TEMPORARY DOCUMENTATION TO GRASP AN IDEA OF WUTS GOING ON UNDER THE HOOD

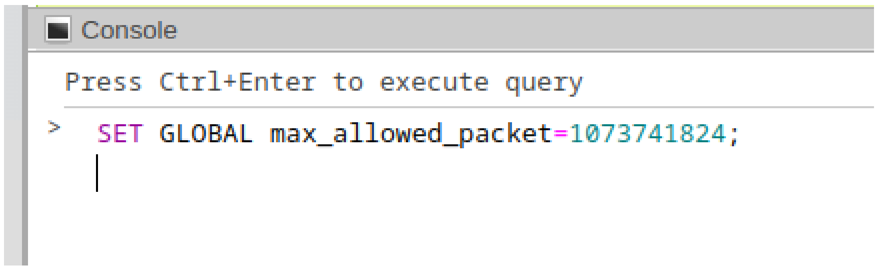
website structure:

four main directories:
ADMIN/   HOME/   CANDIDATE/   RECRUITER/

here is how ima be rolling with this thing
each section on this guide will be focused around a directory, ima go through sql queries, design, screenshots, some css, fonts, bootstrap sass, number of pages, php stuff . . . . . you know the drill.

Now first you gotta download the latest webp zip file, 3.1:

unzip it put it into htdocs, it's highly recommended to use a virtual machine to keep things clean, now look for the sql file probably named webp followed by the version number.
Here is the issue,before importing it we gotta fix something, that sql file got so large it's causing a timeout, to fix that pull up to da console on da phpmyadmin ui and run this command.



```
> SET GLOBAL max_allowed_packet=1073741824;
```

If you wanna read more, here is a stack overflow discussion thread

http://stackoverflow.com/questions/93128/mysql-error-1153-got-a-packet-bigger-than-max-allowed-packet-bytes

also regarding database connection make sure to change the database name and the password accordingly on any file located at **connection/db.php**

Database tables:

admin_login: id, admin_email, admin_pass, admin_username, first_name, last_name, admin_type, date_column.

Recruiters: rec_id, username, email, password, companyname, website, industry, description, phone, address, city, country, zip_code, Foundation, registration_date.

Candidates: can_id, firstname, lastname, Username, Email, password, Phone, Address, City, Country, zipcode, birth, gender, image_name, image_data, registrationdate.

job_offer: job_id, job_title, job_type, salary, job_description, requirements, benefits, creation_date, rec_id, can_id.

Messages: mess_id, message, can_id, job_id, rec_id.

skills: id, skill_name, skill_value, rec_id, job_id.

applied_jobs: app_id, Job_id, Can_id, score.

Yoo it is really worth noting that attribute naming is really important, for instance:

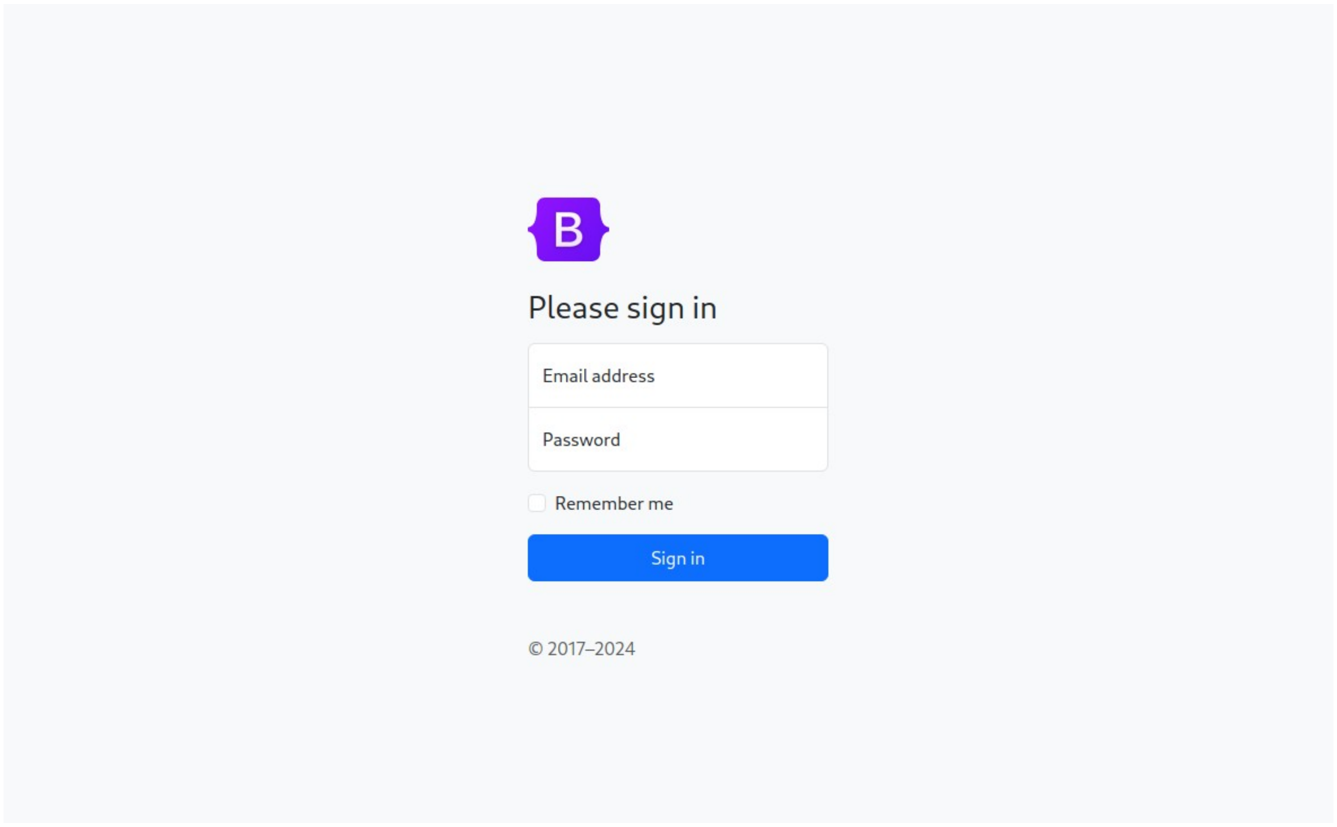
When joining two tables in a SQL query and encountering a scenario where both tables have a column with the same name, such as a foreign key in one table and a primary key in another, you may face difficulties accessing the values of these columns directly using aliases, you gonna be forced to change the name like:

$row['c.job_id']: wont provide the desired value you're looking for.
This yeah of course **$row['job_id'] $row['Job_id']**

now with that said let's kick this off

ADMIN

1. Sign in Interface:



copy pasted from the bootstrap 5 website, examples section.

1.1 sql queries:

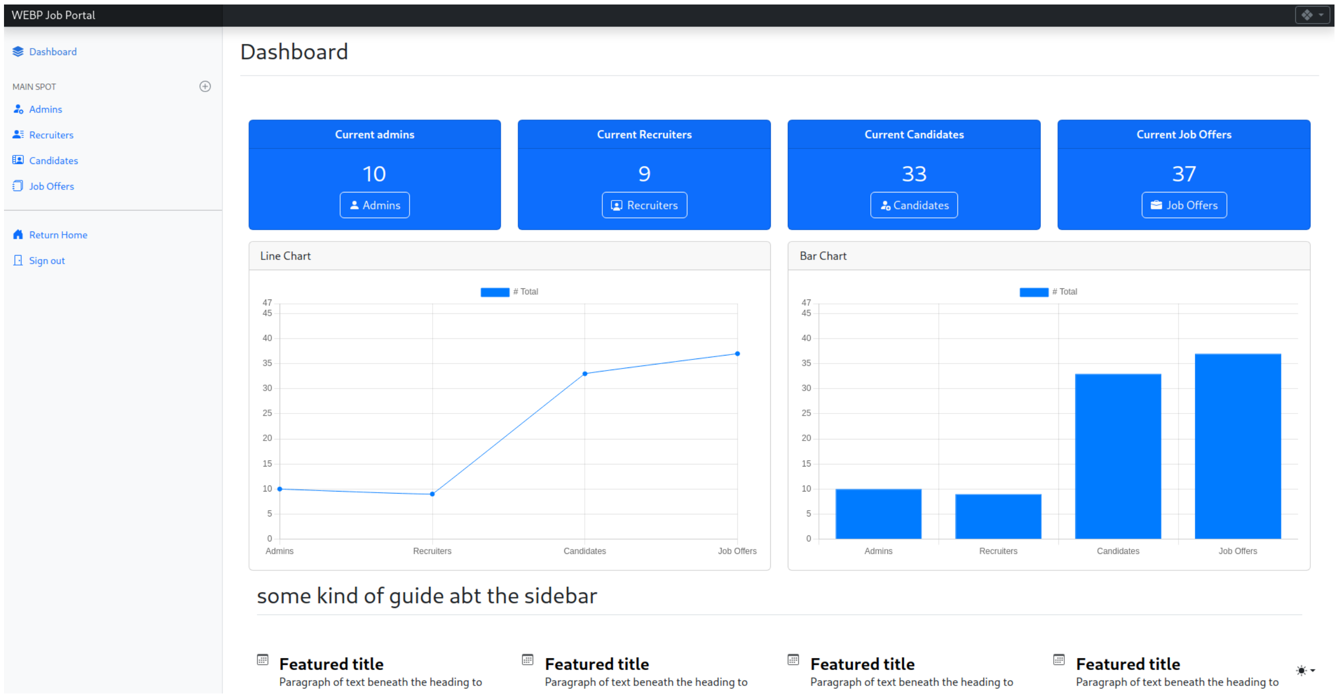
```
select * from admin_login where
admin_email='$email' and admin_pass='$pass'
```

located at phpcode/log_admin_code.php

1.2 stuff you should know:

posting, session, isset(), getting, mysqli ...

1. Dashboard Interface:



Dashboard

Current admins 10 Admins

Current Recruiters 9 Recruiters

Current Candidates 33 Candidates

Current Job Offers 37 Job Offers

some kind of guide abt the sidebar

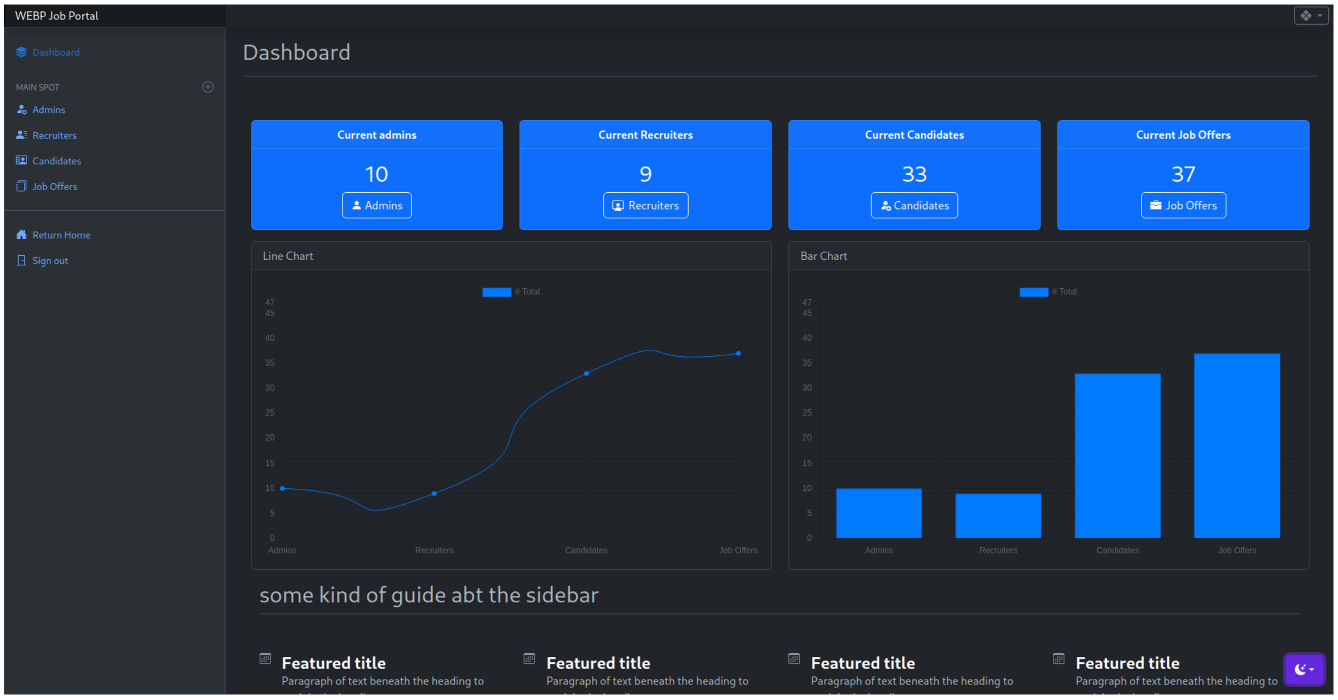
Featured title
Paragraph of text beneath the heading to

Featured title
Paragraph of text beneath the heading to

Featured title
Paragraph of text beneath the heading to

Featured title
Paragraph of text beneath the heading to

the dashboard aint needing any description, it describes itself, oh yeah dark mode.



Dashboard

Current admins 10 Admins

Current Recruiters 9 Recruiters

Current Candidates 33 Candidates

Current Job Offers 37 Job Offers

some kind of guide abt the sidebar

Featured title
Paragraph of text beneath the heading to

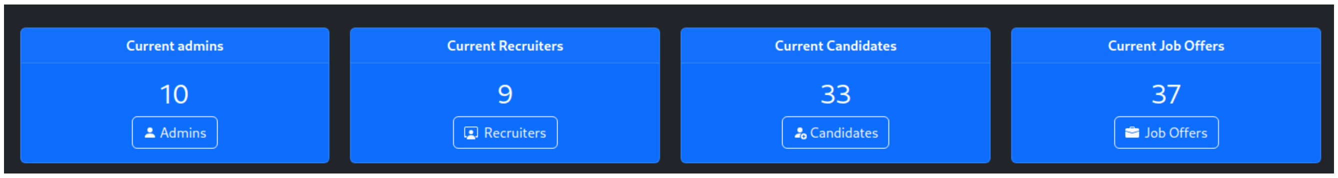
Featured title
Paragraph of text beneath the heading to

Featured title
Paragraph of text beneath the heading to

Featured title
Paragraph of text beneath the heading to

Located at admin_dashboard.php

1.1 sql queries:



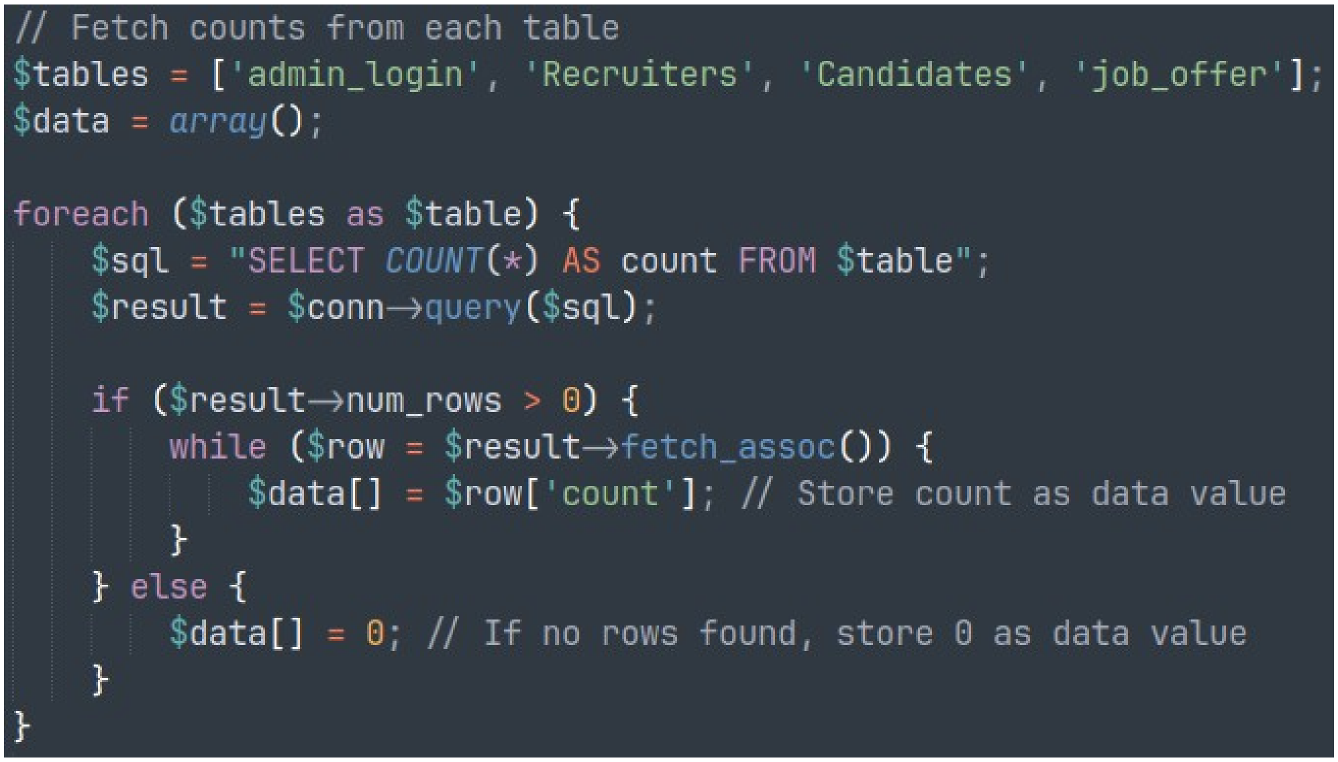
Current admins
10
Admins

Current Recruiters
9
Recruiters

Current Candidates
33
Candidates

Current Job Offers
37
Job Offers

the card numbers are generated using this code:

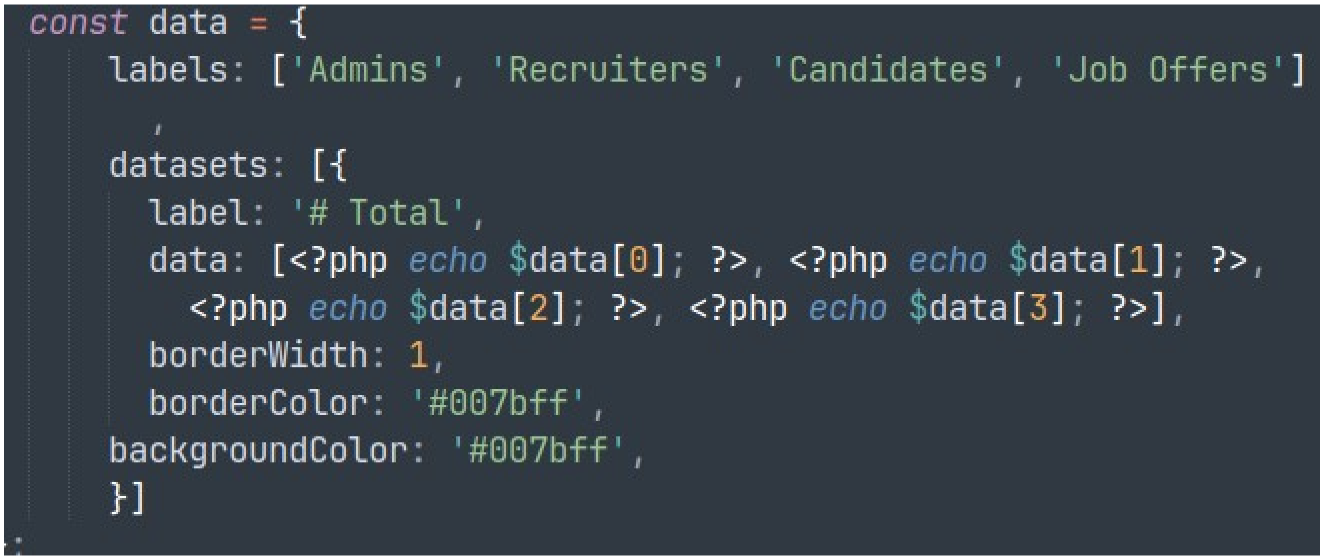


```
// Fetch counts from each table
$tables = ['admin_login', 'Recruiters', 'Candidates', 'job_offer'];
$data = array();

foreach ($tables as $table) {
    $sql = "SELECT COUNT(*) AS count FROM $table";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        while ($row = $result->fetch_assoc()) {
            $data[] = $row['count']; // Store count as data value
        }
    } else {
        $data[] = 0; // If no rows found, store 0 as data value
    }
}
```

next up, the charts, I used a library called charts.js
https://www.chartjs.org/

we feeding the charts using the data array from the code above

still located at the same file



```
const data = {
    labels: ['Admins', 'Recruiters', 'Candidates', 'Job Offers']
    ,
    datasets: [{
      label: '# Total',
      data: [<?php echo $data[0]; ?>, <?php echo $data[1]; ?>,
        <?php echo $data[2]; ?>, <?php echo $data[3]; ?>],
      borderWidth: 1,
      borderColor: '#007bff',
    backgroundColor: '#007bff',
    }]
};
```

now the sidebar:

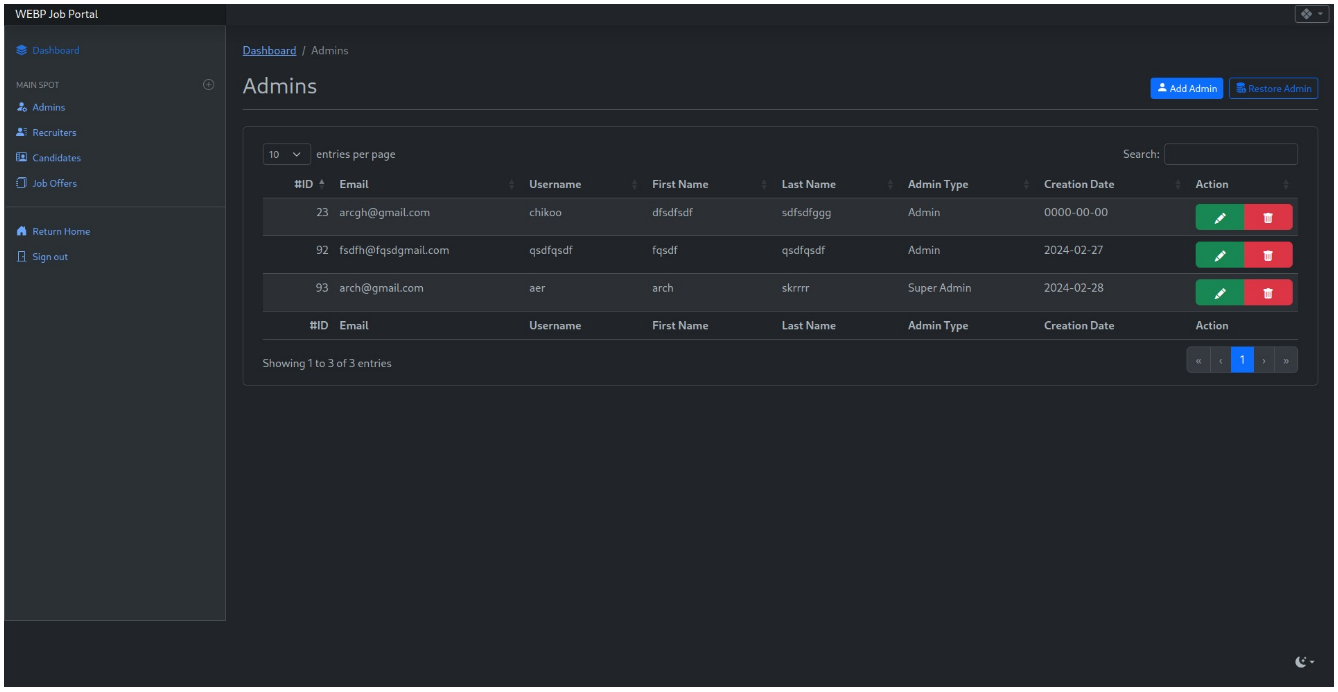
Admins `select * from admin_login where status=1`

Recruiters `select * from Recruiters`

Candidates `select * from Candidates`

Job Offers `select * from job_offer`

1.2 Admins:



WEBP Job Portal

Dashboard

MAIN SPOT

Admins

Recruiters

Candidates

Job Offers

Return Home

Sign out

Dashboard / Admins

Admins

Add Admin   Restore Admin

10 entries per page   Search:

#ID	Email	Username	First Name	Last Name	Admin Type	Creation Date	Action
23	arcgh@gmail.com	chikoo	dfsdfsdf	sdfsdfggg	Admin	0000-00-00	
92	fsdfh@fqsdgmail.com	qsdfqsdf	fqsdf	qsdfqsdf	Admin	2024-02-27	
93	arch@gmail.com	aer	arch	skrrrr	Super Admin	2024-02-28	
#ID	Email	Username	First Name	Last Name	Admin Type	Creation Date	Action

Showing 1 to 3 of 3 entries

« ‹ 1 › »

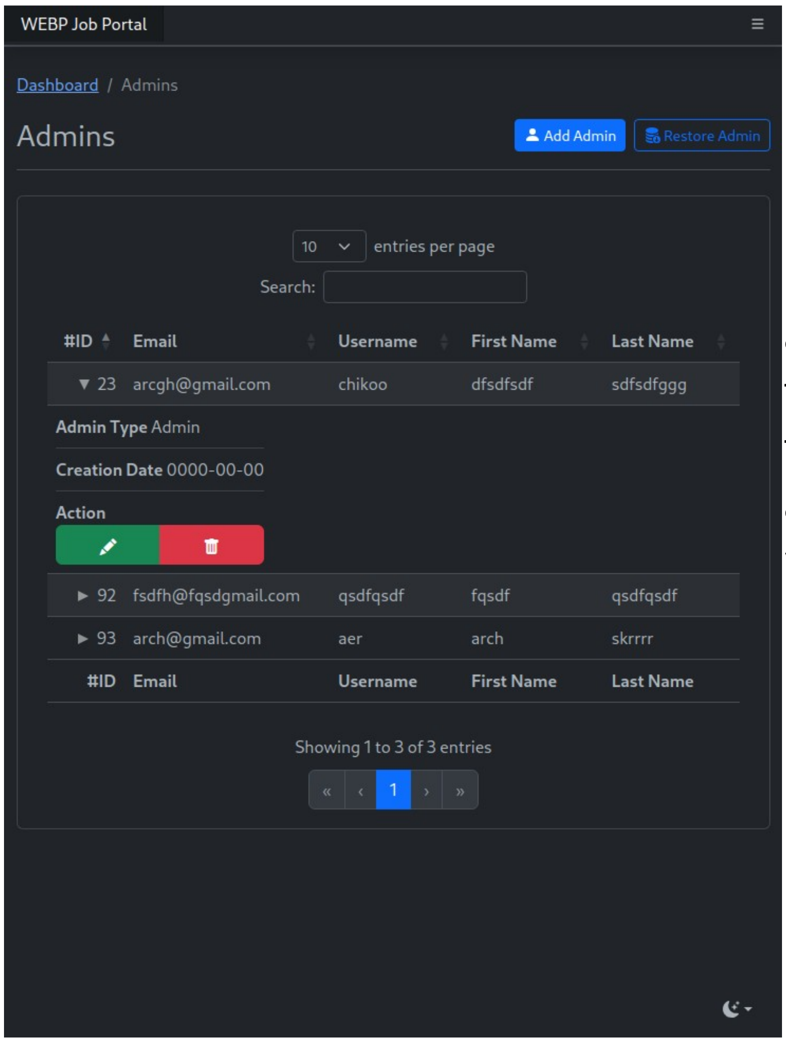
the admin can add/edit/delete admins, the table is from the datatables library
https://datatables.net/

located at: admins.php

here is the js for the table:

```
new DataTable('#example', {
    responsive: true
});
```

here is what responsive true does:



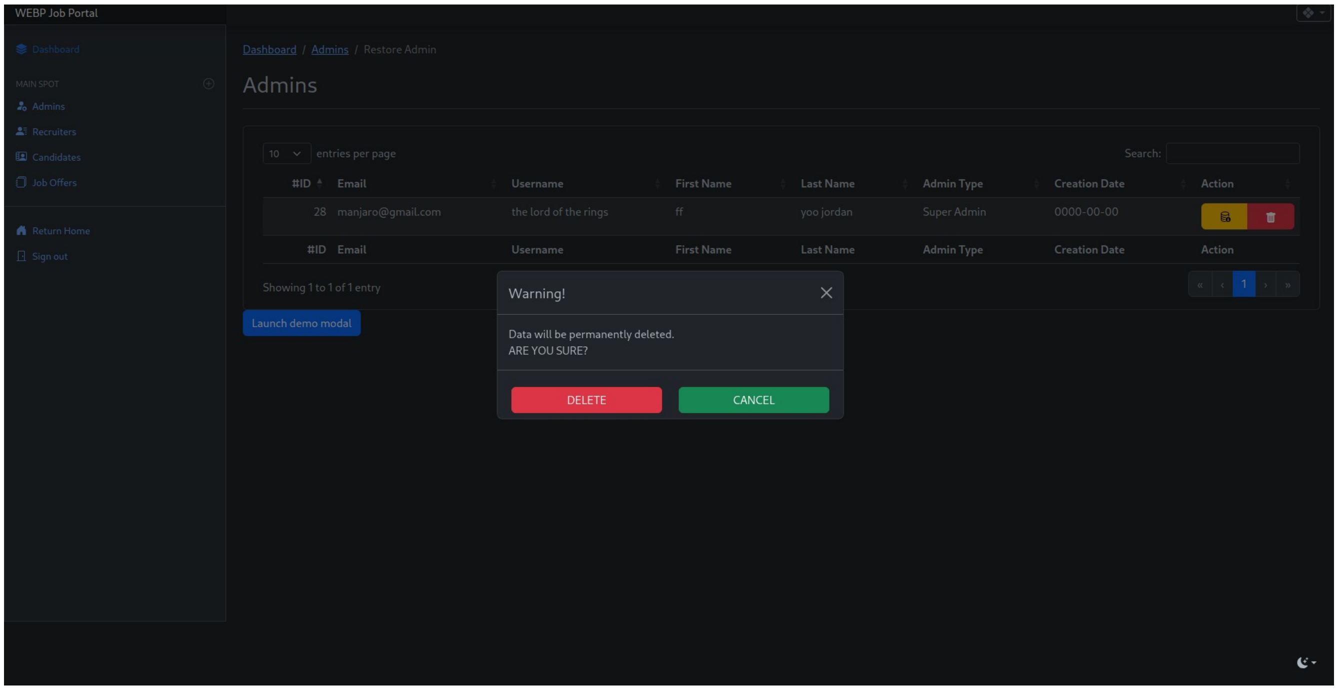
#ID	Email	Username	First Name	Last Name
▼ 23	arcgh@gmail.com	chikoo	dfsdfsdf	sdfsdfggg
Admin Type Admin				
Creation Date 0000-00-00				
Action				
► 92	fsdfh@fqsdgmail.com	qsdfqsdf	fqsdf	qsdfqsdf
► 93	arch@gmail.com	aer	arch	skrrrr
#ID	Email	Username	First Name	Last Name

Showing 1 to 3 of 3 entries

When responsive true is set, DataTables automatically adjusts the layout of the table based on the available space in the viewport or its container.

1.2 restore admin:

confirm delete modal



Admins

10 entries per page Search:

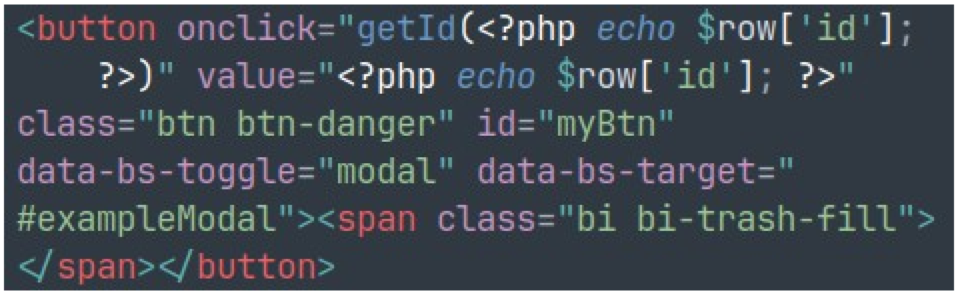
#ID	Email	Username	First Name	Last Name	Admin Type	Creation Date	Action
28	manjaro@gmail.com	the lord of the rings	ff	yoo jordan	Super Admin	0000-00-00	
#ID	Email	Username	First Name	Last Name	Admin Type	Creation Date	Action

Showing 1 to 1 of 1 entry

Warning!

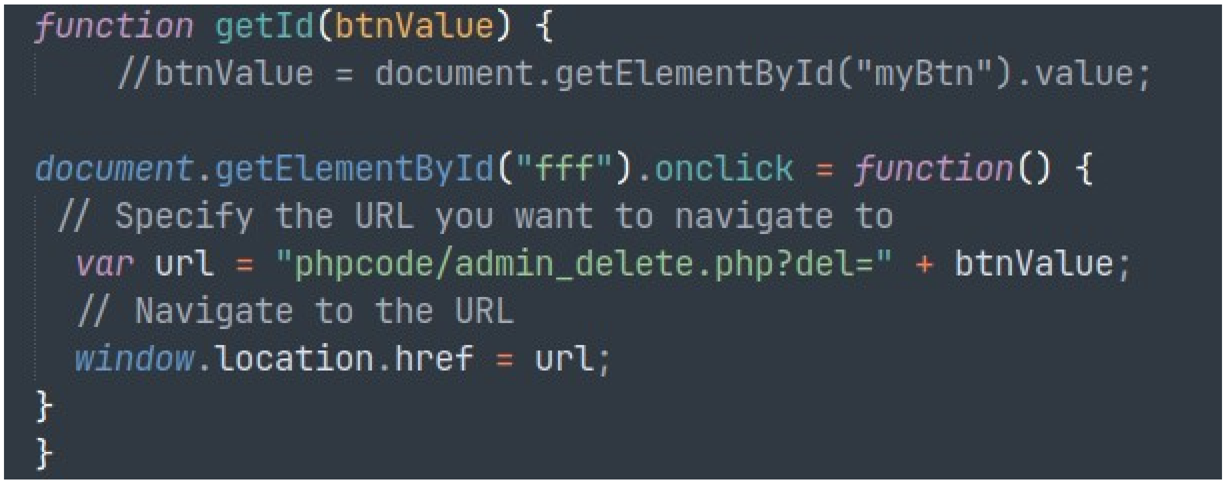
Data will be permanently deleted.
ARE YOU SURE?

the most important thing in this entire dashboard is this goofy modal, spent too much time on it for no reason lol



```
<button onclick="getId(<?php echo $row['id'];
    ?>)" value="<?php echo $row['id']; ?>"
class="btn btn-danger" id="myBtn"
data-bs-toggle="modal" data-bs-target="
#exampleModal"><span class="bi bi-trash-fill">
</span></button>
```

the onclick stuff sets an onclick event handler that calls a JavaScript function named getId() and passes the id value from the PHP variable $row['id'] as a parameter.



```
function getId(btnValue) {
    //btnValue = document.getElementById("myBtn").value;

document.getElementById("fff").onclick = function() {
 // Specify the URL you want to navigate to
  var url = "phpcode/admin_delete.php?del=" + btnValue;
  // Navigate to the URL
  window.location.href = url;
}
}
```

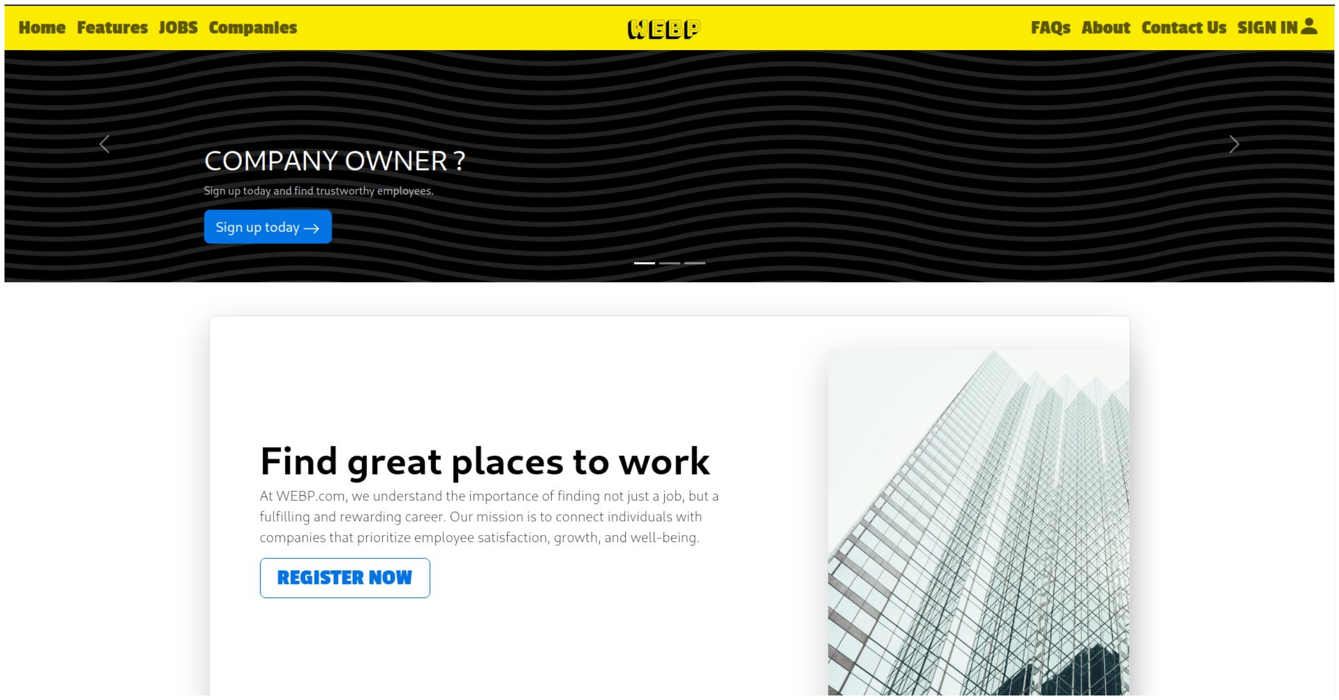
fff is the id of the red delete btn from the modal

located at: admin_restore.php

for candidates, Recruiters, job offers same principal copy paste, but without any restore functionality straight up deleting.

HOME

1. Home Page:



COMPANY OWNER ?

Sign up today and find trustworthy employees.

Sign up today →

Find great places to work

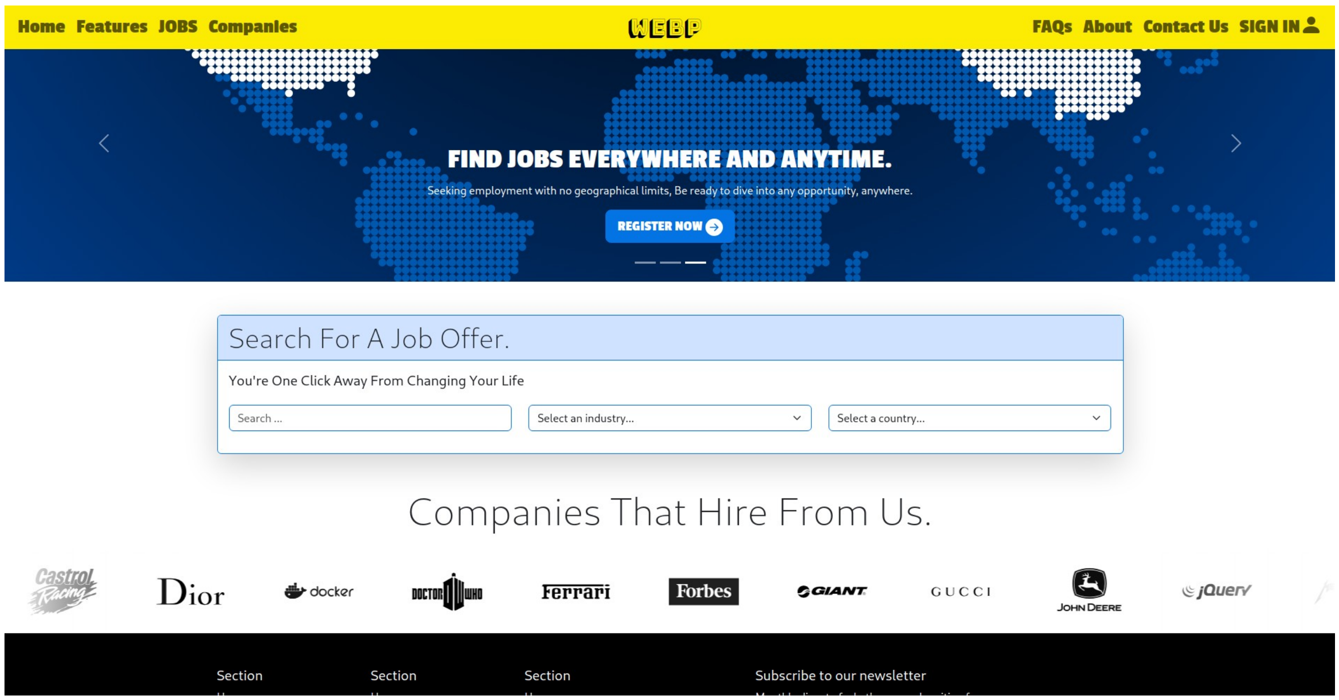
At WEBP.com, we understand the importance of finding not just a job, but a fulfilling and rewarding career. Our mission is to connect individuals with companies that prioritize employee satisfaction, growth, and well-being.

REGISTER NOW

mainly the home page got only two purposes, signing in and lurking through the job offers.

1.1 sql queries:

JOBS_search.php



FIND JOBS EVERYWHERE AND ANYTIME.

Seeking employment with no geographical limits, Be ready to dive into any opportunity, anywhere.

REGISTER NOW

Search For A Job Offer.

You're One Click Away From Changing Your Life

Search ... | Select an industry... | Select a country...

Companies That Hire From Us.

Section Section Section Subscribe to our newsletter

the searching code is located at
phpcode/livesearch3.php

```
$query = "SELECT * FROM job_offer JOIN Recruiters
ON job_offer.rec_id = Recruiters.rec_id WHERE 1=1";

if (!empty($input)) {
    $query .= " AND companyname LIKE '{$input}%'";
}

if (!empty($select1)) {
    $query .= " AND country = '$select1'";
}

if (!empty($select2)) {
    $query .= " AND industry = '$select2'";
}
```

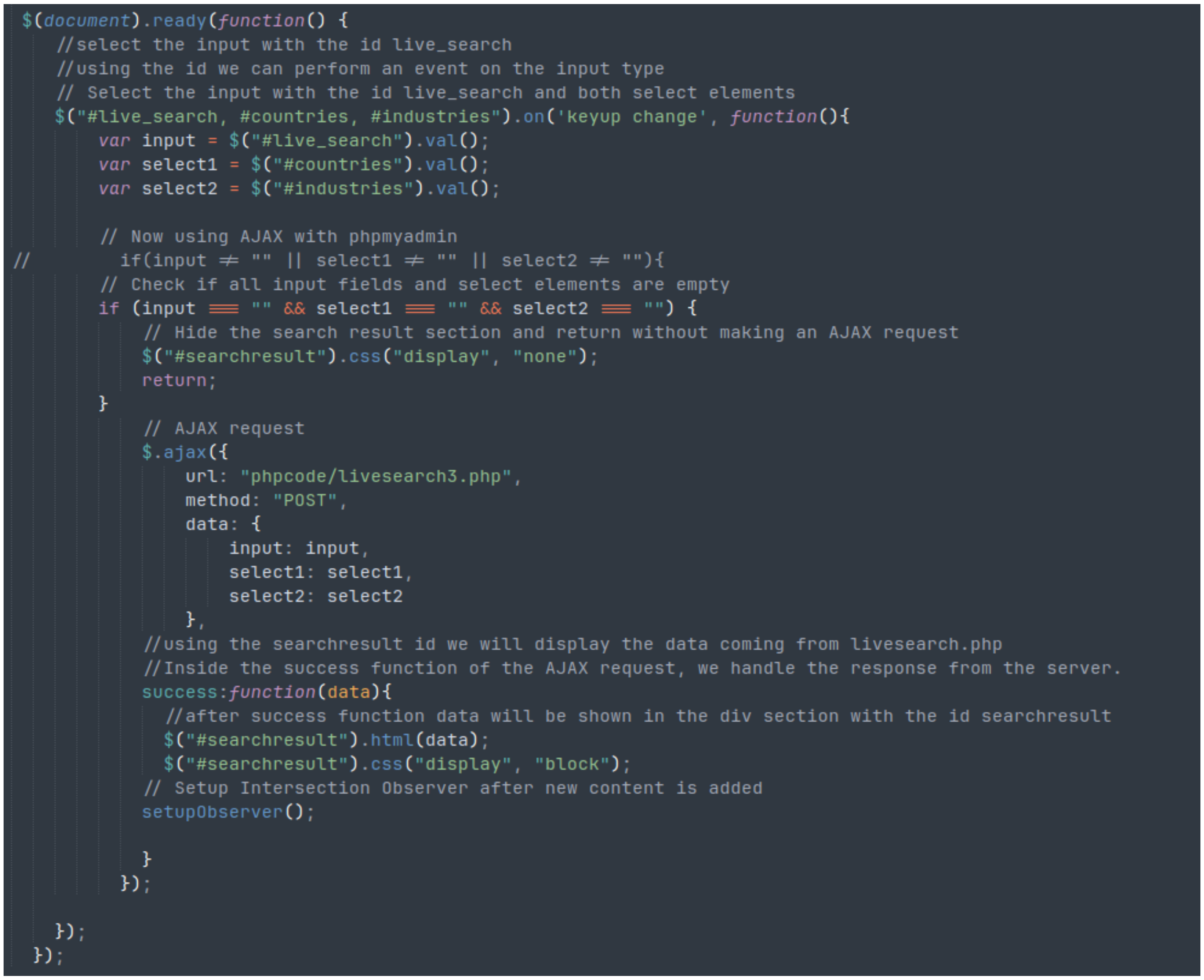
1.2 AJAX:

Read data from a web server - after a web page has loaded.

Update a web page without reloading the page.

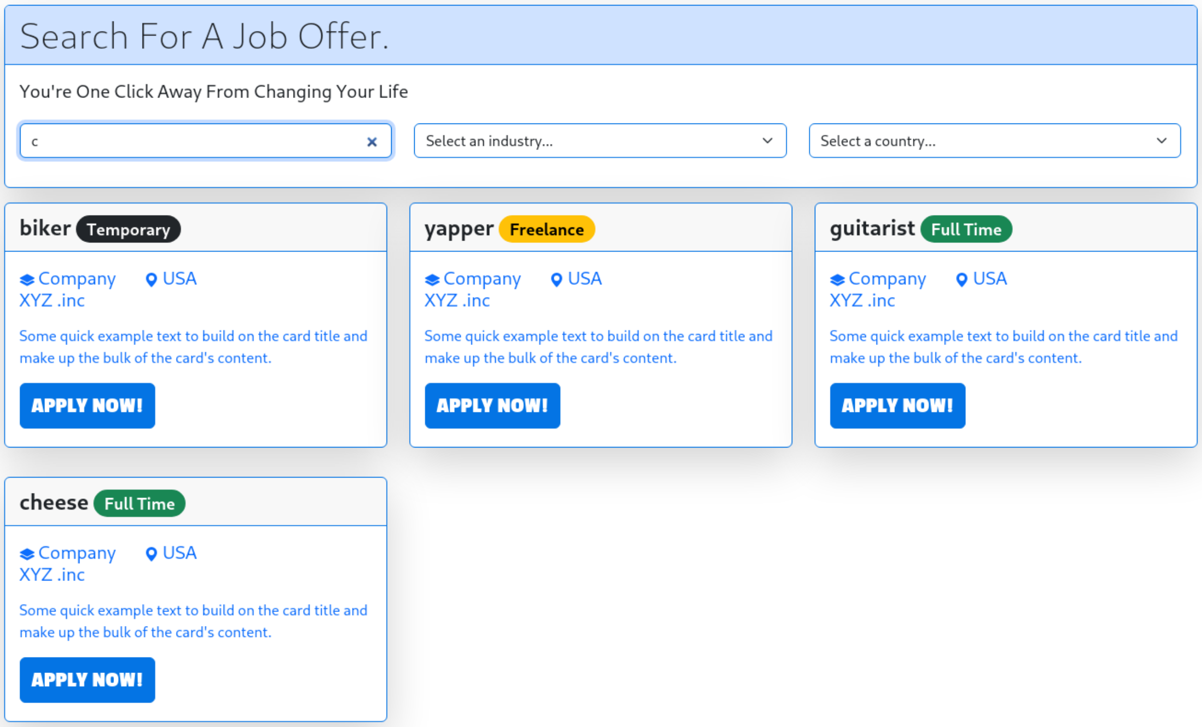
Send data to a web server - in the background.

AJAX
Asynchronous Javascript And XML



```javascript
$(document).ready(function() {
   //select the input with the id live_search
   //using the id we can perform an event on the input type
   // Select the input with the id live_search and both select elements
   $("#live_search, #countries, #industries").on('keyup change', function(){
       var input = $("#live_search").val();
       var select1 = $("#countries").val();
       var select2 = $("#industries").val();

       // Now using AJAX with phpmyadmin
//         if(input != "" || select1 != "" || select2 != ""){
       // Check if all input fields and select elements are empty
       if (input === "" && select1 === "" && select2 === "") {
           // Hide the search result section and return without making an AJAX request
           $("#searchresult").css("display", "none");
           return;
       }
           // AJAX request
           $.ajax({
               url: "phpcode/livesearch3.php",
               method: "POST",
               data: {
                   input: input,
                   select1: select1,
                   select2: select2
               },
           //using the searchresult id we will display the data coming from livesearch.php
           //Inside the success function of the AJAX request, we handle the response from the server.
           success:function(data){
             //after success function data will be shown in the div section with the id searchresult
             $("#searchresult").html(data);
             $("#searchresult").css("display", "block");
           // Setup Intersection Observer after new content is added
           setupObserver();

           }
         });

   });
});
```

Ajax is literally the backbone of the webp project



Search For A Job Offer.

You're One Click Away From Changing Your Life

biker Temporary
Company XYZ .inc USA
Some quick example text to build on the card title and make up the bulk of the card's content.
APPLY NOW!

yapper Freelance
Company XYZ .inc USA
Some quick example text to build on the card title and make up the bulk of the card's content.
APPLY NOW!

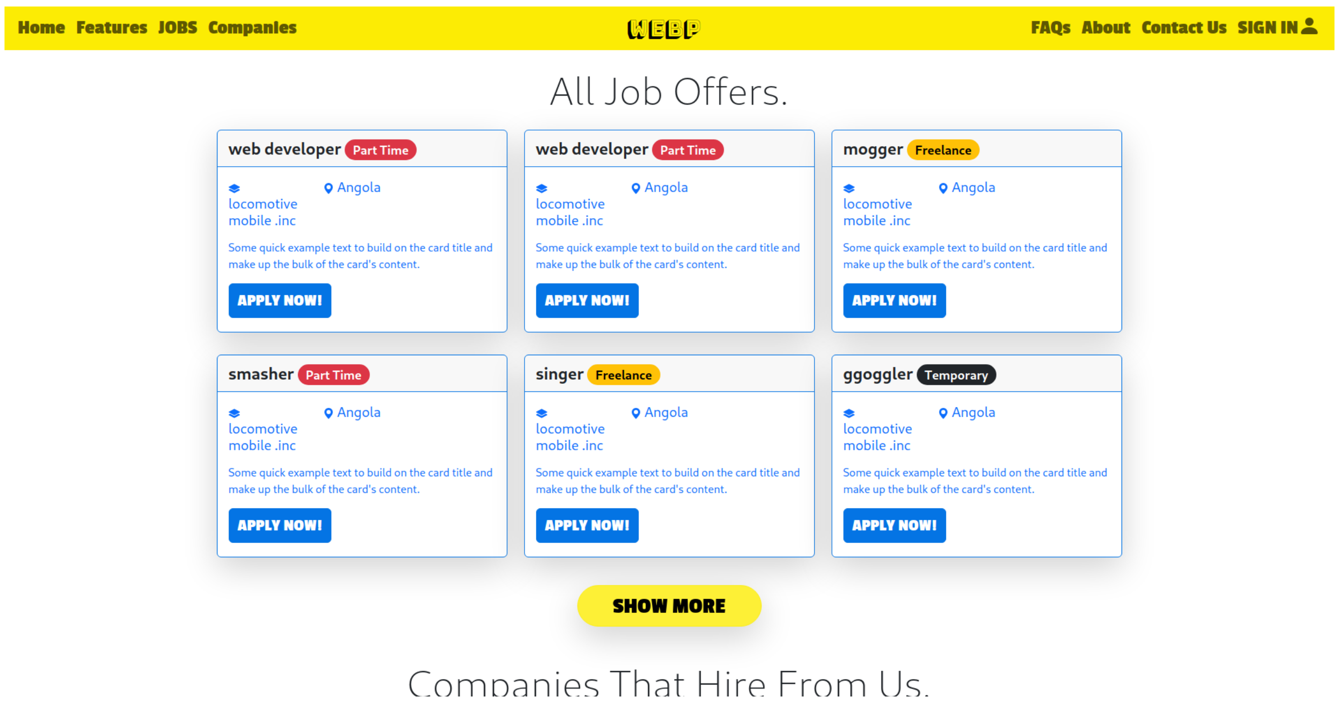
guitarist Full Time
Company XYZ .inc USA
Some quick example text to build on the card title and make up the bulk of the card's content.
APPLY NOW!

cheese Full Time
Company XYZ .inc USA
Some quick example text to build on the card title and make up the bulk of the card's content.
APPLY NOW!

also make sure to understand why we list 6 cards at the time, the logic Is on the same file.

Here is the youtube video from which I got inspired:
https://www.youtube.com/watch?v=Yggrlux69MQ&list=PLVQcN4_vTIsqIrgmJBKlWbNQt4UxOv3ly&index=7&t=380s

1.2 JOBS_list_all.php:



All Job Offers.

web developer Part Time
Angola
locomotive mobile .inc
Some quick example text to build on the card title and make up the bulk of the card's content.

web developer Part Time
Angola
locomotive mobile .inc
Some quick example text to build on the card title and make up the bulk of the card's content.

mogger Freelance
Angola
locomotive mobile .inc
Some quick example text to build on the card title and make up the bulk of the card's content.

smasher Part Time
Angola
locomotive mobile .inc
Some quick example text to build on the card title and make up the bulk of the card's content.

singer Freelance
Angola
locomotive mobile .inc
Some quick example text to build on the card title and make up the bulk of the card's content.

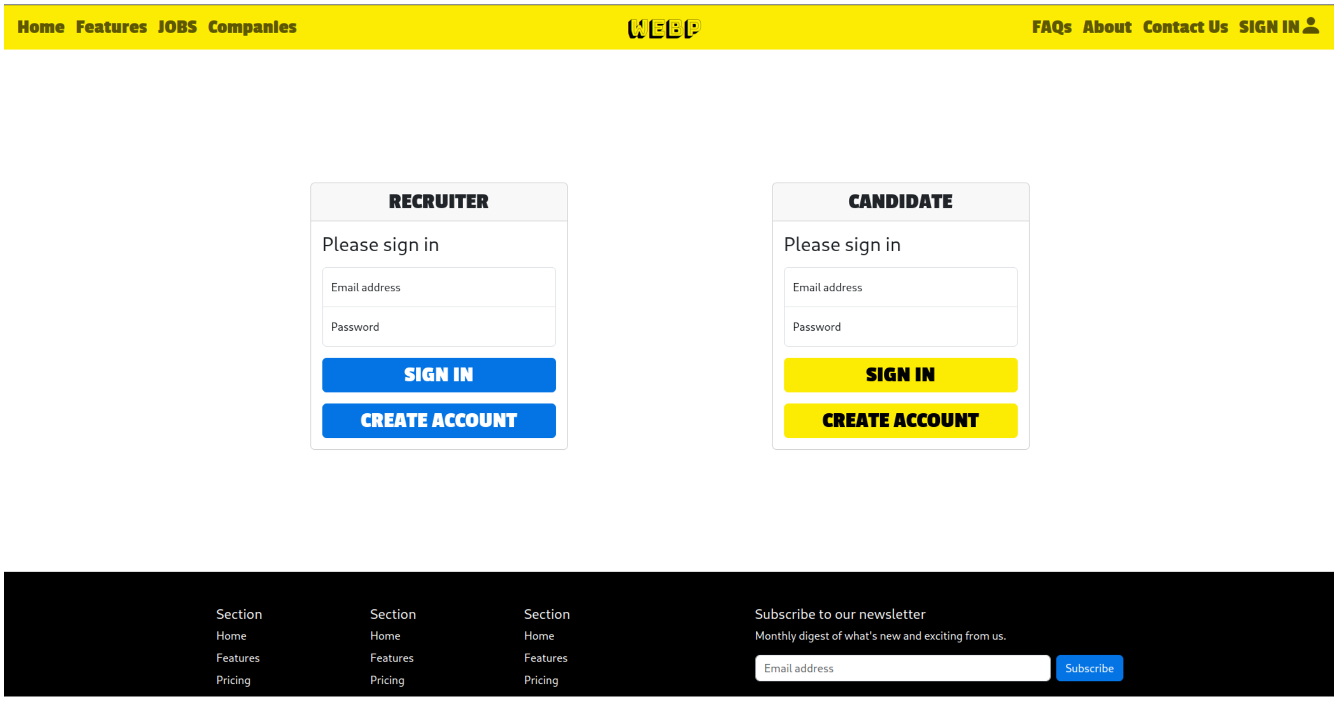
ggoggler Temporary
Angola
locomotive mobile .inc
Some quick example text to build on the card title and make up the bulk of the card's content.

Companies That Hire From Us.

the php code is located at phpcode/all_job_offers.php

Utilizing Ajax, the website can dynamically load and display six additional cards each time the user clicks "Show More", one downside, the btn needs to display that they are no more job offers.

1.3 Sign in page:


Home Features JOBS Companies WEBP FAQs About Contact Us SIGN IN

RECRUITER

Please sign in

Email address

Password

SIGN IN

CREATE ACCOUNT

CANDIDATE

Please sign in

Email address

Password

SIGN IN

CREATE ACCOUNT

Section
Home
Features
Pricing

Section
Home
Features
Pricing

Section
Home
Features
Pricing

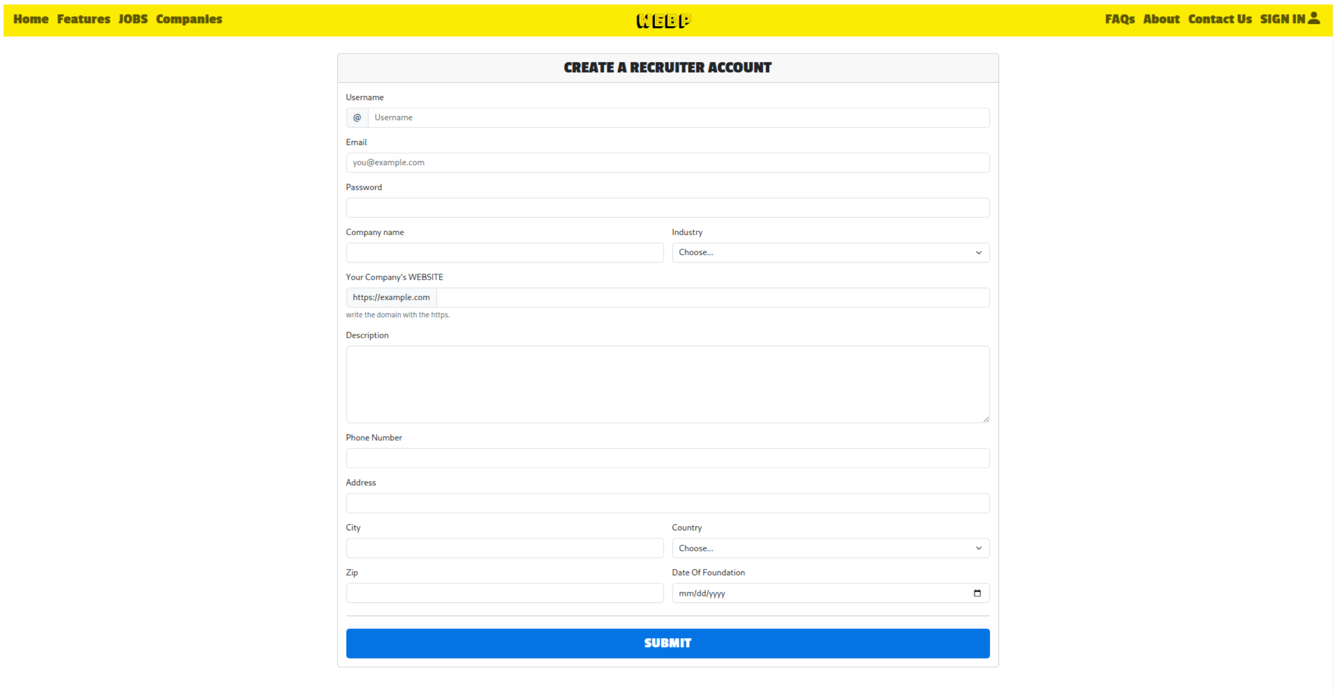
Subscribe to our newsletter

Monthly digest of what's new and exciting from us.

Email address

Subscribe

1.4 Create Recruiter:


Home Features JOBS Companies WEBP FAQs About Contact Us SIGN IN

CREATE A RECRUITER ACCOUNT

Username
@ Username

Email
you@example.com

Password

Company name

Industry
Choose...

Your Company's WEBSITE
https://example.com
write the domain with the https.

Description

Phone Number

Address

City

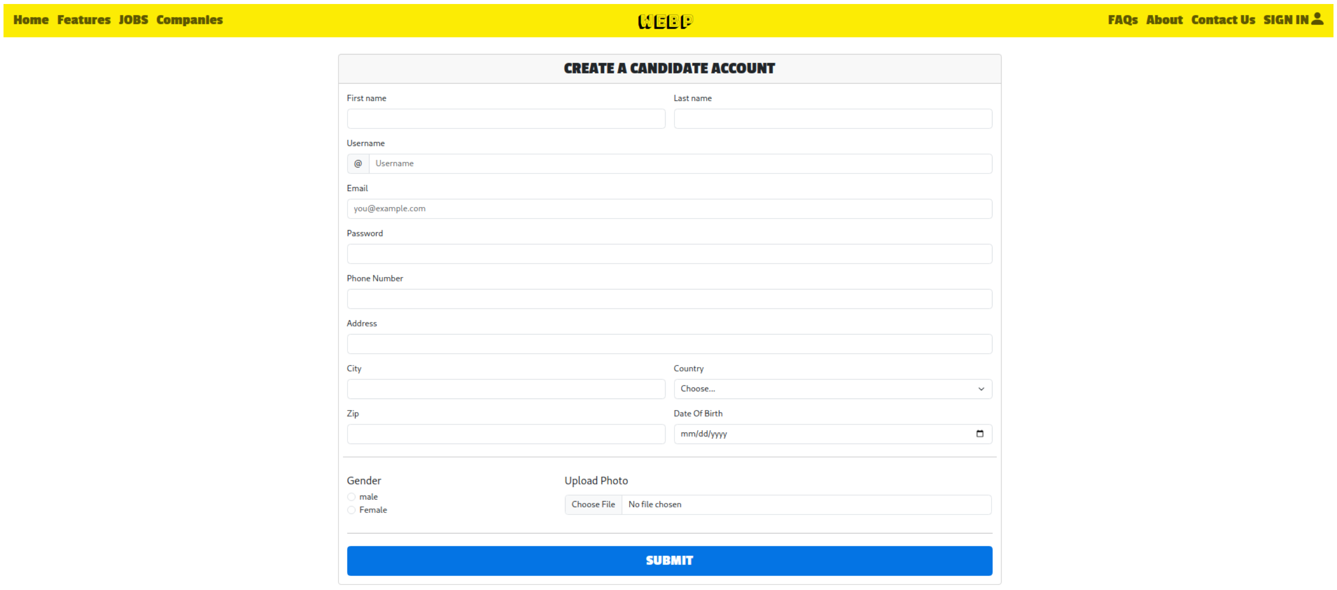
Country
Choose...

Zip

Date Of Foundation
mm/dd/yyyy

SUBMIT

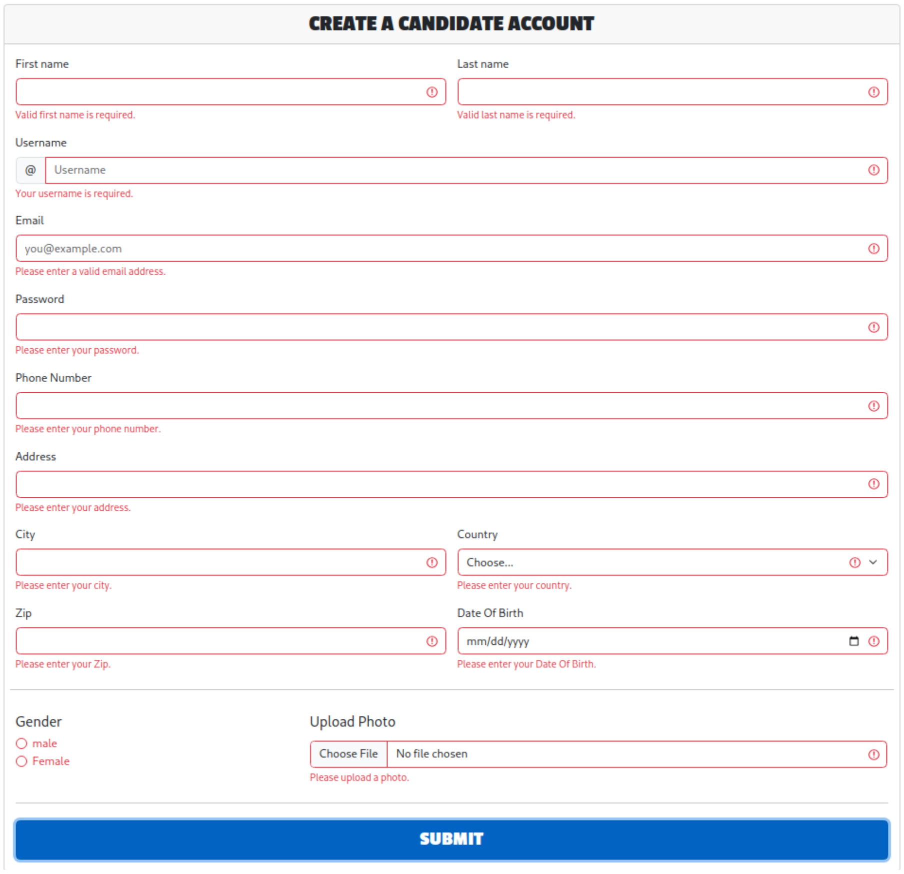
1.4 Create Candidate:



CREATE A CANDIDATE ACCOUNT

First name
Last name
Username
@ Username
Email
you@example.com
Password
Phone Number
Address
City
Country
Choose...
Zip
Date Of Birth
mm/dd/yyyy
Gender
male
Female
Upload Photo
Choose File No file chosen
SUBMIT

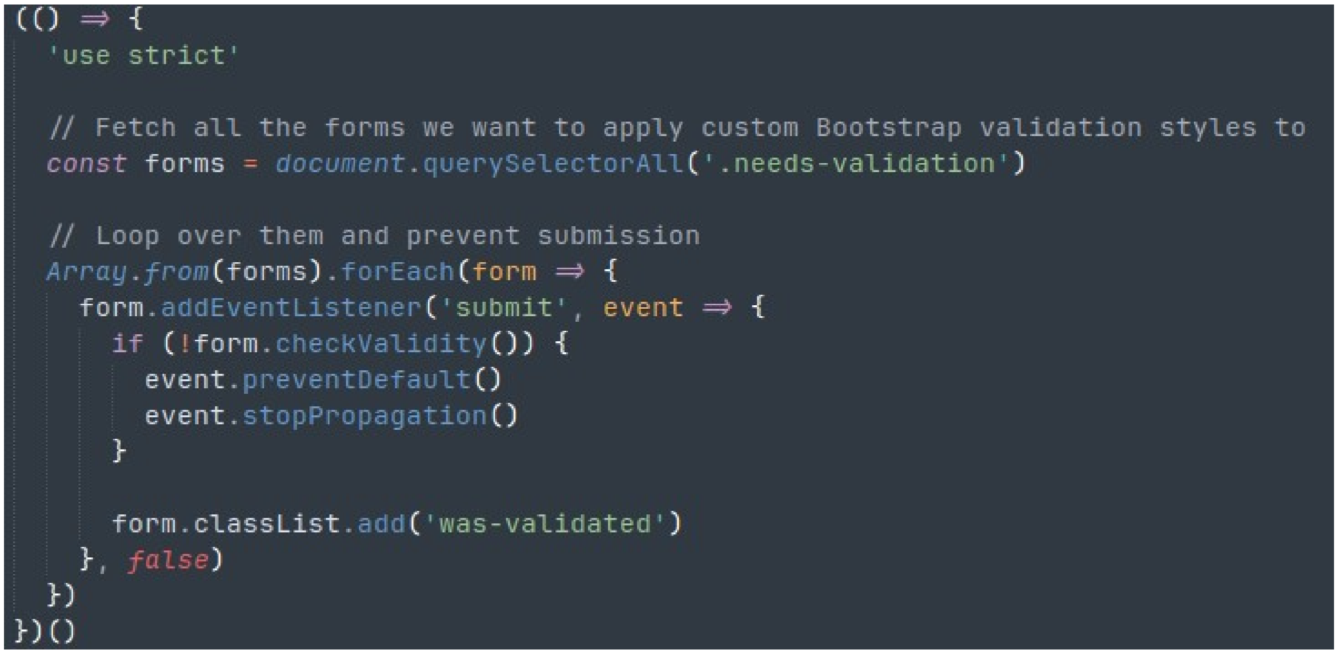
submit check using js



CREATE A CANDIDATE ACCOUNT

First name
Valid first name is required.
Last name
Valid last name is required.
Username
@ Username
Your username is required.
Email
you@example.com
Please enter a valid email address.
Password
Please enter your password.
Phone Number
Please enter your phone number.
Address
Please enter your address.
City
Please enter your city.
Country
Choose...
Please enter your country.
Zip
Please enter your Zip.
Date Of Birth
mm/dd/yyyy
Please enter your Date Of Birth.
Gender
male
Female
Upload Photo
Choose File No file chosen
Please upload a photo.
SUBMIT

js code located at js/submit_check.js



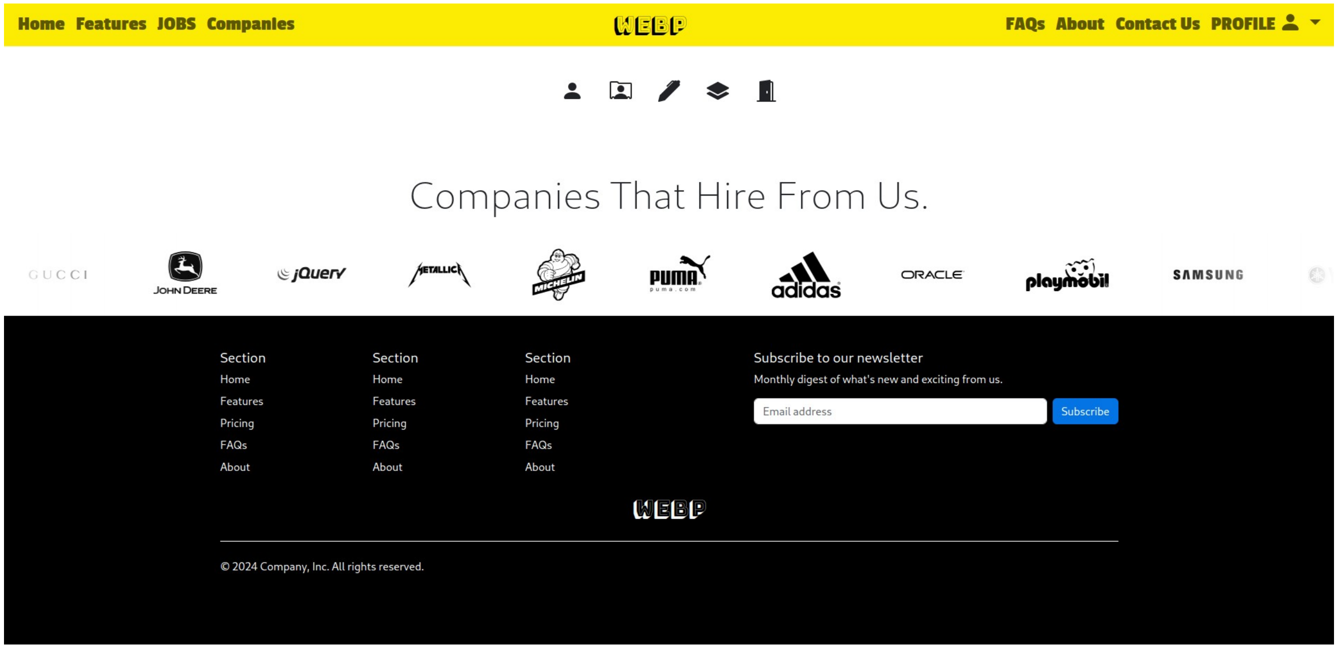
```
(() => {
  'use strict'

  // Fetch all the forms we want to apply custom Bootstrap validation styles to
  const forms = document.querySelectorAll('.needs-validation')

  // Loop over them and prevent submission
  Array.from(forms).forEach(form => {
    form.addEventListener('submit', event => {
      if (!form.checkValidity()) {
        event.preventDefault()
        event.stopPropagation()
      }

      form.classList.add('was-validated')
    }, false)
  })
})()
```

well I believe duts it for the home page, the footer still needs some styling doe.

RECRUITER

1. Recruiter’s magic menu:

the homepage/jobs are the same.



Companies That Hire From Us.

Subscribe to our newsletter

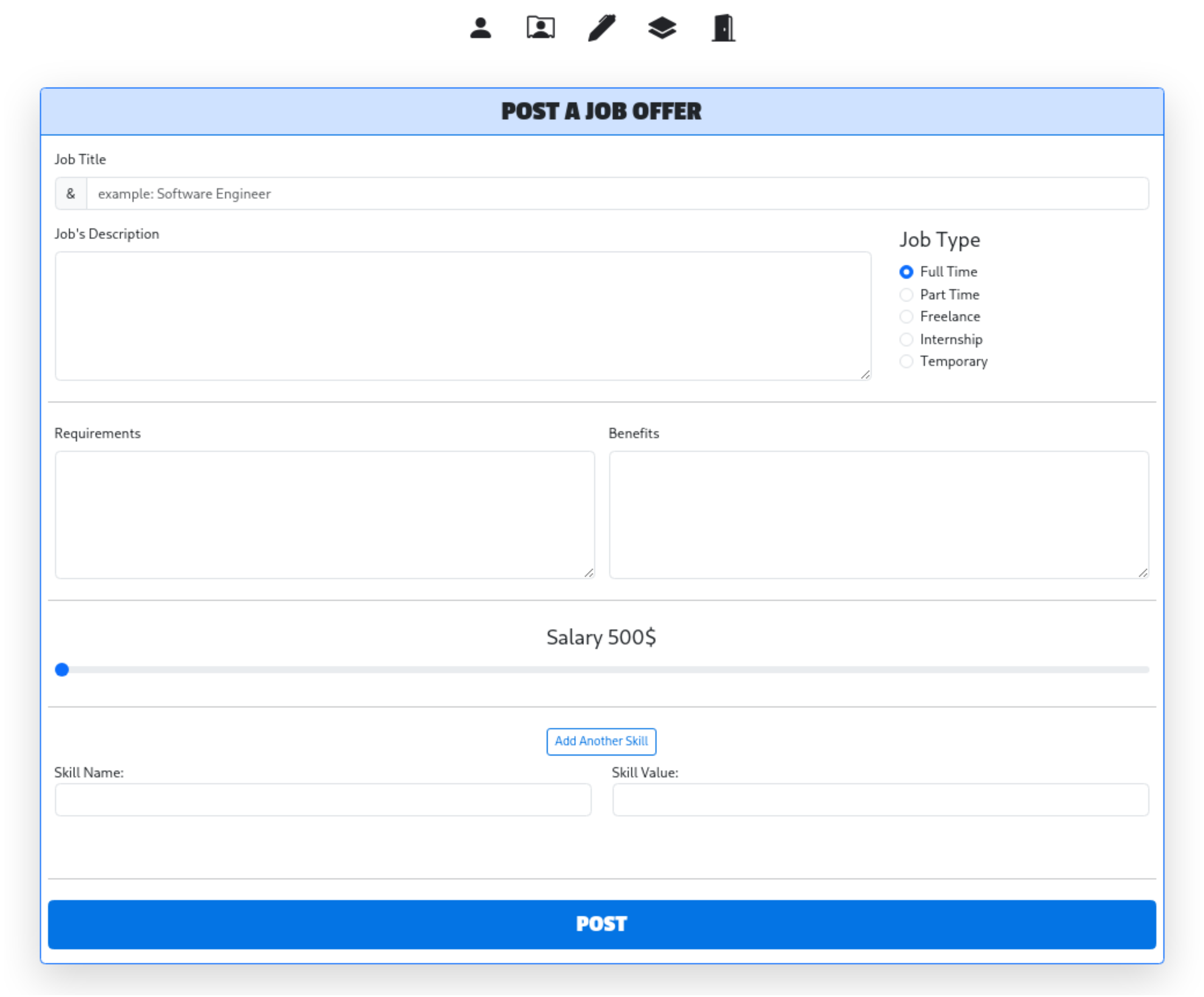
Monthly digest of what's new and exciting from us.

© 2024 Company, Inc. All rights reserved.

Youtube guide:
https://www.youtube.com/watch?v=ArTVfdHOB-M&t=629s

1.1 Posting a Job Offer:

the script that is responsible for creating a job offer is located at phpcode/create_job_offer_code.php, it inserts the values into the job_offer table first then it proceeds to insert the skills prompted by the recruiter into the skills table, of course the skills table needs the rec_id alongside the job_id,


POST A JOB OFFER

Job Title

& example: Software Engineer

Job's Description

Job Type

[✓ Full Time]
Part Time
Freelance
Internship
Temporary

Requirements

Benefits

Salary 500$

Add Another Skill

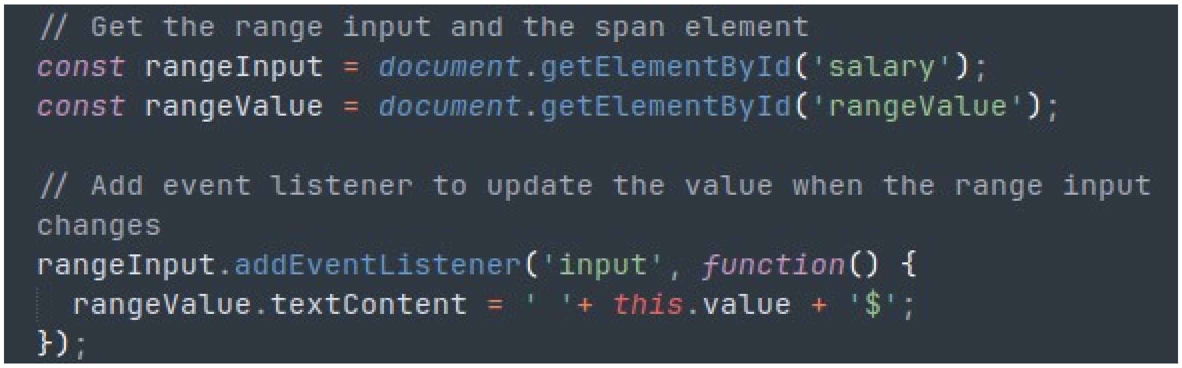
Skill Name:

Skill Value:

POST

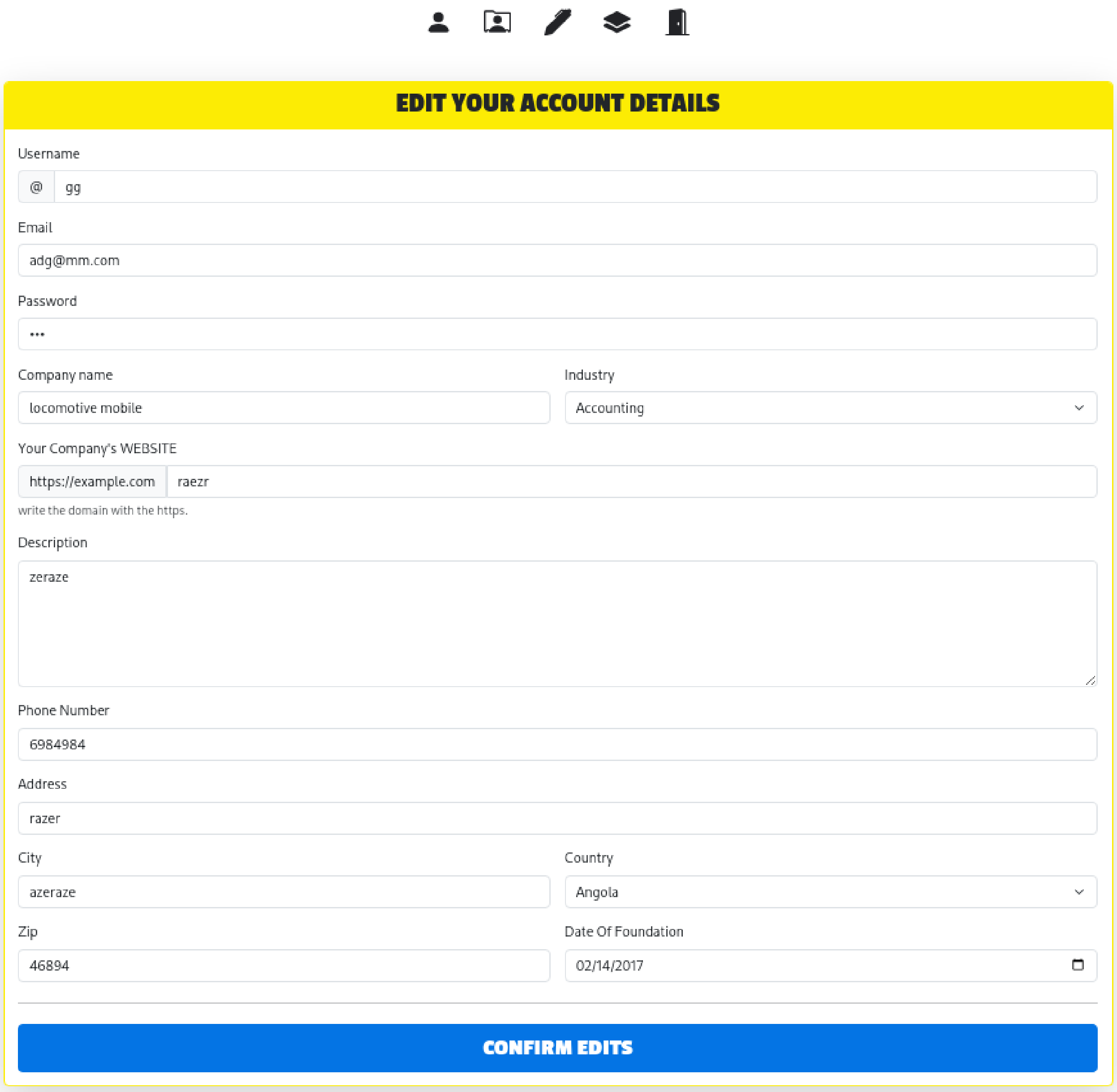
we will need those two foreign keys when the candidate applies to a job.

The js code needed to dynamically update the salary's value when modifying the range's position.


```
// Get the range input and the span element
const rangeInput = document.getElementById('salary');
const rangeValue = document.getElementById('rangeValue');

// Add event listener to update the value when the range input
changes
rangeInput.addEventListener('input', function() {
  rangeValue.textContent = ' '+ this.value + '$';
});
```

Located at post_job_offer.php

1.1 Edit Recruiter:



EDIT YOUR ACCOUNT DETAILS

Username
@ gg

Email
adg@mm.com

Password
•••

Company name
locomotive mobile

Industry
Accounting

Your Company's WEBSITE
https://example.com raezr
write the domain with the https.

Description
zeraze

Phone Number
6984984

Address
razer

City
azeraze

Country
Angola

Zip
46894

Date Of Foundation
02/14/2017

CONFIRM EDITS

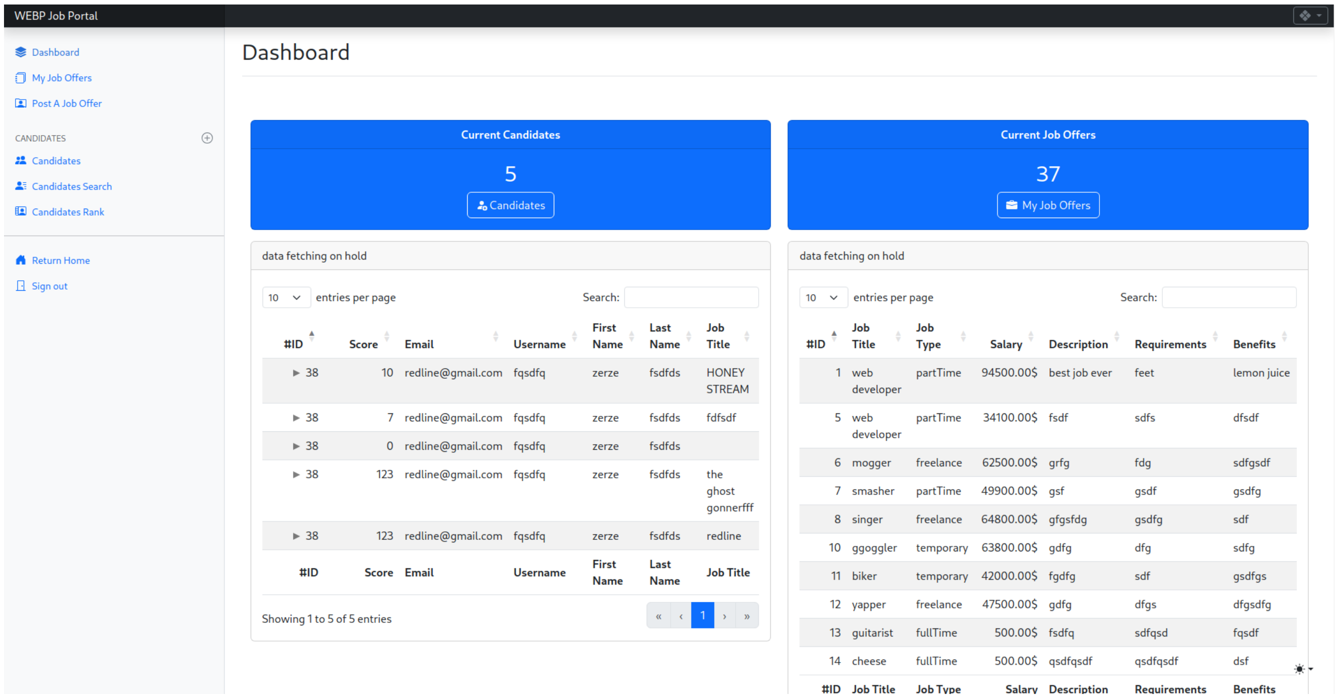
sql query:

```
if ($rec_id !== '' && $email !== '' && $username !
== '' && $password !== '' && $companyname !== '' &&
$industry !== '' && $website !== '' && $description
!== '' && $phone !== '' && $address !== '' && $city
!== '' && $country !== '' && $zip !== '' &&
$foundation !== '') {

  $query=mysqli_query($conn, "UPDATE Recruiters SET
              email = '$email',
              username = '$username',
              password = '$password',
              companyname = '$companyname',
              industry = '$industry',
              website = '$website',
              description = '$description',
              phone = '$phone',
              address = '$address',
              city = '$city',
              country = '$country',
              zip_code = '$zip',
              foundation = '$foundation'
              WHERE rec_id='$rec_id'");

page located at: edit_recruiter.php
script at: phpcode/edit_recruiter_code.php
```

2. Recruiter’s Dashboard:

same stuff as the admin’s dashboard except no charts.

Locatio, DASHBOARD/rec_dashboard.php


WEBP Job Portal

Dashboard
My Job Offers
Post A Job Offer
CANDIDATES
Candidates
Candidates Search
Candidates Rank
Return Home
Sign out

Dashboard

Current Candidates
5
Candidates

Current Job Offers
37
My Job Offers

data fetching on hold

10 entries per page Search:

#ID	Score	Email	Username	First Name	Last Name	Job Title
▶ 38	10	redline@gmail.com	fqsdfq	zerze	fsdfds	HONEY STREAM
▶ 38	7	redline@gmail.com	fqsdfq	zerze	fsdfds	fdfsdf
▶ 38	0	redline@gmail.com	fqsdfq	zerze	fsdfds	
▶ 38	123	redline@gmail.com	fqsdfq	zerze	fsdfds	the ghost gonnerfff
▶ 38	123	redline@gmail.com	fqsdfq	zerze	fsdfds	redline
#ID	Score	Email	Username	First Name	Last Name	Job Title

Showing 1 to 5 of 5 entries « ‹ 1 › »

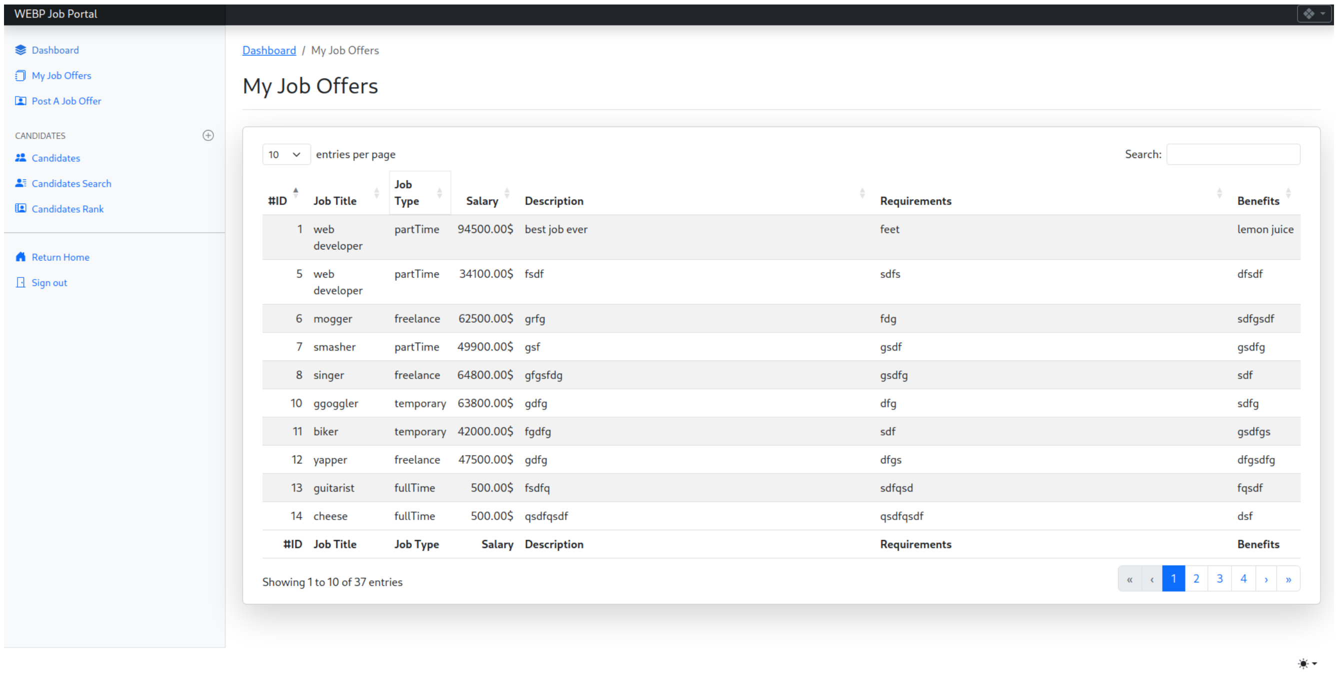
data fetching on hold

10 entries per page Search:

#ID	Job Title	Job Type	Salary	Description	Requirements	Benefits
1	web developer	partTime	94500.00$	best job ever	feet	lemon juice
5	web developer	partTime	34100.00$	fsdf	sdfs	dfsdf
6	mogger	freelance	62500.00$	grfg	fdg	sdfgsdf
7	smasher	partTime	49900.00$	gsf	gsdf	gsdfg
8	singer	freelance	64800.00$	gfgsfdg	gsdfg	sdf
10	ggoggler	temporary	63800.00$	gdfg	dfg	sdfg
11	biker	temporary	42000.00$	fgdfg	sdf	gsdfgs
12	yapper	freelance	47500.00$	gdfg	dfgs	dfgsdfg
13	guitarist	fullTime	500.00$	fsdfq	sdfqsd	fqsdf
14	cheese	fullTime	500.00$	qsdfqsdf	qsdfqsdf	dsf
#ID	Job Title	Job Type	Salary	Description	Requirements	Benefits

2.1 Sql queries:

My Job Offers SELECT * FROM job_offer j INNER JOIN Recruiters r ON r.rec_id = j.rec_id WHERE j.status = 1;


WEBP Job Portal

Dashboard
My Job Offers
Post A Job Offer
CANDIDATES
Candidates
Candidates Search
Candidates Rank
Return Home
Sign out

Dashboard / My Job Offers

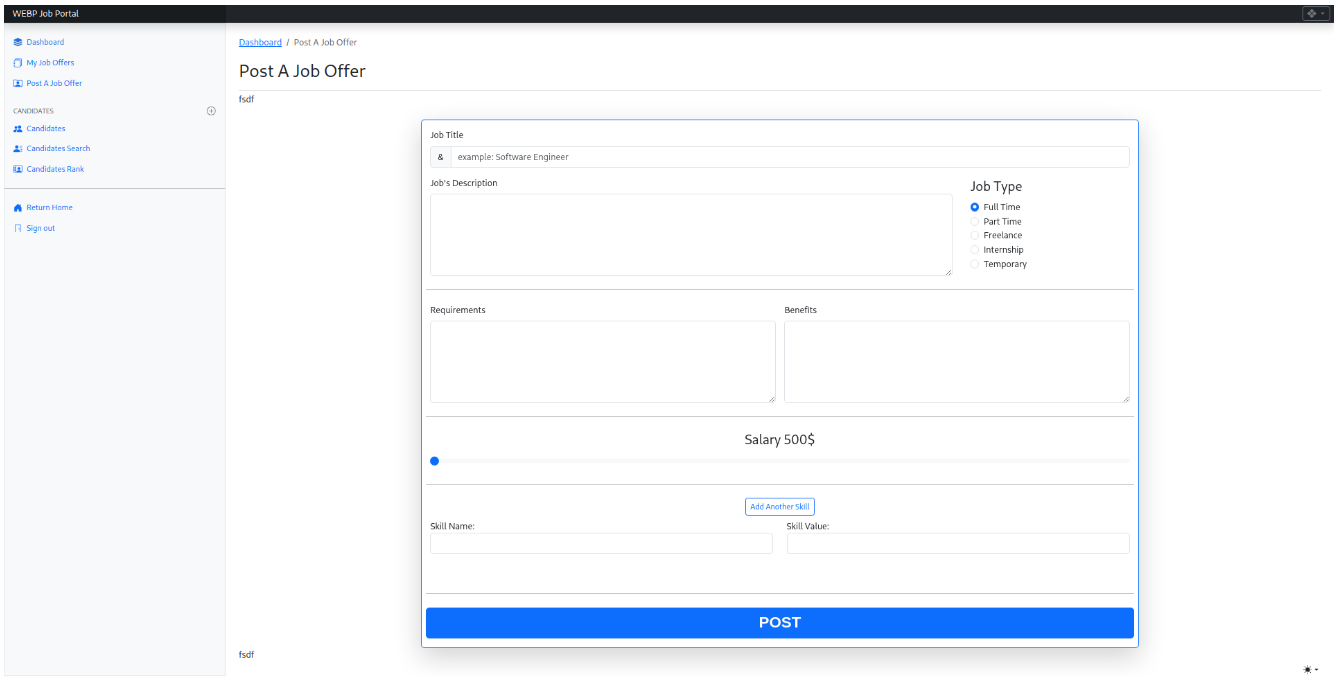
My Job Offers

10 entries per page Search:

#ID	Job Title	Job Type	Salary	Description	Requirements	Benefits
1	web developer	partTime	94500.00$	best job ever	feet	lemon juice
5	web developer	partTime	34100.00$	fsdf	sdfs	dfsdf
6	mogger	freelance	62500.00$	grfg	fdg	sdfgsdf
7	smasher	partTime	49900.00$	gsf	gsdf	gsdfg
8	singer	freelance	64800.00$	gfgsfdg	gsdfg	sdf
10	ggoggler	temporary	63800.00$	gdfg	dfg	sdfg
11	biker	temporary	42000.00$	fgdfg	sdf	gsdfgs
12	yapper	freelance	47500.00$	gdfg	dfgs	dfgsdfg
13	guitarist	fullTime	500.00$	fsdfq	sdfqsd	fqsdf
14	cheese	fullTime	500.00$	qsdfqsdf	qsdfqsdf	dsf
#ID	Job Title	Job Type	Salary	Description	Requirements	Benefits

Showing 1 to 10 of 37 entries « ‹ 1 2 3 4 › »

Post A Job Offer same as the post from the magic menu



WEBP Job Portal

Dashboard
My Job Offers
Post A Job Offer
CANDIDATES
Candidates
Candidates Search
Candidates Rank
Return Home
Sign out

Dashboard / Post A Job Offer

Post A Job Offer

fsdf

Job Title

example: Software Engineer

Job's Description

Job Type

- Full Time
- Part Time
- Freelance
- Internship
- Temporary

Requirements

Benefits

Salary 500$

Add Another Skill

Skill Name:

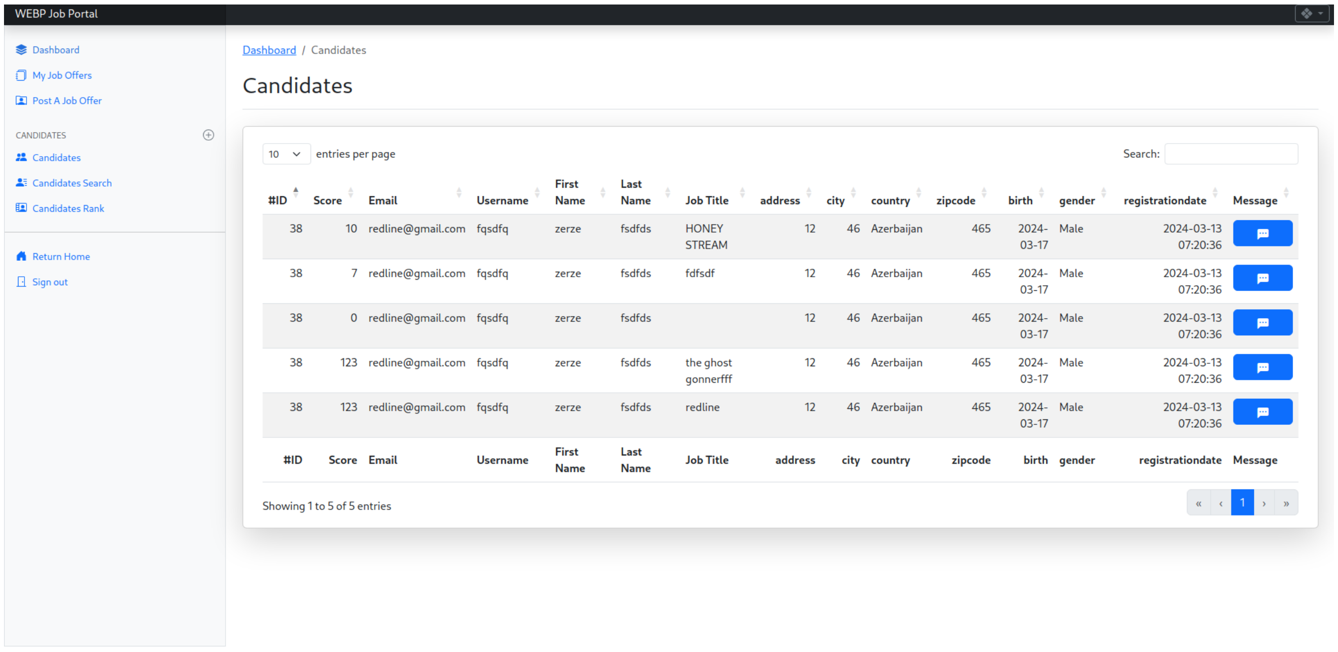
Skill Value:

POST

fsdf

Candidates SELECT * FROM Candidates c INNER JOIN applied_jobs a ON c.can_id = a.Can_id INNER JOIN job_offer j ON a.Job_id = j.job_id WHERE c.status = 1 AND j.rec_id = $recid;

$recid: session global var.



WEBP Job Portal

Dashboard
My Job Offers
Post A Job Offer
CANDIDATES
Candidates
Candidates Search
Candidates Rank
Return Home
Sign out

Dashboard / Candidates

Candidates

10 entries per page

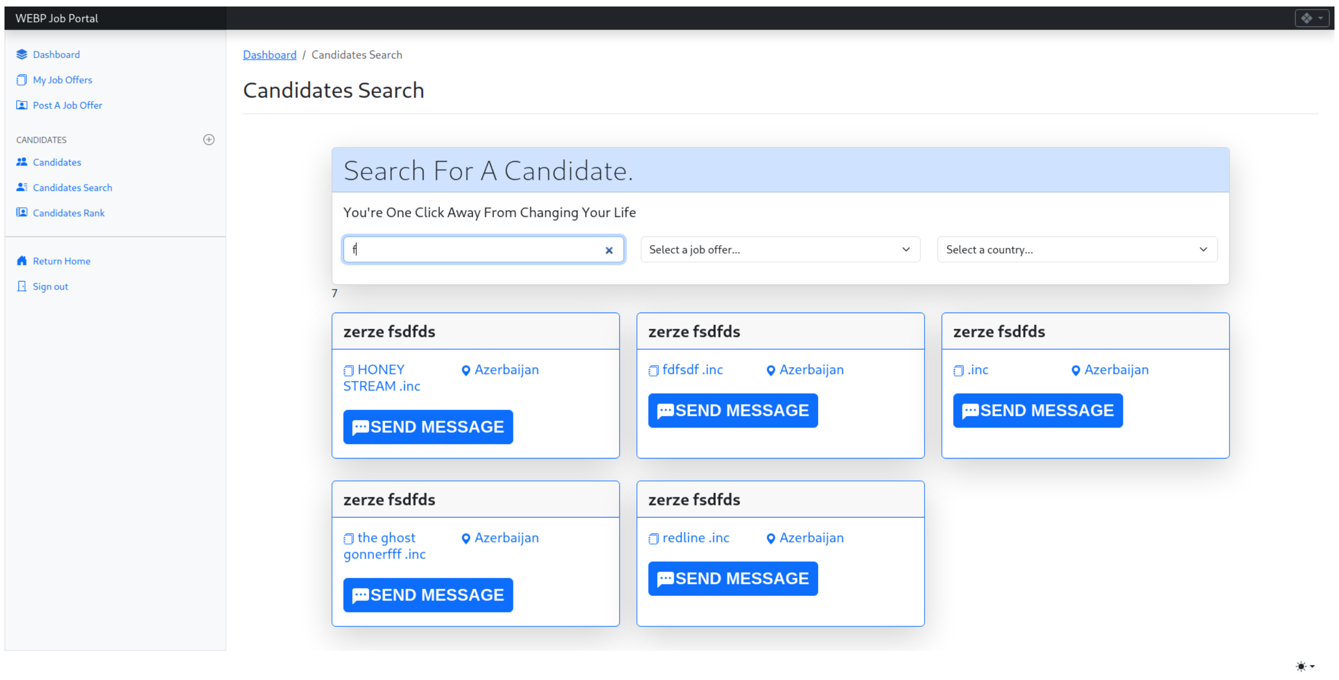
Search:

#ID	Score	Email	Username	First Name	Last Name	Job Title	address	city	country	zipcode	birth	gender	registrationdate	Message
38	10	redline@gmail.com	fqsdfq	zerze	fsdfds	HONEY STREAM	12	46	Azerbaijan	465	2024-03-17	Male	2024-03-13 07:20:36	
38	7	redline@gmail.com	fqsdfq	zerze	fsdfds	fdfsdf	12	46	Azerbaijan	465	2024-03-17	Male	2024-03-13 07:20:36	
38	0	redline@gmail.com	fqsdfq	zerze	fsdfds		12	46	Azerbaijan	465	2024-03-17	Male	2024-03-13 07:20:36	
38	123	redline@gmail.com	fqsdfq	zerze	fsdfds	the ghost gonnerfff	12	46	Azerbaijan	465	2024-03-17	Male	2024-03-13 07:20:36	
38	123	redline@gmail.com	fqsdfq	zerze	fsdfds	redline	12	46	Azerbaijan	465	2024-03-17	Male	2024-03-13 07:20:36	
#ID	Score	Email	Username	First Name	Last Name	Job Title	address	city	country	zipcode	birth	gender	registrationdate	Message

Showing 1 to 5 of 5 entries

« ‹ 1 › »

Candidates Search



WEBP Job Portal

Dashboard
My Job Offers
Post A Job Offer

CANDIDATES
Candidates
Candidates Search
Candidates Rank

Return Home
Sign out

Dashboard / Candidates Search

Candidates Search

Search For A Candidate.

You're One Click Away From Changing Your Life

f

Select a job offer...

Select a country...

7

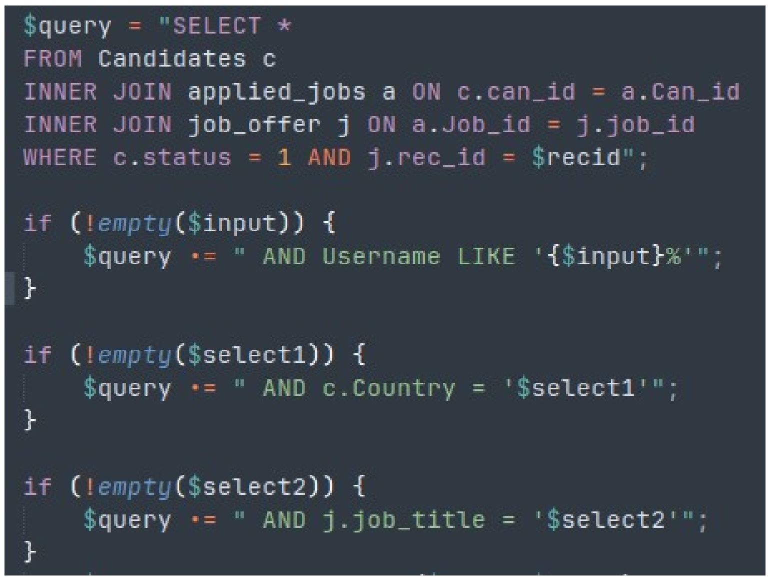
zerze fsdfds
HONEY STREAM .inc — Azerbaijan
SEND MESSAGE

zerze fsdfds
fdfsdf .inc — Azerbaijan
SEND MESSAGE

zerze fsdfds
.inc — Azerbaijan
SEND MESSAGE

zerze fsdfds
the ghost gonnerfff .inc — Azerbaijan
SEND MESSAGE

zerze fsdfds
redline .inc — Azerbaijan
SEND MESSAGE



```php
$query = "SELECT *
FROM Candidates c
INNER JOIN applied_jobs a ON c.can_id = a.Can_id
INNER JOIN job_offer j ON a.Job_id = j.job_id
WHERE c.status = 1 AND j.rec_id = $recid";

if (!empty($input)) {
    $query .= " AND Username LIKE '{$input}%'";
}

if (!empty($select1)) {
    $query .= " AND c.Country = '$select1'";
}

if (!empty($select2)) {
    $query .= " AND j.job_title = '$select2'";
}
```

search by name, job title and candidate's country.

location: DASHBOARD/candidates_search.php
code: DASHBOARD/phpcode/candidates_search_code.php

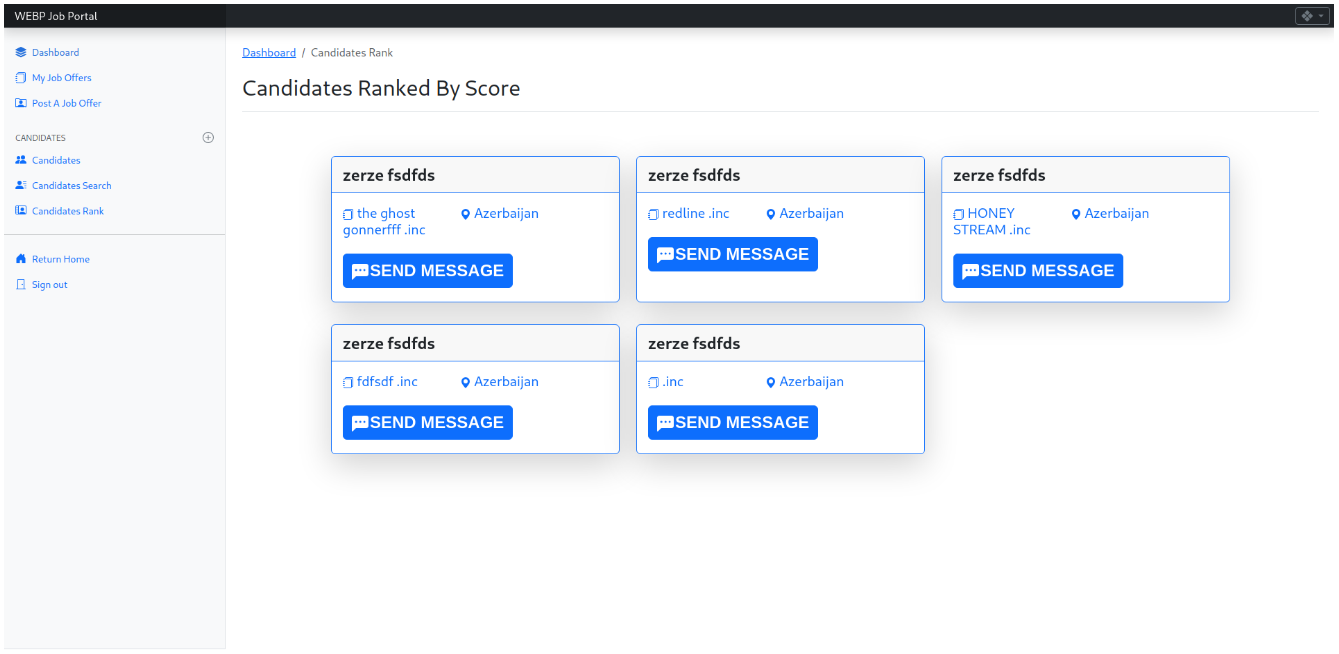
of course once again we using ajax and the 6 cards layout display with a message btn.

Candidates Rank

```
SELECT *
           FROM Candidates c
INNER JOIN applied_jobs a ON c.can_id = a.Can_id
INNER JOIN job_offer j ON a.Job_id = j.job_id
WHERE c.status = 1 AND j.rec_id = $recid
ORDER BY a.score DESC
```

Displaying candidates in descending order based on their score.

To enhance the presentation further, consider incorporating candidate images alongside their scores.


WEBP Job Portal

Dashboard
My Job Offers
Post A Job Offer
CANDIDATES
Candidates
Candidates Search
Candidates Rank
Return Home
Sign out

Dashboard / Candidates Rank

Candidates Ranked By Score

zerze fsdfds the ghost gonnerfff .inc — Azerbaijan SEND MESSAGE	zerze fsdfds redline .inc — Azerbaijan SEND MESSAGE	zerze fsdfds HONEY STREAM .inc — Azerbaijan SEND MESSAGE
zerze fsdfds fdfsdf .inc — Azerbaijan SEND MESSAGE	zerze fsdfds .inc — Azerbaijan SEND MESSAGE	

2.1 Messaging using a simple database table:

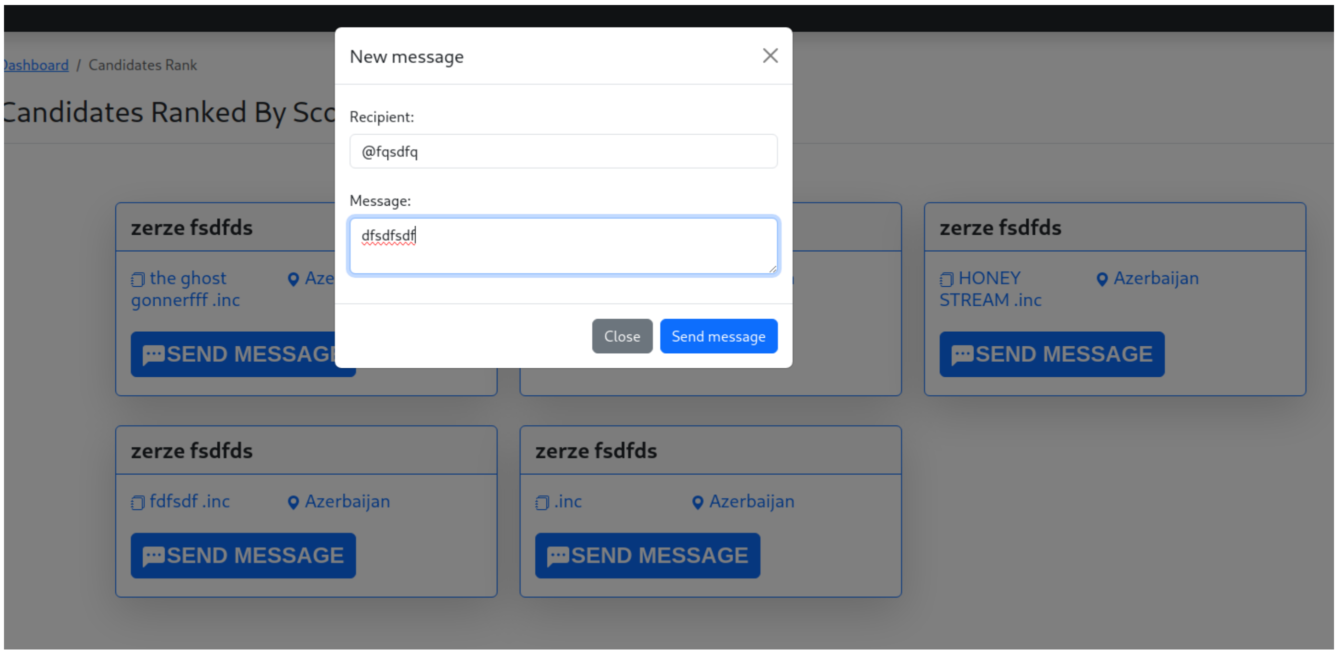
Recruiters can send messages to candidates, but candidates are unable to reply.

XML HttpRequest

Here is the youtube video from which I got the idea:
https://www.youtube.com/watch?v=SkQTMwGLbh4&list=PLVQcN4_vTIsqIrgmJBKlWbNQt4UxOv3ly&index=3

literally everything is explained in this page:
https://www.w3schools.com/xml/xml_http.asp

now the challenge occurs when we wanna send the id's to the modal "send message btn", I already did that the same principal as the confirm delete modal.



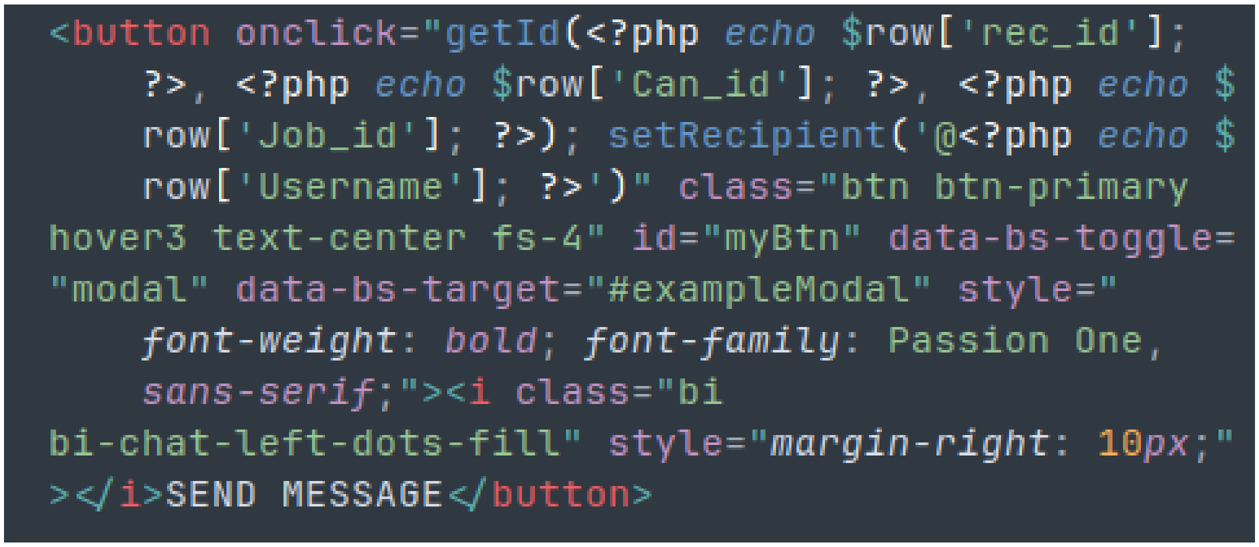
New message

Recipient:
@fqsdfq

Message:
dfsdfsdf

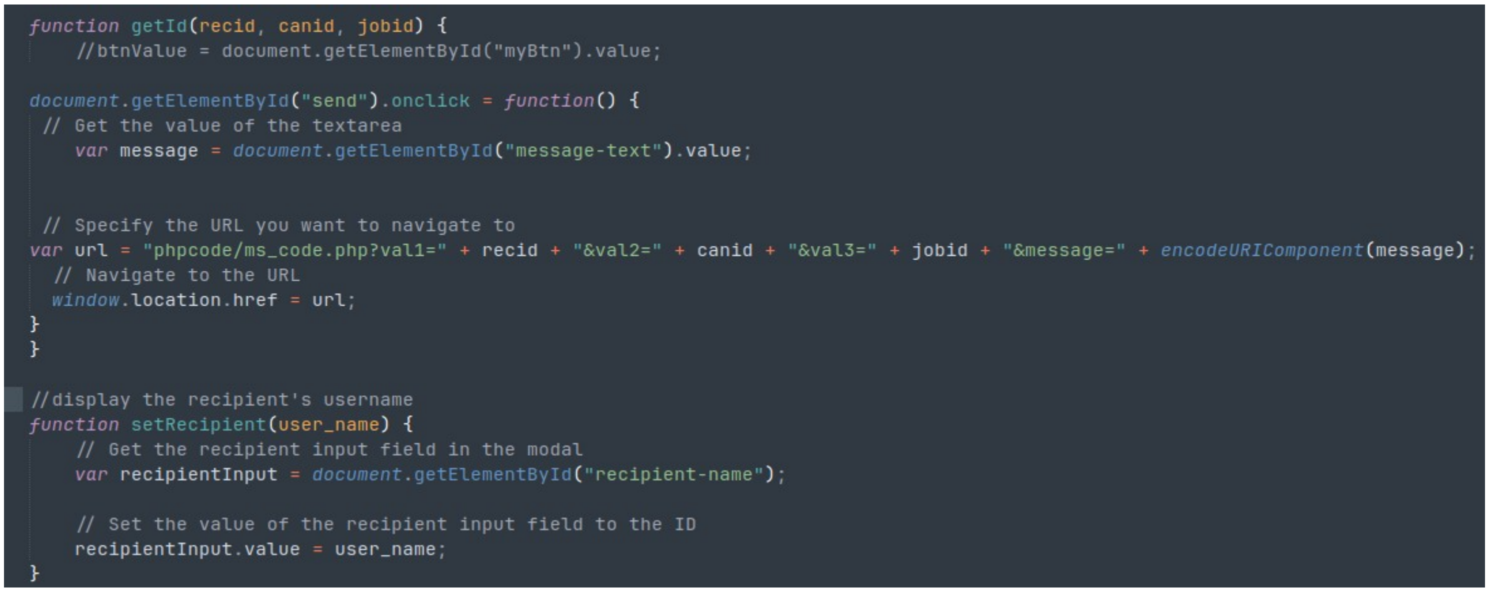
```
Sql query: INSERT INTO messages (message, rec_id,
can_id, job_id) VALUES('$message', '$value1',
'$value2', '$value3');
```

the send message btn:



```
<button onclick="getId(<?php echo $row['rec_id'];
    ?>, <?php echo $row['Can_id']; ?>, <?php echo $
    row['Job_id']; ?>); setRecipient('@<?php echo $
    row['Username']; ?>')" class="btn btn-primary
hover3 text-center fs-4" id="myBtn" data-bs-toggle=
"modal" data-bs-target="#exampleModal" style="
    font-weight: bold; font-family: Passion One,
    sans-serif;"><i class="bi
bi-chat-left-dots-fill" style="margin-right: 10px;"
></i>SEND MESSAGE</button>
```

the js:

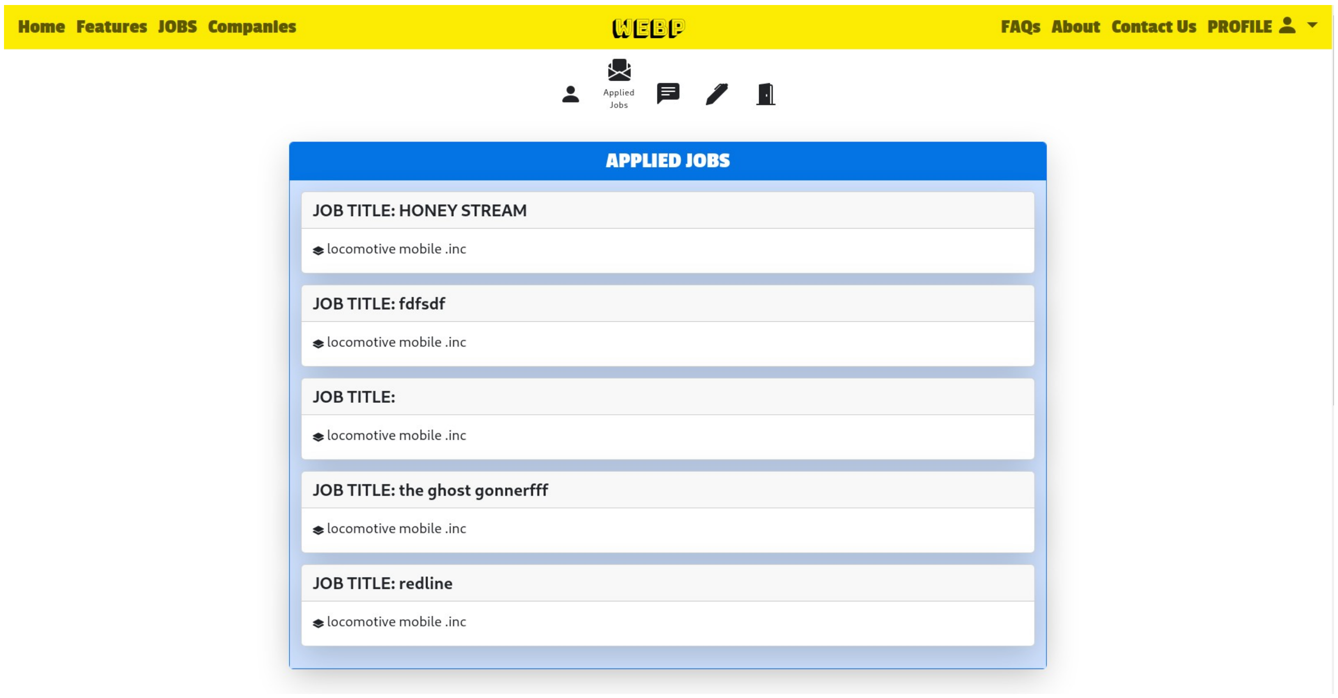


```
function getId(recid, canid, jobid) {
    //btnValue = document.getElementById("myBtn").value;

document.getElementById("send").onclick = function() {
  // Get the value of the textarea
    var message = document.getElementById("message-text").value;


  // Specify the URL you want to navigate to
var url = "phpcode/ms_code.php?val1=" + recid + "&val2=" + canid + "&val3=" + jobid + "&message=" + encodeURIComponent(message);
   // Navigate to the URL
   window.location.href = url;
}
}

//display the recipient's username
function setRecipient(user_name) {
    // Get the recipient input field in the modal
    var recipientInput = document.getElementById("recipient-name");

    // Set the value of the recipient input field to the ID
    recipientInput.value = user_name;
}
```

located at: DASHBOARD/candidates_rank.php
code: DASHBOARD/phpcode/message_code.php

CANDIDATE

nothing new here except for two sections on the magic menu



APPLIED JOBS

JOB TITLE: HONEY STREAM
locomotive mobile .inc

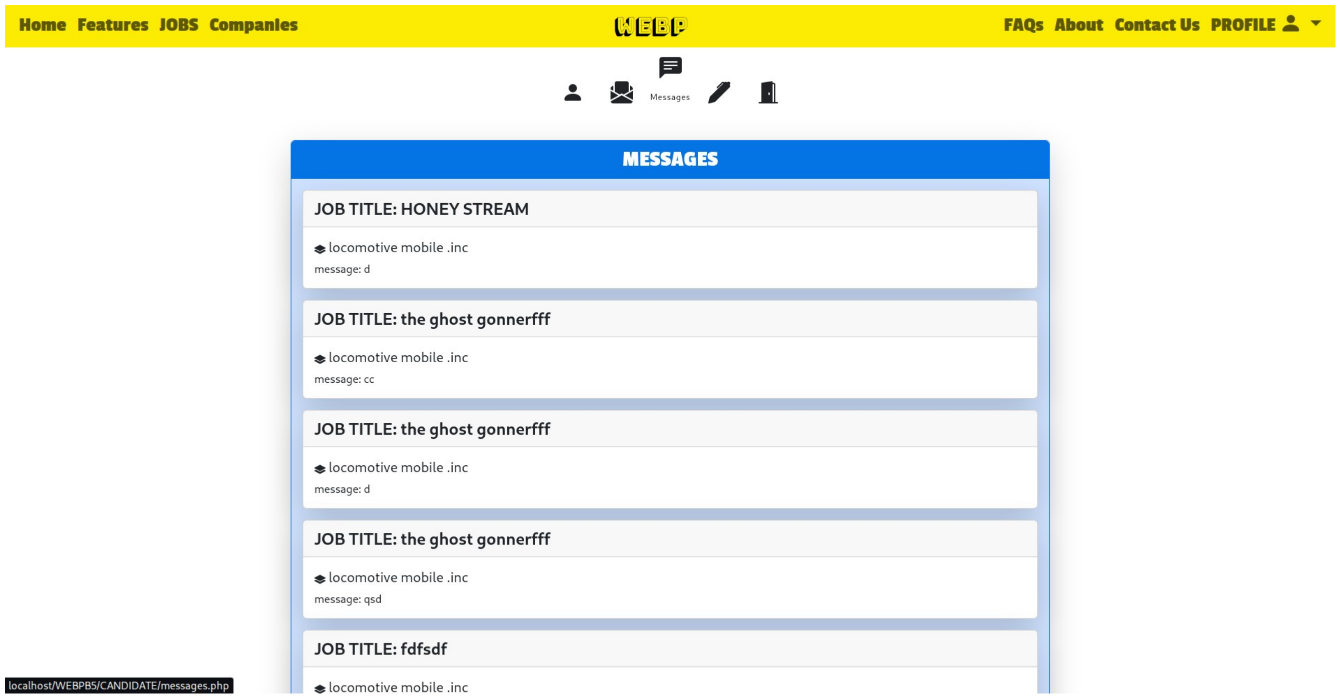
JOB TITLE: fdfsdf
locomotive mobile .inc

JOB TITLE:
locomotive mobile .inc

JOB TITLE: the ghost gonnerfff
locomotive mobile .inc

JOB TITLE: redline
locomotive mobile .inc

sql query: SELECT * FROM applied_jobs a INNER JOIN job_offer j ON a.Job_id = j.job_id INNER JOIN Recruiters r ON r.rec_id = j.rec_id
  WHERE a.Can_id = $can_id";
$can_id: session global var.


Home Features JOBS Companies
WEBP
FAQs About Contact Us PROFILE
Messages
localhost/WEBPB5/CANDIDATE/messages.php

MESSAGES

JOB TITLE: HONEY STREAM
locomotive mobile .inc
message: d

JOB TITLE: the ghost gonnerfff
locomotive mobile .inc
message: cc

JOB TITLE: the ghost gonnerfff
locomotive mobile .inc
message: d

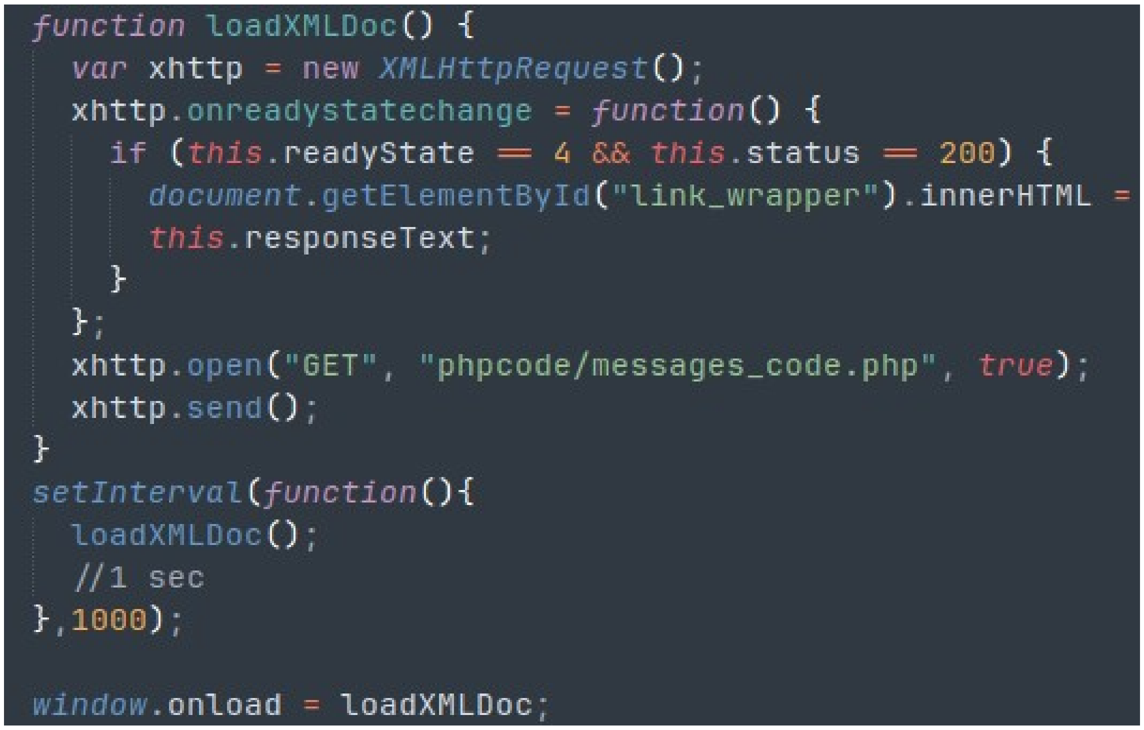
JOB TITLE: the ghost gonnerfff
locomotive mobile .inc
message: qsd

JOB TITLE: fdfsdf
locomotive mobile .inc

sql query: SELECT *
FROM messages m
INNER JOIN job_offer j ON m.job_id = j.job_id
INNER JOIN Recruiters r ON r.rec_id = m.rec_id
WHERE m.can_id =$id"

$id = $_SESSION['can_id'];

ajax stuff:

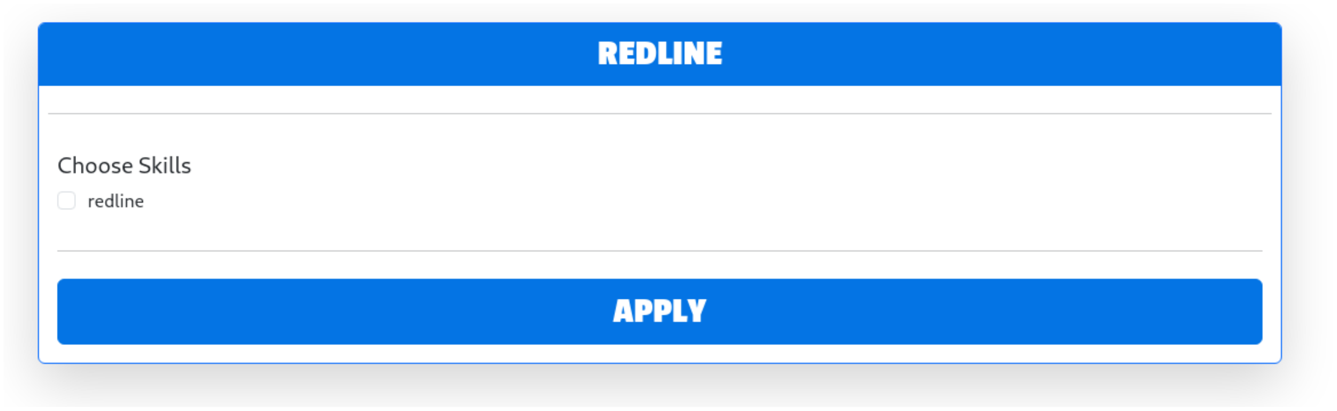


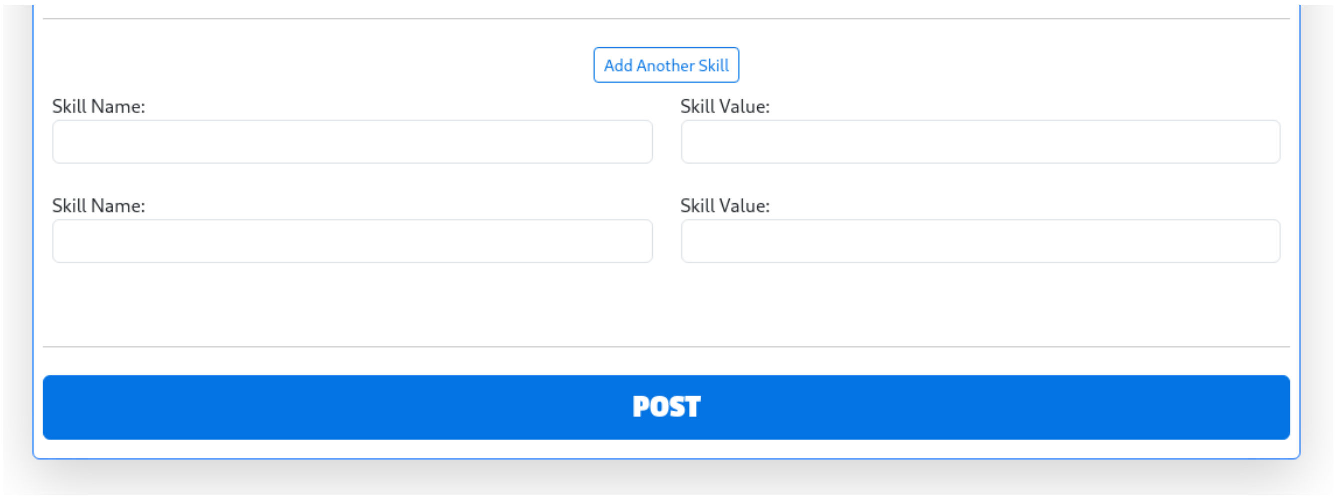
```
function loadXMLDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("link_wrapper").innerHTML =
      this.responseText;
    }
  };
  xhttp.open("GET", "phpcode/messages_code.php", true);
  xhttp.send();
}
setInterval(function(){
  loadXMLDoc();
  //1 sec
},1000);

window.onload = loadXMLDoc;
```

in simple terms, whenever a new row is added to the messages table, the script messages_code.php runs.

Oh yeah I almost forgot the scoring.

Apply to jobs, I didn’t really vibe with this thing that much, you could check the score directory if you wanna try a simple example.
RECRUITER/score





jufu

CSS BASIC ANIMATIONS

@media RULE:

https://www.w3schools.com/cssref/css3_pr_mediaquery.php

media is used to achieve a responsive web design.

THE HOME PAGE SLIDE ANIMATIONS:

there are 8 of them, named scroll#
located at css/scroll.css

incorporating these animations into a static website is incredibly straightforward, the issue arises when we use ajax, they wont work without using this thing called IntersectionObserver.

The **IntersectionObserver** interface of the Intersection Observer API provides a way to asynchronously observe changes in the intersection of a target element with an ancestor element or with a top-level document's viewport. The ancestor element or viewport is referred to as the root.

When an IntersectionObserver is created, it's configured to watch for given ratios of visibility within the root. The configuration cannot be changed once the IntersectionObserver is created, so a given observer object is only useful for watching for specific changes in degree of visibility; however, you can watch multiple target elements with the same observer.

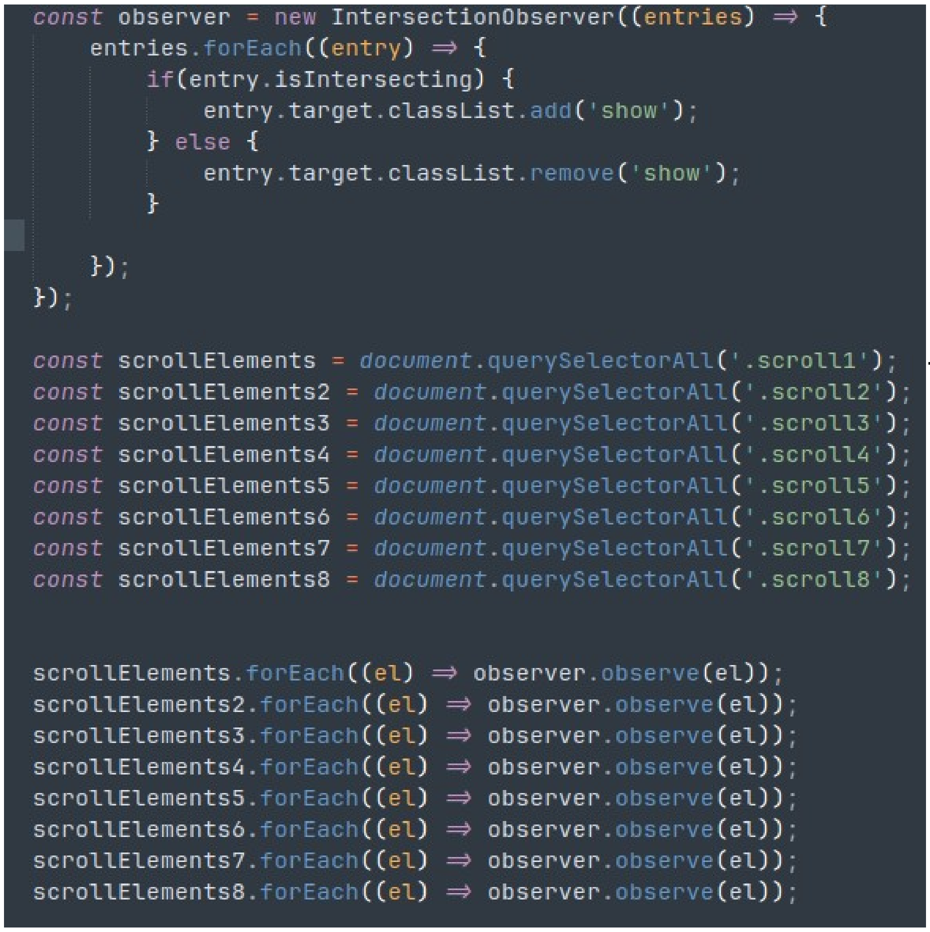
Read more about IntersectionObserver:
https://developer.mozilla.org/en-US/docs/Web/API/IntersectionObserver

we getting a question about this thing during the presentation for sure.

here is how I stumbled upon this api:
https://www.youtube.com/watch?v=T33NN_pPeNI

definition: IntersectionObserver applies animations to elements on the webpage based on their visibility in the viewport, enhancing the user experience by adding dynamic visual effects as the user scrolls through the content.

Yeah I used it on the 6 cards layout animations.

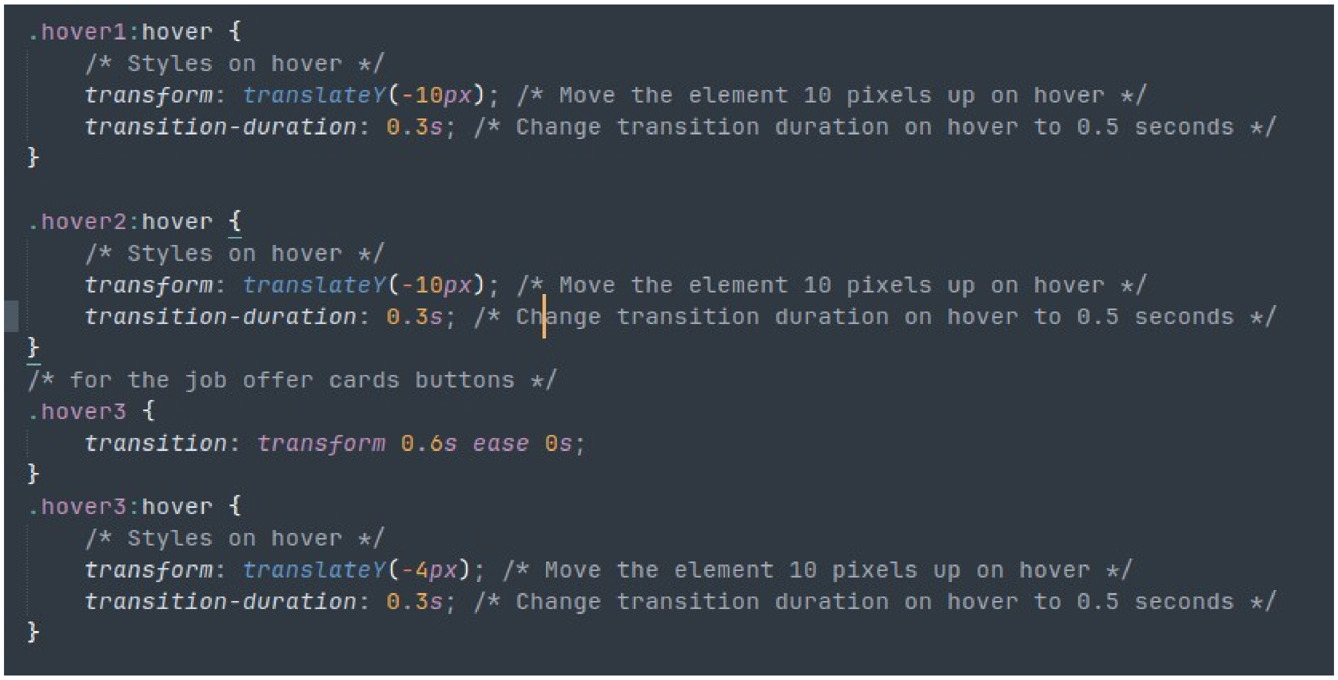


```
const observer = new IntersectionObserver((entries) => {
    entries.forEach((entry) => {
        if(entry.isIntersecting) {
            entry.target.classList.add('show');
        } else {
            entry.target.classList.remove('show');
        }

    });
});

const scrollElements = document.querySelectorAll('.scroll1');
const scrollElements2 = document.querySelectorAll('.scroll2');
const scrollElements3 = document.querySelectorAll('.scroll3');
const scrollElements4 = document.querySelectorAll('.scroll4');
const scrollElements5 = document.querySelectorAll('.scroll5');
const scrollElements6 = document.querySelectorAll('.scroll6');
const scrollElements7 = document.querySelectorAll('.scroll7');
const scrollElements8 = document.querySelectorAll('.scroll8');


scrollElements.forEach((el) => observer.observe(el));
scrollElements2.forEach((el) => observer.observe(el));
scrollElements3.forEach((el) => observer.observe(el));
scrollElements4.forEach((el) => observer.observe(el));
scrollElements5.forEach((el) => observer.observe(el));
scrollElements6.forEach((el) => observer.observe(el));
scrollElements7.forEach((el) => observer.observe(el));
scrollElements8.forEach((el) => observer.observe(el));
```

Located at:
js/scroll.js

you might find the same code on other pages.

THE ON HOVER ANIMATIONS:



```
.hover1:hover {
    /* Styles on hover */
    transform: translateY(-10px); /* Move the element 10 pixels up on hover */
    transition-duration: 0.3s; /* Change transition duration on hover to 0.5 seconds */
}

.hover2:hover {
    /* Styles on hover */
    transform: translateY(-10px); /* Move the element 10 pixels up on hover */
    transition-duration: 0.3s; /* Change transition duration on hover to 0.5 seconds */
}
/* for the job offer cards buttons */
.hover3 {
    transition: transform 0.6s ease 0s;
}
.hover3:hover {
    /* Styles on hover */
    transform: translateY(-4px); /* Move the element 10 pixels up on hover */
    transition-duration: 0.3s; /* Change transition duration on hover to 0.5 seconds */
}
```

these classes are mainly used with buttons like apply now...

located at css/hoverr.css

SASS BOOTSTRAP

we used sass for only two things, get the blue yellow colors

btn-blue
btn-yellow

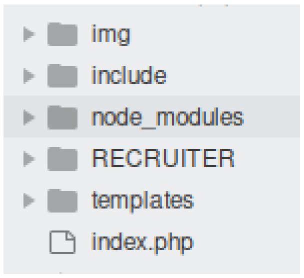
in order to use the blue and yellow class names we gotta recompile the bootstrap source code with those html color codes included.

here is a simple guide:
1. istall npm
in my case: sudo pacman -Sy nodejs

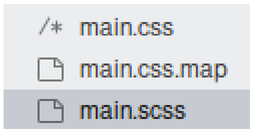
2. initialise npm
on the project directory
sudo npm init

3. now install bootstrap it's recommended to specify the version
sudo npm install bootstrap@5.3.3

a directroy called nodes_modules will show up

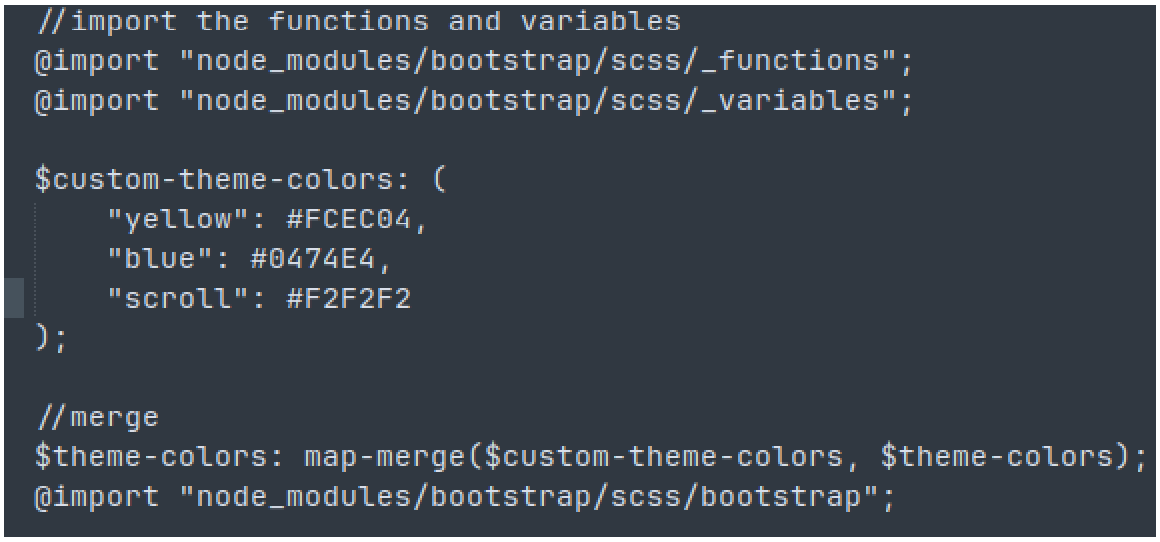


4. next up install sass:
sudo npm install -y sass



5. add the blue yellow colors into the main.scss file:

remember to import the bootstrap functions and variables before declaring the new array of colors.



```
//import the functions and variables
@import "node_modules/bootstrap/scss/_functions";
@import "node_modules/bootstrap/scss/_variables";

$custom-theme-colors: (
    "yellow": #FCEC04,
    "blue": #0474E4,
    "scroll": #F2F2F2
);

//merge
$theme-colors: map-merge($custom-theme-colors, $theme-colors);
@import "node_modules/bootstrap/scss/bootstrap";
```

After that merge the new colors array into the already existing array named them-colors

save and exit then compile the sass code into css cause browsers cant read sass.

```
sudo sass main.scss main.css
```

6. auto compilation:
open a new terminal navigate to the project directory and run this command:

```
sudo sass -watch main.scss main.css
```

so whenever you modify the main.scss file it gets compiled automatically.

7. finally include the bootstrap css and js into the project:

```
<link rel="stylesheet" type="text/css"
href="../main.css">

<script type="text/javascript"
src="../node_modules/bootstrap/dist/js/bootstrap.bu
ndle.min.js"></script>
```

and with that done we can use the blue yellow colors everywhere.

TECHNOLOGIES USED ON THIS PROJECT

1. the basics you know, js css html

2. sublime text editor:

https://www.sublimetext.com/index2

3. bootstrap 5:
https://getbootstrap.com/

4. Intersection Observer API:
https://developer.mozilla.org/en-US/docs/Web/API/Intersection_Observer_API

5. archlinux we gotta include this for sure:
https://archlinux.org/

6. qemu/kvm/virtmanager virtualization stack
https://www.qemu.org/
https://linux-kvm.org/page/Main_Page
https://virt-manager.org/

7. charts.js:
https://www.chartjs.org/

8. datatables library:
https://datatables.net/

9. github

10. node.js:
https://www.npmjs.com/