



UniFlow

(University Workflow Management)

DONE BY -TAMIZHARASAN M

INTRODUCTION

The **Student Academics, Attendance, Scheduling, and Fee Management System** is a web-based application developed using **Java Server Pages (JSP)**, **Java classes**, and **MySQL database**. This system efficiently manages student records, attendance tracking, fee management, grading, and scheduling within an educational institution. It provides an interactive interface for administrators and staff to handle student-related data seamlessly.

PROJECT MODULES & FEATURES

The system consists of **10 core pages**, each handling specific aspects of student management.

1. **INDEX PAGE (DASHBOARD)**
2. **STUDENT LIST (LISTSTUDENT.JSP)**
3. **ADD STUDENT (ADDSTUDENT.JSP)**
4. **ATTENDANCE RECORDS (ATTENDANCERECORDS.JSP)**
5. **ADD ATTENDANCE (ADDATTENDANCE.JSP)**
6. **FEE MANAGEMENT (FEES.JSP)**
7. **ADD FEE (ADDFEE.JSP)**
8. **GRADES MANAGEMENT (GRADES.JSP)**
9. **ADD GRADE (ADDGRADE.JSP)**
10. **CLASS SCHEDULE (SCHEDULE.JSP)**

INDEX PAGE



The index page serves as the main dashboard of the system, providing quick access to all core functionalities. It includes navigation options to list students, add new students, view attendance records, manage fees, manage grades, and view class schedules. This page acts as a central hub, making it easier for administrators to manage student data efficiently.

Acts as the **dashboard** for the system.

- Provides **quick navigation** to key features:
- **List Students**
- **Add New Student**
- **View Attendance Records**
- **Manage Fees**
- **Manage Grades**
- **View Class Schedule**

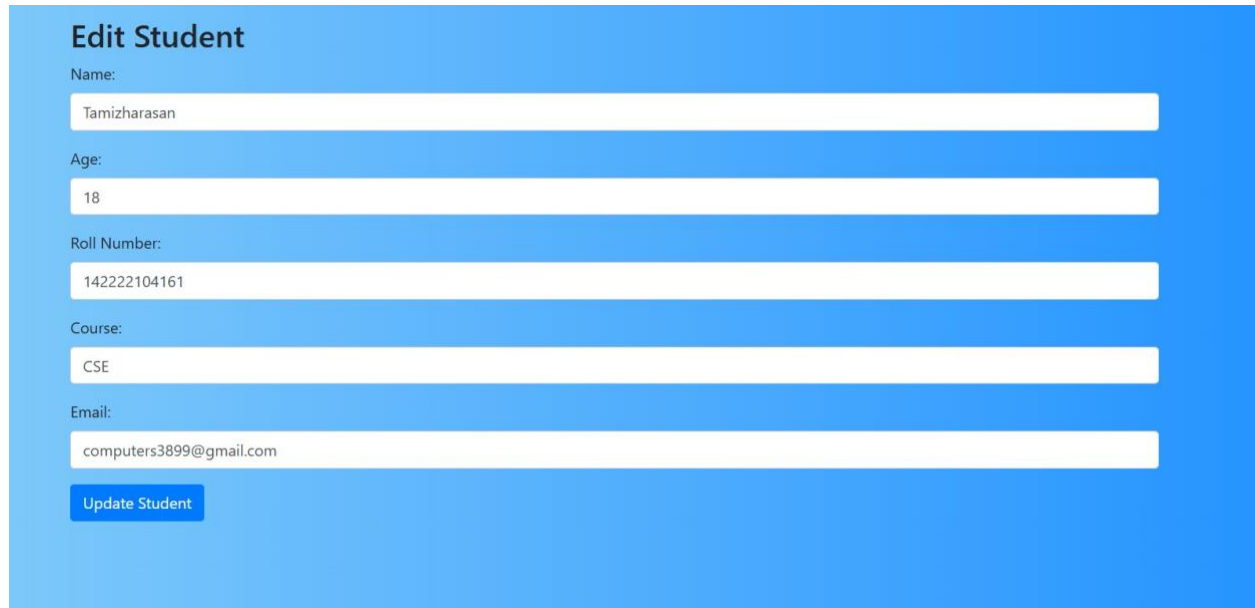
STUDENT LIST PAGE

Student List						
Add New Student						
ID	Name	Age	Roll Number	Course	Email	Actions
10	Tamizharasan	18	142222104161	CSE	computers3899@gmail.com	Edit Delete
11	Rose	21	null	ai	computers3899@gmail.com	Edit Delete
12	Rose	21	null	ai	computers3899@gmail.com	Edit Delete
13	mudarsir	19	null	be	1234@gmail.com	Edit Delete

This page displays a tabular list of students, including details such as student ID, name, age, roll number, course, and email. Each student entry has associated "Edit" and "Delete" buttons, allowing modifications and removal of records. Additionally, there is a button to add a new student, making it convenient for administrators to update the student database.

- Displays a **table of student records** with the following details:
- **ID, Name, Age, Roll Number, Course, Email**
- Each student has **Edit and Delete** buttons for modifications.
- A **button to add a new student** is available.

EDIT PAGE



Edit Student

Name:

Age:

Roll Number:

Course:

Email:

The **edit and delete functionality** allows administrators to manage student records efficiently from the **Student List (listStudent.jsp)** page. Each student entry in the table has an "Edit" button, which redirects the administrator to an **edit form pre-filled with the student's existing details**. The administrator can modify details such as **name, age, roll number, course, and email**, and upon submission, the updated information is stored in the **MySQL database**.

DELETE PAGE

The screenshot displays a web application interface with a blue background. At the top left, the title 'Student List' is shown in bold, with a blue button labeled 'Add New Student' below it. A table with 7 columns (ID, Name, Age, Roll Number, Course, Email, Actions) contains 5 rows of student data. Each row has 'Edit' (yellow) and 'Delete' (red) buttons in the 'Actions' column. A dark gray confirmation dialog box is open in the top right, displaying the text 'localhost:8080 says' and 'Are you sure you want to delete this student?' with 'OK' and 'Cancel' buttons.

ID	Name	Age	Roll Number	Course	Email	Actions
10	Tamizharasan	18	142222104161	CSE	computers3899@gmail.com	<button>Edit</button> <button>Delete</button>
11	Rose	21	null	ai	computers3899@gmail.com	<button>Edit</button> <button>Delete</button>
12	Rose	21	null	ai	computers3899@gmail.com	<button>Edit</button> <button>Delete</button>
13	mudarsir	19	null	be	1234@gmail.com	<button>Edit</button> <button>Delete</button>
14	Victoria	22	null	CSE	victoria233@gmail.com	<button>Edit</button> <button>Delete</button>

The "Delete" button, on the other hand, enables the removal of a student record from the database. When an administrator clicks the delete button, a **confirmation prompt** appears to prevent accidental deletions. Once confirmed, the system executes a **delete query**, permanently removing the student's record from the database. This feature ensures **data integrity** and allows administrators to keep the student database up to date by removing outdated or incorrect entries.

ADD STUDENT PAGE



Add New Student

Name: Victoria

Age: 22

Roll Number: 1002

Course: CSE

Email: victoria233@gmail.com

Add Student

The add student page contains a form where administrators can input new student details, including ID, name, age, roll number, course, and email. Upon submission, the data is stored in the MySQL database and reflected on the student list page. This ensures easy record-keeping and seamless addition of students into the system.

- A form to **input new student details**:
- **ID, Name, Age, Roll Number, Course, Email**
- The data is stored in the **MySQL database** upon submission.

ATTENDANCE RECORDS PAGE

Attendance Records

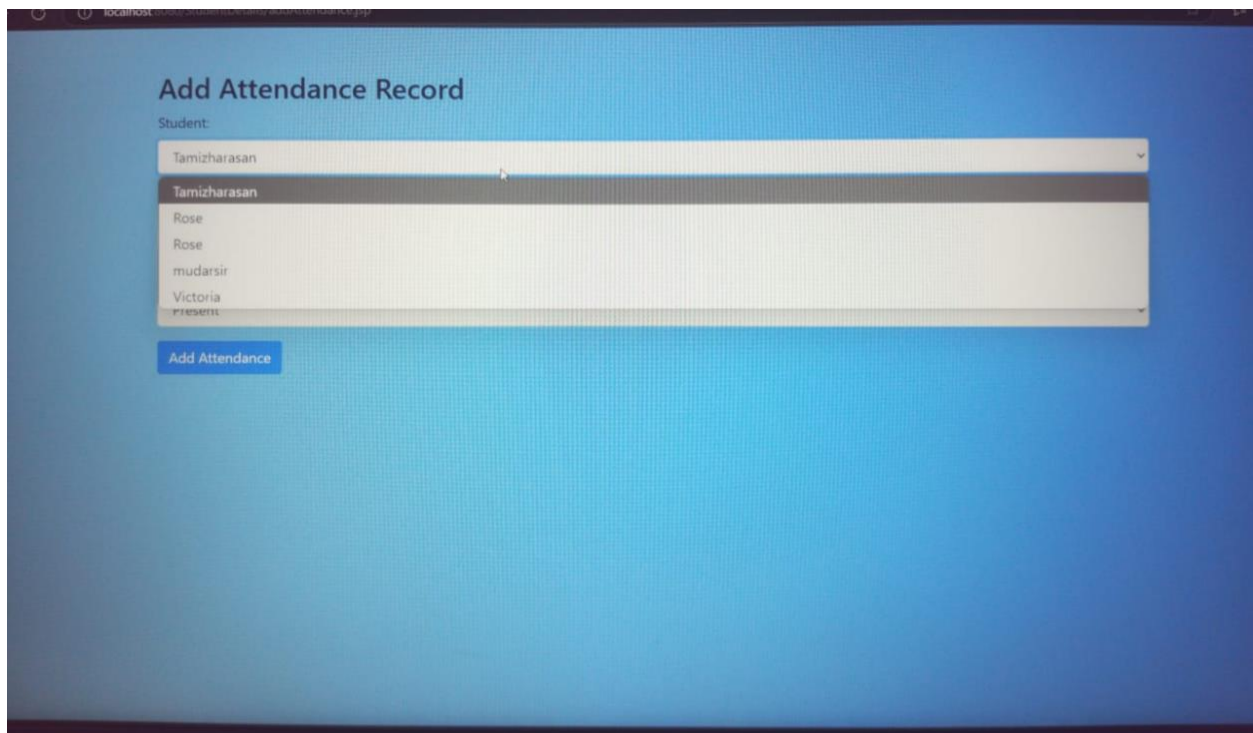
Add New Attendance

ID	Student	Date	Status
1	Tamizharasan	2024-09-02	Present
2	Tamizharasan	2024-09-12	Absent
3	Rose	2024-09-11	Absent

This page provides a list of student attendance records, displaying each student’s ID, name, date, and attendance status (present or absent). Administrators can track attendance history and also add new records. This feature helps in monitoring student participation and maintaining attendance logs efficiently.

- Displays **student attendance history** in a table format:
- **ID, Name, Date, Status (Present/Absent)**
- Provides an option to **add new attendance records**.

ADD ATTENDANCE PAGE



localhost:3000/students/addAttendancePage

Add Attendance Record

Student:

Tamizharasan

Tamizharasan

Rose

Rose

mudarsir

Victoria

Present

Add Attendance

Add Attendance Record

Student:

Tamizharasan

Date:

dd-mm-yyyy

February, 2025

Su	Mo	Tu	We	Th	Fr	Sa
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	1
2	3	4	5	6	7	8

Clear today

Add Attendance Record

Student:

Tamizharasan

Date:

dd-mm-yyyy

Status:

Present

Present

Absent

The add attendance page allows administrators to record student attendance. It includes a dropdown list populated with students already present in the database, along with input fields for selecting the attendance date and marking the student as present or absent. The recorded data is then stored and reflected in the attendance records page.

A form to **record student attendance**:

- **Dropdown list** of students fetched from the database.
- **Attendance Date & Status (Present/Absent)**

FEE MANAGEMENT PAGE

Fees Records

Add New Fee

ID	Student ID	Amount	Due Date	Status
1	10	1000.0	2024-09-02	null
3	10	1000.0	2024-09-03	null
5	10	1000.0	2024-09-03	null
7	10	1222.0	2024-09-02	Pending
17	10	1234.0	2024-09-19	Pending
20	10	1000.0	2024-09-04	Paid
21	10	1000.0	2024-09-11	Not Paid
22	10	1000.0	2024-09-05	Pending

This page displays student fee records in a tabular format, showing details such as student ID, fee amount, due date, and payment status (paid, not paid, or pending). Administrators can track fee payments and outstanding dues, ensuring proper financial management within the institution.

Displays **fee payment status** for students in a tabular format:

- **Student ID, Amount, Due Date, Status (Paid, Not Paid, Pending)**

ADD FEE PAGE

Add New Fee

Student ID

Amount

Due Date

Status

Select your option

Add Fee

Add New Fee

Student ID

Amount

Due Date

Status

Paid

Add Fee

Add New Fee

Student ID
1001

Amount
10000

Due Date
12-02-2025

Status
Paid
Select your option
Paid
Pending
Not Paid

The add fee page provides a form where administrators can enter fee-related details, including student ID, fee amount, due date, and payment status. Once submitted, the data is stored in the database and displayed on the fee management page, helping in maintaining accurate financial records.

- A form to **enter new fee details**:
- **Student ID, Amount, Due Date, Payment Status**
- The entered data is displayed in the **fees.jsp** page.

GRADES MANAGEMENT PAGE

Grades Records

Add New Grade

ID	Student ID	Subject	Grade	Date
1	11	Maths	O	N/A
2	11	Maths	O	2024-09-05
3	12	Physics	C	2024-09-04

This page lists student grades in a structured table, including student ID, subject, grade, and the date of evaluation. Administrators and teachers can review student performance at a glance, making it easier to track academic progress and generate reports if needed.

Displays **student academic performance**:

- **Student ID, Subject, Grade, Date**

ADD GRADE PAGE

Add New Grade

Student ID

Subject

Grade

Date

Add Grade

Add New Grade

Student ID

Subject

Grade

O

O

A+

A

B+

B

C

Fail

The add grade page contains a form where administrators can input student grades. It includes fields for student ID, subject, grade (O, A+, A, B+, B, C, Fail), and date. Once submitted, the grades are stored in the database and reflected on the grades management page, ensuring an organized grading system.

- A form to **assign grades to students**:
- **Student ID, Subject, Grade (O, A+, A, B+, B, C, Fail), Date**
- The entered grades are stored and displayed in **grades.jsp**.

CLASS SCHEDULE PAGE

Schedule Records

Student ID:

Day of Week:

Time Slot:

Subject:

Add Schedule

ID	Student ID	Day of Week	Time Slot	Subject
1	10	null	null	null
2	10	2	10:00 - 11:00	Maths

Schedule Records

Student ID:

20

Day of Week:

3

Time Slot:

12:00 - 1:00

Subject:

Maths

Add Schedule

ID	Student ID	Day of Week	Time Slot	Subject
1	10	null	null	null
2	10	2	10:00 - 11:00	Maths

The schedule page displays the class timetable, showing student ID, day of the week, time slot, and subject. This page allows administrators to manage and update class schedules directly. It helps in organizing academic activities, ensuring that students and teachers have a well-structured timetable to follow.

- Displays the **class timetable**:
- **Student ID, Day of the Week, Time Slot, Subject**
- Allows direct input for scheduling updates.

WORKING REPORT

TECHNOLOGY STACK

- **Frontend:** HTML, CSS, JavaScript (for dynamic UI interactions)
- **Backend:** Java Server Pages (JSP), Servlets, Java Beans (for business logic)
- **Database:** MySQL (for storing and managing student-related data)
- **Server:** Apache Tomcat (for running JSP pages)

DATABASE STRUCTURE

The system utilizes **multiple relational tables**, including:

1. **students** – Stores student details
2. **attendance** – Tracks student attendance
3. **fees** – Manages student fee payments
4. **grades** – Stores academic performance
5. **schedule** – Manages class schedules

SYSTEM WORKFLOW

1. **Student Management:**
 - a. Admin can add, edit, and delete student records.
 - b. The student list is dynamically displayed using JSP.
2. **Attendance Management:**
 - a. Staff can mark student attendance.
 - b. Attendance records can be retrieved and updated.
3. **Fee Management:**
 - a. Admin can track student fees and update payment status.
4. **Grades Management:**
 - a. Staff can assign grades to students based on academic performance.
5. **Class Scheduling:**
 - a. Class schedules are displayed and updated in real-time.

CONCLUSION

This project provides a **user-friendly and efficient** way to manage student data. The integration of **JSP with MySQL** ensures **data integrity and seamless functionality** for educational institutions. The system can be further expanded with features like **student login portals, report generation, and automated notifications**.