Appointment Schedule Project

This is a full-stack web application that facilitates appointment scheduling. The project includes two main parts:

- **Frontend**: Built with React to create a user interface for booking and viewing appointments.
- **Backend**: A Spring Boot application that handles the logic, API endpoints, and database connection.

Project Structure

```
Backend/
    └── Application/
        └── src/
        └── application.properties
Frontend/
    └── Application/
        └── src/
        └── package.json
        └── public/
        └── ...
```

Backend Setup (Spring Boot)

1. Database Configuration

The backend is connected to a MySQL database to store appointments. The database configuration is stored in application.properties under the Backend/Application/src/main/resources/ directory.

Update the database credentials in the application.properties file:

properties

```
spring.datasource.url=jdbc:mysql://localhost:3306/Appointment?useSSL=false&serverTimezone=UTC
spring.datasource.username=root
spring.datasource.password=Dinesh@700
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.jpa.hibernate.ddl-auto=update
spring.datasource.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
spring.datasource.platform=mysql
```

- **spring.datasource.url**: Set the URL to your MySQL database (localhost:3306 is used as an example, you can change it if needed).
- **spring.datasource.username**: MySQL username (default is root).
- **spring.datasource.password**: MySQL password (set to Dinesh@700 as an example).
- **spring.datasource.driver-class-name**: JDBC driver for MySQL.
- **spring.datasource.jpa.hibernate.ddl-auto**: Setting this to update automatically updates the schema

2. Run the pom.xml for Backend Dependencies

Please make sure to run the pom.xml file located in the Backend/Application/ folder using Maven to install all necessary dependencies for the Spring Boot application.

3. Register Admin and Users

- **Admin Registration**: Use @nyu.edu email addresses to register as admin users.
- **User Registration**: Any other email addresses will be treated as regular users.

4. Running the Backend

To run the backend:

1. Navigate to the Backend/Application/ directory.

Run the Spring Boot application using Maven:

./mvnw spring-boot:run

2. The backend will be accessible at http://localhost:8080.

5. MySQL Setup

Make sure MySQL is installed and running. You'll need to create the Appointment database with the following SQL command:

CREATE DATABASE Appointment;

Ensure your database credentials match what is in the application.properties file.

6.User Access Limitation: View Appointments Only for Next Month

In this application, **regular users** can only **view appointments scheduled within the next 30 days** from the current date.

Reason for the Limitation:

This restriction is implemented to:

- **Reduce server load and data overfetching**, especially when the appointment database grows.
- Ensure a **better user experience** by focusing on immediate and upcoming appointments, which are the most relevant for typical users.
- **Prevent users from accessing long-term scheduling data** reserved for administrative planning and system-level forecasting.

Frontend Setup (React)

The frontend is built using React, and it allows users to interact with the backend to view and manage appointments.

1. Install Frontend Dependencies

Navigate to the Frontend/Application/ directory and install the necessary npm packages by running:

npm install

This will install all dependencies listed in the package.json file.

2. Running the Frontend

To run the React application:

1. .After installing the dependencies, start the development server using:
   npm run dev
2. The frontend React application will be available at http://localhost:3000. You can now interact with the backend, making API calls to view or add appointments.

Integration Between Frontend and Backend

1. **Frontend**: The React application communicates with the Spring Boot backend using REST API calls (e.g., GET, POST requests to endpoints like /appointments).

2. **Backend**: The Spring Boot backend handles API requests, connects to the MySQL database, and manages appointment data.

3. **Database**: The MySQL database stores appointment data, with the backend handling CRUD operations through the Java Persistence API (JPA).

Make sure both servers are running:

- Backend: http://localhost:8080

- Frontend: http://localhost:3000

Important Notes

1. **MySQL Setup**: Ensure MySQL is correctly set up, and the Appointment database is created. Make sure the credentials in the application.properties file match your local setup.

2. **Frontend and Backend Communication**: The React frontend sends requests to the Spring Boot backend. Ensure the backend is up and running before accessing the frontend.

3. **Application Deployment**: If deploying, you can configure your application for a production environment, setting the appropriate Mysql database URL and credentials.