



CSE370: Database Systems
Project Report
Project Title: Unity Wave
Github Repository link

Group No: 06, CSE370 Lab Section:14 , Spring 2024		
ID	Name	Contribution
22101114	Labiba Zahin	<p>Frontend:</p> <ul style="list-style-type: none">• Home (index.php)• Donor Home page(DONOR.php)• Donor sign in/Donation Donation(donar.php)• Delete account (remove_donor.php) <p>Backend:</p> <ul style="list-style-type: none">• User registration/ Inserting new Donor (insert_donor.php)• User login (donor_login.php)• User logout ((donor_logout.php)• Showing donor item types(donar.php)• Deletion of Donor account (donor_deleting.php)

22101117	Shawana Maliha	<p>Frontend:</p> <ul style="list-style-type: none"> • Item input and add(add_item.php) • Donor registration process (donor_register.php) • Organization claim for item page(claim.php) <p>Backend:</p> <ul style="list-style-type: none"> • Approving Donor requests (donor_process_request.php) • Inserting quantity and update items list(ad_insert_item.php) • Updating item quantity after donation(insert_item.php) • Removes item quantity with claim of organization(claim_item.php)
22101249	Zarin Tasnim	<p>Frontend:</p> <ul style="list-style-type: none"> • Event list created by admin (show_events_admin.php) • Event list for donor and organization(only_event.php) <p>Backend:</p> <ul style="list-style-type: none"> • Editing event(edit_event.php) • Deleting Event(delete_event.php) • Update_event.php • Insertion of new events(insert.php)
22101343	Tasnim Zaman	<p>Frontend:</p> <ul style="list-style-type: none"> • Organization home page (org_index.php) • Organization login page (organization_login.php) • Organization register page (organization_register.php) • Showing available event table and selecting event page (org_event.php) • Deletion page (org_delete.php) <p>Backend:</p> <ul style="list-style-type: none"> • Organization signing in (signin.php)

		<ul style="list-style-type: none"> ● New organization inserting in the table (insert_organization.php) ● Updating the event attribute in organization table (update_org_event.php) ● Deleting the account (org_deleting)
22101335	Hasan Sarwar Zami	<p><u>Frontend:</u></p> <ul style="list-style-type: none"> ● Admin Dashboard(admin_dashboard.php) ● Organization Request Table(donor_requests.php) ● Donor Request Table(organization_requests.php) <p><u>Backend:</u></p> <ul style="list-style-type: none"> ● Admin Login Panel(admin_login.php) ● Admin Logout (admin_logout.php) ● Approving Organization requests (organization_process_request.php) ● Adding new Events (add_event.php)

Table of Contents

Section No	Content	Page No
1	Introduction	5
2	Project Features	6
3	ER/EER Diagram	7
4	Schema Diagram	8
5	Frontend Development	9-17
6	Backend Development	18-34
7	Source Code Repository	35
8	Conclusion	35
9	References	

Introduction

Welcome to "Unity Wave" – a purpose-driven donation platform designed to empower communities and foster goodwill through charitable contributions. Here, we bring together individuals, organizations, and volunteers in a unified effort to make a positive impact.

Our platform provides a user-friendly interface that allows individuals to register effortlessly and contribute to our community initiatives. Donors can easily specify the types of items they wish to donate along with the quantities, which are then added to our item inventory. Similarly, different organizations can easily specify the types of items they wish to claim for their specific organizations along with the quantities, which are then deleted from our item inventory. Additionally, organizations have the opportunity to join any upcoming events, thus their involvement further enriches our collective efforts.

Unity Wave operates on a role-based access system, ensuring that administrators have complete control over the platform in managing donor and organization registrations, approving requests as well as adding item amounts and categories to the item resources. Our platform eliminates the need for online payments, allowing donors to verify donation details directly through our secure web interface.

Why did we choose a website and DBMS for implementing our project?

- **Data Organization:** With a diverse range of entities including donors, organizations, events, and items, a database management system (DBMS) allows us to efficiently organize and manage this information.
- **Data Security:** By utilizing a DBMS, we prioritize the security and privacy of user data, safeguarding information against unauthorized access and ensuring data integrity.
- **Ease of Maintenance and Updates:** The use of a website and DBMS simplifies data maintenance and updates, facilitating seamless management of donor registrations, event scheduling, and item donations.

Why will people use our website?

In a world where community support is essential, Unity Wave offers individuals a platform to contribute meaningfully and make a difference. Our website enables donors and organizations to participate in initiatives that positively impact lives, providing a transparent and accessible way to engage with charitable activities.

In summary, "Unity Wave" is more than just a donation platform – it's a strategic tool for community empowerment and impact. Join us in creating positive change and making a difference in the lives of those we serve. Together, let's ride the wave of unity towards a brighter future.

Project Features

Our key features for the project are-

Administration handling: The admin dashboard provides comprehensive control over user registrations, donation activities, and item management, ensuring efficient platform oversight.

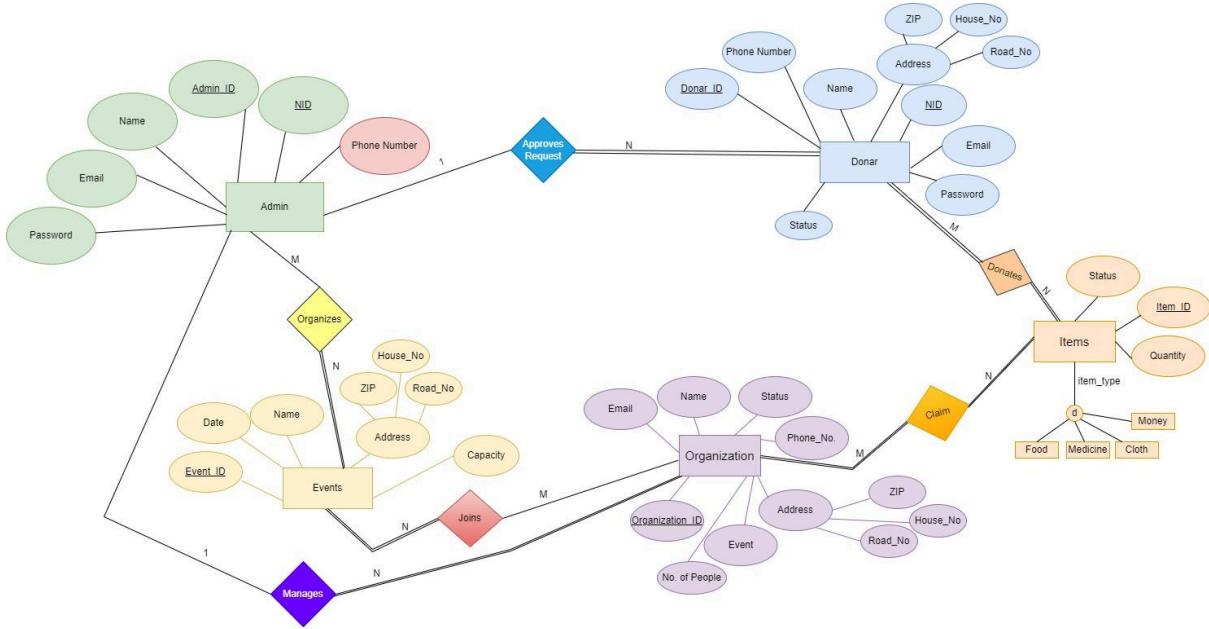
Donation at any amount 09 by registered accounts: Donors can easily register, log in, and participate in donation initiatives, viewing ongoing events and submitting contributions with personalized preferences.

Item collection of various categories: Admin panel allocates for different item types to be availed in the system following which the donors can donate.

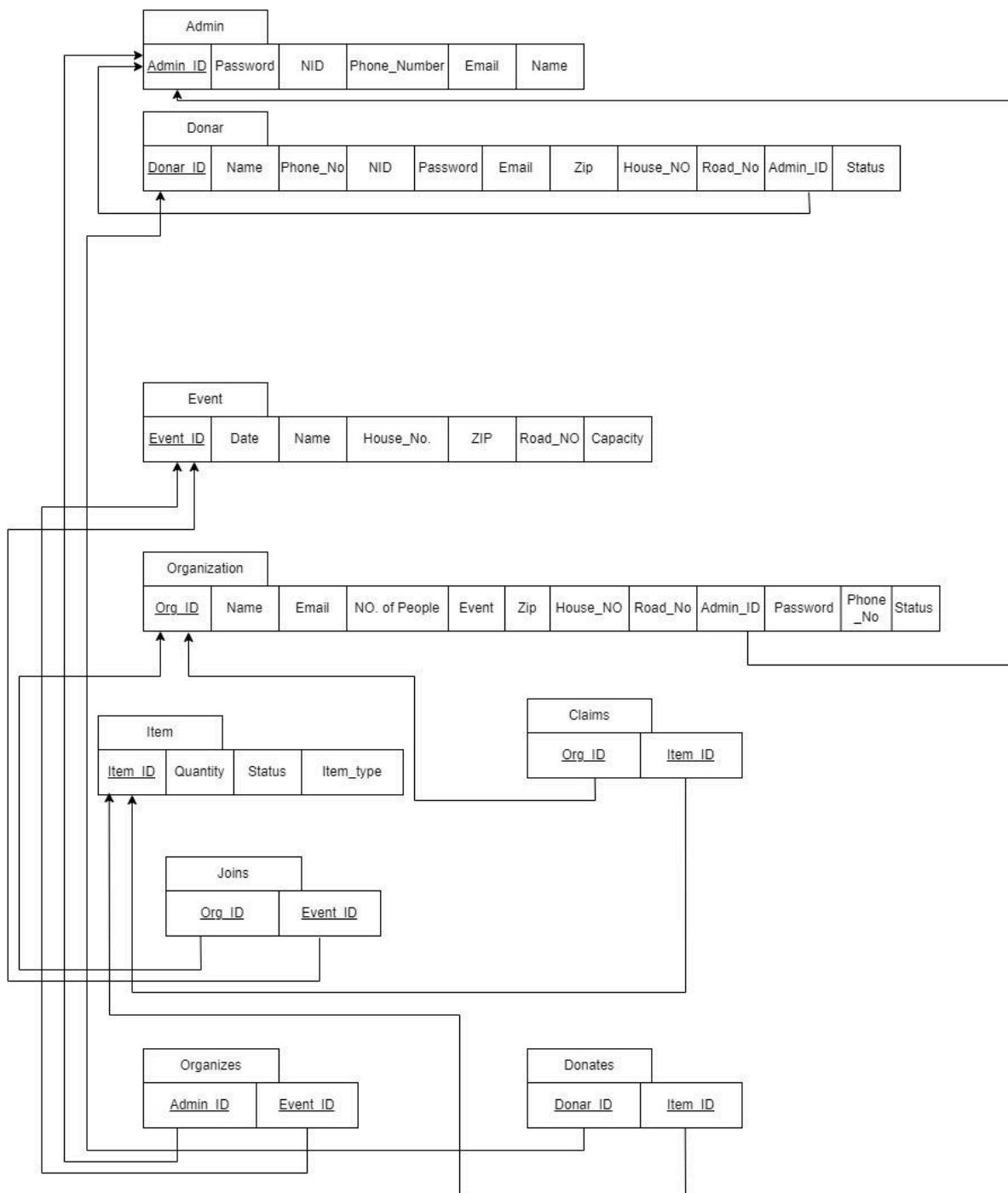
Organization panel participating in community services: Organizations benefit from streamlined registration processes, inventory management tools, and the ability to coordinate donation drives and outreach efforts.

Arrangement and updates about ongoing events: A centralized event hub showcases upcoming activities, and community engagements managed by administrators.

ER/EER Diagram



Schema Diagram



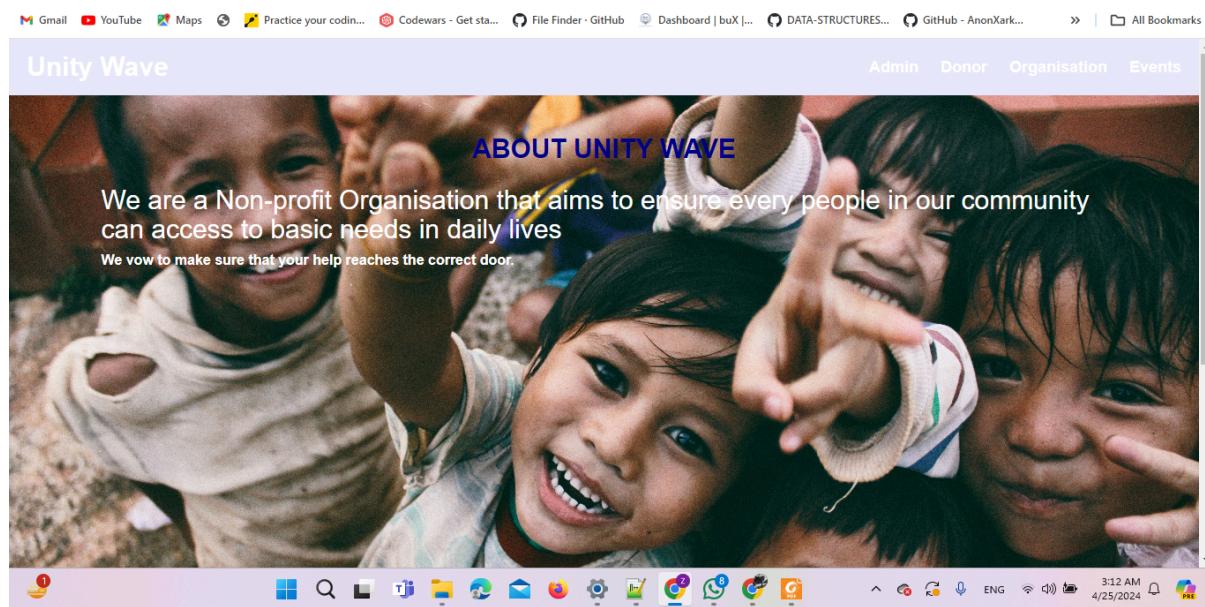
Frontend Development

Contribution of ID : 22101114, Name : Labiba Zahin

(index.php), (DONOR.php),(donar.php),(remove_donor.php):

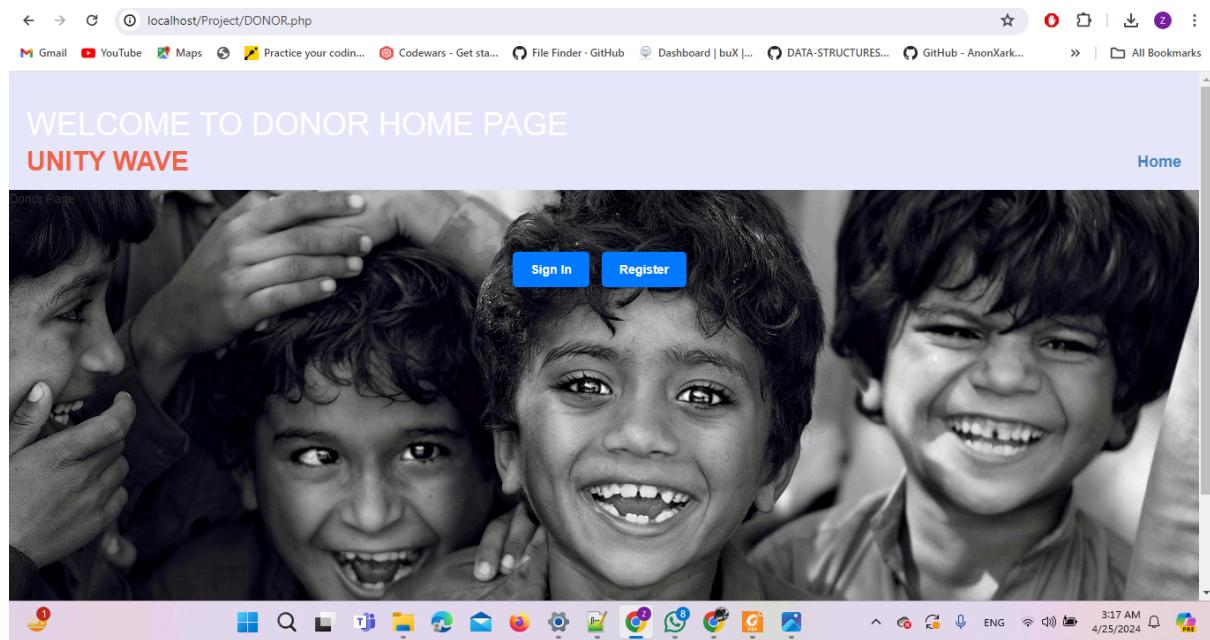
1. Homepage : (index.php)

This is the homepage for our website where we have used a user friendly layout. From here, Admin, Donor, Organization panel can redirect themselves to their specified activities.



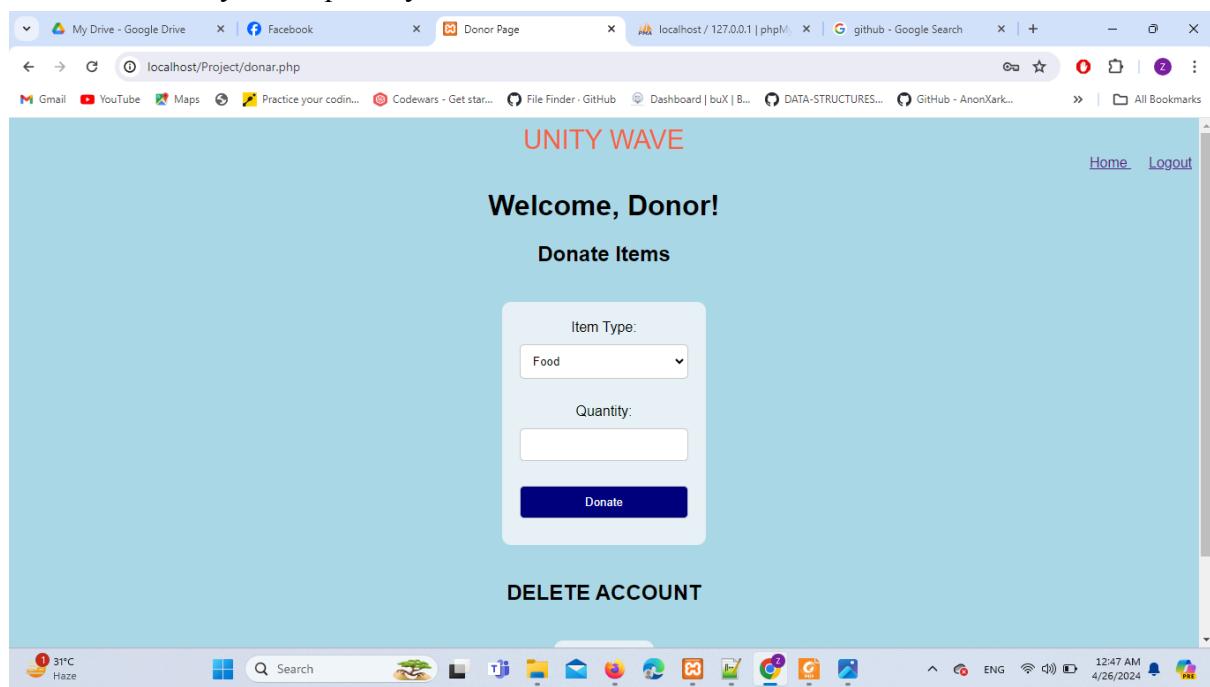
2. Donor Homepage :(DONOR.php):

Here the donor can sign in to donate amounts to our organizations and if new, he will register himself up which will be later on approved by the admin panel.



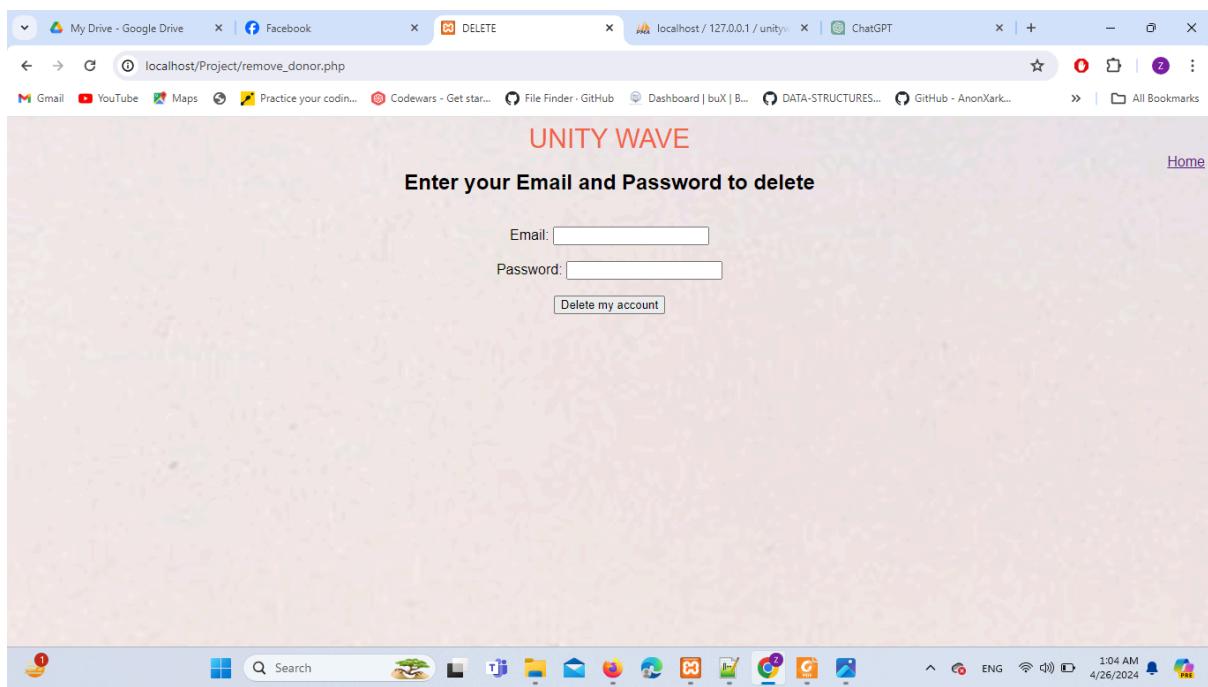
3.(Donor Page /Donate Page): (donar.php)

Upon signing in with valid information, Donors can donate any item that is available in the item list with any valid quantity. ?/



4.DELETE ACCOUNT: (remove_donor.php)

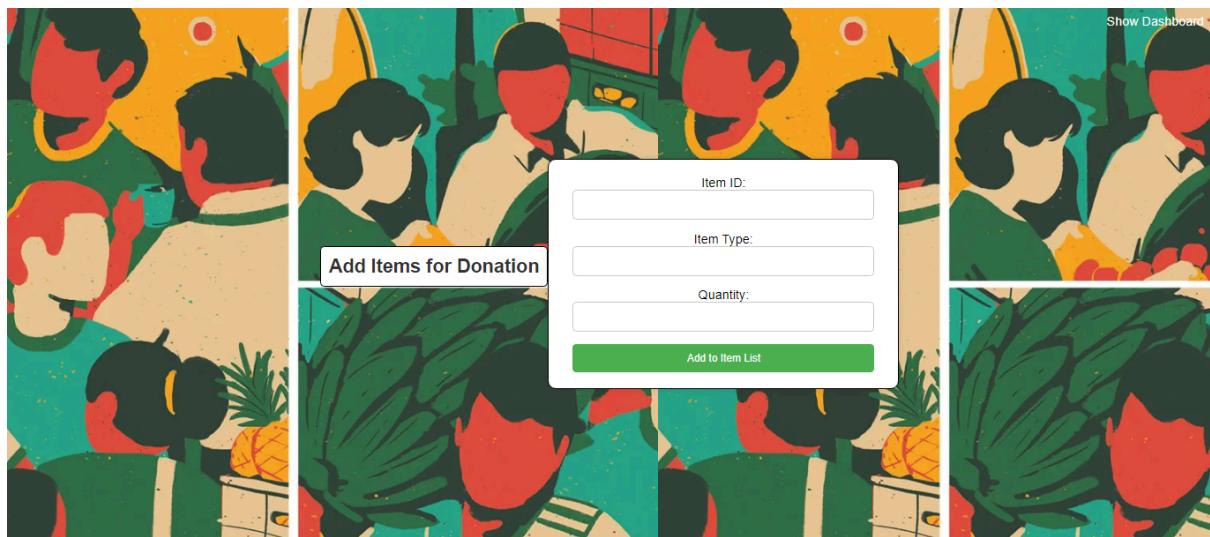
Here Donor if anytime wants to delete the registered account from the website, he can visit this page and by entering his email and password his account will be removed from our website.



Contribution of ID : 22101117, Name : Shawana Maliha (add_item.php, register_donor.php,claim.php)

1. add_item.php

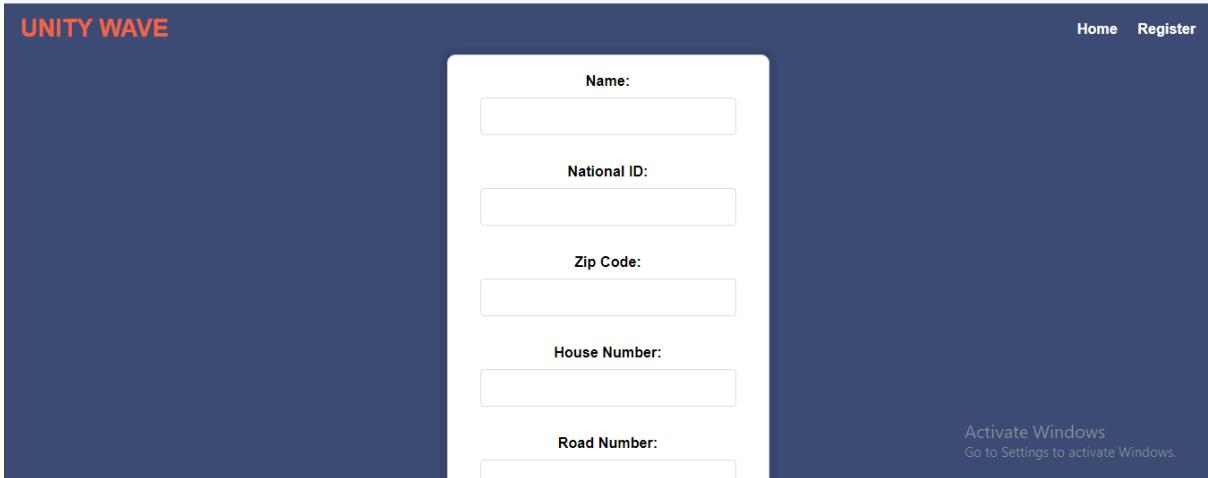
Here a page of adding item types and quantity will be displayed where admin will add to the type and quantity as well as assign an Item_id which will later on redirect him to the insert_item.php



2. register_donor.php

Here donors will fill their information accordingly to get themselves registered and hereby join and participate for our website ongoing donations and activities.

Donor Registration



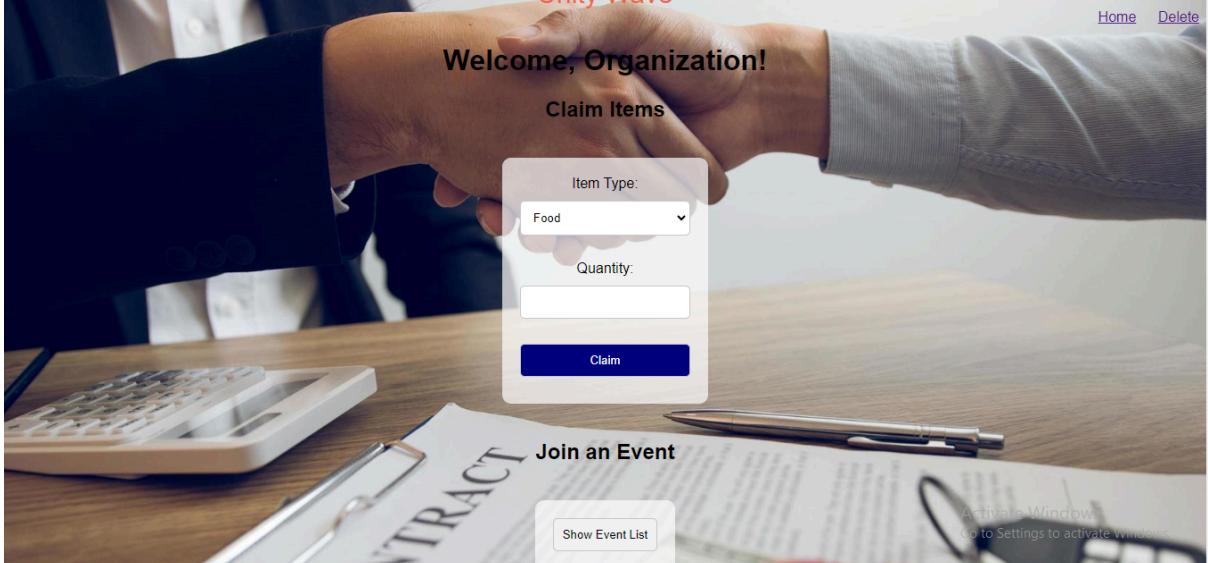
The screenshot shows a registration form titled "Donor Registration" with a dark blue header. The header includes the "UNITY WAVE" logo and navigation links for "Home" and "Register". The form itself has five input fields: "Name:", "National ID:", "Zip Code:", "House Number:", and "Road Number:". To the right of the form, there is a message: "Activate Windows Go to Settings to activate Windows.".

Name:	<input type="text"/>
National ID:	<input type="text"/>
Zip Code:	<input type="text"/>
House Number:	<input type="text"/>
Road Number:	<input type="text"/>

Activate Windows
Go to Settings to activate Windows.

3.claim.php

Organization will insert here which item and quantity it wants to claim for donation purposes. Here organization can take part in participating in various events as well.



The screenshot shows a "Claim Items" form overlaid on a background image of two people shaking hands at a desk. The form has a dropdown menu for "Item Type" set to "Food", a "Quantity" input field, and a "Claim" button. The background image also features a calculator, a pen, and a clipboard with the word "CONTRACT" visible. A "Join an Event" button and a "Show Event List" button are also present. A message on the right side says: "Activate Windows Go to Settings to activate Windows."

Welcome, Organization!

Claim Items

Item Type: Food

Quantity:

Claim

Join an Event

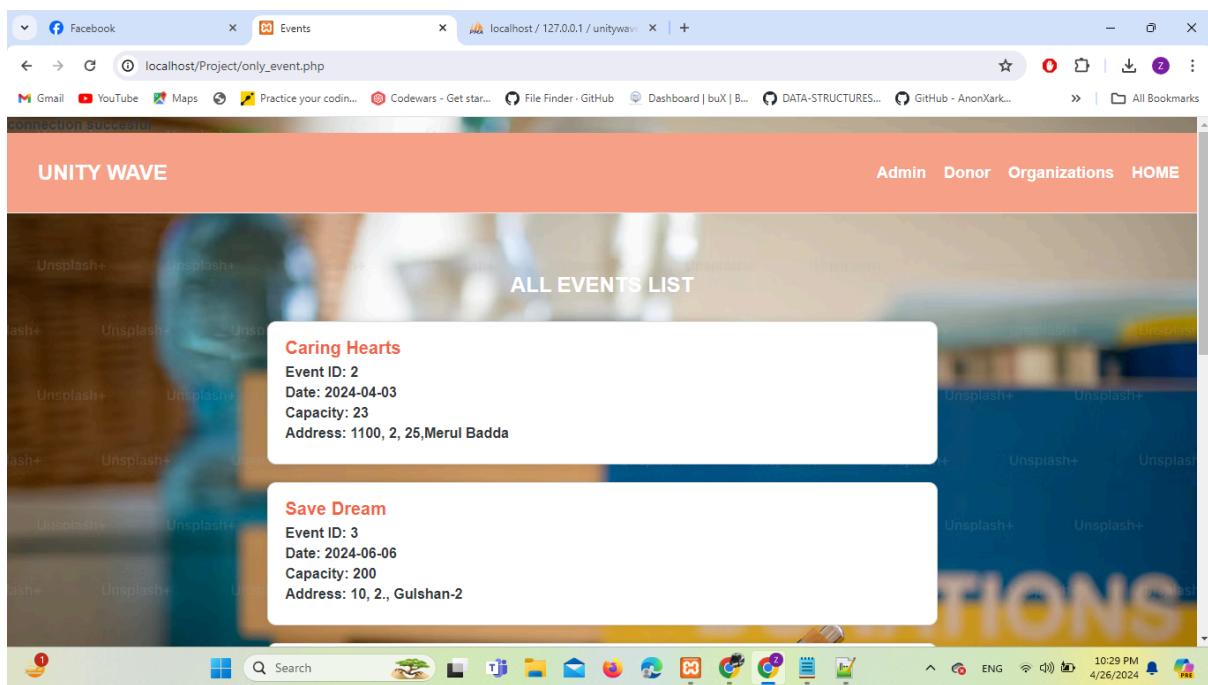
Show Event List

Activate Windows
Go to Settings to activate Windows.

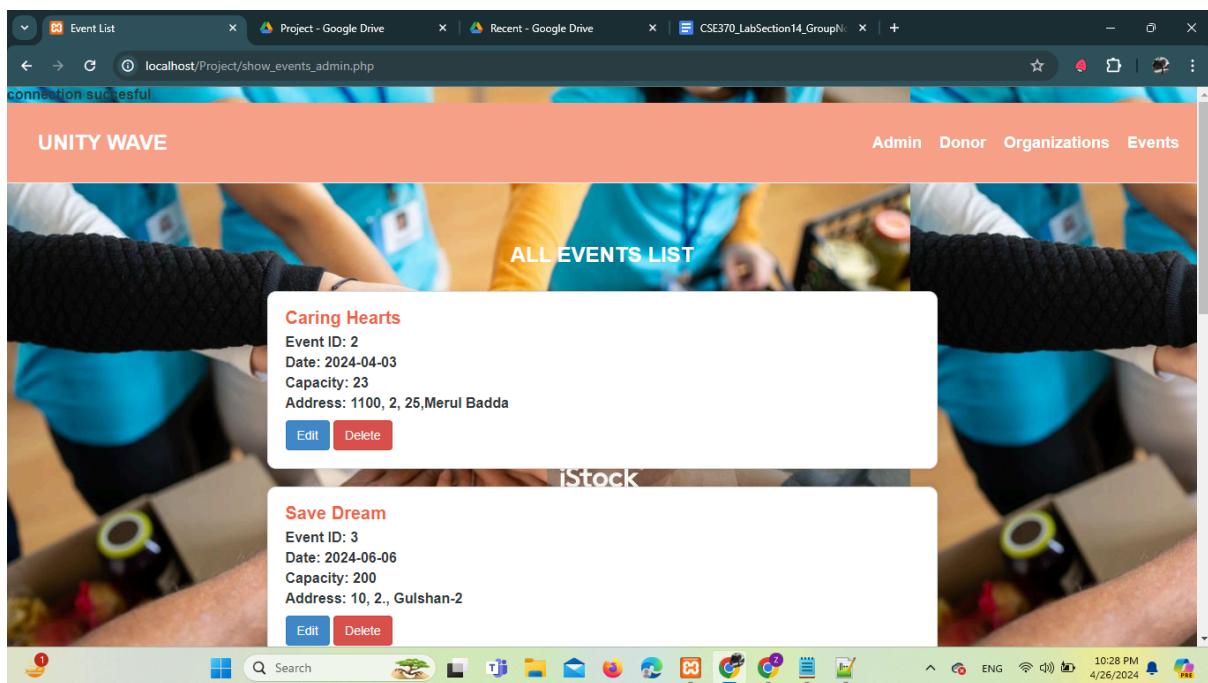
Contribution of ID : 22101249, Name: Zarin Tasnim
(show_events_admin.php, only_event.php)

1. Event Home page (only_event.php)

All available events will be displayed for donor and organization



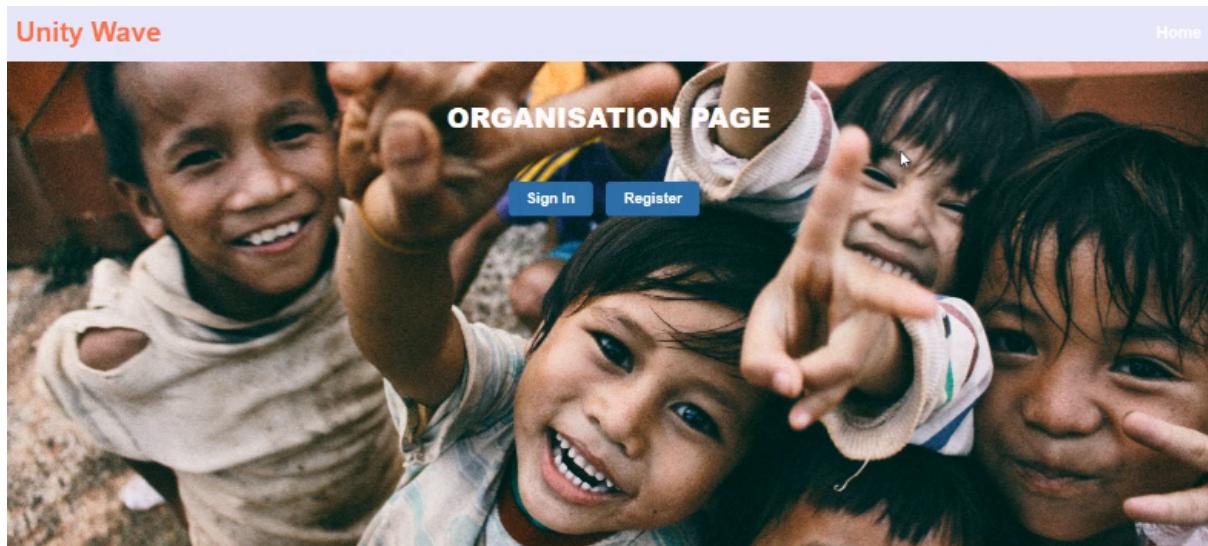
- Existing and newly added events will be displayed and only the admin can access them and update or delete them accordingly.



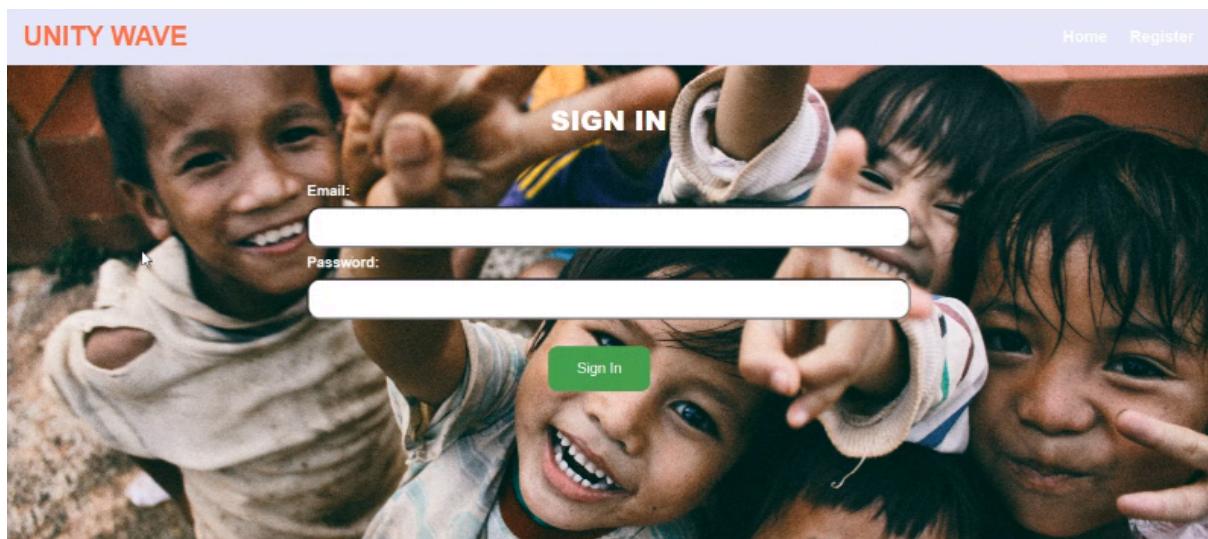
Contribution of ID : 22101343, Name : Tasnim Zaman
(org_index.php, organization_login.php, organization_register.php, org_event.php, org_delete.php)

1. Organization Home Page (org_index.php):

Any organization user can choose the sign in or register option



2. Organization Login Page (organization_login) : Organization can login with email and password in this page



3. Organization Registration form (organization_register.php):

Organizations can insert all their information in this page which will get added in the table with the status “Pending”, when the admin approves the account the organization will be able to claim items.

UNITY WAVE

ADD YOUR ORGANIZATION

[Home](#) [Sign In](#)

Name of Organization:

Number of People:

Zip:

House Number:

Road Number:

Phone Number:

Go to Settings to activate Windows.

4. Organization selecting events (org_event):

Organizations can see the event table and choose which event to join by inserting their email and Event Id from the event table.

Unity Wave

All Event List

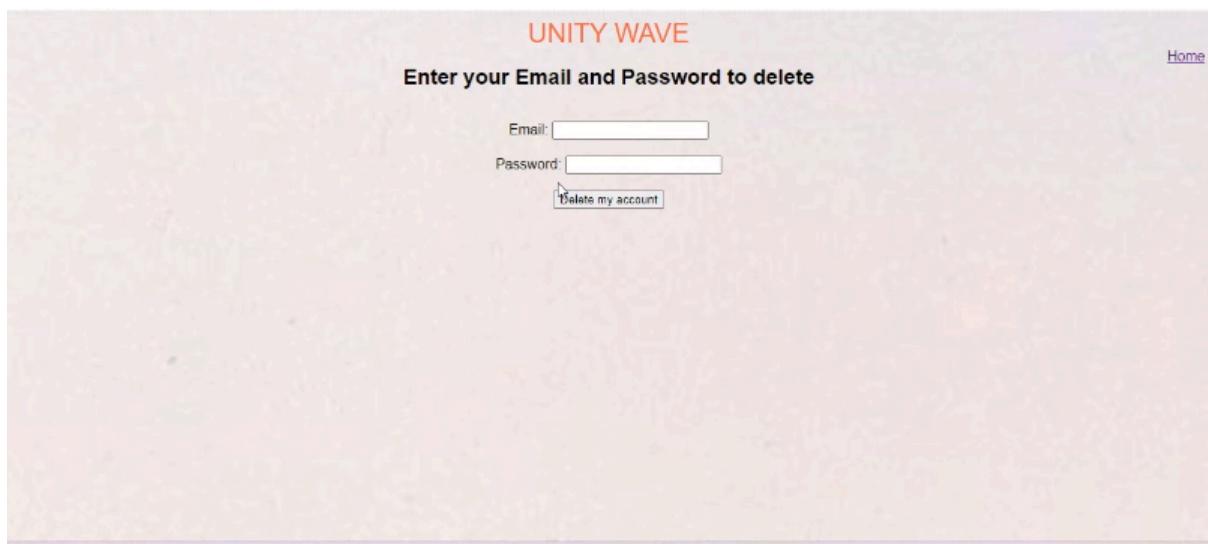
Event ID	Event Name	Date	Zip
House	Road	Capacity	
connection successful			
2	Caring Hearts	2024-04-03	1100
2	25.Merul Badda	23	
3	Save Dream	2024-06-06	10
	2., Gulshan-2	200	
4	Save A life	2024-06-06	1
	2., Gulshan-1	300	
5	bgd	2024-06-13	2
	34 lalmaita	100	
7	we for us	2024-06-26	2
	2.Bardhara	36	
9	wqe	2024-04-23	2
22	21.lal	12	

Email:

10:21 PM 4/26/2024

5. Organization account delete page (org_delete):

Organizations can delete their account by inserting their email and password



Contribution of ID : 22101335, Name : Hasan Sarwar Zami

1. Admin Dashboard(admin_dashboard.php)

The screenshot shows a dual-monitor setup. The left monitor displays the Admin Dashboard at `localhost/unity_wave/admin_dashboard.php`. It features a header with "UNITY WAVE" and a user profile section showing "Admin Name: Hasan", "ID: 1", "Email: hasan@gmail.com", and "Phone No: 01742102030". Below this is a "Events" section with a "Add Event" button, a "Donor Requests" button, and an "Organization Requests" button. The main content area is titled "Items" and lists four items in boxes:

Item Type	Quantity	Status
Food	140	Available
Medicine	50	Available
Cloths	500	Available
Money	125000	Available

The right monitor shows the same dashboard page at `localhost/unity_wave/admin_dashboard.php`, which is identical to the left one.

2. Donor Request List (donor_requests.php)

The screenshot shows a web browser window titled "Donor Request" with the URL "localhost/unity_wave/donor_requests.php". The page has a header "Unity Wave" and a sub-header "DONOR REQUEST LIST". Below the sub-header is a table with the following data:

Donor ID	Name	Email	Phone	Actions
1	John Doe	john@example.com	123-456-7890	<button>Approve</button> <button>Reject</button>
2	Jane Smith	jane@example.com	987-654-3210	<button>Approve</button> <button>Reject</button>

The background of the page features a photograph of several smiling children.

3. Organization Request List (organization_requests.php)

The screenshot shows a web browser window titled "Organization Requests" with the URL "localhost/unity_wave/organization_requests.php". The page has a header "Unity Wave" and a sub-header "Organization Request List". Below the sub-header is a table with the following data:

Organization ID	Name	Email	Phone	Actions
4	Organization B	orgb@example.com	123-456-7890	<button>Approve</button> <button>Reject</button>
5	Organization C	orgc@example.com	987-654-3210	<button>Approve</button> <button>Reject</button>
6	Organization D	orgd@example.com	456-789-0123	<button>Approve</button> <button>Reject</button>

The background of the page features a photograph of three smiling children.

Backend Development

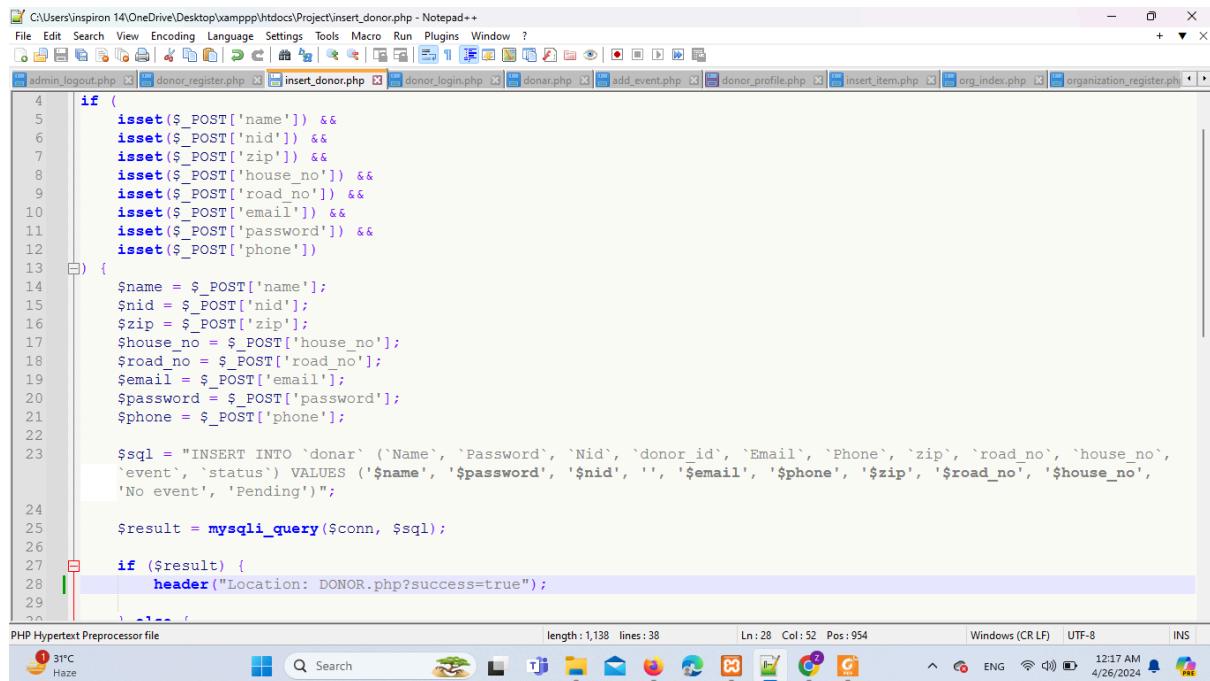
Contribution of ID : 22101114, Name : Labiba Zahin

[\(insert_donor.php\),\(donor_login.php\),\(donor_logout.php\),\(donor_deleting.php\)](#)

1.DONOR REGISTRATION ROUTE:

[\(insert_donor.php\)](#)

To register, INSERT command has been used to insert data into the database



The screenshot shows a Notepad++ window with the file path C:\Users\inspiron 14\OneDrive\Desktop\xampp\htdocs\Project\insert_donor.php. The code is a PHP script for inserting donor information into a MySQL database. It checks if all required POST variables are set and then constructs an SQL INSERT statement. A header redirect is included at the end.

```
4 if (
5     isset($_POST['name']) &&
6     isset($_POST['nid']) &&
7     isset($_POST['zip']) &&
8     isset($_POST['house_no']) &&
9     isset($_POST['road_no']) &&
10    isset($_POST['email']) &&
11    isset($_POST['password']) &&
12    isset($_POST['phone']))
13 {
14     $name = $_POST['name'];
15     $nid = $_POST['nid'];
16     $zip = $_POST['zip'];
17     $house_no = $_POST['house_no'];
18     $road_no = $_POST['road_no'];
19     $email = $_POST['email'];
20     $password = $_POST['password'];
21     $phone = $_POST['phone'];
22
23     $sql = "INSERT INTO `donar` (`Name`, `Password`, `Nid`, `donor_id`, `Email`, `Phone`, `zip`, `road_no`, `house_no`, `event`, `status`) VALUES ('$name', '$password', '$nid', '', '$email', '$phone', '$zip', '$road_no', '$house_no', 'No event', 'Pending')";
24
25     $result = mysqli_query($conn, $sql);
26
27     if ($result) {
28         header("Location: DONOR.php?success=true");
29     }
}
```

2.DONOR LOGIN ROUTE:

[\(donor_login.php\)](#)

To login, SELECT command has been used to check whether email and password match. IF the given information matches, the page redirects to the donation (donar.php) page otherwise Redirects to self.

C:\Users\inspiron 14\OneDrive\Desktop\xampp\htdocs\Project\donor_login.php - Notepad++

```
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
donor_login.php donor.php add_event.php donor_profile.php insert_item.php org_index.php organization_register.php signin.php organization_login.php insert_organization...
34 <section id = "section1">
35     <div class="title"> Donor Login </div>
36
37     <?php
38         require_once('Dbconnect.php');
39
40         if(isset($_POST['email']) && isset($_POST['pass'])){
41             $email = $_POST['email'];
42             $password = $_POST['pass'];
43
44             $sql = "SELECT * FROM donor WHERE email = '$email' AND password = '$password' AND status = 'Approved'";
45             $result = mysqli_query($conn, $sql);
46
47             if(mysqli_num_rows($result) != 0){
48
49                 header("Location: donor.php");
50             }
51             else{
52                 echo "<p style='color:red;'>Invalid email or password. Please provide correct information Or wait for approval.</p>";
53             }
54         }
55     ?>
56
57     <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" class="form_design" method="post">
58         <label>Email:</label> <input type="email" name="email"> <br/>
59         <label>Password:</label> <input type="password" name="pass"> <br/>
60     </form>

```

PHP Hypertext Preprocessor file

length : 2,619 lines : 77 Ln:47 Col:12 Pos:1,628 Windows (CR LF) UTF-8 INS

31°C Haze

Search

12:19 AM 4/26/2024

3.DONOR LOGOUT ROUTE:

(donor_logout.php)

Redirects to Login Page

C:\Users\inspiron 14\OneDrive\Desktop\xampp\htdocs\Project\donor_logout.php - Notepad++

```
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
claim.php claim_item.php update_event.php org_delete.php org_deleting.php donor_deleting.php org_event.php update_org_event.php DONOR.php donor_logout.php...
1 <?php
2
3     session_start();
4     session_unset();
5     session_destroy();
6     header("Location: donor_login.php");
7     exit;
8 ?>
```

PHP Hypertext Preprocessor file

length : 112 lines : 8 Ln:5 Col:19 Pos:64 Windows (CR LF) UTF-8 INS

12:29 AM 4/26/2024

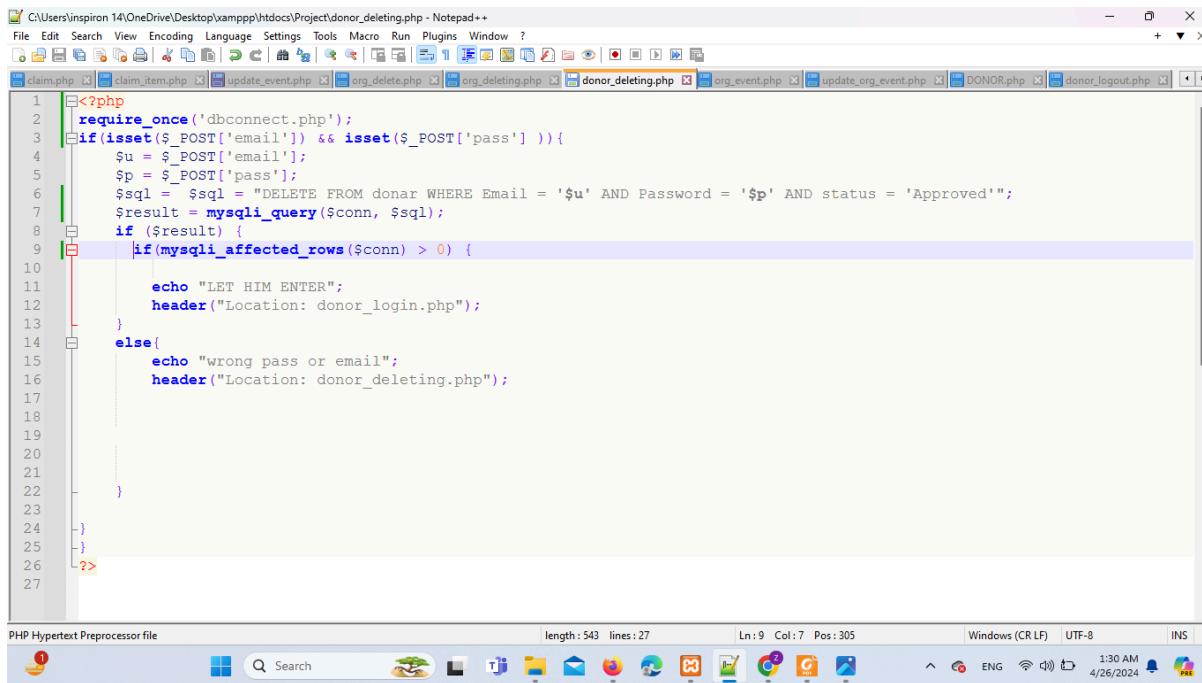
4.Showing donor Available item types: (donar.php)

To Make things easier for the user, the available item types are shown . To ensure this SELECT command has been used.

```
60 <form action="insert_item.php" method="post">
61   <label for="item_type">Item Type:</label>
62   <select name="item_type" id="item_type">
63     <?php
64       require_once('DBconnect.php');
65       $sql = "SELECT DISTINCT item_type FROM items";
66       $result = mysqli_query($conn, $sql);
67       while ($row = mysqli_fetch_assoc($result)) {
68         echo "<option value='" . $row['item_type'] . "'>" . $row['item_type'] . "</option>";
69       }
70       if (isset($_GET['success'])) {
71         echo "<p>Your registration is being approved.</p>";
72       }
73     mysqli_close($conn);
```

5.Delete Profile Route (donor_deleting.php)

In order to delete a profile, DELETE command has been used. It has also been ensured that after deletion , the page redirects to the donor login page.



```
<?php
require_once('dbconnect.php');
if(isset($_POST['email']) && isset($_POST['pass'])) {
  $u = $_POST['email'];
  $p = $_POST['pass'];
  $sql = "DELETE FROM donor WHERE Email = '$u' AND Password = '$p' AND status = 'Approved'";
  $result = mysqli_query($conn, $sql);
  if ($result) {
    if(mysqli_affected_rows($conn) > 0) {
      echo "LET HIM ENTER";
      header("Location: donor_login.php");
    }
  } else {
    echo "wrong pass or email";
    header("Location: donor_deleting.php");
  }
}
?>
```

Contribution of ID : 22101117, Name : Shawana Maliha

(insert_item.php, donor_process_request.php, claim_item.php, insert_item.php)

1. ad_insert_item.php

When admin visits the dashboard and wants to insert items by assigning with an item id , through ad_insert_item the update for he items table is done where it goes to the Item id and updates the quantity(UPDATE) and if the item is new, it assigns to the table as a new row(INSERT INTO).

```
<?php
require_once('Dbconnect.php');
if(isset($_POST['iid']) && isset($_POST['cat']) &&
isset($_POST['quan'])) {
    $item_id = $_POST['iid'];
    $catagory = $_POST['cat'];
    $quantity = $_POST['quan'];
    $check_sql = "SELECT * FROM items WHERE item_id =
'$item_id'";
    $check_result = mysqli_query($conn, $check_sql);

    if(mysqli_num_rows($check_result) > 0) {
        $update_sql = "UPDATE items SET quantity = quantity +
'$quantity' WHERE item_id = '$item_id' AND item_type =
'$catagory'";
        $update_result = mysqli_query($conn, $update_sql);

        if($update_result) {
            echo "Item updated successfully.";
            header("Location: admin_dashboard.php");
        } else {
            echo "<p style='color:red;'>Failed to update
item.</p>";
            header("Location: admin_dashboard.php");
        }
    } else {
        $insert_sql = "INSERT INTO items (item_id, item_type,
quantity, status) VALUES ('$item_id', '$catagory', '$quantity',
'Available')";
        $insert_result = mysqli_query($conn, $insert_sql);

        if($insert_result) {
            echo "Item added successfully.";
            header("Location: admin_dashboard.php");
        } else {
    
```

```

        echo "<p style='color:red;'>Failed to update
item.</p>";
        header("Location: admin_dashboard.php");
    }
}
} else {
    echo "All fields are required.";
}

mysqli_close($conn);
?>

```

2. Donor process request.php

In this the registered donor requests will be approved as soon as the admin gives approval for that donor id (UPDATE) and if rejected it removes him from the table(DELETE).

```

<?php
if (isset($_POST["approve"])) {
require_once("DBconnect.php");

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $donor_id = $_POST["donor_id"];
    $sql = "UPDATE donar SET status = 'Approved' WHERE
donor_id = '$donor_id';

    if (mysqli_query($conn, $sql)) {
        header("Location: donor_requests.php");
        exit();
    } else {
        echo "Error updating record: " . mysqli_error($conn);
    }
}

if (isset($_POST["reject"])) {
    $donor_id = $_POST["donor_id"];
    $sql = "DELETE FROM donar WHERE donor_id = '$donor_id';

    if (mysqli_query($conn, $sql)) {
        header("Location: donor_requests.php");
        exit();
    } else {
        echo "Error deleting record: " . mysqli_error($conn);
    }
}

```

```

        }
    }
?>

```

3. claim_item.php

Here when an organization claims an item, it takes that amount from the item list and updates the items table by reducing the quantity(UPDATE) . For this, it also uses a condition to only remove when that much amount is available to the list. Otherwise, it shows invalid request.

```

<?php
include_once('Dbconnect.php');

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $item_type = $_POST['item_type'];
    $quantity = $_POST['quantity'];
    $check_sql = "SELECT quantity FROM items WHERE item_type =
$item_type";
    $result = $conn->query($check_sql);

    if ($result->num_rows > 0) {
        $row = $result->fetch_assoc();
        $current_quantity = $row['quantity'];

        if ($current_quantity >= $quantity && $quantity > 0) {
            $new_quantity = $current_quantity - $quantity;
            $sql = "UPDATE items SET quantity = $new_quantity WHERE
item_type = '$item_type';

            if ($conn->query($sql) === TRUE) {
                echo "Items claimed successfully.";
            } else {
                echo "Error updating record: " . $conn->error;
            }
        } else {
            echo "Insufficient quantity available or invalid
quantity.";
        }
    } else {
        echo "Item type not found.";
    }

    $conn->close();
}

```

```

}

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Thank You!</title>
    <style>
        body {
            background-color: #ADD8E6;
            font-family: Arial, sans-serif;
            text-align: center;
        }
    </style>
</head>
<body>
    <h1>Thank You!</h1>
    <?php
        if (isset($_GET['message'])) {
            $message = $_GET['message'];
            echo "<p>$message</p>";
        }
    ?>
</body>
</html>

```

4.insert_item.php:

After donation , the quantity of items increases in the database to ensure this UPDATE and SET commands have been used.

```

<?php
include_once('Dbconnect.php');

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $item_type = $_POST['item_type'];
    $quantity = $_POST['quantity'];
    $sql = "UPDATE items SET quantity = quantity + $quantity WHERE
item_type = '$item_type'";
    if ($conn->query($sql) === TRUE) {
        echo "Donation recorded successfully.";
    } else {

```

```
        echo "Error updating record: " . $conn->error;
    }
    $conn->close();
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Thank You!</title>
    <style>
        body {
            background-color: #ADD8E6;
            font-family: Arial, sans-serif;
            text-align: center;
        }
    </style>
</head>
<body>
    <h1>Thank You!</h1>
    <?php
        if (isset($_GET['message'])) {
            $message = $_GET['message'];
            echo "<p>$message</p>";
        }
    ?>
</body>
</html>
```

Contribution of ID : 22101249, Name: Zarin Tasnim (edit_event.php, update_event.php, delete_event.php, insert_event.php)

1.Editing event using SELECT command where existing events can be edited by admin

```
<?php
require_once("DBconnect.php");
if(isset($_GET['id'])) {
    $eventId = $_GET['id'];

    $sql = "SELECT event_id, date, name, capacity, zip, house, road_no FROM events WHERE event_id = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("s", $eventId);
    $stmt->execute();
    $stmt->store_result();

    if ($stmt->num_rows > 0) {
        $stmt->bind_result($eventId, $date, $name, $capacity, $zip, $house, $road_no);

        $stmt->fetch();
    } else {
        echo "Event not found";
        exit();
    }
}?
<form action="update_event.php" method="post" class="mt-4 p-4 border rounded bg-light">
    <input type="hidden" name="event_id" value="<?php echo $eventId; ?>">
    <div class="form-group">
        <label for="date">Date:</label>
        <input type="date" class="form-control" id="date" name="date" value="<?php echo $date; ?>" required>
    </div>
    <div class="form-group">
        <label for="name">Name:</label>
        <input type="text" class="form-control" id="name" name="name" value="<?php echo $name; ?>" required>
    </div>
    <div class="form-group">
        <label for="capacity">Capacity:</label>
        <input type="number" class="form-control" id="capacity" name="capacity" value="<?php echo $capacity; ?>" required>
    </div>
    <div class="form-group">
        <label for="zip">Zip:</label>
        <input type="text" class="form-control" id="zip" name="zip" value="<?php echo $zip; ?>" required>
    </div>
    <div class="form-group">
        <label for="house">House:</label>
        <input type="text" class="form-control" id="house" name="house" value="<?php echo $house; ?>" required>
    </div>
    <div class="form-group">
        <label for="road_no">Road No.:</label>
        <input type="text" class="form-control" id="road_no" name="road_no" value="<?php echo $road_no; ?>" required>
    </div>
    <button type="submit" class="btn btn-primary">Update</button>
</form>
```

2. Updating edited information using UPDATE command. Edited information on existing events can be updated using this. (update.php)

```

<?php

if ($_SERVER["REQUEST_METHOD"] == "POST") {

    require_once ("DBconnect.php");

    $eventId = $_POST['event_id'];
    $date = $_POST['date'];
    $name = $_POST['name'];
    $capacity = $_POST['capacity'];
    $zip = $_POST['zip'];
    $house = $_POST['house'];
    $road_no = $_POST['road_no'];

    $sql = "UPDATE events SET date=?, name=?, capacity=?, zip=?, house=?, road_no=? WHERE event_id=?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("ssssss", $date, $name, $capacity, $zip, $house, $road_no, $eventId);

    if ($stmt->execute()) {
        echo "Event details updated successfully";
    } else {
        echo "Error updating event details: " . $stmt->error;
    }

    $stmt->close();
    $conn->close();
}

?>

```

3. Deleting any event on the list using the command DELETE. Events out of date will be deleted by admin through it.(delete.php)

```

<?php
require_once ("DBconnect.php");

if(isset($_GET['id'])) {
    $eventId = $_GET['id'];

    // Delete the event from the database
    $sql = "DELETE FROM events WHERE event_id = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("i", $eventId);

    if ($stmt->execute()) {
        // Redirect back to the show_events.php page after successful deletion
        header("Location: show_events.php");
        exit();
    } else {
        echo "Error deleting event: " . $conn->error;
    }
} else {
    echo "Event ID not provided";
}

?>

```

4. Insertion of events using INSERT command. Admin will be able to insert events using it.(insert.php)

```

<?php
// first of all, we need to connect to the database
require_once('DBconnect.php');

// we need to check if the input in the form textfields are not empty
if(isset($_POST['eid']) && isset($_POST['ename']) && isset($_POST['address']) && isset($_POST['date']) && isset($_POST['cap'])){
    // write the query to check if this username and password exists in our database
    $u = $_POST['eid'];
    $p = $_POST['ename'];
    $c = $_POST['address'];
    $a = $_POST['date'];
    $x = $_POST['cap'];

    $sql = " INSERT INTO event VALUES( '$u', '$p', '$c', '$a' ,'$x') ";

    //Execute the query
    $result = mysqli_query($conn, $sql);

    //check if this insertion is happening in the database
    if(mysqli_affected_rows($conn)){
        echo "Inserted Successfully";
        header("Location: show_events_admin.php");
    }
    else{
        echo "Insertion Failed";
        header("Location: add_event.php");
    }
}

```

Contribution of ID : 22101343, Name : Tasnim Zaman

- Organization signing in using SELECT (signin.php) :** Letting the organization user access to the next page if they insert valid email and password existing in the ‘organization’ table.

```

<?php

require_once('dbconnect.php');

if(isset($_POST['email']) && isset($_POST['pass']) ){

    $email = $_POST['email'];
    $pass = $_POST['pass'];
    $sql = "SELECT * FROM organization WHERE Email = '$email' AND pass = '$pass' AND Status = 'Approved' ";

    $result = mysqli_query($conn, $sql);

    if(mysqli_num_rows($result) !=0 ) {

        echo "LET HIM ENTER";
        header("Location: claim.php");
    }
}

```

```

    }

    else{
        echo "<p style='color:red;'>Invalid email or password. Please
provide correct information Or wait for approval.</p>";
        header("Refresh: 3; URL=organization_login.php");
        exit();
    }
}

?>

```

2. New organization inserting in the table using INSERT (insert_organization.php)

: Taking the information from the user and inserting it in the ‘organization’ table where event , and organization id is initially empty. Organization id will increment automatically and the status will initially be ‘Pending’.

```

<?php
require_once('dbconnect.php');

if(isset($_POST['name']) && isset($_POST['noofpeople']) &&
isset($_POST['zip']) && isset($_POST['houseno']) &&
isset($_POST['roadno']) && isset($_POST['phn']) &&
isset($_POST['email']) && isset($_POST['pass'])){

    $n = $_POST['name'];
    $no = $_POST['noofpeople'];
    $z = $_POST['zip'];
    $h = $_POST['houseno'];
    $r = $_POST['roadno'];
    $phn = $_POST['phn'];
    $e = $_POST['email'];
    $p = $_POST['pass'];
    $s = 'Pending';

    $sql = "INSERT INTO `organization` (`Org_id`, `Name`, `Email`,
`Pass`, `No_of_people`, `Zip`, `House_No`, `Road_No`, `Phone_no`,
`Event`, `Status`) VALUES ('', '$n', '$e', '$p', '$no', '$z', '$h',
'$r', '$phn', '', '$s')";
}

```

```

$result = mysqli_query($conn, $sql);

if(mysqli_affected_rows($conn)) {

    echo "Inserted Successfully";
    header("Location: organization_login.php");
}

else{
    echo "Insertion Failed";
    header("Location: organization_register.php");

}

?>

```

3. Updating the event attribute in organization table using UPDATE

(update_org_event.php) :

Taking the users email and event id as input and updating the value of ‘event’ where initially it was empty in the ‘organization’ table.

```

<?php
require_once('dbconnect.php');

if(isset($_POST['email']) && isset($_POST['event_id'])) {
    $u = $_POST['email'];
    $p = $_POST['event_id'];
    $sql = "UPDATE organization SET Event = '$p' WHERE Email = '$u' AND
status = 'Approved'";

    $result = mysqli_query($conn, $sql);

    if ($result) {
        if(mysqli_affected_rows($conn) > 0) {

            echo "Event added";
        }
    }
}
else{
    echo "wrong pass or email";
}

```

```
    }

}

?>
```

4. Deleting the account using DELETE (org_deleting):

By taking the email and password the users row from the ‘organization’ table is deleted.

```
<?php

require_once('dbconnect.php');

if(isset($_POST['email']) && isset($_POST['pass'])){

    $u = $_POST['email'];
    $p = $_POST['pass'];
    $sql = $sql = "DELETE FROM organization WHERE Email = '$u' AND
pass = '$p' AND status = 'Approved'";

$result = mysqli_query($conn, $sql);

if ($result) {
    if(mysqli_affected_rows($conn) > 0) {

        echo "LET HIM ENTER";
        header("Location: organization_login.php");
    }
    else{
        echo "wrong pass or email";
        header("Location: deleting.php");
    }
}

}

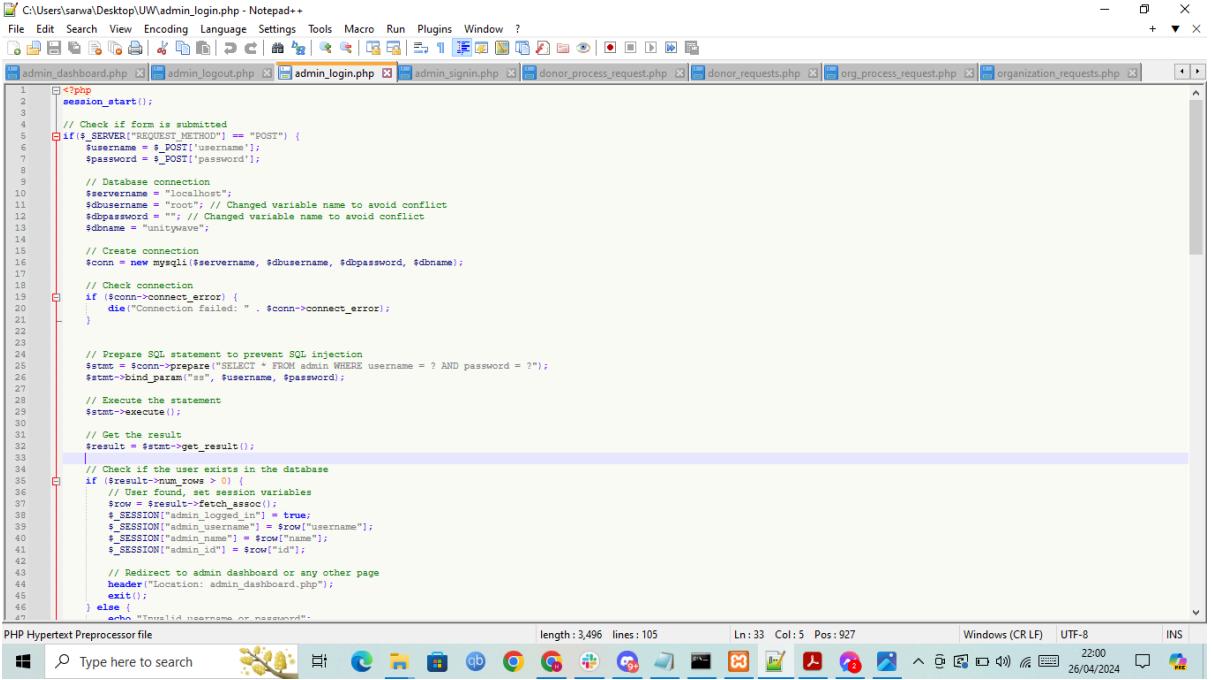
?>
```

Contribution of ID : 22101335, Name : Hasan Sarwar Zami

(admin_login.php, admin_logout.php, add_event.php, org_process_request.php)

1. Admin Login:

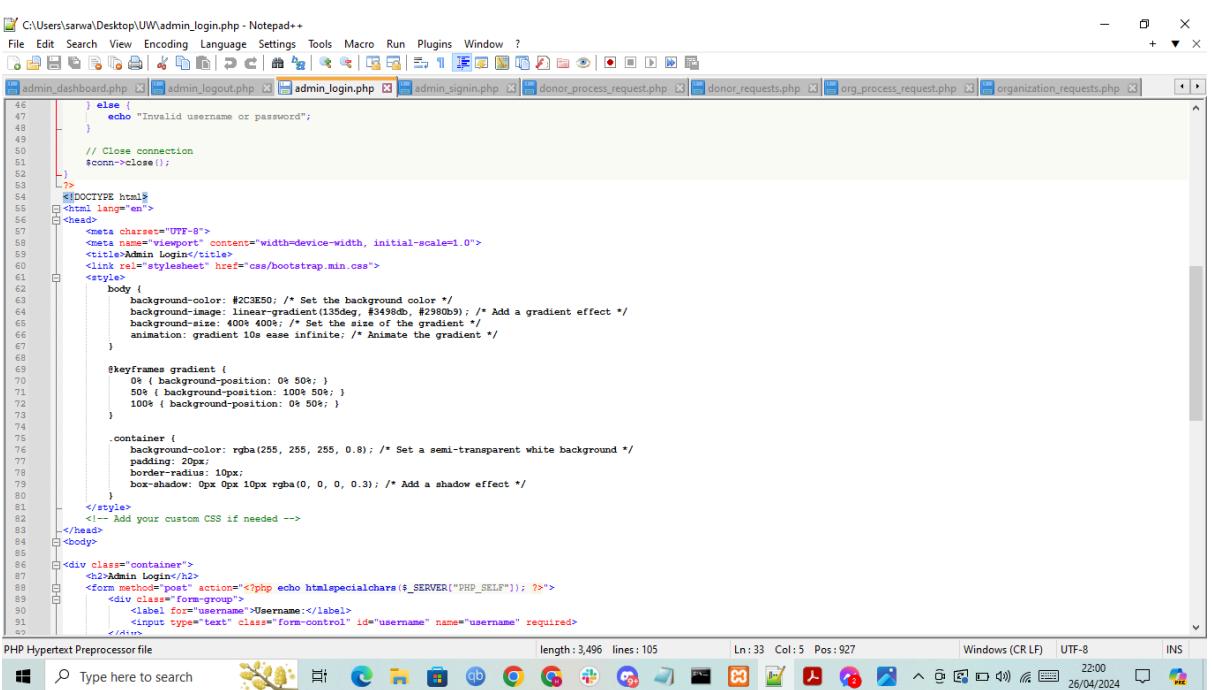
Logging into the admin dashboard where the username and password is fetched from the admin table of the database



```

1 <?php
2 session_start();
3
4 // Check if form is submitted
5 if($_SERVER["REQUEST_METHOD"] == "POST") {
6     $username = $_POST['username'];
7     $password = $_POST['password'];
8
9     // Database connection
10    $servername = "localhost";
11    $dbusername = "root"; // Changed variable name to avoid conflict
12    $dbpassword = ""; // Changed variable name to avoid conflict
13    $dbname = "unitywave";
14
15    // Create connection
16    $conn = new mysqli($servername, $dbusername, $dbpassword, $dbname);
17
18    // Check connection
19    if ($conn->connect_error) {
20        die("Connection failed: " . $conn->connect_error);
21    }
22
23
24    // Prepare SQL statement to prevent SQL injection
25    $stmt = $conn->prepare("SELECT * FROM admin WHERE username = ? AND password = ?");
26    $stmt->bind_param("ss", $username, $password);
27
28    // Execute the statement
29    $stmt->execute();
30
31    // Get the result
32    $result = $stmt->get_result();
33
34    // Check if the user exists in the database
35    if ($result->num_rows > 0) {
36        // User exists, set session variables
37        $row = $result->fetch_assoc();
38        $_SESSION["admin_logged_in"] = true;
39        $_SESSION["admin_username"] = $row["username"];
40        $_SESSION["admin_name"] = $row["name"];
41        $_SESSION["admin_id"] = $row["id"];
42
43        // Redirect to admin dashboard or any other page
44        header("Location: admin_dashboard.php");
45        exit();
46    } else {
47        echo "Invalid username or password";
48    }
49
50    // Close connection
51    $conn->close();
52 }
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Admin Login</title>
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <style>
        body {
            background-color: #2C3E50; /* Set the background color */
            background-image: linear-gradient(135deg, #3498DB, #2980B9); /* Add a gradient effect */
            background-size: 400% 400%; /* Set the size of the gradient */
            animation: gradient 10s ease infinite; /* Animate the gradient */
        }

        @keyframes gradient {
            0% { background-position: 0% 50%; }
            50% { background-position: 100% 50%; }
            100% { background-position: 0% 50%; }
        }

        .container {
            background-color: rgba(255, 255, 255, 0.8); /* Set a semi-transparent white background */
            padding: 20px;
            border-radius: 10px;
            box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.3); /* Add a shadow effect */
        }
    </style>
    <!-- Add your custom CSS if needed -->
</head>
<body>
    <div class="container">
        <h2>Admin Login</h2>
        <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
            <div class="form-group">
                <label for="username">Username:</label>
                <input type="text" class="form-control" id="username" name="username" required>
            </div>
    


```

```

C:\Users\saarva\Desktop\UW\admin_login.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
admin_dashboard.php admin_logout.php admin_login.php admin_signin.php donor_process_request.php donor_requests.php org_process_request.php organization_requests.php
61 <style>
62   body {
63     background-color: #2C3E50; /* Set the background color */
64     background-image: linear-gradient(135deg, #3498db, #2980b9); /* Add a gradient effect */
65     background-size: 400% 400%; /* Set the size of the gradient */
66     animation: gradient 10s ease infinite; /* Animate the gradient */
67   }
68
69   @keyframes gradient {
70     0% { background-position: 0% 50%; }
71     50% { background-position: 100% 50%; }
72     100% { background-position: 0% 50%; }
73   }
74
75   .container {
76     background-color: rgba(255, 255, 255, 0.8); /* Set a semi-transparent white background */
77     padding: 20px;
78     border-radius: 10px;
79     box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.3); /* Add a shadow effect */
80   }
81
82   <!-- Add your custom CSS if needed -->
83
84 </body>
85
86 <div class="container">
87   <h2>Admin Login</h2>
88   <form method="post" action="php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?&gt;"&gt;
89     &lt;div class="form-group"&gt;
90       &lt;label for="username"&gt;Username:&lt;/label&gt;
91       &lt;input type="text" class="form-control" id="username" name="username" required&gt;
92     &lt;/div&gt;
93     &lt;div class="form-group"&gt;
94       &lt;label for="password"&gt;Password:&lt;/label&gt;
95       &lt;input type="password" class="form-control" id="password" name="password" required&gt;
96     &lt;/div&gt;
97     &lt;button type="submit" class="btn btn-primary"&gt;Login&lt;/button&gt;
98   &lt;/form&gt;
99 &lt;/div&gt;
100
101 &lt;script src="js/jquery.js"&gt;&lt;/script&gt;
102 &lt;script src="js/bootstrap.min.js"&gt;&lt;/script&gt;
103 &lt;!-- Add your custom JavaScript if needed --&gt;
104
105 &lt;/body&gt;
&lt;/html&gt;
</pre

PHP Hypertext Preprocessor file



length: 3,496 lines: 105 Ln: 33 Col: 5 Pos: 927 Windows (CR LF) UTF-8 INS



22:00 26/04/2024


```

2. Admin_logout.php

```

C:\Users\saarva\Desktop\UW\admin_logout.php - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
admin_dashboard.php admin_logout.php admin_login.php admin_signin.php donor_process_request.php donor_requests.php org_process_request.php organization_requests.php
1 <?php
2   // Start session
3   session_start();
4
5   // Destroy session
6   session_unset();
7   session_destroy();
8
9   // Redirect to login page
10  header("Location: admin_login.php");
11  exit;
12 ?>

```

PHP Hypertext Preprocessor file

length: 179 lines: 12 Ln: 1 Col: 1 Pos: 1 Windows (CR LF) UTF-8 INS

22:00 26/04/2024

3. Organization Process Request (org_process_request.php)

```

<?php
require_once("DBconnect.php");

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (isset($_POST['approve'])) {
        $org_id = $_POST['org_id'];

        $sql = "UPDATE organization SET status = 'Approved' WHERE org_id = '$org_id'";
        if (mysqli_query($conn, $sql)) {
            header("Location: organization_requests.php");
            exit();
        } else {
            echo "Error updating record: " . mysqli_error($conn);
        }
    }

    if (isset($_POST['reject'])) {
        $org_id = $_POST['org_id'];

        $sql = "DELETE FROM organization WHERE org_id = '$org_id'";
        if (mysqli_query($conn, $sql)) {
            header("Location: organization_requests.php");
            exit();
        } else {
            echo "Error deleting record: " . mysqli_error($conn);
        }
    }
}
?>

```

4. Add_event: adding new events to the database.

```

<?php
// Include database connection
require_once('DBconnect.php');

// Handle form submission
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Debugging: Check what data is being sent via the form
    var_dump($_POST); // Temporary debugging statement

    // Retrieve form data
    $event_id = $_POST['event_id'];
    $date = $_POST['date'];
    $name = $_POST['name'];
    $capacity = $_POST['capacity'];
    $zip = $_POST['zip'];
    $house_no = $_POST['house'];
    $road_no = $_POST['road_no'];
    //address = $_POST['address'];
    $sql = "INSERT INTO 'events' ('event_id', 'date', 'name', 'capacity', 'zip', 'house', 'road_no') VALUES (?, ?, ?, ?, ?, ?, ?)";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("ssssss", $event_id, $date, $name, $capacity, $zip, $house, $road_no);

    if ($stmt->execute()) {
        echo "<p style='color:green;'>Event added Successfully.</p>";
        header("Location: admin_dashboard.php?success=true");
        exit();
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
} else {
    header("Location: donor_register.php?error=missing_fields");
    exit();
}
?>

```

Source Code Repository

Google Drive link:

<https://drive.google.com/drive/folders/164mwybQl5jHwFQA2PrVZ5K76YUiO4FIK>.

Conclusion

In conclusion, Unity Wave stands as a dynamic and inclusive platform that harnesses the power of community engagement and empowers individuals and groups to contribute meaningfully to social causes. Our website prioritizes user experience through secure registration and authentication, guaranteeing personalized interactions for enhancing community engagement and empowerment. Unity Wave embodies the spirit of collaboration and creativity, creating a virtual space where passion meets purpose. Through this project, we have not only met the needs of our users but have also cultivated a vibrant community committed to making a positive impact.

