

# **CASH CRAFT**

A Project report submitted by

**ANJANA SURESH**  
(VAS23MCA - 2011)

to APJ Abdul Kalam Technological University  
in partial fulfillment of the requirements for the award of the degree of  
Master of Computer Applications



**Department of Computer Applications**

Vidya Academy of Science & Technology

Thalakkottukara, Thrissur - 680 501

April 2025

**Department of Computer Applications**  
**Vidya Academy of Science & Technology**  
**Thalakkottukara, Thrissur - 680 501**  
(<http://www.vidyaacademy.ac.in>)



## CERTIFICATE

This is to certify that the report titled **CASH CRAFT** is a bona-fide record of the work related to the paper 20MCA246 MAIN PROJECT done by **ANJANA SURESH (Reg. No. VAS23MCA - 2011)** of S4 MCA (2023 admissions) class of Vidya Academy of Science & Technology, Thrissur - 680501 in partial fulfillment of the requirement for the award of the Degree of Master of Computer Applications of APJ Abdul Kalam Technological University.

**Guide/Supervisor**

Name : Mr.DIJESH P

Signature : .....

Date : April 11, 2025

**Head of Department**

Name : Dr Reji C Joy

Signature : .....

Date : April 11, 2025

(Seal of Department of Computer Applications)

**External Supervisor**

Name : .....

Signature : .....

Date : .....



## Declaration

I, **ANJANA SURESH**, studying in Fourth Semester MCA (2023 admissions) class of Vidya Academy of Science & Technology, Thrissur – 680501, hereby declare that the project report (“CASH CRAFT”) submitted by me for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of APJ Abdul Kalam Technological University, Kerala, is a bona fide work done by me under supervision of Mr.DIJESH P. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place : Thrissur – 680501

Signature of student : .....

Date : April 11, 2025

Name of student : ANJANA SURESH



## Acknowledgment

I wish to record my indebtedness and thankfulness to all who helped me prepare this report titled CASH CRAFT and present it in a satisfactory way. This report is part of my work related to the paper 20MCA246 MAIN PROJECT.

I am especially thankful to my guide and supervisor Mr.DIJESH P in the Department of Computer Applications for giving me valuable suggestions and critical inputs in the preparation of this report. I am also thankful to Dr Reji C Joy, the Head of Computer Applications for encouragement.

My friends in my class have always been helpful and I am grateful to them for their constant help and support.

ANJANA SURESH  
Reg. No. VAS23MCA - 2011  
S4 MCA (2023 Admissions)  
Vidya Academy of Science & Technology  
Thrissur -680 501.



## Synopsis

CashCraft is a web-based platform designed to help users effectively manage their personal finances by tracking income, expenses, savings, and budgeting goals. It empowers users to maintain financial discipline through real-time monitoring of their spending habits and savings progress. With features tailored to personal finance management, users can categorize expenses, set monthly budgets, and receive intelligent alerts for overspending.

CashCraft includes a personalized dashboard that displays key financial metrics such as total income, total expenses, balance, and progress toward financial goals. Users can create savings goals, allocate funds, and track their progress visually. The system supports intuitive income and expense input, along with categorization for better financial insight.

The platform features interactive elements like a motivational quote carousel to encourage financial wellness, and visual representations such as pie charts and progress bars for spending analysis. Additionally, it offers mobile-friendly navigation and a user-centric interface aligned with modern UI/UX trends to enhance user experience.

CashCraft also includes overspending alerts, goal tracking modules, and a budget monitoring system that adapts to user behavior. These features ensure users stay aligned with their financial objectives, making CashCraft a comprehensive tool for budget-conscious individuals striving for smarter financial decisions and long-term savings.

# Contents

<b>Certificate from College</b>	<b>2</b>
<b>Declaration by Student</b>	<b>3</b>
<b>Acknowledgment</b>	<b>4</b>
<b>Synopsis</b>	<b>5</b>
<b>Table of Contents</b>	<b>6</b>
<b>List of Tables</b>	<b>9</b>
<b>List of Figures</b>	<b>10</b>
<b>1 INTRODUCTION</b>	<b>2</b>
1.1 Project Overview . . . . .	2
<b>2 SYSTEM ANALYSIS</b>	<b>3</b>
2.1 The Existing System . . . . .	3
2.2 Proposed System . . . . .	3
<b>3 FEASIBILITY STUDY</b>	<b>5</b>
3.1 Technical Feasibility . . . . .	5
3.2 Economical Feasibility . . . . .	5
3.3 Operational Feasibility . . . . .	5
<b>4 SYSTEM DESIGN</b>	<b>7</b>
4.1 Input Design . . . . .	7
4.2 Output Design . . . . .	7
4.3 Database Design . . . . .	7
4.3.1 Normalization . . . . .	8

4.3.2 Database Tables and Keys . . . . .	8
<b>5 TESTING AND VALIDATION</b>	<b>10</b>
5.1 Testing Methods . . . . .	10
5.1.1 Unit Testing . . . . .	10
5.1.2 Integration Testing . . . . .	10
5.1.3 White Box Testing . . . . .	11
5.1.4 Black Box Testing . . . . .	11
5.2 Validation . . . . .	11
<b>6 SYSTEM SECURITY MEASURES</b>	<b>12</b>
6.0.1 Authentication and Access Control . . . . .	12
6.0.2 Data Protection and Encryption . . . . .	12
6.0.3 Network Security . . . . .	12
6.0.4 Software and Application Security . . . . .	12
6.0.5 Backup and Disaster Recovery . . . . .	13
6.0.6 Monitoring and Auditing . . . . .	13
6.0.7 Physical Security . . . . .	13
6.0.8 Security Awareness and Training . . . . .	13
<b>7 CONCLUSION</b>	<b>14</b>
<b>8 SCOPE FOR FUTURE ENHANCEMENT</b>	<b>15</b>
<b>APPENDICES</b>	<b>16</b>
<b>A SCRUM PROCESS ARTIFACTS</b>	<b>17</b>
A.1 Product Backlog . . . . .	17
A.1.1 Functional Requirements . . . . .	18
A.1.2 Non Functional Requirements . . . . .	18
A.1.3 System Configuration . . . . .	18
A.2 Sprint Details . . . . .	21
A.3 Details of daily scrum meeting . . . . .	21
A.4 Details of bi-weekly meeting . . . . .	22
<b>B VERSIONING</b>	<b>23</b>
<b>C DATA FLOW DIAGRAMS</b>	<b>24</b>
<b>D TABLE STRUCTURE</b>	<b>30</b>

<b>E SAMPLE SCREENSHOTS</b>	<b>32</b>
<b>F REPORTS</b>	<b>37</b>
F.0.1 Login Page . . . . .	37
F.0.2 Register Page . . . . .	38
F.0.3 User Dashboard Page . . . . .	40
F.0.4 Add Income Page . . . . .	62
F.0.5 View Income List Page . . . . .	67
F.0.6 Add Expense Page . . . . .	74
F.0.7 View Expense List Page . . . . .	76
F.0.8 Goal List View Page . . . . .	80
F.0.9 Goal Create Page . . . . .	85
<b>Bibliography</b>	<b>89</b>

# List of Tables

D.1	UserProfile Table Structure . . . . .	30
D.2	Income Table Structure . . . . .	31
D.3	Expense Table Structure . . . . .	31
D.4	Goal Table Structure . . . . .	31

# List of Figures

A.1 Entity-Relationship (ER) Diagram of the System . . . . .	19
C.1 DFD Level 0 . . . . .	24
C.2 DFD Level 1 . . . . .	24
C.3 DFD Level 2.1 . . . . .	25
C.4 DFD Level 2.2 . . . . .	26
C.5 DFD Level 2.3 . . . . .	27
C.6 DFD Level 2.4 . . . . .	28
C.7 DFD Level 2.5 . . . . .	29
E.1 Index Page . . . . .	33
E.2 Register Page . . . . .	33
E.3 Login Page . . . . .	33
E.4 User Dashboard page . . . . .	34
E.5 Edit Profile Page . . . . .	34
E.6 Add Income Page . . . . .	35
E.7 View Income List page . . . . .	35
E.8 Update Income Page . . . . .	35
E.9 Add Expense Page . . . . .	35
E.10 View Expense Page . . . . .	36
E.11 Update Expense Page . . . . .	36
E.12 Goal Page . . . . .	36
E.13 Create New Goal Page . . . . .	36

# PROJECT REPORT

## Chapter 1

# INTRODUCTION

CashCraft is a web-based platform designed to help users manage their personal finances by tracking income, expenses, and savings goals. It allows users to create budgets, categorize spending, and monitor financial progress through a clean and interactive dashboard. With features like overspending alerts and goal tracking, CashCraft provides a simple yet effective solution for improving financial discipline and achieving savings targets.

### 1.1 Project Overview

Many individuals face challenges in managing their finances due to a lack of proper planning, limited financial awareness, and inconsistent tracking of income and expenses. Without a structured system, it becomes difficult to monitor spending habits, set savings goals, or stay within a budget. This often leads to overspending, poor savings, and financial stress.

CashCraft addresses these issues by offering a smart, user-friendly platform for personal finance management. It allows users to record income and expenses, categorize transactions, and set monthly budgets and savings goals. Through visual insights like charts and progress bars, users can better understand their financial behavior. Additionally, features such as overspending alerts, motivational prompts, and mobile-friendly access help users stay consistent and disciplined, promoting healthier financial habits and long-term stability.

## Chapter 2

# SYSTEM ANALYSIS

### 2.1 The Existing System

In the current scenario, many individuals manage their finances using manual methods such as notebooks or spreadsheets, or through multiple disjointed financial apps. These methods are often time-consuming, error-prone, and lack the real-time insights needed for effective budget control. Users may find it difficult to consistently track income, expenses, and savings, leading to poor financial planning and unmonitored overspending.

Existing finance tracking solutions often provide limited features, with minimal customization for budgeting, savings goals, or detailed expense categorization. Additionally, many platforms do not offer visual progress tracking or timely alerts, making it difficult for users to stay within budget or work toward financial goals effectively. As a result, there is a clear need for an integrated and intelligent system like CashCraft that centralizes all essential finance management features in a single, user-friendly platform.

### 2.2 Proposed System

CashCraft aims to provide a centralized, intelligent, and user-friendly platform for effective personal finance management. Unlike traditional manual methods or fragmented financial tools, this web-based system integrates income tracking, expense management, budget planning, and savings goal monitoring into a single cohesive solution. This ensures users can manage all aspects of their finances from one place with improved efficiency and clarity.

The system enables users to categorize expenses, set monthly budgets, and track progress toward savings goals through visual dashboards and real-time analytics. Interactive features such as pie charts, progress bars, and financial summaries help users better understand their spending habits and make informed financial decisions. Additionally, CashCraft includes overspending

alerts and motivational prompts to encourage users to stay within budget and consistently work towards their financial targets.

By combining all essential financial tools into a modern, mobile-friendly interface, CashCraft enhances user engagement and simplifies the process of managing day-to-day finances, promoting better budgeting habits and long-term financial stability.

## Chapter 3

# FEASIBILITY STUDY

A feasibility study evaluates the practicality and viability of implementing the CashCraft system by analyzing technical, economic, and operational aspects to ensure successful development and deployment.

### 3.1 Technical Feasibility

The system is developed as a web-based application using Django (Python) for the backend and HTML, CSS, JavaScript, and Tailwind CSS for the frontend. This technology stack offers scalability, security, and efficient performance. A relational database such as PostgreSQL or MySQL is used to store and manage user and financial data securely. The application supports cloud deployment, ensuring accessibility across devices and ease of future updates or enhancements.

### 3.2 Economical Feasibility

CashCraft is a cost-effective solution as it leverages open-source technologies, reducing licensing and development costs. Hosting and domain registration are the primary expenses. The system can be monetized in the future through premium features or advertisements, making it a financially viable product with potential for long-term sustainability.

### 3.3 Operational Feasibility

The system is designed with a focus on usability and simplicity, allowing users with minimal technical knowledge to manage their finances efficiently. Features like visual dashboards, expense categorization, goal tracking, and overspending alerts make the platform practical and engaging



for day-to-day financial management. The intuitive interface ensures high user adoption and consistent usage.

## Chapter 4

# SYSTEM DESIGN

### 4.1 Input Design

Input design is the process of specifying how users will interact with a system to enter data. It defines the layout, structure, and format of input forms or fields, ensuring data entry is both easy for users and accurate for the system. Input design is crucial for capturing user information correctly, minimizing errors, and optimizing the user experience. Overall, input design aims to make the data entry process seamless, reduce errors, and improve the overall reliability of the system by ensuring data is complete and accurate.

### 4.2 Output Design

Output design focuses on presenting information in a clear, accurate, and user-friendly manner to enhance decision-making and system usability. It ensures data is well-organized, relevant, and accessible through various formats such as reports, dashboards, graphs, or notifications. Outputs can be internal (for organizational use), external (shared with clients or stakeholders), or interactive (dynamic dashboards). Effective output design improves efficiency, readability, and security, making information easily understandable and actionable for users.

### 4.3 Database Design

Database design is the process of structuring and organizing data efficiently to ensure easy storage, retrieval, and management. It involves defining tables, relationships, constraints, and normalization to maintain data integrity and minimize redundancy. A well-designed database improves performance, scalability, and security while optimizing queries for better efficiency. It typically includes key components like entities, attributes, primary keys, foreign keys, and indexing to support

seamless data handling. Proper database design is essential for applications to function smoothly, ensuring accuracy and reliability in data management.

### 4.3.1 Normalization

Normalization is the process of organizing a relational database to reduce data redundancy and improve data integrity. It involves dividing large tables into smaller ones and defining relationships to ensure efficient storage, retrieval, and updating of data. The goal of normalization is to eliminate anomalies in insertion, deletion, and updating while maintaining consistency.

#### 4.3.1.1 First Normal Form

A table is in First Normal Form (1NF) if:

All columns contain atomic (indivisible) values.

Each column contains only one type of data (no multiple values in a single field).

Each row has a unique identifier (primary key).

#### 4.3.1.2 Second Normal Form

A table is in Second Normal Form (2NF) if:

It is already in First Normal Form (1NF) (all values are atomic).

It does not have partial dependencies, meaning every non-key attribute must be fully dependent on the entire primary key and not just part of it.

#### 4.3.1.3 Third Normal Form

A table is in Third Normal Form (3NF) if:

It is already in Second Normal Form (2NF) (no partial dependencies).

It does not have transitive dependencies, meaning non-key attributes should depend only on the primary key and not on another non-key attribute.

### 4.3.2 Database Tables and Keys

A database table is a structured collection of related data organized into rows (records) and columns (fields). Each table typically represents an entity (e.g., Users, Orders, Products) and consists of attributes defining its properties.

#### 4.3.2.1 Primary Key

A unique identifier for each record in a table.

Cannot have NULL values and must be unique.

#### **4.3.2.2 Foreign Key**

A column that establishes a relationship between two tables.

It references a Primary Key from another table.

Ensures referential integrity (no orphan records).

#### **4.3.2.3 Candidate Key**

A set of columns that can uniquely identify a record.

One of them is chosen as the Primary Key.

#### **4.3.2.4 Composite Key**

A key formed by two or more columns to uniquely identify a record.

Used when a single column is not enough to define uniqueness.

#### **4.3.2.5 Alternate Key**

A candidate key that was not selected as the primary key.

#### **4.3.2.6 Super Key**

A set of one or more attributes that can uniquely identify a record.

Includes the Primary Key and any additional columns.

#### **4.3.2.7 Unique Key**

Ensures that the values in a column remain unique but can have NULL values.

## Chapter 5

# TESTING AND VALIDATION

### 5.1 Testing Methods

#### 5.1.1 Unit Testing

Unit testing is a type of software testing that focuses on testing individual units or components of a software application in isolation. A "unit" is typically the smallest testable part of an application, such as a function, method, or class. The goal of unit testing is to verify that each unit works as expected independently of the rest of the system. Unit testing is an essential practice for maintaining high-quality, reliable code. By testing individual functions or methods in isolation, unit tests help identify bugs early and ensure that each component behaves as expected. In your budget tracker, unit tests for functions like calculatebalance, addincome, and addexpense would ensure that each part of your code works correctly, enabling smooth user experiences and reducing the likelihood of unexpected errors.

#### 5.1.2 Integration Testing

Integration testing ensures that different components or modules of an application work together as intended. It focuses on testing the interactions between integrated units to verify data flow, functionality, and interfaces. The goal is to detect issues related to communication between modules, such as incorrect data handling or broken API calls. It can be performed incrementally or using top-down, bottom-up, or big bang approaches. By identifying integration issues early, this testing improves the overall reliability of the system. Successful integration testing ensures a seamless interaction between all components before the final deployment.

### 5.1.3 White Box Testing

White box testing is a software testing technique where the internal structure, design, and implementation of the code are examined and tested. This method requires testers to have knowledge of the code, allowing them to create test cases that cover specific functions, logic, paths, and conditions. It ensures that each line, branch, and pathway of the code functions correctly and meets design specifications. White box testing helps in identifying issues related to code efficiency, logic errors, and security vulnerabilities. Common techniques include path testing, loop testing, and control flow testing. This approach is often used in unit testing and integration testing to improve code quality and robustness.

### 5.1.4 Black Box Testing

Black box testing is a software testing technique that focuses on evaluating the functionality of an application without examining its internal code structure. Testers assess the software by providing inputs and verifying the outputs, ensuring that the system meets requirements and performs as expected. This method doesn't require knowledge of the code or internal workings, making it ideal for testing from a user's perspective. Black box testing helps identify issues related to user interface, functionality, performance, and usability. Common techniques include equivalence partitioning, boundary value analysis, and decision table testing. It is widely used in system and acceptance testing phases.

## 5.2 Validation

Validation is a crucial process that verifies whether inputs, systems, or products meet specified standards and perform as expected. It ensures data integrity, consistency, and correctness, especially in fields like software development and data management. In software, validation checks that applications fulfill user requirements and operate as intended. Data validation prevents errors by ensuring accurate formats, such as date or email inputs, before processing. Similarly, product validation in manufacturing confirms that each item meets quality and safety standards for consumers. In industries like pharmaceuticals, process validation is essential to guarantee consistent, reliable outcomes critical for safety. Overall, validation enhances confidence in system reliability, minimizes errors, and upholds high standards across various applications.

## Chapter 6

# SYSTEM SECURITY MEASURES

System security measures are essential to protect data, applications, and networks from unauthorized access, cyber threats, and data breaches.

### 6.0.1 Authentication and Access Control

Strong Password Policies – Require complex passwords and enforce periodic changes.

Multi-Factor Authentication (MFA) – Adds extra layers of security (e.g., OTP, biometrics).

Role-Based Access Control (RBAC) – Users are given permissions based on their roles.

### 6.0.2 Data Protection and Encryption

Data Encryption – Encrypt sensitive data during storage and transmission (e.g., AES, SSL/TLS).

Database Security – Use techniques like row-level security, access control, and backups.

Data Masking – Hide sensitive data from unauthorized users.

### 6.0.3 Network Security

Firewalls – Monitor and control incoming/outgoing traffic.

Intrusion Detection and Prevention Systems (IDPS) – Detect and block malicious activities.

Virtual Private Networks (VPNs) – Encrypt internet traffic for secure remote access.

### 6.0.4 Software and Application Security

Regular Updates and Patching – Keep software up to date to fix security vulnerabilities.

Secure Coding Practices – Follow OWASP guidelines to prevent SQL Injection, XSS, etc.

Web Application Firewall (WAF) – Protects against attacks like SQL injection and DDoS.

### **6.0.5 Backup and Disaster Recovery**

Automated Backups – Regularly backup data to prevent loss.

Disaster Recovery Plan (DRP) – Have a plan for restoring systems in case of failure.

Cloud Backup Solutions – Use secure cloud storage for redundancy.

### **6.0.6 Monitoring and Auditing**

Log Monitoring – Track and analyze logs for suspicious activities.

Security Information and Event Management (SIEM) – Automates security threat detection.

User Activity Auditing – Monitor user actions to detect insider threats.

### **6.0.7 Physical Security**

Access Control Systems – Restrict physical access to data centers and servers.

Surveillance Cameras – Monitor sensitive areas for unauthorized access.

Biometric Authentication – Use fingerprints or facial recognition for entry.

### **6.0.8 Security Awareness and Training**

Employee Training – Educate users about phishing, social engineering, and best practices.

Incident Response Plan – Ensure employees know how to handle security breaches.

## Chapter 7

# CONCLUSION

CashCraft effectively delivers a comprehensive solution for personal finance management by offering users an intuitive platform to track income, expenses, and savings goals. Through its key features such as budget creation, expense categorization, visual spending analysis, goal tracking, and overspending alerts, the system empowers users to make informed financial decisions and maintain better control over their monetary habits. The application provides a clean, modern, and user-friendly interface with responsive navigation and insightful dashboards, enhancing user experience and engagement. Built using Django's robust backend framework and secured with proper authentication mechanisms, CashCraft ensures data integrity, privacy, and scalability. It caters to a wide range of users looking for a structured and efficient approach to managing their financial well-being. Overall, CashCraft serves as a reliable and intelligent budget management tool that encourages users to build healthier financial habits and achieve their monetary objectives effectively.

## Chapter 8

# SCOPE FOR FUTURE ENHANCEMENT

CashCraft holds great potential for future enhancements that can further enrich the user experience and functionality. Planned improvements include the integration of AI-driven budgeting suggestions, automatic expense categorization using machine learning, and personalized financial insights based on user behavior. Support for multi-currency transactions and integration with banking APIs will provide real-time financial data synchronization, improving accuracy and usability. Additional features such as mobile push notifications for bill reminders, investment tracking, collaborative budgeting for families or roommates, and dark mode UI can offer more versatility and personalization. The inclusion of gamification elements like rewards for achieving savings goals and financial literacy tips can increase motivation and user retention. With these advancements, CashCraft can evolve into a powerful, all-in-one personal finance ecosystem supporting long-term financial wellness.

## **APPENDICES**

## Appendix A

# SCRUM PROCESS ARTIFACTS

### A.1 Product Backlog

Priority	Product Backlog Items	User Story	Estimate (Hours)
1	Database and User Authentication Setup	As a developer, I want to securely store user data and manage user login/registration.	200
2	Dashboard Overview Page	As a user, I want to view my total income, expenses, and balance on a dashboard.	120
3	Expense Tracking and Categorization	As a user, I want to add and categorize my expenses to track where my money goes.	150
4	Budget Creation and Tracking	As a user, I want to create monthly budgets and monitor how much I've spent per category.	130
5	Goal Management Module	As a user, I want to create savings goals and track my progress towards achieving them.	100
6	Spending Analysis with Charts	As a user, I want to see graphical analysis of my spending habits for better financial decisions.	90
7	Alerts for Over-spending	As a user, I want to receive alerts when I exceed my budget to control overspending.	60
8	Motivational Quotes Carousel	As a user, I want to view motivational financial quotes to stay inspired while budgeting.	30
9	Responsive Sidebar Navigation	As a user, I want easy access to all features through a clean and responsive sidebar.	40
10	Goal Deletion Confirmation Modal	As a user, I want to confirm before deleting a goal to prevent accidental loss of data.	20

### A.1.1 Functional Requirements

Functional requirements define the specific tasks, features, and operation that a system must perform to meet user needs. They describe what the system should do, including user interactions, data processing, and system behaviors. These requirements typically cover areas such as user authentication, data management, system workflows, reporting, notifications, and security measures. A well-defined set of functional requirements ensures that the system operates efficiently, meets business objectives, and provides a seamless user experience.

### A.1.2 Non Functional Requirements

Non-functional requirements define the quality attributes and constraints of a system to ensure efficiency, security, and usability. It must provide fast performance, with pages loading within 2 seconds and reports generating results quickly. Security measures such as encryption, authentication, and compliance with data protection regulations must be in place to protect user information. The system should be user-friendly with an intuitive interface, supporting both desktop and mobile devices. Scalability is essential, allowing the platform to handle increased users and data as it grows. Reliability and availability must be ensured with 99.9 percentage uptime and regular backups to prevent data loss. The system should be compatible across various devices and browsers, supporting seamless integrations with third-party APIs. Lastly, maintainability and accuracy should be prioritized, ensuring that data remains correct and the system can be easily updated and expanded in the future.

### A.1.3 System Configuration

System configuration refers to the hardware, software, and network settings required for a system to function efficiently. It includes the operating system, processor, memory, and storage capacity necessary to run applications smoothly. The software stack must be compatible with the system's requirements, including databases, programming languages, and web servers. Proper network configurations such as IP addresses, firewalls, and security protocols ensure safe and uninterrupted connectivity. Performance optimization involves adjusting system settings to enhance speed, stability, and resource utilization. Regular updates and maintenance help keep the system secure and efficient. Backup and recovery mechanisms are essential to prevent data loss in case of failures. A well-configured system ensures reliability, scalability, and security for seamless operations.

#### A.1.3.1 ER Diagram

The Entity-Relationship (ER) Diagram illustrates the major entities used in the system and the relationships among them.

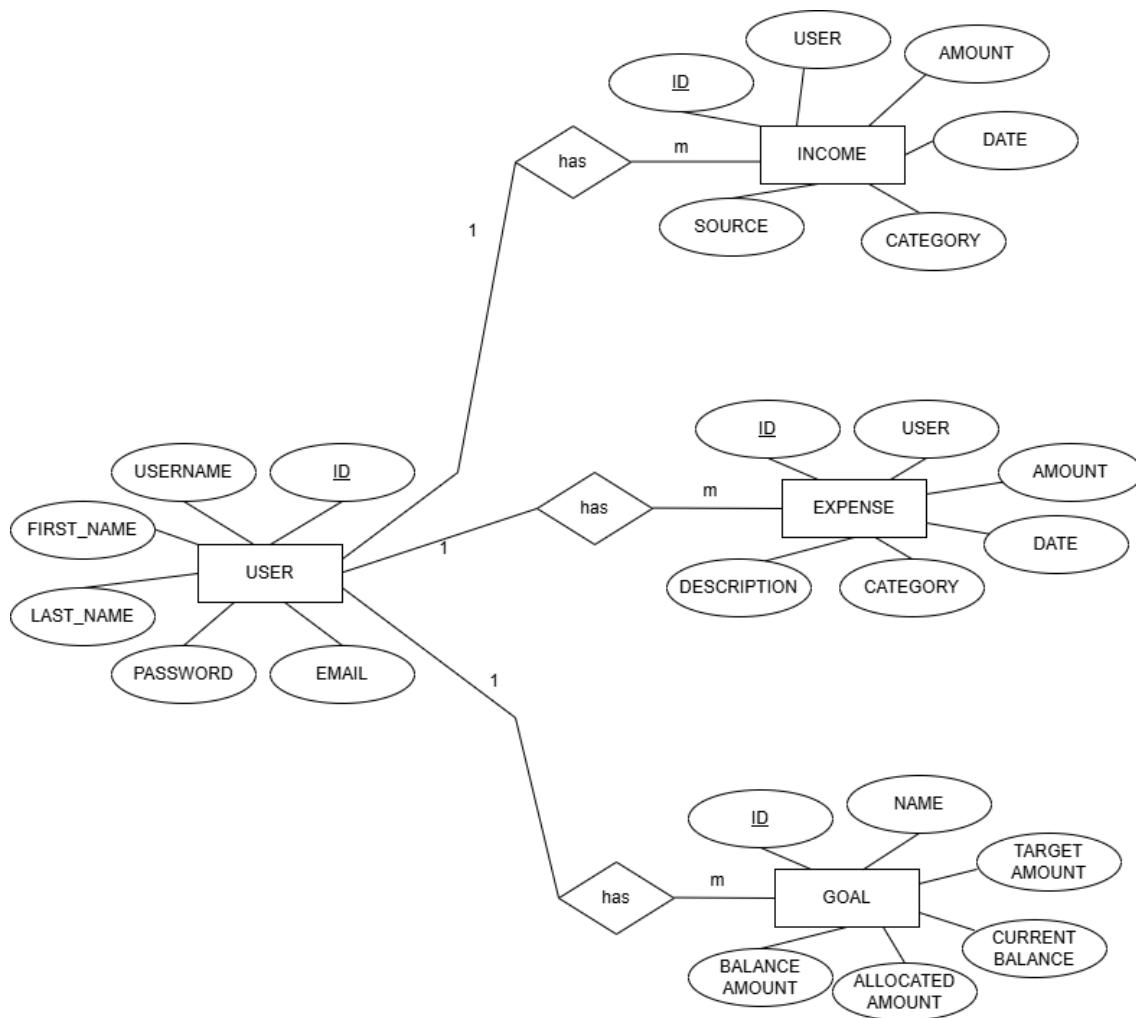


Figure A.1: Entity-Relationship (ER) Diagram of the System

**A.1.3.2 Hardware Configuration**

Processor	Intel Core i5
Hard Disk	40GB to 80GB
Memory	8GB
Display	Intel HD Graphics 620

**A.1.3.3 Software Configuration**

Component	Details
Operating System	Windows 10
Web Server	Django Development Server
Front End	HTML, JavaScript, CSS
Back End	SQLite
Programming Language	Python

## A.2 Sprint Details

Task Name	Description	Priority	Status	Durations (Days)
Sprint 1	Core Features Implementation	High	Completed	10
Task 1	User Registration	High	Completed	2
Task 2	User Login	High	Completed	1
Task 3	Dashboard Layout Design	Medium	Completed	2
Task 4	Income Management Module	High	Completed	2
Task 5	Expense Tracking Module	High	Completed	2
Task 6	Budget Creation	Medium	Completed	2
Task 7	Category-based Expense Visualization	Medium	Completed	1
Task 8	Budget Progress Indicator	Medium	Completed	1
Task 9	Savings Goal Module	High	Completed	2
Task 10	Goal Creation	Medium	Completed	1
Task 11	Goal Editing	Medium	Completed	1
Task 12	Goal Deletion Confirmation	Medium	Completed	1
Task 13	Alerts for Overspending	Medium	Completed	1
Task 14	Motivational Quote Carousel	Low	Completed	1
Task 15	Mobile-Friendly Sidebar Navigation	Medium	Completed	2

## A.3 Details of daily scrum meeting

Daily scrum meetings were held on each day of a sprint. The meetings were held in the office premises of the team at 9:30 am. These scrum meetings were strictly time-boxed to 15 minutes. This was to keep the discussion brisk but relevant.

All team members were required to attend scrum meetings. Since both the Scrum Master and

product owner are committed team members, they are expected to attend and participate. Anyone else was allowed to attend, but was there only to listen. This made scrum meetings an excellent way for a Scrum team to disseminate information.

The daily scrum meeting was not used as a problem-solving or issue resolution meeting. Issues that were raised were taken off-line and usually dealt with by the relevant subgroup immediately after the meeting. During the daily scrum, each team member answered the following three questions:

- What did you do yesterday?
- What will you do today?
- Are there any impediments in your way?

By focusing on what each person accomplished on a day and would accomplish the next day, the team would gain an excellent understanding of what work had been done and what work remained. The daily scrum meeting was not a status update meeting in which a boss was collecting information about who was behind schedule. Rather, it was a meeting in which team members made commitments to each other.

#### A.4 Details of bi-weekly meeting

Bi-weekly Meeting No.	Date	Details
Bi-weekly Meeting 1	19/12/2025	Project Proposal Submission
Bi-weekly Meeting 2	23/01/2025	First Review
Bi-weekly Meeting 3	06/03/2025	Second Review
Bi-weekly Meeting 4	01/04/2025	Third Review

## Appendix B

## VERSIONING

Sprint Number	Current Version Number
Sprint 1	V1.0
Sprint 2	V1.1
Sprint 3	V2.0
Sprint 4	V2.1
Sprint 5	V2.2

## Appendix C

# DATA FLOW DIAGRAMS

### DFD LEVEL 0

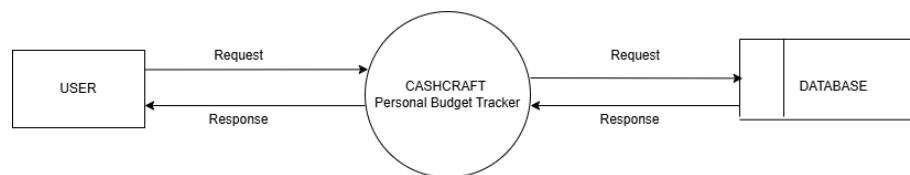


Figure C.1: DFD Level 0

### DFD LEVEL 1

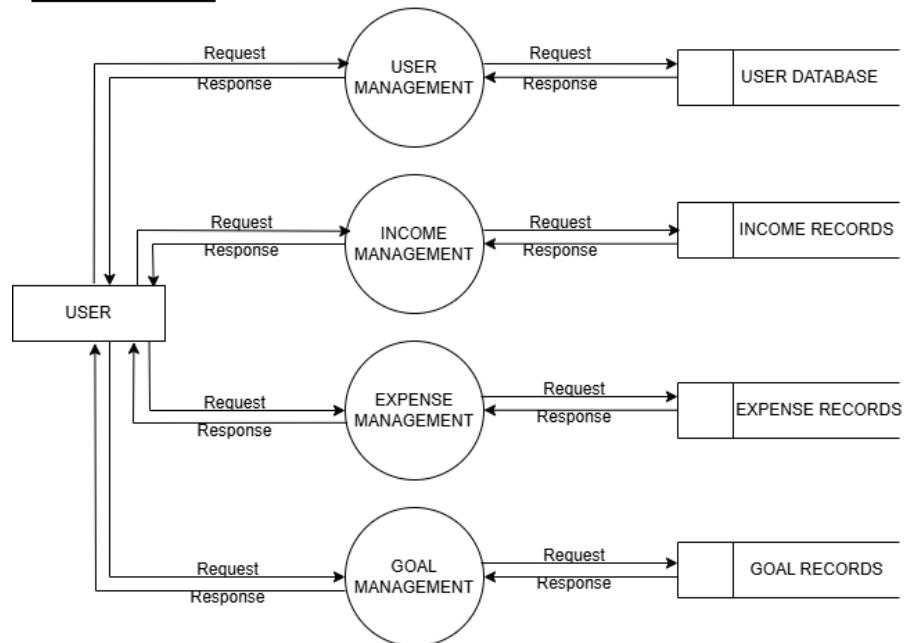


Figure C.2: DFD Level 1

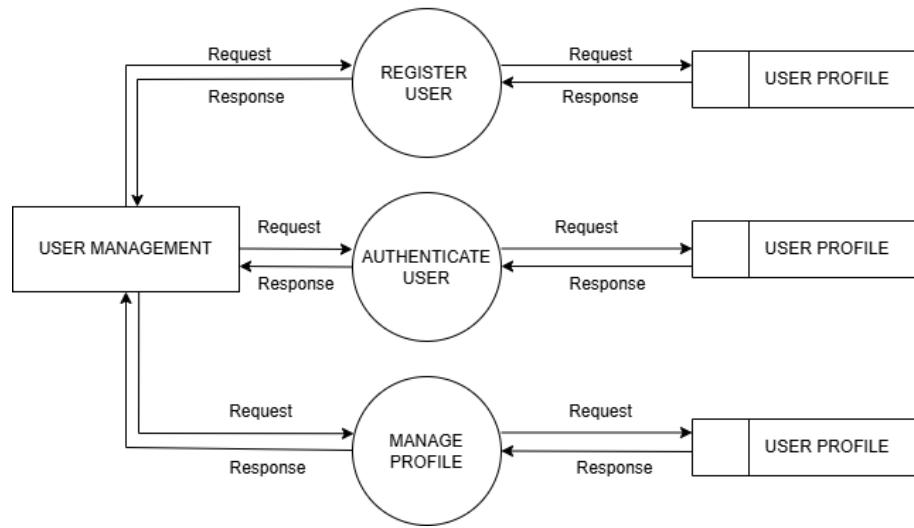
**DFD LEVEL 2.1**

Figure C.3: DFD Level 2.1

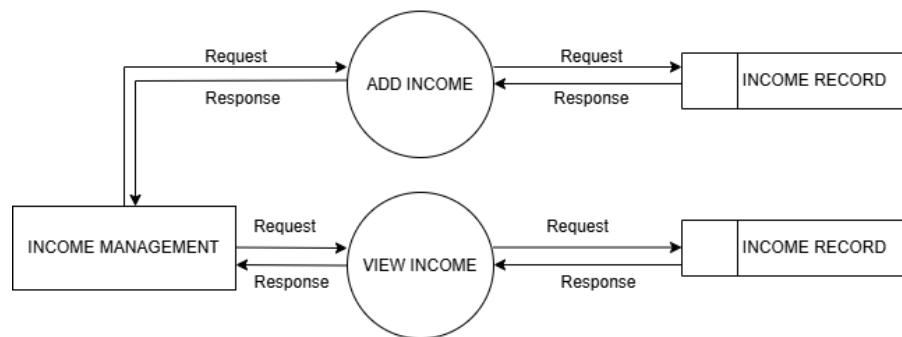
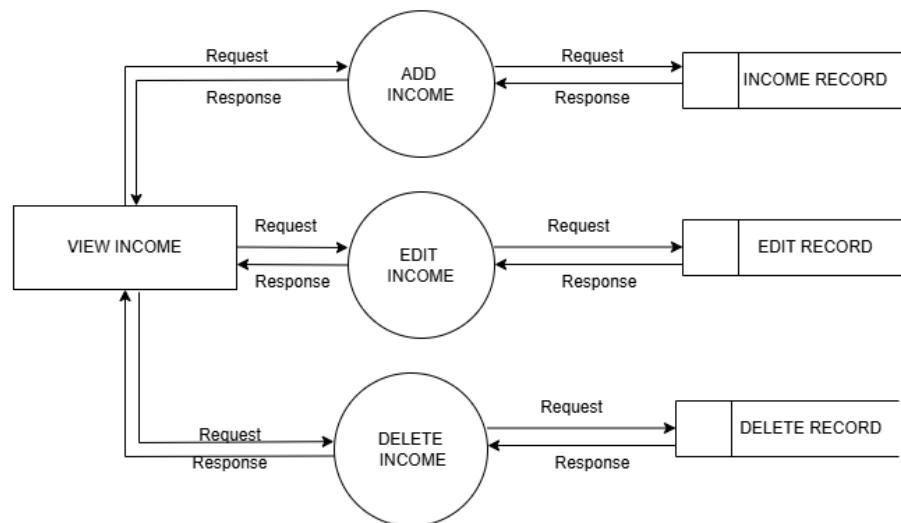
**DFD LEVEL 2.2****DFD LEVEL 2.2.1**

Figure C.4: DFD Level 2.2

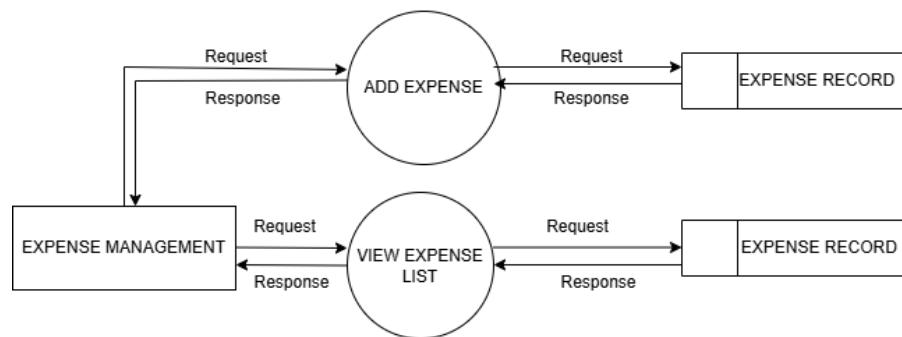
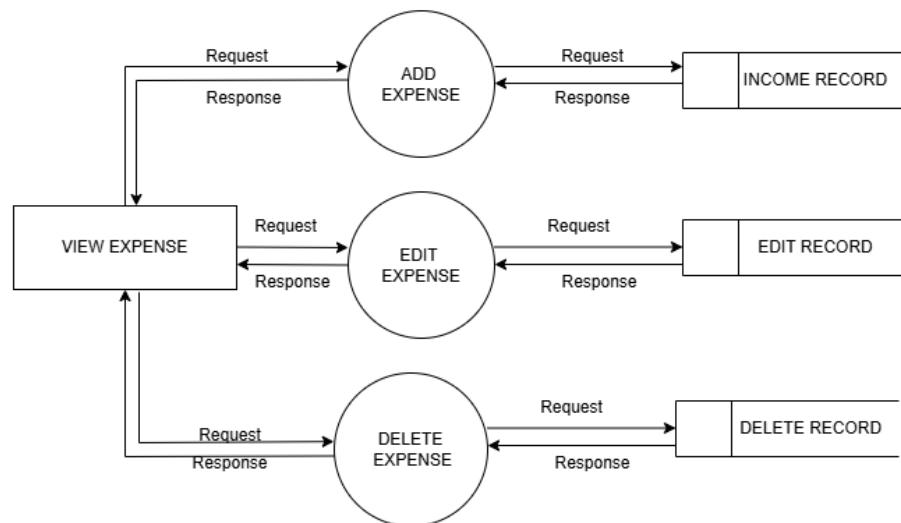
**DFD LEVEL 2.3****DFD LEVEL 2.3.1**

Figure C.5: DFD Level 2.3

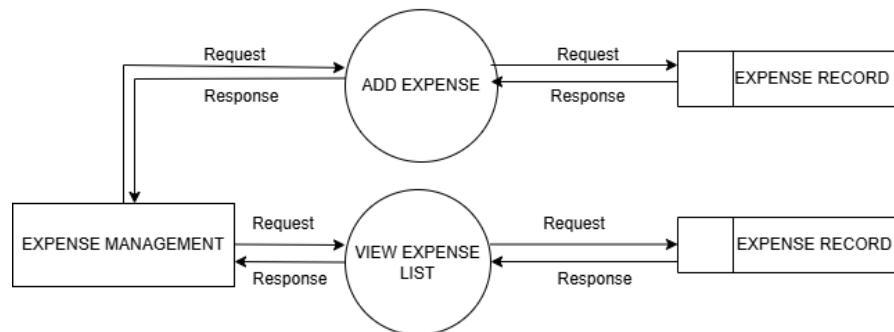
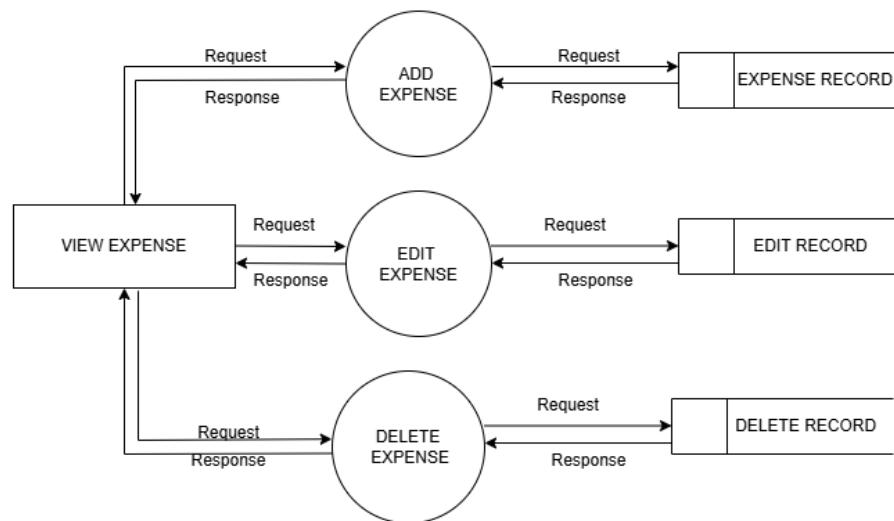
**DFD LEVEL 2.3****DFD LEVEL 2.3.1**

Figure C.6: DFD Level 2.4

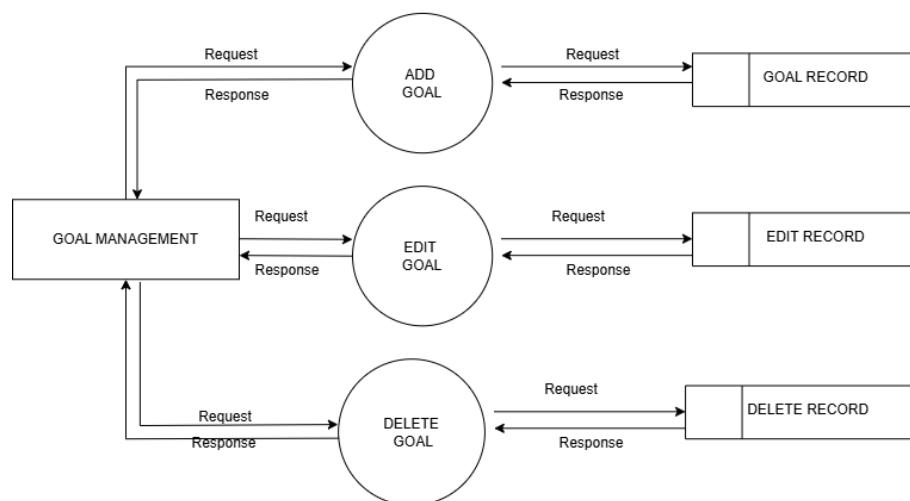
**DFD LEVEL 2.4**

Figure C.7: DFD Level 2.5

## Appendix D

### TABLE STRUCTURE

Sl. No.	Field	Datatype	Constraint	Description
1	id	Integer	Auto-increment, Primary Key	Unique identifier
2	username	varchar	unique, not null	Unique username
3	first_name	varchar	not null	User's first name
4	last_name	varchar	not null	User's last name
5	email	varchar	unique, not null	User's email address
6	password	varchar	not null	User password
7	is_staff	boolean	not null	Django admin site
8	is_active	boolean	not null	User account is active
9	is_superuser	boolean	not null	Grants user all permissions
10	last_login	DateTime	not null	User's last login
11	date_joined	DateTime	not null	User account was created

Table D.1: UserProfile Table Structure

Sl. No.	Field	Datatype	Constraint	Description
1	id	Integer	Auto-increment, Primary Key	Unique identifier for each income entry
2	user	ForeignKey (User)	Not null, On delete CASCADE	Links income to a specific user
3	amount	Decimal(10, 2)	Not null	Income amount with 2 decimal precision
4	date	Date	Not null	Date when the income was received
5	category	Varchar(100)	Default	Income category (e.g., Salary, Bonus)
6	source	Varchar(100)	Default	Source of income (e.g., Employer, Freelance)

Table D.2: Income Table Structure

Sl. No.	Field	Datatype	Constraint	Description
1	id	Integer	Auto-increment, Primary Key	Unique identifier for each expense entry
2	user	ForeignKey (User)	Nullable, On delete CASCADE	Links expense to a specific user
3	amount	Decimal(10, 2)	Not null	Expense amount with 2 decimal precision
4	category	Varchar(50)	Choices: Food, Transport, Shopping, Rent, Other	Type of expense
5	date	Date	Default: time-zone.now	Date of expense
6	description	Text	Optional (nullable)	Additional details about the expense

Table D.3: Expense Table Structure

Sl. No.	Field	Datatype	Constraint	Description
1	id	Integer	Auto-increment, Primary Key	Unique identifier for each goal
2	name	Varchar(255)	Not null	Name of the savings goal
3	target_amount	Decimal(10, 2)	Not null	Total amount to achieve the goal
4	current_balance	Decimal(10, 2)	Default: 0.00	Amount currently saved
5	allocated_amount	Decimal(10, 2)	Default: 0.00	Funds allocated toward the goal
6	remaining_amount	Property (Calculated)	Read-only	Target - (Current Balance + Allocated Amount)

Table D.4: Goal Table Structure

## Appendix E

## SAMPLE SCREENSHOTS

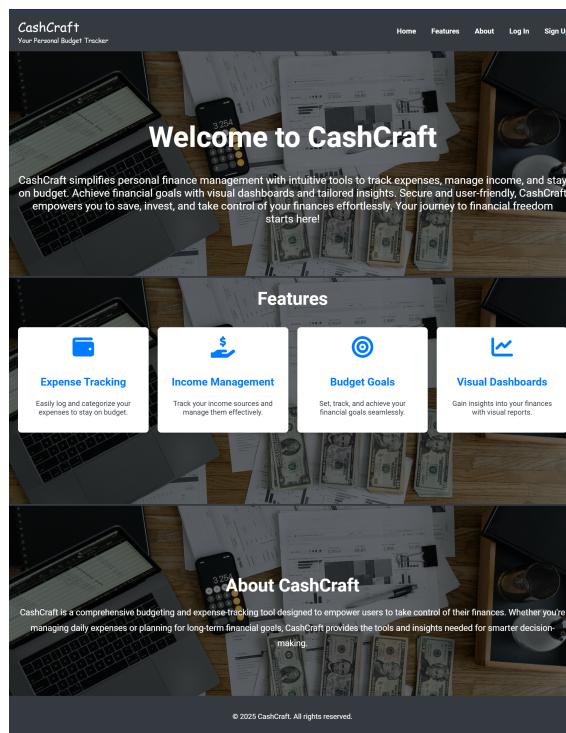


Figure E.1: Index Page

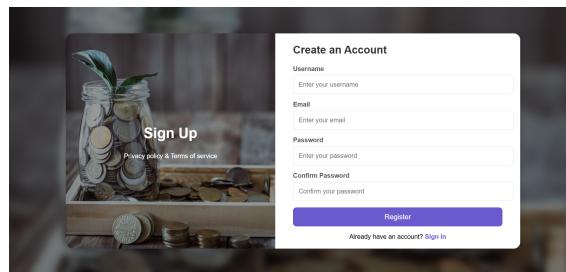


Figure E.2: Register Page

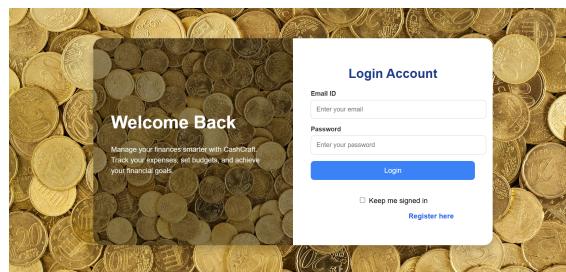


Figure E.3: Login Page

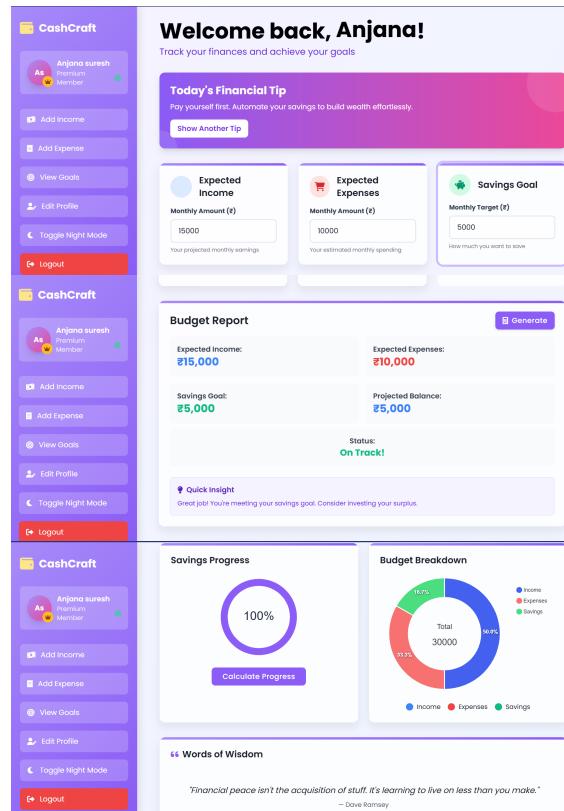


Figure E.4: User Dashboard page

The screenshot shows the 'Edit Your Profile' page. The title 'Edit Your Profile' is at the top, followed by the sub-instruction 'Shape your digital identity with excellence'. The form contains fields for 'First Name' (Anjana), 'Last Name' (suresh), and 'Email' (anjanasuresh1362002@gmail.com). A note above the first name field reads, 'Your personal brand is what people say about you when you're not in the room.' There is a 'Update Profile' button at the bottom.

Figure E.5: Edit Profile Page

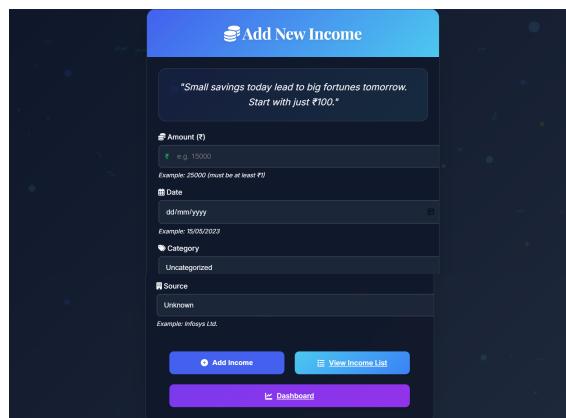


Figure E.6: Add Income Page

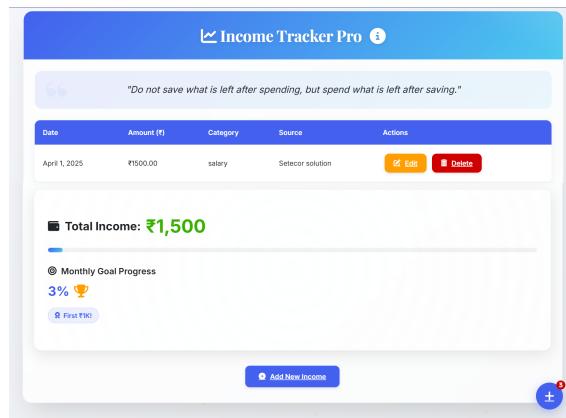


Figure E.7: View Income List page

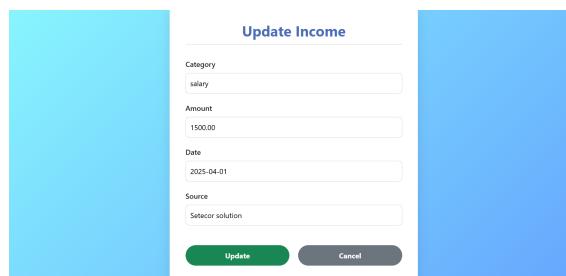


Figure E.8: Update Income Page

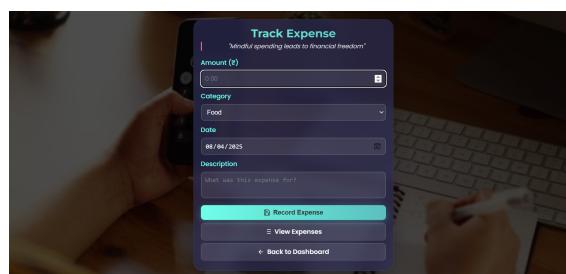


Figure E.9: Add Expense Page

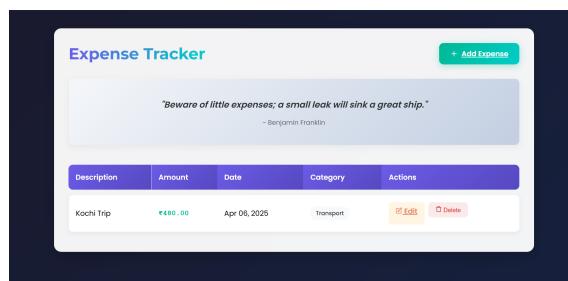


Figure E.10: View Expense Expense Page

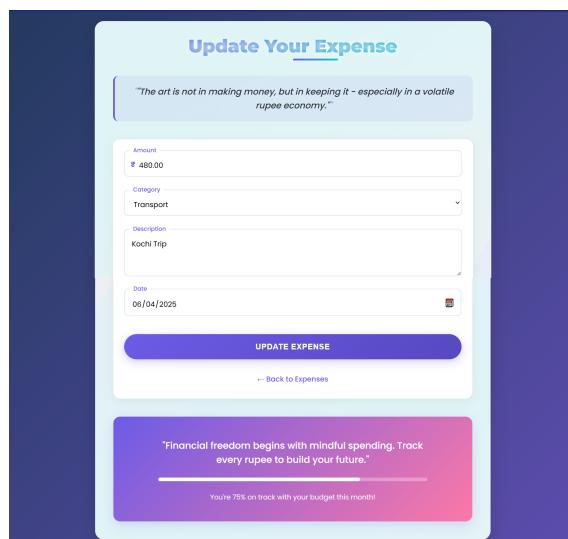


Figure E.11: Update Expense Page

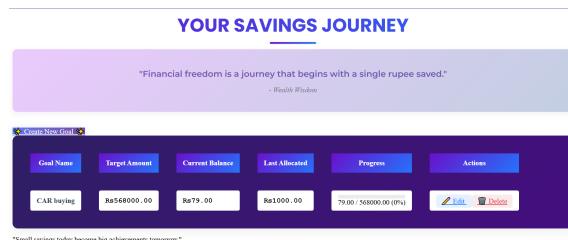


Figure E.12: Goal Page

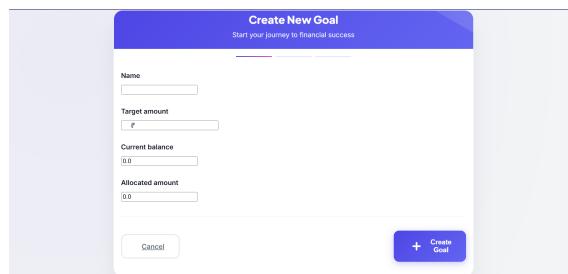


Figure E.13: Create New Goal Page

## Appendix F

# REPORTS

### F.0.1 Login Page

```
{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CashCraft - Login</title>
</head>
<body>
    <div class="container">
        <!-- Welcome Section -->
        <div class="welcome-section">
            <h1>Welcome Back</h1>
            <p>Manage your finances smarter with CashCraft. Track your expenses
               , set budgets, and achieve your financial goals.</p>
        </div>

        <!-- Form Section -->
        <div class="form-section">
            <h2>Login Account</h2>
            <form method="post" action="{% url 'login' %}">
                {% csrf_token %}
                <div class="form-group">
                    <label for="username">Email ID</label>
                    <input type="text" id="username" name="username"
                           placeholder="Enter your email" required>
                </div>
            </form>
        </div>
    </div>
</body>
```

```

<div class="form-group">
    <label for="password">Password</label>
    <input type="password" id="password" name="password"
           placeholder="Enter your password" required>
</div>

    <button type="submit">Login</button>
</form>
<div class="form-footer">
    <p>
        <input type="checkbox" id="keep_signed" name="keep_signed">
        <label for="keep_signed" style="color: black;">Keep me
            signed in</label>
    </p>
</div>
</div>
</div>
</body>
</html>

```

## F.0.2 Register Page

```

{%
    load static
}

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Sign Up</title>
    </head>
    <body>
        <div class="container">
            <!-- Left Panel -->
            <div class="left-panel">
                <h1>Sign Up</h1>
                <p>Privacy policy & Terms of service</p>
            </div>

            <!-- Right Panel (Form) -->
            <div class="form-container">
                <h2>Create an Account</h2>
                <form method="post" action="{% url 'register' %}">
                    {% csrf_token %}
                    <label for="username">Username</label>

```

```
<input type="text" id="username" name="username" placeholder="Enter your username" required>

<label for="email">Email</label>
<input type="email" id="email" name="email" placeholder="Enter your email" required>

<label for="password1">Password</label>
<input type="password" id="password1" name="password1" placeholder="Enter your password" required>

<label for="password2">Confirm Password</label>
<input type="password" id="password2" name="password2" placeholder="Confirm your password" required>

<button type="submit">Register</button>
</form>
<p>Already have an account? <a href="{% url 'login' %}">Sign In</a>
</p>
</div>
</div>

<script>
document.addEventListener("DOMContentLoaded", () => {
    const form = document.querySelector("form");
    const username = document.getElementById("username");
    const email = document.getElementById("email");
    const password1 = document.getElementById("password1");
    const password2 = document.getElementById("password2");

    form.addEventListener("submit", (e) => {
        let isValid = true;
        let message = "";

        // Validate username
        if (username.value.trim() === "") {
            isValid = false;
            message = "Username is required.";
        } else if (username.value.length < 3) {
            isValid = false;
            message = "Username must be at least 3 characters.";
        }

        // Validate email
        const emailPattern = /^[^@\s]+@[^\s]+\.\[^@\s]+$/;
```

```

        if (!emailPattern.test(email.value)) {
            isValid = false;
            message = "Please enter a valid email address.";
        }

        // Validate passwords
        if (password1.value.length < 8) {
            isValid = false;
            message = "Password must be at least 8 characters.";
        } else if (password1.value !== password2.value) {
            isValid = false;
            message = "Passwords do not match.";
        }

        if (!isValid) {
            e.preventDefault();
            alert(message);
        }
    });
});
});

</script>

</body>
</html>

```

### F.0.3 User Dashboard Page

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Personal Budget Tracker | Financial Freedom</title>
    <script src="https://cdn.jsdelivr.net/npm/apexcharts"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.9.1/gsap.min.js"
        ></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.4.0/jspdf.umd.
        min.js"></script>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
        awesome/6.0.0/css/all.min.css">
    <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.
        min.css" rel="stylesheet">
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300
        ;400;500;600;700&display=swap" rel="stylesheet">

```

```
</head>
<body class="bg-gradient-to-br from-purple-50 to-blue-50">
    <!-- Mobile Menu Button -->
    <div class="md:hidden fixed top-4 left-4 z-50">
        <button id="mobileMenuBtn" class="p-2 rounded-lg bg-white shadow-lg text-purple-600">
            <i class="fas fa-bars text-xl"></i>
        </button>
    </div>

    <!-- Sidebar -->
    <div class="sidebar" id="sidebar">
        <div class="flex items-center mb-8">
            <i class="fas fa-wallet text-3xl mr-3 text-yellow-200"></i>
            <h2 class="text-2xl font-bold">CashCraft</h2>
        </div>

        <div class="user-info mb-8 p-4 bg-white bg-opacity-20 rounded-lg">
            <div class="flex items-center">
                <div class="relative w-14 h-14 rounded-full bg-gradient-to-br from-purple-400 to-pink-500 flex items-center justify-center text-white font-bold mr-3 shadow-lg">
                    <!-- User initials -->
                    <span class="z-10">{{ user.first_name|first }}{{ user.last_name|first }}</span>
                <!-- Icon inside circle -->
                <div class="absolute -bottom-1 -right-1 w-6 h-6 rounded-full bg-yellow-400 flex items-center justify-center text-xs">
                    <i class="fas fa-crown text-yellow-800"></i>
                </div>
            </div>
            <div>
                <p class="font-semibold">{{ user.first_name }} {{ user.last_name }}</p>
                <p class="text-sm opacity-80 flex items-center">
                    <span>Premium Member</span>
                    <i class="fas fa-check-circle ml-2 text-green-400"></i>
                </p>
            </div>
        </div>
    </div>
    <div class="user-info mb-8 p-4 bg-white bg-opacity-20 rounded-lg">
        <div class="flex items-center">
```

```
<div class="relative w-14 h-14 rounded-full bg-gradient-to-br
  from-purple-400 to-pink-500 flex items-center justify-
  center text-white font-bold mr-3 shadow-lg">
  <!-- User initials -->
  <span class="z-10">{{ user.first_name|first }}{{ user.
    last_name|first }}</span>

  <!-- Icon inside circle -->
  <div class="absolute -bottom-1 -right-1 w-6 h-6 rounded-
    full bg-yellow-400 flex items-center justify-center
    text-xs">
    <i class="fas fa-crown text-yellow-800"></i>
  </div>
</div>
<div>
  <p class="font-semibold">{{ user.first_name }} {{ user.
    last_name }}</p>
  <p class="text-sm opacity-80 flex items-center">
    <span>Premium Member</span>
    <i class="fas fa-check-circle ml-2 text-green-400"></i>
  </p>
</div>
</div>

<nav>
  <button class="block w-full text-left mb-4 bg-white bg-opacity-20
    hover:bg-opacity-30 px-4 py-3 rounded-lg transition-all
    duration-300 flex items-center" onclick="window.location.href
    ='{{ url 'add_income' }}'>
    <i class="fas fa-money-bill-wave mr-3"></i> Add Income
  </button>
  <button class="block w-full text-left mb-4 bg-white bg-opacity-20
    hover:bg-opacity-30 px-4 py-3 rounded-lg transition-all
    duration-300 flex items-center" onclick="window.location.href
    ='{{ url 'add_expense' }}'>
    <i class="fas fa-receipt mr-3"></i> Add Expense
  </button>
  <button class="block w-full text-left mb-4 bg-white bg-opacity-20
    hover:bg-opacity-30 px-4 py-3 rounded-lg transition-all
    duration-300 flex items-center" onclick="window.location.href
    ='{{ url 'goals:goal_list' }}'>
    <i class="fas fa-bullseye mr-3"></i> View Goals
  </button>
  <button class="block w-full text-left mb-4 bg-white bg-opacity-20
```

```
        hover:bg-opacity-30 px-4 py-3 rounded-lg transition-all
        duration-300 flex items-center" onclick="window.location.href
        ='{{ url 'edit_profile' }}'">
        <i class="fas fa-user-edit mr-3"></i> Edit Profile
    </button>
    <button class="block w-full text-left mb-4 bg-white bg-opacity-20
        hover:bg-opacity-30 px-4 py-3 rounded-lg transition-all
        duration-300 flex items-center" onclick="toggleDarkMode()">
        <i class="fas fa-moon mr-3"></i> Toggle Night Mode
    </button>
    <form method="post" action="{{ url 'logout' }}">
        {% csrf_token %}
        <button type="submit" class="block w-full text-left bg-red-500
            hover:bg-red-600 px-4 py-3 rounded-lg transition-all
            duration-300 flex items-center">
            <i class="fas fa-sign-out-alt mr-3"></i> Logout
        </button>
    </form>

</nav>

<div class="mt-auto pt-6">
    <div class="quote-container text-sm italic opacity-80 mb-4" id="dailyQuote">
        "Financial freedom is available to those who learn about it and
        work for it."
    </div>
    <div class="text-xs opacity-60">
        Last updated: <span id="currentDate"></span>
    </div>
</div>
</div>

<!-- Main Content -->
<div class="main-content" id="mainContent">
    <!-- Header with Animation -->
    <div class="header mb-8">
        <h1 class="text-5xl font-bold text-black bg-clip-text bg-gradient-
            to-r from-purple-600 to-pink-500 mb-2">Welcome back, <span
            class="floating inline-block">{{ user.first_name }}</span>!</h1>
        <br>
        <p class="text-xl text-purple-700 dark:text-purple-300">Track your
            finances and achieve your goals</p>
    </div>
```

```
<!-- Motivational Banner -->
<div class="bg-gradient-to-r from-purple-500 to-pink-500 text-white p-6
rounded-xl mb-8 shadow-lg relative overflow-hidden">
<div class="absolute top-0 right-0 w-32 h-32 bg-white bg-opacity-10
rounded-full -mr-10 -mt-10"></div>
<div class="absolute bottom-0 left-0 w-20 h-20 bg-white bg-opacity
-10 rounded-full -ml-10 -mb-10"></div>
<h2 class="text-2xl font-bold mb-2 relative z-10">Today's Financial
Tip</h2>
<p class="relative z-10" id="financialTip">Pay yourself first.
Automate your savings to build wealth effortlessly.</p>
<button class="mt-4 bg-white text-purple-600 px-4 py-2 rounded-lg
font-semibold hover:bg-opacity-90 transition relative z-10"
onclick="showRandomTip() ">
Show Another Tip
</button>
</div>

<!-- Input Cards with Animated Icons -->
<div class="grid grid-cols-1 md:grid-cols-3 gap-6 mb-8">
<div class="card group">
<div class="p-6">
<div class="flex items-center mb-4">
<div class="w-12 h-12 rounded-full bg-blue-100 dark:bg-
blue-900 flex items-center justify-center text-blue
-600 dark:text-blue-300 mr-4 group-hover:rotate-12
transition-transform">
<i class="fas fa-money-bill-trend-up text-xl"></i>
</div>
<h3 class="text-xl font-semibold">Expected Income</h3>
</div>
<input type="number" id="expectedIncome" class="w-full px-4
py-3 rounded-lg input-field focus:outline-none"
placeholder="0.00" min="0">
<p class="text-xs text-gray-500 dark:text-gray-400 mt-2">
Your projected monthly earnings</p>
</div>
</div>

<div class="card group">
<div class="p-6">
<div class="flex items-center mb-4">
<div class="w-12 h-12 rounded-full bg-red-100 dark:bg-
red-900 flex items-center justify-center text-red
-600 dark:text-red-300 mr-4 group-hover:rotate-12
```

```

        transition-transform">
        <i class="fas fa-cart-shopping text-xl"></i>
    </div>
    <h3 class="text-xl font-semibold">Expected Expenses</h3>
    >
</div>

<input type="number" id="expectedExpenses" class="w-full px-4 py-3 rounded-lg input-field focus:outline-none" placeholder="0.00" min="0">
<p class="text-xs text-gray-500 dark:text-gray-400 mt-2">
    Your estimated monthly spending</p>
</div>
</div>

<div class="card group pulse">
    <div class="p-6">
        <div class="flex items-center mb-4">
            <div class="w-12 h-12 rounded-full bg-green-100 dark:bg-green-900 flex items-center justify-center text-green-600 dark:text-green-300 mr-4 group-hover:rotate-12 transition-transform">
                <i class="fas fa-piggy-bank text-xl"></i>
            </div>
            <h3 class="text-xl font-semibold">Savings Goal</h3>
        </div>
        <input type="number" id="savingsGoal" class="w-full px-4 py-3 rounded-lg input-field focus:outline-none" placeholder="0.00" min="0">
        <p class="text-xs text-gray-500 dark:text-gray-400 mt-2">
            How much you want to save</p>
    </div>
</div>
</div>

<!-- Report Section -->
<div class="card mb-8 transform transition-all hover:shadow-xl">
    <div class="p-6">
        <div class="flex justify-between items-center mb-4">
            <h3 class="text-2xl font-semibold">Budget Report</h3>
            <div class="flex space-x-2">
                <button class="btn bg-purple-500 hover:bg-purple-600 text-white px-4 py-2 rounded-lg flex items-center" onclick="generateReport()">

```

```

        <i class="fas fa-calculator mr-2"></i> Generate
    </button>

    </div>
</div>

<div id="report" class="text-gray-700 dark:text-gray-300 space-y-3">
    <div class="flex justify-between items-center p-4 bg-gray-50 dark:bg-gray-700 rounded-lg">
        <span>Enter your financial details and generate a report</span>
        <i class="fas fa-arrow-right text-purple-500"></i>
    </div>
</div>

<div class="mt-6 bg-purple-50 dark:bg-purple-900 dark:opacity-30 p-4 rounded-lg border border-purple-200 dark:border-purple-800">
    <h4 class="font-semibold text-purple-800 dark:text-purple-200 mb-2 flex items-center">
        <i class="fas fa-lightbulb mr-2"></i> Quick Insight
    </h4>
    <p class="text-sm text-purple-700 dark:text-purple-300" id="quickInsight">
        A budget is telling your money where to go instead of wondering where it went.
    </p>
</div>
</div>
</div>

<!-- Visual Progress Section -->
<div class="grid grid-cols-1 md:grid-cols-2 gap-6 mb-8">
    <div class="card">
        <div class="p-6">
            <h3 class="text-xl font-semibold mb-4">Savings Progress</h3>
            >
            <div class="flex items-center justify-center">
                <svg class="w-48 h-48" viewBox="0 0 100 100">
                    <circle class="text-gray-200 dark:text-gray-700 stroke-width="8" stroke="currentColor" fill="transparent" r="40" cx="50" cy="50" />
                    <circle class="progress-ring__circle text-purple-500" r="40" cx="50" cy="50" />
                </svg>
            </div>
        </div>
    </div>
</div>

```

```
-500" stroke-width="8" stroke-linecap="round"
      stroke="currentColor" fill="transparent" r="40"
      cx="50" cy="50" id="savingsProgress" />
    <text x="50" y="50" font-family="Arial" font-size="14" text-anchor="middle" alignment-baseline="middle" class="fill-current text-gray-800 dark:text-gray-200" id="savingsPercent">0%</text>
  </svg>
</div>
<div class="text-center mt-4">
  <button class="btn bg-purple-500 hover:bg-purple-600 text-white px-6 py-2 rounded-lg" onclick="calculateProgress()">
    Calculate Progress
  </button>
</div>
</div>
</div>

<div class="card">
  <div class="p-6">
    <h3 class="text-xl font-semibold mb-4">Budget Breakdown</h3>
    <div id="chart" class="min-h-[200px]"></div>
    <div class="flex justify-center space-x-4 mt-4">
      <div class="flex items-center">
        <div class="w-4 h-4 bg-blue-500 rounded-full mr-2"></div>
        <span class="text-sm">Income</span>
      </div>
      <div class="flex items-center">
        <div class="w-4 h-4 bg-red-500 rounded-full mr-2"></div>
        <span class="text-sm">Expenses</span>
      </div>
      <div class="flex items-center">
        <div class="w-4 h-4 bg-green-500 rounded-full mr-2"></div>
        <span class="text-sm">Savings</span>
      </div>
    </div>
  </div>
</div>
</div>
```

```
<!-- Motivational Quotes Carousel -->
<div class="card">
  <div class="p-6">
    <h3 class="text-xl font-semibold mb-4 flex items-center">
      <i class="fas fa-quote-left text-purple-500 mr-2"></i>
      Words of Wisdom
    </h3>
    <div class="relative h-32 overflow-hidden">
      <div class="absolute inset-0 flex items-center justify-
          center transition-all duration-500 opacity-0" id="
          quote1">
        <blockquote class="text-center">
          <p class="text-lg italic">"Do not save what is left
              after spending, but spend what is left after
              saving."</p>
        </blockquote>
      </div>
      <div class="absolute inset-0 flex items-center justify-
          center transition-all duration-500 opacity-0" id="
          quote2">
        <blockquote class="text-center">
          <p class="text-lg italic">"Financial peace isn't
              the acquisition of stuff. It's learning to live
              on less than you make."</p>
        </blockquote>
      </div>
      <div class="absolute inset-0 flex items-center justify-
          center transition-all duration-500 opacity-0" id="
          quote3">
        <blockquote class="text-center">
          <p class="text-lg italic">"The habit of saving is
              itself an education; it fosters every virtue,
              teaches self-denial, cultivates the sense of
              order."</p>
        </blockquote>
      </div>
    </div>
    <div class="flex justify-center mt-4 space-x-2">
      <button class="w-3 h-3 rounded-full bg-purple-300 hover:bg-
          purple-500 transition" onclick="showQuote(1)"></button>
      <button class="w-3 h-3 rounded-full bg-purple-300 hover:bg-
          purple-500 transition" onclick="showQuote(2)"></button>
      <button class="w-3 h-3 rounded-full bg-purple-300 hover:bg-
```

```
        purple-500 transition" onclick="showQuote(3)"></button>
    </div>
</div>
</div>

<!-- Notification Toast -->
<div class="fixed bottom-4 right-4 transition-all transform translate-y-20
opacity-0" id="toast">
<div class="bg-green-500 text-white px-6 py-3 rounded-lg shadow-lg flex
items-center">
<i class="fas fa-check-circle mr-2"></i>
<span id="toastMessage">Report generated successfully!</span>
</div>
</div>

<script>
// Initialize GSAP animations
gsap.registerEffect({
    name: "fadeIn",
    effect: (targets, config) => {
        return gsap.from(targets, {
            opacity: 0,
            y: 20,
            duration: 0.8,
            ease: "power2.out",
            stagger: 0.1
        });
    },
    defaults: { duration: 1 }
});

// Set current date
document.getElementById('currentDate').textContent = new Date().
    toLocaleDateString('en-US', {
        weekday: 'long',
        year: 'numeric',
        month: 'long',
        day: 'numeric'
    });

// Mobile menu toggle
const mobileMenuBtn = document.getElementById('mobileMenuBtn');
const sidebar = document.getElementById('sidebar');
```

```
mobileMenuBtn.addEventListener('click', () => {
    sidebar.classList.toggle('sidebar-open');
});

// Dark mode toggle
function toggleDarkMode() {
    document.body.classList.toggle("dark-mode");
    localStorage.setItem('darkMode', document.body.classList.contains(
        'dark-mode'));

    // Animate the toggle
    gsap.fromTo("body",
        { backgroundColor: document.body.classList.contains('dark-mode'
            ') ? '#f8f4ff' : '#0f172a' },
        { backgroundColor: document.body.classList.contains('dark-mode
            ') ? '#0f172a' : '#f8f4ff', duration: 0.5 })
};

}

// Check for saved dark mode preference
if (localStorage.getItem('darkMode') === 'true') {
    document.body.classList.add('dark-mode');
}

// Financial tips array
const financialTips = [
    "Automate your savings to build wealth effortlessly.",
    "Track every expense for a month to identify spending patterns.",
    "The 50/30/20 rule: 50% needs, 30% wants, 20% savings.",
    "Invest in knowledge - the best investment you can make.",
    "Build an emergency fund covering 3-6 months of expenses.",
    "Pay off high-interest debt before focusing on investments.",
    "Review your budget monthly and adjust as needed.",
    "Small daily savings add up to significant amounts over time."
];

// Motivational quotes
const quotes = [
    "Financial freedom is freedom from fear.",
    "A penny saved is a penny earned.",
    "Wealth is not about having money, it's about having options.",
    "Budgeting has only one rule: Don't go over budget.",
    "The art is not in making money, but in keeping it.",
    "Every rupee you save today is a rupee you can invest tomorrow.",
    "Financial peace begins with a well-planned budget."
];
```

```
        "Small amounts saved daily become large amounts over time."
    ];

    // Show random financial tip
    function showRandomTip() {
        const randomTip = financialTips[Math.floor(Math.random() *
            financialTips.length)];
        document.getElementById('financialTip').textContent = randomTip;

        // Animate the change
        gsap.fromTo("#financialTip",
            { opacity: 0, y: 10 },
            { opacity: 1, y: 0, duration: 0.5 })
    };
}

// Show random daily quote
function showDailyQuote() {
    const randomQuote = quotes[Math.floor(Math.random() * quotes.length
        )];
    const quoteElement = document.getElementById('dailyQuote');

    // Animate out
    gsap.to(quoteElement, {
        opacity: 0,
        y: -10,
        duration: 0.3,
        onComplete: () => {
            quoteElement.textContent = `#${randomQuote}`;
            // Animate in
            gsap.to(quoteElement, {
                opacity: 0.8,
                y: 0,
                duration: 0.5
            });
        }
    });
}

// Show specific quote in carousel
let currentQuote = 1;
function showQuote(quoteNum) {
    currentQuote = quoteNum;
    for (let i = 1; i <= 3; i++) {
        const quote = document.getElementById(`quote${i}`);
    }
}
```

```
        if (i === quoteNum) {
            gsap.to(quote, { opacity: 1, duration: 0.5 });
        } else {
            gsap.to(quote, { opacity: 0, duration: 0.5 });
        }
    }

    // Auto-rotate quotes
    setInterval(() => {
        currentQuote = currentQuote % 3 + 1;
        showQuote(currentQuote);
    }, 8000);

    // Calculate savings progress
    function calculateProgress() {
        let income = parseFloat(document.getElementById("expectedIncome") .
            value) || 0;
        let expenses = parseFloat(document.getElementById("expectedExpenses
            ").value) || 0;
        let savingsGoal = parseFloat(document.getElementById("savingsGoal") .
            value) || 0;

        if (income === 0) {
            showToast("Please enter your expected income first");
            return;
        }

        let actualSavings = income - expenses;
        let progressPercentage = Math.min(Math.round((actualSavings /
            savingsGoal) * 100), 100);

        if (savingsGoal === 0) {
            progressPercentage = 0;
        }

        document.getElementById("savingsPercent").textContent = `${
            progressPercentage}%`;

        // Animate the progress circle
        const circle = document.getElementById("savingsProgress");
        const radius = circle.r.baseVal.value;
        const circumference = 2 * Math.PI * radius;
        const offset = circumference - (progressPercentage / 100) *
            circumference;
    }
}
```

```
circle.style.strokeDasharray = `${circumference} ${circumference}`;
circle.style.strokeDashoffset = circumference;

gsap.to(circle, {
    strokeDashoffset: offset,
    duration: 1.5,
    ease: "elastic.out(1, 0.5)"
});

// Update charts
updateChart(income, expenses, actualSavings);
updateTrendChart('monthly');
}

// Update donut chart data
function updateChart(income, expenses, savings) {
    const options = {
        series: [income, expenses, savings],
        chart: {
            type: 'donut',
            height: 300
        },
        labels: ['Income', 'Expenses', 'Savings'],
        colors: ['#4361ee', '#f87171', '#4ade80'],
        responsive: [
            {
                breakpoint: 480,
                options: {
                    chart: {
                        width: 200
                    },
                    legend: {
                        position: 'bottom'
                    }
                }
            ],
            plotOptions: {
                pie: {
                    donut: {
                        labels: {
                            show: true,
                            total: {
                                show: true,
                                label: 'Total',
                                formatter: function (w) {

```

```

        return w.globals.seriesTotals.reduce((a
            , b) => a + b, 0);
    }
}
}
}
}
};

const chart = new ApexCharts(document.querySelector("#chart"),
    options);
chart.render();
}

// Initialize trend chart
let trendChart;
function initializeTrendChart() {
    const options = {
        series: [
            {
                name: 'Income',
                data: [30000, 32000, 35000, 38000, 40000, 42000, 45000]
            }, {
                name: 'Expenses',
                data: [22000, 24000, 26000, 28000, 30000, 32000, 34000]
            }, {
                name: 'Savings',
                data: [8000, 8000, 9000, 10000, 10000, 10000, 11000]
            }],
        chart: {
            type: 'area',
            height: 350,
            stacked: false,
            animations: {
                enabled: true,
                easing: 'easeinout',
                speed: 800
            },
            toolbar: {
                show: true
            }
        },
        colors: ['#4361ee', '#f87171', '#4ade80'],
        dataLabels: {
            enabled: false
        }
    }
}
```

```
        },
        stroke: {
            curve: 'smooth',
            width: 2
        },
        fill: {
            type: 'gradient',
            gradient: {
                opacityFrom: 0.6,
                opacityTo: 0.8,
            }
        },
        legend: {
            position: 'top',
            horizontalAlign: 'center'
        },
        xaxis: {
            categories: ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul'],
            labels: {
                style: {
                    colors: document.body.classList.contains('dark-mode') ? '#e2e8f0' : '#1a202c'
                }
            }
        },
        yaxis: {
            labels: {
                style: {
                    colors: document.body.classList.contains('dark-mode') ? '#e2e8f0' : '#1a202c'
                }
            }
        },
        tooltip: {
            y: {
            }
        }
    };
}

trendChart = new ApexCharts(document.querySelector("#trendChart"),
    options);
trendChart.render();
```

```
}

// Update trend chart based on time period
function updateTrendChart(period) {
    let categories, incomeData, expensesData, savingsData;

    switch(period) {
        case 'monthly':
            categories = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul'];
            incomeData = [30000, 32000, 35000, 38000, 40000, 42000, 45000];
            expensesData = [22000, 24000, 26000, 28000, 30000, 32000, 34000];
            savingsData = [8000, 8000, 9000, 10000, 10000, 10000, 11000];
            break;
        case 'quarterly':
            categories = ['Q1', 'Q2', 'Q3', 'Q4'];
            incomeData = [32000, 38000, 42000, 48000];
            expensesData = [24000, 28000, 32000, 36000];
            savingsData = [8000, 10000, 10000, 12000];
            break;
        case 'yearly':
            categories = ['2020', '2021', '2022', '2023'];
            incomeData = [350000, 420000, 480000, 520000];
            expensesData = [300000, 340000, 380000, 400000];
            savingsData = [50000, 80000, 100000, 120000];
            break;
    }

    trendChart.updateOptions({
        xaxis: {
            categories: categories
        },
        series: [
            {
                name: 'Income',
                data: incomeData
            }, {
                name: 'Expenses',
                data: expensesData
            }, {
                name: 'Savings',
                data: savingsData
            }
        ]
    })
}
```

```
        });
    }

    // Generate report
    function generateReport() {
        let income = parseFloat(document.getElementById("expectedIncome").
            value) || 0;
        let expenses = parseFloat(document.getElementById("expectedExpenses
            ").value) || 0;
        let savingsGoal = parseFloat(document.getElementById("savingsGoal")
            .value) || 0;
        let balance = income - expenses;
        let status = balance >= savingsGoal ? "On Track!" : "Needs
            Improvement!";

        // Determine status color
        let statusColor = balance >= savingsGoal ? "text-green-500" : "text
            -red-500";

        // Create report HTML
        document.getElementById("report").innerHTML =
            <div class="grid grid-cols-1 md:grid-cols-2 gap-4">
                <div class="bg-gray-50 dark:bg-gray-700 p-4 rounded-lg">
                    <p class="font-semibold">Expected Income:</p>
                </div>
                <div class="bg-gray-50 dark:bg-gray-700 p-4 rounded-lg">
                    <p class="font-semibold">Expected Expenses:</p>
                </div>
                <div class="bg-gray-50 dark:bg-gray-700 p-4 rounded-lg">
                    <p class="font-semibold">Savings Goal:</p>
                </div>
                <div class="bg-gray-50 dark:bg-gray-700 p-4 rounded-lg">
                    <p class="font-semibold">Projected Balance:</p>
                </div>
            </div>
            <div class="mt-4 p-4 bg-gray-50 dark:bg-gray-700 rounded-lg
                text-center">
                <p class="font-semibold">Status:</p>
                <p class="text-xl ${statusColor} font-bold">${status}</p>
            </div>
    ';

    // Generate quick insight
    let insight = "";
    if (balance < 0) {
```

```
        insight = "You're spending more than you earn. Consider  
        reducing expenses or increasing income.";  
    } else if (balance >= savingsGoal) {  
        insight = "Great job! You're meeting your savings goal.  
        Consider investing your surplus.";  
    } else if (balance > 0) {  
        insight = "You're saving money but not reaching your goal. Look  
        for areas to cut expenses.";  
    } else {  
        insight = "Every financial journey begins with awareness. Keep  
        tracking to improve your finances.";  
    }  
  
    document.getElementById("quickInsight").textContent = insight;  
  
    // Show toast notification  
    showToast("Report generated successfully!");  
  
    // Create confetti effect if on track  
    if (balance >= savingsGoal && savingsGoal > 0) {  
        createConfetti();  
    }  
  
    // Calculate progress  
    calculateProgress();  
}  
  
// Export PDF  
function exportPDF() {  
    const { jsPDF } = window.jspdf;  
    let doc = new jsPDF();  
  
    // Add title  
    doc.setFontSize(20);  
    doc.setTextColor(99, 102, 241);  
    doc.text("Personal Budget Report", 105, 20, null, null, 'center');  
  
    // Add date  
    doc.setFontSize(12);  
    doc.setTextColor(100, 100, 100);  
    doc.text(`Generated on: ${new Date().toLocaleDateString()}`, 105,  
    30, null, null, 'center');  
  
    // Add user info  
    doc.setFontSize(14);
```

```
doc.setTextColor(0, 0, 0);
doc.text('User: {{ user.first_name }} {{ user.last_name }}', 20,
40);

// Add report content
const reportText = document.getElementById("report").innerText;
const splitText = doc.splitTextToSize(reportText, 170);
doc.text(splitText, 20, 50);

// Add quick insight
doc.setFontSize(12);
doc.setTextColor(99, 102, 241);
doc.text("Quick Insight:", 20, doc.autoTable.previous.finalY + 10);
doc.setTextColor(0, 0, 0);
doc.text(document.getElementById("quickInsight").innerText, 20, doc
.autoTable.previous.finalY + 20);

// Save the PDF
doc.save("budget_report.pdf");

// Show toast notification
showToast("PDF exported successfully!");
}

// Show toast notification
function showToast(message) {
    const toast = document.getElementById("toast");
    document.getElementById("toastMessage").textContent = message;

    // Animate in
    gsap.to(toast, {
        y: 0,
        opacity: 1,
        duration: 0.5,
        ease: "power2.out",
        onComplete: () => {
            // Animate out after delay
            setTimeout(() => {
                gsap.to(toast, {
                    y: 20,
                    opacity: 0,
                    duration: 0.5,
                    ease: "power2.in"
                });
            }, 3000);
        }
    });
}
```

```
        }

    });

}

// Create confetti effect
function createConfetti() {
    const colors = ['#a78bfa', '#8b5cf6', '#c4b5fd', '#a855f7', '#d8b4fe'];

    for (let i = 0; i < 100; i++) {
        const confetti = document.createElement('div');
        confetti.className = 'confetti';
        confetti.style.backgroundColor = colors[Math.floor(Math.random()
            () * colors.length)];
        confetti.style.left = Math.random() * 100 + 'vw';
        confetti.style.borderRadius = Math.random() > 0.5 ? '50%' :
            '0';
        confetti.style.transform = `rotate(${Math.random() * 360}deg)`;
        document.body.appendChild(confetti);
    }
}

// Animate confetti
gsap.to(confetti, {
    y: Math.random() * -500 - 100,
    x: Math.random() * 200 - 100,
    opacity: 1,
    duration: 1.5,
    ease: "power1.out",
    onComplete: () => {
        confetti.remove();
    }
});

}

// Initialize animations on load
document.addEventListener('DOMContentLoaded', () => {
    // Animate sidebar items
    gsap.from(".sidebar button", {
        opacity: 0,
        x: -20,
        duration: 0.6,
        stagger: 0.1,
        delay: 0.3
    });
})
```

```
// Animate cards
gsap.from(".card", {
    opacity: 0,
    y: 30,
    duration: 0.8,
    stagger: 0.1,
    delay: 0.6,
    ease: "back.out(1.7)"
});

// Show first quote
showQuote(1);

// Show random daily quote
showDailyQuote();

// Rotate quotes every 8 seconds
setInterval(showDailyQuote, 24000);

// Add hover effects to buttons
document.querySelectorAll('.btn').forEach(btn => {
    btn.addEventListener('mouseenter', () => {
        gsap.to(btn, {
            y: -2,
            duration: 0.2,
            boxShadow: '0 5px 15px rgba(0,0,0,0.1)'
        });
    });
    btn.addEventListener('mouseleave', () => {
        gsap.to(btn, {
            y: 0,
            duration: 0.2,
            boxShadow: '0 2px 5px rgba(0,0,0,0.1)'
        });
    });
});

// Initialize trend chart
initializeTrendChart();
});

</script>
</body>
</html>
```

#### F.0.4 Add Income Page

```
{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Income Tracker Pro | Financial Dashboard</title>

    <!-- Fonts & Icons -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&family=Playfair+Display:wght@700&display=swap" rel="stylesheet">
</head>
<body>
    <!-- Floating Particles Background -->
    <div class="particles" id="particles"></div>

    <!-- Floating Rupee Symbols -->
    <div class="floating-rupees" id="rupeeContainer"></div>

    <!-- Main Container -->
    <div class="tracker-container">
        <!-- Header with Shine Effect -->
        <div class="tracker-header">
            <i class="fas fa-coins"></i> Add New Income
        </div>

        <!-- Motivational Section -->
        <div class="motivational-section">
            <p class="motivational-quote">
                "Financial freedom begins with awareness. Track every rupee to build your wealth."
            </p>
        </div>

        <!-- Income Form -->
        <form method="POST" id="incomeForm">
            {% csrf_token %}
            <div class="form-group">
                <div class="input-container">
                    <input type="number" name="amount" step="1" min="1" >
                </div>
            </div>
        </form>
    </div>
</body>
```

```

        required
        placeholder="e.g. 15000" id="amountInput">
    </div>
    </div>

<div class="form-group">
    <label for="date"><i class="far fa-calendar-alt"></i> Date</
        label>
    <input type="date" name="date" required
        placeholder="Select date" id="dateInput">
    <span class="input-hint">Example: 15/05/2023</span>
</div>

<div class="form-group">
    <label for="category"><i class="fas fa-tags"></i> Category</
        label>
    <input type="text" name="category" value="Uncategorized"
        placeholder="e.g. Salary, Freelance, Business" id="
            categoryInput">
    <span class="input-hint">Example: Freelance Project</span>
</div>

<div class="form-group">
    <label for="source"><i class="fas fa-building"></i> Source</
        label>
    <input type="text" name="source" value="Unknown"
        placeholder="e.g. Company Name, Client" id="sourceInput"
        >
</div>

<div class="button-group">
    <button type="submit" class="btn-primary" id="submitBtn">
        <i class="fas fa-plus-circle"></i> Add Income
    </button>
    <a href="{% url 'income_list' %}" class="action-btn btn-view">
        <i class="fas fa-list"></i> View Income List
    </a>
    <a href="{% url 'dashboard' %}" class="action-btn btn-dashboard
        ">
        <i class="fas fa-chart-line"></i> Dashboard
    </a>
</div>
</form>

</div>

```

```
<script>
    document.addEventListener("DOMContentLoaded", function() {
        // Create floating particles
        const particleContainer = document.getElementById('particles');
        const colors = ['var(--primary)', 'var(--accent)', 'var(--success)
            ', 'var(--warning)'];

        for (let i = 0; i < 30; i++) {
            const particle = document.createElement('div');
            particle.className = 'particle';
            particle.style.left = Math.random() * 100 + 'vw';
            particle.style.width = (Math.random() * 15 + 5) + 'px';
            particle.style.height = particle.style.width;
            particle.style.backgroundColor = colors[Math.floor(Math.random()
                () * colors.length)];
            particle.style.opacity = Math.random() * 0.2 + 0.05;
            particle.style.animationDuration = (Math.random() * 15 + 10) +
                's';
            particle.style.animationDelay = Math.random() * 10 + 's';
            particleContainer.appendChild(particle);
        }

        // Create floating rupee symbols
        const rupeeContainer = document.getElementById('rupeeContainer');
        for (let i = 0; i < 20; i++) {
            const rupee = document.createElement('div');
            rupee.className = 'rupee-symbol';
            rupee.style.left = Math.random() * 100 + 'vw';
            rupee.style.fontSize = (Math.random() * 20 + 10) + 'px';
            rupee.style.animationDelay = Math.random() * 15 + 's';
            rupee.style.animationDuration = (Math.random() * 10 + 10) + 's
                ';
            rupeeContainer.appendChild(rupee);
        }

        // Amount validation
        const amountInput = document.getElementById('amountInput');
        const amountError = document.getElementById('amountError');

        amountInput.addEventListener('input', function() {
            if (parseFloat(this.value) < 1 && this.value !== '') {
                this.classList.add('error');
                amountError.style.display = 'block';
            } else {

```

```

        this.classList.remove('error');
        amountError.style.display = 'none';
    }
});

// Form submission validation
const form = document.getElementById('incomeForm');
const submitBtn = document.getElementById('submitBtn');

form.addEventListener('submit', function(e) {
    // Validate amount
    if (parseFloat(amountInput.value) < 1) {
        e.preventDefault();
        amountInput.classList.add('error');
        amountError.style.display = 'block';
        return false;
    }

    // If validation passes, proceed with submission
    submitBtn.disabled = true;
    submitBtn.innerHTML = '<i class="fas fa-spinner fa-spin"></i>
    Processing...';

    // Create golden confetti for Indian theme
    createConfetti();

    // Simulate processing delay
    setTimeout(() => {
        submitBtn.innerHTML = '<i class="fas fa-check"></i> Success
        !';
    }, 1500);
});

function createConfetti() {
    const colors = ['var(--gold)', 'var(--rupee)', 'var(--primary)
    ', 'var(--accent)'];

    for (let i = 0; i < 100; i++) {
        const confetti = document.createElement('div');
        confetti.className = 'confetti';
        confetti.style.left = Math.random() * window.innerWidth +
            'px';
        confetti.style.top = -10 + 'px';
        confetti.style.backgroundColor = colors[Math.floor(Math.
            random() * colors.length)];
    }
}

```

```

        confetti.style.width = Math.random() * 8 + 5 + 'px';
        confetti.style.height = Math.random() * 8 + 5 + 'px';
        confetti.style.animationDuration = Math.random() * 3 + 2 +
            's';
        confetti.style.animationDelay = Math.random() * 0.5 + 's';
        document.body.appendChild(confetti);

        // Remove confetti after animation
        setTimeout(() => {
            confetti.remove();
        }, 3000);
    }

    // Random Indian financial quotes
    const quotes = [
        {
            text: "Do not save what is left after spending, but spend
                  what is left after saving.",
            author: "Warren Buffett"
        },
        // Rotate quotes every 8 seconds
    let quoteIndex = 0;
    const quoteElement = document.querySelector('.motivational-quote');
    const authorElement = document.querySelector('.quote-author');

    function changeQuote() {
        quoteElement.style.animation = 'fadeOut 0.5s forwards';
        authorElement.style.animation = 'fadeOut 0.5s forwards';

        setTimeout(() => {
            quoteIndex = (quoteIndex + 1) % quotes.length;
            quoteElement.textContent = `${quotes[quoteIndex].text}`;
        });

        // Add confetti style
        const style = document.createElement('style');
        style.textContent = `
            @keyframes fadeIn {
                from { opacity: 0; }
                to { opacity: 1; }
            }
            @keyframes fadeOut {
                from { opacity: 1; }
                to { opacity: 0; }
            }
        `;
    }
}

```

```

';
document.head.appendChild(style);
});
</script>
</body>
</html>

```

### F.0.5 View Income List Page

```

{%
    load static
}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Income Tracker Pro | Financial Dashboard</title>

    <!-- Fonts & Icons -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
    <link href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&family=Playfair+Display:wght@700&display=swap" rel="stylesheet">
</head>
<body>
    <!-- Floating Particles Background -->
    <div class="particles" id="particles"></div>

    <!-- Main Container -->
    <div class="tracker-container">
        <!-- Header with Shine Effect -->
        <div class="tracker-header">
            <i class="fas fa-chart-line"></i> Income Tracker Pro
            <div class="tooltip">
                <i class="fas fa-info-circle" style="margin-left: 10px;"></i>
                <span class="tooltiptext">Track your income and achieve financial freedom</span>
            </div>
        </div>
    </div>

    <!-- Motivational Section -->
    <div class="motivational-section">
        <p class="motivational-quote">
            "Financial freedom begins with awareness. Track every rupee to

```

```

        build your wealth."
    </p>
</div>

<!-- Income Table -->
<div class="tracker-table-container">
    <table class="tracker-table" id="incomeTable">
        <thead>
            <tr>
                <th>Date</th>
                <th>Category</th>
                <th>Source</th>
                <th>Actions</th>
            </tr>
        </thead>

        <tbody>
            {% for income in incomes %}
            <tr>
                <td>{{ income.date }}</td>
                <td>{{ income.category }}</td>
                <td>{{ income.source }}</td>
                <td>
                    <a href="{% url 'income_update' income.pk %}" class="btn btn-warning btn-custom">
                        <i class="fas fa-edit"></i> Edit
                    </a>
                    <a href="{% url 'income_delete' income.pk %}" class="btn btn-danger btn-custom">
                        <i class="fas fa-trash-alt"></i> Delete
                    </a>
                </td>
            </tr>
            {% empty %}
            <tr>
                <td colspan="6" class="empty-state">
                    <i class="fas fa-piggy-bank"></i>
                    <h4>No income records found</h4>
                    <p>Start tracking your income to see financial growth</p>
                </td>
            </tr>
            {% endfor %}
        </tbody>
    </table>
</div>

```

```

        </tbody>
    </table>
</div>

<!-- Summary Section with Glass Morphism -->
<div class="tracker-summary">
    <div class="progress-container">
        <div class="progress-bar" id="incomeProgress"></div>
    </div>

    <div class="goal-meter">
        <h4><i class="fas fa-bullseye"></i> Monthly Goal Progress</h4>
        <div class="meter-value">
            <span id="goalProgress">0%</span>
            <i class="fas fa-trophy" style="color: var(--warning);"></i>
        </div>
        <div class="achievement-badges" id="achievementBadges"></div>
    </div>
</div>

<!-- Call to Action -->
<div style="text-align:center; margin: 30px;">
    <a href="{% url 'add_income' %}" class="btn btn-primary btn-custom" style="padding: 15px 30px;">
        <i class="fas fa-plus-circle"></i> Add New Income
    </a>
</div>
</div>

<!-- Floating Action Button -->
<a href="{% url 'add_income' %}" class="fab">
    <i class="fas fa-plus"></i>
    <span class="notification-badge" id="reminderBadge">3</span>
</a>

<!-- Chart.js -->
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<script>
    // Create floating particles
    function createParticles() {
        const container = document.getElementById('particles');
        const colors = ['var(--primary)', 'var(--accent)', 'var(--success)', 'var(--warning)'];

```

```

        for (let i = 0; i < 30; i++) {
            const particle = document.createElement('div');
            particle.className = 'particle';
            particle.style.left = Math.random() * 100 + 'vw';
            particle.style.width = (Math.random() * 15 + 5) + 'px';
            particle.style.height = particle.style.width;
            particle.style.backgroundColor = colors[Math.floor(Math.random()
                () * colors.length)];
            particle.style.opacity = Math.random() * 0.2 + 0.05;
            particle.style.animationDuration = (Math.random() * 15 + 10) +
                's';
            particle.style.animationDelay = Math.random() * 10 + 's';
            container.appendChild(particle);
        }
    }

    // Calculate total income with animation
    function calculateTotalIncome() {
        let total = 0;
        document.querySelectorAll("#incomeTable tbody tr td:nth-child(2)").forEach(cell => {

            // Animate the total income counter
            const totalElement = document.getElementById('totalIncome');
            const duration = 1500;
            const start = 0;
            const increment = total / (duration / 16);

            let current = start;
            const timer = setInterval(() => {
                current += increment;
                if (current >= total) {
                    clearInterval(timer);
                    current = total;
                }
            }
            // Update progress bar
            const progressBar = document.getElementById('incomeProgress');
            const progressPercentage = Math.min(100, (total / 50000) * 100); //
                Assuming 50,000 is max
            progressBar.style.width = `${progressPercentage}%`;

            // Update goal progress
            document.getElementById('goalProgress').textContent = `${Math.round
                (progressPercentage)}%`;
        });
    }
}

```

```
// Check for achievements
checkAchievements(total);

// Show confetti if goal is reached
if (progressPercentage >= 100) {
    createConfetti();
    showCelebration();
}

// Check for achievements
function checkAchievements(total) {
    const badgesContainer = document.getElementById('achievementBadges');
    badgesContainer.innerHTML = '';

    const achievements = [
        // Create confetti celebration
        function createConfetti() {
            const colors = ['var(--primary)', 'var(--accent)', 'var(--success)', 'var(--warning)', 'var(--danger)'];

            for (let i = 0; i < 150; i++) {
                const confetti = document.createElement('div');
                confetti.className = 'confetti';
                confetti.style.left = Math.random() * window.innerWidth + 'px';
                confetti.style.top = -10 + 'px';
                confetti.style.backgroundColor = colors[Math.floor(Math.random() * colors.length)];
                confetti.style.width = Math.random() * 10 + 5 + 'px';
                confetti.style.height = Math.random() * 10 + 5 + 'px';
                confetti.style.animationDuration = Math.random() * 3 + 2 + 's';
                confetti.style.animationDelay = Math.random() * 0.5 + 's';
                document.body.appendChild(confetti);
            }

            // Remove confetti after animation
            setTimeout(() => {
                confetti.remove();
            }, 3000);
        }
    ];
}

// Show celebration message
```

```
function showCelebration() {
    const celebration = document.createElement('div');
    celebration.className = 'motivational-section';
    celebration.style.backgroundColor = 'rgba(56, 176, 0, 0.2)';
    celebration.style.borderColor = 'var(--success)';
    celebration.innerHTML =
        `

Congratulations! You've reached your monthly income goal!

`;
}

document.querySelector('.tracker-container').insertBefore(
    celebration, document.querySelector('.tracker-summary'));

// Pulse animation for celebration
celebration.style.animation = 'pulse 2s infinite';

// Remove after 5 seconds
setTimeout(() => {
    celebration.style.opacity = '0';
    celebration.style.transition = 'opacity 1s ease';
    setTimeout(() => {
        celebration.remove();
    }, 1000);
}, 5000);
}

// Rotate motivational quotes
function rotateQuotes() {
    const quotes = [
        {
            text: "Financial peace isn't the acquisition of stuff. It's
                  learning to live on less than you make.",
            author: "Warren Buffett"
        },
        {
            text: "Do not save what is left after spending, but spend
                  what is left after saving.",
            author: "Indian Proverb"
        },
        {
            text: "It's not how much money you make, but how much money
                  you keep.",
            author: "Dhirubhai Ambani"
        },
    ];
}
```

```
{  
    text: "The habit of saving is itself an education; it  
          fosters every virtue.",  
    author: "Ratan Tata"  
,  
{  
    text: "Every rupee saved is a rupee earned for your future  
          self.",  
    author: "Indian Proverb"  
,  
{  
    text: "Financial freedom is when your passive income  
          exceeds your expenses.",  
    author: "Radhika Gupta"  
,  
{  
  
const quoteElement = document.querySelector('.motivational-quote');  
const authorElement = document.querySelector('.quote-author');  
let currentQuote = 0;  
  
setInterval(() => {  
    currentQuote = (currentQuote + 1) % quotes.length;  
    quoteElement.style.opacity = 0;  
    authorElement.style.opacity = 0;  
  
    setTimeout(() => {  
        quoteElement.textContent = `#${quotes[currentQuote].text}`  
        ;  
  
        quoteElement.style.opacity = 1;  
        authorElement.style.opacity = 1;  
    }, 500);  
}, 8000);  
}  
  
// Initialize everything when DOM is loaded  
document.addEventListener("DOMContentLoaded", () => {  
    createParticles();  
    calculateTotalIncome();  
    rotateQuotes();  
  
    // Add fadeIn animation to CSS  
    const style = document.createElement('style');
```

```

        style.textContent = `

@keyframes fadeIn {
    from { opacity: 0; transform: translateY(10px); }
    to { opacity: 1; transform: translateY(0); }
}

`;

document.head.appendChild(style);

// Random reminder notification
setTimeout(() => {
    document.getElementById('reminderBadge').style.display = 'flex'
},
), 2000);
});

</script>
</body>
</html>

```

## F.0.6 Add Expense Page

```

{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Track Expense | Financial Freedom</title>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css"></head>

<body>

<div class="expense-tracker animate__animated animate__fadeInUp">
    <h1>Track Expense</h1>
    <p class="motivation-text">"Mindful spending leads to financial freedom"</p>

    <form method="POST" action="{% url 'expenses:expense_create' %}">
        {% csrf_token %}

        <div class="form-group">
            <input type="number" step="0.01" name="amount" id="amount" class="form-control" required placeholder="0.00" min="0">
        </div>

        <div class="form-group">

```

```

<label for="category">Category</label>
<select name="category" id="category" class="form-control"
        required>
    { % for category in categories %}
        <option value="{{ category }}>{{ category }}</option>
    { % empty %}
        <option disabled>No categories available</option>
    { % endfor %}
</select>
</div>

<div class="form-group">
    <label for="date">Date</label>
    <input type="date" name="date" id="date" class="form-control"
           required>
</div>

<div class="form-group">
    <label for="description">Description</label>
    <textarea name="description" id="description" class="form-
control" placeholder="What was this expense for?"></
textarea>
</div>

<button type="submit" class="btn btn-primary">
    <svg class="btn-icon" viewBox="0 0 24 24" fill="none" stroke="
currentColor" stroke-width="2">
        <path d="M19 21H5a2 2 0 0 1-2-2V5a2 2 0 0 1 2-2h11l5 5v11a2
2 0 0 1-2 2z"></path>
        <polyline points="17 21 17 13 7 13 7 21"></polyline>
        <polyline points="7 3 7 8 15 8"></polyline>
    </svg>
    Record Expense
</button>

<a href="{% url 'expenses:list' %}" class="btn btn-secondary">
    <svg class="btn-icon" viewBox="0 0 24 24" fill="none" stroke="
currentColor" stroke-width="2">
        <line x1="8" y1="6" x2="21" y2="6"></line>
        <line x1="8" y1="12" x2="21" y2="12"></line>
        <line x1="8" y1="18" x2="21" y2="18"></line>
        <line x1="3" y1="6" x2="3.01" y2="6"></line>
        <line x1="3" y1="12" x2="3.01" y2="12"></line>
        <line x1="3" y1="18" x2="3.01" y2="18"></line>
    </svg>

```

```

        View Expenses
    </a>

    <a href="{% url 'dashboard' %}" class="btn btn-secondary">
        <svg class="btn-icon" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2">
            <line x1="19" y1="12" x2="5" y2="12"></line>
            <polyline points="12 19 5 12 12 5"></polyline>
        </svg>
        Back to Dashboard
    </a>
</form>
</div>

<script>
    document.addEventListener('DOMContentLoaded', function() {
        // Set today's date as default
        const today = new Date().toISOString().split('T')[0];
        document.getElementById('date').value = today;

        // Auto-focus amount field
        document.getElementById('amount').focus();
    });
</script>
</body>
</html>

```

### F.0.7 View Expense List Page

```

{%
    block content %
}
<div class="expense-container">
    <div class="header-section">
        <h1 class="page-title">Expense Tracker</h1>
        <a href="{% url 'add_expense' %}" class="add-btn">
            <svg xmlns="http://www.w3.org/2000/svg" width="18" height="18"
                viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-linejoin="round">
                <line x1="12" y1="5" x2="12" y2="19"></line>
                <line x1="5" y1="12" x2="19" y2="12"></line>
            </svg>
            Add Expense
        </a>
    </div>

```

```

<!-- Motivation Section -->
<div class="motivation-card">
    <p class="motivation-quote">"Beware of little expenses; a small leak
        will sink a great ship."</p>
    <p class="motivation-author">- Benjamin Franklin</p>
</div>

{%
    if expenses %}
    <div class="table-responsive">
        <table class="expense-table">
            <thead>
                <tr>
                    <th>Description</th>
                    <th>Amount</th>
                    <th>Date</th>
                    <th>Category</th>
                    <th>Actions</th>
                </tr>
            </thead>
            <tbody>
                {%
                    for expense in expenses %}
                <tr style="--row-index: {{ forloop.counter }};">
                    <td>{{ expense.description|default:"No Description" }}<
                        /td>
                    <td>{{ expense.date|date:"M d, Y" }}</td>
                    <td><span class="category-badge">{{ expense.category }}<
                        /span></td>
                    <td>
                        <div class="action-btns">
                            <a href="{% url 'expenses:expense_update'&
                                expense.pk %}" class="edit-btn">
                                <svg xmlns="http://www.w3.org/2000/svg">
                                    width="14" height="14" viewBox="0 0 24
                                    24" fill="none" stroke="currentColor"
                                    stroke-width="2" stroke-linecap="round"
                                    stroke-linejoin="round">
                                    <path d="M11 4H4a2 2 0 0 0-2 2v14a2 2 0
                                        0 0 2 2h14a2 2 0 0 0 2-2v-7"></
                                    path>
                                    <path d="M18.5 2.5a2.121 2.121 0 0 1 3
                                        3L12 15l-4 1 1-4 9.5-9.5z"></path>
                                </svg>
                                Edit
                            </a>
                            <form action="{% url 'expenses:expense_delete'&
                                expense.id %}" method="post">
                                <input type="hidden" name="csrf_token" value="
                                    {{ csrf_token }}"/>
                                <button type="submit" class="delete-btn">Delete</button>
                            </form>
                        </div>
                    <td>
                </td>
            </tbody>
        </table>
    </div>

```

```

                expense.pk %}" method="post" style="display
                :inline;">
                {% csrf_token %}
                <button type="submit" class="delete-btn">
                    <svg xmlns="http://www.w3.org/2000/svg"
                        width="14" height="14" viewBox="0
                        0 24 24" fill="none" stroke="
                        currentColor" stroke-width="2"
                        stroke-linecap="round" stroke-
                        linejoin="round">
                        <polyline points="3 6 5 6 21 6"></
                        polyline>
                        <path d="M19 6v14a2 2 0 0 1-2 2H7a2
                            2 0 0 1-2-2V6m3 0V4a2 2 0 0 1
                            2-2h4a2 2 0 0 1 2 2v2"></path>
                    </svg>
                    Delete
                </button>
            </form>
        </div>
    </td>
</tr>
{% endfor %}
</tbody>
</table>
</div>

{% else %}
<div class="empty-state">
    <h3 class="empty-title">No Expenses Recorded</h3>
    <p class="empty-text">Start tracking your expenses to gain
        financial control</p>
    <a href="{% url 'expenses:expense_create' %}" class="add-btn" style
        ="margin: 0 auto;">
        Add Your First Expense
    </a>
</div>
{% endif %}
</div>

<script>
    document.addEventListener('DOMContentLoaded', function() {
        // Add animation to rows when they come into view
        const observer = new IntersectionObserver(entries) => {

```

```
        entries.forEach(entry => {
            if (entry.isIntersecting) {
                entry.target.style.opacity = 1;
                entry.target.style.transform = 'translateY(0)';
            }
        });
    }, { threshold: 0.1 });

document.querySelectorAll('.expense-table tbody tr').forEach(row => {
    observer.observe(row);
});

// If there are expenses, show a little celebration
if (document.querySelectorAll('.expense-table tbody tr').length > 0) {
    setTimeout(() => {
        createConfetti();
    }, 1000);
}

function createConfetti() {
    const container = document.querySelector('.expense-container');

    for (let i = 0; i < 30; i++) {
        const confetti = document.createElement('div');
        confetti.className = 'confetti';

        // Random properties
        const colors = ['#6c5ce7', '#00cec9', '#fd79a8', '#fdcb6e', '#00b894'];
        const color = colors[Math.floor(Math.random() * colors.length)];
        const size = Math.random() * 10 + 5;
        const left = Math.random() * 100;
        const animationDuration = Math.random() * 3 + 2;

        confetti.style.backgroundColor = color;
        confetti.style.width = `${size}px`;
        confetti.style.height = `${size}px`;
        confetti.style.left = `${left}%`;
        confetti.style.top = '-10px';
        confetti.style.borderRadius = '50%';
        confetti.style.animationDuration = `${animationDuration}s`;

        container.appendChild(confetti);
    }
}
```

```

        // Add animation
        setTimeout(() => {
            confetti.style.animation = 'confettiFall linear forwards';
        }, 10);

        // Remove after animation
        setTimeout(() => {
            confetti.remove();
        }, animationDuration * 1000);
    }
}

</script>
{%
endblock %}

```

## F.0.8 Goal List View Page

```

{% block content %}
<div class="container mx-auto p-6 max-w-6xl">
    <!-- Main Header -->
    <h1 class="main-header">YOUR SAVINGS JOURNEY</h1>

    <!-- Motivation Section -->
    <div class="motivation-container">
        <p class="motivation-quote">"Every rupee saved is a step closer to your
            dreams."</p>
        <p class="motivation-author">- Financial Freedom</p>
    </div>

    <!-- Achievement Badge (hidden by default, shown via JS) -->
    <div id="achievement-badge" class="hidden fixed top-10 right-10 bg-yellow
        -400 text-black px-4 py-2 rounded-full shadow-lg z-50 font-bold animate
        -bounce">

    </div>

    <div class="flex justify-center mb-8">
        <a href="{% url 'goals:goal_create' %}" class="create-btn px-6 py-3
            rounded-lg text-lg font-semibold transform hover:scale-105
            transition-all duration-300">
        </a>
    </div>

    <div class="goal-card overflow-hidden p-2">
        <div class="overflow-x-auto">

```

```


| Goal Name       | Target Amount                            | Current Balance                            | Last Allocated                              | Progress                                                                                                                                                                                                    | Actions                                                                                                                                           |
|-----------------|------------------------------------------|--------------------------------------------|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| {{ goal.name }} | Rs{{ goal.target_amount floatformat:2 }} | Rs{{ goal.current_balance floatformat:2 }} | Rs{{ goal.allocated_amount floatformat:2 }} | <div class="progress-bar-container"> <div %}="" %}"="" &gt;="goal.target_amount" 100="" class="progress-bar" endif="" goal.current_balance="" goal.target_amount="" if="" widthratio="" {%=""></div> </div> | {{ goal.current_balance floatformat:2 }} / {{ goal.target_amount floatformat:2 }} ({{ widthratio goal.current_balance goal.target_amount 100 }}%) |


```

```

        </a>
        <a href="{% url 'goals:goal_delete' goal.id %}" class="action-btn delete-btn text-sm transform hover:scale-110 transition-transform">
            Delete
        </a>
    </div>
</td>
</tr>
{ % empty %}
<tr>
    <td colspan="6" class="empty-state text-center p-8">
        <div class="text-xl mb-2">No savings goals yet</div>
        <p class="text-gray-500 mb-4">Your financial journey begins with the first step</p>
        <a href="{% url 'goals:goal_create' %}" class="create-btn inline-block px-6 py-2 rounded-lg text-sm font-semibold">
            Create First Goal
        </a>
    </td>
</tr>
{ % endfor %}
</tbody>
</table>
</div>
</div>

<!-- Inspirational Footer --&gt;
&lt;div class="mt-8 text-center text-gray-600 italic"&gt;
    &lt;p&gt;"Small savings today become big achievements tomorrow."&lt;/p&gt;
&lt;/div&gt;
&lt;/div&gt;

&lt;script&gt;
    document.addEventListener('DOMContentLoaded', function() {
        // Progress bar animation
        const progressBars = document.querySelectorAll('.progress-bar');
        const observer = new IntersectionObserver((entries) =&gt; {
            entries.forEach(entry =&gt; {
                if (entry.isIntersecting) {
                    const width = entry.target.style.width;
                    entry.target.style.width = '0';
                }
            })
        })
        observer.observe(progressBars);
    })
&lt;/script&gt;
</pre>

```

```

        setTimeout(() => {
            entry.target.style.width = width;
        }, 100);
    }
});

}, { threshold: 0.1 });
progressBars.forEach(bar => observer.observe(bar));

// Check for completed goals and celebrate
const completedGoals = document.querySelectorAll('.progress-complete');
if (completedGoals.length > 0) {
    celebrateAchievement();
}

// Random motivational quotes
const quotes = [
    {text: "Financial freedom is a journey that begins with a single
        rupee saved.", author: "Wealth Wisdom"},
    {text: "Don't save what is left after spending; spend what is left
        after saving.", author: "Warren Buffett"},
    {text: "The secret to getting ahead is getting started.", author: "
        Mark Twain"},
    {text: "Small amounts saved daily add up to big dreams realized.",
        author: "Financial Freedom"},
    {text: "Every rupee you save is a vote for the future you want.",
        author: "Savings Proverb"}
];

const randomQuote = quotes[Math.floor(Math.random() * quotes.length)];
document.querySelector('.motivation-quote').textContent = ` ${
    randomQuote.text}`;
document.querySelector('.motivation-author').textContent = `- ${
    randomQuote.author}`;

// Animate the quote
setInterval(() => {
    const quoteElement = document.querySelector('.motivation-quote');
    quoteElement.style.transform = 'scale(1.05)';
    setTimeout(() => {
        quoteElement.style.transform = 'scale(1)';
    }, 300);
}, 10000);
});

function celebrateAchievement() {

```

```
const badge = document.getElementById('achievement-badge');
badge.classList.remove('hidden');

// Create confetti
for (let i = 0; i < 50; i++) {
    createConfetti();
}

// Hide badge after 5 seconds
setTimeout(() => {
    badge.classList.add('hidden');
}, 5000);
}

function createConfetti() {
    const confetti = document.createElement('div');
    confetti.className = 'confetti';

    // Random properties
    const colors = ['#ff0000', '#00ff00', '#0000ff', '#ffff00', '#ff00ff',
        '#00ffff'];
    const color = colors[Math.floor(Math.random() * colors.length)];
    const size = Math.random() * 10 + 5;
    const left = Math.random() * window.innerWidth;

    confetti.style.backgroundColor = color;
    confetti.style.width = `${size}px`;
    confetti.style.height = `${size}px`;
    confetti.style.left = `${left}px`;
    confetti.style.top = '-10px';

    document.body.appendChild(confetti);

    // Add animation
    setTimeout(() => {
        confetti.classList.add('celebration-animation');
    }, 10);

    // Remove after animation
    setTimeout(() => {
        confetti.remove();
    }, 3000);
}
</script>
{% endblock %}
```

## F.0.9 Goal Create Page

```

{%
    block content %
}

<div class="form-container">
    <div class="form-card">
        <div class="form-header">
            <h1 class="form-title">
                {%
                    if form.instance.pk %}Edit Goal{%
                    else %}Create New Goal{%
                endif %}
            </h1>
            <p class="form-subtitle">
                {%
                    if form.instance.pk %}Update your financial target{%
                    else %}
                        Start your journey to financial success{%
                    endif %}
                </p>
        </div>

        <div class="form-body">
            <div class="progress-tracker">
                <div class="progress-step active"></div>
                <div class="progress-step {%
                    if form.instance.pk %}active{%
                endif %}"></div>
                <div class="progress-step"></div>
            </div>

            <form method="post" class="space-y-4">
                {%
                    csrf_token %
                }

                {%
                    for field in form %
                }
                <div class="form-group" style="--order: {{ forloop.counter }}">
                    <label for="{{ field.id_for_label }}" class="form-label">
                        {{ field.label }}
                    </label>
                    <div class="{%
                        if field.name == 'target_amount' %}icon-
                        input{%
                            endif %
                        }">
                            {%
                                if field.name == 'target_amount' %
                            }

                            {%
                                endif %
                            }
                            {{ field }}
                        </div>
                    {%
                        if field.errors %
                    }
                    <div class="error-message">
                        {{ field.errors.0 }}
                    </div>
                {%
                    endif %
                }
            </div>
        
```

```

{ % endfor %}

<div class="form-footer">
    <a href="{% url 'goals:goal_list' %}" class="btn btn-
        outline">
        <svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5"
            viewBox="0 0 20 20" fill="currentColor">
            <path fill-rule="evenodd" d="M9.707 16.707a1 1 0
                01-1.414 0l-6-6a1 1 0 00-1.414-1.414L5.414 9H17a1 1 0 00 110 2H5.414L
                .293 4.293a1 1 0 00-1.414z" clip-rule="evenodd"
            </path>
        </svg>
        Cancel
    </a>
    <button type="submit" class="btn btn-primary">
        { % if form.instance.pk %}
        <svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5"
            viewBox="0 0 20 20" fill="currentColor">
            <path fill-rule="evenodd" d="M16.707 5.293a1 1 0
                00 1.414l-8 8a1 1 0 00-1.414-1.414L8 12.586L7.293-7.293a1 1 0
                00-1.414 0z" clip-rule="evenodd" />
        </svg>
        Update Goal
    { % else %}
        <svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5"
            viewBox="0 0 20 20" fill="currentColor">
            <path fill-rule="evenodd" d="M10 5a1 1 0 00 0 11 1
                v3h3a1 1 0 00 110 2h-3v3a1 1 0 00 11-2 0v-3H6a1 1 0
                00-110 2h-3V6a1 1 0 00-11-1z" clip-rule="evenodd" />
        </svg>
        Create Goal
    { % endif %}
    </button>
</div>
</form>
</div>
</div>

<script>
    document.addEventListener('DOMContentLoaded', function() {
        // Add focus class to inputs
        const inputs = document.querySelectorAll('.form-control');

```

```

inputs.forEach(input => {
    input.addEventListener('focus', function() {
        this.parentNode.classList.add('focused');
    });

    input.addEventListener('blur', function() {
        this.parentNode.classList.remove('focused');
    });

    // If there's existing value, add filled class
    if (input.value) {
        input.classList.add('filled');
    }

    input.addEventListener('input', function() {
        if (this.value) {
            this.classList.add('filled');
        } else {
            this.classList.remove('filled');
        }
    });
});

// Animate progress tracker when editing
if (document.querySelector('form[method="post"]').dataset.editing) {
    const steps = document.querySelectorAll('.progress-step');
    steps.forEach((step, index) => {
        setTimeout(() => {
            step.classList.add('active');
        }, index * 300);
    });
}

// Sparkle effect on button hover
const submitBtn = document.querySelector('.btn-primary');
if (submitBtn) {
    submitBtn.addEventListener('mouseenter', function() {
        createSparkles(this);
    });
}
}

function createSparkles(element) {
    const rect = element.getBoundingClientRect();
    const colors = ['#4f46e5', '#ec4899', '#f59e0b', '#10b981', '#fffacd'];

```

```
for (let i = 0; i < 8; i++) {
    const sparkle = document.createElement('div');
    sparkle.className = 'sparkle';

    const size = Math.random() * 6 + 3;
    const left = Math.random() * rect.width;
    const animationDuration = Math.random() * 0.8 + 0.5;
    const color = colors[Math.floor(Math.random() * colors.length)];

    sparkle.style.width = `${size}px`;
    sparkle.style.height = `${size}px`;
    sparkle.style.left = `${left}px`;
    sparkle.style.backgroundColor = color;
    sparkle.style.animationDuration = `${animationDuration}s`;

    element.appendChild(sparkle);

    setTimeout(() => {
        sparkle.remove();
    }, animationDuration * 1000);
}

</script>
{ % endblock %}
```

# Bibliography

- [1] *First Lessons in L<sup>A</sup>T<sub>E</sub>X*, by Dr. V N Krishnachandran, Vidya Academy of Science & Technology, Thrissur - 680 501, 2011.
- [2] Django Software Foundation, *Django Documentation*, <https://docs.djangoproject.com/>, Accessed on April 2025.
- [3] Tailwind Labs, *Tailwind CSS Documentation*, <https://tailwindcss.com/docs>, Accessed on April 2025.
- [4] Mozilla Developer Network (MDN), *HTML, CSS, and JavaScript Guide*, <https://developer.mozilla.org/>, Accessed on April 2025.
- [5] Python Software Foundation, *Python Official Documentation*, <https://docs.python.org/>, Accessed on April 2025.
- [6] S. Krug, *Don't Make Me Think: A Common Sense Approach to Web Usability*, New Riders, 2014.
- [7] Y. S. Wang, *Personal Budgeting: Planning and Managing Finances*, Financial Times Press, 2018.



**Department of Computer Applications**  
Vidya Academy of Science & Technology  
Thalakkottukara, Thrissur - 680 501  
(<http://www.vidyaacademy.ac.in>)