

---

# System Requirements Specification Index

For

## Job Search Application

Version 1.0

IIHT Pvt. Ltd.  
fullstack@iiht.com

## Contents

1	Business-Requirement:	3
1.1	Problem Statement:	3
1.1.1	Job Search Application	3
2.	Template Code Structure	
2.1	Job Controller	
3.	Resources AVAILABLE:	4
4	Suggested WIREFRAMES:	5
5	Business Validations	9
6	Considerations	9
7	Execution Steps to Follow	9

# 1 BUSINESS-REQUIREMENT:

---

## 1.1 PROBLEM STATEMENT:

The purpose of this application is to allow an admin to list out all jobs with different details for job seekers.

### 1.1.1 Job Search Application:

The Job Search Application allows you to:

1. Access the home page (**must show option to add job, search by job name or its description, add any job as favorite and show list of available jobs with number of pages for pagination**).
2. Should be able to add a new job.
3. It can have basic fields like Job Title, Job Description, Minimum Experience & Salary.
4. Should be able to get the list of jobs (**should return data in pages and must have buttons as Asc and Desc in each column to sort list ascending or descending respectively**).
5. Should be able to edit, delete and add to favourite any job.
6. Should be able to search any job on a name or description basis. (**should be implemented using custom query**).
7. Should be able to see all favorite jobs.

## 2. TEMPLATE CODE STRUCTURE:

### 2.1 JOB CONTROLLER

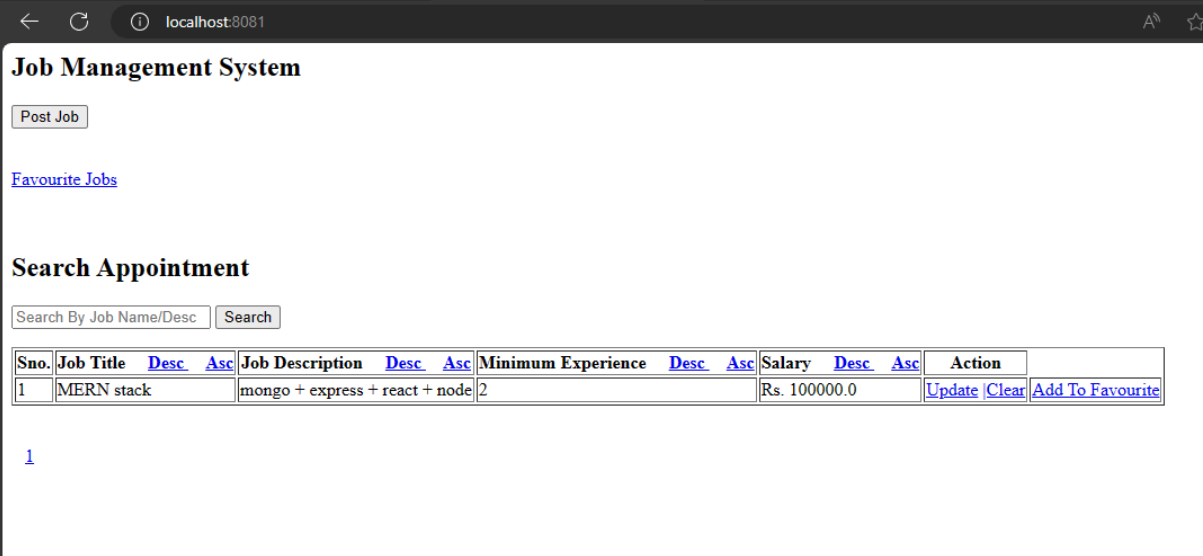
Method Exposed	Purpose
listJobs()	Should return page "list-jobs" with required data.
showFormForAdd()	Should return page "job-add" for adding a Job.
saveJob()	Should save a Job and return "job/list" with required data.
showFormForUpdate()	Should show job details in the page "job-add" to edit a job.
showFormForDelete()	Should delete a job and return "job/list" with required data.
updateIsFav()	Should add that job as favorite and redirect to "job/list".
searchAppointment()	Should search any job out of favorite jobs and return "fav-jobs" page.

## 3. RESOURCES AVAILABLE:

Description	View Pages Name	Remarks
Common UI		
Home Page	list-jobs	Contains a homepage which shows a list of all jobs along with options to add, edit and delete a job.
All Jobs	list-jobs	
Add a job	job-add	
Update a job	job-add	
Favorite jobs	fav-jobs	

## 4 SUGGESTED WIREFRAMES:

### 1. Homepage – Visitor Landing Page



The screenshot shows the homepage of a 'Job Management System' in a web browser. The browser address bar shows 'localhost:8081'. The page has a dark header with navigation icons. Below the header, there's a 'Post Job' button and a link to 'Favourite Jobs'. A 'Search Appointment' section contains a search bar with the placeholder 'Search By Job Name/Desc' and a 'Search' button. Below the search bar is a table with job listings. The table has columns for 'Sno.', 'Job Title', 'Job Description', 'Minimum Experience', 'Salary', and 'Action'. The first row shows a job with 'Sno.' 1, 'Job Title' 'MERN stack', 'Job Description' 'mongo + express + react + node', 'Minimum Experience' 2, and 'Salary' 'Rs. 100000.0'. The 'Action' column for this row contains links for 'Update', 'Clear', and 'Add To Favourite'. Below the table is a pagination link '1'.

Job Management System

Post Job

[Favourite Jobs](#)

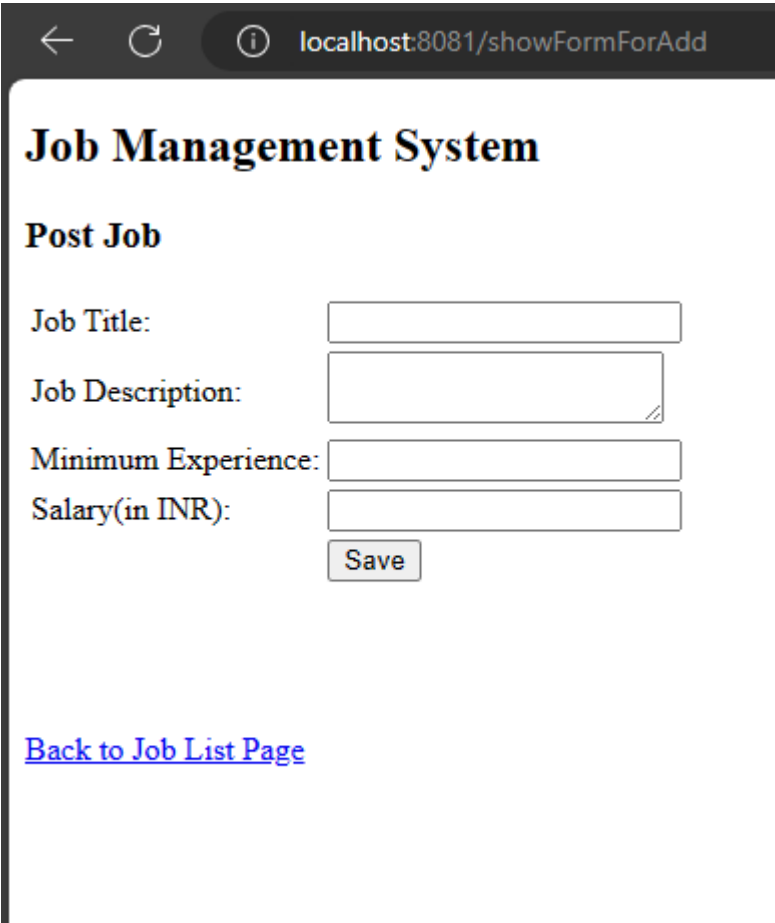
Search Appointment

Search By Job Name/Desc

Sno.	Job Title	Job Description	Minimum Experience	Salary	Action
1	MERN stack	mongo + express + react + node	2	Rs. 100000.0	<a href="#">Update</a> <a href="#">Clear</a> <a href="#">Add To Favourite</a>

[1](#)

### 2. Create a Job



The screenshot shows the 'Create a Job' form in a web browser. The browser address bar shows 'localhost:8081/showFormForAdd'. The page has a dark header with navigation icons. Below the header, there's a 'Job Management System' title and a 'Post Job' section. The 'Post Job' section contains four input fields: 'Job Title:', 'Job Description:', 'Minimum Experience:', and 'Salary(in INR):'. Below these fields is a 'Save' button. At the bottom of the form is a link 'Back to Job List Page'.

Job Management System

Post Job

Job Title:

Job Description:

Minimum Experience:

Salary(in INR):

[Back to Job List Page](#)

### 3. Edit a Job

← ↻ ⓘ localhost:8081/job/showFormForUpdate?jobId=2

## Job Management System

### Post Job

Job Title:

Job Description:

Minimum Experience:

Salary(in INR):

[Back to Job List Page](#)

### 4. Favorite Jobs

← ↻ ⓘ localhost:8081/searchFav

## Job Management System

### Favourite Appointments

Search By Job Name/Desc

Sno.	Job Title <a href="#">Desc</a> <a href="#">Asc</a>	Job Description <a href="#">Desc</a> <a href="#">Asc</a>	Minimum Experience <a href="#">Desc</a> <a href="#">Asc</a>	Salary <a href="#">Desc</a> <a href="#">Asc</a>	Action
1	MERN stack	mongo + express + react + node	2	Rs. 100000.0	<a href="#">Remove Favourite</a>

1

[Back to Job List Page](#)

## 5 BUSINESS VALIDATIONS

---

1. Id must be of type id.
2. JobTitle value not null, min 2 and max 40 characters.
3. Experience value not null
4. Job Description value not null and min 2 and max 40 characters.
5. Salary value not null.
6. isFav value not null.

## 6 CONSIDERATIONS

---

The Code template already contains skeleton methods for service and controller layer. Please write your logic in it.

## 7 EXECUTION STEPS TO FOLLOW

---

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to Application menu  
(Three horizontal lines at left top) → Terminal → New Terminal
3. To build your project use command:  
**mvn clean package -Dmaven.test.skip**
4. To launch your application:  
**java -jar springboot-job-service-0.0.1-SNAPSHOT.war**
5. This editor Auto Saves the code
6. If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use **CTRL+Shift+B**-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
7. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
8. To test any Restful application, the last option on the left panel of IDE, you can find ThunderClient, which is the lightweight equivalent of POSTMAN.
9. This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.

**Note: The application will not run in the local browser**

**10. Default credentials for MySQL:**

- a. Username: **root**
- b. Password: **pass@word1**

**11. To login to mysql instance: Open new terminal and use following command:**

- a. **sudo systemctl enable mysql**
- b. **sudo systemctl start mysql**
- c. **mysql -u root -p**

The last command will ask for password which is '**pass@word1**'

**12. Mandatory: Before final submission run the following command:**

**mvn test**

**13. You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**