

STELLAR TUITION ACADEMY

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF

BACHELOR OF COMPUTER APPLICATIONS

To

MARIAN COLLEGE KUTTIKKANAM (AUTONOMOUS)

Affiliated to

MAHATMA GANDHI UNIVERSITY, KOTTAYAM

By

ALPHY THERESA TOMI (Reg.No:22UBC208)

ANU EMMANUEL (Reg.No:22UBC215)

Guided By

Ms. NEENA K PIUS




DEPARTMENT OF COMPUTER APPLICATIONS
MARIAN COLLEGE KUTTIKKANAM (AUTONOMOUS)

PEERMADE – 685531

OCTOBER, 2024

DECLARATION

We, **ALPHY THERESA TOMI [Reg.no 22UBC208]**, **ANU EMMANUEL [Reg.no 22UBC215]** certify that the mini project titled **“STELLAR TUITION ACADEMY”** is an authentic work carried out by us at Marian College Kuttikkanam (Autonomous). The matter embodied in this project work has not been submitted elsewhere for the award of any degree or diploma to the best of our knowledge and belief.

Signature of the Students: 



Name of the Students: **ALPHY THERESA TOMI, ANU EMMANUEL**

Date: *30/10/2024*

BONAFIDE CERTIFICATE

This is to certify that this project work titled "STELLAR TUITION ACADEMY" is a bonafide record of work done by **Ms. ALPHY THERESA TOMI [Reg.no 22UBC208]**, **Ms. ANU EMMANUEL [Reg.no 22UBC215]** at Marian College Kuttikkanam (Autonomous) in partial fulfillment for the award of the **Degree of Bachelor of Computer Applications of Mahatma Gandhi University, Kottayam.**

This work has not been submitted elsewhere for the award of any other degree to the best of our knowledge.

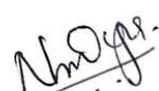
Head of the Department

Dr. Rajimol A

Dept. of Computer Application

Marian College Kuttikkanam

Peermade - 685531


Internal Guide

Ms. Neena K Pius


Dept. of Computer Application

Marian College Kuttikkanam

Peermade - 685531

Submitted for the Viva-Voce Examination held on: 30/10/2024

DEPARTMENT SEAL


Rosamma K S
EXTERNAL EXAMINER



ACKNOWLEDGEMENT

“Gratitude is a feeling which is more eloquent than words, more silent than silence.” In undertaking this project, we needed the direction, assistance and cooperation of various individuals and organizations, which is received in abundance with grace of God, without their unconstrained support, the project could not have been completed.

If words are considered as the symbol of approval and token of acknowledgement, then let the following words play the heralding role of expressing our gratitude.

We wish to acknowledge our sincere gratitude to our Manager, **Rev. Fr. Bobby Alex Mannamplackal** and **Prof. Dr. Ajimon George**, Principal **Marian College Kuttikkanam (Autonomous)**, for all their efforts and administration in educating us in their premier institution.

We extend our gratitude to **Dr. Rajimol A**, Head of the Department of Computer Application, who is a constant source of inspiration and whose advice helped us to complete this project successfully.

We express our deep sense of gratitude to our internal project guide, **Ms. Neena K Pius**, for her profound guidance for the successful completion of this project.

With great enthusiasm we express our gratitude to all the faculty members of Department of Computer Applications for their timely help and support.

Finally, We express our deep appreciation to all our friends and family members for the moral support and encouragement they have given to complete this project successfully.

Alphy Theresa Tomi, Anu Emmanuel

ABSTRACT

ABSTRACT

Our project is a Tuition Center Management System that helps to enhance the learning and teaching process for students and teachers. Various tasks can be performed on the site, including student enrollment, class scheduling, attendance tracking, and performance monitoring. With the login page students, teachers, and admins can access their controls.

Administrators do the overall coordination of the center. They can review admission applications and accept or reject them. Admins have the capability to verify the status of fee payments within the site. Admins are also in charge of maintaining and updating the class schedule. They can add and remove teachers on the basis of availability. Teachers are responsible for inputting student marks and attendance within the site. They can also access the class schedule through the website to stay informed. Students who wish to join the institution must enroll through the admission page to apply and on approval by admin the student gets accepted to the center and thereby can access all the functionalities provided by the website. Students can check their performance, attendance and class schedule in their dashboard. Students can also pay their fees online through the website.

It offers a user-friendly interface for administrators to manage staff and resources efficiently while giving parents and students real-time access to progress reports, class schedule, and fee notifications. The website also supports online payment processing for fees ensuring a seamless and efficient educational experience for all users. This website is developed using HTML, CSS, JavaScript, PHP and MySQL which collectively adds to the creation of a well developed dynamic website.

TABLE OF CONTENTS

TITLE	PAGE NUMBER
ABSTRACT	5
1. INTRODUCTION	9
1.1 ABOUT THE PROJECT	10
1.2 EXISTING SYSTEM	10
1.3 PROPOSED SYSTEM	11
2. SYSTEM ANALYSIS	12
2.1 PROBLEM DEFINITION	13
2.2 FEASIBILITY ANALYSIS	13
2.3 RECOMMENDED IMPLEMENTATIONS	14
3. SOFTWARE REQUIREMENT SPECIFICATION	16
3.1 INTRODUCTION	17
3.2 PURPOSE	17
3.3 SCOPES	17
3.4 TECHNICAL OVERVIEW	18
3.5 STATED REQUIREMENTS	18
3.6 EXTERNAL INTERFACE REQUIREMENTS	20
4. SYSTEM DESIGN	22
4.1 INTRODUCTION	23
4.2 DESIGN METHODOLOGY	24
4.3 SYSTEM ARCHITECTURE AND PROCESS FLOW	28
4.4 MODULE DETAILS	40
4.5 PERFORMANCE ISSUES	40
4.6 SECURITY ISSUES	40
4.7 TABLE DESIGN	41
5. CODING	47

5.1	SELECTION OF SOFTWARE	48
5.2	CODING PHASE	49
6.	TESTING	50
6.1	INTRODUCTION	51
6.2	TESTING	51
6.3	TESTING METHODS	52
6.4	IMPLEMENTATION	55
7.	MAINTENANCE AND ENHANCEMENT	57
7.1	MAINTENANCE	58
7.2	ENHANCEMENT	58
8.	CONCLUSION	59
9.	BIBLIOGRAPHY	61
10.	APPENDIX	63

INTRODUCTION

1. INTRODUCTION

1.1 ABOUT THE PROJECT

The Tuition Center Management System website is a digital platform designed to streamline the educational process for students, teachers, and administrators. This system enhances the learning and teaching experience by providing various functionalities, including student enrollment, class scheduling, attendance tracking, and performance monitoring. With a user-friendly login page, students, teachers, and administrators can access their respective controls easily.

1.1.1 THE PURPOSE AND SCOPE

The primary purpose of the Tuition Center Management System is to establish an efficient online management system for tuition centers. This project aims to facilitate seamless interactions among students, teachers, and administrators while enhancing the overall educational experience. The scope of the project includes:

- **Student Enrollment:** Allowing students to apply for admission through an online portal, with the ability to track their application status.
- **Class Scheduling:** Enabling administrators to maintain and update class schedules, ensuring all users are informed of any changes.
- **Attendance Tracking:** Providing teachers with tools to input and monitor student attendance.
- **Performance Monitoring:** Allowing teachers to input and update student marks, enabling students and parents to track academic progress.
- **User Account Management:** Providing students and teachers with personal dashboards to access the respective functionalities.
- **Online Fee Payment:** Facilitating secure online transactions for fee payments, streamlining the payment process for students and parents.

1.2 EXISTING SYSTEM

In the current educational environment, managing a tuition center often involves manual processes that can be time-consuming and prone to errors. Traditional methods of student enrollment, class scheduling, and performance tracking are often fragmented, leading to inefficiencies and communication gaps between students, teachers, and administrators.

1.3 PROPOSED SYSTEM

The proposed Tuition Center Management System aims to significantly enhance the management of tuition centers by addressing the inefficiencies of traditional methods. Unlike existing systems that rely on manual processes, this system automates key administrative tasks, such as student enrollment, making it easier for students to apply and track their applications.

Real-time access to performance reports, attendance records, and class schedules improves communication and transparency for students and parents, eliminating the delays associated with manual record-keeping. Additionally, teachers can easily input attendance and grades, ensuring accurate and up-to-date information.

The inclusion of an online fee payment system simplifies financial transactions, reducing the administrative burden of cash handling and manual payment tracking. Overall, the proposed system offers a streamlined, efficient, and user-friendly experience that empowers all users and fosters a more effective educational environment.

SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

2.1 PROBLEM DEFINITION

The current methods for managing tuition centers often rely on manual processes that are slow and inefficient. Although some digital tools are used, they are not well organized. Most financial records are still kept on paper, making it hard to track payments and manage student information effectively. To solve these problems, the proposed Tuition Center Management System aims to streamline operations and improve overall efficiency.

2.1.1 ADVANTAGES OF PROPOSED SYSTEM

The proposed system has several advantages over the current methods. It reduces duplication of data through different checks, which helps ensure that information is accurate. The system allows for quick access to information and faster processing, which makes it easier for administrators to do their jobs. Unlike traditional systems, this automated platform saves time on administrative tasks. Additionally, it reduces the need for paper, promoting a more environmentally friendly approach while providing reliable information.

2.2 FEASIBILITY ANALYSIS

Feasibility study is a test of a system proposal according to its workability, ability to meet user needs and effective use of resources. The objective of feasibility is not to solve the problem but to acquire a sense of its scope. The main aim of the feasibility study is to test the technical, social and economic feasibility of the system. The feasibility study can be classified into the following categories:

- Operational Feasibility
- Technical Feasibility
- Economic Feasibility

2.2.1 OPERATIONAL FEASIBILITY

The proposed system is designed to be user-friendly and faster than manual methods. This means less work for administrators, allowing them to manage tasks with less effort. The quick processing speed saves a lot of time, making the project practical to implement.

2.2.2 TECHNICAL FEASIBILITY

Technical feasibility deals with hardware as well as software requirements and to what extent it can support the proposed system. The hardware required is an android phone and software is Android Studio. If the necessary requirements are made available with the system, then the proposed system is said to be technically feasible.

2.2.3 ECONOMIC FEASIBILITY

Economic feasibility is important because the proposed system aims to save time and reduce errors in data entry. While there may be initial costs for software and development, the time saved and the reduction in mistakes will make up for these expenses. The system includes checks to minimize errors, which helps save money in the long run. Thus, the proposed system is economically feasible and offers significant benefits.

2.3 RECOMMENDED IMPLEMENTATIONS

Two principle sources of data are:

1. Written documents
2. Data from the persons, who are involved in the operation of the system under study.

The different fact-finding techniques are:

3. Questionnaires
4. Personal Interviews
5. Observations

Questionnaires

Questionnaires are best methods to probe data out of the customers. In this case, questionnaires were not used for data collection as the administration was small in number and they could be asked questions in a more effective interview.

Personal Interviews

Personal interviews were the main method for gathering information. The owner and staff members were asked about their experiences with the current system, including how they handle

administrative tasks and record-keeping. Their feedback was very helpful and influenced the design of the new system.

Observations

Watching how things work in the tuition center helped understand the daily operations. This observation showed how much paperwork and manual work is involved, which guided the creation of a stronger database model for the new web application.

SOFTWARE REQUIREMENT SPECIFICATION

3.SOFTWARE REQUIREMENT SPECIFICATION

3.1 INTRODUCTION

Requirements specification is the starting step for the development activities. It is currently one of the weak areas of software engineering. During requirement specification, the goal is to produce a document of the client's requirements. This document forms the basis of development and software validation. The basic reason for the difficulty in software requirements specification comes from the fact that there are three interested parties- the client, the end users and the software developer.

3.2 PURPOSE

The origin of most software systems is in the need of a client, who either wants to automate an existing manual system or desires a new software system. The software system itself is created by the developer. Finally, the completed system will be used by the end users. Thus, there are three major parties interested in a new system: the client, the users and the developer. A basic purpose of software requirements specification is to bridge the communication gap. SRS is the medium through which the client and user needs are accurately specified. Indeed, SRS forms the basis of software development. A good SRS should satisfy all the parties, something very hard to achieve, and involves trade-offs and persuasion. Another important purpose of developing an SRS is helping the clients understand their own needs. Advantages are: • An SRS establishes the basis for agreement between the client and the supplier on what the software product will do • An SRS provides a reference for validation of the final product • A high-quality SRS is a prerequisite to high-quality software. • A high-quality SRS reduces the development cost.

3.3 SCOPES

3.3.1 SYSTEM STATEMENT OF SCOPE

Tuition centre management system was developed to manage and handle tuition centre activities. The system provides functions like student enrolment, staff addition, performance viewing, class scheduling etc.

3.4 TECHNICAL OVERVIEW

3.4.1 USER CHARACTERISTICS

The system can be accessed by three types of users, the student, teacher and admin. The students and teachers have only access to their respective dashboards. The admins have access to all the features in the site. The admins and teachers are invisible to the students and admins are invisible to everyone.

3.5 STATED REQUIREMENTS

3.5.1 GENERAL REQUIREMENTS

The system has functional modules divided by student, admin, and teacher

1. Login

- Only registered users like the student, teacher and admin can login and make use of the services provided by the website
- To login the user needs to enter their role, user id and password
- Password can also contain both upper case and lower case alphabetic, numbers and special character
- Admins will be directed to the admin dashboard and teachers to the teacher dashboard
- Students will be directed to the homepage

2. Admission page

Students who want to be a part of the centre to submit an application form in order to gain admission. The admission form includes:

- Name of the applicant
- Contact Number
- Email
- Guardian's phone number
- Date of birth
- School
- Class

- Subjects selected

3. Admin Panel

The admin dashboard provides many features for the admin like addition and dismissal of the teachers, viewing and accepting/rejecting student applications and managing the schedule.

4. Teacher Panel

The teachers can use functions like entering the student's attendance and marks through their dashboard.

5. Student Panel

The students can view their attendance and see their performance in the subjects throughout the semesters. They can also pay their fees

6. Home page

- The main page of the website where the users are taken to first
- If a student's wants to view their schedule or personal details they will have to login with their credentials.
- Only registered users can avail the services of the website

3.5.2 INPUTS

The Tuition Centre Website will collect the information about students details and store it.

3.5.3 PROCESSING

- All types of validation for the data entry are carried out
- View applications, accepting/rejecting applications, adding/removing teachers, managing the schedule are managed by admin
- Marking attendance, uploading student performances are all done by teachers
- Payment

3.5.4 OUTPUTS

This system produces the following outputs:

□ **Student Information:**

- Detailed profiles including name, contact information, guardian's contact information, date of birth, school, class, and selected subjects.
- Attendance records for individual students.
- Performance reports and grades for subjects.
- Fee payment status.

□ **Teacher Information:**

- Detailed profiles including name and subject specialization.
- Class schedules and assigned subjects.
- Attendance records for students in their classes.
- Marks and performance assessments for students.

□ **Admin Information:**

- List of teachers with their details and status.
- Student application statuses (accepted/rejected).
- Class schedules and management reports.

3.6 EXTERNAL INTERFACE REQUIREMENTS

3.6.1 USER INTERFACES

All user interfaces will be GUI interfaces. Interfaces are designed to use with ease and without any confusion. The user interface shall have a pleasing appearance and high functionality.

- Suitable design and pleasing colors are selected to design the window page to make the users comfortable to operate the software.
- Component like textboxes, combo boxes and buttons are selected to make it easy to fill with appropriate data.

3.6.2 HARDWARE INTERFACES

The System needs a computer or any other smart phones or devices with network availability to browse into the web application. No other external hardware is required.

HARDWARE SPECIFICATION

Processor	:	Intel Pentium or higher
RAM	:	256 MB or higher
Hard disk drive	:	100MB is required on disk
Keyboard	:	Standard QWERTY keyboard

IMPLEMENTATION SPECIFICATION

Operating system : Windows OS

3.6.3 SOFTWARE INTERFACES

SOFTWARE SPECIFICATION

Operating System	:	Windows 11
DBMS	:	MySQL
Tool Used	:	PHP

SYSTEM DESIGN

4. SYSTEM DESIGN

4.1 INTRODUCTION

The purpose of design phase is to plan a solution of the problem specified by the analysis phase. This phase is the first step in moving from the problem domain to solution domain.

System design describes the desired features and operation in detail, including screen layouts, business rules, process diagrams, pseudo code and other documentation.

In this phase, the software's overall structure and its nuances are defined. In terms of the client/server technology, the number of tiers needed for the package architecture, input design, output design, the database design, the data structure design etc. are all defined in this phase. Analysis and design are very crucial in the whole development cycle. Any glitch in the design phase could be very expensive to solve in the later stage of the software development. So much care is taken during this phase.

The logical system of the product and the physical characteristics of the system are designed during this phase. The operating environment is established, major resources. Everything requiring user input or approval must be documented and reviewed by the user. The physical characteristics of the system are specified and a detailed design is prepared.

The subsystem identified during design are used to create a detailed structure of the system. Each subsystem is partitioned into one or more design units or modules. Detailed logic specifications are prepared for each software module. The logic of the module is usually specified in a high-level design description language, which is independent of the target language in which the software will eventually be implemented.

A good design must consider:

- Prompt: should be simple and clear to intuitively lead the user to an expected outcome.
- Memory load: Studies show that, under normal circumstances, users have a short-term memory of approximately six words. Ideally, the number of choices of users to select should be four or less. Otherwise, callers become confused and forget the choices presented to them.
- Service reaches ability: It is not pleasant for a person to go through a large number of steps before he reaches a service. Users start to get impatient with more than five steps. Minimize the number of steps a caller must take to reduce frustration

- **Navigation:** Provide a way to navigate back and forth between various dialogue steps. The user should be able to go to different parts of the dialog easily.
- **Phonetic similarity:** Provide a clear set of choices for user to select. Avoid choices with similar pronunciation.
- **Error handling:** Humans make mistakes. Graceful error handling decreases dependency on operators.
- **User update:** Let the user know what is going on and keep him engaged.

For the general design one or more potential design are propose and broadly sketched. Then these alternatives are presented to the users, who choose the design that best suits their requirements while staying with in the project constraints.

Detailed design stage is specification for the user interface, database, programs, hardware, and training and system documentation. Several structured techniques are used during the design phase. To design the software components, the designer transforms the automated processes in the physical data flow diagram into a program structure chart, which decomposes software processes into detailed modules and shows control path between modules

4.2 DESIGN METHODOLOGY

4.2.1 INPUT DESIGN

In the input design, the user oriented inputs are converted into computer recognizable format. The collection of input data is the most expensive part of the system in terms of equipment used, time and number of users involved. Input design is the processes of converting user oriented inputs to a computer based format. The goal of designing input data is to make data entry as easy, logical and free from errors as possible.

Input design is the link between the information system and the users and the skip necessary to put transaction data in to a usable form for processing. Instructing the computer to read data from a written printed document can activate the activity of putting data into the computer for processing or it can occur by keying data directly into the system. The design of input focusing on controlling the amount of input required, controlling the errors, avoid delay extra steps, and keeping the process simple.

System analysis decides the following input design details:

- What data to input
- What medium is to use
- How the data is arranged and coded.
- Data items and transaction needing validation to detect error occurs

Activities performed as part of input design are:

- Data recording
- Data verification
- Data conversion
- Data validation
- Data correction

4.2.2 OUTPUT DESIGN

Output design is a process that involves designing necessary outputs that have to be used by various users according to requirements. Designing computer should proceed in well thought out manner. The term output means any information produced by the information system whether printed or displayed. When analyst design computer output they identified the specific output that is needed to meet the requirement.

Computer is the most important source of information to the users. Efficient intelligent output design should improve the system relationship with the user and help in decision making. When designing the output, system analyst must accomplish the following:

- Determine the information to present
- Decide whether to display, print, speak the information and select the output medium
- Arrange the information acceptable format

The output design is the key to the success of any system. Output is the key between the user and the sensor. The output must be concerned to the system's working, as it should. Output design consists of

displaying specification and procedures as data presentation. User is never left with the confusion as to what is happening without appropriate error and acknowledges message being received.

4.2.3 CODE DESIGN

The coding step is a process that transform design into programming language. It translates a detail design representation of software into a programming language realization. The code design should be done in such a way that the lines of code used in the software should be minimum for the specified design of the solution. The coding should be in modularized manner.

When code is placed in a module, one may hide it from view and give those executable statements a name (the name of the function or procedure). Information hiding is a good thing when it enhances the understanding of a program by letting to focus on a higher level of abstraction. Information hiding is a bad thing when it obscures one's understanding of a program. This usually happens when the name for the module is not chosen accurately.

In this software, the modularized approach is used. Different functions are created for different operations. The name of the module is chosen such a way that it describes what it does, i.e. the name gives the action performed by the module

4.2.4 DATABASE DESIGN

The details about the relevant data that came into lay in the system are identified according to the relationship the tables are designed by following the standard database design methods. The dative for each data in the table is defined. For optimum design of database to have better response time, to have data integrity, to avoid the redundancy and for security of the database tables created and analyzed.

A database system can be defined as a representation of an information system in a computer. The general theme behind a database is to handle information as an integrated whole. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive and flexible for the user. In database design, several specific objectives are considered:

- Controlled redundancy
- Ease of learning and use
- Data independence

- More information at low cost
- Accuracy and integrity
- Recovery from failure
- Privacy and security
- Performance

The schema is the view that helps us the DBMS decide what data in storage it should act upon as requested by the web application program. The subschema is concerned with a relatively small part of scheme. In database design, several views of data must be considered along with the persons who use them. The logical view is what the data look like, regardless of how they stored. The physical view is the way data exists in physical storage. It deals with how data are stored, accessed or related to other data in storage. The logical view are the users view the programmer's view and the overall logical view, called a schema.

A **primary key** is a special relational database table column (or combination of columns) designated to uniquely identify all table records. A **primary key's** main features are: It must contain a unique value for each row of data. It cannot contain null values. There is different primary key for varies tables. For eg: in users_tbl , id is the primary key, in tbl_payment ,id is the primary key and so on.

Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

This project has used a main database having different tables, based on which the operations can perform well.

4.3 SYSTEM ARCHITECTURE AND PROCESS FLOW

4.3.1 USE CASE DIAGRAM

- USE CASE FOR LOGIN

Use Case Name:	Login
Created By:	User
Date created:	28/06/2024
Description:	Allows users to login to their existing accounts
Primary User:	Admin, Teacher, Student
Secondary user:	None
Precondition:	The user is on the tuition center homepage and already has an account on the site
Post Condition:	The user successfully logs into their account
Main Flow:	<ol style="list-style-type: none">1. The user selects 'Login' option2. The system prompts for necessary information (id and password)3. The user provides the necessary details and submits the form4. The user enters into their respective dashboard.

- USE CASE FOR ADMISSION APPLICATION

Use Case Name:	Admission Application
Created By:	User
Date created:	28/06/2024
Description:	Allows users to apply for admission and check its status
Primary User:	User
Secondary user:	None
Precondition:	User navigates to the admission page
Post Condition:	The user successfully applies for the admission and can check its status
Main Flow:	<ol style="list-style-type: none"> 1. The user navigates to admission page 2. The user enters the necessary information 3. The user submits the form 4. Admission application has been submitted successfully and gets their application id 5. User enters their email id and application id to check the status of their application

- USE CASE FOR STUDENT ADMINISTRATION

Use Case Name:	Student Administration
Created by:	User
Date Created:	18/07/2024
Description:	Allows Admin to check details of the students and accept/reject new applications
Primary Actor:	Admin
Secondary Actor:	None
Precondition:	The admin has logged in
Post condition:	The admin has successfully accepted/rejected the applications
Main Flow:	<ol style="list-style-type: none"> 1. The admin navigates their dashboard 2. Admin selects student page 3. Admin can view new applicant's details and confirms/rejects their application 4. The details of the confirmed applicant are entered to the student table 5. Admin may view enrolled student's details and remove students if necessary.

- USE CASE FOR TEACHER ADMINISTRATION

Use Case Name:	Teacher Administration
Created by:	User
Date Created:	31/07/2024
Description:	Allows Admin to enter details of new teacher and add them to the database and also view current teacher and remove them
Primary Actor:	Admin
Secondary Actor:	None
Precondition:	The admin has logged in
Post condition:	The admin has successfully added/deleted teachers
Main Flow:	<ol style="list-style-type: none"> 1. The admin navigates their dashboard 2. Admin selects teacher page 3. If they want to add new teachers: <ol style="list-style-type: none"> i. The admin enters the details into the form ii. The admin clicks the add button and the details of the new teacher is entered into the staff table 4. The admin clicks on the delete button to remove teachers

- USE CASE FOR CLASS SCHEDULE

Use Case Name:	Class schedule
Created by:	User
Date Created:	19/07/2024
Description:	Allows admin to add and update schedule. Teachers and Students can view the schedule.
Primary Actor:	Admin
Secondary Actor:	Teacher, Student
Precondition:	The Admin is logged in
Post condition:	Class schedule is displayed
Main Flow:	<ol style="list-style-type: none"> 1. The admin dashboard page of the website is loaded. 2. The class schedule page is selected. 3. Admin creates or updates schedules 4. Teachers and students view the schedule

- USE CASE FOR STUDENT OVERVIEW

Use Case Name:	Student Overview
Created by:	User
Date Created:	23/08/2024
Description:	Allows teachers to enter marks scored by students and attendance of each day and students to view them
Primary Actor:	Teacher
Secondary Actor:	Student
Precondition:	The teacher is logged in
Post condition:	Marks and attendance are successfully entered.
Main Flow:	<ol style="list-style-type: none"> 1. The teacher logs in and selects the student overview page. 2. The teacher then may enter marks scored by each student in the semester. 3. The teacher can also enter attendance of each student on a specific day. 4. The student, on selecting the marks page on their dashboard, can view their marks as entered by the teacher. 5. On selecting the attendance page, the students can view their attendance status.

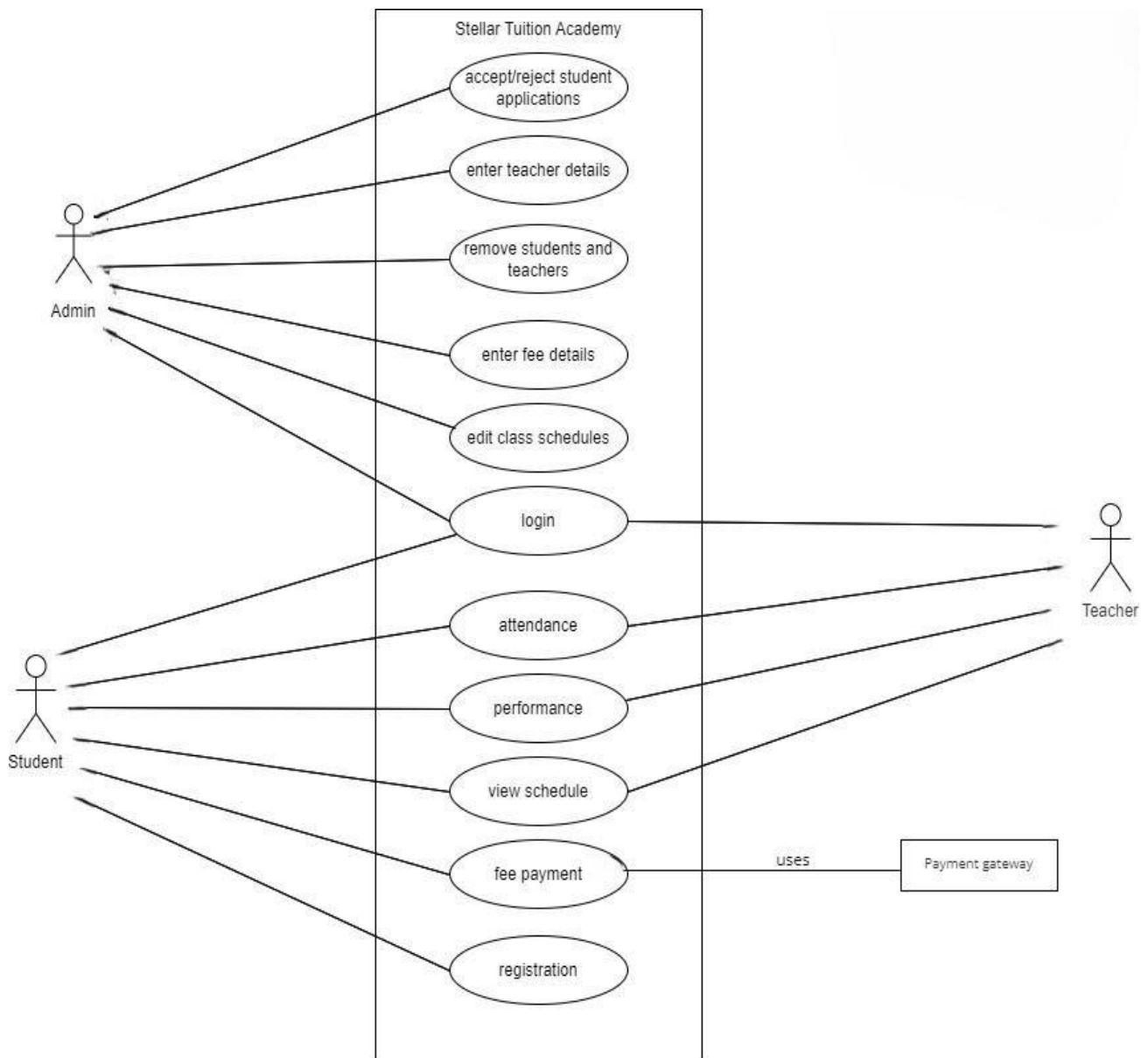
- USE CASE FOR TUITION FEES

Use Case Name:	Tuition Fees
Created by:	User
Date Created:	15/07/2024
Description:	Allows admin to enter the fees details and payment status of students and students to view it
Primary Actor:	Admin
Secondary Actor:	Student
Precondition:	The admin is logged in
Post condition:	Fees details are successfully entered.
Main Flow:	<ol style="list-style-type: none"> 1. The admin logs in and selects the student overview page. 2. The admin then may enter fees details or view/update fees paid by each student in the semester. 3. The student, on selecting the fees page on their dashboard, can view their fee status as entered by the teacher.

- USE CASE FOR FEE PAYMENT

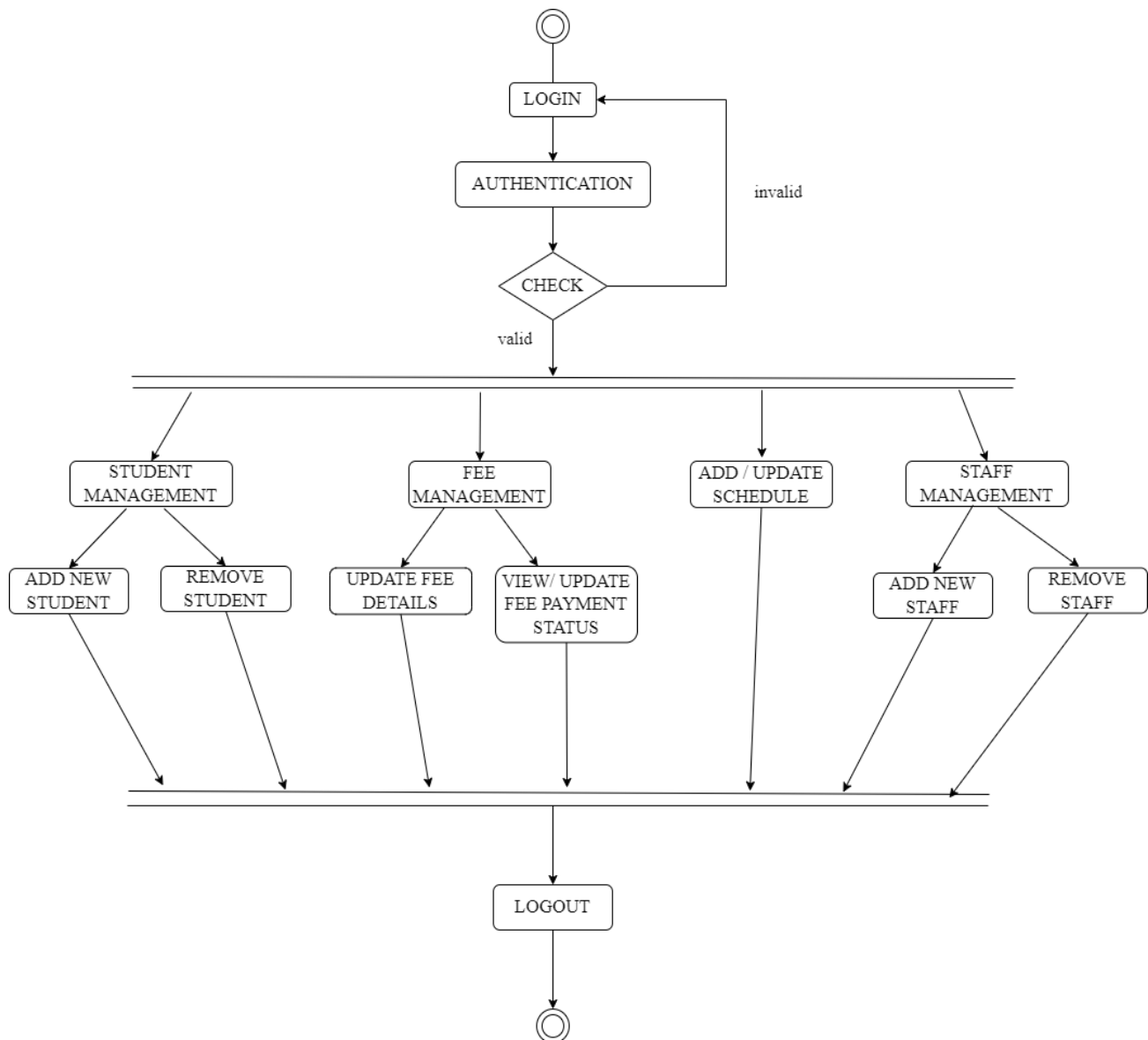
Use Case Name:	Tuition Fees
Created by:	User
Date Created:	18/09/2024
Description:	Allows students to pay fees online and admins to view the payment dashboard
Primary Actor:	Student
Secondary Actor:	Admin
Precondition:	The student is logged in
Post condition:	Fee is successfully paid
Main Flow:	<ol style="list-style-type: none"> 1 The student logs in and selects the fee payment page. 2 The student selects the fees to pay and proceed with the fee payment. 3 After the payment is made, it can be viewed in the fee payment status page of Admin dashboard.

USE CASE DIAGRAM

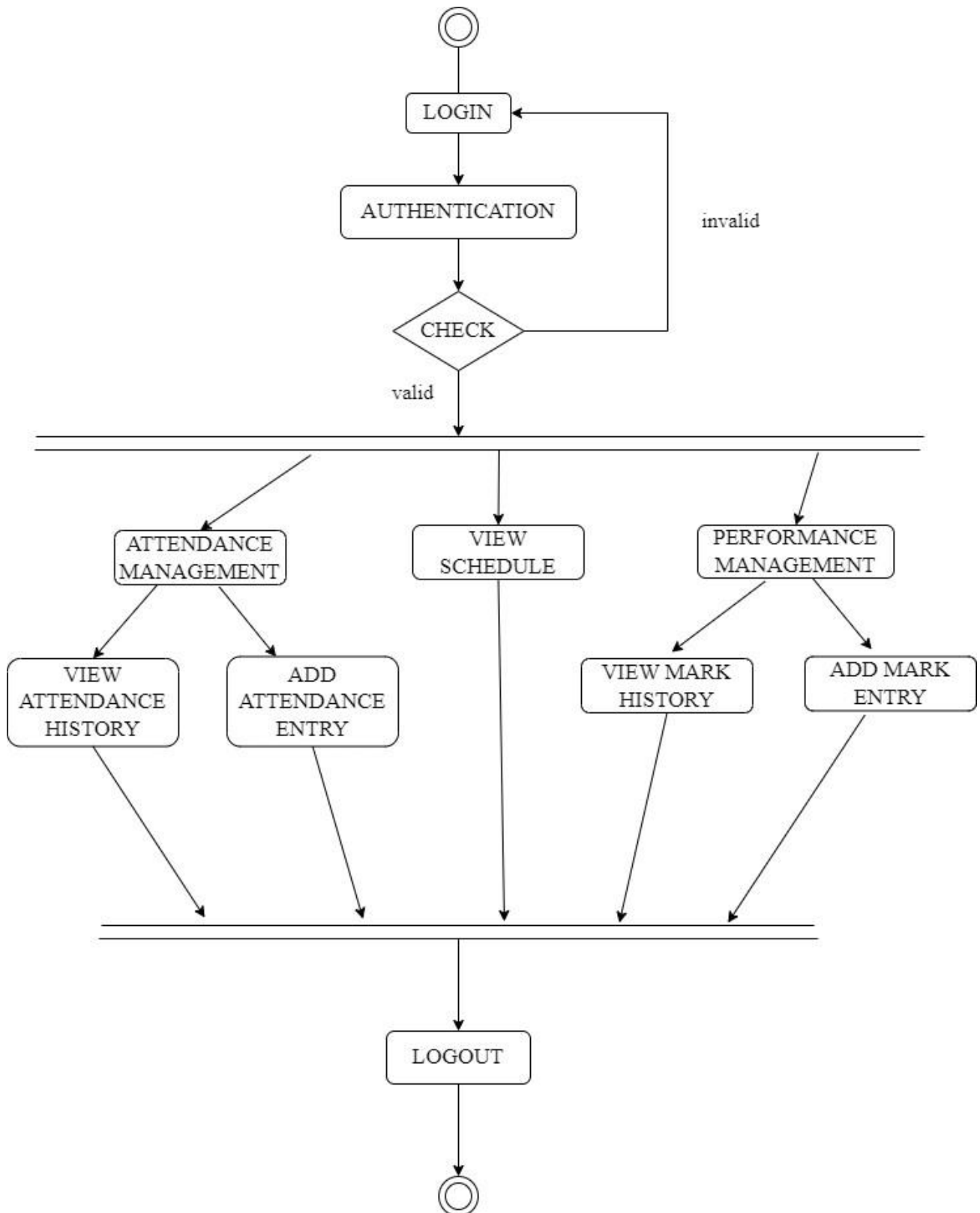


4.3.2 ACTIVITY DIAGRAMS

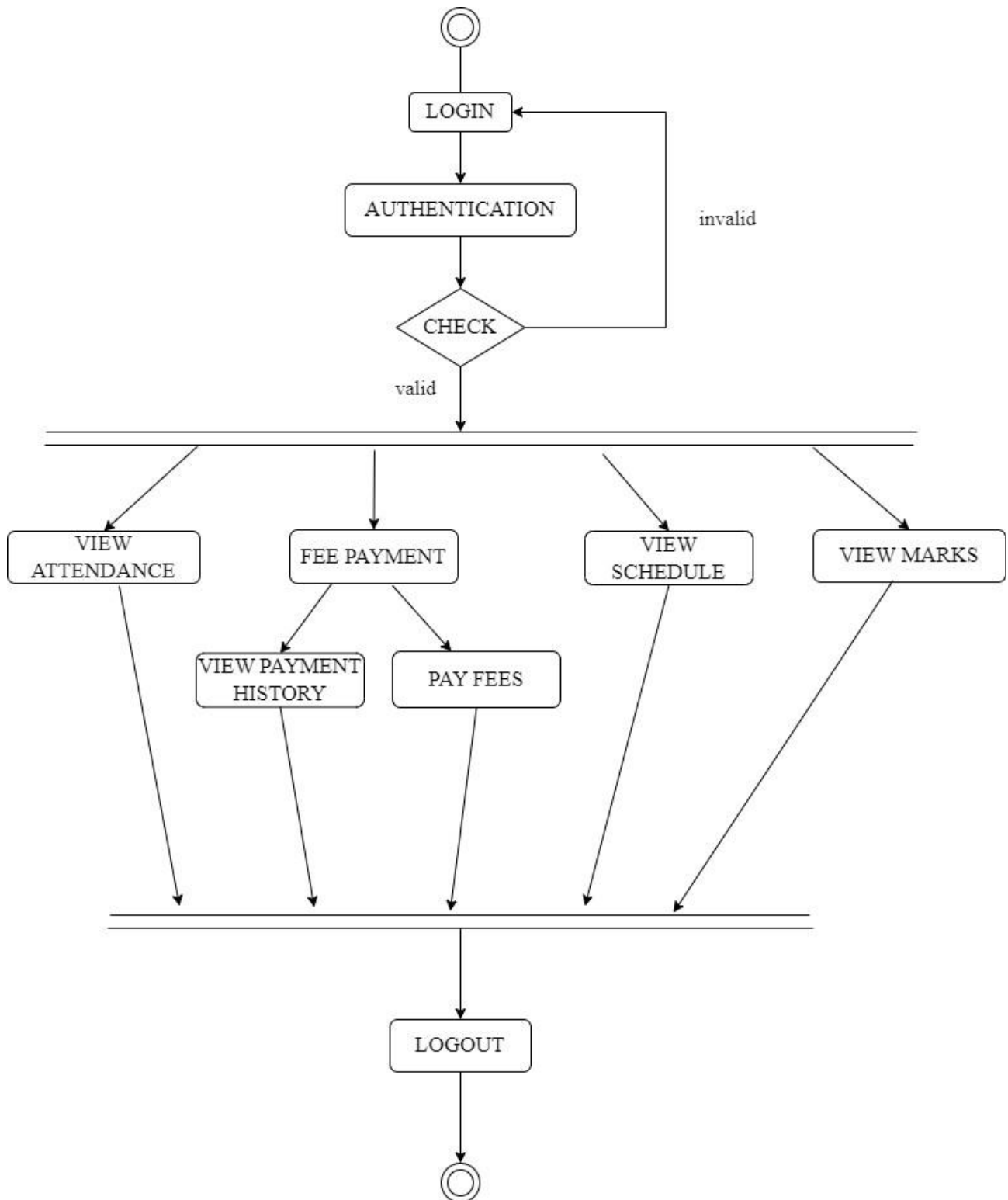
ACTIVITY DIAGRAM FOR ADMIN



ACTIVITY DIAGRAM FOR TEACHER



ACTIVITY DIAGRAM FOR STUDENT



4.4 MODULE DETAILS

This software has 6 module and different parts. They are:

- Login
- Admission Application
- Admin Dashboard
- Teacher Dashboard
- Student Dashboard
- Home page

The login module provides provision for the registered users to login to the site and can avail the services.

The Admission application module is used for new interested students to apply to enroll in any subject provided by the tuition center.

The admin tasks and several processes undertaken by the admin come under the admin dashboard module.

The teacher tasks and several processes undertaken by the teacher are come under the teacher dashboard module.

The student tasks and several processes undertaken by the student come under the student dashboard module.

The different modules and their functions are explained in detail in the above system requirements.

4.5 PERFORMANCE ISSUES

This system should have at least 4GB of RAM. The OS must be Windows 8 or higher.

4.6 SECURITY ISSUES

Only authorized access is permitted to this software. The login is protected by username and password.

4.7 TABLE DESIGN

1. Table name: tbl_stud

Sl No	Field Name	Data Type	Constraint	Description
1	S_ID	Int(20)	Primary key, Not null, Auto-increment	Student ID
2	Fname	Varchar(40)	Not null	First name of the student
3	Lname	Varchar(40)	Not null	Last name of the student
4	Ph_no	Char(10)	Not null	Phone number of the student
5	Email	Varchar(30)	Not null	E-mail of the student
6	Password	Char(8)	Not null, unique	Password to login in the website
7	Guar_ph	Char(10)	Not null	Phone number of the student's guardian
8	DOB	Date	Not null	Date of birth of the student
9	Adm_status	Varchar(30)	Not null	Admission status (Applied/Admitted)

2. Table name: tbl_staff

Sl No	Field Name	Data Type	Constraint	Description
1	ST_ID	Int(20)	Primary key, Not null, Auto-increment	Admin/Teacher ID
2	Fname	Varchar(40)	Not null	First name of the staff
3	Lname	Varchar(40)	Not null	Last name of the staff
3	Ph_no	Char(10)	Not null	Phone number of the staff
4	Email	Varchar(30)	Not null	Email ID of the staff
5	Password	Char(8)	Not null, unique	Password to login in the website
6	Role	Varchar(10)	Not null	Role (Admin/Teacher)

3. Table name: tbl_subject

SI No	Field Name	Data Type	Constraint	Description
1	SUB_ID	Int(20)	Primary key, Not null, Auto-increment	Subject ID
2	Name	Varchar(20)	Not null	Name of the subject
3	Status	Int(2)	Not null	Status of the subject

4. Table name: tbl_allocation

SI No	Field Name	Data Type	Constraint	Description
1	SUBT_ID	Int(20)	Primary key, Not null, Auto-increment	Allocation ID
2	ST_ID	Int(20)	Not null, Foreign key	Teacher ID from Staff table
3	SUB_ID	Int(20)	Not null, Foreign Key	Subject ID from Subjects table

5. Table name: tbl_enrollment

SI No	Field Name	Data Type	Constraint	Description
1	E_ID	Int(20)	Primary key, Not null, Auto-increment	Enrollment ID
2	S_ID	Int(20)	Not null, Foreign key	Student ID from Student table
3	SUB_ID	Int(20)	Not null, Foreign Key	Subject ID from Subjects table

6. Table name: tbl_schedule

SI No	Field Name	Data Type	Constraint	Description
1	SCH_ID	Int(20)	Primary key, Not null, Auto-increment	Schedule ID
2	SUBT_ID	Int(20)	Not null, Foreign Key	Allocation ID from Allocation table
3	Day	Varchar(10)	Not null	Day of the class
4	Time	Time	Not null	Timing of the class

1.

7. Table name: tbl_performance

SI No	Field Name	Data Type	Constraint	Description
1	P_ID	Int(20)	Primary key, Not null, Auto-increment	Performance ID
2	E_ID	Int(20)	Not null, Foreign Key	Enrollment ID from enrollment table
3	S1	Int(2)		Marks scored in Sem 1
4	S2	Int(2)		Marks scored in Sem 2
5	S3	Int(2)		Marks scored in Sem 3

8. Table name: tbl_fees

Sl No	Field Name	Data Type	Constraint	Description
1	F_ID	Int(20)	Primary key, Not null, Auto-increment	Fee ID
2	Type	Varchar(50)	Not null	Fee type (Admission fee, sem fees etc)
3	Amount	Int(10)	Not null	Fee amount
4	Due_date	Date	Not null	Last date to pay the fees

9. Table name: tbl_feepayment

Sl No	Field Name	Data Type	Constraint	Description
1	FP_ID	Int(20)	Primary key, Not null, Auto-increment	Fee payment ID
2	S_ID	Int(20)	Not null, Foreign Key	Student ID from student table
3	F_ID	Int(20)	Not null, Foreign key	Fee ID from Fees table
4	Payment_date	Date	Not null	Date the payment was made

10. Table name: tbl_attendance

SI No	Field Name	Data Type	Constraint	Description
1	ATT_ID	Int(20)	Primary key, Not null, Auto-increment	Attendance ID
2	Att_Date	Int(20)	Not null	Date attendance taken
3	E_ID	Int(10)	Not null, Foreign key	Enrollment ID from enrollment table
4	Status	Int(10)	Not null	Attendance status (Present/Absent)

CODING

5. CODING

5.1 INTRODUCTION

Coding section is where the magic happens. All the planning and the designing done in the previous sections come to life in this section. After this section can only the programmer enjoys the result of his/her hard work when he/she runs the program for the first time.

5.2 SELECTION OF SOFTWARE

PHP

PHP, an acronym for Hypertext Preprocessor, is a versatile server-side scripting language that falls under the broader category of software development. It is widely recognized for its pivotal role in web development and boasts several essential features that make it a preferred choice for building dynamic websites and web applications. Here are some of its key features:

- Open Source
- Database Integration
- Embedded in HTML
- Cross-Platform Compatibility
- Security

MYSQL

MySQL is an open-source relational database system, widely used for web development tasks like data storage, manipulation, and retrieval. It seamlessly integrates into web applications, eliminating the need for complex setup. MySQL is embedded within web development environments, making administrative tasks effortless. It operates as an SQL-based database, storing data in text files on the device. Unlike systems like JDBC, MySQL simplifies data access with its broad range of relational database features. It's features are

- Zero configuration
- Server less
- Stable cross platform database file

- Less memory
- Self-contained
- Transactional

5.3 CODING PHASE

The goal of the coding or programming phase is to translate the design of the system produced during the design phase into code in a given programming language, which can be executed by a computer and that performs the computation specified by the design.

The coding phase affects both testing and maintenance profoundly. The coding phase does not affect the structure of the system; it has great impact on the internal structure of modules, which affects the testability of the system.

The goal of the coding phase is to produce clear simple programs. The aim is not to reduce the coding effect, but to program in a manner so that testing and maintenance costs are reduced.

Programs should not be constructed so that they are easy to write; they should be easy to read and understand. Reading programs is a much more common activity than writing programs. Hence, the goal of the coding phase is to produce simple programs that are clear to understand and modify.

5.3.1 CODING STANDARDS

The standard used in the development of the system is Microsoft Programming standards. It includes naming conventions of variables, constants and objects, standardized formats for labelling and commenting code, spacing, formatting and indenting.

Naming Conventions

The controls are prefixed to indicate their functions. The frames are prefixed with frm, textboxes are prefixed with txt, command buttons with cmd, label boxes with lbl, list boxes with lst, combo boxes with cmb, Date Time Pickers with DTP, Grid with grid and so on.

Labels and Comments

The functions of each control are labelled clearly in the GUI. The code also includes comments so that other developers using the source code in future might understand the module functions better.

TESTING & IMPLEMENTATION

6. TESTING

6.1 INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specifications design and coding. Testing presents an interesting anomaly for the software. Testing is a quality measure process, which reveals the errors in the program. During testing, the program is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected. Testing plays a very critical role in determining the reliability and efficiency of the software and it is a very important stage in software development.

6.2 TESTING

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based systems. Although each test has a different purpose, all work to verify that all system elements have been properly integrated and perform allocated functions.

System testing is done in order to ensure that the system developed doesn't fail at any point. Before implementations, the system is tested with experimental data to ensure that it will meet the specified requirements, special tests data are input for processing and results examined.

6.2.1 TEST PLAN

- **Preparation of test data**

Taking various kinds of test data does the testing. Preparation of test data plays a vital role in the system testing. After preparing, the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and correction are also noted for future use. Two kinds of test data were collected and used:

- **Using live test data**

Live test is those that are actually extracted from organization files. After a system is partially constructed, programmers or analyst often ask users to key in a set of data from their normal activities. Then, the system person uses this data as a way to partially test the system. In order

instance, programmers or analysts extract a set of live data from the files and have entered themselves

Using artificial test data

Artificial test data are created solely for test purpose, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information system department, make possible the testing of all login and control paths through the program.

The most effective test program uses artificial test data generated by person other than those who wrote the program.

In this project invalid data was entered to test whether the program would break or not. These invalid data entries were randomly generated using random people. Many people were given the software for testing the program. They use gibberish values to test if every validation holds strong.

6.3 TESTING METHODS

Testing is generally done at two levels-testing of individual modules and testing of the entire system. During system testing, the system is used experimentally to ensure that the software does not fail that is, that it will run according to its specifications and the results examined. A limited number of uses may be allowed to use the system so analysis can see whether they use it in unforeseen ways. It is preferable to discover any surprise before the organization implements the system and depends on it.

Testing is done throughout system development at various stages. It is always a good practice to test the system at many different levels at various intervals, that is, sub systems, program modules as work progresses and finally the system as a whole. During testing the major activities are concentrated on the examination and modification of the source code. Usually, this testing is to be performed by the person other than the person who has really coded it. This is done in order to ensure more complete and unbiased testing for making the software more reliable.

There are two types of testing:

- Black box testing
- White box testing

6.3.1 WHITE BOX TESTING

In white box testing, the internal logic of the modules is considered. Following levels of testing are performed for the developed project:

6.3.1.1 Unit Testing

This involves the tests carried out on modules programs, which make up a system. This is also called as a program testing. The units in a large system many modules at different levels are needed. Unit testing focuses on the modules, independently of one another, to locate errors. The program should be tested for correctness of logic applied and should detect errors in coding. Before proceeding one must make sure that all the programs are working independently.

6.3.2 BLACK BOX TESTING

The concept of the black box is used to represent a system whose inside workings are not available for inspection. In a black box, the test item is treated as “black”, since its logic is unknown; all that is known is what goes in and what comes out, or the input and output.

6.3.2.1 System Testing

The system testing is conducted on a complete, integrated system to evaluate the system’s compliance with its specified requirement. It falls within scope of black box testing so no knowledge of inner design or logic is needed. As a rule, system testing takes, as its input, all of the integrated software components that have passed integration testing and also the software system itself integrated with any applicable hardware system. The purpose of the integration testing is to detect any inconsistencies between software units.

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commence. The logical design and the physical design should be thoroughly and continually examined on paper ensure that they will work when implemented.

6.3.2.2 Integration Testing

Integration testing is a systematic technique for constructing the program structure, while at the same time conducting tests to uncover errors associated with interfacing. This is the program is constructed and tested in small segments, which makes it easier to isolate and the following common types of integration problems may be observed:

- Version mistakes
- Data integrity violations
- Overlapping function
- Resource problems especially in memory handling
- Wrong type of parameter in function calls

6.3.2.3 Validation Testing

At the culmination of the integration testing, the software was completely assembled as a package, interfacing errors have been uncovered and corrected and a final series of software validation testing began.

In validation testing we test the system functions in a manner that can be reasonably expected by customer, the system was tested against system requirement specification. Different unusual inputs that the users may use were assumed and the outputs were verified for such unprecedented inputs. Deviation or errors discovered at this step are corrected prior to the completion of this project with the help of user by negotiating to establish a method for resolving deficiencies. Thus, the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Validation checking is performed on the: -

Numeric Field: - The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error message. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested module are integrated into a single system.

Character Field: - This field can only contain letters from A-Z and a-z. It is useful for name, address fields and so on.

Check Null Fields: - Before entering values into the database or when updating, a validation is done to check whether any NULL fields are present.

Email Fields: - A email only field with a limit of characters. All the necessary validation checks were verified to see if invalid data ever enters into the database. Null values in fields were also treated as invalid values.

Password Fields: - A password only field with a limit of characters. All the necessary validation

checks were verified to see if invalid data ever enters into the database. Null values in fields were also treated as invalid values.

6.3.3 OUTPUT TESTING

After performing validation test, the next phase is output test of the system, since no system could be useful if it does not produce the desired output in the desired format. By consideration the format of the report/output was generated or displayed and was tested. Here output format was considered in one way: on the display screen.

6.3.4 USER ACCEPTANCE TESTING

User acceptance test of a system is the key factor for the success of the system. The system under consideration was listed for user acceptance by keeping constant touch with the perspective user of the system at the time of design, development and making changes whenever required. This was done with the regards of the following points: -

- Input screen design
- Output design

Users from each of the 2 user types (Admin, user) were selected for user acceptance testing. The Admin was given the software for testing with his username and password. The admin actions are performed and see whether all details are entering into the database and working properly as expected... The customers side is tested using a customer name and password by registering to the system and see he can post property from the website .

6.4 IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned into a working system. The implementation stage is a system project in its own right. It includes careful planning, investigation of current system and its constraints on implementation, design of methods to achieve the changeover, training of the staff in the changeover procedure and evaluation of the changeover method.

The first task in implementation is planning deciding on the methods and time scale to be adopted. Once the planning has been completed the major effort is to ensure that the programs in the system are

working properly when the user has been trained. The complete system involving both computer and user can be executed effectively. Thus, the clear plans are prepared for the activities.

Successful implementation of the new system design is a critical phase in the system life cycle. Implementation means the process of converting a new or a revised system design into an operational one.

MAINTENANCE & ENHANCEMENT

7. MAINTENANCE AND ENHANCEMENT

7.1 MAINTENANCE

This software can be modified as need occurs. Maintenance includes all the activities after installation of the software that is performed to keep the system operational. The process of maintenance involves:

- Understanding the existing software
- Understand the effect of change
- Test for satisfaction

This software requires little to no maintenance. During the testing phase most maintenance duties are performed. If a maintenance requirement occurs, it can be solved with ease

7.2 ENHANCEMENT

This software is divided into modules. Therefore, adding extra functionalities is simple. Enhancements can be added and newer versions can be created with ease as requirements increase and technologies develop.

The modular nature of this software enables seamless integration with future modules. Any function can be added to the web application in the future using another module. This independent module can then be added to the web application with little or no effort.

In future the application can also:

- Online Learning Modules
- AI-Powered Chatbots for Student Queries
- Virtual Classrooms with Video Conferencing
- Advanced Performance Analytics
- Automated Fee Reminders and Notifications
- Mobile App Integration for Easy Access

CONCLUSION

8. CONCLUSION

In today's fast-paced educational environment, technology plays a pivotal role in streamlining operations and enhancing the learning experience. Traditional methods of managing tuition centers are being transformed by digital platforms that simplify tasks for administrators, teachers, and students alike. This Tuition Center Management System reflects the growing importance of technology in education.

Developed with a focus on efficiency and ease of use, this website transcends the usual limitations of manual processes. It offers a seamless platform for managing student enrollment, class schedules, attendance tracking, and fee payments, making it easier for everyone involved to stay organized and informed.

The primary goal of this project has been to create a comprehensive system that benefits all users—students, teachers, and administrators. By providing easy access to information, automating routine tasks, and improving communication, this system significantly enhances the overall management of tuition centers.

As education continues to evolve in the digital age, this project aligns with the increasing demand for streamlined, tech-enabled solutions. It improves the efficiency of managing tuition centers while giving users real-time access to important features. Moving forward, we are committed to refining and expanding the system, incorporating new technologies and user needs to ensure that this platform continues to support and enhance the educational experience.

This project is not just a conclusion but a foundation for future developments, where technology and education come together to improve how we manage and experience learning environments.

BIBLIOGRAPHY

BIBLIOGRAPHY

WEB REFERENCES

- www.w3schools.com
- www.tutorialpoint.com
- Chatgpt
- Blackbox AI
- Google Gemini
- www.templatemo.com
- www.wikipedia.org
- Pinterest

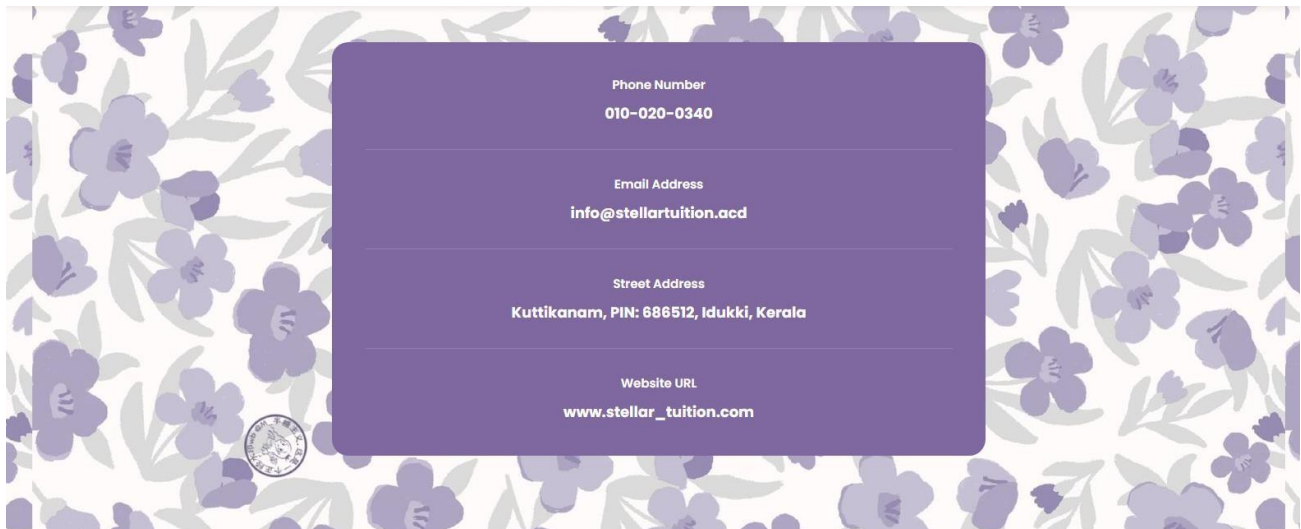
APPENDIX

APPENDIX

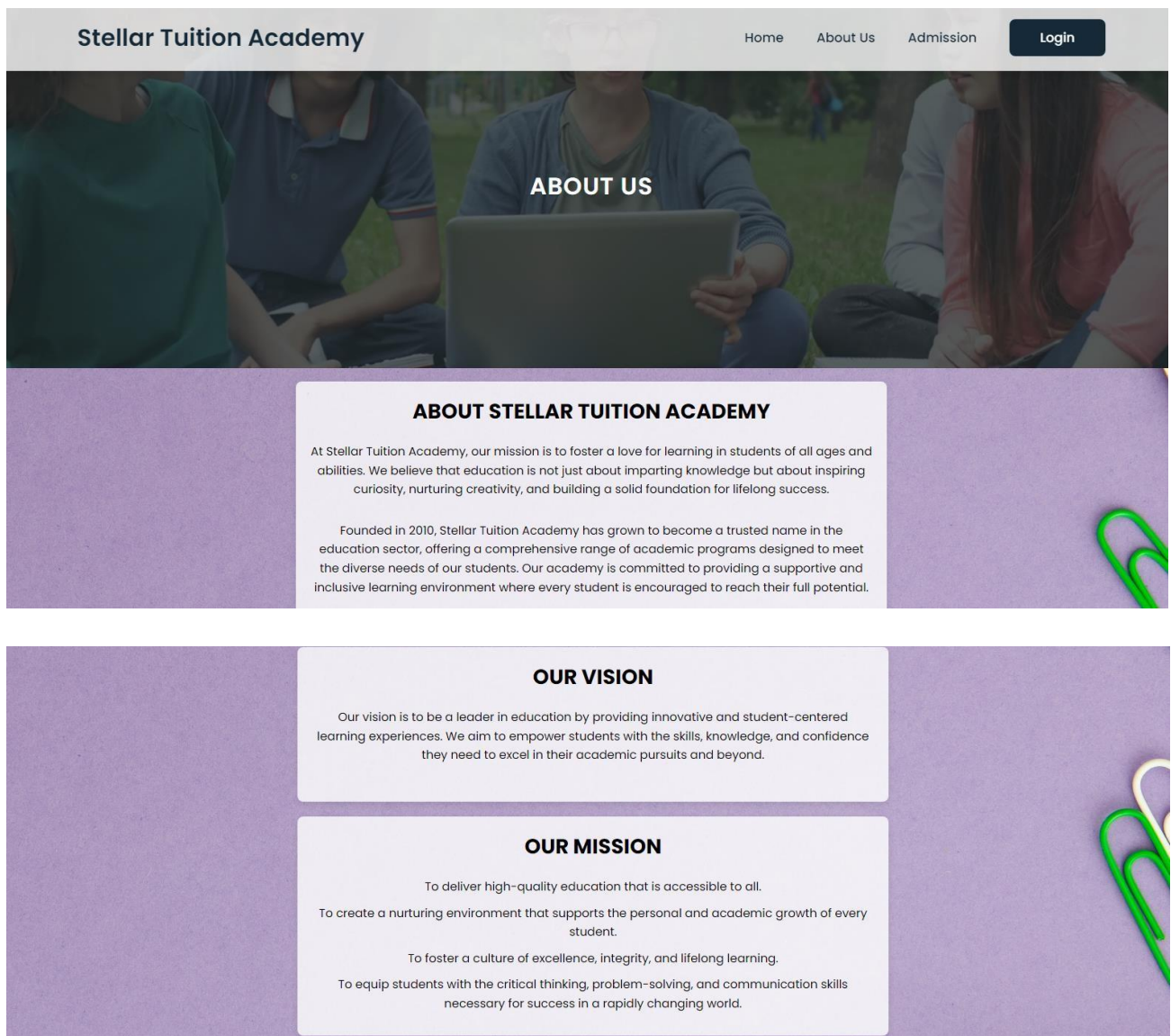
SCREENSHOTS

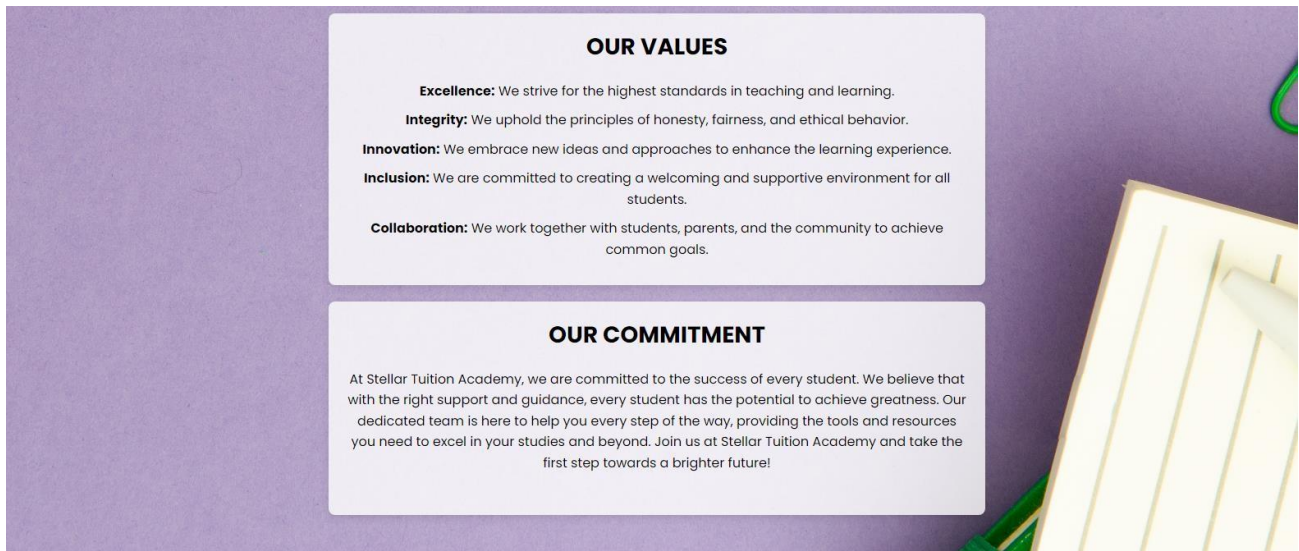
1. HOME PAGE





2. ABOUT US PAGE

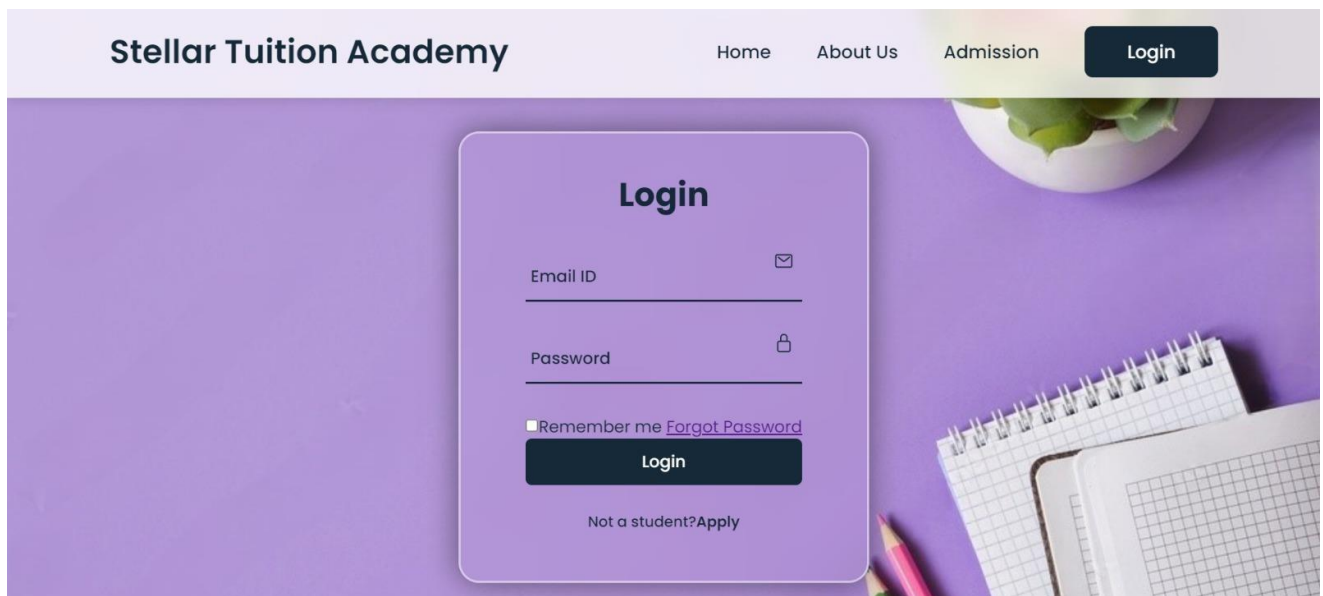




3. ADMISSION APPLICATION PAGE

A purple background with a white box containing an 'Admission Form'. The form fields are: First Name, Last Name of the applicant, Contact Number, Email, Guardian's Phone Number, Date of Birth (with a calendar icon and the format dd-mm-yyyy), Select Class (a dropdown menu), Password, and Confirm Password. A dark blue 'Apply' button is at the bottom. A purple folder with paper clips is visible on the right side of the image.


4. LOGIN PAGE




The login page features a purple header with the academy's name and navigation links. A central white login box contains fields for email and password, a 'Remember me' checkbox, a 'Forgot Password' link, and a 'Login' button. A link to 'Apply' is provided for non-students. The background is a purple desk with school supplies.

Stellar Tuition Academy Home About Us Admission **Login**

Login

Email ID 

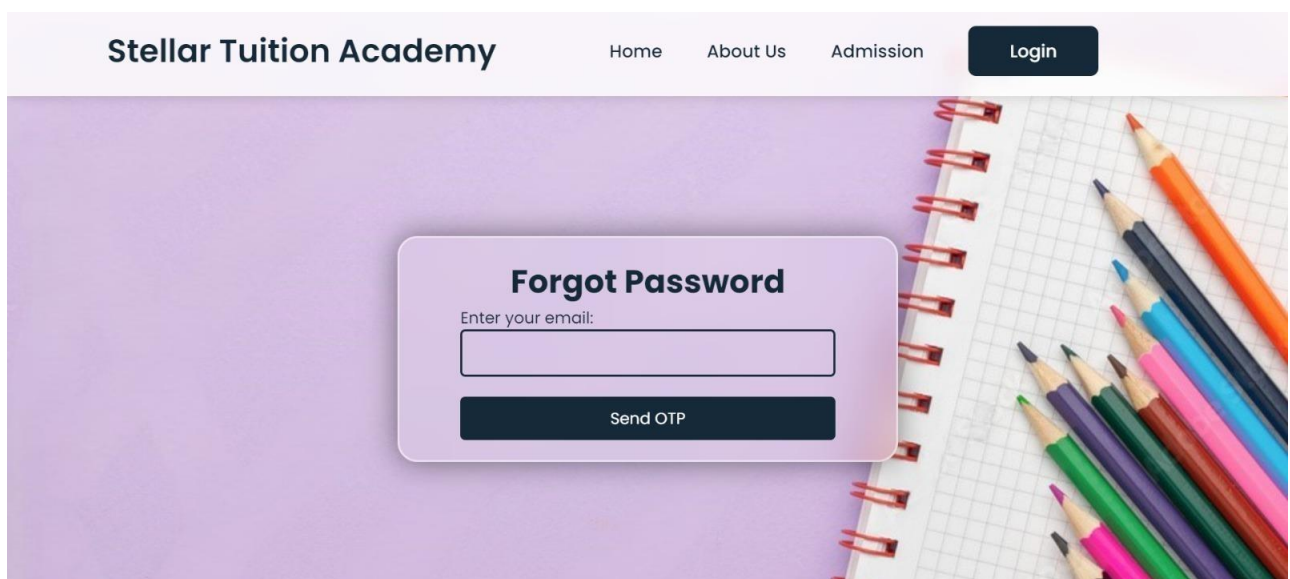
Password 

☐ Remember me [Forgot Password](#)

Login

[Not a student? Apply](#)

5. FORGOT PASSWORD PAGE



The forgot password page has a purple header with navigation links. A central white box prompts the user to enter their email and provides a 'Send OTP' button. The background is a purple desk with school supplies.

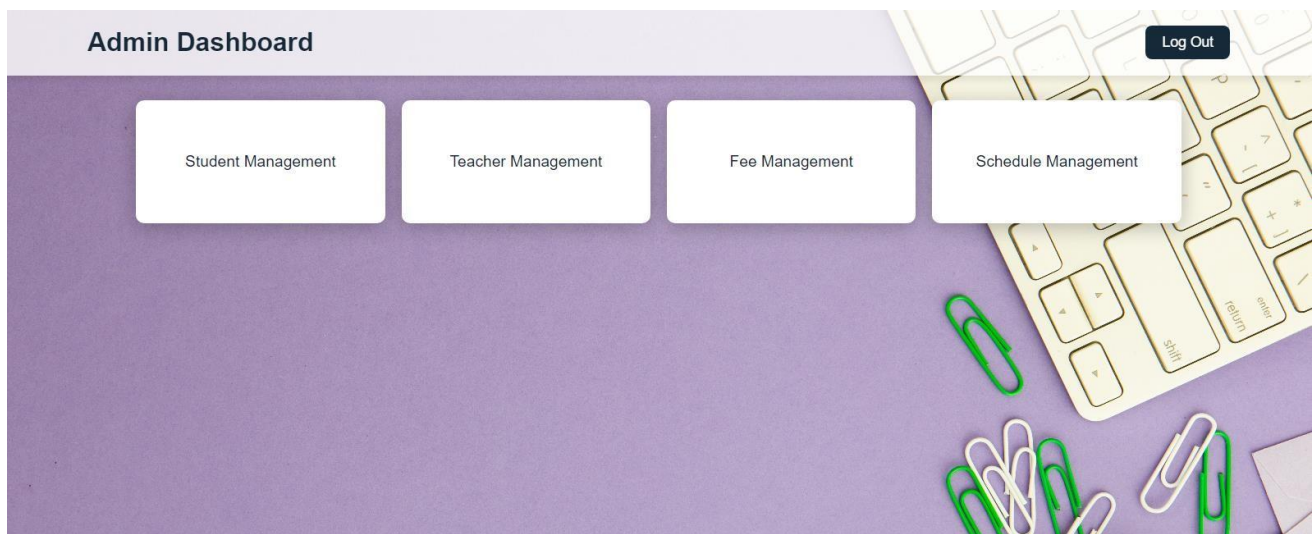
Stellar Tuition Academy Home About Us Admission **Login**

Forgot Password

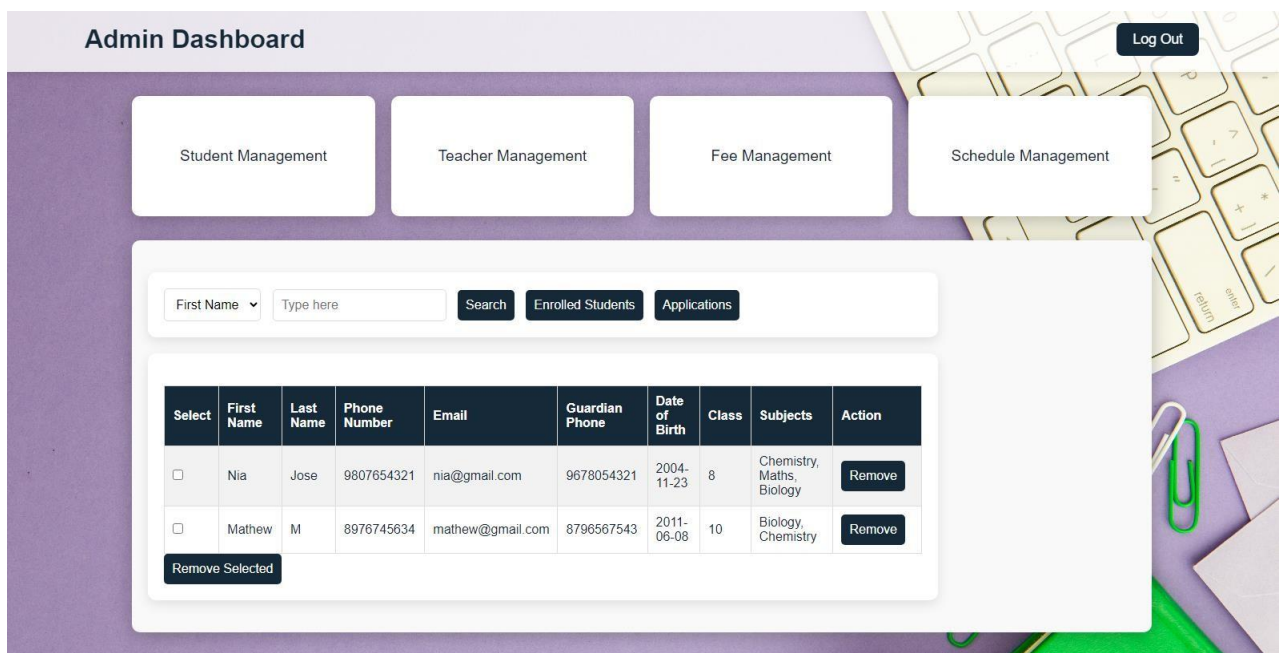
Enter your email:

Send OTP

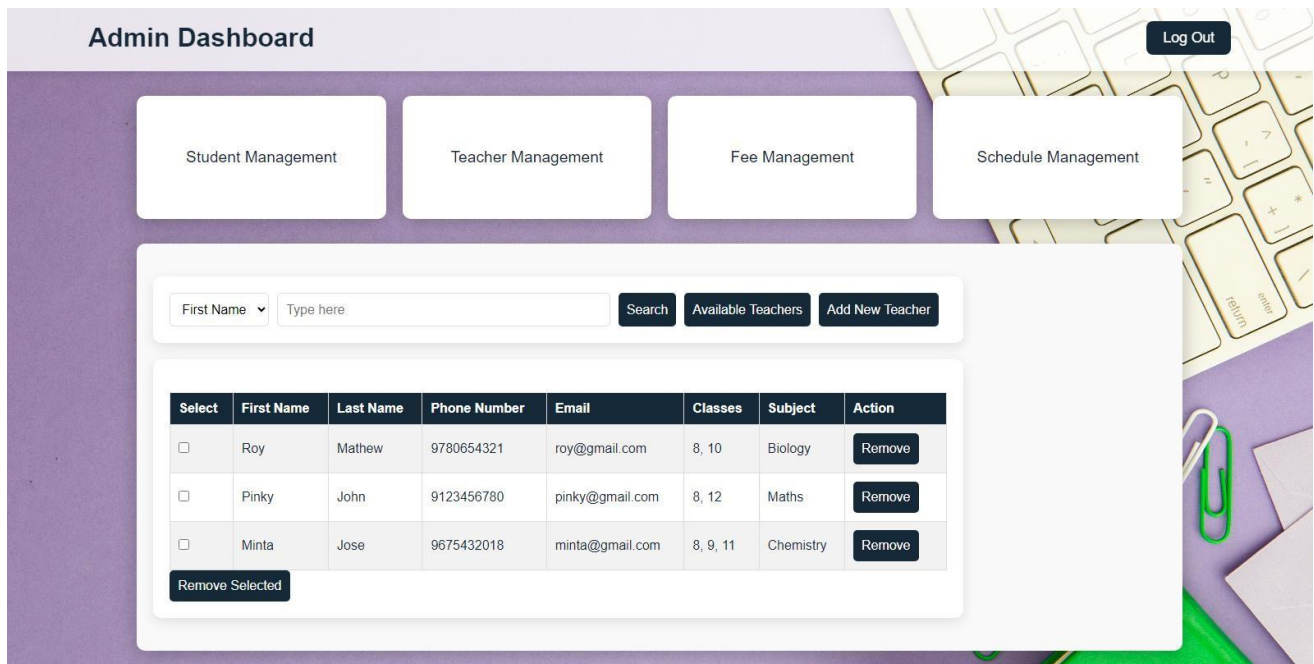
6. ADMIN DASHBOARD



7. ADMIN – STUDENT MANAGEMENT



8. ADMIN – TEACHER MANAGEMENT

A screenshot of an Admin Dashboard with a purple header and background. The header contains the text 'Admin Dashboard' and a 'Log Out' button. Below the header are four white boxes labeled 'Student Management', 'Teacher Management', 'Fee Management', and 'Schedule Management'. The 'Teacher Management' box is active and contains a search bar with a dropdown for 'First Name' and a text input 'Type here'. To the right of the search bar are three buttons: 'Search', 'Available Teachers', and 'Add New Teacher'. Below the search bar is a table with 8 columns: 'Select', 'First Name', 'Last Name', 'Phone Number', 'Email', 'Classes', 'Subject', and 'Action'. The table has three rows of teacher data. Each row has a checkbox in the 'Select' column and a 'Remove' button in the 'Action' column. Below the table is a 'Remove Selected' button.

Admin Dashboard

Log Out

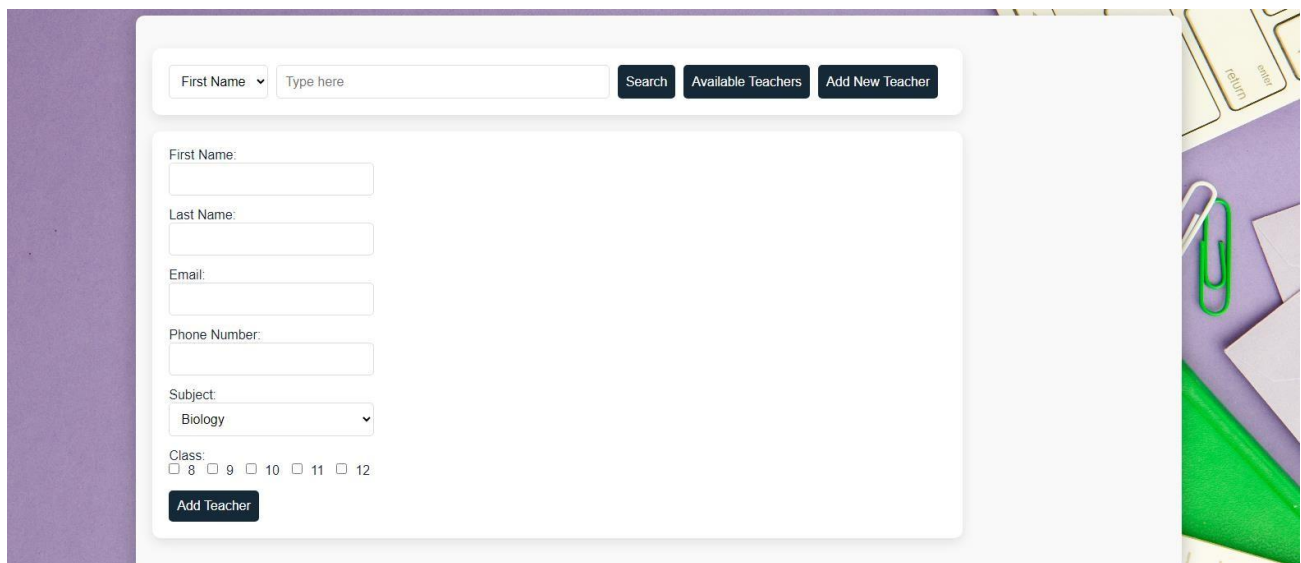
Student Management Teacher Management Fee Management Schedule Management

First Name ▼ Type here Search Available Teachers Add New Teacher

Select	First Name	Last Name	Phone Number	Email	Classes	Subject	Action
<input type="checkbox"/>	Roy	Mathew	9780654321	roy@gmail.com	8, 10	Biology	<button>Remove</button>
<input type="checkbox"/>	Pinky	John	9123456780	pinky@gmail.com	8, 12	Maths	<button>Remove</button>
<input type="checkbox"/>	Minta	Jose	9675432018	minta@gmail.com	8, 9, 11	Chemistry	<button>Remove</button>

Remove Selected

9. ADD NEW TEACHER FORM

A screenshot of the 'Add New Teacher' form. It features a search bar at the top with a dropdown for 'First Name' and a text input 'Type here'. To the right of the search bar are three buttons: 'Search', 'Available Teachers', and 'Add New Teacher'. Below the search bar is a form with several input fields: 'First Name:', 'Last Name:', 'Email:', 'Phone Number:', 'Subject:' (a dropdown menu currently showing 'Biology'), and 'Class:' (a row of checkboxes for 8, 9, 10, 11, and 12). At the bottom of the form is an 'Add Teacher' button.

First Name ▼ Type here Search Available Teachers Add New Teacher

First Name:

Last Name:

Email:

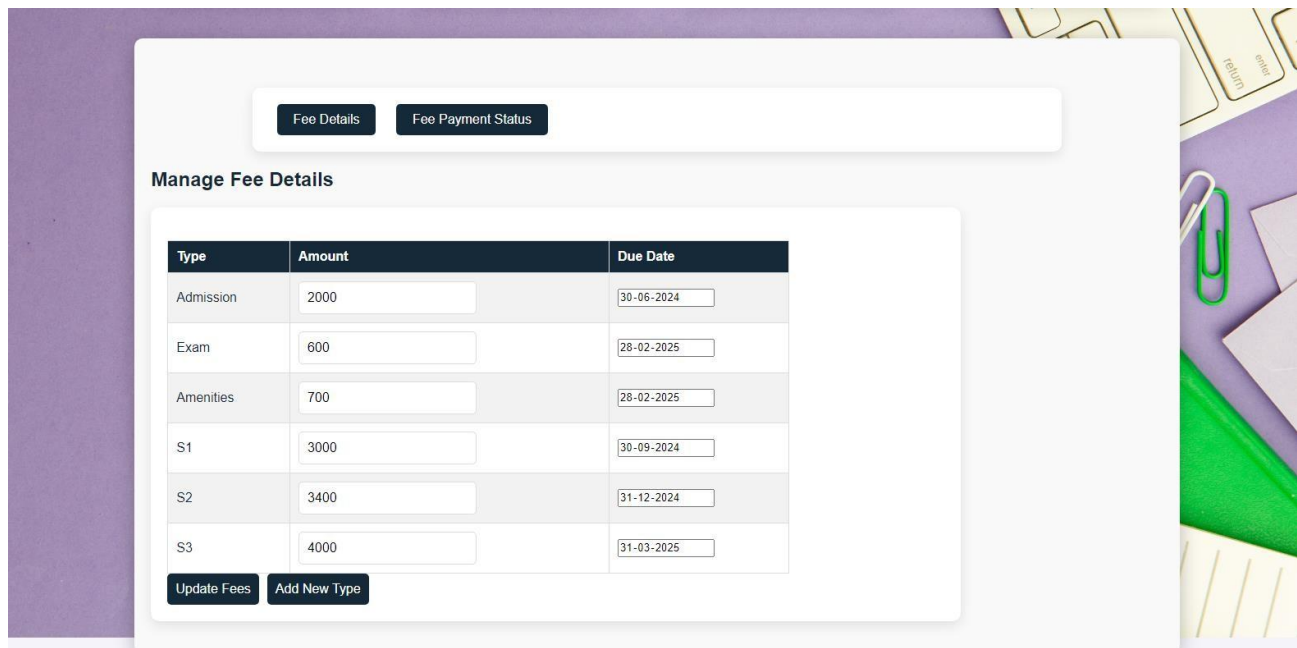
Phone Number:

Subject:
Biology ▼

Class:
☐ 8 ☐ 9 ☐ 10 ☐ 11 ☐ 12

Add Teacher

10. ADMIN – FEE MANAGEMENT

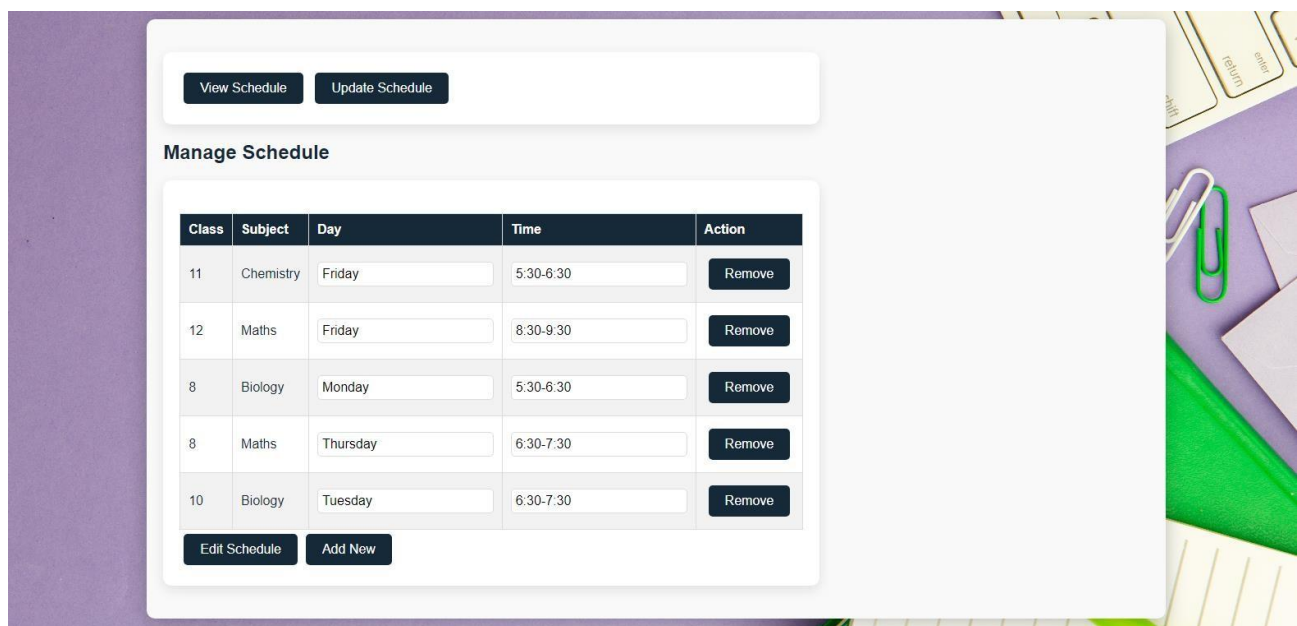


The screenshot displays the 'Manage Fee Details' interface. At the top, there are two buttons: 'Fee Details' (active) and 'Fee Payment Status'. Below the title, a table lists various fee types with their respective amounts and due dates. Each amount and due date is in a text input field. At the bottom of the table, there are two buttons: 'Update Fees' and 'Add New Type'.

Type	Amount	Due Date
Admission	2000	30-06-2024
Exam	600	28-02-2025
Amenities	700	28-02-2025
S1	3000	30-09-2024
S2	3400	31-12-2024
S3	4000	31-03-2025

Update Fees Add New Type

11. ADMIN – SCHEDULE MANAGEMENT

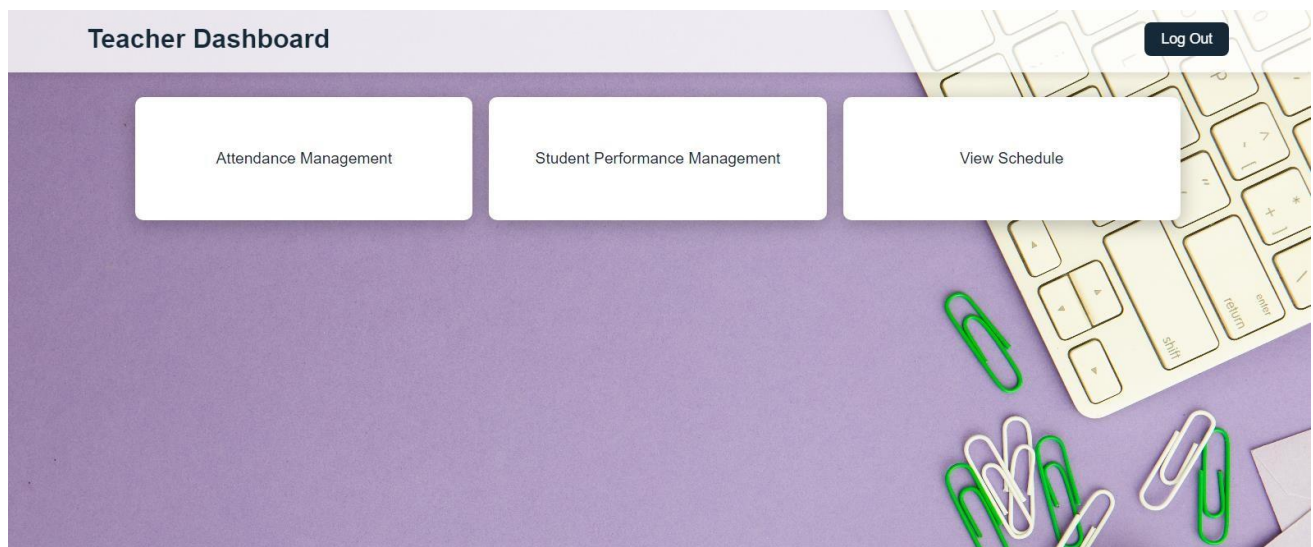


The screenshot displays the 'Manage Schedule' interface. At the top, there are two buttons: 'View Schedule' and 'Update Schedule' (active). Below the title, a table lists class schedules with columns for Class, Subject, Day, Time, and Action. Each Day and Time is in a text input field. The Action column contains a 'Remove' button for each row. At the bottom of the table, there are two buttons: 'Edit Schedule' and 'Add New'.

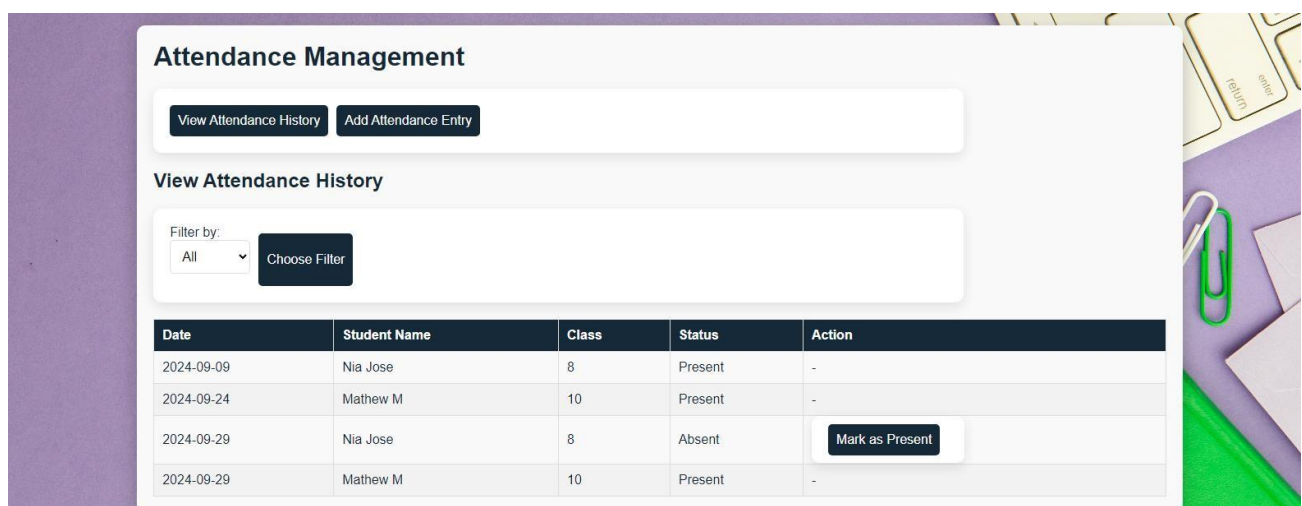
Class	Subject	Day	Time	Action
11	Chemistry	Friday	5:30-6:30	Remove
12	Maths	Friday	8:30-9:30	Remove
8	Biology	Monday	5:30-6:30	Remove
8	Maths	Thursday	6:30-7:30	Remove
10	Biology	Tuesday	6:30-7:30	Remove

Edit Schedule Add New

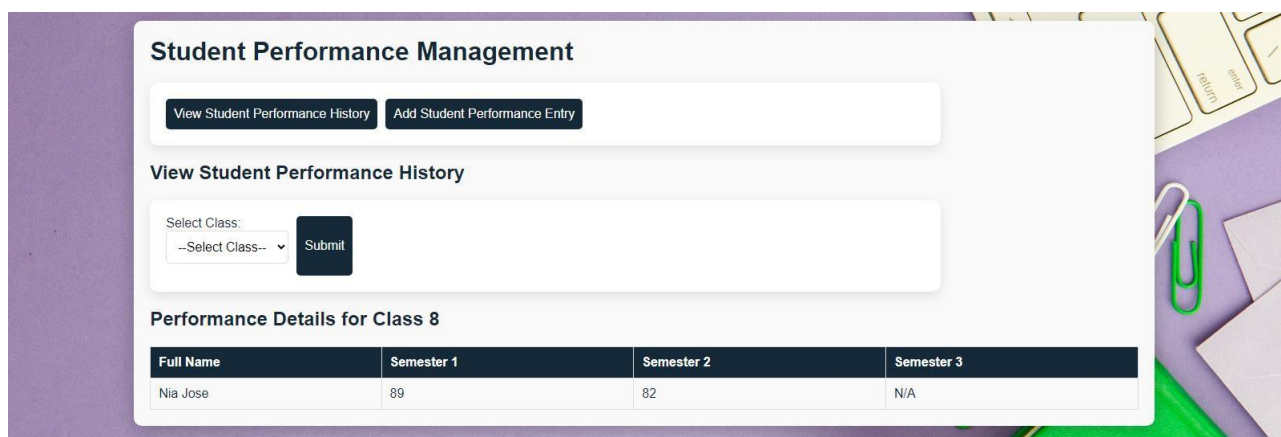
12. TEACHER DASHBOARD



13. TEACHER – ATTENDANCE MANAGEMENT



14. TEACHER – STUDENT PERFORMANCE MANAGEMENT



15. TEACHER - VIEW SCHEDULE

Teacher Dashboard

Log Out

Attendance Management

Student Performance Management

View Schedule

Class	Day	Time
8	Monday	5:30-6:30
10	Tuesday	6:30-7:30

16. STUDENT DASHBOARD

Student Dashboard

Log Out

Attendance Overview

Performance Overview

Schedule Overview

Fee Payment

17. STUDENT – ATTENDANCE OVERVIEW

Student Dashboard

Log Out

Attendance Overview

Performance Overview

Schedule Overview

Fee Payment

Date	Subject	Status
2024-09-09	Biology	Present
2024-09-29	Biology	Absent

18. STUDENT – PERFORMANCE OVERVIEW

Student Dashboard

Log Out

Attendance Overview

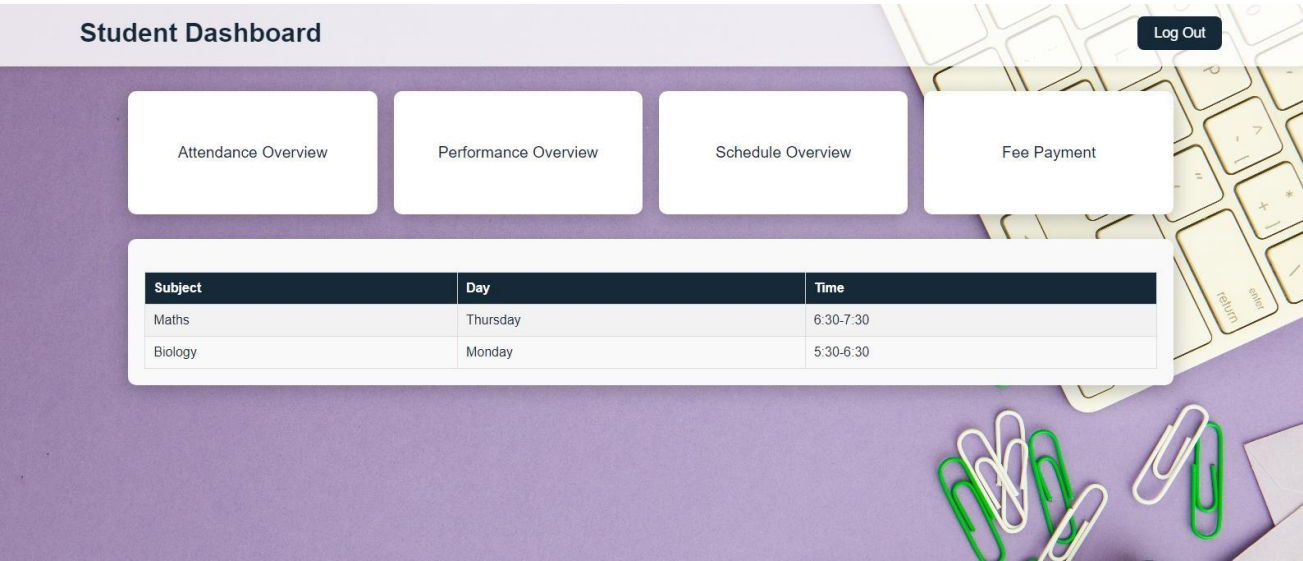
Performance Overview

Schedule Overview

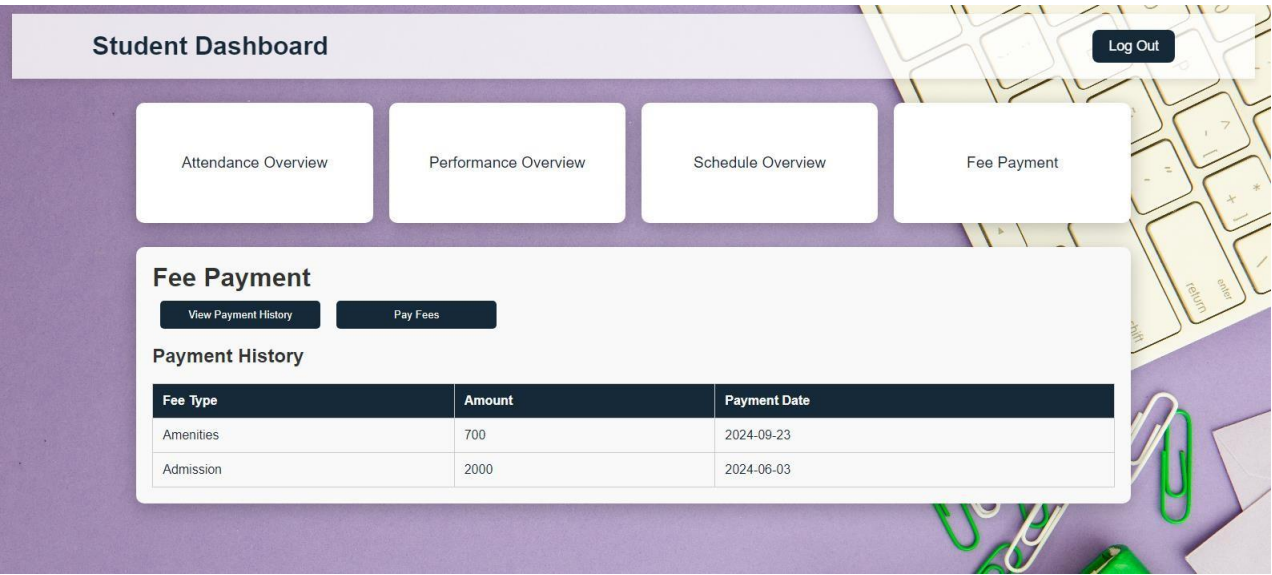
Fee Payment

Subject	Semester 1	Semester 2	Semester 3
Chemistry	N/A	N/A	N/A
Maths	N/A	N/A	N/A
Biology	89	82	N/A

19. STUDENT – SCHEDULE OVERVIEW



20. STUDENT – FEES OVERVIEW



21. STUDENT – FEE PAYMENT FORM

Fee Payment

View Payment HistoryPay Fees

Unpaid Fees

Fee Type	Amount	Due Date	Action
Semester 1	9000	2024-09-30	<button>Pay</button>
Semester 2	10200	2024-12-31	<button>Pay</button>
Exam	600	2025-02-28	<button>Pay</button>
Semester 3	12000	2025-03-31	<button>Pay</button>

Total Amount: 31800

Pay All

Fee Payment

View Payment HistoryPay Fees

Card Number:

Expiry Date (MM/YY):

CVV:

Name on Card:

Verify