My JAVA Project: Student Management System

Team Members:

- Krishnendra Singh Sengar (24SCSE1010438)

- Manjistha Chakraborthy (24SCSE1011261)

Acknowledgement

I sincerely thank my Java teacher for their guidance and support in completing this project. Your expertise helped us understand core concepts and implement this Student Management System effectively.

Special thanks to my team member for her valuable contribution, and for sticking together until the end.

Project Overview

This Java project manages student records with grades and details.

It uses a graphical user interface built with Swing components.
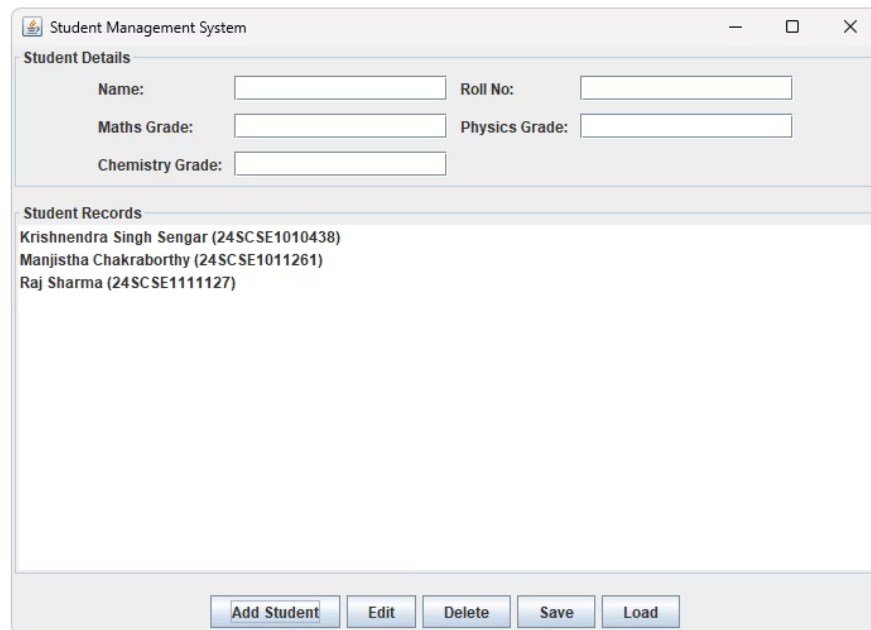
Features

Add, edit, delete, save, and load student data.

Data Structure

Stores students with name, roll number, and subject grades.

Role-based Access

Teacher mode enables full control; student mode is view-only.

Student Class Design

Represents each student with name, roll number, and grades.

Implements Serializable for saving and loading data.

Attributes

- Name

- Roll Number

- Subject Grades (Maths, Physics, Chemistry)

Methods

- Getters and setters for all fields

- toString method for display in list

Graphical User Interface

Built with JFrame and Swing components.

Includes input fields, buttons, and a student list.

Input Fields

Name, Roll No, and grades for three subjects.

Buttons

Add, Edit, Delete, Save, Load functionalities.

Student List

Displays all student records dynamically.

Role Selection and Security

Users select role: Teacher or Student at startup.

Teacher login requires username and password.

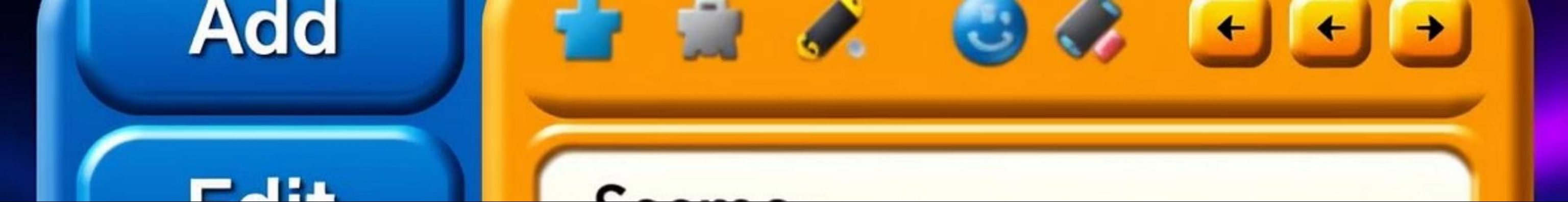## Teacher Mode

Full access with admin login.

Student Mode

Limited access without editing rights.

Core Functionalities

Add new students with complete details.

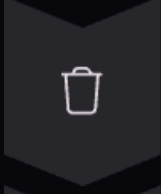Edit and delete existing student records.

Save all data to a file and load it later.

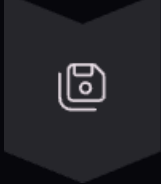**+** Add Student

Input validation ensures all fields are filled.

Ⅰ Edit Student

Select from list and update details.

🗑 Delete Student

Remove selected student from records.

💾 Save & Load

Serialize data to file and reload as needed.

Data Handling and Persistence

Uses Java serialization to save and load student list.

Handles IO exceptions with user notifications.

Saving Data

Writes student list to "students.dat" file.

Loading Data

Reads student list from file and updates UI.
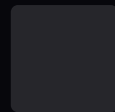
User Interaction and Validation

Input fields clear after operations for ease of use.

Selection in list fills input fields for editing.

Input Validation

Ensures no empty fields before adding or editing.

Field Clearing

Clears inputs after add, edit, or delete actions.

List Selection

Populates fields with selected student data.

Student Details

Name: [   ]   Roll No: [   ]

Maths Grade: [   ]   Physics Grade: [   ]

Chemistry Grade: [   ]

Student Records