EXPENSE TRACKER

Personal Finance Management System

By:

Krupa Devar : 24107065

Shravni Bhosale : 24107025

Shravani Karpe : 24107047

Project Guide : Ms. Aavani Nair

Department of Computer Science Engineering (Data Science)

A.P. Shah Institute of Technology, Thane

Academic Year: 2025-26

Outline

- Introduction
- Problem Statement
- System Design
- Technologies and Methodologies
- Implementation
- Conclusion
- References

Introduction

Real-time Problem Observation:
- Students and individuals struggle with tracking daily expenses
- Lack of simple, accessible expense management tools
- Manual record-keeping is time-consuming and error-prone
- Difficulty in analyzing spending patterns

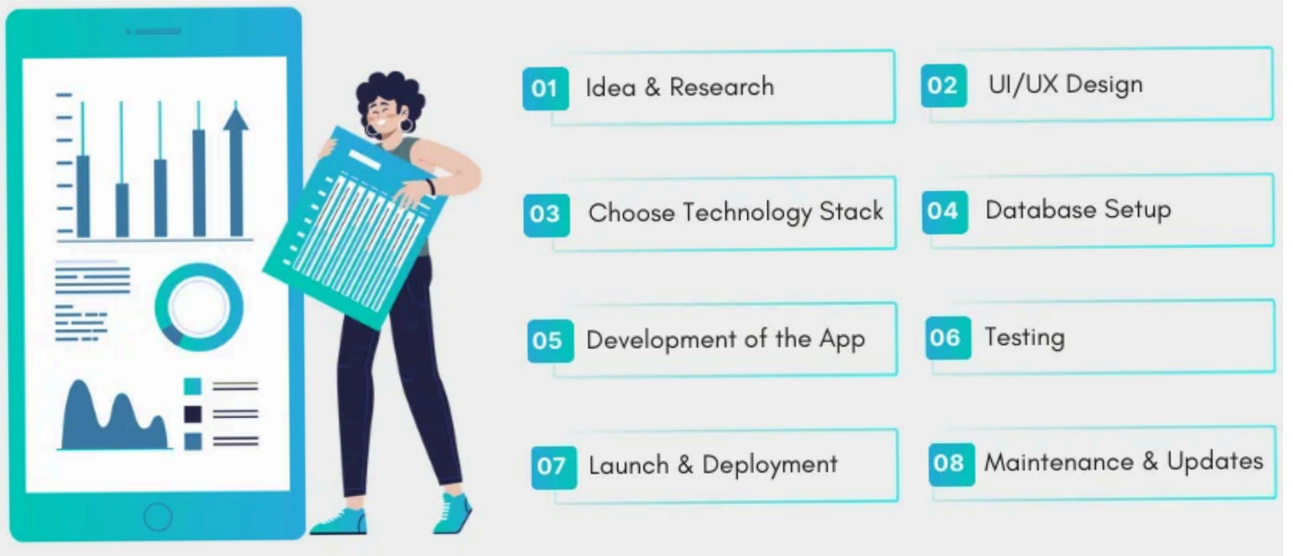
Motivation:
- Need for digital solution to manage personal finances
- Enable better financial decision-making
- Provide instant access to expense history
- Simplify budget tracking for students

Objectives:
- Develop web-based expense tracking application
- Implement database integration for persistent storage
- Create user-friendly interface for expense management
- Enable real-time calculation of total expenditure

Problem Statement

Core Problem:

Individuals lack an efficient system to record, view, and analyze their daily expenses, leading to poor financial awareness and budget management.



Key Points to Address:

- Easy expense entry mechanism
- Categorized expense tracking
- Real-time data storage and retrieval
- Total expenditure calculation
- Cross-platform accessibility

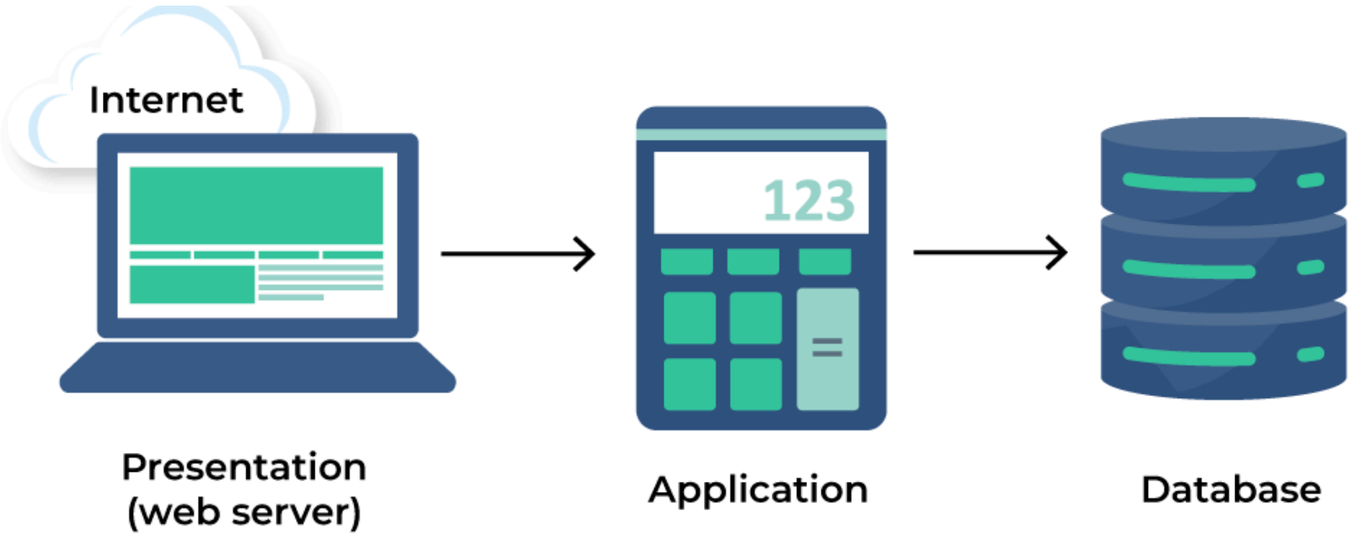
System Architecture

Three-Tier Architecture

Presentation Layer (Frontend):
- JSP pages with HTML/CSS/JavaScript
- User interface for data input and display

Application Layer (Backend):
- Java Servlets handling HTTP requests
- Business logic processing
- Data validation

Data Layer:
- MySQL database
- JDBC connectivity
- Persistent data storage



System Design - Component Diagram

Key Components:

Model Layer:

- Expense.java - Data model with attributes (id, description, amount, category, date)

DAO Layer:

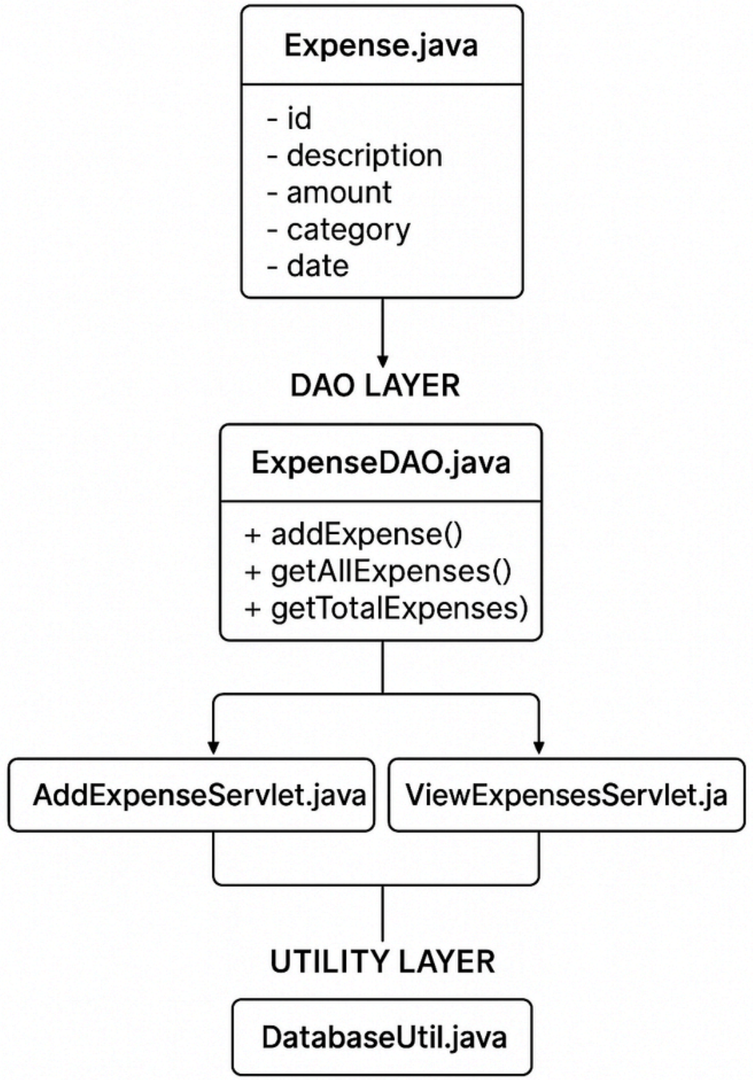
- ExpenseDAO.java - Database operations (CRUD)
- Methods: addExpense(), getAllExpenses(), getTotalExpenses()

Controller Layer:

- AddExpenseServlet.java - Handles expense creation
- ViewExpensesServlet.java - Retrieves and displays expenses

Utility Layer:

- DatabaseUtil.java - Database connection management



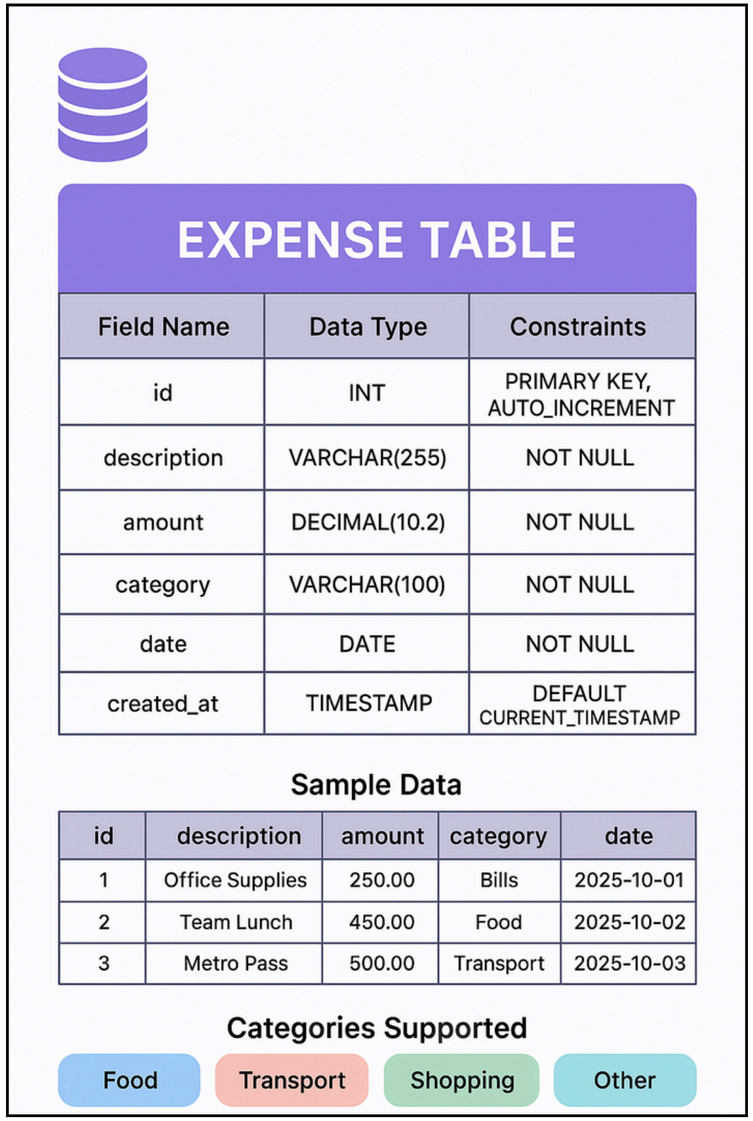
Database Design

Expense Table Schema:

- id (INT, PRIMARY KEY, AUTO_INCREMENT)
- description (VARCHAR, 255)
- amount (DECIMAL, 10,2)
- category (VARCHAR, 100)
- date (DATE)
- created_at (TIMESTAMP)

Categories Supported:

Food, Transport, Shopping, Bills, Entertainment, Other



EXPENSE TABLE

Field Name	Data Type	Constraints
id	INT	PRIMARY KEY, AUTO_INCREMENT
description	VARCHAR(255)	NOT NULL
amount	DECIMAL(10.2)	NOT NULL
category	VARCHAR(100)	NOT NULL
date	DATE	NOT NULL
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP

Sample Data

id	description	amount	category	date
1	Office Supplies	250.00	Bills	2025-10-01
2	Team Lunch	450.00	Food	2025-10-02
3	Metro Pass	500.00	Transport	2025-10-03

Categories Supported

Food | Transport | Shopping | Other

Technologies and Methodologies

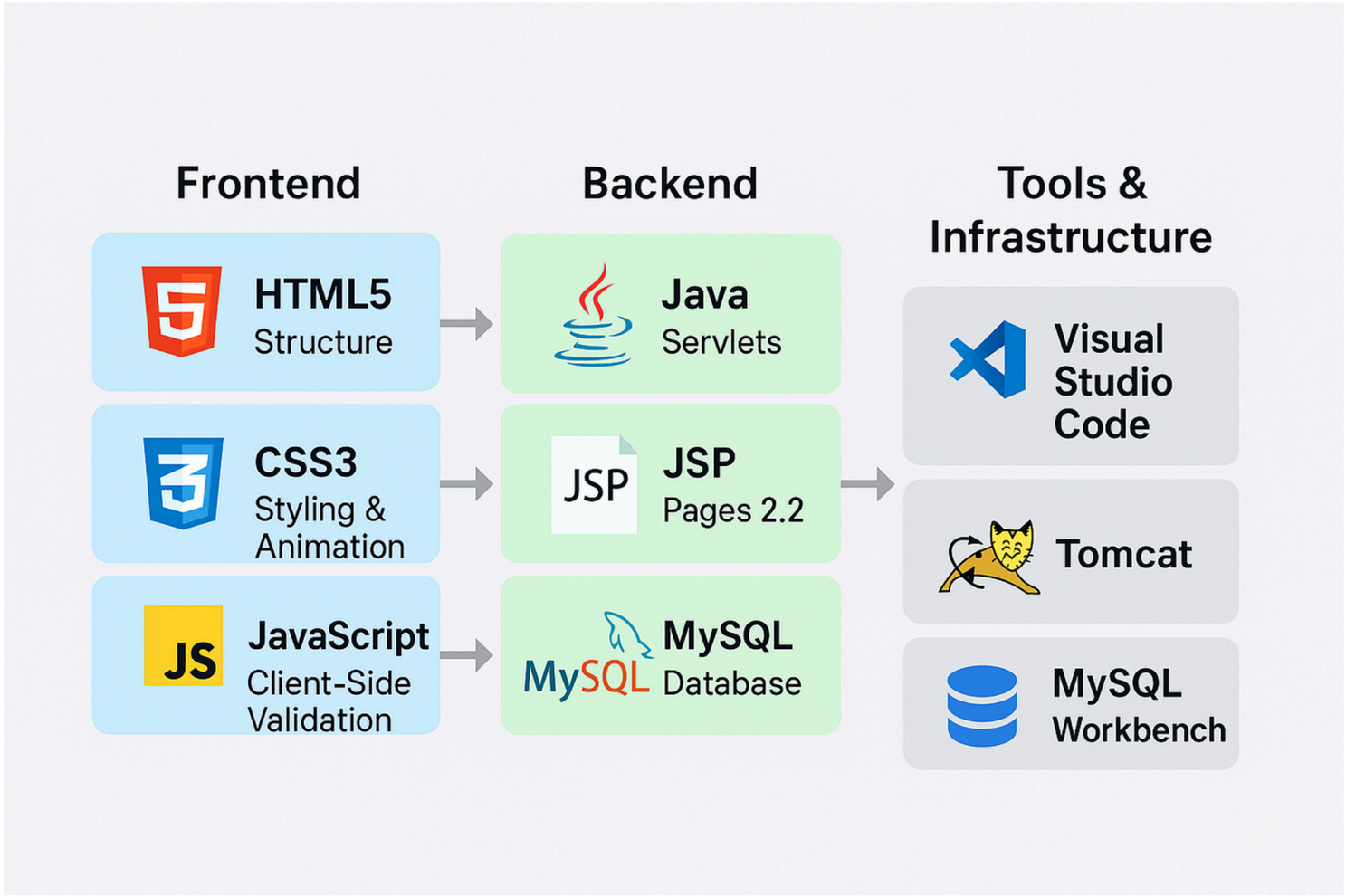
Frontend Technologies:
- HTML5 for structure
- CSS3 for styling and animations
- JavaScript for client-side validation

Backend Technologies:
- Java JDK 11
- Java Servlets API 4.0
- JSP (JavaServer Pages) 2.2

Database:
- MySQL 8.0
- JDBC Driver for connectivity

Server:
- Apache Tomcat 9.0

Development Tools:
- Visual Studio Code
- MySQL Workbench



Implementation - MVC Pattern

Model-View-Controller Architecture:

Model:

- Expense entity class with encapsulation
- Getter/setter methods for data access

View:

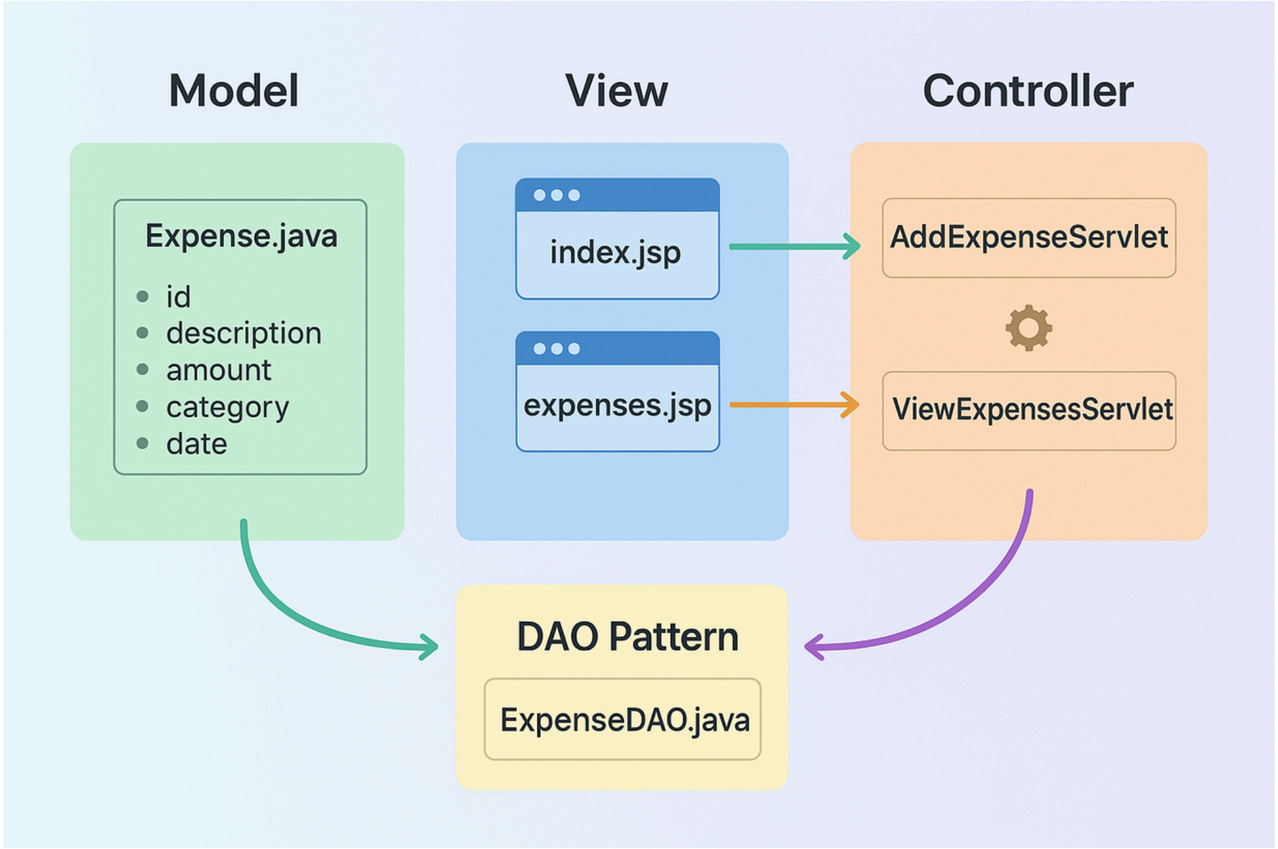
- index.jsp - Expense entry form
- expenses.jsp - Expense list display

Controller:

- Servlets process HTTP requests
- Forward data to appropriate views
- Handle business logic

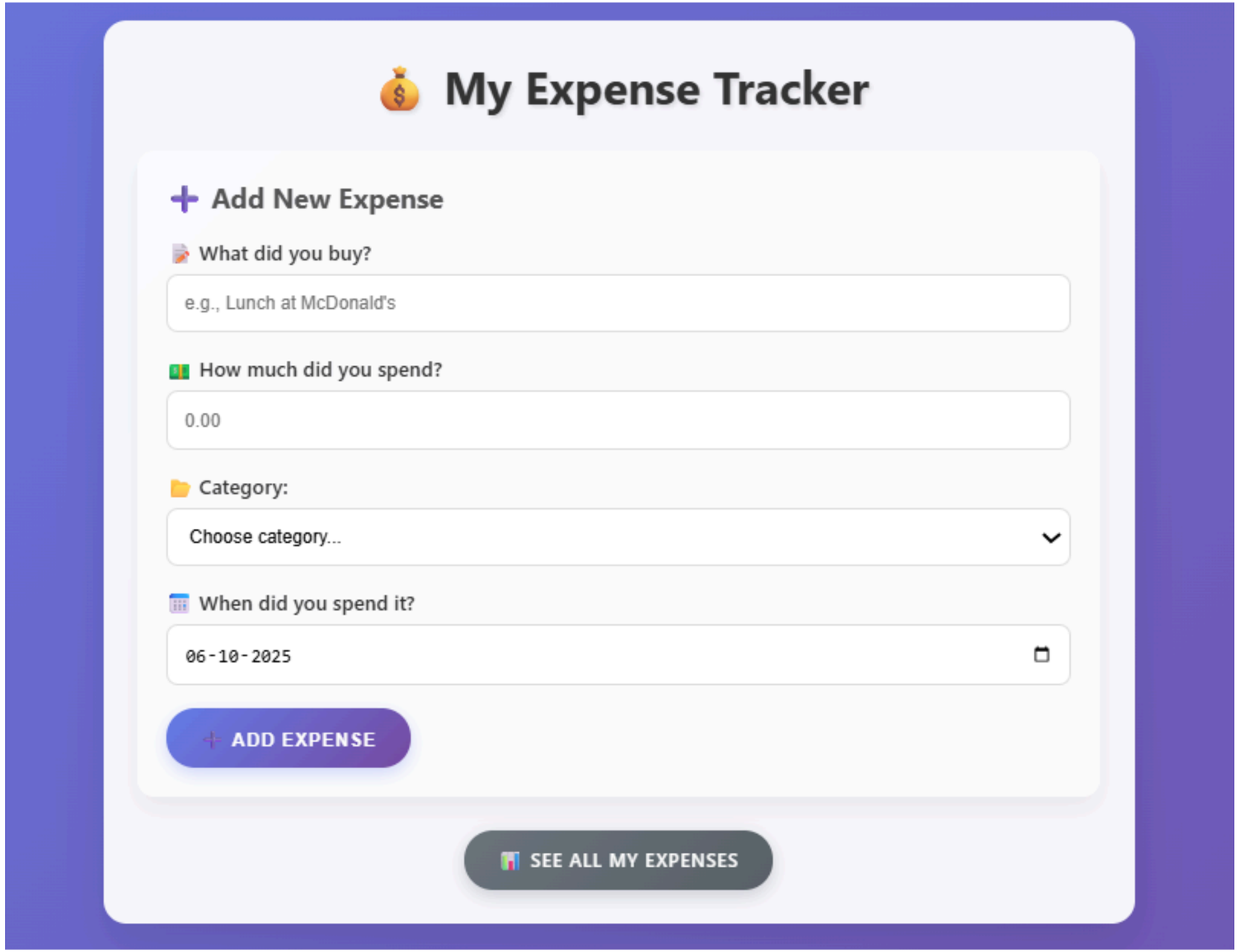
DAO Pattern:

- Separates database operations
- Reusable data access methods



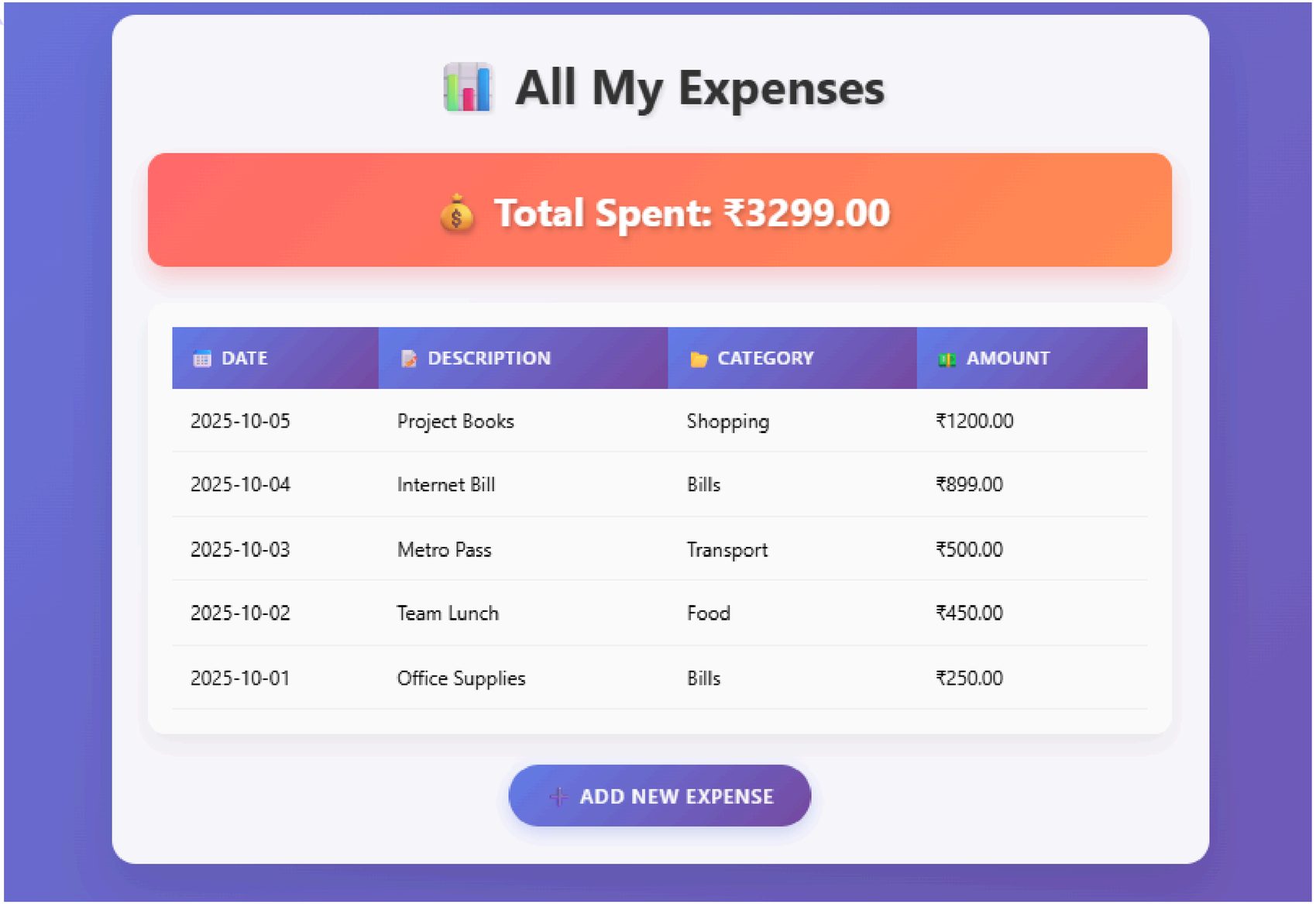
Key Features Implemented

Add Expense Functionality:
- Form with description, amount, category, date fields
- Client-side validation using JavaScript
- Server-side processing via servlet
- Success/error message display



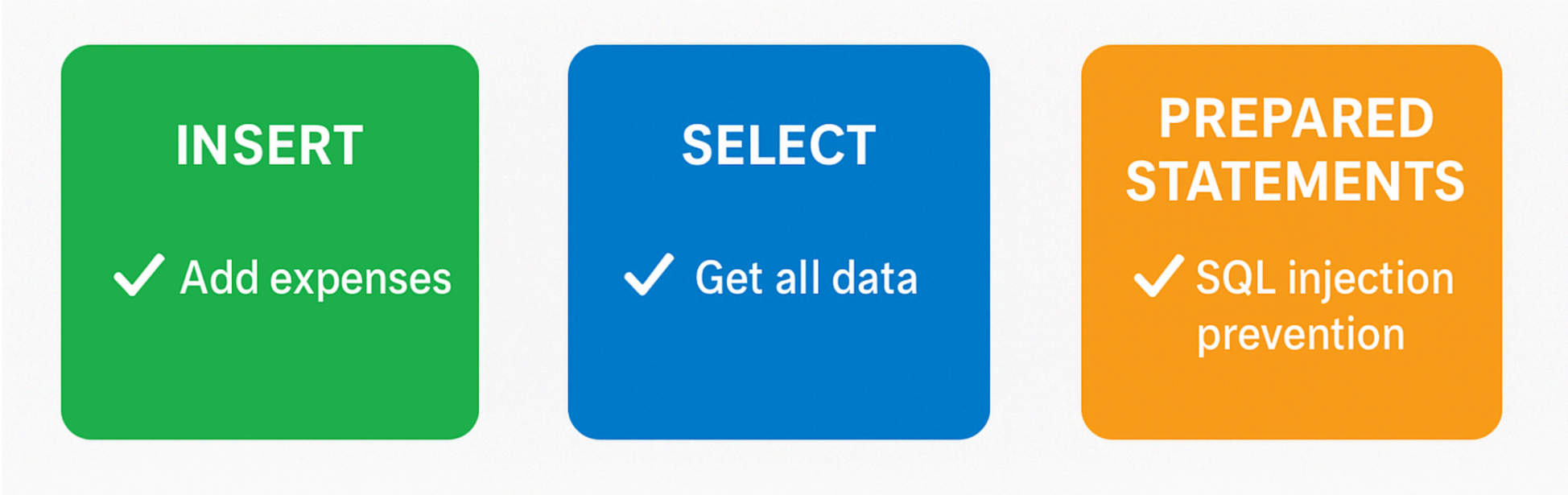
Key Features Implemented

View Expenses Functionality:
- Tabular display of all expenses
- Automatic total calculation
- Sorted by date (newest first)
- Professional UI with gradient design



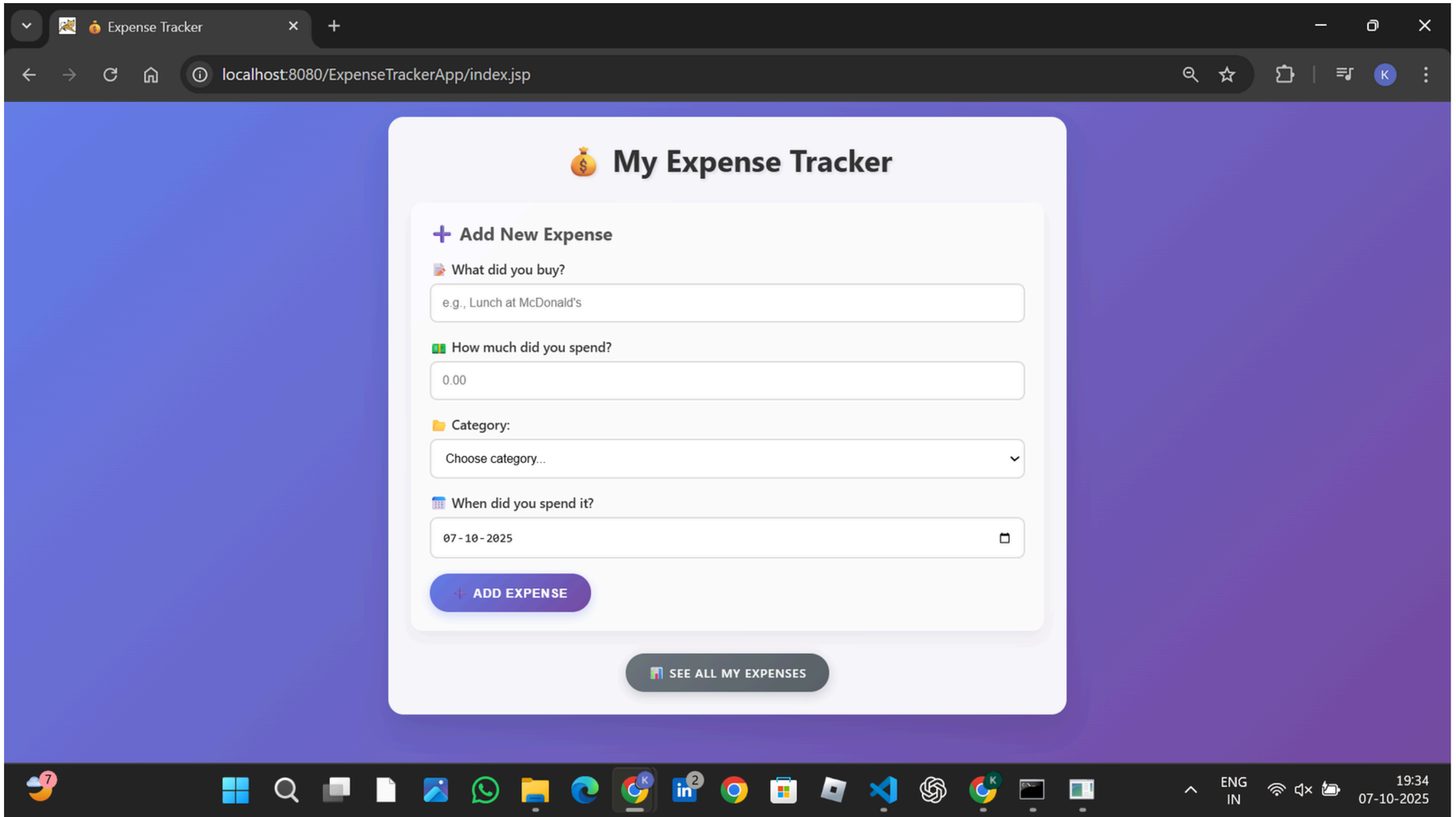
Key Features Implemented

Database Operations:
- INSERT for adding expenses
- SELECT for retrieving data
- Prepared statements for SQL injection prevention



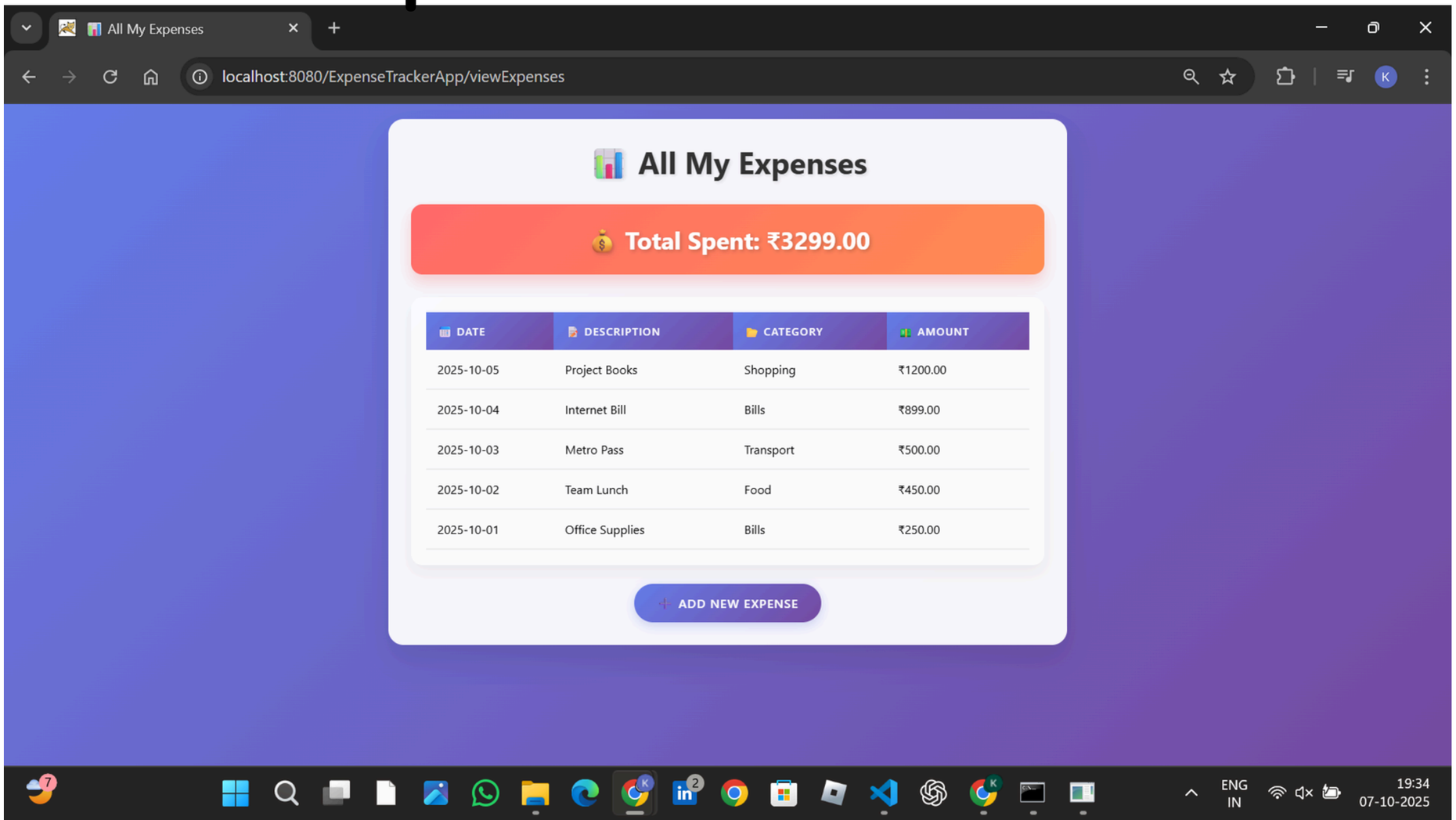
Implementation Screenshots

1. Add Expense Form Interface



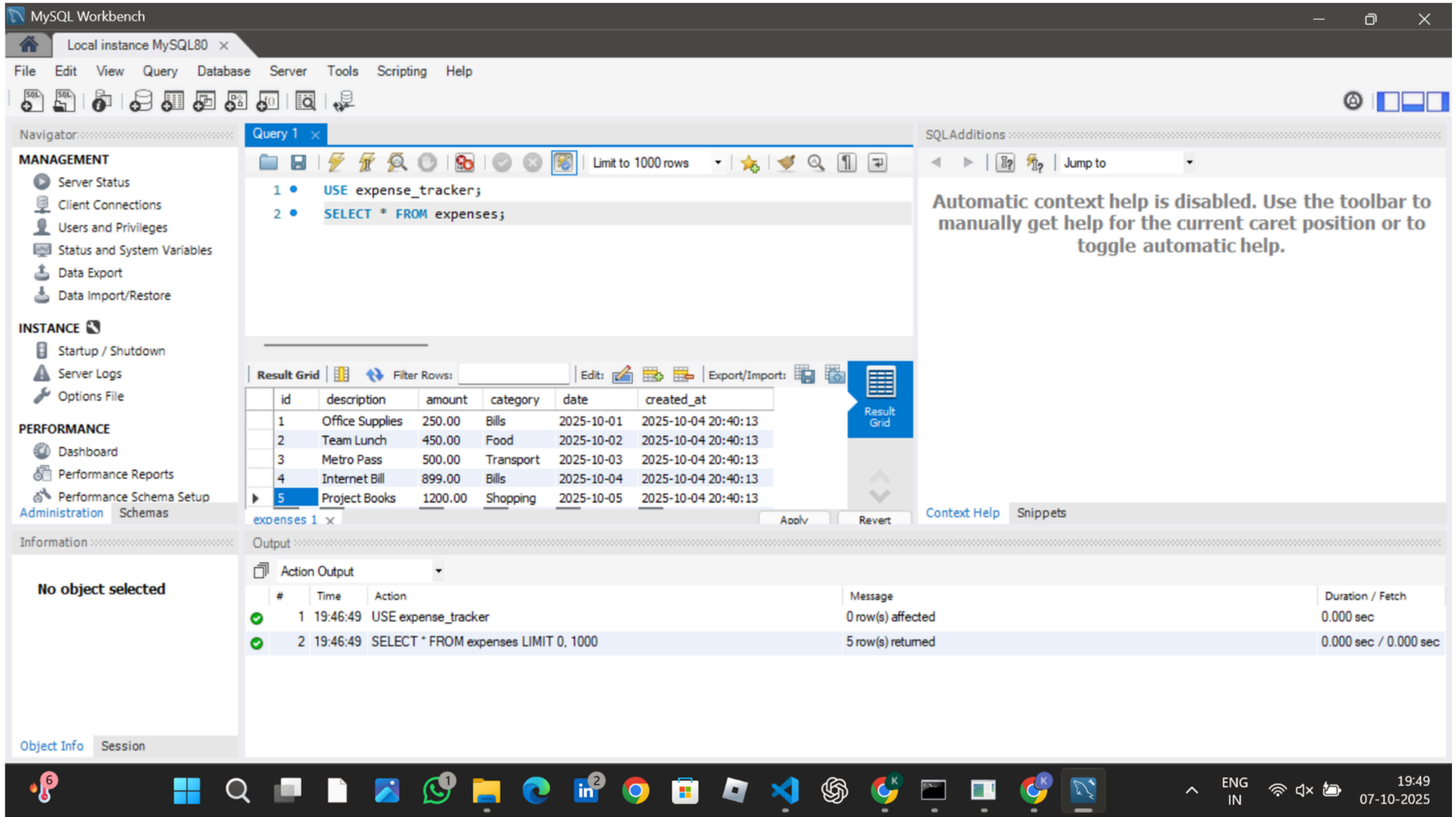
Implementation Screenshots

2. Expenses List with Total



Implementation Screenshots

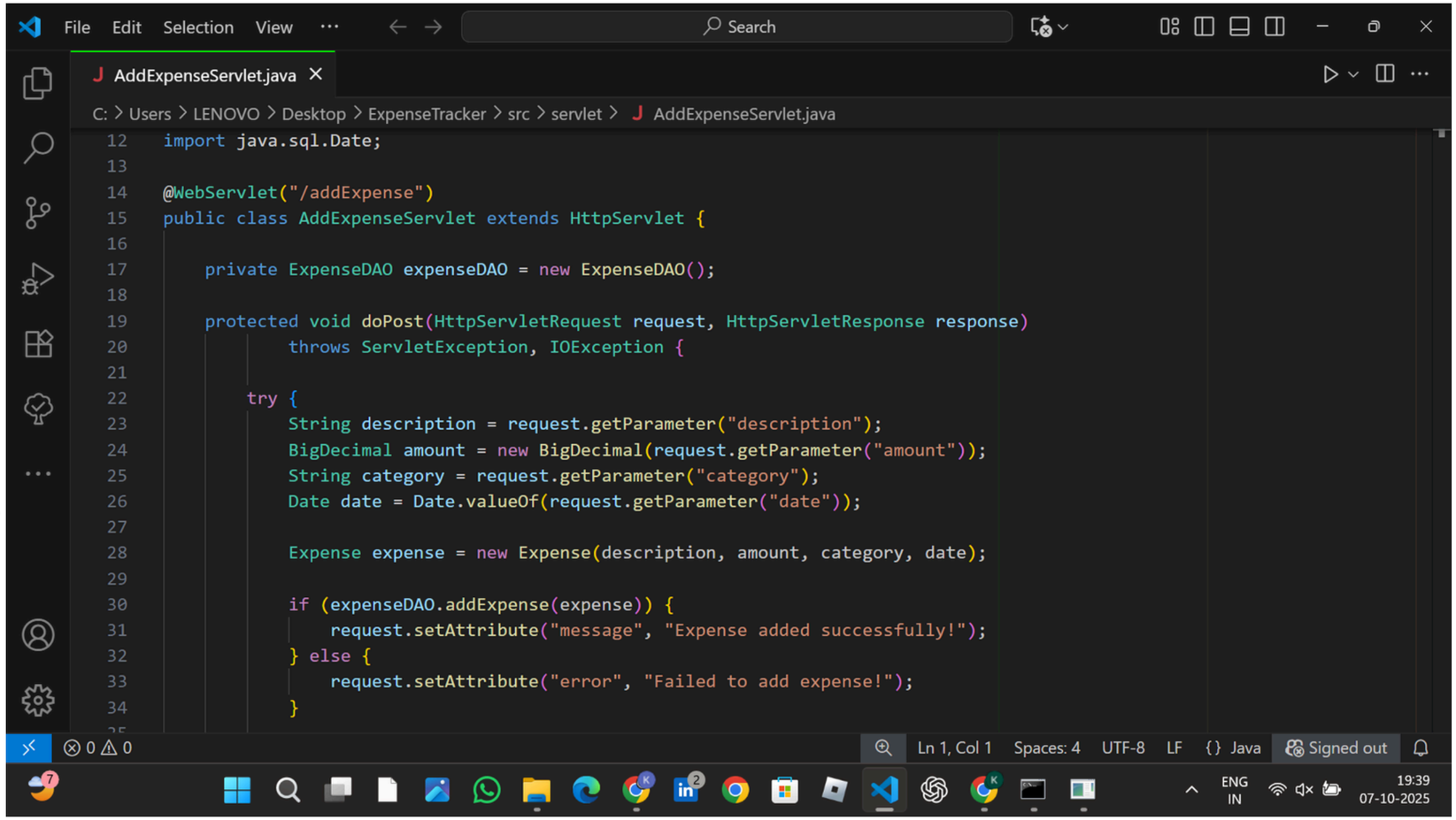
3. Database Table in MySQL Workbench



Code Implementation Highlights

Servlet Example - AddExpenseServlet:

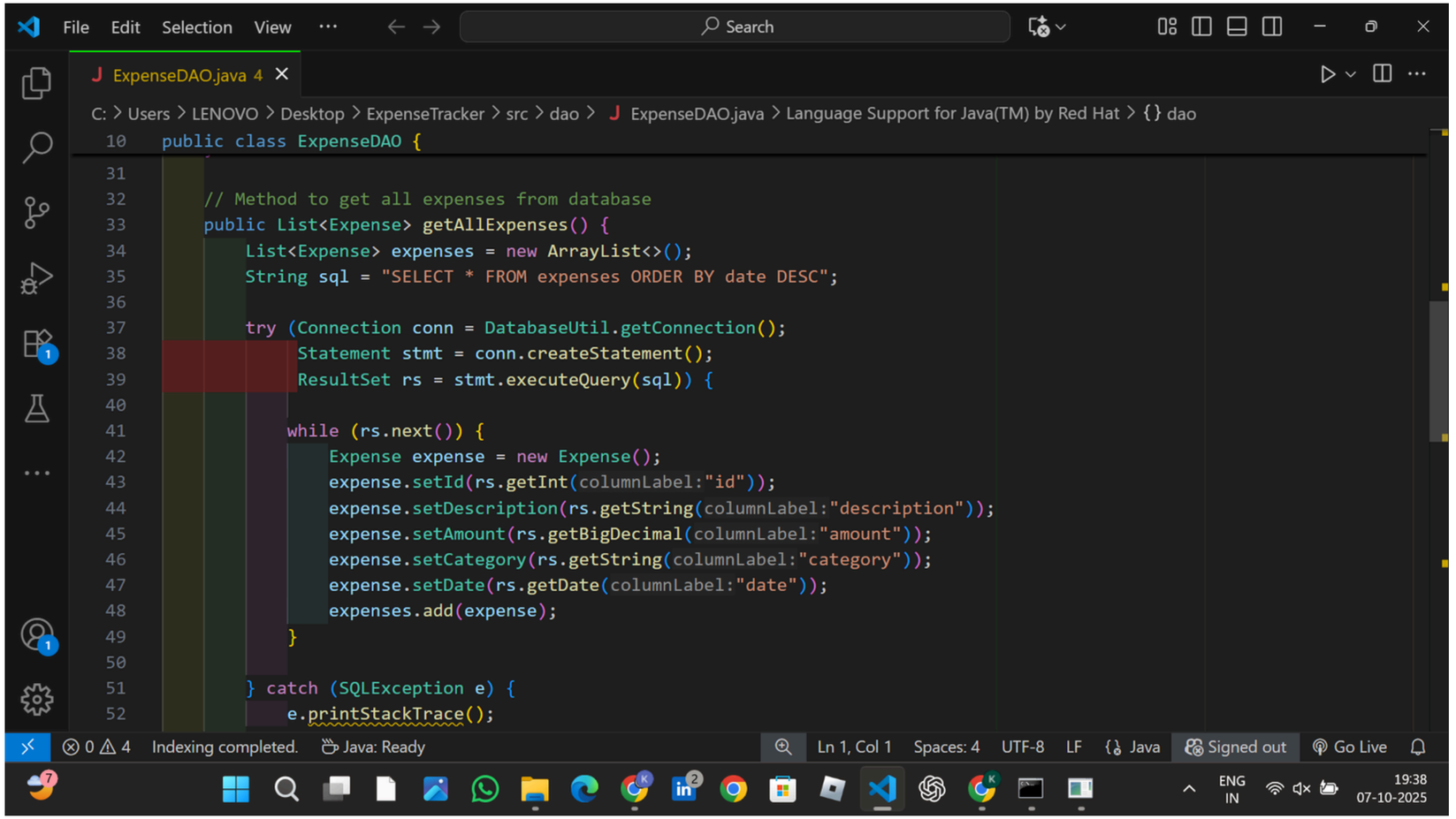
- doPost() method handles form submission
- Retrieves parameters from request
- Creates Expense object
- Calls DAO to persist data
- Forwards to JSP with status message



Code Implementation Highlights

DAO Example - Database Operations:

- getConnection() establishes DB link
- PreparedStatement prevents SQL injection
- ResultSet processes query results
- Exception handling for errors



Testing and Results

Functional Testing:

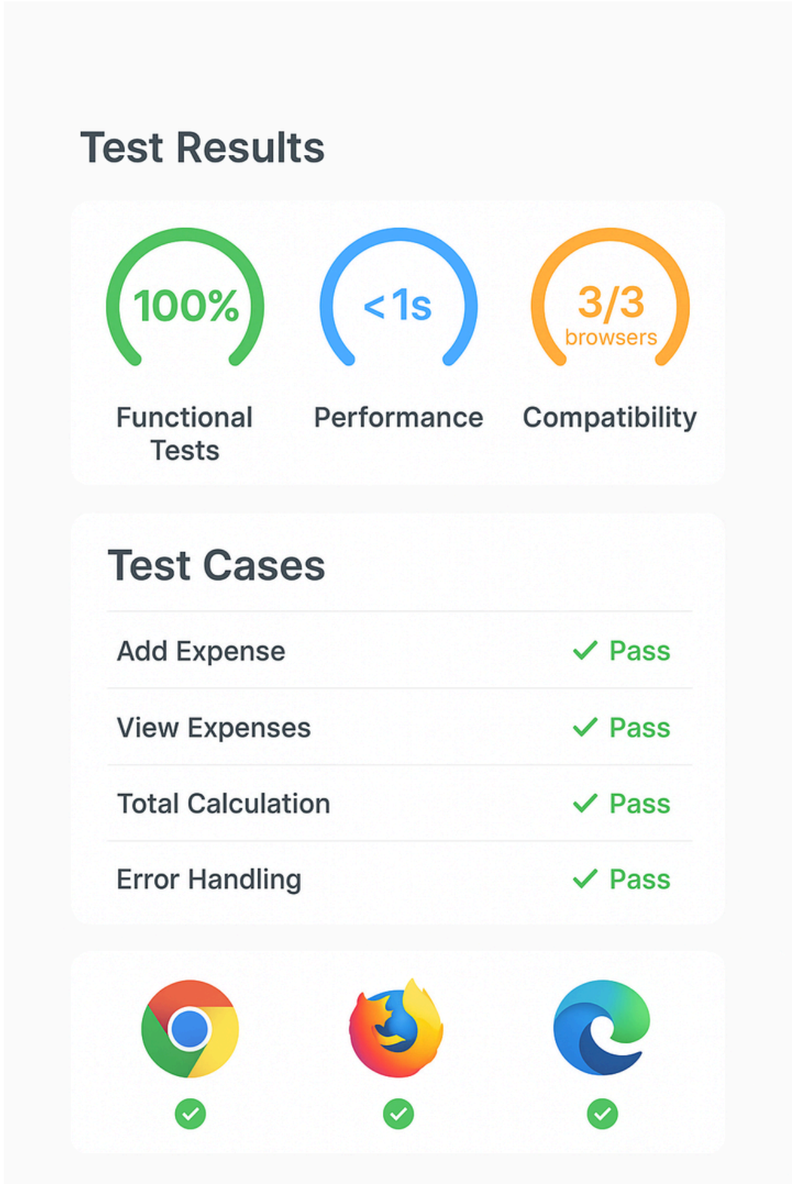
- Successfully adds expenses to database
- Correctly displays all stored expenses
- Accurate total calculation
- Proper error handling

Performance:

- Fast response time (<1 second)
- Efficient database queries
- Smooth user experience

Browser Compatibility:

- Tested on Chrome, Firefox, Edge
- Responsive design works on different screen sizes



Conclusion

Achievements

- Developed fully functional expense tracking system
- Implemented three-tier architecture successfully
- Integrated frontend with backend seamlessly
- Achieved persistent data storage using MySQL

Learning Outcomes

- Practical understanding of Java web development
- Hands-on experience with servlets and JSP
- Database connectivity using JDBC
- Full-stack application deployment

Future Enhancements

- User authentication system
- Delete and edit expense functionality
- Expense filtering by date range
- Data visualization with charts
- Export to PDF/Excel

References

[1] Oracle, "Java Servlet Technology Overview", Oracle Documentation, 2024. https://docs.oracle.com/javaee/7/tutorial/servlets.htm

[2] Oracle, "JavaServer Pages Technology", Oracle Documentation, 2024. https://docs.oracle.com/javaee/7/tutorial/jsps.htm

[3] MySQL, "MySQL 8.0 Reference Manual", MySQL Documentation, 2024. https://dev.mysql.com/doc/refman/8.0/en/

[4] Apache Software Foundation, "Apache Tomcat 9 Documentation", 2024. https://tomcat.apache.org/tomcat-9.0-doc/

[5] Oracle, "JDBC Database Access", Oracle Java Tutorials, 2024. https://docs.oracle.com/javase/tutorial/jdbc/

THANK YOU