

Stock Portfolio Management System - Complete Setup Guide

This guide provides step-by-step instructions to set up and run the Stock Portfolio Management System on your local machine.

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](#)
- Windows: Install using the MySQL Installer
- Mac: `brew install mysql`
- Linux: `sudo apt install mysql-server`

Project Setup

1. Clone the Repository

```
git clone https://github.com/yourusername/stock-portfolio-management-system.git
cd stock-portfolio-management-system
```

2. Start MySQL Server

- Windows: MySQL service should start automatically after installation
- Mac: `brew services start mysql`
- Linux: `sudo systemctl start mysql`

3. Configure Application

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

```
# Database Configuration (update username and password)
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
```

Note: The database will be created automatically - you don't need to create it manually.

4. Build the Application

```
mvn clean install
```

5. Run the Application

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

6. Access the Application

Open your browser and navigate to: <http://localhost:8080>

First-Time Usage

1. Register a New User

- Navigate to: <http://localhost:8080/register>
- Fill in the registration form with username, email, and password
- Click "Register"

2. Login

- Use your newly created credentials to log in
- Default starting cash is \$10,000

3. Add Stocks to Portfolio

- Click "Add New Stock" button
- Enter stock details:
 - Symbol (e.g., AAPL)
 - Name (e.g., Apple Inc.)

- Quantity
- Purchase Price
- Current Price

4. Perform Transactions

- Navigate to "Transactions" tab
- Select "Buy Stock" or "Sell Stock"
- Enter transaction details

5. View Portfolio

- Return to "Portfolio" tab to view your holdings
- See performance metrics on the Dashboard

Troubleshooting

Database Connection Issues

- Verify MySQL is running
- Check username/password in `application.properties`
- Ensure port 3306 is not blocked by firewall

Application Won't Start

- Verify JDK version: `java -version`
- Check for port conflicts (8080 might be in use)
- Review application logs for specific errors

Login Problems

- Ensure you've registered correctly
- Check username/password combination
- Clear browser cookies and try again

Design Patterns Used

This project demonstrates four key design patterns:

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

Technology Stack

- Backend: Spring Boot, Spring Security, Spring Data JPA
- Frontend: Thymeleaf, Bootstrap 5
- Database: MySQL
- Build Tool: Maven

For additional help or to report issues, please open a GitHub issue in the repository.