JOB QUEST: Online Job Portal ForEmployeers And Job Seekers

Course Name: **DAC**

NAME	PRN
PALLAVI ANNAVAJJALA	**200250120009**
LAKSHITA SAHU	**200250120045**
RIA BHADANI	**200250120076**
SAUMYA AGARWAL	**200250120087**
TUNISHA UJAWANE	**200250120101**

WHAT OUR PROJECT DOES?

- Job Quest is a platform for employment opportunities.
- It is an online portal for both job seekers and employers.
- Anyone looking for a job can check the available opportunities and apply using various filters.
- Recruiters can update the job listings according to their requirements.
- From contract positions to full time opportunities, anything can be listed. It can also be used for finding people to collaborate on projects.
- There are filters like job profile, experience, skills, company, job type and location to make the search easy.
- All the required details about job seekers and employers will be available.
- We will be using Angular along with Java for creating this job portal.

WORK FLOW

1. Click on filter button on home page. A dropdown menu opens. filter is a child component on home component.

2. Select any value from the given values. Here we have selected Designation.

3. After clicking on Designation, we are saving the value "designation" in this.y.filtertype(y is the service object) using the function getFiltertype(filtyp:string)on filter component.

4. After selecting Designation, a nested drop-down opens. We can select different designations from it. Suppose if we select Software developer.

5. Sofware developer is stored in this.y.filtervalue upon clicking on "software developer" by calling the function search()on filter component.

6. Inside search function we will navigate to joblisting component. On job-listing page only those records will get displayed which have positionnname as software developer.

7. From joblisting component we can apply to a new job. For that we need to click on the jobid of the given job.

8. In the apply() function of joblisting component, we are cehcking if the user is logged in or not using the sessiion variable.

9. If the user is logged in, we are redirecting to jobdescription page. In job description, we can click on apply button.

10. In jobdescription component, we will be redirected to jobseeekerapp and asked to fill in the form to apply on a given job.

11. After filling the form fields and clicking on apply button.

12. On job seekerapp component, apply button will call submit() function.

Service Layer

1. From angular's service layer function getjoblist(), we are calling "/designationfilter" on port 7001 on Controller of Springboot.

2. The function designationfilter() is using get method, taking one input, that is designation value and returning List<Position>. In our case input is "software developer".

3. The above function is calling designationMultiSelectPosition() of service class Positionmodel.

4. From designationMultiSelectPosition() we are calling our repository function designationfiter().

DAO layer

, 1. The designation filter is running a query and returning only those fields where posiitonname is "software developer.

SOME SCENARIOS WHERE WE GOT STUCK AND HOW DID WE OVERCOME

- **IN FROTEND:**
  - **Model mismmatch**. The service was returning an object array but the calling function was expecting an object.
    Solution: Changed the expected value to object array.

  - ***ngFor** not displaying the fields in the drop-down list from database.
    Solution: Constructors, getters and setters were missing in the entity class of Springboot. Adding them solved the problem.

  - How to do interaction between two components in Angular:
    Solution:Did a POC on (iii.) And (iv.) There are 4 ways-
    1. Parent component to Child component using @Input Decorator
    2. Child component to Parent component using @Output Decorator
    3. Using service layer function based on ‘id selection’
    4. Sending data to the other component using ‘queryParams’ in component.ts
- **IN BACKEND:**

- Spring Boot was creating its own table name (Table_name) if the name is kept in CamelCase (tableName) in the database due to which the columns were not mapped to the database table.
    1. Hibernate maps field names using a physical strategy and an implicit strategy which Spring Boot was not implementing by default. So added these two strategies in application.properties in src/main/resources.
    2. spring.jpa.hibernate.naming.implicit-strategy=org.hibernate.boot.model.naming.ImplicitNamingStrategyLegacyJpaImpl
    3. spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl.
- One of the table names was position. It was giving error while performing any queries on this table because MySQL already has a function POSITION ().
  Solution:Changed the table name to oposition.

- How to Authenticate User credentials and allow access to certain functions only when logged in to their respective accounts (Job Seeker and Recruiter).

Solution :The above strategy was not implemented instead separate links wereprovided for a user based on his role by using Session Storage.

WHAT WERE YOUR LEARNINGS DURING THIS PROJECT?

- We learnt to develop Full Stack application and learnt how Presentation Layer interacts with the Service Layer.
- We have learnt how to use component talking, routing between two components.
- How to Upload a file/image and deciding on the datatypes in MySql to match with Hibernate table columns.
- How to Authenticate User Credentials when the user logins to respective account using Session Storage present in lib.dom.d.ts .
- How to implement joins on two tables in Hibernate.

- Implementing the entire three-tier architecture using Angular as Presentation Layer, Spring Boot as Service Layer and Hibernate as DAO Layer Framework.
- Doing a small POC first before applying the logic in the project code.
- Always first check the value returned by a function by writing console.log() and looking at Console in the Developer Tools of the browser window.
- Scheduling and prioritizing tasks using JIRA by creating sprints.