

## **CONTENTS**

<b>Sr. No.</b>	<b>Content</b>	<b>Page No.</b>
1.	Abstract	7
2.	Introduction <ul style="list-style-type: none"><li>• Background</li><li>• Objectives</li><li>• Scope</li></ul>	8
3.	Implementation <ul style="list-style-type: none"><li>• System Architecture</li><li>• Database Design</li><li>• Tools &amp; Technologies</li><li>• Flow Chart</li><li>• E-R Diagram</li><li>• Code</li></ul>	9
4.	Result <ul style="list-style-type: none"><li>• System Functionality</li><li>• Example Output Tables</li></ul>	13
5.	Conclusion	15
6.	Future Scope	16
7.	References	17

# **MINI PROJECT**

## **ABSTRACT**

The **Personal Finance Management System** is designed to help users manage their financial activities efficiently. This system allows users to track their income, expenses, savings, and investments, providing a comprehensive overview of their financial health. The system offers features such as budgeting, expense categorization, and financial reporting, helping users make informed decisions about their finances. The project leverages database management techniques to store and retrieve financial data securely and efficiently.

# **INTRODUCTION**

## **Background**

Managing personal finances is a critical task for individuals to maintain financial stability and achieve their financial goals. However, many people struggle to keep track of their income and expenses, leading to poor financial decisions. A well-designed Personal Finance Management System can help users organize their finances and provide insights into their spending habits.

## **Objectives**

The primary objectives of this project are:

- To develop a system that allows users to track income and expenses.
- To categorize transactions for better understanding.
- To provide budgeting tools to help users manage their money.
- To generate reports that give insights into spending patterns and savings.

## **Scope**

This system is intended for individuals who want to manage their personal finances. It can be used by anyone looking to keep track of income, expenses, savings, and investments in an organized manner.

# Implementation

## System Architecture

The system is divided into several modules:

1. **User Module:** Handles user registration, login, and profile management.
2. **Income Module:** Allows users to record various sources of income.
3. **Expense Module:** Enables users to input expenses and categorize them (e.g., food, rent, utilities).
4. **Budget Module:** Users can set monthly or yearly budgets for different expense categories.
5. **Reports Module:** Generates reports showing income vs expenses, savings trends, etc.
6. **Investment Module:** Tracks investments in stocks, mutual funds, or other assets.

## Database Design

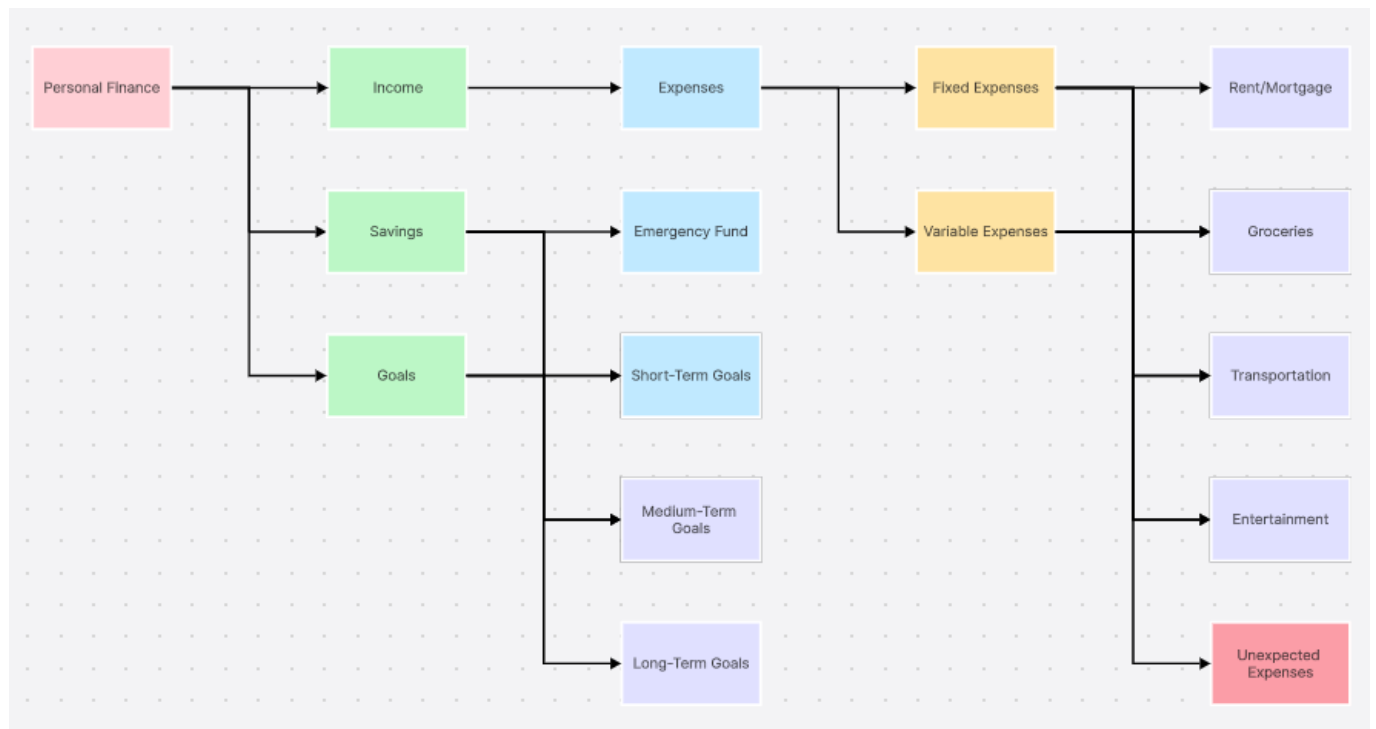
The database consists of the following tables:

- **Users Table:** Stores user information such as username, password (hashed), email, etc.
- **Income Table:** Stores details of income transactions (amount, date, source).
- **Expense Table:** Stores details of expense transactions (amount, date, category).
- **Budget Table:** Stores budget limits for different categories.
- **Investment Table:** Stores details of user investments (type of investment, amount invested).

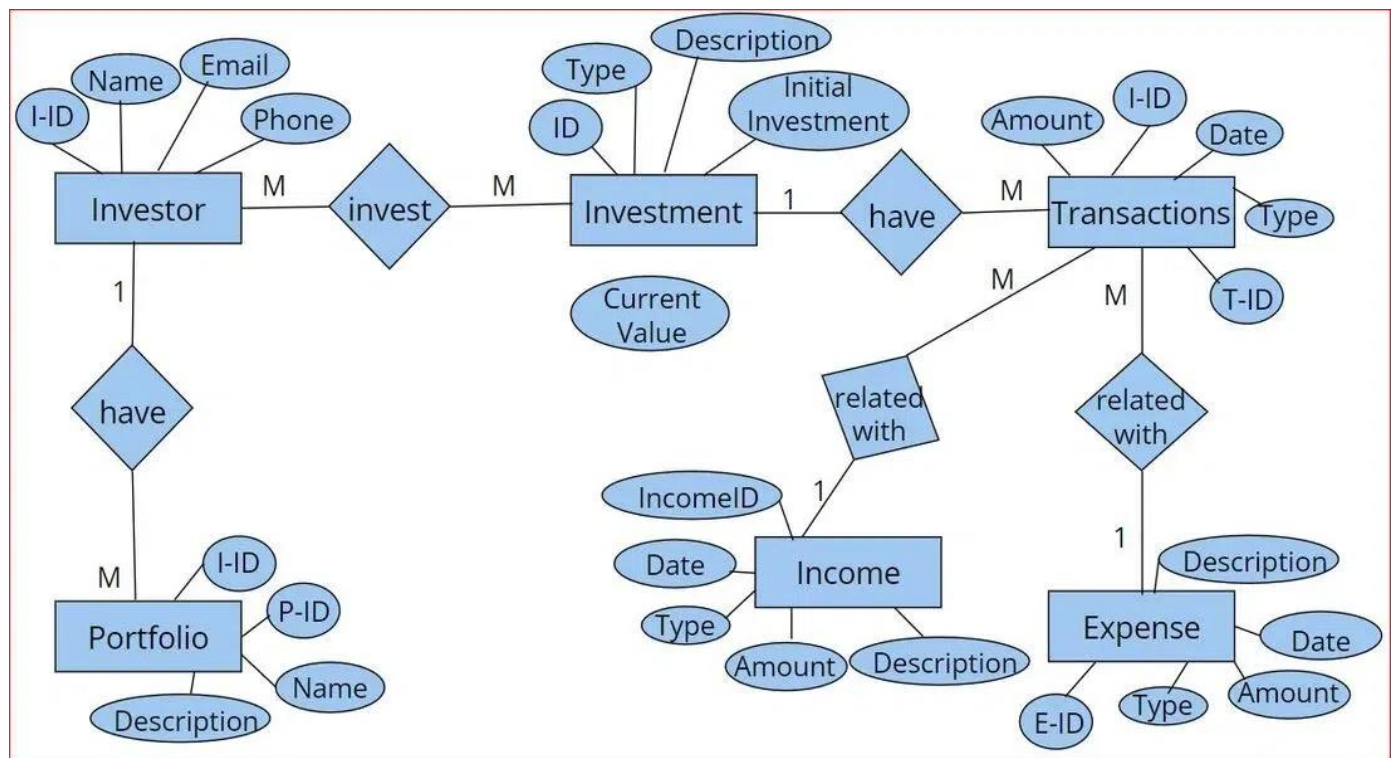
## Tools & Technologies

- **Database Management System (DBMS):** MySQL or PostgreSQL
- **Frontend:** HTML/CSS/JavaScript for the user interface
- **Backend:** PHP/Node.js/Python for handling server-side logic
- **IDE:** Visual Studio Code or any other preferred IDE

## Flow Chart



## E-R Diagram



## Code

-- Create Users Table

```
CREATE TABLE Users (  
    user_id INT PRIMARY KEY AUTO_INCREMENT,  
    username VARCHAR(50) NOT NULL UNIQUE,  
    password VARCHAR(255) NOT NULL,  
    email VARCHAR(100) NOT NULL UNIQUE  
);
```

-- Create Income Table

```
CREATE TABLE Income (  
    income_id INT PRIMARY KEY AUTO_INCREMENT,  
    user_id INT NOT NULL,  
    amount DECIMAL(10, 2) NOT NULL,  
    date DATE NOT NULL,  
    source VARCHAR(100) NOT NULL,  
    FOREIGN KEY (user_id) REFERENCES Users(user_id)  
);
```

-- Create Expenses Table

```
CREATE TABLE Expenses (  
    expense_id INT PRIMARY KEY AUTO_INCREMENT,  
    user_id INT NOT NULL,  
    amount DECIMAL(10, 2) NOT NULL,  
    date DATE NOT NULL,  
    category VARCHAR(100) NOT NULL,  
    FOREIGN KEY (user_id) REFERENCES Users(user_id)  
);
```

-- Create Budget Table

```
CREATE TABLE Budget (  
    budget_id INT PRIMARY KEY AUTO_INCREMENT,  
    user_id INT NOT NULL,  
    category VARCHAR(100) NOT NULL,  
    limit_amount DECIMAL(10, 2) NOT NULL,  
    FOREIGN KEY (user_id) REFERENCES Users(user_id)  
);
```

-- Create Investments Table

```
CREATE TABLE Investments (  
    investment_id INT PRIMARY KEY AUTO_INCREMENT,  
    user_id INT NOT NULL,  
    type VARCHAR(100) NOT NULL,  
    amount_invested DECIMAL(10, 2) NOT NULL,  
    date DATE NOT NULL,  
    FOREIGN KEY (user_id) REFERENCES Users(user_id)  
);
```

-- Insert a User into the Users table

```
INSERT INTO Users (username, password, email) VALUES  
('john_doe', 'hashed_password_123', 'john@example.com');
```

-- Insert Income Entries

```
INSERT INTO Income (user_id, amount, date, source) VALUES  
(1, 1500.00, '2024-01-10', 'Salary'),  
(1, 300.00, '2024-01-15', 'Freelance');
```

-- Insert Expense Entries

```
INSERT INTO Expenses (user_id, amount, date, category) VALUES
```



```
(1, 200.00, '2024-01-11', 'Groceries'),  
(1, 100.00, '2024-01-12', 'Entertainment'),  
(1, 1500.00, '2024-01-14', 'Rent');
```

-- Query to get the user's financial overview (Total Income, Total Expenses, and Net Savings)

```
SELECT
```

```
    (SELECT SUM(amount) FROM Income WHERE user_id = 1) AS Total_Income,  
    (SELECT SUM(amount) FROM Expenses WHERE user_id = 1) AS Total_Expenses,  
    (SELECT SUM(amount) FROM Income WHERE user_id = 1) - (SELECT  
SUM(amount) FROM Expenses WHERE user_id = 1) AS Net_Savings;
```

# RESULT

## System Functionalities

The system was tested with sample data to ensure that all modules function as expected:

1. Users were able to register and log in successfully.
2. Income and expense entries were stored correctly in the database.
3. Budgets were set and tracked against actual expenses.
4. Reports were generated showing detailed insights into user finances.

## Example Output Tables

- Users Table

user_id	username	password	email
1	john_doe	hashed_password_123	<a href="mailto:john@example.com">john@example.com</a>

- Income Table

income_id	user_id	amount	date	source
1	1	1500.00	2024-01-10	Salary
2	1	300.00	2024-01-15	Freelance

- Expenses Table

expense_id	user_id	amount	date	category
1	1	200.00	2024-01-11	Groceries
2	1	100.00	2024-01-12	Entertainment
3	1	1500.00	2024-01-14	Rent

- Financial Overview Query Result

Total_Income	Total_Expenses	Net_Savings
1800.00	1800.00	0.00

## **Conclusion**

The **Personal Finance Management System** successfully fulfills its objective by providing users with an easy-to-use platform for managing their finances. The system allows users to track income and expenses efficiently while offering additional features such as budgeting and investment tracking. By organizing financial data in a structured way and generating insightful reports, the system helps users make informed decisions about their money.

## **Future Scope**

While the current implementation meets basic personal finance management needs, there are several areas where the system can be expanded in future versions:

- **Mobile Application Development:** A mobile app version could be developed for easier access on smartphones.
- **Integration with Banks:** The system could be integrated with bank APIs to automatically fetch transaction data.
- **Advanced Analytics:** Machine learning algorithms could be used to provide predictive analysis on spending habits or investment growth.
- **Multi-Currency Support:** The system could support multiple currencies for global use.
- **Investment Portfolio Optimization:** Advanced tools could be added to help users optimize their investment portfolios based on risk profiles.

## **References**

1. Silberschatz, A., Korth, H., & Sudarshan, S., *Database System Concepts*, McGraw-Hill Education.
2. Elmasri R., Navathe S.B., *Fundamentals of Database Systems*, Addison-Wesley.
3. Date C.J., *An Introduction to Database Systems*, Pearson Education.
4. Online resources on DBMS best practices from websites like Stack Overflow and GeeksforGeeks.