"E-LEARNING RESOURCE MANAGEMENT SYSTEM"

A

Project Report

submitted

in partial fulfillment

for the award of the Degree of

Bachelor of Technology

in Department of Information Technology

Project Mentor:
Ms. Astha Joshi
Assistant Professor

Submitted By :
Aashi Kushwah(21ESKIT001)
Akshita Bhatnagar(21ESKIT012)
Avishi Vijay (21ESKIT027)

Department of Information Technology
Swami Keshvanand Institute of Technology, M & G, Jaipur
Rajasthan Technical University, Kota

Session 2024-2025

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Department of Information Technology

CERTIFICATE

This is to certify that Ms. Aashi Kushwah, a student of B.Tech(Information Technology) VIII th semester has submitted her Project Report entitled ”E-Learning Resource Management System” under my guidance.

Mentor

Ms. Astha Joshi

Assistant Professor

Signature.....

Coordinator

Dr. Priyanka Yadav

Assistant Professor

Signature.....

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Department of Information Technology

CERTIFICATE

This is to certify that Ms. Akshita Bhatnagar, a student of B.Tech(Information Technology) VIII th semester has submitted her Project Report entitled ”E-Learning Resource Management System” under my guidance.

Mentor

Ms. Astha Joshi

Assistant Professor

Signature.....

Coordinator

Dr. Priyanka Yadav

Assistant Professor

Signature.....

Swami Keshvanand Institute of Technology,

Management & Gramothan, Jaipur

Department of Information Technology

CERTIFICATE

This is to certify that Ms. Avishi Vijay, a student of B.Tech(Information Technology) VIII th semester has submitted her Project Report entitled ”E-Learning Resource Management System” under my guidance.

Mentor

Ms. Astha Joshi

Assistant Professor

Signature.....

Coordinator

Dr. Priyanka Yadav

Assistant Professor

Signature.....

DECLARATION

We hereby declare that the report of the project entitled "E-Learning Resource Management System" is a record of an original work done by us at Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur under the mentorship of Ms. Astha Joshi (Dept. of Information Technology) and coordination of Dr. Priyanka Yadav (Dept.of Information Technology). This project report has been submitted as the proof of original work for the partial fulfillment of the requirement for the award of the degree of Bachelor of Technology (B.Tech) in the Department of Information Technology.It has not been submitted anywhere else, under any other program to the best of our knowledge and belief.

Team Members **Signature**

Aashi Kushwah (21ESKIT001)

Akshita Bhatnagar (21ESKIT012)

Avishi Vijay (21ESKIT027)

Acknowledgement

A project of such a vast coverage cannot be realized without help from numerous sources and people in the organization.We take this opportunity to express our gratitude to all those who have been helping us in making this project successful.

We are highly indebted to our faculty mentor Ms. Astha Joshi. She has been a guide, motivator source of inspiration for us to carry out the necessary proceedings for the project to be completed successfully. We also thank our project coordinator Dr. Priyanka Yadav for her co-operation, encouragement, valuable suggestions and critical remarks that galvanized our efforts in the right direction.

We would also like to convey our sincere thanks to Dr. Anil Chaudhary, HOD, Department of Information Technology, for facilitating, motivating and supporting us during each phase of development of the project.Also, we pay our sincere gratitude to all the Faculty Members of Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur and all our Colleagues for their co-operation and support.

Last but not least we would like to thank all those who have directly or indirectly helped and cooperated in accomplishing this project.

Team Members:

Aashi Kushwah (21ESKIT001)

Akshita Bhatnagar (21ESKIT012)

Avishi Vijay (21ESKIT027)

Table of Content

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Problem Statement and Objective

In the modern educational landscape, managing digital learning resources efficiently has become a critical need for institutions, educators, and learners. Traditional systems often lack centralized control, accessibility, and streamlined resource distribution. Objective: The aim of this project is to develop an E-Learning Resource Management System that centralizes, categorizes, and manages educational materials, ensuring easy access, version control, and optimized resource sharing among users.

1.2 Literature Survey /Market Survey/Investigation and Analysis

A review of existing learning management systems (LMS) like Moodle, Google Classroom, and Canvas reveals common issues including limited customization, steep learning curves, and inefficient resource tracking. A market survey indicates an increasing demand for platforms that support modular content delivery, collaboration, and integration with modern tools like video conferencing and cloud storage. This project aims to bridge these gaps with a user-friendly interface, role-based access, and robust backend management.

1.3 Introduction to Project

The E-Learning Resource Management System is a web-based application designed to facilitate the storage, retrieval, sharing, and management of learning resources such as PDFs, videos, assignments, and lecture notes. The system caters to administrators, instructors, and students with role-specific functionalities, ensuring secure

and efficient access to educational content.

1.4 Proposed Logic / Algorithm / Business Plan / Solution / Device

The proposed solution for the E-Learning Resource Management System is based on a robust three-tier architecture comprising the presentation layer, business logic layer, and data layer. The presentation layer is designed using web technologies such as HTML, CSS, and JavaScript to provide an intuitive and responsive user interface. The business logic layer is developed using a backend framework like Django or Node.js, handling user requests, role-based access control, and processing of resource uploads and retrievals. The data layer utilizes a relational database like MySQL or PostgreSQL to store user information, metadata, and learning resources. Role-based access control (RBAC) is implemented to ensure secure and appropriate access to resources based on user type—admin, instructor, or student. Additional features such as file categorization, search filters, and content versioning enhance usability and content management. The system is designed to be scalable and secure, with potential integration points for third-party tools such as Zoom, Google Drive, and email services for communication and notifications.

1.5 Scope of the Project

]The scope of the E-Learning Resource Management System encompasses the design and implementation of a centralized platform for managing educational resources. The system is intended for use in academic institutions such as schools, colleges, and online training centers. It provides a secure environment where administrators can manage users and system settings, instructors can upload and organize learning materials, and students can access, view, and download these resources. The platform supports efficient resource categorization, content version control, and notification systems to keep users informed of updates. Future expansions could include advanced analytics, support for quizzes or assessments, and integration with external learning platforms.

Chapter 2
Software Requirement Specification

2.1 Overall Description

The E-Learning Resource Management System (ERMS) is a comprehensive, cloud-based platform designed to streamline the creation, management, and distribution of educational resources in an online learning environment. Its primary objective is to facilitate the delivery of various educational materials such as courses, videos, assignments, quizzes, and assessments to students while offering instructors tools for content management, student engagement, and progress tracking. The system should support an intuitive user interface that can handle a broad spectrum of learners, including those with minimal technical expertise. It will also ensure that all educational content is easily accessible by students, and provide instructors with tools to monitor performance and provide timely feedback. Furthermore, the platform will be scalable to accommodate a growing number of users and resources, with the flexibility to integrate with other third-party systems, such as Learning Management Systems (LMS), authentication services, and payment gateways.

2.1.1 Product Perspective

2.1.1.1 System Interfaces

The system will need to integrate with various external systems to provide a fully functional and robust experience. First, it will interface with a Learning Management System (LMS) for course creation, enrollment management, and content delivery, ensuring seamless course management. Additionally, third-party authentication systems (such as OAuth-based services, including Google, Facebook, and

LDAP) will be integrated to allow users to authenticate securely and efficiently. The system will also incorporate a payment gateway (like PayPal or Stripe) to handle payments for premium courses or subscriptions. Finally, a cloud storage service (e.g., AWS S3, Google Cloud) will be employed for storing large files like videos, documents, and interactive content, enabling easy access by users and ensuring efficient management of educational resources.

2.1.1.2 User Interfaces

The ERMS will offer a range of user interfaces designed to cater to the needs of different stakeholders. The Admin Interface will provide system administrators with the necessary tools for managing the system at a high level, including user management, resource allocation, system performance monitoring, and access control. This interface will allow admins to configure system settings, create user roles (instructor, student), and access detailed reports on user activity. The Instructor Interface will empower instructors to create, modify, and manage courses, upload educational materials, track student progress, and provide feedback. This interface will include tools for creating interactive assessments such as quizzes, assignments, and exams. The Student Interface will allow learners to access their enrolled courses, view and download materials, complete assignments, and track their learning progress. Additionally, it will enable students to participate in discussions, access grades, and communicate with instructors. Lastly, a Mobile Interface will be optimized for smartphones and tablets, en-

suring that users can interact with the platform even when on the go, supporting offline access and push notifications for important updates.

2.1.1.3 Hardware Interfaces

The ERMS is a cloud-based solution, meaning that users will not need specific hardware requirements to access the platform. However, for optimal performance, the system will be designed to operate efficiently on modern devices, including desktops, laptops, tablets, and smartphones. On the server side, cloud infrastructure (e.g., AWS, Google Cloud) will host the platform, ensuring that the system can scale according to the number of users and available resources. The system must support large multimedia files (e.g., videos, high-resolution images, interactive content) without significant performance degradation. For end-users, devices with at least 2 GB of RAM (minimum) are recommended to ensure smooth operation. While access is possible from lower-end devices, performance may improve with higher specifications.

2.1.1.4 Software Interfaces

In terms of software, the ERMS will use relational databases (such as MySQL or PostgreSQL) to store and manage structured data, including user information, course content, assessments, and grades. It will also employ external services for specific functionalities, such as payment processing (e.g., PayPal or Stripe), live class integration (e.g., Zoom), and video hosting (e.g., Vimeo or AWS). The system will also feature APIs for integration with third-party systems and ser-

vices. The Content Delivery Network (CDN) will be integrated to ensure fast delivery of content, particularly large multimedia files, to users across different regions. APIs will allow the system to interact with other educational platforms for synchronized data transfer and reporting, ensuring that users' progress and course data are consistent across platforms.

2.1.1.5 Communications Interfaces

The system supports standard communication protocols, mainly HTTP and HTTPS for secure data transmission between the client and server. Email communication is also supported using SMTP services for sending alerts, confirmations, and other updates.

2.1.1.6 Memory Constraints

There are minimal memory constraints for end-users. The system is optimized to perform well even on devices with limited memory, while the server must be equipped with sufficient RAM and storage based on the scale of expected traffic and resource uploads.

2.1.1.7 Operations

System operations include user authentication, resource upload and categorization, permission management, and content searching. Admins oversee all operations, instructors handle content uploads, and students primarily interact with resources.

2.1.1.8 Project Functions

The system enables core functions such as secure login/logout, file upload and storage, search and filter functionalities, role-based access control, and real-time notifications. These features are integrated to ensure smooth and efficient operation.

2.1.1.9 User Characteristics

The system is intended for users with basic digital literacy. Admins are typically IT personnel or faculty heads; instructors are subject matter experts or teachers, and students are learners accessing educational content. Each user interacts with the system according to their designated role.

2.1.1.10 Constraints

Constraints include dependency on a stable internet connection, browser compatibility (optimized for modern browsers), and adherence to institutional policies and legal regulations regarding data security and user privacy, such as GDPR.

2.1.1.11 Assumption and Dependencies

The system assumes that users have access to devices with internet connectivity and possess basic computer skills. It also depends on the availability of a secure hosting server, proper domain configuration, and external services such as email gateways for full functionality.

Chapter 3
System Design Specification

3.1 System Architecture

The E-Learning Resource Management System adopts a three-tier architecture comprising the presentation layer, business logic layer, and data layer. The presentation layer is responsible for user interactions and is developed using web technologies such as HTML, CSS, and JavaScript, ensuring a responsive and user-friendly interface. The business logic layer handles the core functionalities of the system, including user authentication, resource management, and access control, implemented using server-side technologies like Node.js or Django. The data layer manages data storage and retrieval, utilizing relational databases like MySQL or PostgreSQL to store user information, resource metadata, and access logs. This architecture ensures scalability, maintainability, and efficient data processing.

3.2 Module Decomposition Description

The system is modularized into distinct components to promote separation of concerns and ease of maintenance. Key modules include:

User Management Module: Handles user registration, authentication, and role-based access control.

Resource Management Module: Facilitates the uploading, catego-

rization, and retrieval of educational resources.

Search and Filtering Module: Enables users to search for resources based on various criteria such as category, type, and keywords.

Notification Module: Manages email and in-app notifications to keep users informed about updates and activities.

Reporting Module: Generates reports on user activities, resource usage, and system performance.

Each module interacts with the others through well-defined interfaces, ensuring cohesive system functionality.

3.3 High Level Design Diagrams

3.3.1 Use Case Diagram

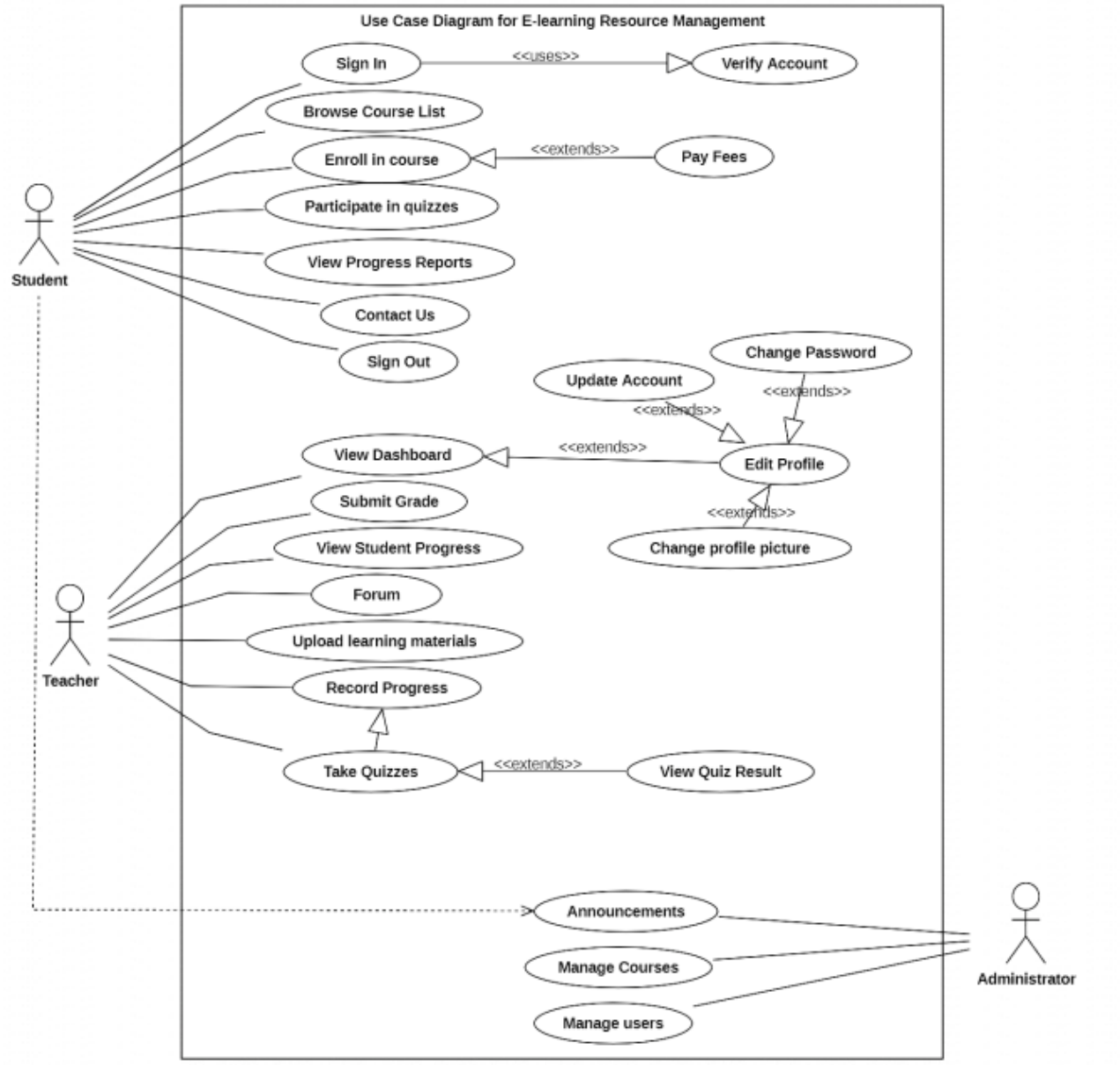

Figure 3.1: Use Case diagram

3.3.2 Activity Diagram

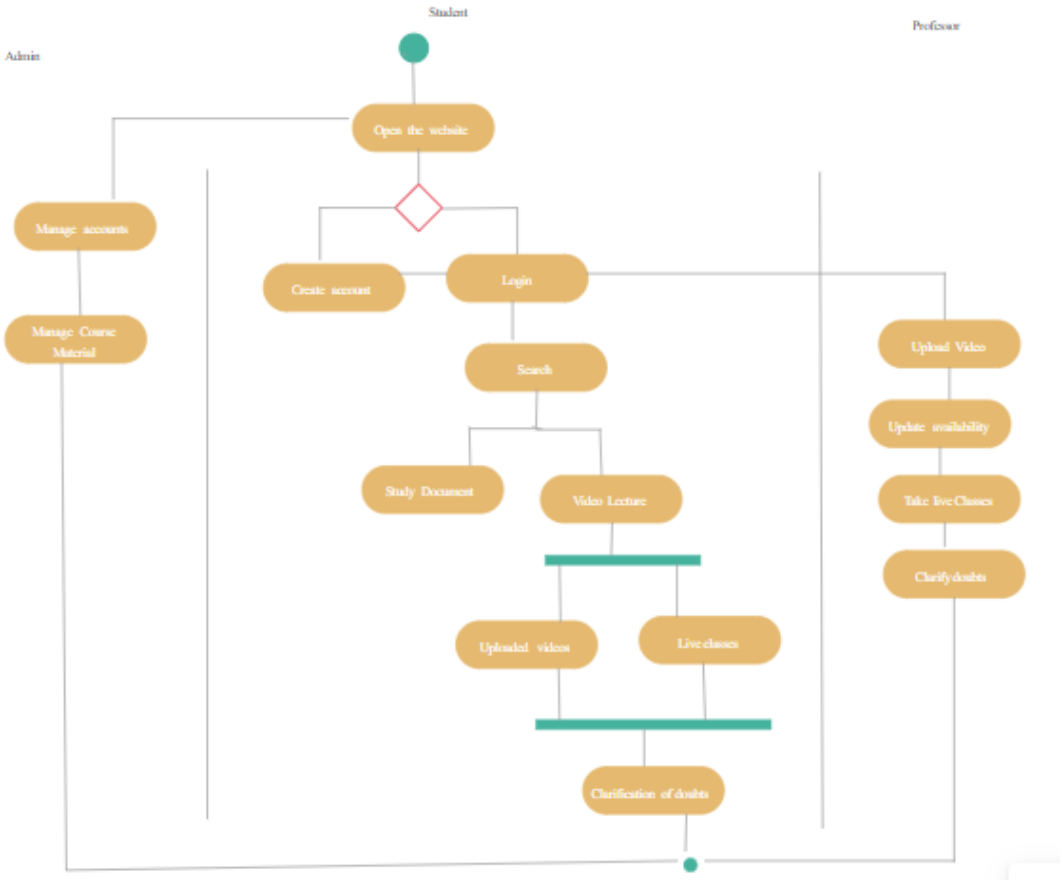

Figure 3.2: Activity diagram

3.3.3 Data-Flow Diagram

Figure 3.3: Zero level DFD diagram

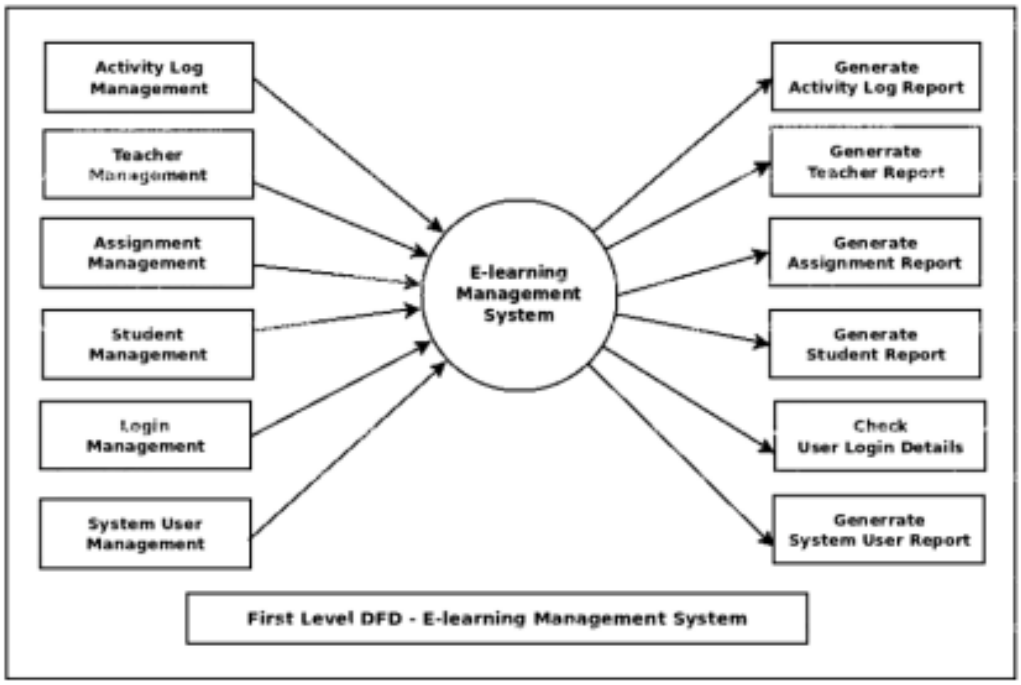

Figure 3.4: First level DFD diagram

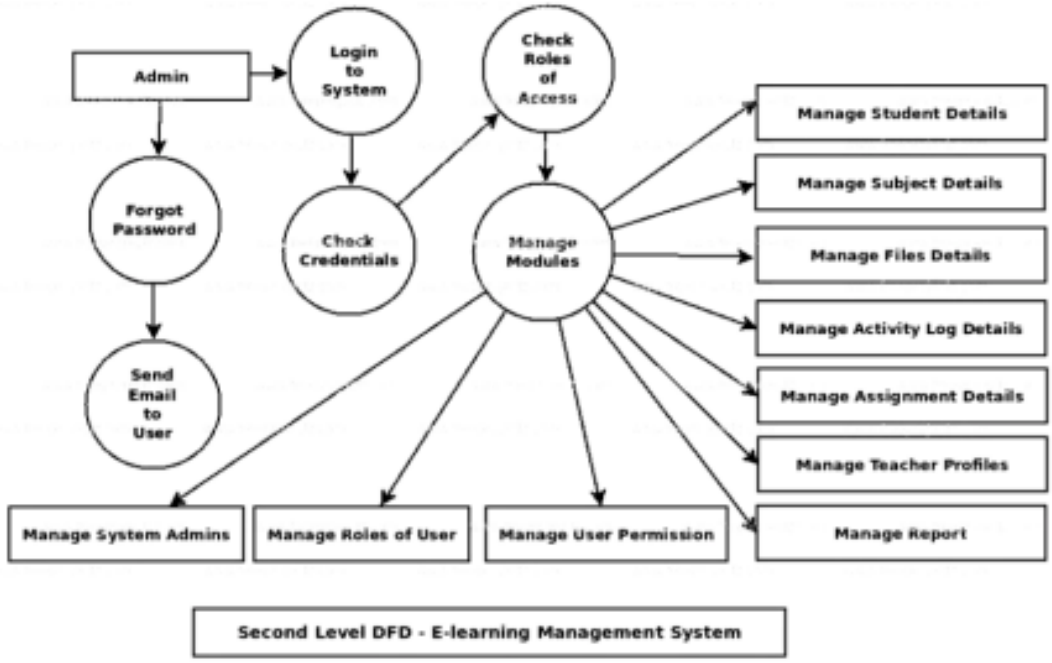

Figure 3.5: Second level DFD diagram

3.3.4 Class Diagram

Figure 3.6: Class diagram

Chapter 4
Methodology and Team

4.1 Introduction to Waterfall Framework

The development methodology adopted for the E-Learning Resource Management System is the Waterfall Model, a widely recognized and traditional software development approach. The Waterfall Model follows a linear and sequential design process, where each phase must be completed before moving on to the next. This methodology is particularly effective when project requirements are well-defined from the beginning, and changes during development are minimal. The process begins with requirement analysis, followed by system design, implementation, testing, deployment, and maintenance. Each of these phases is documented thoroughly, ensuring clarity and consistency throughout the development cycle. This model allows the team to focus on one phase at a time, making it easier to manage tasks, allocate resources, and conduct reviews at each stage. The structured nature of the Waterfall approach ensures that the project follows a disciplined and predictable path, which is essential for academic and institutional software like an e-learning resource platform. Given the scope and goals of this project, the Waterfall Model provided a stable framework to handle design, development, and testing efficiently.

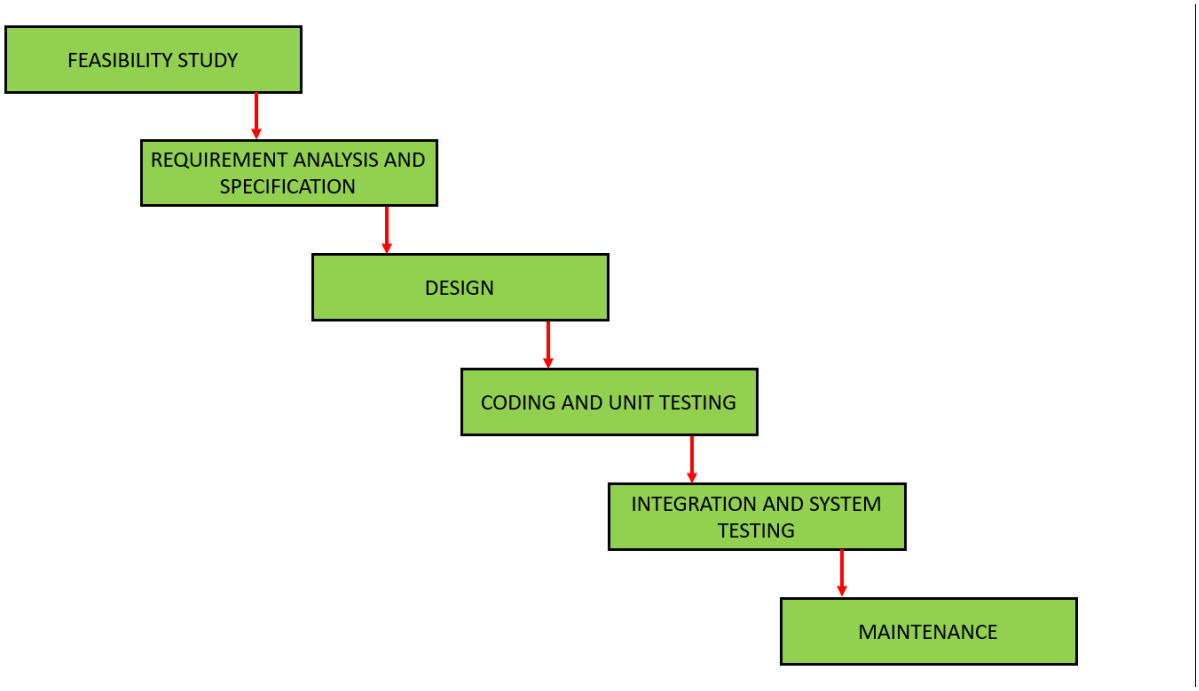

Figure 4.1: WaterFall model

The sequential phases in Waterfall model are-

1. **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.

2. **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

3. **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

4. **Integration and Testing:** All the units developed in the imple-

mentation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

5. **Deployment of system:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

6. **Maintenance:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name ”Waterfall Model”. In this model phases do not overlap.

Waterfall Model Pros & Cons

Advantage The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Disadvantage The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

4.2 Team Members, Roles & Responsibilities

Aashi Kushwah - Database managemnt: Created table for user, courses, student and feedback, etc.
Optimize database for scalability and performance.
Akshita Bhatnagar - Implemented user authentication(login/signup) for students and instructors.
Build APIs to handle dataflow between frontend and Backend.
Avishi Vijay - Created dashboard for students and instructors.
Integrated PHP script for content, author

Chapter 5
Centering System Testing

Testing is a critical phase in the software development lifecycle to ensure that the final product is reliable, functional, and user-friendly. For the E-Learning Resource Management System, various testing methodologies were applied to validate system performance and usability.

5.1 Functionality Testing

Functionality testing was carried out to verify that all features of the system operated as expected according to the requirements specification. This included checking core modules like user authentication, resource upload and download, category-based filtering, and role-specific dashboards. Each user role—admin, instructor, and student—was tested for permission validation to ensure that unauthorized actions were correctly restricted. Test cases were designed and executed for various scenarios including valid/invalid logins, uploading incompatible file types, and accessing restricted content. The system successfully passed all functional tests without any critical issues.

1. Links

   (a) Internal Links: All internal links within the system were checked by manually clicking each link and verifying that it directed

users to the appropriate pages. The system was found to handle internal navigation smoothly, ensuring users could easily transition between pages and access relevant resources.

(b) External Links: Although no external links were included on the website at the time of testing, future enhancements will involve integrating external links. These may include links to candidates' actual profiles, election updates, or other online resources, which will expand the system's connectivity with external websites.

(c) Broken Links : Broken links are those that do not lead to the intended pages. During testing, all links were verified, and no broken links were found. Every link directed the user to the correct page without any issues, ensuring a seamless browsing experience.

2. Forms

(a) Error message for wrong input : The system was tested to ensure it correctly displays error messages when users input incorrect or incomplete data. For instance, when a user enters an invalid password, the system shows a clear error message, notifying the user of the issue. Similarly, if any mandatory fields (such as a username or password) are left empty, error messages prompt the user to complete the required information.

(b) Optional and Mandatory fields : The system identifies manda-

tory fields by marking them with a red asterisk (*), making it clear to users which fields are essential. When users attempt to submit a form without completing mandatory fields, error messages are displayed to guide them in filling in the required details. This ensures that all necessary information is provided before the form can be submitted.

3. Additionally, database connectivity testing was conducted to ensure that the system properly interacts with its underlying database. All database operations, such as adding, retrieving, and modifying data, were thoroughly tested. The system successfully connected to the database, and data was stored, retrieved, and updated without any issues. The integrity and security of the data were verified to ensure that no data loss or unauthorized access occurred during transactions.

5.2 Performance Testing

Performance testing assessed the responsiveness, stability, and scalability of the application under normal and peak usage conditions. The system was tested with a large volume of data uploads, simultaneous logins from multiple users, and frequent access to resource-heavy pages. Load testing was conducted to simulate multiple users accessing and downloading resources concurrently. Results showed that the application maintained acceptable response times and did not crash or slow down under stress, demonstrating the efficiency of its database queries and caching mechanisms.

5.3 Usability Testing

Usability testing focused on the user experience and ease of interaction with the system. Different users were asked to perform key tasks such as uploading files, searching for resources, or modifying their profile settings. Feedback was collected regarding navigation, design clarity, responsiveness, and overall satisfaction. Most users found the interface intuitive and easy to use. Minor adjustments were made based on feedback to improve the placement of buttons and streamline category selection. The system proved to be accessible and user-friendly for all intended user roles.

Chapter 6
Test Execution Summary

The testing phase resulted in the successful validation of all primary system functions. A total of 40 test cases were prepared, covering functional, boundary, and negative scenarios. Out of these, 37 passed on the first run, while 3 required minor fixes. All issues identified were promptly resolved, and re-testing confirmed the system's compliance with the initial requirements. Overall, the system demonstrated strong stability, accurate functionality, and reliable performance under real-world conditions.

Table 6.1: Table to test captions and labels

Chapter 7

Project Screen Shots

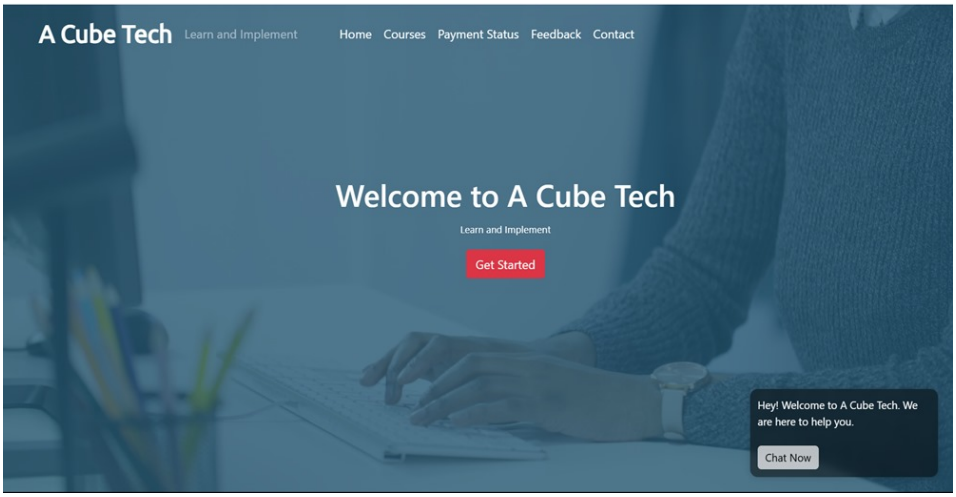

Figure 7.1: Home page

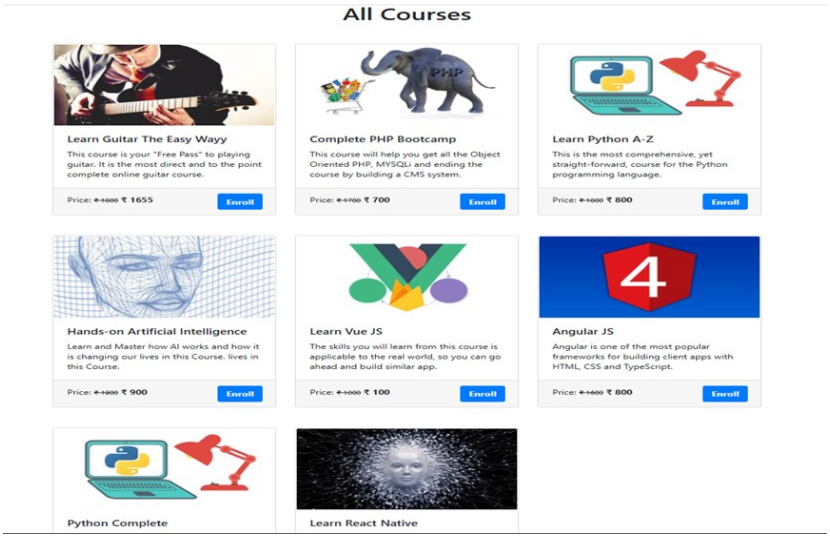

Figure 7.2: Course page

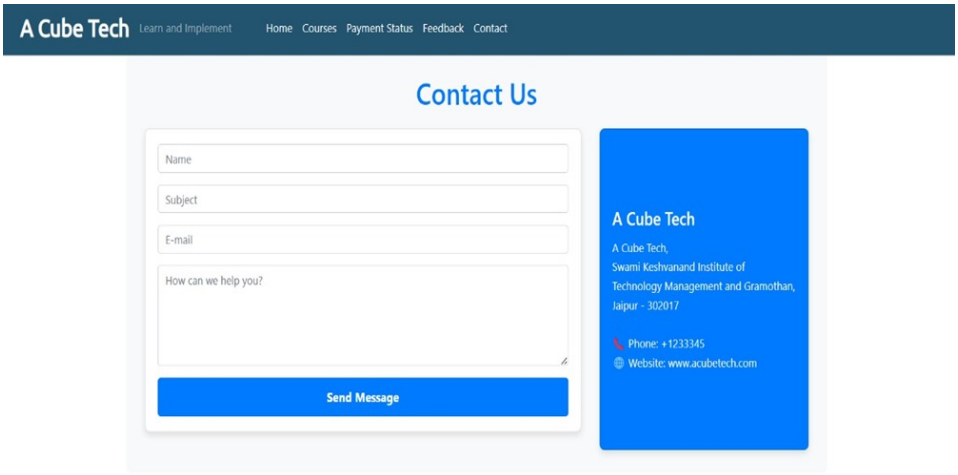

Figure 7.3: Contact us page

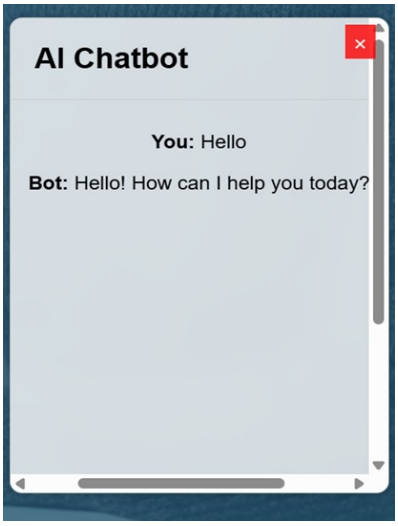

Figure 7.4: Chatbot

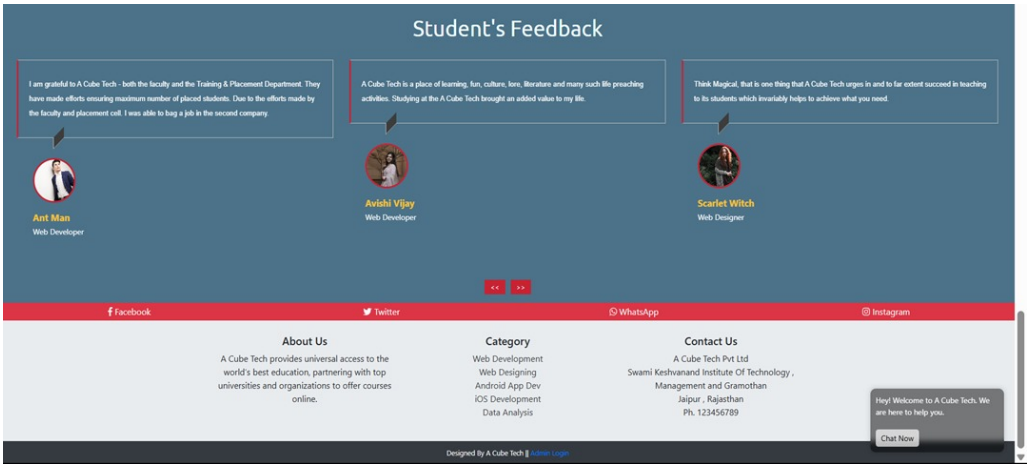

Figure 7.5: Feedback page

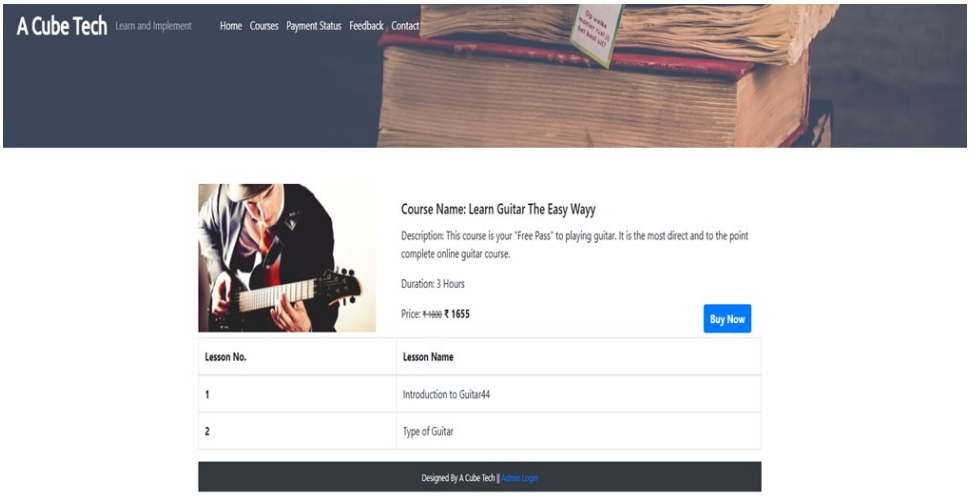

Figure 7.6: Buy now page

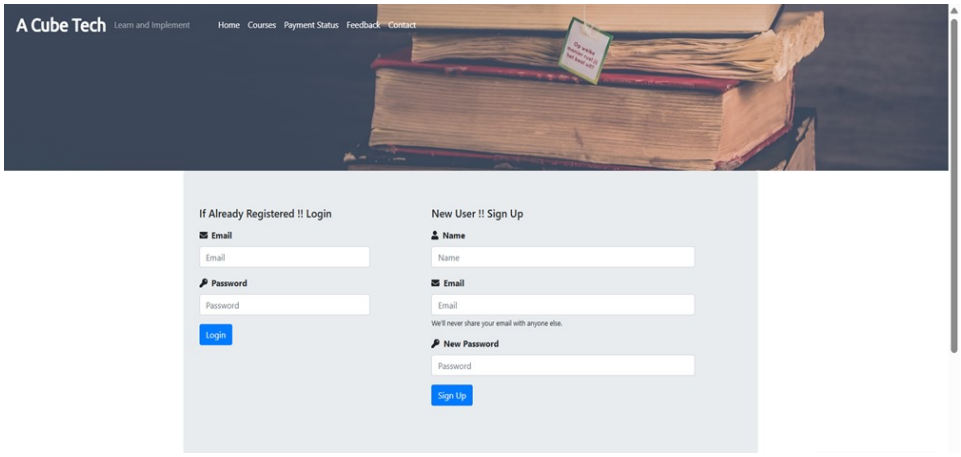

Figure 7.7: Registration page

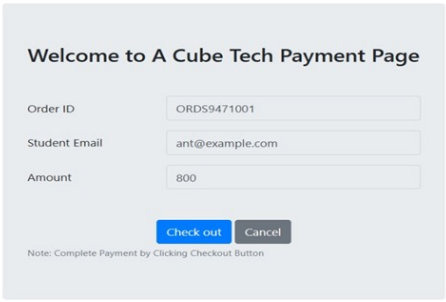

Figure 7.8: Payment page

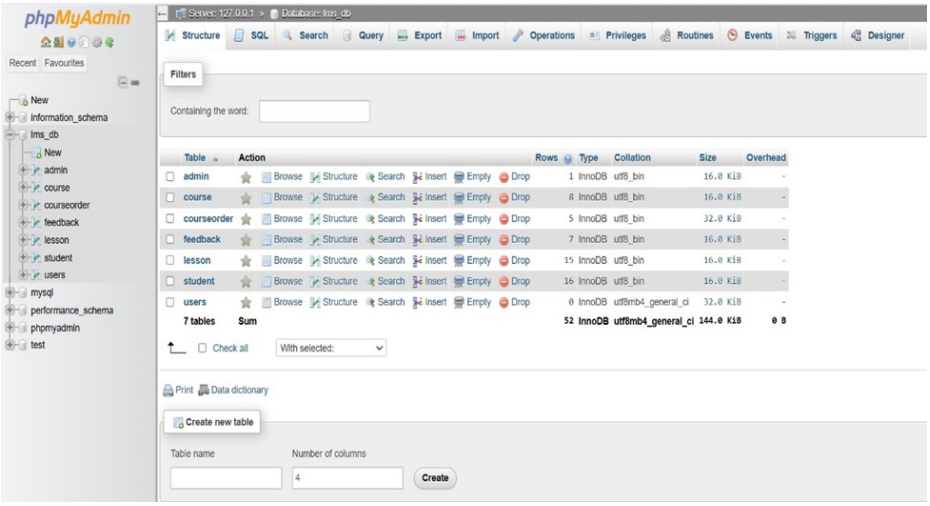

Figure 7.9: Database

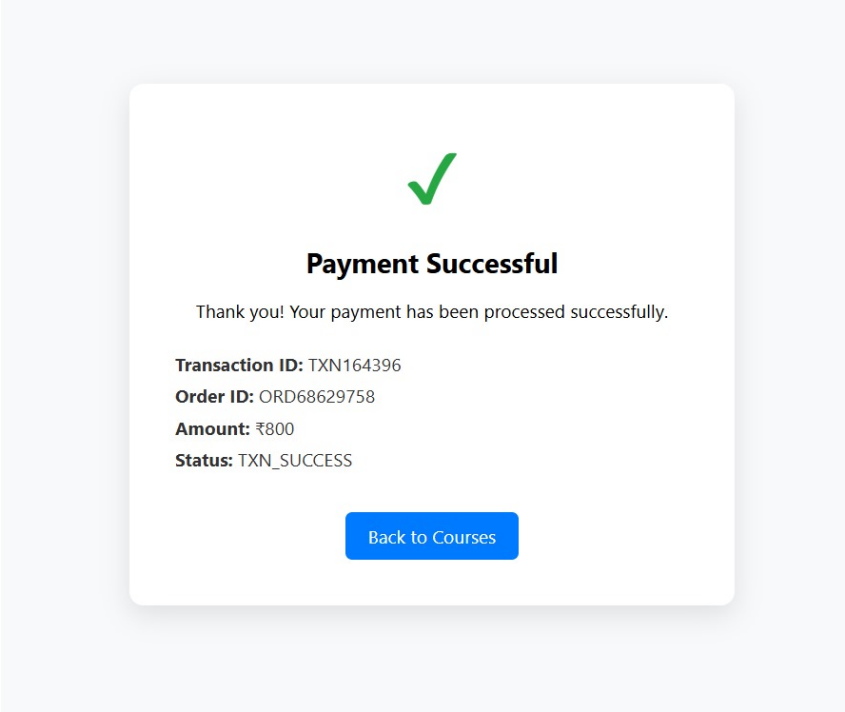

Figure 7.10: Payment done

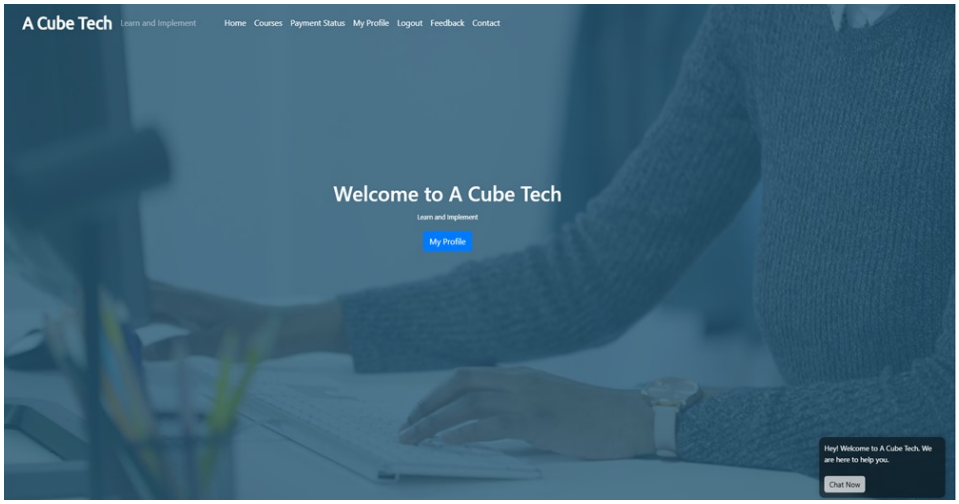

Figure 7.11: Profile page

Figure 7.12: Watch Course

Chapter 8
Project Summary and Conclusions

8.1 Conclusion

In conclusion, the E-Learning Resource Management System is a reliable and adaptable platform that bridges the gap between educators and learners by simplifying access to educational materials. It streamlines the resource management process for instructors while enhancing accessibility for students. With features like role-based access, categorization, and responsive design, the system meets the core objectives outlined during requirement gathering. The structured development and rigorous testing processes have resulted in a high-quality, production-ready application suitable for deployment in real academic environments.

Chapter 9
Future Scope

There is considerable scope to enhance the system further. Future upgrades may include the integration of real-time chat and discussion forums for collaborative learning. Advanced search capabilities using AI or machine learning could be introduced to improve resource discovery. Support for video streaming, embedding interactive content like quizzes, and integration with third-party cloud services such as Google Drive or Dropbox can further increase system flexibility. Additionally, mobile app development for Android and iOS platforms would improve accessibility and user engagement. Analytics dashboards for instructors and administrators could also be developed to monitor usage trends and learning outcomes. These enhancements will make the system more comprehensive and adaptive to the evolving needs of modern education.

References

[1] *https://www.w3schools.com*

[2] *https://www.geeksforgeeks.org*

[3] *ttps://ieeexplore.ieee.org*