Expense Tracker Application

Project Overview

The Expense Tracker Application is a full-stack web application designed to help users record, categorize, analyse, and visualize personal expenses securely.

The system provides end-to-end functionality, starting from user authentication to real-time expense insights, using a Spring Boot backend and a modern JavaScript-based frontend.

This project demonstrates real-world full-stack development, secure REST API design, database integration, and interactive dashboards.

Project Goals & Objectives

The primary objectives of this project are:

Enable secure user registration and login using JWT

Allow users to add, update, and delete expenses

Categorize expenses for better financial organization

Provide visual insights through charts and dashboards

Ensure data security and access control

Build a scalable architecture suitable for production deployment

Key Features Implemented

Authentication & Security

User Registration

User Login

JWT-based authentication

Role-based API access

Secure REST endpoints

💰 Expense Management

Add new expenses

Edit existing expenses

Delete expenses

Category-wise expense management

Analytics & Visualization

Total expenses summary

Category-wise distribution

Monthly expense trends

Pie Chart – Expense Distribution

Bar Chart – Expenses by Category

Line Chart – Expense Trend over time

🌐 Deployment

Backend deployed on Render

PostgreSQL database on cloud

Environment variables for secure configuration

4️⃣ Dataset & Data Model Summary

Main Entities

- User
- Category

- Expense

Key Fields

- Expense Name
- Amount
- Category
- Date
- Comments
- User ID (mapped securely)

Each expense is associated with:

- One user
- One category

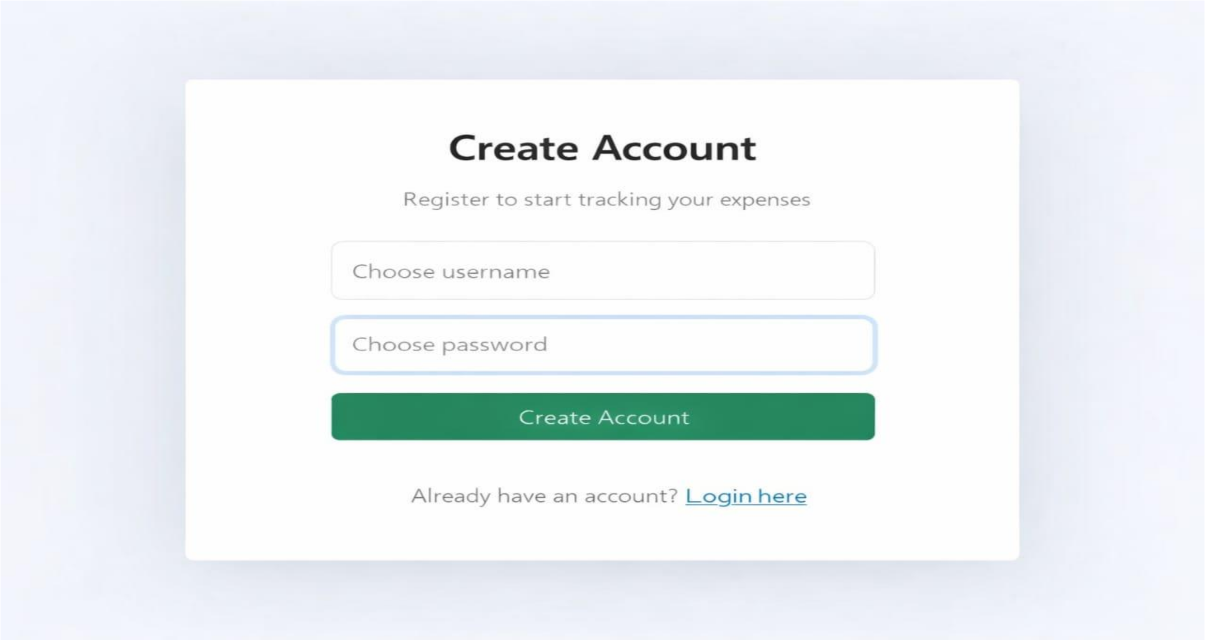
5️⃣ System Architecture (High-Level)

Architecture Layers

Layer	Technology	Purpose
Frontend	HTML, CSS, JavaScript, Bootstrap	User Interface
Backend	Spring Boot, Spring Security	Business Logic
Authentication	JWT	Secure Access
Database	MYSQL / PostgreSQL	Persistent Storage
Deployment	Render	Cloud Hosting

6️⃣ Application Flow (End-to-End)

Step 1: User Registration



Create Account

Register to start tracking your expenses

User creates an account

Credentials are securely stored in the database

Step 2: User Login

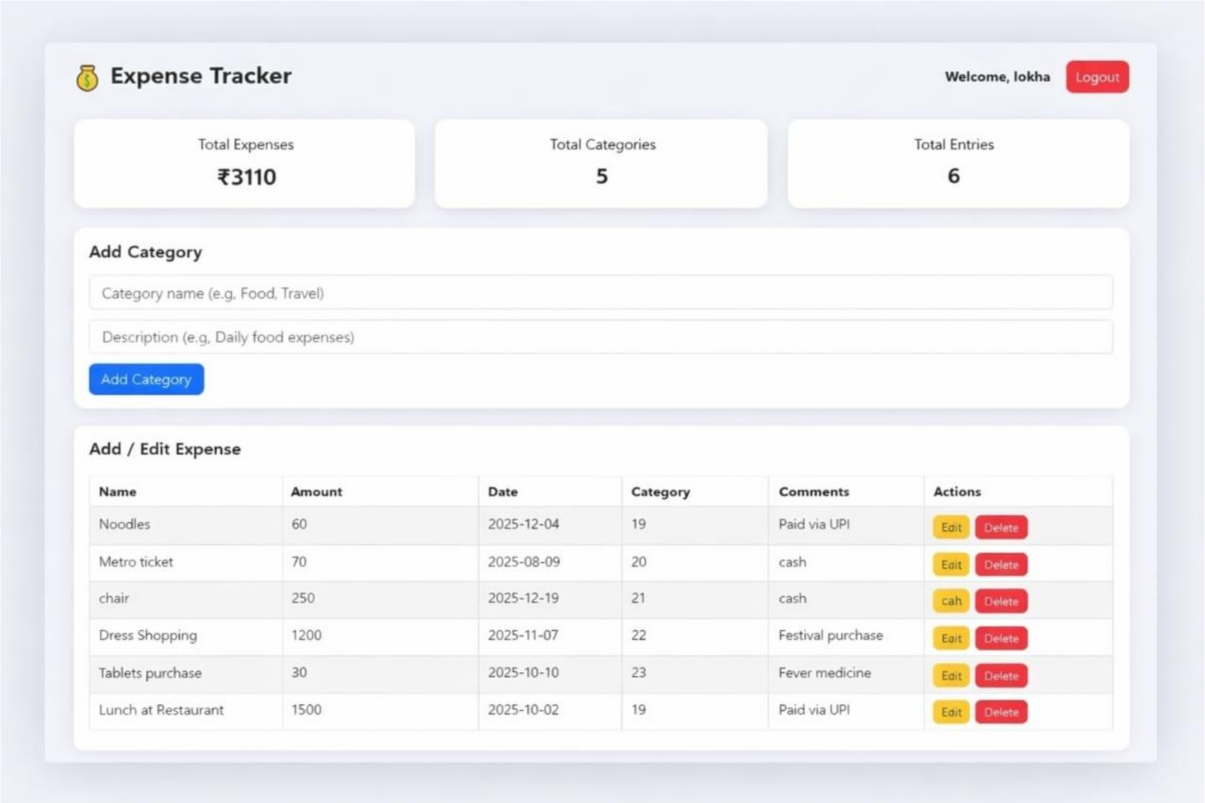
Create Account

Register to start tracking your expenses

JWT token generated upon successful login

Token stored in browser (localStorage)

Step 3: Dashboard Access



Expense Tracker
Welcome, lokha
Logout

Total Expenses	Total Categories	Total Entries
₹3110	5	6

Add Category

Category name (e.g. Food, Travel)
Description (e.g. Daily food expenses)
Add Category

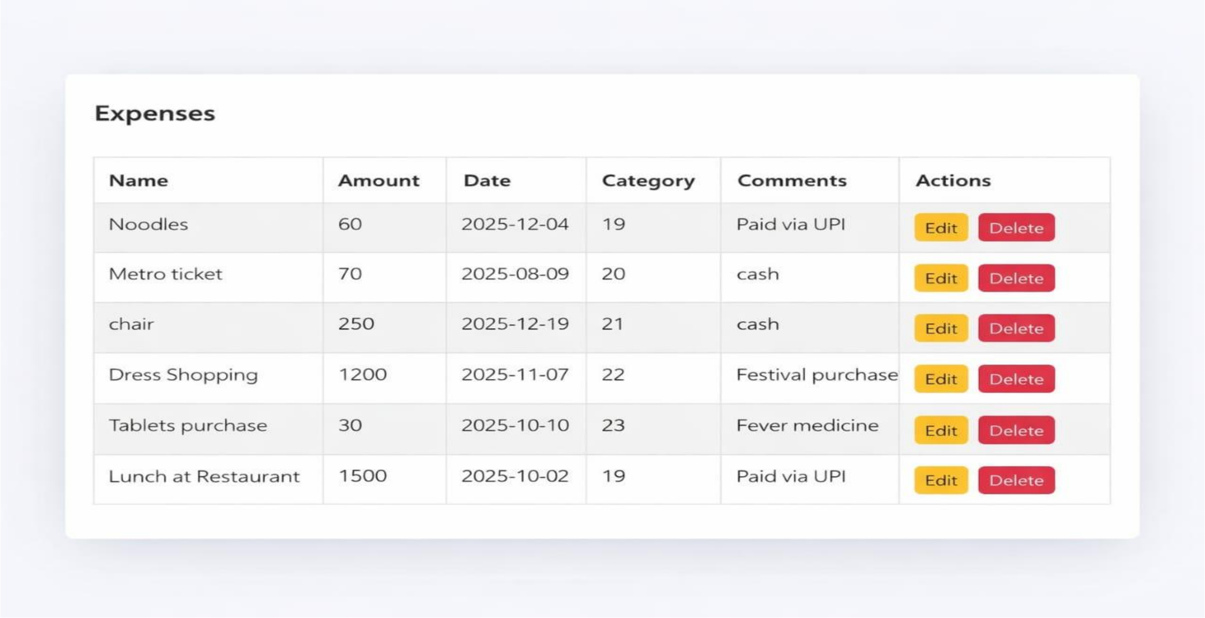
Add / Edit Expense

Name	Amount	Date	Category	Comments	Actions
Noodles	60	2025-12-04	19	Paid via UPI	Edit Delete
Metro ticket	70	2025-08-09	20	cash	Edit Delete
chair	250	2025-12-19	21	cash	cah Delete
Dress Shopping	1200	2025-11-07	22	Festival purchase	Edit Delete
Tablets purchase	30	2025-10-10	23	Fever medicine	Edit Delete
Lunch at Restaurant	1500	2025-10-02	19	Paid via UPI	Edit Delete

Token validated on each API request

User redirected to dashboard after authentication

Step 4: Expense Operations



Expenses

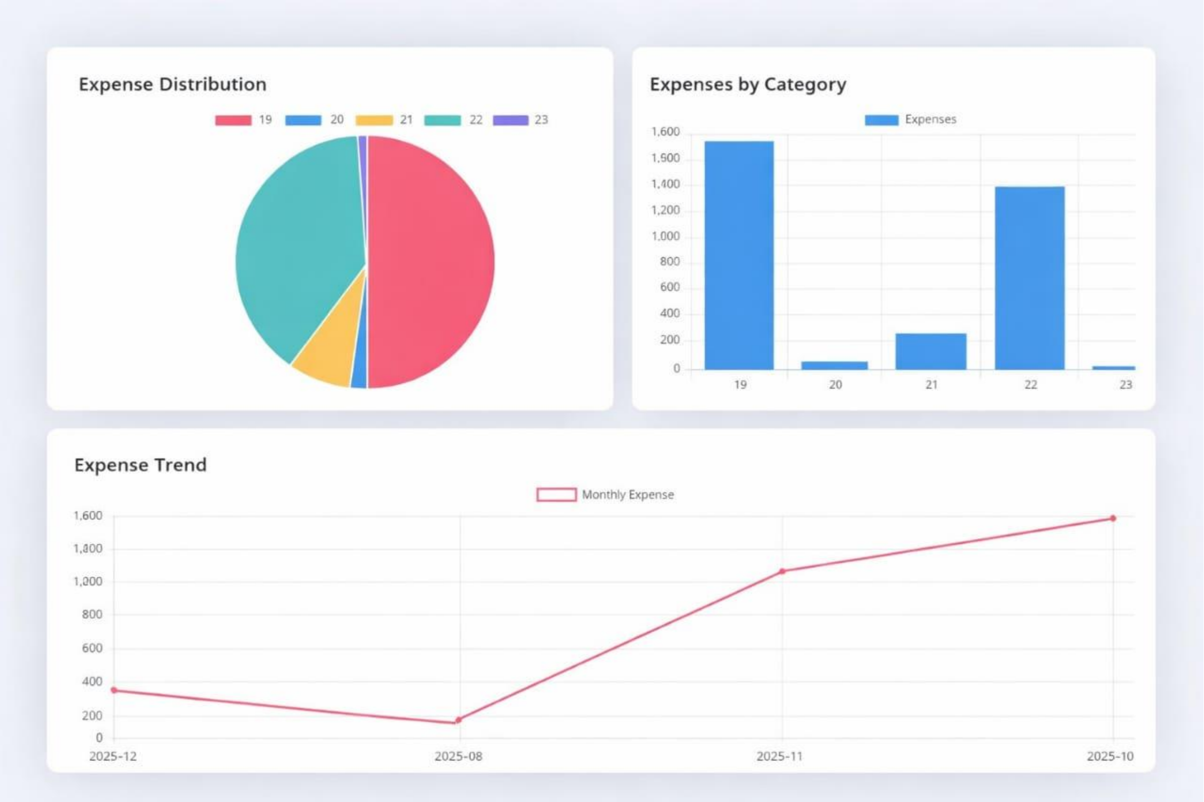
Name	Amount	Date	Category	Comments	Actions
Noodles	60	2025-12-04	19	Paid via UPI	Edit Delete
Metro ticket	70	2025-08-09	20	cash	Edit Delete
chair	250	2025-12-19	21	cash	Edit Delete
Dress Shopping	1200	2025-11-07	22	Festival purchase	Edit Delete
Tablets purchase	30	2025-10-10	23	Fever medicine	Edit Delete
Lunch at Restaurant	1500	2025-10-02	19	Paid via UPI	Edit Delete

Add / Edit / Delete categories

Add / Edit / Delete expenses

Data stored in PostgreSQL

Step 5: Analytics & Visualization



Backend sends expense data via REST APIs

Frontend renders charts dynamically using Chart.js

7 REST API Design

Authentication APIs

POST /api/v1/auth/register

POST /api/v1/auth/login

Category APIs

GET /api/v1/category

POST /api/v1/category

PUT /api/v1/category/{id}

DELETE /api/v1/category/{id}

Expense APIs

GET /api/v1/expenses

POST /api/v1/expenses

PUT /api/v1/expenses/{id}

DELETE /api/v1/expenses/{id}

8️⃣ Frontend Dashboard Insights

The dashboard provides:

Total Expenses (₹)

Total Categories

Total Expense Entries

Expense Distribution Pie Chart

Expenses by Category Bar Chart

Monthly Expense Trend Line Chart

These insights help users understand spending patterns and habits.

9️⃣ Security Considerations

Inspired by best practices from the reference project □, the following security measures were applied.

JWT-based authentication

No exposure of sensitive credentials

Environment variables for DB secrets

Backend API protection using Spring Security

Principle of Least Privilege followed

🔮 Future Enhancements (2025 Ready)

Monthly / Yearly filters

Export reports (CSV / Excel)

Dark mode UI

Role-based admin dashboard

AI-based expense categorization

Real-time notifications

Cloud-native microservices architecture

🔟 Learning Outcomes

Through this project, the following skills were strengthened:

Java Full Stack Development

Spring Boot & Spring Security

JWT Authentication

REST API Design

PostgreSQL Integration

Frontend–Backend Integration

Cloud Deployment (Render)

Real-world project structuring

✅ Conclusion

The Expense Tracker Application is a production-ready full-stack project that showcases practical skills required for Java Full Stack Developer roles.

It combines secure backend development, interactive frontend design, and cloud deployment, making it a strong portfolio project.