A

PROJECT PHASE II

REPORT ON

"Abhinav Eduverse (An E-Learning Platform)"

Submitted to the

Dr. Babasaheb Ambedkar Technological University Lonere, Raigad

/

In fulfillment of the requirements

for the award of the degree

BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING

2024-2025

BY

Shaikh Ashraf Ismail	**2263181242505**
Shirnal Apoorva Vasudeo	**2263181242509**
Golande Pradnya Sunil	**2263181242512**

UNDER THE GUIDANCE OF

Prof. Miss. Anikta Rajeshirke

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ABHINAV EDUCATION SOCIETY 'S
COLLAGE OF ENGINEERING AND TECHNOLOGY

ABHINAV EDUCATION SOCIETY 'S
COLLAGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project Report entitled **"Abhinav Eduverse (An E-Leaning Platform)",** which is being submitted by, **Shaikh Ashraf Ismail, Shirnal Apoorva Vasudeo, Golande Pradnya Sunil** as partial fulfillment for the **Degree Bachelor of Technology** (Computer Science and Engineering) of **DBATU, Lonere.**

This is bonafide work carried under my supervision and guidance.

Place: Pune

Date:

Prof. Miss. Rajeshirke
Project Guide

Prof. Dipak Chavan
Project I/C

Prof. Sagar Morade
Head of Department

Dr. P. J. Patil
Principal

External Examiner [Name & Sign]

SEAL

ACKNOWLEDGEMENT

We would like to express our deep sense of gratitude and sincere thanks to everyone who supported and guided us during the development of our project, **"Abhinav Eduverse – A Full Stack E-Learning Platform."**

We express our heartfelt appreciation to our respected guide, **Prof. Miss. Ankita Rajeshirke**, for her constant guidance, encouragement, and valuable insights throughout the course of this project. Her support has played a crucial role in helping us shape our ideas and bring them into reality.

We are also grateful to our respected **Head of Department, Prof. Sagar Morade**, for his continuous support and for providing all necessary facilities and motivation to complete the project successfully.

We sincerely thank all the faculty members of the **Computer Science and Engineering Department** at **Abhinav Education Society's College of Engineering and Technology (Degree), Wadwadi**, for their invaluable teaching and support throughout our academic journey.

We are especially thankful to our Principal, **Dr. Prashant J. Patil**, for his encouragement, valuable suggestions, and consistent support in ensuring academic excellence.

Last but not least, we would like to thank our families and friends for their patience, motivation, and moral support during every stage of this project.

Team Members:
Shaikh Ashraf Ismail
Shirnal Apoorva Vasudeo
Golande Pardnya Sunil

ABSTRACT

The rapid growth of digital learning platforms has significantly transformed the landscape of modern education. Traditional classroom-based methods are increasingly being complemented — and in many cases replaced — by scalable, interactive, and accessible online learning systems. In this context, we have developed **Abhinav Eduverse**, a comprehensive full-stack E-learning platform tailored to bridge the gap between learners and educators in a virtual environment.

Abhinav Eduverse provides a dual-role system: **teachers** can create and manage courses, add educational modules in the form of videos or lessons, and view enrolled student data, while **students** can browse available courses, enroll, view structured modules, and track their personal learning progress. The application is designed to offer a seamless, intuitive, and engaging user experience for both ends.

The **frontend** of the system is built using modern web technologies ensuring responsiveness and smooth interaction. The **backend** is powered by Spring Boot, enabling a robust, scalable, and secure API-driven architecture. The system follows the MVC design pattern and emphasizes the use of **DTOs** to maintain clean data transactions between layers, improving maintainability and reducing complexity. A **MySQL** database has been used to ensure efficient data persistence and relational integrity.

Key features of the platform include secure user authentication, course/module CRUD operations, video playback integration, progress visualization, and role-based access control. The application also allows teachers to reorder modules, giving them complete control over the flow of course content. Each student's learning progress is tracked and visually represented, promoting engagement and self-paced learning.

The goal of this project is not only to deliver an academic solution but also to offer a scalable foundation that can evolve into a real-world educational platform. Through the design and implementation of Abhinav Eduverse, we have gained practical experience in full-stack development, API integration, responsive UI design, and project lifecycle management.

TABLE OF CONTENTS

CH 1 : SYNOPSIS

1.1 Project Title

Abhinav Eduverse – A Full Stack E-Learning Platform

1.2 Internal Guide

Prof. Miss. Ankita Rajeshirke
Department of Computer Science & Engineering
Abhinav Education Society's College of Engineering and Technology (Degree)

1.3 Technical Words

- Full Stack Development
- React.js
- Spring Boot
- REST API
- DTO (Data Transfer Object)
- MySQL
- Authentication
- Role-Based Access
- Module Management
- Responsive UI
- MVC Architecture
- State Management
- Video Streaming

1.4 Relevant Objectives

- To build a complete E-learning web platform for students and teachers.
- To implement secure role-based login and access control.
- To allow teachers to create, update, and manage course content.
- To enable students to enroll in courses, view videos, and track progress.
- To build a responsive frontend with intuitive navigation and UI.
- To use DTOs and RESTful APIs for clean backend communication.
- To maintain modularity and scalability in system design.

1.5 Motivation

Online learning has become a major part of modern education, especially in remote areas and post-pandemic scenarios. However, many platforms are either too complex or lack personalization for teachers and students. We wanted to create a simple, powerful, and user-friendly learning environment that can serve both students and educators with minimal friction.

Abhinav Eduverse aims to solve these challenges by offering an intuitive interface, flexible module creation, and real-time learning progress tracking — all built using industry-standard tools and technologies.

1.6 Existing System

Current online learning platforms such as Google Classroom, Coursera, and Udemy offer similar functionalities, but they are often either too enterprise-focused, expensive, or lack control from an educator's point of view.

Common drawbacks of these systems include:

- Lack of flexibility in module ordering and control
- Limited tracking of student progress in custom-built courses
- Overwhelming interface or high learning curve
- Minimal backend customization for developers or institutions

Abhinav Eduverse overcomes these issues by offering a custom-built solution with essential E-learning features, intuitive dashboards, and full control over content creation and course flow.

1.7 Problem Statement

In the current educational landscape, traditional classroom methods and basic online tools are insufficient for providing personalized and interactive learning experiences. Most E-learning systems either lack scalability, ease of use, or the ability to track detailed learner progress. Educators also face limitations in customizing course structures or controlling module sequencing.

Additionally, many platforms are expensive, require heavy configuration, or do not provide backend flexibility for enhancements. This gap creates a need for a flexible, modern, and easy-to-use learning system.

1.8 Solving Approach

To address these limitations, we designed and developed **Abhinav Eduverse**, a full-stack web-based E-learning platform that serves both students and educators.

Our solution includes:

- A role-based login system that separates teacher and student dashboards.
- A course creation and module management interface for teachers.
- A student-friendly dashboard with progress tracking and video playback.
- Responsive and intuitive UI using modern frontend tools.
- A secure and scalable backend using Spring Boot and REST APIs.
- Clean backend communication using Data Transfer Objects (DTOs) for optimized data handling.

The system architecture is built with modularity in mind, allowing for future upgrades such as quizzes, discussion forums, and notifications.

CH 2 : INTRODUCTION

2.1 Background and Motivation

Education stands as one of the most fundamental pillars of any society, driving personal growth, economic development, and cultural advancement. Over the past decade, the educational landscape has experienced a profound transformation, shifting away from traditional classroom-based learning towards more dynamic, flexible, and technology-driven methods. This shift has been largely catalyzed by the rapid proliferation of internet access and the ubiquitous use of smart devices, which together enable learners and educators to connect anytime and anywhere.

The emergence of e-learning platforms has become instrumental in facilitating this new mode of education. These platforms provide opportunities for learners to engage with course material interactively and at their own pace, while also offering instructors tools to efficiently manage and deliver content. E-learning is particularly significant in today's world, where remote education is not only convenient but often necessary due to geographic, economic, or health-related constraints.

Despite the rapid adoption of online learning solutions, many existing e-learning platforms tend to fall short in addressing the specific and diverse needs of their users. Common challenges include overly generalized content that lacks personalization, high subscription costs that limit accessibility, and cumbersome user interfaces that complicate navigation. For educators, managing course content, monitoring student progress, and maintaining learner engagement can be overwhelming without the right set of tools. Conversely, students often seek platforms that are intuitive, customizable, and that provide meaningful feedback and progress tracking to help them stay motivated and on course.

2.2 Project Overview: Abhinav Eduverse

To bridge this gap, the Abhinav Eduverse project was conceptualized and developed as a comprehensive full-stack e-learning web application. The primary objective of the system is to deliver a robust, scalable, and user-centric educational platform that caters distinctly to the needs of both students and teachers through role-based access and tailored functionalities.

In Abhinav Eduverse, teachers have the capability to design and manage courses effortlessly. This includes adding instructional videos, creating course modules organized in a flexible sequence, and monitoring student enrollments and performance. On the other hand, students can browse through available courses, enroll with ease, follow structured learning paths through modules, and track their progress in real-time, empowering them to take control of their educational journey.

The platform has been architected with a focus on responsive design to ensure seamless accessibility across a variety of devices, including desktops, tablets, and smartphones. This accessibility enables learning to continue unhindered regardless of the learner's environment or device preferences.

2.3 Technical Architecture and Implementation

The backend of Abhinav Eduverse is implemented using **Spring Boot**, a widely used Java-based framework known for its scalability, maintainability, and robust ecosystem. This choice provides the foundation for building secure RESTful APIs that serve as the communication bridge between the frontend interface and the underlying database.

For persistent data storage, the system utilizes **MySQL**, a relational database management system that offers strong data consistency, transactional support, and efficient querying capabilities. This allows for reliable storage of user information, course data, enrollment records, and progress metrics.

One of the noteworthy design decisions in the project is the use of **Data Transfer Objects (DTOs)** to decouple the internal data models from the API responses. This separation enhances maintainability, prevents issues such as infinite recursion during serialization, and ensures a clean contract between the backend and frontend layers.

The overall system architecture follows the **Model-View-Controller (MVC)** design pattern. This clear division of responsibilities improves code modularity and maintainability. The backend handles data processing and business logic (Model and Controller), while the frontend focuses on presenting data and interacting with users (View).

Frontend communication with the backend APIs is facilitated using asynchronous HTTP requests, typically with libraries such as Axios, ensuring smooth and responsive user experiences without full page reloads.

2.4 Key Features and Functionalities

Abhinav Eduverse encompasses several core features aimed at creating an effective learning ecosystem:

- **Role-Based Access Control:** Users are authenticated and assigned roles (student or teacher) that govern their access to features and data. Upon login, users are directed to their respective dashboards tailored to their needs.
- **Course and Module Management:** Teachers can create courses, add multiple modules per course, upload instructional videos, and sequence modules through drag-and-drop functionality to structure the learning flow.
- **Student Enrollment and Progress Tracking:** Students can enroll in courses, view detailed module content, and track their progress at a granular module level. Progress indicators help motivate learners and provide transparency on completion status.
- **Secure and Scalable API Integration:** The system employs Spring Security to protect endpoints and data, ensuring that only authorized users can access or modify information.
- **Responsive and Intuitive User Interface:** Designed with simplicity and usability in mind, the interface allows users with varying technical skills to navigate and utilize the platform effectively.

2.5 Significance and Benefits

The development of Abhinav Eduverse goes beyond an academic exercise; it represents a practical contribution toward making education more accessible and manageable in a digital-first world. By addressing the shortcomings of existing platforms, it provides a tailored solution that empowers educators with better course management tools and students with personalized, trackable learning experiences.

This platform also lays the groundwork for future enhancements, such as the integration of artificial intelligence for personalized learning recommendations, social features to encourage peer interaction, and advanced analytics for educators to better understand student behavior.

2.6 Learning Outcomes and Project Experience

Throughout the course of this project, valuable insights and skills were gained in multiple domains of software development. These include:

- Conducting comprehensive system analysis and requirement gathering to align development goals with user needs.
- Applying sound software design principles to create a modular, maintainable, and scalable architecture.
- Developing secure backend services and integrating them with a dynamic frontend interface.
- Implementing database design, query optimization, and transaction management for reliable data storage.
- Performing rigorous testing, deployment, and documentation to ensure a high-quality software product.

This project experience has thus not only resulted in a functional e-learning platform but also reinforced core competencies essential for a successful career in software engineering.

CH 3 : LITERATURE SURVEY

3.1 Introduction

The Literature Survey chapter provides a comprehensive review of existing e-learning platforms, relevant technologies, and the challenges faced in the domain of online education. This helps to understand the current state of the field, identify gaps, and justify the need for developing Abhinav Eduverse as a novel, efficient e-learning solution.

3.2 Existing E-learning Platforms

The rapid growth of e-learning in the last decade has led to the emergence of numerous platforms catering to various educational needs. Some of the prominent platforms include:

- **Moodle:** An open-source learning management system widely used by educational institutions. It provides tools for course management, quizzes, grading, and forums. However, Moodle's interface can be complex for beginners and may require dedicated technical staff for customization.
- **Coursera:** A popular MOOC (Massive Open Online Course) platform offering courses from top universities worldwide. While it provides high-quality content and certifications, it operates mostly on a paid model and has limited flexibility for individual course creators.
- **Udemy:** An online marketplace for courses where instructors can upload content and students purchase access. It is popular for its wide range of topics but lacks a uniform structure, and quality varies across courses.
- **Google Classroom:** A free platform integrated with Google's suite of apps, facilitating assignment distribution and grading. It is effective for classroom collaboration but lacks advanced features for large-scale course management or multimedia content delivery.

3.3 Technologies in E-learning Development

Modern e-learning platforms leverage various technologies to create interactive and scalable systems:

- **Frontend Frameworks:** Tools like React.js, Angular, and Vue.js enable dynamic, responsive, and user-friendly interfaces. They allow for component-based development, facilitating reuse and maintainability.
- **Backend Frameworks:** Frameworks such as Spring Boot (Java), Django (Python), and Express.js (Node.js) provide robust APIs and business logic handling. Spring Boot is favored for its modularity, security features, and scalability.
- **Databases:** Relational databases like MySQL and PostgreSQL provide structured data storage with ACID compliance, essential for maintaining course and user data integrity. NoSQL databases (MongoDB, Cassandra) are sometimes used for handling large-scale unstructured data.

- **Multimedia Integration:** Video streaming, interactive quizzes, and real-time chat are critical features supported via technologies like HTML5 video, WebRTC, and RESTful APIs.

3.4 Challenges in Existing Systems

Despite advancements, several challenges persist in the current e-learning ecosystem:

- **Lack of Personalization:** Many platforms offer generic content without adapting to individual learner needs, impacting motivation and effectiveness.
- **Complex User Interfaces:** Educators and learners sometimes find interfaces non-intuitive, limiting platform adoption.
- **Limited Progress Tracking:** Detailed, module-level progress tracking is often missing or inadequate, making it hard to monitor learning paths effectively.
- **Cost Barriers:** Subscription fees and licensing costs restrict access, especially in underserved regions.
- **Security Concerns:** Protecting user data and preventing unauthorized access is critical, yet many platforms lack robust security measures.

3.5 Gap Analysis

The analysis of existing literature and platforms reveals a clear need for an e-learning system that:

- Provides distinct, role-based experiences for teachers and students.
- Supports rich multimedia content with flexible course and module management.
- Offers intuitive progress tracking and personalized learning paths.
- Ensures secure, scalable, and maintainable architecture.
- Is affordable or free to encourage wider accessibility.

3.6 How Abhinav Eduverse Addresses These Gaps

Abhinav Eduverse is designed to bridge these gaps by integrating modern web technologies and best practices:

- **Role-Based Access:** Separate dashboards and functionalities for students and teachers improve usability and operational efficiency.
- **Module Sequencing and Drag-and-Drop:** Teachers can customize course flow, enhancing content delivery.
- **Detailed Progress Tracking:** Students can monitor their learning status at a granular level, improving motivation.
- **Security:** Implementation of Spring Security ensures protected API endpoints and safe data handling.
- **Scalability and Maintainability:** Use of DTOs and MVC architecture facilitates cleaner code and easier future enhancements.

3.7 Summary

This literature survey highlights the evolving nature of e-learning systems and the need for improved, user-focused solutions. By understanding existing technologies and their limitations, the foundation is laid for developing Abhinav Eduverse as a platform that delivers a superior educational experience to both educators and learners.

CH 4 : EXISTING SYSTEM

4.1 Introduction

In today's educational landscape, numerous established e-learning platforms have been developed to facilitate online teaching and learning. These systems range from open-source solutions like Moodle to commercial platforms such as Coursera and Udemy, each designed to meet various educational requirements and audiences. Typically, these platforms provide features such as course creation, content delivery, student enrollment, assessments, and progress tracking through web-based interfaces accessible on multiple devices.

Despite their widespread use and many benefits, these systems share common architectural patterns and feature sets aimed at meeting basic online education needs. Most provide role-based access controls for teachers and students, multimedia support for instructional content, and backend databases for storing user and course data.

4.2 Disadvantages

While existing e-learning systems have transformed education delivery worldwide, they exhibit several limitations affecting usability, flexibility, and scalability:

- **Limited Customization and Flexibility:** Many platforms enforce rigid structures for courses and module sequencing, limiting instructors' ability to tailor the course flow according to pedagogical goals. This reduces engagement and adaptability.
- **Complex User Interfaces:** The user experience on some systems is hindered by cluttered or unintuitive interfaces. Teachers and learners with minimal technical skills may find it difficult to navigate, reducing effective usage.
- **Inadequate Progress Tracking:** Detailed tracking of student progress at the module or lesson level is often insufficient. This lack of granular feedback makes it challenging for both students to gauge their learning and teachers to identify struggling learners.
- **High Cost and Licensing Issues:** Commercial platforms often impose high subscription or licensing fees, which can be prohibitive for learners and educators, especially in developing regions or budget-constrained institutions.
- **Security and Privacy Concerns:** Many platforms do not provide robust mechanisms to safeguard sensitive user information or prevent unauthorized access, raising concerns over data privacy and compliance with regulations.
- **Scalability Constraints:** Some systems are not architected to efficiently support a growing number of users or large multimedia content, resulting in performance degradation during peak usage.
- **Limited Offline Access:** Dependence on stable internet connectivity restricts accessibility for users in areas with unreliable networks, limiting the platform's reach.

4.3 Comparative Analysis

The following table summarizes key features and limitations of popular existing e-learning systems compared to Abhinav Eduverse:

Feature	**Moodle**	**Coursera**	**Udemy**	**Abhinav Eduverse**
Role-Based Access Control	Yes	Limited	Limited	Yes
Progress Tracking	Basic	Moderate	Limited	Detailed Module-Level
Cost	Free/Open Source	Paid	Paid	Free/Open Source
Security Features	Moderate	High	Moderate	High (Spring Security)
Customization Options	Moderate	Low	Low	High
Offline Access	Limited	Limited	Limited	Planned/Extendable

This comparison highlights how Abhinav Eduverse aims to overcome prevalent shortcomings by offering advanced module management, detailed progress tracking, strong security, and flexible customization—all within a scalable, open framework.

4.4 Summary

Existing e-learning platforms have paved the way for accessible education worldwide but still suffer from limitations related to flexibility, usability, cost, and security. These shortcomings motivated the development of Abhinav Eduverse, which strives to provide an adaptable, secure, and user-friendly environment tailored to both educators and learners. By addressing the gaps identified in current systems, Abhinav Eduverse aims to enhance the effectiveness and reach of online education.

CH 5 : PROPOSED SYSTEM

5.1 Introduction

The **Abhinav Eduverse** platform is proposed as a complete and scalable solution for modern e-learning needs. The system addresses the limitations of existing platforms by offering greater flexibility, enhanced role-based functionality, detailed progress tracking, and a secure, responsive interface. Built using full-stack technologies, it provides a robust environment that supports both students and teachers in achieving their goals.

The goal of the proposed system is to simplify course delivery for educators and enhance the learning experience for students. By employing a modular design and following best practices in web development and security, the platform ensures that it is easy to maintain, extend, and use.

The system empowers teachers with tools to create structured, multimedia-rich courses and allows students to learn at their own pace, track their progress, and interact with content in a meaningful way.

5.1.1 Architecture

The architecture of Abhinav Eduverse is designed following the **Three-Tier MVC (Model-View-Controller)** pattern, which promotes clear separation of concerns and enhances scalability.

1. Presentation Layer (Frontend)

This layer is responsible for user interaction. Developed using modern frontend technologies, it includes:

- Dashboards for teachers and students
- Course browsing and enrollment
- Module viewing and video playback
- Responsive design compatible with all devices

2. Business Logic Layer (Backend - Spring Boot)

This layer handles application logic and API processing. Key functions include:

- User authentication and authorization
- Course and module creation and editing
- Enrollment processing and progress management
- Role-based access control
- Secure and efficient RESTful API endpoints

3. Data Layer (MySQL Database)

This layer stores all persistent data. It manages:

- Users, courses, modules, enrollments, and progress
- Relational integrity and indexing for performance
- Clean separation using DTOs (Data Transfer Objects) to prevent serialization issues

Supporting Technologies:

- **Axios (or Fetch):** For asynchronous frontend-backend communication
- **Spring Security:** To secure endpoints and manage user sessions
- **DTO Pattern:** Ensures only required data is exposed to the frontend

The system design supports modularity, reusability, and future enhancement without disrupting existing functionalities.

5.2 Algorithm

The key features of Abhinav Eduverse are implemented through well-structured and efficient logic flows. Below are some core algorithms used in the system:

1. User Authentication and Role Detection

```
if (user != null && password matches) {
    if (user.role == TEACHER) {
        redirect to TeacherDashboard;
    } else if (user.role == STUDENT) {
        redirect to StudentDashboard;
    }
} else {
    show login error;
}
```

Purpose: Ensures only valid users log in and are redirected to appropriate role-based dashboards.

2. **Course Enrollment Check**

```
if (!isEnrolled(studentId, courseId)) {
    enrollStudent(studentId, courseId);
    initializeProgress(studentId, courseModules);
}
```

Purpose: Checks if a student is already enrolled before adding them to a course and sets up progress tracking.

3. Module Progress Tracking

```
for (Module module in courseModules) {
    if (student completes module) {
        updateProgress(studentId, moduleId, COMPLETED);
    }
}
calculateTotalProgress(studentId, courseId);
```

Purpose: Records completed modules and updates the overall course progress percentage.

5. DTO Mapping for API Communication

```
UserDTO dto = new UserDTO();
dto.setId(user.getId());
dto.setName(user.getName());
// Do not expose password or entity relationships
return dto;
```

Purpose: Separates entity logic from API responses to ensure clean, safe, and efficient data exchange.

5.3 Summary

The proposed system—**Abhinav Eduverse**—is a well-structured, secure, and responsive e-learning platform. It resolves common problems found in existing systems by providing:

- Role-based dashboards
- Flexible module and course management
- Detailed, real-time progress tracking
- Secure API access
- Intuitive and mobile-friendly UI

The architectural design, combined with efficient algorithms and modern technologies, makes Abhinav Eduverse a highly scalable and user-centric platform ready for real-world use.

CH 6 : DISSERTATION PLAN

6.1 Purpose of Document

The purpose of this dissertation plan is to present a comprehensive overview of the planning, execution, and management of the project titled **Abhinav Eduverse – A Full-Stack E-learning Platform**. This chapter outlines the strategic approach followed throughout the development lifecycle of the project, covering key aspects such as resource utilization, estimated vs actual effort, risk handling, scheduling, and feasibility.

As the project has been successfully completed, this section also serves to evaluate the effectiveness of the initial plan in guiding the development process and ensuring timely, quality delivery.

6.2 Dissertation Estimates

The project was executed in a modular and phased manner, following a clear schedule with well-defined deliverables at each stage. Time was allocated based on complexity and dependencies. The actual effort was periodically reviewed and adjusted, ensuring efficient use of time and resources.

6.2.1 Reconciled Estimates

Task/Phase	Estimated Duration	Actual Duration	Remarks
Requirement Analysis	5 days	4 days	Completed ahead of schedule
System Design	5 days	6 days	Minor adjustments made during implementation
Database Design	4 days	4 days	On schedule
Frontend Development	10 days	12 days	Extra time spent on responsiveness and UI
Backend Development	10 days	11 days	Included setup of Spring Security and DTOs
Integration & Testing	6 days	6 days	Completed as planned
Final Documentation	5 days	5 days	Final touches added after testing

Total Estimated Duration: 45 days

Total Actual Duration: 48 days

The project was completed within a reasonable margin of the original estimate, demonstrating good planning and effective time management.

6.2.2 Project Resources

The development of the Abhinav Eduverse platform was accomplished using readily available resources and standard tools. These resources supported full-stack development, testing, and deployment.

Resource	Description	Status
Laptop	Personal laptop (Intel i5, 8GB RAM)	Utilized
Frontend Tools	Visual Studio Code, HTML5, CSS3, JavaScript, React	Used
Backend Tools	Spring Boot, Spring Security, REST API development	Used
Database	MySQL with structured schema for users, courses, modules, enrollments	Configured
Testing Tools	Postman for API testing, browser dev tools	Used
Version Control	Git & GitHub for code repository management	Maintained
UI Design Tools	Draw.io, Canva for flowcharts and screen design	Utilized
Time Commitment	~3–4 hours daily during the project timeline	Achieved
Internet Access	Stable broadband used throughout development	Reliable

All project resources were fully utilized and helped ensure the successful and timely delivery of the system.

6.3 Risk Management

Risk management is an essential part of software project planning to anticipate, identify, and minimize the impact of possible issues that could affect the project's success. Although the development of **Abhinav Eduverse** was completed successfully, this section documents the potential risks that were considered during the process, along with mitigation strategies.

6.3.1 Risk Identification

During the planning phase, several possible risks were identified that could have impacted the progress, timeline, or quality of the final product. These included:

Risk ID	Description	Type
R1	Delay in requirement gathering or scope creep	Schedule Risk
R2	Learning curve of new tools (Spring Boot, DTOs)	Technical Risk
R3	API integration issues or communication failures	Integration Risk
R4	Data loss or corruption due to accidental errors	Data Risk
R5	Poor internet or hardware failure	Resource Risk
R6	Incomplete testing or undetected bugs	Quality Risk

6.3.2 Risk Analysis

Each identified risk was assessed based on its probability of occurrence and potential impact on the project. Mitigation strategies were put in place to reduce their effect.

Risk ID	Probability	Impact	Mitigation Strategy
R1	Medium	Medium	Set clear scope early, weekly review of tasks
R2	High	Medium	Allocate extra time for tool learning, refer to documentation
R3	Medium	High	Test API endpoints individually using Postman, ensure clear contracts
R4	Low	High	Regular backups, use of Git for version control
R5	Low	Medium	Use offline development tools, have access to alternate resources
R6	Medium	High	Perform end-to-end testing, write unit tests, and collect user feedback

These risk mitigation strategies ensured that none of the risks caused major disruptions during the development lifecycle. The actual project experienced minimal issues, thanks to proactive planning and regular reviews.

6.4 Dissertation Schedule

The success of a software project depends heavily on how well tasks are scheduled, dependencies are managed, and timelines are tracked. In this section, the planning and scheduling followed during the execution of **Abhinav Eduverse** are presented.

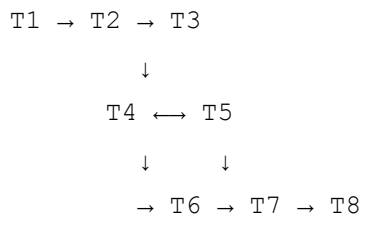
6.4.1 Project Task Set

The entire project was broken down into manageable tasks, each assigned a specific duration and order based on priority and dependency.

Task ID	Task Name	Duration	Dependency
T1	Requirement Gathering	4 days	—
T2	System Design	5 days	T1
T3	Database Design	4 days	T2
T4	Frontend Development	12 days	T2, T3
T5	Backend Development	11 days	T3
T6	API Integration	3 days	T4, T5
T7	Testing and Debugging	6 days	T6
T8	Final Documentation	5 days	T7

6.4.2 Task Network

The project followed a **sequential and modular task network**, as outlined below:



This dependency diagram ensured efficient task management. Parallel development was performed for frontend and backend components after the design phase to reduce total development time.

6.4.3 Project Plan

Below is the high-level **Gantt-style timeline** representing project progress across 6 weeks.

Week	Tasks
1	T1: Requirement Gathering
2	T2: System Design, T3: Database Design
3	T4: Frontend Start, T5: Backend Start
4	T4 & T5 continued
5	T6: API Integration, T7: Testing
6	T8: Final Documentation and Report Compilation

The plan was followed closely, with slight adjustments made for frontend responsiveness and testing feedback.

6.4.4 Management Reporting & Communication

Project progress was monitored using:

- Daily logs of tasks completed and pending
- Git commits and repository history for development tracking
- Weekly summaries of completed milestones
- Personal notes for troubleshooting and backlog planning

Communication with peers and mentors was maintained through regular updates, code reviews, and documentation sharing.

6.5 Feasibility Study

The feasibility study is a critical component of project planning. It assesses whether the proposed solution is practical, sustainable, and beneficial within the given constraints. For **Abhinav Eduverse**, the project was evaluated based on **technical**, **operational**, and **economic** feasibility — all of which were found to be **favorable**.

6.5.1 Technical Feasibility

- The technologies chosen for development — **Spring Boot**, **MySQL**, **HTML/CSS/JavaScript**, and **React** (or other frontend tools used) — are well-supported, widely adopted, and suitable for building scalable web applications.
- Development was carried out on standard hardware with open-source tools, requiring no additional infrastructure investment.
- The use of RESTful APIs, DTOs, and a secure MVC-based architecture ensured efficient data flow, modularity, and easy debugging.
- The development environment (Spring Tool Suite, VS Code, Postman, GitHub) proved sufficient for building and testing the application.

Conclusion: The system is technically feasible and was developed successfully using all planned tools and technologies.

6.5.2 Operational Feasibility

- The platform was designed with role-based access (Student/Teacher), ensuring a smooth, intuitive experience for users with varying technical backgrounds.
- Features like video integration, progress tracking, and drag-and-drop module ordering were implemented successfully and tested.
- The UI was designed to be responsive and user-friendly, ensuring accessibility across devices.
- Teachers were able to create, edit, and manage content, while students could enroll and complete courses — all within a few clicks.

Conclusion: The application is fully functional and user operations were tested to confirm smooth workflows.

6.5.3 Economic Feasibility

- The project was developed using **free and open-source tools**, eliminating the need for licensing costs or third-party subscriptions.
- No special hardware or hosting charges were incurred during development.
- The development was performed independently, so no labor costs were involved.
- In future deployments, the cost remains low as the platform can be hosted on free tiers of services like Render, Railway, or locally for institutional use.

Conclusion: The project is economically viable and sustainable for real-world use or deployment in academic environments.

CH 7 : SOFTWARE REQUIREMENT SPECIFICATION (SRS)

7.1 Functional Requirements

Functional requirements define the core capabilities and system behavior expected from the software. For **Abhinav Eduverse**, the following key functions were identified and implemented:

- User Registration and Authentication
- Role-based Dashboards (Student / Teacher)
- Course Creation and Enrollment
- Module Upload and Drag-and-Drop Ordering
- Video Integration for Content Delivery
- Module-wise Progress Tracking
- Profile Management and Logout

7.1.1 System Features

Feature	Description
User Registration/Login	Users can register and log in securely via role-based access.
Course Management	Teachers can create, update, and delete courses/modules.
Enrollment System	Students can browse and enroll in available courses.
Progress Tracking	Module completion is tracked individually per student.
Video Support	Teachers can add instructional videos to each module.
Drag-and-Drop Ordering	Teachers can reorder modules using a visual drag-and-drop interface.
Role-Based Dashboards	Separate UI components for student and teacher roles.
Secure Logout Functionality	Session management with proper logout features.

7.2 External Interface Requirements

This section outlines the interface-level requirements to ensure smooth interaction between the system and external components.

7.2.1 User Interfaces

- **Login Page**: Input fields for username/password, role selection.
- **Student Dashboard**: List of enrolled courses, progress bar, course continuation option.
- **Teacher Dashboard**: Course management, module uploading, and ordering.
- **Module Page**: Embedded video player, progress tracker, navigation buttons.

UI design was kept minimalistic and responsive, ensuring compatibility with desktops, tablets, and smartphones.

7.2.2 Hardware Interfaces

- No external hardware dependency.
- Application can run on any standard computing device with:
    - Minimum 4 GB RAM
    - Dual-core processor
    - Modern web browser

7.2.3 Software Interfaces

- **Frontend**: Built with HTML5, CSS3, JavaScript, React.js
- **Backend**: Spring Boot (Java-based)
- **Database**: MySQL (Via JPA)
- **APIs**: RESTful architecture using Spring Controllers
- **Testing Tools**: Postman, browser console

7.2.4 Communication Interfaces

- **Client-Server Communication**:
  Uses HTTP over REST APIs. Axios or Fetch is used on the frontend to send requests to the backend.
- **API Endpoints**:
  Follow clean and RESTful naming conventions like `/api/courses`, `/api/modules`, `/api/enrollments`, etc.
- **Data Format**:
  JSON is used for both request and response bodies.

7.3 Non-Functional Requirements

Non-functional requirements describe system qualities and constraints that influence how the system operates. These ensure reliability, usability, and maintainability of the software.

7.3.1 Performance Requirements

- The application should respond to user actions (like login, course load, module view) within **2–3 seconds** under normal network conditions.
- The backend APIs must support **at least 20–30 concurrent users** during peak usage, with low latency.
- Data fetching and rendering should be optimized using efficient queries and pagination (if applicable).

7.3.2 Safety Requirements

- The system should include **session timeouts** to prevent unauthorized access if a user is idle.
- Passwords and sensitive data are **never stored in plain text** and are hashed securely.
- Access to backend operations is restricted based on **role-based authorization**.

7.3.3 Security Requirements

- **Spring Security** is implemented to handle authentication and authorization.
- All API endpoints are secured with role-based checks (e.g., only teachers can modify courses).
- Input validation and sanitization is enforced to prevent **SQL injection** and **XSS attacks**.
- Cross-Origin Resource Sharing (CORS) is configured appropriately to avoid unwanted access.

7.3.4 Software Quality Attributes

Attribute	Description
Reliability	The system is stable under normal usage and recovers gracefully from errors.
Usability	The interface is intuitive and user-friendly, even for non-technical users.
Maintainability	Modular code structure allows for easy debugging and feature expansion.
Portability	Can be hosted on various cloud services or run locally with ease.
Scalability	REST architecture and clean DB schema make scaling simple in the future.

7.4 System Requirements

This section defines the specific requirements for the hardware, software, and database components necessary to run Abhinav Eduverse effectively.

7.4.1 Database Requirements

- The system uses **MySQL** relational database for storing all persistent data such as users, courses, modules, enrollments, and progress tracking.
- The database schema is normalized to avoid redundancy and ensure data integrity.
- Backup and restore mechanisms should be available to prevent data loss.
- Proper indexing is used to optimize query performance on frequently accessed tables.

7.4.2 Software Requirements

- **Backend:**
  - Java 11 or above
  - Spring Boot framework for REST API development
  - **Spring Data JPA** for database interaction and ORM (Object-Relational Mapping)
  - Spring Security for authentication and authorization
  - **Maven** for build and dependency management
- **Frontend:**
  - HTML5, CSS3, JavaScript
  - **React.js** for building the user interface
  - Axios or Fetch API for HTTP communication
- **Database:**
  - MySQL Server 5.7 or later
  - MySQL Workbench for database management and querying
- **Development Environment:**
  - IDE: Spring Tool Suite (STS) or IntelliJ IDEA for backend
  - Visual Studio Code for frontend
  - Postman for API testing

7.4.3 Hardware Requirements

- Minimum specifications to develop and run the platform smoothly:
  - Processor: Intel i3/i5 or equivalent
  - RAM: 4GB minimum, 8GB recommended
  - Storage: 100 GB free disk space for development files and databases
  - Operating System: Windows 10 or later / Linux / macOS
  - Internet connection for development, API testing, and deployment

7.5 SDLC Model to be Applied

For the development of **Abhinav Eduverse**, the **Agile Software Development Life Cycle (SDLC)** model was chosen. This model facilitates iterative development, continuous feedback, and flexibility in handling changing requirements, making it suitable for modern web applications.

Reasons for choosing Agile SDLC:

- **Iterative and Incremental Development:** Features were developed in small modules, tested, and refined based on feedback.
- **Flexibility:** Requirements could evolve during the development phase without disrupting the entire project.
- **Continuous Testing:** Integration and testing occurred throughout the development cycle, reducing bugs and improving quality.
- **User Involvement:** Regular evaluation of the system ensured alignment with user expectations.
- **Efficient Risk Management:** Early detection and resolution of risks and issues.

Agile Process Flow:

1. Requirement Gathering
2. Design & Planning
3. Development (Iterative Sprints)
4. Testing & Feedback
5. Deployment
6. Maintenance & Updates

This SDLC approach contributed significantly to the successful and timely completion of the Abhinav Eduverse platform.

CH 8 : SYSTEM DESIGN

8.1 Data Flow Diagram (DFD)

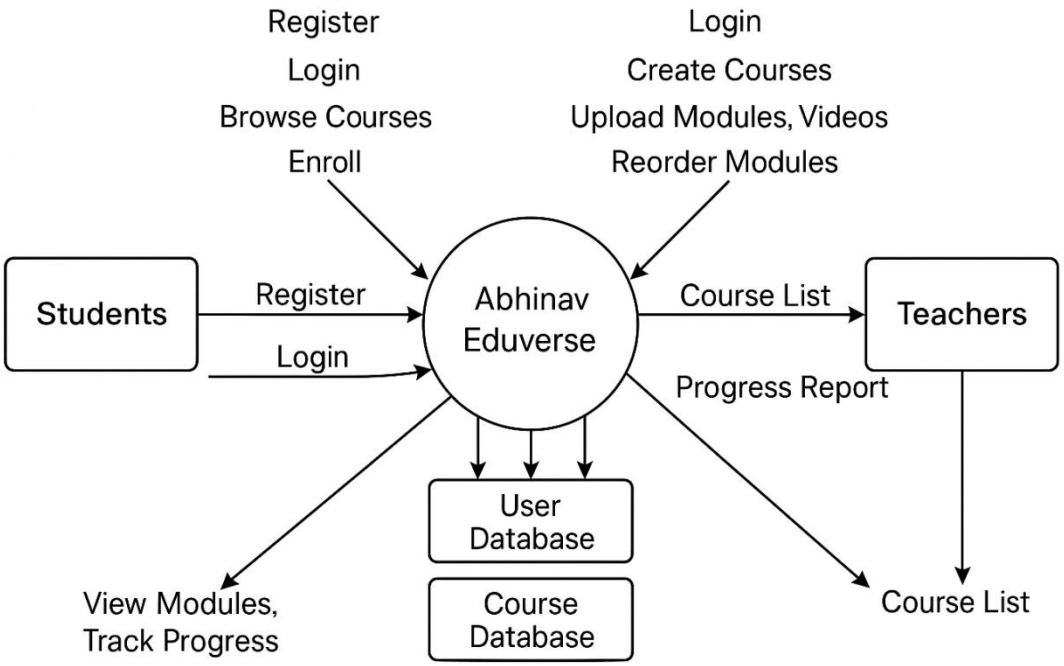
The Data Flow Diagram illustrates how data moves through the Abhinav Eduverse system. It captures the flow of information between various components such as users, system processes, and the database.

Level 0 DFD (Context Diagram)

At the highest level, the system interacts with two main external entities:

- **Students:** Can register, login, browse courses, enroll, view modules, and track progress.
- **Teachers:** Can register, login, create courses, upload modules/videos, reorder modules, and monitor student enrollments.

The system processes these inputs and produces relevant outputs like course lists, progress reports, and notifications.



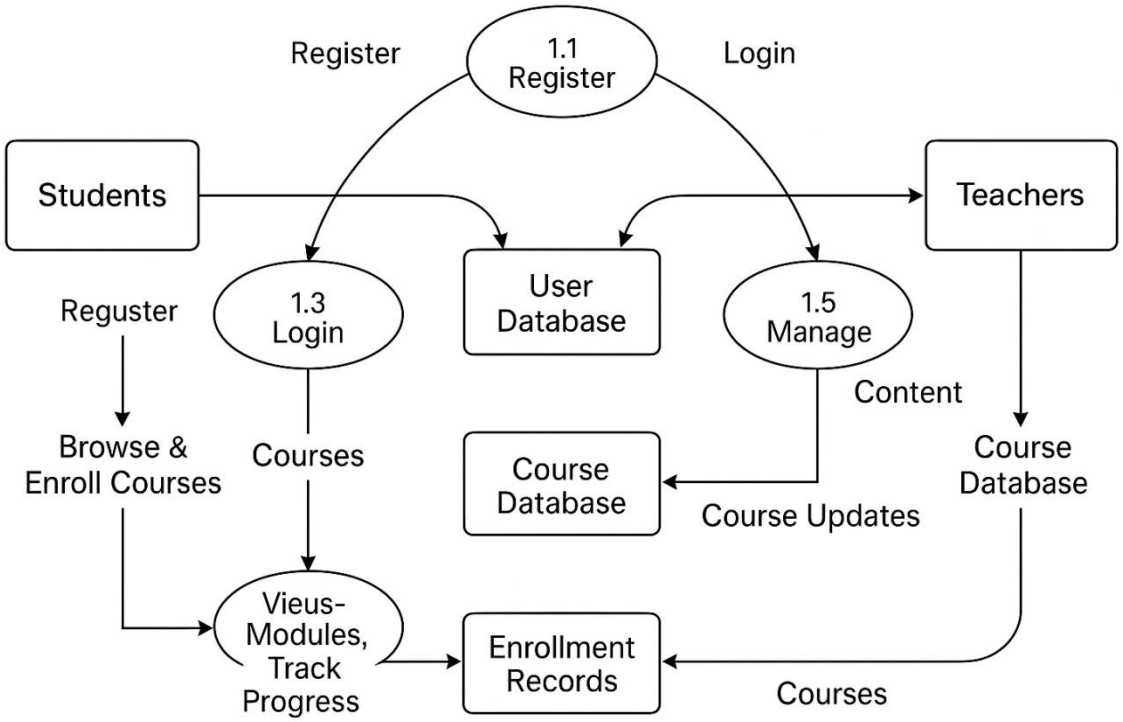
Level 1 DFD (Detailed)

Breaking down the system into key processes:

- **User Authentication:** Validates users and directs them to appropriate dashboards.
- **Course Management:** Handles course creation, updating, deletion by teachers.
- **Enrollment Management:** Manages student course enrollments and progress tracking.
- **Content Delivery:** Provides module video playback and learning material.
- **Progress Tracking:** Updates and reports student progress.

Data stores include:

- User Database
- Course Database
- Enrollment Records



CH 9 : MATHEMATICAL MODEL

9.1 Purpose

The purpose of a mathematical model is to abstract the system's behavior in a formal, logical way using variables, functions, and conditions. For software systems like **Abhinav Eduverse**, a mathematical model helps define the structure of inputs, processes, and outputs in a deterministic way to validate correctness and functionality.

9.2 Problem Definition

Let:

- U be the set of users, where each user is either a **Student (S)** or a **Teacher (T)**
- C be the set of courses
- M be the set of modules per course
- E be the set of enrollments
- P be the set of progress values (percentage of completion)

We define the system as a function:

```
mathematica
CopyEdit
System: f(U, C, M, E) → P
```

This means that, based on a user's enrollment and interaction with courses and modules, a progress value is produced.

9.3 Input

- $U = \{u1, u2, ..., un\}$ → Users (students/teachers)
- $C = \{c1, c2, ..., cm\}$ → Courses
- $M = \{m1, m2, ..., mk\}$ → Modules
- $A = \{a1, a2, ..., al\}$ → Actions performed (Enroll, View Module, Upload Video)

9.4 Output

- Student progress percentage per course
- Course/module visibility for students
- Course content and structure managed by teachers

9.5 Functions

Enroll Function

`Enroll: U × C → E`
A student enrolls in a course, creating an enrollment record.

Progress Function

`Progress: E × M → P`
For each enrollment, progress is calculated based on completed modules.

Create Course Function

`Create: T → C`
Teachers create new courses.

Upload Module Function

`Upload: T × C → M`
Teachers upload or reorder modules within a course.

9.6 Success Conditions

- A valid user must be registered and authenticated.
- Courses and modules must exist before a student can enroll.
- Progress is only calculated if the user is enrolled.

9.7 Failure Conditions

- Invalid login credentials → Denied access
- Attempting to enroll in a non-existent course
- Accessing modules without enrollment
- Uploading modules to a non-owned course (unauthorized access)

CH 10 : CONCLUSION

The project **Abhinav Eduverse** was envisioned, designed, and developed as a full-stack E-learning platform that bridges the gap between educators and learners in a digital environment. It provides structured course delivery, module-wise learning, progress tracking, and intuitive role-based dashboards for both students and teachers.

Throughout the course of the project, we successfully applied key software engineering principles — from requirement gathering, architectural design, backend API development, and frontend interface design, to integration, testing, and documentation. The use of modern technologies like **Spring Boot**, **Spring Security**, **Spring Data JPA**, **MySQL**, and **React.js** helped in building a secure, scalable, and maintainable platform.

By implementing essential features such as drag-and-drop module ordering, video-based learning, and student progress tracking, we ensured both usability and functionality. The project not only fulfilled academic objectives but also simulated real-world software development scenarios, thereby enhancing our technical and problem-solving skills.

Ultimately, **Abhinav Eduverse** stands as a proof-of-concept product that can be enhanced further with features like live classes, quizzes, and AI-based recommendations. This project experience has significantly contributed to our understanding of end-to-end software development and equipped us with practical knowledge for future professional endeavors.

REFRENCES

Official Documentation

1. Spring Boot Documentation – https://docs.spring.io/spring-boot/
2. Spring Security – https://spring.io/projects/spring-security
3. Spring Data JPA – https://spring.io/projects/spring-data-jpa
4. React.js Docs – https://reactjs.org/docs/getting-started.html
5. MySQL Documentation – https://dev.mysql.com/doc/
6. RESTful API Guidelines – https://restfulapi.net/

Web Development Best Practices

7. Mozilla Developer Network (MDN) – https://developer.mozilla.org/en-US/
8. W3Schools Web Technologies – https://www.w3schools.com/
9. Postman API Testing Tool – https://www.postman.com/
10. OWASP Security Guidelines – https://owasp.org/

Books and Academic Sources

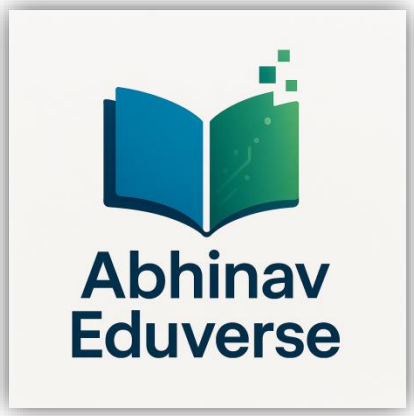
11. "Spring in Action" by Craig Walls (Manning Publications)
12. "Pro JPA 2 in Java EE 8" by Mike Keith and Merrick Schincariol
13. "React Up and Running" by Stoyan Stefanov (O'Reilly)
14. "Designing Web APIs" by Brenda Jin, Saurabh Sahni, Amir Shevat
15. "Clean Architecture" by Robert C. Martin

Software Design & Architecture

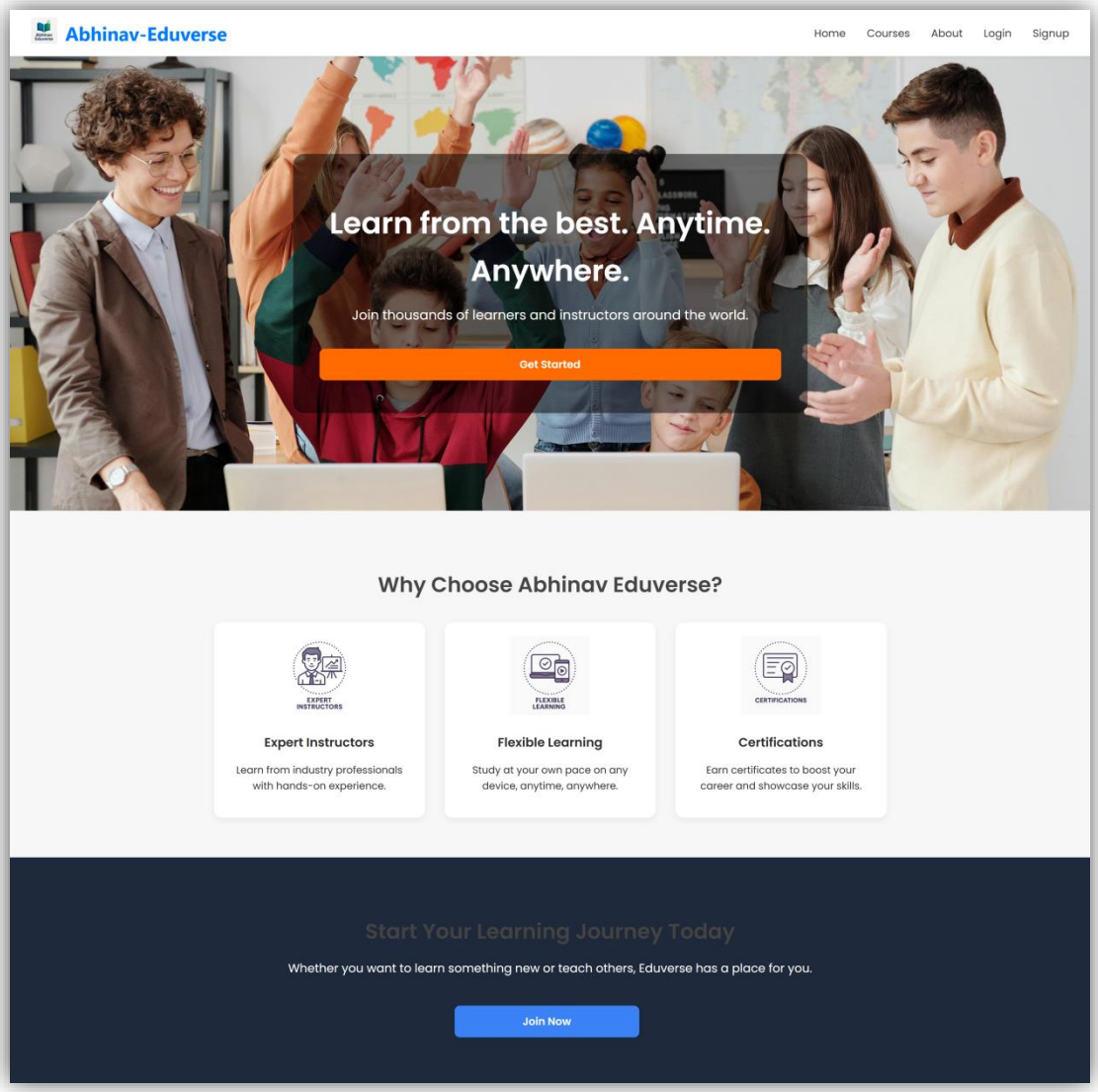
16. Martin Fowler's Blog on Software Architecture – https://martinfowler.com/
17. UML Diagrams Reference – https://www.visual-paradigm.com/
18. Draw.io (diagrams.net) – https://draw.io/

SCREENSHOTS

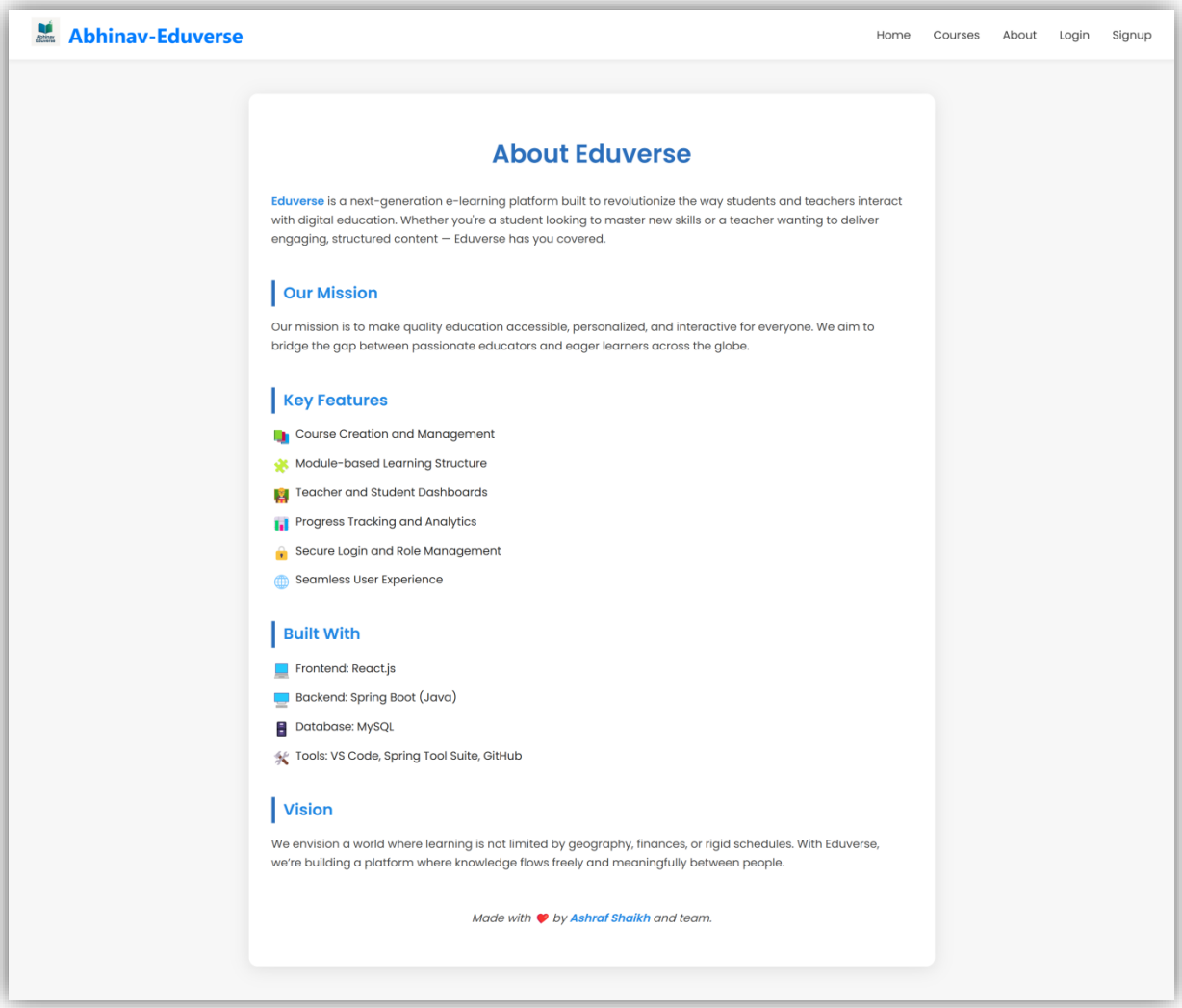
Logo :



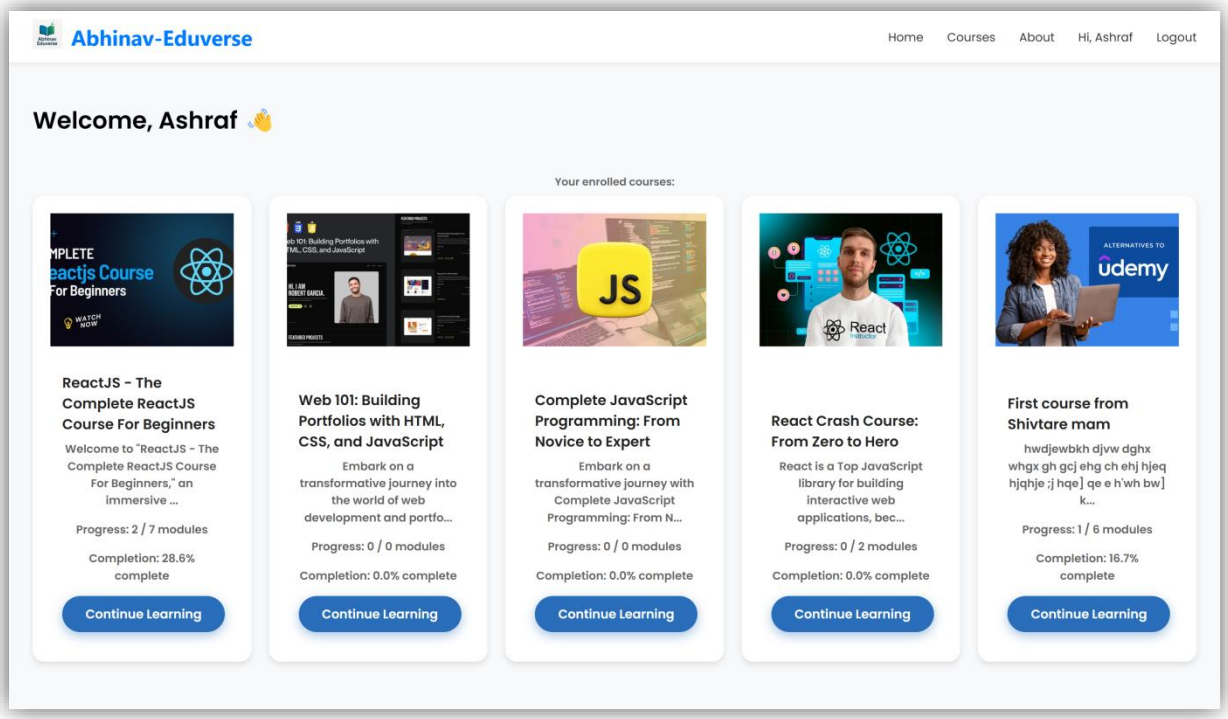
Landing Page :



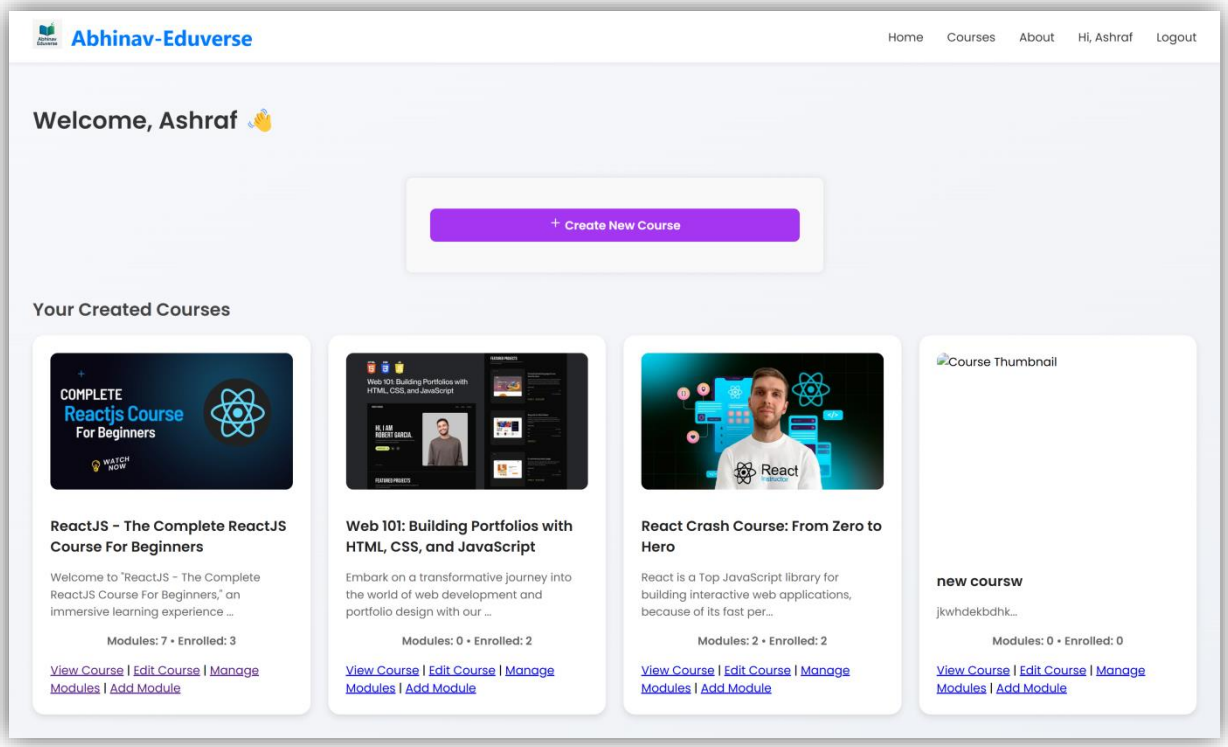
About Page :


Abhinav-Eduverse
Home
Courses
About
Login
Signup
About Eduverse
Eduverse is a next-generation e-learning platform built to revolutionize the way students and teachers interact with digital education. Whether you're a student looking to master new skills or a teacher wanting to deliver engaging, structured content — Eduverse has you covered.
Our Mission
Our mission is to make quality education accessible, personalized, and interactive for everyone. We aim to bridge the gap between passionate educators and eager learners across the globe.
Key Features
Course Creation and Management
Module-based Learning Structure
Teacher and Student Dashboards
Progress Tracking and Analytics
Secure Login and Role Management
Seamless User Experience
Built With
Frontend: React.js
Backend: Spring Boot (Java)
Database: MySQL
Tools: VS Code, Spring Tool Suite, GitHub
Vision
We envision a world where learning is not limited by geography, finances, or rigid schedules. With Eduverse, we're building a platform where knowledge flows freely and meaningfully between people.
Made with ❤ by Ashraf Shaikh and team.

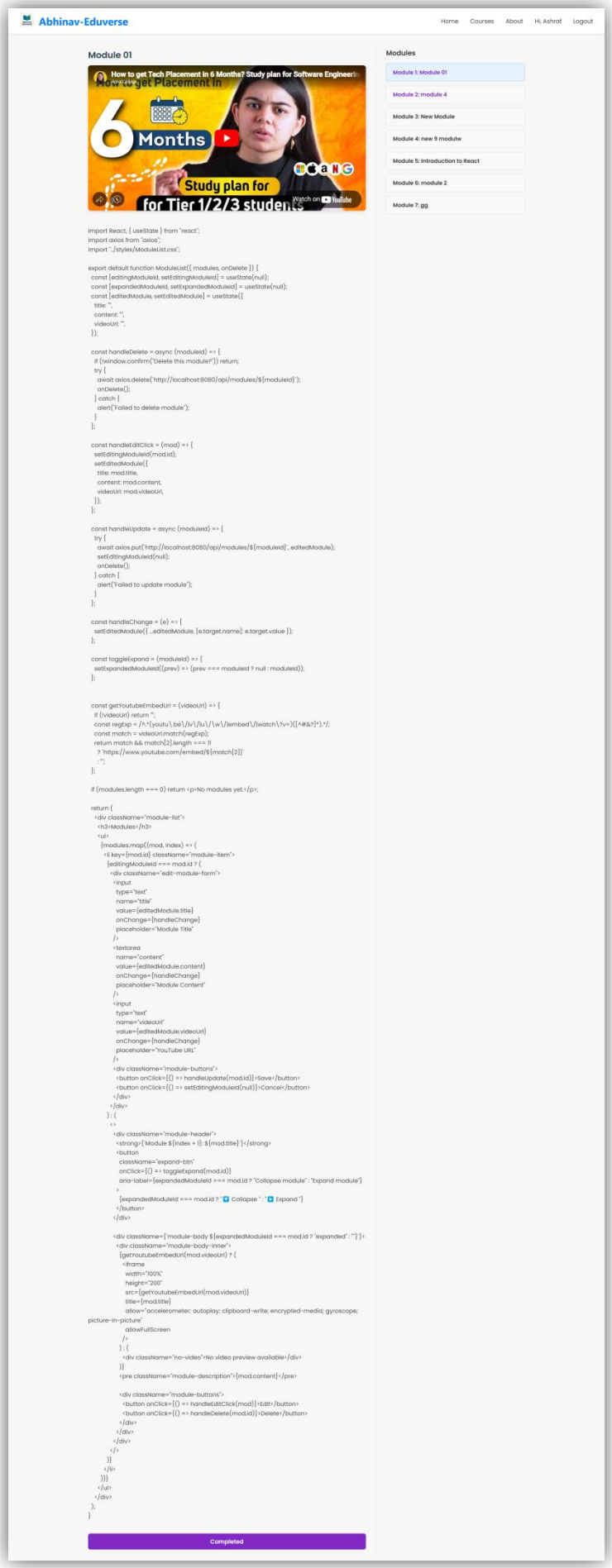
Student Dashboard :



Teacher Dashboard :



Complete Module Page For Students :



Manage Modules For Teacher :