COLLEGE MANAGEMENT SYSTEM

Minor Project Report-I

Submitted by: Priyank Shukla

Enrol no. – 0887CS221018

Course- BACHELOR OF TECHNOLOGY

Department – COMPUTER SCIENCE AND ENGINEERING

Under the guidance of -Dr. Umesh Banodha

(Principal U.I.T. Jhabua)

DR. A.P.J. ABDUL KALAM UNIVERSITY INSTITUTE OF TECHNOLOGY JHABUA (M.P.)

DECLARATION

I PRIYANK SHUKLA (0887CS221018), hereby declare that this project work entitled "COLLEGE MANAGEMENT SYSTEM" was carried out by us under the super vision of Dr. UMESH BANODHA, Principal of Dr, APJ Abdul Kalam University Institute of Technology, Jhabua (M.P.). this project work is submitted to Department of Computer Science and Engineering during the academic year 2024-25.

Place : Jhabua

Date :

PRIYANK SHUKLA *Signature*

DR. A.P.J. ABDUL KALAM UNIVERSITY OF TECHNOLOGY, JHABUA

Session- 2024-25

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE OF APPROVAL

This is to certify that the work embodies in this report entitled **"COLLEGE MANAGEMENT SYSTEM"** being submitted by **PRIYANK SHUKLA (0887CS221018)**, who carried out the project work under our supervision and guidance in the **"Department of Computer Science & Engineering"**, Dr. APJ Abdul Kalam University Institute of Technology, Jhabua (M.P.).

(Internal Examiner)

Name :

Designation : Asst. Professor

Institute : Dr. A.P.J. Abdul Kalam U.I.T. Jhabua

Date :

DR. A.P.J. ABDUL KALAM UNIVERSITY OF TECHNOLOGY, JHABUA

Session- 2024-25

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the work embodies in this report entitled **"COLLEGE MANAGEMENT SYSTEM"** being submitted by **PRIYANK SHUKLA (0887cs221018)**, who carried out the project work under our supervision and guidance in the **"Department of Computer Science & Engineering"**, Dr. APJ Abdul Kalam University Institute of Technology, Jhabua (M.P.).

Guided & Approved By:

Dr. UMESH BANODHA

PRINCIPAL UIT-JHABUA

Department of CSE

Dr. APJ Abdul Kalam UIT, Jhabua

Forwarded By:

Prof. LAL SING CHOUHAN

HOD

Department of CSE

Dr. APJ Abdul Kalam UIT, Jhabua

ABSTRACT

A **College Management System (CMS)** is a comprehensive software solution designed to streamline and automate the operations of academic institutions. This system provides a centralized platform for managing various administrative and academic processes, including student enrollment, faculty management, course scheduling, examination handling, and financial administration.

The CMS fosters better communication among students, faculty, and administrative staff through integrated portals and real-time notifications. It ensures the secure handling of sensitive data such as student records, grades, and financial transactions. Additionally, the system supports report generation and analytics to aid in decision-making and institutional growth.

By digitizing traditional workflows, the College Management System reduces manual labour, minimizes errors, and enhances efficiency, enabling the institution to focus more on academic excellence and student success. The modular design allows customization based on the unique requirements of the college, ensuring scalability and adaptability for future expansions.

This abstract outlines the fundamental purpose and benefits of implementing a College Management System as a modern tool to meet the challenges of managing academic institutions effectively.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Prof. Umesh Banodha, my guide, for his invaluable guidance, support, and encouragement throughout this project. His expertise and insightful feedback have been instrumental in shaping the direction and outcome of this work.

I extend my heartfelt thanks to the entire college staff for their cooperation and assistance in providing essential information about the academic processes, which played a crucial role in understanding and implementing the key features of this project.

Lastly, I am grateful to my peers and family for their constant support and motivation, which kept me focused and determined to complete this project successfully.

Thank you all for your contributions and support in making this project a reality.

Priyank Shukla
(0887CS221018)

TABLE OF CONTENTS

Chapter-1: Introduction

Problem Definition

1.1

Managing the daily operations and academic processes of a college is a complex and time-consuming task, often prone to errors when handled manually. Traditional systems rely on paper-based methods or disconnected software tools, leading to inefficiencies such as:

1. Data Management Challenges: Storing and retrieving large volumes of student, faculty, and administrative data can be cumbersome and susceptible to loss or damage.
2. Lack of Integration: Departments operate in silos, creating communication gaps between students, faculty, and administration.
3. Inefficient Processes: Tasks like student enrollment, timetable scheduling, examination management, and result publication are time-intensive and error-prone when handled manually.
4. Limited Accessibility: Stakeholders often lack real-time access to necessary information, leading to delays and miscommunication.
5. Difficulty in Reporting and Analytics: Generating reports and analysing trends for decision-making is a labour-intensive process in traditional systems.

These challenges highlight the need for a centralized, automated, and scalable solution to effectively manage academic and administrative operations. The proposed College Management System aims to address these issues by providing a unified platform that enhances efficiency, accuracy, and accessibility for all stakeholders involved.

Aim and Objectives

1.2

Aim:

To develop a comprehensive and efficient College Management System that automates and streamlines academic and administrative processes, providing a centralized platform for enhanced communication, data management, and decision-making.

Objectives:

Centralized Data Management:

To create a system that securely stores and manages student, faculty, and administrative data, ensuring easy retrieval and minimal redundancy.

Automation of Academic Processes:

To automate key academic tasks such as student enrolment, timetable scheduling, examination management, and grade publication.

Enhanced Communication:

To provide integrated portals for students, faculty, and administration to facilitate seamless communication and real-time notifications.

User-Friendly Interface:

To design an intuitive and user-friendly interface that accommodates the needs of various stakeholders with minimal technical expertise.

Improved Accessibility:

To enable access to the system through multiple devices, ensuring stakeholders can view and update information in real time.

Efficient Reporting and Analytics:

To incorporate tools for generating reports and analysing data trends, aiding in informed decision-making and institutional growth.

Scalability and Customization:

To ensure the system is scalable and customizable to adapt to the evolving needs of the institution.

This project aims to transform the traditional management system into a modern, efficient, and user-centric solution for academic institutions.

Scope

1.3

The **scope of college management** encompasses all activities, processes, and functions involved in the effective administration and operation of a college or higher education institution. It aims to create a conducive environment for learning, teaching, research, and overall institutional development. Below is an outline of the key areas within the scope of college management:

1. Academic Management

- **Curriculum Development**: Designing and updating courses and programs to align with educational goals and industry needs.
- **Faculty Management**: Recruitment, training, performance appraisal, and professional development of teaching staff.
- **Student Management**: Admission processes, academic counselling, performance monitoring, and student discipline.
- **Research Promotion**: Encouraging faculty and student participation in research, innovation, and academic publications.
- **Examinations and Assessments**: Planning and conducting exams, evaluating results, and ensuring transparency.

---

2. Administrative Management

- **Infrastructure Management**: Maintenance and development of buildings, libraries, laboratories, and campus facilities.
- **Regulatory Compliance**: Ensuring adherence to government policies, accreditation standards, and legal requirements.

- **Resource Allocation**: Managing budgets, staff, and resources to support institutional goals.

---

3. Financial Management

- **Budgeting and Planning**: Allocation of funds for various academic and administrative activities.
- **Fee Management**: Collection of tuition and other fees, along with providing scholarships or financial aid.
- **Audit and Compliance**: Maintaining financial records and ensuring transparency in financial transactions.

---

4. Student Welfare and Development

- **Extracurricular Activities**: Organizing sports, cultural events, and clubs to foster holistic development.
- **Career Services**: Providing placement support, career counselling, and skill development programs.
- **Health and Well-being**: Ensuring access to medical facilities, counselling services, and wellness programs.

---

5. Technology Integration

- **Learning Management Systems (LMS)**: Implementing digital platforms for online learning, assessments, and academic interactions.
- **Data Management**: Using ERP systems for managing student records, attendance, and administrative processes.
- **Digital Communication**: Facilitating efficient communication between faculty, students, and parents through portals or mobile apps.

---

6. Community and External Relations

- **Alumni Engagement**: Building strong alumni networks to support institutional growth and mentoring.
- **Industry Collaboration**: Partnering with industries for internships, research, and skill-building programs.
- **Public Relations**: Enhancing the institution's image through branding, events, and media outreach.

---

7. Strategic Planning and Development

- **Vision and Mission Alignment**: Defining the institution's long-term goals and strategies.
- **Quality Assurance**: Regular monitoring and evaluation of academic and administrative processes for continuous improvement.
- **Sustainability Initiatives**: Promoting eco-friendly practices and sustainability in campus operations.

---

8. Crisis and Risk Management

- **Emergency Preparedness**: Handling situations like natural disasters, pandemics, or other crises.
- **Conflict Resolution**: Addressing grievances and resolving disputes among students, staff, or other stakeholders.
- **Safety and Security**: Implementing measures to ensure the physical and digital security of the campus community.

Project Specifications

1.4

A **College Management System (CMS)** is a centralized platform designed to automate and streamline various academic, administrative, and operational processes within a college. Below are the detailed project specifications:

1. Project Overview

- **Project Title**: College Management System (CMS)
- **Objective**: To create a comprehensive system for managing college operations, including admissions, academics, finance, and student services.
- **Scope**:
  - Automate administrative tasks like fee collection, attendance, and grading.
  - Provide a user-friendly interface for students, faculty, and administrators.
  - Enable seamless communication among stakeholders.

---

2. Functional Specifications

a. User Roles

1. **Administrator**:
   - Manage user accounts (students, faculty, staff).
   - Oversee college operations, such as course scheduling and fee management.
2. **Faculty**:
   - Record attendance.
   - Upload grades and study materials.

    - Communicate with students.

3. **Students**:
    - Register for courses.
    - View attendance, grades, and fee details.
    - Access notifications and announcements.
4. **Parents** (Optional Role):
    - Monitor students' academic progress and attendance.
    - View fee payment status.

b. Core Features

1. **Admissions Management**:
    - Online application and document submission.
    - Admission tracking and enrollment confirmation.
2. **Course Management**:
    - Course creation and scheduling.
    - Faculty assignment and course capacity management.
3. **Attendance Tracking**:
    - Automated attendance tracking and reporting.
    - Notifications for absenteeism.
4. **Examination and Grading**:
    - Online exam scheduling.
    - Grade entry and report generation.
5. **Fee Management**:
    - Online fee payment portal.
    - Automated reminders for due payments.
    - Receipt generation and financial reporting.

6. **Library Management**:
    - Book cataloging and issuance tracking.
    - Late return notifications.
7. **Notifications and Alerts**:
    - Email and SMS notifications for important updates.

---

3. Non-Functional Specifications

- **Performance**:
    - The system should handle at least 500 concurrent users with minimal latency.
    - Response time for any query should be under 2 seconds.
- **Scalability**:
    - The system should support future expansion, including additional modules (e.g., hostel management).
- **Security**:
    - Role-based access control to ensure data privacy.
    - Compliance with data protection laws (e.g., GDPR).
- **Usability**:
    - Intuitive user interface with minimal learning curve.
    - Multilingual support (if applicable).
- **Reliability**:
    - System uptime of 99.5%.
    - Automated data backups to prevent data loss.

---

4. Technical Specifications

- **Technology Stack**:

    - **Frontend**: HTML,CSS.
    - **Backend**: PHP
    - **Database**: PHP Admin
    - **Hosting**: Cloud-based (e.g., AWS, Azure, or Google Cloud).
- **Integration**:
    - Payment gateways (e.g., Stripe, PayPal) for fee collection.
    - Learning Management Systems (LMS) like Moodle (if needed).
- **Hardware Requirements**:
    - Minimum server specifications: 8-core CPU, 16 GB RAM, 1 TB SSD storage.
    - Client devices: Web browsers and mobile apps compatible with Android/iOS.

---

5. Project Timeline

- **Phase 1**: Requirements Gathering (2 weeks).
- **Phase 2**: System Design (3 weeks).
- **Phase 3**: Development (12 weeks).
    - Module 1: Admissions and Course Management.
    - Module 2: Fee and Attendance Management.
    - Module 3: Examination and Library Management.
- **Phase 4**: Testing (4 weeks).
- **Phase 5**: Deployment and Training (3 weeks).
- **Total Duration**: Approximately 6 months.
    - Annual Maintenance: $5,000.

1. **System Features**:
    - Fully functional web-based and mobile applications.
    - Admin dashboard with analytics and reports.
2. **Documentation**:
    - User manuals for students, faculty, and administrators.
    - Technical documentation for system maintenance.
3. **Support**:
    - Initial 6 months of technical support post-deployment.

---

8. Risks and Mitigation

- **Risk**: User res istance to adopting a new system.
    - **Mitigation**: Provide training sessions and user-friendly design.
- **Risk**: Data breaches or security vulnerabilities.
    - **Mitigation**: Implement robust security measures, including encryption and firewalls.

Chapter-2 : Analysis

System requirement analysis

2.1

System Requirement Analysis for a College Management System

System Requirement Analysis (SRA) is a vital phase in developing a **College Management System (CMS)**. It involves identifying, documenting, and validating the system's functional, non-functional, and technical requirements to ensure it meets the stakeholders' needs.

1. Objectives of the System Requirement Analysis

- Understand and document the specific needs of students, faculty, administrators, and other stakeholders.
- Establish clear functional and non-functional requirements.
- Provide a solid foundation for the system design and development process.

---

2. Stakeholder Identification

- **Primary Stakeholders**:
  - Administrators (e.g., college management, admission officers).
  - Faculty and academic staff.
  - Students.
- **Secondary Stakeholders**:
  - Parents/Guardians (optional role).
  - External entities (e.g., accrediting bodies, government regulators).

---

3. Functional Requirements

These describe the specific features and functionalities the system must provide:

a. User Management

- Role-based user accounts for administrators, faculty, students, and parents.
- Secure login and password recovery features.

b. Admissions Management

- Online student registration and document uploads.
- Automated verification and enrollment confirmation.

c. Academic Management

- Course and curriculum creation.
- Faculty assignment to courses.
- Scheduling of classes and exams.

d. Attendance Management

- Digital attendance recording by faculty.
- Automated attendance percentage calculation.
- Notifications to students/parents for absenteeism.

e. Examination and Grading

- Online exam scheduling and result publication.
- Faculty entry of grades and automated result analysis.

f. Financial Management

- Online fee payment portal with receipt generation.
- Automated reminders for fee deadlines.

g. Library Management

- Digital catalog of books and resources.
- Issuance, return, and late fee management.

h. Communication and Notifications

- In-app notifications for important updates (e.g., exam schedules, fee due dates).
- Email and SMS integration.

---

4. Non-Functional Requirements

These define the performance and usability characteristics of the system:

a. Performance

- The system must support 500–1000 concurrent users.
- Response time for user actions should be under 3 seconds.

b. Scalability

- The system should allow for future modules (e.g., hostel management, alumni portal).

c. Security

- Data encryption for sensitive information like student records and financial transactions.
- Role-based access control to ensure restricted access to sensitive data.

d. Reliability

- System uptime of at least 99.5%.
- Regular data backups to prevent data loss.

e. Usability

- Intuitive interface suitable for users with basic technical skills.
- Multilingual support, if necessary.

---

5. Technical Requirements

These specify the technologies and tools required to build the system:

a. Software Requirements

- **Frontend**: HTML,CSS.
- **Backend**: PHP Admin
- **Database**: PHP Amin(Mysql)
- **Operating System**: Compatible with Windows, macOS, and Linux for servers and client devices.

b. Hardware Requirements

- Server: 8-core CPU, 16 GB RAM, 1 TB SSD storage.
- Client Devices: Desktops, laptops, or smartphones with modern web browsers.

c. Network Requirements

- High-speed internet connection for seamless operations.
- Secure connections via HTTPS and VPNs for remote access.

---

10. Deliverables from Requirement Analysis

- **System Requirement Specification (SRS) Document**:
  - Functional, non-functional, and technical requirements.
  - Use case diagrams and process workflows.
- **Prototype or Wireframes**:
  - Visual representation of the system's user interface.

System feasibility

2.2

System Feasibility Analysis for a College Management System

A **System Feasibility Analysis** evaluates whether a College Management System (CMS) can be successfully developed and implemented within the constraints of technology, cost, time, and organizational goals. Below is a detailed analysis of the feasibility dimensions:

---

1. Technical Feasibility

Objective: Assess whether the system can be developed using available technologies and resources.

- **Infrastructure**:
  - Existing servers, computer labs, and network infrastructure can support the CMS.
- **Technology Stack**:
  - Recommended frontend technologies: React.js, Angular.
  - Backend technologies: Django (Python), Spring Boot (Java), or Node.js.
  - Database: MySQL, PostgreSQL, or MongoDB.
- **Expertise**:
  - The in-house IT team or third-party vendors have sufficient expertise in web development, database management, and system integration.

Scalability:

  - The system is designed to handle future increases in student numbers, courses, and additional modules (e.g., hostel management, alumni portal).

Conclusion: The project is technically feasible, given the availability of modern technologies and skilled professionals.

---

2. Economic Feasibility

Objective: Evaluate the financial viability of the project.

- **Estimated Costs**:
  - Development: 10,000/-
  - Implementation: Depend on User's
  - Annual Maintenance: 30,000
- **Expected Benefits**:
  - Reduction in manual administrative costs (e.g., admissions, attendance tracking).
  - Increased efficiency in fee collection, grade entry, and reporting.
  - Enhanced student and parent satisfaction due to improved communication and access to information.
- **Cost-Benefit Analysis**:
  - Break-even period: Estimated within 2–3 years due to operational cost savings.

---

3. Operational Feasibility

Objective: Assess the system's alignment with organizational goals and user acceptance.

- **Stakeholder Acceptance**:

  - Faculty and administrative staff see the value in reducing manual work.
  - Students and parents appreciate digital access to grades, attendance, and notifications.

- **Alignment with Processes**:
  - The system fits seamlessly into existing workflows, such as admissions, fee payments, and attendance tracking.
  - Automation minimizes errors and delays in processes.

- **Training Requirements**:
  - Minimal training required for most users due to an intuitive interface.
  - Administrators and faculty will receive detailed training sessions.

Functional Requirements

2.3

Functional Requirements for a College Management System

The **Functional Requirements** define the specific actions, tasks, and features the College Management System (CMS) must perform to meet the needs of its stakeholders, such as administrators, faculty, students, and parents. These requirements are categorized based on system modules for clarity.

---

1. User Management

- Role-based access control for:
  - **Administrator**: Full access to all modules.
  - **Faculty**: Access to academic and attendance-related modules.
  - **Students**: Access to academic records, attendance, and fee details.
  - **Parents**: View-only access to student performance and fee payment status.
- Secure login with multi-factor authentication.
- Password recovery and management.

---

2. Admissions Management

- Online application submission with document uploads.
- Automated eligibility verification and shortlisting.
- Admission confirmation and enrollment number generation.
- Notification system for application status updates.

---

3. Academic Management

- Course creation and management by administrators.
- Faculty assignment to courses.
- Scheduling of classes, lectures, and examinations.
- Digital syllabus and study material upload by faculty.

---

4. Attendance Management

- Real-time attendance marking by faculty via web or mobile.
- Automatic calculation of attendance percentage.
- View attendance history for students and parents.
- Alerts for low attendance thresholds.

---

5. Examination and Grading

- Exam scheduling with seat allocation.
- Online examination module for conducting assessments (optional).
- Faculty grade submission and moderation.
- Automated grade card generation and report distribution.

---

6. Fee Management

- Online payment portal with multiple payment options.
- Fee structure creation and installment management.
- Automated reminders for upcoming or overdue payments.
- Receipt generation and payment history tracking.

---

7. Library Management

- Digital cataloging of books and resources.
- Book issuance and return tracking.
- Late return notifications and fine management.
- Resource search functionality for students and faculty.

---

8. Notifications and Alerts

- In-app notifications for important updates like exam schedules, fee deadlines, or class changes.
- Email and SMS integration for critical alerts.
- Personalized dashboards for announcements.

---

9. Reports and Analytics

- Attendance, grades, and fee reports for administrators and faculty.
- Performance analysis reports for students.
- Institutional reports for management, such as revenue and enrollment statistics.
- Export options in PDF, Excel, and other formats.

---

10. Communication and Feedback

- Internal messaging system for faculty, students, and administrators.
- Feedback collection from students on courses and faculty.
- Parent-teacher communication channel for academic updates.

---

11. Hostel Management (Optional Module)

- Room allocation and vacancy tracking.

- Fee collection and billing.
- Maintenance request management.

---

12. Transportation Management (Optional Module)

- Bus route and schedule creation.
- Seat allocation for students and faculty.
- Notification system for delays or changes.

Use Cases for Functional Requirements

Example Use Case 1: Attendance Management

1. **Actor**: Faculty
2. **Action**: Faculty logs in, selects the course, marks attendance for students, and submits.
3. **System Response**: Attendance is recorded, and the percentage is updated. Notifications are sent to students with low attendance.

Example Use Case 2: Fee Payment

1. **Actor**: Student
2. **Action**: Student logs in, views the fee structure, selects the payment option, and completes the payment.
3. **System Response**: Payment is processed, receipt is generated, and due amount is updated.

Non-Functional Requirements

2.4

Non-functional requirements (NFRs) specify the quality attributes, performance standards, and constraints that the College Management System (CMS) must meet. These ensure the system operates effectively and delivers a seamless experience for users.

---

1. Performance Requirements

- The system must handle a minimum of 500 concurrent users without performance degradation.
- Response time for user actions should not exceed **2 seconds** under normal load and **5 seconds** under peak load.
- Batch processing tasks (e.g., report generation) must be completed within **5 minutes**.

---

2. Scalability

- The system should support future expansion:
  - Increase in the number of users (students, faculty, administrators).
  - Addition of new modules (e.g., hostel management, alumni tracking).
  - Support for multiple campuses or branch locations.

---

3. Security

- **Authentication**:
  - Role-based access control (RBAC) to restrict system access based on user roles (e.g., student, faculty, administrator).

    - Multi-factor authentication (MFA) for enhanced security.

- **Data Protection**:
    - All sensitive data (e.g., student records, financial transactions) must be encrypted during storage and transmission.
    - Compliance with local and international data protection laws (e.g., GDPR, FERPA).
- **Audit Trails**:
    - Maintain logs of all user activities for accountability and tracking security breaches.

---

4. Usability

- **User-Friendly Interface**:
    - Intuitive and easy-to-navigate UI for users with minimal technical knowledge.
    - Mobile-responsive design for access on smartphones, tablets, and desktops.
- **Accessibility**:
    - Compliance with **WCAG (Web Content Accessibility Guidelines)** to support users with disabilities.
    - Options for text-to-speech and adjustable font sizes.

---

5. Availability and Reliability

- **System Uptime**:
    - The system must guarantee **99.5% availability** to ensure uninterrupted access for users.
- **Backup and Recovery**:
    - Automated daily data backups.

    - Disaster recovery plan to restore full system functionality within 4 **hours** in case of failure.

6. Maintainability

- Modular architecture to allow easy updates, bug fixes, and enhancements.
- Detailed system documentation for developers and administrators.
- Support for automated monitoring and error reporting.

7. Compatibility

- **Browser Compatibility**:
    - The system must support modern web browsers, including Chrome, Firefox, Safari, and Edge.
- **Device Compatibility**:
    - Ensure compatibility with Windows, macOS, Android, and iOS platforms.
- **Integration**:
    - Seamless integration with external systems such as payment gateways, Learning Management Systems (LMS), and government reporting tools.

8. Data Requirements

- **Data Accuracy**:
    - The system must ensure all student records, grades, and financial data are accurate and free from discrepancies.

9. Environmental Requirements

- Optimize system design to reduce server energy consumption.
- Encourage paperless workflows, reducing the college's environmental footprint.

10. Monitoring and Reporting

- Real-time system monitoring to detect and resolve issues proactively.
- Generate usage statistics to assess system performance and user activity.

Platform Specifications

2.5

The **platform specification** outlines the technological, hardware, and software environment required to develop, deploy, and maintain a College Management System (CMS). These specifications ensure the system's efficient functioning, scalability, and compatibility.

---

1. Development Platform

- **Frontend Technologies**:
  - Markup and Styling: HTML5, CSS3.
  - Client-side scripting: PHP
- **Backend Technologies**:

  PHP
- **Database Management System (DBMS)**:

  PHP Admin
- **APIs**:
  - RESTful or GraphQL APIs for communication between client and server.
  - Third-party API integrations for payment gateways, SMS, email, and learning management systems.
- **Version Control**:
  - Git for source code management (GitHub).
- **Development Tools**:
  - IDEs: Visual Studio Code,XAMPP.

---

2. Deployment Platform

- **Operating System**:
  - Server OS: Windows.
  - Client OS: Compatible with Windows, macOS, Android, and iOS.
- **Web Server**:
  - Apache HTTP Server
- **Hosting Environment**:
  - On-Premises: Dedicated college servers with backup capabilities.
  - Cloud Hosting (recommended): AWS, Microsoft Azure, or Google Cloud for scalability and global accessibility.
- **Containerization (Optional)**:
  - Docker or Kubernetes for deploying microservices and ensuring consistent environments.

---

3. Hardware Requirements

Server Requirements:

- **Minimum Configuration**:
  - Processor: Quad-Core CPU (2.5 GHz or higher).
  - RAM: 16 GB.
  - Storage: 500 GB SSD.
  - Network: 1 Gbps Ethernet.
- **Recommended Configuration**:
  - Processor: 8-Core CPU (3.0 GHz or higher).
  - RAM: 32 GB.
  - Storage: 1 TB SSD with RAID for redundancy.

    - Network: High-speed Internet connection with load balancing.

Client Requirements:

- Devices: Desktop, laptop, tablet, or smartphone.
- Browser: Modern browsers like Chrome, Firefox, Safari, or Edge.
- Network: Reliable internet connection (minimum 5 Mbps).

---

5. Software Specifications

- **Database Management Tools**:
    - phpMyAdmin for MySQL, pgAdmin for PostgreSQL.
- **Payment Integration**:
    - Gateways: PayPal, Stripe, or Razorpay.
    - Security: PCI DSS compliance for secure transactions.
- **Notification Services**:
    - Email: SMTP services (e.g., Gmail API, SendGrid).
    - SMS: Twilio, Nexmo.

---

6. Scalability and Extensibility

- **Scalability**:
    - Horizontal scaling for handling increased user load.
- **Extensibility**:
    - Modular architecture to allow easy integration of new functionalities.

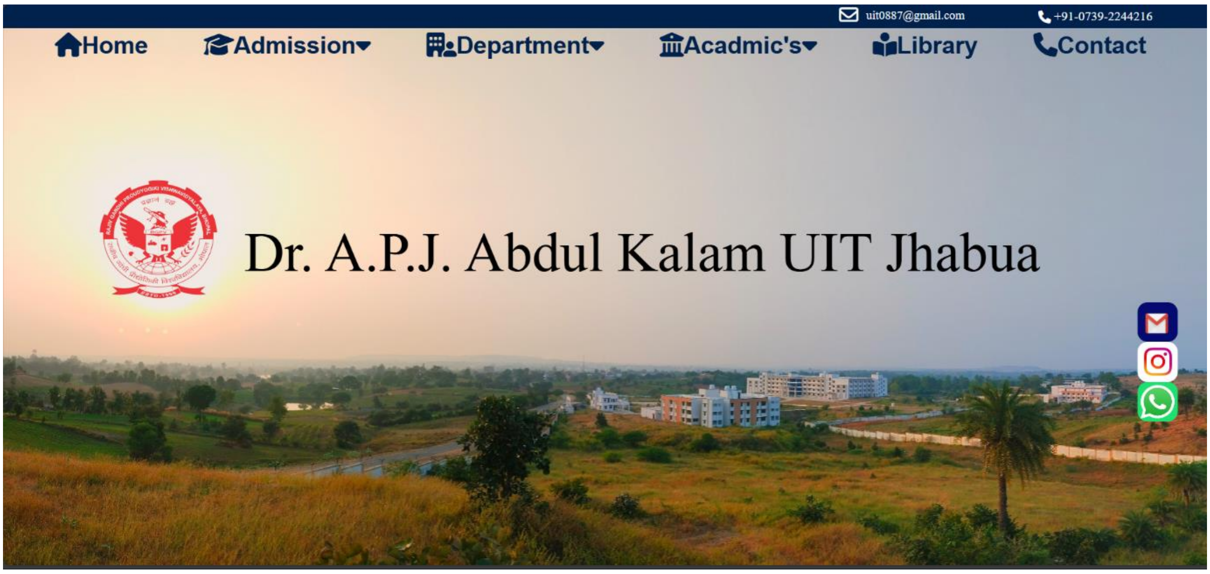
---

7. Backup and Recovery

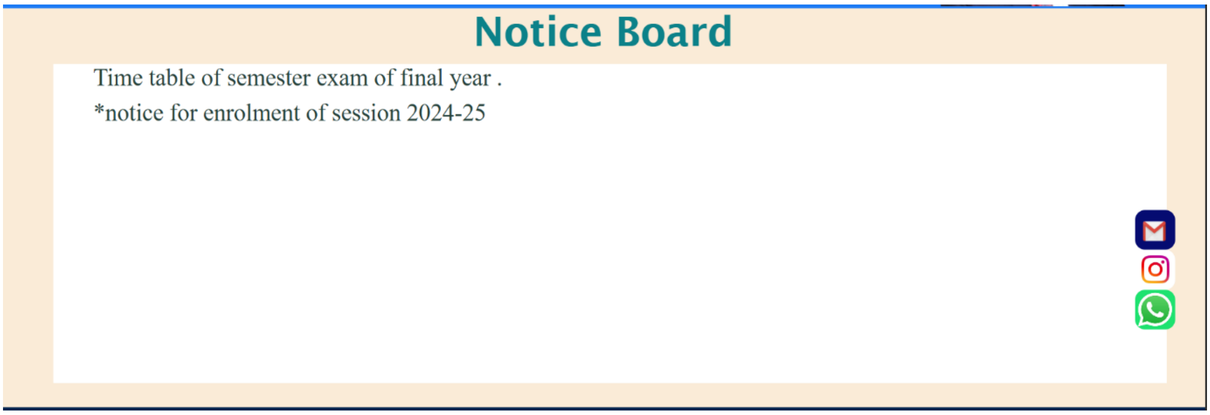
- Automated daily backups to a secure cloud or secondary server.

Software Interface / Communication interface

2.5.2/2.5.3

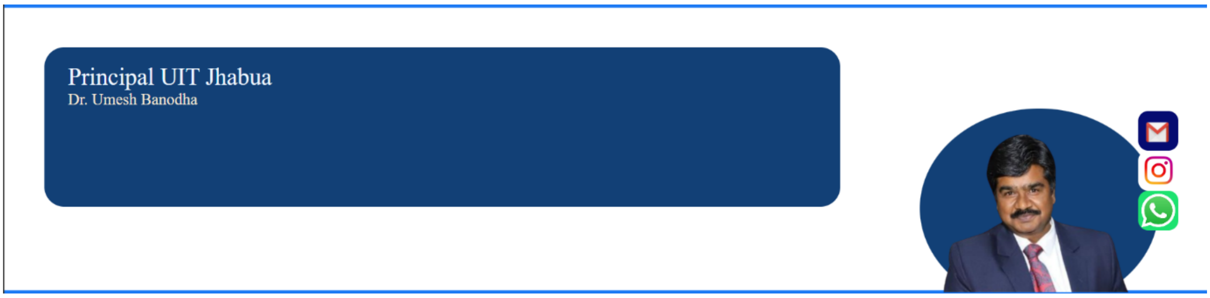


Dr. A.P.J. Abdul Kalam UIT Jhabua

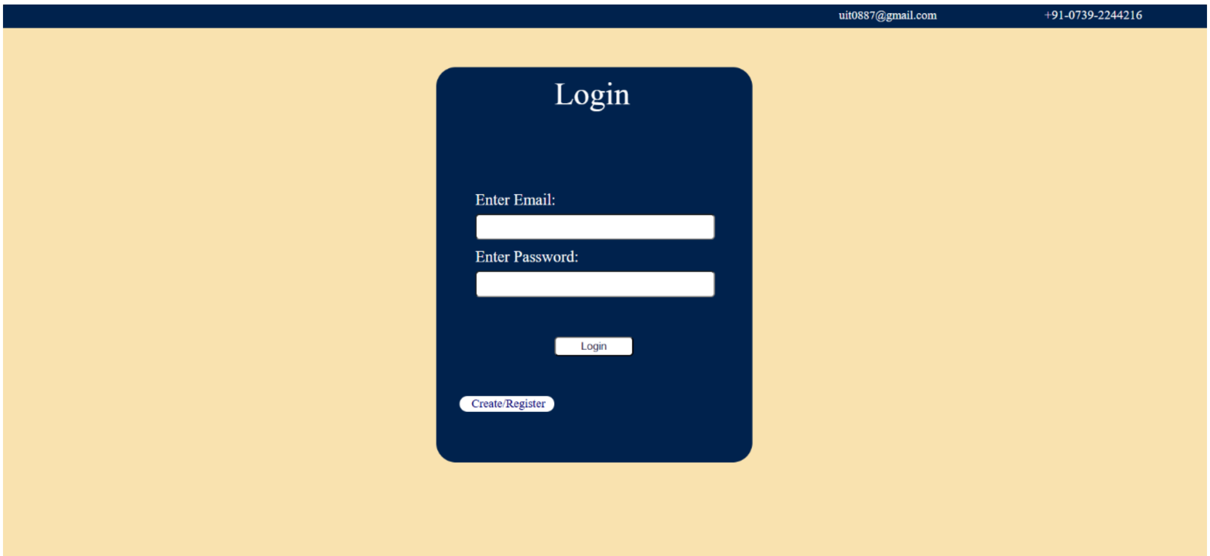

Notice Board

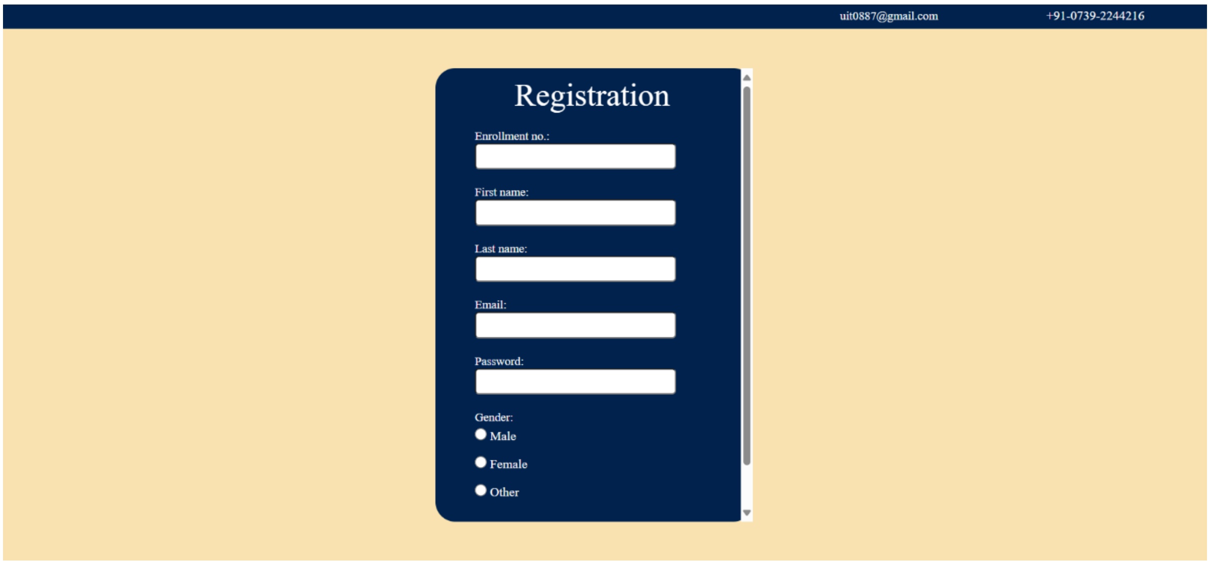
Time table of semester exam of final year .

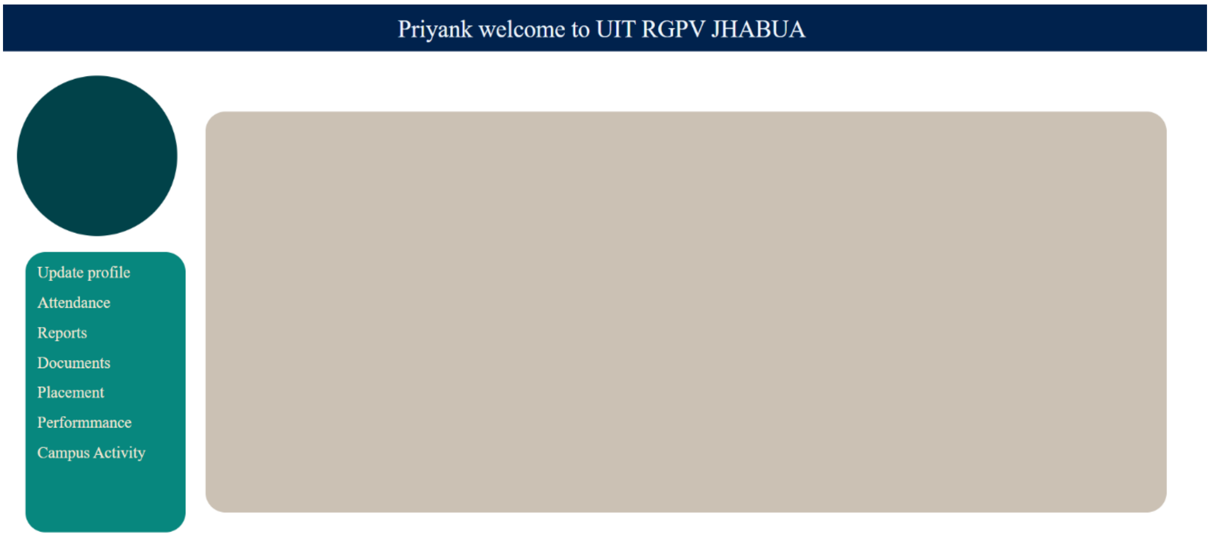
*notice for enrolment of session 2024-25

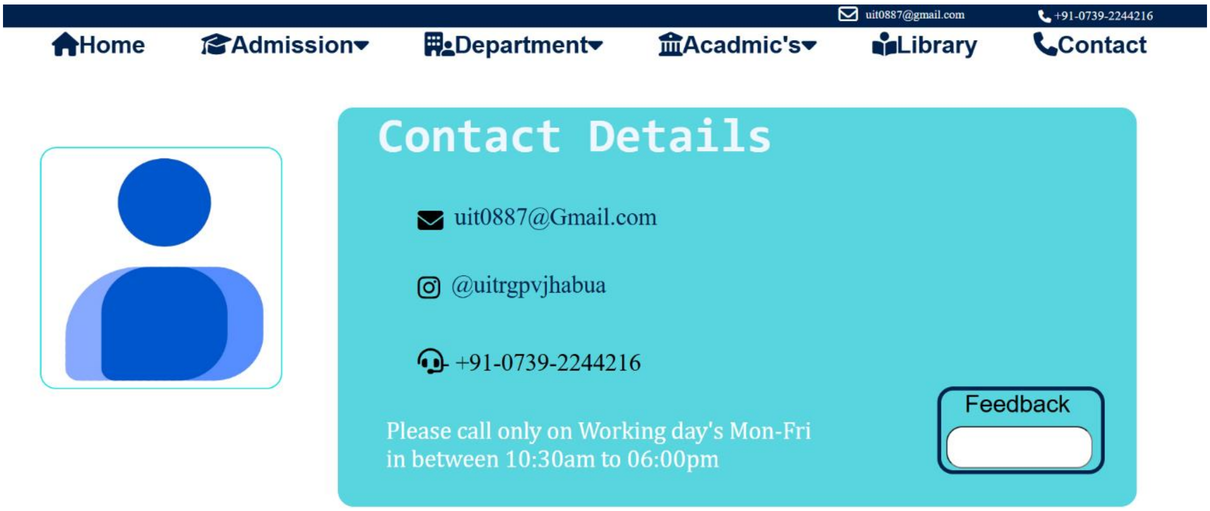

Principal UIT Jhabua

Dr. Umesh Banodha


uit0887@gmail.com
+91-0739-2244216
Login
Enter Email:
Enter Password:
Login
Create/Register


uit0887@gmail.com
+91-0739-2244216
Registration
Enrollment no.:
First name:
Last name:
Email:
Password:
Gender:
○ Male
○ Female
○ Other


Priyank welcome to UIT RGPV JHABUA
Update profile
Attendance
Reports
Documents
Placement
Performmance
Campus Activity



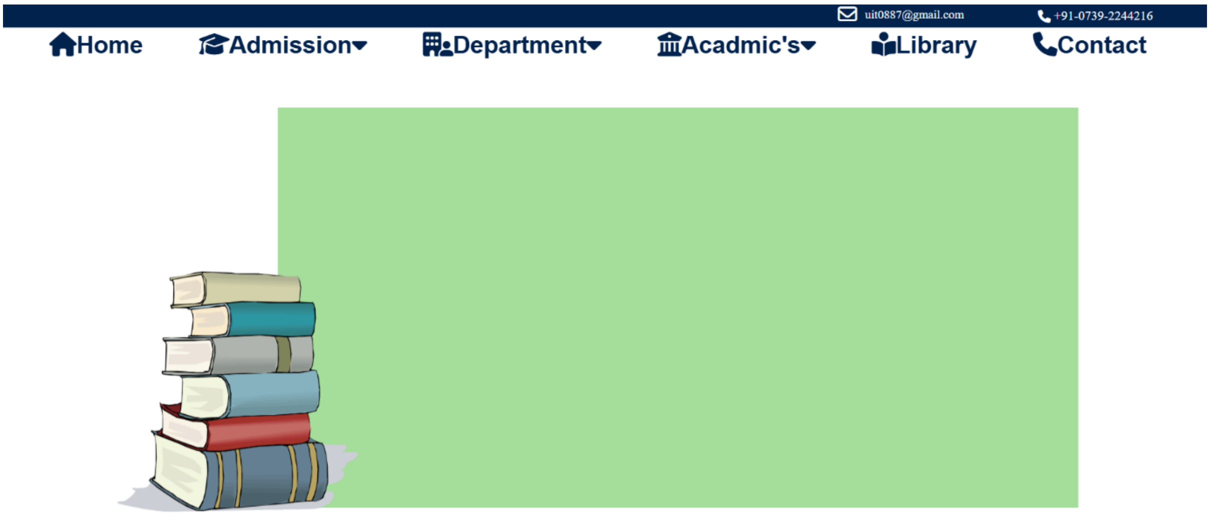
Contact Details

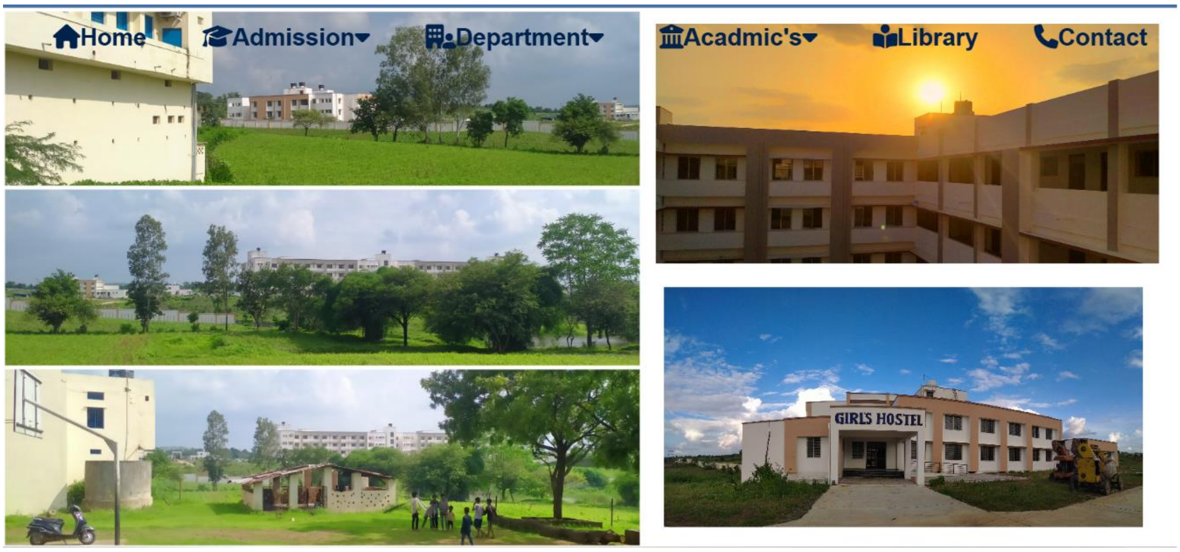
uit0887@Gmail.com

@uitrgpvjhabua

+91-0739-2244216

Please call only on Working day's Mon-Fri
in between 10:30am to 06:00pm





Target users

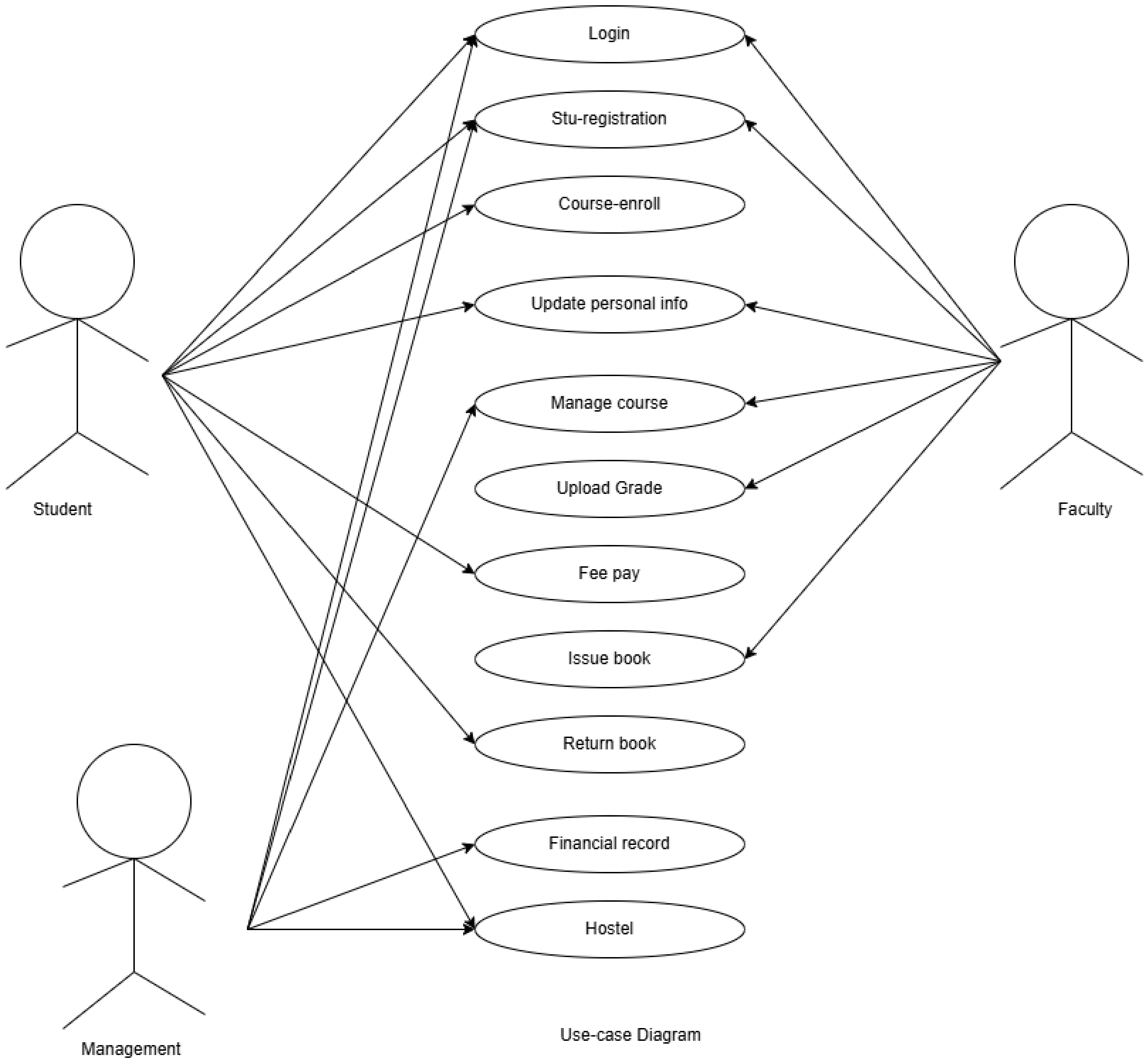
2.6

College Management System is access-able by any one who want to know about the college . But mainly it is for student's related to the course which offer by the institute. And the faculty member and the staff member who were the part of institute.

1. All student's
2. All Faculties
3. Worker related to domain

Use-Case diagrams

2.7



Use-case Diagram

CHAPTER – 3

METHODOLOGY

Agile methodology, known for its iterative and collaborative approach, played a pivotal role in the development of our part-time job portal. This methodology prioritizes customer satisfaction, responsiveness to change, and continuous delivery. The following sections describe how Agile principles and practices were integrated into our project.

1. Agile Principles in Action:

1.1. Customer Collaboration over Contract Negotiation

- We maintained an ongoing dialogue with job seekers, employers, and stakeholders, actively involving them in the development process. This ensured that the portal's features aligned with user needs.

1.2. Individuals and Interactions over Processes and Tools

- We emphasized the importance of communication and collaboration among the development team members. Frequent stand-up meetings, feedback sessions, and close teamwork were integral to our Agile implementation.

1.3. Responding to Change over Following a Plan

- Agile allowed us to be adaptable and responsive to evolving requirements. Change requests were welcomed and integrated, ensuring that the portal evolved as user needs and market conditions changed.

2. Agile Practices Employed:

2.1. User Stories and Backlog

- We used user stories to define features and requirements from the perspective of end users. These user stories were continuously prioritized and added to the product backlog.

2.2. Sprints and Iterations

- The development process was organized into time-boxed iterations (sprints), typically lasting two to four weeks. During each sprint, a set of user stories from the backlog was developed, tested, and made ready for deployment.

2.3. Daily Stand-Up Meetings

- Daily stand-up meetings (Daily Scrums) allowed the team to synchronize their work, discuss progress, and identify and resolve any impediments.

2.4. Continuous Testing and Integration

- Continuous testing and integration practices were followed to ensure that code changes were continuously integrated and tested for quality. This allowed for early bug detection and immediate correction.

2.5. User Acceptance Testing (UAT)

- Frequent UAT was conducted in collaboration with stakeholders to verify that the developed features met user expectations.

---

3. Benefits of Agile:

3.1. Flexibility and Adaptability

- Agile allowed us to respond to user feedback quickly and adjust the project's direction as needed. This flexibility was crucial in accommodating changing market conditions and user preferences.

3.2. Incremental Deliveries

- The iterative nature of Agile ensured that we could deliver value to users early and often. This approach led to a more engaged user base and more rapid adoption.

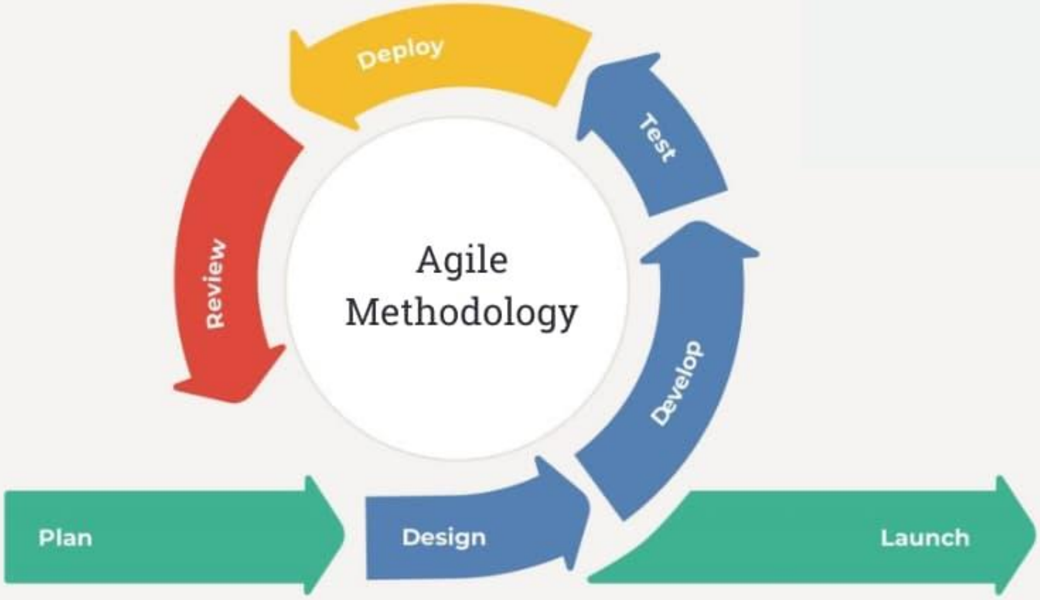
3.3. Transparency and Collaboration

- Agile promoted transparency and collaboration between team members and stakeholders, fostering a sense of shared ownership and responsibility for the project's success.

3.4. User-Centric Development

- By actively involving users throughout the development process, we ensured that the final product met their expectations and needs.

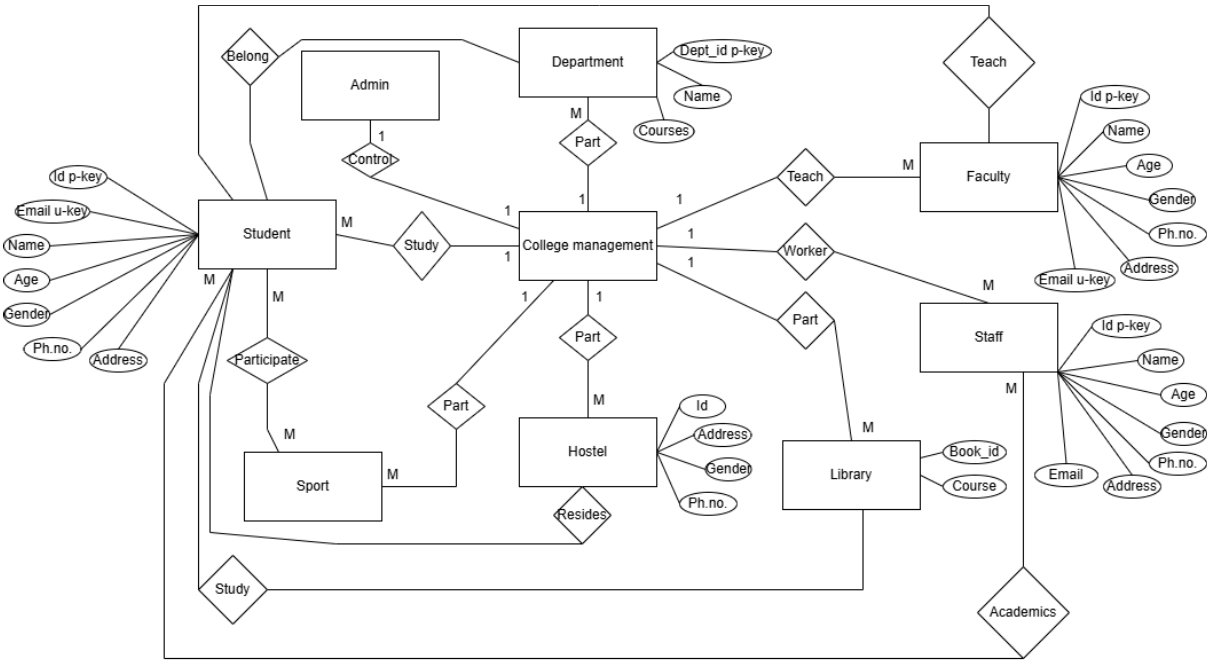
AGILE METHODOLOGY



Chapter-4

Design and Implementation

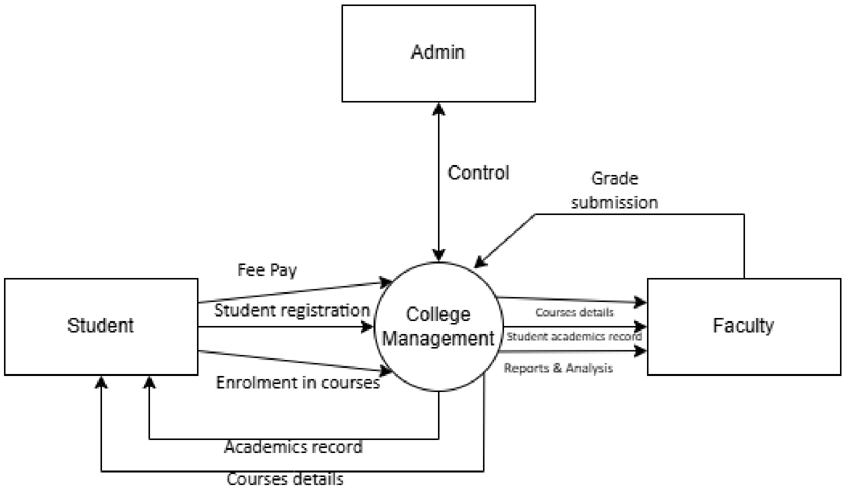
4.1 ER Diagram



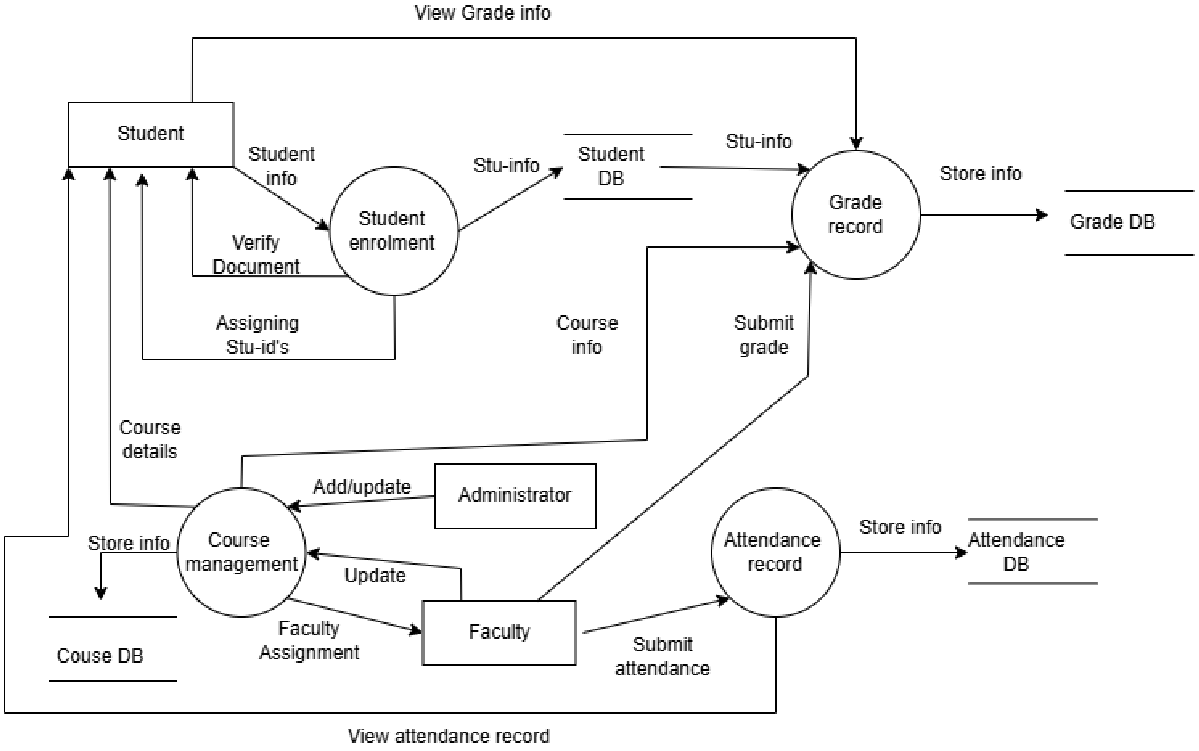
Entities Relationship Diagram of College Management

Data Flow Diagram

4.2



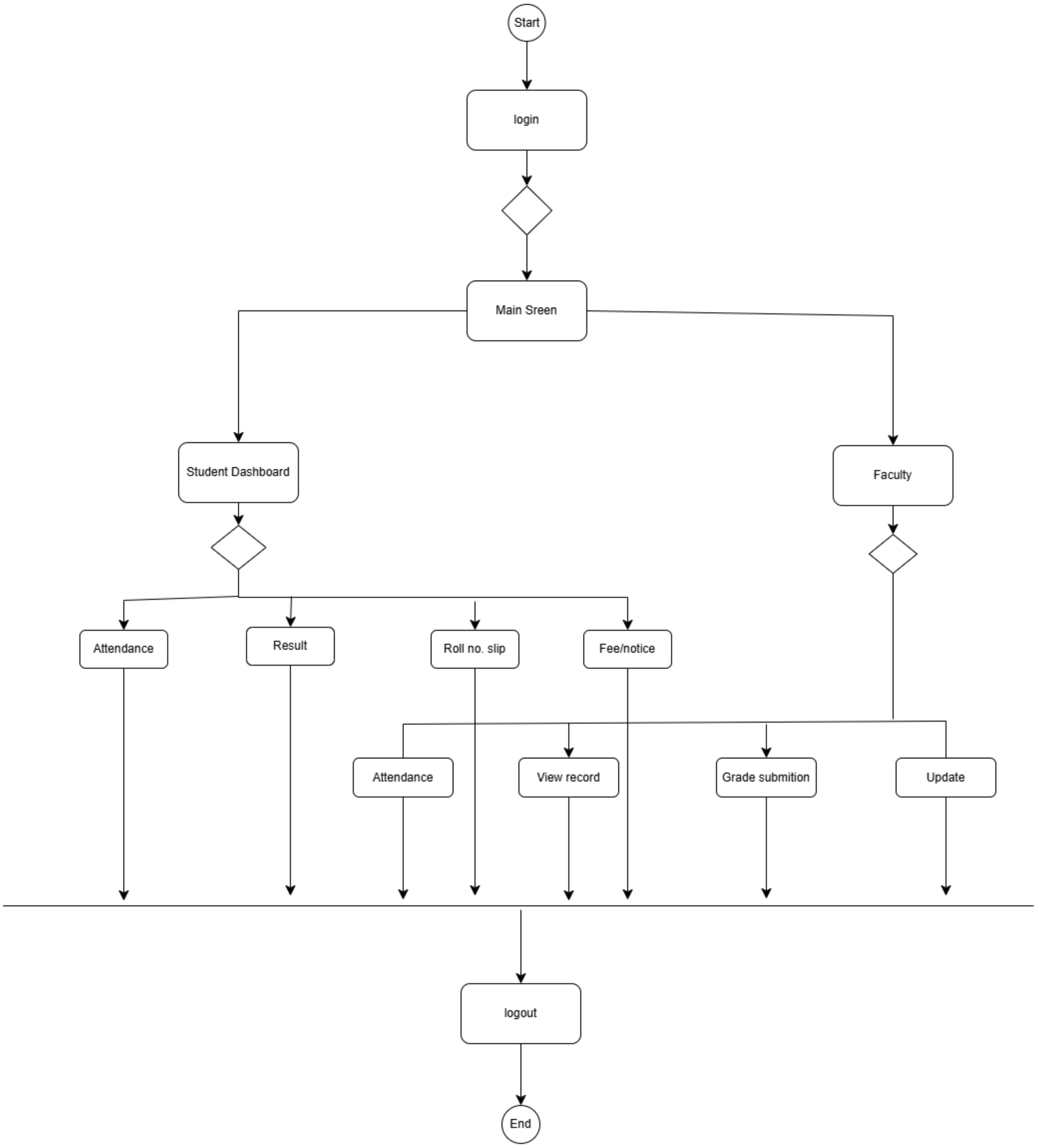
DFD 0-level



DFD 1-level

Activity Diagram

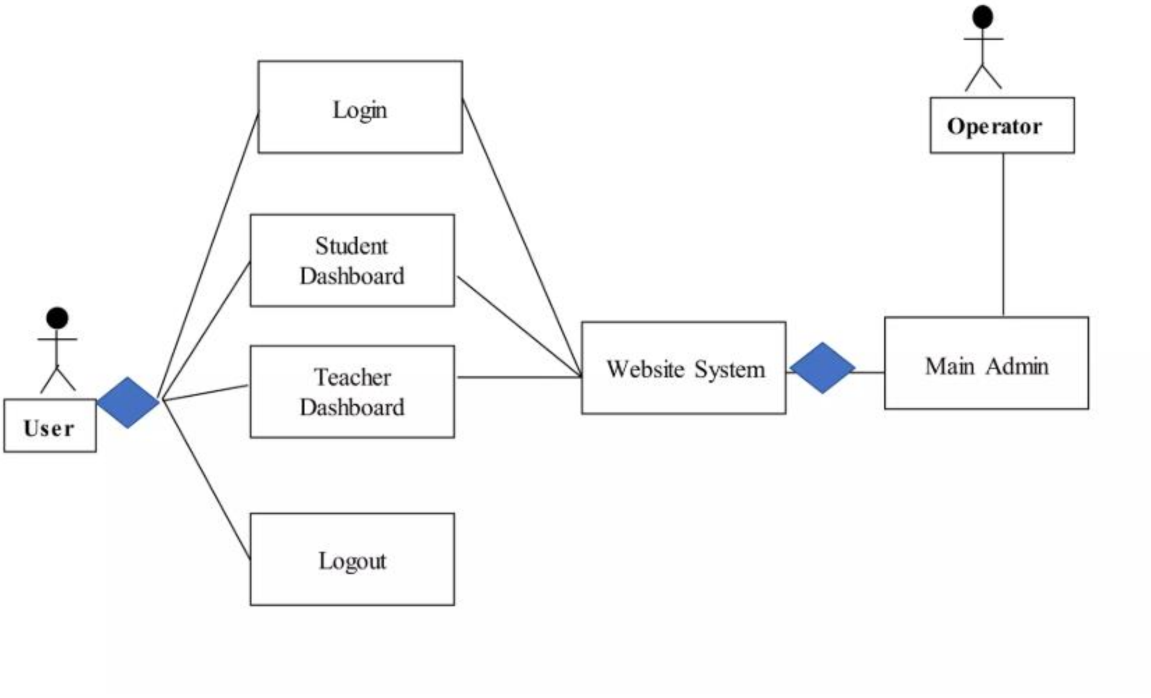
4.3



Activity Diagram

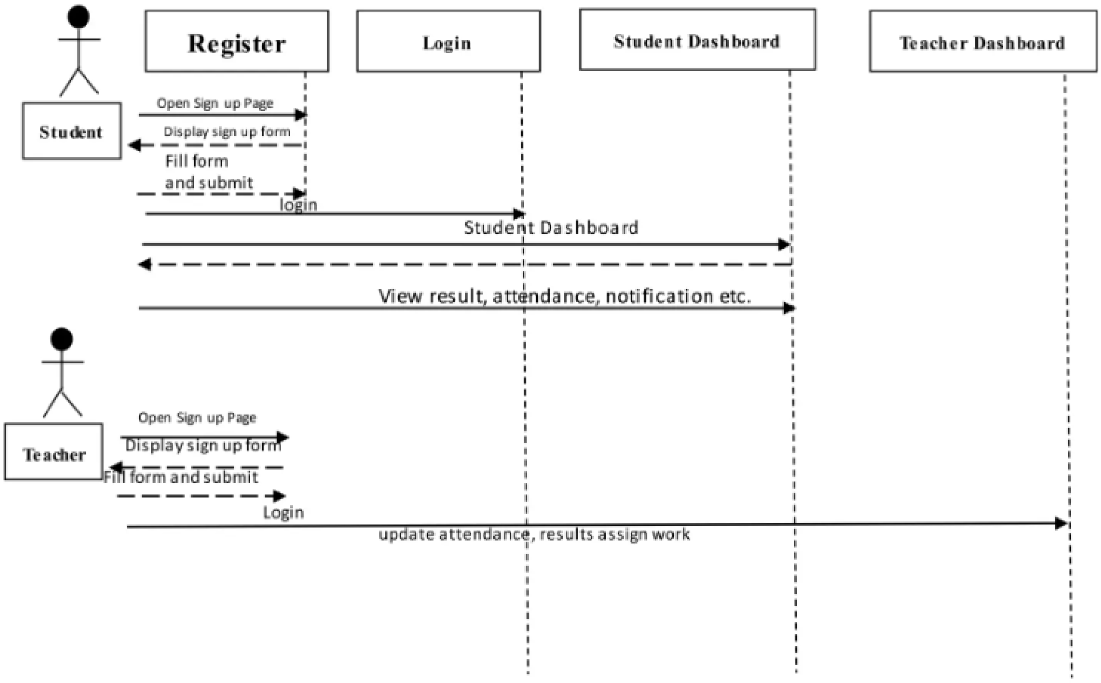
Sequence diagram

4.4



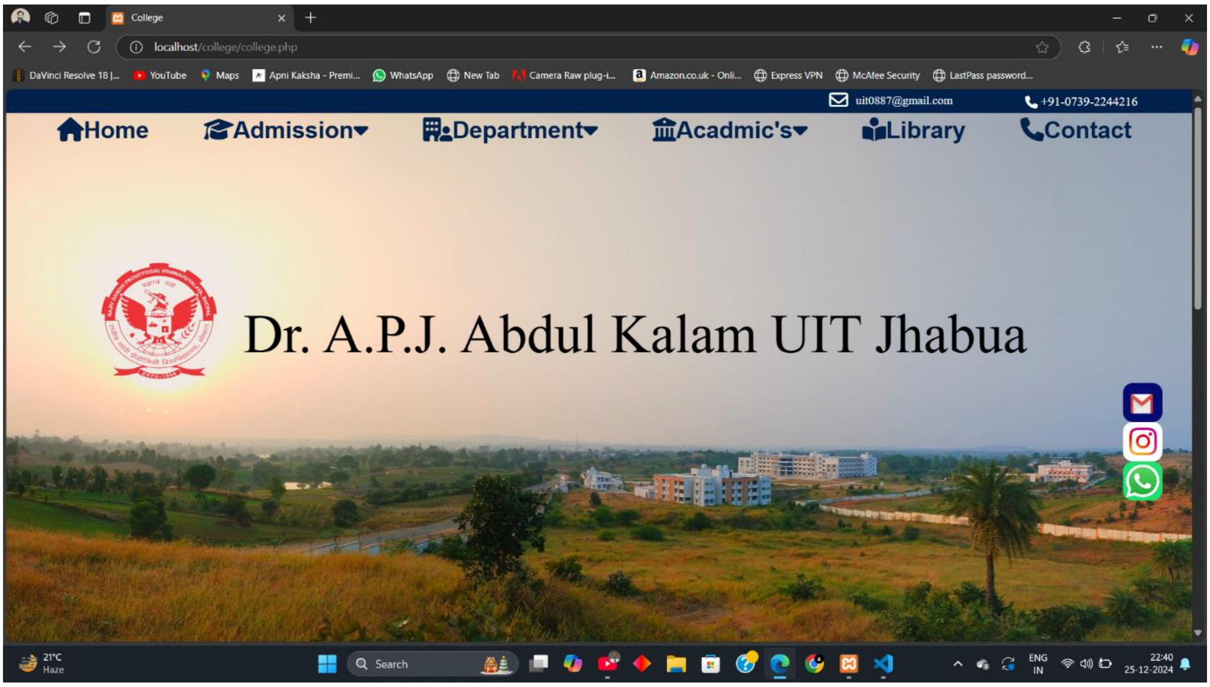
Class Diagram

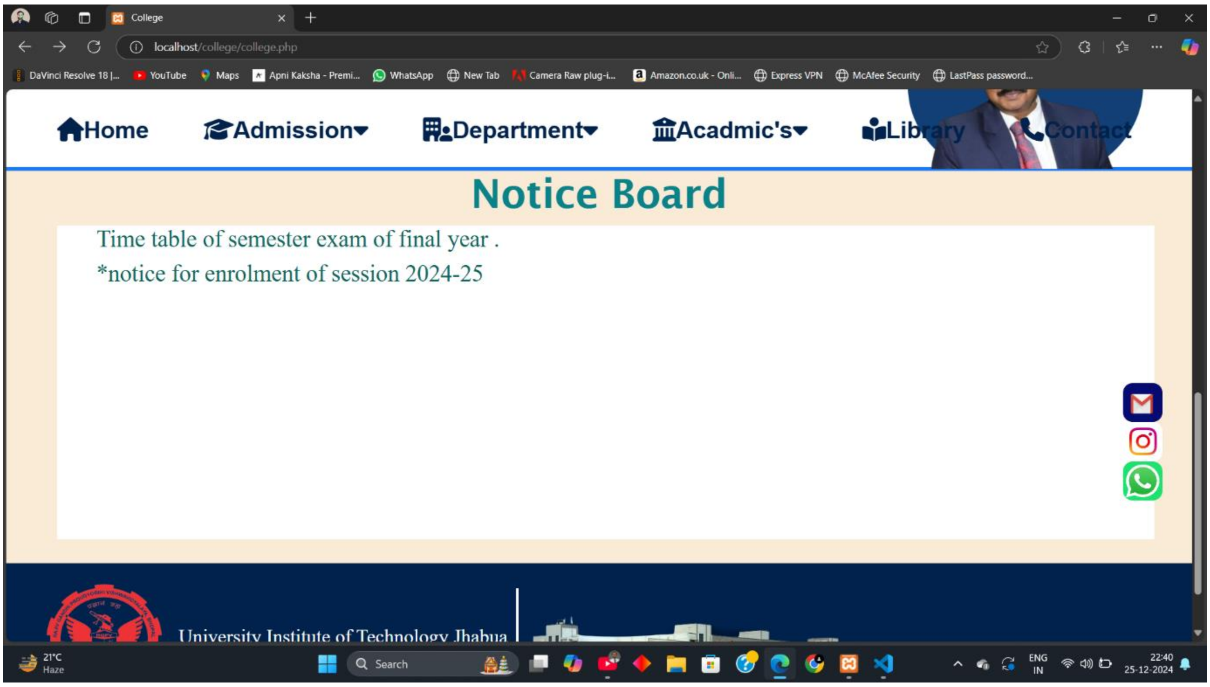
4.5



Chapter-5

Implementation and Testing



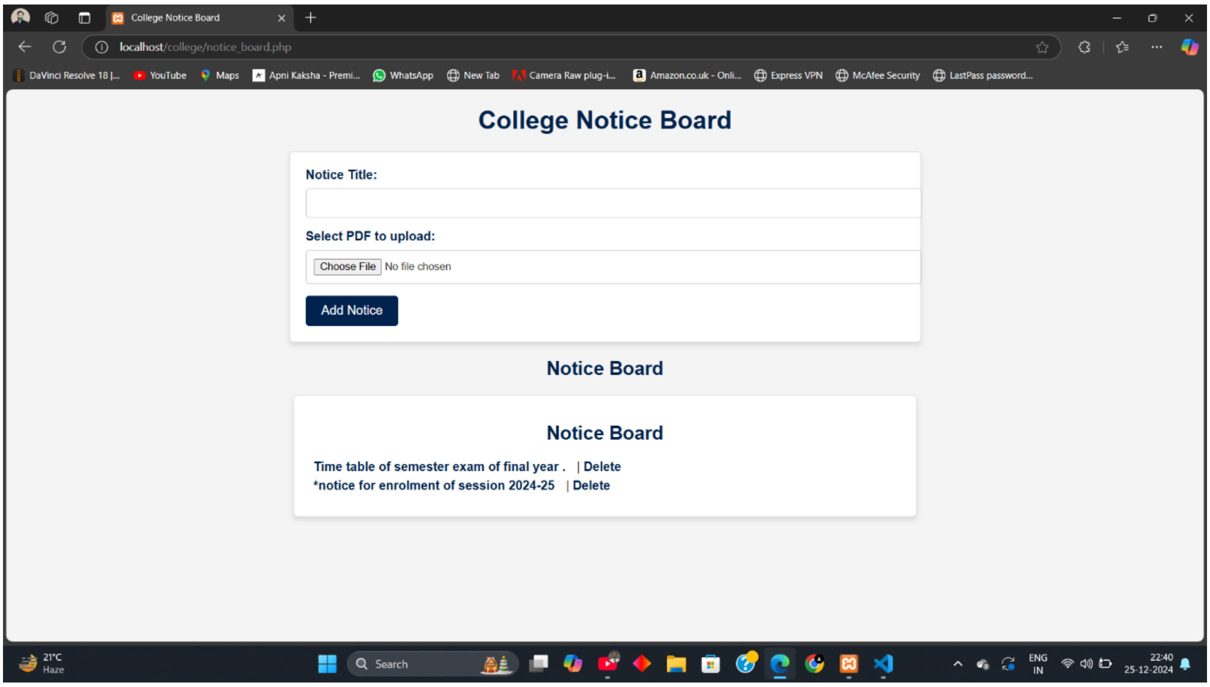


Notice Board

Time table of semester exam of final year .

*notice for enrolment of session 2024-25

University Institute of Technology Jhabua



College Notice Board

Notice Title:

Select PDF to upload:

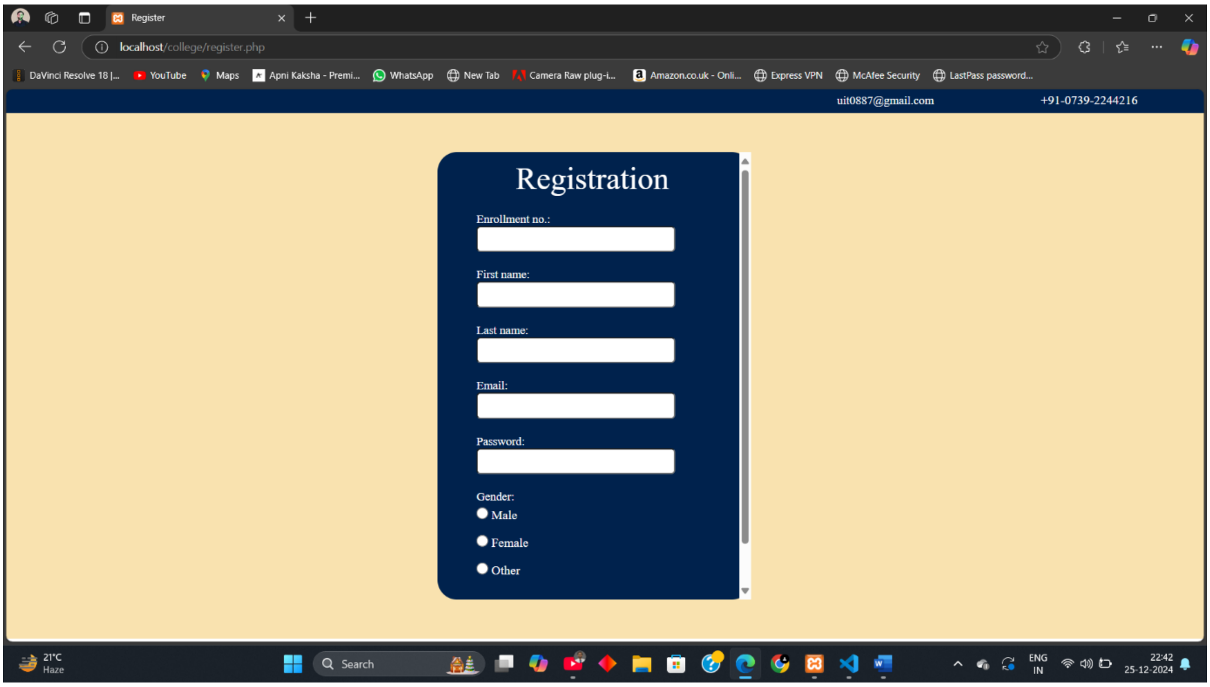
Choose File No file chosen

Add Notice

Notice Board

Notice Board

Time table of semester exam of final year . | Delete
*notice for enrolment of session 2024-25 | Delete



uit0887@gmail.com +91-0739-2244216

Registration

Enrollment no.:

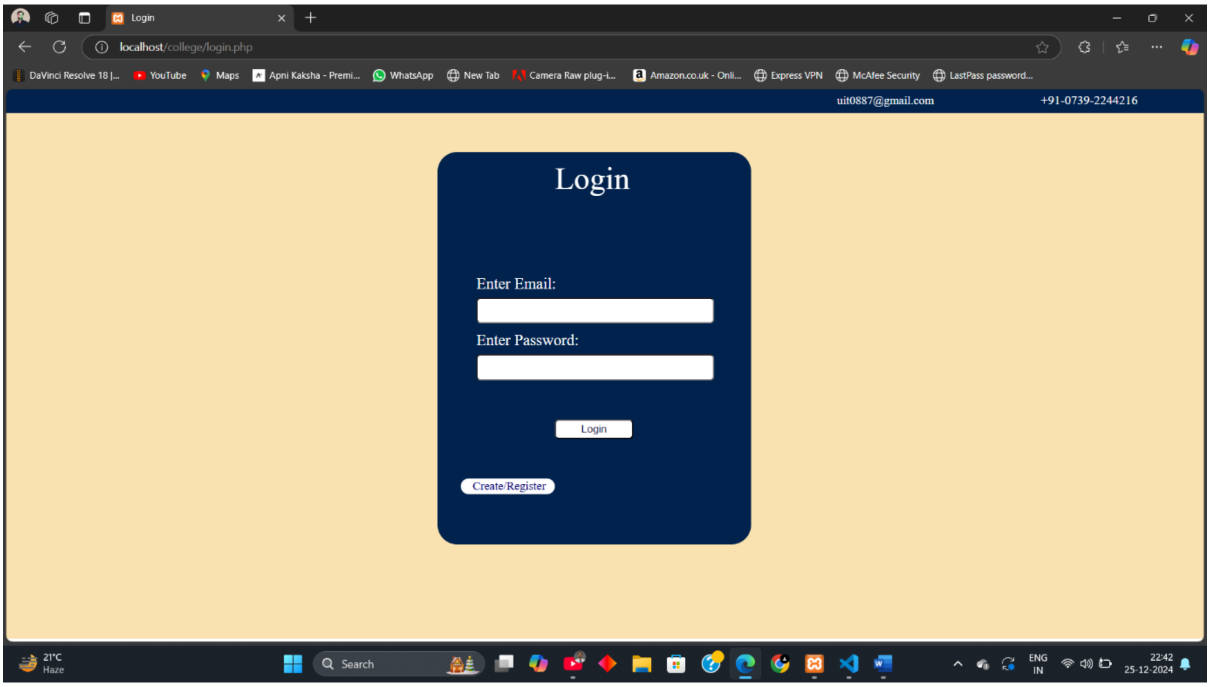
First name:

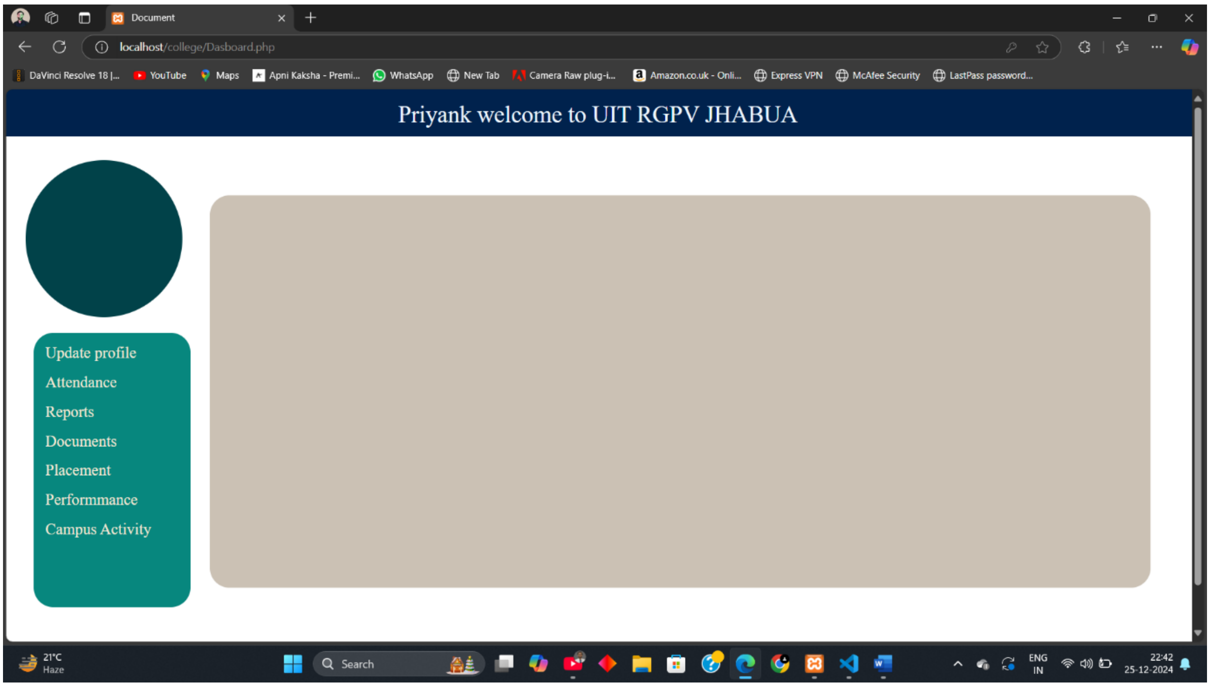
Last name:

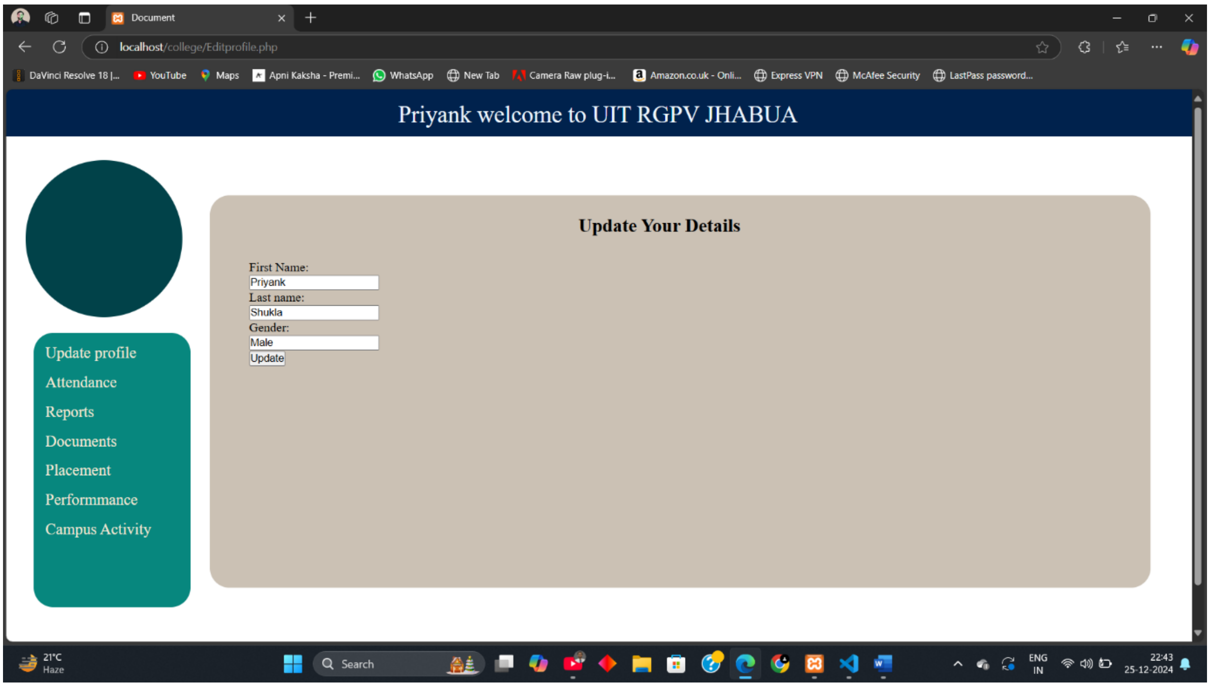
Email:

Password:

Gender:
○ Male
○ Female
○ Other


Login
localhost/college/login.php
uit0887@gmail.com
+91-0739-2244216
Login
Enter Email:
Enter Password:
Login
Create/Register


Document
localhost/college/Dasboard.php
Priyank welcome to UIT RGPV JHABUA
Update profile
Attendance
Reports
Documents
Placement
Performmance
Campus Activity


Document
localhost/college/Editprofile.php
DaVinci Resolve 18 |... YouTube Maps Apni Kaksha - Premi... WhatsApp New Tab Camera Raw plug-i... Amazon.co.uk - Onli... Express VPN McAfee Security LastPass password...
Update profile
Attendance
Reports
Documents
Placement
Performance
Campus Activity
Update
21°C Haze
Search
ENG IN
22:43 25-12-2024

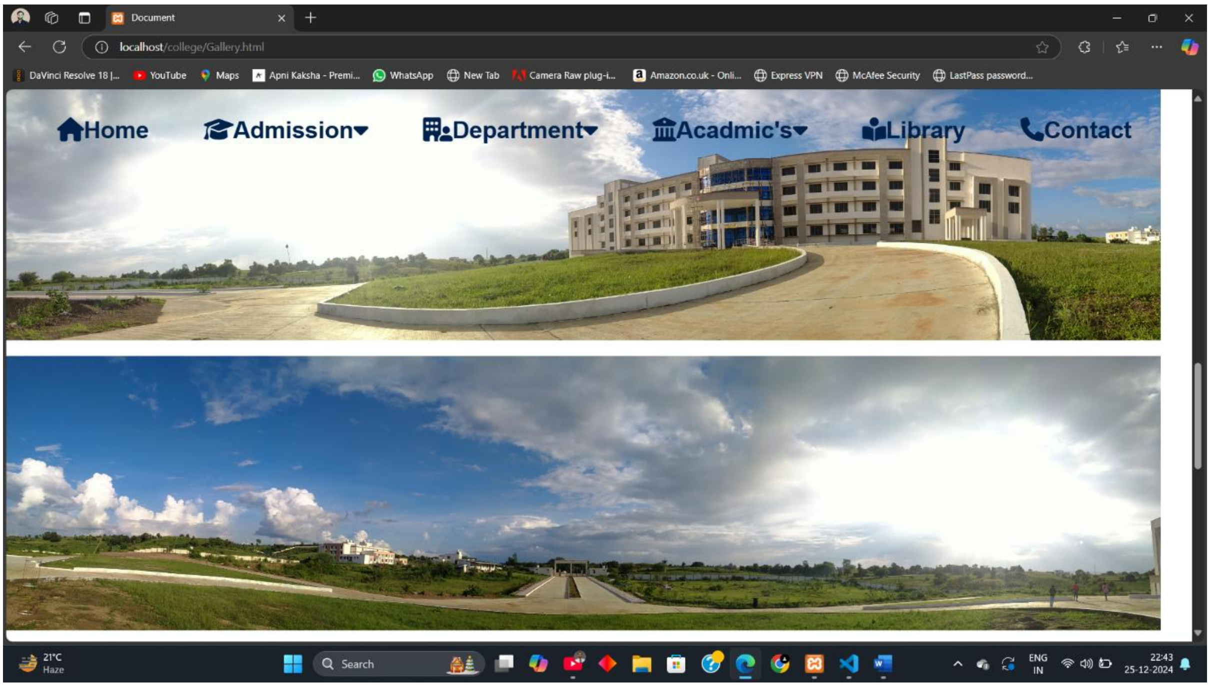
Priyank welcome to UIT RGPV JHABUA

Update Your Details

First Name: Priyank

Last name: Shukla

Gender: Male


Document
localhost/college/Gallery.html
DaVinci Resolve 18 |... YouTube Maps Apni Kaksha - Premi... WhatsApp New Tab Camera Raw plug-i... Amazon.co.uk - Onli... Express VPN McAfee Security LastPass password...
Home
Admission
Department
Acadmic's
Library
Contact
21°C Haze
Search
ENG IN
22:43 25-12-2024

References

Draw.io

Chat-GPT

Senior Project file