Need:

Java jdk, React (Node JS), MySql

IDE: STS and IntelIJ Vs code, Mysql workbench, PostMan

Install from internet:

Install PostMan ---- [https://www.postman.com/downloads/](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqazE4QWVhVmx2clFLcGhLa2xQc2l1dEcwaTdld3xBQ3Jtc0tsTEhFRm9zeld2eHZyY2hXc292elpZdUliaW1XdV8tQWpJVlZEcmlESm9NZG5INGd0ZWp2OGZfUFk1clE5bmY1bmxZTFhUNUlnTHd4bWFpSlh6U3NURVFUTWx6WnFUMUI5WEFwVEJBbDlscmYwSzFNRQ&q=https%3A%2F%2Fwww.postman.com%2Fdownloads%2F&v=4LZKnegAm4g)

------------------------------------------------------------------------------------------------------------------------------------

**Vs code**:

Line format: alt+shif+F

Line comment: //

**React Terminal:**

App Start: npm star ->press Enter

App Stop: ctrl+C -> Y

Clear : ctrl+l

Import react : rfc and press enter.

-----------------------------------------------------------------------------------------------------------------------------------

**1.Create Spring Boot project and packages**

Create using online: search in google-> https://start.spring.io/ -> select and name fill up and also add dependencies -> click generate-> open in Intelij generate file.

**Folder structure in Spring boot:**

under src/main/java/project name package. Create package

1. Model
2. User.java
3. Controller
4. UserController.java
5. Repository
6. UserRepository.java
7. Exception
8. UserNotFound.java
9. UserNotFoundException.java

Step01: create all package and java model class, repository class

Step02: Now connect in database

**Connect database:** Edit application.properties file in spring boot

**2.PostMapping for sending data in database**

Step01: edit userController.java class with @PostMapping annotation ;

Step02: Now check data posted in database using postman. **Run the application and then**

**check by postman:** Go to postman-> Create new collection->Rename->now new Request->add request-> select POST request->input([http://localhost:8080/user](http://localhost:8080/user/1)) -> then select body -> select raw-> select json->inside box writhe json format such as {"name":"..","username":"..", "email":”--” } ->sent click : then data add in database. Check in mysql workbench

**3.@GetMapping for getting data from database**

Step01: edit userController.java class with @GetMapping annotation ;

Step02: Now check data show from database using postman. **Run the application and then**

**check by postman:** Go to postman-> new Request->add request-> select GET request->input([http://localhost:8080/user](http://localhost:8080/user/1)s) -> sent click : then data show from database in post man.

**4.Create React App using npm for frontend**

Step01: **Create React app:** In command Prompts ->Type- cd desktop ->press enter-> type-npx create-react-app projectName-> press enter …

Step02: Now this react app open in VS code.

Step03: Start React app: VS code open terminal-> type- npm start ->press enter open in browser automatic. After Create React app:

**5.Folder Structure for React app**

Folder create under src folder in react app:

1. layout
2. pages
3. users

Now create js file under this folder.

1. layout
2. Navbar.js
3. pages
4. Home.js
5. users
6. AddUser.js
7. EditUser.js
8. ViewUser.js

**6.Create Navbar using Bootstrap and React**

**Now add Library in react:** react sniple.

**Add bootstrap in react:** Go to google-> search npm bootstrap (<https://www.npmjs.com/package/bootstrap>) -> copy install link-> go to react and create new terminal->

Pest copy link in react terminal and press enter now install bootstrap

Using bootstrap from react app: IN App.js file import bootstrap.

* import "../node\_modules/bootstrap/dist/css/bootstrap.min.css";

Step01->Navbar.js (Create a navbar using bootstrap and custom)->import app.js

**7.Create Homepage using Bootstrap table**

Step02->Home.js (Create a navbar using bootstrap and custom)->import app.js

**8.Display Information using AXIOS GET**

**Show date from date base need install npm axios:** Go to google NPM -> from NPM search axios -> copy install link-> pest copy link in react terminal and press enter now install npm axios.

Step01-> Display data using Get Axios in console panel.

Connect font-end react app to backend Spring boot application. In spring boot controller class add @CrossOrigin annotation. Now mapping data in home.js Table. Below show the home JS file.

**9.Configure Routing using REACT-ROUTER-DOM**

Step1-> Now again edit home page and add, edit, delete button.

For react router Dom: Go to google NPM -> from NPM search (react-router-dom)-> copy install link-> pest copy link in react terminal and press enter now install npm react-router-dom.

Step02: import App.js file react router dom.

Step02-> Now add user->custom AddUser.js file, App.js file etc.

**10.Create Register User Form using React**

Step01: custom adduser.jsp

**11.Store User Information inside state**

Step01: custom adduser.jsp, create state, function and etc.

**Add extension React Developer tools:** webstore: (search webstore in browser->select chrome webstore -> select extension-> search: react developer tool -> add extension)

Show input save to userState: inspect react adduser page -> select components in Tab tools-> now input ->

**12.Post data into database AXIOS POST**

Step01: edit adduser.jsp, create function and Navigator etc.

**13.Custom Exception Han.dling using Spring boot**

Step01: create Exception java class and custom in spring boot.

Step02: Get mapping from controller class for exception handling.

Step03: Check in postman: Go to postman-> new Request->add request-> select Get request->input(<http://localhost:8080/user/1>) ->sent click : then show the data by user id 1.

**14.Edit User using PUTMapping**

Step01: Edit UserControl java page with @Putmapping

Step02: check by postman: Go to postman-> new Request->add request-> select PUT request->input(<http://localhost:8080/user/1>) -> then select body -> select raw-> select json->inside box writhe json format such as {"name":"..","username":"..", "email":”--” } ->sent click : then update data by user id 1.

**14.Delete the user of specific id**

Step01: Edit UserControl java page with @DeleteMapping

Step02: Check by postman : Go to postman-> new Request->add request-> select Delect request->input(<http://localhost:8080/user/1>) -> sent click.

Then see the database corresponding user id deleted.

**15.Edit user using AXIOS PUT**

Step01: AppUser.js file copy and pest in EditUser.js file and now custom this file.

**16.DELETE User using AXIOS**

Step01: Edit Home.js file also add deleteUser function.

**17.View Information of specific User**

Step01: Edit viewUser.js file

-----------------------------------------------------------------------------------------------------------------------------------

**All file Give Below:**

|  |  |
| --- | --- |
| **React File Frontend** | **Spring boot backend file** |
|  |  |