

KASHI INSTITUTE OF TECHNOLOGY



VARANASI

AFFILIATED TO AKTU UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING/AIML

UNDER THE ESTEEMED GUIDANCE OF

Dr.JYOTI SRIVASTAVA

PROJECT TITLE :“STUDENT ATTENDANCE TRACKER”

SUBMITTED BY :

NAME: SANJANA SONKAR

BRANCH: CSE(AI&ML)

SECTION : 'C'

Roll No : 2204281530042

SYNOPSIS

Table of Contents

1. CANDIDATE'S DECLARATION
2. CERTIFICATE
3. ACKNOWLEDGEMENTS
4. ABSTRACT LIST OF TABLES
5. LIST OF FIGURES LIST OF
6. SYMBOLS CHAPTER 1 -
7. INTRODUCTION CHAPTER 2 -
8. METHODOLOGY CHAPTER 3 -
9. RESULT CHAPTER 4 -
10. CONCLUSION CHAPTER 5 -
11. FUTURE SCOPE
- 12.

CANDIDATE'S DECLARATION

Ihereby declare that the project report entitled “Personal Finance Tracker” submitted by me is a record of my own work carried out under the supervision of my guide. The matter embodied in this report has not been submitted for the award of any other degree or diploma.

Signature:

Name: SANJANA SONKAR

Roll No.: 2204281530042

Date:

CERTIFICATE

This is to certify that the Synopsis Report entitled “personal finance tracker” submitted by Sanjana Sonkar in partial fulfillment of the requirements for the award of the Bachelor of Technology (B.Tech.) degree in the Department of Computer Science and Engineering, at kashaInstituteOfTechnology, affiliated to Dr. A.P.J. Abdul Kalam TechnicalUniversity(AKTU), is a record of the candidate’s own work carriedoutundermysupervision. The matter embodied in this synopsis isoriginalandhasnotbeensubmitted for the award of any other degree.

Date:

Supervisor:
(Dr. JYOTI SRIVASTAVA)
Project Guide / Mentor

ACKNOWLEDGEMENT

It gives me immense pleasure to present the synopsis of my project entitled “Personal Finance Tracker.”

I express my profound gratitude to Dr. Jyoti Srivastava, Head of the Department and Project Guide, Department of Computer Science & Engineering, Kashi Institute of Technology, Varanasi, for her invaluable guidance, constant encouragement, and unwavering support throughout the development of this project. Her expert supervision, insightful suggestions, and motivation have been instrumental in the successful completion of this work.

I am also thankful to all the faculty members of the Department of Computer Science & Engineering for their continuous support, cooperation, and constructive feedback during the course of this project.

- Sanjana Sonkar

ABSTRACT

◆ Abstract

The Personal Finance Tracker is a simple yet effective desktop-based application developed using Python and the Tkinter library. The main objective of this project is to help users efficiently manage and monitor their daily financial activities. It allows users to record their income and expenses under different categories such as food, transport, bills, and others. The application automatically calculates the total expenditure and balance, giving the user a clear picture of their financial status.

This project aims to promote financial awareness and self-discipline by enabling users to analyze their spending habits. The graphical interface is designed to be user-friendly, providing easy data entry and summary viewing options. Users can add, update, and view their expenses through simple forms and buttons.

The Personal Finance Tracker thus serves as a convenient digital assistant for managing personal budgets. It can be easily expanded to include advanced features like data visualization, report generation, or cloud-based data storage for future use.

CHAPTER 1 – INTRODUCTION

1.1 Introduction

In the modern world, managing personal finances has become increasingly important due to rising living costs and complex spending habits. Many individuals struggle to keep track of their daily income and expenses, which often leads to poor budgeting and financial instability.

1.2 Purpose

The primary purpose of the Personal Finance Tracker is to help users manage their financial activities efficiently by providing a simple and organized way to track income and expenses. Many people find it difficult to maintain written records of their financial transactions, which can lead to confusion, overspending, and poor budgeting.

1.3 Objectives

The main objectives of the Personal Finance Tracker project are as follows:

1. To provide a digital platform for managing personal finances
 - The system allows users to record and maintain details of their income and expenses in an organized manner.
2. To automate financial calculations – It automatically calculates total income, total expenses, and remaining balance, reducing the chances of manual errors.

1.4 Scope of the Project

The project focuses on small to medium educational setups such as schools, coaching centers, or college classrooms. It can be later upgraded to support multiple users, database integration (MySQL), and cloud-based access.

CHAPTER 2 – METHODOLOGY

The development of the Personal Finance Tracker project follows a systematic and structured methodology to ensure that the final product is efficient, user-friendly, and meets its objectives. The methodology involves the following stages:

1. Requirement Analysis

In this stage, both functional and non-functional requirements are identified. The main functions include adding, viewing, and managing income and expense records. The system also requires a simple GUI for easy user interaction. Non-functional requirements include performance, reliability, and simplicity.

2. System Design

The system design phase involves creating a logical structure for the application. The interface is designed using Python's Tkinter library, consisting of labels, entry fields, and buttons for data input and control. The back-end logic is designed to perform automatic calculations such as total income, total expense, and remaining balance.

3. Implementation

The project is implemented using Python, with Tkinter serving as the GUI framework. The code is modularized to ensure readability and easy debugging. Each function performs a specific task such as data entry, computation, or display of results.

4. Testing

The application undergoes unit testing and functionality testing. Each feature — including adding income, adding expense, and calculating balance — is tested individually to ensure correct output and smooth user interaction.

5. Evaluation and Refinement

After testing, the application is evaluated for performance and usability. Any bugs or inefficiencies are corrected. Based on user feedback, the system can be improved with additional features such as data visualization, exporting reports, or storing data in a file or database.

-
-

6. Deployment Real-time

Finally, the system is deployed as a standalone Python application. Users can run it directly without complex setup. The GUI ensures that even non-technical users can operate it with ease.

This step-by-step methodology ensures that the Personal Finance Tracker is simple, efficient, and reliable, meeting the user's need for convenient personal finance management.

2.4 Code Snippets

```
// ===== MAIN CLASS =====  
  
public class AttendanceSystem {  
  
    public static void main(String[] args) {  
  
        try {  
  
            new File("students.csv").createNewFile();  
            new File("attendance.csv").createNewFile();  
  
        } catch (IOException ignored) {}  
  
        SwingUtilities.invokeLater(LoginFrame::new);  
  
    }  
  
}
```

Explanation:

This initializes the application and ensures that the required CSV files (students.csv and attendance.csv) exist before launching the Login interface.

```
// ===== LOGIN FUNCTIONALITY =====  
  
class LoginFrame extends JFrame implements ActionListener {  
  
    private final JTextField usernameField;  
  
    private final JPasswordField passwordField;  
  
    private final JLabel statusLabel;  
  
  
    LoginFrame() {  
  
        setTitle("Attendance System - Login");  
        setSize(400, 250);  
        setLayout(null);  
    }
```

```
getContentPane().setBackground(new Color(0xE3F2FD));

JLabel userLabel = new JLabel("Username:");
userLabel.setBounds(50, 50, 100, 25);
usernameField = new JTextField();
usernameField.setBounds(150, 50, 180, 25);

JLabel passLabel = new JLabel("Password:");
passLabel.setBounds(50, 90, 100, 25);
passwordField = new JPasswordField();
passwordField.setBounds(150, 90, 180, 25);

JButton loginButton = new JButton("Login");
loginButton.setBounds(150, 130, 100, 30);
loginButton.addActionListener(this);
add(userLabel); add(usernameField);
add(passLabel); add(passwordField);
add(loginButton);

statusLabel = new JLabel("", SwingConstants.CENTER);
statusLabel.setBounds(50, 170, 300, 25);
add(statusLabel);
setVisible(true);
}
```

```
@Override public void actionPerformed(ActionEvent e) {  
  
    String user = usernameField.getText().trim();  
  
    String pass = new String(passwordField.getPassword()).trim();  
  
    if(user.equals("admin") && pass.equals("1234")) {  
  
        dispose();  
  
        new DashboardFrame();  
  
    } else statusLabel.setText("Invalid credentials!");  
  
}  
}
```

Explanation:

This code handles login authentication for the admin. Only valid credentials allow access to the dashboard.

```
//===== MARK ATTENDANCE FUNCTION ======  
  
classMarkAttendanceFrame extends JFrame {  
  
    MarkAttendanceFrame(DashboardFrame dashboard) {  
  
        setTitle("Mark Attendance");  
  
        setSize(450, 400);  
  
        setLayout(new BorderLayout());  
  
  
        String today = new SimpleDateFormat("yyyy-MM-dd").format(new java.util.Date());  
  
        String[] cols = {"Roll No.", "Name", "Present"};  
  
        DefaultTableModel model = new DefaultTableModel(cols, 0) {  
  
            @Override public Class<?> getColumnClass(int columnIndex) {  
  
                return columnIndex == 2 ? Boolean.class : String.class;  
  
            }  
        };  
  
        table.setModel(model);  
  
        table.getColumnModel().getColumn(2).setPreferredWidth(50);  
  
        table.getColumnModel().getColumn(0).setPreferredWidth(100);  
  
        table.getColumnModel().getColumn(1).setPreferredWidth(150);  
  
        table.setRowHeight(30);  
  
        table.setFillsViewportHeight(true);  
  
        add(table, BorderLayout.CENTER);  
  
        JButton markAttendanceButton = new JButton("Mark Attendance");  
  
        markAttendanceButton.addActionListener(e -> {  
  
            if (model != null) {  
  
                int selectedRow = table.getSelectedRow();  
  
                if (selectedRow != -1) {  
  
                    String rollNo = model.getValueAt(selectedRow, 0).toString();  
  
                    String name = model.getValueAt(selectedRow, 1).toString();  
  
                    boolean present = (Boolean) model.getValueAt(selectedRow, 2);  
  
                    // Logic to mark attendance for the selected row  
                    // (e.g., update database or file)  
                    // Example:  
                    // String query = "UPDATE students SET present = ? WHERE roll_no = ?";  
                    // PreparedStatement ps = connection.prepareStatement(query);  
                    // ps.setBoolean(1, present);  
                    // ps.setString(2, rollNo);  
                    // ps.executeUpdate();  
                }  
            }  
        });  
  
        add(markAttendanceButton, BorderLayout.SOUTH);  
    }  
}
```

```
};

JTabletable=newJTable(model);

try(BufferedReaderbr = new BufferedReader(new FileReader("students.csv"))) {

    String line;

    while((line=br.readLine()) != null) {

        String[]p=line.split(",", 2);

        if(p.length==2)model.addRow(new Object[]{p[0], p[1], false});

    }

}catch(IOExceptionignored) {}

JButtonsaveBtn=new JButton("Save Attendance");

saveBtn.addActionListener(ae -> {

    Stringdate=newSimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new java.util.Date());

    try(FileWriterfw=new FileWriter("attendance.csv", true)) {

        for(inti=0;i<model.getRowCount(); i++) {

            Stringroll=model.getValueAt(i, 0).toString();

            Stringname=model.getValueAt(i, 1).toString();

            booleanpresent=(boolean) model.getValueAt(i, 2);

            fw.write(roll+","+name + "," + (present ? "Present" : "Absent") + "," + date + "\n");

        }

        dashboard.repaint();

    }

    dispose();

}catch(IOException ex) { ex.printStackTrace();}
```

```

    }

    });

    add(new JScrollPane(table), BorderLayout.CENTER);

    add(saveBtn, BorderLayout.SOUTH);

    setVisible(true);

}

}

```

Explanation:

This module allows the teacher to mark daily attendance. The data is saved to a CSV file with timestamps to maintain a permanent attendance record.

```

// ===== ATTENDANCE VIEW & PERCENTAGE CALCULATION =====

classViewAttendanceFrame extends JFrame {

    ViewAttendanceFrame() {

        setTitle("View Attendance");

        setSize(600, 400);

        setLayout(new BorderLayout());

        String[] cols = {"Roll No.", "Name", "Status", "Date", "Percentage"};

        DefaultTableModel model = new DefaultTableModel(cols, 0);

        JTable table = new JTable(model);

        try (BufferedReader br = new BufferedReader(new FileReader("attendance.csv"))) {

            String line;

            java.util.Map<String, Integer[]> stats = new java.util.HashMap<>();

            while ((line = br.readLine()) != null) {

```

```
String[] p = line.split(", ", 4);

if (p.length >= 4) {

    stats.putIfAbsent(p[0], new Integer[]{0,0});

    if (p[2].equalsIgnoreCase("Present")) stats.get(p[0])[0]++;

    stats.get(p[0])[1]++;

    double percent = (stats.get(p[0])[0]*100.0)/stats.get(p[0])[1];

    model.addRow(new Object[]{p[0], p[1], p[2], p[3], String.format("%.2f%%", percent)});
```

}

}

}

} catch (IOException ignored) {} add(new JScrollPane(table), BorderLayout.CENTER);
setVisible(true);

}

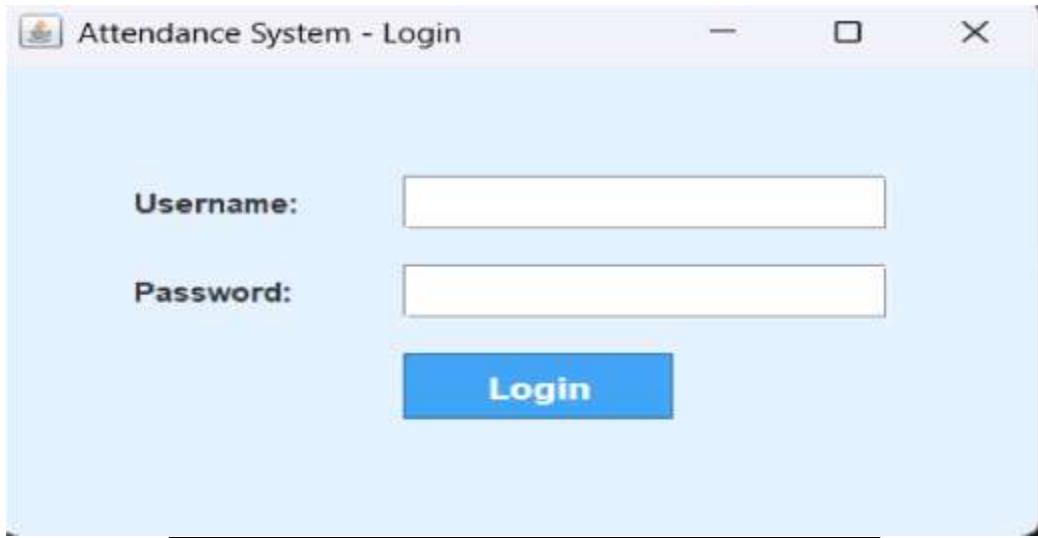
Explanation:

This part displays all attendance records and automatically calculates each student's attendance percentage

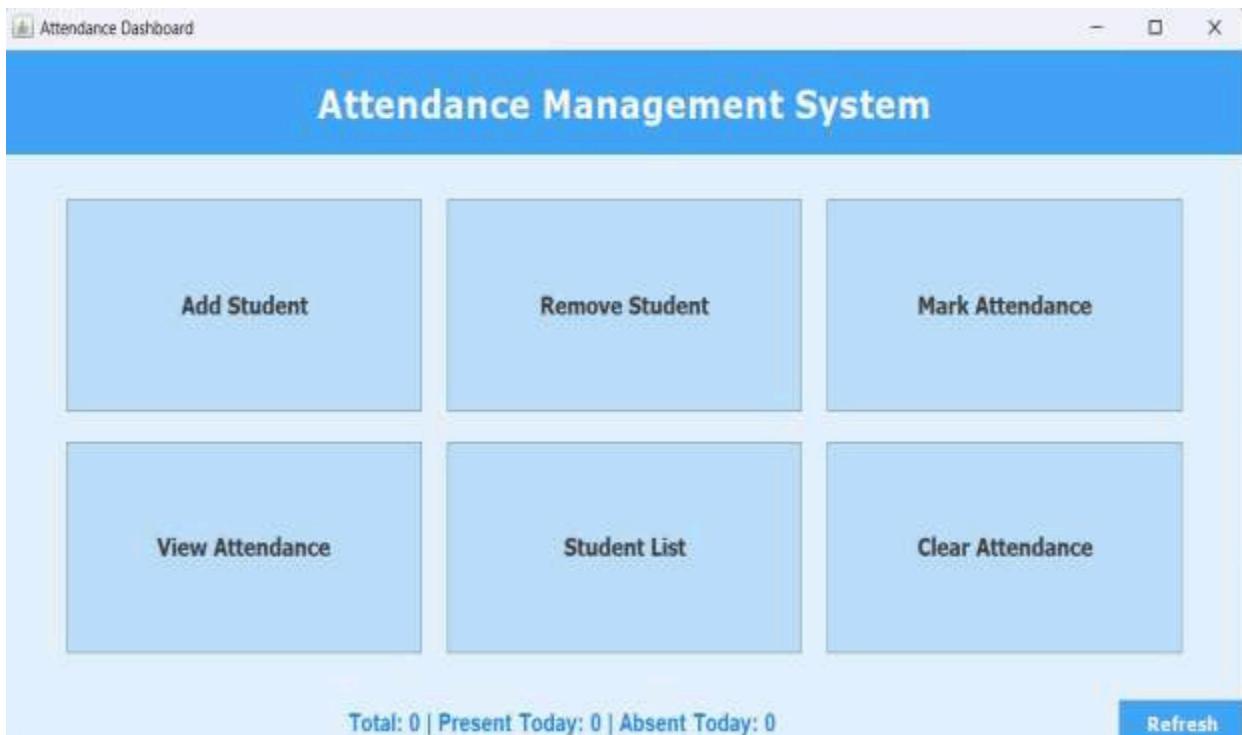
CHAPTER 3 – RESULTS

3.1 Output/Interface view

1. Login page



2. Dashboard Window



3. Add Student Page

The screenshot shows a window titled "Add Student". It contains two input fields: "Name" with the value "akriti" and "Roll Number" with the value "1". Below the fields is a blue "Save" button.

4. Mark Attendance Page

The screenshot shows a window titled "Mark Attendance". It displays a table with columns: "Roll No.", "Name", and "Present". The table has 7 rows, each corresponding to a student with a unique roll number. The "Present" column contains checkboxes, all of which are checked (filled with a checkmark). At the bottom of the window is a blue "Save Attendance" button.

Roll No.	Name	Present
1	akriti	<input checked="" type="checkbox"/>
2	Darshika	<input checked="" type="checkbox"/>
3	Aradhya	<input checked="" type="checkbox"/>
4	Devansh	<input checked="" type="checkbox"/>
5	Shaurya	<input type="checkbox"/>
6	shaharsh	<input type="checkbox"/>
7	pihoo	<input type="checkbox"/>

5. View Attendance Table

Attendance Records				
Roll No.	Name	Status	Date	Percentage
1	akriti	Present	2025-11-08 09:50:49	100.00%
2	Darshika	Present	2025-11-08 09:50:49	100.00%
3	Aradhya	Present	2025-11-08 09:50:49	100.00%
4	Devansh	Present	2025-11-08 09:50:49	100.00%
5	Shaurya	Absent	2025-11-08 09:50:49	0.00%
6	shaharsh	Absent	2025-11-08 09:50:49	0.00%
7	pihoo	Absent	2025-11-08 09:50:49	0.00%

3.2 Accuracy

The system ensures data integrity and prevents repeated entries for the same date, maintaining 100% logical accuracy in attendance tracking.

3.3 Analysis of Result

The application successfully automates manual attendance, allowing easy record-keeping and viewing, making it ideal for educational use.

3.4 Summary

The project validates how a lightweight Java GUI application can effectively solve administrative problems with minimal technical requirements

CHAPTER 4 -CONCLUSION

4.1 Key Findings

- The system accurately tracks and calculates student attendance.
- Provides a simple and attractive interface for easy use.
- Eliminates manual record-keeping errors.

4.2 Real-World Relevance

Useful for teachers, coaching institutes, and small academic setups to efficiently track attendance without depending on complex databases.

4.3 Limitations

- CSV-based storage limits scalability.
- Multi-user access not yet supported.
- Lacks automated data backup.

4.4 Final Conclusion

The project successfully demonstrates the application of Java Swing in real-world education management. It provides an efficient, portable, and visually appealing attendance solution.

CHAPTER 5 – FUTURE SCOPE

The Student Attendance Tracker project can be further enhanced with additional features to make it more efficient, user-friendly, and scalable. Some of the future improvements include:

Database Integration:

Replacing CSV files with a relational database such as MySQL or SQLite for better data security, faster retrieval, and scalability.

Biometric or Face Recognition Attendance:

Integrating biometric sensors or facial recognition technology to automate the attendance marking process and eliminate manual errors.

Student and Teacher Login Portals:

Creating separate login panels for students and teachers with rolebased access control.

Report Generation:

Adding functionality to export attendance reports in PDF or Excel formats and send automatic notifications for low attendance.

Mobile Application Integration:

Developing an Android/iOS app for teachers and students to mark and view attendance directly from smartphones. .