# Expense Tracker

# Project Based Learning (PBL) Report

# for the course

**Software Engineering**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

By

**22R11A05D0 P. Varshita**

**Under the guidance of**

Dr. K. Durga Kalyani



**Department of Computer Science and Engineering**

**Accredited by NBA**

**Geethanjali College of Engineering and Technology**

**(UGC Autonomous)**

(Affiliated to J.N.T.U.H, Approved by AICTE, New Delhi) Cheeryal (V), Keesara (M), Medchal.Dist.-501 301.

# November -2024

**Geethanjali College of Engineering and Technology**

**(UGC Autonomous)**

(Affiliated to J.N.T.U.H, Approved by AICTE, New Delhi) Cheeryal

(V), Keesara (M),

Medchal.Dist.-501 301. =

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Accredited by NBA



# Certificate

This is to certify that the B.Tech PBL report entitled “Expense Tracker” is a bonafide work done **P.Varshita-22R11A05D0** in partial fulfillment of the requirement of the award for the degree of Bachelor of Technology in “**Computer Science and Engineering”** from Geethanjali College of Engineering and Technology, Medchal ,Telangana during the year 20242025.

**Internal Guide HOD-CSE**

## K. Durga kalyani Dr. Sree Lakshmi

## 

## TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Contents** | **Page No** |
| **1** | Introduction  1.1 About Project  1.2 Problem Statement  1.3 Project Scope  1.4 Objectives | 1  1  1  2 |
| **2** | System Design  2.1 System Requirements  2.2 Modules  2.3 ER Diagrams  2.4 Use Case Diagram  2.5 DFDs (Data Flow Diagrams) | 3  4  7  8  9 |
| **3** | Implementation | 10 |
| **4** | Output Screens | 17 |
| **5** | Testing  5.1 Types of Testing | 19 |
| **6** | Conclusion | 21 |
| **7** | References | 22 |

1. **INTRODUCTION**

**1.1 ABOUT THE PROJECT**

An expense tracker is a financial management tool designed to help individuals or organizations monitor and manage their expenses effectively. This project focuses on creating a user-friendly application that allows users to record, categorize, and analyze their daily, weekly, or monthly spending. By providing detailed insights through visual reports and summaries, the expense tracker empowers users to identify spending patterns, set budgets, and make informed financial decisions. The goal of this project is to promote better financial discipline and help users achieve their savings goals through a seamless and intuitive interface.

**1.2 PROBLEM STATEMENT**

Managing personal finances and maintaining control over daily expenses can be challenging for many individuals due to the lack of effective tracking tools, inconsistent budgeting practices, and limited visibility into spending habits. This often leads to overspending, financial stress, and difficulty in achieving savings goals. The absence of a centralized and user-friendly system for recording, categorizing, and analyzing expenses makes it harder to make informed financial decisions and prioritize long-term financial stability. An efficient expense tracker is needed to address these challenges by offering a simple, accessible, and reliable solution for monitoring and managing expenses.

**1.3 PROJECT SCOPE**

The scope of the Expense Tracker project encompasses the development of a user-friendly and feature-rich application aimed at helping individuals and organizations manage their finances effectively. The key features and scope include:

1. **Expense Logging:** Users can record daily expenses, including the date, amount, category, and description, for a comprehensive financial overview.
2. **Categorization:** Expenses can be categorized (e.g., food, transportation, utilities) to facilitate better analysis and organization.
3. **Budgeting Tools:** Users can set monthly or category-specific budgets, receive alerts for overspending, and track their progress toward financial goals.
4. **Analytics and Reports:** The application provides visual insights, such as charts and graphs, to help users analyze spending patterns and make data-driven financial decisions.
5. **Multi-Platform Accessibility:** The tracker will be accessible via mobile apps, web platforms, or desktop applications for convenience and flexibility.
6. **Data Security:** Ensuring secure storage of financial data through encryption and user authentication to maintain privacy and trust.
7. **Future Expansion:** Incorporating additional features, such as income tracking, savings management, and integration with bank accounts, to provide a comprehensive financial management tool.

**OBJECTIVES**

The objectives of an expense tracker project are as follows:

1. **Monitor Financial Activity**: Provide users with a simple and effective way to record and track their daily, weekly, and monthly expenses.
2. **Promote Financial Awareness**: Help users gain insights into their spending patterns through detailed analytics and visualizations like graphs and charts.
3. **Facilitate Budget Management**: Allow users to set budgets for specific categories and track their adherence to these budgets, minimizing overspending.
4. **Encourage Savings**: Enable users to identify areas of excessive spending and make informed decisions to save money and achieve financial goals.
5. **Ensure Accessibility**: Offer a user-friendly interface that is accessible across multiple platforms, including mobile devices, web browsers, and desktops.
6. **Enhance Data Security**: Implement robust security measures to protect users' financial data, ensuring privacy and trust.
7. **Support Goal Tracking**: Help users set and track progress toward financial milestones, such as debt repayment, vacation savings, or emergency funds.
8. **Provide Customization**: Allow users to personalize categories, notifications, and reports to better suit their financial habits and preferences.
9. **Encourage Consistent Usage**: Utilize intuitive design and helpful reminders to make financial tracking a consistent part of users’ routines.
10. **Adapt to Future Needs**: Provide a scalable solution that can integrate additional features, such as income tracking, account linking, or multi-currency support, to meet evolving user requirements.

## 2. SYSTEM DESIGN

**2.1 SYSTEM REQUIREMENTS**

### Software Requirements

* **Operating System** - Windows, macOS, or Linux
* **Browser for Portal Access** - A modern web browser (e.g., Chrome, Firefox)
* **Database Simulation** - Simulated SQL or NoSQL databases
* **Front-End**: HTML, CSS, JavaScript, React.js/Angular.js for dynamic user interfaces.
* **Back-End:** Node.js or Python (Django/Flask) for server-side logic.

### Hardware Requirements

**Client-Side (User’s Device)**:

* Device: Smartphone, Tablet, or PC.
* Processor: Minimum 1 GHz.
* RAM**:** At least 2 GB.
* Storage: 100 MB for browser cache and cookies.
* Network: Stable internet connection (minimum 3 Mbps).

**Server-Side:**

* Processor: Quad-core or higher.
* RAM: 16 GB or more.
* Storage: 500 GB SSD (for application and database).
* Bandwidth: 100 Mbps or higher.
* Hosting: Cloud-based server (e.g., AWS, Azure, Google Cloud).

### 2.2 MODULES

Here are the key modules for an expense tracker application:

1. User Management Module

* User registration and login functionality.
* Profile management for personalized settings (currency, preferences, etc.).
* Password management and account recovery.
* Authentication and authorization (e.g., multi-factor authentication).

2. Expense Tracking Module

* Add, edit, or delete expense entries.
* Categorization of expenses (e.g., food, transportation, utilities).
* Tagging for additional organization.
* Multi-currency support for international users.

3. Income Tracking Module

* Record and manage income sources.
* Link income with expense tracking for accurate budgeting.

4. Budget Management Module

* Set budgets for specific categories or overall spending.
* Receive alerts for budget limits.
* Monitor budget utilization in real-time.

5. Analytics and Reporting Module

* Generate detailed reports (daily, weekly, monthly).
* Provide visualizations such as pie charts, bar graphs, and trend lines.
* Offer insights into spending patterns and trends.

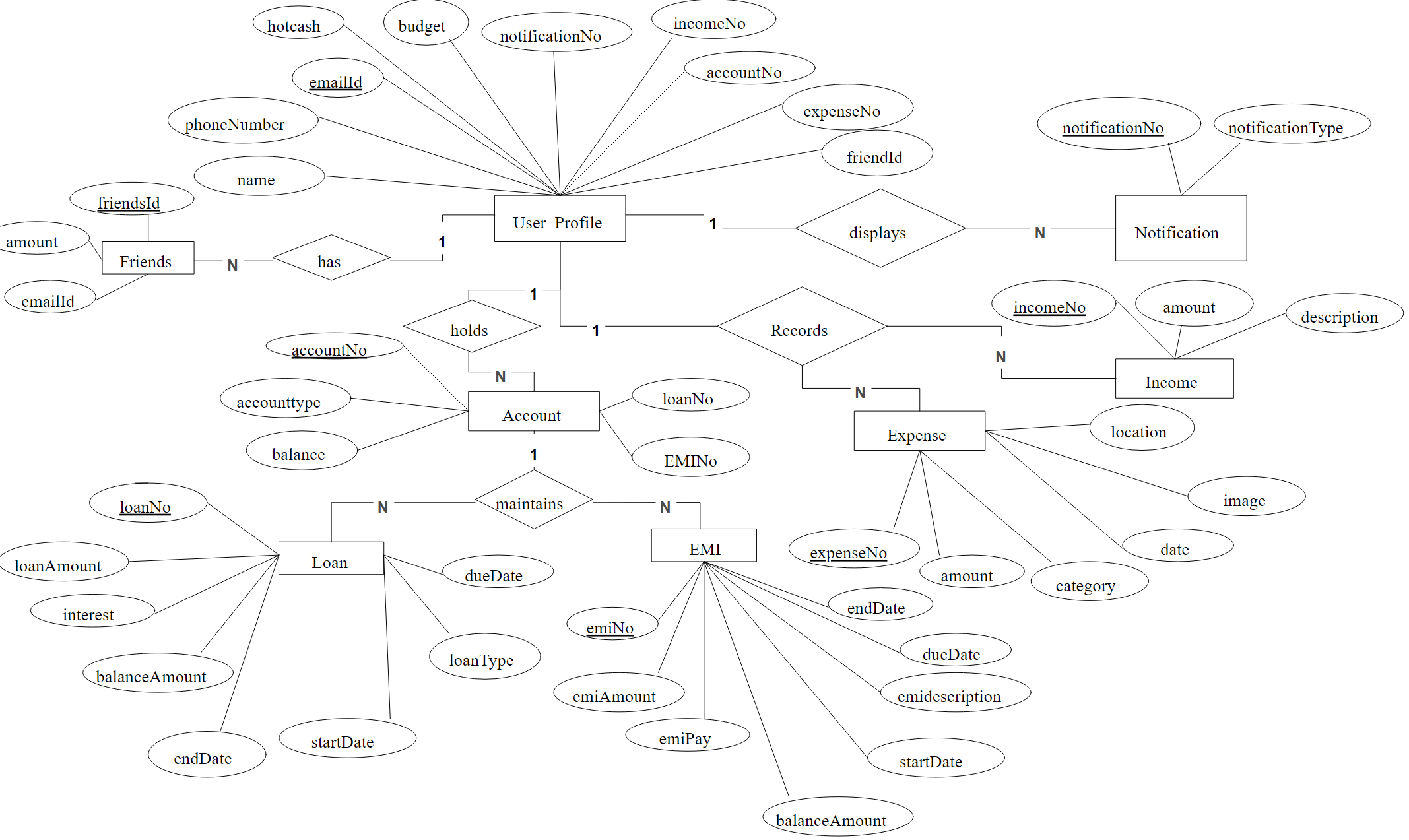
6. Notification and Alerts Module

* Notify users about overspending or nearing budget limits.
* Reminder notifications for logging expenses or setting budgets.
* Customizable alerts for financial goals or milestones.

8. Settings and Customization Module

* Customizable categories and subcategories for expenses.
* Options to set preferred currency and date format.
* Personalization of dashboards and reports.
  1. **2.3 ER DIAGRAM**

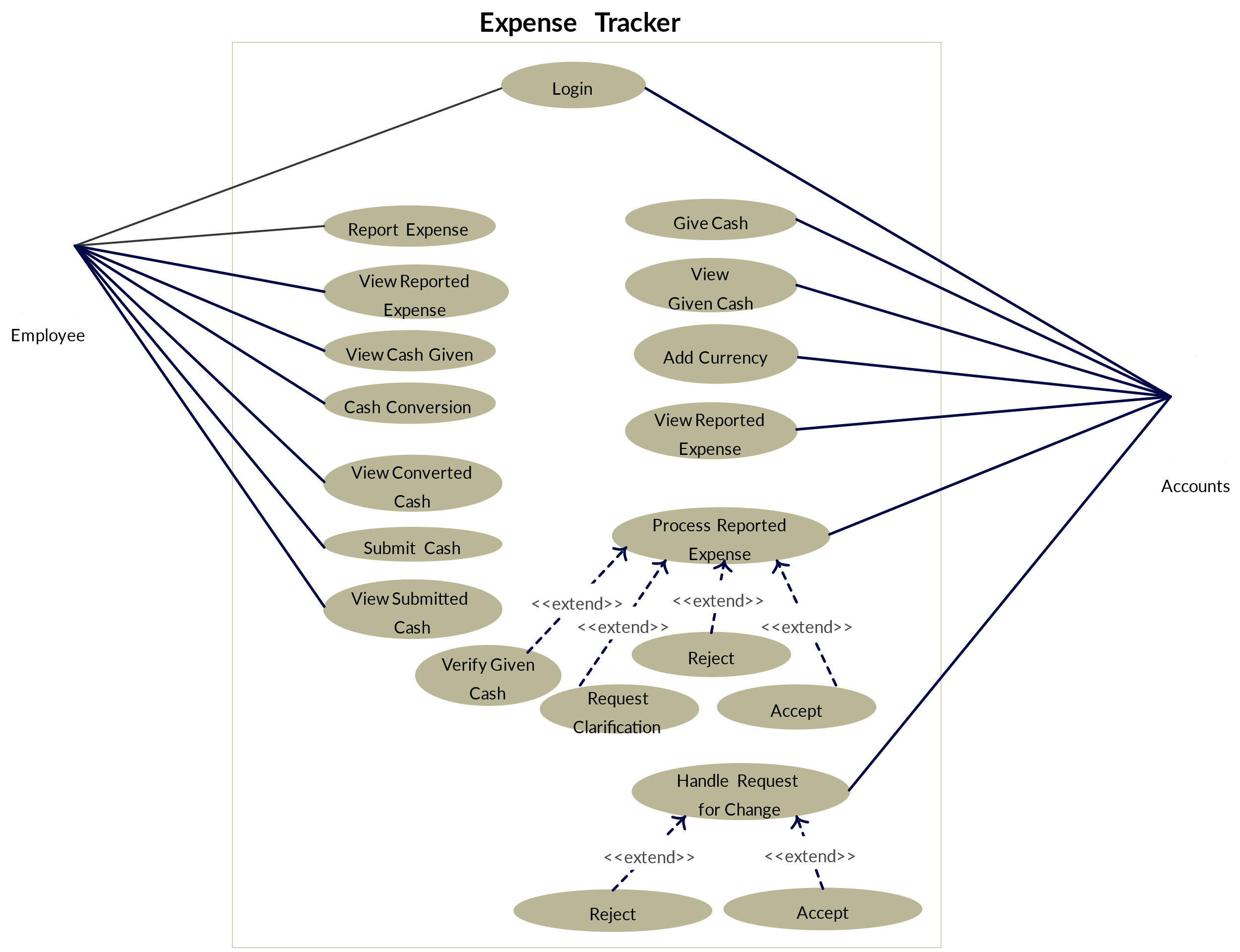
The **ER Diagram** represents the data model of the outlining the Airline reservation system relationships between various entities in the system.



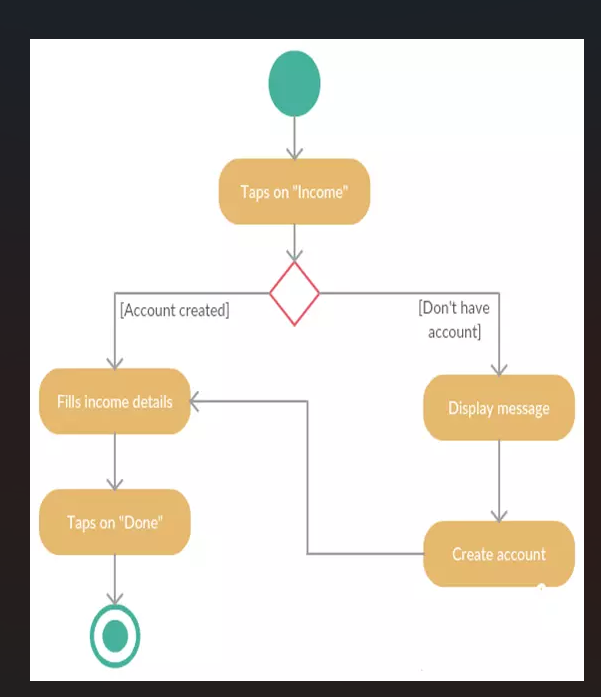


**2.4 USE CASE DIAGRAM**

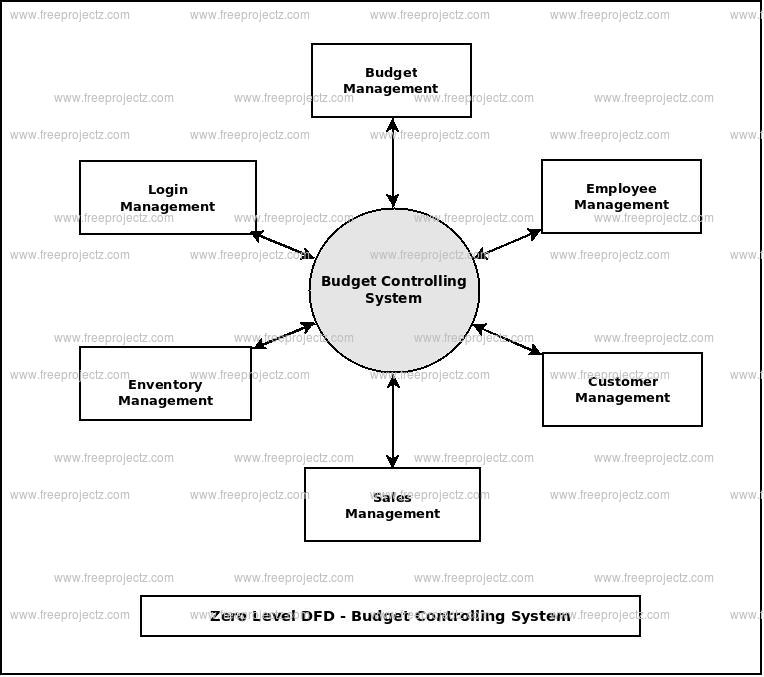
* 1. The Use Case Diagram provides an overview of the interactions between the system and its actors (users and admin). It identifies the primary functions of the website and their stakeholders

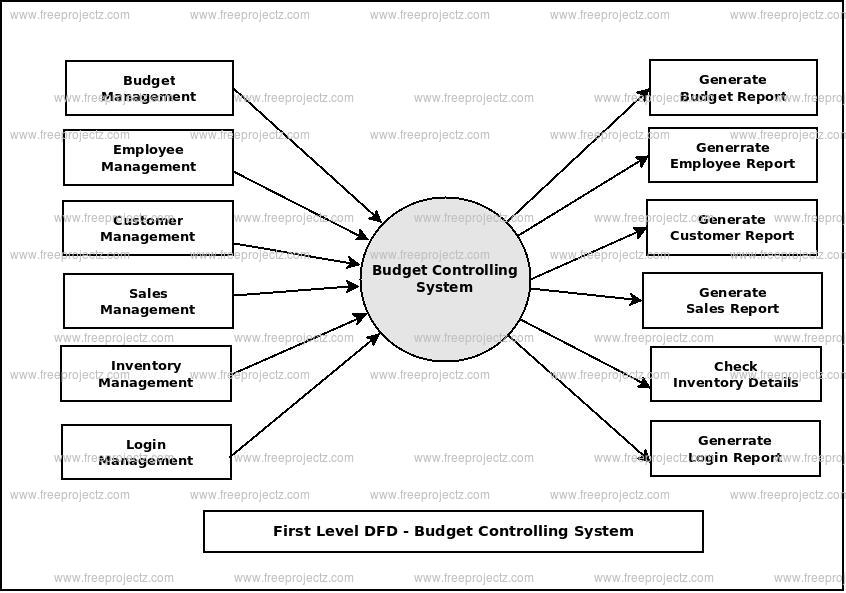


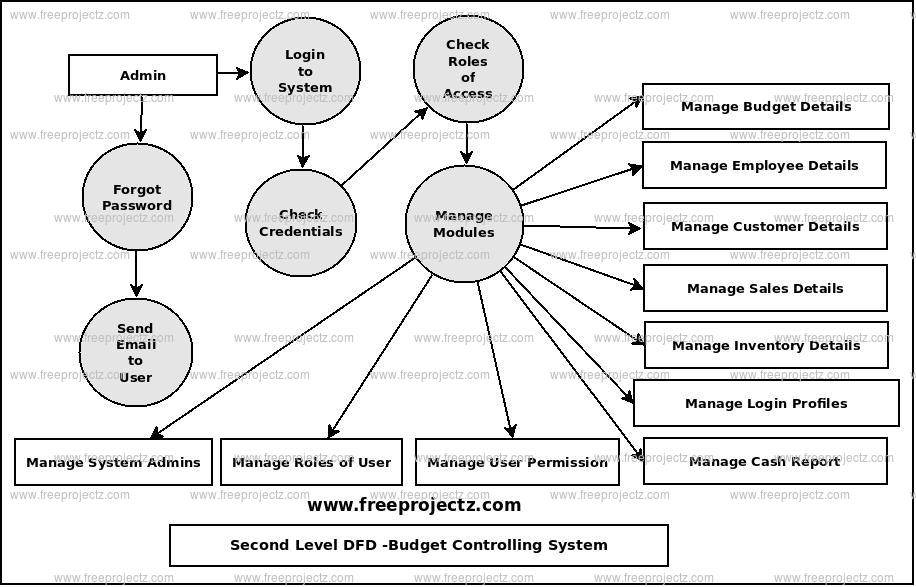
2.5 ACTIVITY DIAGRAM



**2.6 DFDs (DATA FLOW DIAGRAMS)**

****





## 3. IMPLEMENTATION

<!DOCTYPE html>

<html>

<head>

<title>Expense Tracker</title>

<link rel="stylesheet" type="text/css" href="style.css" />

</head>

<body>

<div class="container">

<h1>Expense Tracker</h1>

<form id="expense-form">

<input type="text" id="expense-name" placeholder="Expense Name" required />

<input type="number" id="expense-amount" placeholder="Amount" required />

<select id="expense-category" required>

<option value="" disabled selected>Select Category</option>

<option value="Food">Food</option>

<option value="Transport">Transport</option>

<option value="Entertainment">Entertainment</option>

<option value="Other">Other</option>

</select>

<input type="date" id="expense-date" required />

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

</form>

<div class="expense-table">

<table>

<thead>

<tr>

<th>Expense Name</th>

<th>Amount</th>

<th>Category</th>

<th>Date</th>

<th>Action</th>

</tr>

</thead>

<tbody id="expense-list"></tbody>

</table>

<div class="total-amount">

<strong>Total:</strong> $<span id="total-amount">0</span>

</div>

</div>

<div class="filter">

<label for="filter-category">Filter by Category:</label>

<select id="filter-category">

<option value="All">All</option>

<option value="Food">Food</option>

<option value="Transport">Transport</option>

<option value="Entertainment">Entertainment</option>

<option value="Other">Other</option>

</select>

</div>

</div>

<script src="script.js"></script>

</body>

</html>

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

margin: 0;

padding: 0;

}

.container {

max-width: 800px;

margin: 50px auto;

padding: 20px;

background-color: #fff;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

border-radius: 8px;

}

h1 {

text-align: center;

margin-bottom: 20px;

}

form {

display: flex;

gap: 10px;

margin-bottom: 20px;

}

form input, form select, form button {

padding: 10px;

border: 1px solid #ccc;

border-radius: 4px;

}

form button {

background-color: #28a745;

color: white;

border: none;

cursor: pointer;

}

form button:hover {

background-color: #218838;

}

.expense-table table {

width: 100%;

border-collapse: collapse;

margin-bottom: 20px;

}

.expense-table th, .expense-table td {

padding: 10px;

border: 1px solid #ddd;

text-align: left;

}

.expense-table th {

background-color: #f8f8f8;

}

.total-amount {

text-align: right;

font-size: 18px;

}

.filter {

margin-bottom: 20px;

text-align: right;

}

.filter label {

margin-right: 10px;

}

.filter select {

padding: 10px;

border: 1px solid #ccc;

border-radius: 4px;

}

document.addEventListener("DOMContentLoaded", () => {

    const expenseForm = document.getElementById("expense-form");

    const expenseList = document.getElementById("expense-list");

    const totalAmount = document.getElementById("total-amount");

    const filterCategory = document.getElementById("filter-category");

    let expenses = [];

    expenseForm.addEventListener("submit", (e) => {

        e.preventDefault();

        const name = document.getElementById("expense-name").value;

        const amount = parseFloat(document.getElementById("expense-amount").value);

        const category = document.getElementById("expense-category").value;

        const date = document.getElementById("expense-date").value;

        const expense = {

            id: Date.now(),

            name,

            amount,

            category,

            date

        };

        expenses.push(expense);

        displayExpenses(getFilteredExpenses());

        updateTotalAmount();

        expenseForm.reset();

    });

    expenseList.addEventListener("click", (e) => {

        if (e.target.classList.contains("delete-btn")) {

            const id = parseInt(e.target.dataset.id);

            expenses = expenses.filter(expense => expense.id !== id);

            displayExpenses(getFilteredExpenses());

            updateTotalAmount();

        }

        if (e.target.classList.contains("edit-btn")) {

            const id = parseInt(e.target.dataset.id);

            const expense = expenses.find(expense => expense.id === id);

            document.getElementById("expense-name").value = expense.name;

            document.getElementById("expense-amount").value = expense.amount;

            document.getElementById("expense-category").value = expense.category;

            document.getElementById("expense-date").value = expense.date;

            expenses = expenses.filter(expense => expense.id !== id);

            displayExpenses(getFilteredExpenses());

            updateTotalAmount();

        }

    });

    // Filter by category

    filterCategory.addEventListener("change", () => {

        displayExpenses(getFilteredExpenses());

    });

    // Function to get expenses based on the selected category

    function getFilteredExpenses() {

        const category = filterCategory.value;

        if (category === "All") {

            return expenses;

        } else {

            return expenses.filter(expense => expense.category === category);

        }

    }

    function displayExpenses(expenses) {

        expenseList.innerHTML = "";

        expenses.forEach(expense => {

            const row = document.createElement("tr");

            row.innerHTML = `

                <td>${expense.name}</td>

                <td>$${expense.amount.toFixed(2)}</td>

                <td>${expense.category}</td>

                <td>${expense.date}</td>

                <td>

                    <button class="edit-btn" data-id="${expense.id}">Edit</button>

                    <button class="delete-btn" data-id="${expense.id}">Delete</button>

                </td>

            `;

            expenseList.appendChild(row);

        });

    }

    function updateTotalAmount() {

        const total = getFilteredExpenses().reduce((sum, expense) => sum + expense.amount, 0);

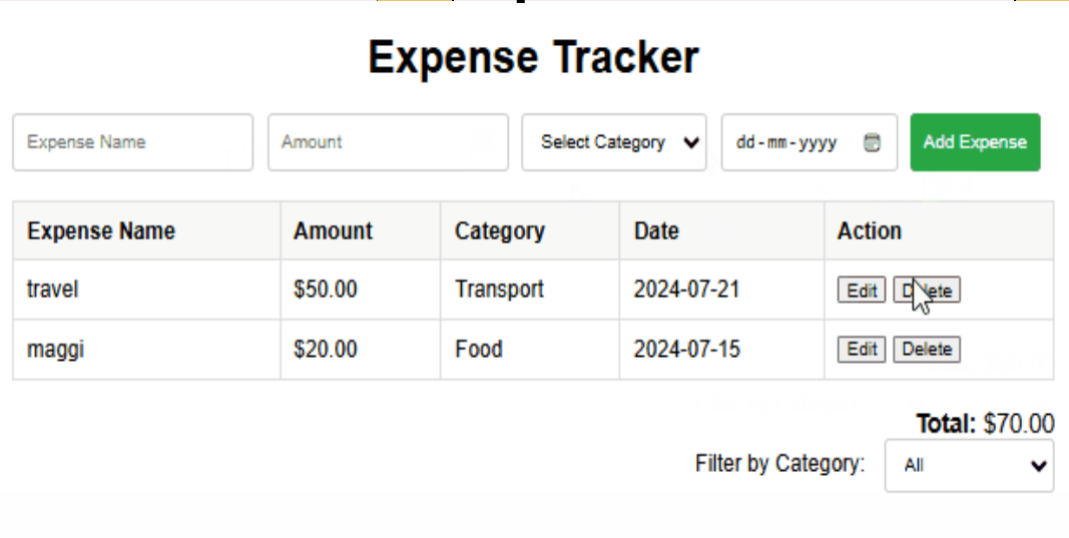
        totalAmount.textContent = total.toFixed(2);

    }

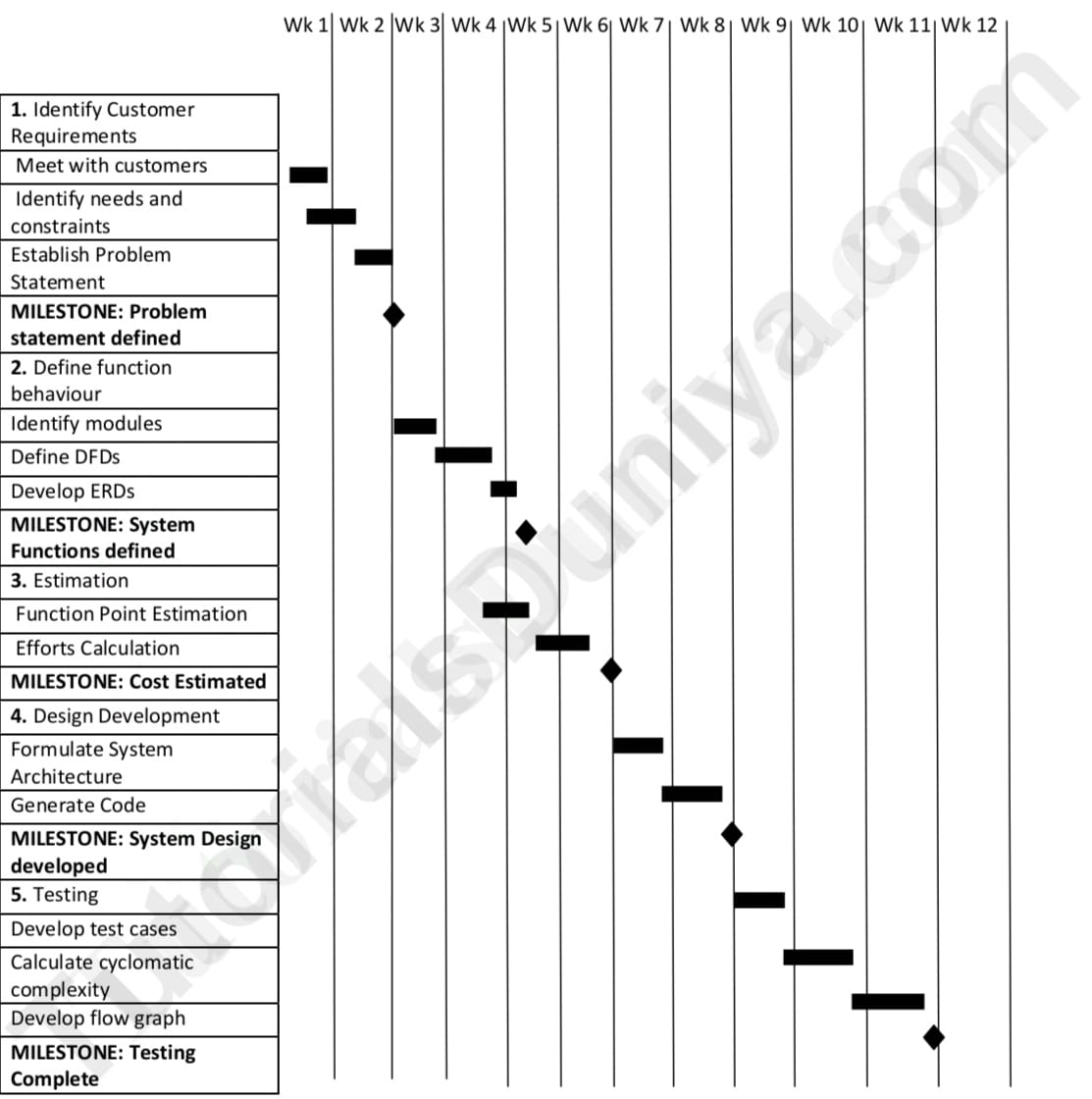
});

## 4. OUTPUT SCREENS

## 



**TIME LINE**



## 5. TESTING

5.1 Types of Testing:

**1. Functional Testing**

* Objective: Verify that all features perform their intended functions.
* Test user registration, login, and profile management.
* Add, edit, and delete expenses or income entries.
* Verify that expense categories and tags function correctly.
* Test budget creation, modification, and alerts for exceeding limits.
* Check data synchronization across devices.

**2. Usability Testing**

* Objective: Ensure the application is user-friendly and intuitive.
* Evaluate the ease of navigation through the app.
* Test the process of adding and categorizing expenses.
* Assess the clarity of reports and visualizations.
* Gather feedback from real users to identify pain points or confusion.

**3. Performance Testing**

* Objective: Test the app’s responsiveness and scalability under different conditions.
* Check load times for dashboards and reports.
* Simulate multiple users adding and accessing data simultaneously.
* Test the app's performance when handling large datasets (e.g., years of expenses).

**4. Security Testing**

* Objective: Ensure user data is safe and secure.
* Verify encryption of sensitive data, such as passwords and financial records.
* Test login authentication, including multi-factor authentication (if implemented).
* Check for vulnerabilities like SQL injection or unauthorized access.

**5. Data Integrity Testing**

* Objective: Validate the accuracy and consistency of stored data.
* Ensure correct calculations for budgets, totals, and spending patterns.
* Verify proper synchronization of data across platforms or devices.
* Test exporting and importing of data for accuracy and format compliance.

**6. Compatibility Testing**

* Objective: Ensure the application works across various platforms and devices.
* Test the app on different operating systems (e.g., Android, iOS, Windows).
* Verify compatibility with different browsers (for web-based versions).
* Test responsiveness on devices of varying screen sizes.

**7. Integration Testing**

* Objective: Ensure seamless interaction between different modules and external services.
* Test integration with bank accounts, credit cards, or third-party payment services.
* Verify cloud sync functionality for data backup and multi-device access.
* Ensure proper functioning of APIs for exporting data or generating reports.

## 5. CONCLUSION

## The Expense Tracker project serves as a practical solution to the growing need for efficient personal financial management. By offering features such as expense recording, budgeting, and insightful analytics, it empowers users to take control of their finances, reduce unnecessary spending, and achieve financial goals. The application’s user-friendly interface, robust security measures, and scalable design make it accessible and reliable for a wide range of users.

## This project not only addresses common financial challenges but also promotes better financial discipline and awareness. Moving forward, the inclusion of advanced features like income tracking, bank integrations, and AI-driven insights can further enhance its functionality and user experience. Overall, the Expense Tracker is a valuable tool for fostering financial stability and long-term planning.

Top of Form

Bottom of Form

## 

## 6. REFERENCES

* <https://en.wikipedia.org/wiki/Expense_management>
* <https://www.geeksforgeeks.org/build-an-expense-tracker-with-html-css-and-javascript/>
* <https://github.com/hassanimran02/ExpenseTracker>