2022

### DBMS Project – Online Job Portal

Submitted By:

MAYANK KUMAR SINGH

PES2UG20CS189

SEC: C

### Contents

1. Short Description and Scope of the Project	3
Overview	3
Features	3
2. ER Diagram	4
3. Relational Schema	5
4.DDL statements - Building the database	6
5. Populating the Database	12
6. Join Queries	14
7.Aggregate Functions	16
8.Set Operations	18
9. Functions and Procedures	
10.Triggers and Cursors	24
11.Developing a Frontend	

### 1. Short Description and Scope of the Project

### Overview:

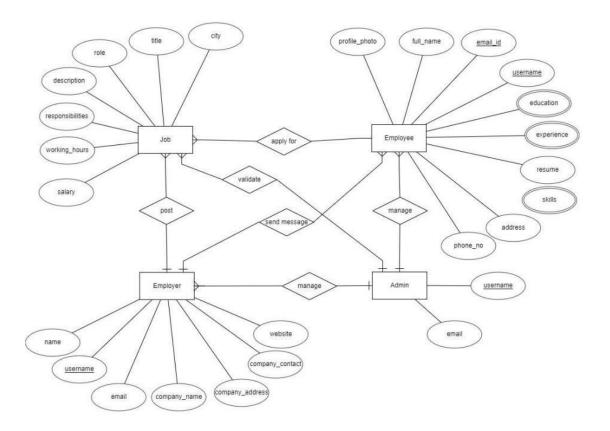
Job Portal is an application which helps the job seeker to find jobs of their interest with advanced search and help employers find the right candidate.

This system uses MySQL connected to a localhost using XAMPP This data can only be accessed by respective users namely admins have the highest privilege and can have access to all the data and users have only selected functionalities.

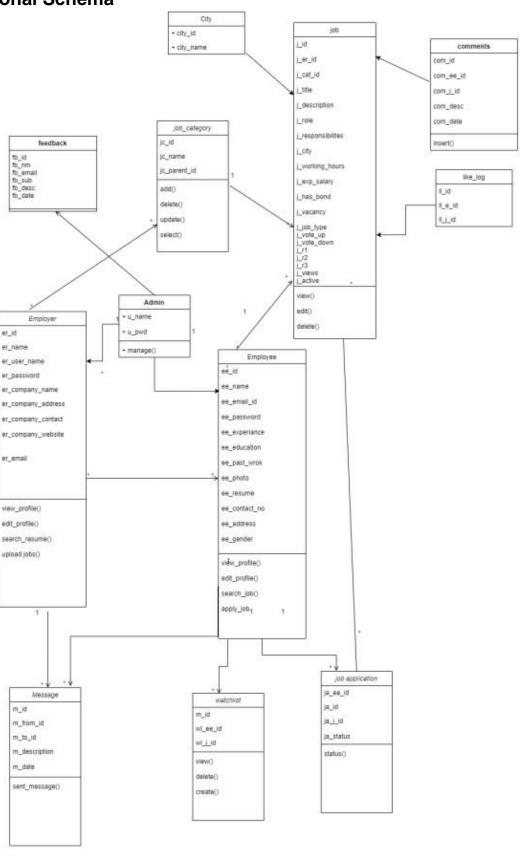
### Features:

- Companies can register themselves on the Job Portal and then the admin has to Reject or Approved the registered Companies to be on the Job Portal.
- After approval, Companies can login with their credentials and create Job Posts.
- Candidates can register themselves on the Job Portal and then can login again with their credentials and apply for the various Job Posts by different companies.
- The Companies can download the Resumes and look at the profile of the candidate who has applied for the Job.
- The Companies can then Reject the Application or Put it under review.
- The Companies and Candidates can contact with each other through Mailbox on the website itself.
- The admin has the complete view of Companies, Applications and Active Jobs.

### • 2. ER Diagram



### 3. Relational Schema



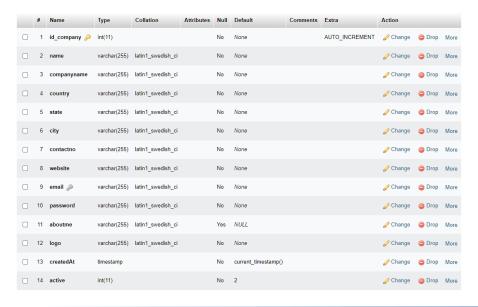
### 4. DDL statements - Building the database

```
CREATE TABLE `admin` (
 `id_admin ` int(11) NOT NULL,
 'username' varchar(255) NOT NULL,
 `password` varchar(255) NOT NULL
);
 SELECT * FROM `admin`
 Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]
  ☐ Show all Number of rows: 25 ➤ Filter rows: Search this table
Extra options
                  ▼ id_admin username password
 ☐ / Edit 3-t Copy 		 Delete 		 2 admin@gmail.com roottoor
CREATE TABLE `apply_job_post` (
 'id_apply' int(11) NOT NULL,
 'id_jobpost' int(11) NOT NULL,
 'id_company' int(11) NOT NULL,
 `id_user` int(11) NOT NULL,
 `status` int(11) NOT NULL
);
 # Name Type Collation Attributes Null Default Comments Extra Action
 □ 1 Id_apply → Int(11) No None AUTO_INCREMENT → Change → Drop More
 □ 2 id_jobpost int(11) No None 

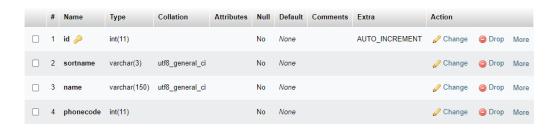
Change © Drop More
 3 id_company int(11) No None
 ☐ 5 status int(11) No 0
 Check all With selected Browse / Change Drop Primary | Unique | Index | Spatia
CREATE TABLE `cities` (
 'id' int(11) NOT NULL,
 `name` varchar(30) NOT NULL,
 `state id` int(11) NOT NULL
);
```



CREATE TABLE `company` ( `id\_company` int(11) NOT NULL, 'name' varchar(255) NOT NULL, `companyname` varchar(255) NOT NULL, `country` varchar(255) NOT NULL, `state` varchar(255) NOT NULL, `city` varchar(255) NOT NULL, `contactno` varchar(255) NOT NULL, 'website' varchar(255) NOT NULL, 'email' varchar(255) NOT NULL, 'password' varchar(255) NOT NULL, `aboutme` varchar(255) NOT NULL `logo` varchar(255) NOT NULL, `createdAt` timestamp NOT NULL, `active` int(11) NOT NULL );



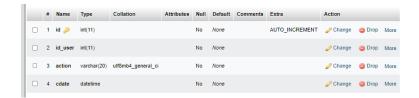
```
CREATE TABLE `countries` (
  `id` int(11) NOT NULL,
  `sortname` varchar(3) NOT NULL,
  `name` varchar(150) NOT NULL,
  `phonecode` int(11) NOT NULL
);
```



# CREATE TABLE `job\_post` ( `id\_jobpost` int(11) NOT NULL, `id\_company` int(11) NOT NULL, `jobtitle` varchar(255) NOT NULL, `description` text NOT NULL, `minimumsalary` varchar(255) NOT NULL, `maximumsalary` varchar(255) NOT NULL, `experience` varchar(255) NOT NULL, `qualification` varchar(255) NOT NULL, `createdat` timestamp NOT NULL );



```
CREATE TABLE `logs` (
  `id` int(11) NOT NULL,
  `id_user` int(11) NOT NULL,
  `action` varchar(20) NOT NULL,
  `cdate` datetime NOT NULL
);
```



## CREATE TABLE `mailbox` ( `id\_mailbox` int(11) NOT NULL, `id\_fromuser` int(11) NOT NULL, `fromuser` varchar(255) NOT NULL, `id\_touser` int(11) NOT NULL, `subject` varchar(255) NOT NULL, `message` text NOT NULL, `createdAt` timestamp NOT NULL );



```
CREATE TABLE `reply_mailbox` (
  `id_reply` int(11) NOT NULL,
  `id_mailbox` int(11) NOT NULL,
  `id_user` int(11) NOT NULL,
  `usertype` varchar(255) NOT NULL,
  `message` text NOT NULL,
  `createdAt` timestamp NOT NULL
);
```



### CREATE TABLE `states` ( `id` int(11) NOT NULL, `name` varchar(30) NOT NULL, `country\_id` int(11) NOT NULL, );



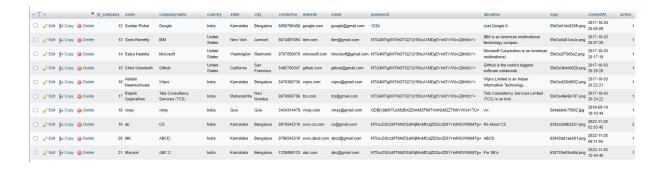
#	Name	Туре	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	id_user 🔑	int(11)			No	None		AUTO_INCREMENT	Change	Drop	More
2	firstname	varchar(255)	latin1_swedish_ci		No	None			Change	Drop	More
3	lastname	varchar(255)	latin1_swedish_ci		No	None			Change	Drop	More
4	email 🔑	varchar(255)	latin1_swedish_ci		No	None			Change	Drop	More
5	password	varchar(255)	latin1_swedish_ci		No	None			Change	Drop	More
6	address	text	latin1_swedish_ci		Yes	NULL			Change	Drop	More
7	city	varchar(255)	latin1_swedish_ci		Yes	NULL			Change	Drop	More
8	state	varchar(255)	latin1_swedish_ci		Yes	NULL			Change	Drop	More
9	contactno	varchar(255)	latin1_swedish_ci		Yes	NULL			Change	Drop	More
10	qualification	varchar(255)	latin1_swedish_ci		Yes	NULL			Change	Drop	More
11	stream	varchar(255)	latin1_swedish_ci		Yes	NULL			Change	Drop	More
12	passingyear	varchar(255)	latin1_swedish_ci		Yes	NULL			Change	Drop	More
13	dob	varchar(255)	latin1_swedish_ci		Yes	NULL			Change	Drop	More
14	age	varchar(255)	latin1_swedish_ci		Yes	NULL			Change	Drop	More
15	designation	varchar(255)	latin1_swedish_ci		Yes	NULL			Change	Drop	More
16	resume	varchar(255)	latin1_swedish_ci		Yes	NULL			Change	Drop	More
17	hash	varchar(255)	latin1_swedish_ci		Yes	NULL			Change	Drop	More
18	active	int(11)			No	1			Change	Drop	More
19	aboutme	text	latin1_swedish_ci		No	None			Change	Drop	More
20	skills	text	latin1_swedish_ci		Yes	NULL			Change	Drop	More

### 5. Populating the database

INSERT INTO users(firstname, lastname, email, password, address, city, state, contactno, qualification, stream, passingyear, dob, age, designation, resume, hash, aboutme, skills) VALUES ('\$firstname', '\$lastname', '\$email', '\$password', '\$address', '\$city', '\$state', '\$contactno', '\$qualification', '\$stream', '\$passingyear', '\$dob', '\$age', '\$designation', '\$file', '\$hash', '\$aboutme', '\$skills');



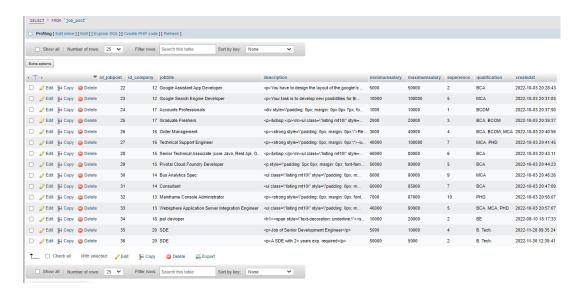
INSERT INTO company(name, companyname, country, state, city, contactno, website, email, password, aboutme, logo) VALUES ('\$name', '\$companyname', '\$country', '\$state', '\$city', '\$contactno', '\$website', '\$email', '\$password', '\$aboutme', '\$file');



INSERT INTO apply\_job\_post(id\_jobpost, id\_company, id\_user) VALUES ('\$\_GET[id]', '\$id\_company', '\$\_SESSION[id\_user]');



INSERT INTO job\_post(id\_company, jobtitle, description, minimumsalary, maximumsalary, experience, qualification) VALUES (?,?,?,?,?,?));



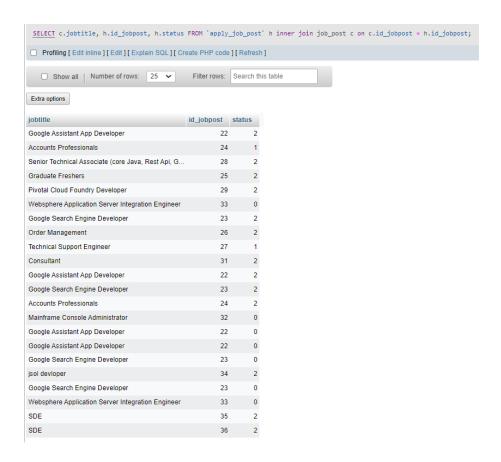
INSERT INTO mailbox (id\_fromuser, fromuser, id\_touser, subject, message) VALUES ('\$\_SESSION[id\_company]', 'company', '\$to', '\$subject', '\$message');



INSERT INTO reply\_mailbox (id\_mailbox, id\_user, usertype, message) VALUES ('\$\_POST[id\_mail]', '\$\_SESSION[id\_company]', 'company', '\$message');

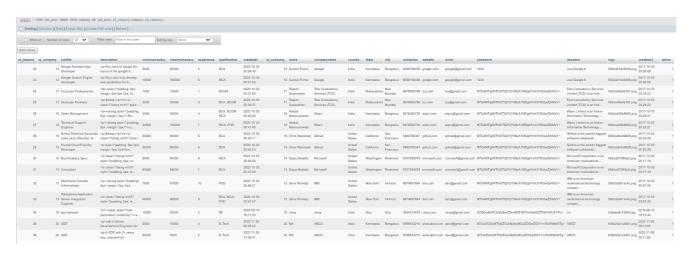
### 6. Join Queries

### 1. Show list of jobs with job title and status



### 2. Show Job Post Details with Company Details

SELECT \* FROM job\_post INNER JOIN company ON job\_post.id\_company=company.id\_company



### 3. LEFT OUTER JOIN

SELECT \* FROM job\_post LEFT OUTER JOIN company ON job\_post.id\_company=company.id\_company;



### 4. CROSS JOIN

SELECT \* FROM job\_post CROSS JOIN company ON job\_post.id\_company=company.id\_company;

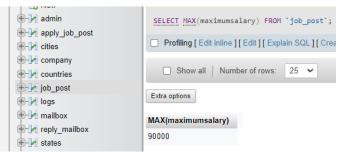


### 7. Aggregate Functions

### 7.1 Show Minimum in Minimum Salary



### 7.2 Show Maximum in Maximum Salary



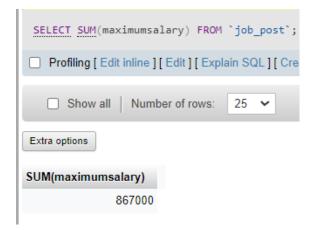
### 7.3 Show Count of Jobs



### 7.4 Show Average Maximum Salary



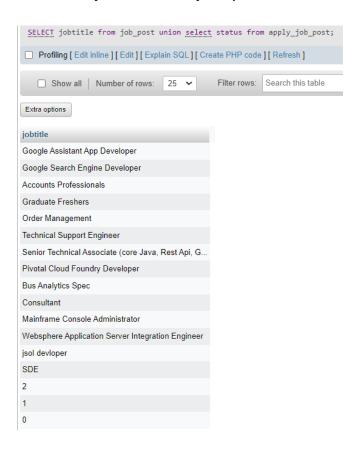
### 7.5 Show Total Maximum Salary



### 8. Set Operations

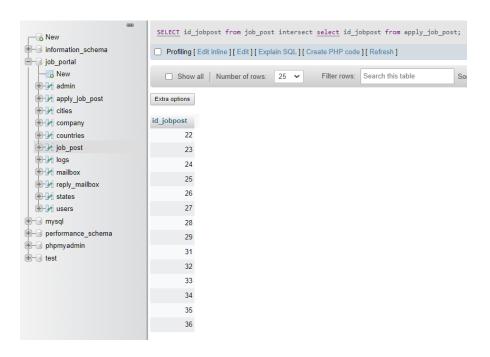
### 8.1 UNION

SELECT jobtitle from job\_post union select status from apply\_job\_post;



### 8.2 INTERSECT

SELECT id\_jobpost from job\_post intersect select id\_jobpost from apply\_job\_post;



### 8.3 UNION ALL

SELECT id\_jobpost from job\_post union all select id\_jobpost from apply\_job\_post;



### 8.4 EXCEPT

SELECT id\_jobpost from job\_post except select id\_jobpost from apply\_job\_post;



### 9. Functions and Procedures

Displays years.

### **FUNCTION**

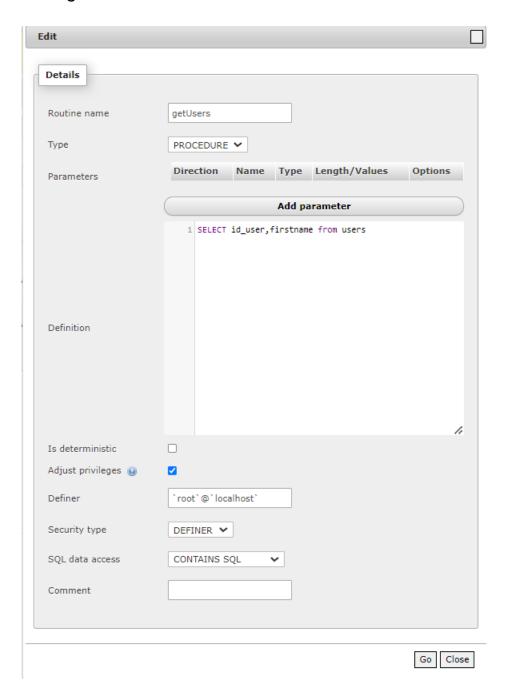


### **OUTPUT**

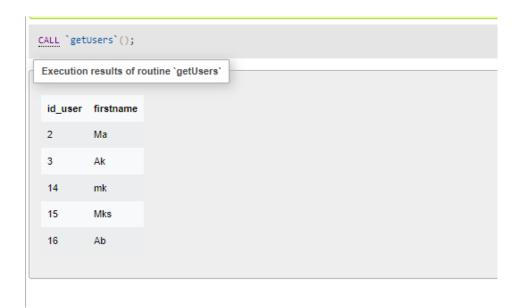


### **PROCEDURE**

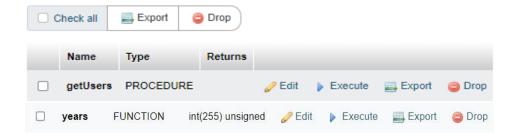
Getting id and first name from users table.



### OUTPUT



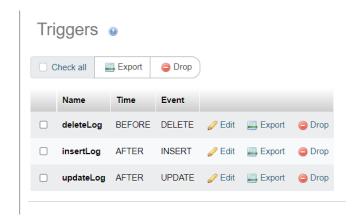
### Routines @

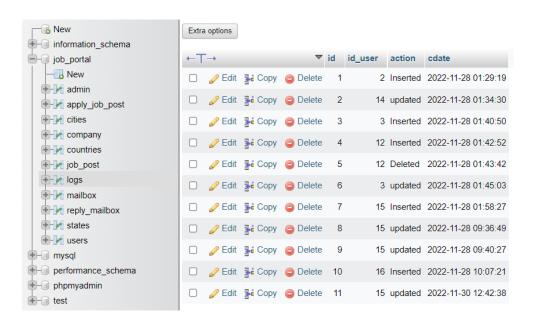


### 10. Triggers and Cursors

### **TRIGGER**

Objective: This trigger adds a log file when a new user has been added to the database.





### **CURSOR**

Objective: When the job post has a status of available, it shows the availability in the database

CREATE PROCEDURE 'getStatus'() NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER BEGIN DECLARE stat TEXT;

DECLARE stat\_cursor CURSOR FOR SELECT status FROM 'job\_post'; OPEN stat\_cursor; FETCH FROM stat\_cursor INTO stat; IF stat="Available" THEN SELECT stat; END IF; CLOSE stat\_cursor; END



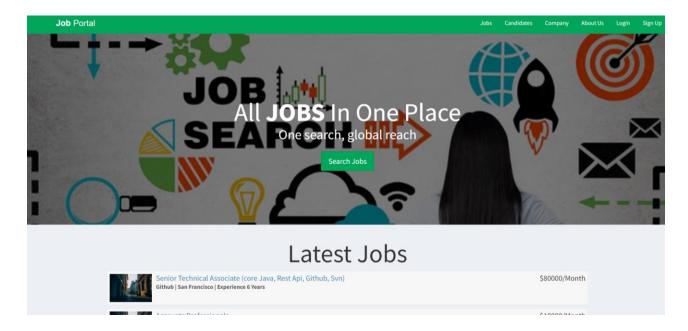
### 11. Developing a Frontend

Higher Level Programming - A simple frontend that is in sync with the backend database is required to be developed . HTML, CSS, PHP and MySQL have been used here.

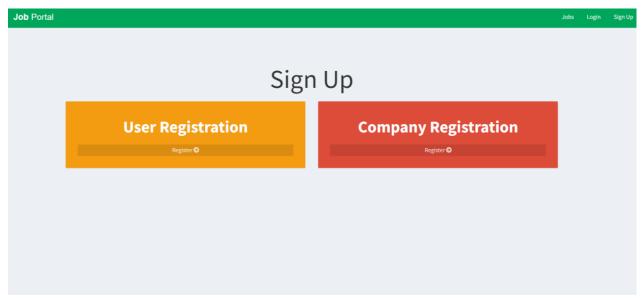
### The frontend supports:

- 1. **Addition, Modification and Deletion** of records from any chosen table company, users, etc
- 2. There is a window to accept and run any SQL statement and display the result.

### 11.1 HOMEPAGE

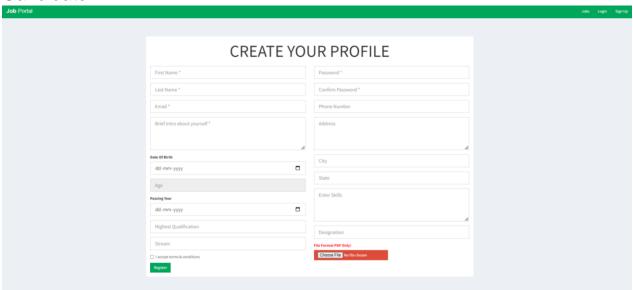


### 11.2 a) Candidate and Company Sign Up Pages:

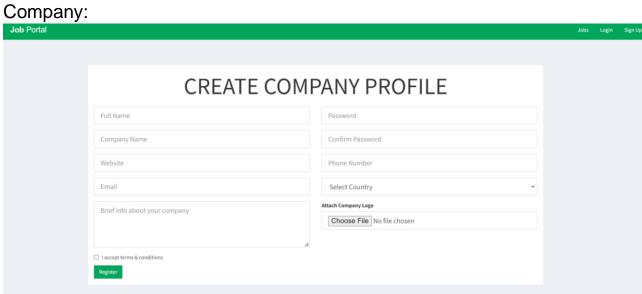


Copyright © 2022-2023 Job Portal. All rights reserved.

### Candidate:

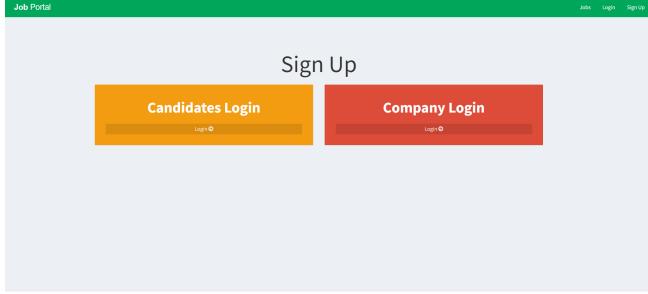


Copyright © 2022-2023 Job Portal. All rights reserved.



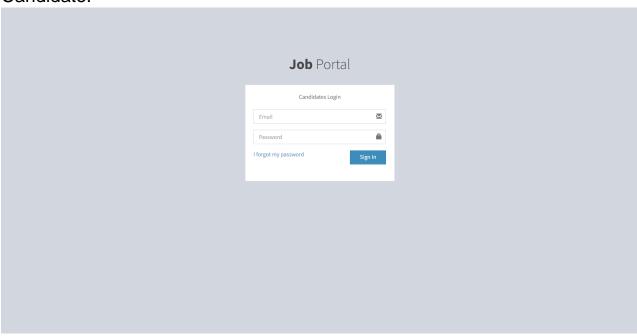
Copyright © 2022-2023 Job Portal. All rights reserved.

### b) Candidate and Company Login Pages:

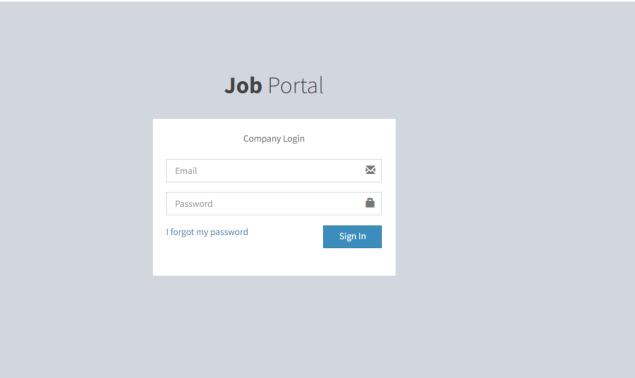


Copyright © 2022-2023 Job Portal. All rights reserved.

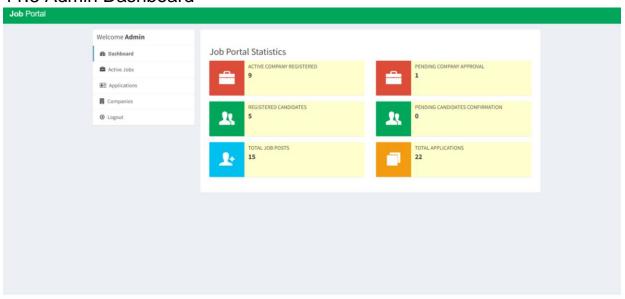
### Candidate:



### Company:

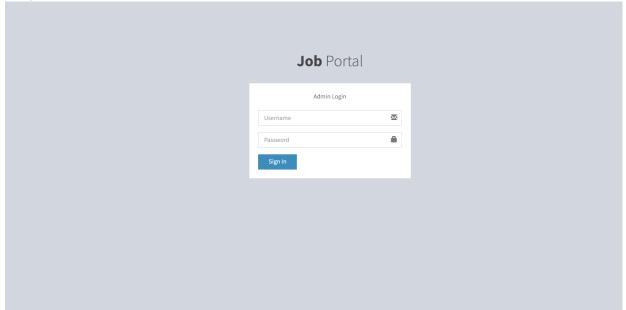


### 11.3 Admin Dashboard

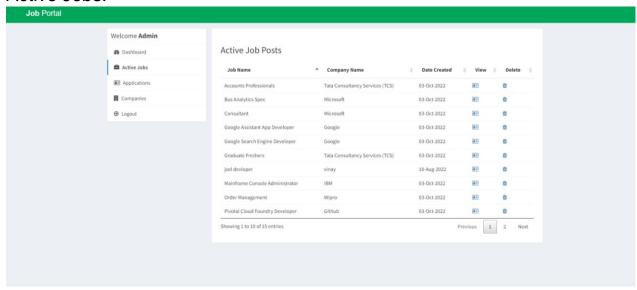


Copyright © 2022-2023 Job Portal. All rights reserved.

### Login:

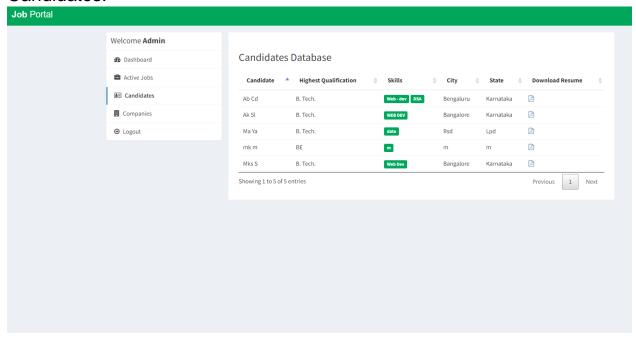


### **Active Jobs:**



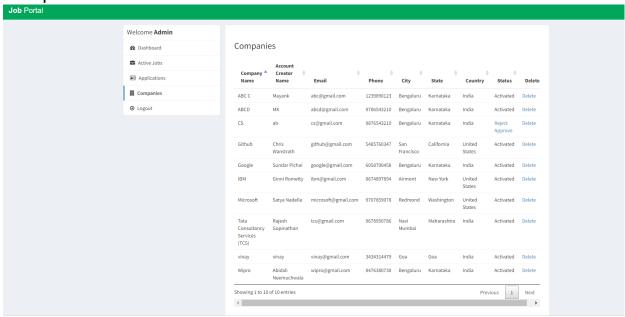
Copyright © 2022-2023 Job Portal. All rights reserved.

### Candidates:



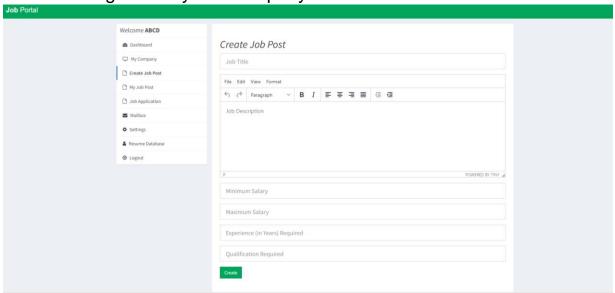
 $\textbf{Copyright} @ \textbf{2022-2023 Job Portal.} \ \textbf{All rights reserved}.$ 

### Companies:

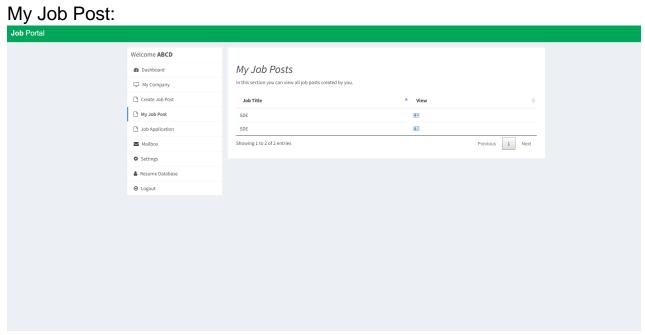


Copyright © 2022-2023 Job Portal. All rights reserved.

### 11.4 Creating a Job by the Company

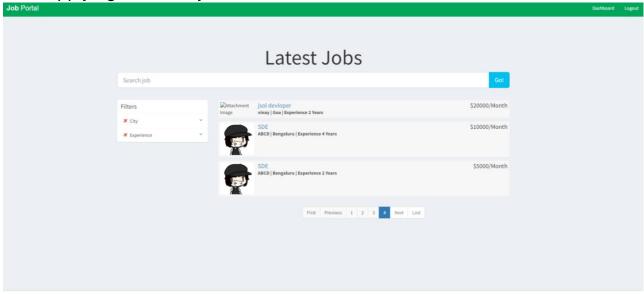


Copyright © 2022-2023 Job Portal. All rights reserved.

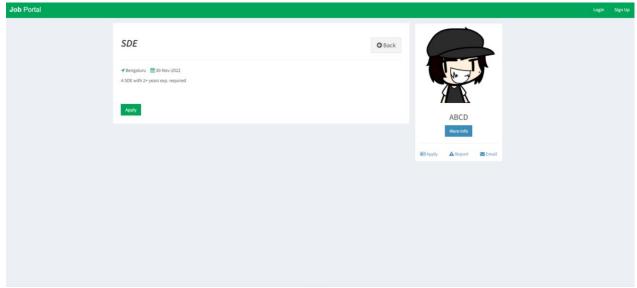


Copyright © 2022-2023 Job Portal. All rights reserved.

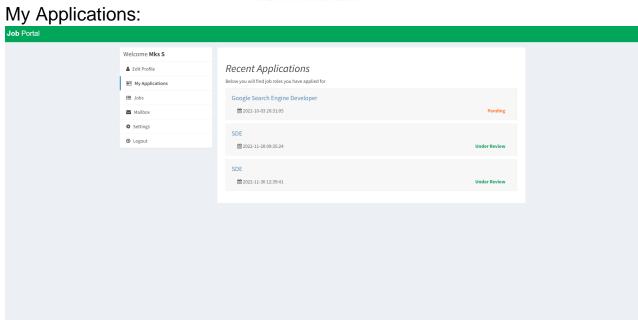
### 11.5 Applying to Jobs by the Candidate



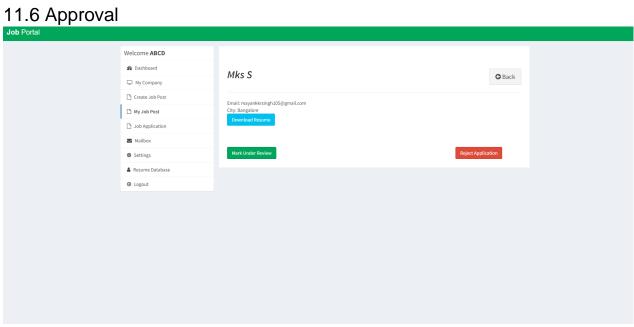
Copyright © 2022-2023 Job Portal. All rights reserved.

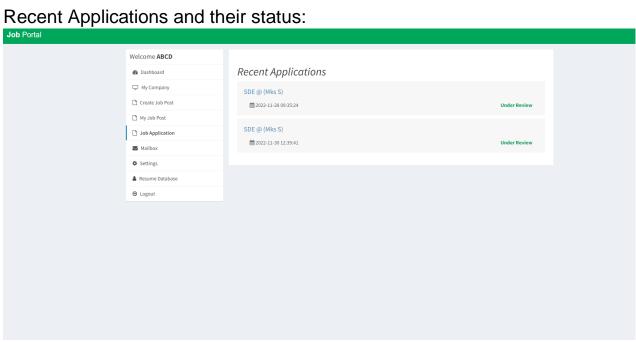


Copyright © 2022-2023 Job Portal. All rights reserved.



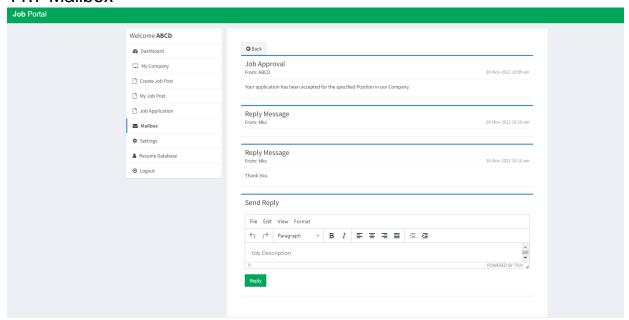
Copyright © 2022-2023 Job Portal. All rights reserved.





Copyright © 2022-2023 Job Portal. All rights reserved.

### 11.7 Mailbox



Copyright © 2022-2023 Job Portal. All rights reserved.

