoi, S. C. B., & Le Hang, V. T. (2021). Exploring factors influencing student satisfaction in online learning during the COVID-19 pandemic: A mixed-methods study. Education Sciences, 11(11), 701.

[9] Frontiers in Education. (2024). Perspectives of teachers and students on the impact of online classrooms during the COVID-19 pandemic.

[10] Russ, S., & Hamidi, F. (2021). Online Learning Accessibility during the COVID-19 Pandemic. In Proceedings of the 18th International Web for All Conference (W4A'21) 1 (pp. 1-4). ACM.

[11] Anderson, T. (2008). The Theory and Practice of Online Learning (2nd ed.). Athabasca University Press.

[12] Bates, A. W. (2015). Teaching in a Digital Age: Guidelines for Designing Teaching and Learning. Tony Bates Associates Ltd.

[13] Moore, M. G., Dickson-Deane, C., & Galyen, K. (2011). e-Learning, online learning, and distance learning environments: Are they the same? The Internet and Higher Education, 14(2), 129-135.

[14]Hew, K. F., & Cheung, W. S. (2014). Students' and instructors' use of technology affordances in blended learning: The interplay of task-technology fit and individual

characteristics. British Journal of Educational Technology, 45(5), 833-844.

[15] Siemens, G. (2005). Connectivism: A Learning Theory for the Digital Age. International Journal of Instructional Technology and Distance Learning.

[16] Al-Fraihat, D., Joy, M., & Sinclair, J. (2020). Evaluating E-learning systems success: An empirical study. Computers in Human Behavior, 102, 67-86.

[17] Rapanta, C., Cantatelli, M., de Freitas, S., Guàrdia, L., & Melchiorre, A. (2020). Online collaborative learning under the microscope: What interaction data tell us. Computers & Education, 143, 103665..

[18] Picciano, A. G. (2002). Beyond student perceptions: Issues of interaction, presence, and performance in an online course. Journal of Asynchronous Learning Networks, 6(1), 21–40.

[19] Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. Expert Systems with Applications, 33(1), 135–146.

[20] Lee, J., & Choi, H. (2011). A review of online course dropout research: Implications for practice and future research. Educational Technology Research and Development, 59(5), 593–61

Edu-Tech

ORIGINALITY REPORT

7%	5%	3%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	4%
2	Submitted to United International University Student Paper	<1%
3	e-jurnal.stiebii.ac.id Internet Source	<1%
4	conference.pixel-online.net Internet Source	<1%
5	www.ijlter.myres.net Internet Source	<1%
6	www.hindawi.com Internet Source	<1%
7	Submitted to University of Pretoria Student Paper	<1%
8	epdf.pub Internet Source	<1%
9	iopscience.iop.org Internet Source	<1%

10	link.springer.com Internet Source	<1 %
11	Submitted to Universiteit van Amsterdam Student Paper	<1 %
12	portalcris.vdu.lt Internet Source	<1 %
13	Submitted to Nottingham Trent University Student Paper	<1 %
14	open.metu.edu.tr Internet Source	<1 %
15	Lynn Senior, Neil Barnes, Joyce I-Hui Chen, Scott Clarke, Heather Sherman. "A Guide to the Diploma in Teaching and Related Qualifications - Level 5 and Beyond", Routledge, 2025 Publication	<1 %
16	oro-vip.open.ac.uk Internet Source	<1 %
17	huskiecommons.lib.niu.edu Internet Source	<1 %
18	malque.pub Internet Source	<1 %
19	Submitted to RMIT University Student Paper	<1 %

20	icatsd2022.iuh.edu.vn Internet Source	<1 %
21	www.thecommunityofinquiry.org Internet Source	<1 %
22	hdl.handle.net Internet Source	<1 %
23	123dok.com Internet Source	<1 %
24	dokumen.pub Internet Source	<1 %
25	wiredspace.wits.ac.za Internet Source	<1 %
26	www.academic-conferences.org Internet Source	<1 %
27	www.irrodl.org Internet Source	<1 %
28	www.ug.edu.gh Internet Source	<1 %
29	Akkilinc, Filiz Resat. "Online Learning and Traditional Face-to-Face Learning: A Comparison Exploring Learning and Learning Experiences Among Students in the HE of the Eastern Mediterranean.", The University of Liverpool (United Kingdom) Publication	<1 %

30	Howard Anderson, Sharon Yull. "BTEC Nationals - IT Practitioners", Routledge, 2019 Publication	<1 %
31	conference.eadtu.eu Internet Source	<1 %
32	internship.daffodilvarsity.edu.bd Internet Source	<1 %
33	sbr.journals.unisel.edu.my Internet Source	<1 %
34	www.asjp.cerist.dz Internet Source	<1 %
35	www.niti.gov.in Internet Source	<1 %
36	Alquran, Hebah. "Towards Sustainable E-Learning Platforms in the Context of Cybersecurity: A TAM-Driven Approach", Eastern Michigan University, 2023 Publication	<1 %
37	Harvey, Nghia Dinh. "Non-Traditional Students' Perceptions of the Learning Management System's (LMS) Support of Self-Regulation Skills: A Phenomenological Study.", National University Publication	<1 %

38	Michael Grahame Moore, William C. Diehl. "Handbook of Distance Education", Routledge, 2018 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		