A

PROJECT REPORT ON

"STUDENT INFORMATION MANAGEMENT SYSTEM"

By

Shewalkar Shreyas Sanjay (32361 - K6)
Bhatkhedkar Aditi Makarand (32407 - K6)
Chavan Nikhil Nandkishor (32409 - K6)
Suryawanshi Komal Valmik (32432 - L6)
Kshirsagar Shrikant Uttam (32433 - L6)
Kulkarni Varad Yogesh (32435 - L6)
Londhe Ankita Shashikant (32438 - L6)
Wedhane Vedant Deepak (32490 - N6)

GUIDE
MR. NILESH SHIRUDE

DEPARTMENT OF
ELECTRONICS AND TELECOMMUNICATION ENGINEERING
PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE – 43

A.Y. 2023-24

INDEX

Sr. No.	Contents	Page No.
1	**Problem Statement**	3
2	**Objectives**	3
3	**Introduction**	4 - 7
4	**Flowchart and Code Link**	8
5	**Result**	9
6	**Conclusion**	10
7	**Applications**	10
8	**Future scope**	10
9	**Copyright Affirmation**	11

1. **PROBLEM STATEMENT:**

Managing student data efficiently in educational institutions is a challenging task. There's a need for a system that can organize student information based on their classes and subjects, while also calculating grades dynamically.

2. **OBJECTIVE :**

1. Develop a user-friendly interface for inputting and viewing student data.
2. Organize student information based on classes and subjects.
3. Dynamically calculate grades based on subject marks.
4. Implement event handling to navigate through different classes and subjects.
5. Ensure scalability and modularity for future enhancements.

3. <u>INTRODUCTION</u>:

In today's era of digitalization, the management of student data plays a crucial role in educational institutions. The Student Data Management System presented here addresses the need for an efficient and organized approach to handle student information and grade calculation. This introduction provides an in-depth overview of the project, covering its background, relevance, project details, inbuilt classes and components used, and the scope of the project.

3.1 Background/context:

Traditional methods of managing student data involve manual record-keeping, which is prone to errors, time-consuming, and lacks scalability. With the advent of technology, there's a growing demand for automated systems that can streamline the process of managing student information, making it more accurate, accessible, and efficient. The Student Data Management System leverages modern programming techniques and graphical user interfaces to fulfill this need.

3.2 Relevance:

The Student Data Management System is relevant in various educational settings, including schools, colleges, universities, and training institutes. It caters to administrators, teachers, and staff who are responsible for maintaining student records and assessing academic performance. By providing a centralized platform for data management and grade calculation, the system enhances productivity, accuracy, and transparency in educational institutions.

3.3 Project Details:

The Student Data Management System is developed using Java programming language and utilizes the Swing framework for creating graphical user interfaces. It features a tree-based navigation system, allowing users to organize student data by classes and subjects. The system dynamically calculates grades based on the marks obtained in different subjects, providing real-time feedback to users. Inbuilt classes and components such as JTree, JTable, DefaultMutableTreeNode, and DefaultTableModel are utilized to implement the core functionality of the system.

1. **Development Language:** Java programming language.
2. **GUI Framework:** Utilizes Swing framework for creating graphical user interfaces.
3. **Navigation System:** Implements a tree-based navigation system to organize student data by classes and subjects.
4. **Grade Calculation:** Dynamically calculates grades based on the marks obtained in different subjects, providing real-time feedback to users.
5. **Data Structure:** Utilizes inbuilt classes and components such as JTree, JTable, DefaultMutableTreeNode, and DefaultTableModel to implement core functionalities.

3.4 Inbuilt Classes and Components Used:

1. JTree:
- ➢ Purpose: Used to create a hierarchical representation of classes and subjects.
- ➢ Explanation: JTree provides a flexible and efficient way to organize data in a tree-like structure. Each node in the tree represents a class or subject, allowing users to navigate through different levels of hierarchy easily.

2. JTable:

➢ Purpose: Employed to display student data in tabular format, facilitating easy viewing and editing.

➢ Explanation: JTable is a versatile component for displaying and editing tabular data. It provides features such as sorting, filtering, and cell editing, enabling users to interact with student data effectively.

3. DefaultMutableTreeNode:

➢ Purpose: Utilized to construct the nodes of the tree structure.

➢ Explanation: DefaultMutableTreeNode is a class that represents a node in a tree structure. It allows for the creation of hierarchical relationships between nodes, enabling the construction of a tree-based navigation system for organizing student data.

4. DefaultTableModel:

➢ Purpose: Used to manage the data structure of tables, enabling dynamic updates and modifications.

➢ Explanation: DefaultTableModel provides a data model for JTable, allowing for the manipulation of tabular data. It supports operations such as adding, removing, and updating rows and columns, making it suitable for dynamically updating student data and grades.
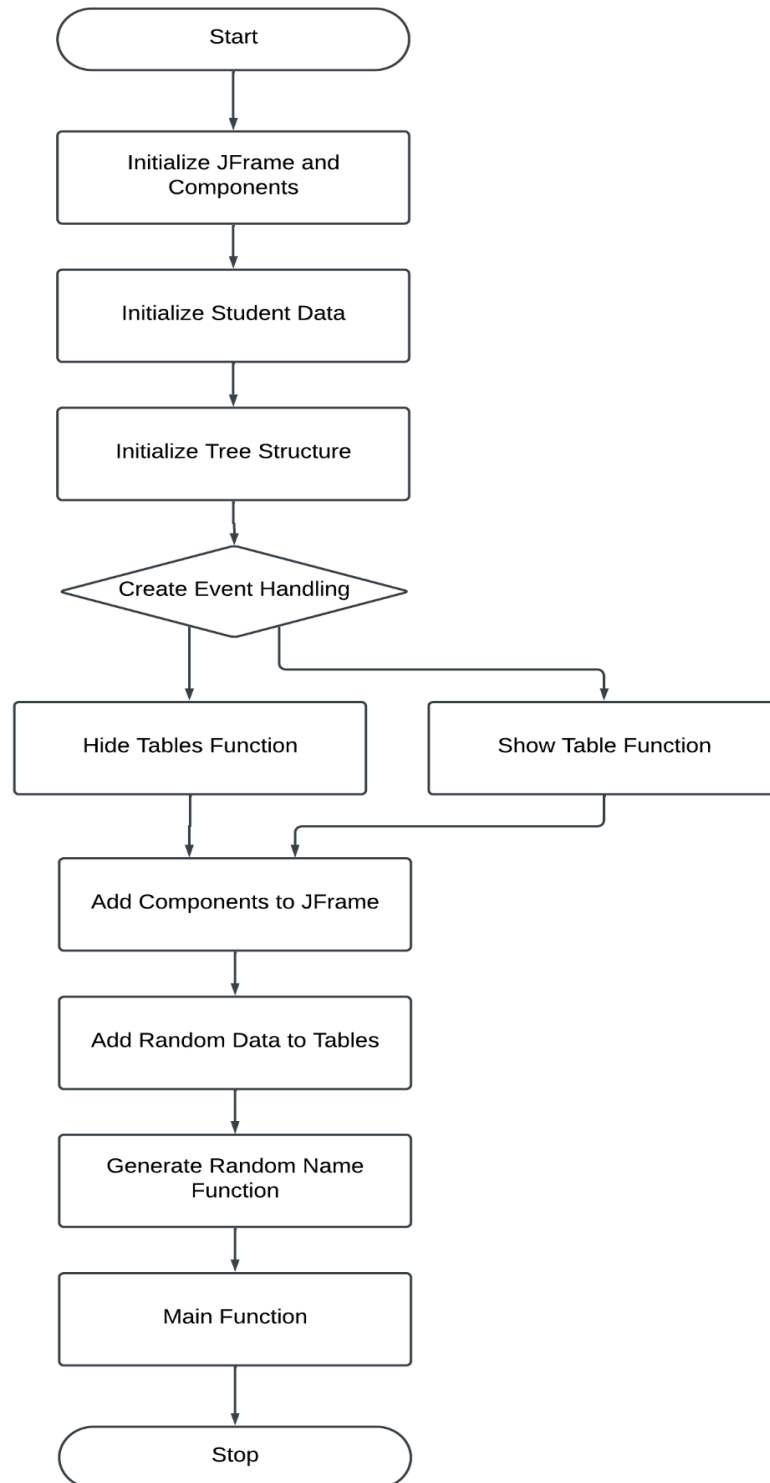
3.5 Scope:

The Student Data Management System serves as a foundation for future enhancements and extensions. Its scope includes:

> ➢ Integration with databases for persistent storage of student records, ensuring data integrity and scalability.
> ➢ Implementation of user authentication and access control mechanisms to ensure data security and privacy.
> ➢ Addition of features such as generating reports, conducting statistical analysis, and generating insights into student performance.
> ➢ Enhancement of the user interface with more interactive elements, customization options, and support for multiple languages.
> ➢ Integration with communication tools for sending notifications and alerts to students, parents, and faculty members regarding grades, attendance, and other relevant information.
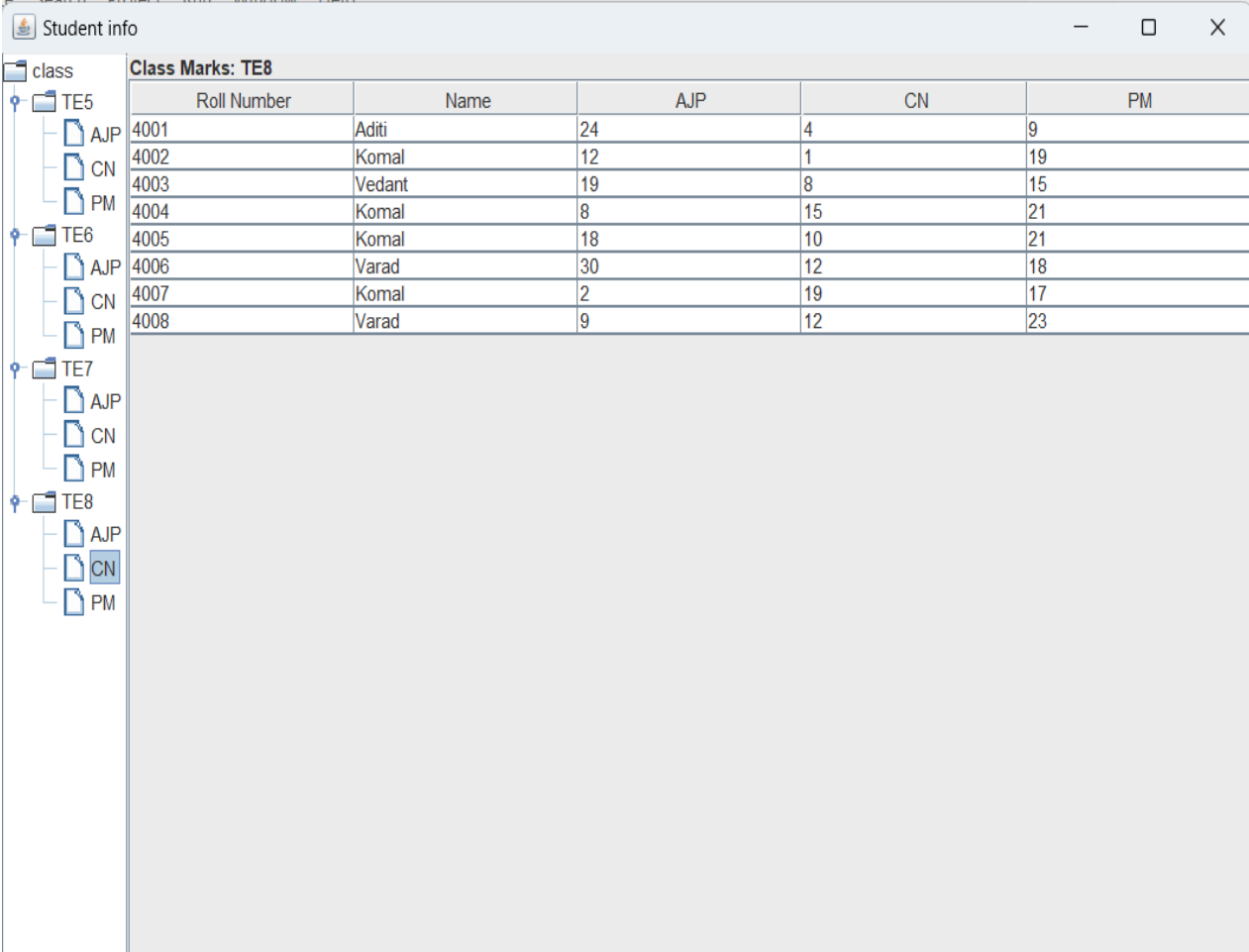
4. Flow Chart and SOURCE CODE:

GitHub: Click here for code link :- CODE

```
                    ╭─────────────────╮
                    │      Start      │
                    ╰─────────────────╯
                            │
                            ▼
                  ┌─────────────────────┐
                  │ Initialize JFrame and│
                  │     Components       │
                  └─────────────────────┘
                            │
                            ▼
                  ┌─────────────────────┐
                  │ Initialize Student Data│
                  └─────────────────────┘
                            │
                            ▼
                  ┌─────────────────────┐
                  │ Initialize Tree Structure│
                  └─────────────────────┘
                            │
                            ▼
                       ◇ Create Event Handling ◇
                       │              │
                       ▼              ▼
            ┌──────────────────┐  ┌──────────────────┐
            │ Hide Tables Function│  │ Show Table Function│
            └──────────────────┘  └──────────────────┘
                       │              │
                       ▼              ▼
                  ┌─────────────────────┐
                  │ Add Components to JFrame│
                  └─────────────────────┘
                            │
                            ▼
                  ┌─────────────────────┐
                  │ Add Random Data to Tables│
                  └─────────────────────┘
                            │
                            ▼
                  ┌─────────────────────┐
                  │ Generate Random Name │
                  │     Function         │
                  └─────────────────────┘
                            │
                            ▼
                  ┌─────────────────────┐
                  │    Main Function     │
                  └─────────────────────┘
                            │
                            ▼
                    ╭─────────────────╮
                    │      Stop       │
                    ╰─────────────────╯
```

5. RESULT:

The system successfully organizes student data and displays it in a structured manner. Users can navigate through different classes and subjects using a tree structure. Grades are calculated dynamically, providing instant feedback to users.

Student info window showing Class Marks: TE8

Roll Number	Name	AJP	CN	PM
4001	Aditi	24	4	9
4002	Komal	12	1	19
4003	Vedant	19	8	15
4004	Komal	8	15	21
4005	Komal	18	10	21
4006	Varad	30	12	18
4007	Komal	2	19	17
4008	Varad	9	12	23

6. CONCLUSION:

In conclusion, the Student Data Management System addresses the need for an efficient and organized approach to manage student data and grade calculation in educational institutions. It leverages modern programming techniques and graphical user interfaces to enhance productivity, accuracy, and transparency. With the potential for future enhancements and extensions, the system holds promise for transforming the way student data is managed and utilized in educational settings.

7. APPLICATIONS:

1. Educational institutions such as schools, colleges, and universities.
2. Training centers and coaching institutes.
3. Online learning platforms.
4. Academic departments within organizations.

8. FUTURE SCOPE:

1. Integration with databases for persistent storage of student records.
2. Implementation of user authentication and access control.
3. Addition of features such as generating reports and statistical analysis of student performance.
4. Enhancement of the user interface with more interactive elements and customization options.
5. Integration with communication tools for notifying students and parents about grades and performance updates.

9. COPYRIGHT AFFIRMATION:

We undersigned pledge and represent that the source code printed in this project report does not violate any proprietary or personal rights of others (including, without limitation, any copyrights or privacy rights); that the Work is factually accurate and contains no matter libelous or otherwise unlawful; that we have substantially participated in the creation of the work and that it represents our original work sufficient for us to claim authorship.

Name of students   **Sign**

1. **Shewalkar Shreyas Sanjay (32361 - K6)**
2. **Bhatkhedkar Aditi Makarand (32407 - K6)**
3. **Chavan Nikhil Nandkishor (32409 - K6)**
4. **Suryawanshi Komal Valmik (32432 - L6)**
5. **Kshirsagar Shrikant Uttam (32433 - L6)**
6. **Kulkarni Varad Yogesh (32435 - L6)**
7. **Londhe Ankita Shashikant (32438 - L6)**
8. **Wedhane Vedant Deepak (32490 - N6)**