

elite

A Good Social Site.

the team:

Nauman
Samejo

248760

Noor
UlAin

256600

Huda
Asif

268420

Kainat
Tasadaq

254678

#project-report

Introduction

Billions of people across the world use social networking to communicate with friends, family and to learn new things which interest them and which is entertaining. Social Networking Sites have emerged as a platform for displaying individual profiles, sharing information, photos, experiences among Internet users as well as forming friendships and sending messages to each-other.

Our project **elite** is a user-friendly web based social site which allows users to have conversations, share information and create contents.

Motivation

Elite gives its users a chance to connect with people around the world.

The users will stay updated about the current events happening in real-time.

Our site can also help users create brand exposure for their businesses, the elderly can feel more connected to the society plus being on a social site is a lot of fun.

All of the above mentioned things compelled us to make a social site for our End Semester project.

Working

The user will **login** after creating an account, he will be welcomed with a **home/timeline** page where posts of his friends can be viewed. The posts can be liked and shared by the user. On clicking the **settings menu**, the user can access his **profile** from where he can view his own posts and also create a new post. After that, there will be **friends** from where he can see his friend list and also unfriend someone if he wants to. He can also **search** for the other users, view their profiles and add them as friends from there. If someone likes or shares the user's posts, he will receive a **notification** as well. In the **gallery** he can view his pictures and also post new pictures while in **about** he can see his own info. While in our chat system, he can send and receive messages to others.

Sessions are a simple way to store data for individual users against a unique session ID. In this project we used sessions. This can be used to persist state information between page requests. Session IDs are normally sent to the browser via ID and are used to retrieve existing session data.

Functional Requirements

A functional requirement describes what a software system should do. In our case,

- User Registration
- Adding new friends and unfriending the previous ones
- Creating, Sharing, Liking posts
- Viewing others profiles
- Adding and Sharing your Info
- Sending and Receiving message

Non-functional Requirements

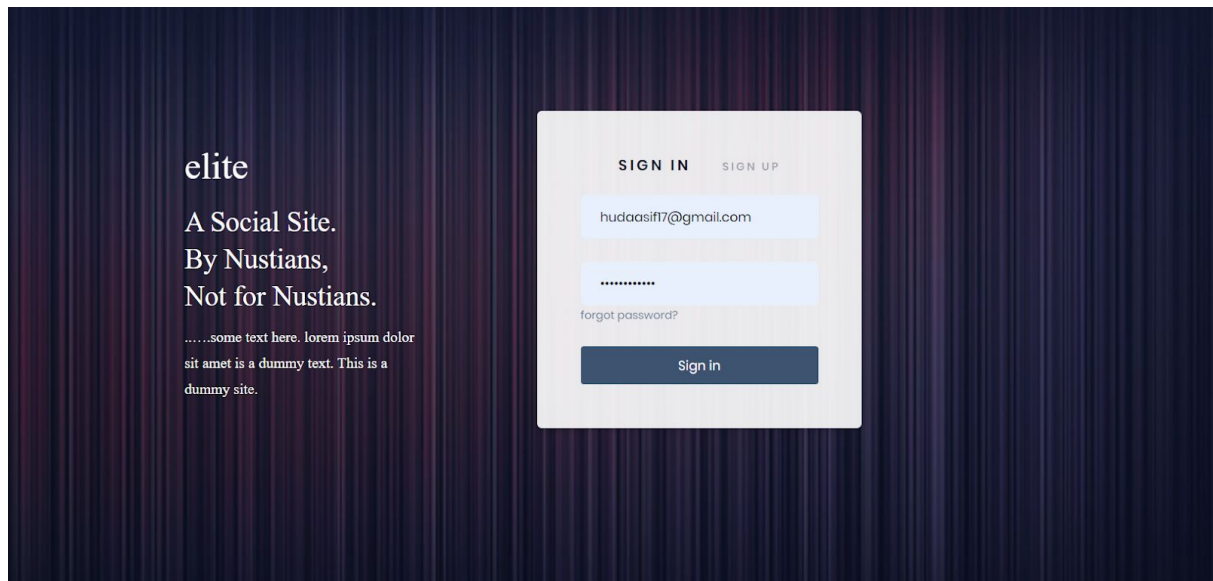
Non-functional requirements place constraints on how the system will do so:

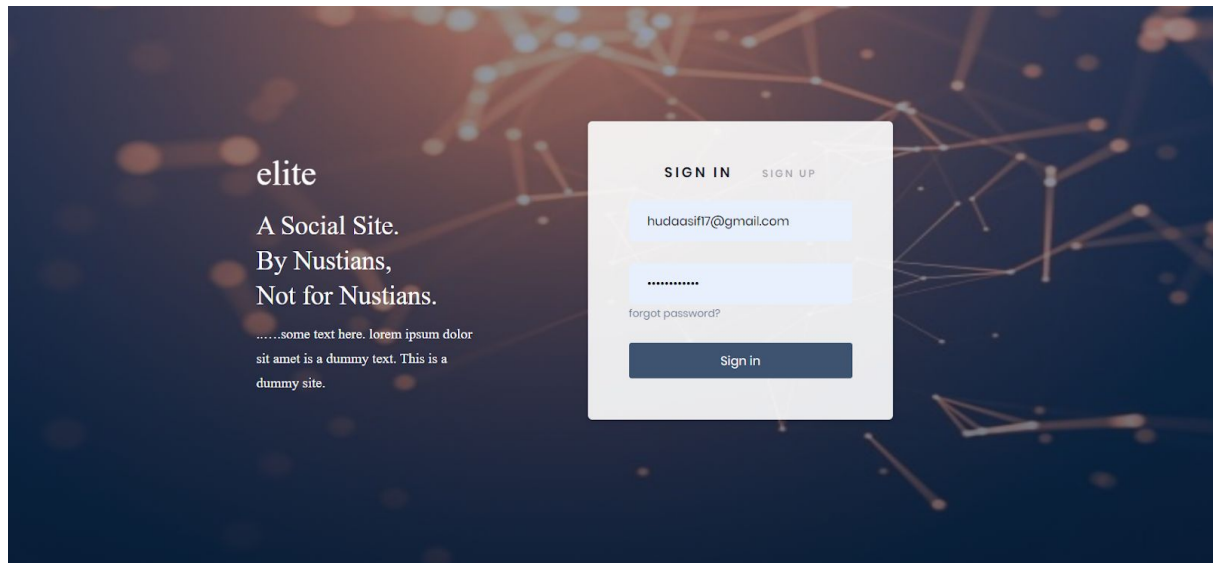
- Privacy
- Performance
- Reliability
- Usability

Description Of the Website

Login System

Since the login system is the first step towards the access of a website, hence we have added certain background images overlaid with translucent color and text which would change on page load in order to make it eye-catching for the users. The color overlay creates a better background while being much more visually interesting than a solid background color. Some of the screenshots are given:



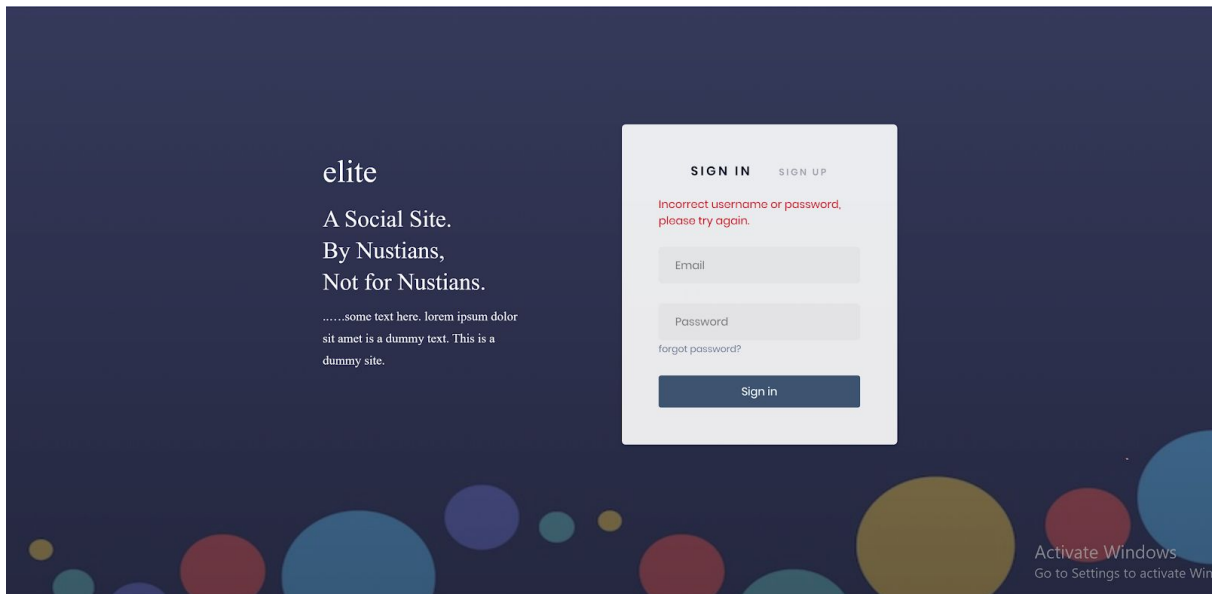


We have used a function inside a JavaScript where location of the images is stored in an array. On page load, we have generated a random number. Through which we get the image location and put that as the background of the element.

```
<script type="text/javascript">
  function changeImg(imgNumber) {
    var myImages = ["img/Background-Image1.jpg", "img/bgimg-2.jpg", "img/bgimg-3.jpeg", "img/bgimg-4.jpg",
      "img/bgimg-5.jpg", "img/bgimg-6.jpg", "img/bgimg-7.jpg", "img/bgimg-8.jpg", "img/bgimg-9.jpg"];
    var imgShown = document.body.style.backgroundImage;
    var newImgNumber = Math.floor(Math.random() * myImages.length);
    document.body.style.backgroundImage = 'url(' + myImages[newImgNumber] + ')';
  }
  window.onload = changeImg;
</script>
```

Sign In Page

The very first page is the sign in page where the site demands credentials. A user credential is a user Email and password authentication token that is bound to a particular user. If the user enters an invalid information the page will show an error message as shown below .



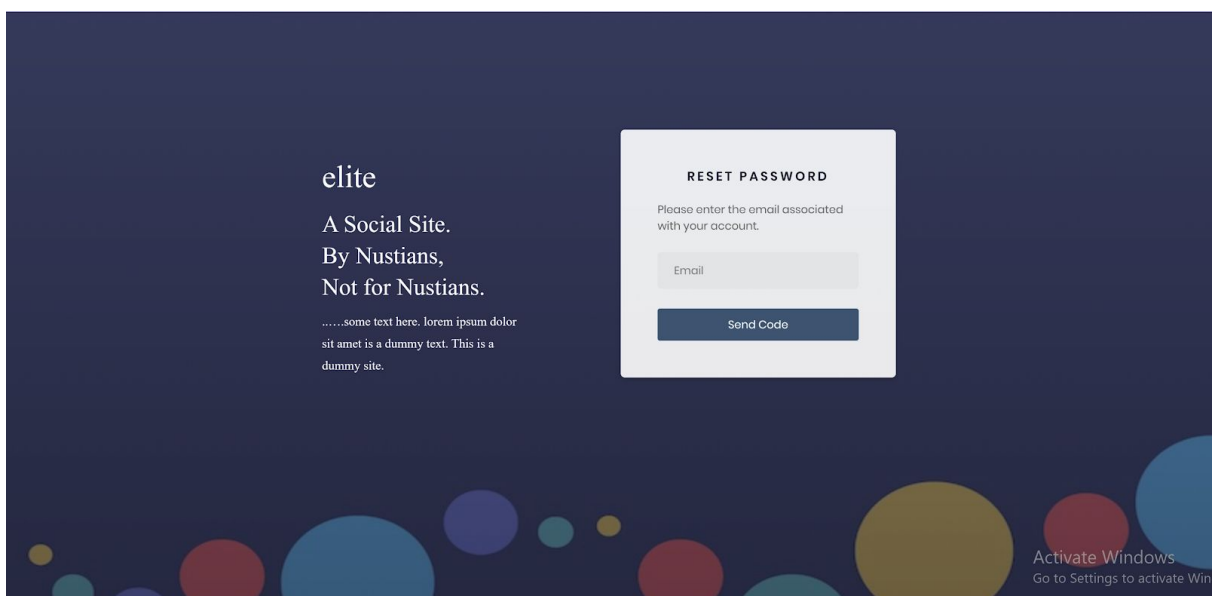
For the coding part of it, inside the php tag placed inside the sign-in form an else-if logic was generated i.e, if the user enters valid information he will move to the next page else show the error message at the top of the signin form(shown in red).

As you sign-in or sign-up into this site your login in status would get modified as “Online”

Forget Password

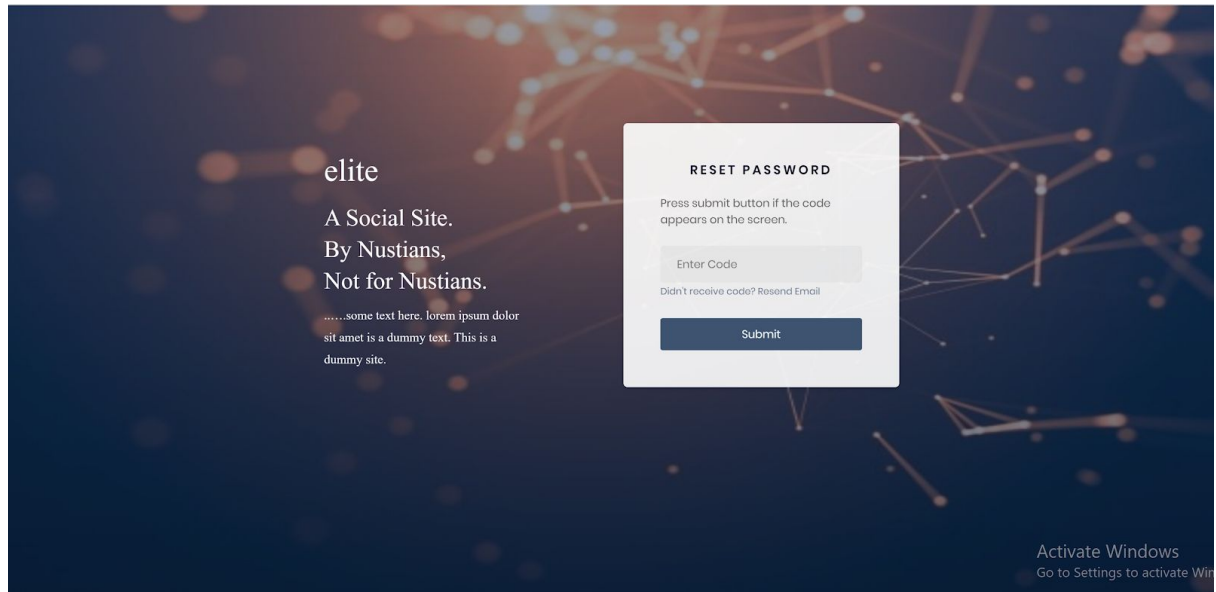
Sometime it happens that the password slips out of the mind in this case you can recover your account through the following steps,

- 1) Head to the Elite sign-in page and click the “Forgot Password” link(Shown in blue in the above screenshot).
- 2) Enter your Email associated with your account.

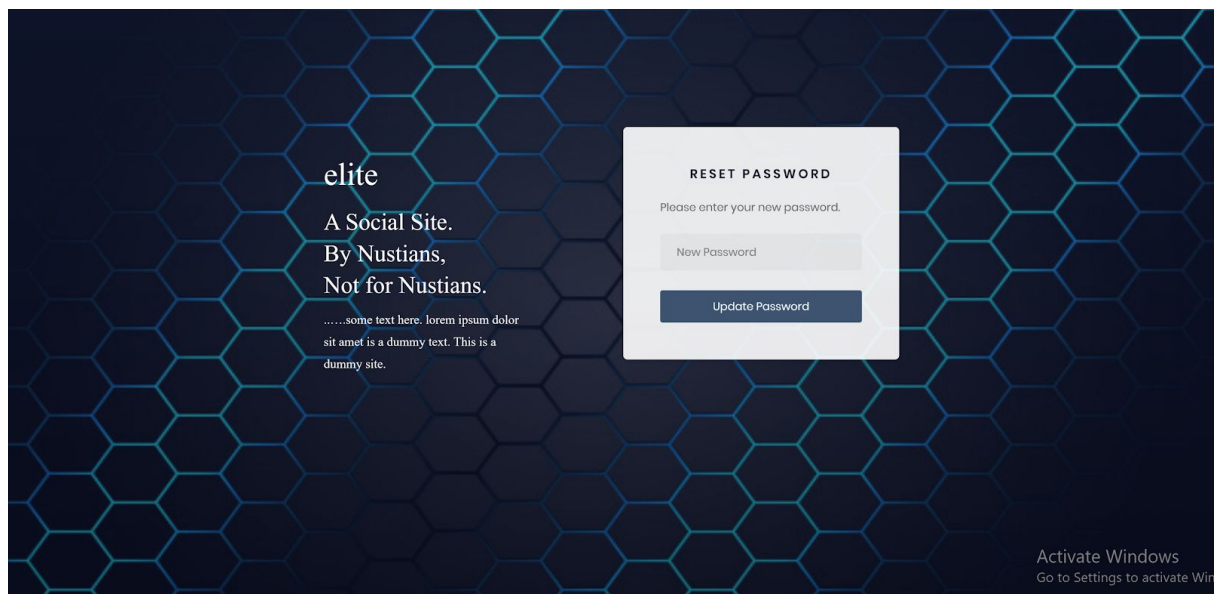


3) You will Receive a code. Enter that code .Press 'Submit'.

In a case you didn't receive the Email you can resent the Email through the link below in blue,



4) Enter the new password and press 'Update Password'

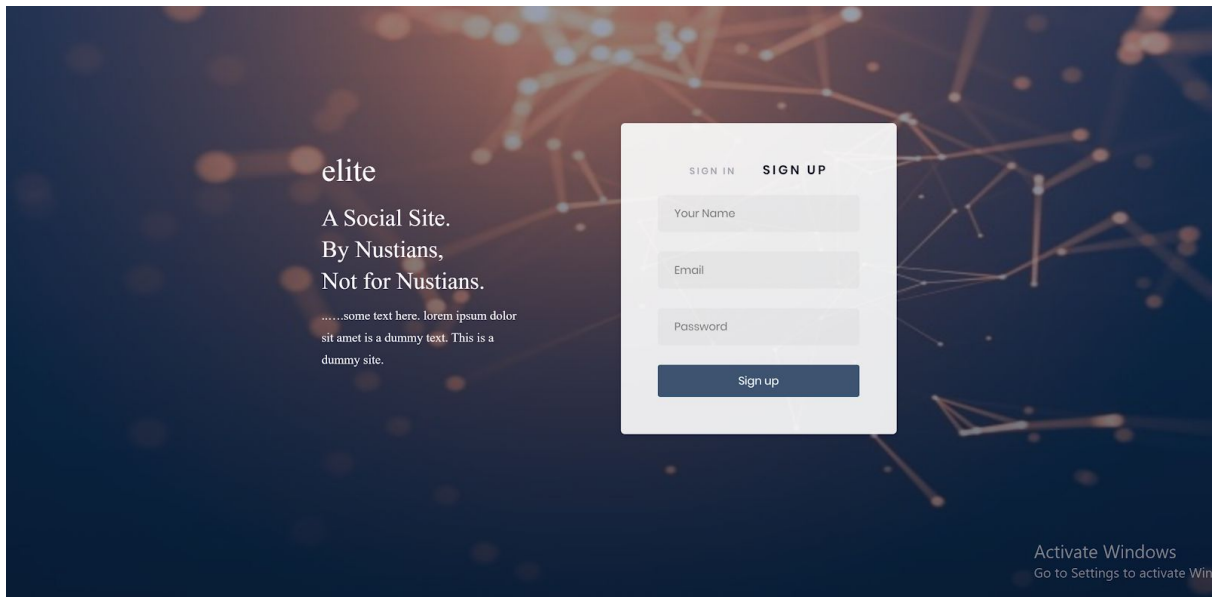


Following the mentioned step the user will be directed to the signin where to sign into his account with an updated password.

Signup page

To create your own account you just need to create your credentials as a user. Here, A user credential is a username, password and Email account authentication token that is bound to a particular user to sign up.

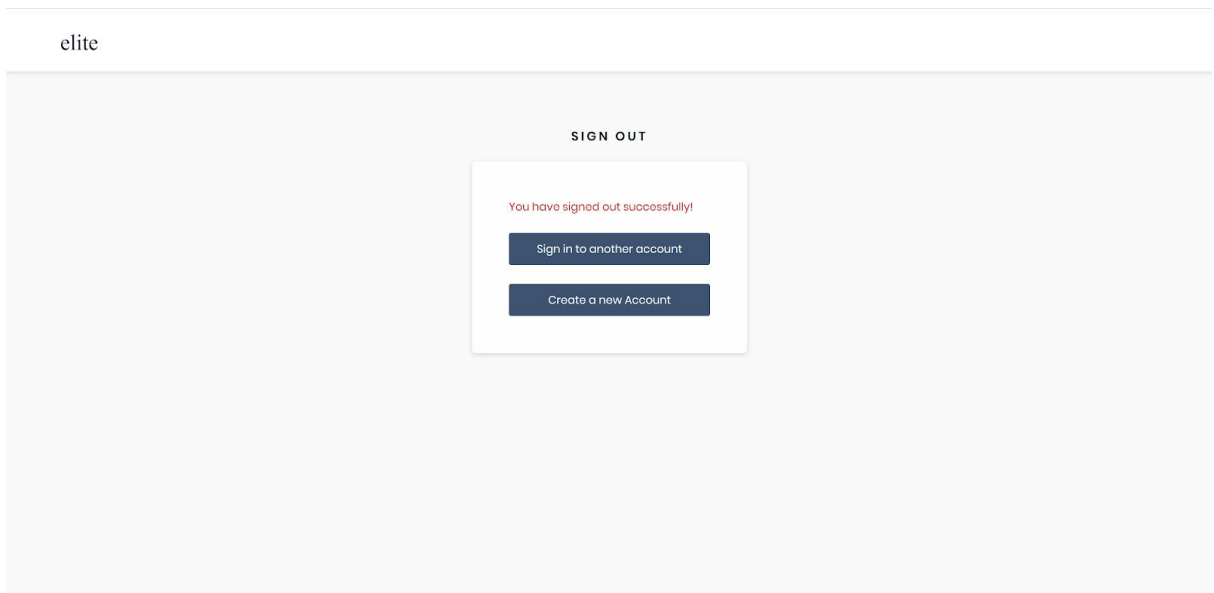
This will add users information into the login table where the credentials were saved.



Signout Page

You can move to the sign-out page through the settings-menu. Once you click the signout option the user status gets “Offline”.

From this page you can either move to sign-in page or the sign-out. The choice of the page is user dependent.



Home page

After successfully signing in/up into the account the user will move to “**Home page**”. Here the user will be able to see the posts of his friends in a descending order i.e., the most recent posts will appear at the top of the page.

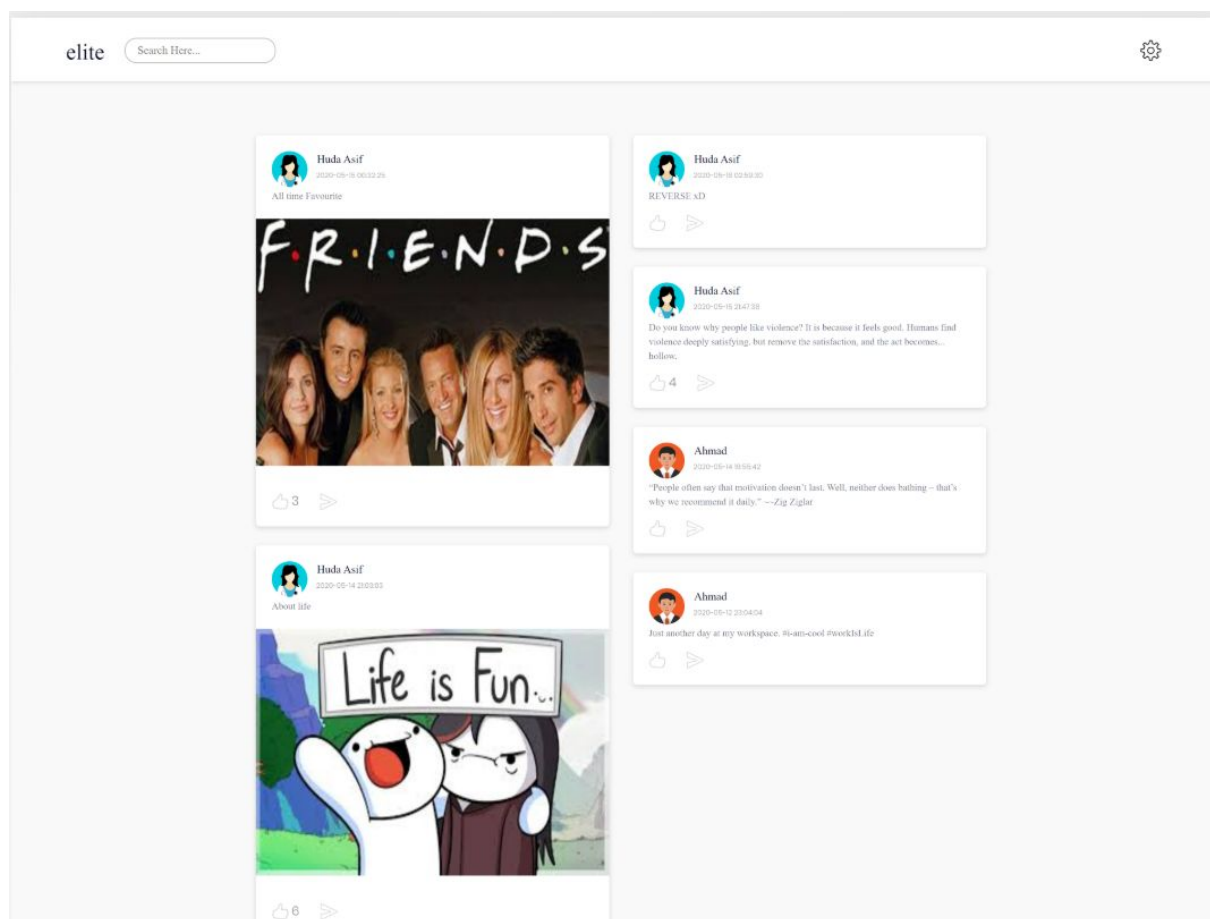
The Home has a '2 column page layout'. It was designed to differentiate between the posts type. As shown in the screenshot below the post with both status and an image will appear in the left column while the one with status only will appear in the right column.

Profile photo, posting date and time, Friend's name along with the status and the posted image will appear as a part of a single post.

The logged in user can like and share the posts visible on the home page. Like count will appear at the side of the like button. While just after sharing a post that post will appear on your personal profile page and on friends home page.

You can view your friends account through the profile pic appearing on the post.

As the new user will have no friends at first so they will not be able to see any post but a notice "Add friends to see posts" on the home page. Eventually, their friends list will grow up so they will be able to see posts on their homepage.



For the development, nested while loops were used. where at first friends data was fetched and then their posts which were displayed in descending order (Latest post at the top).


```

while($row2=mysqli_fetch_array($result2)){
    //Showing Different Friends
    $id=$row2['friends'];

    $query3 = "SELECT * FROM user_post where user_id = '$id' and post != '' order by post_id DESC";
    $result3 = mysqli_query($conn,$query3) or die( mysqli_error($conn));

    while($row3=mysqli_fetch_array($result3)){

        $likes = $row3['likes'];

        $query = "SELECT * FROM login WHERE user_id = '$id'";
        $result = mysqli_query($conn,$query) or die( mysqli_error($conn));
        $row=mysqli_fetch_array($result);

        $que = "SELECT * FROM profile WHERE user_id = '$id'";
        $res = mysqli_query($conn,$que) or die( mysqli_error($conn));
        $r=mysqli_fetch_array($res);

        $pass = $row3['post_id'];
        $passmail = $row['e_mail'];

    }
}

<div class="form-postn">

    <div class= "account">

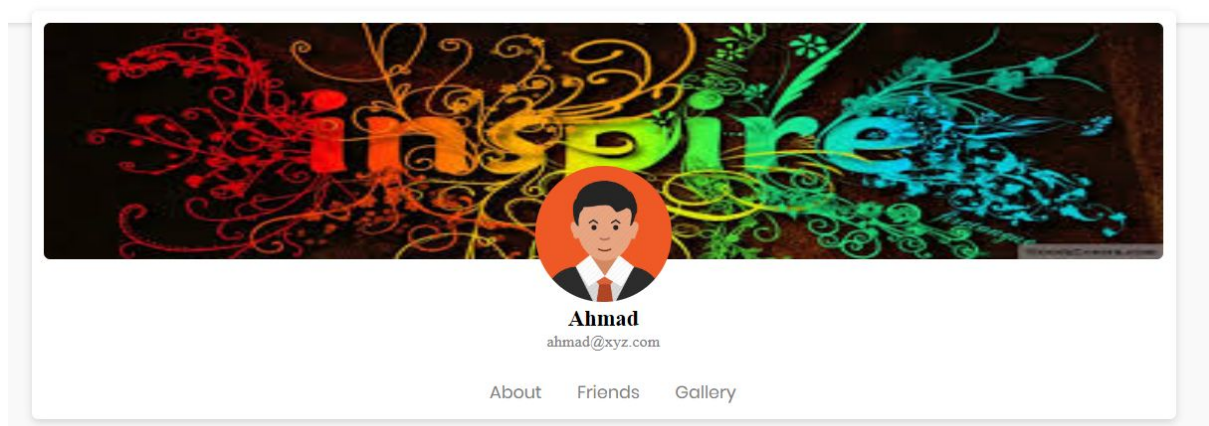
        <div class="profile">
            <a href="profileothers.php?mail=<?php echo($passmail) ?>">
                <img src= <?php echo $r['profile_url'] ?> alt="Avatar" class="avatarprofile">
            </a>
        </div>

        <div class="avatarname">
            <?php echo $row['userName'] ?>
        </div>
    </div>

```

Profile Page

The profile section consists of the cover pic and the profile pic along with the user's name and login account. The already signed-in users have their information stored in our database and will be shown the exact username and login ID they have given to our site during the signup process.



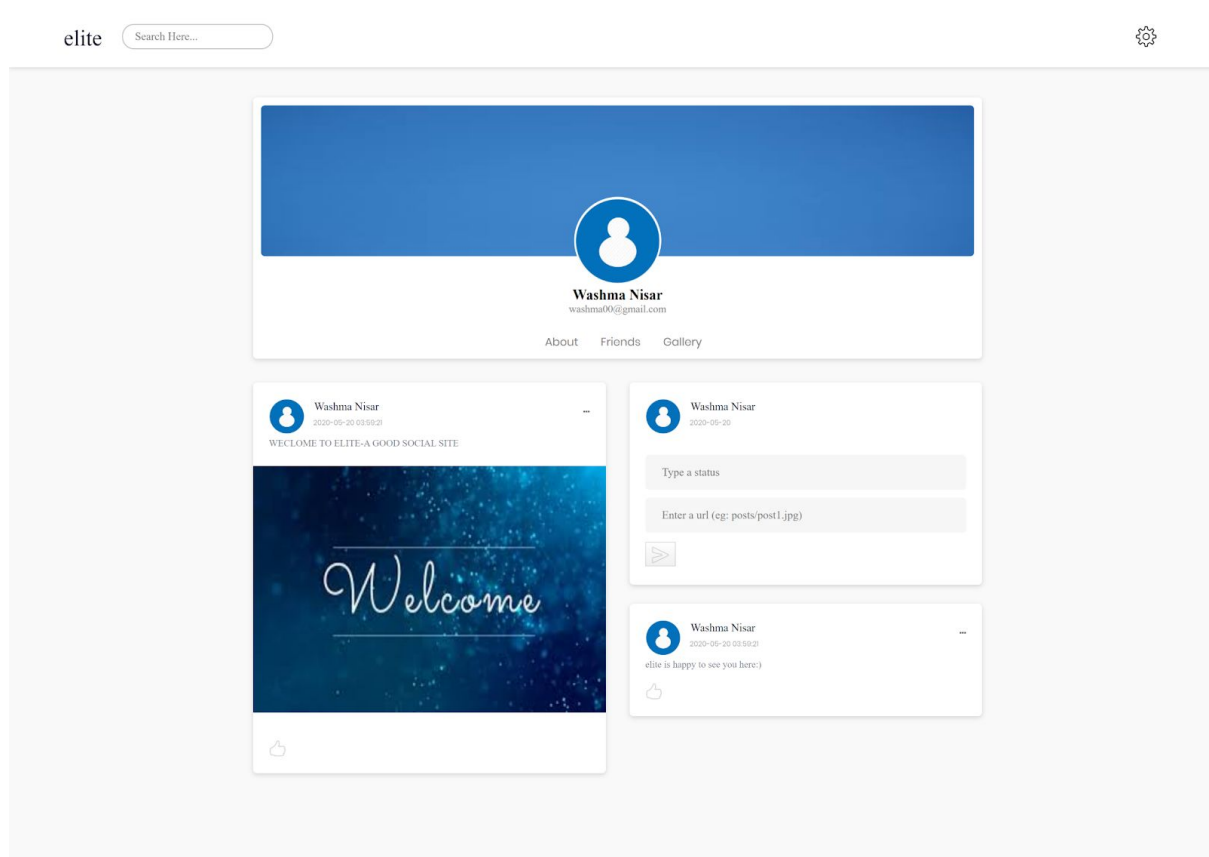
While the new users who have to sign up will see default cover and profile image already set. The three items at the bottom of the profile part (About, Friends, Gallery) will give users access to the other pages with the same profile section at the top of each one of them.

Moreover they will be overwhelmed by the site with the welcome post and status. If the user wants, he can delete those posts lately For posts, the profile page will also follow '2 column layout'. The user can see his posts and can create posts as well.

The user can delete his post with the delete option which appears as a drop down menu at the top right corner of the post. The user can like his own post through the like button.

Users Can create their own post through the form present at the top right corner of the posts section. They can post a single status or a post with both status and an image

The new post with pic will appear after following this step, "setting(at menu bar)->profile".



We have used divs for the cover and profile pic with classes assigned to them from our CSS stylesheet in order to give them the desired layout as shown above while the list items at the bottom are also placed inside a div.

For the PHP part, we first connected our code to the database and then fetched the username and mail from the login table plus the profile and cover image from the profile table using SELECT statement in database query respectively.

```
<?php

$conn = mysqli_connect('localhost','elite','','cs344_project');

$mail_p = $_SESSION["mail"];
$_SESSION["mail"] = $mail_p;
$sql1 = "SELECT userName,e_mail FROM login where e_mail='".$mail_p."' ";
$result1 = mysqli_query($conn,$sql1) or die( mysqli_error($conn));
$row1 = mysqli_fetch_array($result1);

$sql2 = "SELECT * FROM profile JOIN login using (user_id) where e_mail='".$mail_p."' ";
$result2 = mysqli_query($conn,$sql2) or die( mysqli_error($conn));
$row2 = mysqli_fetch_array($result2);
```

While the images are set default if the data fetched for them from the profile table for the respective user (who just signed-up) is empty.

```
$image1 = $row2['cover_url'];
if (empty($image1)) $image1 = "default.jpg";
$image2 = $row2['profile_url'];
if (empty($image2)) $image2 = "default2.jpg";
```

Since the list items on click will open the respective page for the user so they are provided with the link for those pages i.e.

```
$url1= "about.php" ;
$url2= "friends.php" ;
$url3= "gallery.php";
```

```
<div class="Profilelist-div">
  <br>
  <ul class="Profilelist">
    <li class="left">
      <a href="<?php echo $url1 ?>">About</a>
    </li>
    <li class="left">
      <a href="<?php echo $url2 ?>">Friends</a>
    </li>
    <li class="left">
      <a href="<?php echo $url3 ?>">Gallery</a>
    </li>
  </ul>
</div>
```

Responsiveness:


Through the use of CSS media queries, we have made our website pages responsive with design and development which responds to user's behavior and environment based on screen size, orientation and platform.


We have provided the link for bootstrap in our code and applied the classes of container and row to the divs containing the profile part. Their adjustment in the various screens is done by using col and offsets. For the profile part, code is:

```
<div class="container">
  <div class="row">
    <div class="col-lg-10 offset-lg-1 col-md-10 offset-md-1">
```

About Page

The About page consists of information about the user. The information has 3 parts; Personal, contacts and residence information. On the right of the page, you can see the options of Edit Info and Add Info for the signed-in and signed-up users respectively.

elite 



Noor
nain.bscl8seecs@seecs.edu.pk

About Friends Gallery

About

Personal Information

Name:	Noor
Age:	23
Gender	Female
Works as:	Computer Scientist
Works At:	Software house, Islamabad
Date of Birth:	30Dec,1997
Status:	Single

Contact Information

Number:	+923332600234
Email:	nain.bscl8seecs@seecs.edu.pk

Residence Information

Country:	Pakistan
City:	Islamabad
Address:	H-12

For the user to edit the info, they will be guided to another page containing the form. On making changes to the required fields, they will submit the form with being redirected to the about page with their edited info. Or they opened the form but don't want any changes, they can press the link About at the top of the form to go back to the about page.

About

Personal Information
Age:
Gender:
Works as:
Works at:
Date of Birth:
Status:


Contact Information
Number:

Residence Information
Country:
City:
Address:

Similarly, we have a form for adding information for the users new to our site which will be filled by them. Some fields are mandatory to be filled (e.g. Country, Gender etc.) but some can be left where the users don't want to give their certain personal info.

About

Personal Information
user-id:
Age:
Gender:

 Please fill out this field.

Works as:
Works at:
Date of Birth:
Status:

Contact Information
Number:

Residence Information
Country:
City:
Address:

Coding:

All the information regarding the user will be present in the about table of the database which will be fetched by the simple SELECT statement using the email with which the user logged in as the condition with JOIN in between the two tables (about and login).

```
$sql3 = "SELECT * FROM about JOIN login using (user_id) where e_mail='".$mail_p."' ";
$result3 = mysqli_query($conn,$sql3) or die( mysqli_error($conn));
$row3 = mysqli_fetch_array($result3);
```

For edit form, the data from the text fields will be taken through POST method and that data will then be used to update the about table in the database through UPDATE query as shown:

```
if(isset($_POST['edit'])){
    $age = mysqli_real_escape_string($conn, $_POST['age']);
    $gender = mysqli_real_escape_string($conn, $_POST['gender']);
    $work_as = mysqli_real_escape_string($conn, $_POST['work_as']);
    $works_at = mysqli_real_escape_string($conn, $_POST['works_at']);
    $dob = mysqli_real_escape_string($conn, $_POST['dob']);
    $status = mysqli_real_escape_string($conn, $_POST['status']);
    $number = mysqli_real_escape_string($conn, $_POST['number']);
    $country = mysqli_real_escape_string($conn, $_POST['country']);
    $city = mysqli_real_escape_string($conn, $_POST['city']);
    $address = mysqli_real_escape_string($conn, $_POST['address']);
    //updating the table
    $sql = "UPDATE about join login on login.user_id = about.user_id SET about.age='$age',about.gender='$gender',
    about.work_as= '$work_as', about.works_at='$works_at', about.dob='$dob', about.status='$status',
    about.number= '$number', about.country= '$country', about.city='$city', about.address= '$address' where login.e_mail= '$mail_p' ";
    $result = mysqli_query($conn,$sql);
}
```

While for displaying the user's current data in the form as it is opened, we will be selecting data associated with the particular user mail i.e.

```
<?php
$mail_p = $_SESSION["mail"];
$sql1 = "SELECT * FROM about JOIN login using (user_id) where e_mail='".$mail_p."' ";
//selecting data associated with this particular id
$result1 = mysqli_query($conn,$sql1);
$res=mysqli_fetch_array($result1);
$age = $res['age'];
$gender = $res['gender'];
$work_as = $res['work_as'];
$works_at = $res['works_at'];
$dob = $res['dob'];
$status = $res['status'];
$number = $res['number'];
$country = $res['country'];
$city = $res['city'];
$address = $res['address'];

$conn->close();
?>
```

In the same way, for adding info, the data written into the form by the user will be fetched to the database through INSERT query and on submitting, that data will be displayed in the new user's about page i.e.

```
if(isset($_POST['add'])) {
    $age = mysqli_real_escape_string($conn, $_POST['age']);
    $gender = mysqli_real_escape_string($conn, $_POST['gender']);
    $work_as = mysqli_real_escape_string($conn, $_POST['work_as']);
    $works_at = mysqli_real_escape_string($conn, $_POST['works_at']);
    $dob = mysqli_real_escape_string($conn, $_POST['dob']);
    $status = mysqli_real_escape_string($conn, $_POST['status']);
    $number = mysqli_real_escape_string($conn, $_POST['number']);
    $country = mysqli_real_escape_string($conn, $_POST['country']);
    $city = mysqli_real_escape_string($conn, $_POST['city']);
    $address = mysqli_real_escape_string($conn, $_POST['address']);
    $user_id = mysqli_real_escape_string($conn, $_POST['id']);
    //insert data to database
    $sql = "INSERT INTO about(user_id,age,gender,work_as,works_at,dob,status,number,country,city,address)
    VALUES ('$user_id','$age','$gender','$work_as','$works_at','$dob','$status','$number','$country','$city','$address')";
    $result = mysqli_query($conn,$sql) or die( mysqli_error($conn));
}
```

Friends Page

In the friends page, you can view your friend list. You can also unfriend. To delete friends you just need to place the cursor over it. using the hover property we made the button to change its color and hover content so that the user can easily identify the functionality of the button i.e, to remove the person from friends list.

```
<button class="follow-frnd f-right" data-hover="unfriend" onclick="location.href='phplinks/deleteFriend.php?i=?php echo $i ?>&you=?php echo $you ?>'>
    Friend
</button>
```

When we press the unfriend button both of the users remove themselves from their friend list.

Keeping in view the security concerns, when the users views friend list of some other person he would not be able to unfriend someone from the friend list.

Noor Ul Ain
noorain908@gmail.com

About Friends Gallery

Friends

<p>Huda Asif hudaasif7@gmail.com</p>	unfriend
<p>Ahmad ahmad@xyz.com</p>	Friend

Activate Windows
Go to Settings to activate W

<ul class="row12">

<?php

```
$conn = mysqli_connect('localhost','elite','','cs344_project');
```

```
$mail = $_SESSION['mail'];
```

```
$_SESSION['email'] = $mail;
```

```
// fetching user_id
```

```
$query1 = "SELECT * FROM login WHERE e_mail = '$mail' ";
```

```
$result1 = mysqli_query($conn,$query1) or die( mysqli_error($conn));
```

```
$row1=mysqli_fetch_array($result1);
```

```
$value=$row1['user_id'];
```

```
//fetching Friends
```

```
$query2 = "SELECT * FROM friends WHERE user_id = '$value'";
```

```
$result2 = mysqli_query($conn,$query2) or die( mysqli_error($conn));
```

```
while($row2=mysqli_fetch_array($result2)){
```

```
//Showing Different Friends
```

```
$id=$row2['friends'];
```

```
$query3 = "SELECT * FROM login where user_id = '$id'";
```

```
$result3 = mysqli_query($conn,$query3) or die( mysqli_error($conn));
```

```
$row3=mysqli_fetch_array($result3);
```

```
$req = $row3['user_id'];
```

```
$fmail = $row3['e_mail'];
```

```
// Fetching my id
```

```
$me = "SELECT user_id from login where e_mail= '$mail'";
```

```
$meo = mysqli_query($conn,$me) or die( mysqli_error($conn));
```

```
$medo=mysqli_fetch_array($meo);
```

```
$i = $medo['user_id'];
```

```
$me1 = "SELECT * from profile where user_id= '$req'";
```

```
$meo1 = mysqli_query($conn,$me1) or die( mysqli_error($conn));
```

```

// Fetching friends
$unfriend = "SELECT user_id from login where e_mail= '$fmail'";
$unfriendgo = mysqli_query($conn,$unfriend) or die( mysqli_error($conn));
$unfrienddo=mysqli_fetch_array($unfriendgo);
$you = $unfrienddo['user_id'];
?>
<li class="clear">
    <div class="profile ">
        <a href='profileothers.php?mail=?php echo($fmail) ?>'>
            
        </a>
    </div>

    <div class="avatarname ">
        <a>
            <?php echo $row3['userName']; ?>
        </a>
    </div>

    <br>

    <div class="searchstatus f-left">
        <a>
            <?php echo $row3['e_mail']; ?>
        </a>
    </div>

    <button class="follow-frnd f-right" data-hover="unfriend" onclick="location.href='phplinks/deleteFr:
        Friend
    </button>

</li>

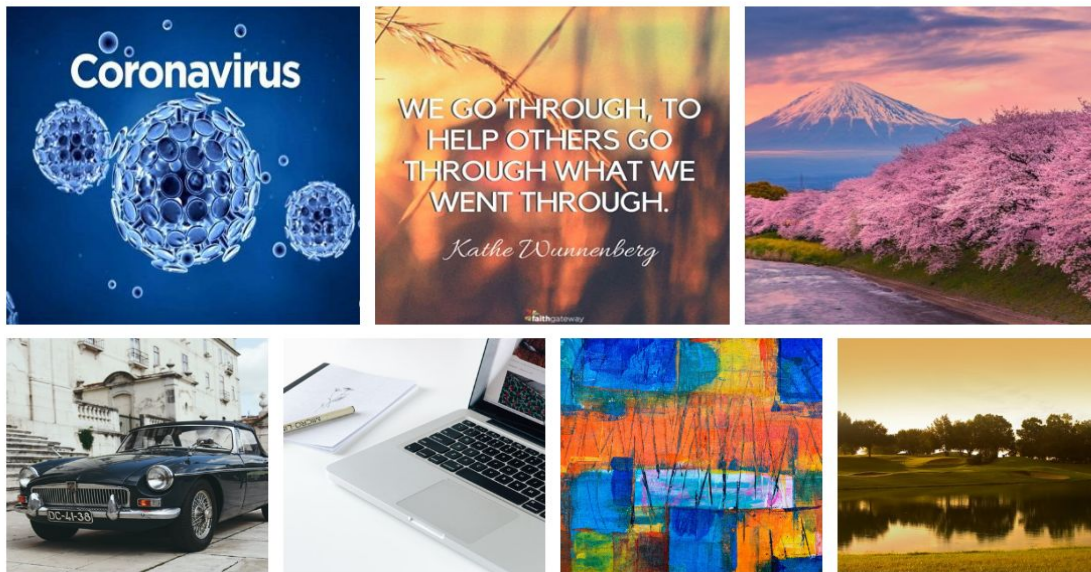
<!--ennd of while loop-->
<?php
}
}

```

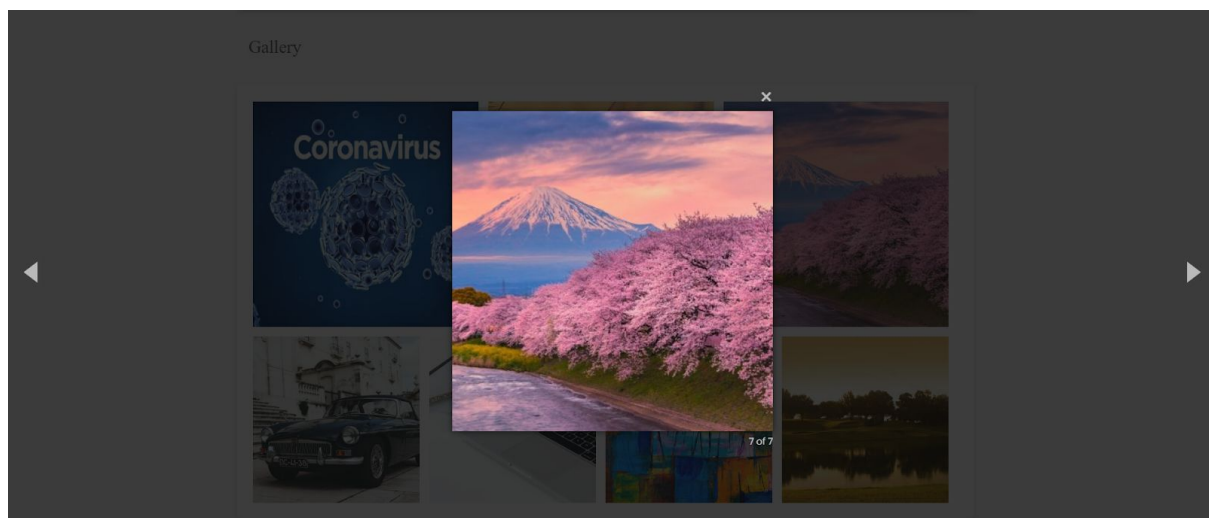
Gallery Page

The Gallery section of the website will help users upload, manage and show the images to their friends. The images collection is the vital part of a social site and for that purpose, we have made a responsive and appealing Gallery page which will also serve as a collection of memories for the users.

Gallery



Upon clicking, the images will be displayed which the user then can swipe forward and backward as shown:



For uploading more pictures, they will provide us with the location of the image they tend to upload and hence the image will then become a part of their gallery.



Coding:

Since our page layouts are div based, hence we have used div where we provided the image source inside the anchor tag with the href to the same image location. Display flex property is applied to the div where images are placed which makes our layout structure flexible and responsive without using float or positioning.

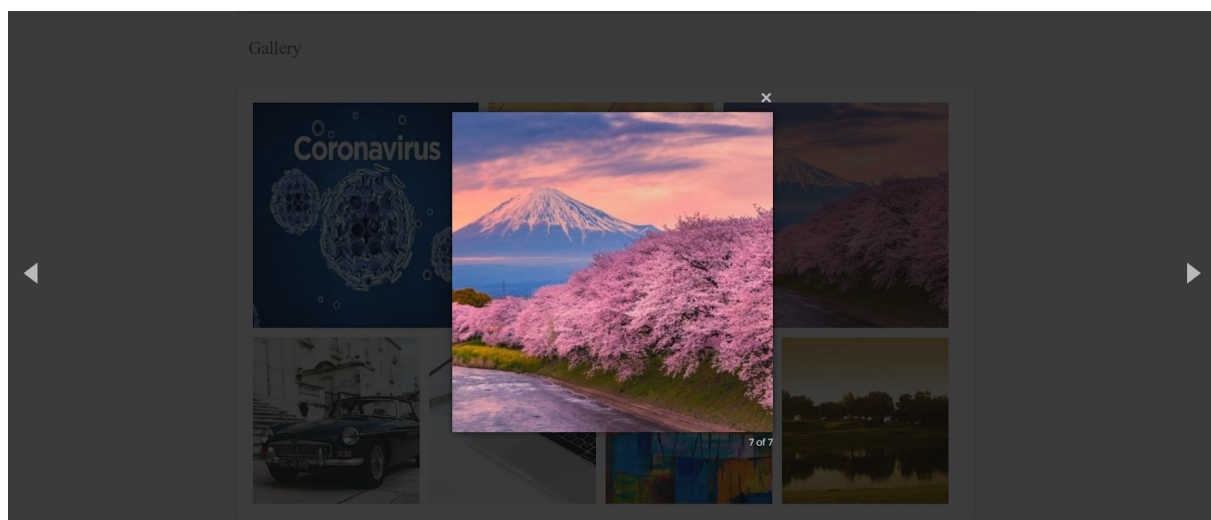
```
.gallery{  
  display: flex;  
  flex-wrap: wrap-reverse;  
  justify-content: centre;  
}
```

In addition, when hovered, a transform property is applied on images which gives them a 2D or 3D transformation. In our case we have used transform scale.

```
.gallery-section .image:hover img{  
  transform: scale(1.1);  
}
```




We have used Magnific Popup which is an open source jQuery lightbox to independently display images one by one.



For this purpose, we have provided the links to the Magnific Popup jQuery libraries in our code for the gallery section.

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/magnific-popup.js/1.1.0/jquery.magnific-popup.min.js"></script>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/magnific-popup.js/1.1.0/magnific-popup.min.css">
```

And have included the following code in the script:

```

$(".gallery").magnificPopup({
  delegate: 'a',
  type: 'image',
  gallery:{
    enabled: true
  }
})

```

Now for the PHP part, the location of pics is stored in the gallery table of the database which will be fetched through a SELECT statement with user id from which the user logged in and the images will be displayed by giving those urls in the tag.

```

$path2 = "SELECT * FROM gallery WHERE user_id='$required_id' " ;
$resultp2 = mysqli_query($conn,$path2) or die( mysqli_error($conn));
while($rowp2 = mysqli_fetch_array($resultp2)){
    $gimg = $rowp2['gallery_url'];
?>
<a href="<?php echo "img/".$gimg ?>" class="image">
    " alt="">
</a>
<?php
}
?>

```

And for adding images, we will use the INSERT statement and POST method.

```

if(isset($_POST['image'])){
    $required_id= $_rowp['user_id'];
    $value1 = $_POST["image"];
    $query = "INSERT INTO gallery (user_id, gallery_url) value ('$required_id','$value1')";
    mysqli_query($conn, $query);
}

```

Responsiveness:

As discussed above, we have used lightbox for providing best user experience with any device in addition to the flex property we have used in media queries in our CSS stylesheet for gallery.

```

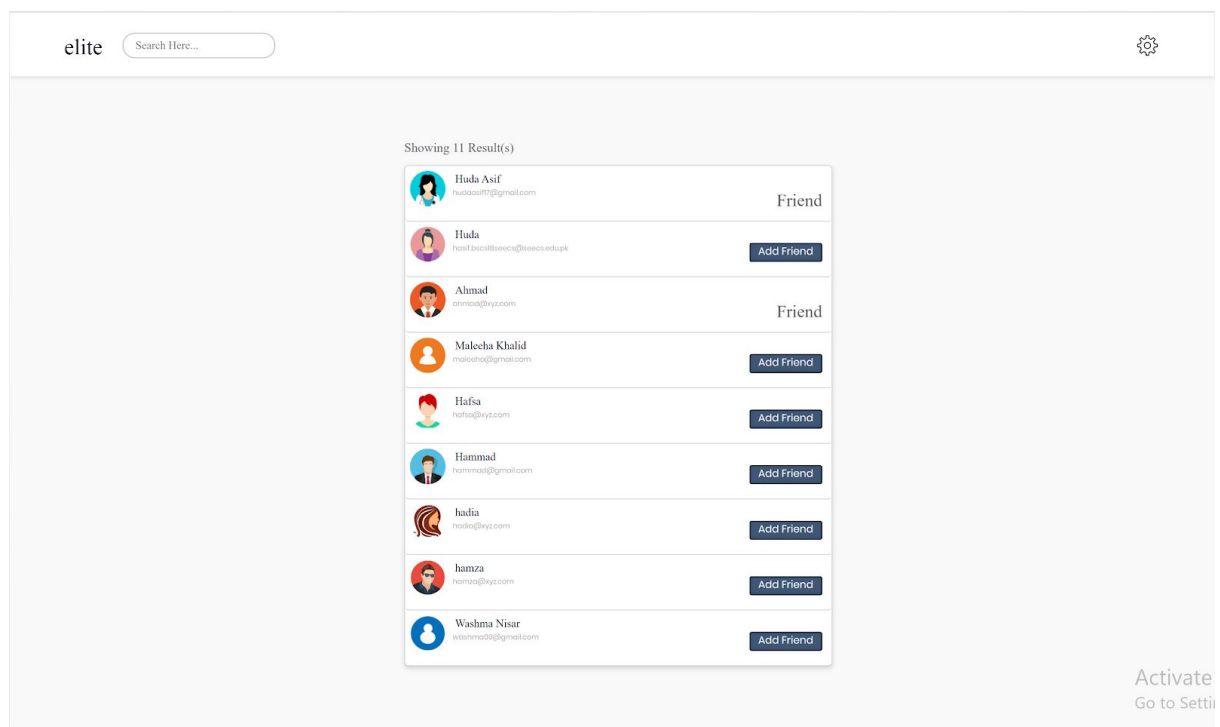
@media screen and (max-width:960px){
  .gallery-section .image{
    flex: 33.33%
  }
}
@media screen and (max-width:768px){
  .gallery-section .image{
    flex: 50%
  }
}
@media screen and (max-width:480px){
  .gallery-section .image{
    flex: 100%;
    cursor: pointer;
  }
}

```

SEARCH-INDEX PAGE

After searching for someone's id, the user can view the result on the search index page. This page was also developed using the while loop logic. each and every person corresponding to the result would be displayed at each iteration of the loop one by one.

From here you can add others as ur friend and can also get a view if someone is already your friend or not



Coding:

Firstly, we have introduced the search bar in the home, profile and other pages for the user to search for other users using our site.

elite

Search Here...

For this, we got the username from the search bar through POST and then using SELECT query matched the searched user to the usernames in our database and then proceeded to the search page which will display all those users that the query could found from the database.

```
if(isset($_POST['search'])){
    $searchq = $_POST["search"];
    $url= "search-index.php?search=" . $searchq;
    $searchq = preg_replace("#[^0-9a-z]#i","", $searchq);
    $query = "SELECT * FROM login where userName LIKE '%$searchq%' ";
    $result = mysqli_query($conn,$query) or die( mysqli_error($conn));
    $count = mysqli_num_rows($result);
    if($count==0){
        header("Location: ". $url);
    }
    else{
        while($row=mysqli_fetch_array($result)){
            header("Location: ". $url);
        }
    }
}
```

For getting required info from the database, we have again used queries and also used if else statements for showing whether the database had the searched user's info or not. For instance,

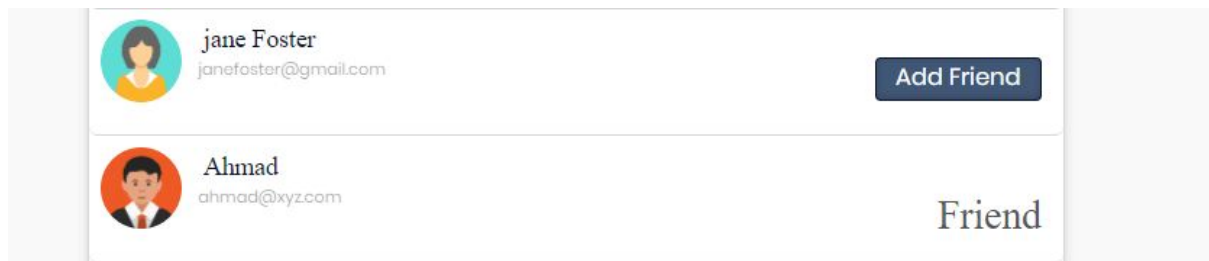
Showing 0 Result(s)

There were no search results!

```
$search = $_GET["search"];
$query1 = "SELECT * FROM profile join login using (user_id) where userName LIKE '%$search%' ";
$result1 = mysqli_query($conn,$query1) or die( mysqli_error($conn));
$count1 = mysqli_num_rows($result1);
if($count1==0){
    echo "<b>". "There were no search results!". "</b>";
}
else{
    while($row=mysqli_fetch_array($result1)){
        $user_id = $row['user_id'];
```

The search page will also show whether the searched users are your friends or if not then whether you want to add them in your friend list. We also achieved it through select query and if else statement in a way that if the searched user's id is found against the logged in user id in the

friends table then that particular user will be shown as Friend of the logged in user. For instance,



Now for the add friend part, the logged in user's id plus the searched user's id will be fetched and will then be inserted to the friends table through INSERT query such that after adding the searched user as friends, the logged in user will also become a friend of that searched user.

```
$id1 = $_GET['myid'];
$mail_a = $_GET['mail'];
$path = "SELECT user_id from login where e_mail= '$mail_a' ";
$go = mysqli_query($conn,$path) or die( mysqli_error($conn));
$went = mysqli_fetch_array($go);
$id2 = $went['user_id'];

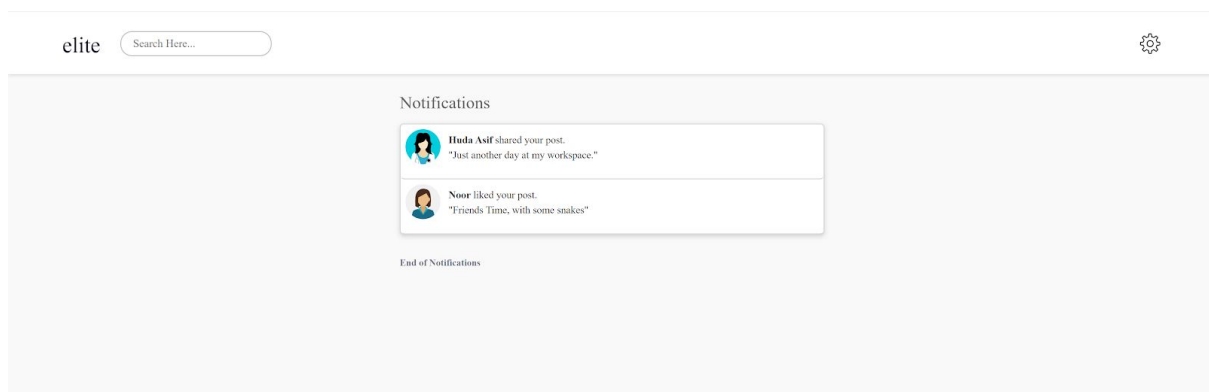
$add = "INSERT INTO friends(user_id,friends) VALUES ('$id1', '$id2') " ;
$addgo = mysqli_query($conn,$add) or die( mysqli_error($conn));

$add2 = "INSERT INTO friends(user_id,friends) VALUES ('$id2', '$id1') " ;
$addgo2 = mysqli_query($conn,$add2) or die( mysqli_error($conn));
```

After adding a friend, the user will be redirected to its friends page where the recently added friend will be shown to him.

Notification:

The site Notifies the user of the information that his post has been shared or liked along with the name of the person.



```

$posr = $val['post_id'];
$share = $val['sharedby'];
$like = $val['likedby'];

if(!empty($share)){
    $query2 = "SELECT * FROM login WHERE user_id = '$share' ";
    $result2 = mysqli_query($conn,$query2) or die( mysqli_error($conn));
    $row=mysqli_fetch_array($result2);

    $query3 = "SELECT * FROM profile WHERE user_id = '$share' ";
    $result3 = mysqli_query($conn,$query3) or die( mysqli_error($conn));
    $row3=mysqli_fetch_array($result3);

    $q = "SELECT * FROM user_post WHERE post_id = '$posr' ";
    $r = mysqli_query($conn,$q) or die( mysqli_error($conn));
    $ro=mysqli_fetch_array($r);

    $output="shared";
}
else{
    $query2 = "SELECT * FROM login WHERE user_id = '$like' ";
    $result2 = mysqli_query($conn,$query2) or die( mysqli_error($conn));
    $row=mysqli_fetch_array($result2);

    $query3 = "SELECT * FROM profile WHERE user_id = '$like' ";
    $result3 = mysqli_query($conn,$query3) or die( mysqli_error($conn));
    $row3=mysqli_fetch_array($result3);

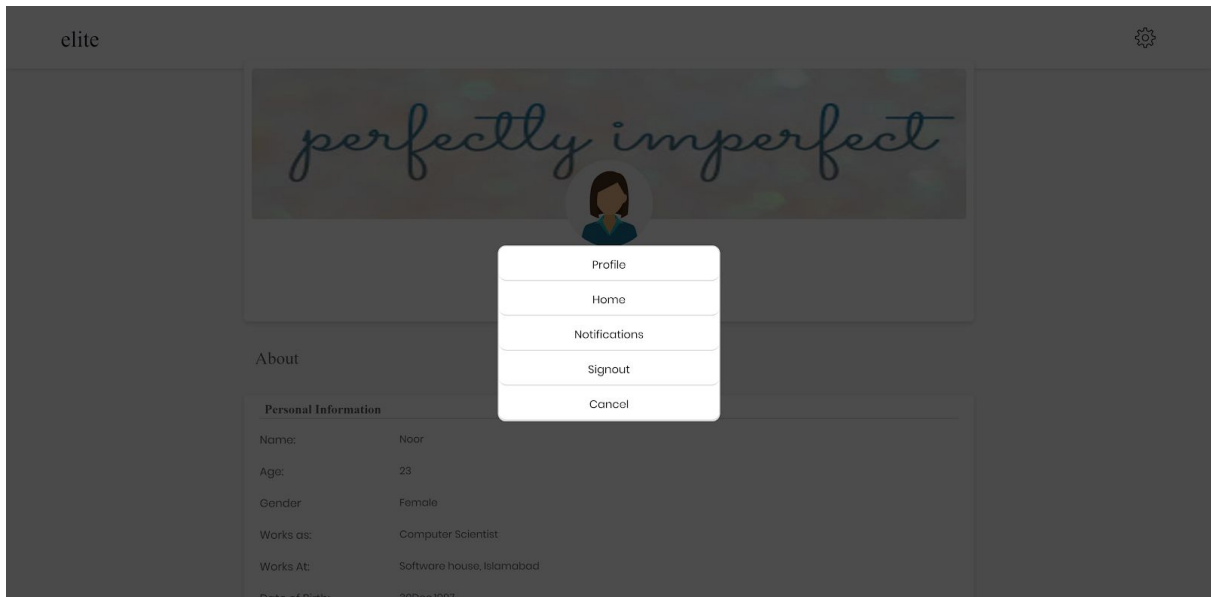
    $q = "SELECT * FROM user_post WHERE post_id = '$posr' ";
    $r = mysqli_query($conn,$q) or die( mysqli_error($conn));
    $ro=mysqli_fetch_array($r);

    $output="liked";
}
}
}

```

Settings-Menu:

Settings-Menus present at the right corner of the nav-bar through this you can move to different pages. In the development of this part z-index and the visibility feature are the most important. the visible was changed from hidden to visible by the use of toggle class. You can close the menu-index either by clicking the cancel option or by clicking any whereon the black area(having less opacity)



```

</div>
<div onclick="close1();" class="bg-modal1" id="myDiv">
  <div class="menu">
    <ul>
      <li class="clear"><a href='profile.php'>Profile</a></li>
      <li class="clear"><a href='home.php'>Home</a></li>
      <li class="clear"><a href='notifications.php'>Notifications</a></li>
      <li class="clear"><a href="signout.php">Signout</a></li>
      <li onclick="close1();" class="clear" ><a href="" id="close">Cancel</a></li>
    </ul>
  </div>
</div>
</section>
</div>

<script type="text/javascript">

  function appear(){
    document.getElementById("myDiv").classList.toggle("bg-modal2");
  }

  function close1(){
    document.getElementById("myDiv").classList.toggle("bg-modal2");
  }

```

Chat System



A preview of the chat app, adaptive layout (meaning it follows nicely on every size, and works well on phones).

Video preview can be found here: <https://youtu.be/hjovCI88FLA>

Overview

This chat-app is a module of our project, the social site. It can be deployed to a different url, etc. It fetches the current signed in user on the social site, and fetches it's details from the database.

In the given preview, "Nauman Samejo" is the logged in user. And the list on side is the friend list of "Nauman Samejo".

This is an instant-messaging platform, and to make it so, we had to use different technologies than PHP. Those are covered in the "Technologies" section.

Technologies

To make the chat system an instant messaging system, we had a few options, one of which was to use php, as the whole project does, but the issue here is that the php does not work nicely with Web Sockets, so, instead, in php we had to do work with AJAX-long-polling, which is significantly slower in performance.

Instead, we had to use the following technologies:

- React - Provides a nicer way to build component based web apps.
- Redux - Useful state management for the project.
- NodeJS - For setting up web socket and api server
- ExpressJS - Routing, using REST api
- Web Sockets - Instant messaging
- MySQL - The database we have used in the whole project, chat system authenticated and read/sent messages there.
- Redux Saga - Helping manage state asynchronously

Challenges

There were a couple of challenges faced when developing the chat module of the app. Some of them are:

Instant Messaging:

This was the main problem that we had to solve, because if done inefficiently, it can deliver late-messages, can be heavy on CPU, etc.

As discussed above, the choices we had were to choose either AJAX long polling (which I find to be very inefficient and CPU-demanding), and the other was to choose web sockets.

The solution to this was to use web sockets, and a whole different tech stack to support the web sockets, and develop the app nicely.

The solution to this one led to another problem, which is, the problem of “Authorising the user”

Authorizing the user

The solution of the previous challenge led to this problem.

As we are now using two different tech stacks, it comes at a price, for example, the api, which does the backend stuff and authorization cannot be the same on the XAMPP server's host and port. This led to the problem of not being able to access data such as Cookies, Session from the main app, which helps us to validate the user login.

The solution to this we chose was to change login + logout logic in our main app, so, instead of just setting session data when logging in/out the user, we first created a REST api, which received some parameters to set the session on it's own port, and in log in/out functions, we called that REST api with some user authentication details. Problem solved.

There were some other tiny few problems, all solved, the chat app is working well now.

Conclusion

To conclude, we did our best in developing such a site in the time span given to us while hoping that our product will stand out and be liked by the users.