ONLINE FEE PAYMENT SYSTEM FOR STUDENTS

Minor Project Report Submitted to

SRI PADMAVATI MAHILA VISVAVIDYALAYAM

in Partial fulfilment of the requirement for the

MASTER OF COMPUTER APPLICATIONS

III SEMESTER

By

ANASURI SRAVANI (2023MCA16003)

BADIREDDI LAVANYA (2023MCA16010)

Under the guidance of

Prof. T. SUDHA

Accredited by **NAAC with A**$^{+}$ Grade ISO 9001: 2015 Certified

DEPARTMENT OF COMPUTER SCIENCE

SRI PADMAVATI MAHILA VISVAVIDYALAYAM

(Women's University)

Tirupati-517502(A.P), Andhra Pradesh

FEBRUARY, 2025

DEPARTMENT OF COMPUTER SCIENCE

SRI PADMAVATI MAHILA VISVAVIDYALAYAM

(Women's University)

Tirupati - 517502, Andhra Pradesh, India Accredited with

A+ Grade by NAAC

CERTIFICATE

This is to certify that the project work entitled "**ONLINE FEE PAYMENT SYSTEM FOR STUDENTS"** is a Bonafide record of work carried out by ANASURI SRAVANI (2023MCA16003) and BADIREDDI LAVANYA (2023MCA16010) in the **Department of Computer Science**, **Sri Padmavati Mahila Visvavidyalayam**, Tirupati in partial fulfilment of the requirements of III Semester of **MASTER OF COMPUTER APPLICATIONS**. The content of the Project Report has not been submitted to any other University / Institute for the award of any degree.

Guide ***Head of the Department***

DECLARATION

We hereby declare that MCA III Semester Minor Project entitled **"ONLINE FEE PAYMENT SYSTEM FOR STUDENTS"** was done at the **Department of Computer Science,Sri Padmavati Mahila Visvavidyalayam**, Tirupati, in the year 2024-2025 under the guidance of **Prof. T SUDHA** in partial fulfilment of requirements of MCA III Semester. We also declare that this project is our original contribution of the best of our knowledge and belief. We further declare that this work has not been submitted for the award of any other degree of this or any other university/Institution.

Signature of the Students

ACKNOWLEDGEMENT

We are greatly indebted to our guide **Prof. T. SUDHA** for taking keen interest on our project work and providing valuable suggestions in all the possible areas of improvement.

We express our sincere thanks to the teaching staff of the Department of Computer Science for extending support and encouragement to us in all the stages of the project work.

We gratefully acknowledge and express our gratitude to the non-teaching staff of the Computer Science Department who supported us in preparing the project report.

Signature of the Students

INDEX

ABSTRACT

Online Fee Payment System for Students is a streamlined digital platform designed to simplify the fee payment process for students in educational institutions. The system offers an intuitive and user-friendly interface that enables students to securely pay various types of fees, including tuition, examination, and hostel fees, with ease. Without the need for backend technologies, the platform efficiently manages student data using local storage, ensuring quick access and smooth performance. Students can view their personal details and make payments through simple forms, enhancing convenience and reducing administrative workload. The system also ensures secure transactions and generates automated receipts upon successful payments. The ultimate goal of this project is to provide an accessible, efficient, and secure fee management solution that promotes ease of use while minimizing manual effort for both students and educational institutions.

1. INTRODUCTION

The **Online Fee Payment System for Students** is a web-based application designed to simplify and automate the process of paying educational fees. This system provides a convenient, efficient, and secure platform for students to make payments related to tuition, examination, and hostel fees without the need for physical transactions. Developed with a focus on user-friendly functionality, the system eliminates the complexities associated with traditional fee payment methods, such as long queues, manual paperwork, and time-consuming verification processes.

This system operates entirely on front-end technologies, with student data being securely managed through the browser's **localStorage**. By eliminating the need for backend servers or databases, the platform ensures faster performance, ease of access, and reduced system complexity. The absence of backend infrastructure also reduces development costs and makes the system lightweight, making it suitable for institutions that require simple, scalable solutions.

The platform provides a **personalized dashboard** where students can view their details, including name, enrollment number, course, and other relevant academic information, which are retrieved directly from localStorage. Students can effortlessly fill out forms for various fee categories, submit their payments, and receive instant confirmations. The system automatically validates form inputs to ensure accuracy and completeness before processing the payment.

Security is a key consideration in the system’s design. Although it does not involve complex backend security protocols, the platform implements front-end validation techniques to ensure data integrity. Additionally, the system generates **automated payment confirmations**, reducing the chances of errors and offering students immediate assurance of their successful transactions.

The key objective of this project is to create an accessible, reliable, and secure fee management solution that reduces administrative efforts, minimizes errors, and enhances the overall efficiency of fee collection processes for educational institutions. By offering a

paperless, cashless, and straightforward process, the Online Fee Payment System contributes to a more organized, eco-friendly, and modern approach to managing educational finances.

1.1 UNIVERSITY PROFILE

Sri Padmavati Mahila Visvavidyalayam (SPMVV) is a prestigious women's university located in Tirupati, Andhra Pradesh. It was established in the year of 1983 by Sri N. T. Rama Rao, the Former Chief Minister of Andhra Pradesh. The university has earned recognition for its academic excellence, particularly in women's education, offering undergraduate, postgraduate, and research programs across various fields.

It has also been accredited by the National Assessment and Accreditation Council (NAAC) with an 'A+' grade. The university is recognized for its commitment in providing a supportive and inclusive learning environment, enabling women to excel in their careers and significantly contribute to society.

Objectives:

- **Empowerment of Women**: To provide higher education opportunities that enhance the social, economic, and intellectual empowerment of women.
- **Academic Excellence**: To offer academic programs that promote excellence, innovation, and research in diverse fields.
- **Holistic Development**: To foster the overall development of students, including intellectual, moral, and emotional growth.
- **Community Service**: To contribute to social development through outreach programs and community service initiatives.
- **Global Competence**: To equip women with the skills and knowledge necessary for success in global markets and societal advancement.

2. PROBLEM DEFINITION

2.1 AIM

The aim of the **Online Fee Payment System for Students** is to develop a simple, efficient, and secure web-based platform that enables students to easily pay tuition, examination, and hostel fees. The system eliminates the need for manual processes by using front-end technologies and **localStorage** for data management, ensuring a smooth, user-friendly, and paperless fee payment experience.

2.2 PROBLEM DEFINITION

Traditional fee payment methods create inefficiencies, errors, and administrative burdens, impacting students and institutions. Manual fee collection often results in delays and inaccuracies, making financial management difficult.

Key Problems:

1. **Inefficient Traditional Processes** – Long queues and paperwork create inconvenience.
2. **Human Errors** – Manual data entry leads to payment discrepancies.
3. **Limited Accessibility** – Students lack a remote-friendly digital payment option.
4. **Administrative Burden** – Manual tracking increases staff workload.

The system addresses these issues by offering a secure, streamlined, and accessible web-based solution.

2.2.1 EXISTING SYSTEM

Current systems for student fee payment are largely dependent on traditional, manual processes or fragmented digital solutions that lack the integration necessary to provide a seamless and efficient payment experience. These systems include in-person fee collection at administrative offices, basic online payment portals, and third-party payment applications, each presenting unique challenges:

1. **Manual Payment Processes:** Many institutions still rely on physical, paper-based systems where students must stand in long queues, fill out forms, and make payments in person, leading to time-consuming procedures.

2. **Fragmented Digital Solutions:** Some institutions use basic online portals that are not fully integrated with student records, requiring separate platforms to view details, make payments, and confirm transactions.

3. **High Risk of Errors:** Manual data entry in traditional systems increases the chances of errors in recording payment information, causing discrepancies that can lead to disputes and delays.

4. **Limited Accessibility:** Students often face challenges accessing payment systems remotely, especially if the platforms are not mobile-friendly or require specific technical setups.

5. **Lack of Real-Time Confirmation:** In many systems, students do not receive immediate confirmation of their payments, causing uncertainty and the need for follow-ups with administrative staff.

6. **Administrative Overload:** Fee collection, verification, and record-keeping place a significant burden on administrative staff, reducing efficiency and increasing the risk of mismanagement.

These limitations hinder the efficiency, accuracy, and convenience of fee payment processes, both for students and educational institutions. The **Online Fee Payment System for Students** aims to address these gaps by providing an integrated, user-friendly, and accessible platform that simplifies fee management while reducing administrative workload.

2.3 OBJECTIVES

The primary objective of the **Online Fee Payment System for Students** is to develop a simple, efficient, and secure web-based platform that streamlines the fee payment process for students and educational institutions. This system aims to:

1. **Simplify the Fee Payment Process:** Provide an easy-to-use interface for students to submit various types of fees, including tuition, examination, and hostel fees, eliminating the need for physical transactions.

2. **Enhance Efficiency:** Automate key tasks such as data entry, fee calculations, and payment confirmations to reduce administrative workload and improve overall operational efficiency.

3. **Ensure Secure Transactions:** Implement secure data handling and transaction protocols to guarantee the confidentiality and integrity of student payment information.

4. **Improve Accessibility:** Offer a platform that can be accessed by students from any device with an internet connection, ensuring the system is available at all times and from any location.

5. **Reduce Errors and Discrepancies:** Minimize human errors by automating processes and ensuring accurate data management, leading to fewer discrepancies in payment records.

6. **Promote a Paperless Environment:** Facilitate digital fee payments and reduce reliance on physical paperwork, contributing to a more sustainable, eco-friendly approach to fee management.

This system provides a reliable and accessible fee management solution, simplifying the process while reducing administrative workload.

3. SYSTEM ANALYSIS

3.1 Software Requirements Specifications

Introduction

Purpose

The purpose of the **Online Fee Payment System for Students** is to simplify and streamline the fee payment process within educational institutions. This document specifies the system's functionalities, performance expectations, and technical requirements. It aims to provide clarity to all stakeholders, ensuring that the final product meets the desired objectives, reduces administrative workload, and offers students a hassle-free payment experience.

Scope

The scope of the **Online Fee Payment System for Students** is to develop a web-based platform that allows students to securely and conveniently pay their academic-related fees, including tuition, examination, and hostel fees.

Functional Requirements

These requirements define what the system must do:

User Management

- Allow students to register with basic details (name, email, password).
- Provide secure login/logout functionality to manage student sessions.

Dashboard

- Display student details like name, enrollment number, course, and fee-related information retrieved from localStorage.

Fee Payment Module

- Offer separate forms for different types of fees (tuition, examination, hostel) with fields for entering relevant payment details.

Payment Confirmation

- Display immediate confirmation messages after successful payment form submission.
- Provide a summary of the payment details for student reference.

LocalStorage Management

- Handle the storage and retrieval of student data securely within the browser using localStorage.

Ensure data persistence across sessions within the constraints of localStorage.

Non-Functional Requirements

These requirements define how the system should perform:

Performance

- The website should load within 2-3 seconds on a standard internet connection.
- Support smooth navigation and quick form submissions without noticeable delays.

Security

- Protect sensitive user data by implementing front-end validation and security measures to prevent unauthorized access.
- Ensure that student information stored in localStorage is handled securely, with appropriate data validation to prevent tampering.

Usability

- The interface should be intuitive, allowing students with basic web navigation skills to use the system effortlessly.
- Provide a responsive design that ensures usability across devices, including desktops, tablets, and smartphones.

Scalability

- The system should be designed in a modular way to allow for future enhancements, such as payment history tracking or integration with real-time payment gateways.

Maintainability

- Code should be clean, modular, and well-documented to support easy maintenance and future updates.

Portability:

Ensure compatibility with major web browsers (Google Chrome, Mozilla Firefox, Microsoft Edge, Safari).

Reliability

- Ensure the system is stable with minimal crashes, maintaining data integrity within localStorage.

3.2 System Requirements

3.2.1 Hardware Requirements:

These are the hardware essentials required to develop and utilize the **Online Fee Payment System for Students**:

Development Environment

- A system with at least **4 GB RAM** and **128 GB SSD** for smooth front-end development.
- **Stable internet connectivity** for accessing external libraries, frameworks, and hosting services.

3.2.2 Software Requirements (Tools and Frameworks):

These are the technical requirements for the development of the **Online Fee Payment System for Students**:

Frontend

- **HTML, CSS, JavaScript** for building the structure, design, and interactive functionalities of the website.

Data Management

localStorage for storing and managing student details and fee-related data directly in the browser.

3.3 Feasibility Study

3.3.1. Operational Feasibility

Operational feasibility determines whether the **Online Fee Payment System for Students** will function effectively in the intended environment and meet the needs of its users and stakeholders.

Target Audience:

- The system is designed for **students**, **academic staff**, and **administrative personnel** who manage fee-related processes. Its simple and user-friendly interface ensures accessibility for users with varying levels of technical proficiency.

Ease of Use:

- The platform features an **intuitive design** with clear navigation, allowing students to easily fill out fee forms and view their details stored in the system.

Long-Term Viability:

- The system can be easily **updated or modified** to add new features, improve the interface, or enhance data validation mechanisms.

3.3.2. Technical Feasibility

Technical feasibility assesses whether the project can be developed and implemented using the available technology, tools, and resources.

Technology Availability:

- The project uses standard web technologies like **HTML, CSS, and JavaScript**.
- All required tools and libraries are **widely available** and supported across modern browsers.

Development Tools:

- Development is supported by tools like **Visual Studio Code** for coding, **GitHub** for version control, and **browser developer tools** for debugging.

Cost-Benefit Analysis:

- The system reduces **manual administrative work**, minimizes paperwork, and eliminates the need for traditional fee collection processes, leading to **cost savings** for educational institutions.

3.4 Modeling Approaches

3.4.1 UML Diagrams:

The **Unified Modeling Language (UML)** is a standardized modeling language used to visualize, specify, construct, and document the components of a software system. It helps in creating clear and structured blueprints for system development, ensuring better understanding and communication among project

Why We Need UML:

- **Efficient Collaboration:** Complex systems like the **Online Fee Payment System for Students** involve multiple teams.
- **Bridging Technical Gaps:** Non-technical stakeholders, such as educational administrators, may not understand programming code. UML helps them visualize system workflows, functionalities, and data flows in an easy-to-understand format.
- **Improved System Design:** UML helps identify design issues early in the development process, saving time and reducing costs. It allows teams to **visualize user interactions, workflows, and the system's architecture** before actual implementation.

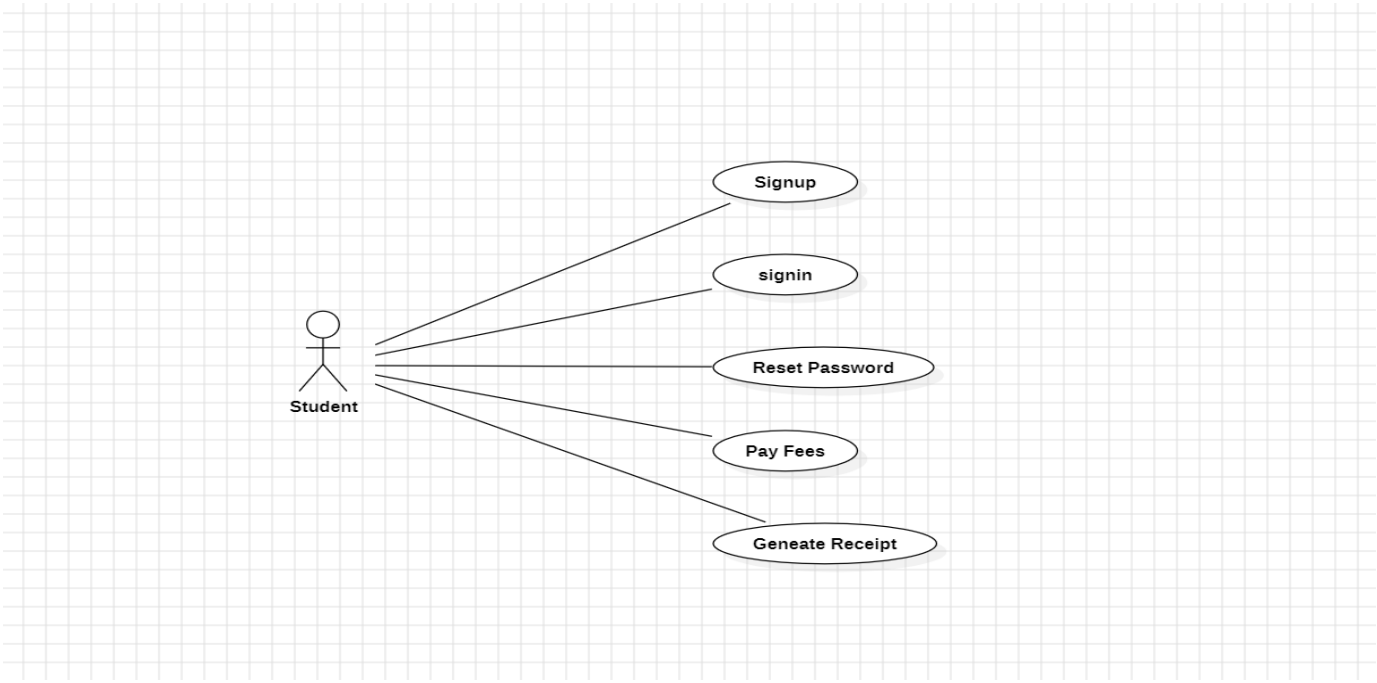
For the **Online Fee Payment System for Students**, the following UML diagrams can be created to represent different aspects of the system:

- **Use Case Diagram:** Illustrates the interactions between users (students, administrators) and the system, covering activities such as student registration, fee payment, and form submission.
- **Class Diagram:** Represents the static structure of the system, showing entities like **Student, Fee Form, and Payment Details**, along with their attributes, operations, and relationships.

- **Activity Diagram:** Depicts the workflow of processes like **fee submission, data validation, and confirmation generation**, showing the sequence of actions and decision points.
- **Sequence Diagram:** Describes the sequence of interactions between the system components and users during specific operations, such as **logging in, filling out fee forms, and receiving payment confirmation**.
- **Collaboration Diagram:** Illustrates the modular structure of the system, showing how the front-end components (like UI forms) and data-handling modules interact.

3.4.1.1 Use Case Diagram

The **Use Case Diagram** for the **Online Fee Payment System** represents how a **student interacts** with the system to perform tasks like **signing in, selecting a fee type, making payments, and generating receipts**. It defines the system's functional requirements and provides a **clear overview of user interactions with various system features**. This diagram helps identify the **actors (users)** and their interactions with **different functionalities**, ensuring a well-structured **user flow** for payment processing.

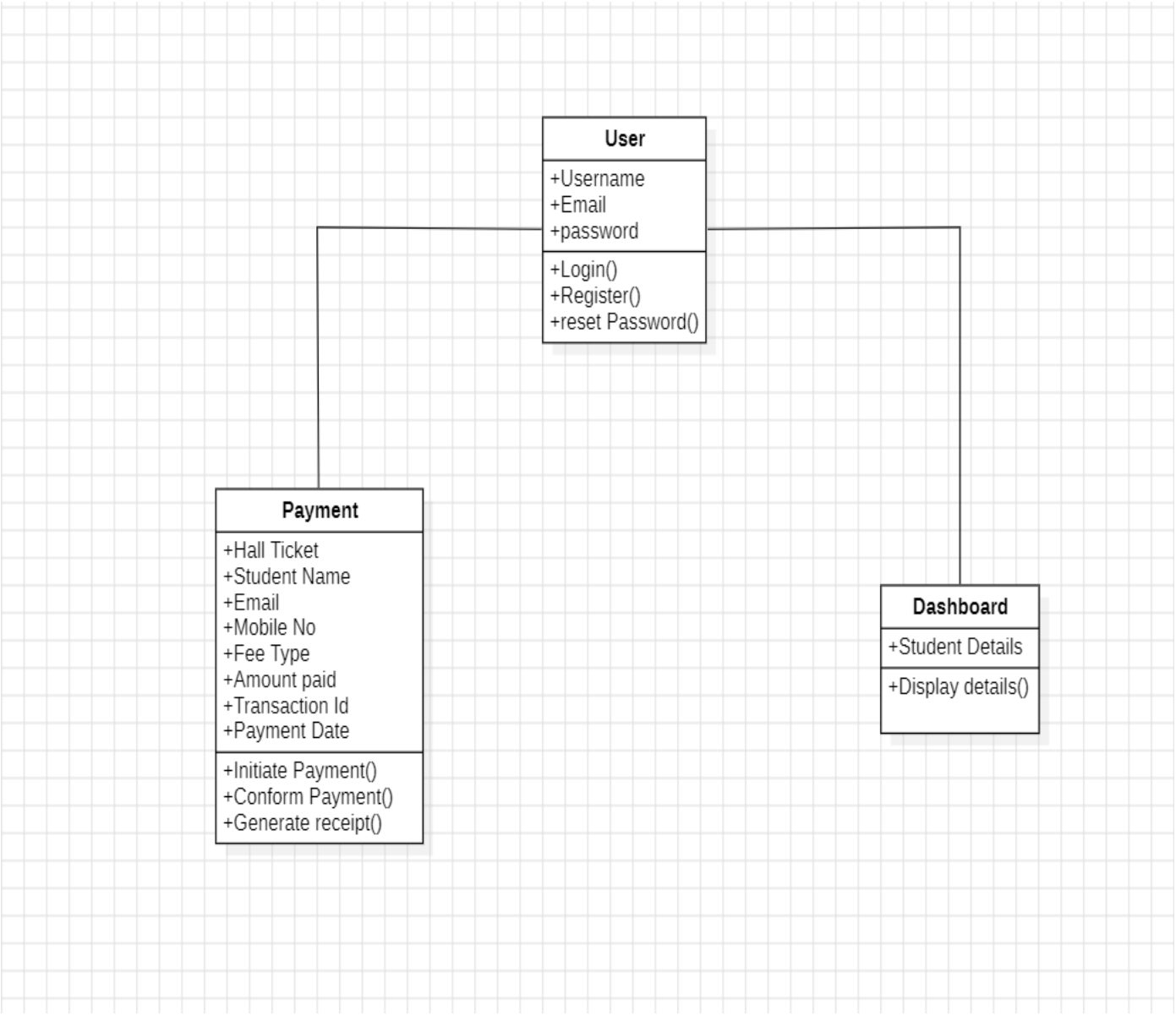


Use Case Diagram for Online Fee Payment System for students

STATE DIAGRAM OF VIRTUAL DRIVE MASTER

3.4.1.2 Class Diagram

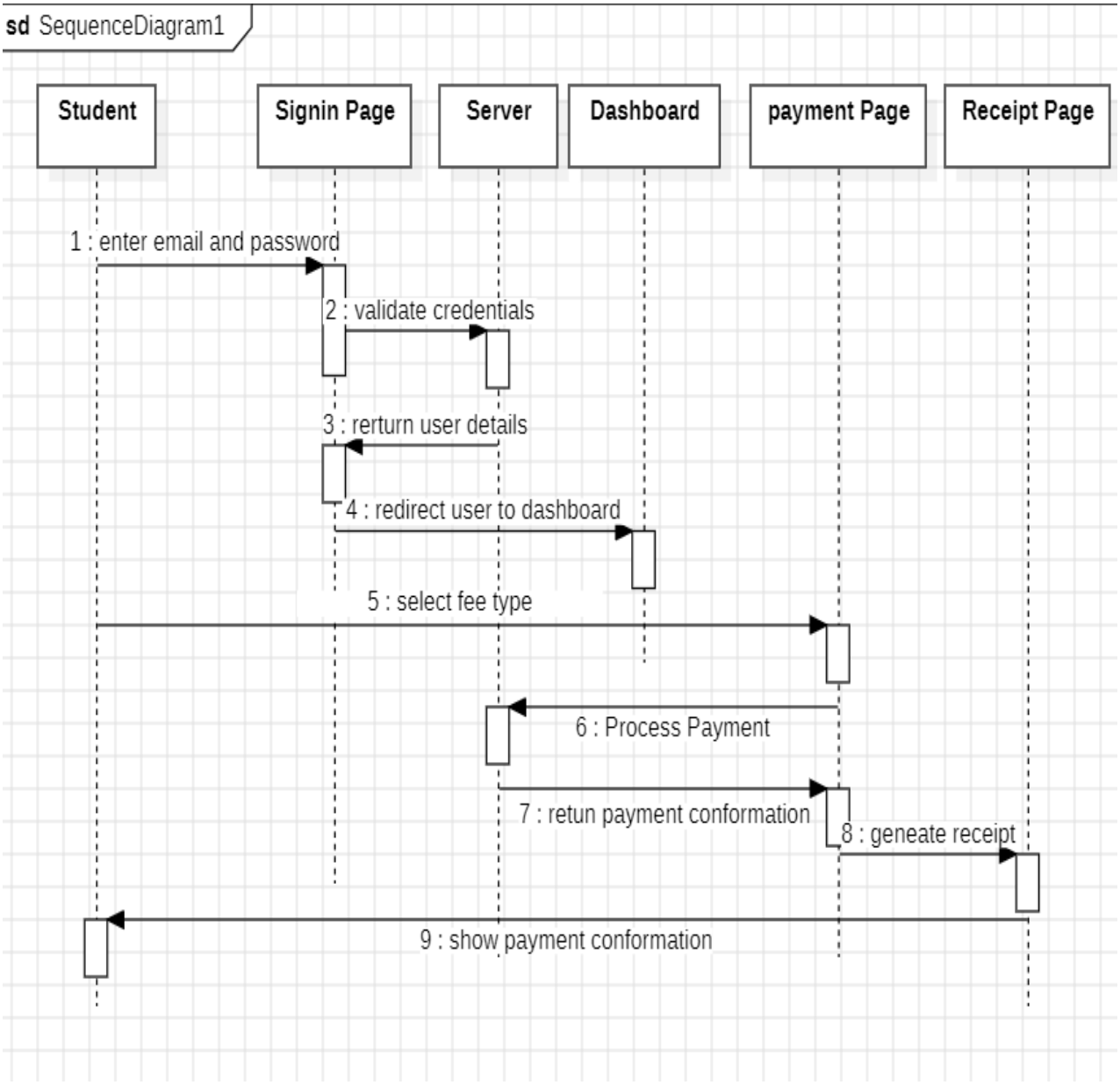
The **Class Diagram** defines the **static structure** of the **Online Fee Payment System**, representing key classes such as **User, Payment, and Dashboard** along with their attributes and methods. It establishes relationships between classes, ensuring a **well-structured system design** that supports secure fee transactions and receipt generation. This diagram is essential for **database and system architecture design**, as it helps in defining **data storage, object relationships, and method implementation** for seamless system operations.



Class Diagram for *Online Fee Payment System for students*

3.4.1.3 Sequence Diagram

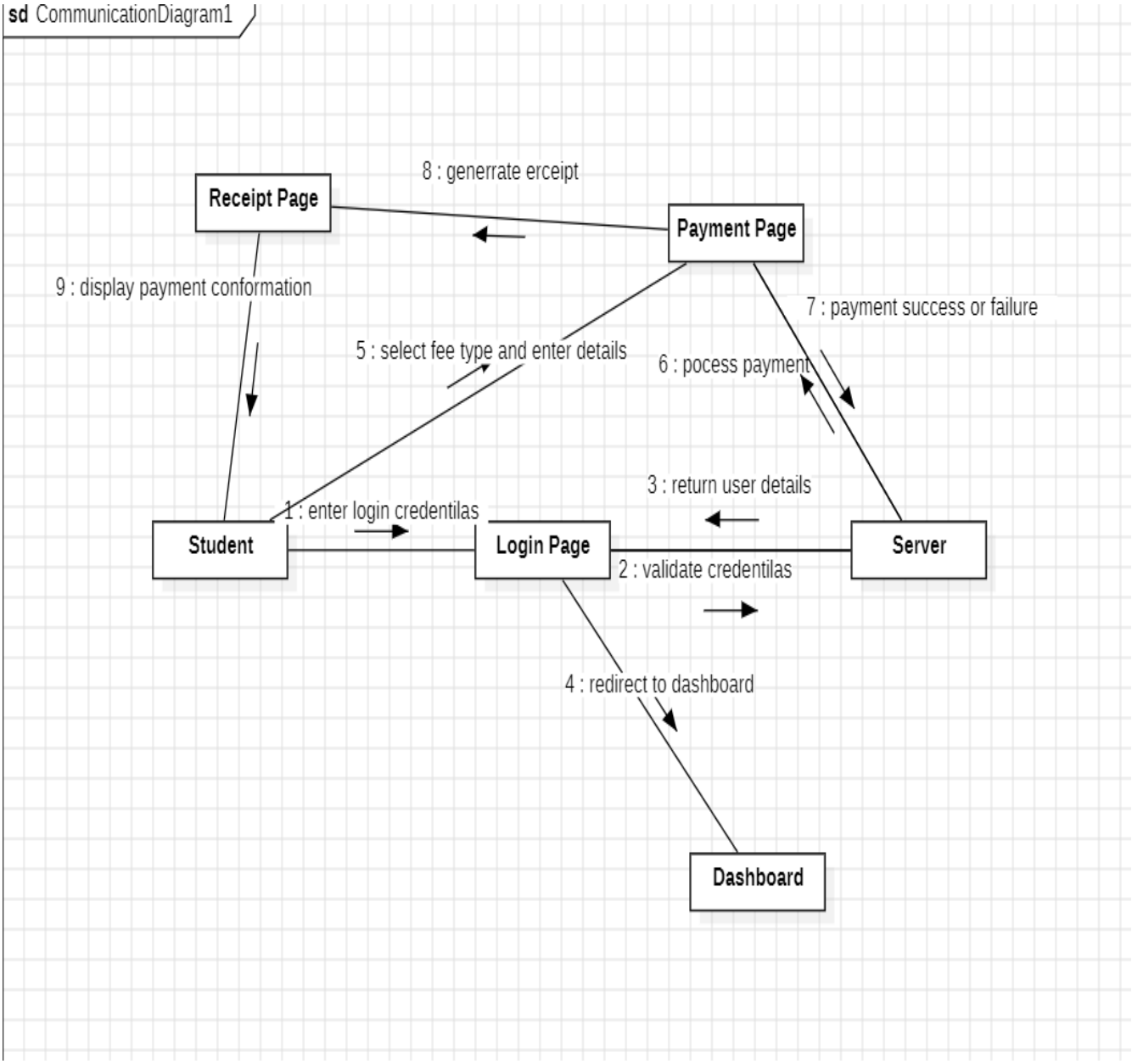
The **Sequence Diagram** provides a **time-based interaction flow** of how messages are exchanged between the **student and system components**. It depicts the **sequential process** of logging in, selecting a fee type, processing payment, and generating a receipt. This ensures **clear visualization of execution order and message flow**. The diagram helps in understanding **how requests and responses are processed**, ensuring a smooth transaction process and allowing developers to troubleshoot potential **delays or errors** in system communication.



Sequence Diagram for Virtual Drive Master

3.4.1.4 Collaboration Diagram

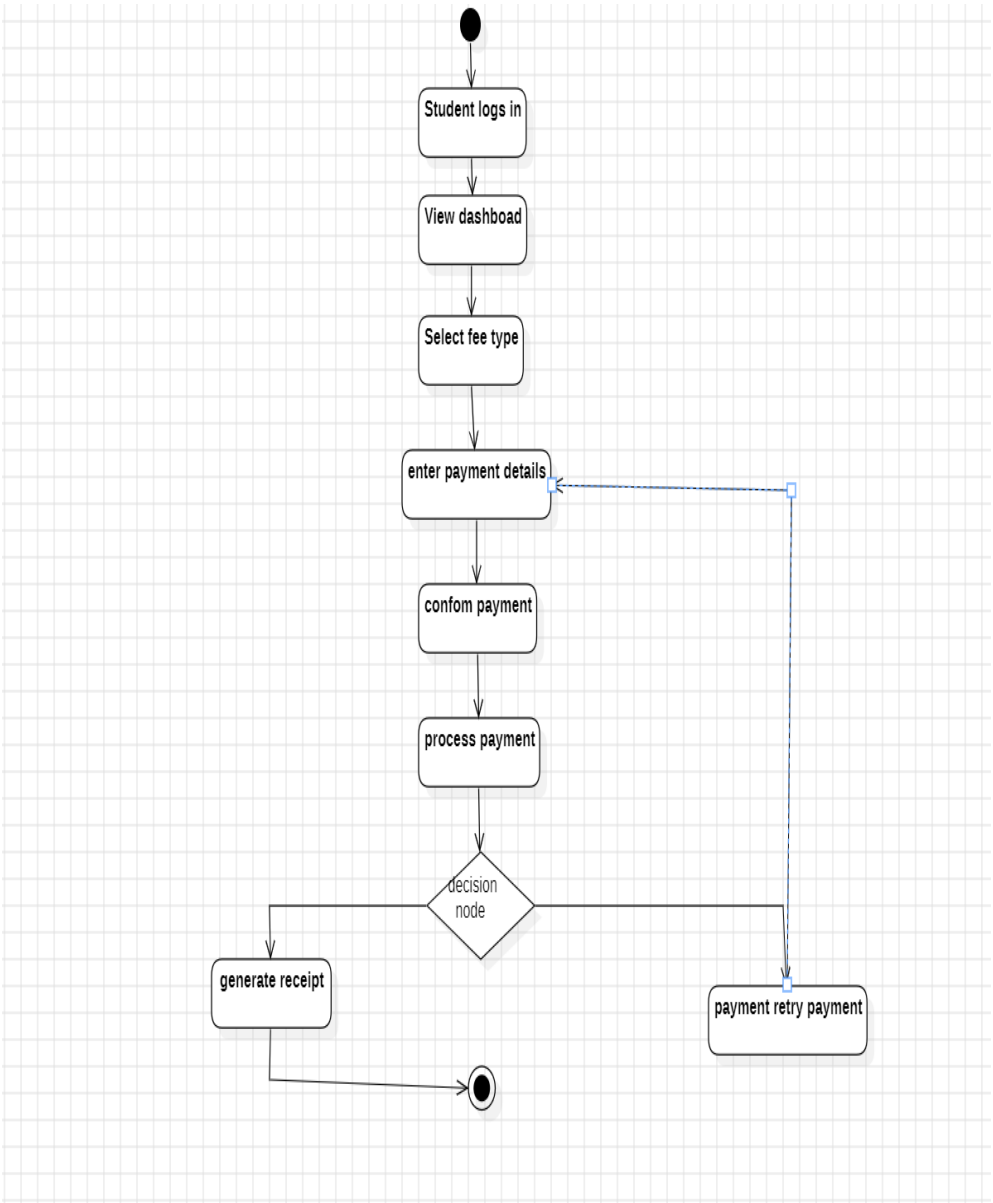
The **Collaboration Diagram** shows the **interaction between system components** during the fee payment process. It represents objects such as the **LoginPage, Dashboard, PaymentPage, Server, and ReceiptPage**, highlighting how they exchange messages. The diagram focuses on **how objects communicate to complete the transaction**, rather than the sequence of interactions. It helps developers **visualize dependencies between components**, making it easier to understand **data flow and interaction logic** within the system.



Collaboration Diagram Online Fee Payment System for students

3.4.1.5 Activity Diagram

The **Activity Diagram** illustrates the **step-by-step flow of actions** in the **fee payment process**. It begins with the student logging in, selecting a fee type, entering payment details, and processing the transaction. The system handles both **successful and failed payments**, ensuring a structured and efficient **workflow for fee transactions**. The diagram uses **decision nodes** to manage different conditions, such as a **payment failure, where the user can retry**, or a **successful transaction leading to receipt generation**.



Activity Diagram for Online Fee Payment System for students

4. SYSTEM DESIGN

4.1 Design principle

The design principles for the Online Fee Payment System focus on creating a secure, user-friendly, and scalable platform that allows students to conveniently pay their fees while enabling administrators to manage financial records efficiently. These principles ensure that the system operates smoothly and provides an optimal experience for all users involved.

- Usability: The system should provide a simple and intuitive interface for students to log in, select their fee type, enter payment details, and complete transactions easily.
- Scalability: The platform should be designed to handle an increasing number of students and transactions without performance degradation, ensuring smooth operation during peak payment periods.
- Modularity: The system follows a modular approach, separating functionalities such as authentication, fee selection, payment processing, and receipt generation, making it easier to maintain and upgrade.
- Data Integrity: Ensuring accurate and consistent data management is critical, particularly for tracking student details, payments, and transaction records to prevent errors and discrepancies.
- Security: The system must implement strong encryption and secure payment gateways to protect sensitive information such as student credentials, transaction details, and payment history.
- Flexibility: The system should be able to support various payment methods, including UPI, bank transfers, credit/debit cards, and digital wallets, making it adaptable to different institutions**.**
- Automation: Automated receipt generation, transaction history updates, and financial reports should be integrated to minimize manual work and improve efficiency.

These principles help ensure that the Online Fee Payment System for Students is secure, scalable, and user-friendly, improving both the administrative process and student experience when managing fee payments.

5. SYSTEM TESTING

System testing is a critical phase in the **software development lifecycle**, ensuring that the **Online Fee Payment System for Students** functions as expected. It involves testing the **integrated system** against **specified requirements** to identify and fix any defects before deployment.

5.1 Testing schemes

To ensure comprehensive validation of the **Online Fee Payment System**, several testing schemes are applied:

5.1.1 Unit Testing

Unit testing is used to test individual functions or code modules at the most micro-level. The developer conducts unit testing to check if each module of code is working properly.

Case: Testing the functionality of the registration form, ensuring it correctly validates input fields like email, password, and student ID before allowing account creation.

5.1.2 Integration Testing

Integration testing focuses on verifying the **interaction between different modules** to ensure they work **seamlessly together**. It primarily relies on **black-box testing** while also incorporating some **white-box testing techniques**.

Case: Testing the integration between the **login module** and the **dashboard**, ensuring that after authentication, students are directed to the **correct fee payment section**.

5.1.3 Functional Testing

Functional testing is a **black-box testing** method focused on the **functional requirements** of the application. Testers ensure that the software functions as specified.
Case: Verifying that students can **register, log in, select fee type, make payments, and generate receipts** without errors.

5.1.4 System Testing

System testing is a more comprehensive test that evaluates the entire system as a whole. This is also known as alpha testing. It ensures that the system functions correctly under different conditions.

Case: Testing the full student journey, from logging in, selecting a fee type, completing the payment process, and downloading the payment receipt.

5.1.5 Performance Testing

Performance testing evaluates the system's responsiveness, speed, and scalability under different load conditions. The objective is to identify any bottlenecks in processing.

5.1.6 White Box Testing

White box testing, also known as clear-box testing, focuses on the internal logic, flow, and structure of the code. The tester has full knowledge of the system's code.

5.1.7 Black Box Testing

Black box testing evaluates the system from an end-user perspective without requiring knowledge of the internal code structure.

5.2 Test Cases

Below are the test cases categorized into different modules to **ensure comprehensive testing** of the **Online Fee Payment System**.**User Management Module**

1. User Management Module

Registration Form

Test Case	**Input**	**Expected Outcome**
Validate all fields are required	Leave fields empty	Display error message: "All fields are required."
Validate password strength	Enter password (If weak)	Display error message: "Password must be strong."
Validate email format	Enter invalid email	Display error message: "Enter a valid email address."
Successful registration	Enter valid details	User registered successfully, redirect to login page.

Login Form –

Test Case	**Input**	**Expected Outcome**
Validate credentials are required	Leave email and password empty	Display error message: "Email and password are required."
Validate incorrect credentials	Enter incorrect email/password	Display error message: "Invalid email or password."
Successful login	Enter valid email and password	User logged in successfully, redirect to dashboard.

2. Fee Payment Module

Payment Selection

Test Case	**Input**	**Expected Outcome**
Validate fee selection	Do not select a fee type	Display error message: "Please select a fee type."

Validate fee amount display	Select Examination Fee	Display the correct fee amount for Examination Fee.
Validate payment method selection	Do not select a payment method	Display error message: "Please select a payment method."

Payment Processing

Test Case	**Input**	**Expected Outcome**
Validate required payment details	Leave payment fields empty	Display error message: "All payment details are required."
Validate invalid card details	Enter incorrect card number	Display error message: "Invalid card number."
Successful payment	Enter valid payment details	Transaction successful, redirect to receipt page.

3. Receipt Generation Module

Test Case	**Input**	**Expected Outcome**
Verify receipt generation	Complete a successful payment	Display the receipt with transaction details.
Validate receipt download	Click “Download Receipt” button	Receipt downloaded successfully in PDF format.
Ensure receipt contains correct details	Check transaction ID on receipt	Display the correct transaction ID, date, and amount.

4. System Navigation

Test Case	**Input**	**Expected Outcome**
Validate navigation bar links	Click on “Home”	Redirect to the Homepage.
Ensure dashboard access after login	Click on "Dashboard" after login	Display student details and fee options.
Ensure logout functionality works	Click "Logout" button	End session and redirect to login page.

6. IMPLEMENTATION

The **implementation phase** of the **Online Fee Payment System for Students** involves setting up the **system requirements, developing the user interface, testing, deployment, and ongoing maintenance**. This phase ensures the system is **functional, secure, and user-friendly** for both students and administrators.

• System Requirements Analysis

• Identify User Requirements:

- **Students:** Log in, view fee details, select fee type (Examination/Hostel), make payments, and download receipts.
- **Administrators:** Manage student records, verify payments, generate financial reports, and monitor transactions.
- **Payment Gateway:** Process student transactions via multiple payment methods (UPI, Credit/Debit Card, Bank Transfer).

• Define Key Functionalities:

- **User Authentication:** Secure login and password recovery for students.
- **Dashboard Management:** Display student details, pending fees, and payment status.
- **Fee Payment Processing:** Allow students to select fee type and complete transactions.
- **Receipt Generation:** Automatically generate and allow download of payment receipts.
- **Admin Panel:** Enable administrators to track transactions and generate reports.

• Frontend Development

Develop an **intuitive and user-friendly interface** using modern web technologies:

- **HTML, CSS, JavaScript** – For structuring and styling web pages.
- **Frameworks (React, Angular, or Vue.js)** – To enhance user experience and ensure smooth navigation.
- **Responsive Design (Bootstrap, Tailwind CSS)** – To make the system accessible on both mobile and desktop devices.

• Testing

• Unit Testing:

- Test **individual components**, such as the **registration form, login functionality, and fee selection**.
- Verify **payment validation** (e.g., correct amount, valid payment methods).

• Integration Testing:

- Ensure **smooth interaction** between the **dashboard, payment gateway, and receipt generation**.
- Test the **workflow from login → fee selection → payment processing → receipt download**.

• Deployment

- Deploy the system on **cloud platforms** such as **AWS, Firebase, or Vercel** for scalability and performance.
- Ensure **secure payment processing** with **SSL encryption** and **secure API calls**.
- Optimize **load balancing** for **high traffic periods**, such as semester fee payment deadlines.

• Maintenance and Updates

• Bug Fixes and Optimizations:

- Continuously **monitor system performance** and **fix issues** related to payments or user authentication.

• Feature Updates:

- Based on **student and admin feedback**, add features like **reminders for unpaid fees** and **multiple payment attempts**.
- Improve the **receipt management system** by integrating **email receipt functionality**.

• Payment System Updates:

- Integrate new **payment gateways** to offer **more transaction options**.
- Ensure compliance with **security standards** for **safe online transactions**.

7. CONCLUSION

The **Online Fee Payment System** successfully streamlines the fee payment process by providing a secure, efficient, and user-friendly platform for students and educational institutions. The system ensures seamless transactions through a structured dashboard, multiple payment methods, automated receipt generation, and a responsive interface accessible across devices. With its modular design, it enables smooth navigation, secure authentication, and efficient localStorage-based data management, reducing manual workload and paperwork.

Limitations

While the system meets its primary objectives, it has certain limitations:

- **No Payment History Tracking:** Students cannot access past transaction records.
- **Limited Payment Gateway Integration:** The system currently supports basic payment methods and lacks advanced options like recurring payments or mobile wallets.
- **No Due Date Notifications:** Students are not alerted about upcoming fee deadlines, leading to potential late payments.
- **Basic Admin Features:** Lacks financial reporting, analytics, and automated verification tools for administrators.

Future Enhancements

To improve system functionality and scalability, the following enhancements are proposed:

- **Payment History Tracking:** Enabling students to view past transactions for better financial management.
- **Enhanced Payment Gateways:** Supporting additional payment methods like EMIs, mobile wallets, and international payments.
- **Automated Due Date Notifications:** Implementing email/SMS reminders to notify students of upcoming fee deadlines.
- **Advanced Admin Dashboard:** Introducing financial reports, analytics, and automated verification for improved fee management.
- **Multi-Institution Support:** Expanding the system's adaptability for different colleges and universities.

By addressing these limitations and implementing future improvements, the Online Fee Payment System can evolve into a **comprehensive, automated, and data-driven financial management tool**, ensuring a seamless fee payment experience for students and administrators.

8. BIBILOGRAPHY

Referred Books –

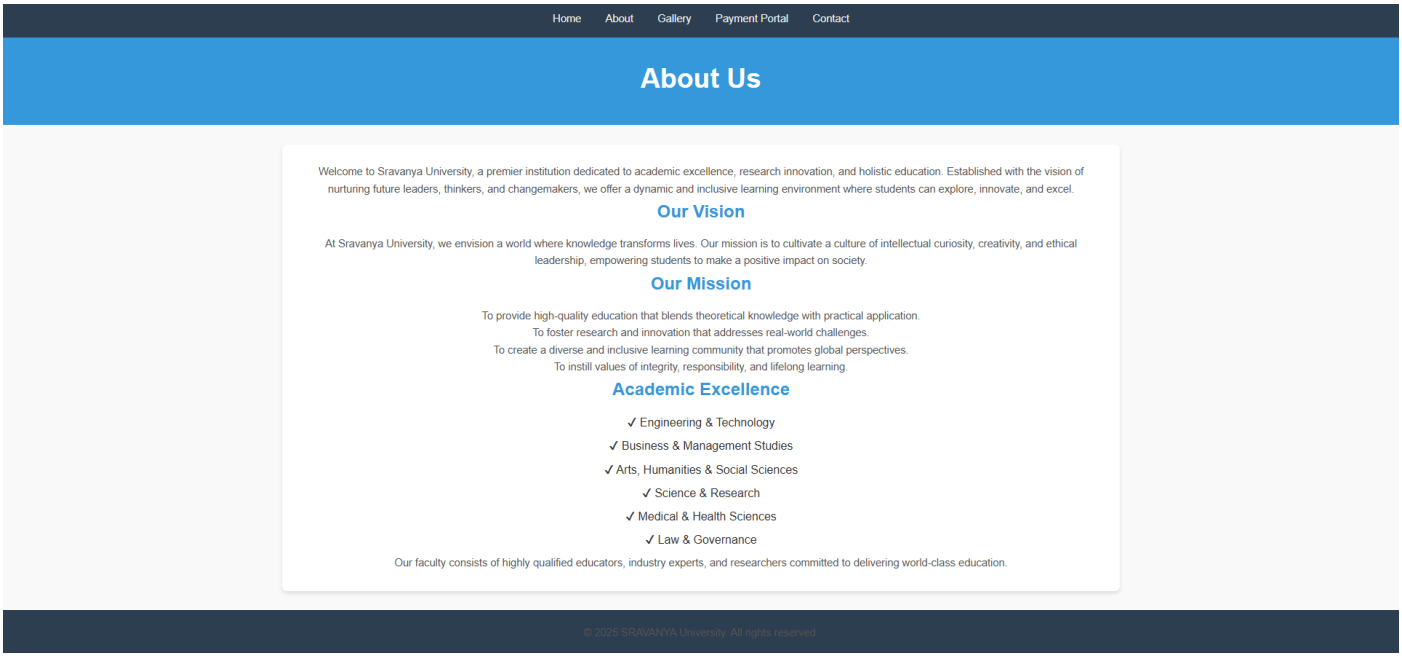
1. *"Software Engineering: A Practitioner's Approach"* – Roger S. Pressman, Bruce R. Maxim, McGraw-Hill Education, 8th Edition, 2014.
2. *"Systems Analysis and Design: An Object-Oriented Approach with UML"* – Alan Dennis, Barbara Wixom, David Tegarden, Wiley, 5th Edition, 2018.
3. *"An Introduction to Database Systems"* – C. J. Date, Pearson Education, 8th Edition, 2019.
4. *"Software Testing: Principles and Practices"* – Srinivasan Desikan, Gopalaswamy Ramesh, Pearson Education, 2006.
5. *"The Unified Modeling Language User Guide"* – Grady Booch, James Rumbaugh, Ivar Jacobson, Addison-Wesley, 2nd Edition, 2005.

Referred URLs –

- https://www.w3schools.com/js/
- https://developer.mozilla.org/en-US/
- https://www.geeksforgeeks.org/web-development/
- https://stripe.com/docs
- https://www.npci.org.in/upi

9.APPENDICES

APPENDIX A:Screens



Home About Gallery Payment Portal Contact

Contact Us

If you have any questions, please reach out to the appropriate department below:

University Management Contacts

Admissions Office
Address: 123 University Ave, Tirupati, AndhraPradesh, 517502
Email: admissions@university.edu
Phone: (123) 456-7890

Financial Aid Office
Address: 456 College Rd, Tirupati, AndhraPradesh, 517502
Email: financialaid@university.edu
Phone: (123) 456-7891

Registrar's Office
Address: 789 Academic St, Tirupati, AndhraPradesh, 517502
Email: registrar@university.edu
Phone: (123) 456-7892

Student Affairs
Address: 321 Campus Dr, Tirupati, AndhraPradesh, 517502
Email: studentaffairs@university.edu
Phone: (123) 456-7893

IT Support
Address: 654 Tech Blvd, Tirupati, AndhraPradesh, 517502
Email: itsupport@university.edu
Phone: (123) 456-7894

Career Services
Address: 987 Career Way, Tirupati, AndhraPradesh, 517502
Email: careerservices@university.edu
Phone: (123) 456-7895

Home About Gallery Payment Portal Contact

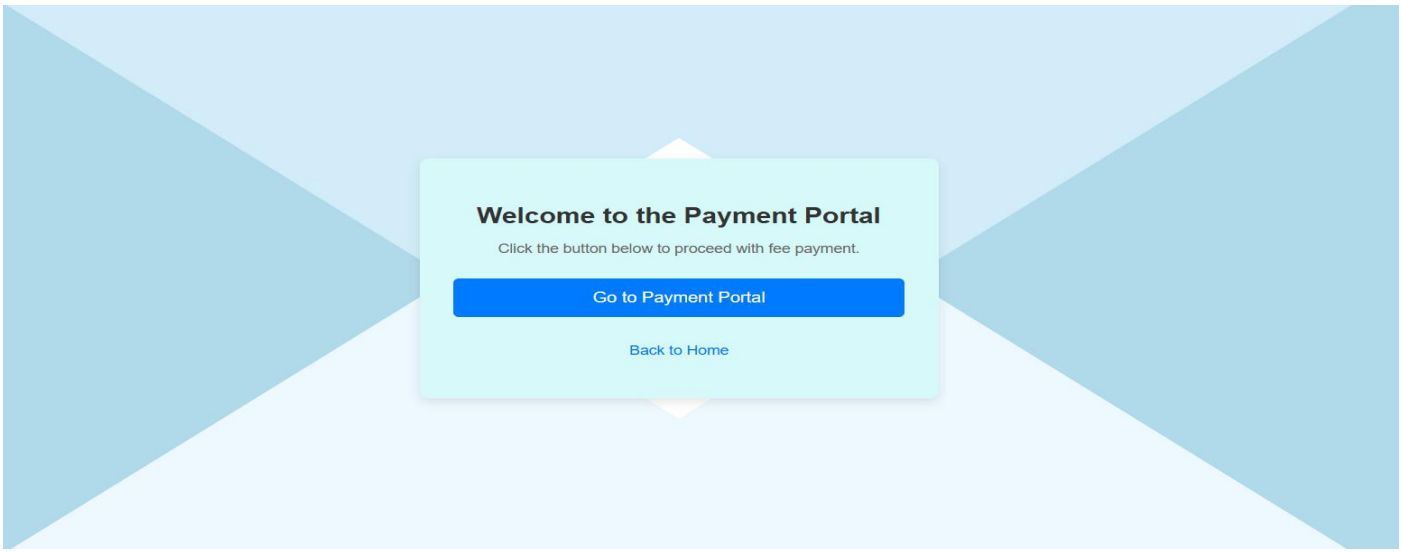
Gallery

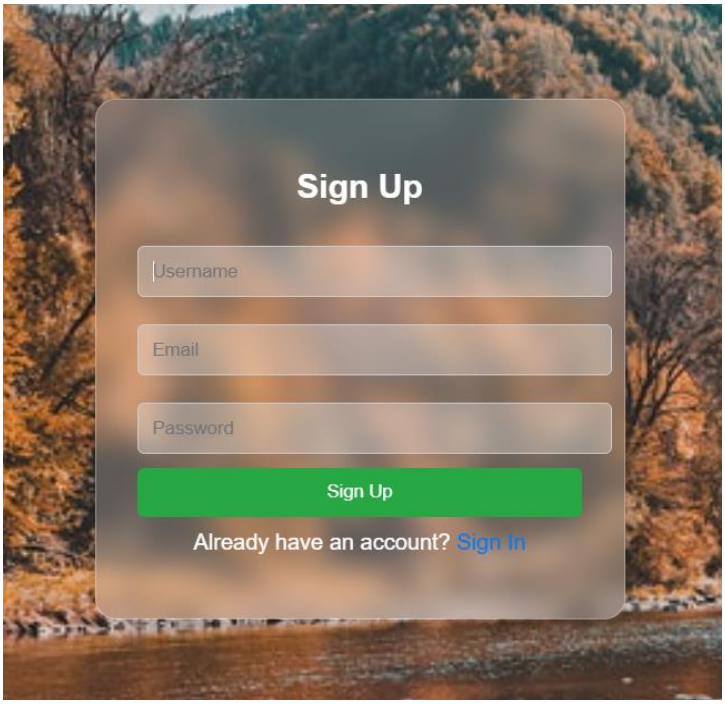
Check out our campus events and activities.

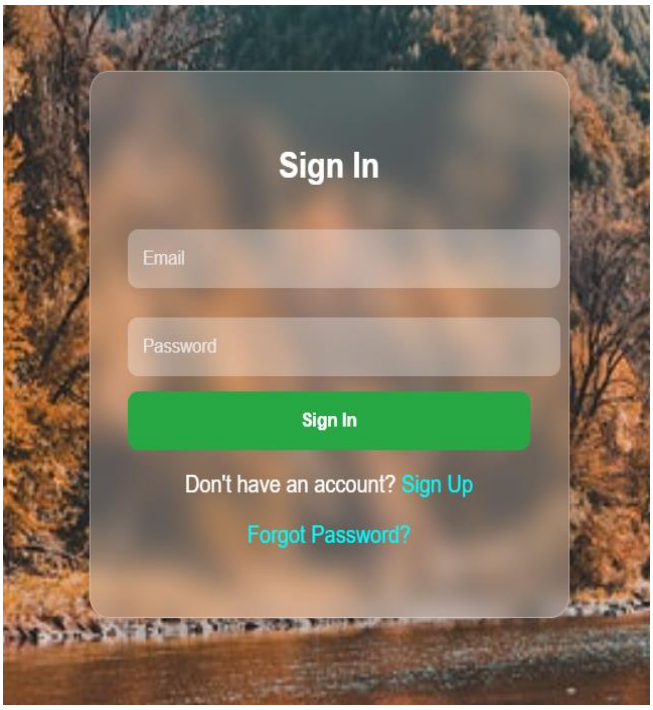


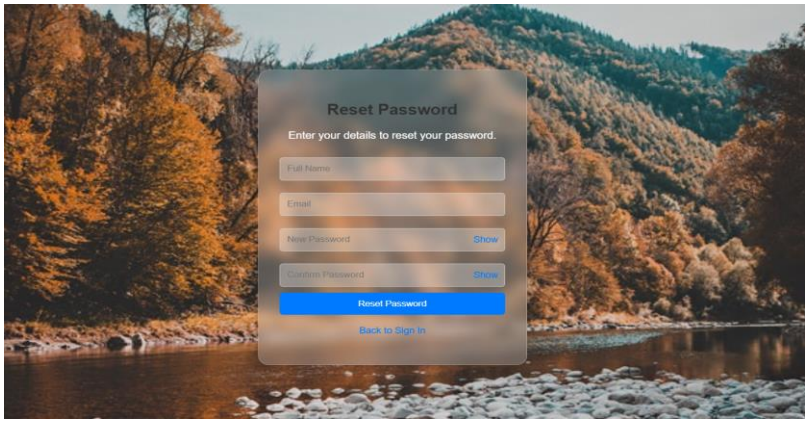


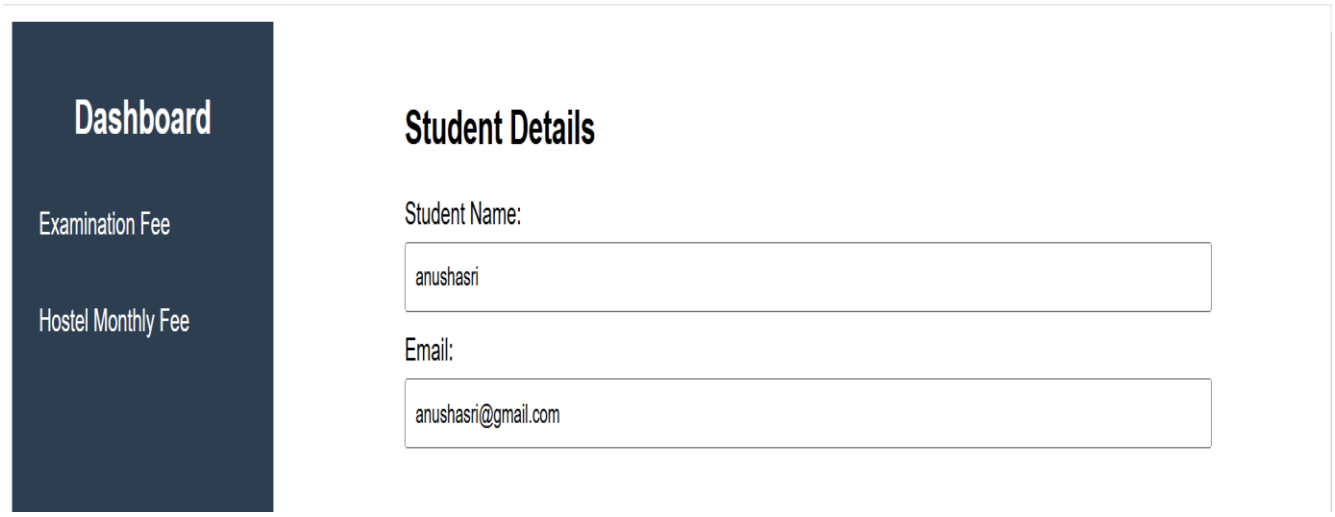


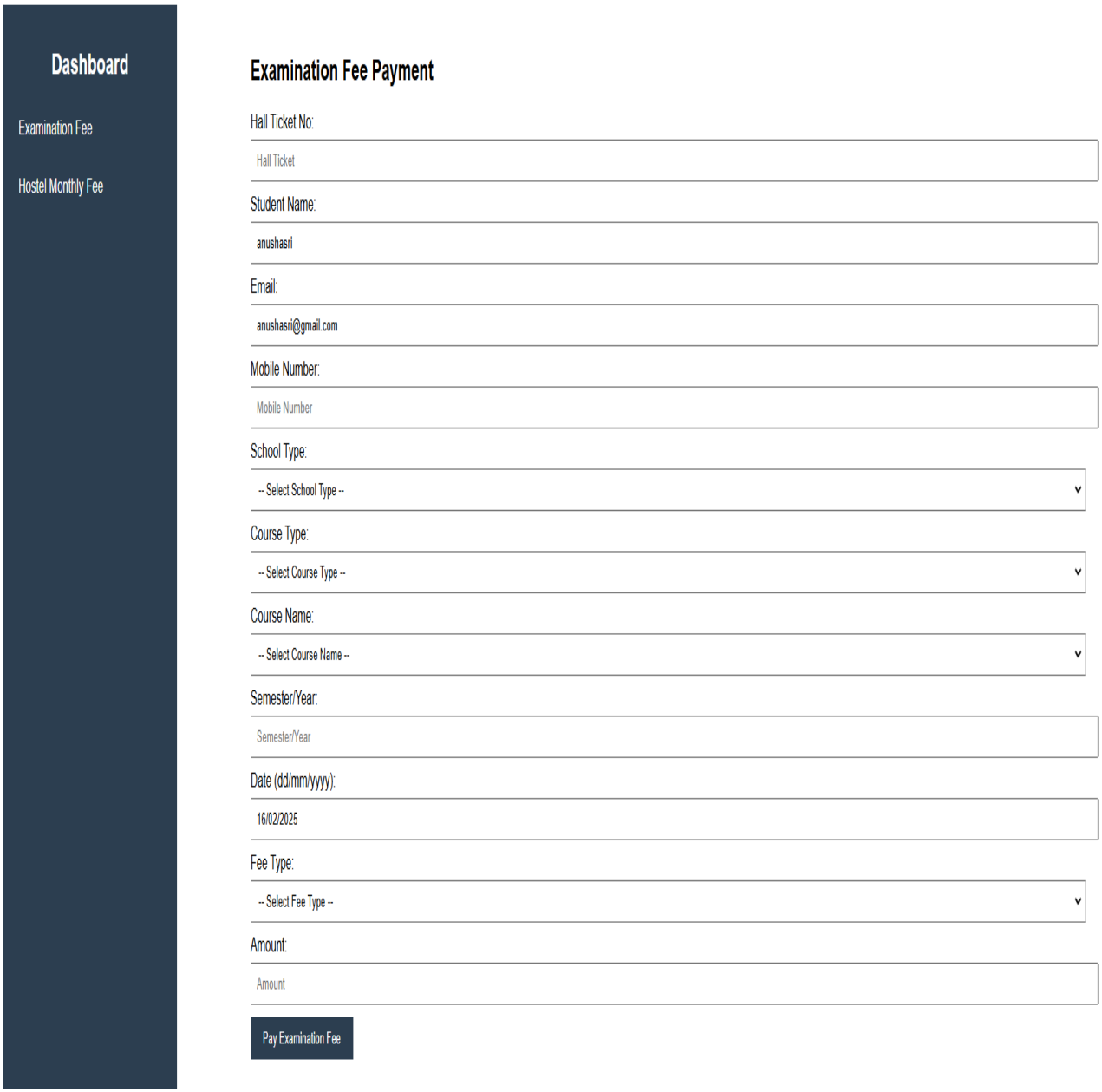


Welcome to the Payment Portal
Click the button below to proceed with fee payment.
Go to Payment Portal
Back to Home

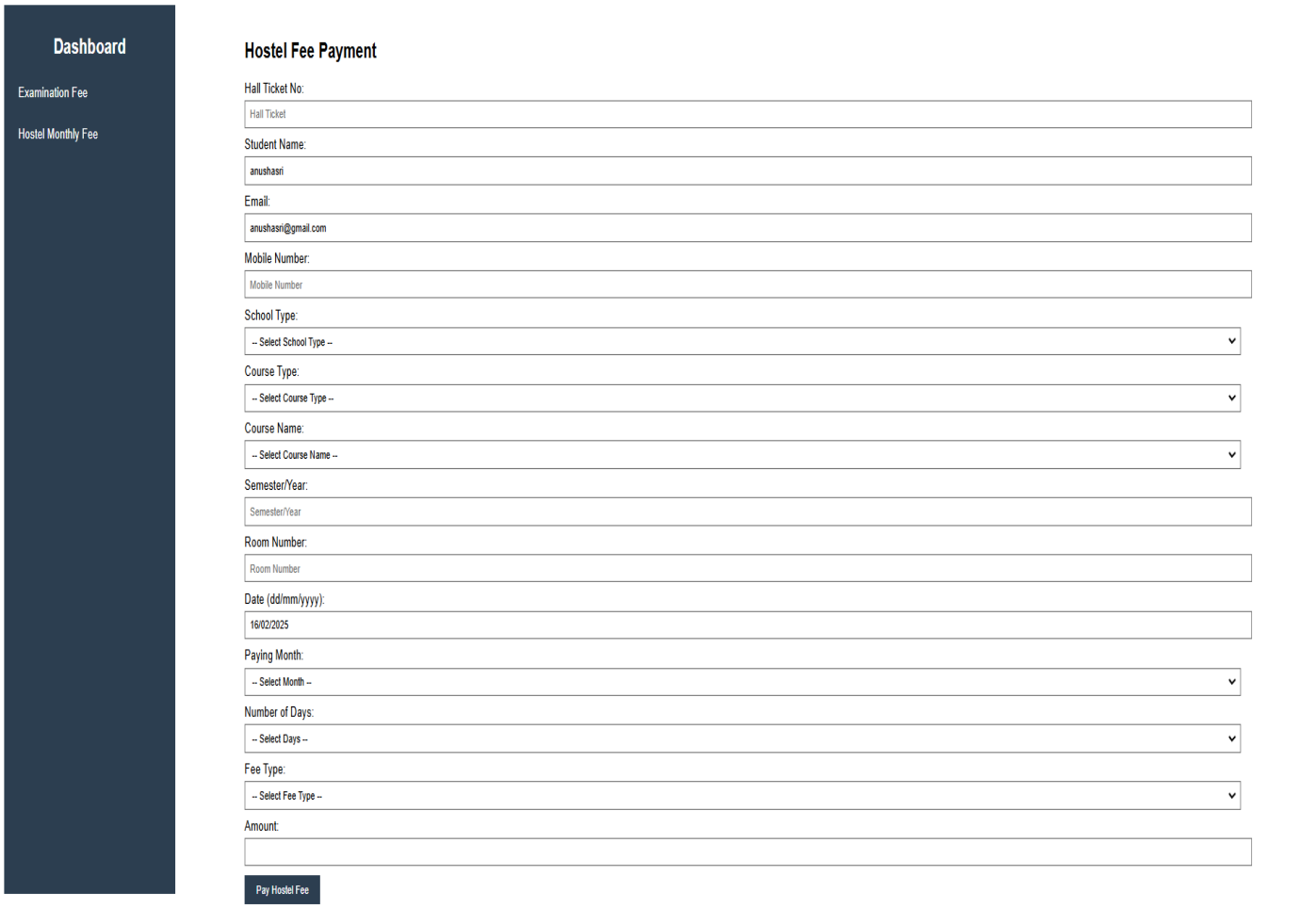

Sign Up
Username
Email
Password
Sign Up
Already have an account? Sign In

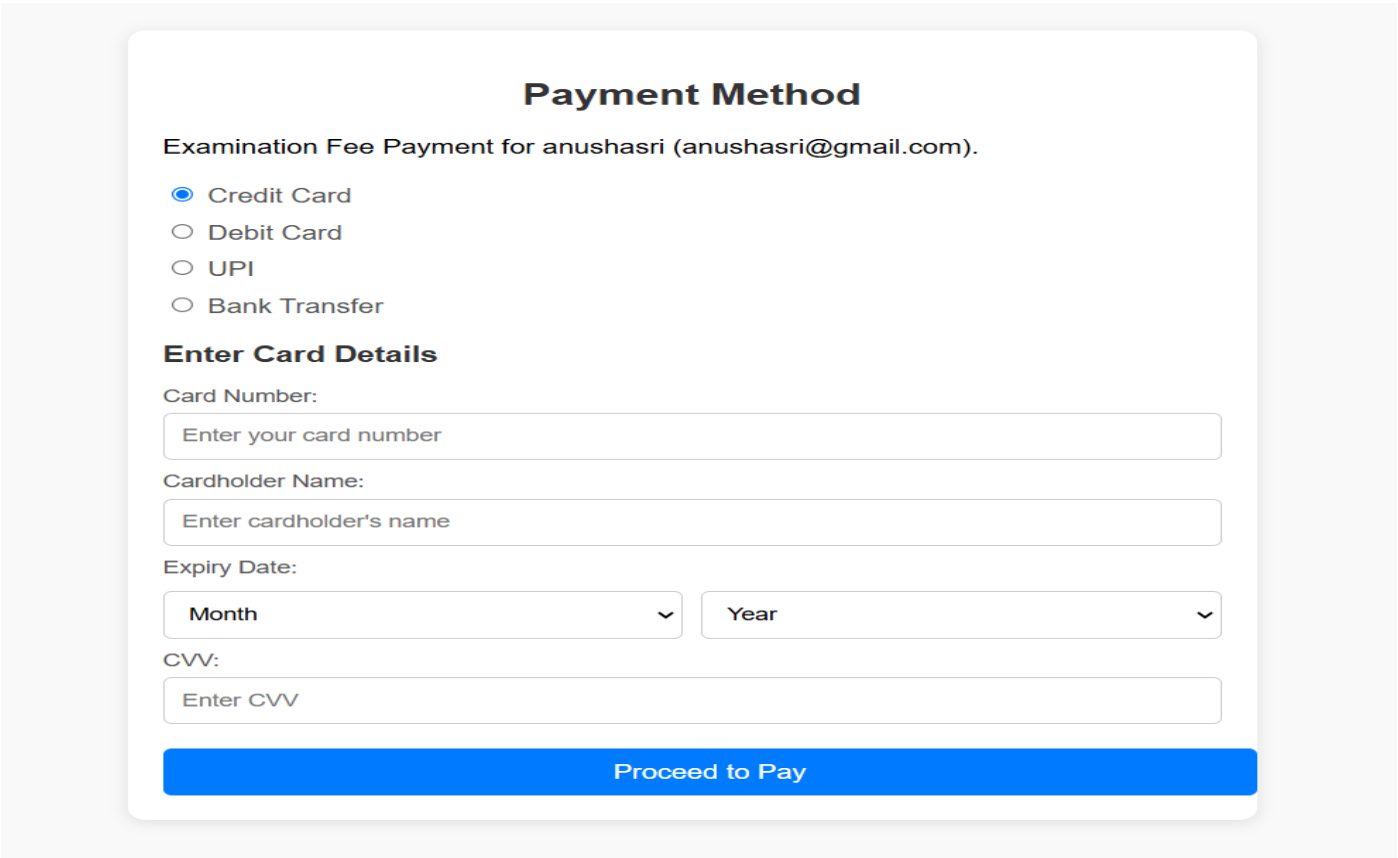

Sign In
Email
Password
Sign In
Don't have an account? Sign Up
Forgot Password?


Reset Password
Enter your details to reset your password.
Full Name
Email
New Password
Show
Confirm Password
Show
Reset Password
Back to Sign In


Dashboard
Examination Fee
Hostel Monthly Fee
Student Details
Student Name:
anushasri
Email:
anushasri@gmail.com


Dashboard
Examination Fee
Hostel Monthly Fee
Examination Fee Payment
Hall Ticket No:
Hall Ticket
Student Name:
anushasri
Email:
anushasri@gmail.com
Mobile Number:
Mobile Number
School Type:
-- Select School Type --
Course Type:
-- Select Course Type --
Course Name:
-- Select Course Name --
Semester/Year:
Semester/Year
Date (dd/mm/yyyy):
16/02/2025
Fee Type:
-- Select Fee Type --
Amount:
Amount
Pay Examination Fee


Dashboard
Examination Fee
Hostel Monthly Fee
Hostel Fee Payment
Hall Ticket No:
Hall Ticket
Student Name:
anushasri
Email:
anushasri@gmail.com
Mobile Number:
Mobile Number
School Type:
-- Select School Type --
Course Type:
-- Select Course Type --
Course Name:
-- Select Course Name --
Semester/Year:
Semester/Year
Room Number:
Room Number
Date (dd/mm/yyyy):
16/02/2025
Paying Month:
-- Select Month --
Number of Days:
-- Select Days --
Fee Type:
-- Select Fee Type --
Amount:
Pay Hostel Fee


Payment Method
Examination Fee Payment for anushasri (anushasri@gmail.com).
Credit Card
Debit Card
UPI
Bank Transfer
Enter Card Details
Card Number:
Enter your card number
Cardholder Name:
Enter cardholder's name
Expiry Date:
Month
Year
CVV:
Enter CVV
Proceed to Pay

Payment Receipt

SRAVANYA University

Student Name	anushasri
Email	anushasri@gmail.com
Mobile Number	1234567890
Hall Ticket No	16001
Course Type	B.Sc
Course Name	Mathematics
Semester	1
Exam Fee Type	semester_fee
Transaction ID	TXN27876147
Amount Paid	₹1000
Payment Date	2/16/2025
Status	Successful

Download Receipt as PDF

Payment Receipt

Examination Fee Receipt

Student Name	:	anushasri
Email	:	anushasri@gmail.com
Mobile Number	:	1234567890
Hall Ticket No	:	16001
Course Type	:	B.Sc
Course Name	:	Mathematics
Semester	:	1
Exam Fee Type	:	semester_fee
Transaction ID	:	TXN27876147
Amount Paid	:	¹1000
Payment Date	:	2/16/2025
Status	:	Successful

Thank you for your payment!

APPENDIX B: User Manual

1. Introduction

The **Online Fee Payment System** enables students to **securely and conveniently** pay their fees online. This manual provides a brief guide on **how to register, log in, make payments, and download receipts**.

2. System Requirements

- **Device Compatibility:** Desktop, Laptop, Tablet, Smartphone
- **Browser Support:** Chrome, Firefox, Edge, Safari
- **Internet Connection:** Required for transactions

3. Getting Started

- **Sign Up:** Enter your name, email, password, student ID, and phone number.
- **Login:** Enter your registered email and password to access the dashboard.
- **Forgot Password:** Use the **"Forgot Password"** option to reset your credentials.

4. Fee Payment Process

- **Select Fee Type** (Examination Fee or Hostel Fee).
- **Enter Payment Details** and choose a method (**UPI, Debit/Credit Card, Bank Transfer**).
- **Confirm and Complete Payment** by clicking **"Proceed to Pay"**.
- **Receive Confirmation** and download the receipt.

5. Receipt Generation

- Go to **"Payment Receipt"**, click **"Download"**, and save the receipt in **PDF format**.

6. System Navigation

Feature	• Function
Dashboard	• View student details and payment status
Pay Fees	• Select fee type and complete payment
Transaction Status	• Check payment confirmation
Payment Receipt	• Download transaction receipt
Logout	• Securely exit the system

7. Troubleshooting & FAQs

- **Failed Payment?** Check your internet, verify details, or contact your bank.
- **Forgot Password?** Use the **"Forgot Password"** option to reset it.
- **Payment Confirmation?** Check the **transaction status** or download the receipt.

8. Contact Support

For assistance, contact:

- **Email:** support@sravanyauniversityfeesystem.com
- **Phone:** +91- (123) 456-7891

This manual provides a **quick and easy guide** to using the **Online Fee Payment System for Students**.

APPENDIX C: Source Code

//Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Online Fee Payment System</title>
    <style>
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}body {
    font-family: 'Arial', sans-serif;
    line-height: 1.6;
    background-color: #f4f4f9;
    color: #333;
}nav {
    background-color: #2c3e50;
    padding: 1rem 0;
    position: sticky;
    top: 0;
    z-index: 10;
}nav ul {
    display: flex;
    justify-content: center;
    list-style-type: none;
}nav ul li {
    margin: 0 20px;
}
nav ul li a {
    color: #ecf0f1;
    text-decoration: none;
    font-size: 1.1rem;
    transition: color 0.3s ease;
}
```

```
nav ul li a:hover {
    color: #3498db;
}ul {
            list-style-type: none;
        }ul li {
            font-size: 1.2rem;
            margin-bottom: 1.5rem;
            line-height: 1.6;
            padding: 1rem;
            border-bottom: 1px solid #ecf0f1;
            transition: background-color 0.3s ease;
        }ul li:hover {
            background-color: #ecf0f1;
            border-radius: 50px;
        }ul li strong {
            color: #3498db;
            font-size: 1.3rem;
        }a {
            color: #3498db;
            text-decoration: none;
            transition: color 0.3s ease;
        }header {
    text-align: center;
    padding: 3rem 0;
    background-color: #3498db;
    color: white;}
}header p {
    font-size: 1.2rem;
    font-weight: 300;
}.carousel {
    position: relative;
    width: 90%;
    max-width: 1500px;
    height: 800px;
    margin: 20px auto;
    overflow: hidden;
    border: 2px solid #ccc;
```

```
    border-radius: 10px;
}.carousel-images {
    display: flex;
    transition: transform 0.5s ease;
    height: 100%;
}carousel-images img {
    width: 100%;
    height: 100%;
    object-fit:cover;
    min-width: 100%;
}
.carousel-button {
    position: absolute;
    top: 50%;
    transform: translateY(-50%);
    background-color: rgba(255, 255, 255, 0.7);
    border: none;
    padding: 10px;
    cursor: pointer;
    border-radius: 5px;
}
.prev {
    left: 10px;
}
.next {
    right: 10px;
}footer {
    background-color: #2c3e50;
    text-align: center;
    padding: 1rem;
    color: white;
    margin-top: 3rem;
}/* Responsive Design */
@media (max-width: 768px) {
    header h1 {
        font-size: 2.5rem;
    }
```

```
    header p {
        font-size: 1.1rem;
    }
    .carousel-button {
        font-size: 1.5rem;
        padding: 8px;
    }
}</style></head>
<body><nav>
        <ul><li><a href="index.html">Home</a></li>
            <li><a href="about.html">About</a></li>
            <li><a href="gallery.html">Gallery</a></li>
            <li><a href="payment.html">Payment Portal</a></li>
            <li><a href="contact.html">Contact</a></li> </ul></nav>
    <header><h1>SRAVANYA UNIVERSITY</h1>
        <p>Learn Today, Lead Tomorrow.</p></header>
    <div class="carousel">
        <div class="carousel-images">
            <img src="./images/carousel1.jpg" alt="Slide 1">
            <img src="./images/carousel2.jpg" alt="Slide 2">
            <img src="./images/carousel3.jpeg" alt="Slide 3">
            <img src="./images/carousel4.jpg" alt="Slide 4">
            <img src="./images/carousel5.jpg" alt="Slide 5">
        </div><button class="carousel-button prev" onclick="moveSlide(-1)">&#10094;</button>
        <button class="carousel-button next" onclick="moveSlide(1)">&#10095;</button></div>
    <footer><p>&copy; 2025 SRAVANYA University. All rights reserved.</p></footer>
    <script>
        let currentSlide = 0;
        function showSlide(index) {
            const slides = document.querySelectorAll('.carousel-images img');
            if (index >= slides.length) {
                currentSlide = 0;
            } else if (index < 0) {
                currentSlide = slides.length - 1;
            } else {
                currentSlide = index;
            }const offset = -currentSlide * 100; // Move the carousel
```

```
        document.querySelector('.carousel-images').style.transform = `translateX(${offset}%)`;
    }
    function moveSlide(direction) {
        showSlide(currentSlide + direction);
    }
    showSlide(currentSlide);
  </script>
</body>
</html>
```

//about.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>About Us</title>
  <style>
    * {margin: 0;padding: 0;
      box-sizing: border-box;
    }
    /* Body and General Styles */
    body {
      font-family: 'Arial', sans-serif;
      line-height: 1.6;
      background-color: #f9f9f9;
      color: #333; }
    nav {
      background-color: #2c3e50;
      padding: 1rem 0;
      position: sticky;
      top: 0;
      z-index: 10;}
    nav ul {
      display: flex;
      justify-content: center;
      list-style-type: none;}
    nav ul li { margin: 0 20px;}
```

```
nav ul li a {
        color: #ecf0f1;
        text-decoration: none;
        font-size: 1.1rem;
        transition: color 0.3s ease; }
    nav ul li a:hover {color: #3498db;}
    header {
        text-align: center;
        padding: 2rem 0;
        background-color: #3498db;
        color: white;}
    header h1 {
        font-size: 2.8rem;
        margin-bottom: 0.5rem;}
    .container {
        width: 60%;
        margin: 2rem auto;
        background-color: white;
        padding: 2rem;
        border-radius: 8px;
        box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);}
    h3 {
        font-size: 1.8rem;
        color: #3498db;
        margin-bottom: 1rem;
        text-align: center;  }
    p {
        font-size: 1.1rem;
        color: #555;
        line-height: 1.6;
        text-align: center;}
    /* Responsive Design */
    @media (max-width: 768px) {
        .container {width: 90%;padding: 1.5rem;}
        header h1 {
            font-size: 2.2rem;
        }}</style>
```

```
</head>
<body>
  <nav> <ul><li><a href="index.html">Home</a></li>
          <li><a href="about.html">About</a></li>
           <li><a href="gallery.html">Gallery</a></li>
           <li><a href="payment.html">Payment Portal</a></li>
       <li><a href="contact.html">Contact</a></li>
     </ul> </nav>
  <header><h1>About Us</h1></header>
  <div class="container">
       <p>Welcome to Sravanya University, a premier institution dedicated to academic excellence, research innovation, and holistic education. Established with the vision of nurturing future leaders, thinkers, and changemakers, we offer a dynamic and inclusive learning environment where students can explore, innovate, and excel.</p>
     <h3>Our Vision</h3>
     <p>At Sravanya University, we envision a world where knowledge transforms lives. Our mission is to cultivate a culture of intellectual curiosity, creativity, and ethical leadership, empowering students to make a positive impact on society.</p>
     <h3>Our Mission</h3>
     <p> To provide high-quality education that blends theoretical knowledge with practical application.<br>
       To foster research and innovation that addresses real-world challenges.<br>
       To create a diverse and inclusive learning community that promotes global perspectives.<br>
       To instill values of integrity, responsibility, and lifelong learning.
     </p><h3>Academic Excellence</h3>
<ul>   <li>✓ Engineering & Technology</li>
       <li>✓ Business & Management Studies</li>
       <li>✓ Arts, Humanities & Social Sciences</li>
       <li>✓ Science & Research</li>
       <li>✓ Medical & Health Sciences</li>
       <li>✓ Law & Governance</li></ul>
     <p>Our faculty consists of highly qualified educators, industry experts, and researchers committed to delivering world-class education.</p></div>
  <footer>
     <p>&copy; 2025 SRAVANYA University. All rights reserved.</p>
  </footer>
</body>
</html>
```

//contact.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Contact Us</title>
    <style>
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;}
        /* Body and General Styles */
        body {
            font-family: 'Arial', sans-serif;
            line-height: 1.6;
            background-color: #f9f9f9;
            color: #333;
        }
        /* Header Section */
        header {
            text-align: center;
            padding: 3rem 0;
            background-color: #3498db;
            color: white;
        }
        header h1 {
            font-size: 3.5rem;
            margin-bottom: 0.5rem;
            letter-spacing: 2px;
        }
        header p {font-size: 1.2rem;font-weight: 300;}
        /* Contact Section */
        section {
            max-width: 1000px;
            margin: 20px auto;
            background-color: white;
            padding: 20px;
```

```
    border-radius: 8px;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}section h2 {
    font-size: 2.5rem;
    color: #3498db;
    text-align: center;
    margin-bottom: 1.5rem;
}ul {list-style-type: none; padding: 0; }
ul li {
    font-size: 1.2rem;
    margin-bottom: 1.5rem;
    line-height: 1.6;
    padding: 1rem;
    border-bottom: 1px solid #ecf0f1;
    transition: background-color 0.3s ease;
}ul li:hover {background-color: #ecf0f1;border-radius: 50px;}
ul li strong {
    color: #3498db;
    font-size: 1.3rem;
}a {
    color: #3498db;
    text-decoration: none;
    transition: color 0.3s ease;
}a:hover {color: #2c3e50;  }
footer {
    background-color: #2c3e50;
    text-align: center;
    padding: 1.5rem;
    color: white;
    margin-top: 2rem;
}/* Responsive Design */
@media (max-width: 768px) {
    header h1 {
        font-size: 2.5rem;
    }header p {font-size: 1.1rem;  }
    section {width: 95%;
        padding: 1.5rem;
```

```
        }section h2 {font-size: 2rem;}
        ul li {font-size: 1rem;}}
    </style>
</head>
<body>
    <nav>
        <ul>
            <li><a href="index.html">Home</a></li>
            <li><a href="about.html">About</a></li>
            <li><a href="gallery.html">Gallery</a></li>
            <li><a href="payment.html">Payment Portal</a></li>
            <li><a href="contact.html">Contact</a></li>
        </ul>
    </nav>
    <header ><h1>Contact Us</h1>
        <p>If you have any questions, please reach out to the appropriate department below:</p></header>
    <section>
        <h2>University Management Contacts</h2>
        <ul>
            <li><strong>Admissions Office</strong><br>
                Address: 123 University Ave, Tirupati, AndhraPradesh, 517502<br>
                Email: <a href="mailto:admissions@university.edu">admissions@university.edu</a><br>
                Phone: (123) 456-7890</li>
            <li><strong>Financial Aid Office</strong><br>
                Address: 456 College Rd, Tirupati, AndhraPradesh, 517502<br>
                Email: <a href="mailto:financialaid@university.edu">financialaid@university.edu</a><br>
                Phone: (123) 456-7891 </li>
            <li><strong>Registrar's Office</strong><br>
                Address: 789 Academic St, Tirupati, AndhraPradesh, 517502<br>
                Email: <a href="mailto:registrar@university.edu">registrar@university.edu</a><br>
                Phone: (123) 456-7892</li>
            <li><strong>Student Affairs</strong><br>
                Address: 321 Campus Dr, Tirupati, AndhraPradesh, 517502<br>
                Email: <a href="mailto:studentaffairs@university.edu">studentaffairs@university.edu</a><br>
                Phone: (123) 456-7893</li>
            <li> <strong>IT Support</strong><br>
                Address: 654 Tech Blvd, Tirupati, AndhraPradesh, 517502<br>
```

```
            Email: <a href="mailto:itsupport@university.edu">itsupport@university.edu</a><br>
            Phone: (123) 456-7894 </li>
          <li><strong>Career Services</strong><br>
            Address: 987 Career Way, Tirupati, AndhraPradesh, 517502<br>
            Email: <a href="mailto:careerservices@university.edu">careerservices@university.edu</a><br>
            Phone: (123) 456-7895</li>
      </ul>
   </section>
   <footer><p>&copy; 2025 SRAVANYA University. All rights reserved.</p></footer>
</body>
</html>
```

//payment.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Payment Portal</title>
   <style>
   body {
      font-family: Arial, sans-serif;
     background: url('https://images.blocksurvey.io/cdn-cgi/imagedelivery/G6cPOuGZ4Z_bEV13gBxixw/template-category/event-experience.svg/l') no-repeat center center fixed; /* Add your image path here */
      background-size: cover;
      background-color: #f4f4f9;
      margin: 0;
      padding: 0;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
   }
   .container {
      background: rgb(214, 249, 249);
      padding: 30px;
      border-radius: 8px;
      box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.1);
```

```
        text-align: center;
        width: 400px;
    }    h2 { color: #333;  margin-bottom: 10px; }
    p {color: #666;font-size: 14px; }
  </style>
</head>
<body>
    <div class="container">
        <h2>Welcome to the Payment Portal</h2>
        <p>Click the button below to proceed with fee payment.</p>
        <button id="paymentBtn">Go to Payment Portal</button>
        <p><a href="index.html">Back to Home</a></p></div>
    <script>
        document.getElementById("paymentBtn").addEventListener("click", function() {
            window.location.href = "signin.html"; // Redirects to the sign-in page
        });
    </script>
</body></html>
```

//signin.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sign In</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background: url('https://mdbootstrap.com/img/Photos/Horizontal/Nature/6-col/img%20(122).jpg') no-repeat
center center fixed;
            background-size: cover;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;}
        .container {
            background: rgba(255, 255, 255, 0.2);
```

```
            backdrop-filter: blur(10px);
            -webkit-backdrop-filter: blur(10px);
            padding: 30px;
            border-radius: 15px;
            box-shadow: 0 8px 32px rgba(0, 0, 0, 0.2);
            text-align: center;
            width: 320px;
            color: white;
            border: 1px solid rgba(255, 255, 255, 0.3);
        }h2 {margin-bottom: 20px;font-weight: bold;}
        input {width: 100%;
            padding: 12px;
            margin: 10px 0;
            border: none;
            border-radius: 8px;
            background: rgba(255, 255, 255, 0.3);
            color: white;
            outline: none;
        }input::placeholder { color: rgba(255, 255, 255, 0.7);}
        button {
            width: 100%;
            padding: 12px;
            background: #28a745;
            border: none;
            color: white;
            border-radius: 8px;
            cursor: pointer;
            font-weight: bold;
            transition: background 0.3s ease;}
        button:hover {background: #218838;}
        p { margin-top: 15px;}
        a {color: #00ffff;text-decoration: none; }
        a:hover { text-decoration: underline; }
    </style>
</head>
<body>
    <div class="container">
```

```
    <h2>Sign In</h2>
    <form onsubmit="signin(event)">
      <input type="email" id="email" placeholder="Email" required>
      <input type="password" id="password" placeholder="Password" required>
      <button type="submit">Sign In</button>
      <p>Don't have an account? <a href="signup.html">Sign Up</a></p>
      <p><a href="forgot_password.html">Forgot Password?</a></p>
    </form></div>
  <script>
    function signin(event) {
      event.preventDefault();
      let email = document.getElementById("email").value;
      let password = document.getElementById("password").value;
      let users = JSON.parse(localStorage.getItem("users")) || [];
      let user = users.find(u => u.email === email && u.password === password);
      if (user) {
        localStorage.setItem("currentUser", JSON.stringify(user));
        alert("Sign-in successful!");
        window.location.href = "dashboard.html";
      } else {
        alert("Invalid email or password. Please try again.");
      } }
  </script>
</body>
</html>
```

//signup.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Sign Up</title>
  <style>
    body {
      font-family: Arial, sans-serif;
       background: url('https://mdbootstrap.com/img/Photos/Horizontal/Nature/6-col/img%20(122).jpg') no-repeat
center center fixed;
```

```
    background-size: cover;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    margin: 0;}
.container {
    background: rgba(255, 255, 255, 0.2);
    backdrop-filter: blur(10px);
    -webkit-backdrop-filter: blur(10px);
    padding: 30px;
    border-radius: 15px;
    box-shadow: 0 8px 32px rgba(0, 0, 0, 0.2);
    text-align: center;
    width: 320px;
    color: white;
    border: 1px solid rgba(255, 255, 255, 0.3);
}h2 {margin-bottom: 20px;}
input {
    width: 100%;
    padding: 10px;
    margin: 10px 0;
    border: 1px solid #ccc;
    border-radius: 5px;
    background: rgba(255, 255, 255, 0.3);
    color: white;
    outline: none;
}button {
    width: 100%;
    padding: 10px;
    background: #28a745;
    border: none;
    color: white;
    border-radius: 5px;
    cursor: pointer;
}button:hover {background: #218838;}
p {margin-top: 10px;}
```

```
        a {
            color: #007bff;
            text-decoration: none;
        }a:hover {text-decoration: underline;  }
    </style>
</head>
<body>
    <div class="container">
        <h2>Sign Up</h2>
        <form onsubmit="signup(event)">
            <input type="text" id="username" placeholder="Username" required>
            <input type="email" id="email" placeholder="Email" required>
            <input type="password" id="password" placeholder="Password" required>
            <button type="submit">Sign Up</button>
            <p>Already have an account? <a href="signin.html">Sign In</a></p>
        </form></div>
    <script>
        function signup(event) {
            event.preventDefault(); // Prevent form submission
            let username = document.getElementById("username").value;
            let email = document.getElementById("email").value;
            let password = document.getElementById("password").value;
            // Get existing users from localStorage
            let users = JSON.parse(localStorage.getItem("users")) || [];
            // Check if email is already registered
            let existingUser = users.find(user => user.email === email);
            if (existingUser) {
                alert("An account with this email already exists!");
                return;
            }// Save new user data to localStorage
            const newUser = { username, email, password };
            users.push(newUser);
            localStorage.setItem("users", JSON.stringify(users));
            localStorage.setItem("currentUser", JSON.stringify(newUser));
            alert("Registration successful!");
            window.location.href = "signin.html";
        }
```

```
   </script></body>
</html
```

//forgot_password.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8"><meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Forgot Password</title>
   <style>
   body {font-family: Arial, sans-serif;
          background: url('https://mdbootstrap.com/img/Photos/Horizontal/Nature/6-col/img%20(122).jpg') no-repeat center center fixed;
          background-size: cover;
          display: flex;
          justify-content: center;
          align-items: center;
          height: 100vh;
          margin: 0;
       }.container {
          background: rgba(255, 255, 255, 0.2);
          backdrop-filter: blur(10px);
          -webkit-backdrop-filter: blur(10px);
          padding: 30px;
          border-radius: 15px;
          box-shadow: 0 8px 32px rgba(0, 0, 0, 0.2);
          text-align: center;
          width: 320px;
          color: white;
          border: 1px solid rgba(255, 255, 255, 0.3);
       }h2 { color: #333;}
       /* Form Input */
       .input-container {
          position: relative;
          width: 100%;
       }input {
          width: 100%;
          padding: 10px;
```

```
        margin: 10px 0;
        border: 1px solid #ccc;
        border-radius: 5px;
        box-sizing: border-box;
        background: rgba(255, 255, 255, 0.3);
        color: white;
        outline: none;
    }
    /* Show Password Toggle */
    .toggle-password {
        position: absolute;
        right: 10px;
        top: 50%;
        transform: translateY(-50%);
        cursor: pointer;
        font-size: 14px;
        color: #007bff;
    }
    button {
        width: 100%;
        padding: 10px;
        background: #007bff;
        border: none;
        color: white;
        border-radius: 5px;
        cursor: pointer;
        transition: background 0.3s;
    }
    button:hover {background: #0056b3; }
    p a {
        color: #007bff;
        text-decoration: none;
        font-size: 14px;}
    p a:hover {
        text-decoration: underline;
    }
</style>
```

```
</head>
<body>
  <div class="container">
    <h2>Reset Password</h2>
    <p>Enter your details to reset your password.</p>
    <form onsubmit="resetPassword(event)">
      <input type="text" id="name" placeholder="Full Name" required>
      <input type="email" id="email" placeholder="Email" required>
      <div class="input-container">
        <input type="password" id="new_password" placeholder="New Password" required>
        <span class="toggle-password" onclick="togglePassword('new_password')">Show</span></div>
      <div class="input-container">
        <input type="password" id="confirm_password" placeholder="Confirm Password" required>
        <span class="toggle-password" onclick="togglePassword('confirm_password')">Show</span></div>
      <button type="submit">Reset Password</button></form>
    <p><a href="signin.html">Back to Sign In</a></p> </div>
  <script>
    function resetPassword(event) {
      event.preventDefault(); // Prevent form submission
      let name = document.getElementById("name").value;
      let email = document.getElementById("email").value;
      let newPassword = document.getElementById("new_password").value;
      let confirmPassword = document.getElementById("confirm_password").value;
      // Get stored users from Local Storage
      let users = JSON.parse(localStorage.getItem("users")) || [];
      // Find the user by email
      let userIndex = users.findIndex(user => user.email === email);
      if (userIndex === -1) {
        alert("No account found with this email!");
        return;
      }if (newPassword !== confirmPassword) {
        alert("Passwords do not match!");
        return;}
      // Update password for the user
      users[userIndex].password = newPassword;
      localStorage.setItem("users", JSON.stringify(users));
      alert("Password reset successful! Redirecting to Sign In...");
```

```
        window.location.href = "signin.html"; // Redirect to sign-in page
    }function togglePassword(fieldId) {
        let field = document.getElementById(fieldId);
        let toggleBtn = field.nextElementSibling;
        if (field.type === "password") {
            field.type = "text";
            toggleBtn.textContent = "Hide";
        } else {
            field.type = "password";
            toggleBtn.textContent = "Show";
        }}</script>
</body></html>
```

//dashboard.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8"><meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Fee Payment Dashboard</title>
  <style>
    /* CSS for Sidebar and Forms */
    body {
        font-family: Arial, sans-serif;
        display: flex;
    }.sidebar {
        width: 350px;
        height:1000px;
        background-color: #2c3e50;
        color: white;
        padding: 15px;
    }.sidebar h2 { text-align: center; margin-bottom: 20px}
    .sidebar a:hover {
        background-color: #34495e;
    }.content {
        margin-left: 100px;
        margin-right: 100px;
        padding: 20px;
        width: 100%;
```

```
    }form-container {margin-bottom: 30px;}
    label {display: block; margin-top: 10px; }
    input, select {
      width: 100%;
      padding: 8px;
      margin-top: 5px;
    }button {
      background-color: #2c3e50;
      color: white;
      padding: 10px 20px;
      margin-top: 10px;
      border: none;
      cursor: pointer;
    }button:hover {background-color: #34495e;}
    .hidden {display: none;}
  </style>
</head>
<body>
  <div class="sidebar">
    <h2>Dashboard</h2>
    <a href="#" onclick="showForm('examFeeForm')">Examination Fee</a>
    <a href="#" onclick="showForm('hostelFeeForm')">Hostel Monthly Fee</a></div>
  <div class="content" >
    <!-- Default Form (Empty Labels) -->
    <div id="defaultForm" class="form-container" >
      <h2>Student Details</h2>
      <label>Student Name:</label>
      <input type="text" id="defaultName" placeholder="Enter your name" readonly >
      <label>Email:</label>
      <input type="email" id="defaultEmail" placeholder="Enter your email" readonly>  </div>
    <!-- Examination Fee Form (Hidden Initially) -->
    <div id="examFeeForm" class="form-container hidden">
      <h2>Examination Fee Payment</h2>
      <form onsubmit="saveExamFee(event)">
        <label>Hall Ticket No:</label>
        <input type="text" id="hallTicket" placeholder="Hall Ticket" required>
        <label>Student Name:</label>
```

```
            <input type="text" id="studentName" placeholder="Full Name" required readonly>
            <label>Email:</label>
            <input type="email" id="studentEmail" placeholder="Email" required readonly>
            <label>Mobile Number:</label>
            <input type="tel" id="MobileNumber" placeholder="Mobile Number" required>
            <label for="schoolType">School Type:</label>
            <select id="schoolType" onchange="updateCourseTypes()" required>
               <option value="">-- Select School Type --</option>
               <option value="School of Sciences">School of Sciences</option>
               <option value="School of Social Sciences">School of Social Sciences</option>
               <option value="School of Engineering & Technology">School of Engineering & Technology</option>
               <option value="School of Nursing">School of Nursing</option>
            </select><label for="courseType">Course Type:</label>
            <select id="courseType" onchange="updateCourseNames()" required>
               <option value="">-- Select Course Type --</option></select>
            <label for="courseName">Course Name:</label>
            <select id="courseName" required>
               <option value="">-- Select Course Name --</option> </select>
                <label>Semester/Year:</label>  <input type="text" id="semesterYear" placeholder="Semester/Year"
required>
            <label>Date (dd/mm/yyyy):</label>
            <input type="text" id="examDate" readonly />
            <label for="examFeeType">Fee Type:</label>
            <select id="examFeeType" required>
               <option value="">-- Select Fee Type --</option>
               <option value="semester_fee">Semester Fee</option>
               <option value="late_fee">Late Fee</option>
               <option value="college_fee">College Fee</option></select>
            <label>Amount:</label>
            <input type="text" id="examAmount" placeholder="Amount">
            <button type="submit">Pay Examination Fee</button></form></div>
      <!-- Hostel Fee Form (Hidden Initially) -->
      <div id="hostelFeeForm" class="form-container hidden">
         <h2>Hostel Fee Payment</h2>
         <form onsubmit="saveHostelFee(event)">
            <label>Hall Ticket No:</label>
            <input type="text" id="hallTicket" placeholder="Hall Ticket" required>
```

```
<label>Student Name:</label>
<input type="text" id="hostelName" placeholder="Full Name" required>
<label>Email:</label>
<input type="email" id="hostelEmail" placeholder="Email" required>
<label>Mobile Number:</label>
<input type="tel" id="hostelMobileNumber" placeholder="Mobile Number" required>
<label for="hostelSchoolType">School Type:</label>
<select id="hostelSchoolType" onchange="updateHostelCourseTypes()" required>
   <option value="">-- Select School Type --</option>
   <option value="School of Sciences">School of Sciences</option>
   <option value="School of Social Sciences">School of Social Sciences</option>
   <option value="School of Engineering & Technology">School of Engineering & Technology</option>
   <option value="School of Nursing">School of Nursing</option>  </select>
<label for="hostelCourseType">Course Type:</label>
<select id="hostelCourseType" onchange="updateHostelCourseNames()" required>
   <option value="">-- Select Course Type --</option>
</select><label for="hostelCourseName">Course Name:</label>
<select id="hostelCourseName" required>
   <option value="">-- Select Course Name --</option></select>
<label>Semester/Year:</label>
<input type="text" id="hostelSemesterYear" placeholder="Semester/Year" required>
<label>Room Number:</label>
<input type="text" id="hostelRoomNumber" placeholder="Room Number" required>
<label>Date (dd/mm/yyyy):</label>
<input type="text" id="hostelDate" readonly />
<label for="payingMonth">Paying Month:</label>
<select id="payingMonth" onchange="updateDaysDropdown()" required>
   <option value="">-- Select Month --</option>
   <option value="January">January</option>
   <option value="February">February</option>
   <option value="March">March</option>
   <option value="April">April</option>
   <option value="May">May</option>
   <option value="June">June</option>
   <option value="July">July</option>
   <option value="August">August</option>
   <option value="September">September</option>
```

```
            <option value="October">October</option>
            <option value="November">November</option>
            <option value="December">December</option> </select>
        <label for="numberOfDays">Number of Days:</label>
        <select id="numberOfDays" required>
            <option value="">-- Select Days --</option> </select>
        <label for="hostelFeeType">Fee Type:</label>
        <select id="hostelFeeType" required>
            <option value="">-- Select Fee Type --</option>
            <option value="hostel_fee">Hostel Fee</option>
            <option value="maintenance_fee">Maintenance Fee</option> </select>
        <label>Amount:</label>
        <input type="text" id="hostelAmount" readonly>
        <button type="submit">Pay Hostel Fee</button></form></div></div>
<script>
    // JavaScript for form visibility and dropdowns
    function showForm(formId) {
        const forms = document.querySelectorAll('.form-container');
        forms.forEach(form => form.classList.add('hidden')); // Hide all forms
        document.getElementById(formId).classList.remove('hidden'); // Show the selected form
    }// Function to load student details from localStorage
    function loadStudentDetails() {
        let currentUser = JSON.parse(localStorage.getItem("currentUser"));
        if (currentUser) {
            document.getElementById("defaultName").value = currentUser.username;
            document.getElementById("defaultEmail").value = currentUser.email;
            // Autofill the Examination Fee Payment form
            document.getElementById("studentName").value = currentUser.username;
            document.getElementById("studentEmail").value = currentUser.email;
            // Autofill the Hostel Fee Payment form
            document.getElementById("hostelName").value = currentUser.username;
            document.getElementById("hostelEmail").value = currentUser.email;
        }} // Update course types based on school type for Examination Fee
    function updateCourseTypes() {
        const schoolType = document.getElementById('schoolType').value;
        const courseTypeSelect = document.getElementById('courseType');
        courseTypeSelect.innerHTML = '<option value="">-- Select Course Type --</option>';
```

```
    let courseTypes = [];
    if (schoolType === 'School of Sciences') {
        courseTypes = ['B.Sc', 'M.Sc'];
    } else if (schoolType === 'School of Social Sciences') {
        courseTypes = ['B.A', 'M.A'];
    } else if (schoolType === 'School of Engineering & Technology') {
        courseTypes = ['B.Tech', 'M.Tech'];
    } else if (schoolType === 'School of Nursing') {
        courseTypes = ['B.Sc Nursing', 'M.Sc Nursing'];
    }courseTypes.forEach(type => {
        const option = document.createElement('option');
        option.value = type;
        option.textContent = type;
        courseTypeSelect.appendChild(option);
    });} // Update course names based on course type for Examination Fee
function updateCourseNames() {
    const courseType = document.getElementById('courseType').value;
    const courseNameSelect = document.getElementById('courseName');
    courseNameSelect.innerHTML = '<option value="">-- Select Course Name --</option>';
    let courseNames = [];
    if (courseType === 'B.Sc') {
        courseNames = ['Physics', 'Chemistry', 'Mathematics'];
    } else if (courseType === 'M.Sc') {
        courseNames = ['Physics', 'Chemistry', 'Biology'];
    } else if (courseType === 'B.A') {
        courseNames = ['History', 'English', 'Sociology'];
    } else if (courseType === 'M.A') {
        courseNames = ['English', 'Sociology', 'Psychology'];
    } else if (courseType === 'B.Tech') {
        courseNames = ['Computer Science', 'Mechanical Engineering', 'Electrical Engineering'];
    } else if (courseType === 'M.Tech') {
        courseNames = ['Computer Science', 'Mechanical Engineering'];
    } else if (courseType === 'B.Sc Nursing') {
        courseNames = ['General Nursing', 'Pediatric Nursing'];
    } else if (courseType === 'M.Sc Nursing') {
        courseNames = ['Psychiatric Nursing', 'Community Health Nursing'];
    }courseNames.forEach(name => {
```

```
        const option = document.createElement('option');
        option.value = name;
        option.textContent = name;
        courseNameSelect.appendChild(option);
    });} // Update course types based on school type for Hostel Fee
function updateHostelCourseTypes() {
    const schoolType = document.getElementById('hostelSchoolType').value;
    const courseTypeSelect = document.getElementById('hostelCourseType');
    courseTypeSelect.innerHTML = '<option value="">-- Select Course Type --</option>';
    let courseTypes = [];
    if (schoolType === 'School of Sciences') {
        courseTypes = ['B.Sc', 'M.Sc'];
    } else if (schoolType === 'School of Social Sciences') {
        courseTypes = ['B.A', 'M.A'];
    } else if (schoolType === 'School of Engineering & Technology') {
        courseTypes = ['B.Tech', 'M.Tech'];
    } else if (schoolType === 'School of Nursing') {
        courseTypes = ['B.Sc Nursing', 'M.Sc Nursing'];
    }courseTypes.forEach(type => {
        const option = document.createElement('option');
        option.value = type;
        option.textContent = type;
        courseTypeSelect.appendChild(option);
    });}
function updateHostelCourseNames() {
    const courseType = document.getElementById('hostelCourseType').value;
    const courseNameSelect = document.getElementById('hostelCourseName');
    courseNameSelect.innerHTML = '<option value="">-- Select Course Name --</option>';
    let courseNames = [];
    if (courseType === 'B.Sc') {
        courseNames = ['Physics', 'Chemistry', 'Mathematics'];
    } else if (courseType === 'M.Sc') {
        courseNames = ['Physics', 'Chemistry', 'Biology'];
    } else if (courseType === 'B.A') {
        courseNames = ['History', 'English', 'Sociology'];
    } else if (courseType === 'M.A') {
        courseNames = ['English', 'Sociology', 'Psychology'];
```

```
    } else if (courseType === 'B.Tech') {
        courseNames = ['Computer Science', 'Mechanical Engineering', 'Electrical Engineering'];
    } else if (courseType === 'M.Tech') {
        courseNames = ['Computer Science', 'Mechanical Engineering'];
    } else if (courseType === 'B.Sc Nursing') {
        courseNames = ['General Nursing', 'Pediatric Nursing'];
    } else if (courseType === 'M.Sc Nursing') {
        courseNames = ['Psychiatric Nursing', 'Community Health Nursing'];
    }courseNames.forEach(name => {
        const option = document.createElement('option');
        option.value = name;
        option.textContent = name;
        courseNameSelect.appendChild(option);
    });
}// Function to set default date in dd/mm/yyyy format
function setDefaultDate() {
    const today = new Date();
    const dd = String(today.getDate()).padStart(2, '0');
    const mm = String(today.getMonth() + 1).padStart(2, '0'); // January is 0!
    const yyyy = today.getFullYear()
    const formattedDate = `${dd}/${mm}/${yyyy}`;
    // Set default date in the form fields
    document.getElementById('examDate').value = formattedDate;
    document.getElementById('hostelDate').value = formattedDate;
}// function to set DaysDropdown
function updateDaysDropdown() {
const month = document.getElementById("payingMonth").value;
const daysDropdown = document.getElementById("numberOfDays");
// Clear previous options
daysDropdown.innerHTML = '<option value="">-- Select Days --</option>';
let days = 31; // Default
if (month === "February") {
    days = 28; // Non-leap year default
    const currentYear = new Date().getFullYear();
    if ((currentYear % 4 === 0 && currentYear % 100 !== 0) || (currentYear % 400 === 0)) {
        days = 29; // Leap year
    }} else if (["April", "June", "September", "November"].includes(month)) {
```

```
    days = 30; // Months with 30 days} // Populate days dropdown based on the selected month
for (let i = 1; i <= days; i++) {
    const option = document.createElement("option");
    option.value = i;
    option.textContent = i;
    daysDropdown.appendChild(option);
}} // Update amount for examination fee form
function updateExamAmount() {
    const feeType = document.getElementById('examFeeType').value;
    const examAmountInput = document.getElementById('examAmount');
    let amount = '';
    if (feeType === 'semester_fee') {
        amount = '1000';
    } else if (feeType === 'late_fee') {
        amount = '300';
    } else if (feeType === 'college_fee') {
        amount = '25000';
    }examAmountInput.value = amount ? `₹${amount}` : '';   }
// Update amount for hostel fee form
function updateHostelAmount() {
    const feeType = document.getElementById('hostelFeeType').value;
    const hostelAmountInput = document.getElementById('hostelAmount')
    let amount = '';
    if (feeType === 'hostel_fee') {
        amount = '1500';
    } else if (feeType === 'maintenance_fee') {
        amount = '500';
    }   hostelAmountInput.value = amount ? `₹${amount}` : '';
}   // Event listener for 'examFeeType' to update exam fee amount
document.getElementById('examFeeType').addEventListener('change', updateExamAmount);
// Event listener for 'hostelFeeType' to update hostel fee amount
document.getElementById('hostelFeeType').addEventListener('change', updateHostelAmount);
// Placeholder function for saving exam fee and redirecting to Payment Method page
function saveExamFee(event) {
    event.preventDefault();
    const hallTicket = document.getElementById('hallTicket').value;
    const studentName = document.getElementById('studentName').value;
```

```
    const studentEmail = document.getElementById('studentEmail').value;
    const mobileNumber = document.getElementById('MobileNumber').value;
    const courseType = document.getElementById('courseType').value;
    const courseName = document.getElementById('courseName').value;
    const semester = document.getElementById('semesterYear').value;
    const feeTypeDetails = document.getElementById('examFeeType').value;
    const amountPaid = document.getElementById('examAmount').value;
    // Store in localStorage
    localStorage.setItem("feeType", "examination");
    localStorage.setItem("hallTicket", hallTicket);
    localStorage.setItem("studentName", studentName);
    localStorage.setItem("studentEmail", studentEmail);
    localStorage.setItem("mobileNumber", mobileNumber);
    localStorage.setItem("courseType", courseType);
    localStorage.setItem("courseName", courseName);
    localStorage.setItem("semester", semester);
    localStorage.setItem("feeTypeDetails", feeTypeDetails);
    localStorage.setItem("amountPaid", amountPaid);
    // Redirect to Payment Method page with query parameters
    const queryParams = new URLSearchParams({
        feeType: 'examination',
        hallTicket,
        studentName,
        studentEmail,
        mobileNumber,
        courseType,
        courseName,
        semester,
        feeTypeDetails,
        amountPaid,
    }).toString();
    window.location.href = `paymentMethod.html?${queryParams}`;
}// Placeholder function for saving hostel fee and redirecting to Payment Method page
function saveHostelFee(event) {
    event.preventDefault();
    const hallTicket = document.getElementById('hallTicket').value;
    const studentName = document.getElementById('hostelName').value;
```

```
        const studentEmail = document.getElementById('hostelEmail').value;
        const mobileNumber = document.getElementById('hostelMobileNumber').value;
        const courseType = document.getElementById('hostelCourseType').value;
        const courseName = document.getElementById('hostelCourseName').value;
        const semester = document.getElementById('hostelSemesterYear').value;
        const roomNumber = document.getElementById('hostelRoomNumber').value;
        const feeTypeDetails = document.getElementById('hostelFeeType').value;
        const amountPaid = document.getElementById('hostelAmount').value;
        // Store in localStorage
        localStorage.setItem("feeType", "hostel");
        localStorage.setItem("hallTicket", hallTicket);
        localStorage.setItem("studentName", studentName);
        localStorage.setItem("studentEmail", studentEmail);
        localStorage.setItem("mobileNumber", mobileNumber);
        localStorage.setItem("courseType", courseType);
        localStorage.setItem("courseName", courseName);
        localStorage.setItem("semester", semester);
        localStorage.setItem("roomNumber", roomNumber);
        localStorage.setItem("feeTypeDetails", feeTypeDetails);
        localStorage.setItem("amountPaid", amountPaid);
        // Redirect to Payment Method page with query parameters
        const queryParams = new URLSearchParams({
            feeType: 'hostel',
            hallTicket,
            hostelName,
            hostelEmail,
        }).toString();
        window.location.href = `paymentMethod.html?${queryParams}`;
    }// Set default date when the page loads
    window.onload = function () {
        loadStudentDetails();
        setDefaultDate(); // Call the function to set default date
    };
  </script></body></html>
```

//paymentMethod.html

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Payment Method</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f9f9f9;
        }.container {
            max-width: 600px;
            margin: 50px auto;
            background: #fff;
            padding: 20px;
            border-radius: 10px;
            box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
        }h2 {
            text-align: center;
            margin-bottom: 20px;
            font-size: 24px;
            color: #333;
        }.payment-options {
            margin-top: 20px;
        }.payment-options label {
            display: block;
            margin: 10px 0;
            font-size: 16px;
            color: #555;
        }.card-details {
            margin-top: 20px;
            display: none;
        }.card-details h3 {
            font-size: 18px;
            margin-bottom: 15px;
            color: #333;
        }.card-details label {
            display: block;
            margin-top: 10px;
```

```
            font-size: 14px;
            color: #555;
        }.card-details input,
        .card-details select {
            width: 100%;
            padding: 10px;
            margin-top: 5px;
            font-size: 14px;
            border: 1px solid #ccc;
            border-radius: 5px;
            box-sizing: border-box;
        }
        .card-details input:focus,
        .card-details select:focus {
            outline: none;
            border-color: #007BFF;
            box-shadow: 0 0 5px rgba(0, 123, 255, 0.5);
        }.expiry-date {
            display: flex;
            gap: 10px;
            margin-top: 5px;
        }.pay-button {
            display: block;
            width: 100%;
            padding: 10px;
            background-color: #007BFF;
            color: #fff;
            text-align: center;
            text-decoration: none;
            border-radius: 5px;
            margin-top: 20px;
        }.pay-button:hover {
            background-color: #0056b3;}
    </style>
</head>
<body>
    <div class="container">
```

```
<h2>Payment Method</h2>
<p id="feeDetails"></p>
<div class="payment-options">
<label> <input type="radio" name="paymentMethod" value="Credit Card" onclick="showCardDetails()">
Credit Card </label>
<label><input type="radio" name="paymentMethod" value="Debit Card" onclick="showCardDetails()"> Debit Card</label>
<label><input type="radio" name="paymentMethod" value="UPI" onclick="showUpiDetails()">UPI</label>
<label><input type="radio" name="paymentMethod" value="Bank Transfer" onclick="showBankDetails()"> Bank Transfer</label> </div>
<div class="card-details" id="cardDetails">
<h3>Enter Card Details</h3>
<label>Card Number:</label>
<input type="text" id="cardNumber" placeholder="Enter your card number" required>
<label>Cardholder Name:</label>
<input type="text" id="cardholderName" placeholder="Enter cardholder's name" required>
<label>Expiry Date:</label>
<div class="expiry-date">
<select id="expiryMonth" required>
<option value="" disabled selected>Month</option>
<option value="01">January</option>
<option value="02">February</option>
<option value="03">March</option>
<option value="04">April</option>
<option value="05">May</option>
<option value="06">June</option>
<option value="07">July</option>
<option value="08">August</option>
<option value="09">September</option>
<option value="10">October</option>
<option value="11">November</option>
<option value="12">December</option></select></div>
<label>CVV:</label>
<input type="text" id="cvv" placeholder="Enter CVV" required> </div>
<div class="card-details" id="upiDetails" style="display: none;">
<h3>Enter UPI Transaction Details</h3>
<label>UPI ID:</label>
<input type="text" id="upiId" placeholder="Enter your UPI ID">
```

```
        <label>Transaction ID:</label>
        <input type="text" id="upiTransactionId" placeholder="Enter UPI Transaction ID"> </div>
    <div class="card-details" id="bankDetails" style="display: none;">
        <h3>Enter Bank Transaction Details</h3>
        <label>Account Holder Name:</label>
        <input type="text" id="accountHolder" placeholder="Enter Account Holder Name">
        <label>Account Number:</label>
        <input type="text" id="accountNumber" placeholder="Enter Account Number">
        <label>IFSC Code:</label>
        <input type="text" id="ifscCode" placeholder="Enter IFSC Code">
    </div><a href="#" class="pay-button" onclick="proceedToPay()">Proceed to Pay</a>  </div>
<script>
    // Get query parameters from the URL
    const urlParams = new URLSearchParams(window.location.search);
    const feeType = urlParams.get('feeType');
    const studentName = urlParams.get('studentName') || urlParams.get('hostelName');
    const email = urlParams.get('studentEmail') || urlParams.get('hostelEmail');
    // Display fee details dynamically
    const feeDetails = document.getElementById('feeDetails');
    if (feeType === 'examination') {
        feeDetails.textContent = `Examination Fee Payment for ${studentName} (${email}).`;
    } else if (feeType === 'hostel') {
        feeDetails.textContent = `Hostel Fee Payment for ${studentName} (${email}).`;
    }function showCardDetails() {
        document.getElementById("cardDetails").style.display = "block";
        document.getElementById("upiDetails").style.display = "none";
        document.getElementById("bankDetails").style.display = "none";
    }function showUpiDetails() {
        document.getElementById("cardDetails").style.display = "none";
        document.getElementById("upiDetails").style.display = "block";
        document.getElementById("bankDetails").style.display = "none";
    }function showBankDetails() {
        document.getElementById("cardDetails").style.display = "none";
        document.getElementById("upiDetails").style.display = "none";
        document.getElementById("bankDetails").style.display = "block";
    }// Populate expiryYear dropdown with current year and next 10 years
    function populateExpiryYear() {
```

```
        const currentYear = new Date().getFullYear();
        const expiryYearSelect = document.getElementById('expiryYear');
        for (let i = 0; i <= 10; i++) {
            const yearOption = document.createElement('option');
            yearOption.value = currentYear + i;
            yearOption.textContent = currentYear + i;
            expiryYearSelect.appendChild(yearOption);
        }}// Call the function when the page loads
    populateExpiryYear();
    function proceedToPay() {
    const feeType = localStorage.getItem("feeType");
    const feeTypeDetails = localStorage.getItem("feeTypeDetails"); // Retrieve specific fee type
    let amountPaid = "N/A";
    if (feeType === "examination") {
        if (feeTypeDetails === "semester_fee") {
            amountPaid = "₹1000";
        } else if (feeTypeDetails === "late_fee") {
            amountPaid = "₹300";
} else if (feeTypeDetails === "college_fee") {
            amountPaid = "₹25000";  }
    } else if (feeType === "hostel") {
        if (feeTypeDetails === "hostel_fee") {
            amountPaid = "₹1500";
        } else if (feeTypeDetails === "maintenance_fee") {
            amountPaid = "₹500";}}
    const selectedMethod = document.querySelector('input[name="paymentMethod"]:checked');
    const cardNumber = document.getElementById('cardNumber')?.value || '';
    const cardholderName = document.getElementById('cardholderName')?.value || '';
    const expiryMonth = document.getElementById('expiryMonth')?.value || '';
    const expiryYear = document.getElementById('expiryYear')?.value || '';
    const cvv = document.getElementById('cvv')?.value || '';
    const upiId = document.getElementById('upiId')?.value || '';
    const accountHolder = document.getElementById('accountHolder')?.value || '';
    const accountNumber = document.getElementById('accountNumber')?.value || '';
    const ifscCode = document.getElementById('ifscCode')?.value || '';
    if (!selectedMethod) {
        alert('Please select a payment method.');
```

```
            return;
        }let requiredFields = [];
            if (selectedMethod.value === 'UPI') {
                requiredFields = ['upiId', 'upiTransactionId'];
            } else if (selectedMethod.value === 'Bank Transfer') {
                requiredFields = ['accountHolder', 'accountNumber', 'ifscCode'];
            } else if (selectedMethod.value === 'Credit Card' || selectedMethod.value === 'Debit Card') {
                requiredFields = ['cardNumber', 'cardholderName', 'expiryMonth', 'expiryYear', 'cvv'];
            }
            for (let field of requiredFields) {
                if (!document.getElementById(field).value) {
                    alert('Please fill out all required details.');
                    return;  }  }
        localStorage.setItem("transactionId", `TXN${Math.floor(Math.random() * 100000000)}`);
        localStorage.setItem("amountPaid", amountPaid);
        localStorage.setItem("paymentDate", new Date().toLocaleDateString());
    if (feeType === "examination") {
        const hallTicket = new URLSearchParams(window.location.search).get("hallTicket");
        const studentName = new URLSearchParams(window.location.search).get("studentName");
        const studentEmail = new URLSearchParams(window.location.search).get("studentEmail");
        localStorage.setItem("hallTicket", hallTicket);
        localStorage.setItem("studentName", studentName);
        localStorage.setItem("studentEmail", studentEmail);
    } else if (feeType === "hostel") {
        const hallTicket = new URLSearchParams(window.location.search).get("hallTicket");
        const studentName = new URLSearchParams(window.location.search).get("hostelName");
        const studentEmail = new URLSearchParams(window.location.search).get("hostelEmail");
        localStorage.setItem("hallTicket", hallTicket);
        localStorage.setItem("hostelName", studentName);
        localStorage.setItem("hostelEmail", studentEmail);
    }    window.location.href = "receipt.html?feeType=" + feeType;}
    </script>
</body>
</html>
```

//receipt.html

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Payment Receipt</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f9f9f9;
        }.container {
            max-width: 600px;
            margin: 50px auto;
            background: #fff;
            padding: 20px;
            border-radius: 10px;
            box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
        }h2 {margin-bottom: 20px;font-size: 24px;text-align: center;
            color: #333;
        }  .details-table {
            width: 100%;
            margin: 20px 0;
            border-collapse: collapse;
        } .details-table th,
        .details-table td {
            text-align: left;
            padding: 10px;
            border-bottom: 1px solid #ddd;
            font-size: 14px;
        }  .details-table th {
            font-weight: bold;
            background-color: #f5f5f5;
        }.download-btn {
            display: block;
            margin: 20px auto;
            padding: 10px 20px;
            background-color: #007BFF;
```

```
            color: #fff;
            text-decoration: none;
            border-radius: 5px;
            text-align: center;
            font-size: 16px;
        }
        .download-btn:hover {
            background-color: #0056b3;
        }
    </style></head><body>
    <div class="container">
        <h2>Payment Receipt</h2>
        <img id="college-logo" src="https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcSjE2kyon7waVjPTo1Um_2BcIjku0T2F0EWgA&s" alt="College Logo"
style="display: block; margin: 0 auto; width: 100px; height: auto;">
        <h3 style="text-align: center;">SRAVANYA University</h3>
        <table class="details-table" id="detailsTable"></table>
        <a href="#" class="download-btn" onclick="generatePDF()">Download Receipt as PDF</a>  </div>
 <script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.4.0/jspdf.umd.min.js"></script>
    <script>
        // Retrieve data from localStorage
        const feeType = localStorage.getItem("feeType") || "N/A";
        const hallTicket = localStorage.getItem("hallTicket") || "N/A";
        const studentName = localStorage.getItem("studentName") || "N/A";
        const email = localStorage.getItem("studentEmail") || "N/A";
        const mobileNumber = localStorage.getItem("mobileNumber") || "N/A";
        const courseType = localStorage.getItem("courseType") || "N/A";
        const courseName = localStorage.getItem("courseName") || "N/A";
        const semester = localStorage.getItem("semester") || "N/A";
        const roomNumber = localStorage.getItem("roomNumber") || "N/A";
        const feeTypeDetails = localStorage.getItem("feeTypeDetails") || "N/A";
        const amountPaid = localStorage.getItem("amountPaid") || "N/A";
        const transactionId = localStorage.getItem("transactionId") || "N/A";
        const paymentDate = localStorage.getItem("paymentDate") || "N/A";
        const status = "Successful";
        const detailsTable = document.getElementById("detailsTable");
        // Create table rows dynamically
        const rows = [
```

```
    ["Student Name", studentName],
    ["Email", email],
    ["Mobile Number", mobileNumber],
    ["Hall Ticket No", hallTicket],
    ...(feeType === "examination"
      ? [
        ["Course Type", courseType],
        ["Course Name", courseName],
        ["Semester", semester],
        ["Exam Fee Type", feeTypeDetails],
      ]
      : [
        ["Course Type", courseType],
        ["Course Name", courseName],
        ["Semester", semester],
        ["Room Number", roomNumber],
        ["Hostel Fee Type", feeTypeDetails],
      ]),
    ["Transaction ID", transactionId],
    ["Amount Paid", amountPaid],
    ["Payment Date", paymentDate],
    ["Status", status],
];
rows.forEach(([label, value]) => {
    const row = `<tr>
      <th>${label}</th>
      <td>${value}</td>
    </tr>`;
    detailsTable.innerHTML += row;
});// Generate PDF
async function generatePDF() {
    const { jsPDF } = window.jspdf;
    const doc = new jsPDF();
    doc.setFont("Helvetica", "bold");
    doc.setFontSize(18);
    doc.text("Payment Receipt", 105, 20, { align: "center" });
    doc.setFont("Helvetica", "normal");
```

```
        doc.setFontSize(14);
        doc.text(
          feeType === "examination" ? "Examination Fee Receipt" : "Hostel Fee Receipt",
          105,
          30,
          { align: "center" }
        );
        // Separator line
        doc.setLineWidth(0.5);
        doc.line(15, 35, 195, 35);
        // Details
        let yPosition = 45; // Start below the line
        const labelX = 20; // X position for the labels
        const colonX = 80; // X position for the colon
        const valueX = 85; // X position for the values
        rows.forEach(([label, value]) => {
          // Draw label
          doc.setFont("Helvetica", "bold");
          doc.setFontSize(12);
          doc.text(label, labelX, yPosition);
          // Draw colon
          doc.setFont("Helvetica", "normal");
          doc.text(":", colonX, yPosition);
          // Draw value
          doc.text(value, valueX, yPosition);
          yPosition += 10; // Move to the next line
        });
        doc.line(15, yPosition, 195, yPosition);
        yPosition += 10;
        doc.setFont("Helvetica", "italic");
        doc.setFontSize(10);
        doc.text("Thank you for your payment!", 105, yPosition, { align: "center" });
        doc.save(`${feeType}_receipt.pdf`);
      }
    </script>
</body>
</html>
```