

# Visvesvaraya Technological University

Jnana Sangama, Belagavi - 590018



## **A Mini Project (MMCL106) Report on**

### **“FINANCIAL TRANSACTION MANAGEMENT”**

*Mini Project Report submitted in partial fulfilment of the requirement for  
the award of the degree of*

**MASTER OF COMPUTER APPLICATIONS**

**Submitted by**

**CHINMAYEE B L**

**1KS24MC008**

**KEERTHIKA L**

**1KS24MC013**

**VEENA SERI**

**1KS24MC029**

Under the guidance of

**DR. SNEHA GIRISH**



**DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS**

**K. S. Institute of Technology**

**#14, Raghuvanahalli, Kanakapura Road, Bengaluru - 560109**

**2024 - 2025**

# K. S. Institute of Technology

#14, Raghuvanahalli, Kanakapura Road, Bengaluru - 560109

## DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS



### CERTIFICATE

Certified that the Mini Project (**MMCL106**) entitled “**FINANCIAL TRANSACTION MANAGEMENT**” is a bonafide work conducted by:

**CHINMAYEE B L**

**1KS24MC008**

**KEERTHIKA L**

**1KS24MC013**

**VEENA SERI**

**1KS24MC029**

in partial fulfilment for I semester MCA, as a part of skill development activity work in the branch of Master of Computer Applications prescribed by **Visvesvaraya Technological University, Belagavi** during the Academic year of 2024-2025. It is certified that all the corrections and suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The Mini Project Report has been approved as it satisfies the academic requirements in the report of project work prescribed for the Master of Computer Applications.

.....  
**Signature of the Guide**

[Dr. Sneha Girish]

.....  
**Signature of the HOD**

[Dr. Sneha Girish]

## DECLARATION

We, the undersigned students of 1<sup>st</sup> semester, department of Master of Computer Applications, KSIT, declare that our Mini Project Work “**FINANCIAL TRANSACTION MANAGEMENT**”, is a bonafide work of ours. Our project is neither a copy nor by means a modification of any other project.

We also declare that this project was not entitled for submission to any other university in the past and shall remain the only submission made and will not be submitted by us to any other university in the future.

Place:

Bengaluru Date:

09-03-2025

**Name and USN**

**Signature**

**CHINMAYEE B L (1KS24MC008)**

.....

**KEERTHIKA L (1KS24MC013)**

.....

**VEENA SERI (1KS24MC029)**

.....

## ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task will be incomplete without the mention of the individuals, we are indebted to, who through guidance and providing facilities have served as a beacon of light and crowned our efforts with success.

Primarily, our sincere prayer goes to the almighty, whose grace made us realize our objective and conceive this project. We take pleasure in expressing our profound sense of gratitude to our parents for helping us complete our Mini Project Work successfully.

We take this opportunity to express our sincere gratitude to our college **K.S. Institute of Technology**, Bengaluru for providing the environment to work on our project.

We would like to express our gratitude to our **MANAGEMENT**, K.S. Institute of Technology, Bengaluru, for providing an incredibly good infrastructure and all the kindness forwarded to us in carrying out this project work in college.

We would like to express our gratitude to **Dr. K.V.A Balaji**, CEO, K.S. Group of Institutions, Bengaluru, for his valuable guidance.

We would like to express our gratitude to **Dr. Dilip Kumar K**, Principal/Director, K.S. Institute of Technology, Bengaluru, for his continuous support.

We would like to extend our gratitude to **Dr. Sneha Girish**, Professor and Head, Department of Master of Computer Applications, for providing incredibly good facilities and all the support forwarded to us in conducting this Project Work successfully.

We would like to thank our Project coordinator, **DR. Sneha Girish**, Professor and Head, Department of Master of Computer Applications for their help and support provided to conduct the Project Work successfully.

We are also thankful to the teaching and non-teaching staff of the department of Master of Computer Applications, KSIT for helping us in completing the Project work.

**CHINMAYEE B L  
KEERTHIKA L  
VEENA SERI**

## ABSTRACT

Financial Transaction Management system is designed to provide users with an efficient and user-friendly platform for tracking and managing their financial activities. Developed using Visual Studio Code with HTML and JavaScript, the system allows users to input their income and expense transactions, categorizing them by description, amount, date, and type.

The system offers robust visualization tools, displaying income and expense data through interactive charts that allow users to view these categories both separately and together. This visual representation aids users in quickly understanding their financial trends and making informed decisions. Additionally, the system highlights key financial metrics, such as the minimum and maximum transaction amounts, providing further insights into spending and saving patterns.

To enhance usability, This Financial Transaction Management system also includes a transaction history feature that displays the three most recent transactions along with its date, offering users a quick reference to their latest financial activities. The backend of the system is powered by MongoDB, ensuring secure and efficient storage of transaction data, while Postman is used as an API for smooth data management and communication.

## **TABLE OF CONTENTS**

<b>Chapter No.</b>	<b>Title</b>	<b>Page No.</b>
<b>1.</b>	<b>INTRODUCTION</b>	<b>7</b>
<b>2.</b>	<b>SOFTWARE REQUIREMENTS</b>	<b>8</b>
<b>3.</b>	<b>IMPLEMENTATION DETAILS</b>	<b>9</b>
<b>4.</b>	<b>FLOW CHART</b>	<b>10-12</b>
<b>5.</b>	<b>RESULTS SCREEN SHOTS</b>	<b>13-15</b>
<b>6.</b>	<b>FUTURE ENHANCEMENTS</b>	<b>16</b>
<b>7.</b>	<b>CONCLUSION</b>	<b>17</b>
<b>8.</b>	<b>REFERENCES</b>	<b>18</b>

### Chapter 1

#### INTRODUCTION

In today's digital age, effective financial management is crucial for both individuals and businesses. Traditional methods of tracking finances, such as manual record-keeping, are often inefficient and prone to errors. The Financial Transaction Management system offers a modern solution by allowing users to efficiently track their incomes and expenses. Users can input detailed financial transactions, categorized by description, amount, and type, with all data securely stored in a MongoDB database.

The system enhances financial oversight with interactive charts that visually represent income and expense data. Users can easily analyse their financial trends by viewing income and expenses separately in these charts. Additionally, the system displays the three most recent transactions and highlights the minimum and maximum amounts recorded, providing quick insights into financial activity.

Developed using Visual Studio Code with HTML and JavaScript and integrated with Postman as an API for seamless data management, the Financial Transaction Management system is designed to be user-friendly and comprehensive. It offers a reliable and efficient tool for individuals seeking to manage their finances more effectively.

## Chapter 2

### SOFTWARE REQUIREMENTS

**1. Operating System:** Windows 11(22621.2861)

**2. Backend Environment:**

- Node.js

**3. Libraries and Frameworks:**

- Express.js: For building the API.
- MongoDB: For database management.
- Mongoose: For object modeling.
- Chart.js: For generating charts.

**4. Frontend Environment:**

- **HTML:** for structuring web pages.
- **JavaScript:** for client-side scripting.

**5. IDE/Code Editor:**

- **VS Code:** For writing and managing code.
- **Postman:** For testing API endpoints.

**6. Database:**

- **MongoDB:** Storing financial transaction



## **Chapter 3**

### **IMPLEMENTATION DETAILS**

#### **1. System Design and Architecture:**

- Define system architecture including front-end and back-end components.
- Create wireframes for user interface layout and functionality.

#### **2. Development:**

- Set up MongoDB database and integrate with the front-end using APIs.
- Implement web pages using HTML and JavaScript for transaction entry.

#### **3. Testing:**

- Test all features to ensure data entry, display, and chart work correctly.
- Collect feedback from users to improve the interface and functionality.

#### **4. Deployment:**

- Deploy the system on a suitable web server.
- Provide user training and documentation.

#### **5. Maintenance:**

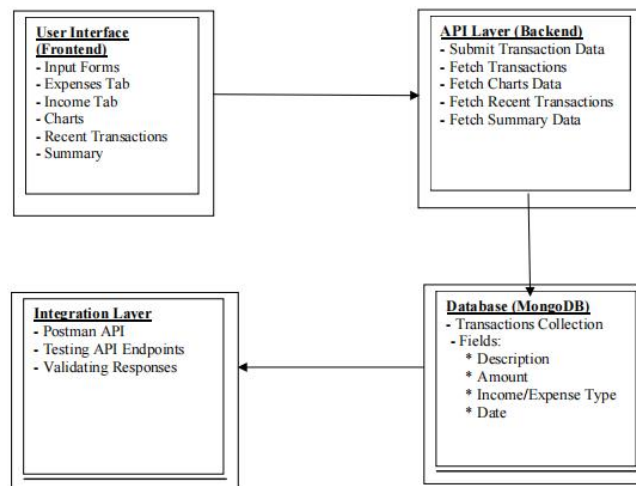
- Monitor system performance and address any issues.
- Update the system based on user feedback and technological advancements.

This modular description outlines the essential components and requirements for developing a financial transaction management system, ensuring effective tracking, visualization, and management of personal finances.

### Chapter 4

#### FLOW CHART

Block diagram:



1. User Interface: Allows users to input and view transactions, display charts, and see recent transactions and summaries. It provides forms for entering transaction data and displays results to the user.

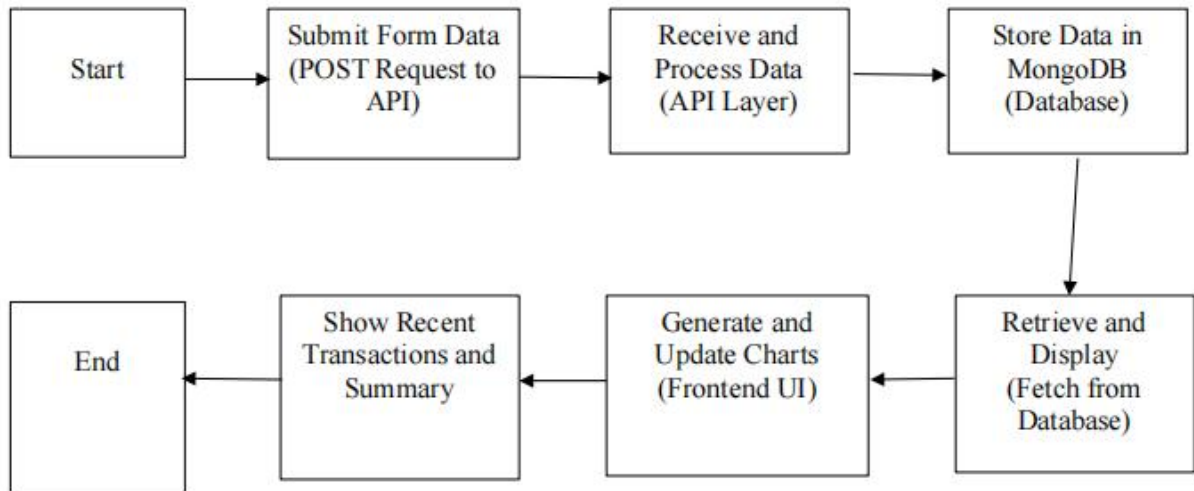
2. API Layer: Handles requests from the frontend, processes transaction data, and retrieves information from the database. It processes requests for submitting transactions, fetching data for display, and retrieving charts and summaries. It ensures data is correctly processed and retrieved.

3. Database (MongoDB): Stores transaction details such as descriptions, amounts, and types. It serves as the backend storage system for all transaction records, which the API layer queries to provide information to the frontend.

4. Integration Layer: Postman is used to test API endpoints and ensure that data is being correctly sent and received between the frontend, API layer, and database.

This setup ensures efficient data flow and interaction between users, the backend, and storage.

### WORKFLOW:



1) Start: The process begins when the user accesses the website. The user navigates to the financial transaction management website.

2) Submit Form Data (POST Request to API): The user inputs transaction details such as description, amount, and type into a form on the website. The user submits the form, sending a POST request to the API with the transaction details.

3) Receive and Process Data (API Layer): The API receives the data, processes it, and performs validation.

4) Store Data in MongoDB (Database): The API stores the validated transaction data in the MongoDB database for persistent storage.

5) Retrieve and Display Data (Fetch from Database): The API retrieves the updated data, transactions, charts, recent transactions, and summaries from MongoDB.

6) Generate and Update Charts (Frontend UI): The frontend updates the charts to reflect the latest income and expense data.

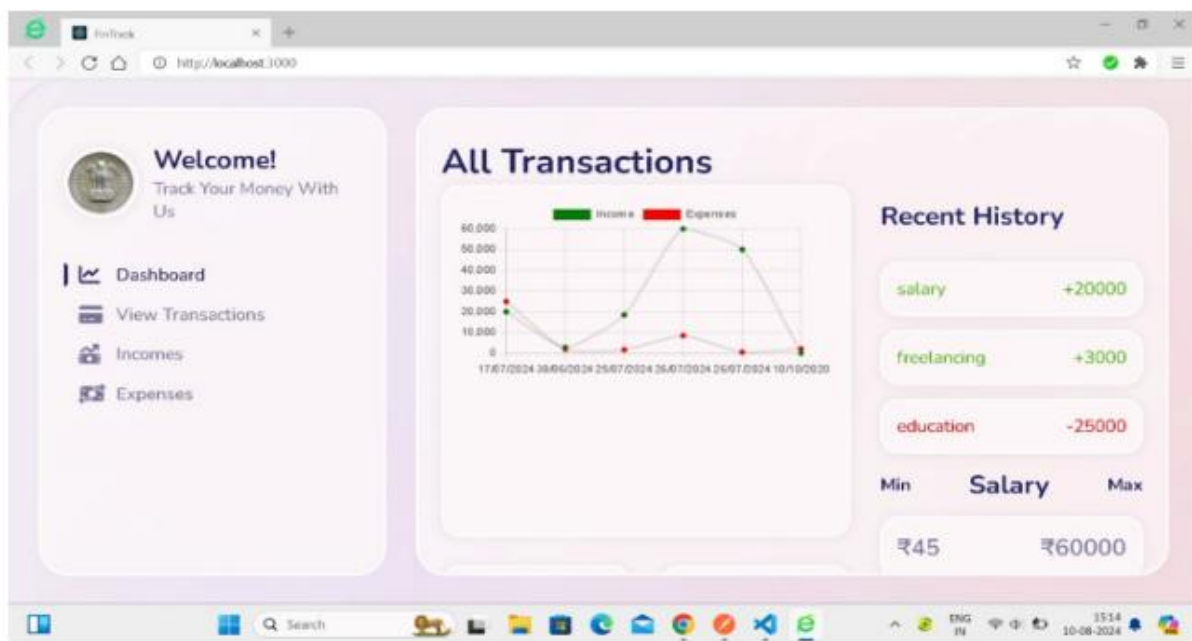
7) Show Recent Transactions and Summary: The frontend displays the recent transactions and summary information to the user.

8) End: The workflow concludes with updated information being visible to the user

## Chapter 5

### RESULTS SCREENSHOTS

**Dashboard:** This page contains the Chart representation, Total Income, Total Expenses, Total Balance, History, Minimum and Maximum amount.

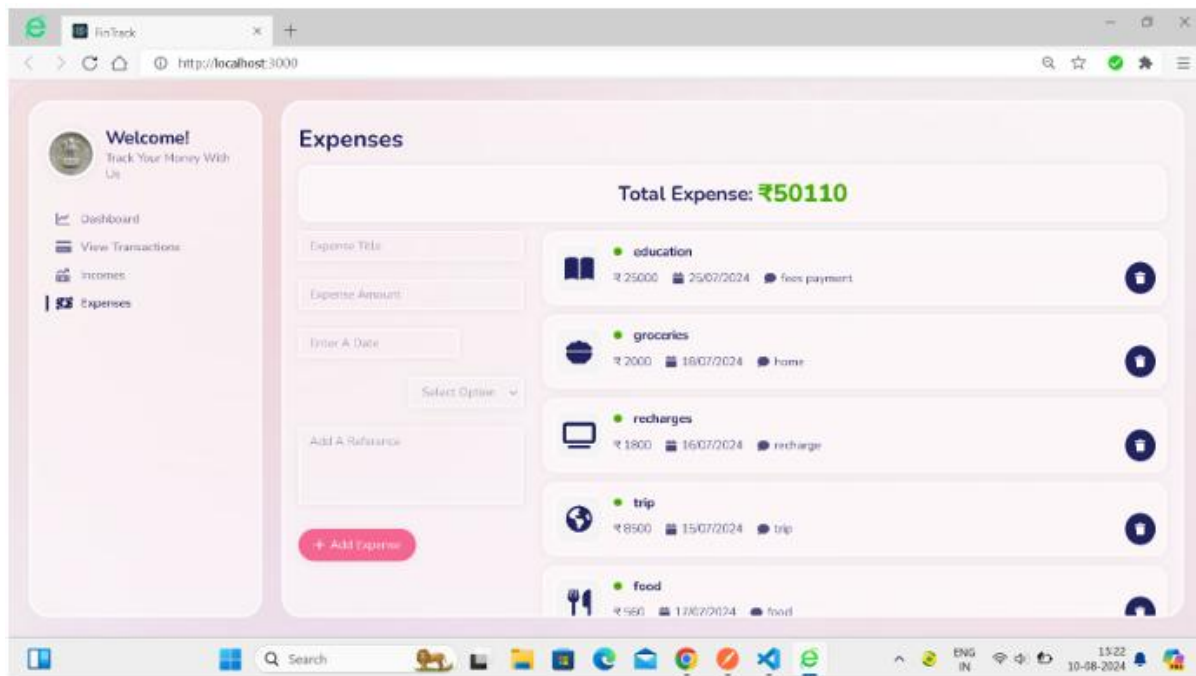


**Incomes:** The User can enter the following details in this page- Salary title, Salary amount, Date, Type of Salary, Reference

The screenshot shows the 'Incomes' page. On the left is a sidebar with a 'Welcome!' message and a menu: Dashboard, View Transactions, Incomes, and Expenses. The main area has a 'Total Income: ₹151545' at the top. Below it is a form to 'Add income' with fields for Salary Title, Salary Amount, Entry & Date, and a 'Select Option' dropdown. At the bottom of the form is a '+ Add income' button. To the right of the form is a list of existing income entries:

salary	₹ 20000	17/07/2024	income
freelancing	₹ 3000	30/06/2024	freelancing
youtube earnings	₹ 18000	25/07/2024	earnings
stock market	₹ 60000	26/07/2024	Stocks
education	₹ 10000	26/07/2024	education

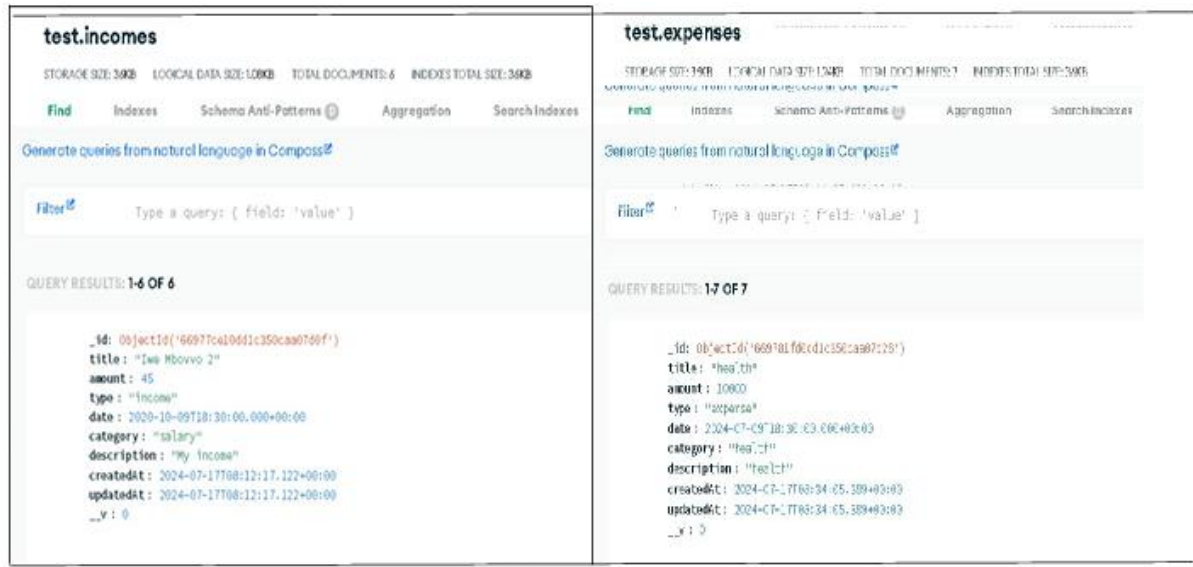
**Expenses:** The User can enter the following details in this page- Expense title, Expense amount, Date, Type of Expense, Reference and submit



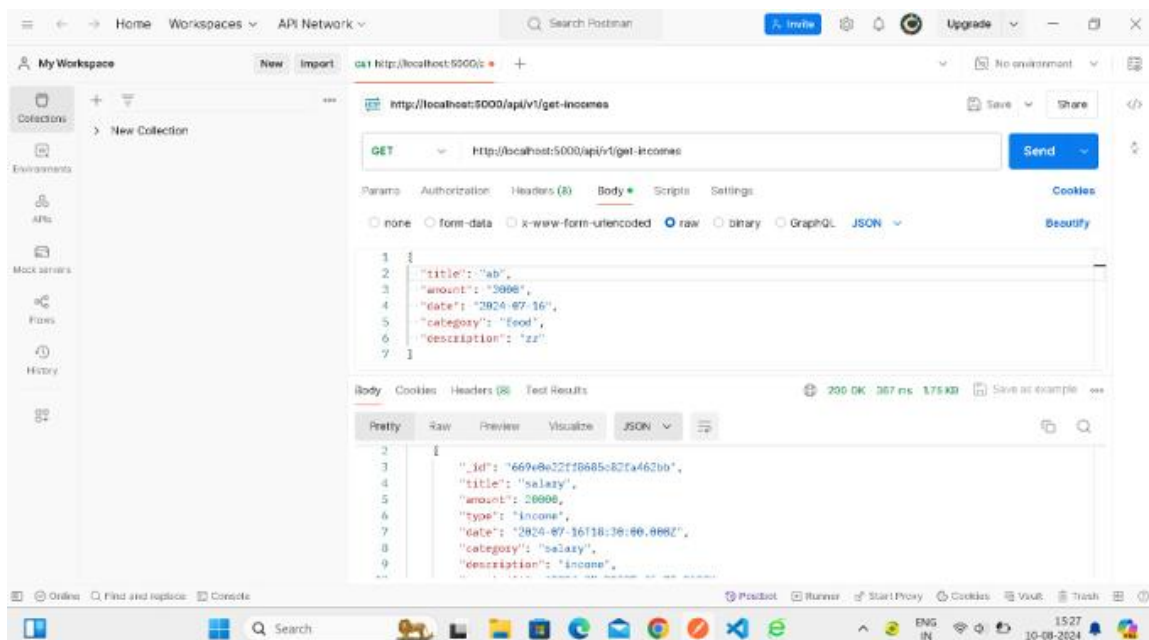
**Representation of Incomes and Expenses Separately in the Chart.**



**MongoDB:** This Database stores all the user filled data in it.



**Postman:** This API have Get, Post, Put, Delete options for managing the database



## Chapter 6

### FUTURE ENHANCEMENT

- **User Authentication & Security Enhancements**

To enhance security, the system can incorporate **two-factor authentication (2FA)** via email or SMS, ensuring that only authorized users access their financial data. Additionally, implementing **role-based access control** can allow different user permissions, such as admin and regular users. Sensitive financial data can also be encrypted using **AES encryption**, providing an extra layer of security and preventing unauthorized access.

- **Automated Bill Payments & Reminders**

A **bill payment automation feature** can be introduced to enable users to set up **automatic payments** for recurring expenses, such as rent, utilities, and subscriptions. The system can also include a **reminder system** that sends **email or SMS notifications** for upcoming payments, overdue bills, or low account balance alerts. This feature helps users stay on top of their financial obligations without manual tracking.

- **Mobile App Version**

To improve accessibility and usability, a **mobile application (Android & iOS)** can be developed, allowing users to manage their financial transactions on the go. The mobile app can sync data in real time with the web application, ensuring seamless updates. Additionally, an **offline mode** can be introduced, enabling users to log transactions without an active internet connection, with automatic syncing once the device is online.

## **Chapter 7**

### **CONCLUSION**

The Financial Transaction Management System serves as a reliable and user-friendly platform for tracking and managing financial activities. By allowing users to categorize transactions, visualize financial trends through interactive charts, and securely store data using MongoDB, the system enhances financial awareness and decision-making.

Future enhancements such as two-factor authentication (2FA), automated bill payments, and a mobile application will significantly improve security, efficiency, and accessibility. Integrating these features will ensure better financial planning, reduce manual effort, and provide users with real-time insights into their spending and savings. With continuous improvements, the system has the potential to become a comprehensive financial assistant, simplifying money management for individuals and businesses alike.



## Chapter 8

### REFERENCES

#### REVIEW THROUGH BOOKS

- Haverbeke, M. (2018). Eloquent JavaScript: A Modern Introduction to Programming (3rd ed.). No Starch Press.
- Duckett, J. (2014). JavaScript and JQuery: Interactive Front-End Web Development. Wiley
- Westerveld, D. (2021). API Testing and Development with Postman. Packt Publishing.
- Chodorow, K. (2019). MongoDB: The Definitive Guide (3rd ed.). O'Reilly Media.

#### REVIEW THROUGH WEB REFERENCE

- [Java Script Libraries and Framework](#)
- [How to get started with MongoDB](#)
- [Use of Postman for API Testing](#)

