Creating a new react project:

* npx create-react-app nap-tracker

Install libraries

* npm install react-router-dom axios recharts bootstrap

Project structure

nap-tracker-dashboard/

├── public/

├── src/

│ ├── components/

│ │ ├── AssetList.js

│ │ ├── PriceChart.js

│ │ └── Sidebar.js

│ ├── pages/

│ │ └── Dashboard.js

│ ├── App.js

│ └── index.js

**Backend**

1. Create a new project
2. Select Java as language
3. Select maven
4. Configure group and artifact ID
5. Configure POM.xml:

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>3.2.4</version> <!-- use the latest stable version -->  
 <relativePath/> <!-- lookup parent from repository -->  
 </parent>  
  
 <groupId>own.naptracker</groupId>  
 <artifactId>backend</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <properties>  
 <maven.compiler.source>8</maven.compiler.source>  
 <maven.compiler.target>8</maven.compiler.target>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 </properties>  
 <dependencies>  
 <dependency>  
 <groupId>jakarta.persistence</groupId>  
 <artifactId>jakarta.persistence-api</artifactId>  
 <version>3.1.0</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-jpa</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.postgresql</groupId>  
 <artifactId>postgresql</artifactId>  
 <scope>runtime</scope>  
 </dependency>  
 </dependencies>  
  
</project>

1. Create and configure application.properties under resources to configure db connection:

spring.datasource.url=jdbc:postgresql://localhost:5432/naptracker  
spring.datasource.username=postgres  
spring.datasource.password=N4sh4mw3@2014@N  
  
  
spring.jpa.hibernate.ddl-auto=none  
spring.jpa.show-sql=true  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect  
  
server.port=8080

1. Create Asset model in model package to test db connection
2. Create repository interface

eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJlbW1lIiwiaWF0IjoxNzQ2MTE0MDYyLCJleHAiOjE3NDYxNTAwNjJ9.jj\_QyXmWavlrxKEdflCfi-yF7a3wj5Fyzjlu\_zrrKxI

Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJlbW1lIiwiaWF0IjoxNzQ2MTE0MDYyLCJleHAiOjE3NDYxNTAwNjJ9.jj\_QyXmWavlrxKEdflCfi-yF7a3wj5Fyzjlu\_zrrKxI

**WHEN THE PROJECT DISAPPEARED:**

1. Go to the project configuration, check if module is not missing
2. If Module is missing, import it and add pom.xml of the project.

**In storing expired tokens in logout, storing keys, token in redis is better practice**

1. Run Docker desktop
2. After the run Docker engine, in terminal irun or install redis: docker run -d --name redis-server -p 6379:6379 redis
3. List all keys stored in redis:

* docker exec -it redis-server redis-cli
* KEYS \*