ANUDIP FOUNDATION

A Project Report On

“Expense Tracker Web Application”

Developed by

Name	Student ID
Subham Kumar Prasad	AF0481831

Under Guidance

Of

Rajshri Thete

Index

Expense Tracker Web Application

ABSTRACT

The Expense Tracker Web Application is a simple tool designed to help people manage their money. Many individuals find it hard to keep track of how much they earn and where they spend it. This project allows users to record their daily income and expenses in a clear and organized way, helping them better understand their spending habits.

The application is built using basic web technologies like Python (Flask) for the backend and HTML, CSS, and JavaScript for the frontend. Data is saved locally in JSON files, and users can view summaries and charts showing their total income, expenses, and balance. It also provides features like generating monthly reports in PDF and Excel formats, making financial records easy to access and share.

Overall, this project is a user-friendly and helpful system that can be used by anyone who wants to improve their personal budgeting. It is secure, lightweight, and easy to use, making it a great solution for students, professionals, or anyone who wants better control over their finances without using complex or paid apps.

1. A simple web application to help users track their income and expenses.
2. Designed to improve personal budgeting and financial awareness.
3. Allows users to add, categorize, and view financial transactions.
4. Built using Python (Flask), HTML, CSS, JavaScript, and JSON.
5. Stores data locally in JSON files for lightweight usage.
6. Generates monthly reports in PDF and Excel formats.
7. Shows financial summaries and charts using Chart.js.
8. Secure, easy to use, and works on both desktop and mobile browsers.
9. Best suited for students, professionals, or anyone who wants to manage money better.
10. Free and offline, unlike complex or paid budgeting apps.

ACKNOWLEDGEMENT

The project “Expense Tracker Web Application” is the Project work carried out by

Name	Enrollment No
Subham Kumar Prasad	AF0481831

I would like to express my sincere gratitude to everyone who supported and guided me throughout the development of this project.

First and foremost, I am thankful to **Dr. D.Y. Patil ACS College, Pimpri, Pune** and the **Department of Computer Science** for providing me with the opportunity to work on this industrial training project.

I am especially thankful to **Anudip Foundation for Social Welfare** for giving me real-world exposure and helping me understand the practical applications of web development.

I would also like to thank my mentors, faculty members, and friends for their valuable suggestions, constant support, and encouragement during the entire project development process.

Last but not least, I express my heartfelt thanks to my family for their constant motivation and belief in me.

OBJECTIVE

From a developer's point of view, this project was also a great opportunity to apply practical knowledge of web development using Python (Flask), JavaScript, HTML/CSS, and file handling with JSON. It allowed the integration of front-end and back-end development, user authentication, data visualization, and report generation all essential skills in real-world software development.

Key Factors Driving the Motivation:

1. **Lack of Simple Financial Tools**
   Many available budgeting apps are either too complicated or require payment to access full features. This project aims to fill the gap by offering a simple and free solution that helps users manage their money without confusion or cost.
2. **Unawareness of Spending Habits**
   Most people do not realize how much they are spending daily. A tool that tracks income and expenses can help users clearly see their spending patterns and avoid unnecessary expenses.
3. **Need for Organized Financial Records**
   People often forget to note down small transactions. This application helps them keep an organized digital record that can be accessed anytime, avoiding manual logs or paper-based tracking.
4. **Educational Purpose for Developer**
   Developing this app provided a chance to learn and apply skills in Flask, JavaScript, data handling, and user interface design. It helped understand how to build a complete project from back-end to front-end.

PROBLEM DEFINATION:

In today's fast-paced world, managing personal finances has become increasingly challenging. People often find it difficult to keep track of their income and daily expenses, leading to poor budgeting, overspending, and financial stress. While many digital finance tools are available in the market, they are often too complex, cluttered with unnecessary features, or locked behind paid subscriptions. This discourages everyday users from adopting digital budgeting solutions.

Many users, especially students and working professionals, require a simple yet effective tool that can help them monitor their spending habits, categorize transactions, and visualize their financial data. Existing manual methods like notebooks or spreadsheets are not user-friendly, and they lack automation, alerts, and summaries, making them inefficient for regular use.

There is a need for an easy-to-use, responsive web-based solution that allows users to log income and expenses, generate meaningful reports, and view monthly summaries in a clean and understandable format. Such a system should also ensure data privacy and provide export options to help users keep personal financial records.

This project, the **Expense Tracker Web Application**, is developed to solve these problems by providing a streamlined, free, and accessible platform where users can take control of their financial life in a simple and intuitive way.

The problem this project addresses is the **lack of an efficient, simple, and accessible tool for real-time expense management**. Users often need a platform where they can:

- Add expenses quickly and categorize them
- View summaries of their spending habits
- Set budgets and receive alerts
- Visualize data through charts and graphs
- Ensure their financial data remains secure

Moreover, existing systems may have limitations such as being mobile-only, requiring payment for core features, or lacking customization options. Hence, there is a clear need for a **free, open, and accessible web-based expense tracker** that balances functionality with simplicity.

Purpose / Objective:

Purpose

The purpose of this project is to help users manage their personal finances in an easy and organized way. Many people lose track of their income and daily expenses, which makes it hard to save or plan for the future. This web application allows users to record their income and expenses, so they can understand their financial habits and make better decisions.

Objectives

- To develop a simple and user-friendly platform for tracking income and expenses.
- To allow users to categorize transactions (e.g., food, transport, bills) for better understanding.
- To provide users with summary reports that show their financial status at a glance.
- To ensure that user data is securely stored and easily accessible.
- To help users develop better money management skills through regular tracking.

Goals

1. **Easy Financial Tracking:**
   Users should be able to add, edit, and delete income or expense entries with ease.
2. **Clear Visualization:**
   The app should offer summary views like total income, total expense, and remaining balance, along with graphs and charts.
3. **Data Export:**
   Users should be able to download their monthly financial reports in PDF and Excel formats for record-keeping or offline access.
4. **Secure Access:**
   The application should include user authentication features like login and password protection to keep financial data private.
5. **Responsiveness:**
   The app should work smoothly on desktops, tablets, and mobile devices, making it accessible from anywhere.

FEASIBILITY STUDY

1. Introduction

A feasibility study is conducted to determine whether a proposed project is practical and achievable with the available resources, technology, and time. It helps identify possible obstacles, risks, and the overall success rate of the project before starting its development. For the Expense Tracker Web App, the study focuses on its technical, operational, economic, and legal aspects.

2. Purpose of the Feasibility Study

The main purpose of this feasibility study is to evaluate whether the Expense Tracker Web Application can be developed and implemented effectively within a given scope and limitations. It aims to:

- Identify technical requirements and constraints.
- Determine if the solution meets user needs.
- Ensure the project is cost-effective.
- Analyze legal and operational impacts.
- Minimize risk and increase the chances of project success.

3. Feasibility Types

3.1 Technical Feasibility

This assesses whether the current technology is sufficient to build the application. In this project:

- The app is built using Python (Flask), HTML, CSS, JavaScript, and JSON.
- All technologies are open-source and well-documented.
- Tools like Chart.js, ReportLab, and OpenPyXL support features like charts and report generation.
- Therefore, the project is technically feasible with basic to moderate programming knowledge.

3.2 Operational Feasibility

This checks whether the system will function effectively and meet user needs in real-world conditions.

- The application is easy to use and doesn’t require technical skills.
- Users can manage financial data with simple forms and reports.
- The app supports both desktop and mobile browsers.
- As a result, the system is operationally feasible and user-friendly.

3.3 Economic Feasibility

This evaluates whether the project is financially viable.

- Since the app uses open-source tools, there is no cost for software or licensing.
- Development and deployment can be done on a local machine or free hosting services.
- Maintenance cost is low as it is a lightweight application.
- Hence, it is economically feasible for individual or educational use.

3.4 Legal Feasibility

This ensures that the project does not violate any legal or compliance regulations.

- The application is developed for academic and personal use.
- All libraries used are open-source and free for use.
- No sensitive user data (like bank accounts or real-time payment info) is collected.
- Thus, it is legally feasible for deployment and use.

4. Risk Analysis

Although the project is feasible in most areas, some risks still exist:

- Loss of data due to local file corruption (no cloud backup).
- Security risks if login/session features are not well-handled.
- Limited scalability (meant for single-user only).
- No professional customer support in case of issues.

These risks are minimal and manageable with careful implementation and future enhancements.

1. Preliminary Investigation

Problem Identification

Managing personal finances is often neglected due to the complexity of existing tools or lack of habit. Most individuals do not track daily spending, leading to poor budgeting and overspending.

Proposed Solution

Develop a lightweight, user-friendly web application where users can log their expenses, set monthly budgets, and view insightful reports. The system will operate offline-first using Flask and MySQL and include visual data representation.

Feasibility Study

- **Technical:**
  The application is built using well-supported technologies like Python (Flask), MySQL, HTML/CSS/JS, making it easy to develop and maintain.
- **Operational:**
  Easy-to-use interface ensures that even non-technical users can track their expenses effectively. Minimal system requirements increase accessibility.
- **Economic:**
  Development costs are minimal, relying on open-source tools. No third-party licensing required. Ideal for individual use or community deployment.

2. System Analysis

Existing Systems

Existing platforms like Walnut, Money Manager, and Google Sheets either require constant connectivity or offer too many features that overwhelm basic users.

Requirements Gathering

- User login/signup
- Add/view/delete expenses
- Set monthly income & budget
- Category-wise analysis
- Excel report downloads
- Interactive charts

Constraints

- Single-user mode (initially)
- Works on modern browsers only
- Limited to local database in offline mode

3. System Design

Architecture

- Client-server architecture using Flask (backend)
- RESTful design pattern
- MVC-like folder structure

Flow

1. User logs in → Dashboard
2. Add Expense → Save to DB
3. Fetch Data → Display Table/Charts
4. Export → Generate Report

Database Tables

- **users:** user_id, username, email, password
- **expenses:** expense_id, user_id, amount, category, description, date
- **incomes:** income_id, user_id, amount, source, date
- **budgets:** budget_id, user_id, month, year, amount
- **reports:** report_id, user_id, type, file_path, generated_on

4. Coding

Backend

- Language: Python
- Framework: Flask
- DBMS: MySQL
- ORM: Raw SQL queries / optional use of SQLAlchemy

Frontend

- HTML5, CSS3, Tailwind CSS
- JavaScript (for interactivity and charts via Chart.js)
- Jinja2 Templating engine

5. Security

- **User Authentication:**
  Password hashing using Werkzeug, login/session management, CSRF protection.
- **Data Protection:**
  Input sanitization, parameterized queries to prevent SQL injection, `.env` for sensitive credentials.

6. Testing

Unit Testing

- Each function (e.g., add_expense, generate_report) tested independently using sample data.

Functional Testing

- End-to-end checks for login, budget setting, report generation, and filtering.

User Testing

- Feedback collected from a small group of real users to refine usability and error handling.

7. Implementation

Deployment

- Local deployment for testing
- Can be deployed on cloud services (Heroku, PythonAnywhere, etc.)
- Includes `.env` for secure credentials management

Future Scope

- Multi-user support
- Mobile-friendly version
- Cloud sync and notifications
- Integration with payment APIs

Project Scheduling (Day-wise)

Day	Task
Day 1	Requirements gathering and planning
Day 2	Database design (ERD, table creation)
Day 3	Backend setup (Flask app structure)
Day 4	User authentication system

Day	Task
Day 5	Expense and income modules
Day 6	Budget tracking and visualizations
Day 7	Report generation (Excel download)
Day 8	Frontend UI improvements (Tailwind CSS)
Day 9	Testing: unit, functional, user
Day 10	Final debugging, documentation, and deployment

1. Project Title:

Expense Tracker Web Application

2. Purpose:

The purpose of this application is to provide users with a simple and intuitive platform to track daily expenses, manage monthly budgets, and generate reports to understand their financial behavior. It helps users build better financial habits by giving insights into their income and spending patterns.

3. Scope:

This web-based system allows users to:

- Record expenses with categories and descriptions
- Set monthly income and budget
- View and download monthly reports in Excel format
- Visualize spending with charts
- Filter/search expenses
- Operate primarily in offline/local mode (with Flask & MySQL)

4. Functional Requirements:

- User registration and login functionality
- Add, edit, and delete expense records
- Set and update monthly income and budget
- View all expenses in a sortable/filterable table
- Generate and download reports
- Display spending insights via pie charts and summary cards
- Category and date-based filtering of expenses

5. Non-Functional Requirements:

- The application should load pages in under 2 seconds
- Must be compatible with modern browsers (Chrome, Firefox, Edge)
- Responsive UI across desktops, tablets, and smartphones
- Secure storage of user credentials and session data
- Easy-to-navigate and minimal UI design
- Reliable data handling with MySQL backend

6. Software and Tools Required:

- **Frontend:** HTML5, Tailwind CSS, JavaScript
- **Backend:** Python 3.7+, Flask framework
- **Database:** MySQL Server
- **Chart Library:** Chart.js
- **Dependencies:** Flask extensions (Flask-Login, Flask-MySQLdb)
- **Environment Management:** Virtualenv
- **IDE/Text Editor:** VS Code, PyCharm
- **Version Control:** Git & GitHub

7. System Architecture:

- **Client:** Browser (UI)
- **Web Server:** Flask app (RESTful backend)
- **Database Server:** MySQL for structured data
- **Storage:** Local file system for reports
- **Data Flow:**
  Browser ↔ Flask App ↔ MySQL DB
  Flask ↔ Chart.js / Excel / Reports

8. Assumptions:

- User has basic internet and computer knowledge
- MySQL is properly installed and configured
- System is used by a single user (for the first version)
- Reports are accessed locally or via browser download

9. Constraints:

- No real-time cloud sync in initial version
- Limited to predefined budget categories unless manually expanded
- Currently supports only one user session at a time
- Reports are downloaded manually in Excel format

10. Future Enhancements (Optional Ideas):

- Add multi-user support with roles (admin, regular user)

- Enable cloud sync (Google Drive, Dropbox)
- Add mobile app version
- SMS/email notifications for budget thresholds
- AI-based suggestions to optimize spending
- Dark mode and UI themes customization

Project Title: Expense Tracker Web Application

1. Overview:

The Expense Tracker is a web-based application that enables users to manage personal finances by tracking income and expenses. It helps users analyze their spending habits, generate monthly reports, and maintain budget discipline through a user-friendly interface and interactive data visualization.

2. User Roles:

- **User**:
  - Register and log in
  - Add, update, and delete expenses
  - Set monthly income and budget
  - Generate and download reports
  - View spending summaries and visual charts

3. Functional Requirements:

3.1 User Authentication:

- Secure user registration with validation
- Login system with session management
- Password protection and hashing
- Logout functionality

3.2 Expense Tracking:

- Add new expense records (amount, category, description, date)
- Edit or delete existing expenses
- View expenses in a sortable and searchable table
- Filter expenses by category or date range

3.3 Budget Management:

- Set and update monthly income and budget
- Visual indicators for budget usage (progress bar, warnings)
- Summary of savings (Income - Expenses)

3.4 Report Generation:

- Generate and download monthly reports (Excel format)
- Reports include category-wise spending, monthly totals, and savings
- Visual representations using pie charts and graphs

3.5 Database Interactions:

- Store all user data in MySQL database
- Fetch, insert, update, and delete transactions
- Maintain separate tables for users, expenses, and budget

4. Non-Functional Requirements:

- Should be responsive on different screen sizes
- Compatible with major browsers
- Load time should be <2 seconds for main dashboard
- Secure communication with proper session handling
- Data validation at both client and server sides

5. Technical Stack:

- **Frontend**: HTML5, Tailwind CSS, JavaScript
- **Backend**: Python (Flask)
- **Database**: MySQL
- **Visualization**: Chart.js
- **Environment**: Virtualenv
- **Tools**: Git, GitHub, VS Code

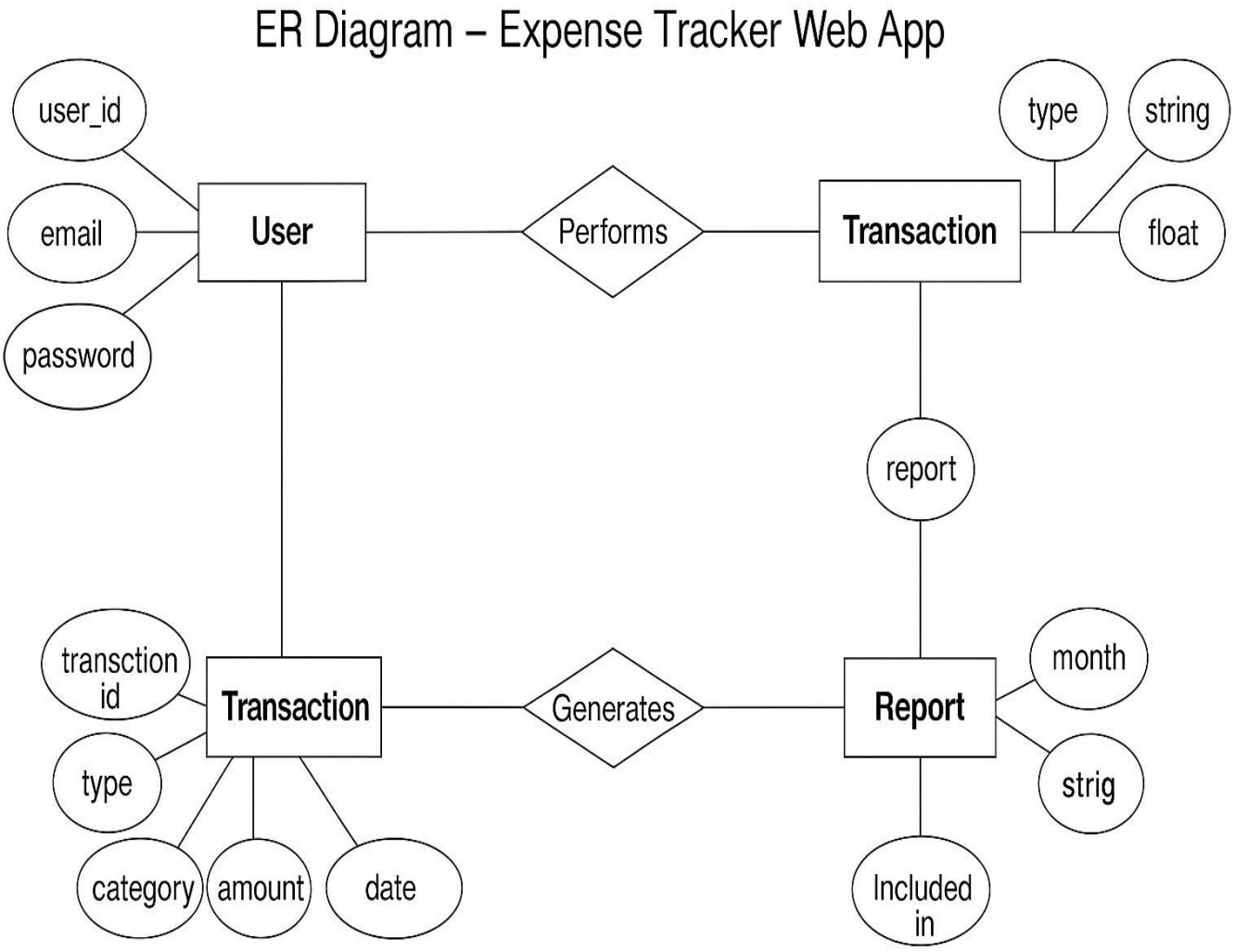
6. Error Handling:

- Input validation errors (empty fields, invalid values)
- Display friendly error messages for form inputs
- Handle server/database connection errors gracefully
- Redirect users on session timeout or unauthorized access

7. Assumptions:

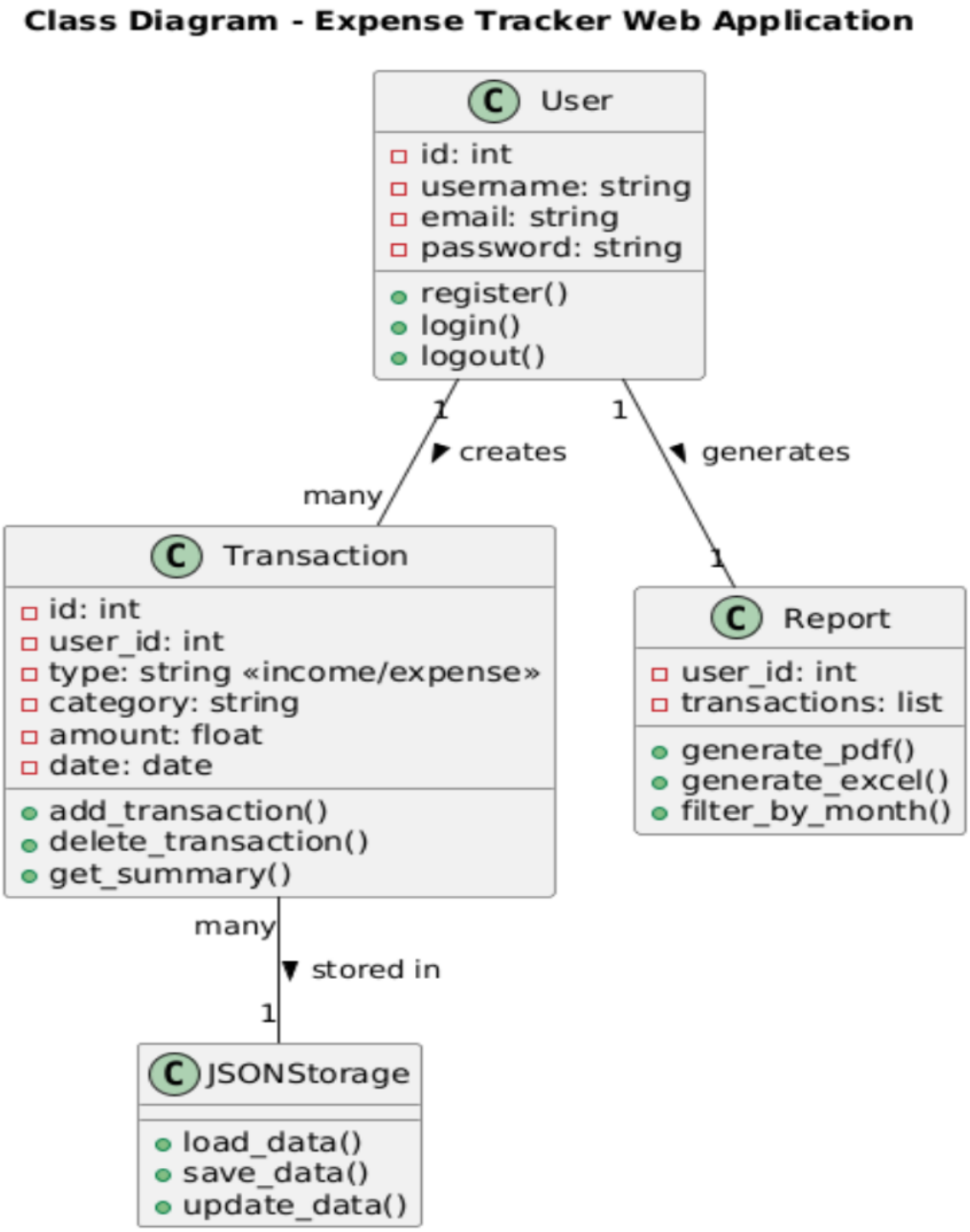
- MySQL database is set up and accessible
- Users have basic knowledge of web usage
- Application is run on a local or single-host environment
- Only one user will access the system in the initial version

ER DIAGRAM:

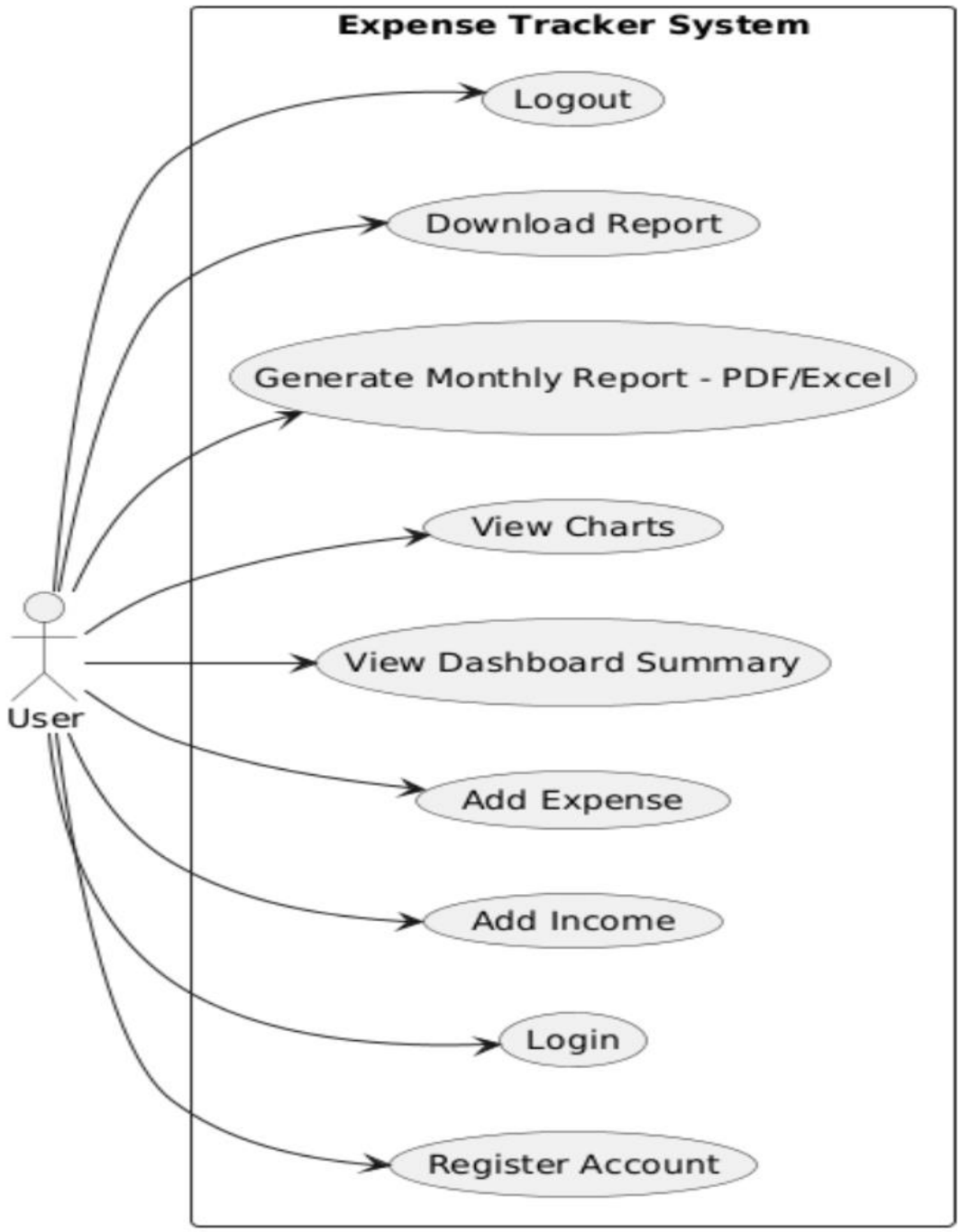


ER Diagram – Expense Tracker Web App

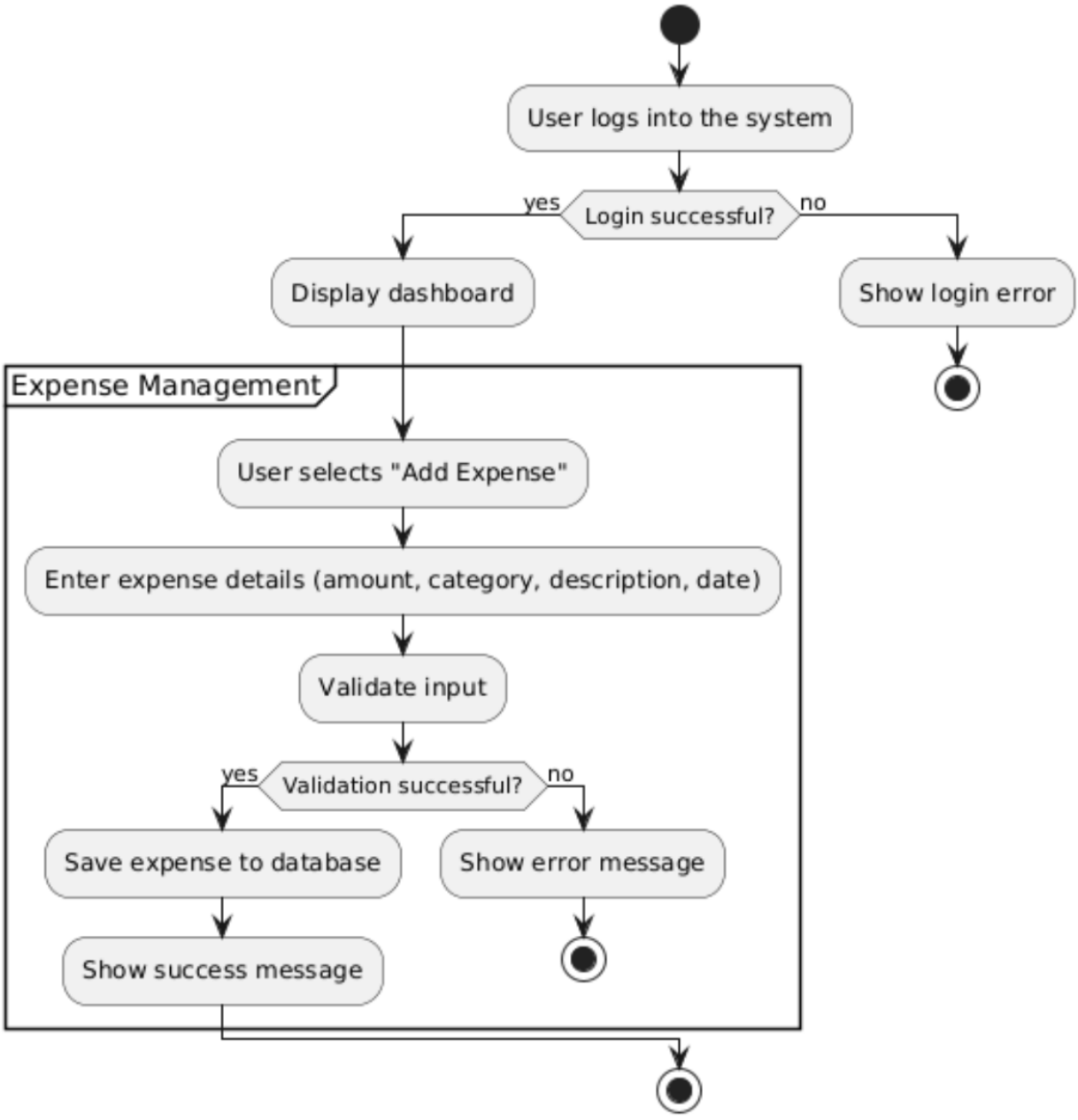
CLASS DIAGRAM:



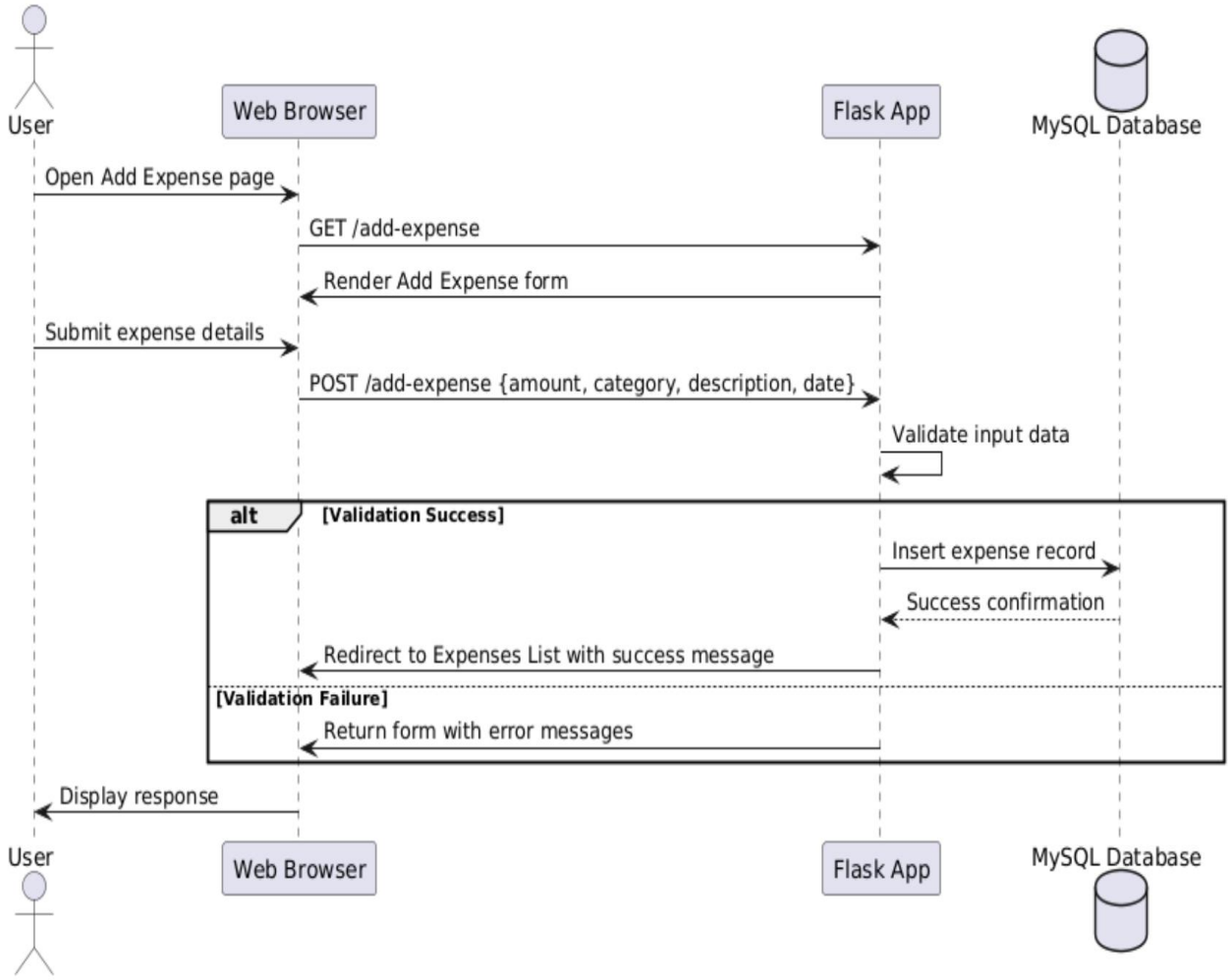
USE CASE DIAGRAM:



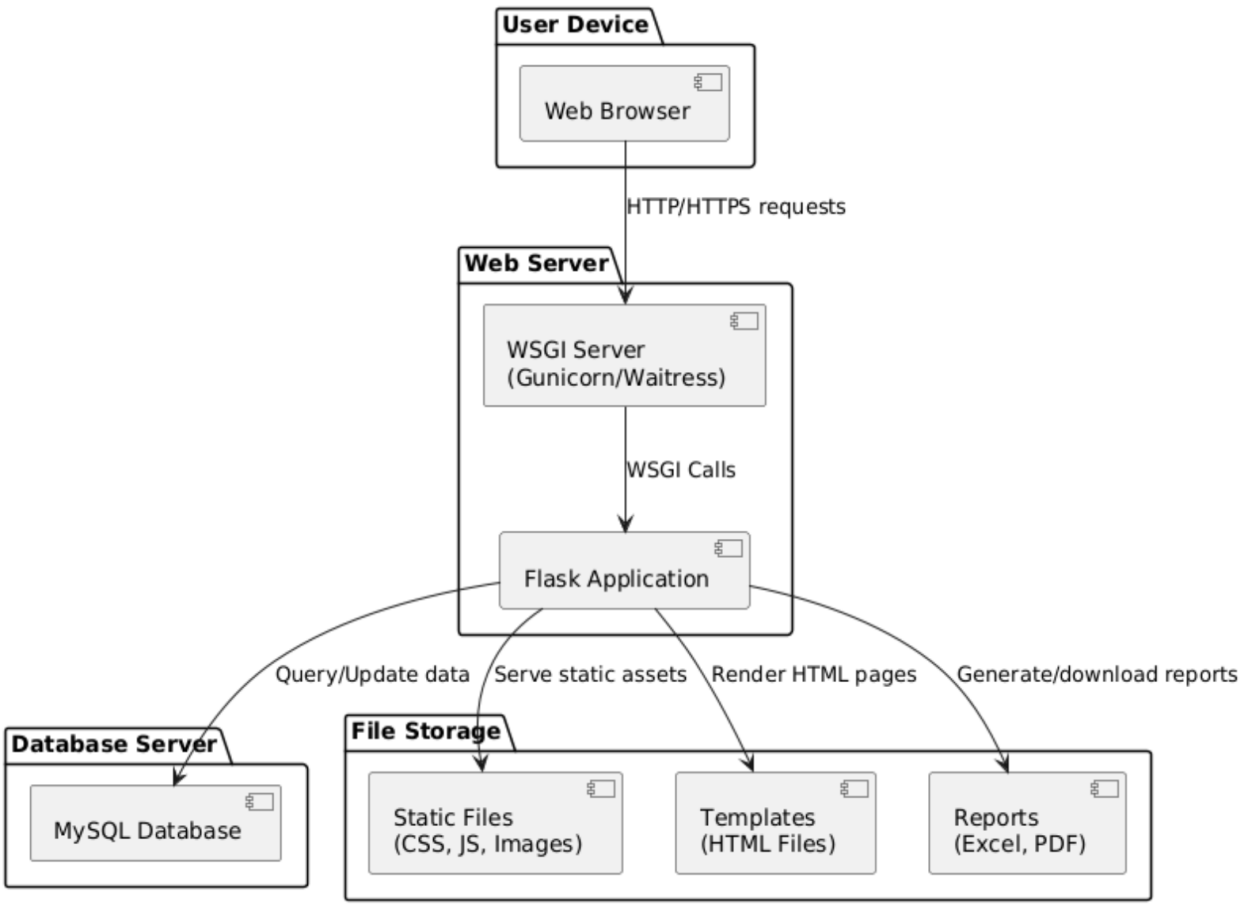
ACTIVITY DIAGRAM:



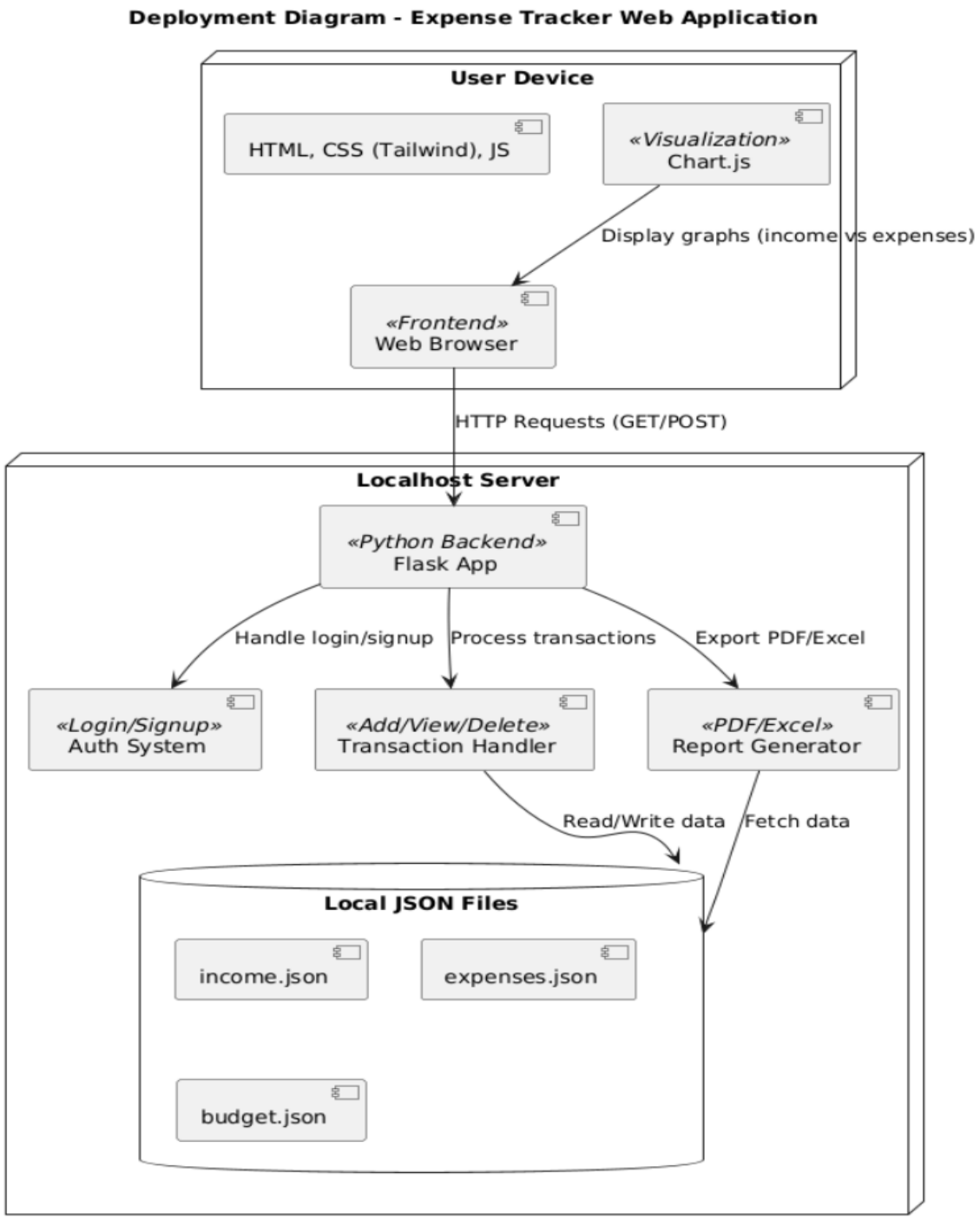
SEQUENCE DIAGRAM:



COMPONENT DIAGRAM:



DEPLOYMENT DIAGRAM:



DATA DICTIONARY:

1. Expenses Table

Field Name	Data Type	Description	Constraints	Example
id	INT	Unique identifier for each expense	Primary Key, Auto Increment	1, 2, 3...
amount	DECIMAL(10,2)	Amount of the expense	NOT NULL, > 0	1500.00
category	VARCHAR(50)	Category of the expense	NOT NULL	"Food", "Transport"
description	TEXT	Detailed description of the expense	NULL allowed	"Monthly groceries"
date	DATE	Date of the expense	NOT NULL	"2024-03-15"

1.1 Settings Table

Field Name	Data Type	Description	Constraints	Example
setting_key	VARCHAR(50)	Unique identifier for setting	Primary Key	"monthly_income"
value	TEXT	Value of the setting	NOT NULL	"50000"
user_id	VARCHAR(50)	User identifier	Primary Key	"user123"

2. Configuration Data

2.1 Application Settings

Setting Name	Type	Description	Default Value
MYSQL_HOST	String	Database host address	"localhost"
MYSQL_USER	String	Database username	"root"
MYSQL_PASSWORD	String	Database password	""

MYSQL_DB	String	Database name	"expense_tracker"
UPLOAD_FOLDER	String	Reports storage path	"./reports"
HOST	String	Application host	"0.0.0.0"
PORT	Integer	Application port	5000
DEBUG	Boolean	Debug mode flag	True
SECRET_KEY	String	Session security key	"your-secret-key-here"

3. Session Data

3.1 User Session

Field Name	Type	Description	Example
user	String	Username of logged-in user	"subham"
session_id	String	Unique session identifier	"123"

4. Report Data

4.1 Monthly Report Structure

Field Name	Type	Description	Example
monthly_income	Decimal	Total monthly income	50000.00
monthly_expense	Decimal	Total monthly expenses	35000.00
total_expenses	Decimal	Sum of all expenses	35000.00
net_savings	Decimal	Income minus expenses	15000.00
category_totals	Object	Expenses by category	{"Food": 10000, "Transport": 5000}

5. Expense Categories

5.1 Standard Categories

Category Name	Description	Example Expenses
Food	Food and groceries	Groceries, Restaurants
Transport	Transportation costs	Fuel, Public transport
Utilities	Basic utilities	Electricity, Water
Entertainment	Leisure activities	Movies, Games
Shopping	General shopping	Clothes, Electronics
Healthcare	Medical expenses	Medicines, Doctor visits
Education	Educational expenses	Books, Courses
Others	Miscellaneous expenses	Gifts, Donations

6. File Storage

6.1 Report Files

File Type	Format	Description	Naming Convention
Monthly Report	PDF	Monthly expense summary	monthly_report_YYYY_MM.pdf
Budget Report	PDF	Budget analysis report	budget_report_YYYY_MM.pdf

7. Data Validation Rules

7.1 Expense Validation

Field	Validation Rules
amount	- Must be positive number - Maximum 2 decimal places - Cannot be null
category	- Must be one of predefined categories - Cannot be null - Maximum 50 characters
description	- Optional - Maximum 1000 characters

date	- Must be valid date - Cannot be future date - Cannot be null

7.2 Budget Validation

Field	Validation Rules
monthly_income	- Must be positive number - Maximum 2 decimal places
monthly_expense	- Must be positive number - Maximum 2 decimal places

8. Data Relationship

8.1 Primary Relationship

Table	Related To	Relationship Type	Description
expenses	settings	Indirect	Expenses are tracked against budget settings
settings	user_id	Direct	Settings are associated with specific users

9. Data Security

9.1 Sensetive Data

Data Type	Security Measures
Passwords	- Hashed using Werkzeug security - Never stored in plain text
Session Data	- Encrypted using secret key - Time-limited validity
Database Credentials	- Stored in environment variables - Not hardcoded in application

1. User Authentication Module

Description:
Handles user registration, login, and logout functionalities. Ensures data security through password hashing and session management.

Features:

- Sign Up
- Sign In
- Session control
- Password validation and hashing

2. Expense Management Module

Description:
Allows users to add, update, delete, and view their daily expenses. It supports categorization and filtering of transactions.

Features:

- Add new expense
- Edit/delete existing expense
- Filter expenses by date/category
- Display in searchable and sortable table

3. Budget Management Module

Description:
Enables users to set monthly income and budget. It provides visual indicators for budget usage and notifies if expenses exceed limits.

Features:

- Set monthly income and budget
- Track expenses against budget
- Show savings summary (income – expenses)
- Budget usage progress bar

4. Report Generation Module

Description:
Generates downloadable monthly reports summarizing expenses, income, and savings. Reports are provided in Excel format.

Features:

- Monthly report generation
- Export to Excel
- Category-wise breakdown
- Expense summary

5. Data Visualization Module

Description:
Presents spending data through dynamic charts and visual elements to enhance user understanding.

Features:

- Pie charts for category-wise spending
- Summary cards (income, expense, savings)
- Real-time chart updates

6. Database Interaction Module

Description:
Manages communication between the Flask backend and MySQL database. Performs CRUD operations for users, expenses, and budgets.

Features:

- Insert/update/delete records
- Secure and optimized queries
- Separate tables for users, budgets, expenses

7. Admin/Settings Module (Optional/Future Scope)

Description:
A placeholder for future implementation where an admin can manage users, backup data, and perform analytics.

Features:

- User management
- Report backup

1. User Authentication Module

Table Name: `users`
Description: Stores user credentials and session information.

Field Name	Data Type	Description
id	INT (PK, AI)	Unique identifier
username	VARCHAR(100)	User's name
email	VARCHAR(255)	Unique email
password_hash	VARCHAR(255)	Encrypted password
created_at	DATETIME	Registration timestamp

2. Expense Management Module

Table Name: `expenses`
Description: Stores all expense records made by the user.

Field Name	Data Type	Description
id	INT (PK, AI)	Unique expense ID
user_id	INT (FK)	References `users(id)`
amount	DECIMAL(10,2)	Expense amount
category	VARCHAR(100)	Category (e.g., Food, Travel)
description	TEXT	Optional notes or details
date	DATE	Date of expense
created_at	DATETIME	Record timestamp

3. Budget Management Module

Table Name: `budgets`
Description: Stores user's income and monthly budget details.

Field Name	Data Type	Description
id	INT (PK, AI)	Unique ID
user_id	INT (FK)	References `users(id)`
month	VARCHAR(10)	Month (e.g., 'June')
year	INT	Year
income	DECIMAL(10,2)	Monthly income
budget_limit	DECIMAL(10,2)	Monthly budget limit
created_at	DATETIME	Record timestamp

4. Report Generation Module

Table Name: No separate table. Reports are generated dynamically using data from `expenses` and `budgets` and downloaded as Excel.

Description:

- Data from both `expenses` and `budgets` tables is queried and processed to generate reports using Python libraries (like `pandas`, `openpyxl`, etc.).

5. Data Visualization Module

Description:
No separate table; this module fetches processed data (category-wise sums, monthly totals) from:

- `expenses` (for category and total expenses)
- `budgets` (for limits and income)
- Passed to Chart.js in JSON format.

6. Database Interaction Module

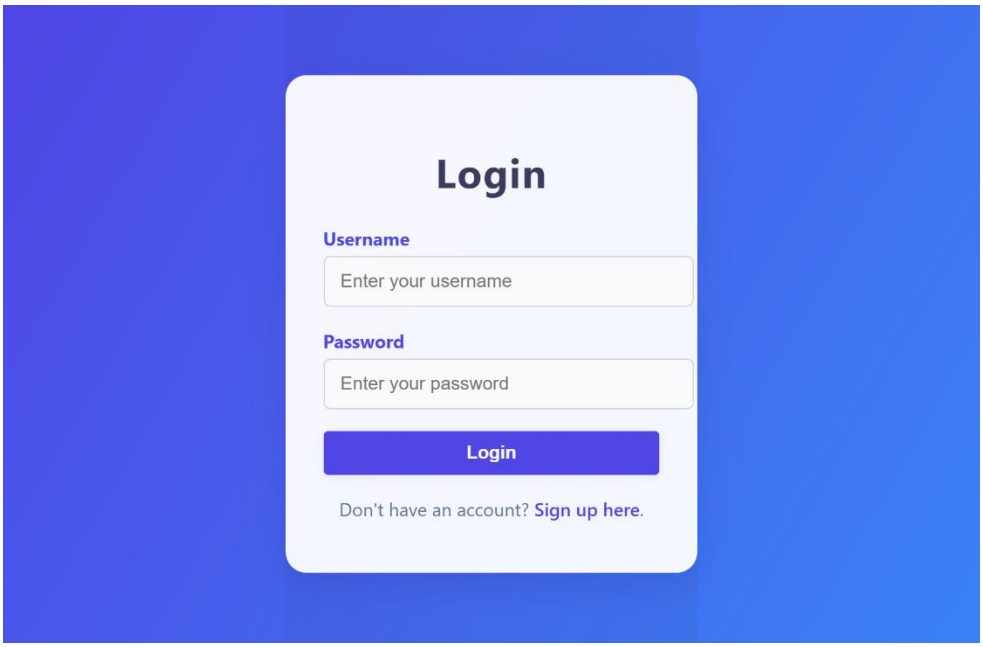
Functions/Methods (Python Flask):

```
python
CopyEdit
def get_expenses_by_user(user_id)
def add_expense(data)
def update_expense(expense_id, data)
def delete_expense(expense_id)
def get_budget_for_month(user_id, month, year)
def update_budget(data)
def get_monthly_summary(user_id, month, year)
```

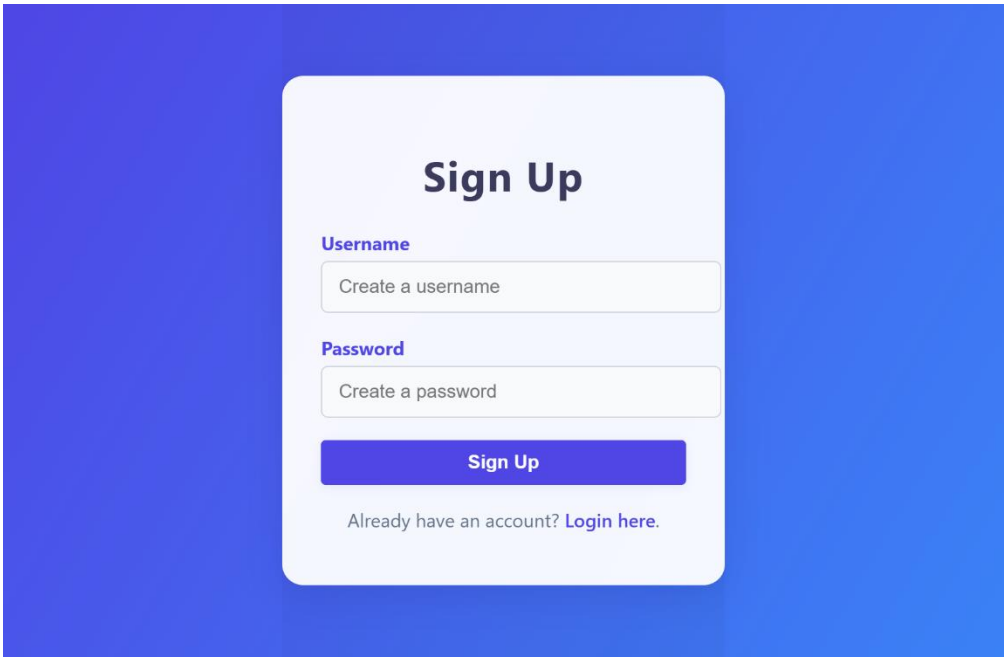
All of these functions interact with the above tables via SQL queries or SQLAlchemy ORM.

Screen shot:

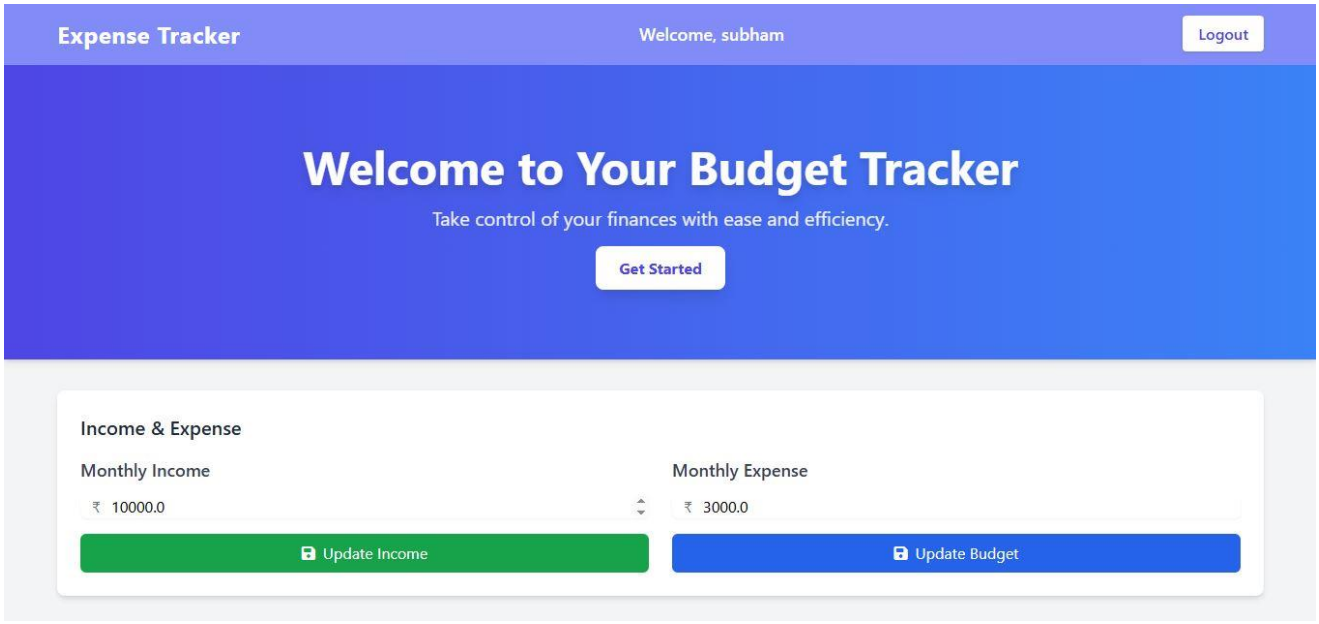
Login Screen:



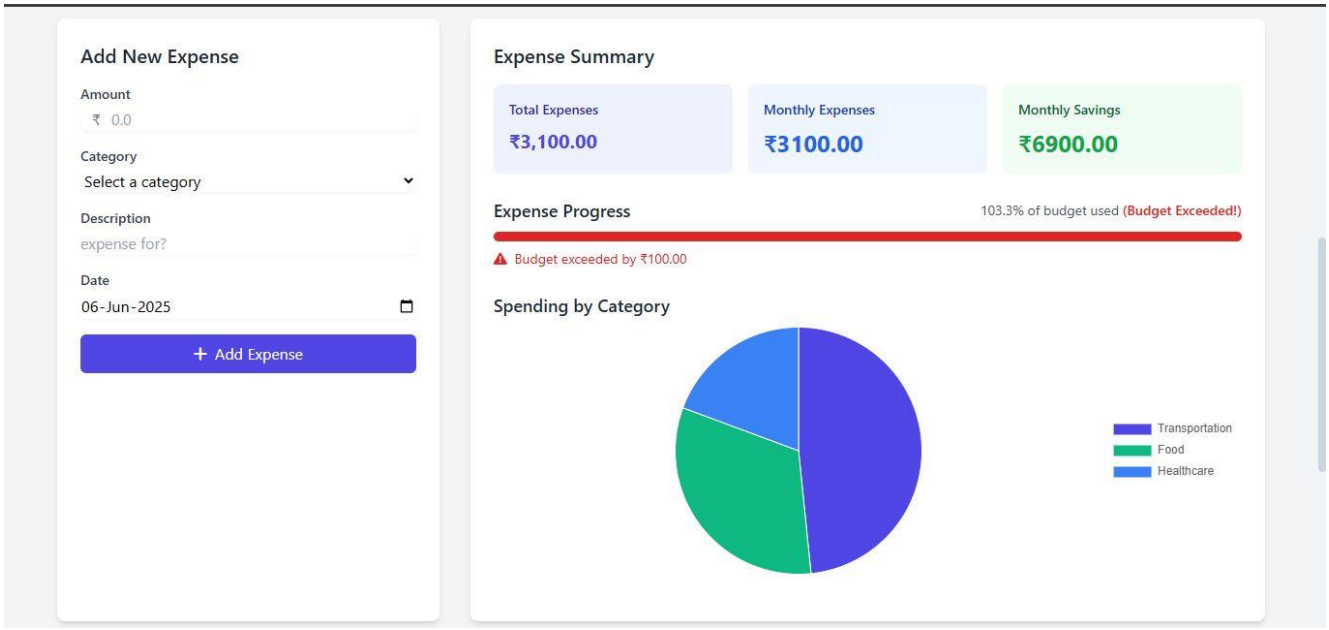
Signup Screen:



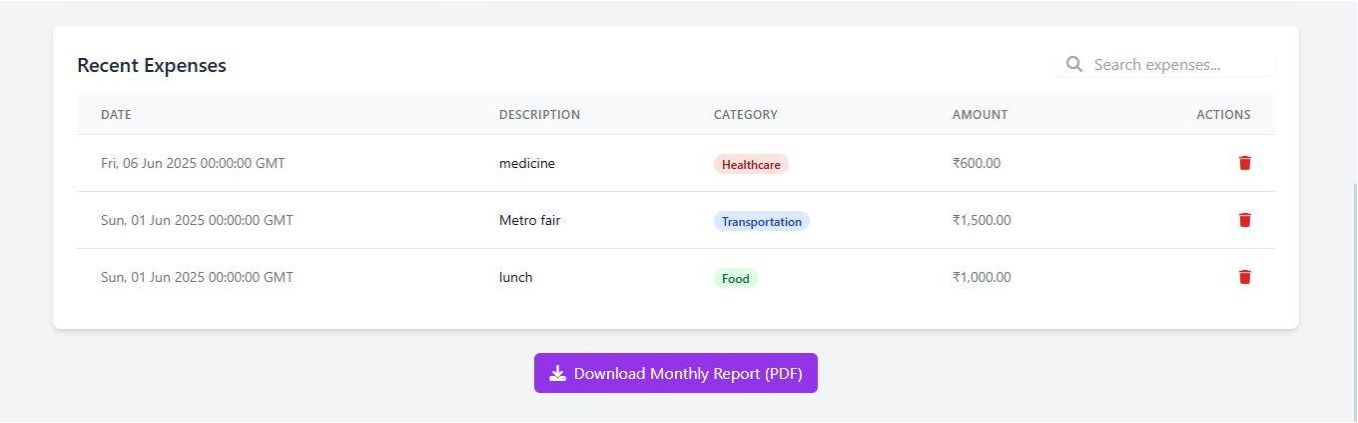
Home Screen:



Expense Summary DashBoard Screen:



Recent Expense Screen:



Monthly Report PDF Screen:

Monthly Expense Report

Generated on: June 06, 2025

Summary

Metric	Amount (■)
Monthly Income	■10,000.00
Monthly Budget	■3,000.00
Total Expenses	■3,100.00
Net Savings	■6,900.00

Detailed Expenses

Date	Category	Description	Amount (■)
2025-06-01	Transportation	Metro fair	■1,500.00
2025-06-01	Food	lunch	■1,000.00
2025-06-06	Healthcare	medicine	■600.00

CODING

Index.html

```
<!DOCTYPEhtml>
<htmllang="en">
<head>
<metacharset="UTF-8"/>
<metaname="viewport"content="width=device-width,initial-scale=1.0"/>
<title>ExpenseTracker</title>
<scriptsrc="https://cdn.tailwindcss.com"></script>
<link
rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css"
/>
<scriptsrc="https://cdn.jsdelivr.net/npm/chart.js"></script>
<link
rel="stylesheet"
href="{{url_for('static',filename='css/styles.css')}}"
/>
</head>
<bodyclass="bg-gray-100min-h-screen">
<!--NavigationBar-->
<navclass="bg-indigo-400text-whitepy-4shadow-lg">
<divclass="containermx-autoflexjustify-betweenitems-centerpx-4">
<!--Logo-->
<divclass="text-2xlfont-boldtracking-wide">ExpenseTracker</div>
<!--WelcomeUser-->
{%ifuser%}
<divclass="text-lgfont-medium">Welcome,{{user}}</div>
{%endif%}
<!--LogoutButton-->
<div>
<a
href="/logout"
class="bg-whitetext-indigo-600px-4py-2roundedshadowhover:bg-gray-200font-semiboldtransition"
>Logout</>
</div>
```

```
</div>
</nav>
<!--HeroSection-->
<sectionclass="bg-indigo-600text-whitepy-20shadow-md">
<divclass="containermx-autotext-centerpx-4">
<h1class="text-5xlfont-extraboldmb-4drop-shadow-lg">
WelcometoYourBudgetTracker
</h1>
<pclass="text-lgmb-6opacity-90">
Takecontrolofyourfinanceswitheaseandefficiency.
</p>
<a
href="#"
class="bg-whitetext-indigo-600px-8py-3rounded-lgshadow-lghover:bg-gray-200font-boldtext-
lgtransition"
>GetStarted</a
>
</div>
</section>

<divclass="containermx-autopx-4py-8">
<!--IncomeandBudgetTrackingCard-->
<divclass="bg-whiterounded-lgshadow-mdp-6mb-8">
<h2class="text-xlfont-semiboldtext-gray-800mb-4">
Income&Expense
</h2>
<divclass="gridgrid-cols-1md:grid-cols-2gap-6">
<!--IncomeSection-->
<div>
<h3class="text-lgfont-mediumtext-gray-700mb-3">
MonthlyIncome
</h3>
<form
id="update-income-form"
action="/update_income"
method="POST"
class="space-y-4"
```

```
>
<divclass="relative">
<span
class="absoluteleft-3top-1/2transform-translate-y-1/2text-gray-500"
>₹</span
>
<input
type="number"
id="monthly-income"
name="income"
step="0.01"
required
value="{{monthly_income}}"
class="pl-8w-fullrounded-mdborder-gray-300shadow-smfocus:border-indigo-500focus:ringfocus:ring-indigo-200focus:ring-opacity-50"
placeholder="0.0"
/>
</div>
<button
type="submit"
class="w-fullbg-green-600text-whitepy-2px-4rounded-mdhover:bg-green-700transitionduration-300flexitems-centerjustify-center"
>
<iclass="fasfa-savemr-2"></i>UpdateIncome
</button>
</form>
</div>

<!--BudgetSection-->
<div>
<h3class="text-lgfont-mediumtext-gray-700mb-3">
MonthlyExpense
</h3>
<form
id="update-budget-form"
action="/update_budget"
method="POST"
```

```
class="space-y-4"
>
<divclass="relative">
<span
class="absoluteleft-3top-1/2transform-translate-y-1/2text-gray-500"
>₹</span
>
<input
type="number"
id="monthly-budget"
name="budget"
step="0.01"
required
value="{{monthly_budget}}"
class="pl-8w-fullrounded-mdborder-gray-300shadow-smfocus:border-indigo-500focus:ringfocus:ring-indigo-200focus:ring-opacity-50"
placeholder="Setyourmonthlybudget"
/>
</div>
<button
type="submit"
class="w-fullbg-blue-600text-whitepy-2px-4rounded-mdhover:bg-blue-700transitionduration-300flexitems-centerjustify-center"
>
<iclass="fasfa-savemr-2"></i>UpdateBudget
</button>
</form>
</div>
</div>
</div>

<divclass="gridgrid-cols-1lg:grid-cols-3gap-8">
<!--AddExpenseCard-->
<divclass="bg-whiterounded-lgshadow-mdp-6col-span-1">
<h2class="text-xlfont-semiboldtext-gray-800mb-4">
AddNewExpense
</h2>
```

```
<formaction="/add_expense"method="POST"class="space-y-4">
<div>
<label
for="amount"
class="blocktext-smfont-mediumtext-gray-700mb-1"
>Amount</label
>
<divclass="relative">
<span
class="absoluteleft-3top-1/2transform-translate-y-1/2text-gray-500"
>₹</span
>
<input
type="number"
id="amount"
name="amount"
step="0.01"
required
class="pl-8w-fullrounded-mdborder-gray-300shadow-smfocus:border-indigo-500focus:ringfocus:ring-indigo-200focus:ring-opacity-50"
placeholder="0.0"
/>
</div>
</div>

<div>
<label
for="category"
class="blocktext-smfont-mediumtext-gray-700mb-1"
>Category</label
>
<select
id="category"
name="category"
required
class="w-fullrounded-mdborder-gray-300shadow-smfocus:border-indigo-500focus:ringfocus:ring-indigo-200focus:ring-opacity-50"
```

```
>
<optionvalue="">Selectacategory</option>
<optionvalue="Food">Food</option>
<optionvalue="Transportation">Transportation</option>
<optionvalue="Entertainment">Entertainment</option>
<optionvalue="Housing">Housing</option>
<optionvalue="Utilities">Utilities</option>
<optionvalue="Healthcare">Healthcare</option>
<optionvalue="Shopping">Shopping</option>
<optionvalue="Other">Other</option>
</select>
</div>

<div>
<label
for="description"
class="blocktext-smfont-mediumtext-gray-700mb-1"
>Description</label
>
<input
type="text"
id="description"
name="description"
required
class="w-fullrounded-mdborder-gray-300shadow-smfocus:border-indigo-500focus:ringfocus:ring-indigo-
200focus:ring-opacity-50"
placeholder="expensefor?"
/>
</div>

<div>
<label
for="date"
class="blocktext-smfont-mediumtext-gray-700mb-1"
>Date</label
>
<input
```

```
type="date"
id="date"
name="date"
class="w-fullrounded-mdborder-gray-300shadow-smfocus:border-indigo-500focus:ringfocus:ring-indigo-200focus:ring-opacity-50"
/>
</div>

<button
type="submit"
class="w-fullbg-indigo-600text-whitepy-2px-4rounded-mdhover:bg-indigo-700transitionduration-300flexitems-centerjustify-center"
>
<iclass="fasfa-plusmr-2"></i>AddExpense
</button>
</form>
</div>

<!--SummaryandChartsCard-->
<divclass="bg-whiterounded-lgshadow-mdp-6col-span-1lg:col-span-2">
<h2class="text-xlfont-semiboldtext-gray-800mb-4">
ExpenseSummary
</h2>

<divclass="gridgrid-cols-1md:grid-cols-3gap-4mb-6">
<divclass="bg-indigo-50rounded-lgp-4">
<h3class="text-smfont-mediumtext-indigo-800mb-2">
TotalExpenses
</h3>
<p
class="text-xlfont-boldtext-indigo-600break-words"
id="total-expenses"
>
₹0.00
</p>
</div>
<divclass="bg-blue-50rounded-lgp-4">
```

```
<h3class="text-smfont-mediumtext-blue-800mb-2">
MonthlyExpenses
</h3>
<pclass="text-2xlfont-boldtext-blue-600"id="monthly-expenses">
₹{{"%.2f"|format(monthly_expenses)}}
</p>
</div>
<divclass="bg-green-50rounded-lgp-4">
<h3class="text-smfont-mediumtext-green-800mb-2">
MonthlySavings
</h3>
<pclass="text-2xlfont-boldtext-green-600"id="monthly-savings">
₹{{"%.2f"|format(monthly_income-monthly_expenses)}}
</p>
</div>
</div>

<!--BudgetProgress-->
<divclass="mb-6">
<divclass="flexjustify-betweenitems-centermb-2">
<h3class="text-lgfont-mediumtext-gray-800">
ExpenseProgress
</h3>
<spanclass="text-smtext-gray-600"id="budget-percentage">
{%ifmonthly_budget>0%}{{
"%.1f"|format(budget_warning.percentage)}}%ofbudgetused{%
ifbudget_warning.is_exceeded%}
<spanclass="text-red-600font-semibold"
>(BudgetExceeded!)</span
>
{%endif%}{%else%}Nobudgetset{%endif%}
</span>
</div>

<divclass="w-fullbg-gray-200rounded-fullh-2.5">
{%ifmonthly_budget>0%}{%ifbudget_warning.percentage>100
%}
```

```
<div
class="bg-red-600h-2.5rounded-full"
style="width:100%"
></div>
{%else%}
<div
class="{%ifbudget_warning.percentage>90%}bg-yellow-600{%else%}bg-green-600{%endif%}h-
2.5rounded-full"
data-percentage="{{budget_warning.percentage}}"
></div>
{%endif%}{%else%}
<div
class="bg-gray-400h-2.5rounded-full"
style="width:0%"
></div>
{%endif%}
</div>

{%ifmonthly_budget>0%}
<div
class="mt-2text-sm{%ifbudget_warning.is_exceeded%}text-red-600{%else%}text-green-
600{%endif%}"
>
{%ifbudget_warning.is_exceeded%}
<iclass="fasfa-exclamation-trianglemr-1"></i>
Budgetexceededby₹{{"%.2f"|format(-budget_warning.remaining)}}
{%else%}
<iclass="fasfa-check-circlemr-1"></i>
₹{{"%.2f"|format(budget_warning.remaining)}}remaininginbudget
{%endif%}
</div>
{%endif%}
</div>

<divclass="mb-6">
<h3class="text-lgfont-mediumtext-gray-800mb-2">
SpendingbyCategory
```

```
</h3>
<divclass="h-64">
<canvasid="category-chart"></canvas>
</div>
</div>
</div>
</div>

<!--ExpenseList-->
<divclass="mt-8bg-whiterounded-lgshadow-mdp-6">
<divclass="flexjustify-betweenitems-centermb-4">
<h2class="text-xlfont-semiboldtext-gray-800">RecentExpenses</h2>
<divclass="relative">
<input
type="text"
id="search-expenses"
placeholder="Searchexpenses..."
class="pl-10rounded-mdborder-gray-300shadow-smfocus:border-indigo-500focus:ringfocus:ring-indigo-
200focus:ring-opacity-50"
/>
<i
class="fasfa-searchabsoluteleft-3top-1/2transform-translate-y-1/2text-gray-400"
></i>
</div>
</div>

<divclass="overflow-x-auto">
<tableclass="min-w-fulldivide-ydivide-gray-200">
<theadclass="bg-gray-50">
<tr>
<th
class="px-6py-3text-lefttext-xsfont-mediumtext-gray-500uppercasetracking-wider"
>
Date
</th>
<th
class="px-6py-3text-lefttext-xsfont-mediumtext-gray-500uppercasetracking-wider"
```

```
>
Description
</th>
<th
class="px-6py-3text-lefttext-xsfont-mediumtext-gray-500uppercasetracking-wider"
>
Category
</th>
<th
class="px-6py-3text-lefttext-xsfont-mediumtext-gray-500uppercasetracking-wider"
>
Amount
</th>
<th
class="px-6py-3text-righttext-xsfont-mediumtext-gray-500uppercasetracking-wider"
>
Actions
</th>
</tr>
</thead>
<tbody
class="bg-whitedivide-ydivide-gray-200"
id="expense-table-body"
>
{%ifexpenses%}{%forexpenseinexpenses%}
<tr>
<tdclass="px-6py-4whitespace-nowraptext-smtext-gray-500">
{{expense.date}}
</td>
<tdclass="px-6py-4whitespace-nowraptext-smtext-gray-900">
{{expense.description}}
</td>
<tdclass="px-6py-4whitespace-nowrap">
<span
```

```
class="px-2inline-flextext-xsleading-5font-semiboldrounded-full{%ifexpense.category=='Food'%}bg-green-100text-green-800{%elifexpense.category=='Transportation'%}bg-blue-100text-blue-800{%elifexpense.category=='Entertainment'%}bg-purple-100text-purple-800{%elifexpense.category=='Housing'%}bg-yellow-100text-yellow-800{%elifexpense.category=='Utilities'%}bg-gray-100text-gray-800{%elifexpense.category=='Healthcare'%}bg-red-100text-red-800{%elifexpense.category=='Shopping'%}bg-pink-100text-pink-800{%else%}bg-indigo-100text-indigo-800{%endif%}"
>
{{expense.category}}
</span>
</td>
<tdclass="px-6py-4whitespace-nowraptext-smtext-gray-500">
₹{{"%.2f"|format(expense.amount)}}
</td>
<td
class="px-6py-4whitespace-nowraptext-righttext-smfont-medium"
>
<form
action="/delete_expense/{{expense.id}}"
method="POST"
class="inline"
>
<button
type="submit"
class="text-red-600hover:text-red-900"
>
<iclass="fasfa-trash"></i>
</button>
</form>
</td>
</tr>
{%endfor%}{%else%}
<tr>
<td
colspan="5"
class="px-6py-4text-centertext-smtext-gray-500"
>
```

```
Noexpensesfound.Addyourfirstexpense!
</td>
</tr>
{%endif%}
</tbody>
</table>
</div>
</div>

<!--DownloadReportButton-->
<divclass="mt-6text-center">
<a
href="/download_report"
class="inline-flexitems-centerpx-4py-2bg-purple-600text-whiterounded-mdhover:bg-purple-
700transitionduration-300"
download
>
<iclass="fasfa-downloadmr-2"></i>DownloadMonthlyReport(PDF)
</a>
</div>
</div>

<scriptsrc="{{url_for('static',filename='js/main.js')}}"></script>
<script>
letcategoryChart=null;

//Updatethepiechartdata
functionupdatePieChart(){
fetch("/api/categories")
.then((response)=>response.json())
.then((data)=>{
constctx=document
.getElementById("category-chart")
.getContext("2d");

//Destroyexistingchartifitexists
if(categoryChart){
```

```
categoryChart.destroy();
}

//Onlycreatechartifthereisdata
if(data.length>0){
categoryChart=newChart(ctx,{
type:"pie",
data:{
labels:data.map((item)=>item.name),
datasets:[
{
data:data.map((item)=>parseFloat(item.value)),
backgroundColor:[
"#4F46E5",//indigo
"#10B981",//green
"#3B82F6",//blue
"#8B5CF6",//purple
"#F59E0B",//yellow
"#EF4444",//red
"#EC4899",//pink
"#6B7280",//gray
],
borderWidth:1,
},
],
},
options:{
responsive:true,
maintainAspectRatio:false,
plugins:{
legend:{
position:"right",
},
tooltip:{
callbacks:{
label:function(context){
constvalue=context.raw;
```

```
consttotal=context.dataset.data.reduce(
(a,b)=>a+b,
0
);
constpercentage=((value/total)*100).toFixed(1);
return`₹${value.toLocaleString(
"en-IN"
)}(${percentage}%)`;
},
},
},
},
},
});
}else{
//Clearthechartareaandshowamessage
ctx.clearRect(0,0,ctx.canvas.width,ctx.canvas.height);
ctx.fillStyle="#6B7280";
ctx.textAlign="center";
ctx.textBaseline="middle";
ctx.font="14pxArial";
ctx.fillText(
"Noexpensesforcurrentmonth",
ctx.canvas.width/2,
ctx.canvas.height/2
);
}
})
.catch((error)=>
console.error("Errorfetchingcategorydata:",error)
);
}

//Updatetotalexpensesandmonthlydata
functionupdateExpenses(){
//Fetchtotalexpenses
fetch("/api/expenses")
```

```
.then((response)=>response.json())
.then((data)=>{
consttotal=data.reduce(
(sum,expense)=>sum+parseFloat(expense.amount),
0
);
document.getElementById(
"total-expenses"
).textContent=`₹${total.toLocaleString("en-IN",{
minimumFractionDigits:2,
maximumFractionDigits:2,
})}`;

//Updatetheexpensetable
consttableBody=document.getElementById("expense-table-body");
if(data.length===0){
tableBody.innerHTML=`
<tr>
<tdcolspan="5"class="px-6py-4text-centertext-smtext-gray-500">
Noexpensesfound.Addyourfirstexpense!
</td>
</tr>
`;
return;
}

tableBody.innerHTML=data
.map(
(expense)=>`
<tr>
<tdclass="px-6py-4whitespace-nowraptext-smtext-gray-500">
${expense.date}
</td>
<tdclass="px-6py-4whitespace-nowraptext-smtext-gray-900">
${expense.description}
</td>
<tdclass="px-6py-4whitespace-nowrap">
```

```
<spanclass="px-2inline-flextext-xsleading-5font-semiboldrounded-full${
expense.category==="Food"
?"bg-green-100text-green-800"
:expense.category==="Transportation"
?"bg-blue-100text-blue-800"
:expense.category==="Entertainment"
?"bg-purple-100text-purple-800"
:expense.category==="Housing"
?"bg-yellow-100text-yellow-800"
:expense.category==="Utilities"
?"bg-gray-100text-gray-800"
:expense.category==="Healthcare"
?"bg-red-100text-red-800"
:expense.category==="Shopping"
?"bg-pink-100text-pink-800"
:"bg-indigo-100text-indigo-800"
}">
${expense.category}
</span>
</td>
<tdclass="px-6py-4whitespace-nowraptext-smtext-gray-500">
₹${parseFloat(expense.amount).toLocaleString("en-IN",{
minimumFractionDigits:2,
maximumFractionDigits:2,
})}
</td>
<tdclass="px-6py-4whitespace-nowraptext-righttext-smfont-medium">
<formaction="/delete_expense/${
expense.id
}"method="POST"class="inline">
<buttontype="submit"class="text-red-600hover:text-red-900">
<iclass="fasfa-trash"></i>
</button>
</form>
</td>
</tr>
`
```

```
)
.join("");
})
.catch((error)=>console.error("Errorfetchingexpenses:",error));

//Fetchmonthlysummary
fetch("/api/monthly_summary")
.then((response)=>response.json())
.then((data)=>{
//Updatemonthlyexpenses
document.getElementById(
"monthly-expenses"
).textContent=`₹${data.monthly_expenses.toLocaleString("en-IN",{
minimumFractionDigits:2,
maximumFractionDigits:2,
})}`;

//Updatemonthlysavings(onlyiftherareexpenses)
constsavingsText=
data.monthly_expenses>0
?`₹${data.savings.toLocaleString("en-IN",{
minimumFractionDigits:2,
maximumFractionDigits:2,
})}`
:"₹0.00";
document.getElementById("monthly-savings").textContent=
savingsText;
})
.catch((error)=>
console.error("Errorfetchingmonthlysummary:",error)
);
}

//Searchfunctionality
document
.getElementById("search-expenses")
.addEventListener("input",function(e){
```

```
constsearchTerm=e.target.value.toLowerCase();
constrows=document.querySelectorAll("#expense-table-bodytr");

rows.forEach((row)=>{
consttext=row.textContent.toLowerCase();
row.style.display=text.includes(searchTerm)?"":"none";
});
});

//Callupdatefunctionswhenthepageloads
document.addEventListener("DOMContentLoaded",function(){
//Handleincomeformsubmission
document
.getElementById("update-income-form")
.addEventListener("submit",asyncfunction(e){
e.preventDefault();
constform=e.target;
constincome=form.querySelector('input[name="income"]').value;

try{
constresponse=awaitfetch("/update_income",{
method:"POST",
headers:{
"Content-Type":"application/x-www-form-urlencoded",
},
body:`income=${encodeURIComponent(income)}`,
});

constdata=awaitresponse.json();

if(data.success){
alert("Incomeupdatedsuccessfully!");
//Refreshthepagetoshowupdatedvalues
window.location.reload();
}else{
alert(`Failedtoupdateincome:${data.error}`);
}
```

```
}catch(error){
console.error("Error:",error);
alert("Failedtoupdateincome.Pleasetryagain.");
}
});

//Handlebudgetformsubmission
document
.getElementById("update-budget-form")
.addEventListener("submit",asyncfunction(e){
e.preventDefault();
constform=e.target;
constbudget=form.querySelector('input[name="budget"]').value;

try{
constresponse=awaitfetch("/update_budget",{
method:"POST",
headers:{
"Content-Type":"application/x-www-form-urlencoded",
},
body:`budget=${encodeURIComponent(budget)}`,
});

constdata=awaitresponse.json();

if(data.success){
alert("Budgetupdatedsuccessfully!");
//Refreshthepagetoshowupdatedvalues
window.location.reload();
}else{
alert(`Failedtoupdatebudget:${data.error}`);
}
}catch(error){
console.error("Error:",error);
alert("Failedtoupdatebudget.Pleasetryagain.");
}
});
```

```
//Initializethedateinputwithtoday'sdate
consttoday=newDate().toISOString().split("T")[0];
document.getElementById("date").value=today;

//Updateexpensestableandcharts
updateExpenses();
updatePieChart();
});

//Addeventlistenerfordownloadbutton
document
.querySelector('a[href="/download_report"]')
.addEventListener("click",asyncfunction(e){
e.preventDefault();
try{
constresponse=awaitfetch("/download_report");
if(!response.ok){
consterrorData=awaitresponse.json();
thrownewError(errorData.error||"Failedtodownloadreport");
}

//GetthefilenamefromtheContent-Dispositionheader
constcontentDisposition=response.headers.get(
"Content-Disposition"
);
constfilename=contentDisposition
?contentDisposition.split("filename=")[1].replace(/"/g,"")
:"monthly_report.pdf";

//Createablobfromtheresponse
constblob=awaitresponse.blob();

//Createadownloadlink
consturl=window.URL.createObjectURL(blob);
consta=document.createElement("a");
a.href=url;
```

```
a.download=filename;
document.body.appendChild(a);
a.click();

//Cleanup
window.URL.revokeObjectURL(url);
document.body.removeChild(a);
}catch(error){
console.error("Errordownloadingreport:",error);
alert("Failedtodownloadreport:"+error.message);
}
});
</script>
</body>
</html>

<!DOCTYPEhtml>
<htmllang="en">
<head>
<metacharset="UTF-8"/>
<metaname="viewport"content="width=device-width,initial-scale=1.0"/>
<title>ExpenseTracker</title>
<scriptsrc="https://cdn.tailwindcss.com"></script>
<link
rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css"
/>
<scriptsrc="https://cdn.jsdelivr.net/npm/chart.js"></script>
<link
rel="stylesheet"
href="{{url_for('static',filename='css/styles.css')}}"
/>
</head>
<bodyclass="bg-gray-100min-h-screen">
<!--NavigationBar-->
<navclass="bg-indigo-400text-whitepy-4shadow-lg">
<divclass="containermx-autoflexjustify-betweenitems-centerpx-4">
```

```
<!--Logo-->
<divclass="text-2xlfont-boldtracking-wide">ExpenseTracker</div>
<!--WelcomeUser-->
{%ifuser%}
<divclass="text-lgfont-medium">Welcome,{{user}}</div>
{%endif%}
<!--LogoutButton-->
<div>
<a
href="/logout"
class="bg-whitetext-indigo-600px-4py-2roundedshadowhover:bg-gray-200font-semiboldtransition"
>Logout</a
>
</div>
</div>
</nav>

<!--HeroSection-->
<sectionclass="bg-indigo-600text-whitepy-20shadow-md">
<divclass="containermx-autotext-centerpx-4">
<h1class="text-5xlfont-extraboldmb-4drop-shadow-lg">
WelcometoYourBudgetTracker
</h1>
<pclass="text-lgmb-6opacity-90">
Takecontrolofyourfinanceswitheaseandefficiency.
</p>
<a
href="#"
class="bg-whitetext-indigo-600px-8py-3rounded-lgshadow-lghover:bg-gray-200font-boldtext-
lgtransition"
>GetStarted</a
>
</div>
</section>

<divclass="containermx-autopx-4py-8">
<!--IncomeandBudgetTrackingCard-->
```

```
<divclass="bg-whiterounded-lgshadow-mdp-6mb-8">
<h2class="text-xlfont-semiboldtext-gray-800mb-4">
Income&Expense
</h2>
<divclass="gridgrid-cols-1md:grid-cols-2gap-6">
<!--IncomeSection-->
<div>
<h3class="text-lgfont-mediumtext-gray-700mb-3">
MonthlyIncome
</h3>
<form
id="update-income-form"
action="/update_income"
method="POST"
class="space-y-4"
>
<divclass="relative">
<span
class="absoluteleft-3top-1/2transform-translate-y-1/2text-gray-500"
>₹</span
>
<input
type="number"
id="monthly-income"
name="income"
step="0.01"
required
value="{{monthly_income}}"
class="pl-8w-fullrounded-mdborder-gray-300shadow-smfocus:border-indigo-500focus:ringfocus:ring-
indigo-200focus:ring-opacity-50"
placeholder="0.0"
/>
</div>
<button
type="submit"
class="w-fullbg-green-600text-whitepy-2px-4rounded-mdhover:bg-green-700transitionduration-
300flexitems-centerjustify-center"
```

```
>
<iclass="fasfa-savemr-2"></i>UpdateIncome
</button>
</form>
</div>

<!--BudgetSection-->
<div>
<h3class="text-lgfont-mediumtext-gray-700mb-3">
MonthlyExpense
</h3>
<form
id="update-budget-form"
action="/update_budget"
method="POST"
class="space-y-4"
>
<divclass="relative">
<span
class="absoluteleft-3top-1/2transform-translate-y-1/2text-gray-500"
>₹</span
>
<input
type="number"
id="monthly-budget"
name="budget"
step="0.01"
required
value="{{monthly_budget}}"
class="pl-8w-fullrounded-mdborder-gray-300shadow-smfocus:border-indigo-500focus:ringfocus:ring-indigo-200focus:ring-opacity-50"
placeholder="Setyourmonthlybudget"
/>
</div>
<button
type="submit"
```

```
class="w-fullbg-blue-600text-whitepy-2px-4rounded-mdhover:bg-blue-700transitionduration-
300flexitems-centerjustify-center"
>
<iclass="fasfa-savemr-2"></i>UpdateBudget
</button>
</form>
</div>
</div>
</div>

<divclass="gridgrid-cols-1lg:grid-cols-3gap-8">
<!--AddExpenseCard-->
<divclass="bg-whiterounded-lgshadow-mdp-6col-span-1">
<h2class="text-xlfont-semiboldtext-gray-800mb-4">
AddNewExpense
</h2>
<formaction="/add_expense"method="POST"class="space-y-4">
<div>
<label
for="amount"
class="blocktext-smfont-mediumtext-gray-700mb-1"
>Amount</label
>
<divclass="relative">
<span
class="absoluteleft-3top-1/2transform-translate-y-1/2text-gray-500"
>₹</span
>
<input
type="number"
id="amount"
name="amount"
step="0.01"
required
class="pl-8w-fullrounded-mdborder-gray-300shadow-smfocus:border-indigo-500focus:ringfocus:ring-
indigo-200focus:ring-opacity-50"
placeholder="0.0"
```

```
/>
</div>
</div>

<div>
<label
for="category"
class="blocktext-smfont-mediumtext-gray-700mb-1"
>Category</label
>
<select
id="category"
name="category"
required
class="w-fullrounded-mdborder-gray-300shadow-smfocus:border-indigo-500focus:ringfocus:ring-indigo-200focus:ring-opacity-50"
>
<optionvalue="">Selectacategory</option>
<optionvalue="Food">Food</option>
<optionvalue="Transportation">Transportation</option>
<optionvalue="Entertainment">Entertainment</option>
<optionvalue="Housing">Housing</option>
<optionvalue="Utilities">Utilities</option>
<optionvalue="Healthcare">Healthcare</option>
<optionvalue="Shopping">Shopping</option>
<optionvalue="Other">Other</option>
</select>
</div>

<div>
<label
for="description"
class="blocktext-smfont-mediumtext-gray-700mb-1"
>Description</label
>
<input
type="text"
```

```
id="description"
name="description"
required
class="w-fullrounded-mdborder-gray-300shadow-smfocus:border-indigo-500focus:ringfocus:ring-indigo-200focus:ring-opacity-50"
placeholder="expensefor?"
/>
</div>

<div>
<label
for="date"
class="blocktext-smfont-mediumtext-gray-700mb-1"
>Date</label
>
<input
type="date"
id="date"
name="date"
class="w-fullrounded-mdborder-gray-300shadow-smfocus:border-indigo-500focus:ringfocus:ring-indigo-200focus:ring-opacity-50"
/>
</div>

<button
type="submit"
class="w-fullbg-indigo-600text-whitepy-2px-4rounded-mdhover:bg-indigo-700transitionduration-300flexitems-centerjustify-center"
>
<iclass="fasfa-plusmr-2"></i>AddExpense
</button>
</form>
</div>

<!--SummaryandChartsCard-->
<divclass="bg-whiterounded-lgshadow-mdp-6col-span-1lg:col-span-2">
<h2class="text-xlfont-semiboldtext-gray-800mb-4">
```

```
ExpenseSummary
</h2>

<divclass="gridgrid-cols-1md:grid-cols-3gap-4mb-6">
<divclass="bg-indigo-50rounded-lgp-4">
<h3class="text-smfont-mediumtext-indigo-800mb-2">
TotalExpenses
</h3>
<p
class="text-xlfont-boldtext-indigo-600break-words"
id="total-expenses"
>
₹0.00
</p>
</div>
<divclass="bg-blue-50rounded-lgp-4">
<h3class="text-smfont-mediumtext-blue-800mb-2">
MonthlyExpenses
</h3>
<pclass="text-2xlfont-boldtext-blue-600"id="monthly-expenses">
₹{{"%.2f"|format(monthly_expenses)}}
</p>
</div>
<divclass="bg-green-50rounded-lgp-4">
<h3class="text-smfont-mediumtext-green-800mb-2">
MonthlySavings
</h3>
<pclass="text-2xlfont-boldtext-green-600"id="monthly-savings">
₹{{"%.2f"|format(monthly_income-monthly_expenses)}}
</p>
</div>
</div>

<!--BudgetProgress-->
<divclass="mb-6">
<divclass="flexjustify-betweenitems-centermb-2">
<h3class="text-lgfont-mediumtext-gray-800">
```

```
ExpenseProgress
</h3>
<spanclass="text-smtext-gray-600"id="budget-percentage">
{%ifmonthly_budget>0%}{{
"%.1f"|format(budget_warning.percentage)}}%ofbudgetused{%
ifbudget_warning.is_exceeded%}
<spanclass="text-red-600font-semibold"
>(BudgetExceeded!)</span
>
{%endif%}{%else%}Nobudgetset{%endif%}
</span>
</div>

<divclass="w-fullbg-gray-200rounded-fullh-2.5">
{%ifmonthly_budget>0%}{%ifbudget_warning.percentage>100
%}
<div
class="bg-red-600h-2.5rounded-full"
style="width:100%"
></div>
{%else%}
<div
class="{%ifbudget_warning.percentage>90%}bg-yellow-600{%else%}bg-green-600{%endif%}h-
2.5rounded-full"
data-percentage="{{budget_warning.percentage}}"
></div>
{%endif%}{%else%}
<div
class="bg-gray-400h-2.5rounded-full"
style="width:0%"
></div>
{%endif%}
</div>

{%ifmonthly_budget>0%}
<div
```

```
class="mt-2text-sm{%ifbudget_warning.is_exceeded%}text-red-600{%else%}text-green-
600{%endif%}"
>
{%ifbudget_warning.is_exceeded%}
<iclass="fasfa-exclamation-trianglemr-1"></i>
Budgetexceededby₹{{"%.2f"|format(-budget_warning.remaining)}}
{%else%}
<iclass="fasfa-check-circlemr-1"></i>
₹{{"%.2f"|format(budget_warning.remaining)}}remaininginbudget
{%endif%}
</div>
{%endif%}
</div>

<divclass="mb-6">
<h3class="text-lgfont-mediumtext-gray-800mb-2">
SpendingbyCategory
</h3>
<divclass="h-64">
<canvasid="category-chart"></canvas>
</div>
</div>
</div>
</div>

<!--ExpenseList-->
<divclass="mt-8bg-whiterounded-lgshadow-mdp-6">
<divclass="flexjustify-betweenitems-centermb-4">
<h2class="text-xlfont-semiboldtext-gray-800">RecentExpenses</h2>
<divclass="relative">
<input
type="text"
id="search-expenses"
placeholder="Searchexpenses..."
class="pl-10rounded-mdborder-gray-300shadow-smfocus:border-indigo-500focus:ringfocus:ring-indigo-
200focus:ring-opacity-50"
/>
```

```
<i
class="fasfa-searchabsoluteleft-3top-1/2transform-translate-y-1/2text-gray-400"
></i>
</div>
</div>

<divclass="overflow-x-auto">
<tableclass="min-w-fulldivide-ydivide-gray-200">
<theadclass="bg-gray-50">
<tr>
<th
class="px-6py-3text-lefttext-xsfont-mediumtext-gray-500uppercasetracking-wider"
>
Date
</th>
<th
class="px-6py-3text-lefttext-xsfont-mediumtext-gray-500uppercasetracking-wider"
>
Description
</th>
<th
class="px-6py-3text-lefttext-xsfont-mediumtext-gray-500uppercasetracking-wider"
>
Category
</th>
<th
class="px-6py-3text-lefttext-xsfont-mediumtext-gray-500uppercasetracking-wider"
>
Amount
</th>
<th
class="px-6py-3text-righttext-xsfont-mediumtext-gray-500uppercasetracking-wider"
>
Actions
</th>
</tr>
</thead>
```

```
<tbody
class="bg-whitedivide-ydivide-gray-200"
id="expense-table-body"
>
{%ifexpenses%}{%forexpenseinexpenses%}
<tr>
<tdclass="px-6py-4whitespace-nowraptext-smtext-gray-500">
{{expense.date}}
</td>
<tdclass="px-6py-4whitespace-nowraptext-smtext-gray-900">
{{expense.description}}
</td>
<tdclass="px-6py-4whitespace-nowrap">
<span
class="px-2inline-flextext-xsleading-5font-semiboldrounded-full{%ifexpense.category=='Food'%}bg-green-100text-green-800{%elifexpense.category=='Transportation'%}bg-blue-100text-blue-800{%elifexpense.category=='Entertainment'%}bg-purple-100text-purple-800{%elifexpense.category=='Housing'%}bg-yellow-100text-yellow-800{%elifexpense.category=='Utilities'%}bg-gray-100text-gray-800{%elifexpense.category=='Healthcare'%}bg-red-100text-red-800{%elifexpense.category=='Shopping'%}bg-pink-100text-pink-800{%else%}bg-indigo-100text-indigo-800{%endif%}"
>
{{expense.category}}
</span>
</td>
<tdclass="px-6py-4whitespace-nowraptext-smtext-gray-500">
₹{{"%.2f"|format(expense.amount)}}
</td>
<td
class="px-6py-4whitespace-nowraptext-righttext-smfont-medium"
>
<form
action="/delete_expense/{{expense.id}}"
method="POST"
class="inline"
>
<button
```

```
type="submit"
class="text-red-600hover:text-red-900"
>
<iclass="fasfa-trash"></i>
</button>
</form>
</td>
</tr>
{%endfor%}{%else%}
<tr>
<td
colspan="5"
class="px-6py-4text-centertext-smtext-gray-500"
>
Noexpensesfound.Addyourfirstexpense!
</td>
</tr>
{%endif%}
</tbody>
</table>
</div>
</div>

<!--DownloadReportButton-->
<divclass="mt-6text-center">
<a
href="/download_report"
class="inline-flexitems-centerpx-4py-2bg-purple-600text-whiterounded-mdhover:bg-purple-
700transitionduration-300"
download
>
<iclass="fasfa-downloadmr-2"></i>DownloadMonthlyReport(PDF)
</a>
</div>
</div>

<scriptsrc="{{url_for('static',filename='js/main.js')}}"></script>
```

```
<script>
letcategoryChart=null;

//Updatethepiechartdata
functionupdatePieChart(){
fetch("/api/categories")
.then((response)=>response.json())
.then((data)=>{
constctx=document
.getElementById("category-chart")
.getContext("2d");

//Destroyexistingchartifitexists
if(categoryChart){
categoryChart.destroy();
}

//Onlycreatechartifthereisdata
if(data.length>0){
categoryChart=newChart(ctx,{
type:"pie",
data:{
labels:data.map((item)=>item.name),
datasets:[
{
data:data.map((item)=>parseFloat(item.value)),
backgroundColor:[
"#4F46E5",//indigo
"#10B981",//green
"#3B82F6",//blue
"#8B5CF6",//purple
"#F59E0B",//yellow
"#EF4444",//red
"#EC4899",//pink
"#6B7280",//gray
],
borderWidth:1,
```

```
},
],
},
options:{
responsive:true,
maintainAspectRatio:false,
plugins:{
legend:{
position:"right",
},
tooltip:{
callbacks:{
label:function(context){
constvalue=context.raw;
consttotal=context.dataset.data.reduce(
(a,b)=>a+b,
0
);
constpercentage=((value/total)*100).toFixed(1);
return`₹${value.toLocaleString(
"en-IN"
)}(${percentage}%)`;
},
},
},
},
},
},
});
}else{
//Clearthechartareaandshowamessage
ctx.clearRect(0,0,ctx.canvas.width,ctx.canvas.height);
ctx.fillStyle="#6B7280";
ctx.textAlign="center";
ctx.textBaseline="middle";
ctx.font="14pxArial";
ctx.fillText(
"Noexpensesforcurrentmonth",
```

```
ctx.canvas.width/2,
ctx.canvas.height/2
);
}
})
.catch((error)=>
console.error("Errorfetchingcategorydata:",error)
);
}

//Updatetotalexpensesandmonthlydata
functionupdateExpenses(){
//Fetchtotalexpenses
fetch("/api/expenses")
.then((response)=>response.json())
.then((data)=>{
consttotal=data.reduce(
(sum,expense)=>sum+parseFloat(expense.amount),
0
);
document.getElementById(
"total-expenses"
).textContent=`₹${total.toLocaleString("en-IN",{
minimumFractionDigits:2,
maximumFractionDigits:2,
})}`;

//Updatetheexpensetable
consttableBody=document.getElementById("expense-table-body");
if(data.length===0){
tableBody.innerHTML=`
<tr>
<tdcolspan="5"class="px-6py-4text-centertext-smtext-gray-500">
Noexpensesfound.Addyourfirstexpense!
</td>
</tr>
`;
```

```
return;
}

tableBody.innerHTML=data
.map(
(expense)=>`
<tr>
<tdclass="px-6py-4whitespace-nowraptext-smtext-gray-500">
${expense.date}
</td>
<tdclass="px-6py-4whitespace-nowraptext-smtext-gray-900">
${expense.description}
</td>
<tdclass="px-6py-4whitespace-nowrap">
<spanclass="px-2inline-flextext-xsleading-5font-semiboldrounded-full${
expense.category==="Food"
?"bg-green-100text-green-800"
:expense.category==="Transportation"
?"bg-blue-100text-blue-800"
:expense.category==="Entertainment"
?"bg-purple-100text-purple-800"
:expense.category==="Housing"
?"bg-yellow-100text-yellow-800"
:expense.category==="Utilities"
?"bg-gray-100text-gray-800"
:expense.category==="Healthcare"
?"bg-red-100text-red-800"
:expense.category==="Shopping"
?"bg-pink-100text-pink-800"
:"bg-indigo-100text-indigo-800"
}">
${expense.category}
</span>
</td>
<tdclass="px-6py-4whitespace-nowraptext-smtext-gray-500">
₹${parseFloat(expense.amount).toLocaleString("en-IN",{
minimumFractionDigits:2,
```

```
maximumFractionDigits:2,
})}
</td>
<tdclass="px-6py-4whitespace-nowraptext-righttext-smfont-medium">
<formaction="/delete_expense/${
expense.id
}"method="POST"class="inline">
<buttontype="submit"class="text-red-600hover:text-red-900">
<iclass="fasfa-trash"></i>
</button>
</form>
</td>
</tr>
`
)
.join("");
})
.catch((error)=>console.error("Errorfetchingexpenses:",error));

//Fetchmonthlysummary
fetch("/api/monthly_summary")
.then((response)=>response.json())
.then((data)=>{
//Updatemonthlyexpenses
document.getElementById(
"monthly-expenses"
).textContent=`₹${data.monthly_expenses.toLocaleString("en-IN",{
minimumFractionDigits:2,
maximumFractionDigits:2,
})}`;

//Updatemonthlysavings(onlyiftherareexpenses)
constsavingsText=
data.monthly_expenses>0
?`₹${data.savings.toLocaleString("en-IN",{
minimumFractionDigits:2,
maximumFractionDigits:2,
```

```
})}`
:"₹0.00";
document.getElementById("monthly-savings").textContent=
savingsText;
})
.catch((error)=>
console.error("Errorfetchingmonthlysummary:",error)
);
}

//Searchfunctionality
document
.getElementById("search-expenses")
.addEventListener("input",function(e){
constsearchTerm=e.target.value.toLowerCase();
constrows=document.querySelectorAll("#expense-table-bodytr");

rows.forEach((row)=>{
consttext=row.textContent.toLowerCase();
row.style.display=text.includes(searchTerm)?"":"none";
});
});

//Callupdatefunctionswhenthepageloads
document.addEventListener("DOMContentLoaded",function(){
//Handleincomeformsubmission
document
.getElementById("update-income-form")
.addEventListener("submit",asyncfunction(e){
e.preventDefault();
constform=e.target;
constincome=form.querySelector('input[name="income"]').value;

try{
constresponse=awaitfetch("/update_income",{
method:"POST",
headers:{
```

```
"Content-Type":"application/x-www-form-urlencoded",
},
body:`income=${encodeURIComponent(income)}`,
});

constdata=awaitresponse.json();

if(data.success){
alert("Incomeupdatedsuccessfully!");
//Refreshthepagetoshowupdatedvalues
window.location.reload();
}else{
alert(`Failedtoupdateincome:${data.error}`);
}
}catch(error){
console.error("Error:",error);
alert("Failedtoupdateincome.Pleasetryagain.");
}
});

//Handlebudgetformsubmission
document
.getElementById("update-budget-form")
.addEventListener("submit",asyncfunction(e){
e.preventDefault();
constform=e.target;
constbudget=form.querySelector('input[name="budget"]').value;

try{
constresponse=awaitfetch("/update_budget",{
method:"POST",
headers:{
"Content-Type":"application/x-www-form-urlencoded",
},
body:`budget=${encodeURIComponent(budget)}`,
});
```

```
constdata=awaitresponse.json();

if(data.success){
alert("Budgetupdatedsuccessfully!");
//Refreshthepagetoshowupdatedvalues
window.location.reload();
}else{
alert(`Failedtoupdatebudget:${data.error}`);
}
}catch(error){
console.error("Error:",error);
alert("Failedtoupdatebudget.Pleasetryagain.");
}
});

//Initializethedateinputwithtoday'sdate
consttoday=newDate().toISOString().split("T")[0];
document.getElementById("date").value=today;

//Updateexpensestableandcharts
updateExpenses();
updatePieChart();
});

//Addeventlistenerfordownloadbutton
document
.querySelector('a[href="/download_report"]')
.addEventListener("click",asyncfunction(e){
e.preventDefault();
try{
constresponse=awaitfetch("/download_report");
if(!response.ok){
consterrorData=awaitresponse.json();
thrownewError(errorData.error||"Failedtodownloadreport");
}

//GetthefilenamefromtheContent-Dispositionheader
```

```
constcontentDisposition=response.headers.get(
"Content-Disposition"
);
constfilename=contentDisposition
?contentDisposition.split("filename=")[1].replace(/"/g,"")
:"monthly_report.pdf";

//Createablobfromtheresponse
constblob=awaitresponse.blob();

//Createadownloadlink
consturl=window.URL.createObjectURL(blob);
consta=document.createElement("a");
a.href=url;
a.download=filename;
document.body.appendChild(a);
a.click();

//Cleanup
window.URL.revokeObjectURL(url);
document.body.removeChild(a);
}catch(error){
console.error("Errordownloadingreport:",error);
alert("Failedtodownloadreport:"+error.message);
}
});
</script>
</body>
</html>
```

Sign-up.html:

```
<!DOCTYPEhtml>
<htmllang="en">
<head>
<metacharset="UTF-8"/>
<metaname="viewport"content="width=device-width,initial-scale=1.0"/>
<title>SignUp</title>
<linkrel="stylesheet"href="/static/css/styles.css"/>
```

```
</head>
<body>
<divclass="form-container">
<h1>SignUp</h1>
<formaction="/signup"method="POST">
<labelfor="username">Username:</label>
<inputtype="text"id="username"name="username"required/>

<labelfor="password">Password:</label>
<inputtype="password"id="password"name="password"required/>

<buttontype="submit">SignUp</button>
</form>
<p>Alreadyhaveanaccount?<ahref="/login">Loginhere</a>.</p>
</div>
</body>
</html>

/*CustomstylesbeyondTailwindCSS*/

/*Animationforformsubmissionfeedback*/
@keyframespulse{
0%{
box-shadow:0000rgba(99,102,241,0.7);
}
70%{
box-shadow:00010pxrgba(99,102,241,0);
}
100%{
box-shadow:0000rgba(99,102,241,0);
}
}

.pulse-animation{
animation:pulse1.5sinfinite;
}
```

```
/*Customscrollbar*/
::-webkit-scrollbar{
width:8px;
height:8px;
}

::-webkit-scrollbar-track{
background:#f1f5f9;
border-radius:4px;
}

::-webkit-scrollbar-thumb{
background:#cbd5e1;
border-radius:4px;
}

::-webkit-scrollbar-thumb:hover{
background:#94a3b8;
}

/*Mobileresponsivenessimprovements*/
@media(max-width:640px){
.container{
padding-left:12px;
padding-right:12px;
}

.table-responsive{
display:block;
width:100%;
overflow-x:auto;
-webkit-overflow-scrolling:touch;
}
}

/*Stylesfortheformcontainer*/
.form-container{
```

```
max-width:400px;
margin:50pxauto;
padding:20px;
background:#ffffff;
border-radius:8px;
box-shadow:04px6pxrgba(0,0,0,0.1);
}

.form-containerh1{
font-size:24px;
margin-bottom:20px;
color:#4f46e5;
}

.form-containerlabel{
display:block;
margin-bottom:8px;
font-weight:bold;
color:#374151;
}

.form-containerinput{
width:100%;
padding:10px;
margin-bottom:20px;
border:1pxsolid#d1d5db;
border-radius:4px;
}

.form-containerbutton{
width:100%;
padding:10px;
background:#4f46e5;
color:#ffffff;
border:none;
border-radius:4px;
cursor:pointer;
```

```
font-size:16px;
}

.form-containerbutton:hover{
background:#4338ca;
}

/*HeroSectionStyles*/
section.bg-indigo-600{
background:linear-gradient(90deg,#4f46e5,#3b82f6);
}

section.bg-indigo-600h1{
font-size:3rem;
font-weight:bold;
margin-bottom:1rem;
}

section.bg-indigo-600p{
font-size:1.25rem;
margin-bottom:1.5rem;
}

section.bg-indigo-600a{
font-size:1rem;
font-weight:bold;
padding:0.75rem1.5rem;
border-radius:0.5rem;
transition:background-color0.3sease;
}

section.bg-indigo-600a:hover{
background-color:#e5e7eb;
}

body{
min-height:100vh;
```

```
margin:0;
font-family:"SegoeUI","Roboto",Arial,sans-serif;
}

.form-bg{
min-height:100vh;
display:flex;
align-items:center;
justify-content:center;
background:rgba(0,0,0,0.03);
}

.modern-login{
box-shadow:08px32px0rgba(31,38,135,0.18);
backdrop-filter:blur(4px);
border-radius:18px;
padding:36px32px28px32px;
background:rgba(255,255,255,0.95);
border:1pxsolidrgba(255,255,255,0.3);
transition:box-shadow0.3s;
}

.modern-loginh1{
font-size:2.2rem;
font-weight:700;
color:#3b3b5c;
margin-bottom:1.5rem;
letter-spacing:1px;
text-align:center;
}

.modern-loginlabel{
font-size:1rem;
color:#4f46e5;
margin-bottom:0.3rem;
}
```

```
.modern-logininput{
padding:12px14px;
border-radius:6px;
border:1pxsolid#d1d5db;
margin-bottom:1.2rem;
font-size:1rem;
background:#f8fafc;
transition:border0.2s;
}

.modern-logininput:focus{
border:1.5pxsolid#4f46e5;
outline:none;
background:#fff;
}

.btn-primary{
background:linear-gradient(90deg,#4f46e5,#3b82f6);
color:#fff;
border:none;
border-radius:6px;
padding:12px0;
font-size:1.1rem;
font-weight:600;
cursor:pointer;
box-shadow:02px8pxrgba(79,70,229,0.08);
transition:background0.2s,box-shadow0.2s;
}

.btn-primary:hover{
background:linear-gradient(90deg,#6366f1,#2563eb);
box-shadow:04px16pxrgba(59,130,246,0.13);
}

.signup-link{
text-align:center;
margin-top:1.2rem;
```

```
color:#64748b;
font-size:1rem;
}

.signup-linka{
color:#4f46e5;
text-decoration:none;
font-weight:500;
transition:color0.2s;
}

.signup-linka:hover{
color:#2563eb;
text-decoration:underline;
}

document.addEventListener('DOMContentLoaded',function(){
//Fetchexpensesdataforchartsandsummary
fetchExpenses();

//Fetchmonthlysummarydata
fetchMonthlySummary();

//Initializesearchfunctionality
initSearch();
});

//FetchexpensesdatafromtheAPI
asyncfunctionfetchExpenses(){
try{
constresponse=awaitfetch('/api/expenses');
constexpenses=awaitresponse.json();

//Updatesummarymetrics
updateSummaryMetrics(expenses);

//Fetchcategorydataandupdatechart
```

```
fetchCategoryData();
}catch(error){
console.error('Errorfetchingexpenses:',error);
}
}

//FetchmonthlysummarydatafromtheAPI
asyncfunctionfetchMonthlySummary(){
try{
constresponse=awaitfetch('/api/monthly_summary');
constdata=awaitresponse.json();

//Updatemonthlysummarymetrics
updateMonthlySummary(data);
}catch(error){
console.error('Errorfetchingmonthlysummary:',error);
}
}

//Updatesummarymetricsbasedonexpensesdata
functionupdateSummaryMetrics(expenses){
consttotalElement=document.getElementById('total-expenses');

if(expenses.length===0){
totalElement.textContent='₹0.00';
return;
}

//Calculatetotalwithexplicitnumberconversion
consttotal=expenses.reduce((sum,expense)=>sum+parseFloat(expense.amount),0);

//UpdateDOMelements
totalElement.textContent='₹'+total.toFixed(2);
}

//Updatemonthlysummarymetrics
functionupdateMonthlySummary(data){
```

```
constmonthlyExpensesElement=document.getElementById('monthly-expenses');
constmonthlySavingsElement=document.getElementById('monthly-savings');
constbudgetPercentageElement=document.getElementById('budget-percentage');

if(monthlyExpensesElement){
monthlyExpensesElement.textContent='₹'+data.monthly_expenses.toFixed(2);
}

if(monthlySavingsElement){
monthlySavingsElement.textContent='₹'+data.savings.toFixed(2);
}

if(budgetPercentageElement&&data.monthly_budget>0){
constpercentage=(data.monthly_expenses/data.monthly_budget)*100;
budgetPercentageElement.textContent=percentage.toFixed(1)+'%ofbudgetused';

//Updatebudgetprogressbar
constprogressBar=budgetPercentageElement.parentElement.nextElementSibling.querySelector('div');
if(progressBar){
if(percentage>100){
progressBar.style.width='100%';
progressBar.className='bg-red-600h-2.5rounded-full';
}else{
progressBar.style.width=percentage+'%';
progressBar.className=percentage>90?'bg-yellow-600h-2.5rounded-full':'bg-green-600h-2.5rounded-
full';
}
}
}
}

//Fetchcategorydataforthepiechart
asyncfunctionfetchCategoryData(){
try{
constresponse=awaitfetch('/api/categories');
constcategoryData=awaitresponse.json();
```

```
if(categoryData.length>0){
createCategoryChart(categoryData);
}else{
//Handleemptydata
constchartContainer=document.getElementById('category-chart');
chartContainer.innerHTML='<divclass="flexh-fullitems-centerjustify-centertext-gray-
400">Nocategorydataavailable</div>';
}
}catch(error){
console.error('Errorfetchingcategorydata:',error);
}
}

//Createthecategorypiechart
functioncreateCategoryChart(categoryData){
constctx=document.getElementById('category-chart').getContext('2d');

//Chartcolors
constcolors=[
'rgba(99,102,241,0.8)',//Indigo
'rgba(16,185,129,0.8)',//Green
'rgba(59,130,246,0.8)',//Blue
'rgba(217,70,239,0.8)',//Purple
'rgba(245,158,11,0.8)',//Yellow
'rgba(239,68,68,0.8)',//Red
'rgba(236,72,153,0.8)',//Pink
'rgba(107,114,128,0.8)'//Gray
];

//PreparedataforChart.js
constlabels=categoryData.map(item=>item.name);
constvalues=categoryData.map(item=>item.value);

//Createchart
newChart(ctx,{
type:'doughnut',
data:{
```

```
labels:labels,
datasets:[{
data:values,
backgroundColor:colors.slice(0,categoryData.length),
borderWidth:0
}]
},
options:{
responsive:true,
maintainAspectRatio:false,
plugins:{
legend:{
position:'right',
labels:{
font:{
size:12
},
padding:15
}
},
tooltip:{
callbacks:{
label:function(context){
constlabel=context.label||'';
constvalue=context.raw.toFixed(2);
return`${label}:₹${value}`;
}
}
}
},
cutout:'70%'
}
});
}

//Initializeexpensesearchfunctionality
functioninitSearch(){
```

```
constsearchInput=document.getElementById('search-expenses');

if(searchInput){
searchInput.addEventListener('input',function(){
constquery=this.value.toLowerCase();
constrows=document.querySelectorAll('#expense-table-bodytr');

rows.forEach(row=>{
if(row.cells.length>1){//Skipemptystaterow
constdate=row.cells[0].textContent.toLowerCase();
constdescription=row.cells[1].textContent.toLowerCase();
constcategory=row.cells[2].textContent.toLowerCase();

if(date.includes(query)||description.includes(query)||category.includes(query)){
row.style.display='';
}else{
row.style.display='none';
}
}
});
});
}
}
```

app.py:

```
document.addEventListener('DOMContentLoaded',function(){
//Fetchexpensesdataforchartsandsummary
fetchExpenses();

//Fetchmonthlysummarydata
fetchMonthlySummary();

//Initializesearchfunctionality
initSearch();
});
```

```
//FetchexpensesdatafromtheAPI
asyncfunctionfetchExpenses(){
try{
constresponse=awaitfetch('/api/expenses');
constexpenses=awaitresponse.json();

//Updatesummarymetrics
updateSummaryMetrics(expenses);

//Fetchcategorydataandupdatechart
fetchCategoryData();
}catch(error){
console.error('Errorfetchingexpenses:',error);
}
}

//FetchmonthlysummarydatafromtheAPI
asyncfunctionfetchMonthlySummary(){
try{
constresponse=awaitfetch('/api/monthly_summary');
constdata=awaitresponse.json();

//Updatemonthlysummarymetrics
updateMonthlySummary(data);
}catch(error){
console.error('Errorfetchingmonthlysummary:',error);
}
}

//Updatesummarymetricsbasedonexpensesdata
functionupdateSummaryMetrics(expenses){
consttotalElement=document.getElementById('total-expenses');

if(expenses.length===0){
totalElement.textContent='₹0.00';
return;
}
```

```
//Calculatetotalwithexplicitnumberconversion
consttotal=expenses.reduce((sum,expense)=>sum+parseFloat(expense.amount),0);

//UpdateDOMelements
totalElement.textContent='₹'+total.toFixed(2);
}

//Updatemonthlysummarymetrics
functionupdateMonthlySummary(data){
constmonthlyExpensesElement=document.getElementById('monthly-expenses');
constmonthlySavingsElement=document.getElementById('monthly-savings');
constbudgetPercentageElement=document.getElementById('budget-percentage');

if(monthlyExpensesElement){
monthlyExpensesElement.textContent='₹'+data.monthly_expenses.toFixed(2);
}

if(monthlySavingsElement){
monthlySavingsElement.textContent='₹'+data.savings.toFixed(2);
}

if(budgetPercentageElement&&data.monthly_budget>0){
constpercentage=(data.monthly_expenses/data.monthly_budget)*100;
budgetPercentageElement.textContent=percentage.toFixed(1)+'%ofbudgetused';

//Updatebudgetprogressbar
constprogressBar=budgetPercentageElement.parentElement.nextElementSibling.querySelector('div');
if(progressBar){
if(percentage>100){
progressBar.style.width='100%';
progressBar.className='bg-red-600h-2.5rounded-full';
}else{
progressBar.style.width=percentage+'%';
progressBar.className=percentage>90?'bg-yellow-600h-2.5rounded-full':'bg-green-600h-2.5rounded-
full';
}
```

```
}
}
}

//Fetchcategorydataforthepiechart
asyncfunctionfetchCategoryData(){
try{
constresponse=awaitfetch('/api/categories');
constcategoryData=awaitresponse.json();

if(categoryData.length>0){
createCategoryChart(categoryData);
}else{
//Handleemptydata
constchartContainer=document.getElementById('category-chart');
chartContainer.innerHTML='<divclass="flexh-fullitems-centerjustify-centertext-gray-
400">Nocategorydataavailable</div>';
}
}catch(error){
console.error('Errorfetchingcategorydata:',error);
}
}

//Createthecategorypiechart
functioncreateCategoryChart(categoryData){
constctx=document.getElementById('category-chart').getContext('2d');

//Chartcolors
constcolors=[
'rgba(99,102,241,0.8)',//Indigo
'rgba(16,185,129,0.8)',//Green
'rgba(59,130,246,0.8)',//Blue
'rgba(217,70,239,0.8)',//Purple
'rgba(245,158,11,0.8)',//Yellow
'rgba(239,68,68,0.8)',//Red
'rgba(236,72,153,0.8)',//Pink
'rgba(107,114,128,0.8)'//Gray
```

```
];

//PreparedataforChart.js
constlabels=categoryData.map(item=>item.name);
constvalues=categoryData.map(item=>item.value);

//Createchart
newChart(ctx,{
type:'doughnut',
data:{
labels:labels,
datasets:[{
data:values,
backgroundColor:colors.slice(0,categoryData.length),
borderWidth:0
}]
},
options:{
responsive:true,
maintainAspectRatio:false,
plugins:{
legend:{
position:'right',
labels:{
font:{
size:12
},
padding:15
}
},
tooltip:{
callbacks:{
label:function(context){
constlabel=context.label||'';
constvalue=context.raw.toFixed(2);
return`${label}:₹${value}`;
}
```

```
}
}
},
cutout:'70%'
}
});
}

//Initializeexpensesearchfunctionality
functioninitSearch(){
constsearchInput=document.getElementById('search-expenses');

if(searchInput){
searchInput.addEventListener('input',function(){
constquery=this.value.toLowerCase();
constrows=document.querySelectorAll('#expense-table-bodytr');

rows.forEach(row=>{
if(row.cells.length>1){//Skipemptystaterow
constdate=row.cells[0].textContent.toLowerCase();
constdescription=row.cells[1].textContent.toLowerCase();
constcategory=row.cells[2].textContent.toLowerCase();

if(date.includes(query)||description.includes(query)||category.includes(query)){
row.style.display='';
}else{
row.style.display='none';
}
}
});
});
}
}
```

confi.py:

importos

```
fromdotenvimportload_dotenv

#Loadenvironmentvariablesfrom.envfile
load_dotenv()

classConfig:
#Databaseconfiguration
MYSQL_HOST=os.getenv('MYSQL_HOST','localhost')
MYSQL_USER=os.getenv('MYSQL_USER','root')
MYSQL_PASSWORD=os.getenv('MYSQL_PASSWORD','')
MYSQL_DB=os.getenv('MYSQL_DB','expense_tracker')

#Fileuploadconfiguration
UPLOAD_FOLDER=os.path.join(os.path.dirname(os.path.abspath(__file__)),'reports')

#Ensuretheuploadfolderexists
os.makedirs(UPLOAD_FOLDER,exist_ok=True)

#Serverconfiguration
HOST='0.0.0.0'
PORT=5000
DEBUG=True

#Applicationconfiguration
SECRET_KEY=os.getenv('SECRET_KEY','your-secret-key-here')

init_db.py:

importmysql.connector
fromconfigimportConfig

definit_database():
try:
#ConnecttoMySQLserver
conn=mysql.connector.connect(
host=Config.MYSQL_HOST,
user=Config.MYSQL_USER,
```

```
password=Config.MYSQL_PASSWORD,
auth_plugin='mysql_native_password'
)
cursor=conn.cursor()

#Createdatabaseifitdoesn'texist
cursor.execute(f"CREATEDATABASEIFNOTEXISTS{Config.MYSQL_DB}")
cursor.execute(f"USE{Config.MYSQL_DB}")

#Dropexistingtablesiftheyexist
cursor.execute("DROPTABLEIFEXISTSexpenses")
cursor.execute("DROPTABLEIFEXISTSsettings")

#Createexpensestable
cursor.execute('''
CREATETABLEexpenses(
idINTAUTO_INCREMENTPRIMARYKEY,
amountDECIMAL(10,2)NOTNULL,
categoryVARCHAR(50)NOTNULL,
descriptionTEXT,
dateDATENOTNULL
)
''')

#Createsettingstable
cursor.execute('''
CREATETABLEsettings(
setting_keyVARCHAR(50),
valueTEXTNOTNULL,
user_idVARCHAR(50),
PRIMARYKEY(setting_key,user_id)
)
''')

#Insertdefaultsettings
cursor.execute('''
INSERTINTOsettings(setting_key,value,user_id)
```

```
VALUES
('monthly_income','0','default'),
('monthly_expense','0','default')
''')

conn.commit()
print("Databaseinitializedsuccessfully!")

exceptmysql.connector.Erroraserr:
print(f"Errorinitializingdatabase:{err}")
finally:
if'conn'inlocals():
cursor.close()
conn.close()

if__name__=='__main__':
init_database()

run.py:

importos
importsys
importwebbrowser
fromthreadingimportTimer
fromappimportapp
fromconfigimportConfig

defopen_browser():
"""OpenthebrowsertotheapplicationURLafterashortdelay"""
webbrowser.open('http://127.0.0.1:5000/')

if__name__=="__main__":
#CheckifFlaskisinstalled
try:
fromflaskimportFlask
exceptImportError:
print("Flaskisnotinstalled.InstallingFlask...")
```

```
os.system(f"{sys.executable}-mpipinstallflask")

#SetFlaskenvironmentvariables
os.environ['FLASK_APP']='app.py'
os.environ['FLASK_ENV']='development'

#Openbrowserafterashortdelay
Timer(1.5,open_browser).start()

#StarttheFlaskapplication
print("StartingExpenseTrackerapplication...")
print(f"Accesstheapplicationat:http://{Config.HOST}:{Config.PORT}/")
print("PressCTRL+Ctostoptheserver.")

#RuntheFlaskapplication
app.run(
host=Config.HOST,
port=Config.PORT,
debug=Config.DEBUG
)
```

Error Handling

Error handling ensures the system behaves gracefully when unexpected situations arise, improving user experience and maintaining data integrity.

1. User Input Errors

Description: Handles invalid or missing user inputs in forms.

Scenario	Handling Approach
Missing required fields	Flash message (e.g., “All fields are required”)
Invalid email format	Regex validation + user message
Negative/zero amount entered	Client-side + server-side validation
Invalid date format	HTML5 date input + server-side fallback

2. Authentication & Authorization Errors

Description: Secures routes and prevents unauthorized access.

Scenario	Handling Approach
Incorrect login credentials	Flash error: “Invalid username or password”
Accessing protected page without login	Redirect to login with warning message
Session timeout or missing cookie	Force logout and redirect

3. Database Errors

Description: Handles issues connecting to or querying the MySQL database.

Scenario	Handling Approach
Database connection failure	Try-except block + user-friendly error page
SQL integrity constraint (e.g. duplicate email)	Flash error: “Email already registered”
Invalid data insertion (e.g. NULL fields)	Transaction rollback + error log

4. File/Report Generation Errors

Description: Handles problems during Excel export or file operations.

Scenario	Handling Approach
Missing data while generating reports	Flash message: “No data available for report”
File write permission denied	Log error + user alert
Export library error (e.g., openpyxl)	Try-except with detailed traceback logging

5. Server-Side Errors

Description: Prevent app crash on unexpected exceptions.

- All major Flask routes include `try-except` blocks.
- Use Flask error handlers for common HTTP status codes:

```
python
CopyEdit
@app.errorhandler(404)
def not_found_error(e):
    return render_template("404.html"), 404

@app.errorhandler(500)
def internal_error(e):
    # Optionally log the error here
    return render_template("500.html"), 500
```

6. Client-Side (JavaScript/UI) Errors

Description: Validate forms and handle UI logic issues before submission.

- Validate forms using JavaScript before sending to backend.
- Disable submit buttons after first click to prevent duplicates.
- Handle Chart.js errors for missing/incomplete data.

BIBLIOGRAPHY

- **Reitz, Miguel.** (2021). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.
    - A comprehensive guide to building web applications using Flask, covering everything from setup to deployment.

- **McKinney, Wes.** (2017). *Python for Data Analysis*. O'Reilly Media.
    - Focuses on data manipulation and analysis using Python libraries such as Pandas and NumPy.

- **VanderPlas, Jake.** (2016). *Python Data Science Handbook*. O'Reilly Media.
    - Covers core data science tools and techniques including data visualization using matplotlib and seaborn.

- **Lutz, Mark.** (2013). *Learning Python*. O'Reilly Media.
    - A thorough exploration of Python language fundamentals, suitable for both beginners and intermediate developers.

- **Beazley, David.** (2009). *Python Essential Reference*. Addison-Wesley.
    - A concise and authoritative reference book for experienced Python developers.

- **Robbins, Jennifer.** (2018). *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics*. O'Reilly Media.
    - Useful for understanding web design principles and how to integrate HTML/CSS with Flask applications.

- **Kropf, Peter.** (2015). *Software Engineering: The Development Process*. Springer.
    - Useful for understanding the software development life cycle, including requirement analysis and testing.

- **MySQL Documentation.** (2024). *MySQL 8.0 Reference Manual*. Oracle.
    - Official and detailed documentation on MySQL queries, functions, and database management.