**Project Setup Guide**

**1. Overview**

This project uses **React** for the Frontend and **Spring Boot** for the Backend.

**2. Requirements**

* **Node.js** (>= 14.x) and **npm** (>= 6.x)
* **Java Development Kit** (JDK 17)
* **Maven** or **Gradle** to build the Backend.
* **IDE**:
  + Visual Studio Code (Frontend)
  + IntelliJ IDEA or Eclipse (Backend)

**3. Frontend Setup**

**3.1. Dependencies**

Key libraries used in the Frontend:

* **react-router-dom**: For routing management.
* **react-bootstrap**: Bootstrap components.
* **Material UI**: UI component library.
* **react-icons**: Icon library.
* **react-select**: Custom dropdown components.
* **axios**: API calls.
* **xlsx**, **exceljs**: Excel file handling.
* **papaparse**: CSV file parsing and processing.
* **file-saver**: Save files from the browser.
* **styled-components**: CSS-in-JS solution.
* **notyf**: Toast notifications.
* **date-fns**: Date formatting and manipulation.
* **eslint**: Code linting.
* **vite**: Fast build tool for React applications.

**3.2. Installation and Run**

1. **Clone the project**:
2. git clone <repository-link>

cd frontend

1. **Install dependencies**:

npm install

1. **Run the application**:

npm run dev

The application will run at http://localhost:3000.

1. **Build for Production**:

npm run build

**4. Backend Setup**

**4.1. Dependencies**

The Backend uses **Spring Boot** and additional libraries:

|  |  |
| --- | --- |
| **Spring Boot** | **External Libraries** |
| Spring Data JPA | ModelMapper |
| Spring Boot DevTools | Lombok |
| Spring Web | JSON Web Token (JWT) |
| Spring Security | Spring Filter |
| Spring Boot Starter Validation |  |

**4.2. Installation and Run**

1. **Clone the project**:
2. git clone <repository-link>

cd backend

1. **Build and run the Backend**:
2. mvn clean install

mvn spring-boot:run

1. **API Server** will run at http://localhost:8080.

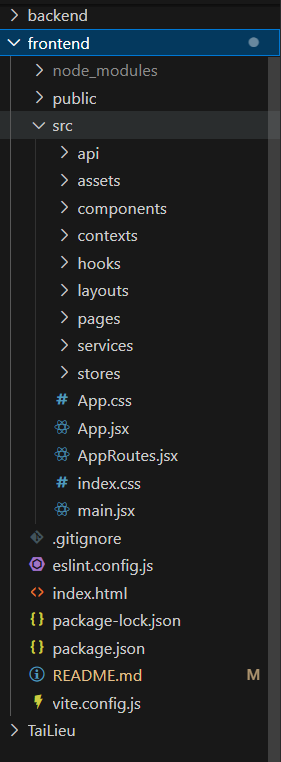
**5. Environment Variables**

Create a .env file for the Frontend:

VITE\_API\_URL=http://localhost:8080/api

A screenshot of a computer

Description automatically generated**6. Project Structure**



**7. Deployment Guide**

**7.1. Frontend Deployment**

1. **Build Frontend**:

npm run build

The build output will be in the dist/ folder.

1. **Deploy to Server**:
   * Use services like **Vercel**, **Netlify**, or upload to a server with **Nginx** installed.
   * Set the build folder as the root directory on the server.

**7.2. Backend Deployment**

1. **Build Backend**:

mvn clean package

A backend.jar file will be generated in the target/ folder.

1. **Deploy to Server**:
   * Upload backend.jar to the server.
   * Run the Backend with the command:

java -jar backend.jar

* + Configure the firewall to open port 8080 or update the port in application.yml.

1. **Configure Reverse Proxy (Nginx)**
   * Install **Nginx** and create a Proxy Pass configuration.
   * Set up **HTTPS** using **Certbot** for your domain.

**Example Nginx Configuration**:

server {

listen 80;

server\_name example.com;

location / {

root /path/to/frontend;

index index.html index.htm;

}

location /api/ {

proxy\_pass http://localhost:8080/api/;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

}

}

1. **Reload Nginx**:

sudo systemctl reload nginx

**8. Run Full Project**

1. **Start the Backend** first.
2. **Run the Frontend** and connect to the API using VITE\_API\_URL.
3. Deploy to the server following the **Deployment Guide**.

**9. References**

* [Spring Boot Documentation](https://spring.io/projects/spring-boot)
* [React Documentation](https://reactjs.org/)
* [Material UI Documentation](https://mui.com/)
* [Axios Documentation](https://axios-http.com/)
* [JWT Guide](https://jwt.io/)
* [Certbot Guide](https://certbot.eff.org/)