# LAB10

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class PersonalFinanceApp {
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new
FinanceAppFrame().setVisible(true));
    }
}

class FinanceAppFrame extends JFrame {
    private DefaultTableModel expenseTableModel;
    private DefaultTableModel incomeTableModel;
    private JLabel totalExpensesLabel;
    private JLabel totalIncomeLabel;

    public FinanceAppFrame() {
        setTitle("Personal Finance Manager");
        setSize(800, 600);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        JTabbedPane tabbedPane = new JTabbedPane();

        // Dashboard Tab
        JPanel dashboardPanel = createDashboardPanel();
        tabbedPane.addTab("Dashboard", dashboardPanel);

        // Expense Tracker Tab
        JPanel expensePanel = createExpensePanel();
        tabbedPane.addTab("Expense Tracker", expensePanel);

        // Income Tracker Tab
        JPanel incomePanel = createIncomePanel();
        tabbedPane.addTab("Income Tracker", incomePanel);
```

# LAB10

```
        add(tabbedPane);
    }

    private JPanel createDashboardPanel() {
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(2, 1, 10, 10));

        totalExpensesLabel = new JLabel("Total Expenses: $0.00",
JLabel.CENTER);
        totalIncomeLabel = new JLabel("Total Income: $0.00",
JLabel.CENTER);

        totalExpensesLabel.setFont(new Font("Arial", Font.BOLD, 16));
        totalIncomeLabel.setFont(new Font("Arial", Font.BOLD, 16));

        panel.add(totalExpensesLabel);
        panel.add(totalIncomeLabel);

        return panel;
    }

    private JPanel createExpensePanel() {
        JPanel panel = new JPanel(new BorderLayout());

        expenseTableModel = new DefaultTableModel(new
String[]{"Description", "Amount"}, 0);
        JTable expenseTable = new JTable(expenseTableModel);

        JPanel inputPanel = new JPanel();
        JTextField descriptionField = new JTextField(15);
        JTextField amountField = new JTextField(10);
        JButton addButton = new JButton("Add Expense");

        inputPanel.add(new JLabel("Description:"));
        inputPanel.add(descriptionField);
        inputPanel.add(new JLabel("Amount:"));
        inputPanel.add(amountField);
        inputPanel.add(addButton);

        addButton.addActionListener(e -> {
```

# LAB10

```
            String description = descriptionField.getText();
            String amountText = amountField.getText();

            if (!description.isEmpty() && !amountText.isEmpty()) {
                try {
                    double amount = Double.parseDouble(amountText);
                    expenseTableModel.addRow(new Object[]{description,
amount});
                    updateTotals();
                    descriptionField.setText("");
                    amountField.setText("");
                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(this, "Invalid amount.
Please enter a valid number.");
                }
            } else {
                JOptionPane.showMessageDialog(this, "Please fill in all
fields.");
            }
        });

        panel.add(new JScrollPane(expenseTable), BorderLayout.CENTER);
        panel.add(inputPanel, BorderLayout.SOUTH);

        return panel;
    }

    private JPanel createIncomePanel() {
        JPanel panel = new JPanel(new BorderLayout());

        incomeTableModel = new DefaultTableModel(new String[]{"Source",
"Amount"}, 0);
        JTable incomeTable = new JTable(incomeTableModel);

        JPanel inputPanel = new JPanel();
        JTextField sourceField = new JTextField(15);
        JTextField amountField = new JTextField(10);
        JButton addButton = new JButton("Add Income");

        inputPanel.add(new JLabel("Source:"));
```

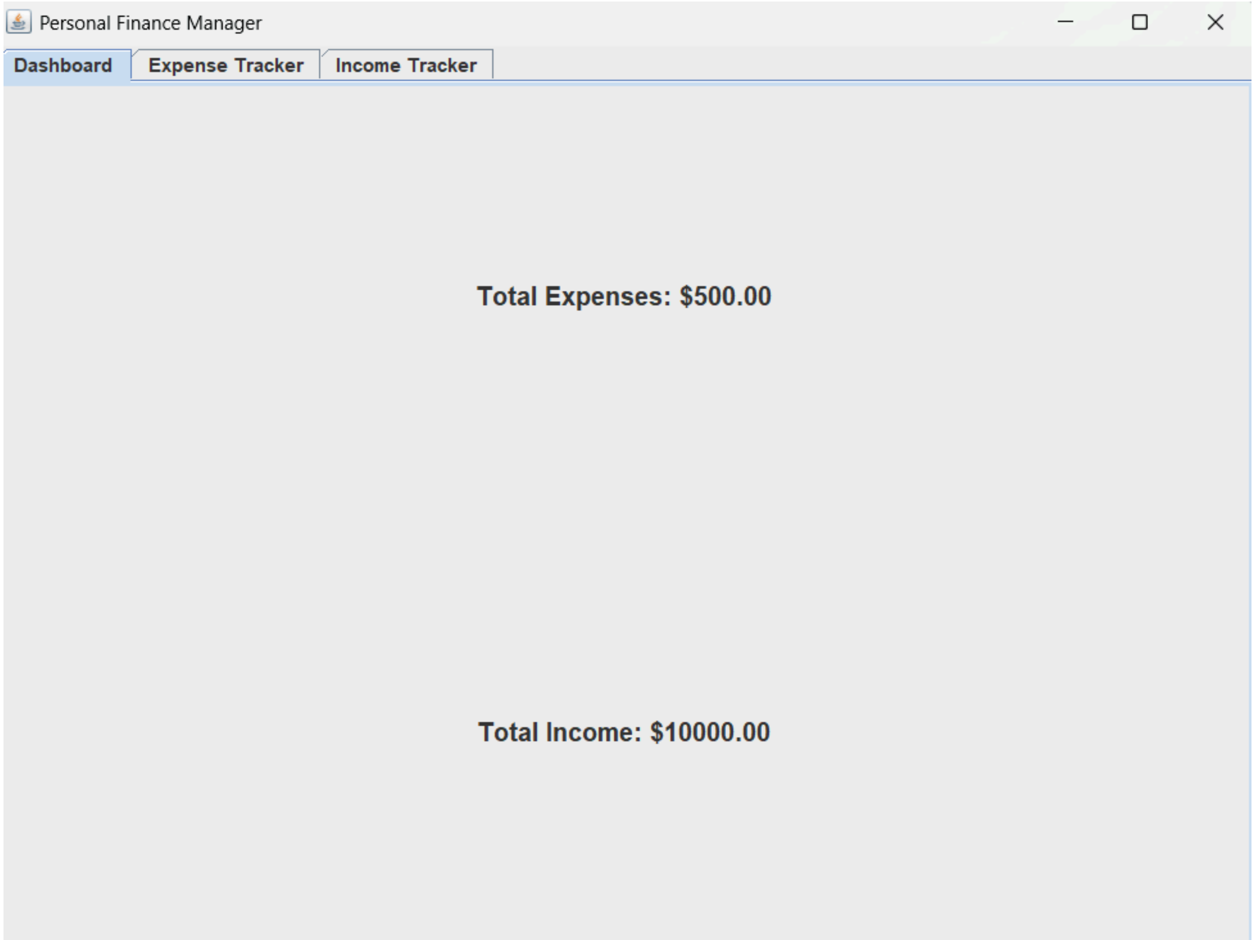
# LAB10

```
        inputPanel.add(sourceField);
        inputPanel.add(new JLabel("Amount:"));
        inputPanel.add(amountField);
        inputPanel.add(addButton);

        addButton.addActionListener(e -> {
            String source = sourceField.getText();
            String amountText = amountField.getText();

            if (!source.isEmpty() && !amountText.isEmpty()) {
                try {
                    double amount = Double.parseDouble(amountText);
                    incomeTableModel.addRow(new Object[]{source, amount});
                    updateTotals();
                    sourceField.setText("");
                    amountField.setText("");
                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(this, "Invalid amount.
Please enter a valid number.");
                }
            } else {
                JOptionPane.showMessageDialog(this, "Please fill in all
fields.");
            }
        });

        panel.add(new JScrollPane(incomeTable), BorderLayout.CENTER);
        panel.add(inputPanel, BorderLayout.SOUTH);

        return panel;
    }

    private void updateTotals() {
        double totalExpenses = 0.0;
        for (int i = 0; i < expenseTableModel.getRowCount(); i++) {
            totalExpenses += (double) expenseTableModel.getValueAt(i, 1);
        }
        totalExpensesLabel.setText(String.format("Total Expenses: $%.2f",
totalExpenses));
```

# LAB10

```
        double totalIncome = 0.0;
        for (int i = 0; i < incomeTableModel.getRowCount(); i++) {
            totalIncome += (double) incomeTableModel.getValueAt(i, 1);
        }
        totalIncomeLabel.setText(String.format("Total Income: $%.2f",
totalIncome));
    }
}
```

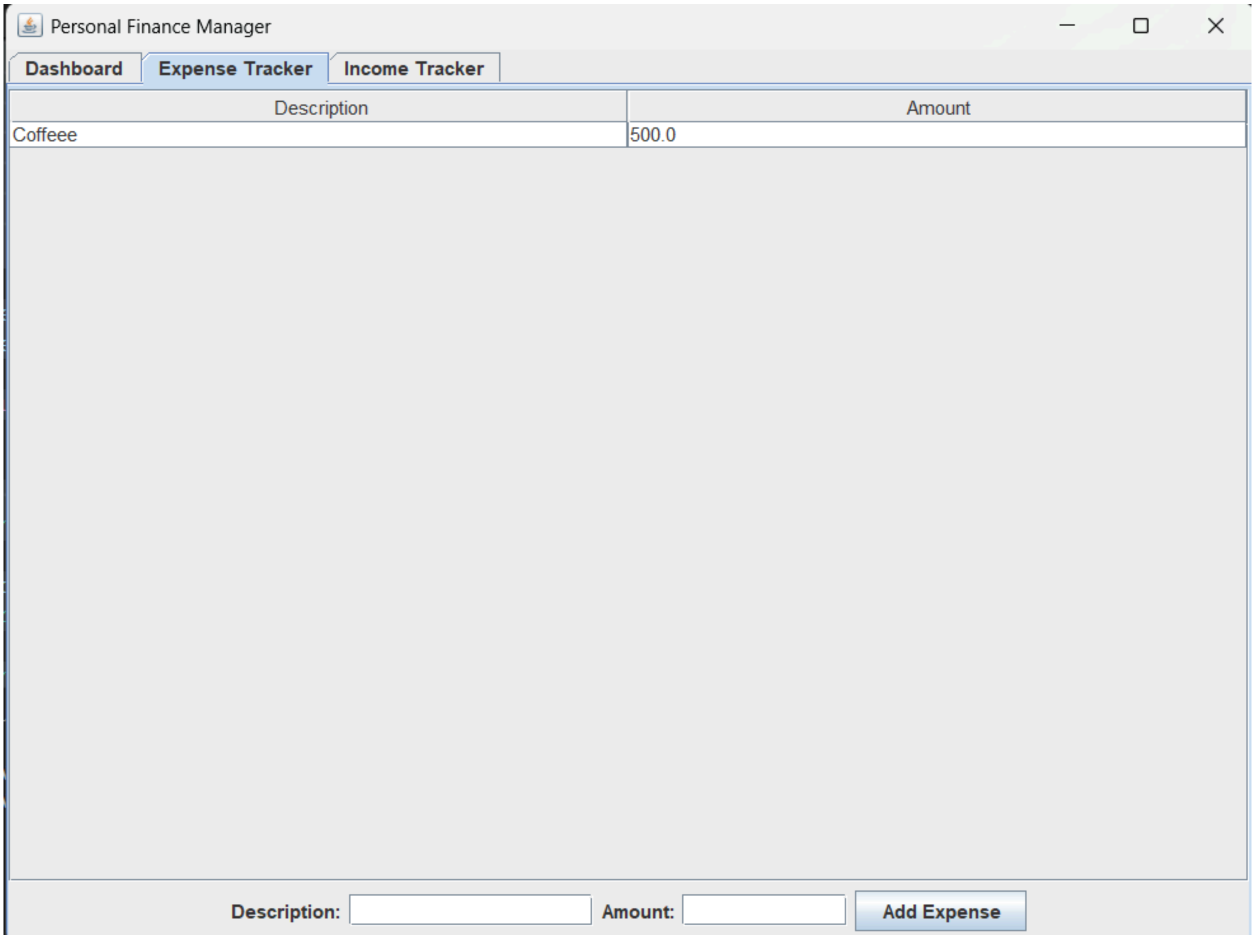
OUTPUT:



Total Expenses: $500.00

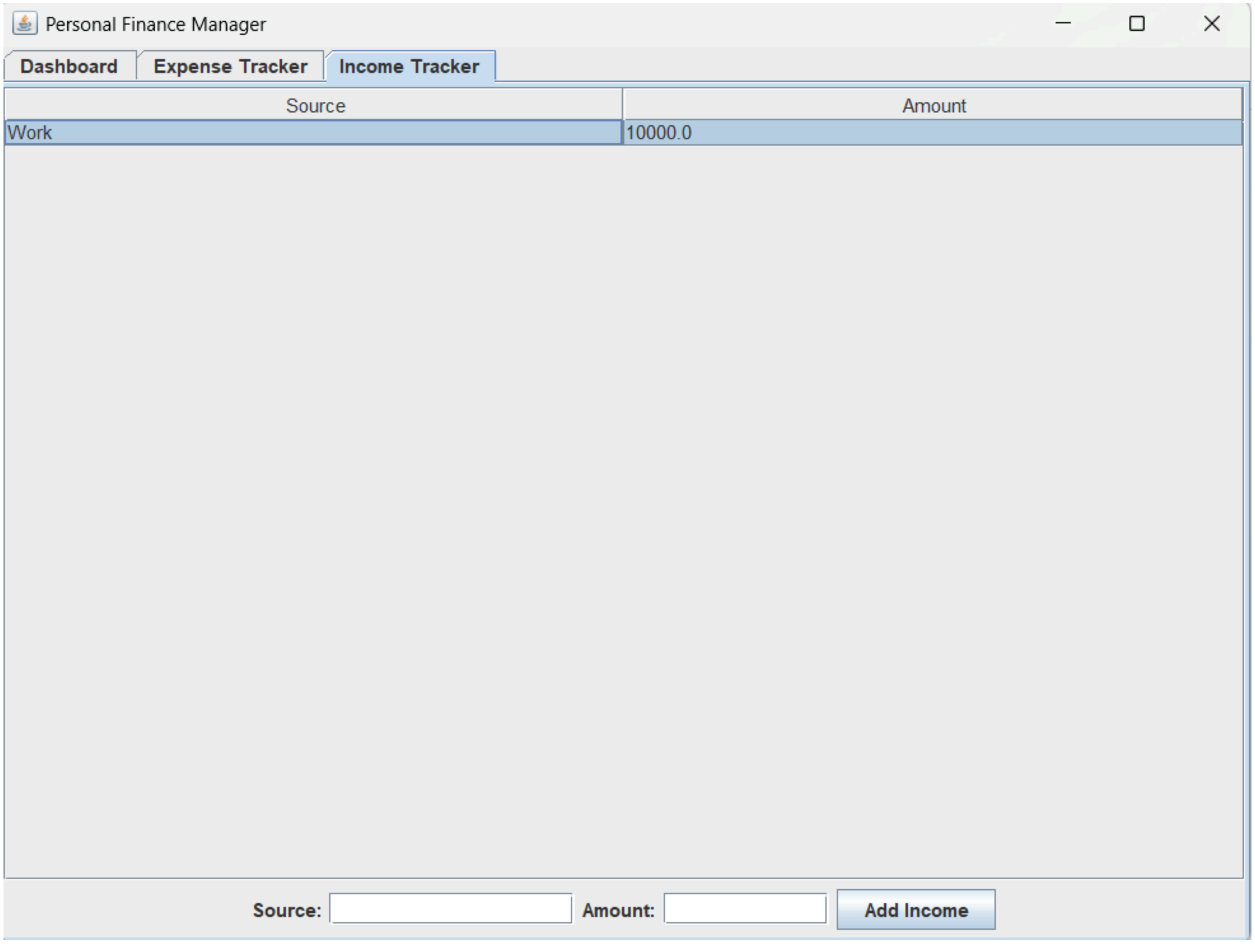
Total Income: $10000.00

# LAB10



| Description | Amount |
|---|---|
| Coffeee | 500.0 |

# LAB10



| Source | Amount |
|---|---|
| Work | 10000.0 |