Expense tracker

A MINI-PROJECT REPORT

Submitted by

SRAVANTHIKA.S 241001258

NIRTHYA THARA.H.S 241001153

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

INFORMATION OF TECHNOLOGY

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2025

RAJALAKSHMIENGINEERINGCOLLEGE

(AnAutonomousInstitution)

RAJALAKSHMINAGAR,THANDALAM-602105

BONAFIDE CERTIFICATE

Certified that this mini project titled "Expense Tracker" is the bona fide work of Sravanthika .S , Nirthya Thara.H.S , who have carried out the mini project work under my supervision for the course CS23332- Database Management Systems during the academic year 2025- 2026.

SIGNATURE

Prof. Subashree

Dept. of Information Technology

Rajalakshmi Engineering College Chennai

Submitted for the Viva Voce of the Mini Project held on: .....

INTERNALEXAMINER EXTERNALEXAMINER

OVERVIEW:

The Expense Tracker project is a Java-based application designed to help users efficiently manage and monitor their daily financial activities. It allows users to record expenses, categorize them (e.g., Food, Travel, Shopping), and view summaries of their spending patterns over time.

The system uses a Java frontend for user interaction and MySQL for data storage and retrieval.

This project aims to promote financial discipline and awareness by providing clear insights into how users spend their money. It also supports features such as income tracking, balance calculation, and categorized expense analysis.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman MR. S. MEGANATHAN and the chairperson DR. M.THANGAM MEGANATHAN for their timely support and encouragement.

Weare greatly indebted to our respected and honorable principal Dr. S.N. MURUGESAN forhis able support and guidance. No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department Dr.P.VALARMATHIE and our Deputy Head Of The Department Dr.M.BABU PEOFE for being ever supporting force during our project work.

We also extend our sincere and hearty thanks to our internal guide Prof. Subashree ma'am ,forhisvaluable guidance and motivation during the completion of this project. Our sincere thanks to our family members, friends and other staff members of computer science engineering.

1. NIRTHYA THARA H S

2. SRAVANTHIKA

TABLE OF CONTENTS

1. Introduction

In today's fast-paced life, managing personal finances manually can be challenging. The **Expense Tracker** application provides a digital solution to record daily transactions and generate meaningful statistics.

Objectives:

- To provide an easy way to record income and expenses.
- To categorize expenses for better tracking.
- To generate monthly or yearly reports.
- To store and retrieve data securely using a database.

Technologies Used:

Component	Technology
Frontend	Html (Swing/JavaFX)
Backend	MySQL Database
Language	Java

2.System Design and Architecture

The **system architecture** consists of three main layers:

1. **User Interface Layer:**
   Built using Java (Swing or JavaFX), allowing users to enter and view data easily.
2. **Application Logic Layer:**
   Handles data processing, calculations (like total expenses), and interaction between the UI and the database.
3. **Database Layer:**
   Stores all user data including income, expenses, and categories in **MySQL**.

Database Tables:

Table Name	Description	Key Fields
users	Stores user login details	user_id, username, password
income	Stores income records	income_id, user_id, amount, date
expenses	Stores expense records	expense_id, user_id, category, amount, date
categories	Stores expense categories	category_id, name

3. Implementation Details

Modules:

1. **User Authentication Module:**
   Allows users to register and log in securely.
2. **Income Management Module:**
   Enables users to add and view their income sources.
3. **Expense Management Module:**
   Allows users to record daily expenses under various categories.
4. **Report Generation Module:**
   Summarizes data monthly or yearly and shows total spending vs. savings.
5. **Category Management Module:**
   Enables creation, modification, or deletion of expense categories.

4. Results and Discussion

After implementing the Expense Tracker, users can:

- Easily **add, update, and delete** expense records.
- View **summary charts** to analyze their spending.
- Identify **savings trends** over time.

Example Output (Figure 2):

Figure 2: Expense Summary View

Category | Amount (₹)

Food | 4500

Travel | 1200

Shopping | 2000

Total | 770

5.Coding

1. Backend using Java and Mysql

Java code

Add expense

```
package com.expensetracker.controller;


import com.expensetracker.model.User;
import com.expensetracker.util.DBConnection;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;


public class AddExpenseServlet extends HttpServlet {


    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {


        System.out.println("✅ AddExpenseServlet triggered!");


        String description = request.getParameter("description");
        String category = request.getParameter("category");
```

```
        String amountStr = request.getParameter("amount");
        String date = request.getParameter("date");

        double amount = Double.parseDouble(amountStr);

        // ✅ Get logged-in user info from session
        HttpSession session = request.getSession(false);
        if (session == null || session.getAttribute("user") == null) {
            // User not logged in, redirect to login
            response.sendRedirect("login.jsp?error=Please+login+first");
            return;
        }

        // ✅ Retrieve user object from session
        User user = (User) session.getAttribute("user");
        int userId = user.getId();
        String username = user.getName();

        System.out.println("User ID: " + userId);
        System.out.println("Username: " + username);

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        try (Connection conn = DBConnection.getConnection()) {
            // ✅ Insert expense with userId and username
            String sql = "INSERT INTO expenses (description, category, amount, date, user_id, username)
VALUES (?, ?, ?, ?, ?, ?)";
            PreparedStatement stmt = conn.prepareStatement(sql);
            stmt.setString(1, description);
            stmt.setString(2, category);
```

```
            stmt.setDouble(3, amount);
            stmt.setString(4, date);
            stmt.setInt(5, userId);
            stmt.setString(6, username);

            int rowsInserted = stmt.executeUpdate();

            if (rowsInserted > 0) {
//                out.println("<html><body style='font-family: sans-serif; text-align:center;'>");
//                out.println("<h2 style='color:green;'>Expense added successfully!</h2>");
//                out.println("<a href='add_expense.html'>Add Another</a> | <a
href='view_expenses.jsp'>View All</a>");
//                out.println("</body></html>");
                response.sendRedirect("view_expenses.jsp");
            } else {
                out.println("<h2 style='color:red;'>Failed to add expense!</h2>");
            }

        } catch (SQLException e) {
            e.printStackTrace();
            out.println("<h3>Error: " + e.getMessage() + "</h3>");
        }
    }
}
```

Add user expense

```
package com.expensetracker.controller;

import com.expensetracker.dao.UserDAO;
import com.expensetracker.model.User;
import jakarta.servlet.*;
import jakarta.servlet.annotation.WebServlet;
```

```
import jakarta.servlet.http.*;
import java.io.IOException;

@WebServlet("/addUser")
public class AddUserServlet extends HttpServlet {
    private UserDAO userDAO = new UserDAO();

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {

        String name = request.getParameter("name");
        String email = request.getParameter("email");
        String password = request.getParameter("password");
        String role = request.getParameter("role");

        User newUser = new User();
        newUser.setName(name);
        newUser.setEmail(email);
        newUser.setPassword(password);
        newUser.setRole(role);

        boolean success = userDAO.register(newUser);
        if (success) {
            response.sendRedirect("manageUsers");
        } else {
            response.getWriter().println("<h3>Error adding user. Try again.</h3>");
        }
    }
```

```
}
```

Edit User

```
package com.expensetracker.controller;

import com.expensetracker.dao.UserDAO;
import com.expensetracker.model.User;
import jakarta.servlet.*;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.*;
import java.io.IOException;

@WebServlet("/edit_user")
public class EditUserServlet extends HttpServlet {
    private UserDAO userDAO = new UserDAO();

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        int id = Integer.parseInt(request.getParameter("id"));
        User user = userDAO.getUserById(id);

        request.setAttribute("user", user);
        RequestDispatcher dispatcher = request.getRequestDispatcher("edit_user.jsp");
        dispatcher.forward(request, response);
    }
}
```

Update User

```
package com.expensetracker.controller;


import com.expensetracker.util.DBConnection;
import jakarta.servlet.*;
import jakarta.servlet.http.*;
import jakarta.servlet.annotation.*;
import java.io.*;
import java.sql.*;


@WebServlet("/update_expense")
public class UpdateExpenseServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {


        int id = Integer.parseInt(request.getParameter("id"));
        String description = request.getParameter("description");
        String category = request.getParameter("category");
        double amount = Double.parseDouble(request.getParameter("amount"));
        String date = request.getParameter("date");


        try (Connection conn = DBConnection.getConnection()) {
            String sql = "UPDATE expenses SET description=?, category=?, amount=?, date=?
WHERE id=?";
            PreparedStatement ps = conn.prepareStatement(sql);
            ps.setString(1, description);
            ps.setString(2, category);
            ps.setDouble(3, amount);
```

```
            ps.setString(4, date);
            ps.setInt(5, id);
            ps.executeUpdate();
        } catch (Exception e) {
            e.printStackTrace();
        }

        response.sendRedirect("view_expenses.jsp");
    }
}
```

View expense tracker

```
package com.expensetracker.controller;

import com.expensetracker.dao.ExpenseDAO;
import com.expensetracker.model.User;
import jakarta.servlet.*;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.*;
import java.io.IOException;
import java.util.Map;

@WebServlet("/monthlyExpenditure")
public class ViewMonthlyExpenditureServlet extends HttpServlet {
    private ExpenseDAO expenseDAO = new ExpenseDAO();

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse res)
            throws ServletException, IOException {
```

```
        HttpSession session = req.getSession(false);
        User user = (User) (session != null ? session.getAttribute("user") : null);

        if (user == null) {
            res.sendRedirect("login.jsp");
            return;
        }

        // ✅ Declare correct type for data
        Map<String, Map<String, Double>> monthlyTotals;

        // ✅ Choose query based on role
        if ("ADMIN".equalsIgnoreCase(user.getRole())) {
            monthlyTotals = expenseDAO.getAllUsersMonthlyTotalsByCategory();
        } else {
            monthlyTotals = expenseDAO.getMonthlyTotalsByCategory(user.getId());
        }

        // ✅ Send to JSP
        req.setAttribute("monthlyTotals", monthlyTotals);
        RequestDispatcher rd = req.getRequestDispatcher("monthly_expenditure.jsp");
        rd.forward(req, res);
    }
}
```

Dbms queries using Mysql

To Create Table

```
CREATE TABLE expenses (
  id INT PRIMARY KEY AUTO_INCREMENT,
  description VARCHAR(50),
  category VARCHAR(20),
  amount INT,
  date DATE,
  user_id INT,
  username VARCHAR(20)
);
CREATE TABLE users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(50),
    email VARCHAR(100),
    password VARCHAR(50),
    role ENUM('USER', 'ADMIN') NOT NULL
);
```

Frontend using Html code

Add expense

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Add Expense</title>

    <style>
        /* Global styles */
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        body {
            font-family: 'Poppins', sans-serif;
            background-color: #f8f9fa;
            color: #333;
        }

        /* Navbar */
        .navbar {
            background-color: #343a40;
            color: white;
            padding: 15px 30px;
            display: flex;
            justify-content: space-between;
            align-items: center;
            box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
        }
```

```
.navbar h1 {
    font-size: 22px;
}

.navbar ul {
    list-style: none;
    display: flex;
    gap: 20px;
}

.navbar a {
    color: white;
    text-decoration: none;
    font-weight: 500;
}

.navbar a:hover {
    text-decoration: underline;
}

/* Container */
.container {
    width: 400px;
    margin: 80px auto;
    background: white;
    padding: 30px;
    border-radius: 10px;
    box-shadow: 0 0 15px rgba(0, 0, 0, 0.1);
}

button {
    width: 100%;
```

```
    margin-top: 15px;
    padding: 10px;
    background-color: #28a745;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
}

button:hover {
    background-color: #218838;
}

h2 {
    text-align: center;
    margin-bottom: 15px;
}

label {
    display: block;
    margin-top: 10px;
}

input {
    width: 100%;
    padding: 8px;
    margin-top: 5px;
    border-radius: 5px;
    border: 1px solid #ccc;
}
select {
    width: 100%;
```

```
            padding: 8px;
            margin-top: 5px;
            border-radius: 5px;
            border: 1px solid #ccc;
        }

    </style>
</head>
<body>
<div class="navbar">
    <h1>Expense Tracker</h1>
    <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="add_expense.html">Add Expense</a></li>
        <li><a href="view_expenses.jsp">View Expenses</a></li>
    </ul>
</div>

<div class="container">
    <h2>Add New Expense</h2>
    <form action="addExpense" method="post">
        <label for="description">Description:</label>
        <input type="text" id="description" name="description" required>

        <label for="category">Category:</label>
        <select id="category" name="category" required>
            <option value="">-- Select Category --</option>
            <option value="Income">Income</option>
            <option value="Expense">Expense</option>
        </select>

        <label for="amount">Amount:</label>
        <input type="number" id="amount" name="amount" step="1" required>
```

```
        <label for="date">Date:</label>
        <input type="date" name="date" id="date" required>

        <button type="submit">Add</button>
    </form>

</div>
</body>
</html>
```

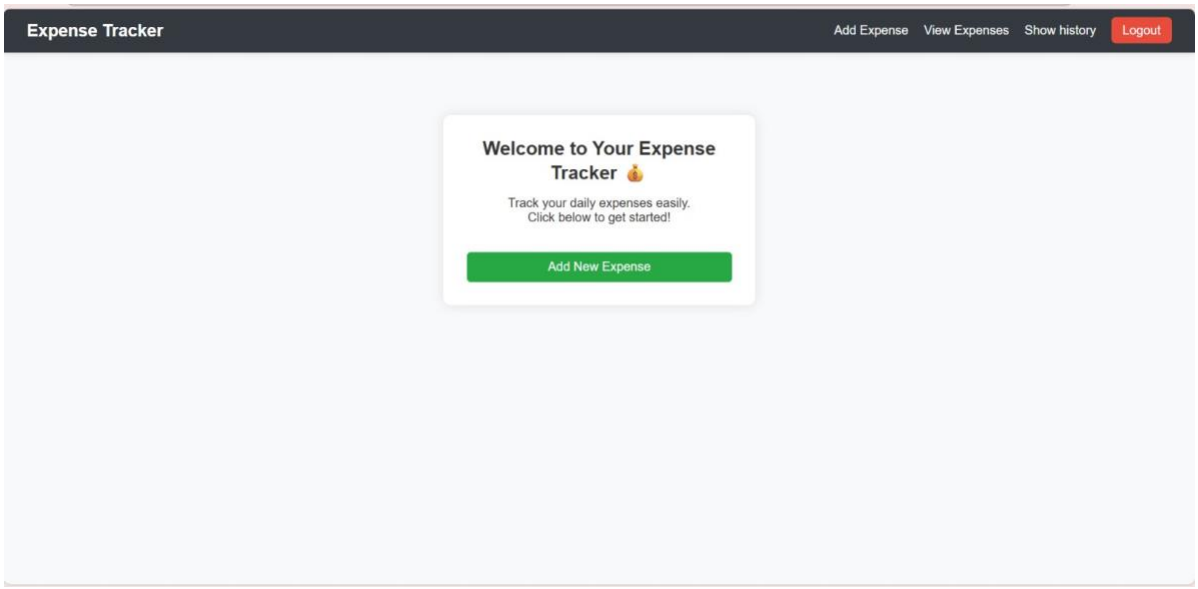
Edit expense

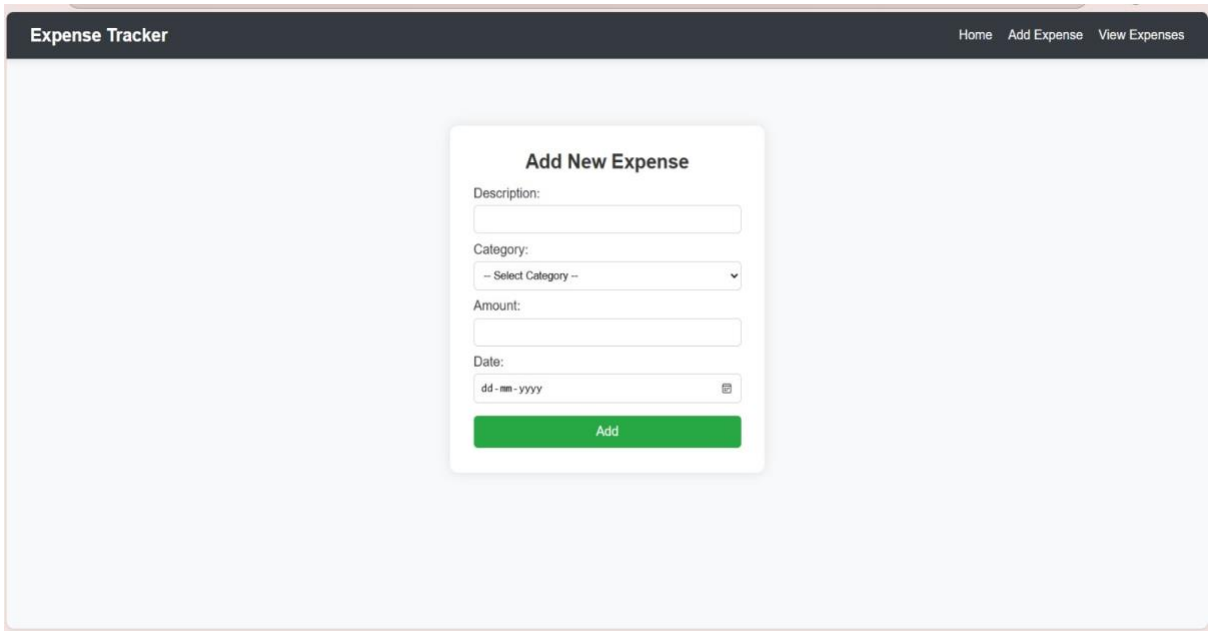
```
<%@ page import="java.sql.*, com.expensetracker.util.DBConnection" %>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Edit Expense</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f8f9fa;
        }
        .container {
            width: 40%;
            margin: 80px auto;
            background-color: #fff;
            padding: 25px;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0,0,0,0.1);
        }
        input, select {
```

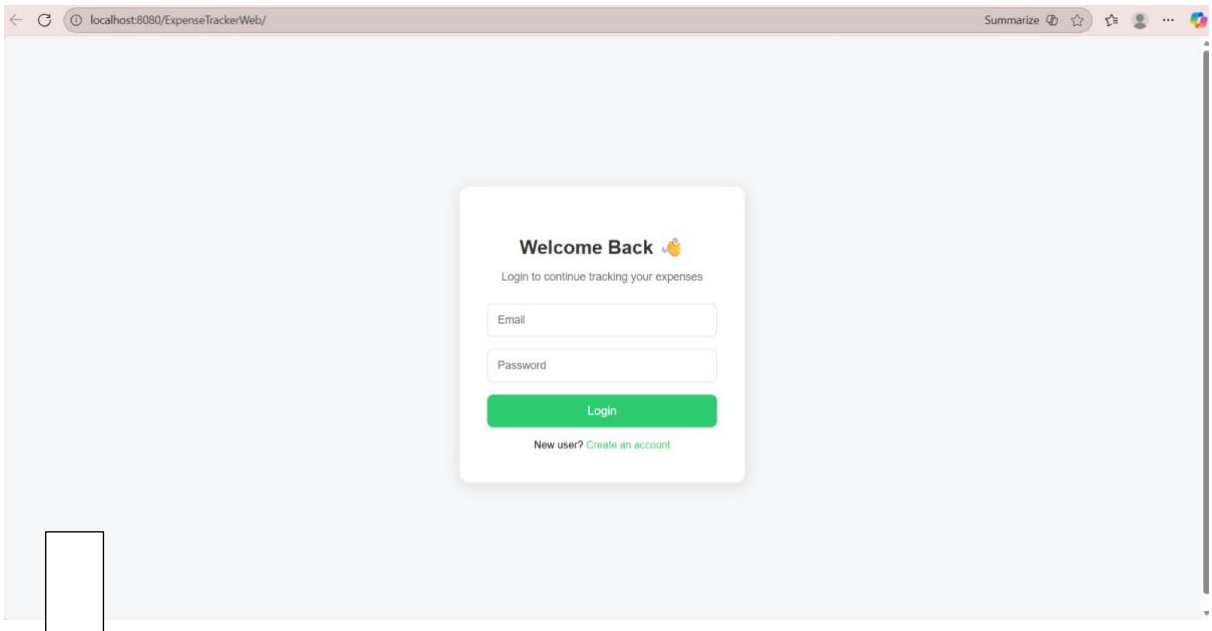
```
            width: 100%;
            padding: 10px;
            margin: 10px 0;
        }
        button {
            background-color: #28a745;
            color: white;
            padding: 10px 15px;
            border: none;
            border-radius: 5px;
            cursor: pointer;
        }
        button:hover {
            background-color: #218838;
        }
    </style>
</head>
<body>
<div class="container">
    <h2>Edit Expense</h2>

    <%
        int id = Integer.parseInt(request.getParameter("id"));
        String description = "", category = "", date = "";
        double amount = 0;

        try (Connection conn = DBConnection.getConnection()) {
            String sql = "SELECT * FROM expenses WHERE id=?";
            PreparedStatement ps = conn.prepareStatement(sql);
            ps.setInt(1, id);
            ResultSet rs = ps.executeQuery();
            if (rs.next()) {
                description = rs.getString("description");
```

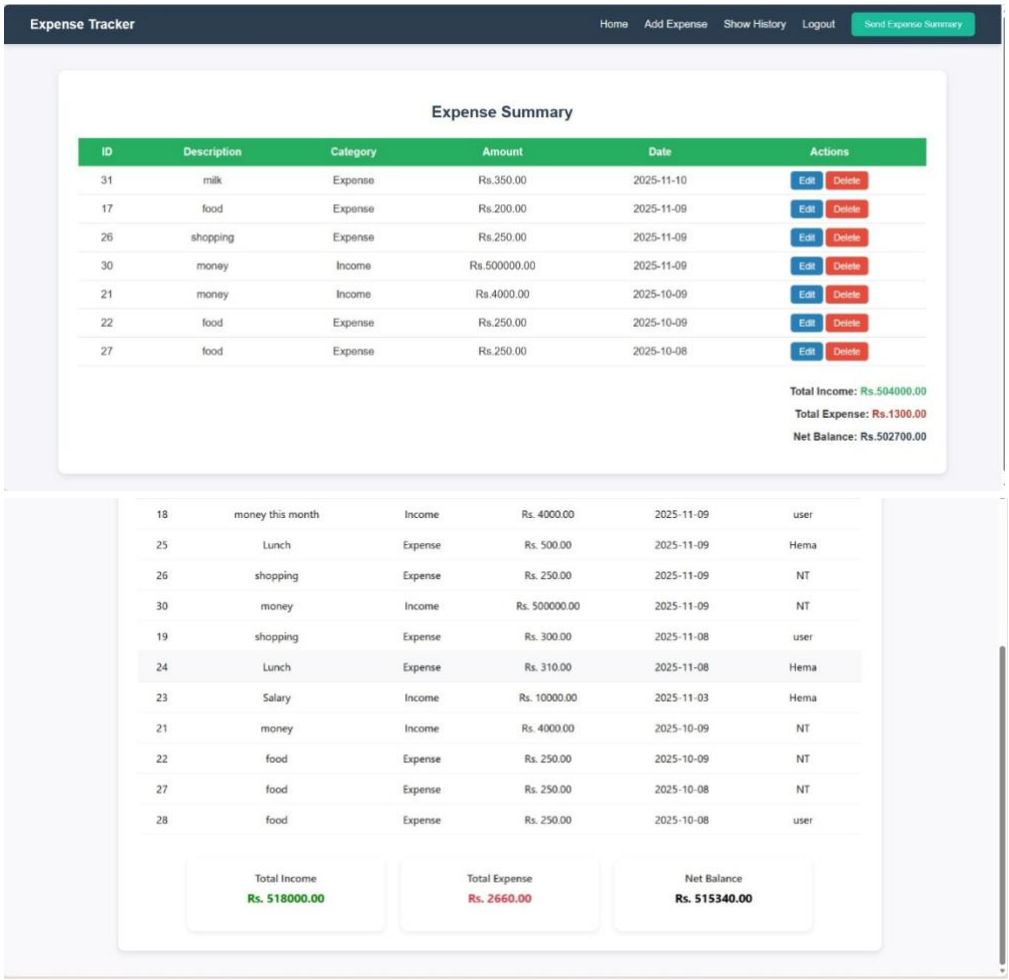
```
            category = rs.getString("category");
            amount = rs.getDouble("amount");
            date = rs.getString("date");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
%>

<form action="update_expense" method="post"> <!-- ✅ maps to your UpdateExpenseServlet -->
    <input type="hidden" name="id" value="<%= id %>">

    <label>Description:</label>
    <input type="text" name="description" value="<%= description %>" required>

    <label>Category:</label>
    <select name="category" required>
        <option value="Income" <%= "Income".equals(category) ? "selected" : "" %>>Income</option>
        <option value="Expense" <%= "Expense".equals(category) ? "selected" : "" %>>Expense</option>
    </select>

    <label>Amount:</label>
    <input type="number" name="amount" value="<%= amount %>" step="0.01" required>

    <label>Date:</label>
    <input type="date" name="date" value="<%= date.substring(0,10) %>" required>

    <button type="submit">Update Expense</button>
</form>
</div>
</body>
</html
```

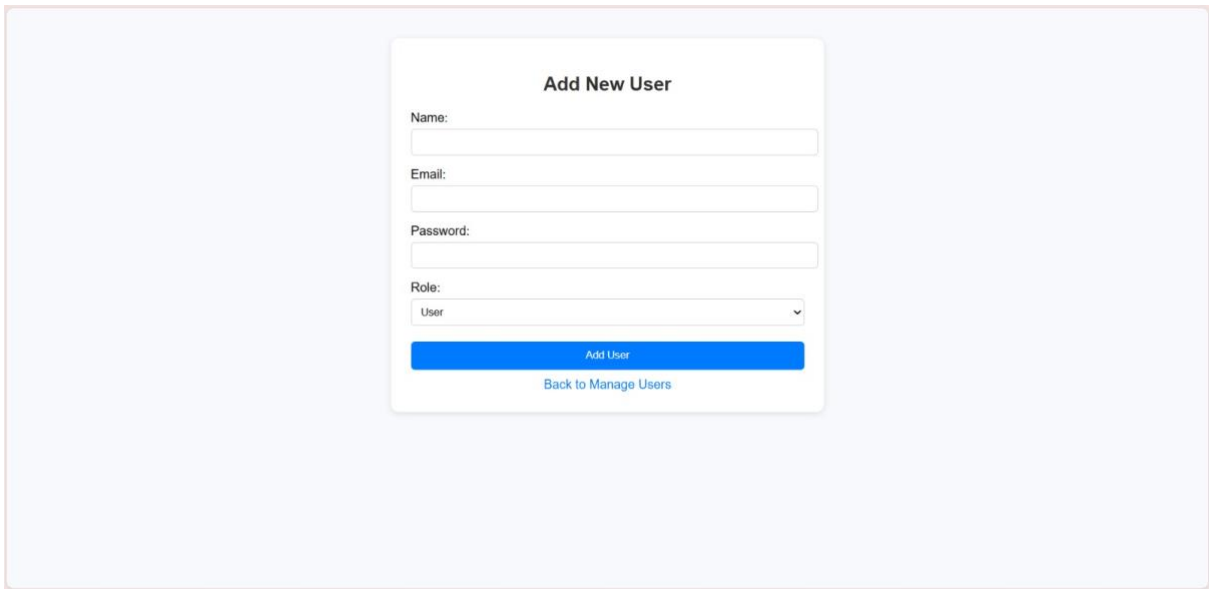
6.Screen shot

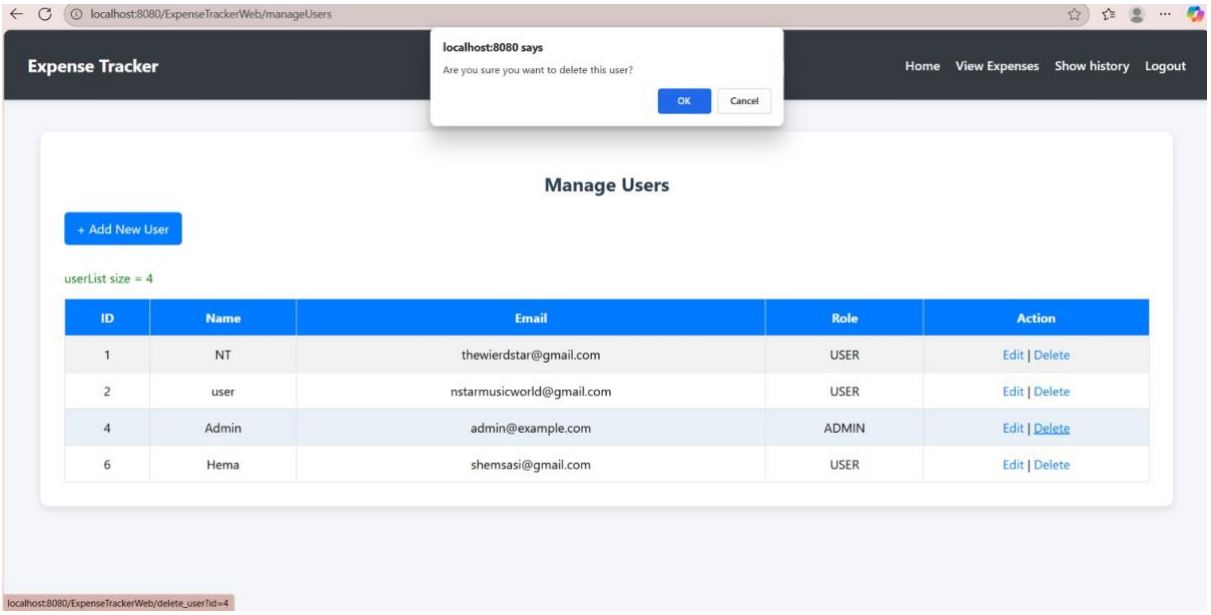







Expense Tracker
Home
Add Expense
Show History
Logout
Send Expense Summary
Expense Summary
ID Description Category Amount Date Actions
31 milk Expense Rs.350.00 2025-11-10 Edit Delete
17 food Expense Rs.200.00 2025-11-09 Edit Delete
26 shopping Expense Rs.250.00 2025-11-09 Edit Delete
30 money Income Rs.500000.00 2025-11-09 Edit Delete
21 money Income Rs.4000.00 2025-10-09 Edit Delete
22 food Expense Rs.250.00 2025-10-09 Edit Delete
27 food Expense Rs.250.00 2025-10-08 Edit Delete
Total Income: Rs.504000.00
Total Expense: Rs.1300.00
Net Balance: Rs.502700.00
18 money this month Income Rs. 4000.00 2025-11-09 user
25 Lunch Expense Rs. 500.00 2025-11-09 Hema
26 shopping Expense Rs. 250.00 2025-11-09 NT
30 money Income Rs. 500000.00 2025-11-09 NT
19 shopping Expense Rs. 300.00 2025-11-08 user
24 Lunch Expense Rs. 310.00 2025-11-08 Hema
23 Salary Income Rs. 10000.00 2025-11-03 Hema
21 money Income Rs. 4000.00 2025-10-09 NT
22 food Expense Rs. 250.00 2025-10-09 NT
27 food Expense Rs. 250.00 2025-10-08 NT
28 food Expense Rs. 250.00 2025-10-08 user
Total Income
Rs. 518000.00
Total Expense
Rs. 2660.00
Net Balance
Rs. 515340.00


Add New User
Name:
Email:
Password:
Role:
User
Add User
Back to Manage Users



7. Conclusion and Future Scope

The **Expense Tracker** project successfully helps users monitor and manage their personal finances digitally.
It demonstrates the integration of **Java and MySQL** for building an efficient and user-friendly financial management system.

Future Enhancements:

- Integration with **mobile apps** for portability.
- Adding **graphical charts** for visual analysis.
- Introducing **AI-based suggestions** for budgeting and savings.
- Cloud-based storage for data backup.

8. References

1. *Java: The Complete Reference* by Herbert Schildt
2. *Database System Concepts* by Abraham Silberschatz
3. MySQL Official Documentation – https://dev.mysql.com/doc/
4. Oracle Java Tutorials – https://docs.oracle.com/javase/tutorial/
5. W3Schools Java Database Connectivity Guide – https://www.w3schools.com/java/java_mysql.asp