PROJECT REPORT ON
"EDUNEST
A FULL-STACK LEARNING MANAGEMENT SYSTEM (LMS)"
COURSE CODE: CSE-436
COURSE TITLE: FINAL YEAR PROJECT

Department of Computer Science and Engineering
Metropolitan University, Sylhet
August 07, 2025

"EDUNEST: A FULL-STACK LEARNING MANAGEMENT SYSTEM (LMS)"

COURSE CODE: CSE-436

COURSE TITLE: FINAL YEAR PROJECT

Supervisor

Abu Jafar Md. Jakaria

Senior Lecturer
Department of Computer Science and Engineering
Metropolitan University, Sylhet

Project Reviewers

Md. Mushtaq Shahriyar Rafee
Assistant Professor
Department of Computer Science and Engineering
Metropolitan University, Sylhet

Golam Mostofa Naeem
Lecturer
Department of Computer Science and Engineering
Metropolitan University, Sylhet

ACKNOWLEDGEMENTS

To begin, we offer all of our thanks to the Almighty for enabling strength, wisdom, and opportunity to complete this final year project. It has been a long and arduous journey, but every moment has been fruitful learning.

We are grateful to our project supervisor, **Abu Jafar Md. Jakaria**, Senior Lecturer in the Department of CSE. His support was like a lighthouse. He illuminatingly helped us to overcome every predicament when we thought we were stuck - with advice, encouragement, and most importantly, interest in our project's progression.

We would also like to express our immense gratefulness to **Md. Mahfujul Hasan**, Associate Professor and Head of the CSE Department. His continual support and environment for students like us to complete aspirational projects like this has helped us bring this project to fruition.

We would also like to distinguish and give thanks to our project reviewers, **Md. Mushtaq Shahriyar Rafee**, Assistant Professor and **Golam Mostofa Naeem**, Lecturer. Their suggestions for our project were crucial in the decisions which we made during our reflections on the project. It was a pleasure to have them get involved with giving us a different perspective on our project through feedback.

We would like to acknowledge our friends and classmates who so kindly contributed their ideas, and acted as the initial "users" of our platform and provided us with their honest feedback.

Last, but not least, we thank our families. The love, support, and encouragement provided were the foundation of everything we accomplished along the way. Thank you.

STUDENT'S DECLARATION

We, **Al Jamil** and **Ahmad Bakhtiar Nayeem**, affirm that the undergraduate project titled "**EduNest - A Full-Stack Learning Management System**" we submitted for the consideration and fulfillment of our degree requirements by Metropolitan University, Sylhet is an authentic and original product of our work and development. We also further affirm that this project has not been submitted as a whole or part for any other qualification.

We have justified all international sources of information and material used in this project, including references, citations and other sources and material as appropriate.

We understand that the attempted to mispresent the work of any others, to be ours is serious offense and will be dealt with seriously in accordance with the policies of Metropolitan University and other applicable policies.

(Signature and Date)

Al Jamil
ID: 222-112-016
Batch: 31st (Eve)
Department of Computer Science and Engineering
Metropolitan University, Sylhet

(Signature and Date)

Ahmad Bakhtiar Nayeem
ID: 222-112-017
Batch: 31st (Eve)
Department of Computer Science and Engineering
Metropolitan University, Sylhet

CERTIFICATE OF APPROVAL

This certificate is to affirm that the undergraduate project "**EduNest—A Full-Stack Learning Management System**" has been successfully fulfilled under my supervision of this project. This project was submitted by **Al Jamil** and **Ahmad Bakhtiar Nayeem** students of Metropolitan University, Sylhet enrolled in Computer Science and Engineering program.

We confirm that, we have supervised development of this project and our satisfaction regarding the quality, originality and consistency with academic standards of the project. The project shows students' understanding of the subject and how theory could solve practical problems.

06.08.25

(Signature and Date)

Abu Jafar Md. Jakaria

Senior Lecturer

Department of Computer Science and Engineering

Metropolitan University, Sylhet

CERTIFICATE OF REVIEW

This is to confirm that we have evaluated the undergraduate project titled, "**EduNest - A Full-Stack Learning Management System**" submitted by **Al Jamil** and **Ahmad Bakhtiar Nayeem**, from Computer Science and Engineering Department, Metropolitan University, Sylhet.

Upon assessment of this project, to the best of our knowledge and ability, the team has undertaken considerable research and practical work in this project; and as such, the criteria of the bachelor degree have been adequately satisfied in terms of academic requirements. The project clearly shows a good grasp of the topic, wide range of analysis, and originality in their creative aspect in the form of problem-solving.

06.08.25

(Signature and Date)

Md. Mushtaq Shahriyar Rafee

Assistant Professor

Department of Computer Science and Engineering

Metropolitan University, Sylhet

Naeem 06/08/2025

(Signature and Date)

Golam Mostofa Naeem

Lecturer

Department of Computer Science and Engineering

Metropolitan University, Sylhet

ABSTRACT

Online learning has never been more critical in our modern society. This project, **EduNest**, is our response to the growing demand for a modern and user-friendly Learning Management System (LMS). **EduNest** was built with the MERN stack, which includes MongoDB, Express.js, React.js, and Node.js. Using this technology stack allowed us to create a fast, scalable and consistent platform.

EduNest was built to serve 3 types of users: Students, Instructors, and a Admin. The Students can subscribe to and buy courses, watch video lessons, and track their learning journey. The Instructors can create their own courses, upload content, and see how well their courses perform with a full analytics dashboard. The Admin has full control over the entire system, including managing users, courses, and the money generated in the system.

We wanted to build a complete, real-world application that can be powerful for Instructors and simple for Students. This report will take you through our journey from the very first project idea until we produced a final product.

THIS PROJECT PAPER SUBMITTED FOR THE IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF 'BACHELOR OF SCIENCE' IN COMPUTER SCIENCE AND ENGINEERING

Chapter		Contents	Page No.
Chapter: 1		**Introduction**	11-14
	1.1	Introduction	12-12
	1.2	Project Objectives	12-12
	1.3	Project Overview	13-13
	1.4	Problem definition and proposed solutions	13-14
	1.5	Motivation	14-14
Chapter: 2		**Feasibility Study**	15-17
	2.1	Introduction	16-16
	2.2	Technical Feasibility	16-16
	2.3	Operational Feasibility	16-16
	2.4	Economic Feasibility	17-17
Chapter: 3		**System Analysis and Requirements**	18-20
	3.1	System Analysis	19-19
	3.2	Requirement Analysis (Task Breakdown)	19-20
	3.3	Requirement Analysis	20-20
Chapter: 4		**Process Model and Design**	21-23
	4.1	Development Methodology	22-22
	4.2	System Architecture	23-23
	4.3	Database Design	23-23
Chapter: 5		**System Diagrams**	24-33
	5.1	Use Case Diagrams	25-27
	5.2	System Architecture Diagram	28-28
	5.3	Data Flow Diagram	29-31
	5.4	Entity-Relationship Diagram (ERD) Diagram	32-32
	5.4	Sequence Diagram	33-33
Chapter: 6		**System Design**	34-56
	6.1	Database and Required Collections	35-35
	6.2	System Interface	35-56

Chapter		Contents	Page No.
Chapter: 7		**System Testing and QA Integration**	57-61
	7.1	Testing Approach	58-58
	7.2	Testing Methodologies	59-59
	7.3	Industry-Standard QA Test Cases	59-61
Chapter: 8		**System Documentation**	62-63
	8.1	Discussion and Conclusion	63-63
	8.2	Limitations and Outcome	63-63
	8.3	Future Scope	63-63
		Reference	64

Chapter One

Introduction

Contents:

- Introduction
- Project Objectives
- Project Overview
- Problem Definition and Proposed Solution
- Motivation

1.1 Introduction

Welcome to EduNest. In a world where learning is no longer confined to classrooms, we wanted to build a digital space—a "nest"—where knowledge can be shared and grown. EduNest is a modern online learning platform designed to connect instructors who have knowledge to share with students who are eager to learn.

Our platform is a complete system, built from the ground up using the powerful MERN stack. This means we used modern JavaScript technologies for everything, from the user-friendly interface you see to the powerful server that runs in the background.

1.2 Project Objectives

Our main objectives for this project were plain from the beginning. We desired to:

- **Build a Full-Stack Application:** Create a development-ready, functioning web application using the MERN stack in a manner that displays our ability to complete both frontend and backend development.
- **Facilitate different user roles:** Design a system that works for students, instructors and an admin with roles tailored to their needs.
- **Develop a good user experience:** Create a platform that is simple and fun to use, is clean, user friendly and easy to navigate.
- **Solve full-real world tasks:** Implement complex features including secure online payments through Stripe, video uploads through Cloudinary, and automatic emails like registrations and certificates.
- **Make it scalable:** Develop a platform that will allow growth over time and allow itself to cope with additional users and users.

1.3 Project Overview

EduNest isn't just simply a website; it is a learning ecosystem. It comes in three sections for three types of users:

- **The Student Panel:** This panel is for students. Students create an account, they can search for interesting courses, select and pay securely online, and get learning right away. They can view their progress, take quizzes, and ask the instructor questions.
- **The Instructor Panel:** This is where the teachers are creative. Instructors can create and upload their courses, develop their video lectures, and communicate with their students as well as view their course stats and a revolutionary dashboard to view how many people are purchasing their courses and how much money they are profiting.
- **The Admin Panel:** The Admin Panel is the entirety of the platform. The Admin is able to manage all users and courses, approve new instructors, and check and manage all financial transactions in order to keep the platform executing without obstacles.

1.4 Problem Definition and Proposed Solution

The Problem: Many online learning platforms available today can feel old, complicated, or too expensive. For instructors, it can be difficult to set up and manage their courses. For students, finding high-quality, relevant courses can be a challenge. There was a need for a platform that is modern, simple, and powerful all at once.

Our Solution: EduNest is designed to solve these problems.

- For Instructors, we provide an all-in-one dashboard that makes it simple to create courses, upload videos, and track earnings.
- For Students, we offer a clean, easy-to-navigate interface to find and purchase courses, with a seamless learning experience.
- For Everyone, we use modern technology to make the platform fast, secure, and reliable.

1.5 Motivation

Our motivation for creating EduNest came from two points. First, we observed the amazing growth of online learning, in particular, in Bangladesh, and wanted to be a part of that transformation and create something that contributed to people's learning and career progressions.

Second, as computer science students, we wanted to push ourselves. Creating a complete Learning Management System (LMS) is a complex project that involves knowledge of many technical areas. This was a project that allowed us to take everything we learned in our studies to build a real-world application that we can be proud of.

Chapter Two

Feasibility Study

Contents:

- Introduction
- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

2.1 Introduction

Before we wrote a single line of code, we had to ensure that our idea was practical. A Feasibility Study gives us insight into whether or not a project is possible to build and if it makes sense to build. We viewed it from three angles: technical, operational, and economic.

2.2 Technical Feasibility

- **Is the technology available?** Yes. The MERN stack is a popular and well-supported set of technologies. React, Node.js, and MongoDB are all open source and free to use with large communities of developers actively sharing knowledge and solutions.

- **Do we have the skills?** Yes. While at the CSE department we did hard courses on JavaScript and web development which gave us the basis we needed on the core skills for the MERN stack.

- **Can it do the work?** Yes. The MERN stack is built for what is referred to in the industry as scalability meaning we can manage increasing numbers of users and data. Payments and videos will be managed by industry leading services like Stripe and Cloudinary designed for reliability.

2.3 Operational Feasibility

- **Will people use it?** We believe so. EduNest is designed for simplicity and usability. The dashboards for the instructors and for admins are configured to make the platform easy to manage.

- **Does it solve the problem?** Yes. It offers a modern and complete solution for online learning that is in demand.

2.4 Economic Feasibility

Is it affordable to build? Yes. Since we used open-source technologies like React and Node.js, there were no initial software costs.

What are the running costs? The main costs would be for hosting the website and database, and for using third-party services like Stripe (which takes a small percentage of each transaction) and Cloudinary (which charges based on usage). These costs are manageable and would only grow as the platform becomes more successful.

Chapter Three

System Analysis and Requirements

Contents:

- System Analysis
- Requirement Analysis (Task Breakdown)
- System Requirements

3.1 System Analysis

To create a good system, you need to know what the system is supposed to do. We broke these requirements into two categories:

- **Functional Requirements:** These are the things the system is supposed to do. An example of functional requirements can include "a user must be able to create an account" or "an instructor must be able to upload a video."
- **Non-Functional Requirements:** These are the qualities that the system must have. An example of non-functional requirements is that the website needs to be fast, secured, and easy to use on desktops and mobile phones.

3.2 Requirement Analysis (Task Breakdown)

We created a detailed list of every single feature needed for the project. This helped us stay organized and make sure we didn't miss anything.

Student / User Panel

- Create and log into an account with email verification.
- Reset password if forgotten.
- Create and update a personal profile with a profile picture.
- Search and filter courses by category and price.
- Purchase courses securely using the Stripe payment gateway.
- View personal transaction history and download invoices.
- Access all enrolled courses from a "My Learning" page.
- Track progress through courses and individual lectures.
- Watch video lectures in a built-in player.
- Take quizzes to test knowledge.
- View quiz results and receive a certificate via email upon passing.
- Ask questions and reply to answers in a Q&A section for each lecture.
- Apply to become an instructor.

Instructor Panel

- View a dashboard with key analytics (revenue, sales, student count).
- See detailed financial statements and request payouts.

- Create, edit, and manage courses.
- Publish or unpublish courses.
- Upload video lectures and other course materials to Cloudinary.
- Create and manage quizzes for courses.
- View a list of all enrolled students and their progress.
- Answer student questions in the Q&A section.

Admin Panel

- View a main dashboard with platform-wide statistics.
- Manage all users (students and instructors), including creating, enabling, or disabling accounts.
- Approve or refuse applications from students who want to become instructors.
- Manage all courses on the platform, including the ability to publish or delete any course.
- Create and manage course categories.
- View all financial transactions and manage instructor payout requests.

3.3 Requirement Analysis

Software and Technologies Used:

- **Frontend:** React.js, Redux Toolkit, RTK Query, Tailwind CSS, ShadCN UI
- **Backend:** Node.js, Express.js
- **Database:** MongoDB with Mongoose
- **Authentication:** JSON Web Tokens (JWT)
- **Payment:** Stripe
- **File/Video Storage:** Cloudinary
- **Email Service:** Nodemailer
- **Development Tools:** Vite, Git, VS Code

Chapter Four

Process Model

Contents:

- Development Methodology
- System Architecture
- Database Design

4.1 Development Methodology

For this particular project we chose an Agile development approach, which differs from the more traditional Waterfall model, where one stage of development must be fully completed before continuing to the next.

Using an agile approach, we built the application in small pieces, or iterations. In our example of working on the login feature, we would build the entire feature, test that the login feature worked as intended before moving to the next feature. This approach enabled us to remain flexible, make changes easily, and see our progress, one iteration at a time

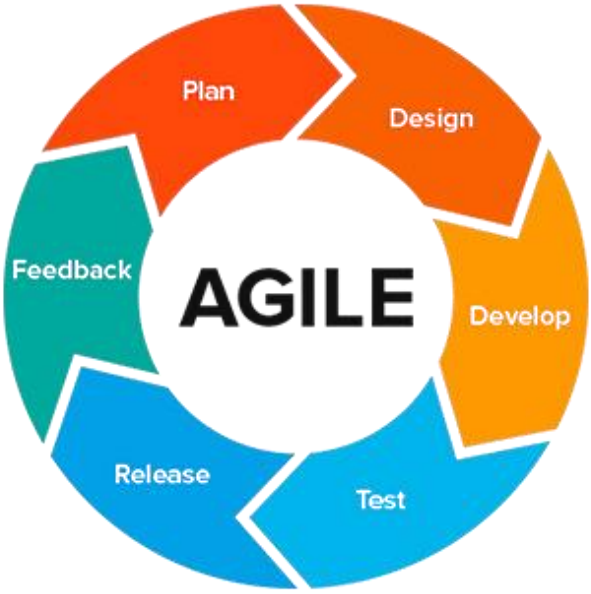


Fig-4.1: Agile Model

This methodology will help us:

- To produce a better-quality system, in terms of documentation standards, To maintainability and consistency of software
- To ensure that the user requirements are met completely
- To have better control over the project
- Finally, to understand and learn some of the software engineering techniques.

4.2 System Architecture

EduNest is built on a client-server architecture. This means we have two main parts that talk to each other over the internet:

- **The Client (Frontend):** This is the part you see and interact with in your web browser. It's built with React and is responsible for showing the user interface. When you click a button, the client sends a request to the server.
- **The Server (Backend):** This is the "brain" of the application. It's built with Node.js and Express.js. It receives requests from the client, interacts with the database to get or save information, and sends a response back to the client.

4.3 Database Design:

We used MongoDB, which is a NoSQL database. Unlike traditional SQL databases that use tables with rows and columns, MongoDB stores data in flexible, JSON-like documents. This was a great choice for our project because it allowed us to easily store complex data, like a user's profile with their list of enrolled courses.

Our main database collections (which are like tables) include:

- **users:** Stores information about students, instructors, and the admin.
- **courses:** Contains all the details about each course, including its title, description, price, and a list of its lectures.
- **lectures:** Stores information for each individual lesson.
- **coursePurchases:** Keeps a record of every transaction made on the platform.
- **quizzes and quizAttempts:** Manages the quizzes and the results of students who take them.

Chapter Five

System Diagrams

Contents:

- Use Case Modeling
- System Architecture Diagram
- Data Flow Diagram
- Entity-Relationship Diagram (ERD)
- Sequence Diagram

5.1 Use Case Diagrams

A use case diagram shows what different users can do in the system

5.1.1 Student Use Case Diagram

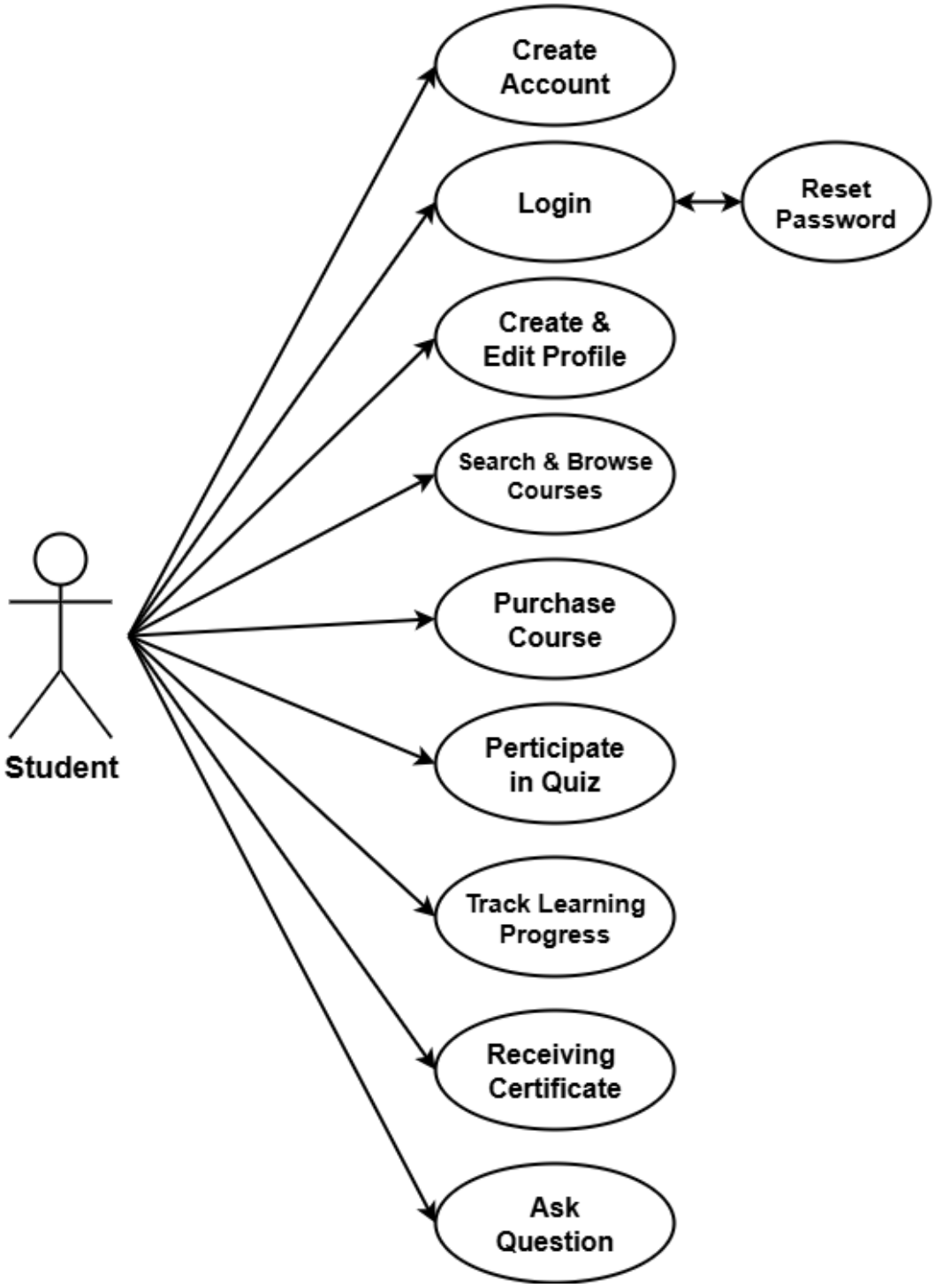


Fig-5.1.1: Use Case Diagram of Student

5.1.2 Instructor Use Case Diagram

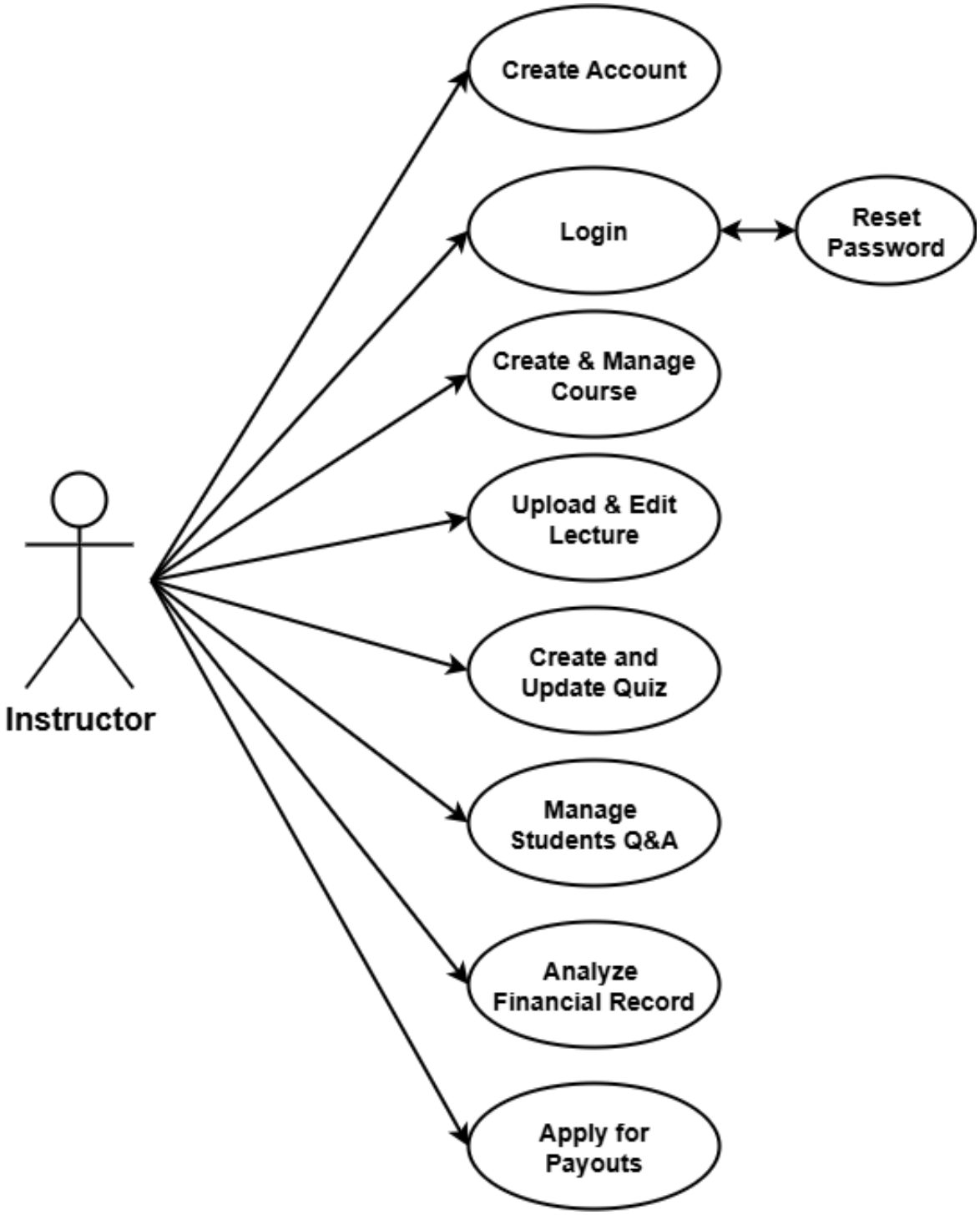


Fig-5.1.2: Use Case Diagram of Instructor

5.1.3 Admin Use Case Diagram

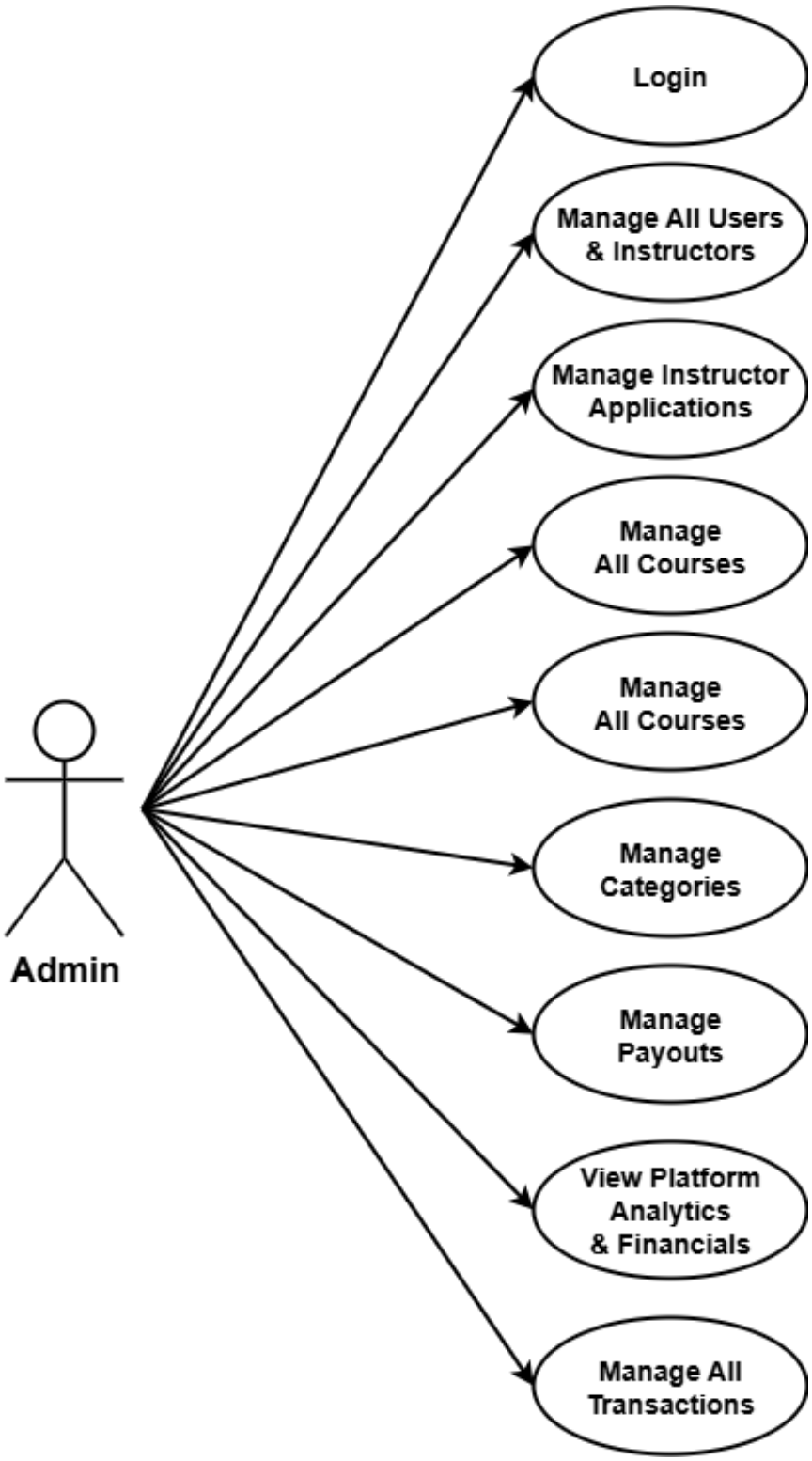


Fig-5.1.3: Use Case Diagram of Admin

5.2 System Architecture Diagram

This diagram shows how all the different parts of our system connect and work together.

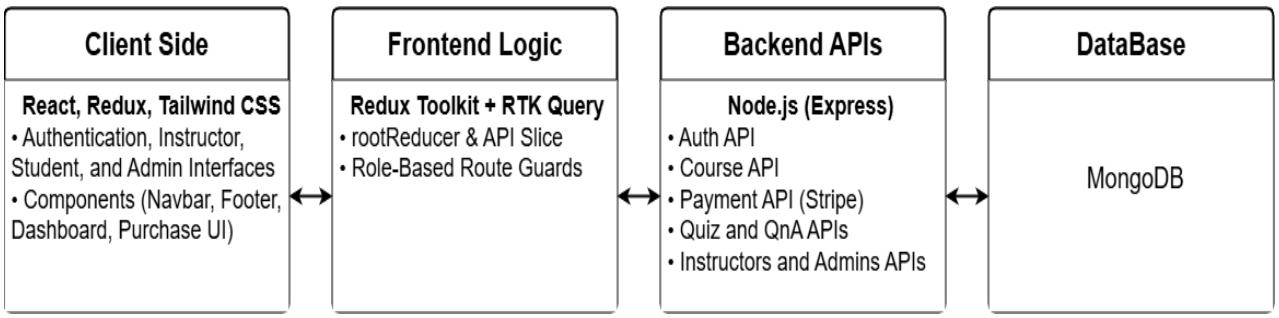


Fig-5.2.1 System Architecture Diagram

5.3 Data Flow Diagram

A Data Flow Diagram represents the flows of information between the system and all external entities that may have interact with a system. The entire software system is shown as a single process.

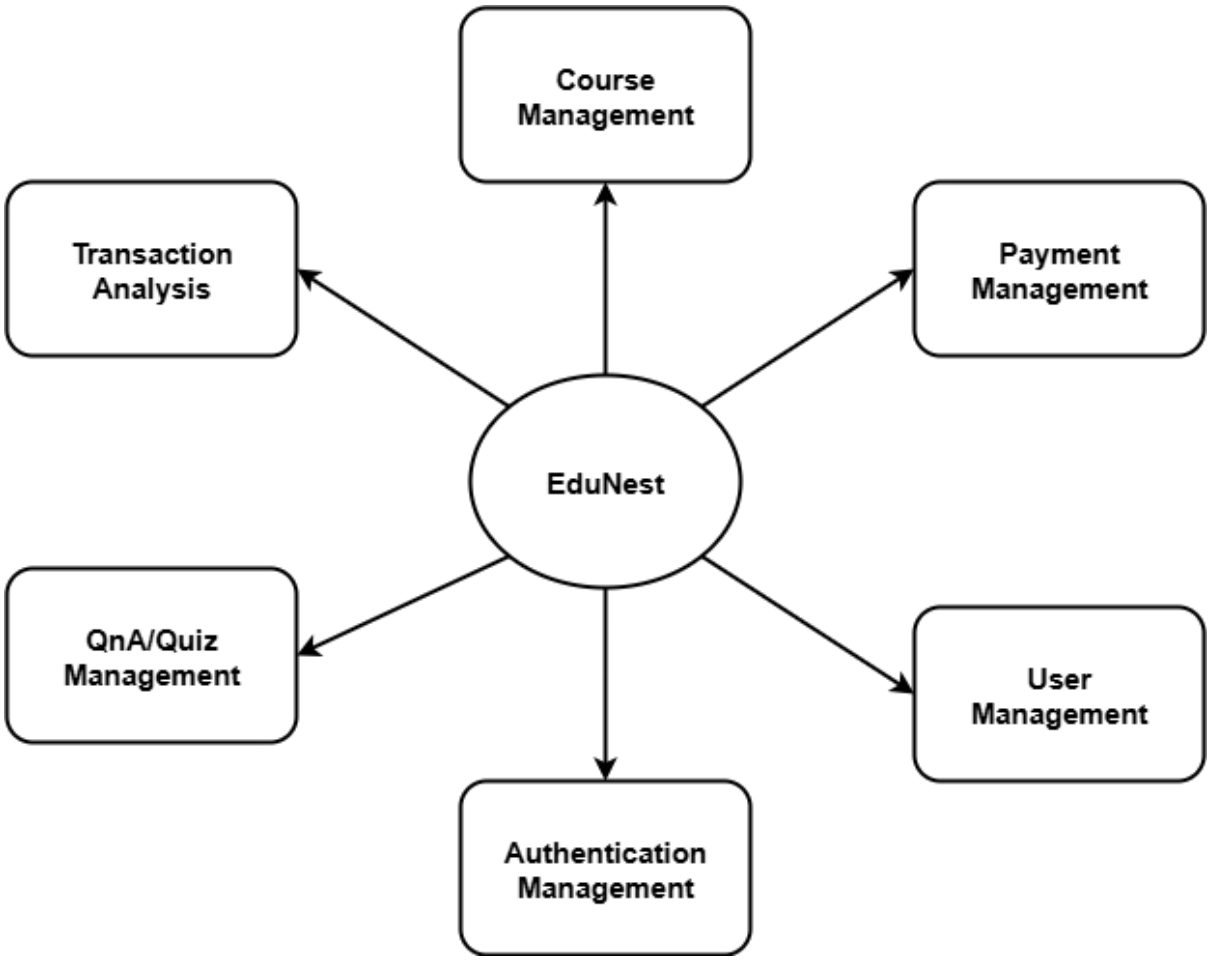


Fig-5.3.1: Level-0 DFD Diagram

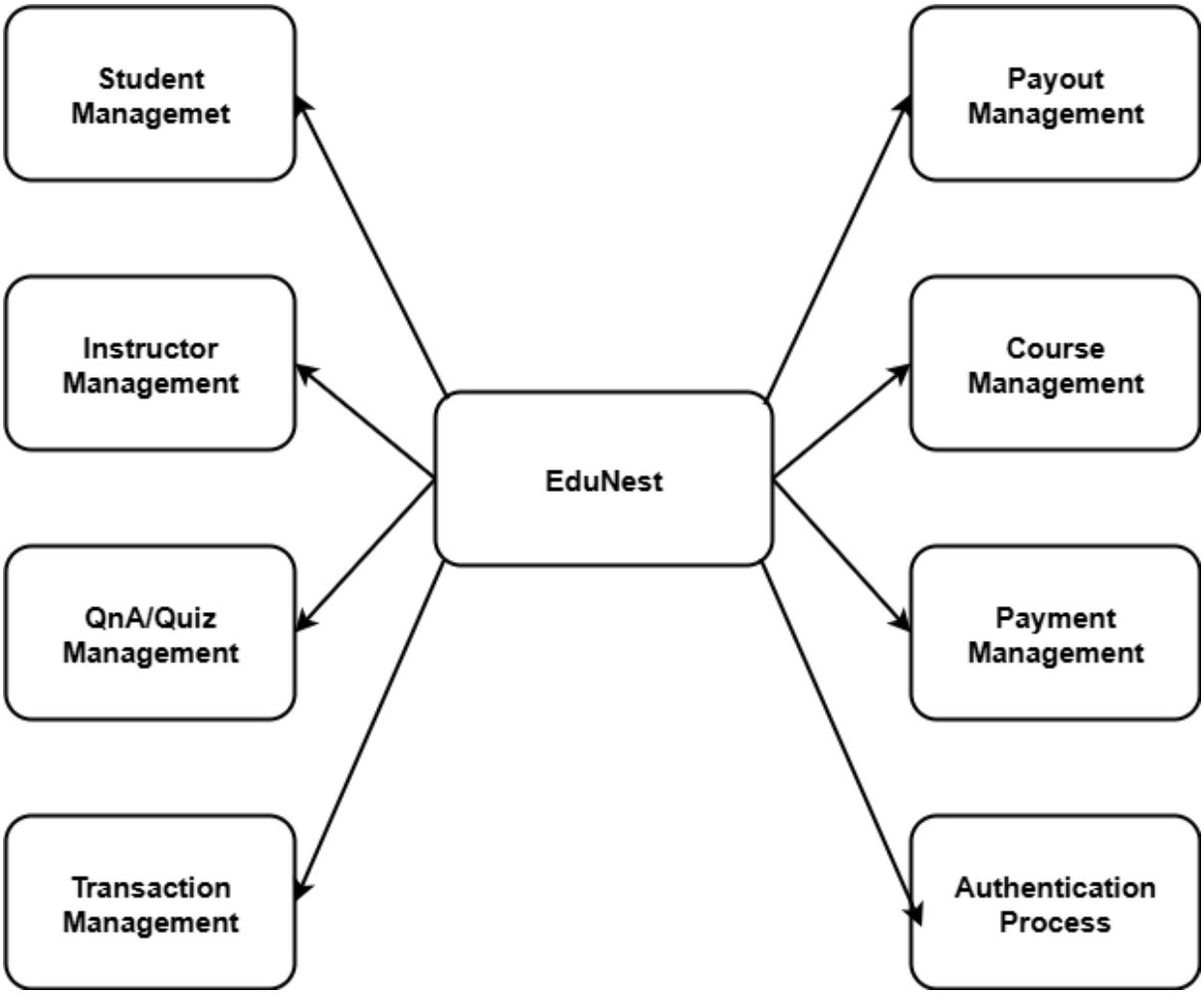


Fig-5.3.2: Level-1 DFD Diagram

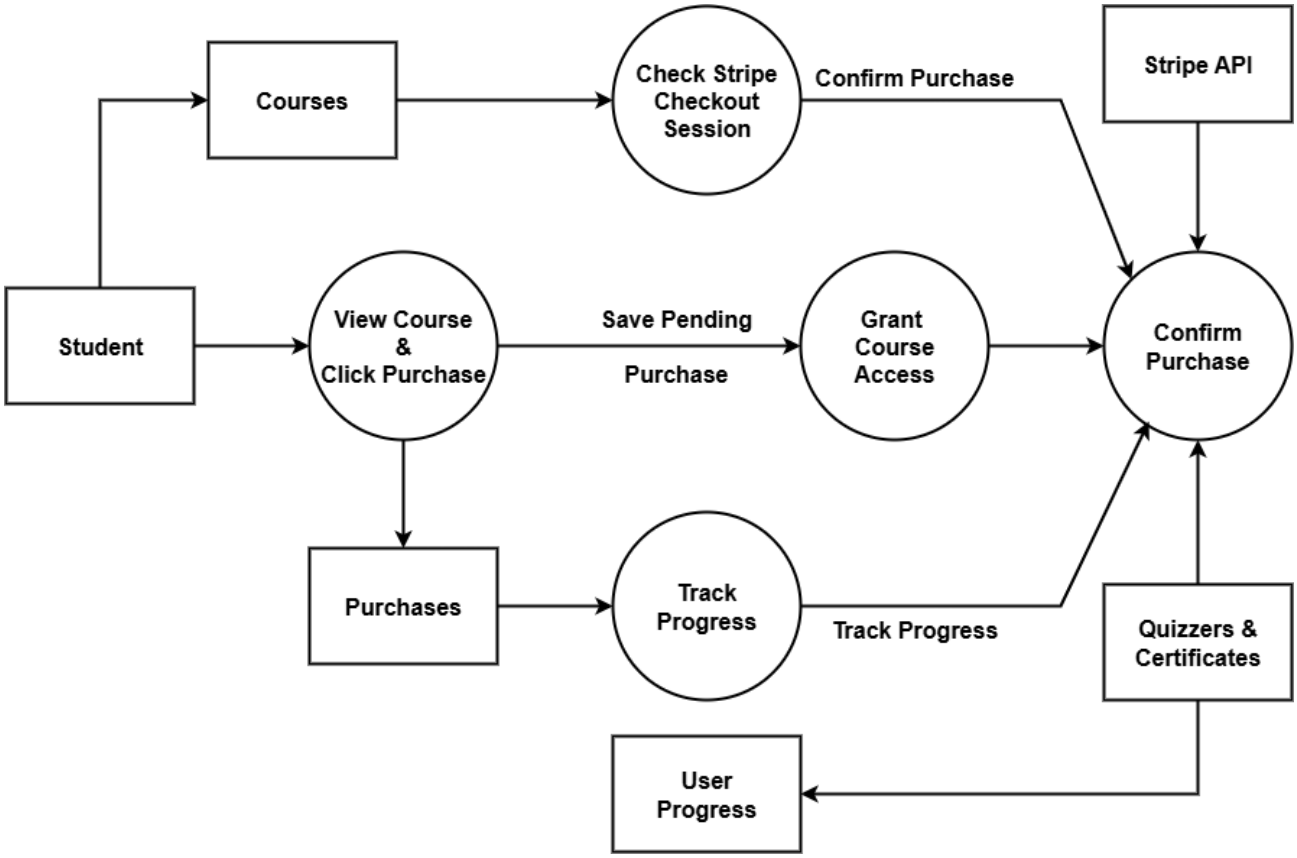


Fig-5.3.3: Level-2 DFD Diagram

5.4 Entity-Relationship Diagram (ERD) Diagram

E-R diagram is Entity Relationship diagram which can describes interrelated things of interest in a specific domain of knowledge.

E-R diagrams of this system are given below.

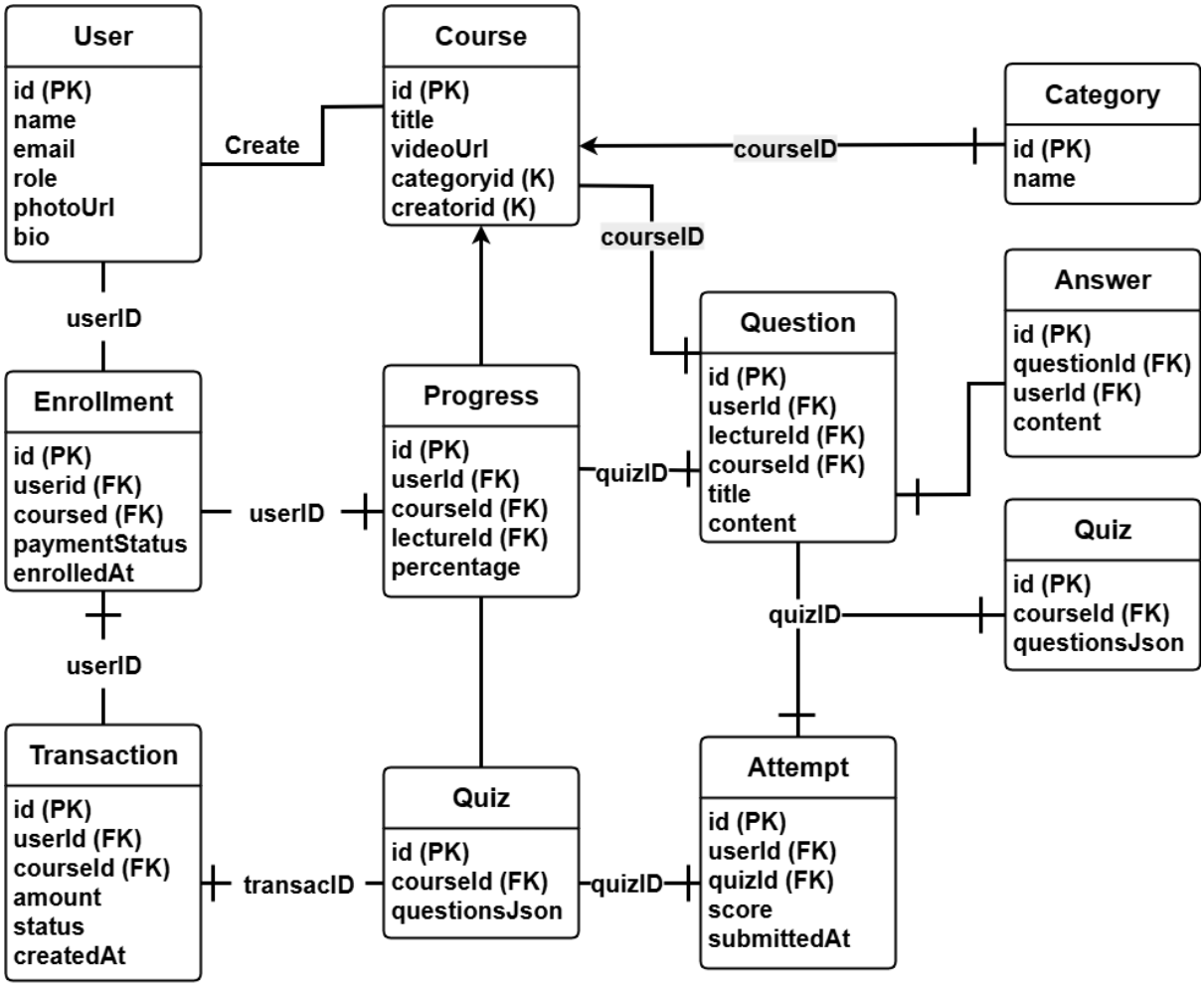


Fig-5.4 Entity-Relationship Diagram

5.5 Sequence Diagram

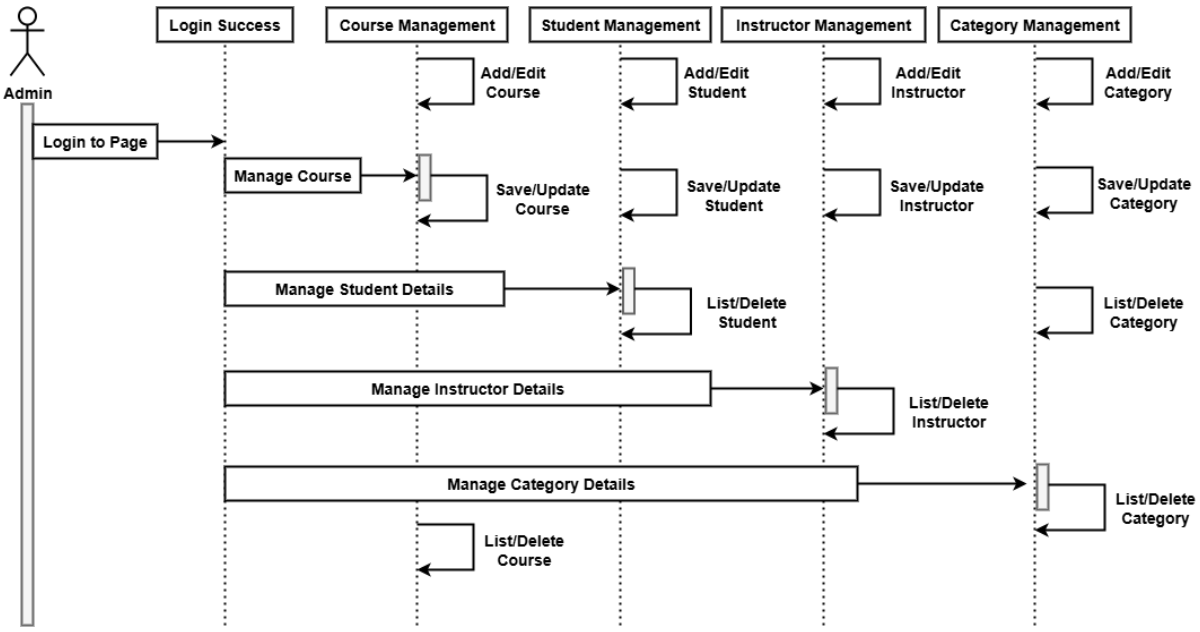


Fig-5.5 Sequence Diagram of Learning Management System

Chapter Six

System Design

Contents:

- Database and Required Collections
- System Interface

6.1 Database and Required Collections

For this project, we utilized MongoDB, a NoSQL document-based database, for storing all application data. Unlike traditional SQL databases, MongoDB's flexible, JSON-like document structure was ideal for our application, allowing us to easily store complex and nested data, such as a user's profile with their list of enrolled courses. We used Mongoose as an Object Data Modeling (ODM) library to interact with our database.

Our main database collections include:

- **users:** Stores information for all roles (students, instructors, admin), including profiles, credentials, enrolled courses, and payout details.
- **courses:** Contains details for each course, such as title, description, price, category, creator, and references to its lectures and quiz.
- **lectures:** Stores information for each individual lesson, including its title and the URL for the video content hosted on Cloudinary.
- **coursePurchases:** Keeps a detailed record of every transaction, linking a user to a purchased course and storing payment status and amount.
- **quizzes:** Contains the structure for each course's quiz, including the questions, options, and correct answers.
- **quizAttempts:** Stores the results of each quiz taken by a student, including their score and detailed answers.
- **payouts:** Tracks payout requests made by instructors and their status (pending, completed, etc.).
- **categories**: Manages the list of available course categories

6.2 System Interface

Interfaces for any web should be as simple as possible with attractive outlook and ease of work. In this project "**EduNest- A Full Stack Learning Management System**" various pages are used for information storing, processing and display purpose. Some of these pages are shown be

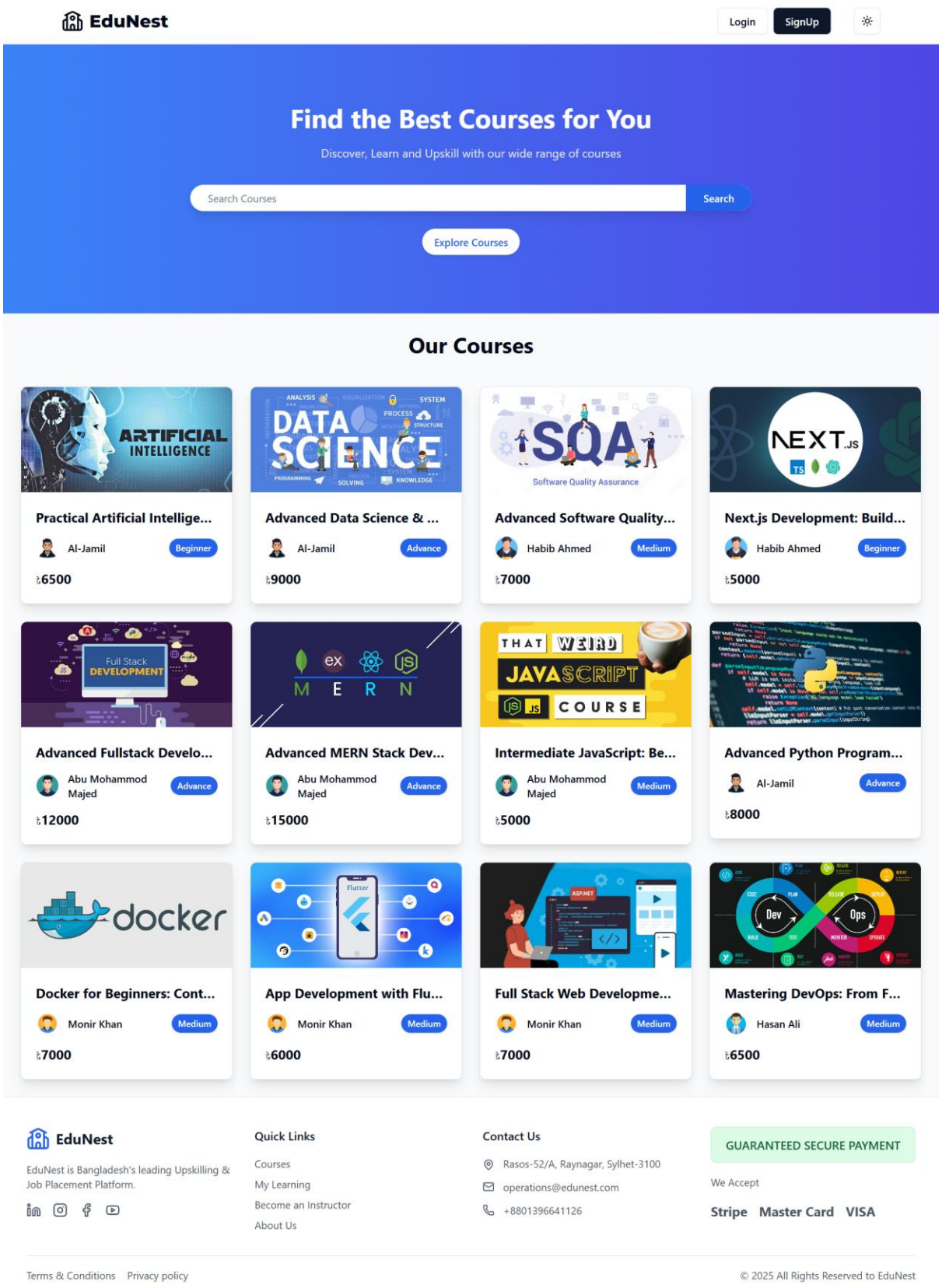


Fig-6.2.1: Home Page (Light Version)

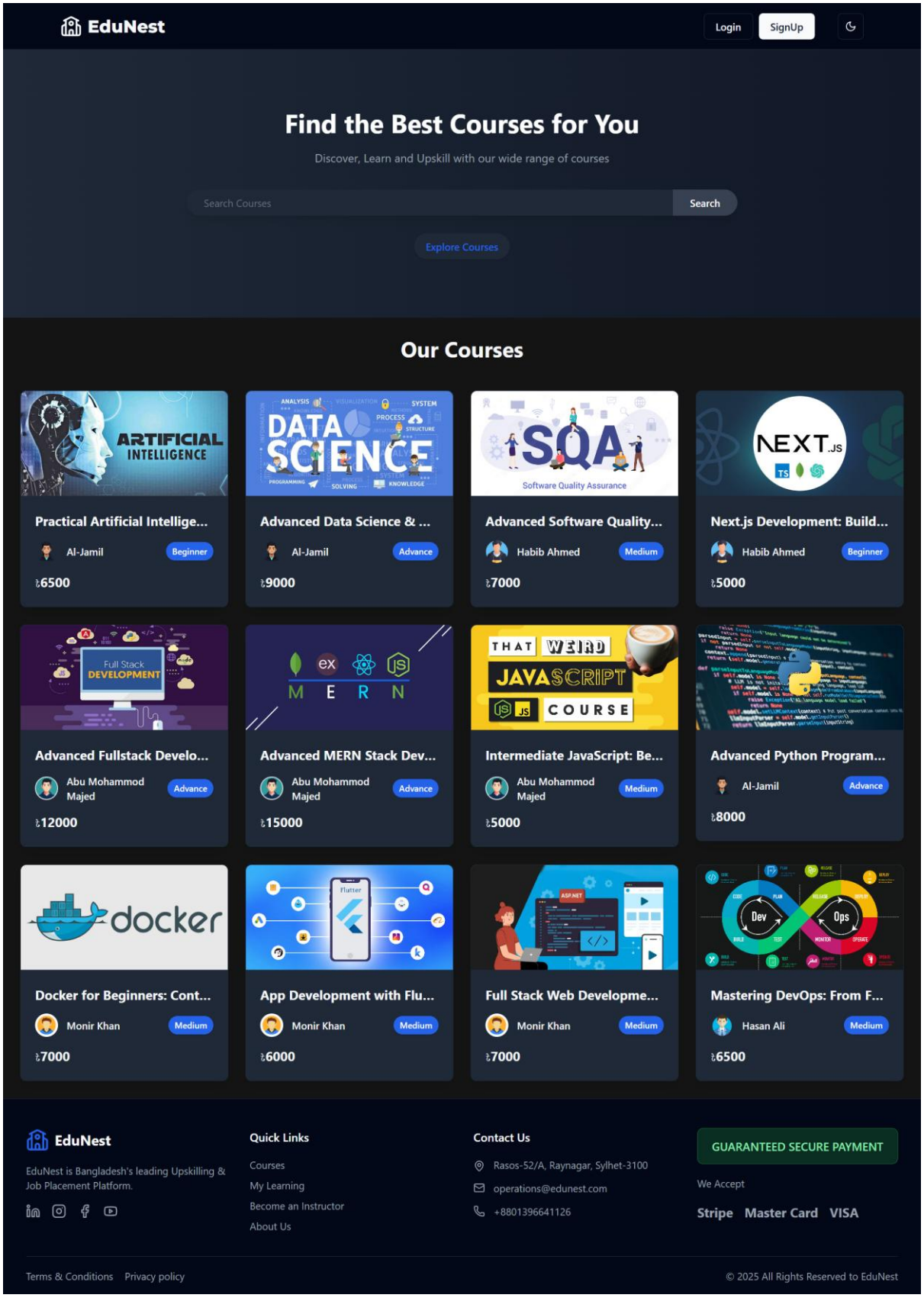


Fig-6.2.2: Home Page (Dark Version)

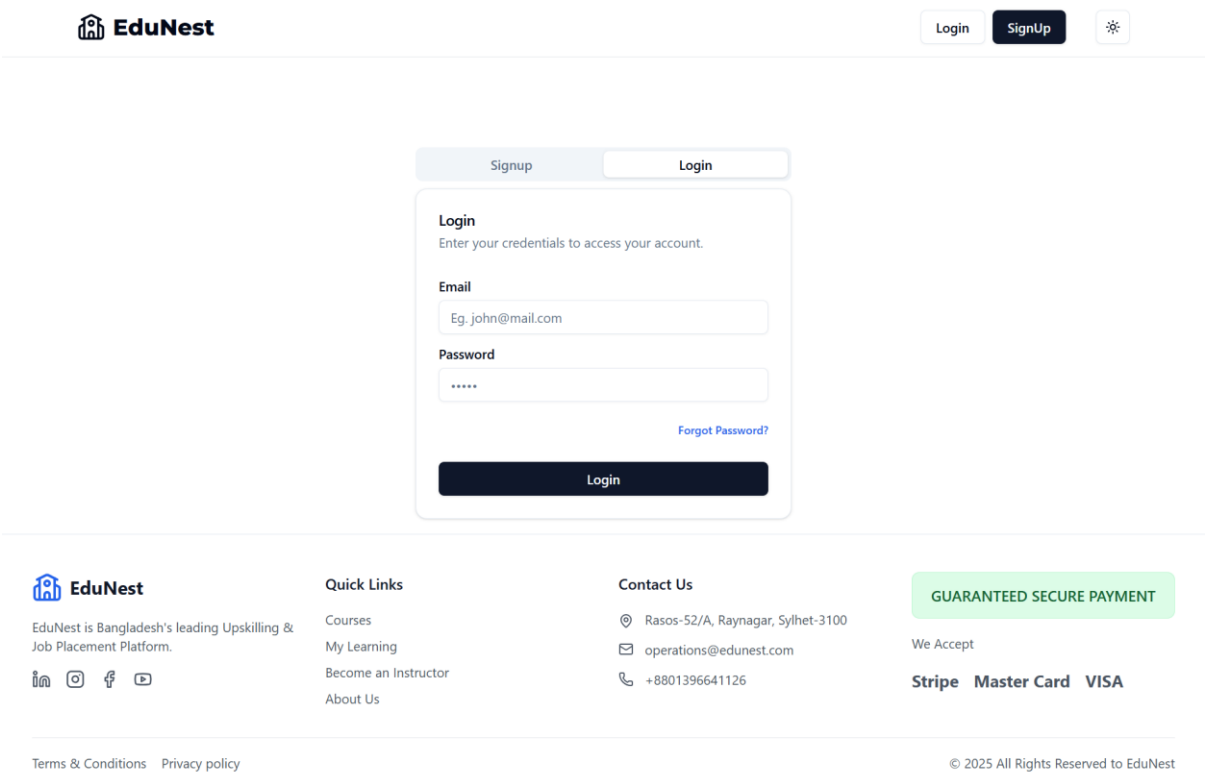


Fig-6.2.3: Login Page (For All User)

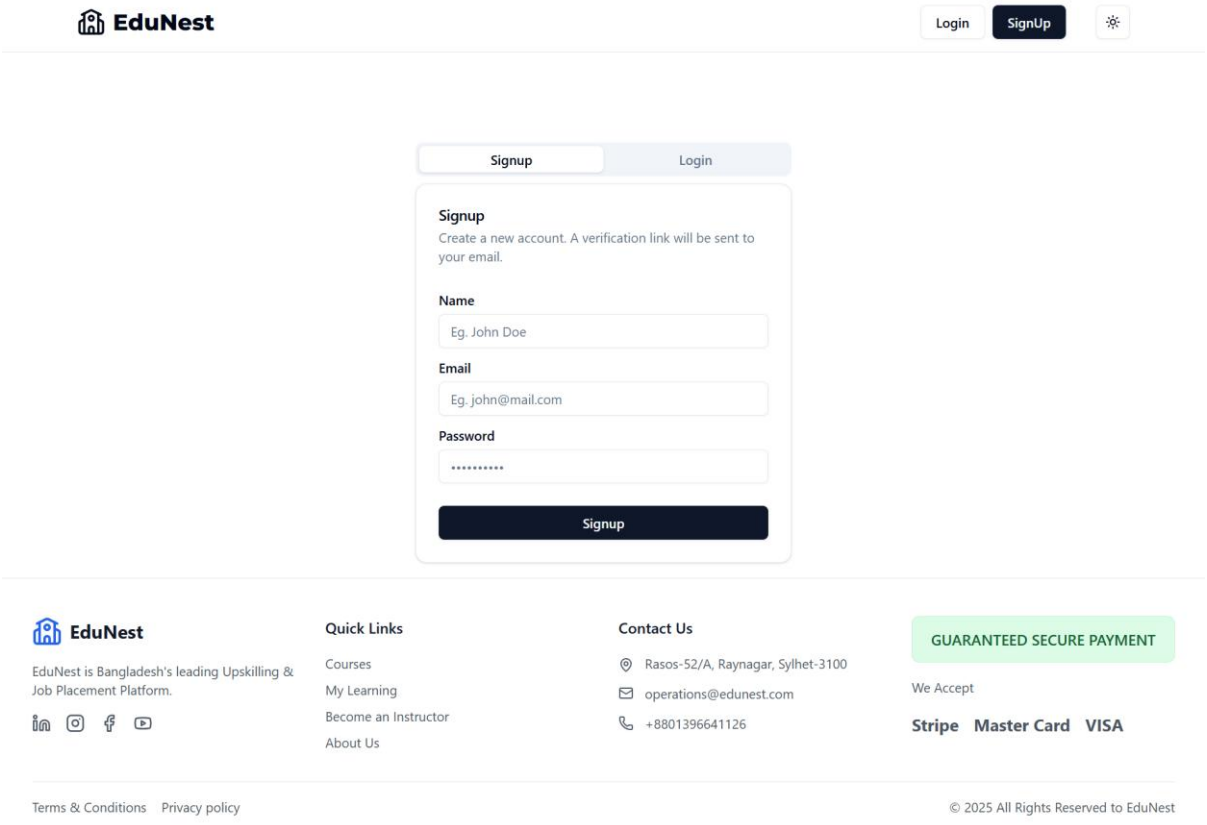


Fig-6.2.4: SignUp Page (For All User)

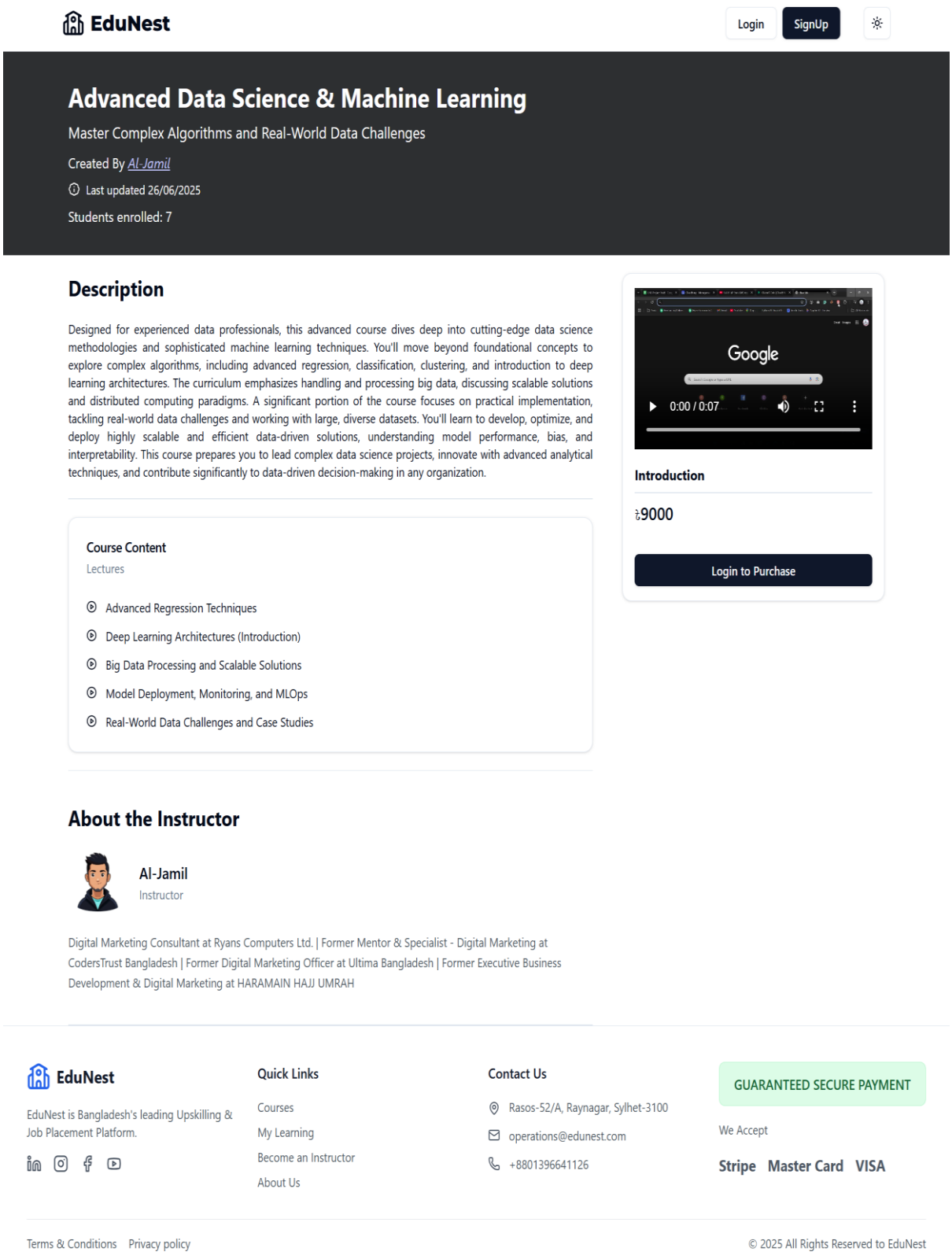


Fig-6.2.5: Course Detail Page (For All User)

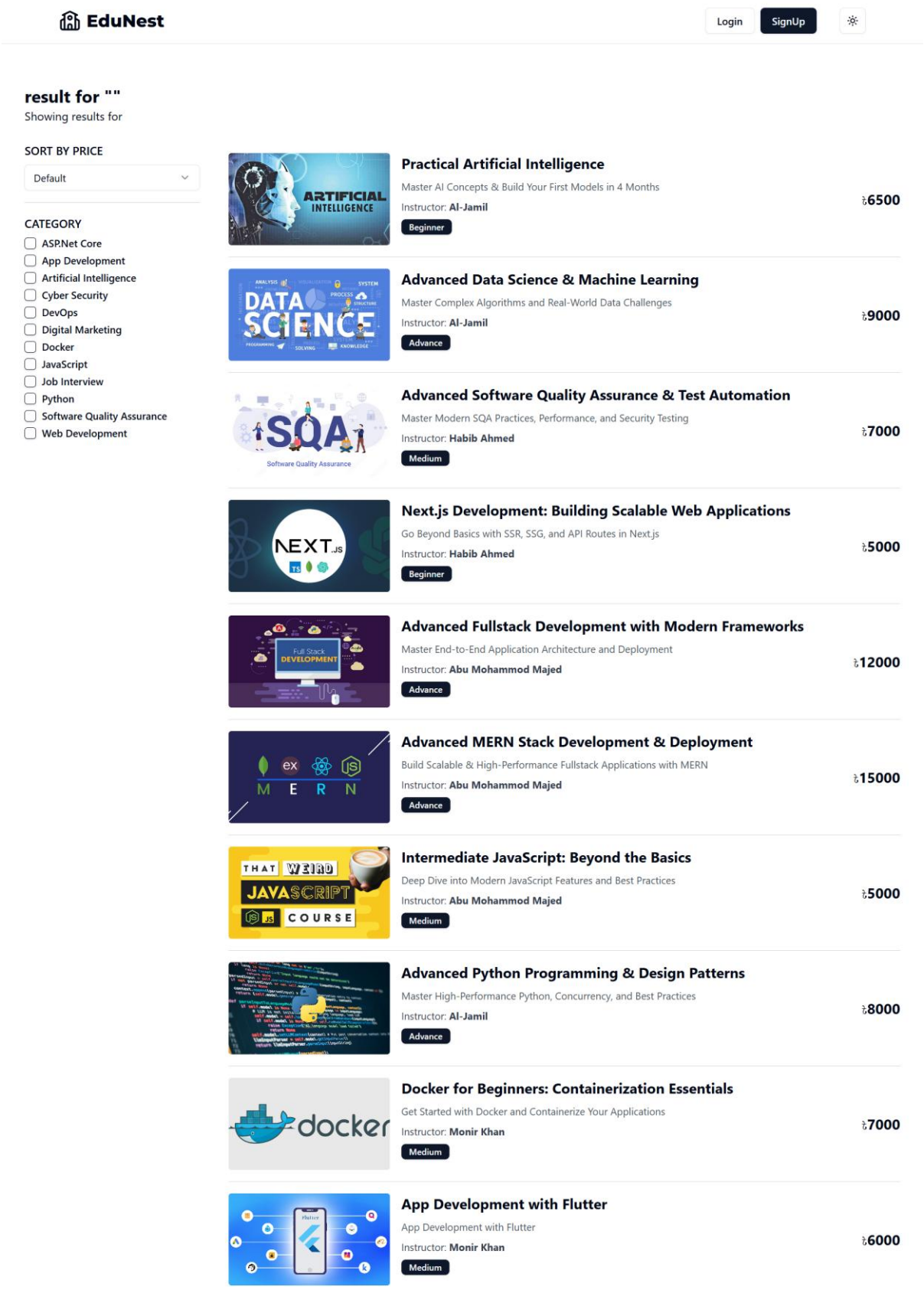


Fig-6.2.6: Course Search & Filter Page (For All User)

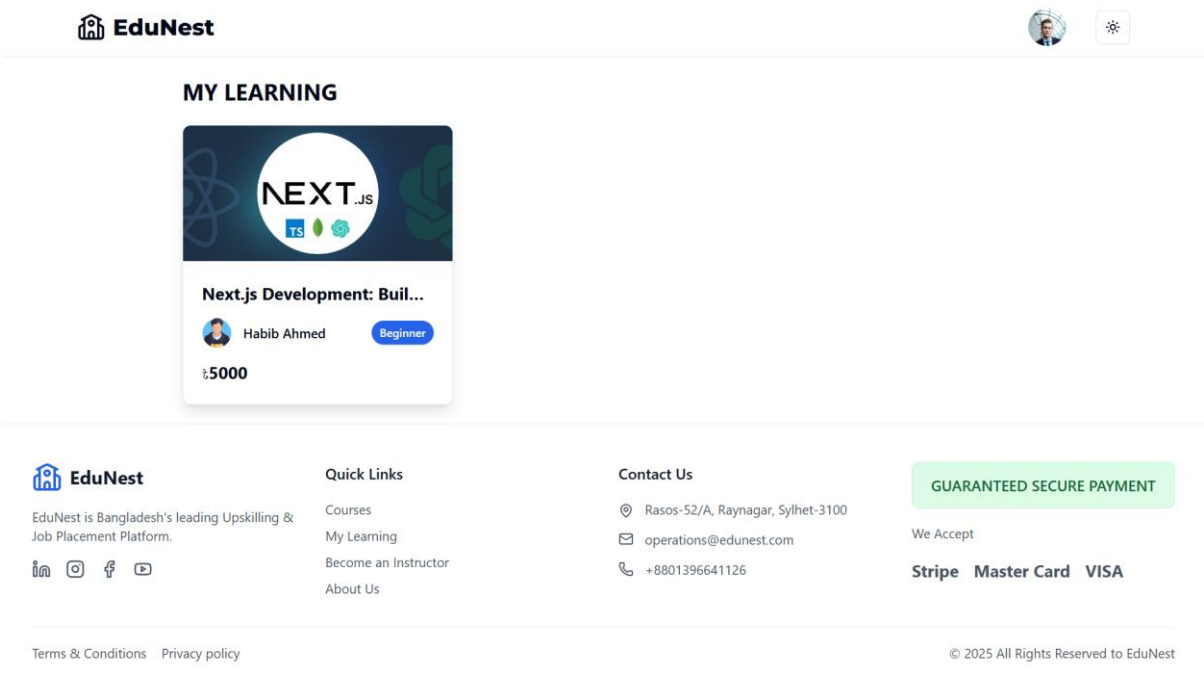


Fig-6.2.7: My Learning Page (For Student)

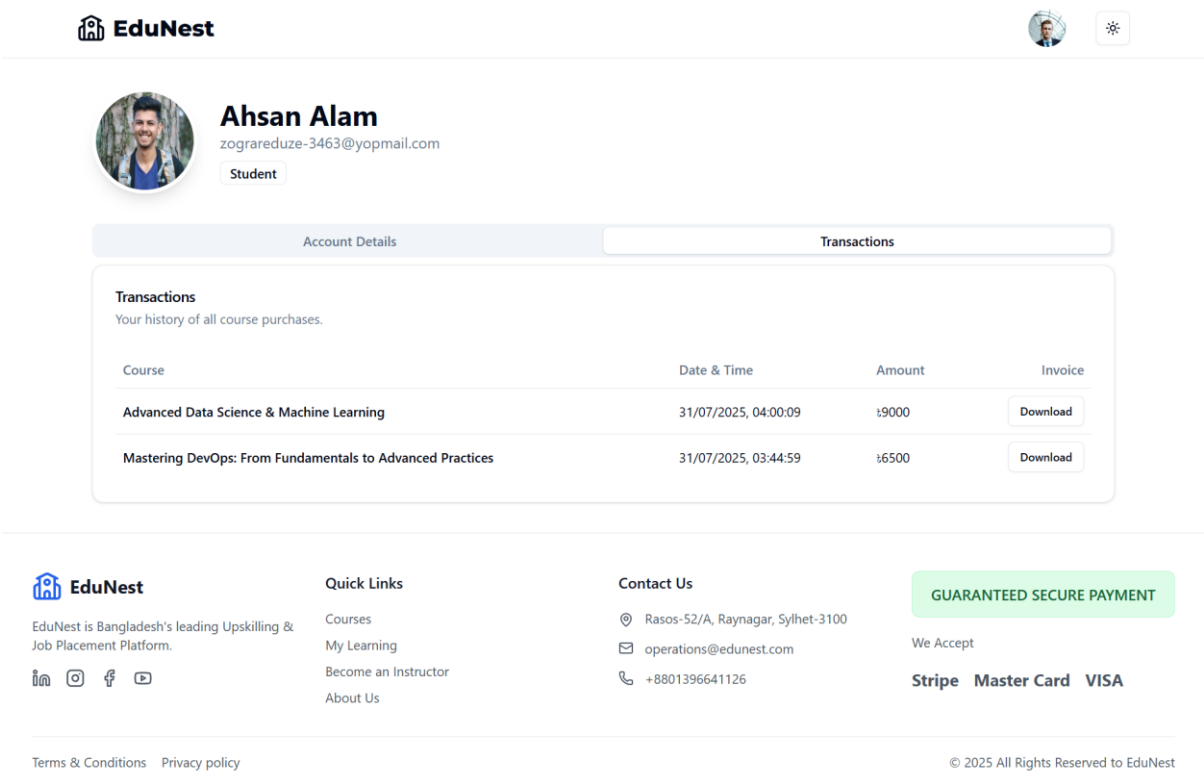


Fig-6.2.8: Transaction Page (For Student)

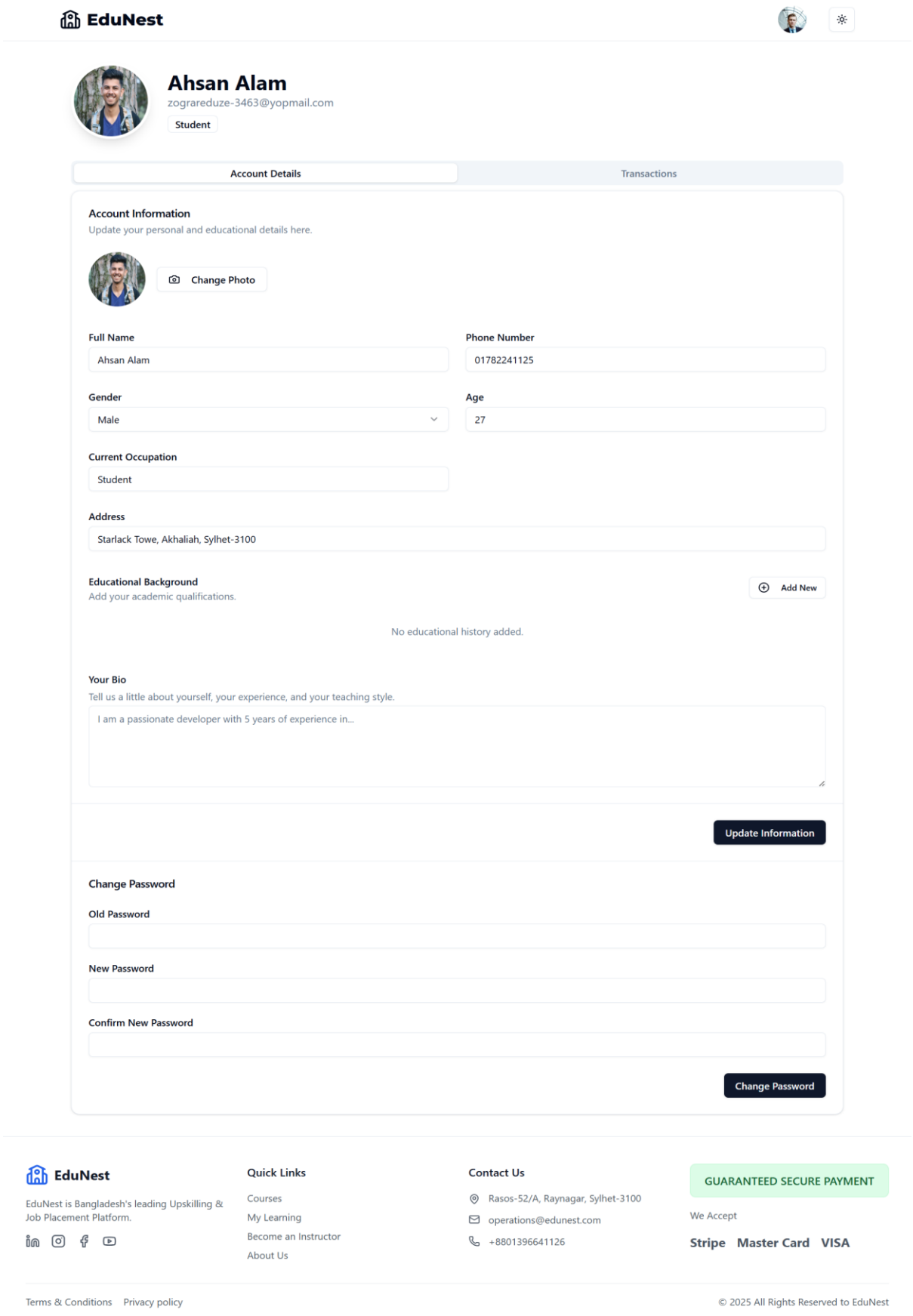


Fig-6.2.9: My Learning Page (For Student)

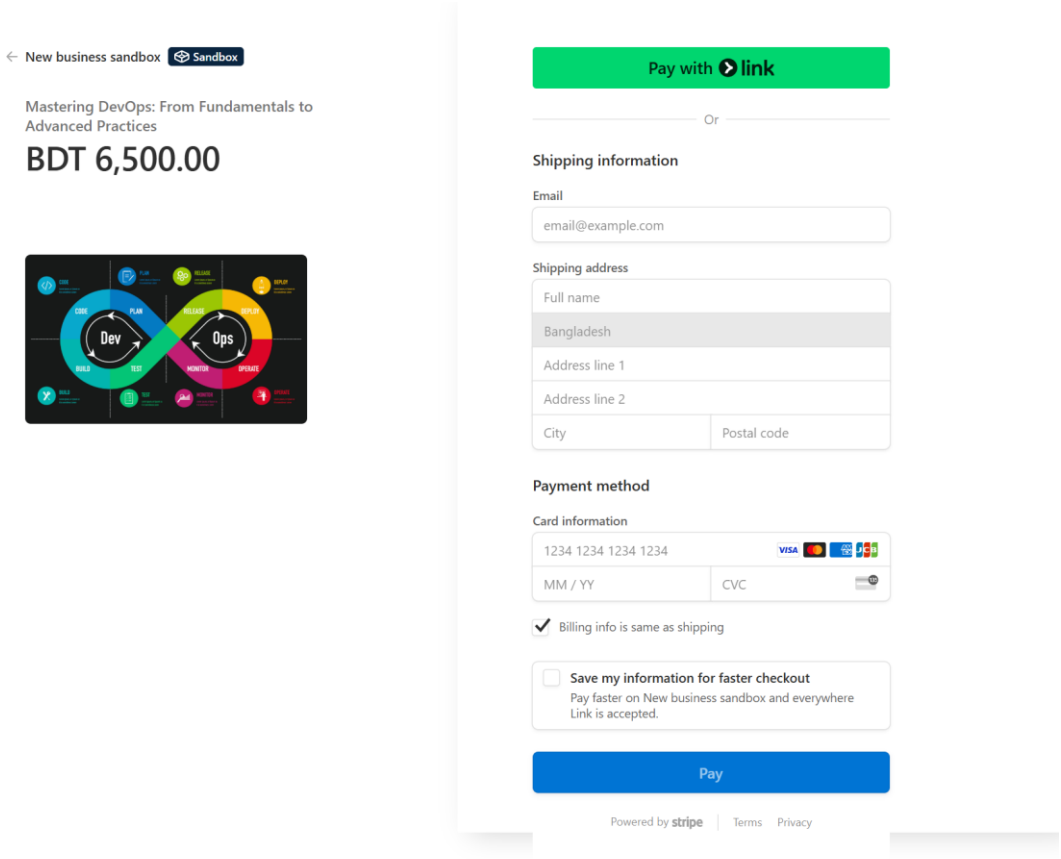


Fig-6.2.10: Payment Landing Page (For Student)

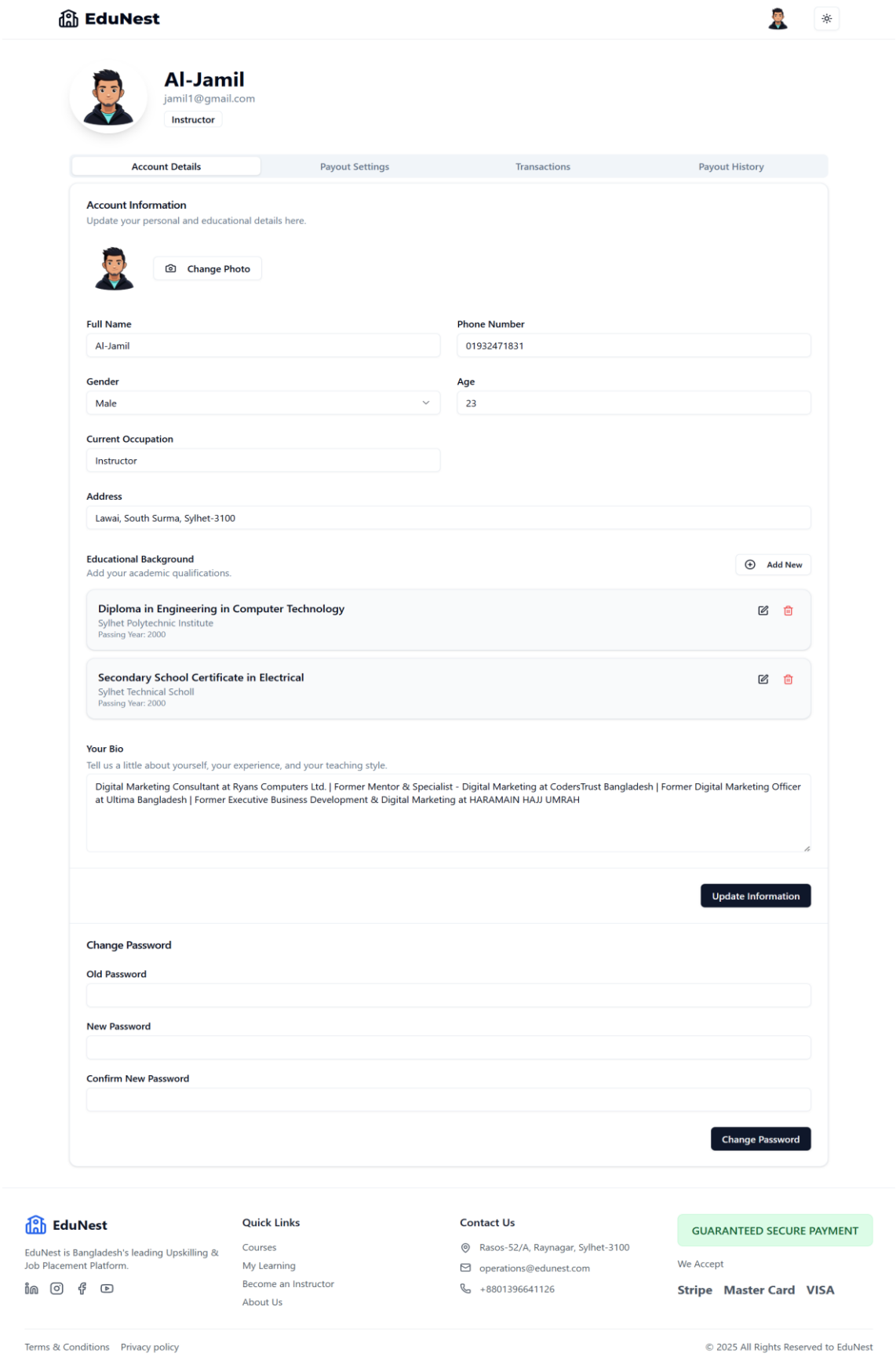


Fig-6.2.10: Profile Page (For Instructor)

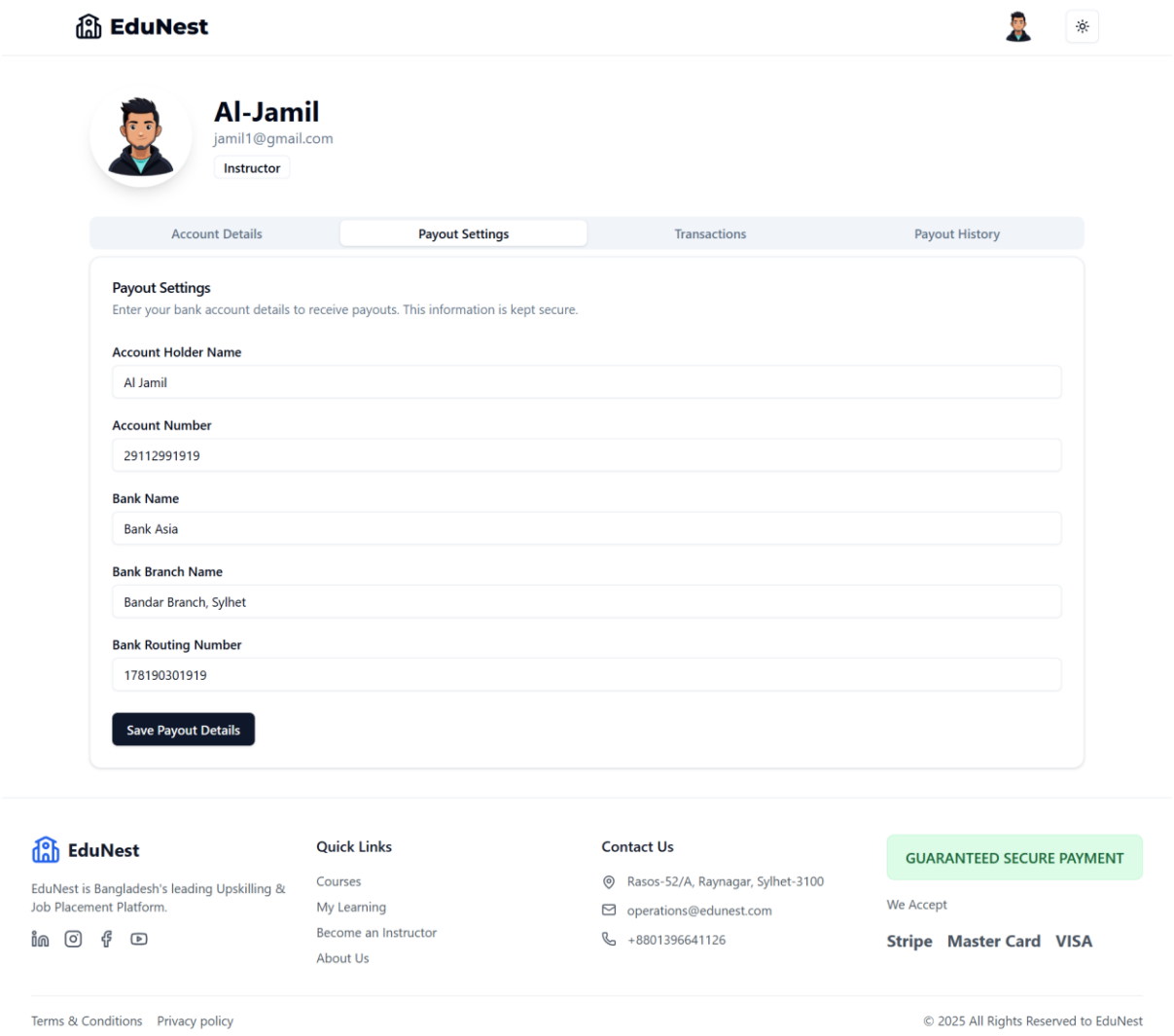


Fig-6.2.11: Instructors Payout Setting Page

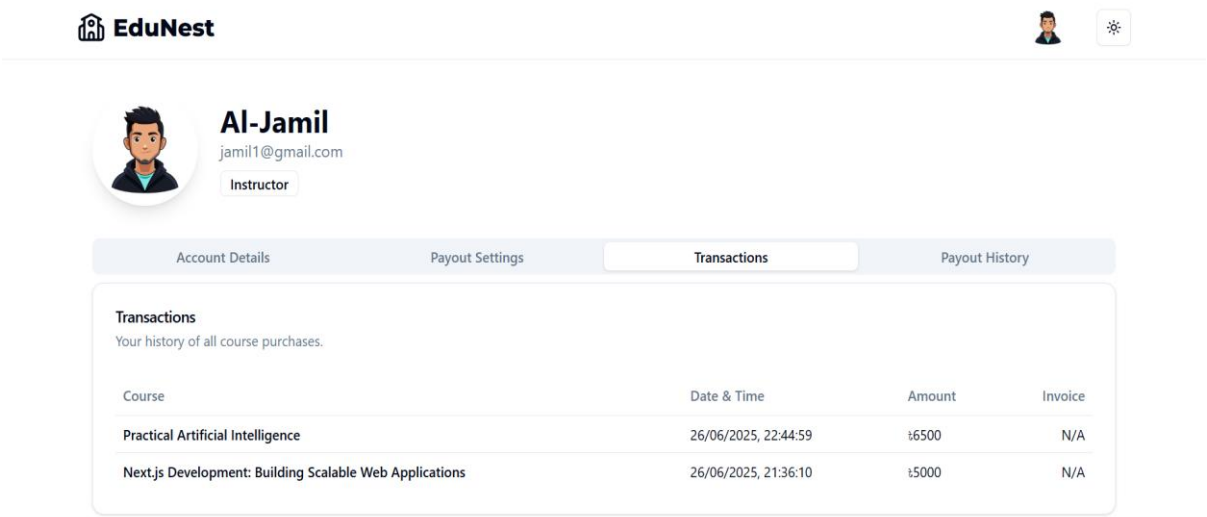


Fig-6.2.12: Transactions Page (For Instructor)

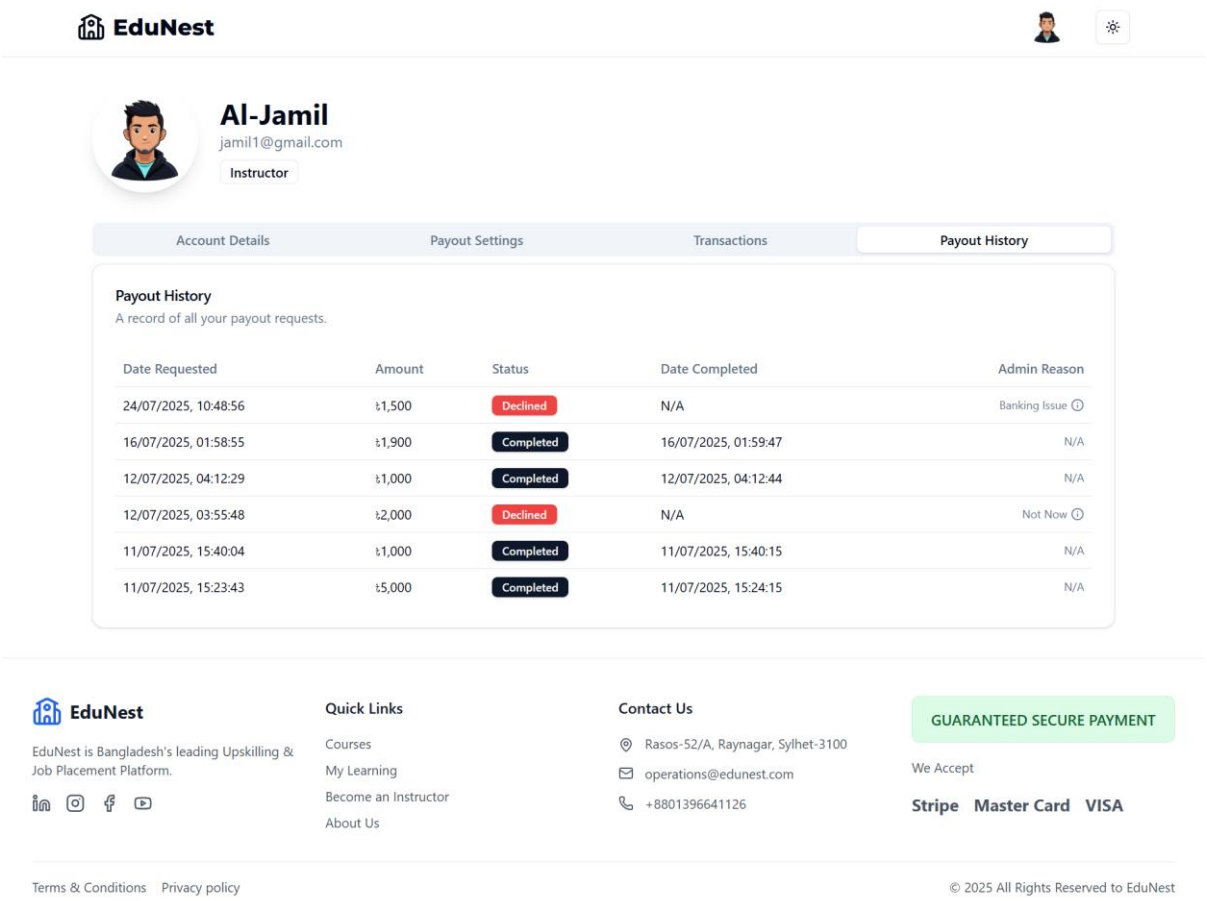


Fig-6.2.13: Payout History Page (For Instructor)

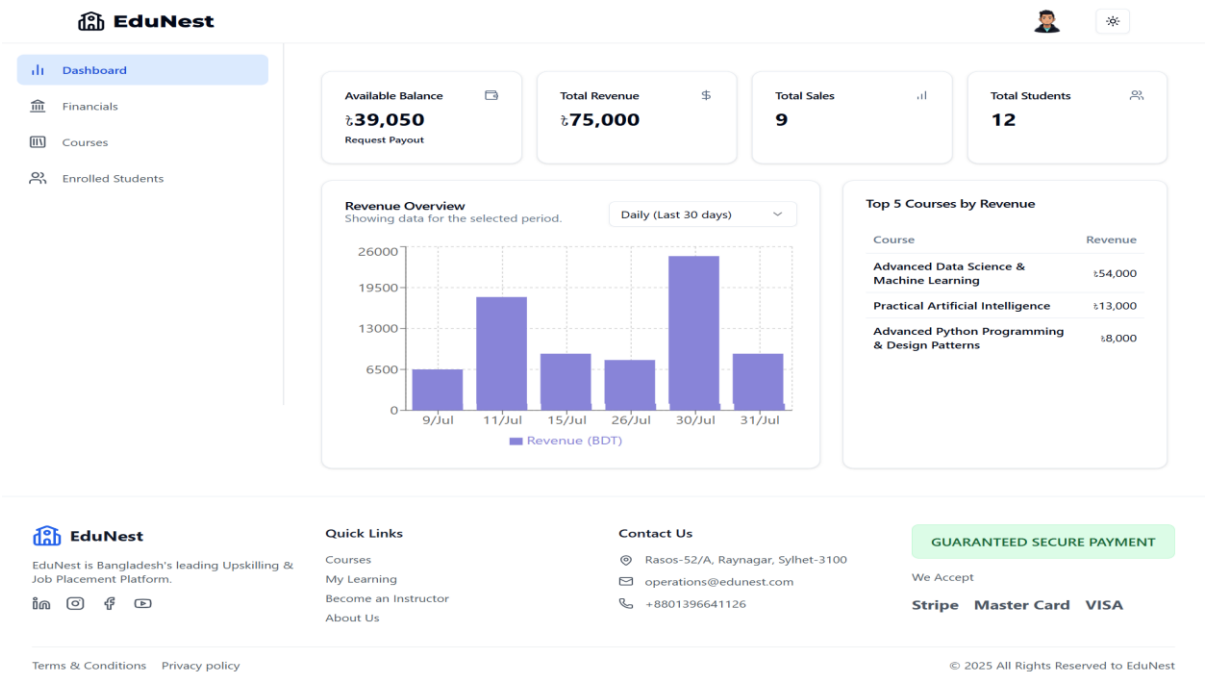


Fig-6.2.14: Dashboard Page (For Instructor)

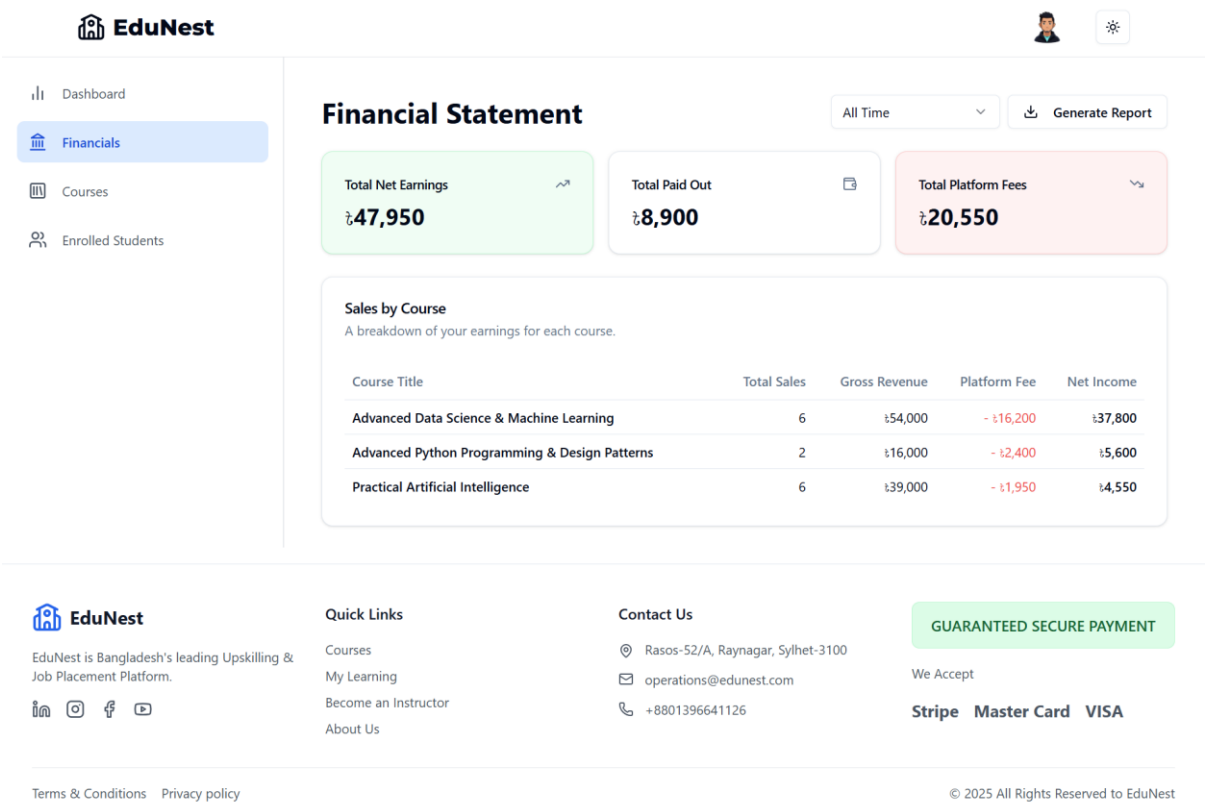


Fig-6.2.15: Financial Dashboard Page (For Instructor)

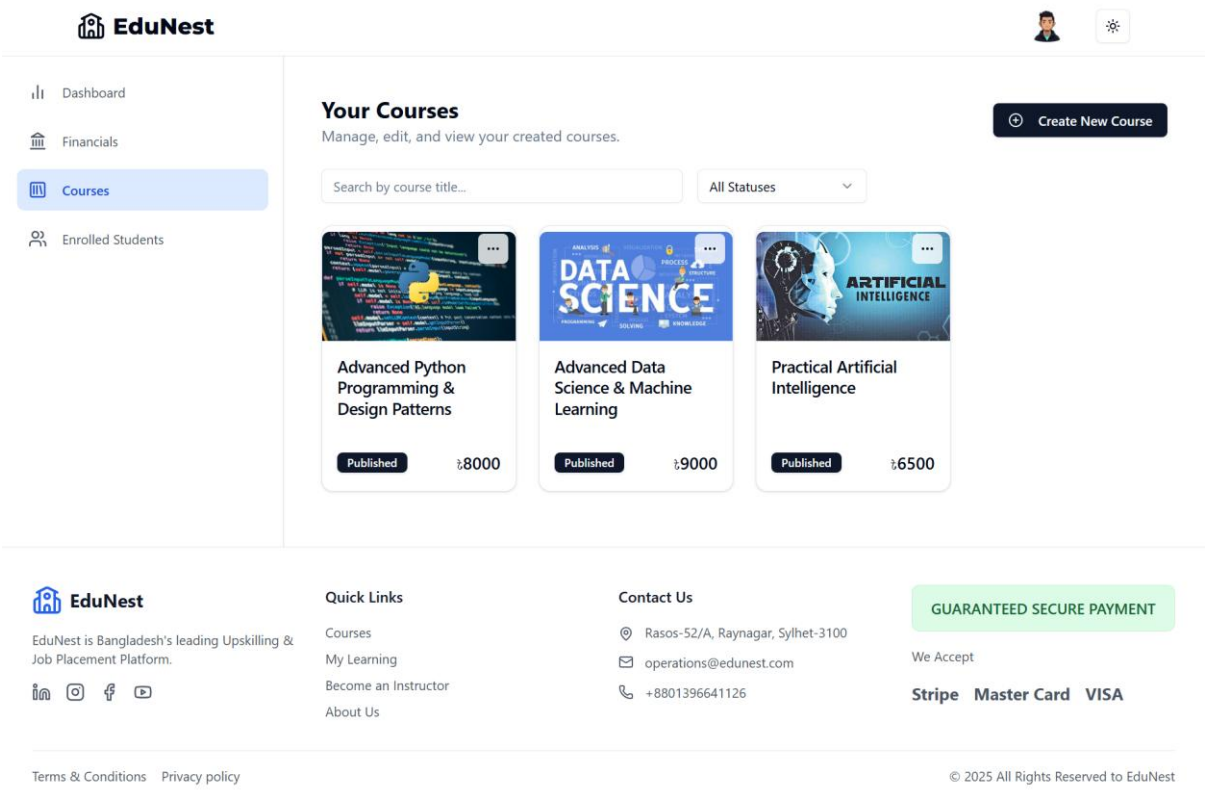


Fig-6.2.16: Course Management (For Instructor)

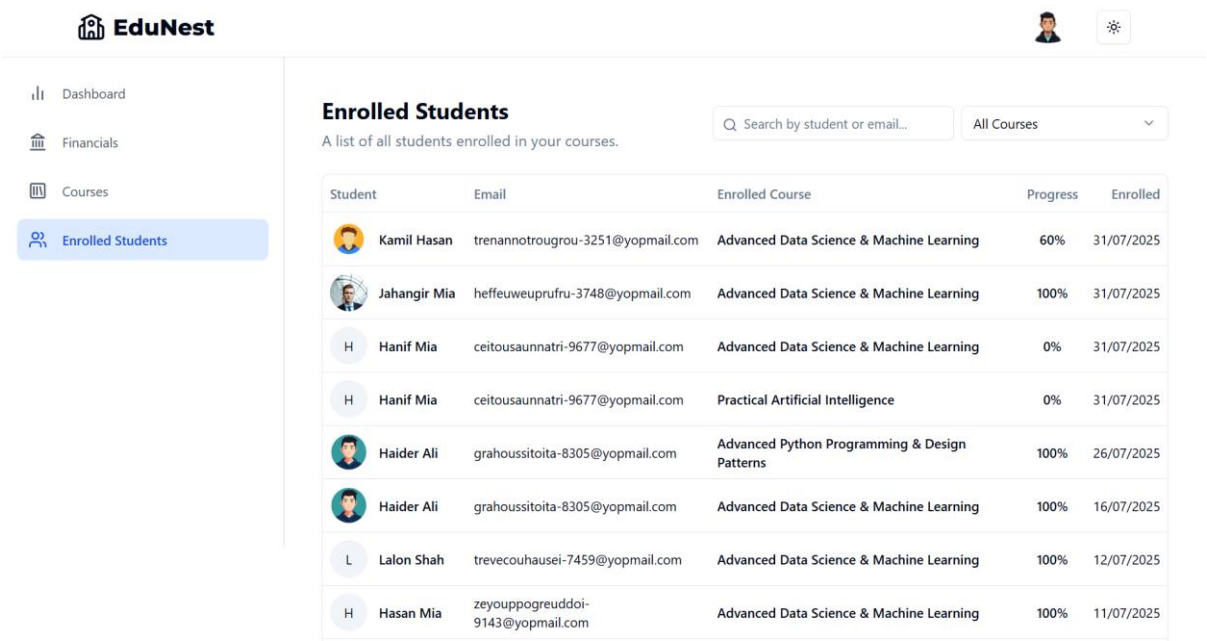


Fig-6.2.17: Enrolled Student Details (For Instructor)

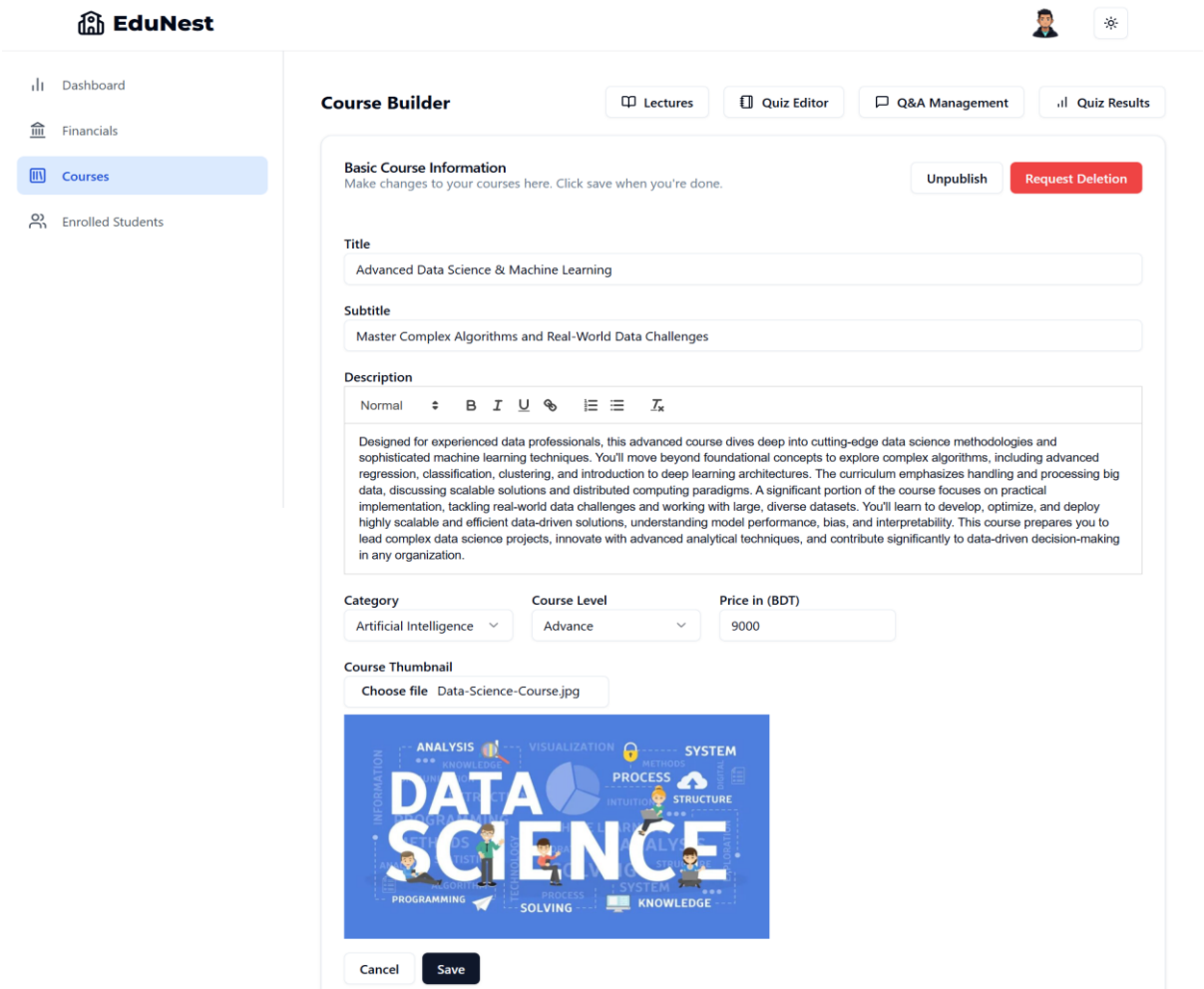


Fig-6.2.18: Course Builder Page (For Instructor)

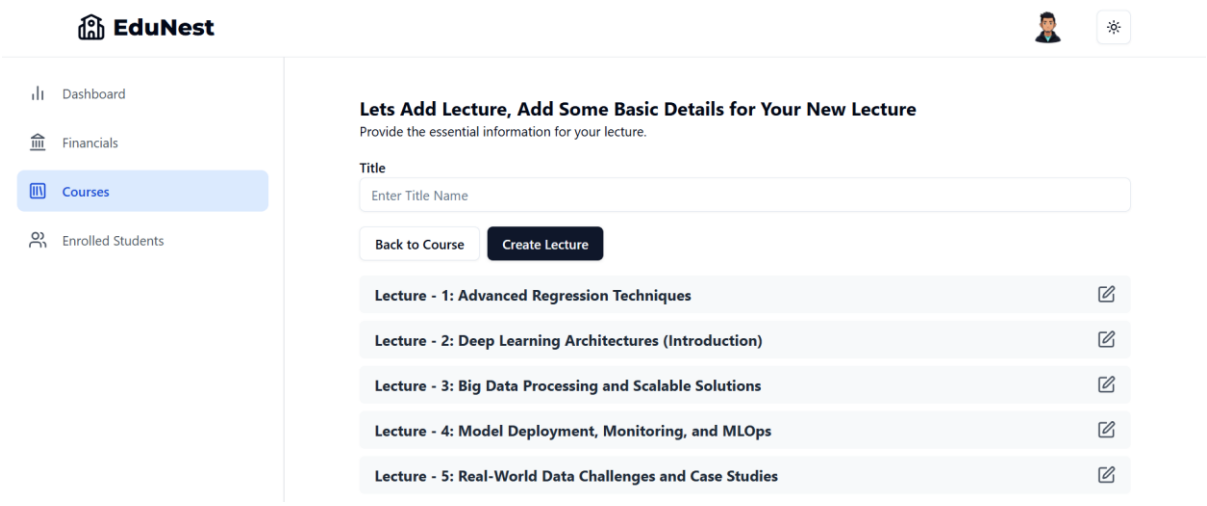


Fig-6.2.19: Lecture Management System (For Instructor)

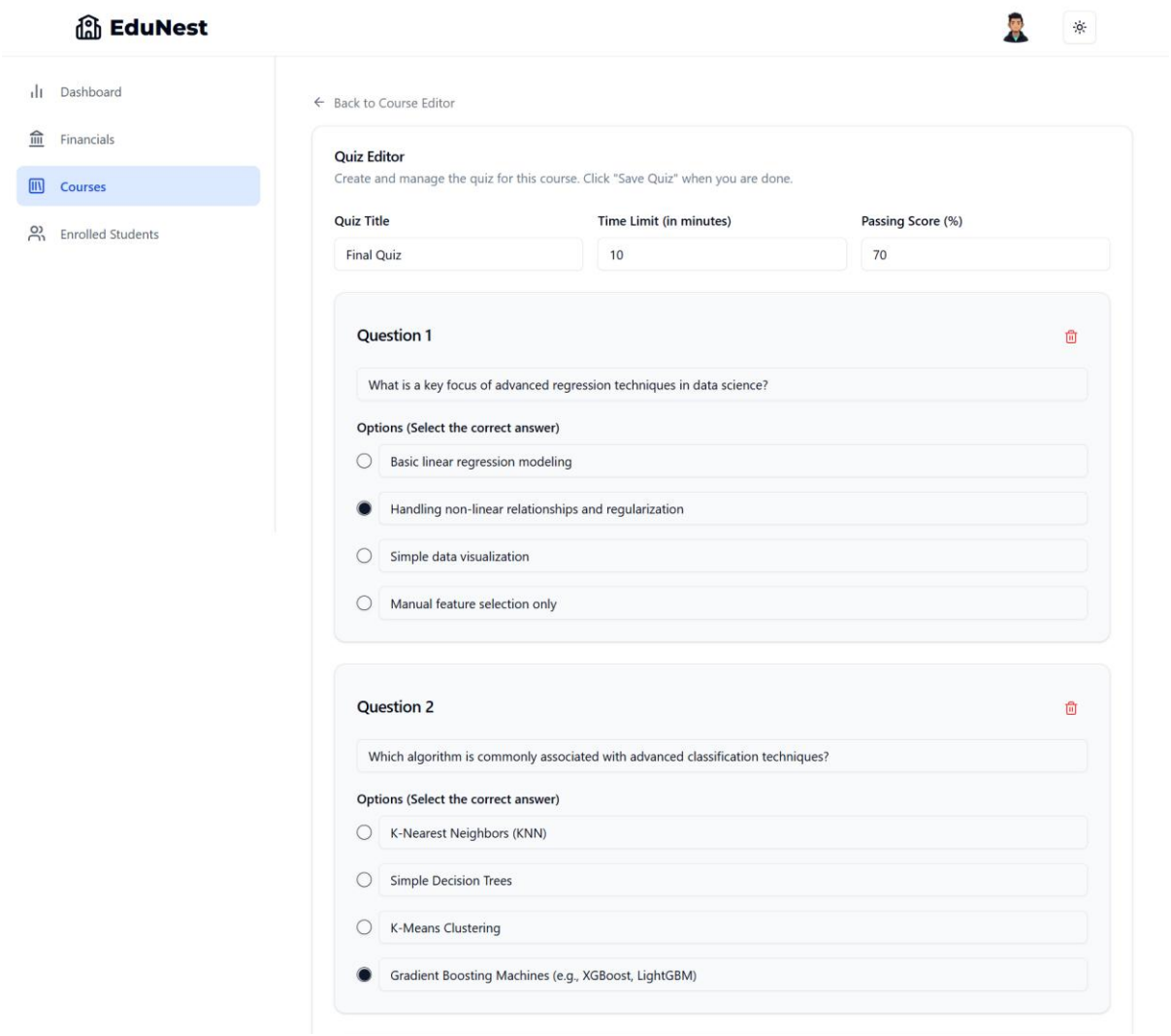


Fig-6.2.20: Quiz Management Page (For Instructor)

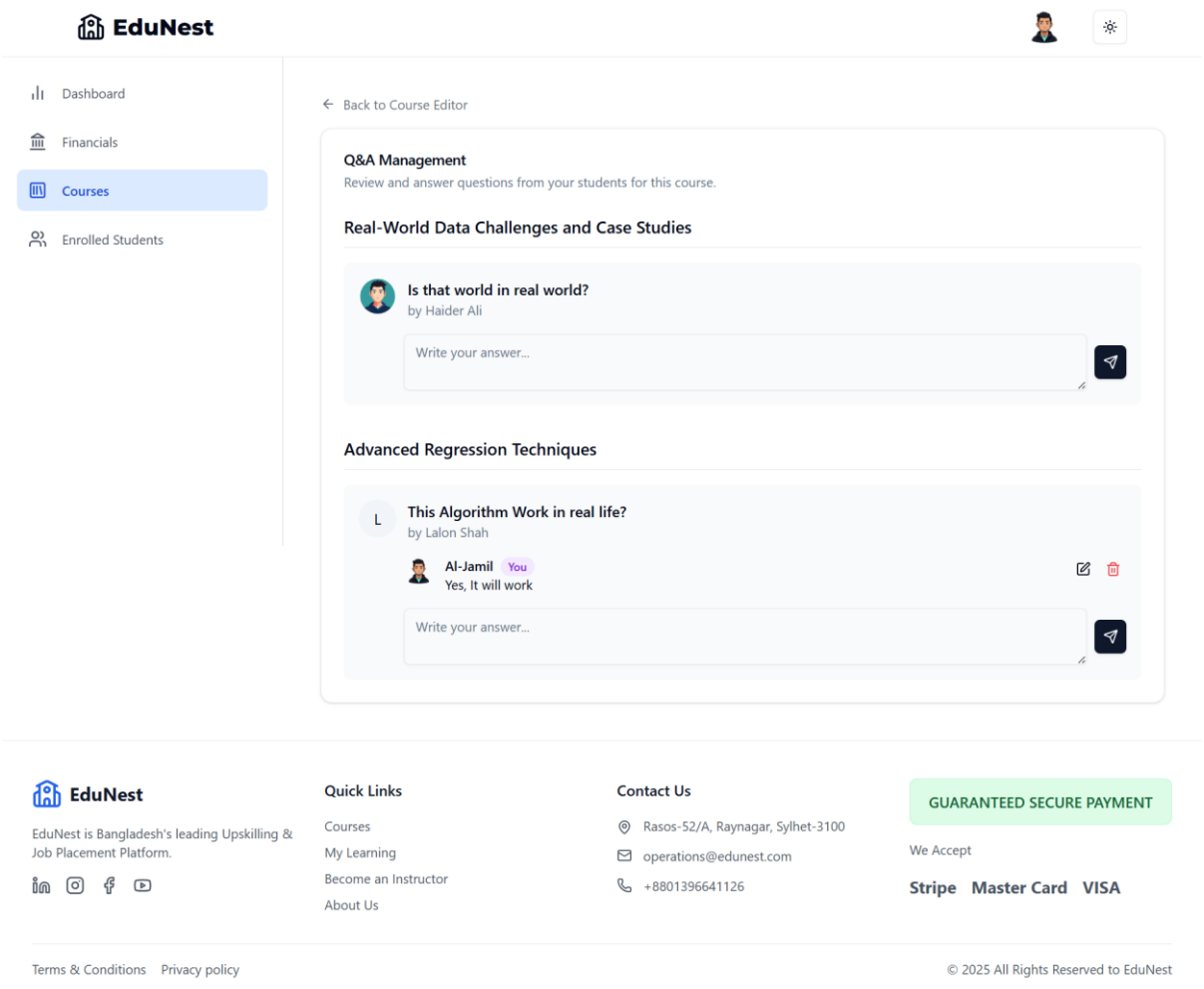
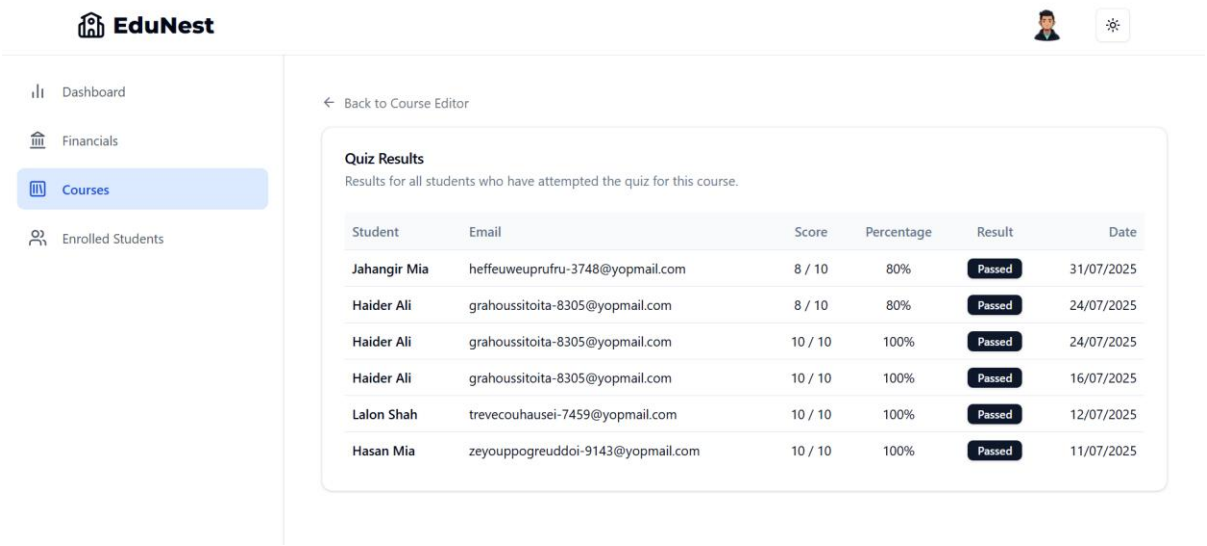


Fig-6.2.21: QnA Management Page (For Instructor)



Student	Email	Score	Percentage	Result	Date
Jahangir Mia	heffeuweuprufru-3748@yopmail.com	8 / 10	80%	Passed	31/07/2025
Haider Ali	grahoussitoita-8305@yopmail.com	8 / 10	80%	Passed	24/07/2025
Haider Ali	grahoussitoita-8305@yopmail.com	10 / 10	100%	Passed	24/07/2025
Haider Ali	grahoussitoita-8305@yopmail.com	10 / 10	100%	Passed	16/07/2025
Lalon Shah	trevecouhausei-7459@yopmail.com	10 / 10	100%	Passed	12/07/2025
Hasan Mia	zeyouppogreuddoi-9143@yopmail.com	10 / 10	100%	Passed	11/07/2025

Fig-6.2.22: Quiz Result Page (For Instructor)

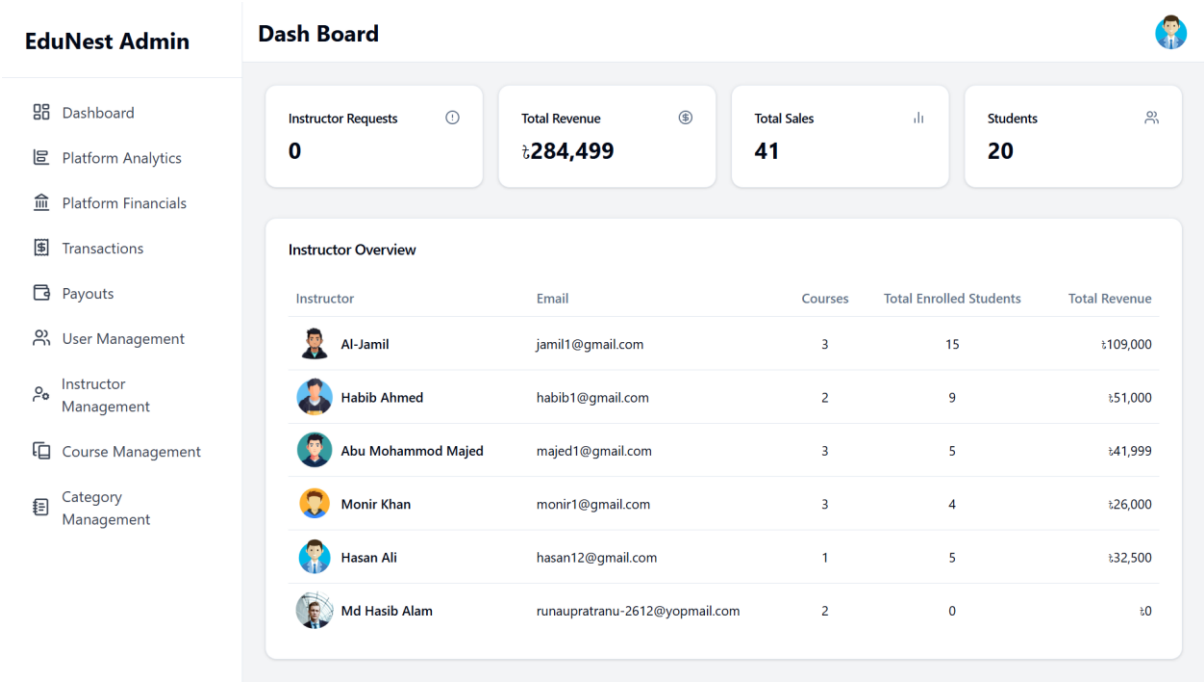


Fig-6.2.23: Admin Panel

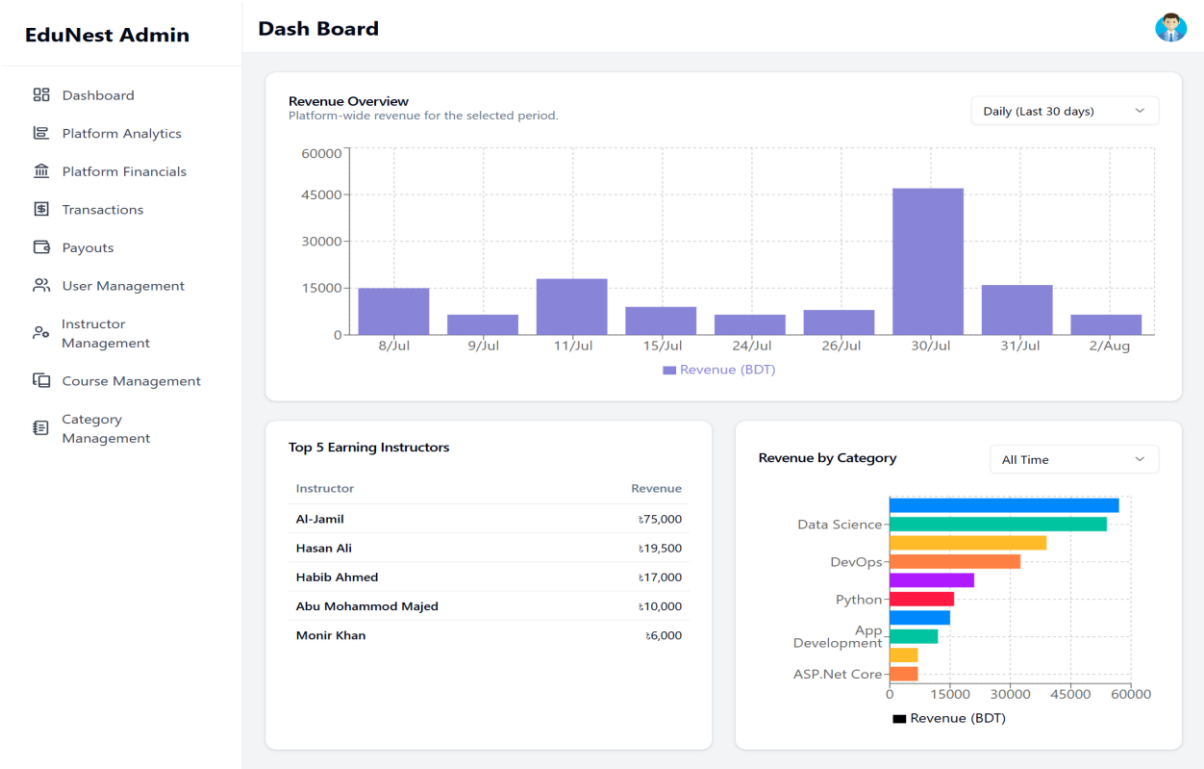


Fig-6.2.24: Admin Panel (Platform Analytics)

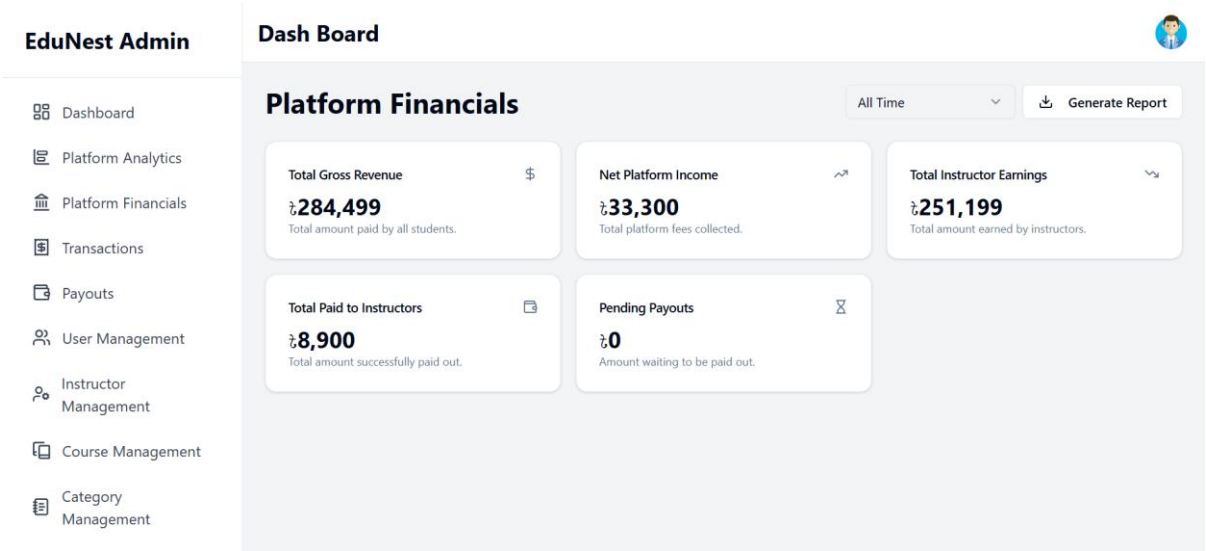


Fig-6.2.25: Platform Financials (Admin Side)

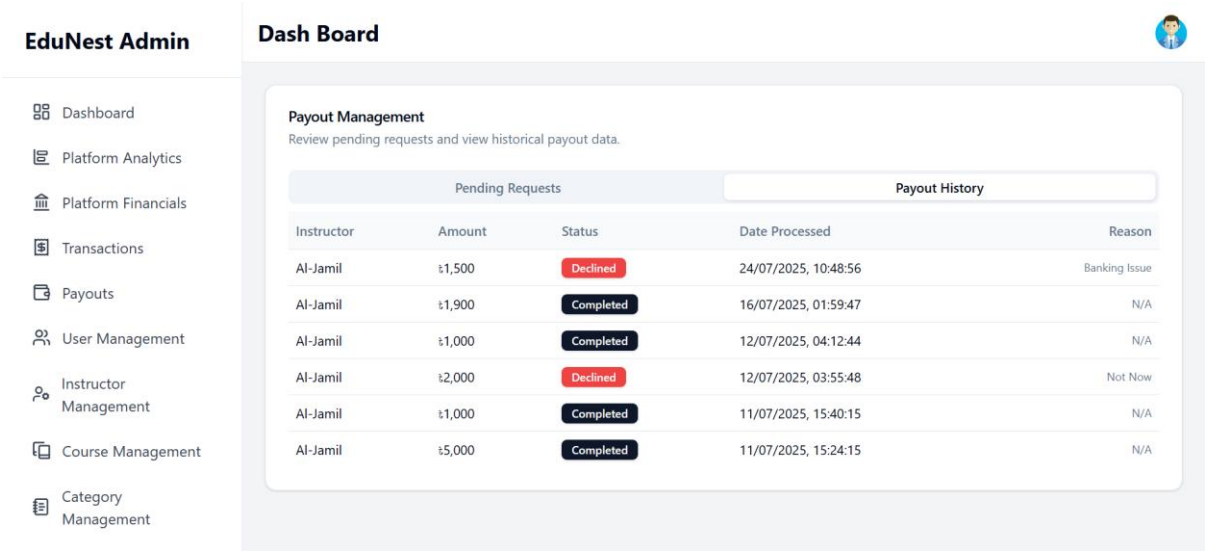
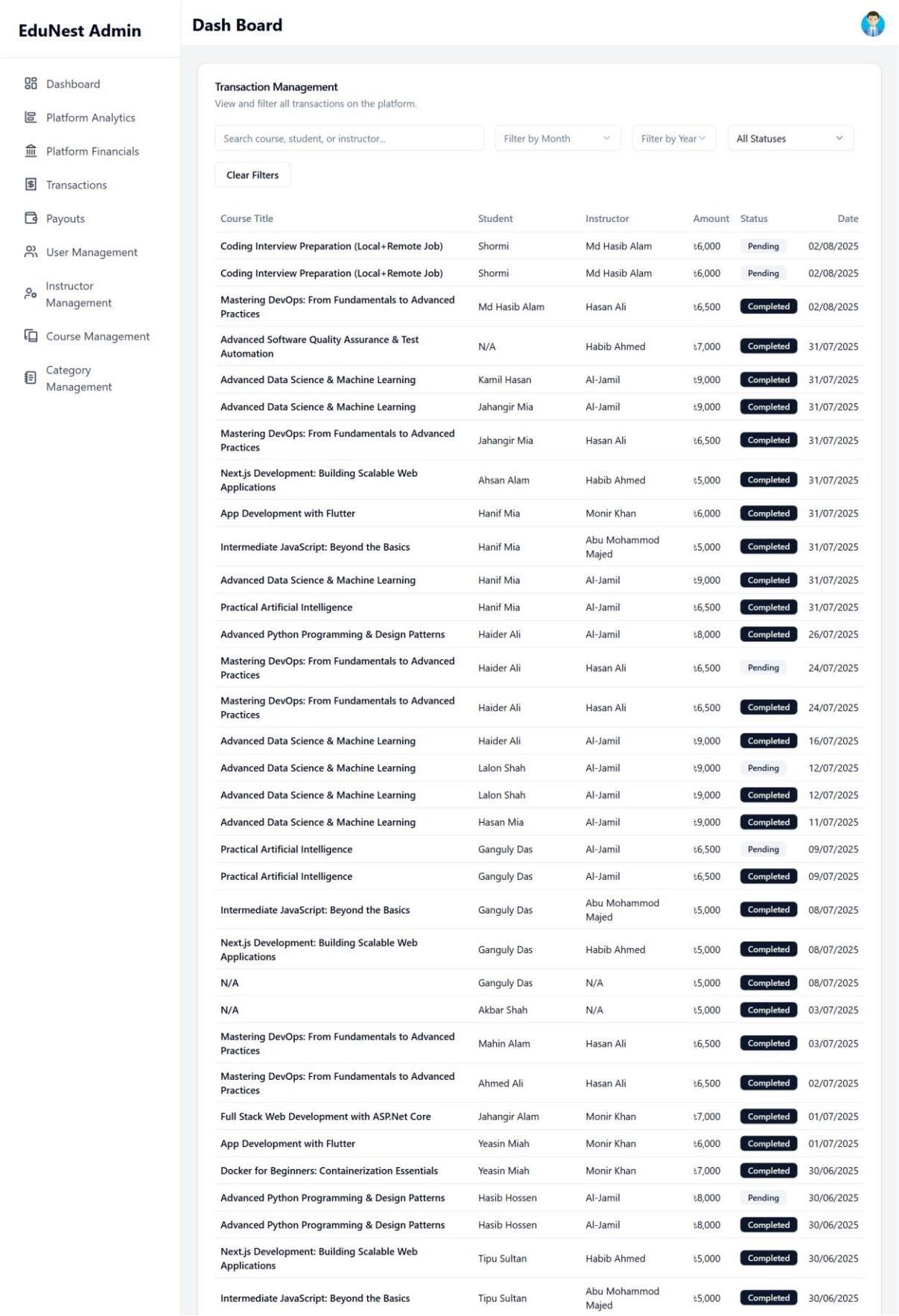


Fig-6.2.26: Payout Management



Course Title	Student	Instructor	Amount	Status	Date
Coding Interview Preparation (Local+Remote Job)	Shormi	Md Hasib Alam	৳6,000	Pending	02/08/2025
Coding Interview Preparation (Local+Remote Job)	Shormi	Md Hasib Alam	৳6,000	Pending	02/08/2025
Mastering DevOps: From Fundamentals to Advanced Practices	Md Hasib Alam	Hasan Ali	৳6,500	Completed	02/08/2025
Advanced Software Quality Assurance & Test Automation	N/A	Habib Ahmed	৳7,000	Completed	31/07/2025
Advanced Data Science & Machine Learning	Kamil Hasan	Al-Jamil	৳9,000	Completed	31/07/2025
Advanced Data Science & Machine Learning	Jahangir Mia	Al-Jamil	৳9,000	Completed	31/07/2025
Mastering DevOps: From Fundamentals to Advanced Practices	Jahangir Mia	Hasan Ali	৳6,500	Completed	31/07/2025
Next.js Development: Building Scalable Web Applications	Ahsan Alam	Habib Ahmed	৳5,000	Completed	31/07/2025
App Development with Flutter	Hanif Mia	Monir Khan	৳6,000	Completed	31/07/2025
Intermediate JavaScript: Beyond the Basics	Hanif Mia	Abu Mohammod Majed	৳5,000	Completed	31/07/2025
Advanced Data Science & Machine Learning	Hanif Mia	Al-Jamil	৳9,000	Completed	31/07/2025
Practical Artificial Intelligence	Hanif Mia	Al-Jamil	৳6,500	Completed	31/07/2025
Advanced Python Programming & Design Patterns	Haider Ali	Al-Jamil	৳8,000	Completed	26/07/2025
Mastering DevOps: From Fundamentals to Advanced Practices	Haider Ali	Hasan Ali	৳6,500	Pending	24/07/2025
Mastering DevOps: From Fundamentals to Advanced Practices	Haider Ali	Hasan Ali	৳6,500	Completed	24/07/2025
Advanced Data Science & Machine Learning	Haider Ali	Al-Jamil	৳9,000	Completed	16/07/2025
Advanced Data Science & Machine Learning	Lalon Shah	Al-Jamil	৳9,000	Pending	12/07/2025
Advanced Data Science & Machine Learning	Lalon Shah	Al-Jamil	৳9,000	Completed	12/07/2025
Advanced Data Science & Machine Learning	Hasan Mia	Al-Jamil	৳9,000	Completed	11/07/2025
Practical Artificial Intelligence	Ganguly Das	Al-Jamil	৳6,500	Pending	09/07/2025
Practical Artificial Intelligence	Ganguly Das	Al-Jamil	৳6,500	Completed	09/07/2025
Intermediate JavaScript: Beyond the Basics	Ganguly Das	Abu Mohammod Majed	৳5,000	Completed	08/07/2025
Next.js Development: Building Scalable Web Applications	Ganguly Das	Habib Ahmed	৳5,000	Completed	08/07/2025
N/A	Ganguly Das	N/A	৳5,000	Completed	08/07/2025
N/A	Akbar Shah	N/A	৳5,000	Completed	03/07/2025
Mastering DevOps: From Fundamentals to Advanced Practices	Mahin Alam	Hasan Ali	৳6,500	Completed	03/07/2025
Mastering DevOps: From Fundamentals to Advanced Practices	Ahmed Ali	Hasan Ali	৳6,500	Completed	02/07/2025
Full Stack Web Development with ASP.Net Core	Jahangir Alam	Monir Khan	৳7,000	Completed	01/07/2025
App Development with Flutter	Yeasin Miah	Monir Khan	৳6,000	Completed	01/07/2025
Docker for Beginners: Containerization Essentials	Yeasin Miah	Monir Khan	৳7,000	Completed	30/06/2025
Advanced Python Programming & Design Patterns	Hasib Hossen	Al-Jamil	৳8,000	Pending	30/06/2025
Advanced Python Programming & Design Patterns	Hasib Hossen	Al-Jamil	৳8,000	Completed	30/06/2025
Next.js Development: Building Scalable Web Applications	Tipu Sultan	Habib Ahmed	৳5,000	Completed	30/06/2025
Intermediate JavaScript: Beyond the Basics	Tipu Sultan	Abu Mohammod Majed	৳5,000	Completed	30/06/2025

Fig-6.2.27: Transactions Analytics

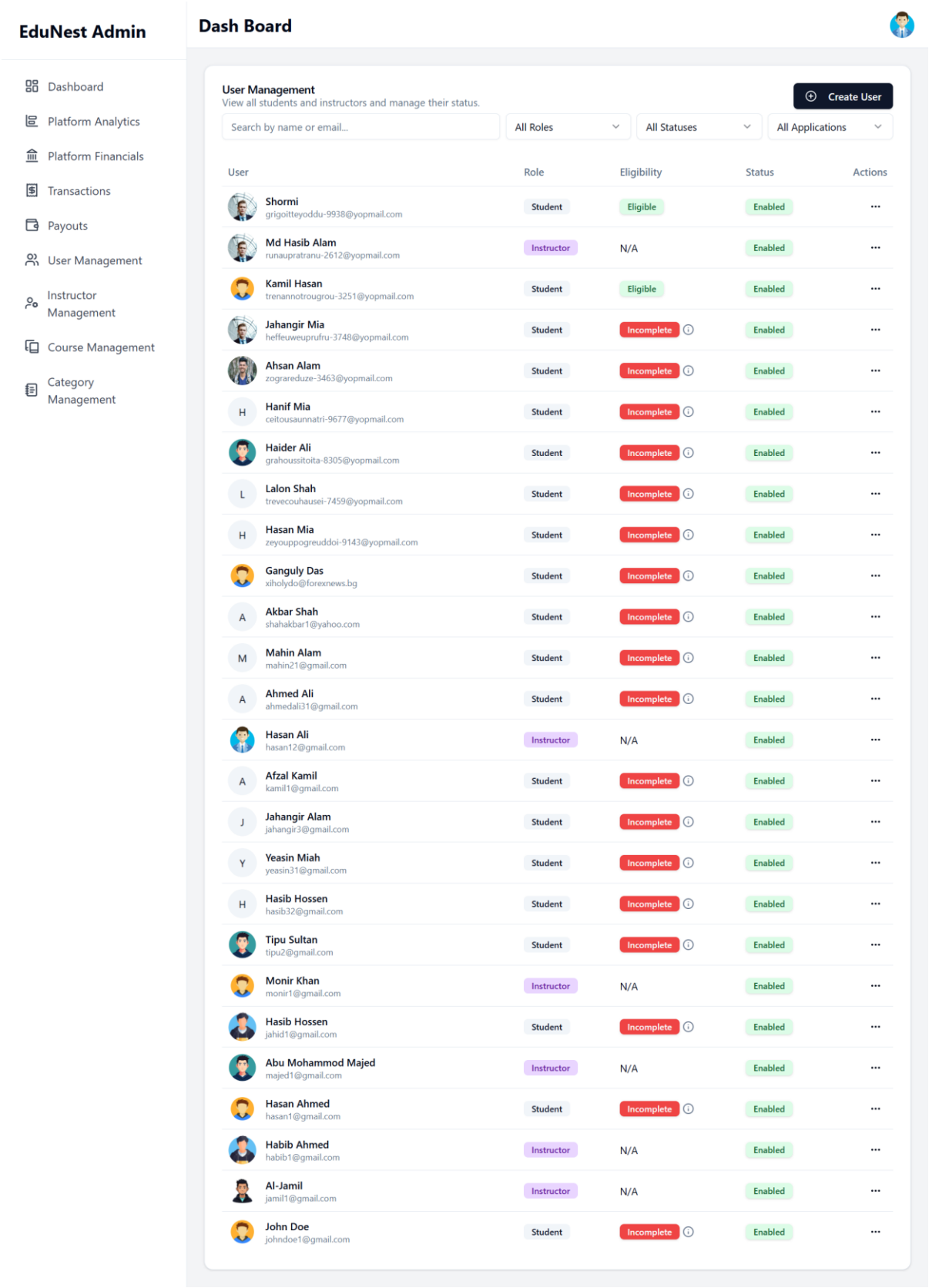


Fig-6.2.28: User Management

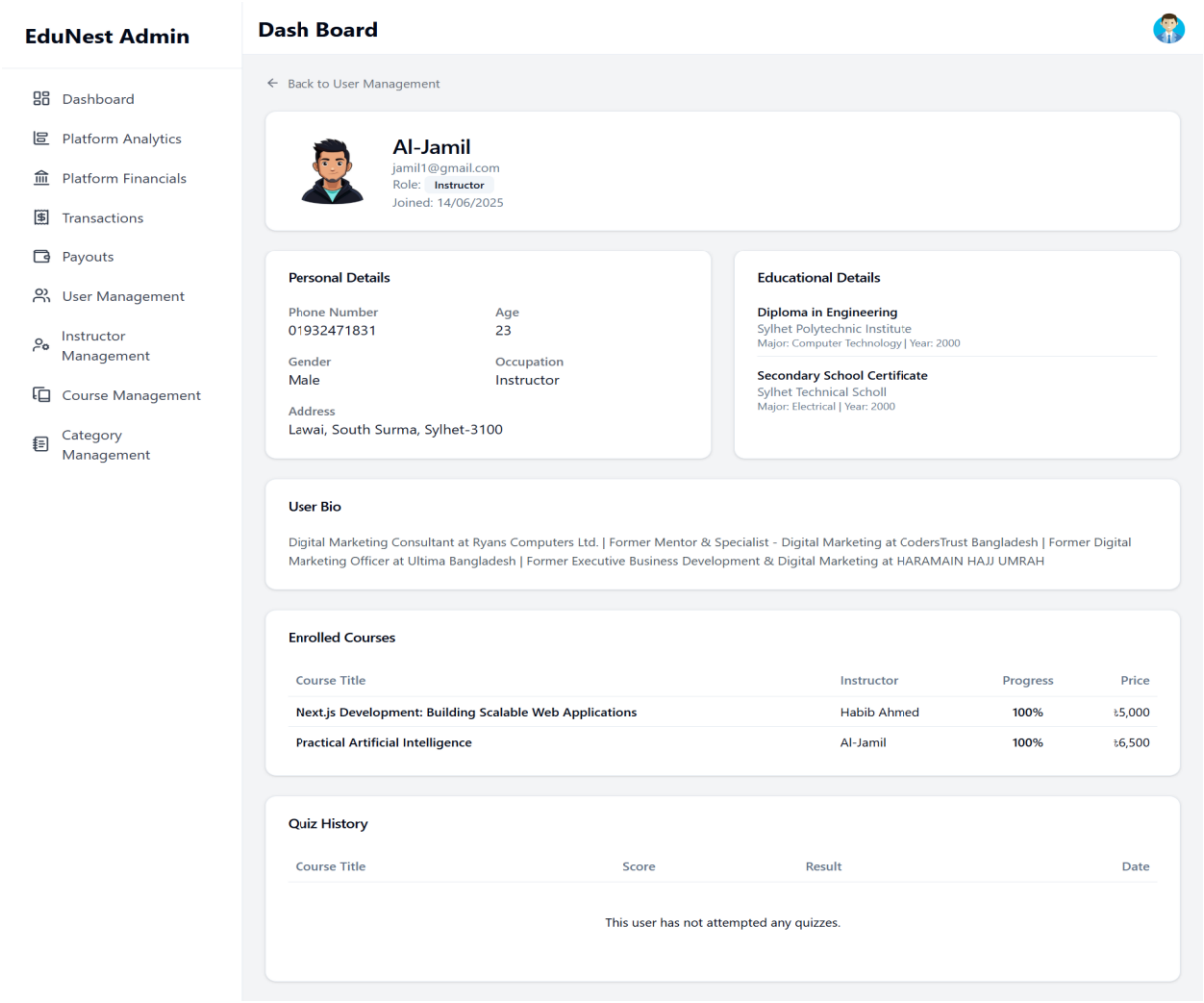


Fig-6.2.29: User Details from Admin Side (Student/Instrcutor)

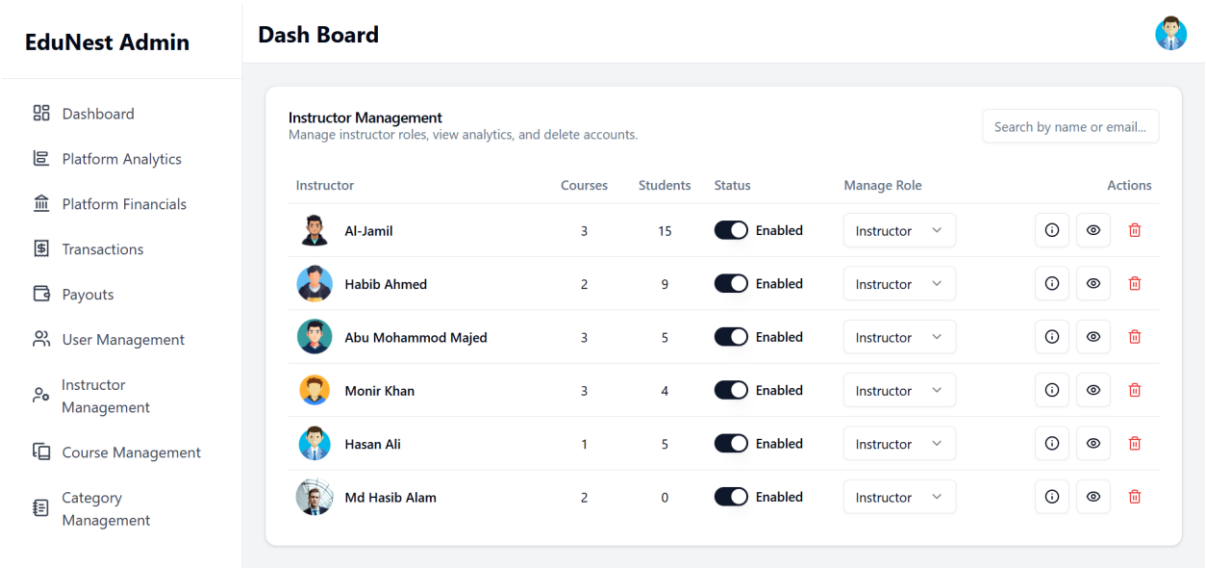


Fig-6.2.30: Instructor Management

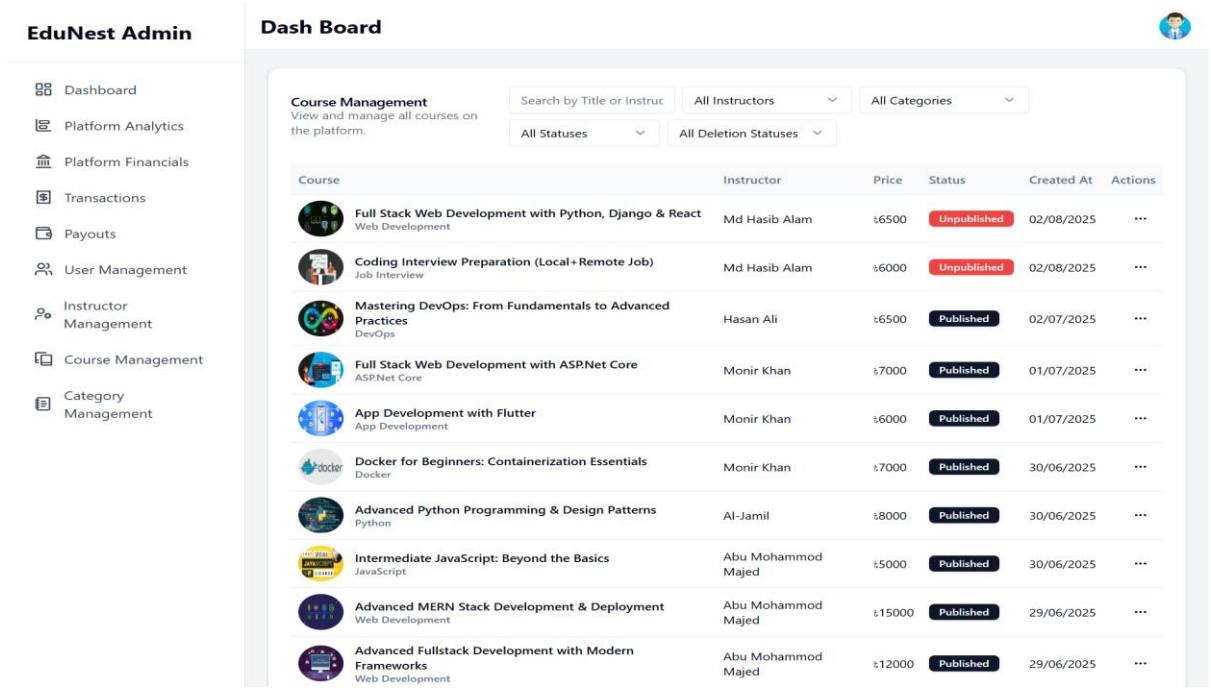


Fig-6.2.31: Course Management (Admin Side)

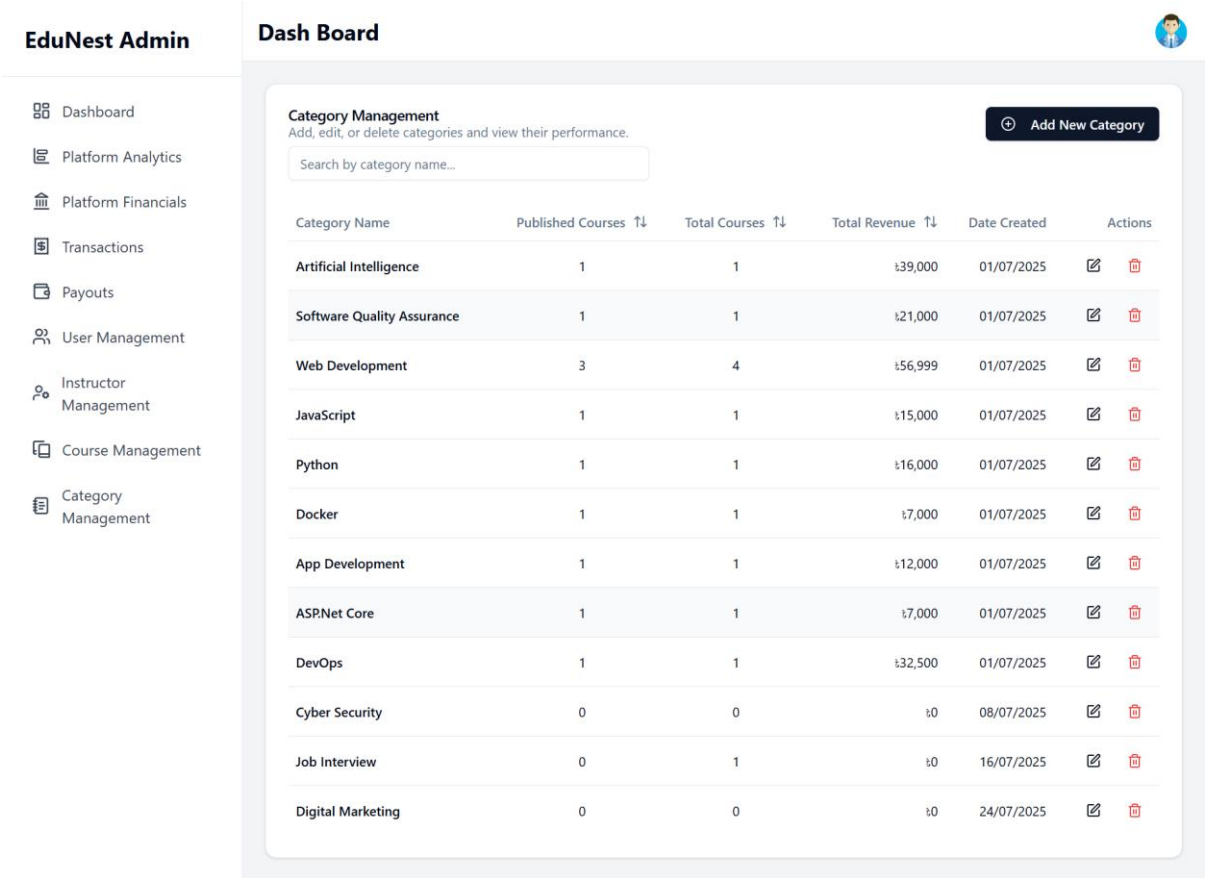


Fig-6.2.32: Category Management

Chapter Seven

System Testing and QA Integration

Contents:

- Testing Approach
- Testing Methodologies
- Industry-Standard QA Test Cases

7.1 Testing Approach

Testing plays a vital role in the software development lifecycle, making sure that the final product is dependable, functional, and free from any bugs. For the **EduNest** project, we implemented a comprehensive testing strategy to ensure the application was validated at every stage of development. Our approach included:

Unit Testing: We tested the smallest individual components of the application in isolation.

- **Backend:** This involved testing individual controller functions (e.g., the register function in user.controller.js) to ensure they correctly handle requests, perform validation, and interact with the database models.
- **Frontend:** We tested individual React components (e.g., the BuyCourseButton component's loading and disabled states), and Redux slices (e.g., the state changes in authSlice.js) to confirm they behaved as expected.

Integration Testing: This phase focused on verifying that different parts of the application work together correctly.

- A key area was testing the communication between the frontend and backend. For example, we tested the entire login flow, ensuring that a useLoginUserMutation call from the frontend's authApi.js successfully authenticated against the backend's /login route, which in turn generated a JWT token, set a cookie, and updated the Redux store on the client-side.

End-to-End (E2E) Testing: We simulated real user scenarios from start to finish to test the complete workflow of the application. This involved testing entire user journeys, such as a student registering, purchasing a course, and watching a lecture, or an instructor creating a new course from their dashboard.

7.2 Testing Methodologies

To ensure comprehensive coverage, we employed both black-box and white-box testing methodologies.

- **White-Box Testing:** Also known as structural testing, this was conducted by us, the developers, who have full knowledge of the codebase.

Example: We performed code walkthroughs of critical backend logic, such as the stripeWebhook controller function, to verify that it correctly handles the checkout.session.completed event, updates the CoursePurchase status, adds the course to the user's enrolled courses, and updates the instructor's balance.

- **Black-Box Testing:** Also known as behavioral testing, this was performed from an end-user's perspective without any knowledge of the internal code structure.

Example: We tested the "Forgot Password" functionality by entering a valid email and expecting to see a success message indicating a reset link was sent. The internal logic of token generation and email sending was not considered, only the final, observable outcome.

7.3 Industry-Standard QA Test Cases

To make our testing process more effective, we created detailed test cases that focus on the application's key features. Each test case outlines a clear set of instructions to check a specific function. Below, you'll find examples tailored for each user role.

Test Case ID: EN-001 (Student)

Feature: Course Purchase and Enrollment Flow

Priority: High

Title: Verify a logged-in student can successfully purchase a course and see it in their "My Learning" section.

Preconditions:

- A valid, logged-in student account exists.
- The course to be tested is published and has a price.
- The student is not already enrolled in the course.
- Valid Stripe test credit card details are available.

Test Steps:

- Navigate to a course detail page and click the "Purchase Course" button.
- User is redirected to the Stripe checkout page. Verify course title and price are correct.
- Fill in all required fields on the Stripe page and click "Pay".
- After successful payment, verify the user is redirected to the Course Progress page.
- Navigate to the "My Learning" page and confirm the newly purchased course is visible.
- Navigate to the "Profile" to "Transactions" tab and verify a new transaction record exists with an option to download the invoice.

Pass/Fail Criteria: Pass if all expected results are met; otherwise, fail.

Test Case ID: EN-002 (Instructor)

Feature: Course Creation and Management

Priority: High

Title: Verify an instructor can create a new course, add a video lecture, and publish it.

Preconditions: A valid, logged-in instructor account exists.

Test Steps:

- From the instructor dashboard, navigate to "Courses" and click "Create New Course."
- Enter a title and select a category, then click "Create Course."
- On the "Course Builder" page, add a lecture and upload a video file.
- Click the "Publish" button.

Expected Results:

- The course is initially created as a "Draft."
- The video lecture is successfully uploaded to Cloudinary and associated with the course.
- After publishing, the course status changes to "Published" and it becomes visible on the main site.

Test Case ID: EN-003 (Admin)

Feature: User Management

Priority: High

Title: Verify an admin can approve a student's pending application to become an instructor.

Preconditions: A logged-in admin account and a student account with instructorApplicationStatus: "pending" exists.

Test Steps:

- Navigate to "User Management" in the admin panel.
- Locate the user with the pending application.
- Use the "Actions" menu to change the user's role to "Instructor."

Expected Results:

- The user's role is successfully updated in the database and on the UI.
- The "Pending" status badge is removed.
- The user receives an email notification confirming their role has been upgraded to Instructor.

Chapter Eight

System Documentation

Contents:

- Discussion and Conclusion
- Limitations and Outcome
- Future Scope

8.1 Discussion and Conclusion

Building EduNest has been an incredible journey filled with both rewards and challenges. We kicked things off with a bold idea, and through thoughtful planning and hard work, we've turned it into a fully functional application. This project really pushed us to get a grip on a range of modern web technologies, from creating a secure backend API using Node.js to designing a dynamic and responsive frontend with React and Redux. We take great pride in having developed a platform that's user-friendly for students, powerful for instructors, and easy to manage for administrators.

8.2 Limitations and Outcome

While we are proud of what we have built, no project is without its limitations.

- Currently, EduNest is a web-only platform; we have not developed dedicated mobile apps for iOS or Android.
- While our analytics are detailed, they could be expanded further with more advanced data visualizations and reporting features.

8.3 Future Scope

EduNest was built with the future in mind. There are many exciting features we would love to add in the future, such as:

- **Live Classes:** Integrating a feature for instructors to host live video sessions.
- **Community Forums:** Creating a space for students to interact with each other and discuss course topics.
- **Mobile Apps:** Developing dedicated mobile applications for an even better learning experience on the go.
- **Advanced AI Features:** Using artificial intelligence to recommend courses to students based on their interests and learning history.

References:

- Lemay DJ, Baek C, Doleck T. Comparison of learning analytics and educational datamining: A topic modeling approach. Comput Edu: Artif Intell. 2021;2:100016.
- Ahmed M, Ahmad MS, Yusoff MZM. A collaborative framework for multiagent systems. Int J Agent Technol Syst (IJATS). 2011;3:1-18.
- Durnali M. Managing the Utilization of Technologies in Adult Education, Training, and Administration: The Case Study of Turkish MoNE. in Utilizing Technology, Knowledge, and Smart
- Systems in Educational Administration and Leadership. IGI Global2020; 1-19.
- S. A. Bafna, P. D. Dutonde, S. S. Mamidwar, M. S. Korvate & D. Shirbhare. (2022). Review on study and usage of mern stack. International Journal for Research in Applied Science & Engineering Technology, 10(II).
- MongoDB Documentation: https://mongodb.com/docs/
- React.js Documentation: https://reactjs.org/docs/getting-started.html
- Node.js Documentation: https://nodejs.org/en/docs/
- Stripe API Documentation: https://stripe.com/docs
- MDN Web Docs: https://developer.mozilla.org
- Express.js Documentation: https://expressjs.com/
- Stripe API Documentation: https://stripe.com/docs
- Tailwind CSS Documentation: https://tailwindcss.com/docs/
- Redux Toolkit Documentation: https://redux-toolkit.js.org/