

Computer Science & Engineering Department   
D.Y. Patil College of Engineering & Technology, Kasaba Bawada Kolhapur

MINI PROJECT REPORT

SYB-CSE, AOOC, May 2025

STUDENT FEE PAYMENT SYSTEM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Batch | Group No. | Juno ID | Roll No. | Student Name | Sign |
| S1 | G2 | EN23254258 | 08 | Sanika R. Dhumal |  |
| EN23234620 | 09 | Samruddhi V. Awate |  |
| EN23218368 | 10 | Rajnandini R. Jagadale |  |
| EN23259311 | 12 | Kalyani N. Patil |  |
| EN23253817 | 19 | Prasad D. Pise |  |

Course Faculty: Dr. Kapil B. Kadam

# PROBLEM STATEMENT & INTRODUCTION

Problem Statement:

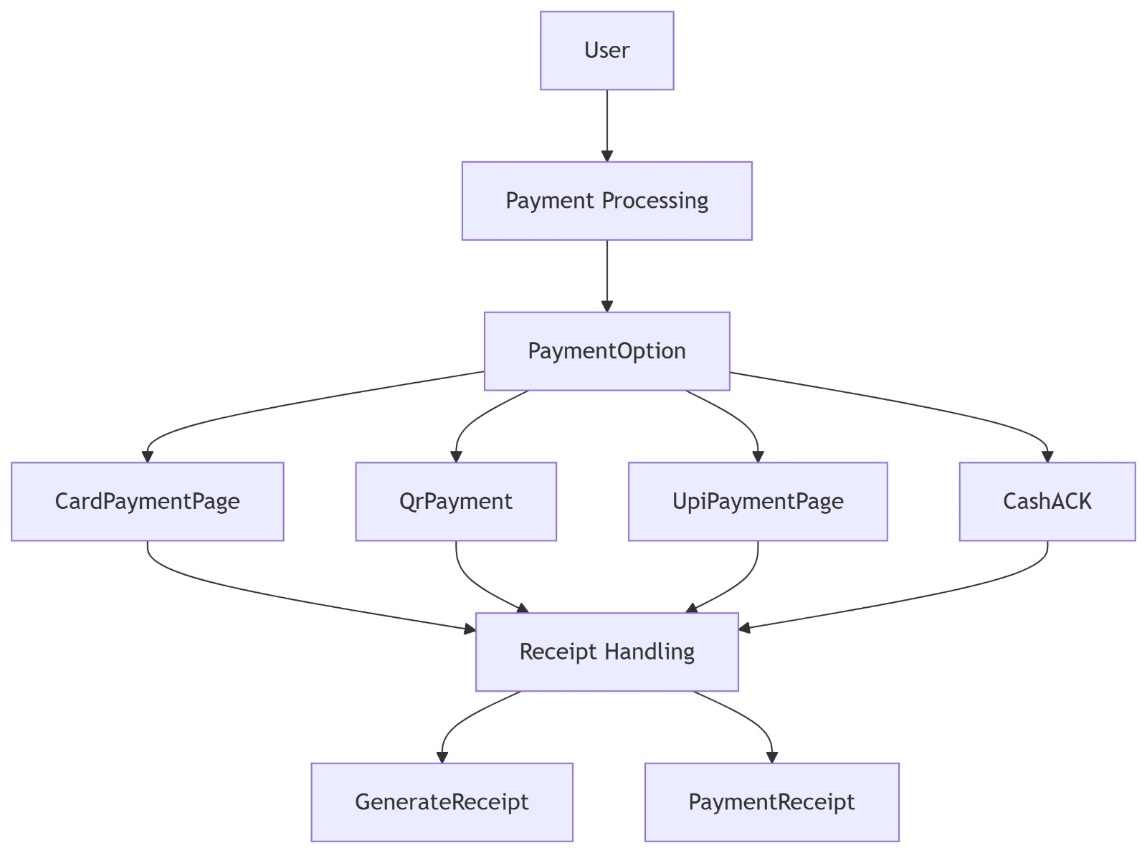
Managing student payments in educational institutions often involves manual processes that are time-consuming, prone to errors, and lack real-time tracking. The absence of a centralized digital solution creates challenges in ensuring payment security, tracking transaction history, and generating receipts promptly.

Introduction:

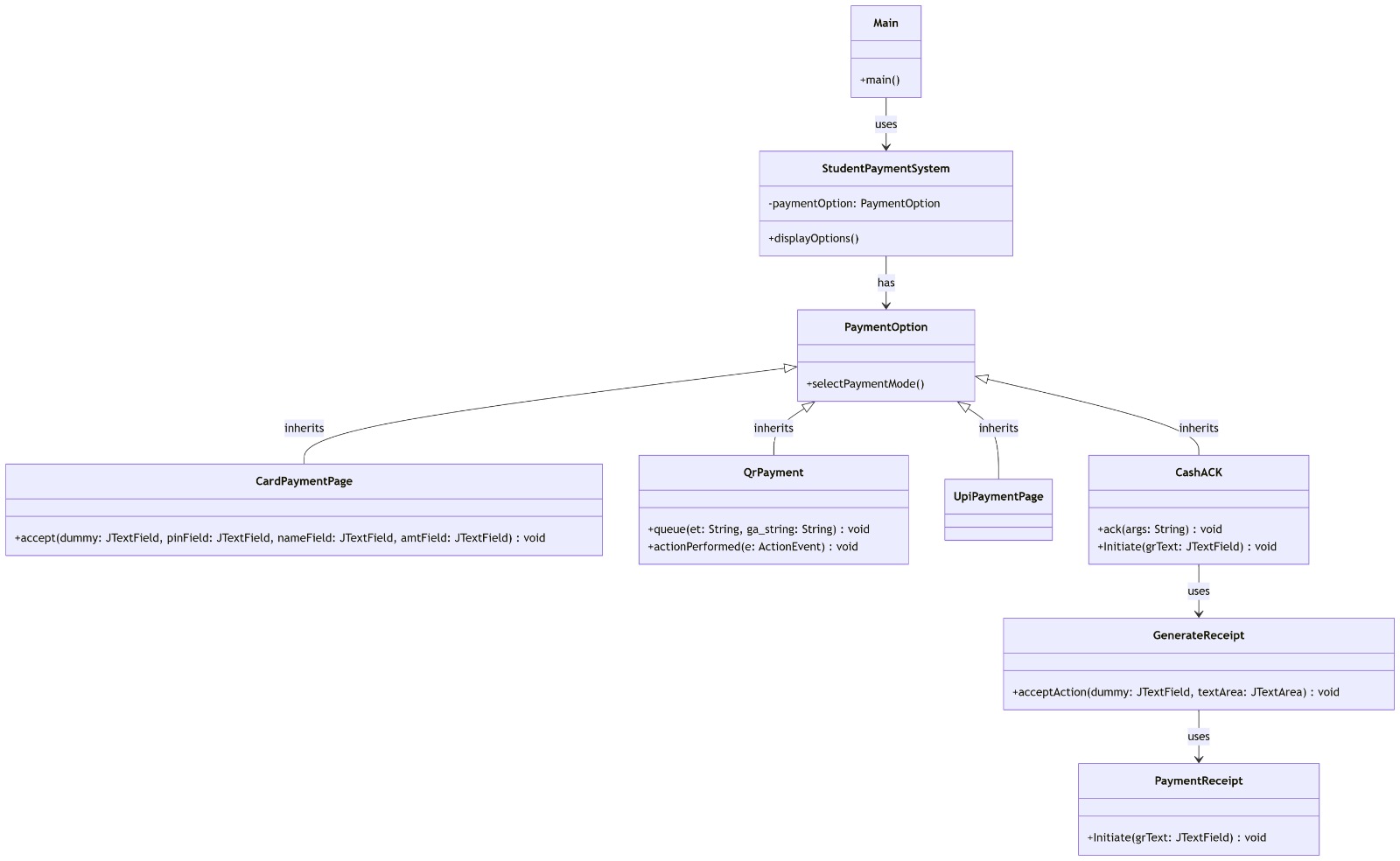
The Student Payment System is a Java-based desktop application designed to streamline and digitize the process of collecting payments from students. The system offers multiple payment methods including Card, UPI, QR code, and Cash, allowing students to choose their preferred payment option. The application ensures secure transactions and automatically generates digital receipts. Built using Java Swing and integrated with external libraries like iTextPDF for receipt generation, the system improves payment accuracy, enhances user experience, and reduces administrative burden.

# 

# 2. SYSTEM ARCHITECTURE



# 3. CLASS DIAGRAM



# 

# 4. MODULE DESCRIPTION

The Student Payment System is composed of multiple interdependent modules, each responsible for managing a specific task in the overall payment workflow. Below is a breakdown of each module with detailed functionality and integration points:

1. StudentPaymentSystem.java (Main Module)

Purpose: Acts as the starting point of the application.

Functions:

Displays a welcome or introduction screen when the application launches.

Initializes the main frame and GUI components.

Directs the user to the PaymentOption module for further actions.

Significance: Sets up the environment for user interaction and serves as the base of the application’s flow.

2. PaymentOption.java

Purpose: Provides the interface for selecting the payment method.

Functions:

Offers multiple buttons/options: Card Payment, UPI Payment, QR Payment, and Cash

On button click, opens the corresponding module using action listeners.

Integration:

Bridges the user interaction with the specific payment modules.

Significance: Core navigation module allowing flexible payment choice.

3. CardPaymentPage.java

Purpose: Facilitates payment through debit/credit cards.

Functions:

Captures user input including cardholder name, card number, expiry date, and CVV.

Basic input validation (e.g., format check).

Simulates payment confirmation.

Triggers GenerateReceipt after successful transaction.

Significance: Mimics a secure card transaction system.

4. UpiPaymentPage.java

Purpose: Enables payment using UPI ID.

Functions:

Accepts a UPI ID from the user.

Verifies the format of UPI input.

Simulates a transaction processing window.

Upon success, calls GenerateReceipt.

Significance: Modernizes the payment process with a popular digital method.

5. QrPayment.java

Purpose: Simulates payment using QR code scanning.

Functions:

Displays a QR code image that can be scanned using any UPI app.

Shows a confirmation button after scanning.

On confirmation, invokes GenerateReceipt.

Significance: Enhances usability with visual-based payment.

6. CashACK.java

Purpose: Handles manual cash payments.

Functions:

Shows a confirmation screen stating that payment has been accepted in cash.

Does not require user input; assumes manual verification.

Passes control to GenerateReceipt.

Significance: Supports traditional offline payment method.

7. GenerateReceipt.java

Purpose: Automatically generates a payment receipt.

Functions:

Collects details from the payment module (e.g., name, amount).

Uses iTextPDF library to generate a PDF receipt.

Saves receipt to the local system and displays success message.

Significance: Provides proof of payment and adds professional polish.

8. BackgroundPanel.java

Purpose: Enhances GUI aesthetics by allowing background images.

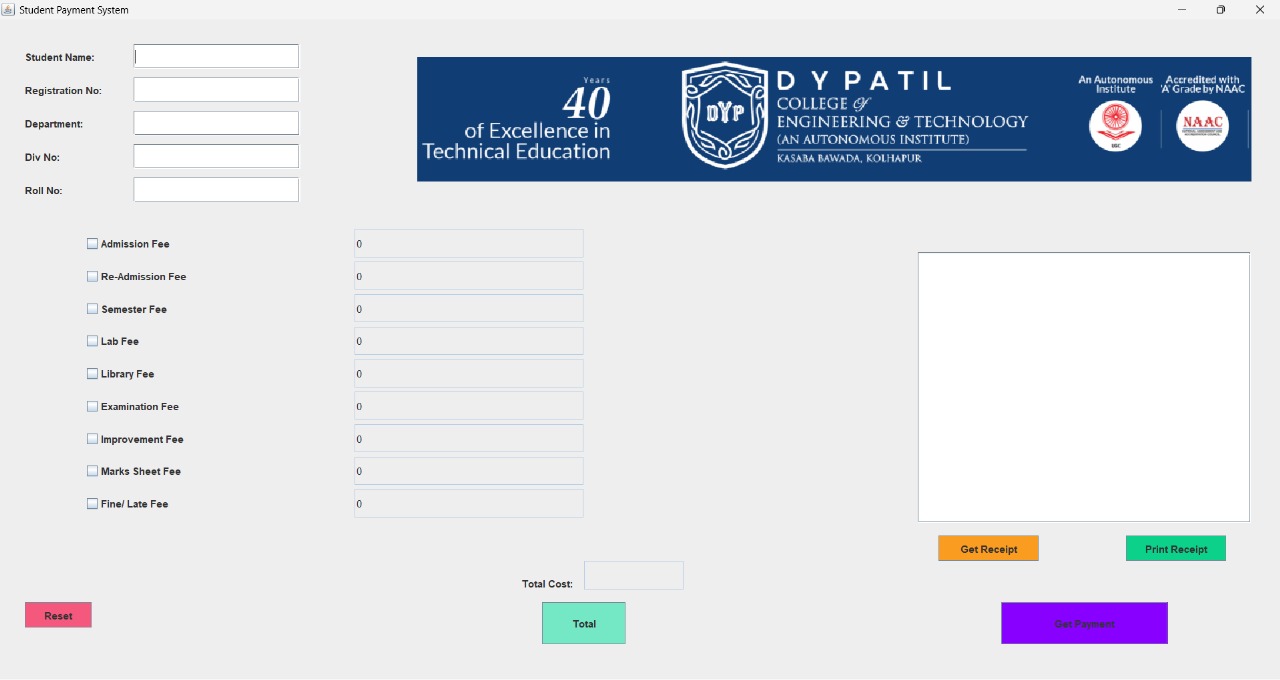
Functions:

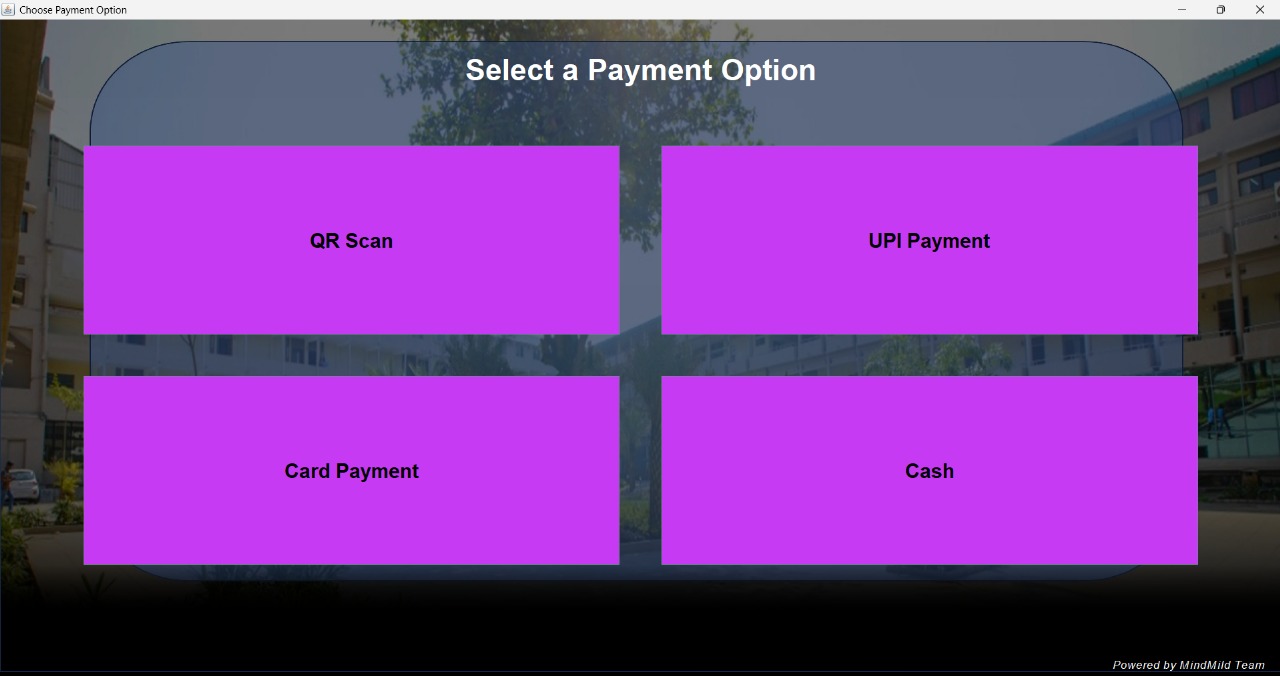
Overrides JPanel paintComponent to set an image as the background.

Can be reused across different GUI windows.

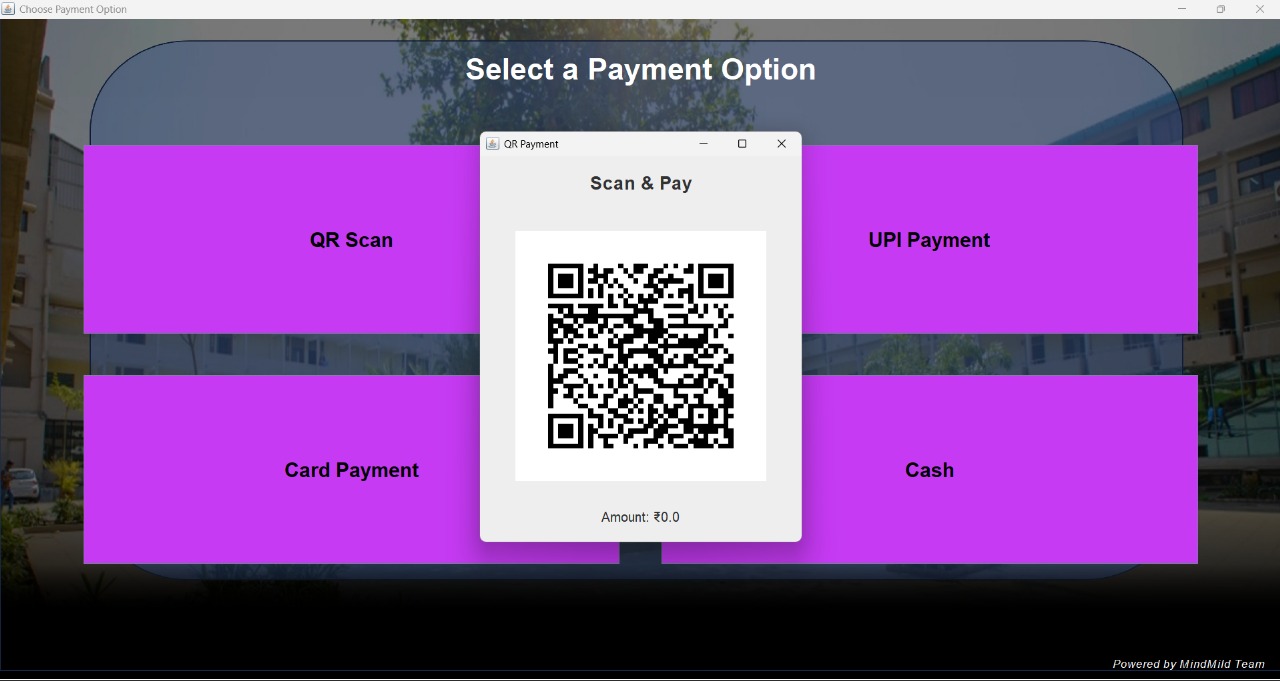
Significance: Improves visual appeal and user experience.

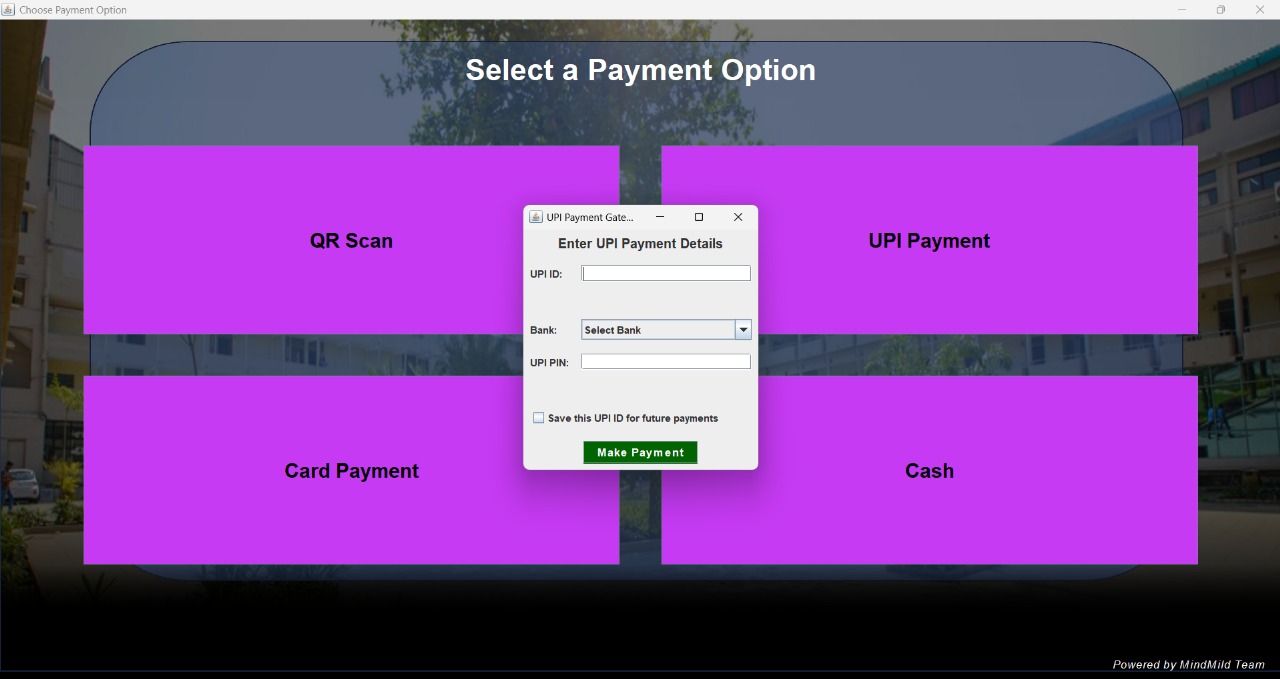
# 5. SCREEN-SHOTS

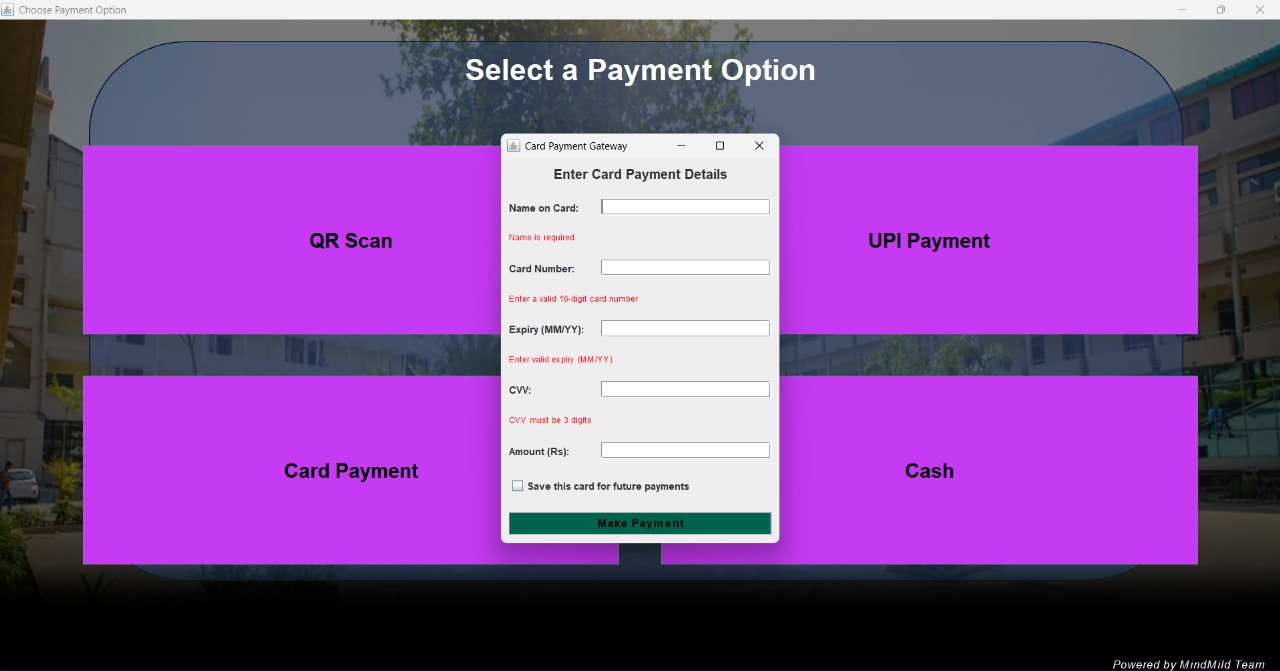


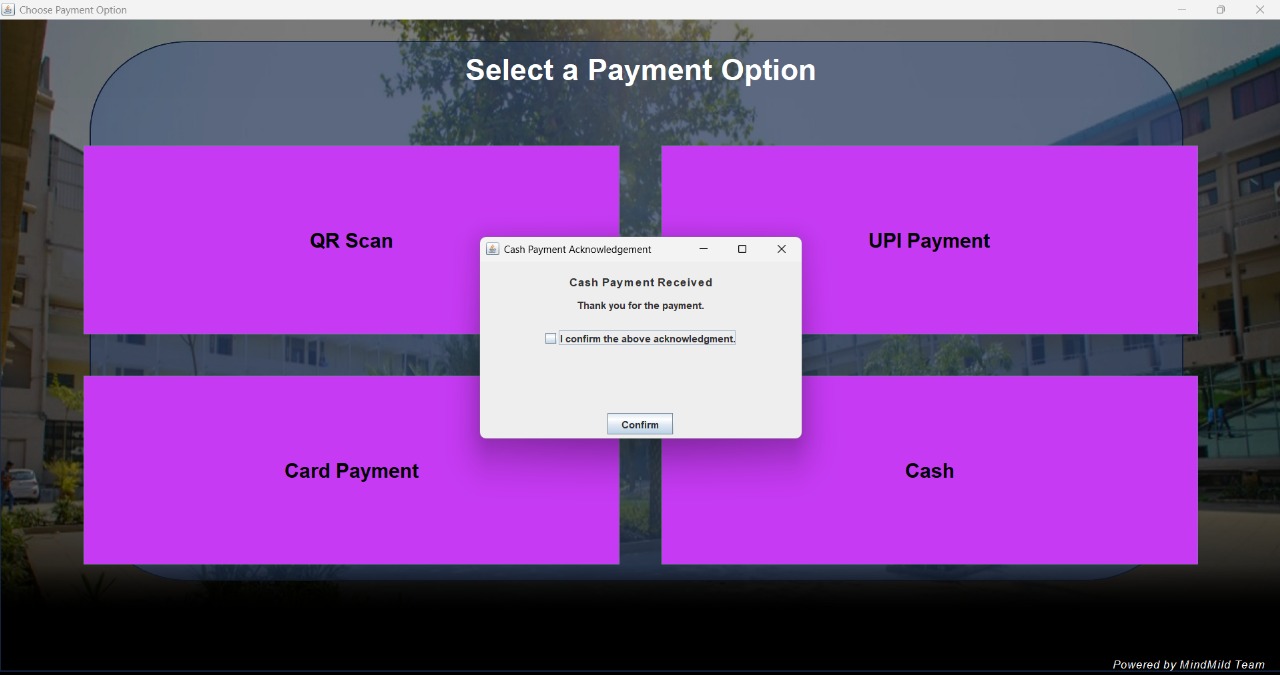


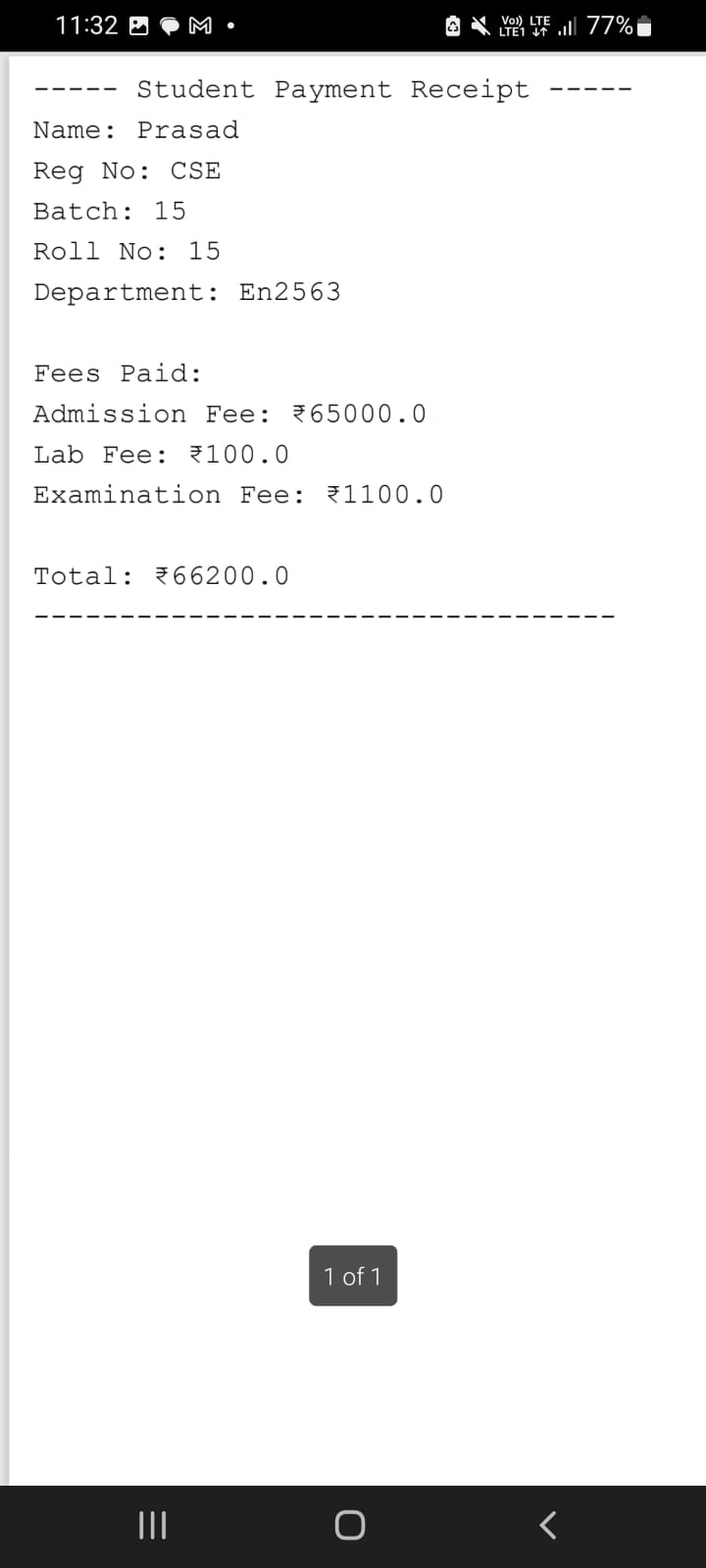


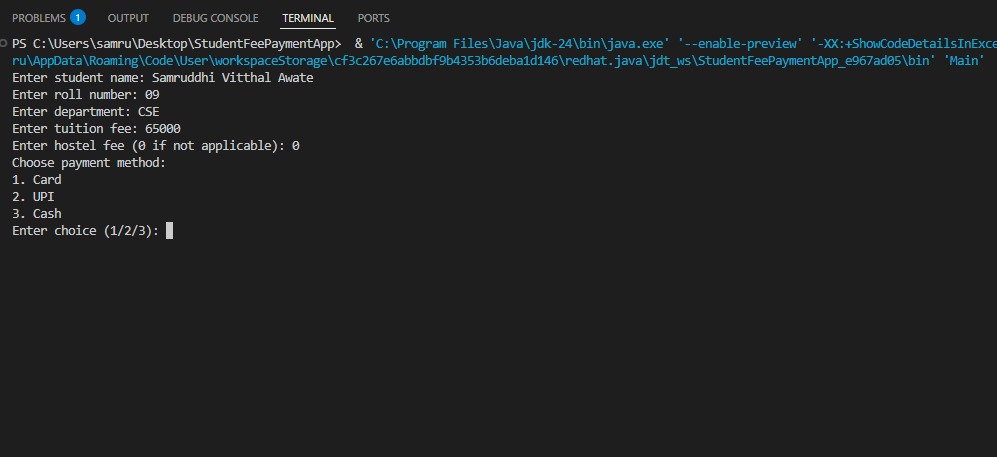


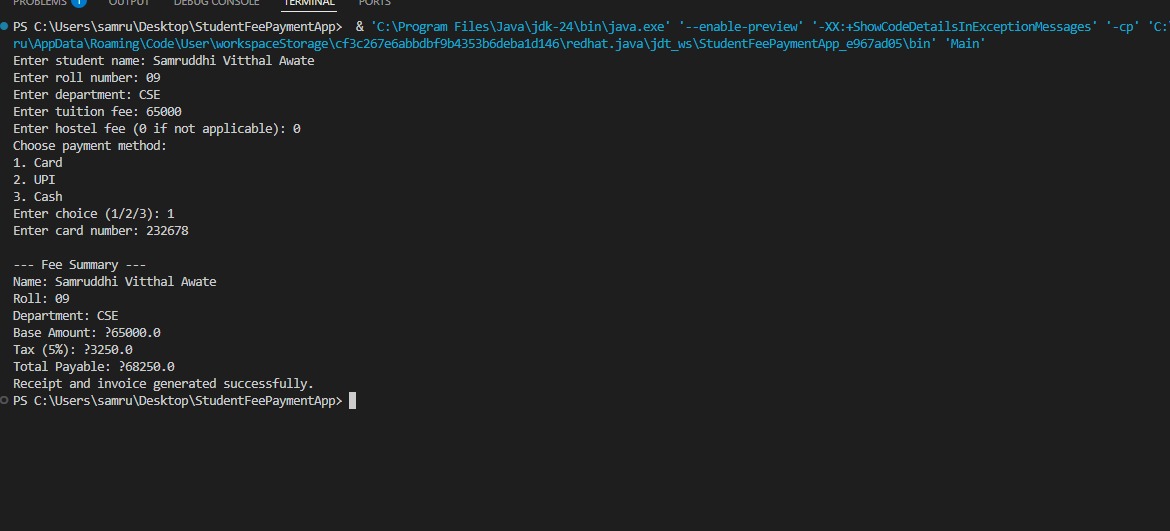


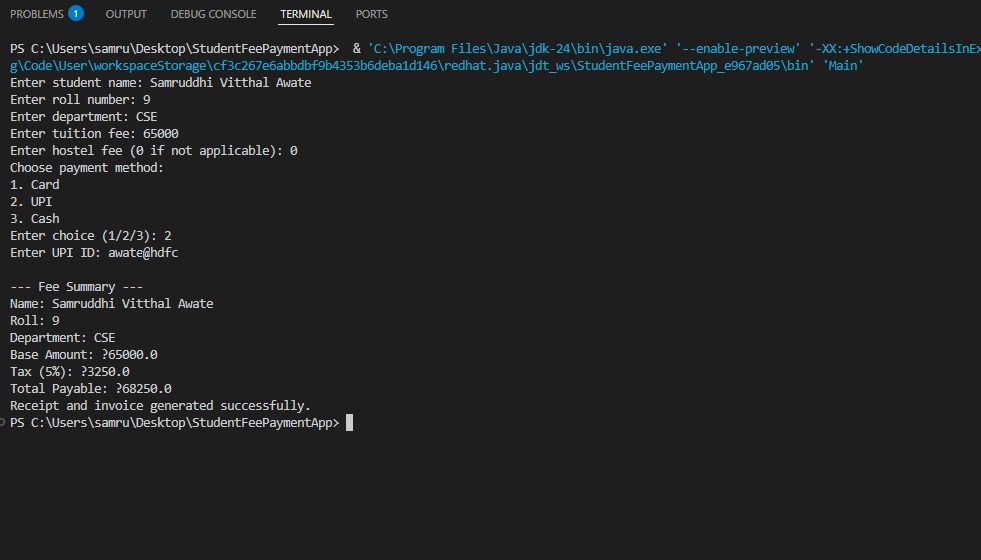


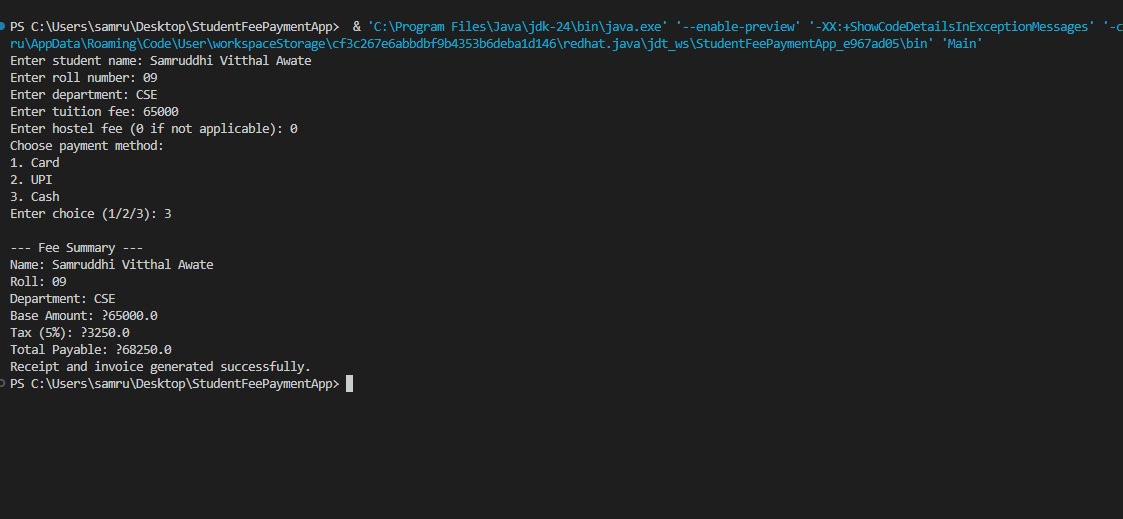












# 6. GIT-HUB LINK

<https://github.com/Sanika2805/Javaproject>

<https://github.com/samruddhiawate/Javaproject>

<https://github.com/Rajnandini16/Java_miniProject.git>

<https://github.com/Kalyani1196/JavaProject.git>

<https://github.com/Prasad5683/Java_Mini_Project>