BANGLADESH ARMY UNIVERSITY OF SCIENCE AND TECHNOLOGY (BAUST), SAIDPUR

Project report-02

Department: Computer Science and Engineering

Course Title: Web engineering project

Course no: CSE 3200

Experiment No: 02

Name of the Experiment: Final Project Presentation

Submitted To:

Name: Md. Al- Hasan

Designation: Lecturer

Contents

Memories Chronicle

(An event planner company website)

▪ Objectives:

- ➢ Enable Stability by monitoring all events that occurs throughout the IT infrastructure to allow for normal service operations and detect and escalate exceptions.
- ➢ To make an event planner hiring process easier for all.
- ➢ To provide basis of service assurance, reporting and service improvement.
- ➢ To help event planner to speed up process of event planning and grow the business through the use of modern technology.
- ➢ To help clients making there events stress free.

▪ Team Member:

- ➢ Abdullah Ibna Mojib (180101001)
- ➢ Soumita Mondol (180101002)
- ➢ Mahmudul Hasan Pranto (180101003)
- ➢ Moumita Roy (180101004)

▪ Entity-relationship diagram (ER diagram):

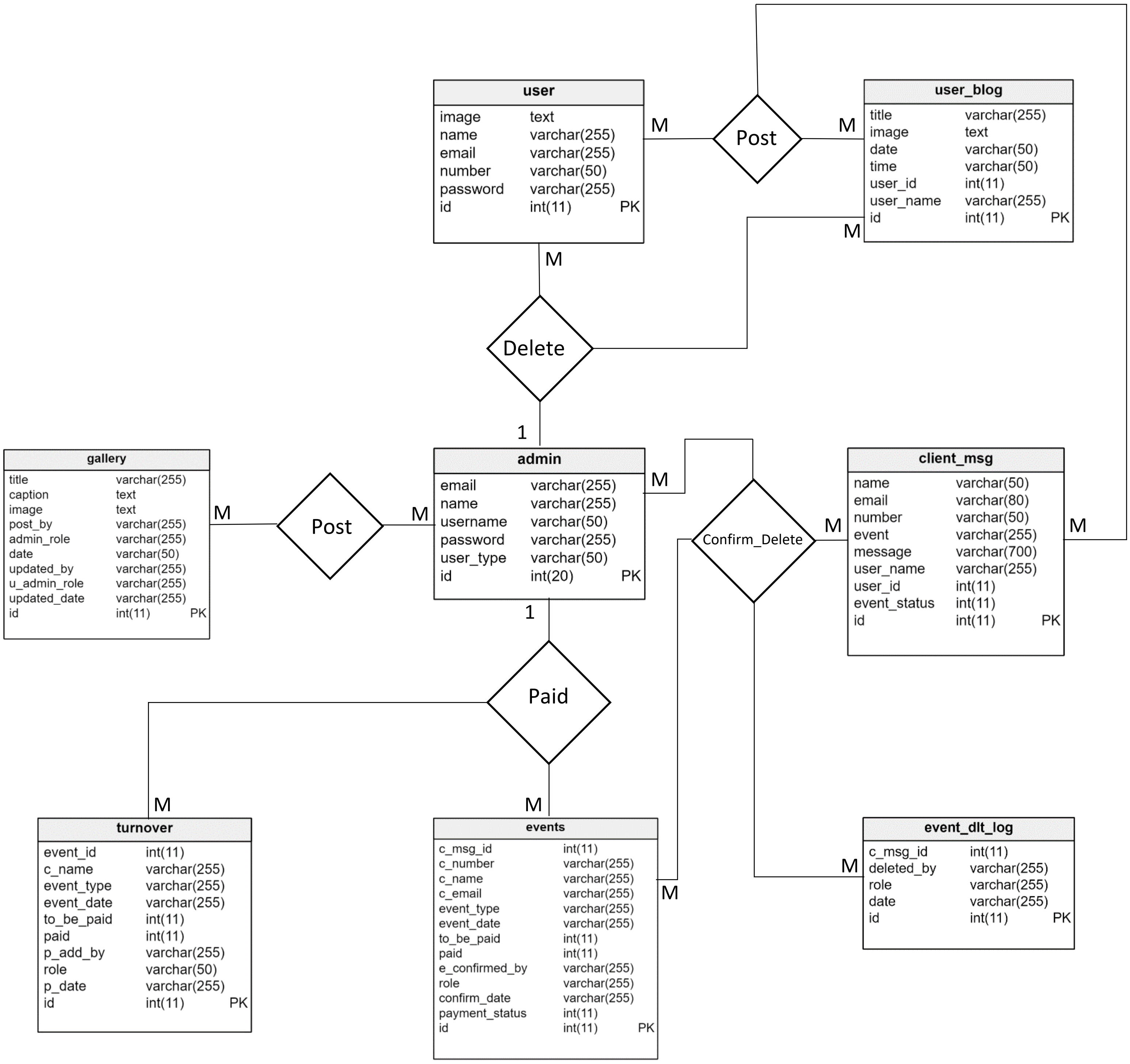


Fig. Entity-relationship diagram for Memories Chronicle

▪ Features & Description:

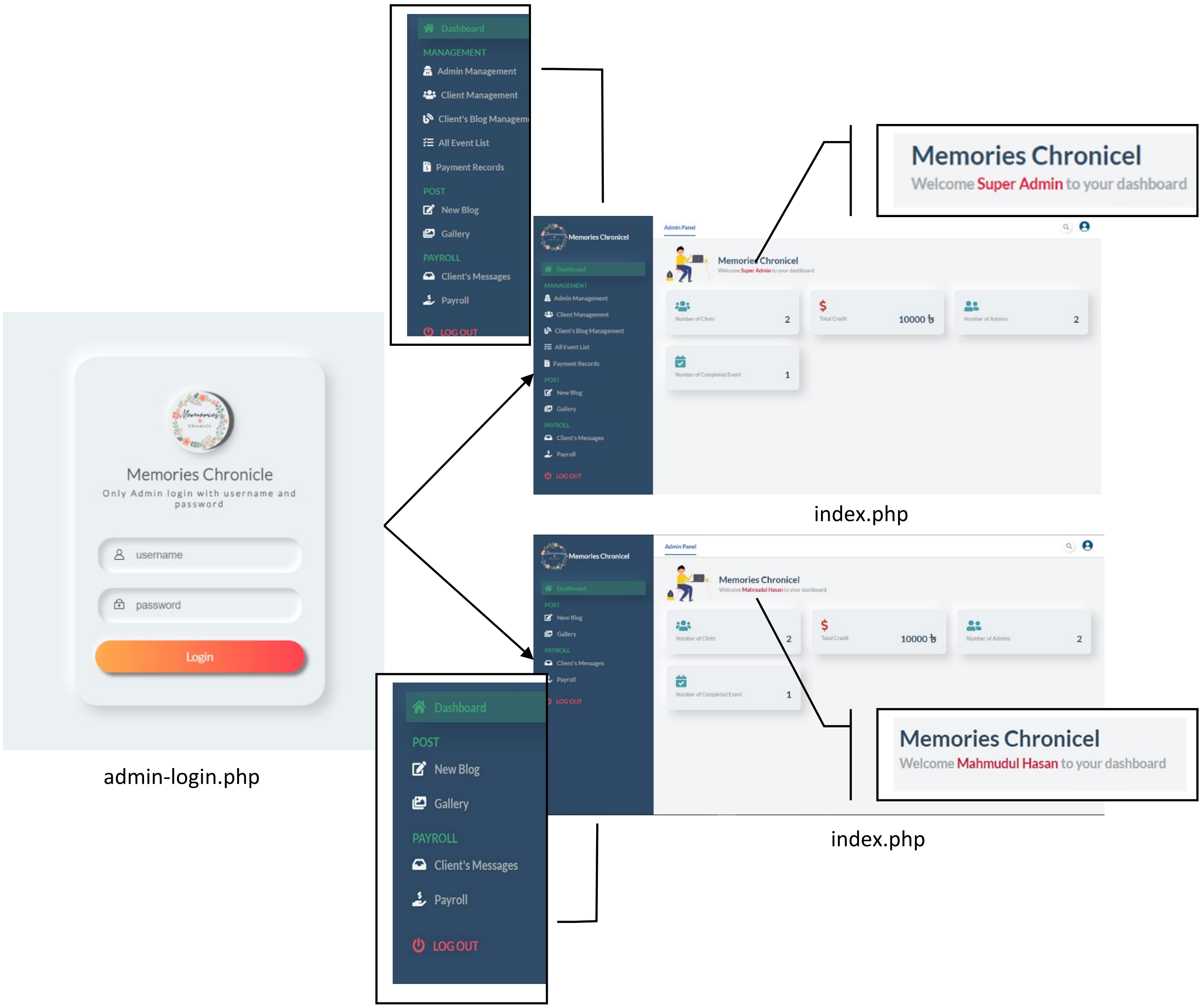
Let’s divide the Website into two part.

☞ Admin Side

☞ User Side

Admin Side

1. Admin Login and Admin Dashboard:

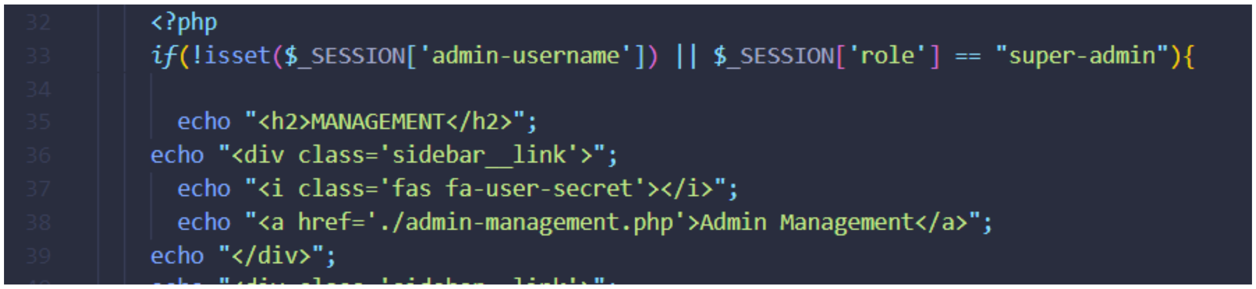


admin-login.php

index.php

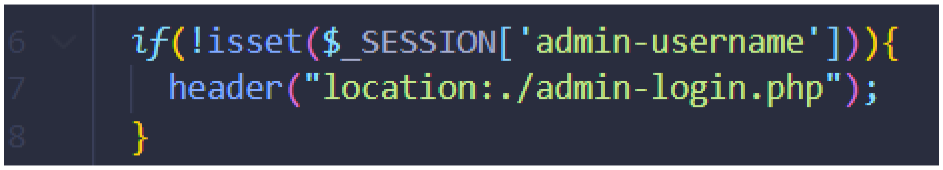
index.php

Behind Admin login form and Admin Dashboard there is two page named admin-login.php and index.php. There are 2 types of admin one is **super-admin** and another one is **normal admin**_if a super-admin or normal admin logged in he/she can see their name above the dashboard page. But there is some difference in the action menu. Super-admin has extra features called **Management** which a normal admin doesn't have. Management section have some extra features that gives a super admin some extra controls over the website. To do so we check whether an admin is super admin or not by checking the role. We use the below condition in the sidebar menu code.



```
<?php
if(!isset($_SESSION['admin-username']) || $_SESSION['role'] == "super-admin"){

  echo "<h2>MANAGEMENT</h2>";
echo "<div class='sidebar__link'>";
  echo "<i class='fas fa-user-secret'></i>";
  echo "<a href='./admin-management.php'>Admin Management</a>";
echo "</div>";
```

For security we check in every page any of admin logged in or not. If not it will automatically redirect to the login page. To implement all this we use **Session variables**. Which is created when any admin logged in.



```
if(!isset($_SESSION['admin-username'])){
  header("location:./admin-login.php");
}
```

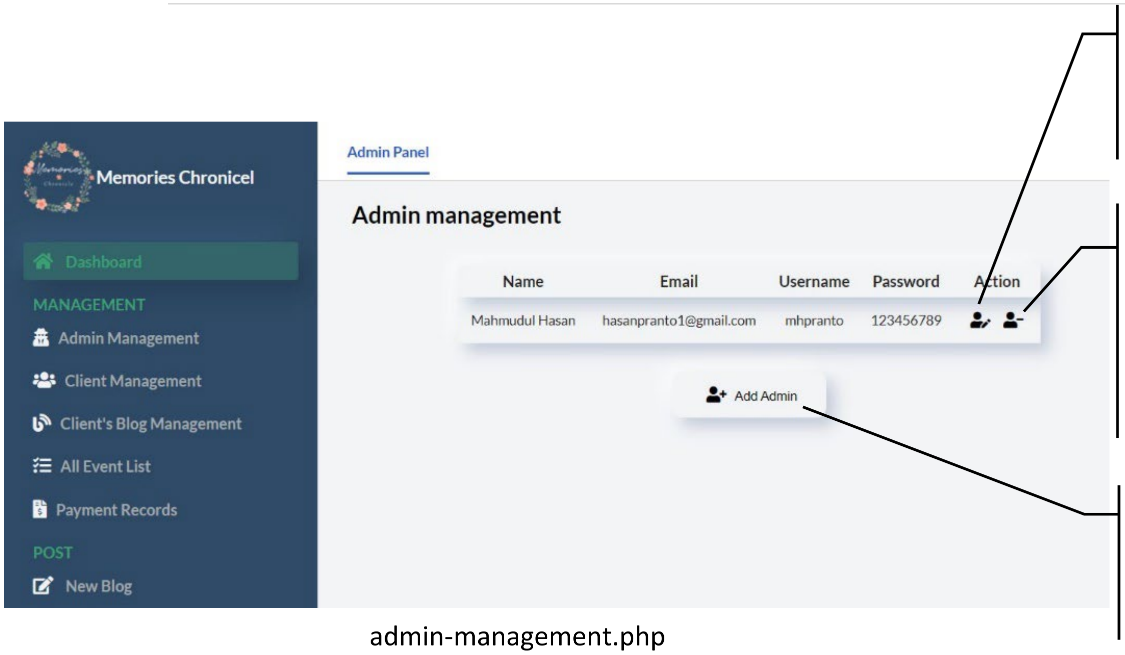
There are some other features like number of client, number of admin, total completed event & total credit. This all values are coming from database automatically because of some basic MySQL query.

2. Management Section:

This section is only visible for Super-admin. Management section has 5 sub category,

2.1. Admin Management:

In this a super admin can manage admins for his/her website. He/she can add/remove new admins or can edit existence admins credentials or information. These all implement by MySQL Insert, Update, and Delete query along with html form, input, and table tag.



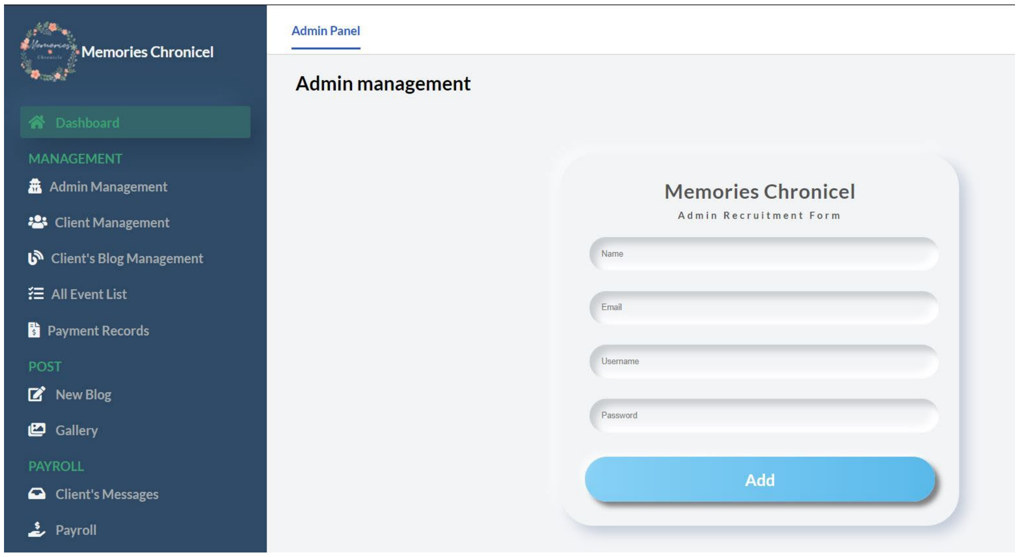
Name	Email	Username	Password	Action
Mahmudul Hasan	hasanpranto1@gmail.com	mhpranto	123456789	

This button will redirect to the admin recruitment form for edit in admin-edit.php

This button will redirect to admin-remove.php. Which will execute delete query and pop up ***"Admin remove successfully"*** in alert box.

This button will redirect to the admin recruitment form in admin-add.php

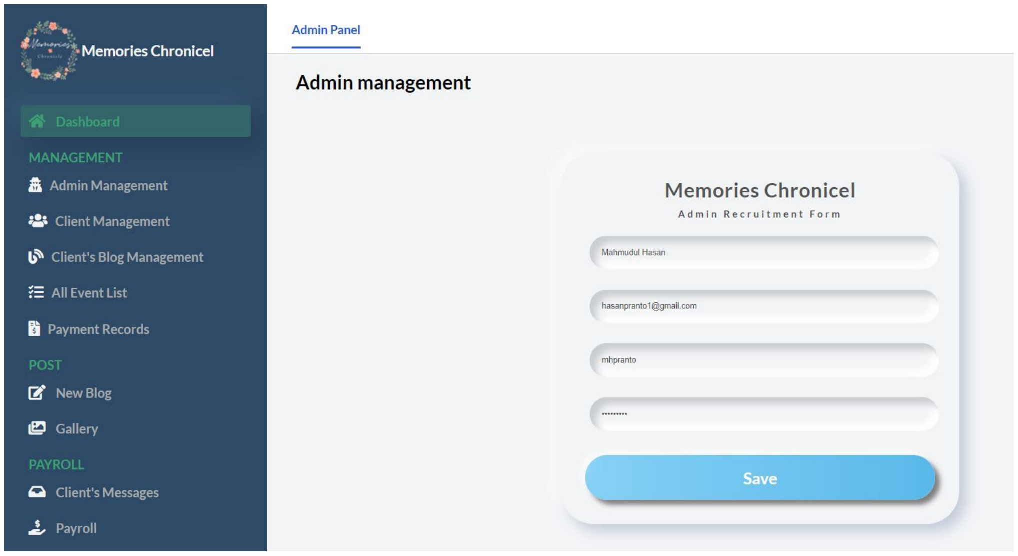
admin-management.php



When super admin add all the necessary information for a new admin and press Add button it will execute Insert query and add the information for a new admin in the database and pop up ***"Admin added Successfully"*** in alert box.

admin-add.php

When super admin edit all the necessary information that he want and press save button it will execute update query and updated the information for that admin in the database. To make sure specific admin we use admin id which is sent and retrieve through the URL by using GET method. After that ***"Information Updated Successfully"*** message pop up in alert box.

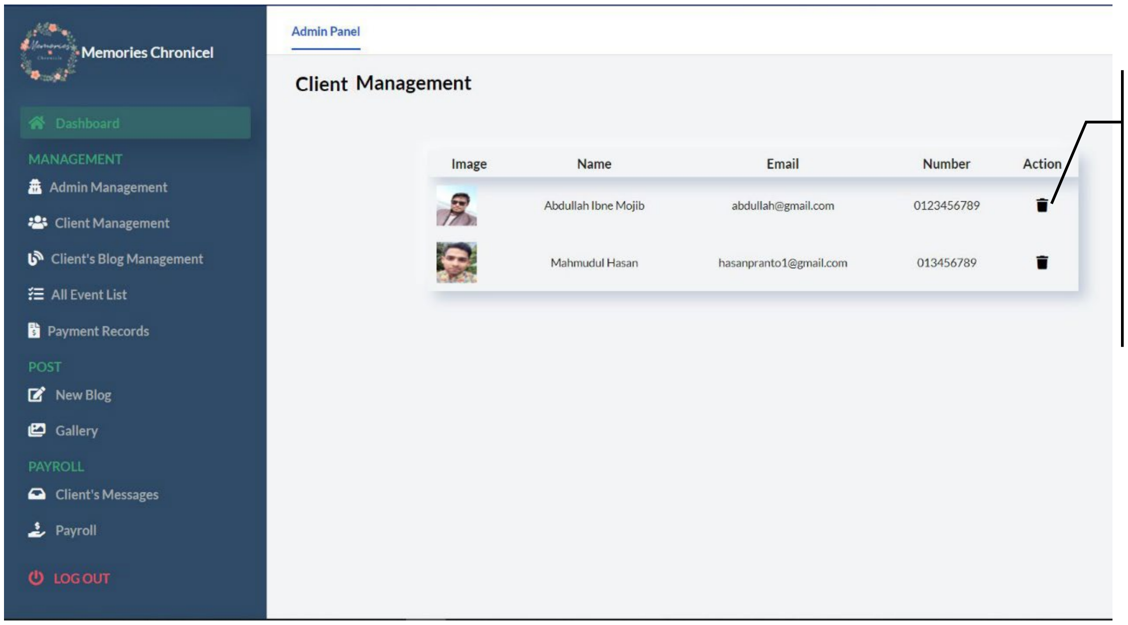


Mahmudul Hasan
hasanpranto1@gmail.com
mhpranto
....

admin-edit.php

2.2. Client Management:

Client management option will redirect to the client-management.php page. This page shows every registered client's information and super admin can see information and only have access to **Delete** clients if he/she wants to.



Client Management

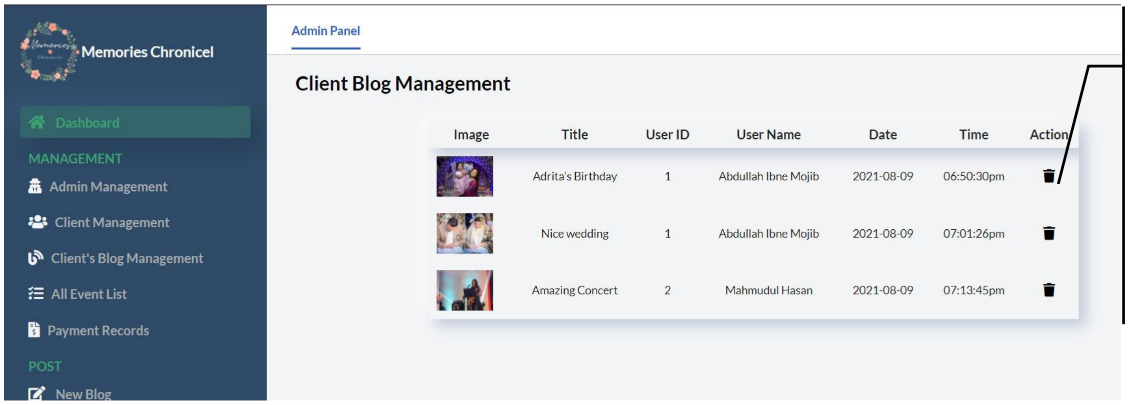
Image	Name	Email	Number	Action
	Abdullah Ibne Mojib	abdullah@gmail.com	0123456789	
	Mahmudul Hasan	hasanpranto1@gmail.com	013456789	

This button will redirect to client-remove.php. Which will execute delete query and pop up ***"Client remove successfully"*** in alert box. To specify clients we use GET method like we use before.

Client-management.php

2.3. Client's Blog Management:

Like Client management, Client's Blog management option will redirect to the client blog management.php page. This page shows every posts that clients were posted and super admin can see them and only have access to **Delete** them if he/she wants to.



Client Blog Management

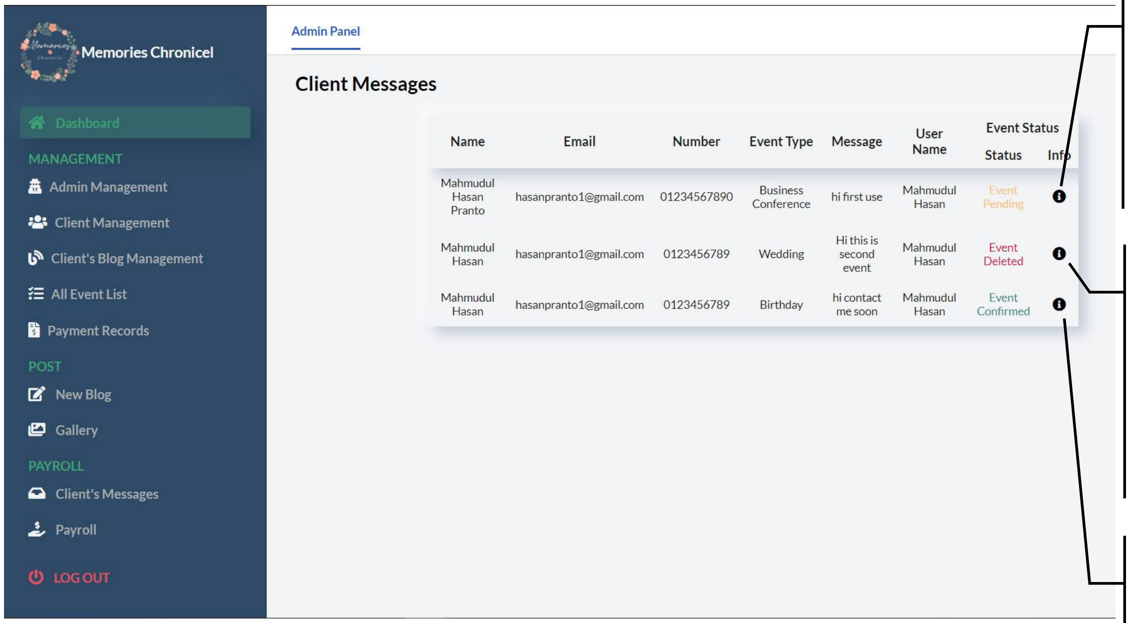
Image	Title	User ID	User Name	Date	Time	Action
	Adrita's Birthday	1	Abdullah Ibne Mojib	2021-08-09	06:50:30pm	
	Nice wedding	1	Abdullah Ibne Mojib	2021-08-09	07:01:26pm	
	Amazing Concert	2	Mahmudul Hasan	2021-08-09	07:13:45pm	

This button will redirect to client-post-remove.php. Which will execute delete query and pop up ***"Clients post remove successfully"*** in alert box. To specify clients we use GET method like we use before.

Clients blog management.php

2.4. All Event List:

All event list option features a client messages page. Where each and every messages shown which are sent by user. The messages shown along with the status and an ‘i’ button. The button redirect to the information page for individual status.



Client Messages

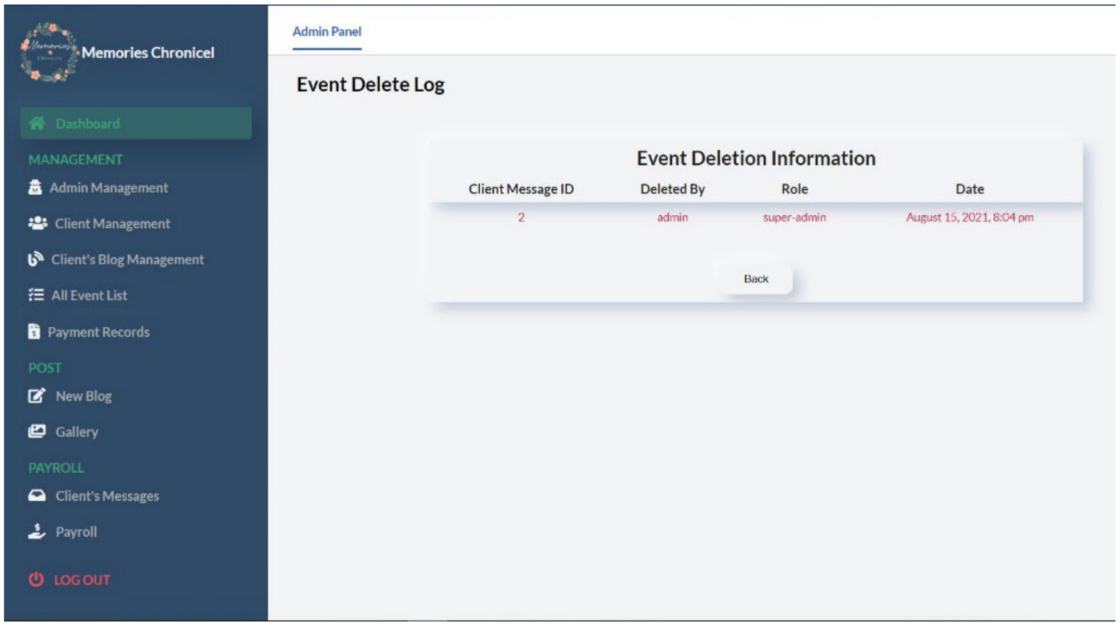
Name	Email	Number	Event Type	Message	User Name	Event Status: Status	Event Status: Info
Mahmudul Hasan Pranto	hasanpranto1@gmail.com	01234567890	Business Conference	hi first use	Mahmudul Hasan	Event Pending	ℹ
Mahmudul Hasan	hasanpranto1@gmail.com	0123456789	Wedding	Hi this is second event	Mahmudul Hasan	Event Deleted	ℹ
Mahmudul Hasan	hasanpranto1@gmail.com	0123456789	Birthday	hi contact me soon	Mahmudul Hasan	Event Confirmed	ℹ

All-event-list.php

If the status is “**Event Pending”** the i button will redirect to the order.php page where the messages are listed for either confirmation or deletion.

If the status is “**Event Deleted”** the i button will redirect to the delete-log.php page where the super admin can see who and when deleted the message for an event.

If the status is “**Event Confirmed”** the i button will redirect to the payroll.php page where the admins can add payment for the events that are completed or ongoing.



Event Delete Log

Event Deletion Information

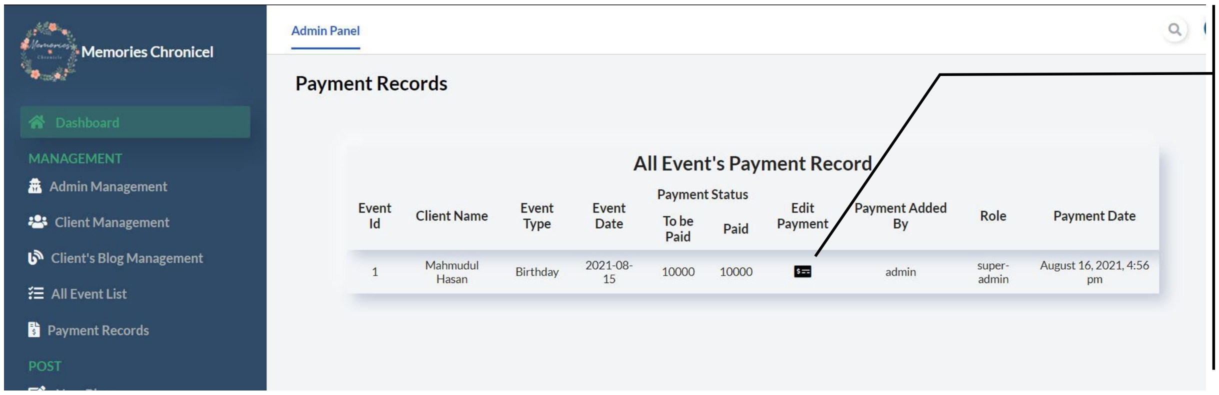
Client Message ID	Deleted By	Role	Date
2	admin	super-admin	August 15, 2021, 8:04 pm

Delete-log.php

Event Delete log page shows the information of a deleted information. It shows when the event message is deleted and by whom. We use the same technique of GET method to show the specific event info.

2.5. Payment Records:

Payment Records page shows all the completed event along with payment in the payroll.php page. If any change need to be done in payment amount it can be change here and this action can be done by super admin only. No other admins can't change any amount after they submitted once any event paid amount in payroll page.



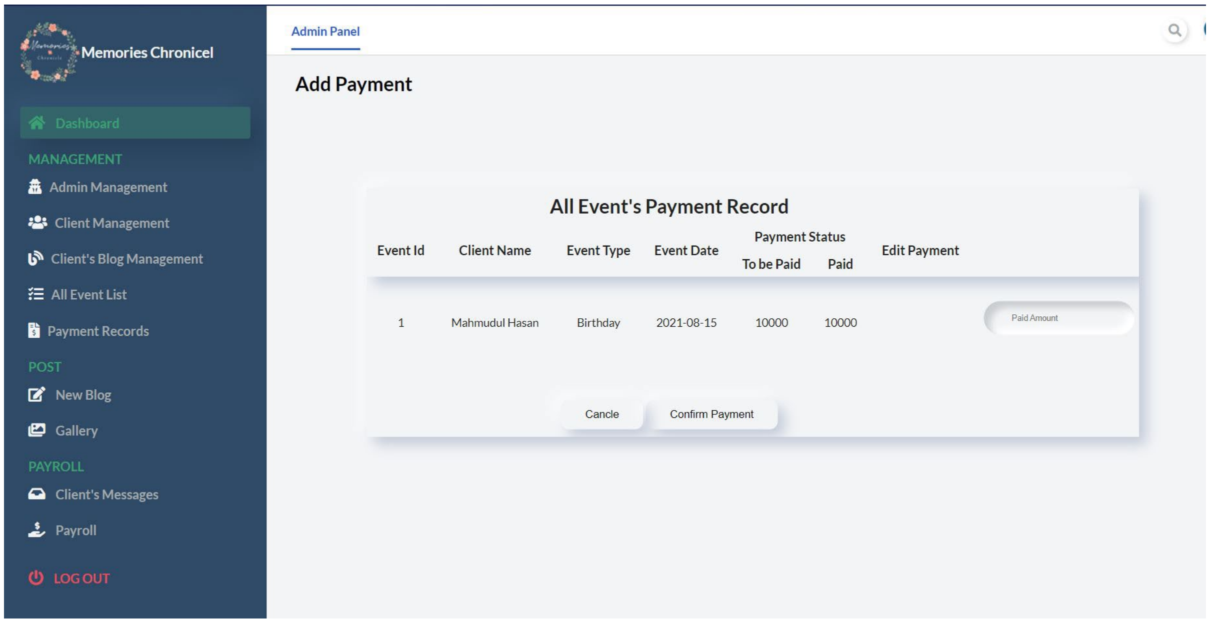
Payment Records

All Event's Payment Record

Event Id	Client Name	Event Type	Event Date	Payment Status		Edit Payment	Payment Added By	Role	Payment Date
				To be Paid	Paid				
1	Mahmudul Hasan	Birthday	2021-08-15	10000	10000		admin	super-admin	August 16, 2021, 4:56 pm

This button allows super admin to override the paid amount that previously added in payroll section if necessary. This button redirect to the edit-payment.php to edit the amount.

Payment records.php



Add Payment

All Event's Payment Record

Event Id	Client Name	Event Type	Event Date	Payment Status		Edit Payment
				To be Paid	Paid	
1	Mahmudul Hasan	Birthday	2021-08-15	10000	10000	

Edit-payment.php page allows to edit the paid amount that previously added. When super admin add new amount and confirm then the amount update the paid amount in database.

Edit-payment.php

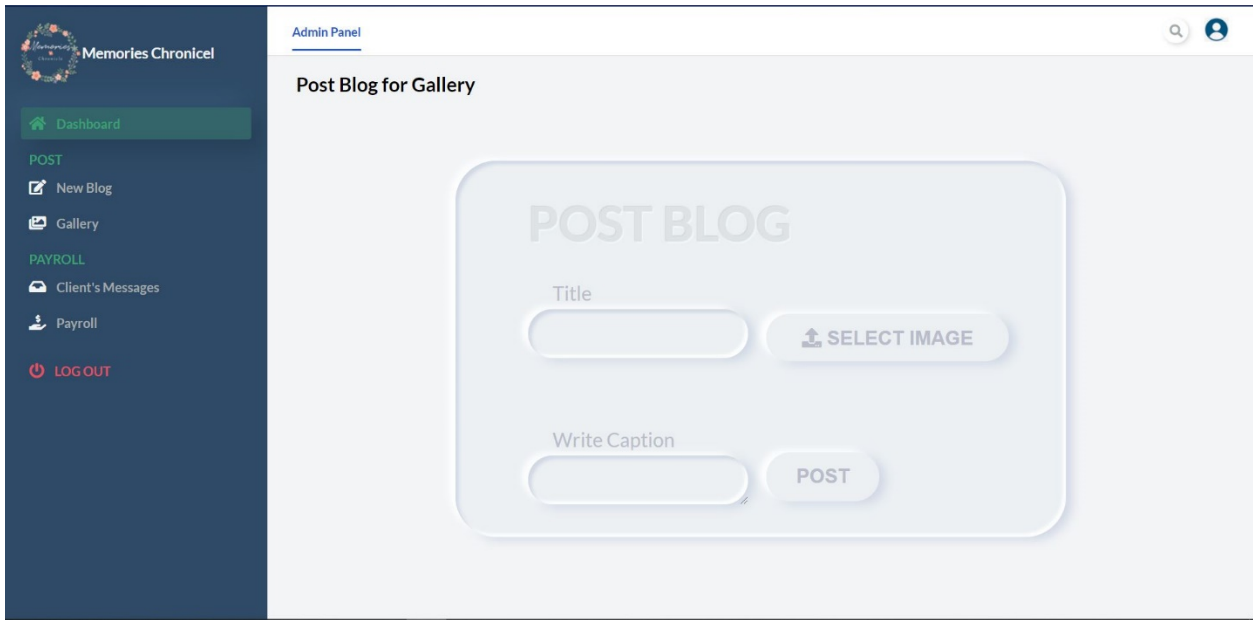
And this is the last sub category of management section these 5 options can only visible to super admin and only super admin has full rights on these 5.From now on all the options are common for all type of admin.

3. Post Section :

Post section consist of 2 sub category. Both these category/option is for post/edit post in gallery. This section is accessible from both admin and super admin.

3.1. New Blog:

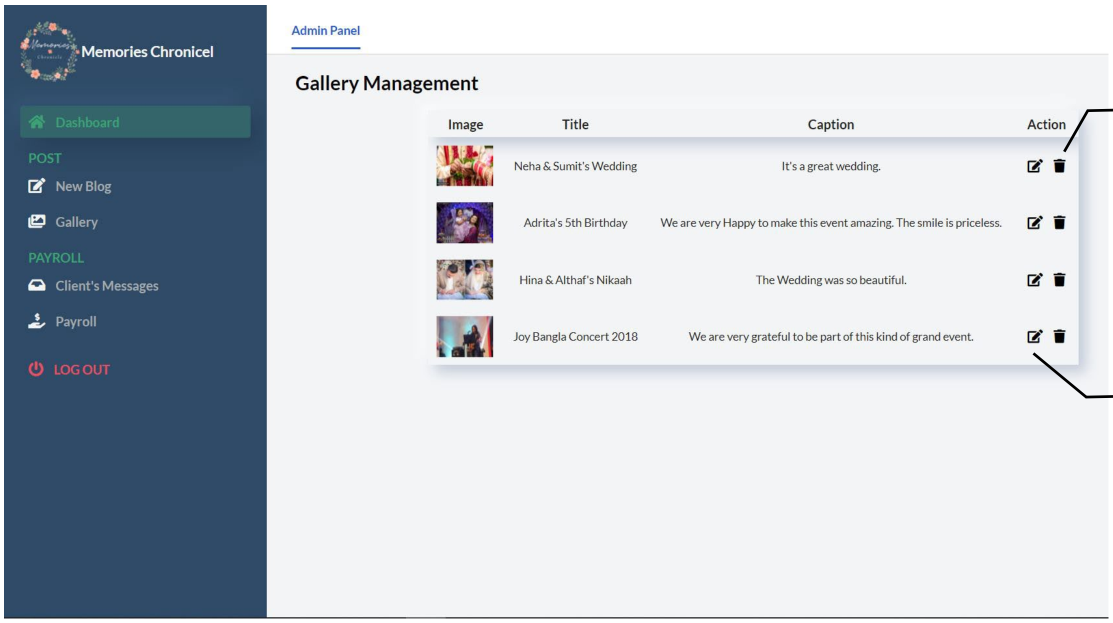
New blog page (blog.php) is for posting images for the gallery in the user side. It consist of a form which take image, title, and caption for the post.



blog.php

3.2. Gallery:

Gallery.php page is for editing post in the gallery. Admins can edit or delete post from gallery. All the posts enlisted in this page.



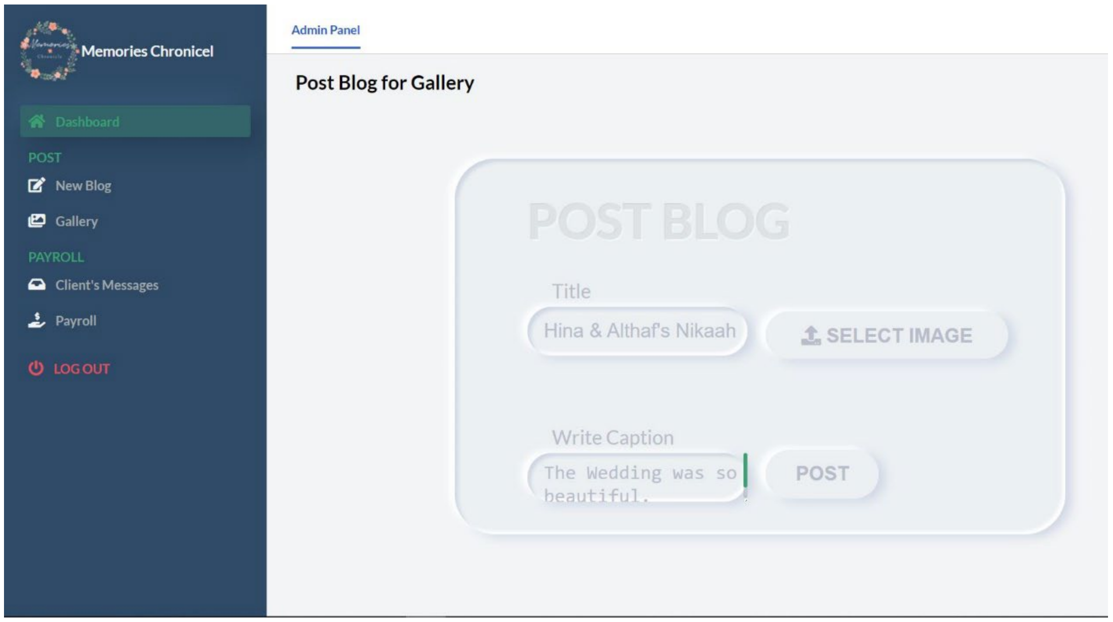
Gallery Management

Image	Title	Caption	Action
	Neha & Sumit's Wedding	It's a great wedding.	
	Adrita's 5th Birthday	We are very Happy to make this event amazing. The smile is priceless.	
	Hina & Althaf's Nikaah	The Wedding was so beautiful.	
	Joy Bangla Concert 2018	We are very grateful to be part of this kind of grand event.	

This button is to delete a post from the gallery. It redirects to the gallery-post-remove.php page. We use the same GET method technique to specify the post to be deleted.

This button is to edit a post from the gallery. This button redirect to the gallery-post-edit.php.

gallery.php



Post Blog for Gallery

Title: Hina & Althaf's Nikaah

Write Caption: The Wedding was so beautiful.

In this page according to post id data will show in the input field to edit. After editing admins can press Post button to update the content of that specific post. We use GET method to specify the post and Update query to update the post in the database.

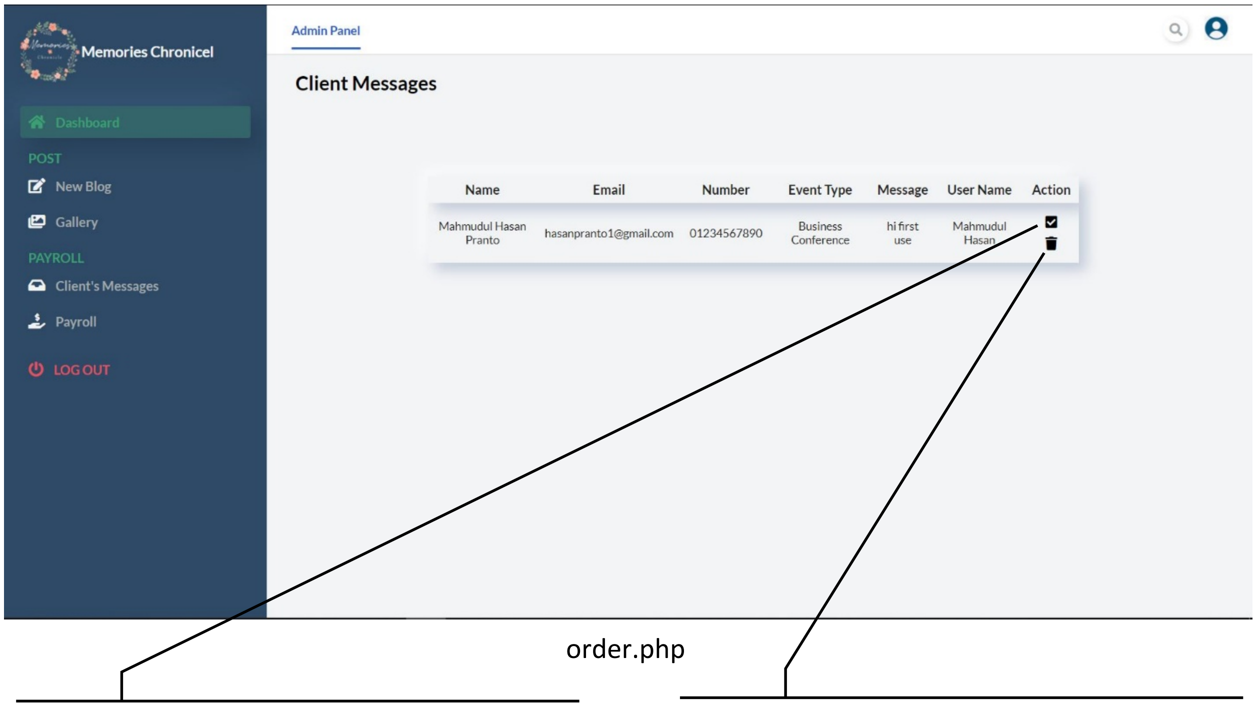
gallery-post-edit.php

4. Payroll:

This is the last section of the admin panel. In this section there are 2 category. This section also accessible by both admin and super admin like the previous section we describe before.

4.1. Client's messages :

Order.php page AKA client's message page shows a list of messages which are sent by client as there event request. Admins has the ability to delete the message or confirm the event. Client gives the necessary information to communicate. If any admin contacts with the client and it's confirmed to do the event. Then the admin simply confirm the event by pressing confirm button and filling up a confirmation form in the event-confirmation.php page. All the client messages comes from contact page from user side.



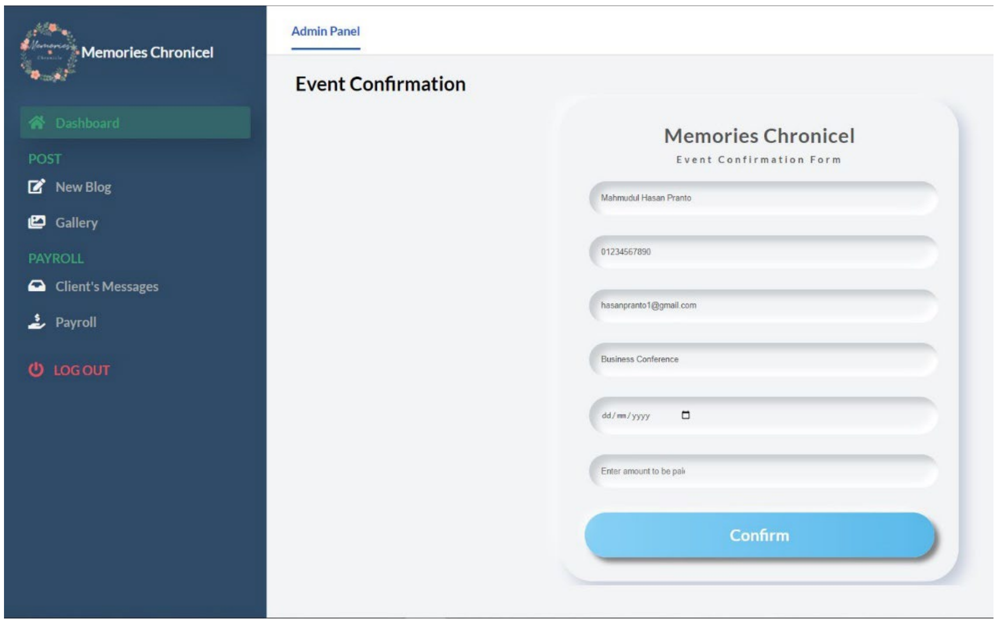
Client Messages

Name	Email	Number	Event Type	Message	User Name	Action
Mahmudul Hasan Pranto	hasanpranto1@gmail.com	01234567890	Business Conference	hi first use	Mahmudul Hasan	☑ 🗑

order.php

This button redirect admins to the event-confirmation.php page where they need to fill up a confirmation form which send data to the payroll page

This button redirect admins to the event-remove.php page and execute an update query to update the active status in the corresponding data table. We didn't delete the message because we keep the data to see in the delete log and all event list page. Here also we use GET method to specify the message.



Event Confirmation

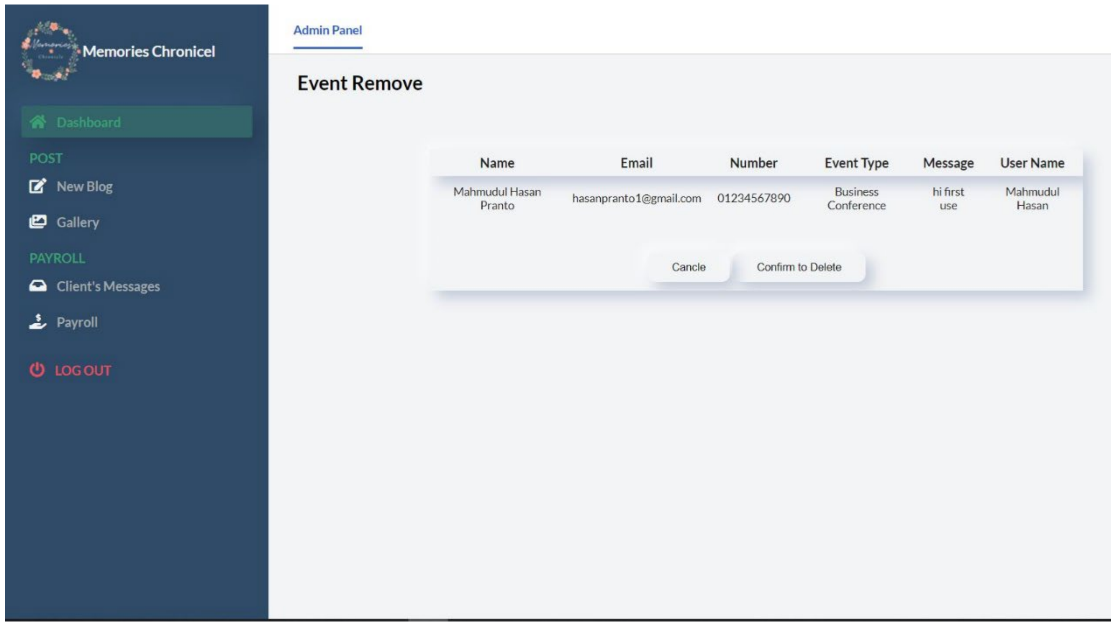
Memories Chronicel

Event Confirmation Form

Mahmudul Hasan Pranto
01234567890
hasanpranto1@gmail.com
Business Conference
dd / mm / yyyy
Enter amount to be pai

Event-confirmation.php

This page has a form. And some input field is disabled to edit because those data is coming from the specific message that an admin wants to confirm. Admin can just input the date and amount to be paid and press the confirm button. After pressing the button all data along with message id sent to the table name events. From which data will be shown in the payroll page.



Event Remove

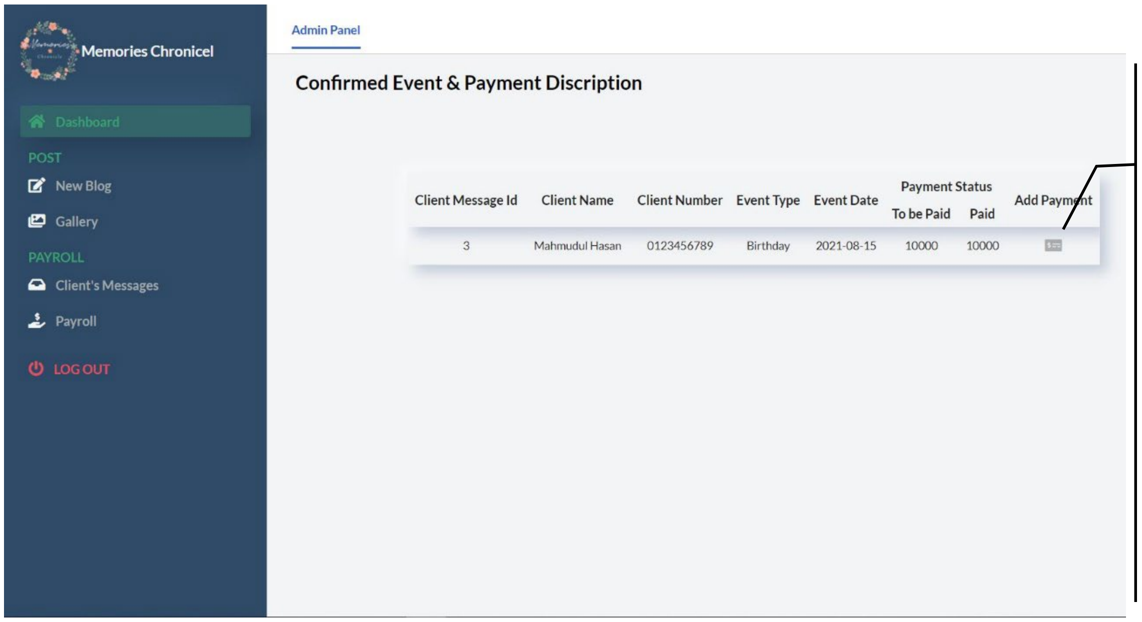
Name	Email	Number	Event Type	Message	User Name
Mahmudul Hasan Pranto	hasanpranto1@gmail.com	01234567890	Business Conference	hi first use	Mahmudul Hasan

Event-remove.php

This page appears when admins try to delete any event. Means when they try to delete any client message it shows the page if he/she is confirm to delete the message. If admins press confirms then it execute a query to just update the active status in the client_msg table in the database.

4.2. Payroll:

Payroll page shows the list of confirmed events along with a button to add payment amount. This page is to add the payment amount given by the clients for their events. After adding payment it send data to the turnover table which record shown in the payment records page.



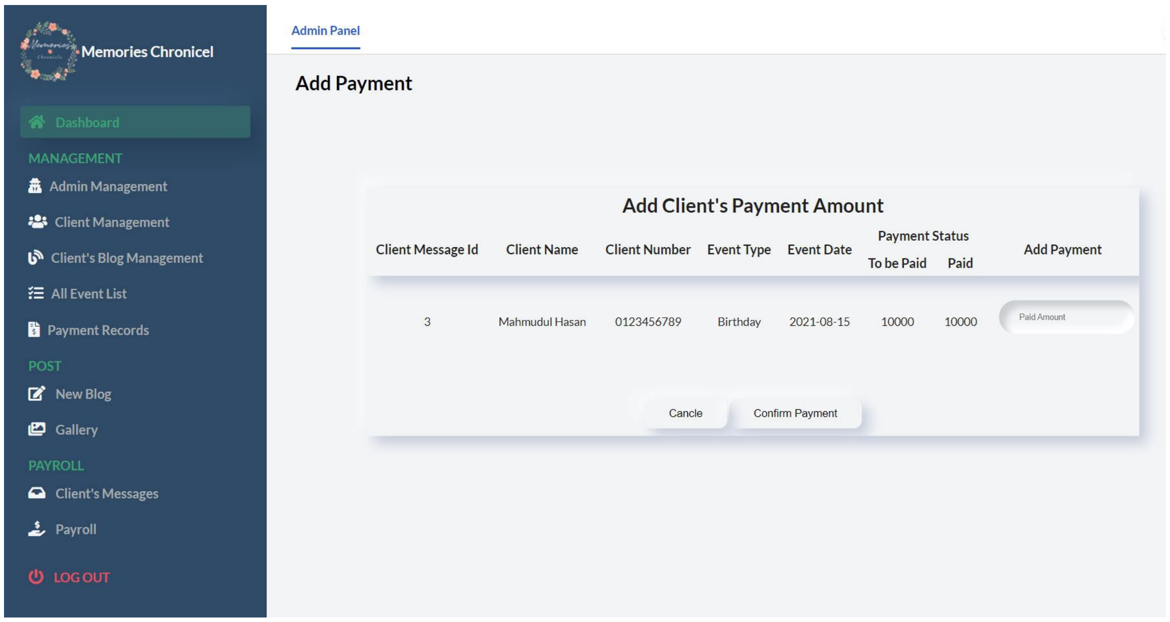
Confirmed Event & Payment Discription

Client Message Id	Client Name	Client Number	Event Type	Event Date	Payment Status		Add Payment
					To be Paid	Paid	
3	Mahmudul Hasan	0123456789	Birthday	2021-08-15	10000	10000	

Add payment button send admins to the add-payment.php where they can add the amount of paid and it will inserted in events and turnover table in the database.

Payroll.php

Admin can add payment amount through this page once. After adding amount the button will be disable. If he wants to edit the amount he have to go through the payment records page. Which can be accessible only super admin.



Add Payment

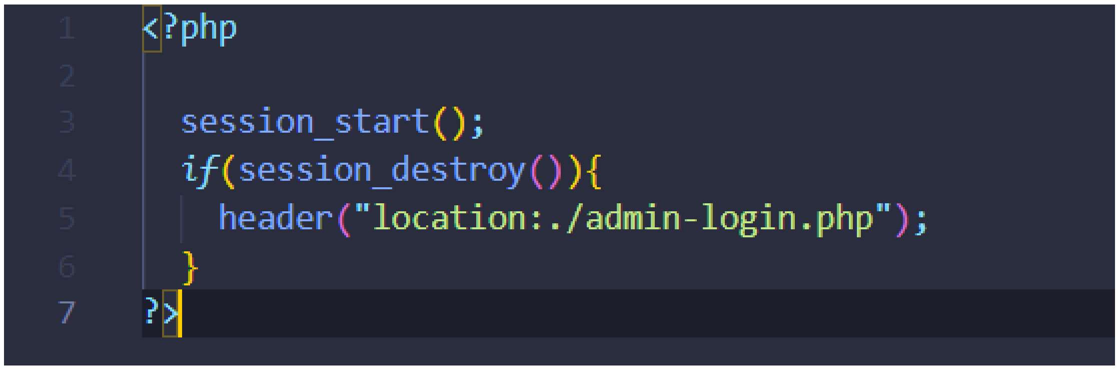
Add Client's Payment Amount

Client Message Id	Client Name	Client Number	Event Type	Event Date	Payment Status		Add Payment
					To be Paid	Paid	
3	Mahmudul Hasan	0123456789	Birthday	2021-08-15	10000	10000	Paid Amount

Add-payment.php

5. Logout :

This option logged out any admin from the admin panel by using logout.php. This file includes a code to destroy the session variables and redirect to the admin-login.php page again.



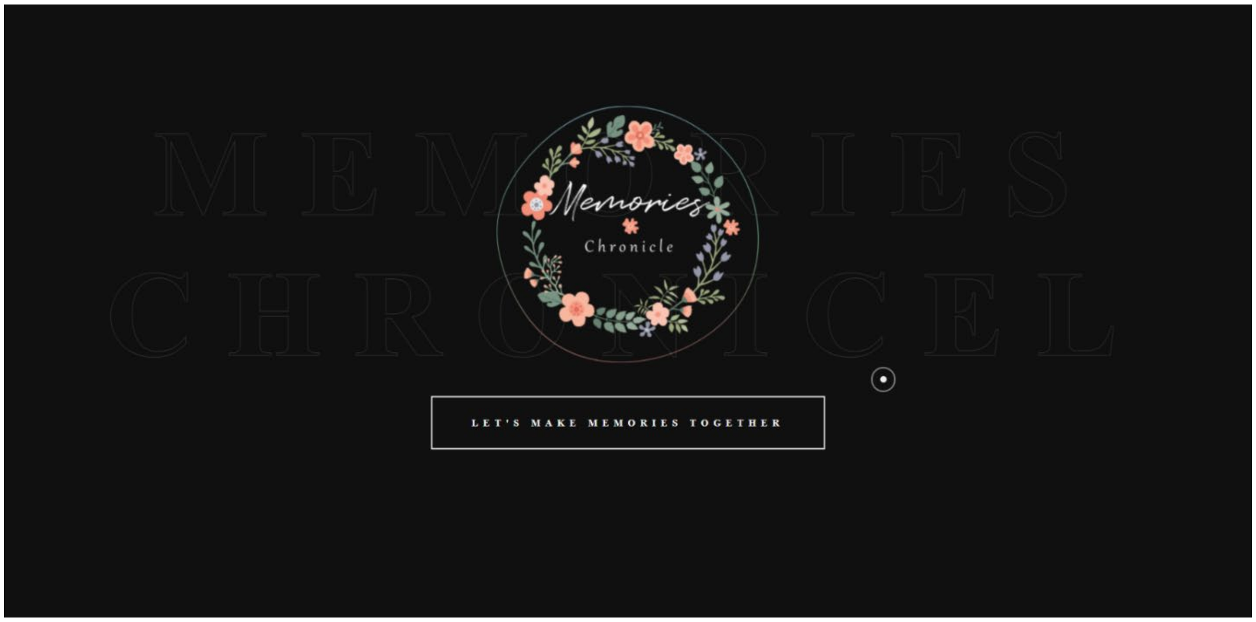
```php
<?php

  session_start();
  if(session_destroy()){
    header("location:./admin-login.php");
  }
?>
```

User Side

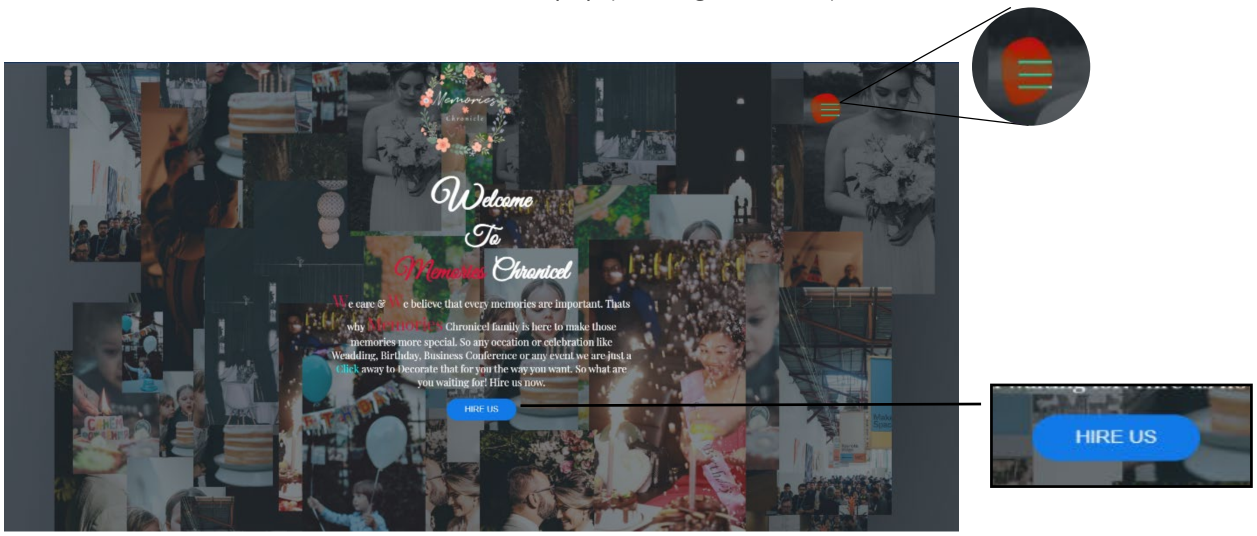
Memories Chronicle is a website for event planner. To make it eye catchy for user we uses follow a specific color palate for the whole website along with some js animations. We also use custom page transition and cursor by using html, js and CSS. Basically in user side all pages are accessible for all kinds of visitor accepts profile page and if anyone wants to contact with us through a contact form they must have an account. We follow the Glassmorphism CSS style almost everywhere in the entire website to give an amazing user experience to the visitors. Let's explore more about the website for user side,

6. Home Page and Menu :

For home page we start with a transition along with the site logo middle and site name at the back and a button that redirect to the main home page. All these transition page and main home page code are in the same file called index.php. In the home page there is some text describe the website and a hire us button along with a background js photo shuffling animation. On the top right corner there is a hamburger menu which opens the menu.



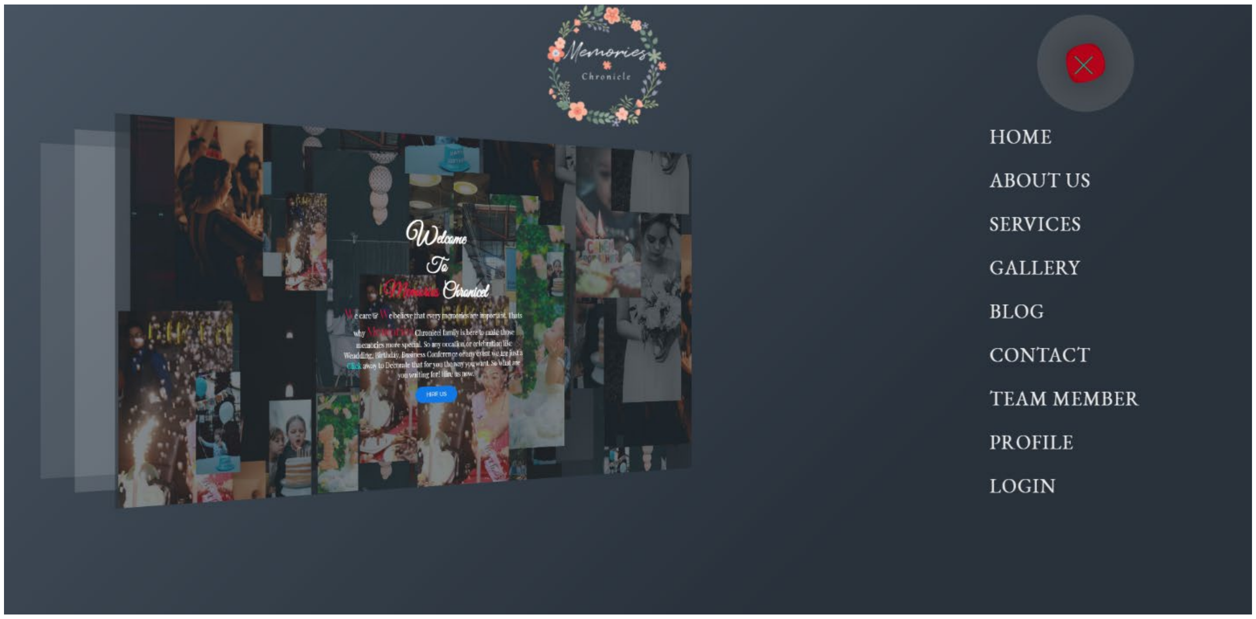
index.php (starting transition)



We care & We believe that every memories are important. Thats why Memories Chronicel family is here to make those memories more special. So any occation or celebration like Weadding, Birthday, Business Conference or any event we are just a Click away to Decorate that for you the way you want. So what are you waiting for! Hire us now.

index.php (home page)

Hire us button redirect to the contact.php page and the upper right corners hamburger menu leads to the menu option of the website. In the menu section we use html list element to implement the menu items. When the menu section is active, we make the site a perspective view along with the menu items. We follow this same menu option to all our web pages.



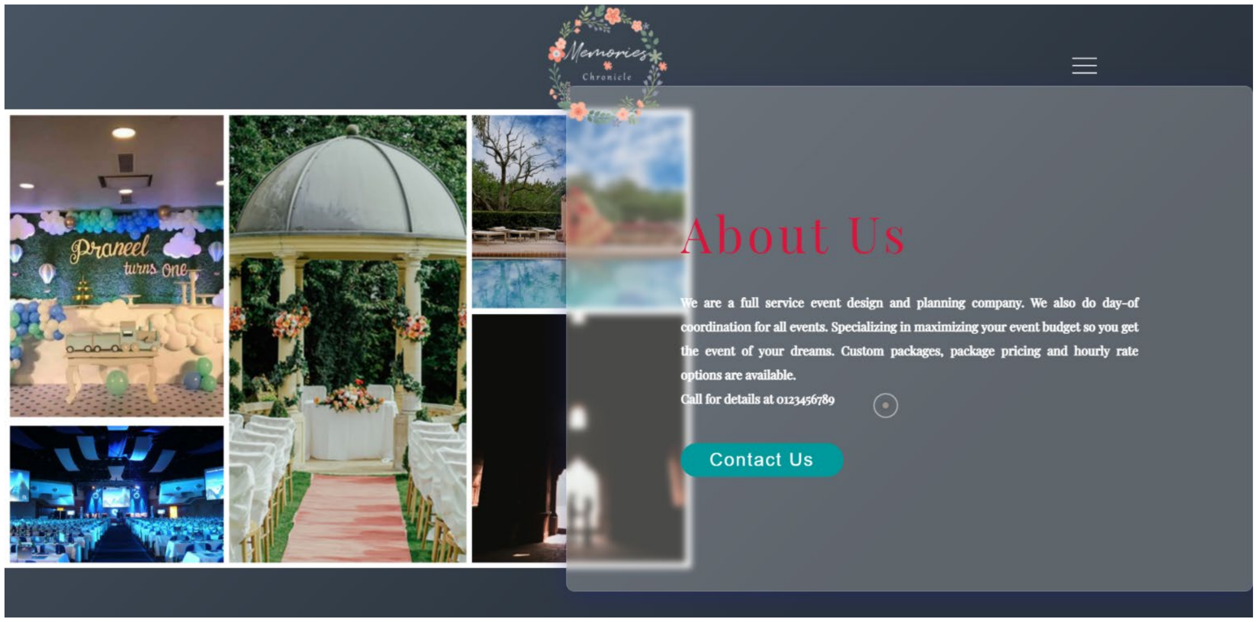
Welcome To Memories Chronicel

We care & We believe that every memories are important. Thats why Memories Chronicel family is here to make those memories more special. So any occation or celebration like Weadding, Birthday, Business Conference or any event we are just a Click away to Decorate that for you the way you want. So what are you waiting for! Hire us now.

index.php (menu section)

7. About us :

About us page or about.php page consist of sum code along with some text to describe the website. Also there is a button called **“Contact Us”** to redirect user to the contact us.php page.



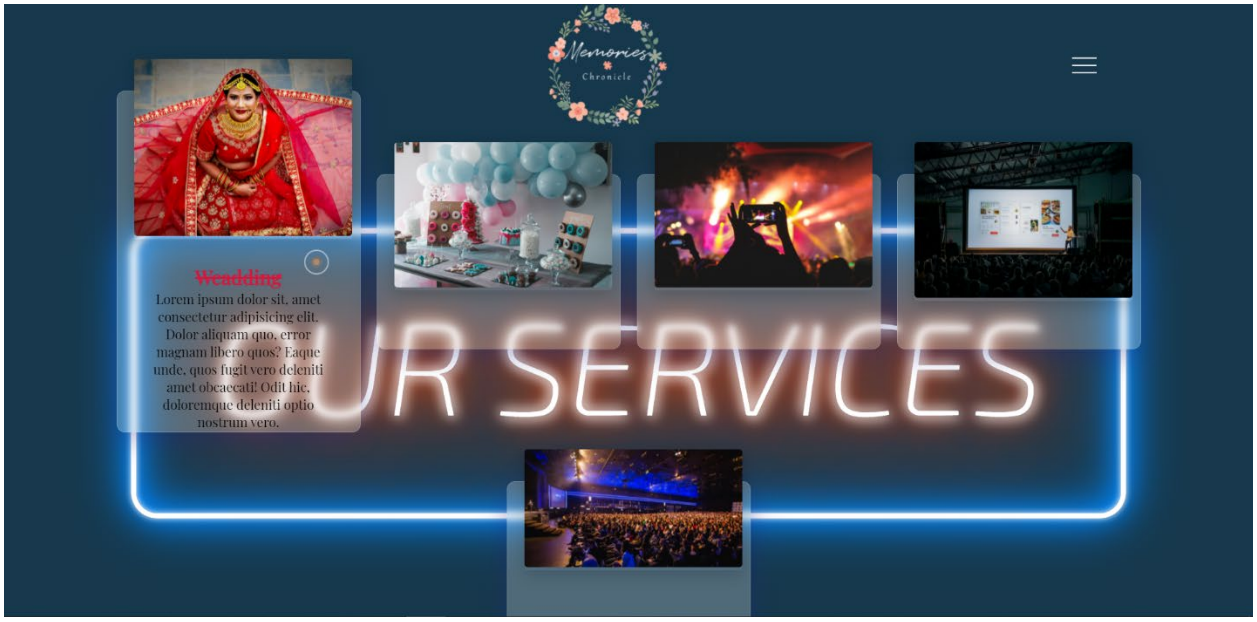
About Us

We are a full service event design and planning company. We also do day-of coordination for all events. Specializing in maximizing your event budget so you get the event of your dreams. Custom packages, package pricing and hourly rate options are available.
Call for details at 0123456789

About.php

8. Services :

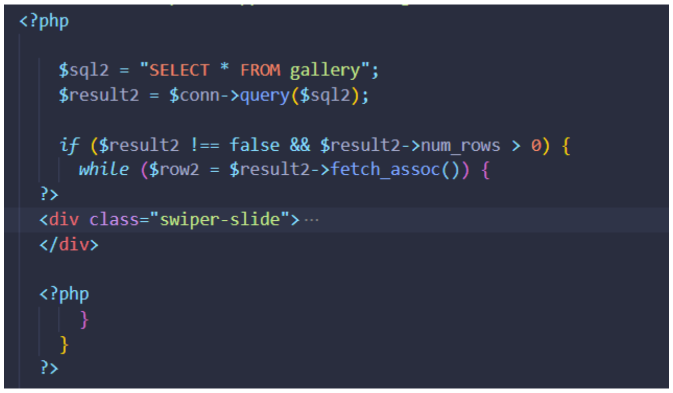
Services page is have multiple card item along with an image and a hover effect to describe the services of our webpage. We use CSS animation to implement a background neon effected text that shows behind the card items.


Memories Chronicle
Weadding
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Dolor aliquam quo, error magnam libero quos? Eaque unde, quos fugit vero deleniti amet obcaecati! Odit hic, doloremque deleniti optio nostrum vero.
UR SERVICES

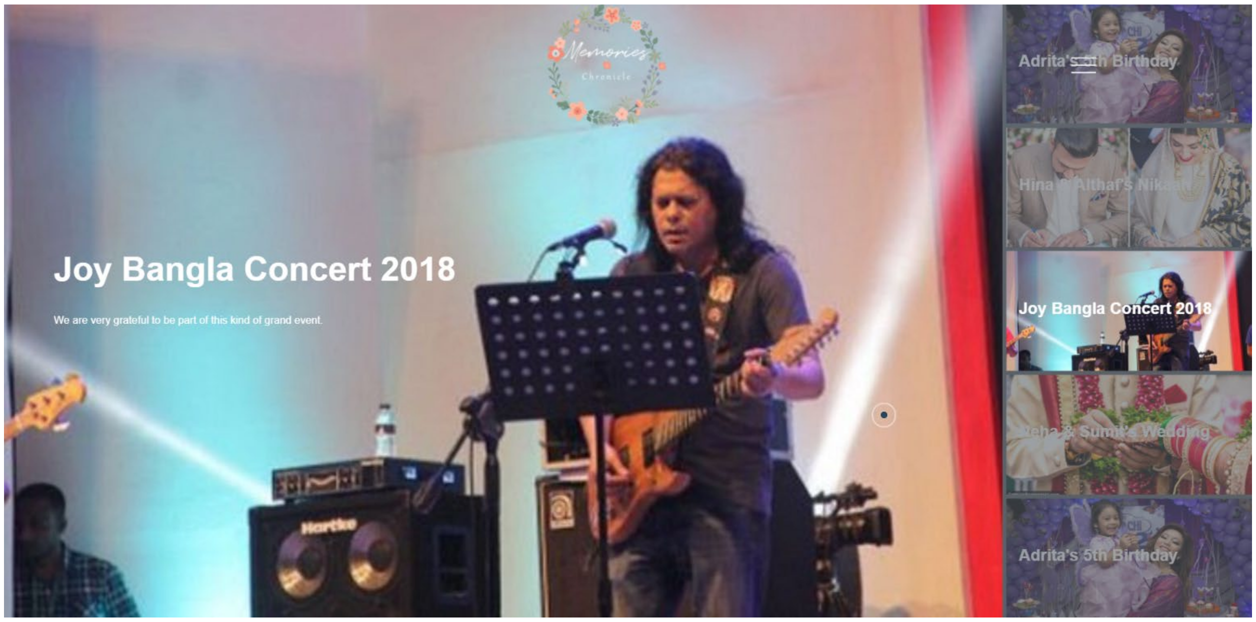
services.php

9. Gallery :

In gallery.php page we use an image changing effect using js. Which shows the images that Admins are uploaded through Admin panel. So that means Gallery page is controlled by the admin to showcase the work that website done previously. We use swipe js to implement the smooth image slider and image sidebar. All the data comes from the table named “gallery” in the database and by using for loop we print each of the data in the page.



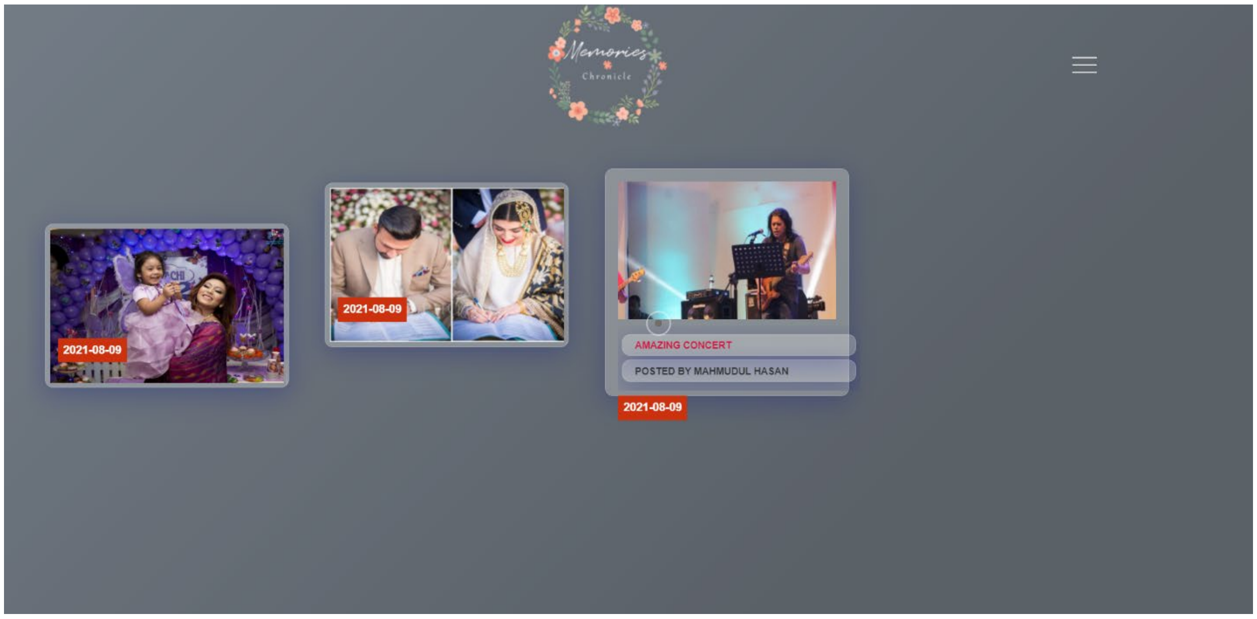
```
<?php

    $sql2 = "SELECT * FROM gallery";
    $result2 = $conn->query($sql2);

    if ($result2 !== false && $result2->num_rows > 0) {
      while ($row2 = $result2->fetch_assoc()) {
  ?>
  <div class="swiper-slide">···
  </div>

  <?php
      }
    }
  ?>
```



gallery.php

10. Blog :

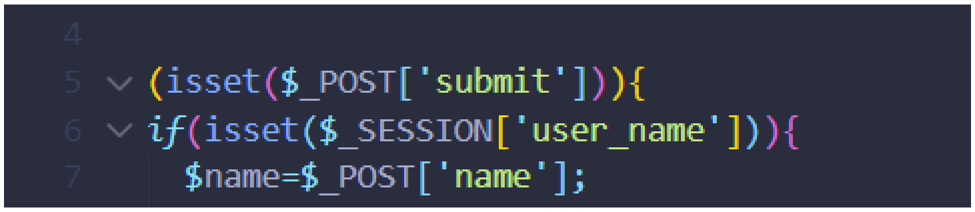
In blog.php page users or visitors can see the blog posted by other users along with title, posted by and date of the post. It's a horizontal post slider to maintain the entire esthetics of the website.



Blog.php

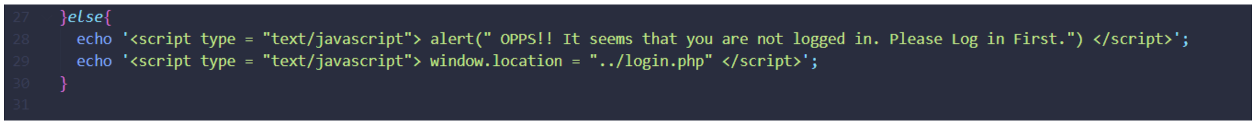
11. Contact :

Contact.php page consist of a contact form. Which is represent as event request. Users can fill up the form to contact with admins by giving necessary contact information and the type of event that he/she want to be done. Contact us page is accessible for all visitor and user but if someone want to send message he/she have to log in with his/her account. It means **user login** is required for this. This system also implement by using session variables.



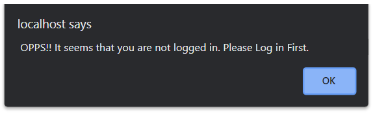
```
(isset($_POST['submit'])){
if(isset($_SESSION['user_name'])){
  $name=$_POST['name'];
```

All these check happens in contactcheck.php after the form trying to submit. If a visitor wants to send message it shows an alert message. That he didn't logged in and send him/her to the login page. From where the can create new account or login to the account.



```
}else{
  echo '<script type = "text/javascript"> alert(" OPPS!! It seems that you are not logged in. Please Log in First.") </script>';
  echo '<script type = "text/javascript"> window.location = "../login.php" </script>';
}
```

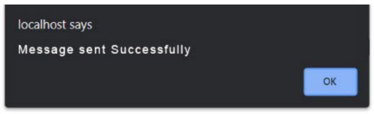
After login any user can send messages to the admins for their event.



localhost says

OPPS!! It seems that you are not logged in. Please Log in First.

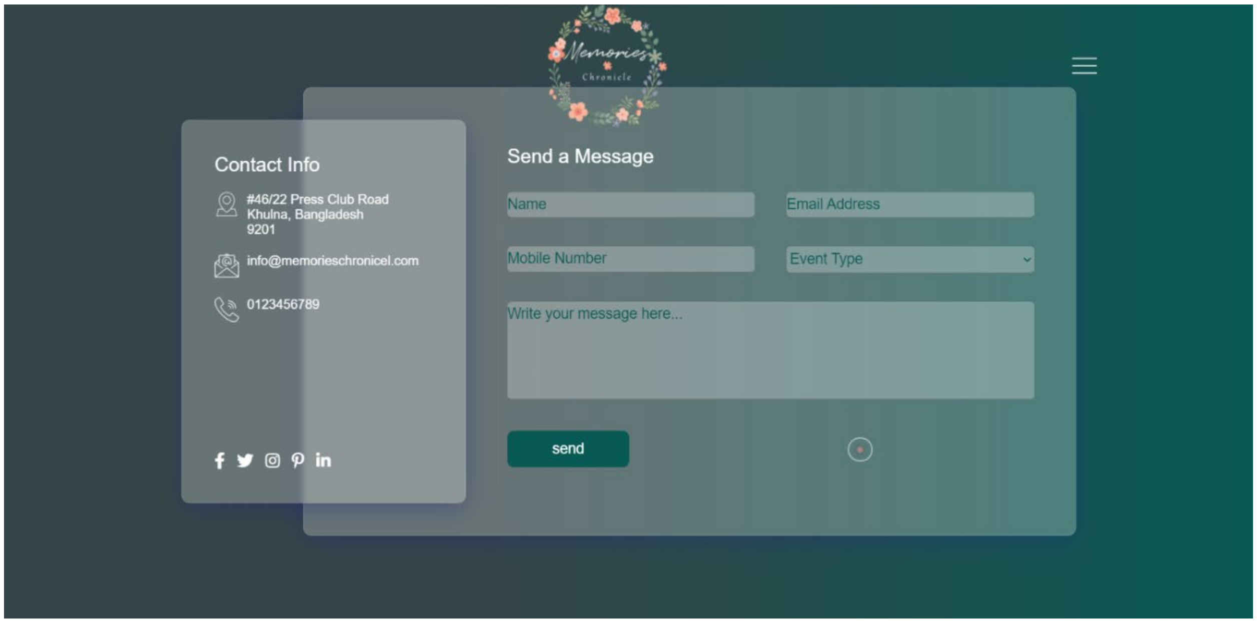
And if the message sent successfully the can show a successful message in the alert box as well.



localhost says

Message sent Successfully

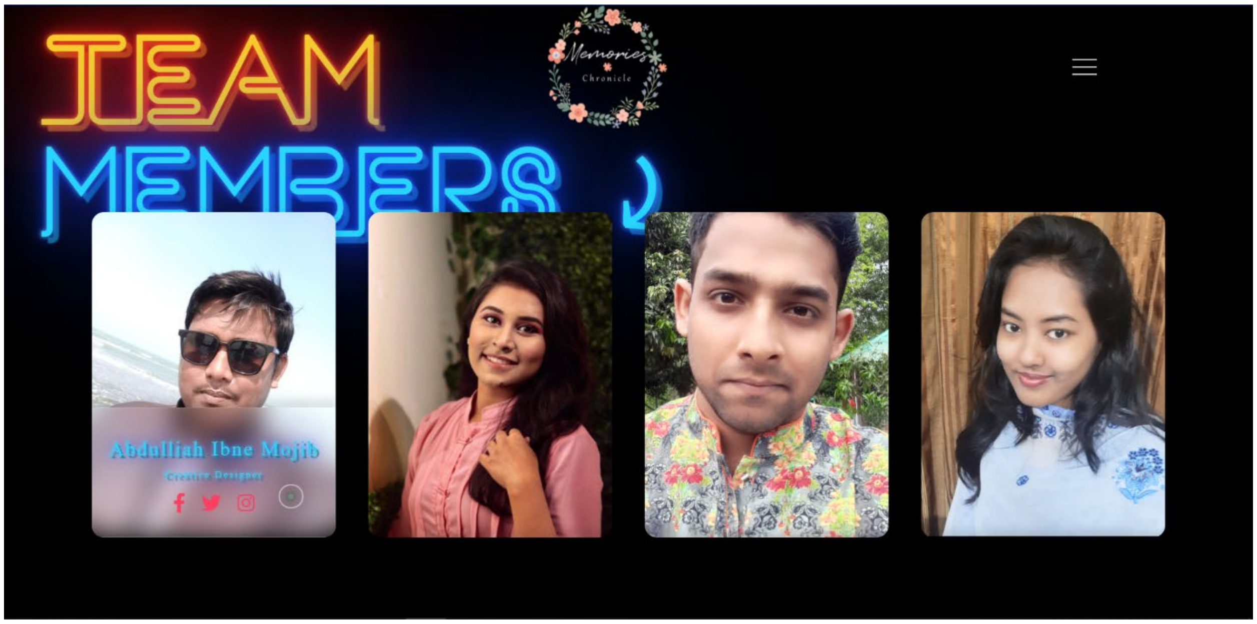
But to send message it's necessary to have an account in the website. Visitors cant sent any messages.



contact.php

12. Team Member :

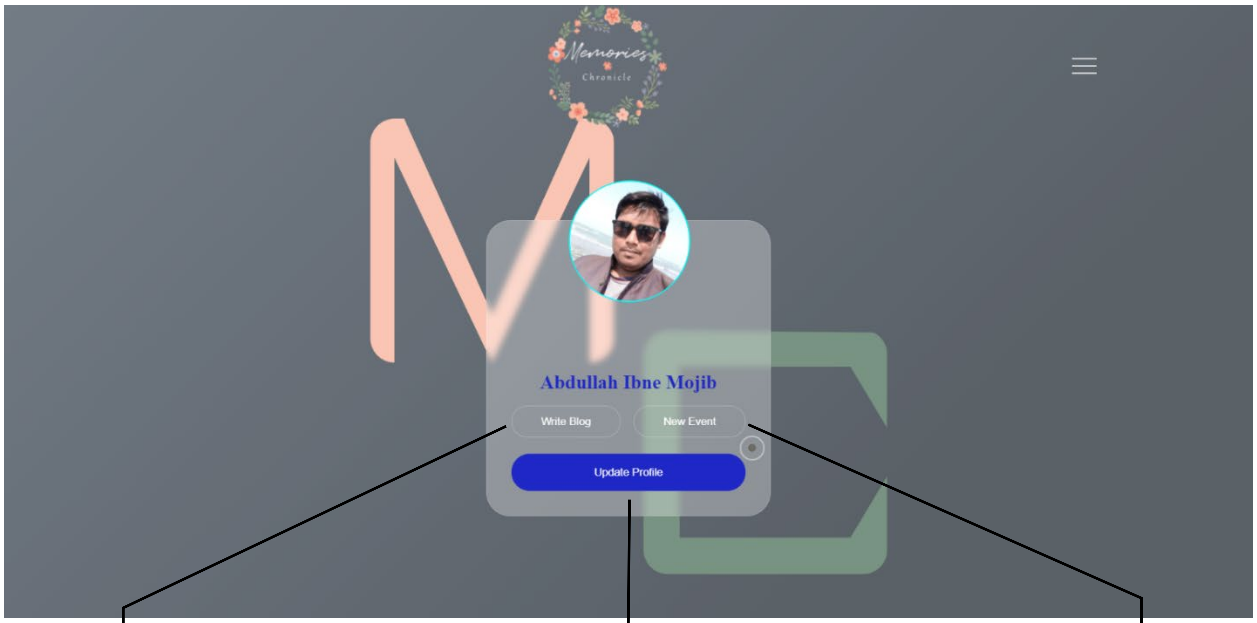
The page team member or team.php shows the information of people who develop this website. It’s a card view and on the hover effect the information appears. There is also a neon text effect which is created by CSS. Just like services section.



Team.php

13. Profile :

Profile page also comes with the same **“login required”** features like the contact us page. User must have account to go to his/her profile page. We implement the same session variable concept like we discuss before in the contact section. The profile.php page checks the same condition before the page load and if a user doesn’t have any account it will automatically redirect to the login and register page along with the same alert box so that they can create one and come back to the page. If a user already logged in then he/she can access the profile page by clicking profile option in the menu section.



This button redirect user to the writeblog.php page where is a form to write a blog post for the blog page in the website. Those posts are visible to all visitors, admins and users.

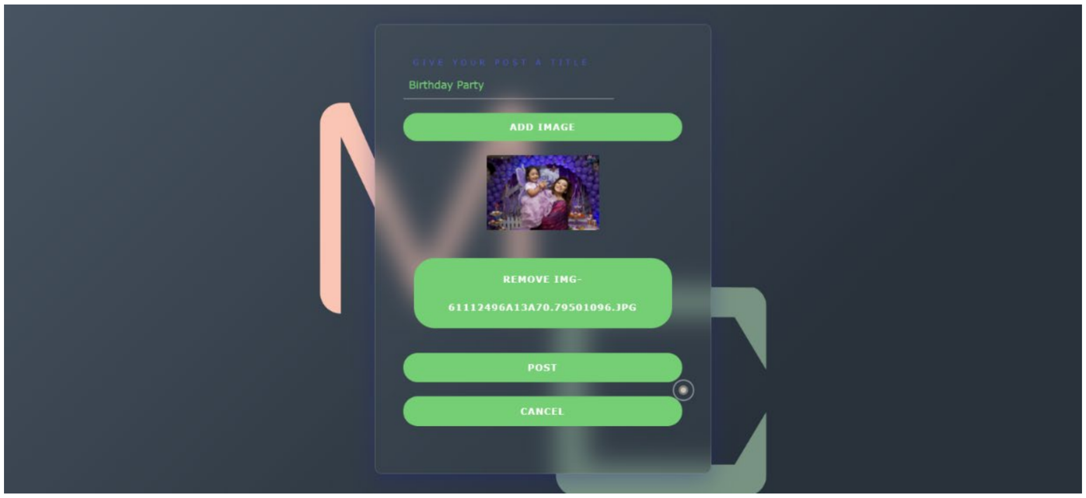
This button redirect users to the contact.php page so that they can sent message request for a new event.

This button redirect users to the update profile.php page there they got a form if they change any information of their profile they can do it from here. Here we also use GET method to send & retrieve user id for the update query.

Profile.php

13.1. Write Blog:

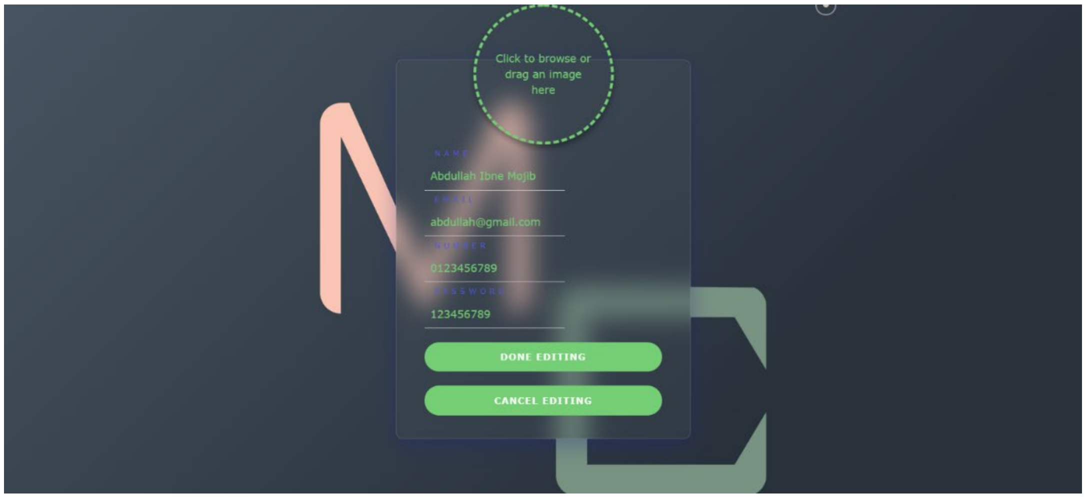
The Writeblog.php has a form that takes a title for the post and an image prom user after the post button press it execute an INSERT query to insert the blog data along with the date time and user name in the database.


GIVE YOUR POST A TITLE
Birthday Party

writeblog.php

13.2. Update Profile :

Update profile.php page get a user id from the profile page when the user press on the update profile button. Then it shows data in a form according to the user id to change if the user wants. If the user change information and press save button the data will updated in the database table using Update query.

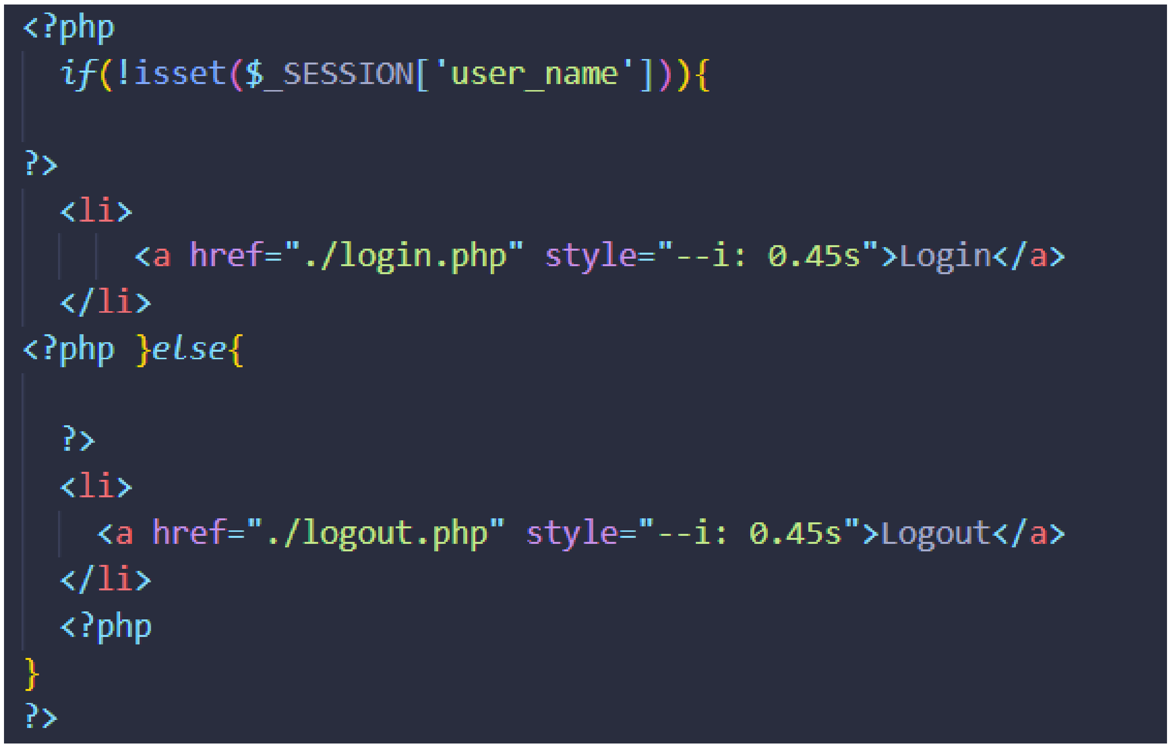


NAME
Abdullah Ibne Mojib
EMAIL
abdullah@gmail.com
NUMBER
0123456789
PASSWORD
123456789

Updateprofile.php

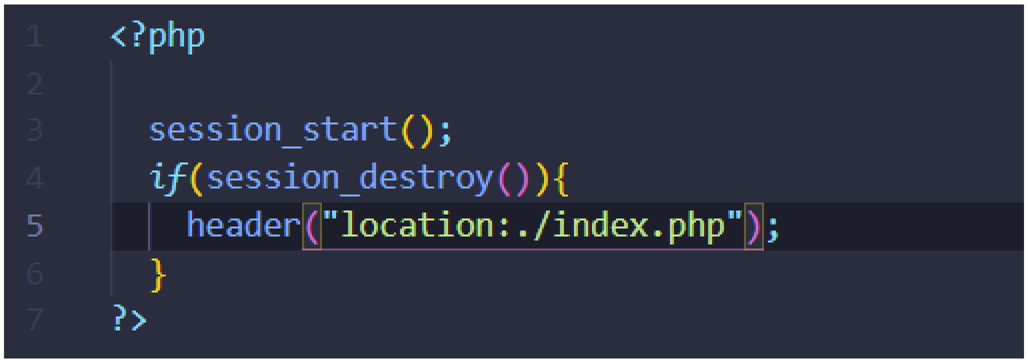
14. Login or Logout & Register :

In the menu item login and logout option dynamically appears. When a user is not logged in or a visitor visited the website they see a login option in the menu item if a user logged in he saw a log out option instate of a login option. The login option redirect to the login.php page and the logout option redirects to the logout.php. Appearance of different option is achieved by checking session variable called user_name.



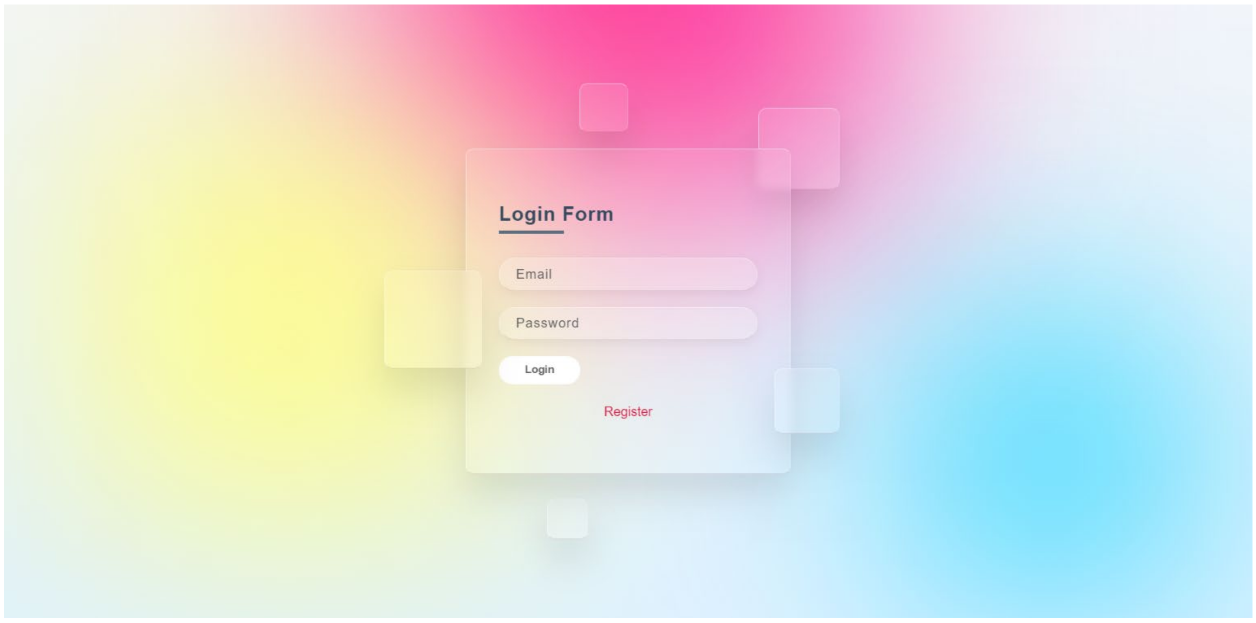
```php
<?php
  if(!isset($_SESSION['user_name'])){

?>
  <li>
      <a href="./login.php" style="--i: 0.45s">Login</a>
  </li>
<?php }else{

  ?>
  <li>
    <a href="./logout.php" style="--i: 0.45s">Logout</a>
  </li>
  <?php
}
?>
```

If a logged in user press the logout option the session variable will destroy and he will redirect to the home page.



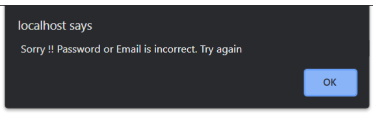
```php
<?php

  session_start();
  if(session_destroy()){
    header("location:./index.php");
  }
?>
```

If a visitor or user choose to login he will redirected to the login.php. In login.php page there is a login form and also a register form to register new user. Bellow the login button there is a button called “**register**”. Which is use to toggle between login form and a register form. If an existing user wants to login he/she must be input their email and password in the input field and then press the login button.



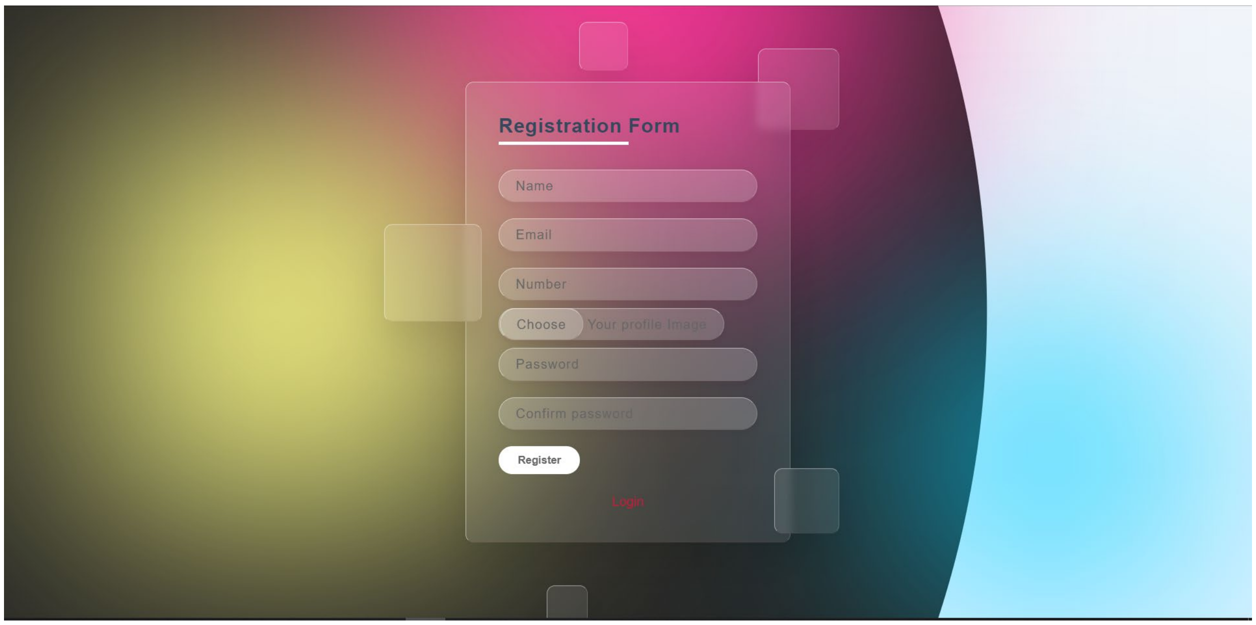
login.php (login form)

If the user uses wrong credentials, an alert box will pop up and shows that **“Sorry!! Password or email is incorrect. Try again”.**


localhost says

Sorry !! Password or Email is incorrect. Try again

If the visitor wants to register. He just need to press the Register button bellow the login button. Which will toggle the form to register form.



login.php (registration form)

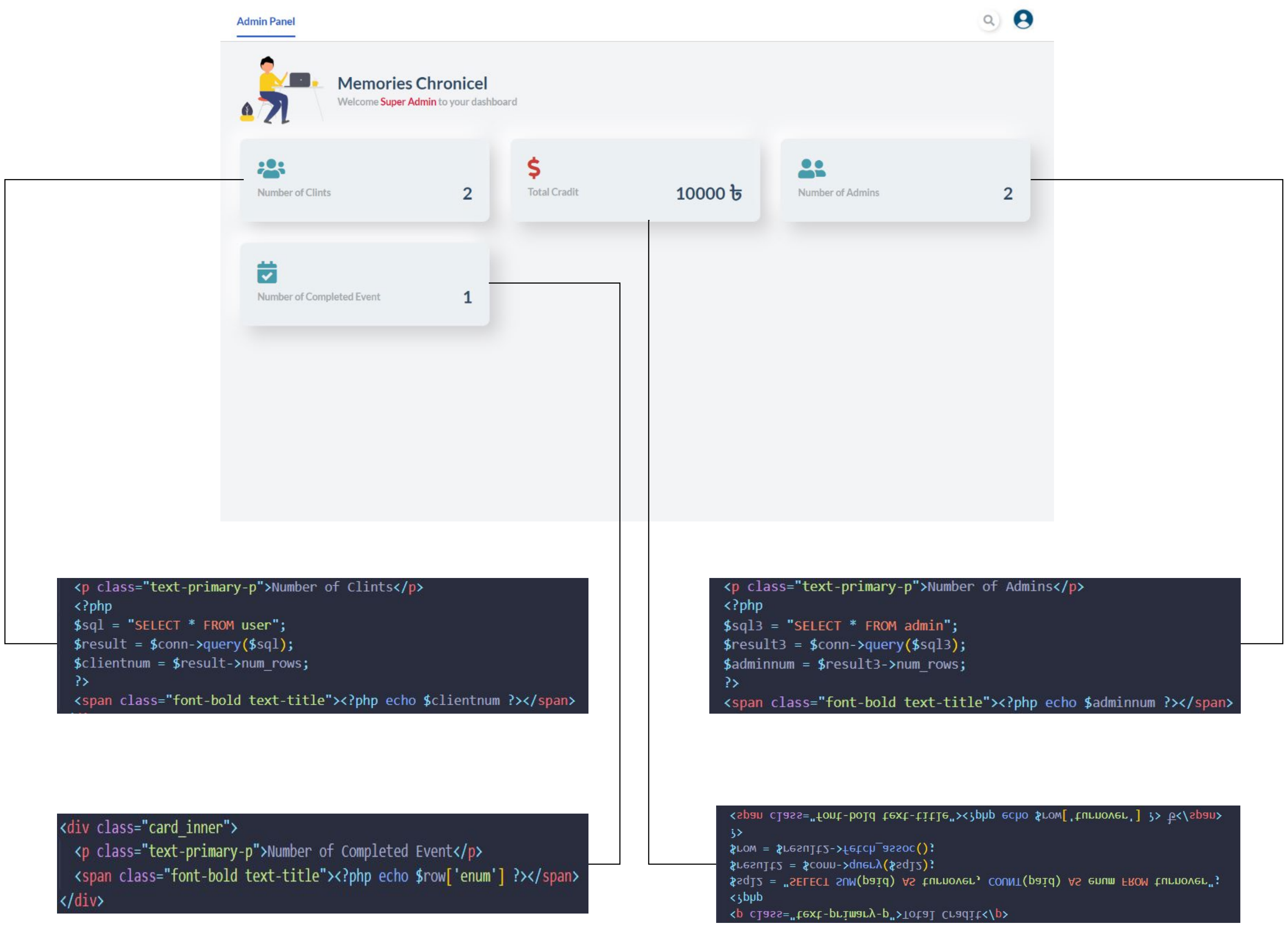
After fill up all the necessary information in the registration form and after press the register button the form data will send to registrationcheck.php page and execute a Insert query to insert all the data in the database. After that a new user can login into the website. And for the login credential check login form data go through the logincheck.php. Which consist of MySQL query to fetch data and check according to given data.

Important Code Snippets & Description:

Important code snippets discuss about the logical codes behind some important features that the website offers.

I. Cards on admin dashboard:

There are 4 cards on the admin dashboard page. All the card value is dynamically updated by running some MySQL query with some aggregate function.



```
<p class="text-primary-p">Number of Clints</p>
<?php
$sql = "SELECT * FROM user";
$result = $conn->query($sql);
$clientnum = $result->num_rows;
?>
<span class="font-bold text-title"><?php echo $clientnum ?></span>
```

```
<p class="text-primary-p">Number of Admins</p>
<?php
$sql3 = "SELECT * FROM admin";
$result3 = $conn->query($sql3);
$adminnum = $result3->num_rows;
?>
<span class="font-bold text-title"><?php echo $adminnum ?></span>
```

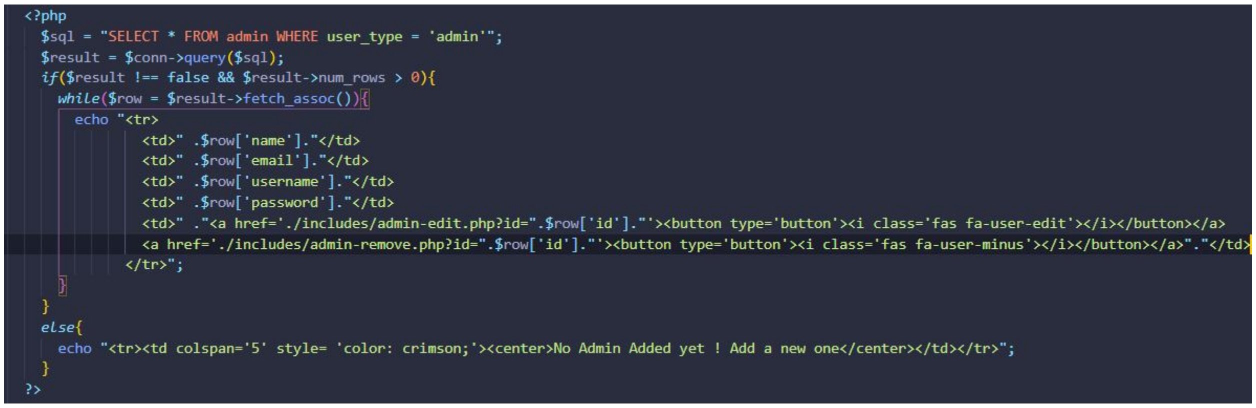
```
<div class="card_inner">
  <p class="text-primary-p">Number of Completed Event</p>
  <span class="font-bold text-title"><?php echo $row['enum'] ?></span>
</div>
```

```
<p class="text-primary-p">Total Cradit</p>
<?php
$sql2 = "SELECT SUM(paid) AS turnover, COUNT(paid) AS enum FROM turnover";
$result2 = $conn->query($sql2);
$row = $result2->fetch_assoc();
?>
<span class="font-bold text-title"><?php echo $row['turnover'] ?> ৳</span>
```

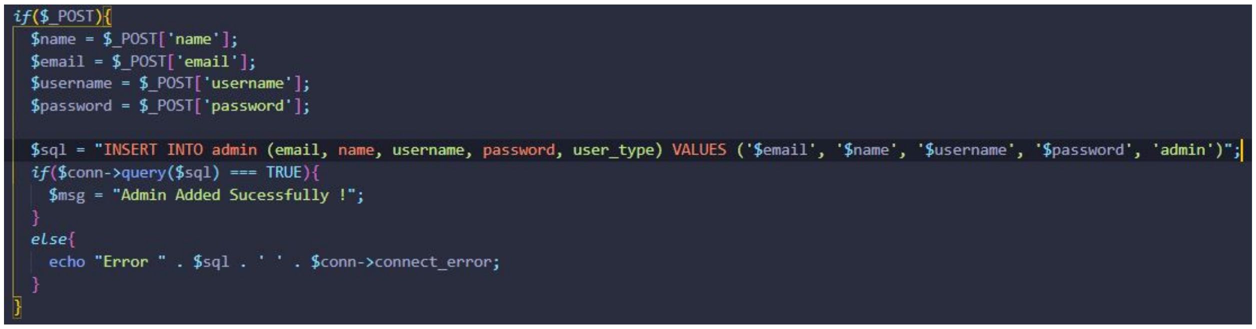
Code for Admin Dashboard card

II. Code for admin Management:

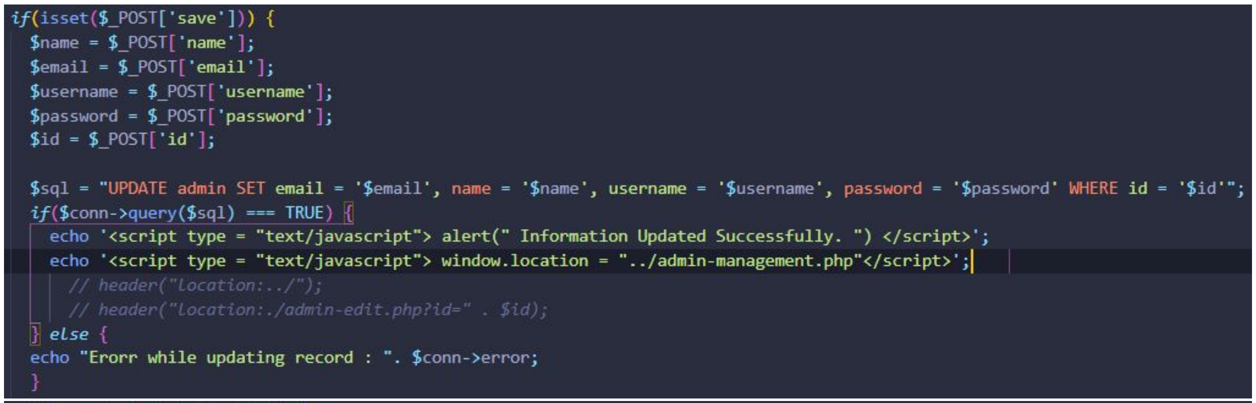
There are 4 operations in admin management Show, ADD, EDIT & DELETE. These operations achieved by 4 basic MySQL operation Select, Insert, Update & Delete.



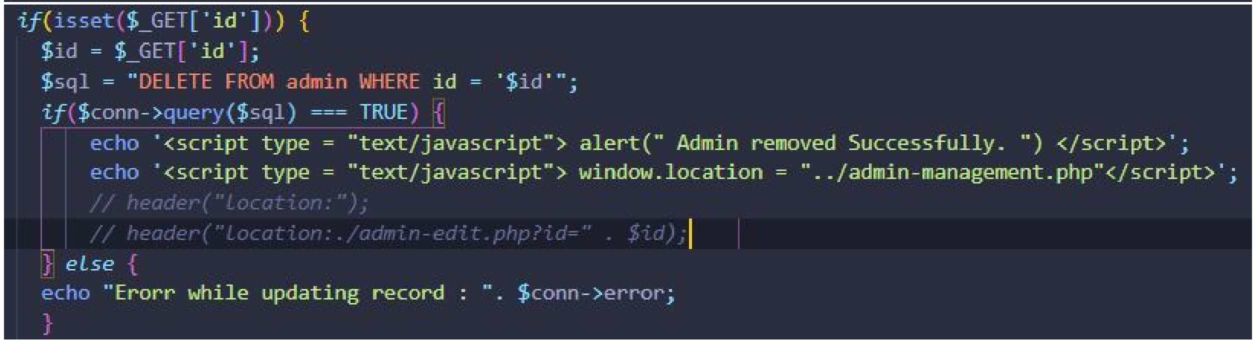
```php
<?php
  $sql = "SELECT * FROM admin WHERE user_type = 'admin'";
  $result = $conn->query($sql);
  if($result !== false && $result->num_rows > 0){
    while($row = $result->fetch_assoc()){
      echo "<tr>
              <td>" .$row['name']."</td>
              <td>" .$row['email']."</td>
              <td>" .$row['username']."</td>
              <td>" .$row['password']."</td>
              <td>" ."<a href='./includes/admin-edit.php?id=".$row['id']."'><button type='button'><i class='fas fa-user-edit'></i></button></a>
              <a href='./includes/admin-remove.php?id=".$row['id']."'><button type='button'><i class='fas fa-user-minus'></i></button></a>"."</td>
            </tr>";
    }
  }
  else{
    echo "<tr><td colspan='5' style= 'color: crimson;'><center>No Admin Added yet ! Add a new one</center></td></tr>";
  }
?>
```



```php
if($_POST){
  $name = $_POST['name'];
  $email = $_POST['email'];
  $username = $_POST['username'];
  $password = $_POST['password'];

  $sql = "INSERT INTO admin (email, name, username, password, user_type) VALUES ('$email', '$name', '$username', '$password', 'admin')";
  if($conn->query($sql) === TRUE){
    $msg = "Admin Added Sucessfully !";
  }
  else{
    echo "Error " . $sql . ' ' . $conn->connect_error;
  }
}
```



```php
if(isset($_POST['save'])) {
  $name = $_POST['name'];
  $email = $_POST['email'];
  $username = $_POST['username'];
  $password = $_POST['password'];
  $id = $_POST['id'];

  $sql = "UPDATE admin SET email = '$email', name = '$name', username = '$username', password = '$password' WHERE id = '$id'";
  if($conn->query($sql) === TRUE) {
    echo '<script type = "text/javascript"> alert(" Information Updated Successfully. ") </script>';
    echo '<script type = "text/javascript"> window.location = "../admin-management.php"</script>';
      // header("location:../");
      // header("location:./admin-edit.php?id=" . $id);
  } else {
  echo "Erorr while updating record : ". $conn->error;
  }
```

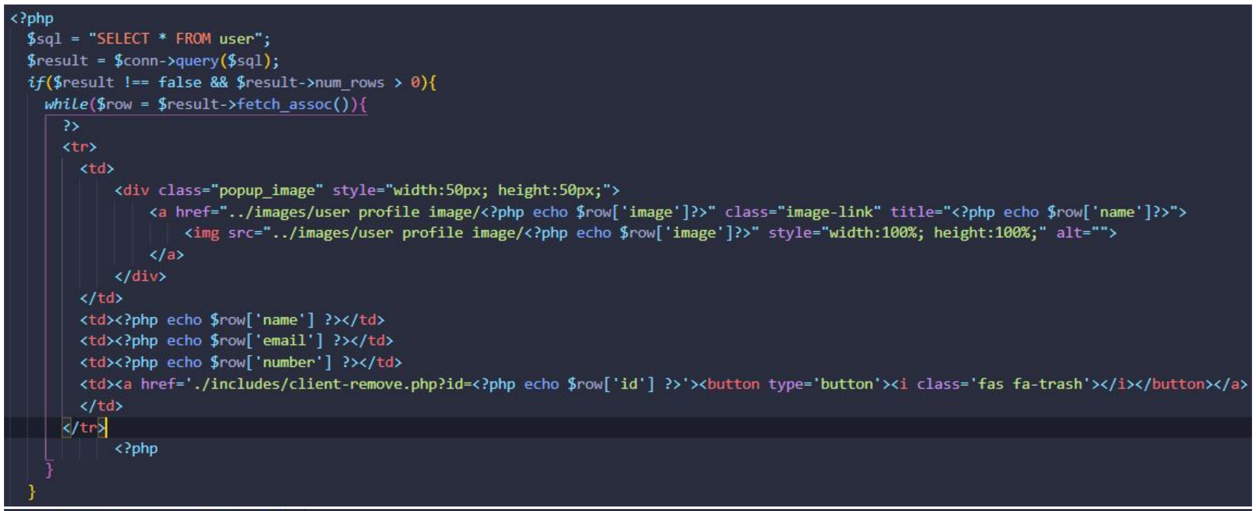


```php
if(isset($_GET['id'])) {
  $id = $_GET['id'];
  $sql = "DELETE FROM admin WHERE id = '$id'";
  if($conn->query($sql) === TRUE) {
      echo '<script type = "text/javascript"> alert(" Admin removed Successfully. ") </script>';
      echo '<script type = "text/javascript"> window.location = "../admin-management.php"</script>';
      // header("location:");
      // header("location:./admin-edit.php?id=" . $id);
  } else {
  echo "Erorr while updating record : ". $conn->error;
  }
```

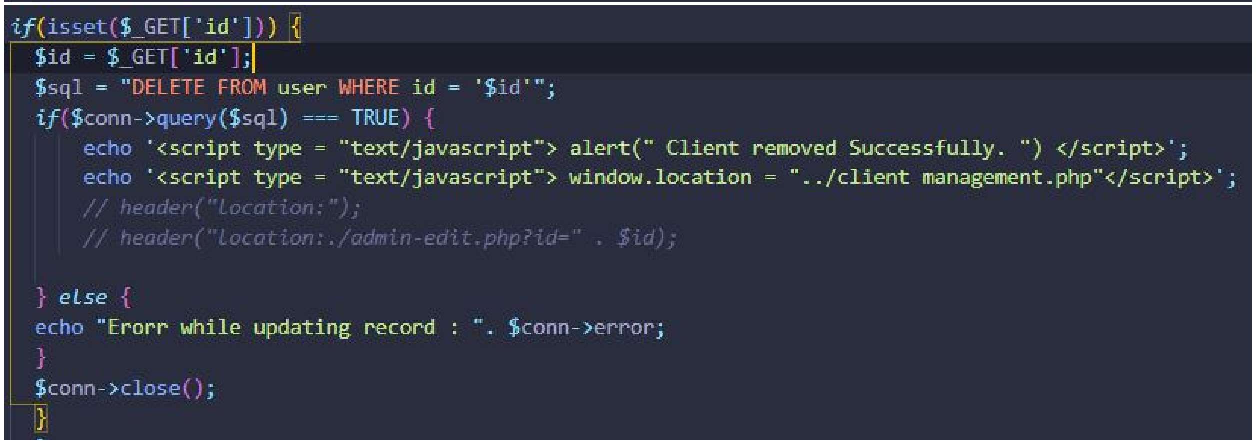
The 4 operations behind admin management

III. Code for client management:

In client management there is only two operation and that is SELECT & DELETE. Which also created by MySQL query.



```
<?php
  $sql = "SELECT * FROM user";
  $result = $conn->query($sql);
  if($result !== false && $result->num_rows > 0){
    while($row = $result->fetch_assoc()){
      ?>
      <tr>
        <td>
            <div class="popup_image" style="width:50px; height:50px;">
                <a href="../images/user profile image/<?php echo $row['image']?>" class="image-link" title="<?php echo $row['name']?>">
                    <img src="../images/user profile image/<?php echo $row['image']?>" style="width:100%; height:100%;" alt="">
                </a>
            </div>
        </td>
        <td><?php echo $row['name'] ?></td>
        <td><?php echo $row['email'] ?></td>
        <td><?php echo $row['number'] ?></td>
        <td><a href='./includes/client-remove.php?id=<?php echo $row['id'] ?>'><button type='button'><i class='fas fa-trash'></i></button></a>
        </td>
      </tr>
            <?php
    }
  }
```



```
if(isset($_GET['id'])) {
  $id = $_GET['id'];
  $sql = "DELETE FROM user WHERE id = '$id'";
  if($conn->query($sql) === TRUE) {
      echo '<script type = "text/javascript"> alert(" Client removed Successfully. ") </script>';
      echo '<script type = "text/javascript"> window.location = "../client management.php"</script>';
      // header("Location:");
      // header("Location:./admin-edit.php?id=" . $id);

  } else {
  echo "Erorr while updating record : ". $conn->error;
  }
  $conn->close();
}
```

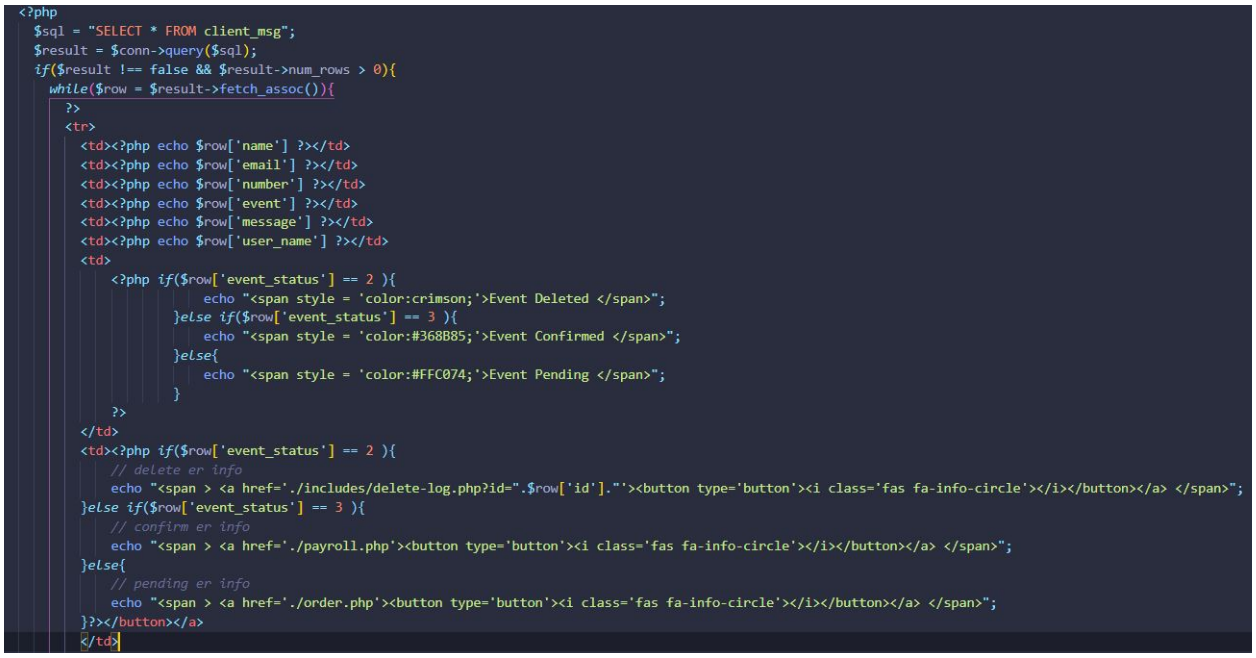
Client list show and delete operation from client management

IV. Code for Client blog management:

The logical codes are same as Client management discuss above just this time the table name is different. The table name is ***'user_blog'*** instate of ***'user'*** in the MySQL query.

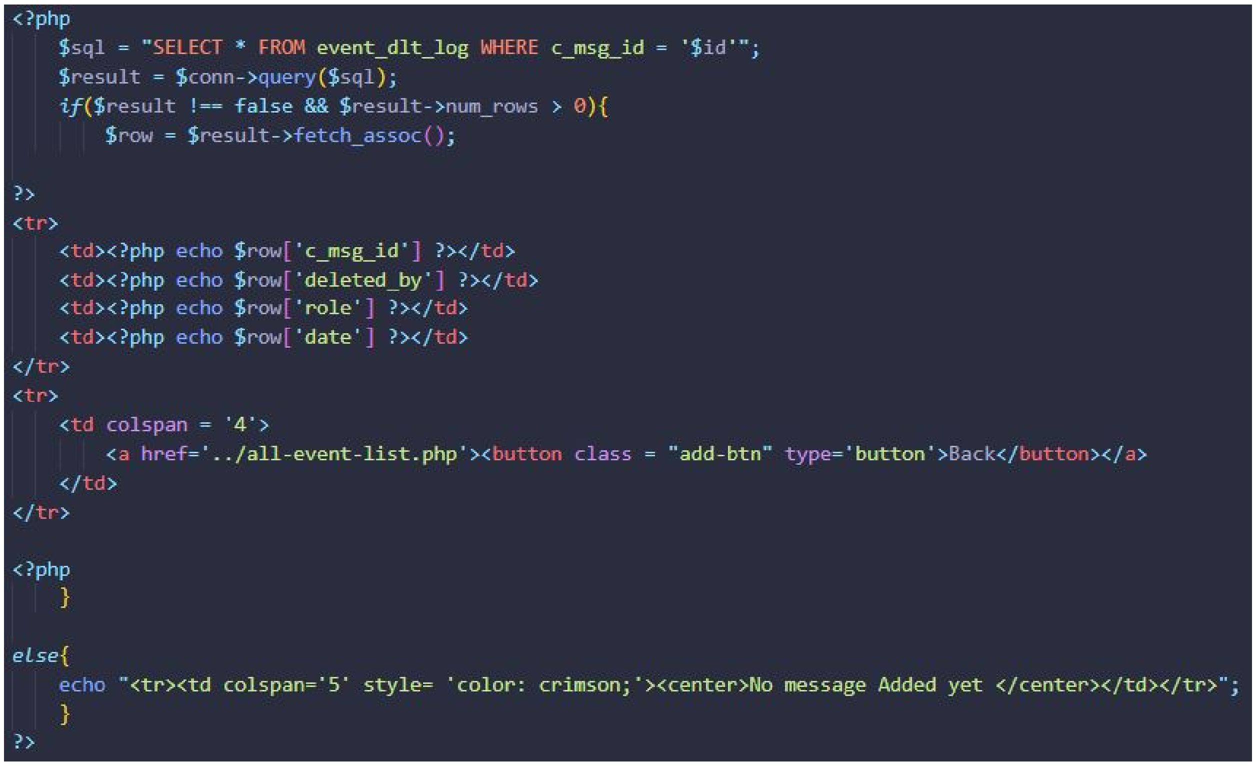
V. Code for all event list:

Since this page shows the status of events. We control the status by checking the active status in the database table. Also here we showcase the event deleted by whom and when in a separate page. And the other status linked to the other pages like client messages and payroll page.



```php
<?php
  $sql = "SELECT * FROM client_msg";
  $result = $conn->query($sql);
  if($result !== false && $result->num_rows > 0){
    while($row = $result->fetch_assoc()){
      ?>
      <tr>
        <td><?php echo $row['name'] ?></td>
        <td><?php echo $row['email'] ?></td>
        <td><?php echo $row['number'] ?></td>
        <td><?php echo $row['event'] ?></td>
        <td><?php echo $row['message'] ?></td>
        <td><?php echo $row['user_name'] ?></td>
        <td>
            <?php if($row['event_status'] == 2 ){
                        echo "<span style = 'color:crimson;'>Event Deleted </span>";
                    }else if($row['event_status'] == 3 ){
                        echo "<span style = 'color:#368B85;'>Event Confirmed </span>";
                    }else{
                        echo "<span style = 'color:#FFC074;'>Event Pending </span>";
                    }
            ?>
        </td>
        <td><?php if($row['event_status'] == 2 ){
            // delete er info
            echo "<span > <a href='./includes/delete-log.php?id=".$row['id']."'><button type='button'><i class='fas fa-info-circle'></i></button></a> </span>";
        }else if($row['event_status'] == 3 ){
            // confirm er info
            echo "<span > <a href='./payroll.php'><button type='button'><i class='fas fa-info-circle'></i></button></a> </span>";
        }else{
            // pending er info
            echo "<span > <a href='./order.php'><button type='button'><i class='fas fa-info-circle'></i></button></a> </span>";
        }?></button></a>
        </td>
```

List of events with status and a i button

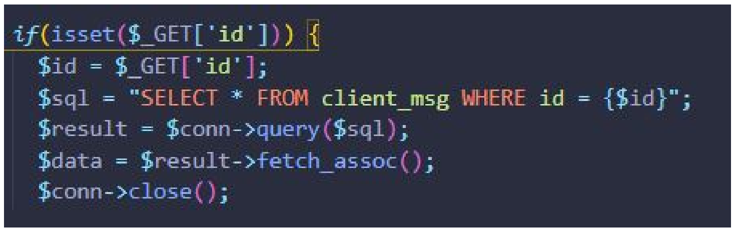


```php
<?php
    $sql = "SELECT * FROM event_dlt_log WHERE c_msg_id = '$id'";
    $result = $conn->query($sql);
    if($result !== false && $result->num_rows > 0){
        $row = $result->fetch_assoc();

?>
<tr>
    <td><?php echo $row['c_msg_id'] ?></td>
    <td><?php echo $row['deleted_by'] ?></td>
    <td><?php echo $row['role'] ?></td>
    <td><?php echo $row['date'] ?></td>
</tr>
<tr>
    <td colspan = '4'>
        <a href='../all-event-list.php'><button class = "add-btn" type='button'>Back</button></a>
    </td>
</tr>

<?php
    }

else{
    echo "<tr><td colspan='5' style= 'color: crimson;'><center>No message Added yet </center></td></tr>";
    }
?>
```

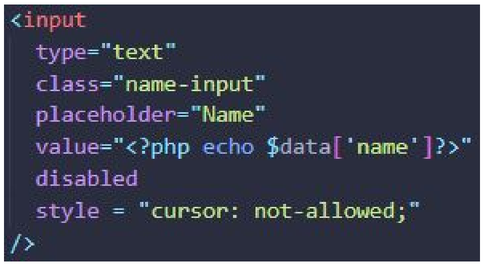
Deleted event's log table

VI. Event confirmation page:

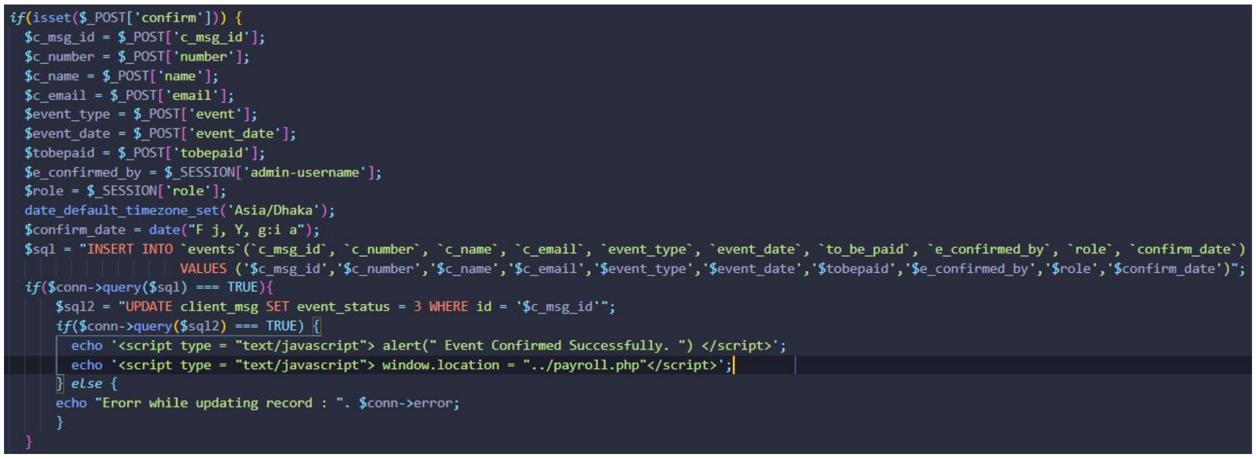
In this page there is a form with some input element is **disabled**. Cause those information was given by the user. Admin can only add the event date and payment amount.



```php
if(isset($_GET['id'])) {
  $id = $_GET['id'];
  $sql = "SELECT * FROM client_msg WHERE id = {$id}";
  $result = $conn->query($sql);
  $data = $result->fetch_assoc();
  $conn->close();
```



```html
<input
  type="text"
  class="name-input"
  placeholder="Name"
  value="<?php echo $data['name']?>"
  disabled
  style = "cursor: not-allowed;"
/>
```



```php
if(isset($_POST['confirm'])) {
  $c_msg_id = $_POST['c_msg_id'];
  $c_number = $_POST['number'];
  $c_name = $_POST['name'];
  $c_email = $_POST['email'];
  $event_type = $_POST['event'];
  $event_date = $_POST['event_date'];
  $tobepaid = $_POST['tobepaid'];
  $e_confirmed_by = $_SESSION['admin-username'];
  $role = $_SESSION['role'];
  date_default_timezone_set('Asia/Dhaka');
  $confirm_date = date("F j, Y, g:i a");
  $sql = "INSERT INTO `events`(`c_msg_id`, `c_number`, `c_name`, `c_email`, `event_type`, `event_date`, `to_be_paid`, `e_confirmed_by`, `role`, `confirm_date`)
                        VALUES ('$c_msg_id','$c_number','$c_name','$c_email','$event_type','$event_date','$tobepaid','$e_confirmed_by','$role','$confirm_date')";
  if($conn->query($sql) === TRUE){
      $sql2 = "UPDATE client_msg SET event_status = 3 WHERE id = '$c_msg_id'";
      if($conn->query($sql2) === TRUE) {
        echo '<script type = "text/javascript"> alert(" Event Confirmed Successfully. ") </script>';
        echo '<script type = "text/javascript"> window.location = "../payroll.php"</script>';
      } else {
      echo "Erorr while updating record : ". $conn->error;
      }
  }
```

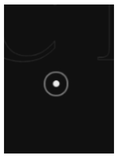
Event confirmation process

VII. Home Page:

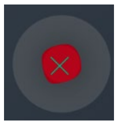
Home page consist of some animation and some transition between pages all achieved by JS.

7.01.Cursor:

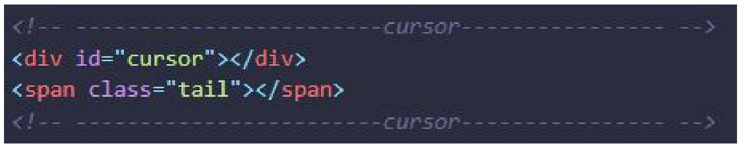
There is a custom cursor that gives the website a modern look. It is achieved by HTML class, CSS & JS.



This courser has a hover effect when it goes on any anchor tag then it shows a breathing effect.

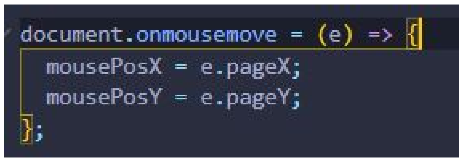


The JavaScript part of this effect is given bellow,



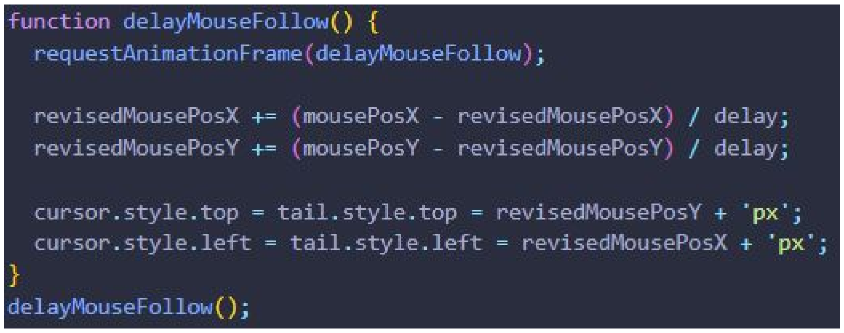
```
<!-- ------------------------cursor---------------- -->
<div id="cursor"></div>
<span class="tail"></span>
<!-- ------------------------cursor---------------- -->
```

HTML part for cursor



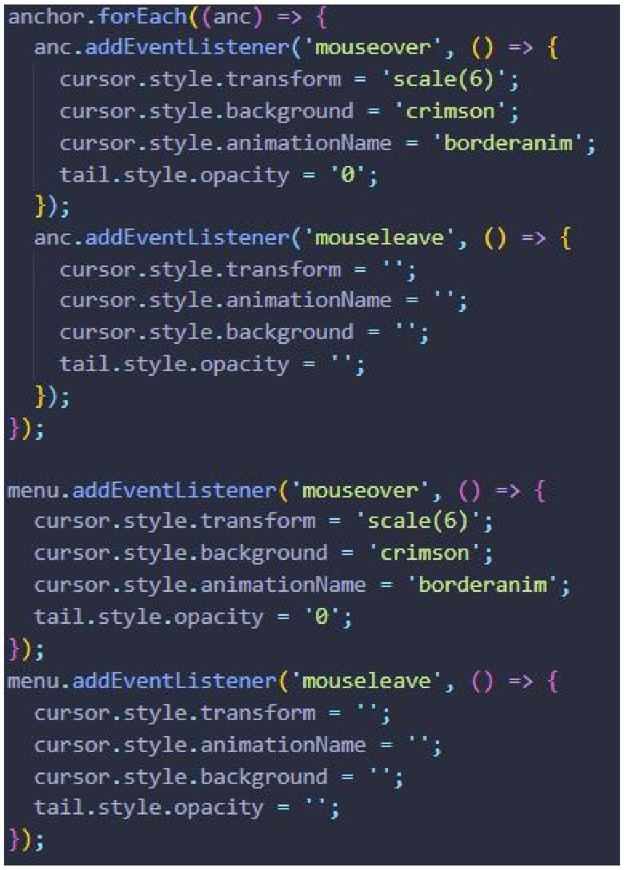
```
document.onmousemove = (e) => {
  mousePosX = e.pageX;
  mousePosY = e.pageY;
};
```

Track cursor



```
function delayMouseFollow() {
  requestAnimationFrame(delayMouseFollow);

  revisedMousePosX += (mousePosX - revisedMousePosX) / delay;
  revisedMousePosY += (mousePosY - revisedMousePosY) / delay;

  cursor.style.top = tail.style.top = revisedMousePosY + 'px';
  cursor.style.left = tail.style.left = revisedMousePosX + 'px';
}
delayMouseFollow();
```

Make cursor movement smooth by adding delay

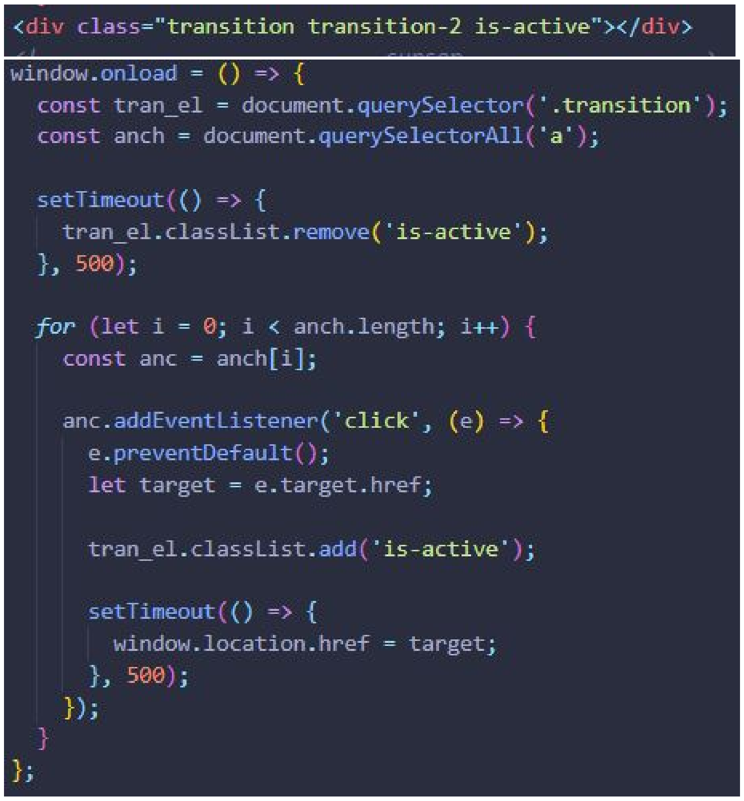


```
anchor.forEach((anc) => {
  anc.addEventListener('mouseover', () => {
    cursor.style.transform = 'scale(6)';
    cursor.style.background = 'crimson';
    cursor.style.animationName = 'borderanim';
    tail.style.opacity = '0';
  });
  anc.addEventListener('mouseleave', () => {
    cursor.style.transform = '';
    cursor.style.animationName = '';
    cursor.style.background = '';
    tail.style.opacity = '';
  });
});

menu.addEventListener('mouseover', () => {
  cursor.style.transform = 'scale(6)';
  cursor.style.background = 'crimson';
  cursor.style.animationName = 'borderanim';
  tail.style.opacity = '0';
});
menu.addEventListener('mouseleave', () => {
  cursor.style.transform = '';
  cursor.style.animationName = '';
  cursor.style.background = '';
  tail.style.opacity = '';
});
```

Add styling on hover on anchor tag

7.02.Transition:

There is a transition between page changes and this is achieved by using **onload()** method, HTML class & CSS.



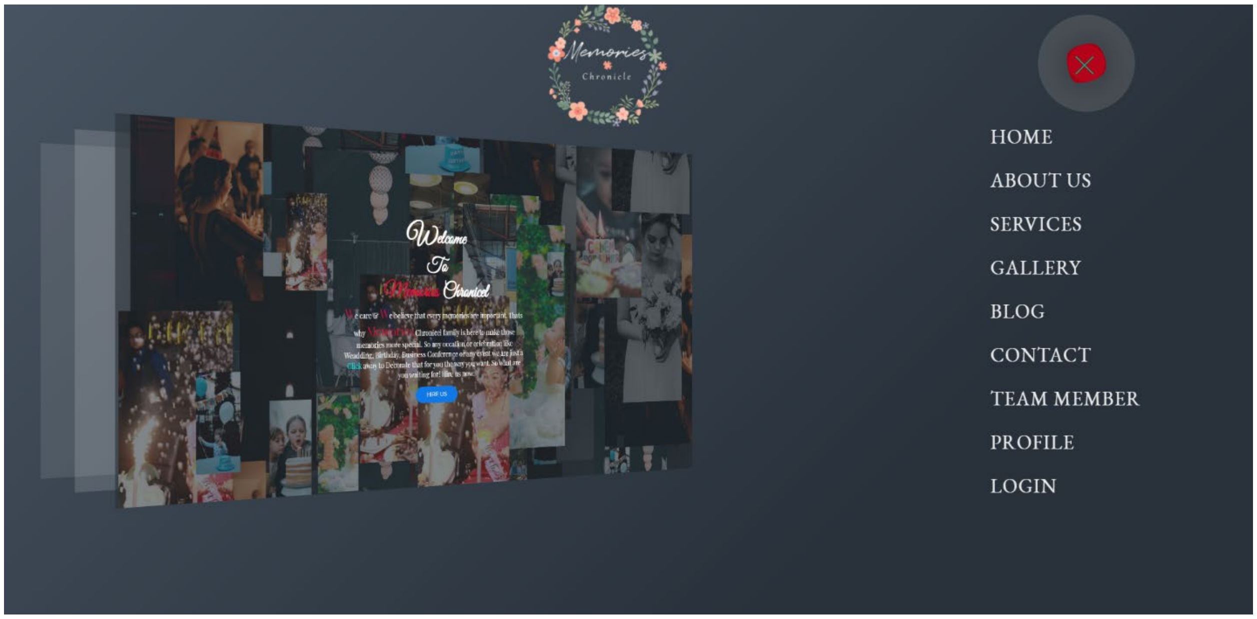
```
<div class="transition transition-2 is-active"></div>
```

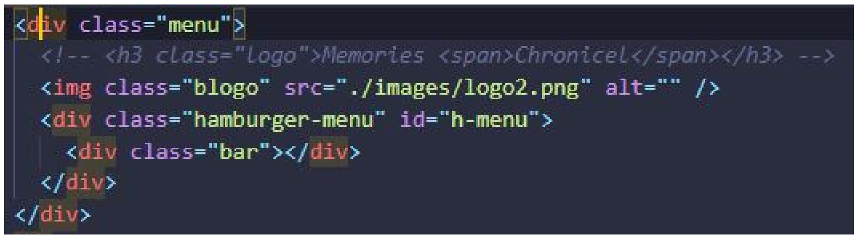
```
window.onload = () => {
  const tran_el = document.querySelector('.transition');
  const anch = document.querySelectorAll('a');

  setTimeout(() => {
    tran_el.classList.remove('is-active');
  }, 500);

  for (let i = 0; i < anch.length; i++) {
    const anc = anch[i];

    anc.addEventListener('click', (e) => {
      e.preventDefault();
      let target = e.target.href;

      tran_el.classList.add('is-active');

      setTimeout(() => {
        window.location.href = target;
      }, 500);
    });
  }
};
```

HTML class & JS for page load transition

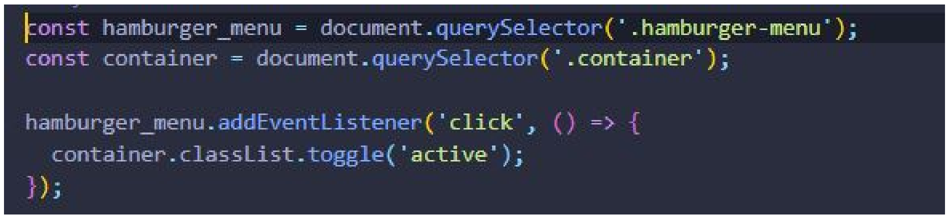
7.03.Menu:

We use a perspective hamburger menu with the help of CSS properties, HTML elements & JS.





```
<div class="menu">
  <!-- <h3 class="logo">Memories <span>Chronicel</span></h3> -->
  <img class="blogo" src="./images/logo2.png" alt="" />
  <div class="hamburger-menu" id="h-menu">
    <div class="bar"></div>
  </div>
</div>
```

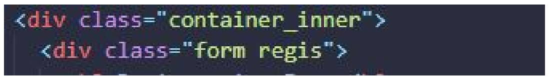


```
const hamburger_menu = document.querySelector('.hamburger-menu');
const container = document.querySelector('.container');

hamburger_menu.addEventListener('click', () => {
  container.classList.toggle('active');
});
```

Perspective Hamburger menu

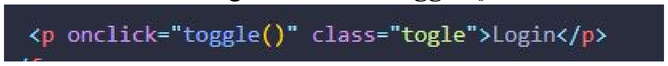
VIII. Login & Registration page switching transition:

On the login page there is a transition between 2 form switching. This transition also created by using HTML classes, CSS & JS.

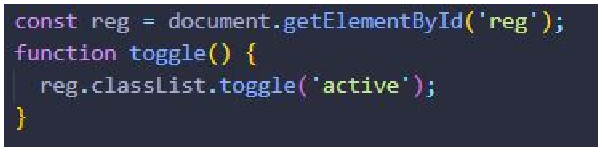


```
<div class="container_inner">
  <div class="form regis">
```

This div class changes when the **toggle ()** method call.



```
  <p onclick="toggle()" class="togle">Login</p>
```



```
const reg = document.getElementById('reg');
function toggle() {
  reg.classList.toggle('active');
}
```

▪ Development Tools:

Software Requirement		
Type	Tools Name	Availability
Operating System	Windows 10 (64bit)	✓
Language	HTML, CSS, PHP, js, SCSS	✓
Server	Apache	✓
Database	MySQL	✓
IDE	VS Code, Sublime text	✓

▪ Source Code:

CLICK HERE FOR CODE