

Job Portal

Kavali Sri Vyshnavi Devi

Sahaja Nandyala

1 Overview:

Job portal that has resume management, job demands that both companies and job seekers can use. In this we have two interfaces one is for a company and other is for a job seeker. A company can login and post new jobs, check all jobs they posted, and check profiles of all job seekers who have applied to that respective job. A job seeker can login and see all the jobs that are posted by different companies and apply to jobs with his profile which includes his education details, experience details etc. Keeping this in mind we have framed a database called jobportal with tables company_user_account, jobseeker_user_account, company, seeker_profile, education_details, experience_details, seeker_skill_set, skill_set, job_post, job_post_activity, .

2 Database Schema:

```
company_user_account(company_id, password, email)
```

```
jobseeker_user_account(company_id, password, email)
```

```
company(company_id, company_name, company_description, business_stream,  
company_website_url)
```

```
seeker_profile(user_id, first_name, last_name, gender, phone_no)
```

```
experience_details(user_id, job_title, starting_date,  
completion_date,  
company_name, description)
```

```
education_details(user_id, education_level, major,  
institute_name, starting_date, completion_date, percentage)
```

```

job_post(job_post_id,
company_id, job_type, created_date, end_date,
job_description, job_location)
job_post_activity(user_id,job_post_id company_id, apply_date)

```

```

skill_set( skill_name)

```

```

seeker_skill_set(user_id, skill_name)

```

3 ER Model:

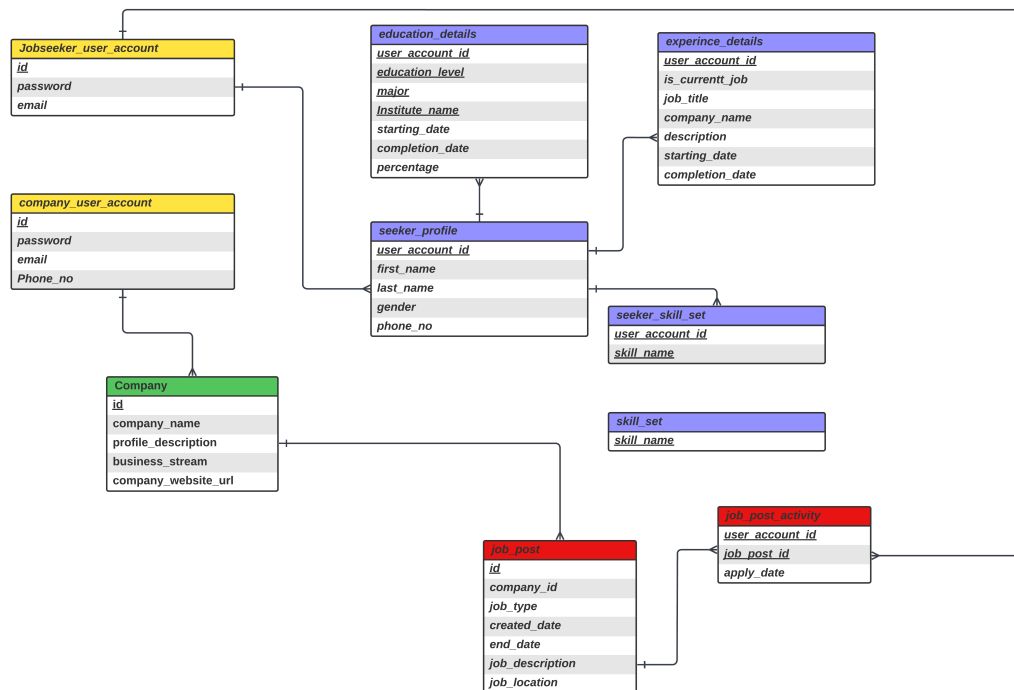


Figure 1: ER Diagram

4 Integrity Constraints

Relation	Primary Key	Foreign Key(Referencing table)
jobseeker_user_account	user_id	-
company_user_account	company_id	-
company	company_id	company_user_account(company_id)
seeker_profile	user_id	jobseeker_user_account(user_id)
education_details	user_id, educational_level, major	seeker_profile(user_id)
experience_details	user_id, job_title, company_name	seeker_profile(user_id)
skill_set	user_id, skill_name	skill_set(skill_name)
job_post	job_post_id, company_id	company(company_id)
job_post_activity	user_id, job_post_id, company_id	seeker_profile(user_id), job_post(job_post_id), company(company_id)

5 Users:

User: company

Password: Company_Apple1

```
create user 'company'@'localhost' identified by 'Company_Apple1';
grant all privileges on Apple.* to 'company'@'localhost';
-- use this command 'mysql -u company -p;' to access the user
```

6 Relational Database Design:

```
create database jobportal;
use jobportal;
```

```
create table jobseeker_user_account(
user_id varchar(10),
password varchar(10),
email varchar(20),
primary key (user_id)
);
```

```
create table company_user_account(
company_id varchar(10),
password varchar(10),
email varchar(20),
primary key (company_id)
);
```

```

create table company(
company_id varchar(10),
company_name varchar(20),
profile_description varchar(100),
business_stream varchar(10),
company_website_url varchar(50),
primary key (company_id),
foreign key (company_id) references company_user_account(company_id)
);

```

```

create table seeker_profile(
user_id varchar(10),
first_name varchar(15),
last_name varchar(15),
gender varchar(10),
phone_no varchar(10),
primary key (user_id),
foreign key(user_id) references jobseeker_user_account(user_id));

```

```

create table education_details(
user_id varchar(10),
educational_level varchar(15),
major varchar(15),
institute_name varchar(20),
starting_date varchar(10),
completion_date varchar(10),
percentage numeric(5,2),
primary key (user_id, educational_level, major),
foreign key (user_id) references seeker_profile(user_id)
);

```

```

create table experience_details(
user_id varchar(10),
job_title varchar(10),
starting_date varchar(10),
completion_date varchar(10),
company_name varchar(15),
description varchar(100),
primary key (user_id, job_title, company_name),
foreign key (user_id) references seeker_profile(user_id)

```

```
);
```

```
create table skill_set(  
skill_name varchar(20),  
primary key (skill_name)  
);
```

```
create table seeker_skill_set(  
user_id varchar(10),  
skill_name varchar(10),  
primary key (user_id,skill_name),  
foreign key (skill_name) references skill_set(skill_name)  
);
```

```
create table job_post(  
job_post_id varchar(10),  
company_id varchar(10),  
job_type varchar(10),  
created_date varchar(10),  
end_date varchar(10),  
job_description varchar(200),  
job_location varchar(20),  
primary key (job_post_id,company_id),  
foreign key (company_id) references company(company_id)  
);
```

```
create table job_post_activity(  
user_id varchar(10),  
job_post_id varchar(10),  
company_id varchar(10),  
apply_date varchar(20),  
primary key (user_id, job_post_id, company_id),  
foreign key (user_id) references seeker_profile(user_id),  
foreign key (job_post_id) references job_post(job_post_id),  
foreign key (company_id) references company(company_id)  
);
```

7 Tech Stacks Used

- Frontend- Javascript, html, css, bootstrap
- Backend- Sql, ExpressJs, NodeJs

8 Interface Designs

To build user interface Html, Css, Scss, Bootstrap, JQuery, Javascript are used. In every page which is being rendered in website there is a navigation bare to navigate through the pages/sections. Upon success or failure of login of applying or filling a form javascript alert function is used to notify the user. The results like user information is shown in form of jquery data tables. there are also user and admin control acces so that user can not go to admin portal and admin can not use user portal. There are also other pages like profile and dashboard in company and user interfaces.

9 Different Routers And Their Details

/ : In this page for login of admin and user buttons are shown. This router is to display the home page of the whole website.

/adminLogin : Here, the admin that is company side users can login into the portal by entering valid ID and password. To login as a admin one can use the credentials present in table company_user_account.

/userLogin : Here, the user that is tyhe applicant can login into the portal by entering valid ID and password. To login as a user one can use the credentials present in table jobseeker_user_account.

/login : This router validates login of user and admin accordingly sends response 200 to the login.ejs page which goes to dashboard of respective user type on success.

/register : This page renders the registration if an user is new and wanted to create the account in the portal. This page is also rendered based on user type which is set in home page.

/companyFormSubmit : When company user (admin) registers for the first time this router validates the information entered by the company user and stores the data in database and in respective tables. Upon successful registration the user is shown alert message and rendered to login page.

/userFormSubmit : When user (applicant) registers for the first time this router validates the information entered by the company user and stores the data in database.

- /userDashboard** : When user (applicant) logins successfully the page goes to this router and renders user dashboard which is like a welcome page for the user.
- /userProfile** : When user (applicant) logged in he/she can access their profile which shows education details, experience details and details entered by them during registration.
- /userJobListings** : When user (applicant) logged in he/she can access all the job posts present which can be applied or that are applied already can be shown to the user.
- /userJobDetails** : When user (applicant) logged in he/she can access the job details when clicked on which they wanted to apply for and in this page description of job is shown in detail.
- /apply** : When user clicks on apply button when it is previously not applied the data on click goes to this router and the information is fetched through the request and it is stored in the respective tables.
- /companyDashboard** : When company (admin) logins successfully the page goes to this router and renders user dashboard which is like a welcome page for the admin.
- /companyJobPostings** : When company (admin) logins all the jobs posted by that particular company is shown and there will be a click button which renders a page to show who all applied.
- /companyJobPostingsApplied** : When company (admin) logins all the jobs posted by that particular company and the clicked particular job is shown to the company so that they can access the information of the user who all applied to the particular job.
- /companyJobPostNew** : When company (admin) clicks on post a new job link on navigation bar it goes to this router which renders the form to post a new job.
- /companyJobPostNewFormSubmit** : When company (admin) fills the form to post a new job and submit this the data is sent to this router through the http get request in which the information is validated and inserts data into respective tables.