PERSONAL FINANCE MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

NEKASHRI S(2303811724322076)

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 **" PERSONAL FINANCE MANAGEMENT SYSTEM"** is the bonafide work of **NEKASHRI S(2303811724322076)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

[illegible] S. Geetha

Signature	**Signature**
Dr. T. AVUDAIAPPAN M.E.,Ph.D.,	Mrs.S. GEETHA M.E.,
HEAD OF THE DEPARTMENT,	**SUPERVISOR,**
Department of Artificial Intelligence,	Department of Artificial Intelligence,
K. Ramakrishnan College of Technology,	K. Ramakrishnan College of Technology,
Samayapuram, Trichy -621 112.	Samayapuram, Trichy -621 112.

S

Submitted for the viva-voce examination held on 3.12.24

S. Geetha [illegible]

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

I declare that the project report on "**PERSONAL FINANCE MANAGEMENT SYSTEM** " 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.**

Nekashri ,S

Signature

NEKASHRI S

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 acknowledgment 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 conductive 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 Personal Finance Manager is a desktop application designed to streamline financial tracking and management. It enables users to record income, expenses, and categorize transactions while monitoring their budget and savings goals. The application provides real-time insights, including warnings for overspending and unmet savings targets, through an intuitive graphical user interface. Key features include dynamic balance updates, transaction history viewing, and monthly financial summaries. This project aims to simplify personal financial planning, helping users make informed decisions and maintain financial discipline. Future enhancements include data persistence, advanced analytics, multi-platform support, and AI-powered insights. The Personal Finance Manager also focuses on empowering users by providing clear insights into spending patterns and financial trends. It promotes proactive decision-making by encouraging users to set and achieve long-term financial goals. By integrating robust features and considering future scalability, the application serves as a versatile tool for comprehensive financial management.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Managing personal finances is essential for achieving financial stability and long-term goals. The Personal Finance Manager assists users in tracking income, expenses, and budgets effectively through an intuitive interface. It allows users to record transactions, set financial goals, and monitor progress in real-time. With features such as transaction history, financial summaries, and alerts for overspending or unmet savings targets, the tool promotes disciplined financial planning. Its user-friendly design ensures accessibility for individuals seeking to organize their finances efficiently and make informed decisions.

1.2 OBJECTIVE

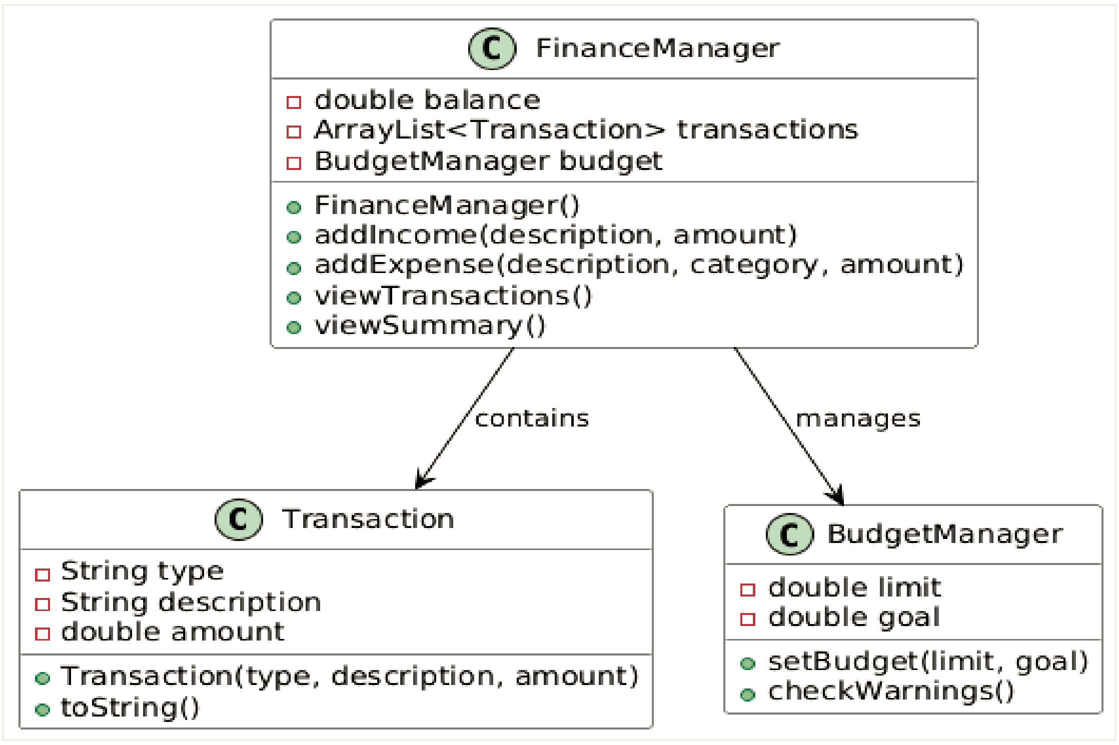
The Personal Finance Manager aims to provide users with a structured and effective approach to managing their finances. It focuses on streamlining the recording of transactions, monitoring expenses, and setting financial goals. By delivering features like real-time alerts for budget management, detailed transaction summaries, and goal tracking, the tool promotes better financial habits. It seeks to empower users to make data-driven decisions, maintain financial discipline, and achieve their monetary objectives with greater ease and clarity. Furthermore, it encourages proactive financial planning and fosters awareness of spending patterns to help users stay on track. With its intuitive interface, the tool is designed to cater to users of varying financial expertise.

CHAPTER 2
PROJECT METHODOLOGY

2.1 PROPOSED WORK

The proposed personal finance management system aims to develop a tool that helps users organize their finances by tracking transactions, income, expenses, and setting financial goals. It will feature budget monitoring with real-time alerts for overspending and savings goal tracking. The user-friendly interface will allow easy navigation and provide insights into spending patterns. Future improvements, such as data persistence, graphical analytics, and multi-platform support, will enhance the tool's functionality and impact.

2.2 BLOCK DIAGRAM



CHAPTER 3
JAVA PROGRAMMING

3.1 EVENT-DRIVEN PROGRAMMING

- **Action Listener Implementation:**

  The program adopts the event-driven programming model by responding to user interactions through event listeners. The PersonalFinanceManagerAWT class implements the ActionListener interface to listen for button clicks, triggering specific actions like adding income, recording expenses, or viewing transactions.

- **Event Handlers for User Action:**

  Each button in the application is associated with a corresponding event handler method (addIncome, addExpense, etc.), ensuring that the correct functionality is executed when a button is clicked.

3.2 AWT GUI COMPONENTS

- **Use of AWT Components:**

  The program employs AWT components such as Frame, Button, TextField, TextArea, Label, and GridBagLayout. These components are used to design a flexible and responsive graphical user interface, facilitating organized input and output through a grid-based layout.

- **WindowListener for Closing Events:**

  The application uses WindowListener to listen for window-closing events, ensuring that the program exits cleanly when the user closes the application window.

- **Text Formatting for Display:**

  To display transaction summaries and user messages, the program uses string manipulation techniques, including string concatenation and String.format(). This enables clear and user-friendly outputs within the GUI components.

CHAPTER 4
MODULE DESCRIPTION

3.1 TRANSACTION CLASS MODULE

Represents financial transactions (Income/Expense) with details like type, category, description, and amount. Serves as the foundational data structure to store and manage financial records.

3.2 PERSONALFINANCEMANAGERAWT CLASS MODULE

The main application class integrates the GUI, transaction management, and financial calculations. It coordinates all modules, serving as the central point for managing finances and user interactions. Acts as the central controller, coordinating user input, transactions, and financial calculations.

3.3 GUI COMPONENTS AND LAYOUT MODULE

Organizes input fields, buttons, and display areas using GridBagLayout for a user-friendly design that ensures seamless input and output handling.

3.4 ACTION HANDLERS AND EVENTS MODULE

Processes user actions (e.g., button clicks), links them to operations like adding transactions or generating summaries and handles input validation while dynamically updating financial records.

3.5 BUDGET AND WARNING MODULE

Tracks budget and savings goals, issuing alerts for overspending or unmet savings targets. It helps users maintain financial discipline by providing real-time warnings and reminders.

CHAPTER 5
CONCLUSION

The Personal Finance Management System successfully addresses the need for an organized and efficient way to manage personal finances. By offering features such as transaction tracking, income and expense management, budget monitoring, and savings goal tracking, the system empowers users to maintain financial discipline and make informed decisions. The user-friendly interface ensures accessibility, making financial management easy for users with varying levels of expertise. The system’s ability to provide real-time alerts and insightful summaries encourages better financial planning. With potential future enhancements like data persistence, graphical analytics, and multi-platform support, the tool can evolve to meet the growing needs of users and offer even greater value. Ultimately, this system fosters a proactive approach to financial management, helping users stay on track with their financial goals. By simplifying complex financial processes, it serves as a valuable resource for both short-term budgeting and long-term financial planning. The project has the potential to expand further with additional features and integrations, providing an all-encompassing financial management solution.

REFERENCES:

a) **Oracle Documentation (2021).** *Java AWT API*. Oracle. Retrieved from https://docs.oracle.com/javase/8/docs/api/java/awt/package-summary.html

b) **Java Tutorials (2021).** *How to Use Buttons, Check Boxes, and Radio Buttons in AWT*. Oracle. Retrieved from https://docs.oracle.com/javase/tutorial/uiswing/components/button.html

c) **Madrigal, L. (2019).** *Building a Personal Finance Application with Java*. Java World.

d) **Rita, A. (2018).** *Designing Financial Management Systems Using Java and Swing*. ACM Digital Library.

e) **Wiley, A. (2019).** *Managing Your Money: Budgeting and Financial Planning Tools for Java Developers*. Wiley Online. ISBN: 978-1119478660.

APPENDICES

APPENDIX A – SOURCE CODE

```
import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList;

class Transaction {
    String type;
    String category;
    String description;
    double amount;

    Transaction(String type, String category, String description, double amount) {
        this.type = type;
        this.category = category;
        this.description = description;
        this.amount = amount;
    }

    @Override
    public String toString() {
        return type + " (" + category + "): " + description + " - $" + amount;
    }
}

public class PersonalFinanceManagerAWT extends Frame implements
ActionListener {
    private double balance = 0.0;
    private double budgetLimit = 0.0;
    private double savingsGoal = 0.0;
    private double totalExpenses = 0.0;
    private ArrayList<Transaction> transactions = new ArrayList<>();

    private TextField descriptionField, amountField, categoryField, budgetField, goalField;
    private TextArea transactionArea;
    private Label balanceLabel, budgetLabel, expenseLabel, goalLabel;

    public PersonalFinanceManagerAWT() {
        setLayout(new GridBagLayout());
        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(5, 5, 5, 5);
        gbc.fill = GridBagConstraints.HORIZONTAL;

        // Input Fields
        gbc.gridx = 0; gbc.gridy = 0;
        add(new Label("Description:"), gbc);
        gbc.gridx = 1;
        descriptionField = new TextField(15);
        add(descriptionField, gbc);

        gbc.gridx = 0; gbc.gridy = 1;
        add(new Label("Amount:"), gbc);
        gbc.gridx = 1;
        amountField = new TextField(10);
        add(amountField, gbc);
```

```
gbc.gridx = 0; gbc.gridy = 2;
add(new Label("Category:"), gbc);
gbc.gridx = 1;
categoryField = new TextField(15);
add(categoryField, gbc);

// Buttons
Button addIncomeButton = new Button("Add Income");
Button addExpenseButton = new Button("Add Expense");
Button viewTransactionsButton = new Button("View Transactions");
Button setBudgetButton = new Button("Set Budget & Goal");
Button viewSummaryButton = new Button("Monthly Summary");

gbc.gridx = 0; gbc.gridy = 3;
add(addIncomeButton, gbc);
gbc.gridx = 1;
add(addExpenseButton, gbc);

gbc.gridx = 0; gbc.gridy = 4;
add(viewTransactionsButton, gbc);
gbc.gridx = 1;
add(setBudgetButton, gbc);

gbc.gridx = 0; gbc.gridy = 5;
add(viewSummaryButton, gbc);

// Budget and goal fields
gbc.gridx = 0; gbc.gridy = 6;
```

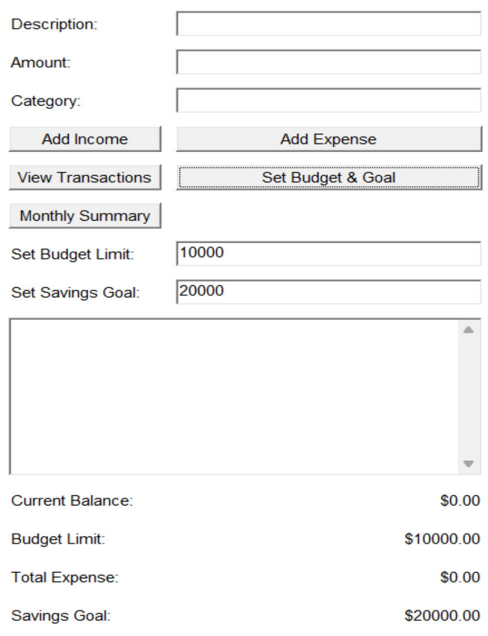
```
add(new Label("Set Budget Limit:"), gbc);
gbc.gridx = 1;
budgetField = new TextField(10);
add(budgetField, gbc);

gbc.gridx = 0; gbc.gridy = 7;
add(new Label("Set Savings Goal:"), gbc);
gbc.gridx = 1;
goalField = new TextField(10);
add(goalField, gbc);

// Transaction area
gbc.gridx = 0; gbc.gridy = 8;
gbc.gridwidth = 2;
transactionArea = new TextArea(8, 40);
add(transactionArea, gbc);

// Status Labels
gbc.gridwidth = 1;
gbc.gridx = 0; gbc.gridy = 9;
add(new Label("Current Balance:"), gbc);
gbc.gridx = 1;
balanceLabel = new Label("$0.00", Label.RIGHT);
add(balanceLabel, gbc);

gbc.gridx = 0; gbc.gridy = 10;
add(new Label("Budget Limit:"), gbc);
gbc.gridx = 1;
budgetLabel = new Label("$0.00", Label.RIGHT);
```

```
add(budgetLabel, gbc);

gbc.gridx = 0; gbc.gridy = 11;
add(new Label("Total Expense:"), gbc);
gbc.gridx = 1;
expenseLabel = new Label("$0.00", Label.RIGHT);
add(expenseLabel, gbc);

gbc.gridx = 0; gbc.gridy = 12;
add(new Label("Savings Goal:"), gbc);
gbc.gridx = 1;
goalLabel = new Label("$0.00", Label.RIGHT);
add(goalLabel, gbc);
// Add action listeners
addIncomeButton.addActionListener(this);
addExpenseButton.addActionListener(this);
viewTransactionsButton.addActionListener(this);
setBudgetButton.addActionListener(this);
viewSummaryButton.addActionListener(this);
// Frame properties
setTitle("Personal Finance Manager");
setSize(600, 500);
setVisible(true);

addWindowListener(new WindowAdapter() {
   public void windowClosing(WindowEvent e) {
      System.exit(0);
   }
});
```

```
}

@Override
public void actionPerformed(ActionEvent e) {
    String command = e.getActionCommand();
    try {
        switch (command) {
            case "Add Income":
                String incomeDesc = descriptionField.getText();
                double incomeAmount = Double.parseDouble(amountField.getText());
                addIncome(incomeDesc, incomeAmount);
                break;
            case "Add Expense":
                String expenseDesc = descriptionField.getText();
                String category = categoryField.getText();
                double expenseAmount = Double.parseDouble(amountField.getText());
                addExpense(expenseDesc, category, expenseAmount);
                break;
            case "View Transactions":
                viewTransactions();
                break;
            case "Set Budget & Goal":
                setBudgetAndGoal();
                break;
            case "Monthly Summary":
                viewMonthlySummary();
                break;
        }
    } catch (NumberFormatException ex) {
            transactionArea.append("Invalid input. Please enter valid numbers for
amounts.\n");
        }
    }

    private void addIncome(String description, double amount) {
        transactions.add(new Transaction("Income", "N/A", description, amount));
        balance += amount;
        updateLabels();
         transactionArea.append("Income added: " + description + " - $" + amount +
"\n");
    }

    private void addExpense(String description, String category, double amount) {
        transactions.add(new Transaction("Expense", category, description, amount));
        balance -= amount;
        totalExpenses += amount;
        checkBudgetWarnings();
        updateLabels();
        transactionArea.append("Expense added: " + category + " - " + description + " -
$" + amount + "\n");
    }

    private void setBudgetAndGoal() {
        budgetLimit = Double.parseDouble(budgetField.getText());
        savingsGoal = Double.parseDouble(goalField.getText());
        updateLabels();
    }
    private void viewTransactions() {
        transactionArea.setText("");
        for (Transaction transaction : transactions) {
            transactionArea.append(transaction + "\n");
        }
    }

    private void viewMonthlySummary() {
        transactionArea.append("\n--- Monthly Summary ---\n");
        transactionArea.append("Total Income: $" + (balance + totalExpenses) + "\n");
        transactionArea.append("Total Expenses: $" + totalExpenses + "\n");
        transactionArea.append("Net Balance: $" + balance + "\n");
    }
    private void checkBudgetWarnings() {
        if (totalExpenses > budgetLimit) {
                transactionArea.append("Warning: You have exceeded your budget
limit!\n");
        } else if (totalExpenses > 0.8 * budgetLimit) {
            transactionArea.append("Alert: You are nearing your budget limit!\n");
        }
        if (balance < savingsGoal) {
            transactionArea.append("Reminder: You are below your savings goal.\n");
        }
    }
    private void updateLabels() {
        balanceLabel.setText(String.format("$%.2f", balance));
        budgetLabel.setText(String.format("$%.2f", budgetLimit));
        expenseLabel.setText(String.format("$%.2f", totalExpenses));
        goalLabel.setText(String.format("$%.2f", savingsGoal));
    }

    public static void main(String[] args) {
        new PersonalFinanceManagerAWT();
    }
}
```

APPENDIX B - SCREENSHOTS

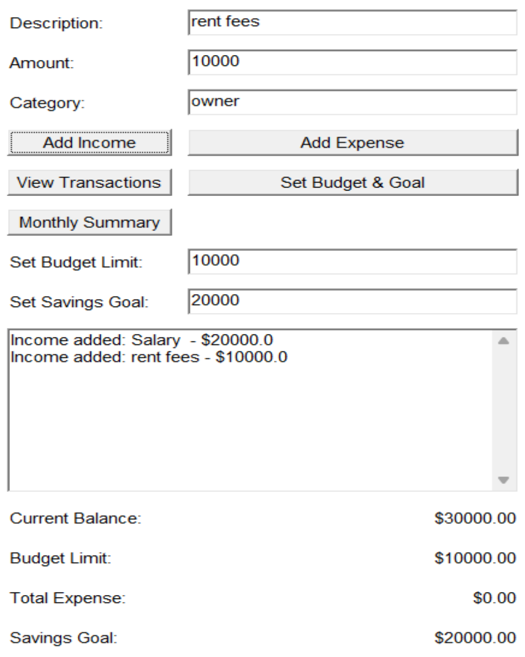
Set Budget and Goal



Description:	
Amount:	
Category:	
Set Budget Limit:	10000
Set Savings Goal:	20000

Current Balance:	$0.00
Budget Limit:	$10000.00
Total Expense:	$0.00
Savings Goal:	$20000.00

Add Income

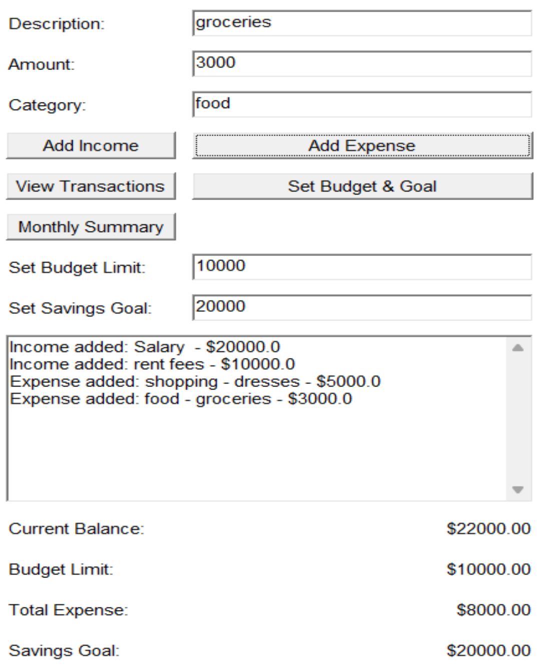


Description:	rent fees
Amount:	10000
Category:	owner
Set Budget Limit:	10000
Set Savings Goal:	20000

```
Income added: Salary  - $20000.0
Income added: rent fees - $10000.0
```

Current Balance:	$30000.00
Budget Limit:	$10000.00
Total Expense:	$0.00
Savings Goal:	$20000.00

Add Expense


Description: groceries

Amount: 3000

Category: food

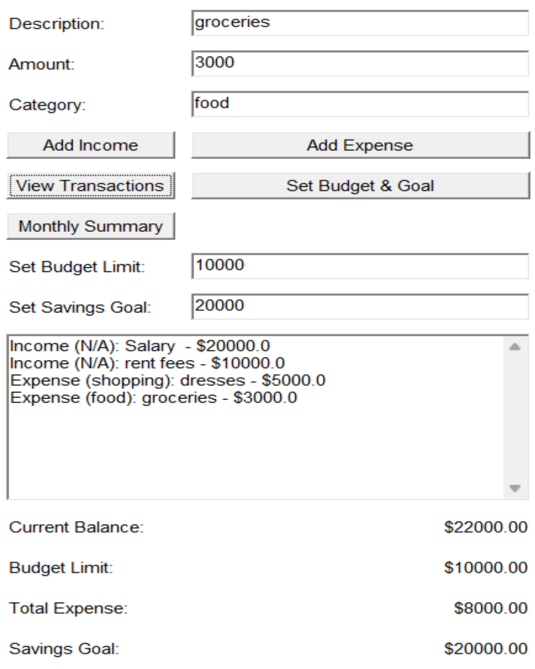
Set Budget Limit: 10000

Set Savings Goal: 20000

```
Income added: Salary  - $20000.0
Income added: rent fees - $10000.0
Expense added: shopping - dresses - $5000.0
Expense added: food - groceries - $3000.0
```

Current Balance:	$22000.00
Budget Limit:	$10000.00
Total Expense:	$8000.00
Savings Goal:	$20000.00

View Transaction


Description: groceries

Amount: 3000

Category: food

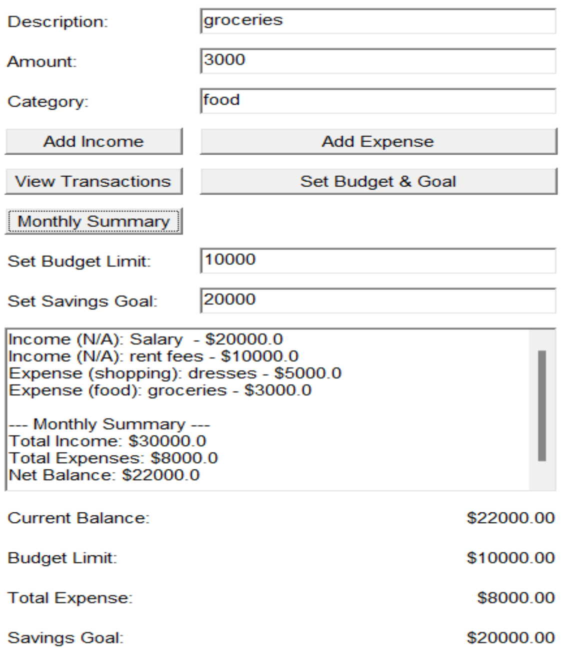
Set Budget Limit: 10000

Set Savings Goal: 20000

```
Income (N/A): Salary  - $20000.0
Income (N/A): rent fees - $10000.0
Expense (shopping): dresses - $5000.0
Expense (food): groceries - $3000.0
```

Current Balance:	$22000.00
Budget Limit:	$10000.00
Total Expense:	$8000.00
Savings Goal:	$20000.00

Monthly Summary

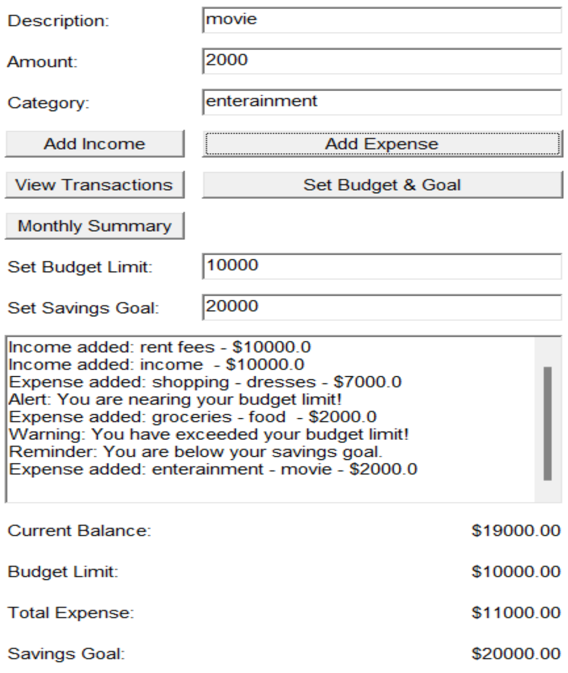


Description:	groceries
Amount:	3000
Category:	food
Set Budget Limit:	10000
Set Savings Goal:	20000

```
Income (N/A): Salary  - $20000.0
Income (N/A): rent fees - $10000.0
Expense (shopping): dresses - $5000.0
Expense (food): groceries - $3000.0

--- Monthly Summary ---
Total Income: $30000.0
Total Expenses: $8000.0
Net Balance: $22000.0
```

Current Balance:	$22000.00
Budget Limit:	$10000.00
Total Expense:	$8000.00
Savings Goal:	$20000.00

Alert & Warning Message



Description:	movie
Amount:	2000
Category:	enterainment
Set Budget Limit:	10000
Set Savings Goal:	20000

```
Income added: rent fees - $10000.0
Income added: income  - $10000.0
Expense added: shopping - dresses - $7000.0
Alert: You are nearing your budget limit!
Expense added: groceries - food  - $2000.0
Warning: You have exceeded your budget limit!
Reminder: You are below your savings goal.
Expense added: enterainment - movie - $2000.0
```

Current Balance:	$19000.00
Budget Limit:	$10000.00
Total Expense:	$11000.00
Savings Goal:	$20000.00