PHANAINCE – PROJECT SETUP REPORT

Submitted by:

Aashirwad Pradhan
MCA – Sapthagiri NPS University

Chapter 1

Introduction

The Project Setup Report provides a complete overview of the installation, configuration, and environment preparation steps required to build and run **Phainance**, a full-stack expense tracking system.
This report documents:

- Required software
- Backend setup
- Frontend setup
- Database initialization
- API configuration
- Testing and running the project

The purpose of this document is to ensure the system can be deployed consistently across any machine.

Chapter 2

Software Requirements

2.1 System Requirements

- Windows 10 / 11 or Ubuntu 20+
- Minimum 4 GB RAM
- Minimum 10 GB free storage
- Java JDK 21+
- Node.js 18+
- MySQL Server 8+
- Modern web browser (Chrome/Edge)

2.2 Tools Used

Category	Tools
Backend	Java, Spring Boot, Maven
Frontend	React (Vite), JavaScript
Database	MySQL
Testing	Postman, Browser DevTools
IDE	IntelliJ IDEA, VS Code

Chapter 3

Backend Setup (Spring Boot)

3.1 Installing Java & Maven

- Install **Java JDK 21**
- Verify installation:
- java -version
- mvn -version

3.2 Importing Backend Project

- Open **IntelliJ IDEA**
- Select **Open Project**
- Choose backend/ folder
- Maven automatically resolves dependencies from pom.xml

3.3 Configuring Application Properties

Inside
src/main/resources/application.properties

spring.datasource.url=jdbc:mysql://localhost:3306/expense_tracker

spring.datasource.username=root

spring.datasource.password=YOUR_PASSWORD

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.format_sql=true

spring.main.allow-circular-references=true

3.4 Start the Backend Server

Run:

mvn spring-boot:run

Backend will start on:

http://localhost:8080

Chapter 4

Frontend Setup (React + Vite)

4.1 Installing Node.js

Install Node.js LTS version.

Check installation:

node -v

npm -v

4.2 Install Frontend Dependencies

Inside frontend/:

npm install

4.3 Configure Axios Base URL

Inside:
frontend/src/api/axios.js

```
export default axios.create({
  baseURL: "http://localhost:8080/api",
});
```

4.4 Start Frontend

npm run dev

Frontend runs at:

http://localhost:5173

Chapter 5

Database Setup (MySQL)

5.1 Create Database

Run in MySQL:

```
CREATE DATABASE IF NOT EXISTS expense_tracker;
USE expense_tracker;
```

5.2 Create Tables

```
CREATE TABLE IF NOT EXISTS users (
  id BIGINT AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(100) NOT NULL UNIQUE,
  email VARCHAR(150) NOT NULL UNIQUE,
  password VARCHAR(255) NOT NULL
);

CREATE TABLE IF NOT EXISTS categories (
  id BIGINT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(255) NOT NULL,
  UNIQUE KEY ux_categories_name (name)
);
```

```
CREATE TABLE IF NOT EXISTS expenses (
  id BIGINT AUTO_INCREMENT PRIMARY KEY,
  amount DOUBLE,
  description VARCHAR(255),
  date DATE,
  category_id BIGINT,
  FOREIGN KEY (category_id) REFERENCES categories(id)
);

CREATE TABLE IF NOT EXISTS budgets (
  id BIGINT AUTO_INCREMENT PRIMARY KEY,
  month VARCHAR(20) NOT NULL,
  limit_amount DOUBLE
);
```

5.3 Insert Default Categories

```
INSERT INTO categories (name)
VALUES ('Food'), ('Rent'), ('Utilities'), ('Entertainment'), ('Other')
ON DUPLICATE KEY UPDATE name=name;
```

Chapter 6

API Configuration & Authentication

6.1 JWT Authentication

- Login generates JWT token
- Token stored in localStorage
- Token appended in Axios headers:

```
headers: { Authorization: `Bearer ${token}` }
```

6.2 Important API Endpoints

Module	Method	Endpoint
Auth	POST	/api/auth/login
Auth	POST	/api/auth/signup
Expenses	GET	/api/expenses
Expenses	POST	/api/expenses
Budget	GET	/api/budget
Budget	POST	/api/budget
Categories	GET	/api/categories

Chapter 7

Running the Project

7.1 Start MySQL Server

Ensure MySQL is running.

7.2 Launch Backend

mvn spring-boot:run

7.3 Launch Frontend

npm run dev

7.4 Open the Application

Visit:

http://localhost:5173

Chapter 8

Verification & Testing

Testing Performed

- User signup/login
- Adding expenses
- Editing & deleting expenses
- Setting budget
- Chart & analytics loading
- Category-based filtering
- Notifications & validation

Manual Testing Tools

- Postman
- Browser DevTools
- MySQL Workbench

Chapter 9

Conclusion

The project setup for **Phainance** was successfully completed using modern web development tools. The entire system — frontend, backend, and database — is fully configured for execution.
This structured setup ensures smooth deployment, scalability, and maintainability for future enhancements.