

# JOB PORTAL

## Introduction:

Job Portal is an application which is specifically build for job purpose i.e., to search jobs, to find the candidates for a particular designation. This application is usefull for job seekers and the HR managers. The user can create the profile as employee or the recruiter. The employees can upload their resumes and update the profile with necessary details. Employees can also apply the for the job created by the recruiter. Similarly the recruiter can create a job with the required details of the job and publish it, so that the job is visible to all the job seekers.

## Application Description:

Job Portal application is build using node js. REST API's are build in node js to perform various functionaliy of the application.  
The database used here is MongoDB.

## API's:

1.

**API Name:** /api/v1/register

**Description:** The user needs to sign up an account. Based on the type of the user (employee/recruiter) the register screen will be displayed.

**API Type:** POST

2.

**API Name:** /api/v1/login

**Description:** The user gives the username, password and user type (employee/recruiter) to login the account.

**API Type:** POST

3.

**API Name:** /api/v1/updateprofile

**Description:** Based on the type of user the update screen is displayed. If the user is an employee then update resume, skills, education..etc, if the user is an recruiter then the update profile screen askes for creating a job, search resumes..etc.

**API Type:** POST

4.

**API Name:** /api/v1/getdetails

**Description:** Once the user logs in the details of the user which is already saved is fetched and returned to the UI.

**API Type:** GET

5.

**API Name:** /api/v1/getdetails

**Description:** Once the user logs in the details of the user which is already saved is fetched and returned to the UI.

**API Type:** GET

6.

**API Name:** /api/v1/uploadresume

**Description:** A employee can upload their resume by a button click.Initially the file is stored in a tempfolder and then it is renamed with the unique userid and it is moved to /uploads/employee/resume.

**API Type:** POST

7.

**API Name:** /api/v1/uploadphoto

**Description:** A user can upload their photo by a button click.Initially the file is stored in a tempfolder and then it is renamed with the unique userid and it is moved to /uploads/usertype/photo.

**API Type:** POST

8.

**API Name:** /api/v1/deleteaccount

**Description:** The userid is passed to the api to delete the corresponding account.

**API Type:** POST

9.

**API Name:** /api/v1/createjob

**Description:** The recruiter can create a job by providing the details to create a job like organisation,job description, skills, years of experience...etc

**API Type:** POST

10.

**API Name:** /api/v1/searchresumes

**Description:** The recruiter can search the employee's profile using skills or designation.

**API Type:** GET

11.

**API Name:** /api/v1/sendmail

**Description:** The recruiter can send a mail to the employee regarding the job in a button click. From the UI the employee's mail is send to the api and using the portal's mail the job application is sent to the employee's mail.

**API Type:** POST

12.

**API Name:** /api/v1/sendmessage

**Description:** The recruiter can send the job application inside the job portal itself.The job description is sent to the employee's profile by the recruiter.

**API Type:** POST

13.

**API Name:** /api/v1/viewapplication

**Description:** The user can view the job application status.The employee can see the status of the job application which he has applied, similarly the recruiter can view the status of the job application which he has sent to the employee.

**API Type:** GET

14.

**API Name:** /api/v1/changestatus

**Description:** The user can change the application status from viewed, to applied, to accepted.

**API Type:** *POST*

15.

**API Name:** /api/v1/searchjobs

**Description:** The employee can search the jobs based on the skills, designation or location.

**API Type:** *GET*

16.

**API Name:** /api/v1/applyjob

**Description:** The employee can apply a job. By selecting a job the job id is passed to the api and the data job is applied.

**API Type:** *POST*

**Database Name:**

JOB\_PORTAL

**Collections Used:**

1.

**Collection Name:** account\_details

**Description:** To store the user account details.

2.

**Collection Name:** job\_details

**Description:** To store the jobs created by the recruiters.

3.

**Collection Name:** message\_details

**Description:** To store the messages between the employee and the recruiter of the job portal.

4.

**Collection Name:** appliedjob\_details

**Description:** To store the job applications and their respective details.