

EXPENSE TRACKER APPLICATION



A PROJECT REPORT

Submitted by

SHAHUL HAMEED M (2303811724321101)

in partial fulfillment of requirements for the award of the course

CGB1201 – JAVA PROGRAMMING

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by
AICTE, New Delhi)

SAMAYAPURAM – 621 112

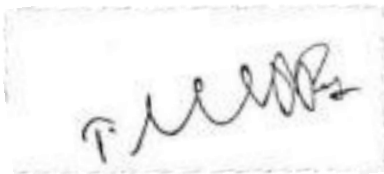
DECEMBER, 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on “ **EXPENSE TRACKER APPLICATION**” is the bonafide work of **SHAHUL HAMEED M (2303811724321101)** who carried out the project work during the academic year 2024 - 2025 under my supervision.



Signature

Dr. T. AVUDAIAPPAN M.E.,Ph.D.,

HEAD OF THE DEPARTMENT,

Department of Artificial Intelligence,
K. Ramakrishnan College of Engineering,
Samayapuram, Trichy -621 112.



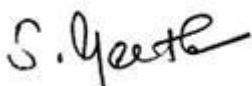
Signature

Mrs. S. GEETHA M.E.,

SUPERVISOR,

Department of Artificial Intelligence,
K. Ramakrishnan College of Engineering,
Samayapuram, Trichy -621 112.

Submitted for the viva-voce examination held on 3.12.24



INTERNAL EXAMINER



EXTERNAL EXAMINER

DECLARATION

I declare that the project report on “**EXPENSE TRACKER APPLICATION**” is the result of original work done by me and best of my knowledge, similar work has not been submitted to “**ANNA UNIVERSITY CHENNAI**” for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfillment of the requirement of the award of the **CGB1201 – JAVA PROGRAMMING**.



Signature

SHAHUL HAMEED M

Place: Samayapuram

Date: 3/12/2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and indebtedness to our institution, **“K. Ramakrishnan College of Technology (Autonomous)”**, for providing us with the opportunity to do this project.

I extend our sincere acknowledgement and appreciation to the esteemed and honourable Chairman, **Dr. K. RAMAKRISHNAN, B.E.**, for having provided the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director, **Dr. S. KUPPUSAMY, MBA, Ph.D.**, for forwarding our project and offering an adequate duration to complete it.

I would like to thank **Dr. N. VASUDEVAN, M.TECH., Ph.D.**, Principal, who gave the opportunity to frame the project to full satisfaction.

I thank **Dr.T.AVUDAIAPPAN, M.E.,Ph.D.**, Head of the Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for providing her encouragement in pursuing this project.

I wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mrs.S.GEETHA M.E.**, Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for her incalculable suggestions, creativity, assistance and patience, which motivated us to carry out this project.

I render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards.

MISSION OF THE INSTITUTION

- Be a centre of excellence for technical education in emerging technologies by exceeding the needs of industry and society.
- Be an institute with world class research facilities.
- Be an institute nurturing talent and enhancing competency of students to transform them as all- round personalities respecting moral and ethical values.

VISION AND MISSION OF THE DEPARTMENT

To excel in education, innovation and research in Artificial Intelligence and Data Science to fulfill industrial demands and societal expectations.

Mission 1: To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.

Mission 2: To collaborate with industry and offer top-notch facilities in a conducive learning environment.

Mission 3: To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.

Mission 4: To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

PEO 1: Compete on a global scale for a professional career in Artificial Intelligence and Data Science.

PEO 2: Provide industry-specific solutions for the society with effective communication and ethics.

PEO 3: Hone their professional skills through research and lifelong learning initiatives.

PROGRAM OUTCOMES

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO 1:** Capable of working on data-related methodologies and providing industry-focussed solutions.
- **PSO2:** Capable of analysing and providing a solution to a given real-world problem by designing an effective program.

ABSTRACT

The Expense Tracker Application is a user-friendly Java-based solution designed to help individuals efficiently manage their daily expenses and budgets. Built using Java Swing, the application offers a seamless interface for users to record, categorize, and monitor their financial transactions. Users can add expenses under predefined categories, such as Food, Transport, Shopping, Bills, and Miscellaneous, along with specifying the amount and date. A budget-setting feature enables users to define a financial limit, allowing the system to calculate and display total expenses and the remaining budget dynamically. The expense summary is presented in a tabular format, providing a clear and organized overview of spending patterns. With the goal of promoting financial discipline, the application helps users identify areas of overspending, thereby encouraging better money management and savings. The project emphasizes key programming concepts such as object-oriented design, event handling, and GUI development, making it a practical and educational implementation for tracking personal finances. Future enhancements may include advanced analytics, graphical insights, and integration with cloud services for improved scalability and accessibility. This project demonstrates the practical application of Java programming to solve everyday problems, offering an effective tool for financial awareness and management.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	viii
1	INTRODUCTION	1
	1.1 OBJECTIVE	1
	1.2 OVERVIEW	1
2	PROJECT METHODOLOGY	2
	2.1 PROPOSED WORK	2
	2.2 BLOCK DIAGRAM	3
3	JAVA PROGRAMMING CONCEPTS	4
	3.1 OBJECT-ORIENTED PROGRAMMING (OOP)	4
	3.2 SWING FRAMEWORK	4
	3.3 COLLECTIONS FRAMEWORK	4
4	MODULE DESCRIPTION	5
	4.1 USER INPUT MODULE	5
	4.2 EXPENSE MANAGEMENT MODULE	5
	4.3 BUDGET TRACKING MODULE	5
	4.4 EXPENSE SUMMARY MODULE	6
5	CONCLUSION	7
	REFERENCES	8
	APPENDICES	9
	APPENDIX-A SOURCE CODE	9
	APPENDIX-B SCREEN SHOT	15

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The Expense Tracker application is a desktop software solution developed using Java and designed to streamline personal financial management. The application features a user-friendly graphical interface created with Java Swing, allowing users to input expense details such as category, amount, and date. It also enables users to set a financial budget, which is dynamically updated based on their recorded expenses. The application processes these inputs and displays the results in a well-organized summary, showing total expenses, categorized spending, and the remaining budget.

1.2 OBJECTIVE

The primary objective of the Expense Tracker application is to provide users with a comprehensive tool for managing their finances effectively and conveniently. It is designed to allow users to record their daily expenses, categorize them based on different spending areas (e.g., food, transport, shopping), and set financial budgets. By tracking spending in real time, the application ensures that users are aware of their financial status at any given moment. The app also aims to promote better financial habits by helping users analyze their spending patterns, control unnecessary expenditures, and ultimately save money. Through its intuitive interface, the application seeks to make personal finance management accessible and efficient for users of all technical backgrounds.

CHAPTER 2

PROJECT METHODOLOGY

2.1 PROPOSED WORK

User Interface (UI):

The user interacts with the application through a graphical interface where they can input expense details such as category, amount, date, and set budgets. The UI is built using Java Swing, offering an intuitive experience for the user.

Input Handling:

When the user enters the expense data and clicks on the "Add Expense" button, the system processes this input. It validates the entered details and stores the expense in a manageable format for further use.

Expense Calculation & Budgeting:

The application calculates the total expenses based on user inputs, compares them against the budget, and updates the financial summary. The system allows users to set and track their budget, displaying the remaining balance as they add new expenses.

Display & Reporting:

The processed information is displayed on the screen, showing the user a categorized summary of their expenses, total expenses, and the remaining budget. The interface updates in real-time as new expenses are added, giving the user a clear view of their financial standing.

2.2 BLOCK DIAGRAM

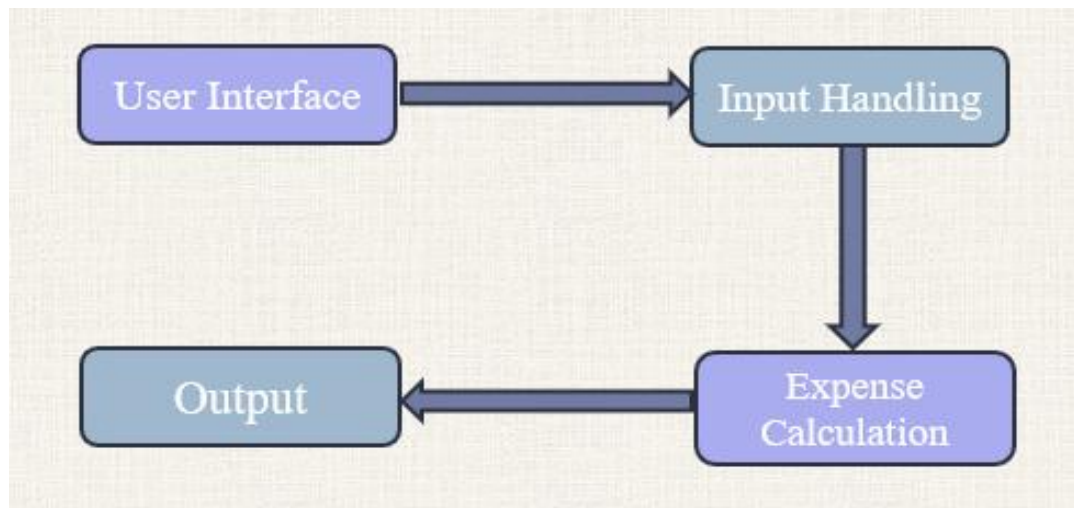


Fig 2.1 Block Diagram

CHAPTER 3

JAVA PROGRAMMING CONCEPTS

3.1 OBJECT-ORIENTED PROGRAMMING (OOP)

- **Encapsulation:** The Expense class encapsulates expense details (category, amount, date) to maintain data integrity and ensure modularity.
- **Abstraction:** Complex functionalities like adding expenses and updating summaries are abstracted into methods for ease of use.
- **Reusability:** Classes and methods are reusable, enabling efficient expansion of the application's features without redundant code.

3.2 SWING FRAMEWORK

- **Graphical Interface:** Components like JFrame, JTextField, JButton, and JLabel are used to create a user-friendly and interactive UI.
- **Event Handling:** Listeners (ActionListener) manage user actions, such as button clicks, for a seamless interaction experience.
- **Layout Management:** Layouts like BorderLayout and GridLayout ensure organized and responsive placement of UI elements.

4.3 COLLECTIONS FRAMEWORK

- **Dynamic Storage:** An ArrayList is used to store and manage expense records efficiently, allowing dynamic addition and retrieval of data.
- **Iterative Processing:** The ArrayList enables iteration through stored expenses to calculate totals and display summaries.
- **Flexibility:** The framework provides scalability, ensuring the application can handle varying amounts of expense data effortlessly.

CHAPTER 4

MODULE DESCRIPTION

4.1 USER INPUT MODULE

Description: This module captures user inputs such as category, amount, date, and budget through the graphical user interface.

Key Points:

- Provides input fields for expenses and budget details.
- Validates the inputs for correctness (e.g., non-empty, valid amount).
- Enables users to easily interact with the application.

4.2 EXPENSE MANAGEMENT MODULE

Description: This module processes and manages the expense records added by the user.

Key Points:

- Adds new expenses with category, amount, and date details to the system.
- Stores the expenses dynamically in a structured format (ArrayList).
- Maintains a categorized list of all expenses for further analysis.

4.3 BUDGET TRACKING MODULE

Description: This module manages the user-defined budget and tracks expenses against it.

Key Points:

- Allows users to set and update their financial budget.
- Calculates the total expenses and the remaining budget dynamically.

- Alerts users if expenses exceed the set budget.

4.4 EXPENSE SUMMARY MODULE

Description: This module generates and displays the summarized expense data for the user.

Key Points:

- Displays expenses in a tabular format with columns like category, amount, and date.
- Provides insights into total expenses and remaining budget in real-time.
- Ensures clarity through an organized and user-friendly summary layout.

CHAPTER 5

CONCLUSION

The Expense Tracker application successfully fulfills its intended purpose of helping users record, categorize, and monitor their daily expenses. By providing real-time updates on total expenses and remaining budgets, the application empowers users to manage their finances effectively. This project demonstrates the practical application of key Java concepts such as Object-Oriented Programming, Swing for GUI design, and the Collections framework. These technologies were seamlessly integrated to deliver a robust and user-friendly application. With features like categorized expense tracking, budget setting, and real-time financial summaries, the application provides valuable insights into spending habits. It helps users cultivate better financial practices, avoid overspending, and promote savings. The project has significant potential for future enhancements, such as persistent storage (database integration), advanced analytics for spending trends, and mobile platform compatibility. These improvements can make the application even more versatile and impactful for users. The Expense Tracker project not only serves as a reliable tool for personal finance management but also highlights the importance of using technology to simplify everyday tasks.

REFERENCES:

Books:

- *Java: The Complete Reference* by Herbert Schildt – A comprehensive guide to Java programming, including Swing and collections.
- *Head First Java* by Kathy Sierra and Bert Bates – An easy-to-understand book for learning Java concepts interactively.

Websites:

- [Oracle Java Documentation](#) – The official reference for Java programming and Swing framework.
- [GeeksforGeeks](#) – A resourceful website for Java tutorials, examples, and explanations.
- W3Schools - Java – An excellent platform for Java beginners to learn concepts with practical examples.

APPENDICES

APPENDIX-A SOURCE CODE

```
import javax.swing.*;
import javax.swing.border.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.util.ArrayList;

public class ExpenseTracker extends JFrame {
    private ArrayList<Expense> expenses = new ArrayList<>();
    private double totalBudget = 0;

    // Components
    private JTextField amountField, dateField, budgetField;
    private JComboBox<String> categoryComboBox;
    private JTable expenseTable;
    private DefaultTableModel tableModel;
    private JLabel totalExpensesLabel, balanceLabel;

    public ExpenseTracker() {
        setTitle("Expense Tracker");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(600, 500);
        setLayout(new BorderLayout());

        // Main Panel
        JPanel mainPanel = new JPanel(new BorderLayout());
```

```

mainPanel.setBorder(new EmptyBorder(10, 10, 10, 10));
mainPanel.setBackground(new Color(240, 248, 255));

// Input Panel
JPanel inputPanel = new JPanel(new GridLayout(5, 2, 10, 10));
inputPanel.setBorder(new TitledBorder("Add Expense"));

inputPanel.add(new JLabel("Category:"));
categoryComboBox = new JComboBox<>(new String[]{"Food",
"Transport", "Shopping", "Bills", "Miscellaneous"});
inputPanel.add(categoryComboBox);

inputPanel.add(new JLabel("Amount (₹):"));
amountField = new JTextField();
inputPanel.add(amountField);

inputPanel.add(new JLabel("Date (DD/MM/YYYY):"));
dateField = new JTextField();
inputPanel.add(dateField);

inputPanel.add(new JLabel("Set Budget (₹):"));
budgetField = new JTextField();
inputPanel.add(budgetField);

JButton addExpenseButton = new JButton("Add Expense");
JButton setBudgetButton = new JButton("Set Budget");

inputPanel.add(addExpenseButton);
inputPanel.add(setBudgetButton);

```

```

// Output Panel
JPanel outputPanel = new JPanel(new BorderLayout());
outputPanel.setBorder(new TitledBorder("Expense Summary"));

String[] columnNames = {"Category", "Amount (₹)", "Date"};
tableModel = new DefaultTableModel(columnNames, 0);
expenseTable = new JTable(tableModel);
JScrollPane tableScrollPane = new JScrollPane(expenseTable);
outputPanel.add(tableScrollPane, BorderLayout.CENTER);

// Summary Panel
JPanel summaryPanel = new JPanel(new GridLayout(2, 1));
totalExpensesLabel = new JLabel("Total Expenses: ₹0");
balanceLabel = new JLabel("Remaining Budget: ₹0");
summaryPanel.add(totalExpensesLabel);
summaryPanel.add(balanceLabel);
summaryPanel.setBackground(new Color(245, 255, 250));

// Add Panels
mainPanel.add(inputPanel, BorderLayout.NORTH);
mainPanel.add(outputPanel, BorderLayout.CENTER);
mainPanel.add(summaryPanel, BorderLayout.SOUTH);

add(mainPanel);

// Event Listeners
addExpenseButton.addActionListener(e -> addExpense());
setBudgetButton.addActionListener(e -> setBudget());

```

```

        setVisible(true);
    }

    private void addExpense() {
        try {
            String category = (String) categoryComboBox.getSelectedItem();
            double amount = Double.parseDouble(amountField.getText());
            String date = dateField.getText();

            if (amount <= 0 || date.isEmpty()) {
                JOptionPane.showMessageDialog(this, "Please enter valid
details!", "Error", JOptionPane.ERROR_MESSAGE);
                return;
            }

            expenses.add(new Expense(category, amount, date));
            updateSummary();
            addExpenseToTable(category, amount, date);

            amountField.setText("");
            dateField.setText("");
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(this, "Invalid amount!", "Error",
JOptionPane.ERROR_MESSAGE);
        }
    }

    private void addExpenseToTable(String category, double amount, String

```

```

date) {
    tableModel.addRow(new Object[] {category, "₹" + String.format("%.2f",
amount), date});
}

private void setBudget() {
    try {
        totalBudget = Double.parseDouble(budgetField.getText());
        updateSummary();
        budgetField.setText("");
    } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(this, "Invalid budget amount!",
"Error", JOptionPane.ERROR_MESSAGE);
    }
}

private void updateSummary() {
    double totalExpenses =
expenses.stream().mapToDouble(Expense::getAmount).sum();
    totalExpensesLabel.setText("Total Expenses: ₹" + totalExpenses);
    balanceLabel.setText("Remaining Budget: ₹" + (totalBudget -
totalExpenses));
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(ExpenseTracker::new);
}

// Inner Class to Represent an Expense

```

```
class Expense {  
    private String category;  
    private double amount;  
    private String date;  
  
    public Expense(String category, double amount, String date) {  
        this.category = category;  
        this.amount = amount;  
        this.date = date;  
    }  
  
    public String getCategory() {  
        return category;  
    }  
  
    public double getAmount() {  
        return amount;  
    }  
  
    public String getDate() {  
        return date;  
    }  
}  
}
```

APPENDIX-B SCREEN SHOT

The screenshot shows the 'Expense Tracker' application window. The 'Add Expense' section has four input fields: 'Category' (a dropdown menu showing 'Food' and a list of options: Food, Transport, Shopping, Bills, Miscellaneous), 'Amount (₹):', 'Date (DD/MM/YYYY):', and 'Set Budget (₹):'. Below these are two buttons: 'Add Expense' and 'Set Budget'. The 'Expense Summary' section contains an empty table. At the bottom, a green bar displays 'Total Expenses: ₹0' and 'Remaining Budget: ₹0'.

Category	Amount (₹)	Date
----------	------------	------

Total Expenses: ₹0
Remaining Budget: ₹0

The screenshot shows the 'Expense Tracker' application window after a user has added an expense and set a budget. The 'Add Expense' section now has values: 'Category' is 'Food', 'Amount (₹)' is empty, 'Date (DD/MM/YYYY)' is empty, and 'Set Budget (₹)' is '2000'. The 'Expense Summary' table now contains one row. At the bottom, a green bar displays 'Total Expenses: ₹200.0' and 'Remaining Budget: ₹1800.0'.

Category	Amount (₹)	Date
Food	₹200.00	28/11/24

Total Expenses: ₹200.0
Remaining Budget: ₹1800.0