

Softwares used:

Bootstrap : framework for developing responsive web pages

HTML/CSS : For the front end

Php : for the backend

Mysql : For database support

Xampp: For aiding in database manipulation and locally hosting the webpage

Table descriptions:

Admin : has the admin username and password

Role:

Has the details about the active companies registered, candidates registered, total job posts and total applications.

Has the power to remove any user or candidate

Any company before being able to recruit and post jobs must be verified by the admin.

Apply\_job\_post\_table:

Use of:

Id\_apply: acts as primary key with auto increment on

Id\_job\_post: acts as the id for the GET method of php for opening the specific webpage having job description

Id company/user : id of user or the company

Status: status of the job post depending on the admin. A job post is published only after validation by the admin

City, state, country table :

Consist of a large database of the cities, states and tables connected together by id of state and countries. They are created separately since while creating forms the option to select country comes first, then the state and then the city. None of these are compulsory so the 3 tables are kept separately so that if a user only provides country or country-state it works as well.

Company Table:

Consists of all the details of the country. The logo is stored as image format in the uploads folder.

Any company before being to operate must be able to be verified from the admin when their active column changes from 2 to 1.

Users:

Similar to the companies these include all the users looking for jobs. The resume is stored as pdf(size smaller than 5 mb) and is stored in the uploads folder. The passwords are stored after applying decoding so that even the admin or the superuser can't access the password.

Mailbox/Reply mailbox:

We have included an inbuilt mailing technique b/w the users and the companies, the user can have conversation with the employee who have interest in their application. The design is taken up from gmail and all the information is stored in the sql file i.e. the message, from, to and the date.

---

How the connection with mysql is made:

In all the php pages which require access to database we include the db.php.

db.php is a php which is used for establishing the sql connection. Rather than writing all of that separately in all the php files, we have created one db file and included this in all which we require.

For running any SQL query we just write the query similar to that in SQL and then run it using the connection established.

```
Eg: $result = $conn->query($sql);
```

Role of each php file in the program:

Index.php : The main php file that opens up at the start. Contains link to other pages like login and register and also the admin login.

Shows 4(using sql limit) randomly selected job offers at the centre.

Jobs.php: Contains the job offers 4 per page(again by using limit 4, for this code is written in jobpagination.php). The filters applied are using the help of search.php which applies the filters of city and experience. Currently the cities are navi-mumbai and Bengaluru but more can be added as required.

Clicking on any of these open the view-job-post.php

View-job-post.php : also has an id at the end of url required to identified which job post page to open as all have the same format. This id is sent using the GET method of php.

Login/Signup/Register php files: These are the pages which open the login and signup pages. The details of username and password are checked using the checklogin.php and checkcompany login.php. All the login and register pages have the standard php code for validation of phone number, email id, pdf size etc. These can be seen at W#Schools itself.

Folders pf user/company/admin: These contain the php pages to be loaded when the corresponding people are logged in. This is known by using the session global variable of php to know which of these is set. The php files within are pretty simple to those mentioned above with the exception that these are a little more specified for the particular kind of user.

SQL Queries:

The common files (no user/company/admin specified):

File: addcompany.php

Queries:

```
$sql = "SELECT email FROM company WHERE email='$email'";
```

To check if the email already exists so that every new company created has a new email.

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

---

Inserting new data retrieved from the form into the company table

---

File: adduser.php

Queries:

```
$sql = "SELECT email FROM users WHERE email='$email'";
```

Similar to the above to check if the file already exists.

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

For inserting data into the user table

---

File: apply.php (not a file directly associated with the front end but called for processing of another file)

```
$sql = "SELECT * FROM job_post WHERE id_jobpost='$_GET[id]'";
```

To retrieve all the data from the job\_post table where id is taken from the GET function of php.

```
$sql1 = "SELECT * FROM apply_job_post WHERE id_user='$_SESSION[id_user]' AND id_jobpost='$row[id_jobpost]'";
```

Used to check if the user has applied to the job post or not.

```
$sql = "INSERT INTO apply_job_post(id_jobpost, id_company, id_user) VALUES ('$_GET[id]', '$id_company', '$_SESSION[id_user]')";
```

Whenever a user applies for a job the details need to be added to the apply\_job\_post table.

---

File: checkcompanylogin.php

```
$sql = "SELECT id_company, companyname, email, active FROM company WHERE email='$email' AND password='$password'";
```

To check for if the login is correct or not.

Similar query in the checklogin as well.

---

File: city.php

```
$sql = "SELECT * FROM cities WHERE state_id='$_POST[id]'";
```

To select the cities from states. Similar is with the case in state.php in a layer by layer like fashion.

---

File: index.php

```
$sql = "SELECT * FROM job_post Order By Rand() Limit 4";
```

To select 4 random job offers to display at the main screen.

Other sql commands are the simple select commands to get the count of various elements.

---

File: jobpagination.php

```
$sql = "SELECT * FROM job_post LIMIT $start_from, $limit";
```

Similar to above case for choosing 4 job posts at a time. Limit variable is set to 4.

---

File: search.php

```
$sql = "SELECT * FROM job_post WHERE jobtitle LIKE '%$search%' LIMIT $start_from, $limit";
```

To search specific company on the basis of inputted name.

---

File: view-job-post.php

```
$sql = "SELECT * FROM job_post INNER JOIN company ON job_post.id_company=company.id_company WHERE id_jobpost='$_GET[id]'";
```

To select details of the specific company on the page on the basis of id sent by GET method of php.

---

Similar queries in the user, company and admin php files for the specific user types.

Scopes of improvement:

There are a few hyperlinks which don't work properly like the report button which is mainly down to it not being hosted on any server

The webpage is hosted locally at the moment. Needs to be run on the server.