Stock Portfolio Management System - Setup and User Guide

Overview

The **Stock Portfolio Management System** is a Spring Boot application that allows users to manage their stock portfolios. The system includes features for portfolio management, transaction processing, and portfolio analysis.

Features

- **User Authentication**: Secure login and registration
- **Portfolio Management**: Add, update, and delete stocks
- **Transaction Processing**: Buy and sell stocks
- **Portfolio Analysis**: Performance and risk analysis
- **Real-Time Portfolio Valuation**

Technology Stack

- **Backend**: Java 21, Spring Boot, Spring Security, Spring Data JPA
- **Frontend**: Thymeleaf, Bootstrap 5
- **Database**: MySQL 8.0+
- **Build Tool**: Maven

Prerequisites

1. **Java Development Kit (JDK) 21**
   - Download from Oracle JDK or OpenJDK
   - Verify installation: `java -version`
2. **Maven 3.8+**
   - Download from Maven
   - Verify installation: `mvn -version`
3. **MySQL 8.0+**
   - Download from MySQL
   - Start MySQL service:
     - Windows: Use MySQL Workbench or Services
     - Mac: `brew services start mysql`
     - Linux: `sudo systemctl start mysql`
4. **Git**
   - Download from Git
   - Verify installation: `git --version`

Setup Instructions

Step 1: Clone the Repository

```
git clone https://github.com/PES1202203344/OOAD-Mini-Project.git
cd OOAD-Mini-Project
```

Step 2: Configure the Database

1. Ensure MySQL is running.
2. No need to manually create the database; it will be created automatically by the application.

3. Open `src/main/resources/application.properties` and update the database credentials:

   ```
   spring.datasource.url=jdbc:mysql://localhost:3306/portfolio_db?createDatabaseIfNotExist=true
   spring.datasource.username=your_mysql_username
   spring.datasource.password=your_mysql_password
   spring.jpa.hibernate.ddl-auto=update
   spring.jpa.show-sql=true
   ```

Step 3: Build and Run the Application

1. Build the project:

   ```
   mvn clean install
   ```

2. Run the application:

   ```
   mvn spring-boot:run
   ```

3. Access the application at http://localhost:8080.

Step 4: First-Time Usage

1. Register a new user account:
   - Navigate to http://localhost:8080/register.
   - Fill in the registration form with username, email, and password.
2. Log in using your credentials.
3. Start managing your portfolio:
   - Add stocks to your portfolio.
   - Execute buy and sell transactions.
   - Monitor your portfolio's performance on the dashboard.

Project Structure

```
src/main/java/com/portfolio/management/
├── config/           # Security and application configuration files
├── controller/       # MVC controllers for handling requests
├── model/            # Entity classes representing database tables
├── repository/       # Data access interfaces (Spring Data JPA)
├── service/          # Business logic and design pattern implementations
└── util/             # Utility classes for helper methods
```

Design Patterns Used

1. **Singleton Pattern**: For configuration management
2. **Factory Pattern**: For creating transactions
3. **Observer Pattern**: For stock price updates
4. **Strategy Pattern**: For portfolio analysis

Troubleshooting

Common Issues

1. **Database Connection Issues**
   - Ensure MySQL is running.
   - Verify credentials in `application.properties`.
   - Check if port `3306` is open.
2. **Build Failures**
   - Ensure you have JDK 21+ installed.
   - Clean Maven cache:

     ```
     mvn clean install
     ```
3. **Login Issues**

   - Ensure you've registered correctly.
   - Check username/password combination.

4. **Whitelabel Error Page**

   - Ensure all templates are in `src/main/resources/templates`.
   - Check logs for specific errors.

Security Notes

1. Never commit sensitive information like passwords or API keys to GitHub.
2. Use `.gitignore` to exclude sensitive files like `application.properties`:

   ```
   src/main/resources/application.properties
   ```

Pushing to GitHub

1. Initialize Git in your project directory:

```
git init
```

2. Add all files to Git:

```
git add .
git commit -m "Initial commit"
```

3. Add the remote repository:

```
git remote add origin https://github.com/PES1202203344/OOAD-Mini-Project.git
```

4. Push your code to GitHub:

```
git push -u origin main
```