# Expense Tracker App - Project Plan

## Title Page

• Project Name: Expense Tracker App

• Team Members: Dushyant Bali, Tegveer Singh, Akash Rana, Sai Jeevan, Mandeep Singh, Khagendra, and Parkash

• Git Repository: <https://github.com/ranasault/AWA-lab-3.git>

## Project Goal

This project aims to develop a user-friendly expense tracker application that enables users to efficiently manage their expenses, categorize transactions, and generate reports. The app will feature an intuitive interface and ensure data security.

## Roles and Responsibilities

• Project Manager (Tegveer Singh): Oversees project timeline, assigns tasks, and ensures deadlines are met.

• Front-End Developer(Akash Akash & Sai Jeevan): Develops the user interface using React.js and ensures a smooth user experience.

• Back-End Developer(Prakash): Implements server-side logic and manages data storage.

• Database Administrator (Dushyant Bali & Mandeep Singh): Designs and maintains the database structure for efficient data retrieval.

• QA Tester(Khagendra Bastola): Tests the application for bugs and ensures a high-quality user experience.

• UI/UX Designer (Tegveer Singh): Creates wireframes and ensures the application is visually appealing.

## Project Elements & Tools

• Front-End: React.js, Tailwind CSS

• Back-End: Node.js, Express.js

• Database: MongoDB or PostgreSQL

• Authentication: Firebase Auth or OAuth

• Deployment: Vercel/Netlify for Front-End, AWS/Heroku for Back-End

• Version Control: Git & GitHub

## Timeline & Milestones

• Week 1: Project setup, Git repository creation, initial UI wireframes

• Week 1: Develop authentication & user account management

• Week 2: Implement expense tracking functionality

• Week 2: Data visualization & report generation

• Week 3: Testing & debugging

• Week 3: Deployment & final review

## Risks & Mitigation

• Scope Creep: Clearly define project requirements and adhere to the plan.

• Technical Challenges: Allocate time for research and troubleshooting.

• Time Constraints: Set realistic milestones and regularly review progress.

• Data Security Risks: Implement encryption and secure authentication.

## Communication & Work Plan

• Meetings: Weekly stand-ups via Zoom/Google Meet

• Collaboration: Slack/Discord for daily communication

• Task Management: Trello/Jira for tracking progress

• Code Reviews: Pull requests and peer reviews on GitHub

## Test Plan Development

### Establish and Identify Components

• Define the application components that need testing.

• Determine individual and integrated testing strategies.

### Roles and Responsibilities

• Assign testing responsibilities to team members.

• Identify testers for unit, integration, and system testing.

### Document the Test Plan

• Develop a complete list of test cases.

• Set a timeframe for creating and executing tests.

• Establish procedures for handling failed or inconclusive test results.