Technical Document for Personal Task Manager Prototype
Website: https://infost790s1.uwmsois.com/

Introduction:

This technical document describes the development of a personal task manager prototype, which is a web-based application developed to help users manage their daily tasks and events. The system was developed using modern web development tools and technologies, including HTML and CSS. The goal of this document is to provide a comprehensive understanding of the system's architecture, design, and functionality.

System Architecture:

The system architecture is designed as a client-server model, where the client is the web browser, and the server is a PHP server. The web pages are developed using HTML, CSS, and JavaScript, which are then executed by the browser. The browser communicates with the server using HTTP requests, and the server sends back the requested data as an HTTP response.

System Design:

The system's design is based on a Model-View-Controller (MVC) architecture, which separates the system into three components: the Model, the View, and the Controller. The Model component represents the data and business logic of the system, the View component is responsible for rendering the user interface, and the Controller component handles user input and interacts with the Model and View components.

System Functionality:

The system provides users with the ability to create, edit, and delete tasks and events, which can be organized by date and time. The system also allows users to set task priorities, and due dates. The user interface is designed to be intuitive and easy to use, with drag-and-drop functionality for rearranging tasks and events. Additionally, the system includes a search function that allows users to search for specific tasks and events.

Functionality:

The Personal Task Manager Prototype offers the following functionality to users:

1. User registration and login: The application allows users to create an account and login using their email and password.
2. Task creation: Users can create tasks by entering a title, description, due date, priority, and category. The application also allows users to edit or delete tasks.

3. Task filtering and sorting: The application allows users to filter tasks by category or priority and sort them by due date.
4. User management: The application allows superusers to manage users, including creating new accounts, updating user information, and deleting accounts.

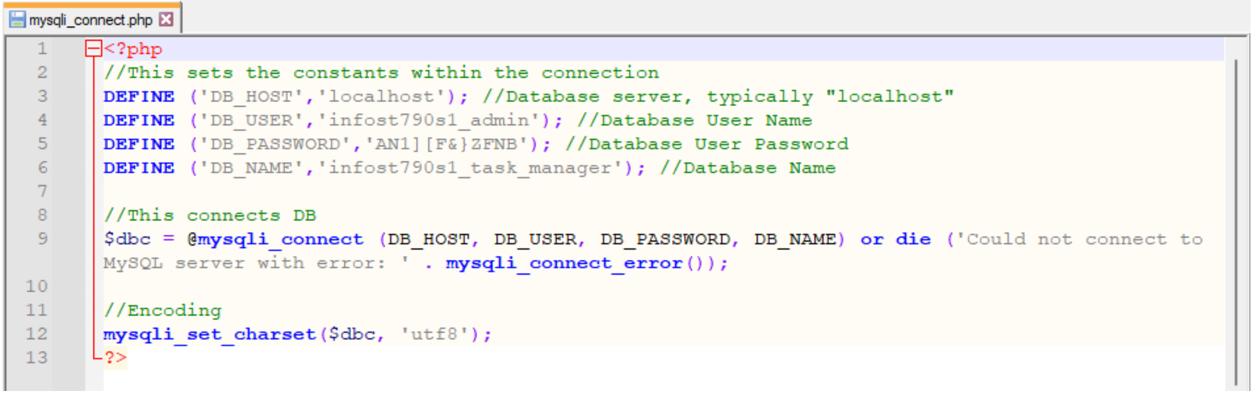
Technical Details:

The Personal Task Manager Prototype is developed using various technologies, including:

1. HTML: Used for creating the structure and content of the web pages.
2. CSS: Used for styling the web pages and making them visually appealing.
3. MySQL: Used for storage of data and data access through SQL queries.

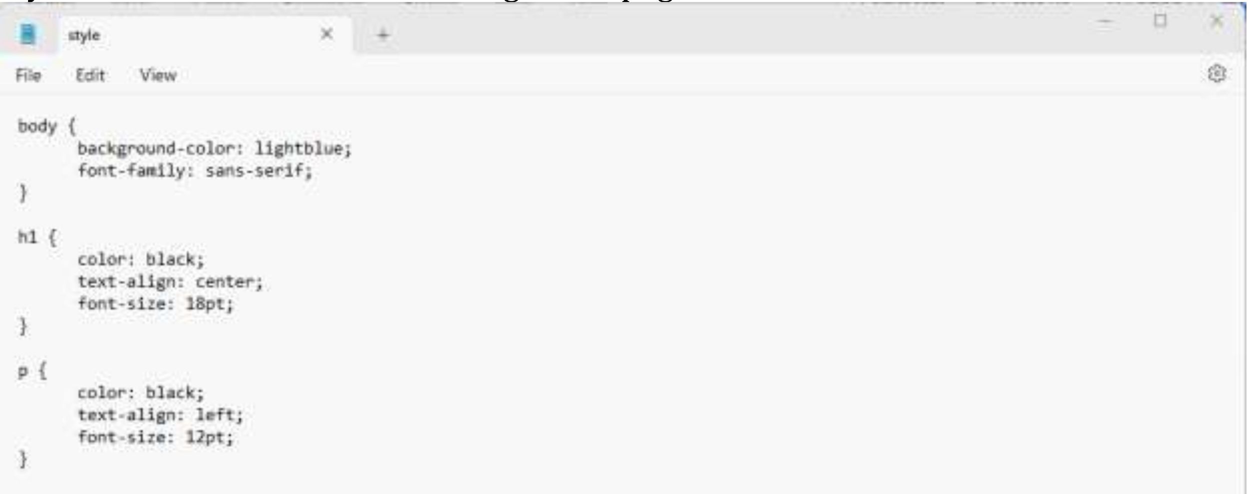
Code:

mysqli_connect.php: Connects the database with the website pages.



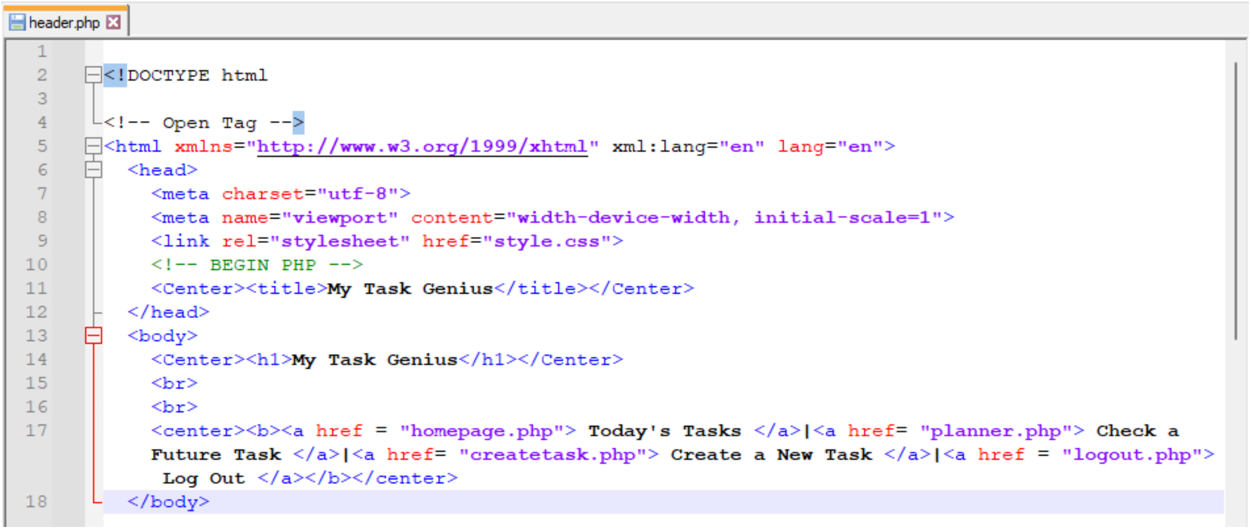
```php
<?php
//This sets the constants within the connection
DEFINE ('DB_HOST','localhost'); //Database server, typically "localhost"
DEFINE ('DB_USER','infost790s1_admin'); //Database User Name
DEFINE ('DB_PASSWORD','AN1][F&}ZFNB'); //Database User Password
DEFINE ('DB_NAME','infost790s1_task_manager'); //Database Name

//This connects DB
$dbc = @mysqli_connect (DB_HOST, DB_USER, DB_PASSWORD, DB_NAME) or die ('Could not connect to MySQL server with error: ' . mysqli_connect_error());

//Encoding
mysqli_set_charset($dbc, 'utf8');
?>
```

Style.css: Provides basic formatting to the pages.



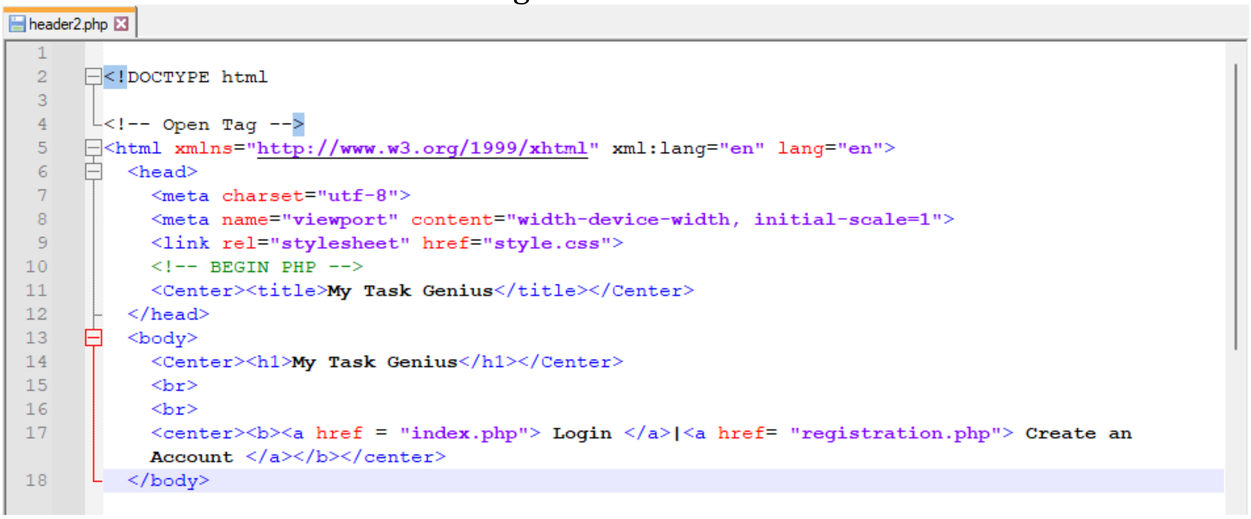
```css
body {
	background-color: lightblue;
	font-family: sans-serif;
}

h1 {
	color: black;
	text-align: center;
	font-size: 18pt;
}

p {
	color: black;
	text-align: left;
	font-size: 12pt;
}
```

header.php: General header applied to pages once the user is logged in. This provides links to several other pages within the program.



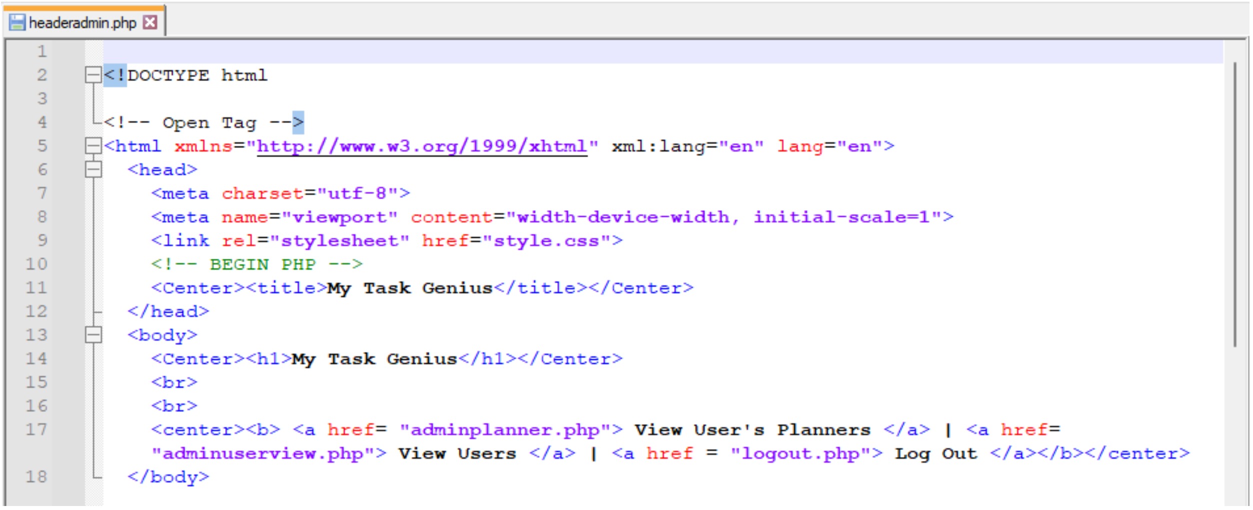
```
<!DOCTYPE html

<!-- Open Tag -->
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width-device-width, initial-scale=1">
    <link rel="stylesheet" href="style.css">
    <!-- BEGIN PHP -->
    <Center><title>My Task Genius</title></Center>
  </head>
  <body>
    <Center><h1>My Task Genius</h1></Center>
    <br>
    <br>
    <center><b><a href = "homepage.php"> Today's Tasks </a>|<a href= "planner.php"> Check a Future Task </a>|<a href= "createtask.php"> Create a New Task </a>|<a href = "logout.php"> Log Out </a></b></center>
  </body>
```

header2.php: General header for an individual who is not signed in. This provides links for account creation and to login.



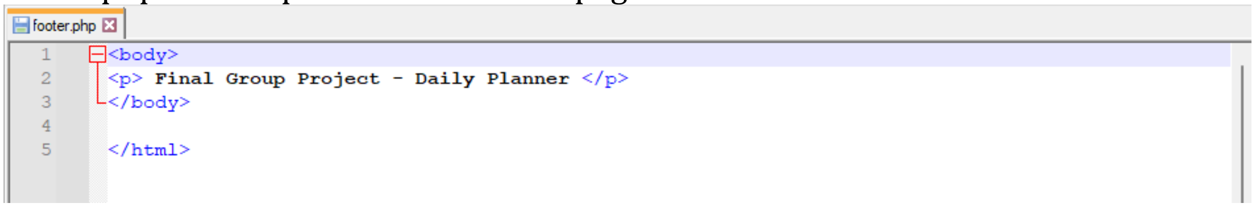
```
<!DOCTYPE html

<!-- Open Tag -->
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width-device-width, initial-scale=1">
    <link rel="stylesheet" href="style.css">
    <!-- BEGIN PHP -->
    <Center><title>My Task Genius</title></Center>
  </head>
  <body>
    <Center><h1>My Task Genius</h1></Center>
    <br>
    <br>
    <center><b><a href = "index.php"> Login </a>|<a href= "registration.php"> Create an Account </a></b></center>
  </body>
```

headeradmin.php: General header for when an individual is logged into an administrator account. Provides links to various administrator functions.



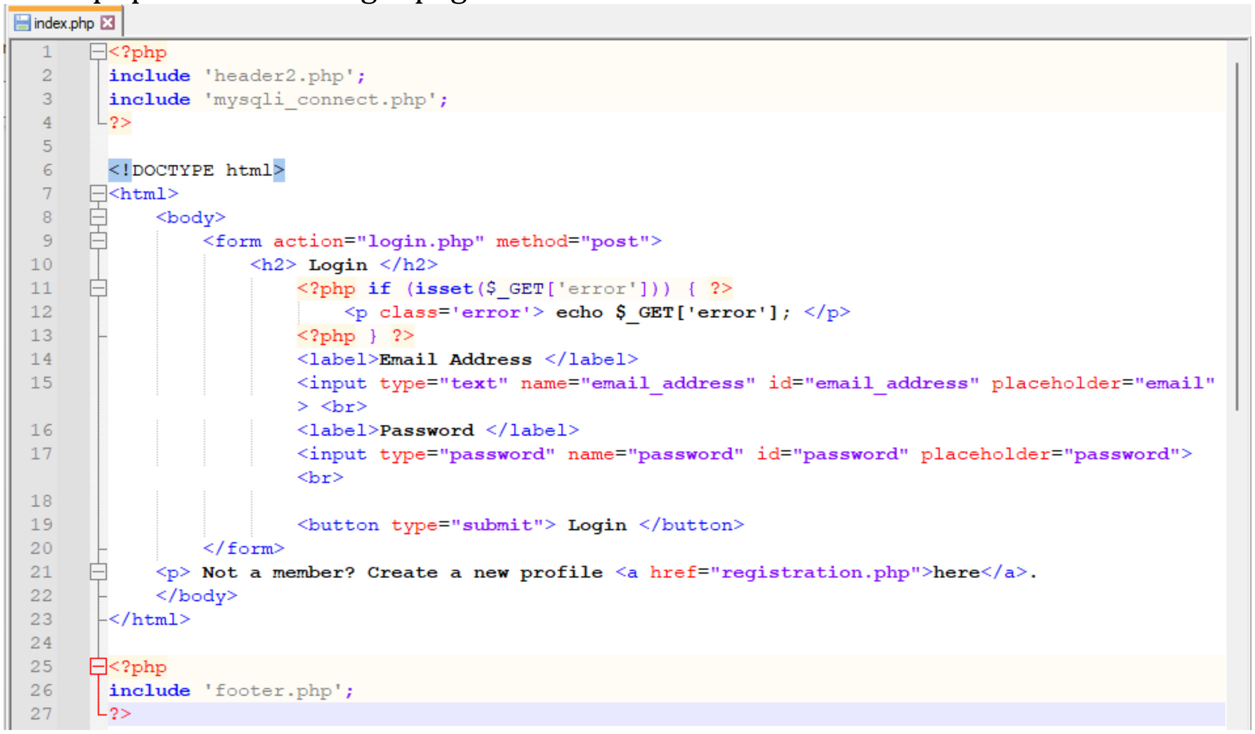
```
<!DOCTYPE html

<!-- Open Tag -->
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width-device-width, initial-scale=1">
    <link rel="stylesheet" href="style.css">
    <!-- BEGIN PHP -->
    <Center><title>My Task Genius</title></Center>
  </head>
  <body>
    <Center><h1>My Task Genius</h1></Center>
    <br>
    <br>
    <center><b> <a href= "adminplanner.php"> View User's Planners </a> | <a href= "adminuserview.php"> View Users </a> | <a href = "logout.php"> Log Out </a></b></center>
  </body>
```

footer.php: Footer provided to all web pages.



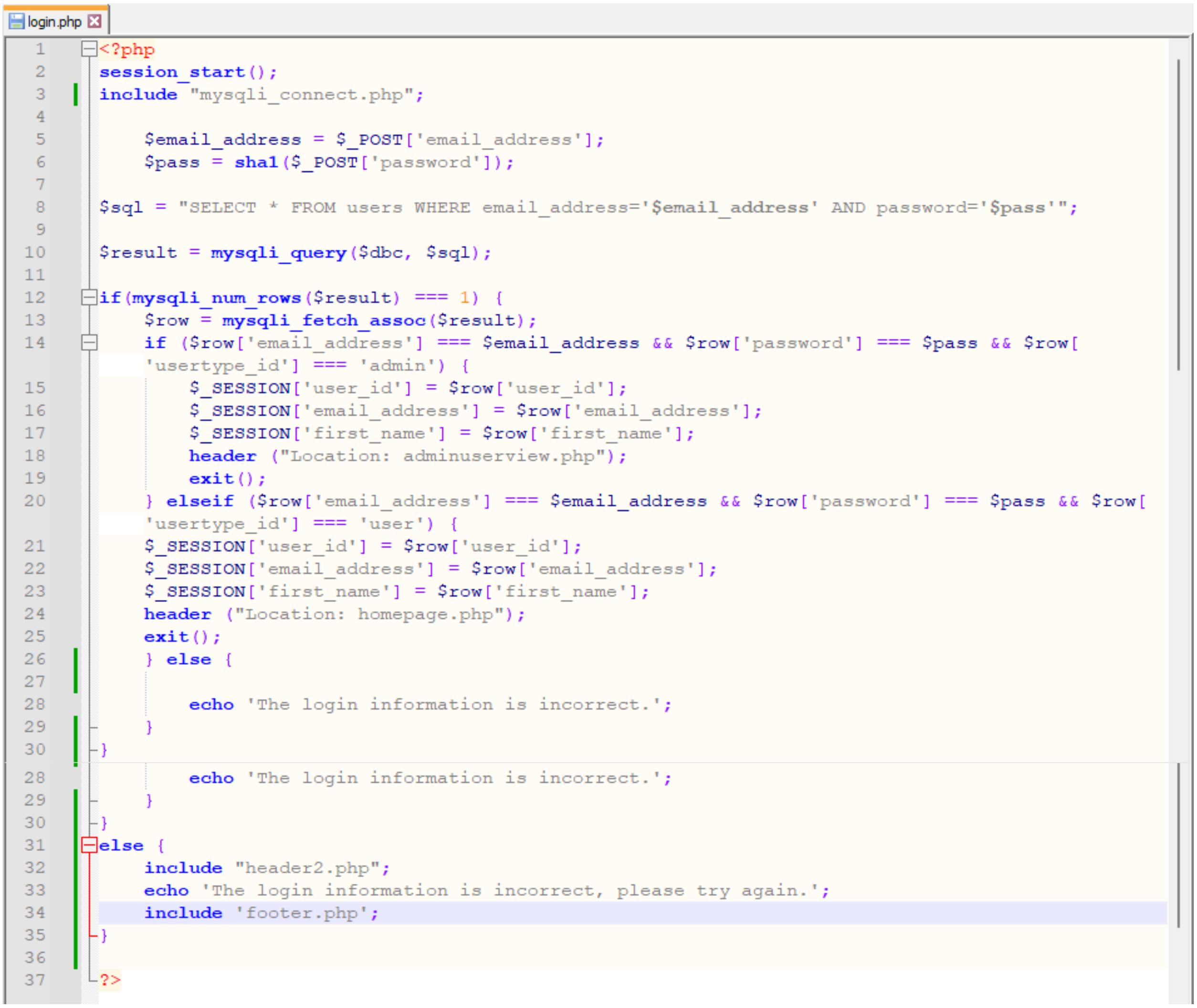
```
<body>
<p> Final Group Project - Daily Planner </p>
</body>

</html>
```

index.php: Acts as the login page for all users.



```
<?php
include 'header2.php';
include 'mysqli_connect.php';
?>

<!DOCTYPE html>
<html>
    <body>
        <form action="login.php" method="post">
            <h2> Login </h2>
                <?php if (isset($_GET['error'])) { ?>
                    <p class='error'> echo $_GET['error']; </p>
                <?php } ?>
                <label>Email Address </label>
                <input type="text" name="email_address" id="email_address" placeholder="email" > <br>
                <label>Password </label>
                <input type="password" name="password" id="password" placeholder="password"> <br>

                <button type="submit"> Login </button>
        </form>
    <p> Not a member? Create a new profile <a href="registration.php">here</a>.
    </body>
</html>

<?php
include 'footer.php';
?>
```

login.php: Passes login information from index.php to database and creates session.

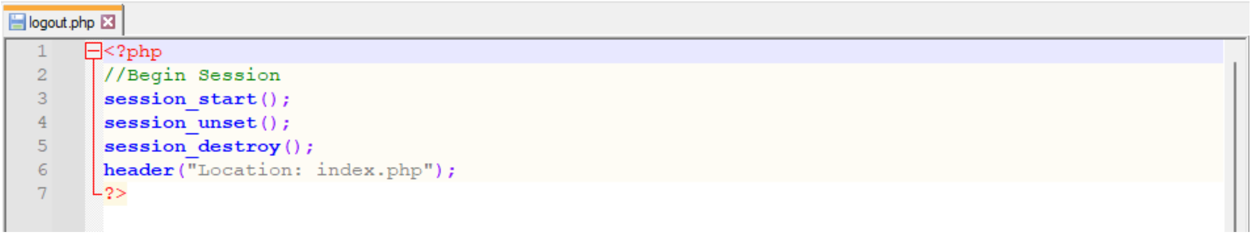


```php
<?php
session_start();
include "mysqli_connect.php";

    $email_address = $_POST['email_address'];
    $pass = sha1($_POST['password']);

$sql = "SELECT * FROM users WHERE email_address='$email_address' AND password='$pass'";

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

if(mysqli_num_rows($result) === 1) {
    $row = mysqli_fetch_assoc($result);
    if ($row['email_address'] === $email_address && $row['password'] === $pass && $row['usertype_id'] === 'admin') {
        $_SESSION['user_id'] = $row['user_id'];
        $_SESSION['email_address'] = $row['email_address'];
        $_SESSION['first_name'] = $row['first_name'];
        header ("Location: adminuserview.php");
        exit();
    } elseif ($row['email_address'] === $email_address && $row['password'] === $pass && $row['usertype_id'] === 'user') {
    $_SESSION['user_id'] = $row['user_id'];
    $_SESSION['email_address'] = $row['email_address'];
    $_SESSION['first_name'] = $row['first_name'];
    header ("Location: homepage.php");
    exit();
    } else {

        echo 'The login information is incorrect.';
    }
}
```

```php
        echo 'The login information is incorrect.';
    }
}
```

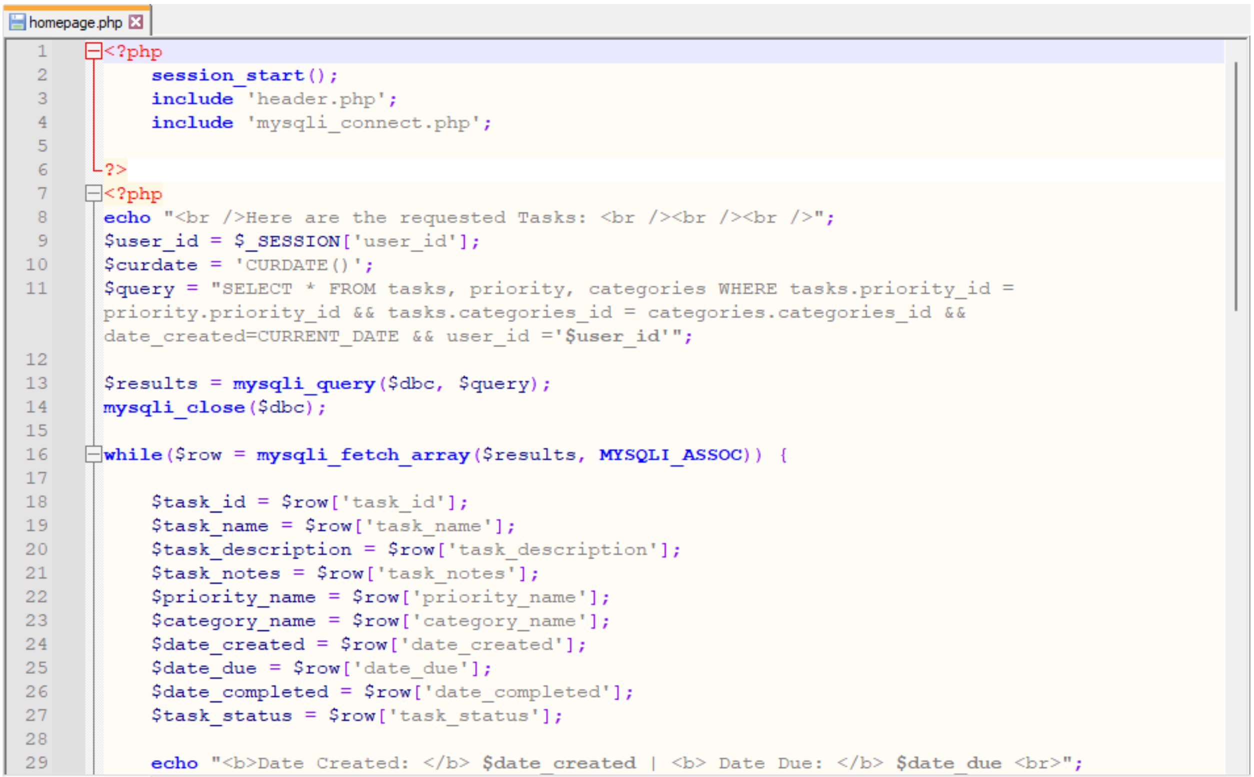
```php
else {
    include "header2.php";
    echo 'The login information is incorrect, please try again.';
    include 'footer.php';
}

?>
```

logout.php: Ends user session.

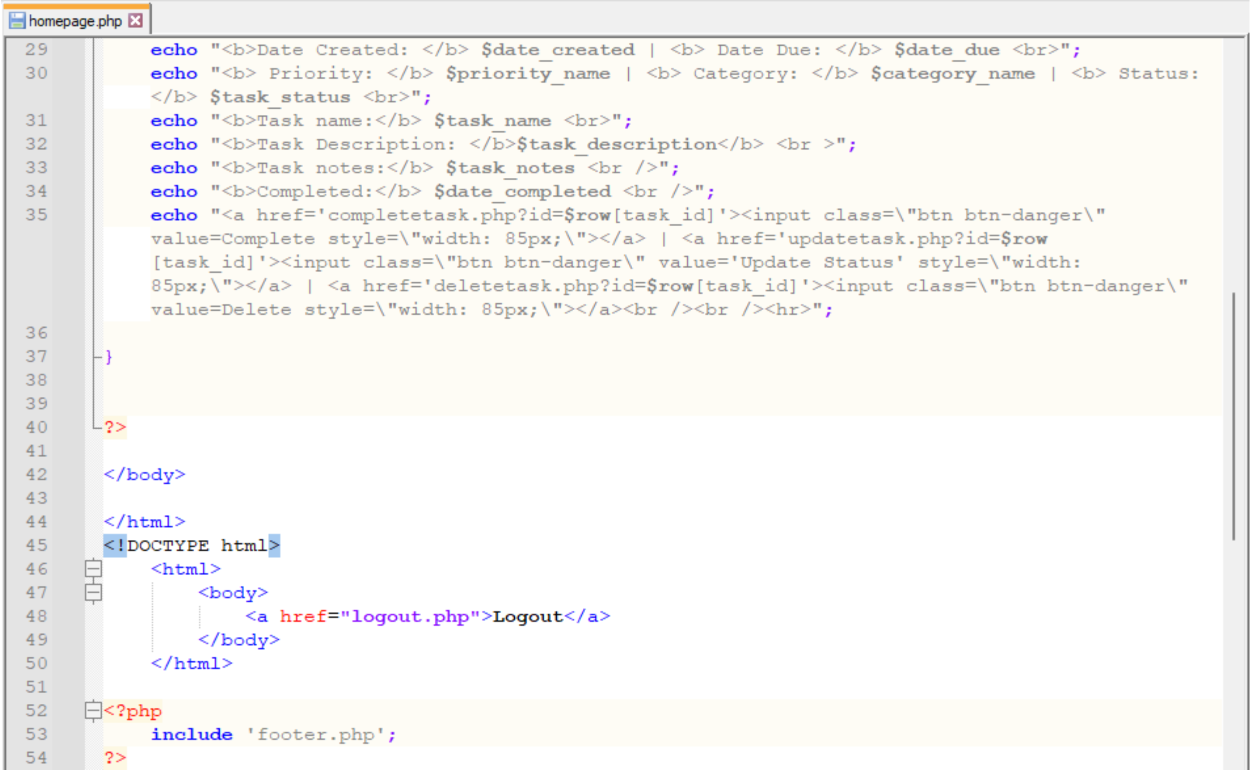


```php
<?php
//Begin Session
session_start();
session_unset();
session_destroy();
header("Location: index.php");
?>
```

homepage.php: Acts as the homepage for users. Shows the current day’s tasks.

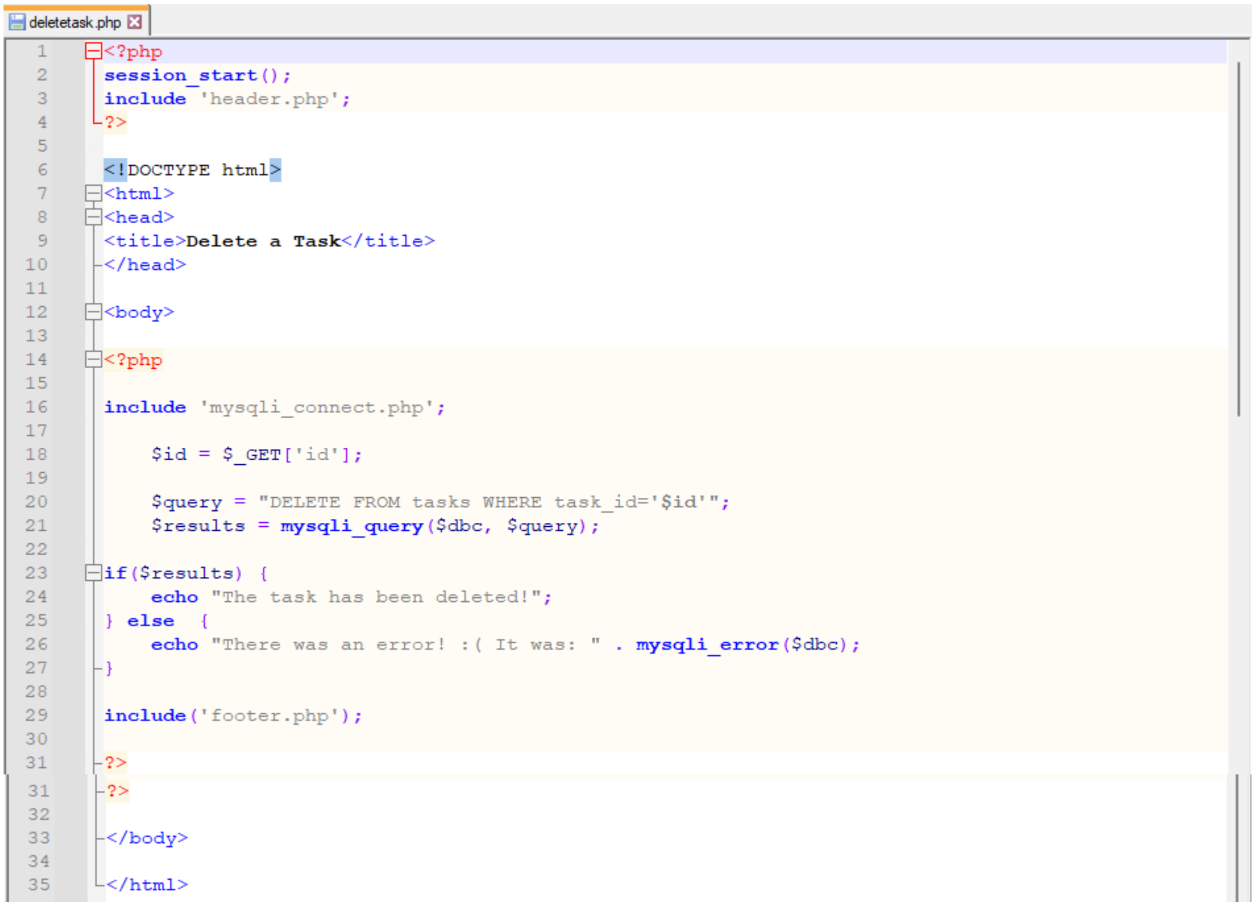


```php
<?php
    session_start();
    include 'header.php';
    include 'mysqli_connect.php';

?>
<?php
echo "<br />Here are the requested Tasks: <br /><br /><br />";
$user_id = $_SESSION['user_id'];
$curdate = 'CURDATE()';
$query = "SELECT * FROM tasks, priority, categories WHERE tasks.priority_id = priority.priority_id && tasks.categories_id = categories.categories_id && date_created=CURRENT_DATE && user_id ='$user_id'";

$results = mysqli_query($dbc, $query);
mysqli_close($dbc);

while($row = mysqli_fetch_array($results, MYSQLI_ASSOC)) {

    $task_id = $row['task_id'];
    $task_name = $row['task_name'];
    $task_description = $row['task_description'];
    $task_notes = $row['task_notes'];
    $priority_name = $row['priority_name'];
    $category_name = $row['category_name'];
    $date_created = $row['date_created'];
    $date_due = $row['date_due'];
    $date_completed = $row['date_completed'];
    $task_status = $row['task_status'];

    echo "<b>Date Created: </b> $date_created | <b> Date Due: </b> $date_due <br>";
```



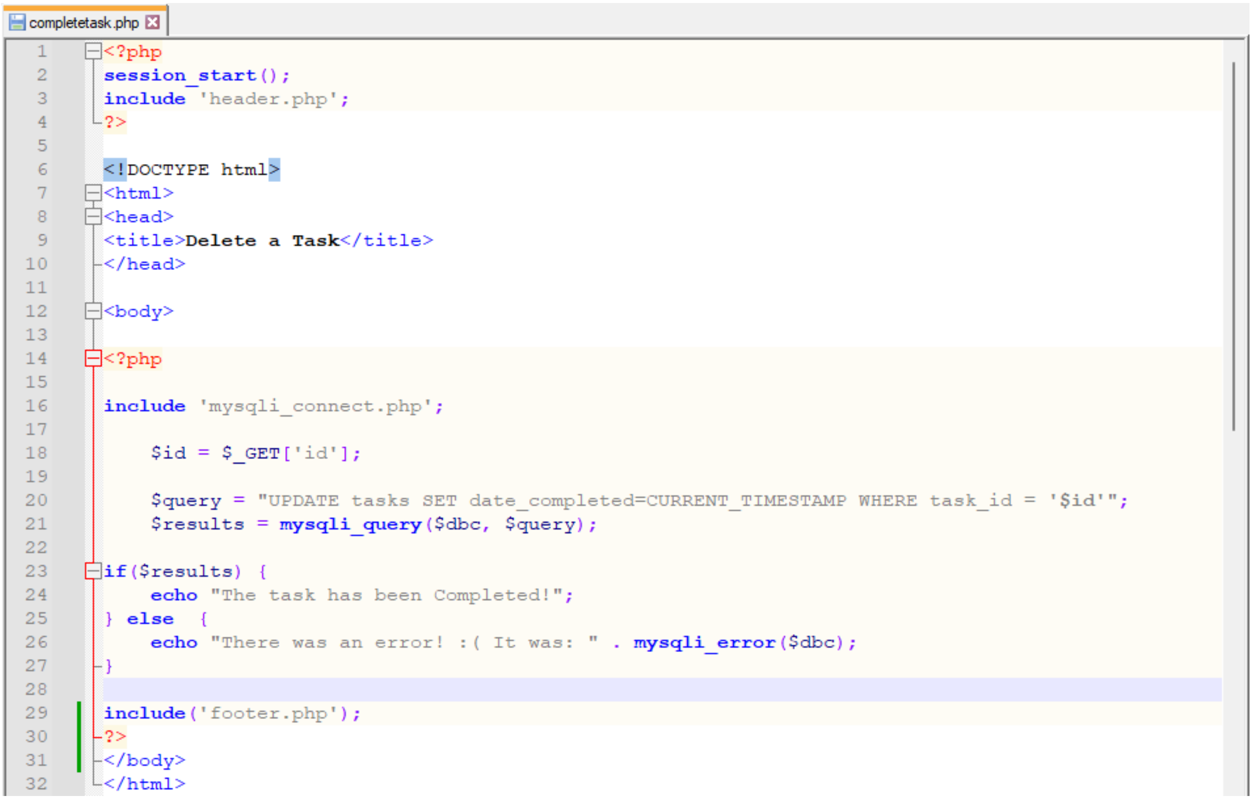
```php
    echo "<b>Date Created: </b> $date_created | <b> Date Due: </b> $date_due <br>";
    echo "<b> Priority: </b> $priority_name | <b> Category: </b> $category_name | <b> Status: </b> $task_status <br>";
    echo "<b>Task name:</b> $task_name <br>";
    echo "<b>Task Description: </b>$task_description</b> <br >";
    echo "<b>Task notes:</b> $task_notes <br />";
    echo "<b>Completed:</b> $date_completed <br />";
    echo "<a href='completetask.php?id=$row[task_id]'><input class=\"btn btn-danger\" value=Complete style=\"width: 85px;\"></a> | <a href='updatetask.php?id=$row[task_id]'><input class=\"btn btn-danger\" value='Update Status' style=\"width: 85px;\"></a> | <a href='deletetask.php?id=$row[task_id]'><input class=\"btn btn-danger\" value=Delete style=\"width: 85px;\"></a><br /><br /><hr>";

}



?>

</body>

</html>
<!DOCTYPE html>
    <html>
        <body>
            <a href="logout.php">Logout</a>
        </body>
    </html>

<?php
    include 'footer.php';
?>
```

deletetask.php: Deletes the selected task.



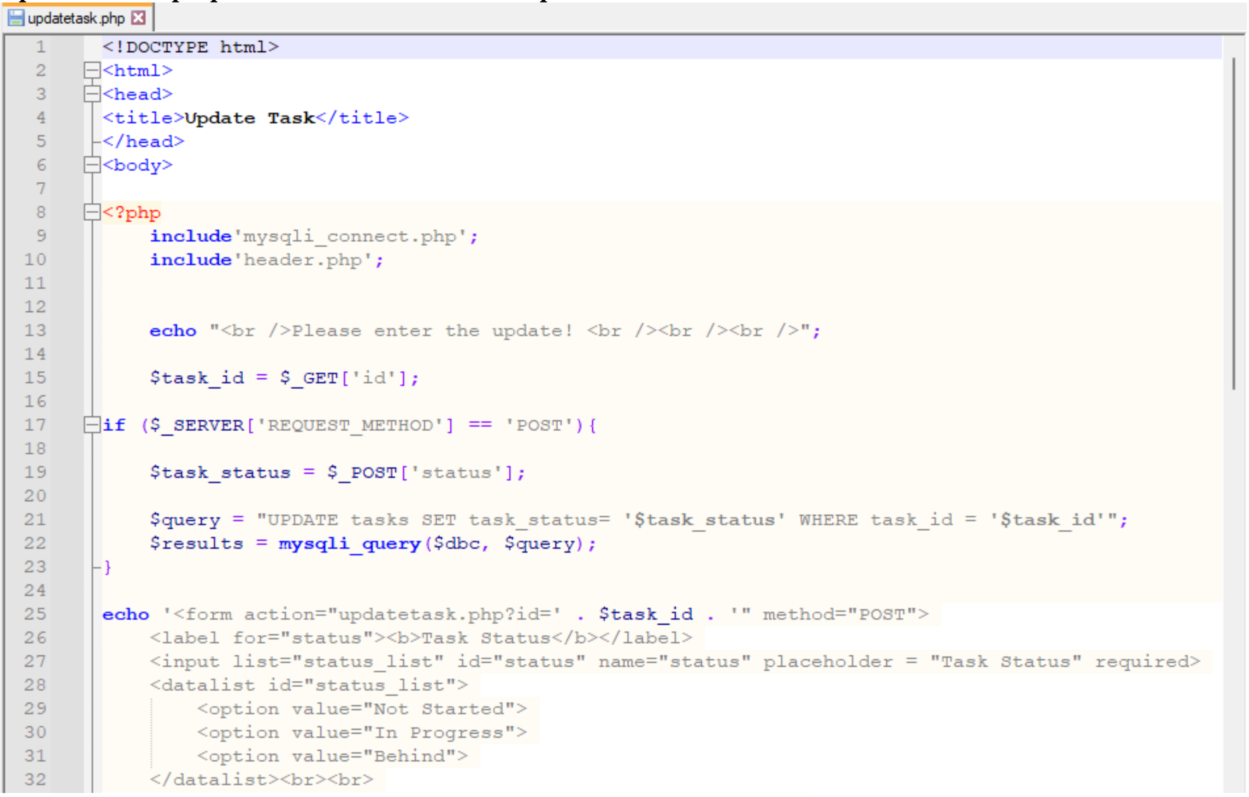
```
<?php
session_start();
include 'header.php';
?>

<!DOCTYPE html>
<html>
<head>
<title>Delete a Task</title>
</head>

<body>

<?php

include 'mysqli_connect.php';

    $id = $_GET['id'];

    $query = "DELETE FROM tasks WHERE task_id='$id'";
    $results = mysqli_query($dbc, $query);

if($results) {
    echo "The task has been deleted!";
} else  {
    echo "There was an error! :( It was: " . mysqli_error($dbc);
}

include('footer.php');

?>
?>

</body>

</html>
```

completetask.php: Adds the current date and time to the selected task completed field.

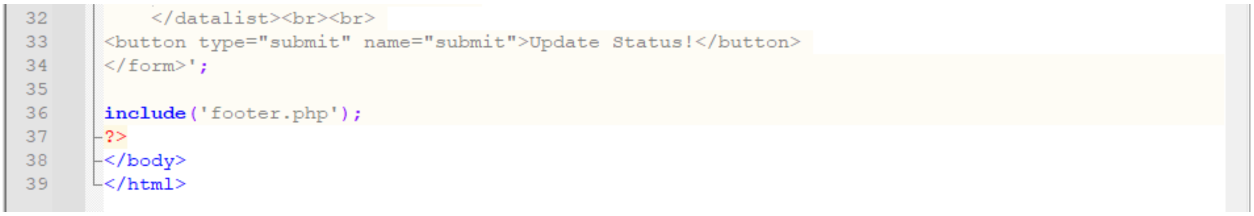


```php
<?php
session_start();
include 'header.php';
?>

<!DOCTYPE html>
<html>
<head>
<title>Delete a Task</title>
</head>

<body>

<?php

include 'mysqli_connect.php';

    $id = $_GET['id'];

    $query = "UPDATE tasks SET date_completed=CURRENT_TIMESTAMP WHERE task_id = '$id'";
    $results = mysqli_query($dbc, $query);

if($results) {
    echo "The task has been Completed!";
} else  {
    echo "There was an error! :( It was: " . mysqli_error($dbc);
}

include('footer.php');
?>
</body>
</html>
```

updatetask.php: Allows the user to update the status of the selected task.

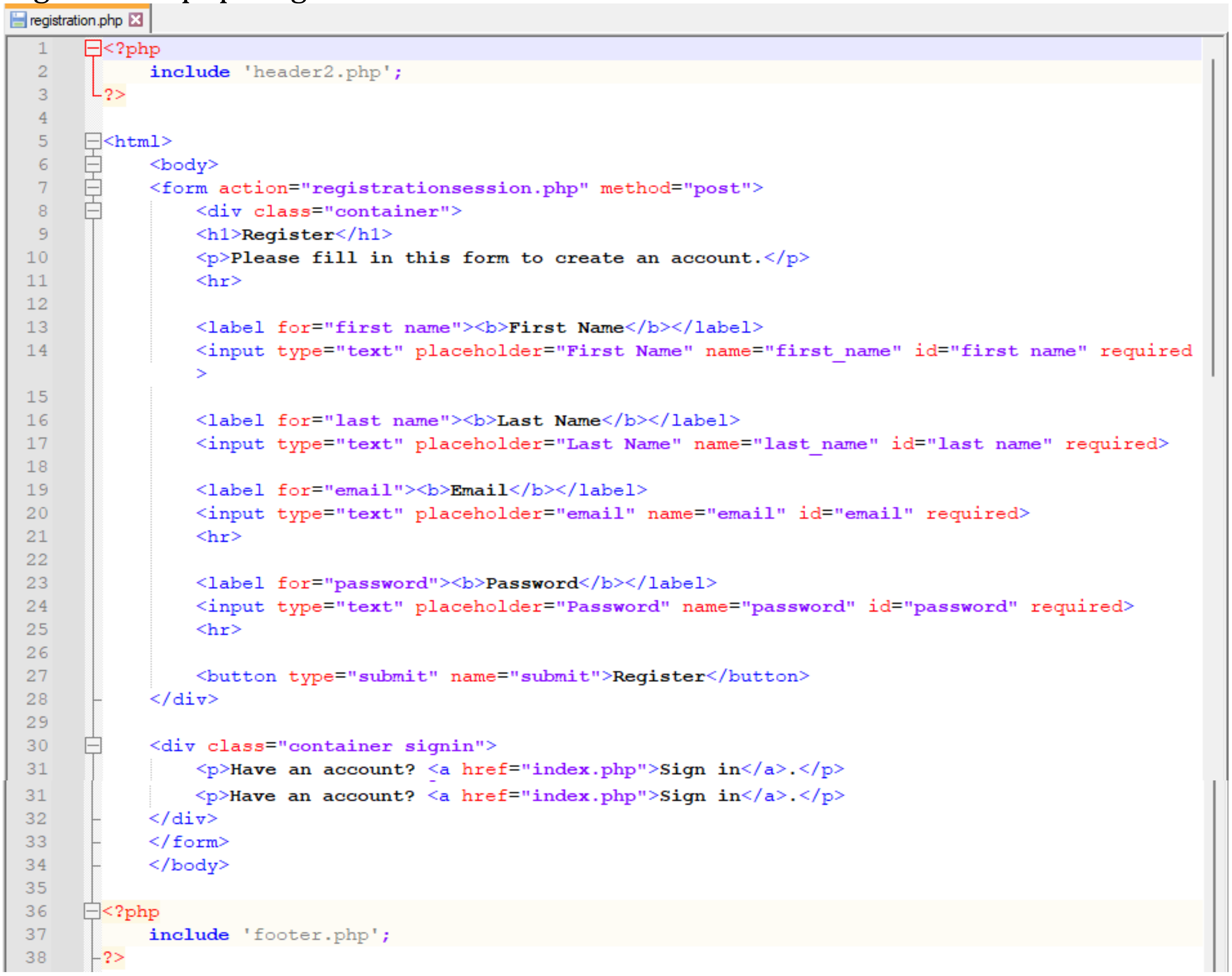


```php
<!DOCTYPE html>
<html>
<head>
<title>Update Task</title>
</head>
<body>

<?php
    include'mysqli_connect.php';
    include'header.php';


    echo "<br />Please enter the update! <br /><br /><br />";

    $task_id = $_GET['id'];

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

    $task_status = $_POST['status'];

    $query = "UPDATE tasks SET task_status= '$task_status' WHERE task_id = '$task_id'";
    $results = mysqli_query($dbc, $query);
}

echo '<form action="updatetask.php?id=' . $task_id . '" method="POST">
    <label for="status"><b>Task Status</b></label>
    <input list="status_list" id="status" name="status" placeholder = "Task Status" required>
    <datalist id="status_list">
        <option value="Not Started">
        <option value="In Progress">
        <option value="Behind">
    </datalist><br><br>
```



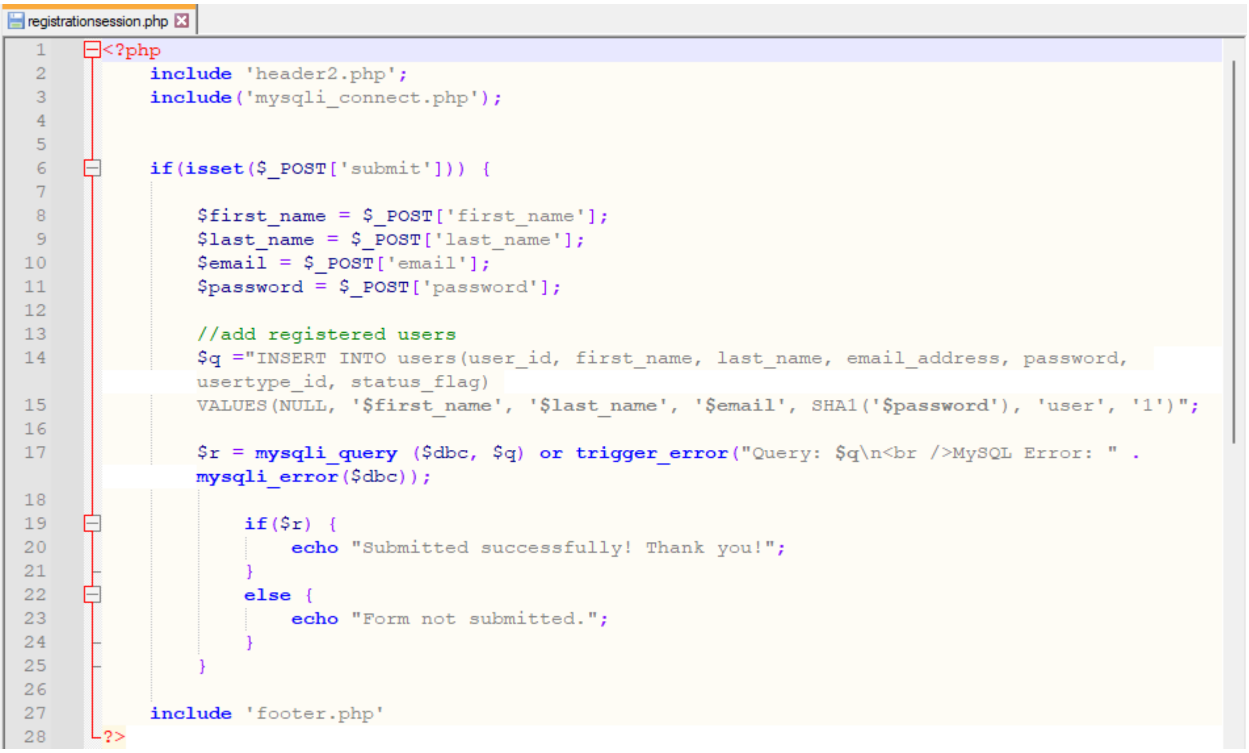
```
    </datalist><br><br>
<button type="submit" name="submit">Update Status!</button>
</form>';

include('footer.php');
?>
</body>
</html>
```

registration.php: Page to create a new user account.



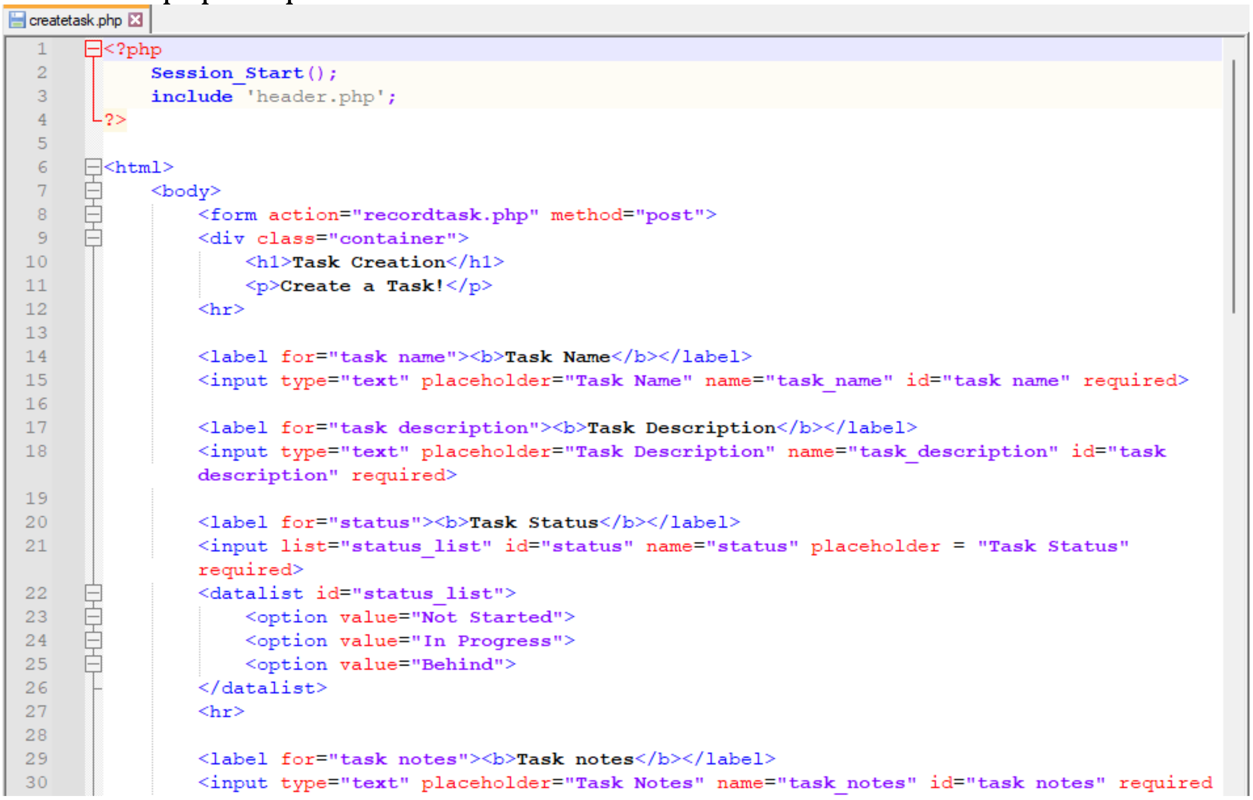
```
<?php
    include 'header2.php';
?>

<html>
    <body>
    <form action="registrationsession.php" method="post">
        <div class="container">
        <h1>Register</h1>
        <p>Please fill in this form to create an account.</p>
        <hr>

        <label for="first name"><b>First Name</b></label>
        <input type="text" placeholder="First Name" name="first_name" id="first name" required
        >

        <label for="last name"><b>Last Name</b></label>
        <input type="text" placeholder="Last Name" name="last_name" id="last name" required>

        <label for="email"><b>Email</b></label>
        <input type="text" placeholder="email" name="email" id="email" required>
        <hr>

        <label for="password"><b>Password</b></label>
        <input type="text" placeholder="Password" name="password" id="password" required>
        <hr>

        <button type="submit" name="submit">Register</button>
    </div>

    <div class="container signin">
        <p>Have an account? <a href="index.php">Sign in</a>.</p>
        <p>Have an account? <a href="index.php">Sign in</a>.</p>
    </div>
    </form>
    </body>

<?php
    include 'footer.php';
?>
```

registrationsession.php: Passes information from registration.php to the database.

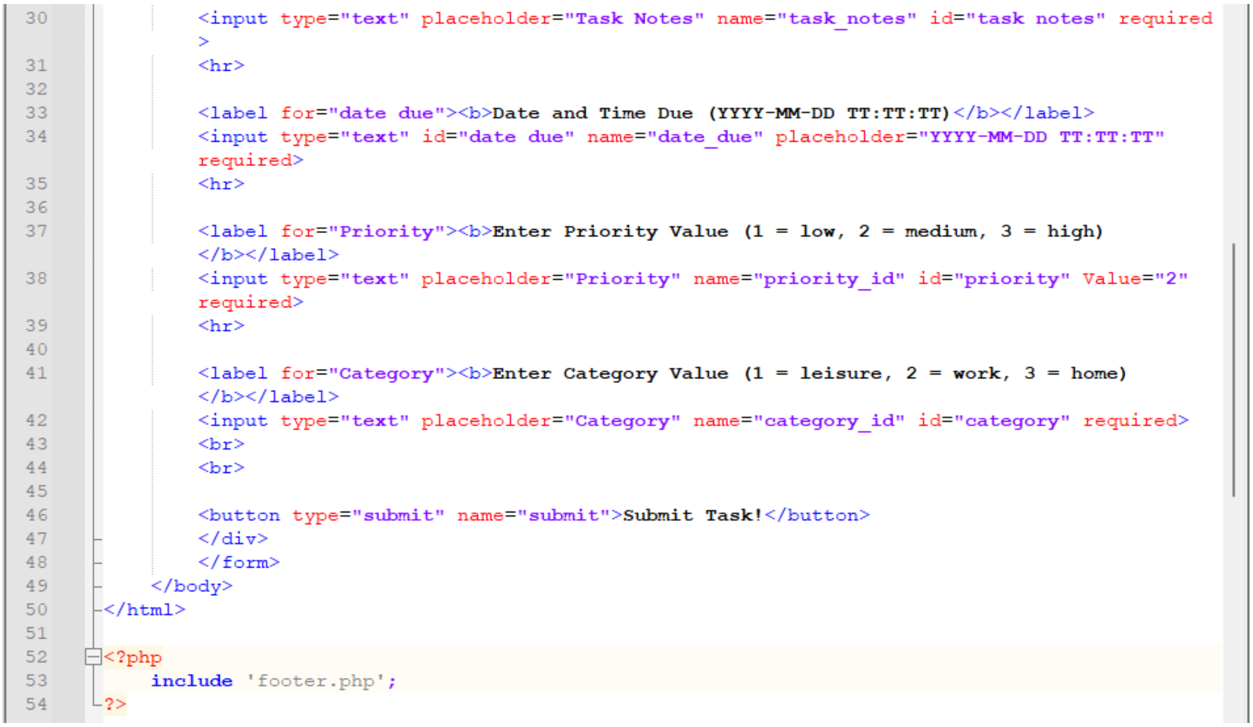


```
<?php
    include 'header2.php';
    include('mysqli_connect.php');


    if(isset($_POST['submit'])) {

        $first_name = $_POST['first_name'];
        $last_name = $_POST['last_name'];
        $email = $_POST['email'];
        $password = $_POST['password'];

        //add registered users
        $q ="INSERT INTO users(user_id, first_name, last_name, email_address, password,
        usertype_id, status_flag)
        VALUES(NULL, '$first_name', '$last_name', '$email', SHA1('$password'), 'user', '1')";

        $r = mysqli_query ($dbc, $q) or trigger_error("Query: $q\n<br />MySQL Error: " .
        mysqli_error($dbc));

            if($r) {
                echo "Submitted successfully! Thank you!";
            }
            else {
                echo "Form not submitted.";
            }
        }

    include 'footer.php'
?>
```

createtask.php: Requests information to create a new task.

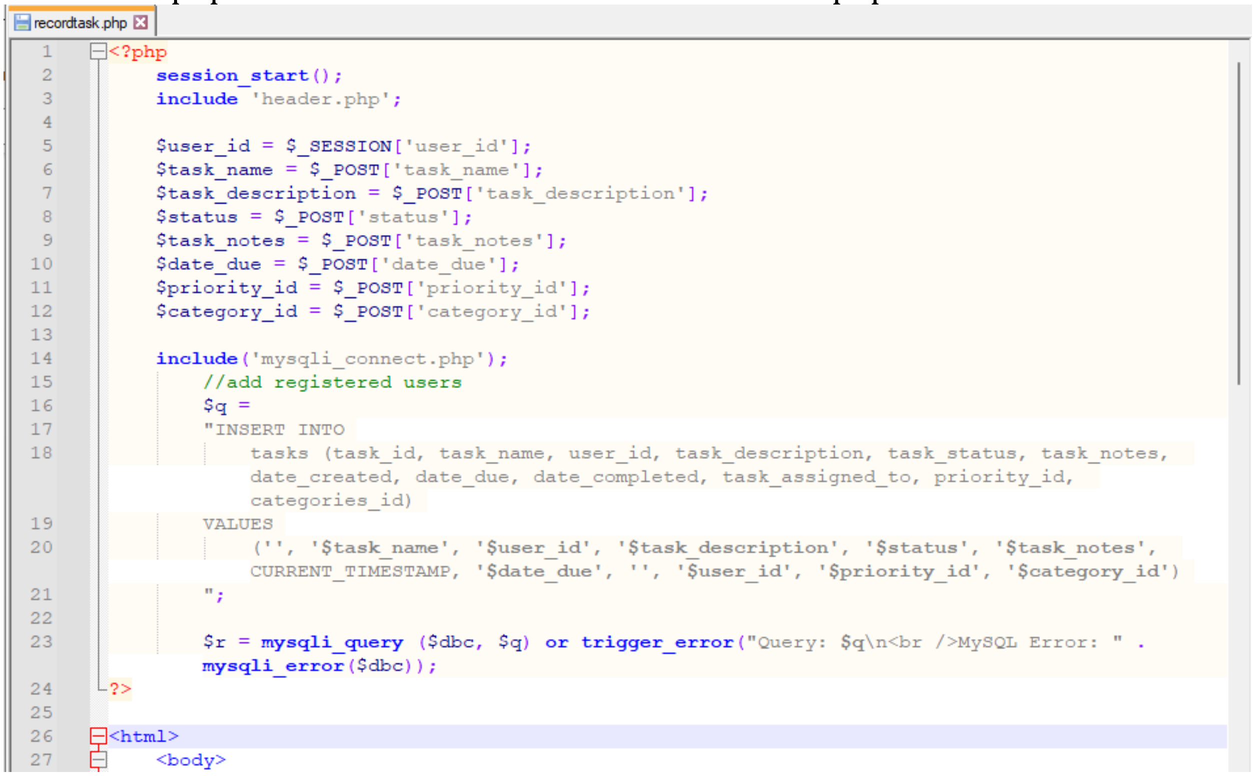


```
<?php
    Session_Start();
    include 'header.php';
?>

<html>
    <body>
        <form action="recordtask.php" method="post">
        <div class="container">
            <h1>Task Creation</h1>
            <p>Create a Task!</p>
        <hr>

        <label for="task name"><b>Task Name</b></label>
        <input type="text" placeholder="Task Name" name="task_name" id="task name" required>

        <label for="task description"><b>Task Description</b></label>
        <input type="text" placeholder="Task Description" name="task_description" id="task
        description" required>

        <label for="status"><b>Task Status</b></label>
        <input list="status_list" id="status" name="status" placeholder = "Task Status"
        required>
        <datalist id="status_list">
            <option value="Not Started">
            <option value="In Progress">
            <option value="Behind">
        </datalist>
        <hr>

        <label for="task notes"><b>Task notes</b></label>
        <input type="text" placeholder="Task Notes" name="task_notes" id="task notes" required
```

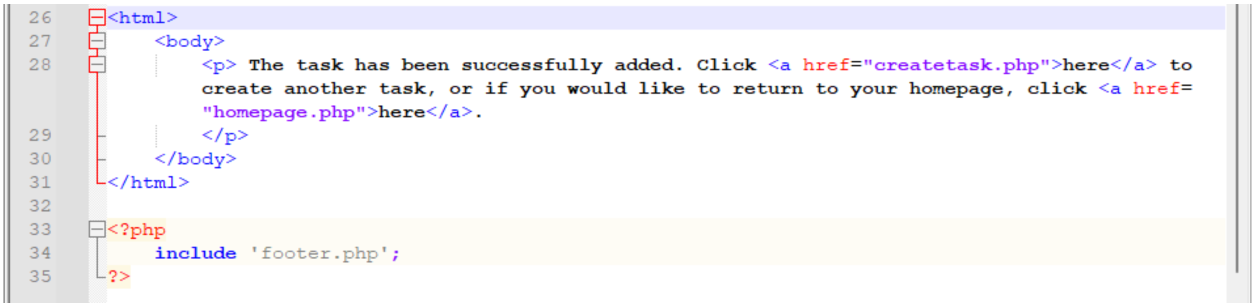


```
        <input type="text" placeholder="Task Notes" name="task_notes" id="task notes" required >
        <hr>

        <label for="date due"><b>Date and Time Due (YYYY-MM-DD TT:TT:TT)</b></label>
        <input type="text" id="date due" name="date_due" placeholder="YYYY-MM-DD TT:TT:TT" required>
        <hr>

        <label for="Priority"><b>Enter Priority Value (1 = low, 2 = medium, 3 = high)</b></label>
        <input type="text" placeholder="Priority" name="priority_id" id="priority" Value="2" required>
        <hr>

        <label for="Category"><b>Enter Category Value (1 = leisure, 2 = work, 3 = home)</b></label>
        <input type="text" placeholder="Category" name="category_id" id="category" required>
        <br>
        <br>

        <button type="submit" name="submit">Submit Task!</button>
        </div>
        </form>
    </body>
</html>

<?php
    include 'footer.php';
?>
```

recordtask.php: Passes the information from createtask.php to the database.

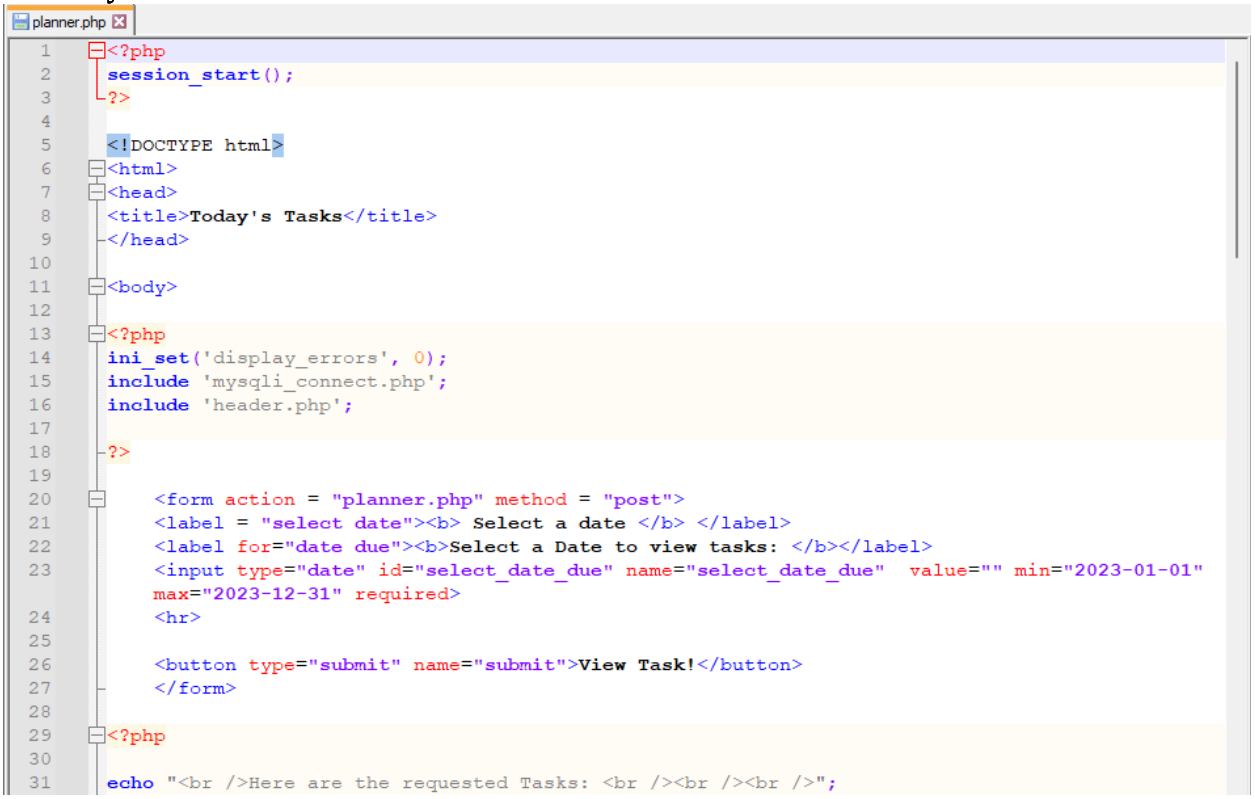


```
<?php
    session_start();
    include 'header.php';

    $user_id = $_SESSION['user_id'];
    $task_name = $_POST['task_name'];
    $task_description = $_POST['task_description'];
    $status = $_POST['status'];
    $task_notes = $_POST['task_notes'];
    $date_due = $_POST['date_due'];
    $priority_id = $_POST['priority_id'];
    $category_id = $_POST['category_id'];

    include('mysqli_connect.php');
        //add registered users
        $q =
        "INSERT INTO
            tasks (task_id, task_name, user_id, task_description, task_status, task_notes, date_created, date_due, date_completed, task_assigned_to, priority_id, categories_id)
        VALUES
            ('', '$task_name', '$user_id', '$task_description', '$status', '$task_notes', CURRENT_TIMESTAMP, '$date_due', '', '$user_id', '$priority_id', '$category_id')
        ";

        $r = mysqli_query ($dbc, $q) or trigger_error("Query: $q\n<br />MySQL Error: " . mysqli_error($dbc));
?>

<html>
    <body>
```

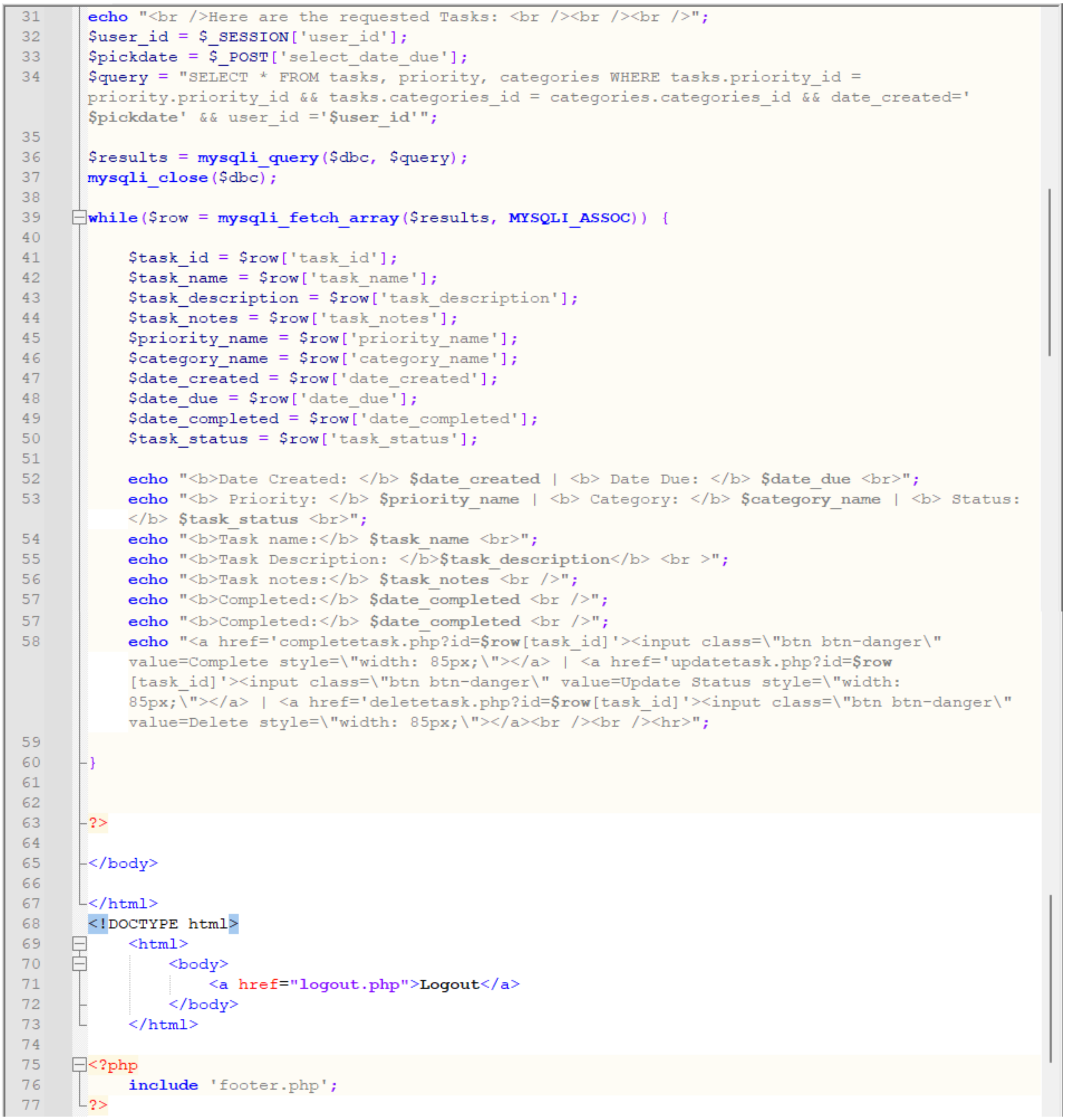


```
<html>
    <body>
        <p> The task has been successfully added. Click <a href="createtask.php">here</a> to create another task, or if you would like to return to your homepage, click <a href="homepage.php">here</a>.
        </p>
    </body>
</html>

<?php
    include 'footer.php';
?>
```

planner.php: Allows the user to enter a date and returns any tasks scheduled for that day.

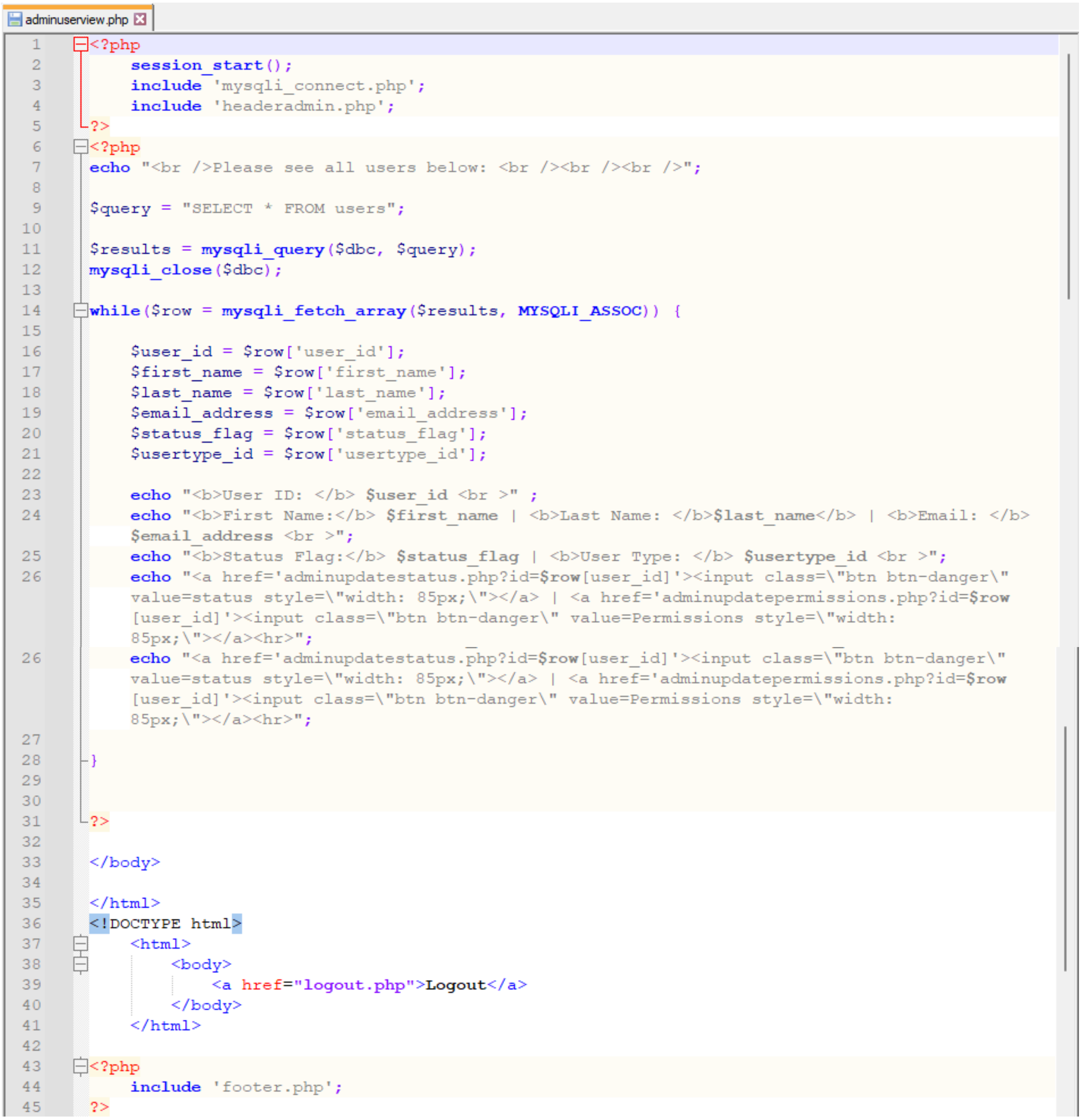


```
<?php
session_start();
?>

<!DOCTYPE html>
<html>
<head>
<title>Today's Tasks</title>
</head>

<body>

<?php
ini_set('display_errors', 0);
include 'mysqli_connect.php';
include 'header.php';

?>

    <form action = "planner.php" method = "post">
    <label = "select date"><b> Select a date </b> </label>
    <label for="date due"><b>Select a Date to view tasks: </b></label>
    <input type="date" id="select_date_due" name="select_date_due"  value="" min="2023-01-01" max="2023-12-31" required>
    <hr>

    <button type="submit" name="submit">View Task!</button>
    </form>

<?php

echo "<br />Here are the requested Tasks: <br /><br /><br />";
```



```
echo "<br />Here are the requested Tasks: <br /><br /><br />";
$user_id = $_SESSION['user_id'];
$pickdate = $_POST['select_date_due'];
$query = "SELECT * FROM tasks, priority, categories WHERE tasks.priority_id = priority.priority_id && tasks.categories_id = categories.categories_id && date_created='$pickdate' && user_id ='$user_id'";

$results = mysqli_query($dbc, $query);
mysqli_close($dbc);

while($row = mysqli_fetch_array($results, MYSQLI_ASSOC)) {

    $task_id = $row['task_id'];
    $task_name = $row['task_name'];
    $task_description = $row['task_description'];
    $task_notes = $row['task_notes'];
    $priority_name = $row['priority_name'];
    $category_name = $row['category_name'];
    $date_created = $row['date_created'];
    $date_due = $row['date_due'];
    $date_completed = $row['date_completed'];
    $task_status = $row['task_status'];

    echo "<b>Date Created: </b> $date_created | <b> Date Due: </b> $date_due <br>";
    echo "<b> Priority: </b> $priority_name | <b> Category: </b> $category_name | <b> Status: </b> $task_status <br>";
    echo "<b>Task name:</b> $task_name <br>";
    echo "<b>Task Description: </b>$task_description</b> <br >";
    echo "<b>Task notes:</b> $task_notes <br />";
    echo "<b>Completed:</b> $date_completed <br />";
    echo "<b>Completed:</b> $date_completed <br />";
    echo "<a href='completetask.php?id=$row[task_id]'><input class=\"btn btn-danger\" value=Complete style=\"width: 85px;\"></a> | <a href='updatetask.php?id=$row[task_id]'><input class=\"btn btn-danger\" value=Update Status style=\"width: 85px;\"></a> | <a href='deletetask.php?id=$row[task_id]'><input class=\"btn btn-danger\" value=Delete style=\"width: 85px;\"></a><br /><br /><hr>";

}


?>

</body>

</html>
<!DOCTYPE html>
    <html>
        <body>
            <a href="logout.php">Logout</a>
        </body>
    </html>

<?php
    include 'footer.php';
?>
```

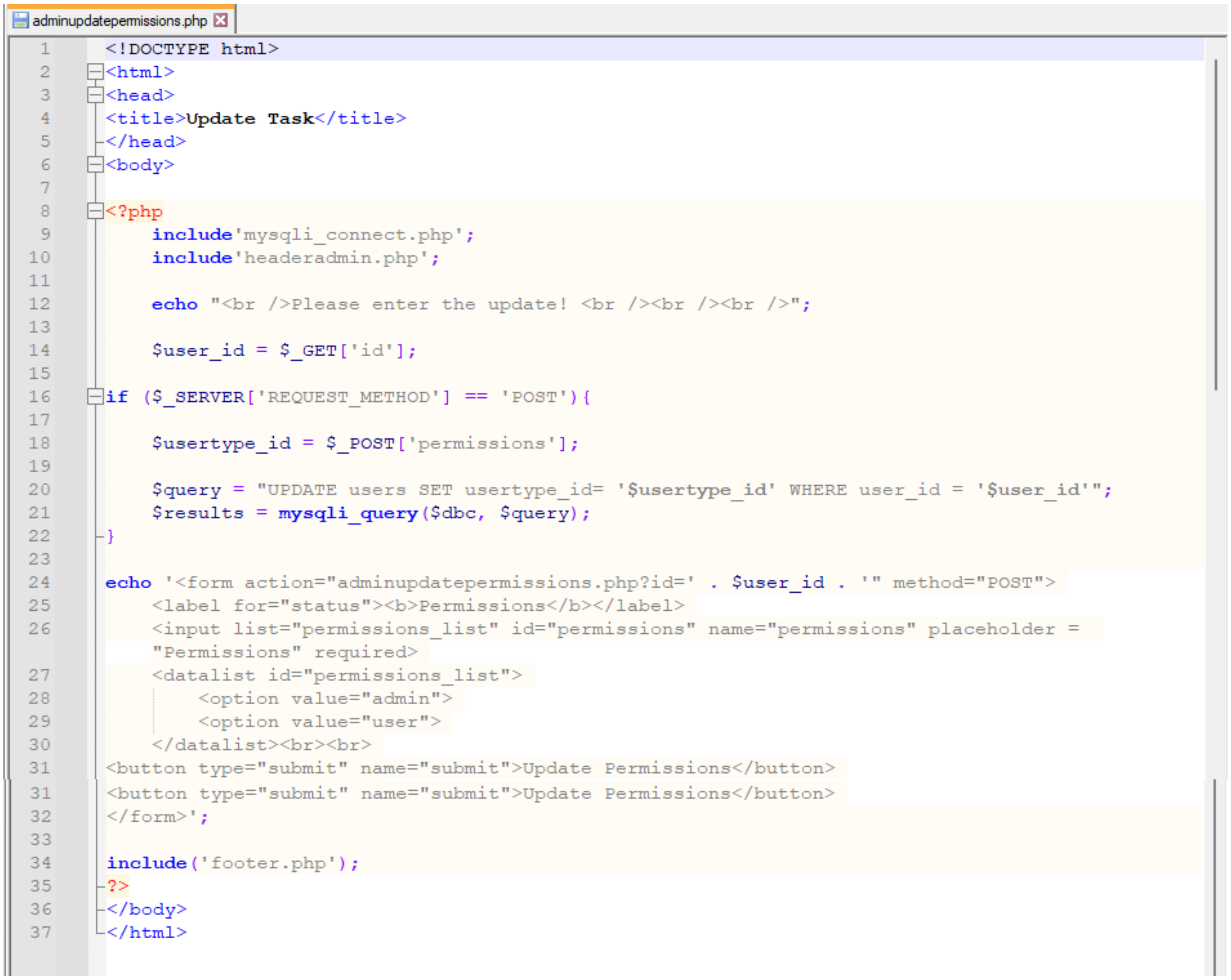
adminuserview.php: Administrator homepage, shows all of the users and information related to them.



adminuserview.php

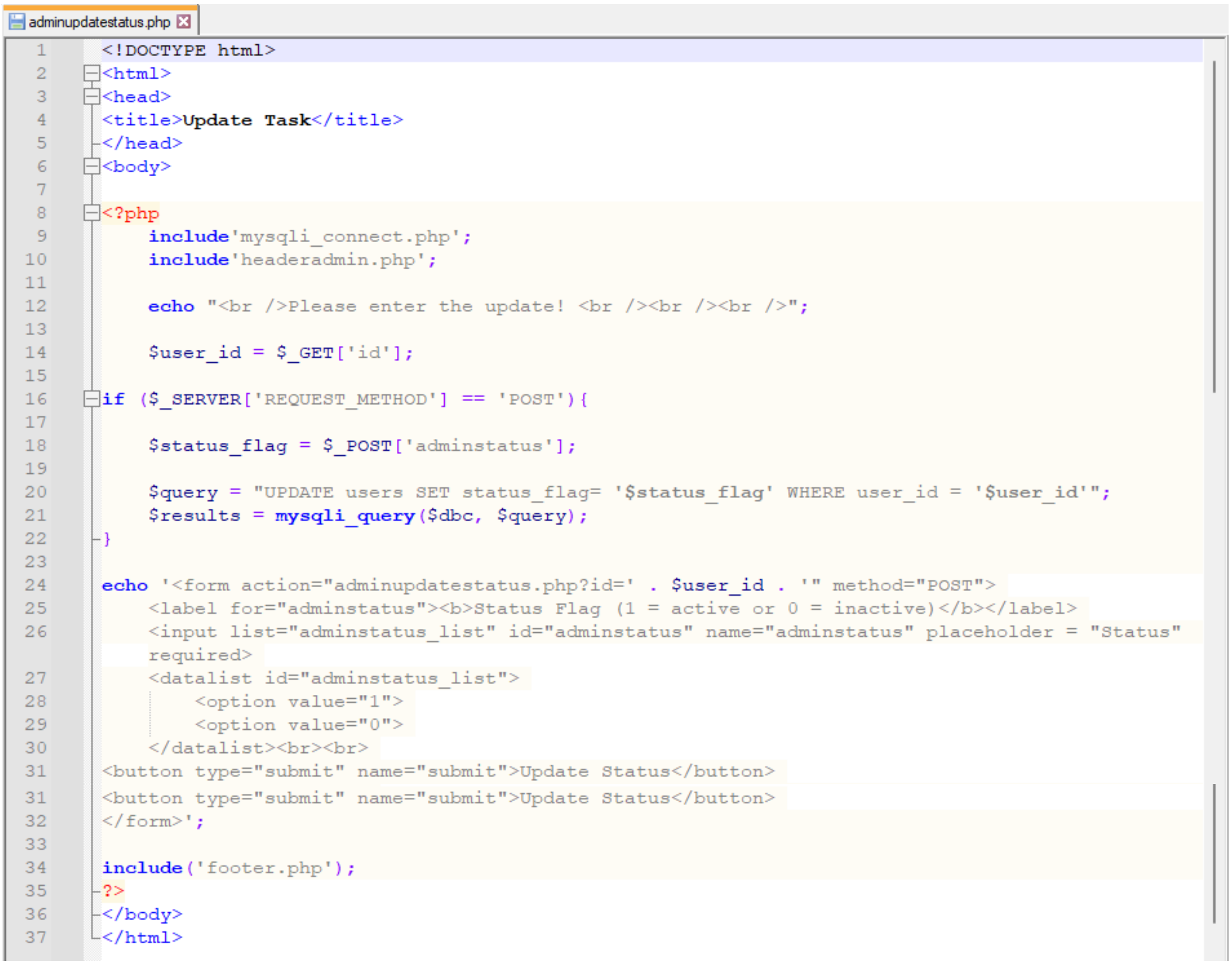
```php
<?php
    session_start();
    include 'mysqli_connect.php';
    include 'headeradmin.php';
?>
<?php
echo "<br />Please see all users below: <br /><br /><br />";

$query = "SELECT * FROM users";

$results = mysqli_query($dbc, $query);
mysqli_close($dbc);

while($row = mysqli_fetch_array($results, MYSQLI_ASSOC)) {

    $user_id = $row['user_id'];
    $first_name = $row['first_name'];
    $last_name = $row['last_name'];
    $email_address = $row['email_address'];
    $status_flag = $row['status_flag'];
    $usertype_id = $row['usertype_id'];

    echo "<b>User ID: </b> $user_id <br >" ;
    echo "<b>First Name:</b> $first_name | <b>Last Name: </b>$last_name</b> | <b>Email: </b> $email_address <br >";
    echo "<b>Status Flag:</b> $status_flag | <b>User Type: </b> $usertype_id <br >";
    echo "<a href='adminupdatestatus.php?id=$row[user_id]'><input class=\"btn btn-danger\" value=status style=\"width: 85px;\"></a> | <a href='adminupdatepermissions.php?id=$row[user_id]'><input class=\"btn btn-danger\" value=Permissions style=\"width: 85px;\"></a><hr>";
    echo "<a href='adminupdatestatus.php?id=$row[user_id]'><input class=\"btn btn-danger\" value=status style=\"width: 85px;\"></a> | <a href='adminupdatepermissions.php?id=$row[user_id]'><input class=\"btn btn-danger\" value=Permissions style=\"width: 85px;\"></a><hr>";

}


?>

</body>

</html>
<!DOCTYPE html>
    <html>
        <body>
            <a href="logout.php">Logout</a>
        </body>
    </html>

<?php
    include 'footer.php';
?>
```

adminupdatepermissions.php: Allows an administrator to promote or demote another administrator.



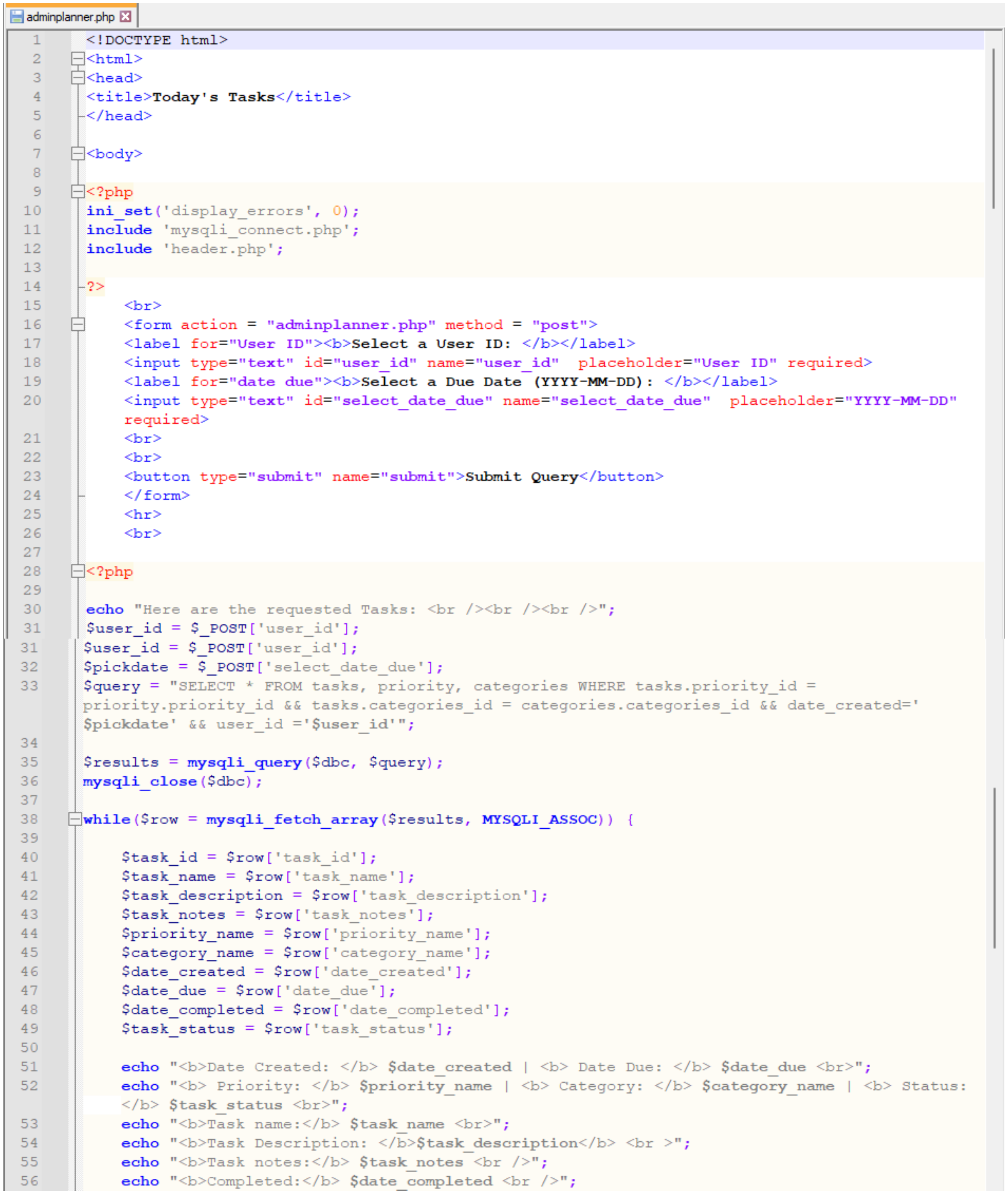
```
<!DOCTYPE html>
<html>
<head>
<title>Update Task</title>
</head>
<body>

<?php
    include'mysqli_connect.php';
    include'headeradmin.php';

    echo "<br />Please enter the update! <br /><br /><br />";

    $user_id = $_GET['id'];

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

    $usertype_id = $_POST['permissions'];

    $query = "UPDATE users SET usertype_id= '$usertype_id' WHERE user_id = '$user_id'";
    $results = mysqli_query($dbc, $query);
}

echo '<form action="adminupdatepermissions.php?id=' . $user_id . '" method="POST">
    <label for="status"><b>Permissions</b></label>
    <input list="permissions_list" id="permissions" name="permissions" placeholder =
    "Permissions" required>
    <datalist id="permissions_list">
        <option value="admin">
        <option value="user">
    </datalist><br><br>
<button type="submit" name="submit">Update Permissions</button>
<button type="submit" name="submit">Update Permissions</button>
</form>';

include('footer.php');
?>
</body>
</html>
```

adminupdatestatus.php: Allows the administrator to mark a user account as active or inactive.

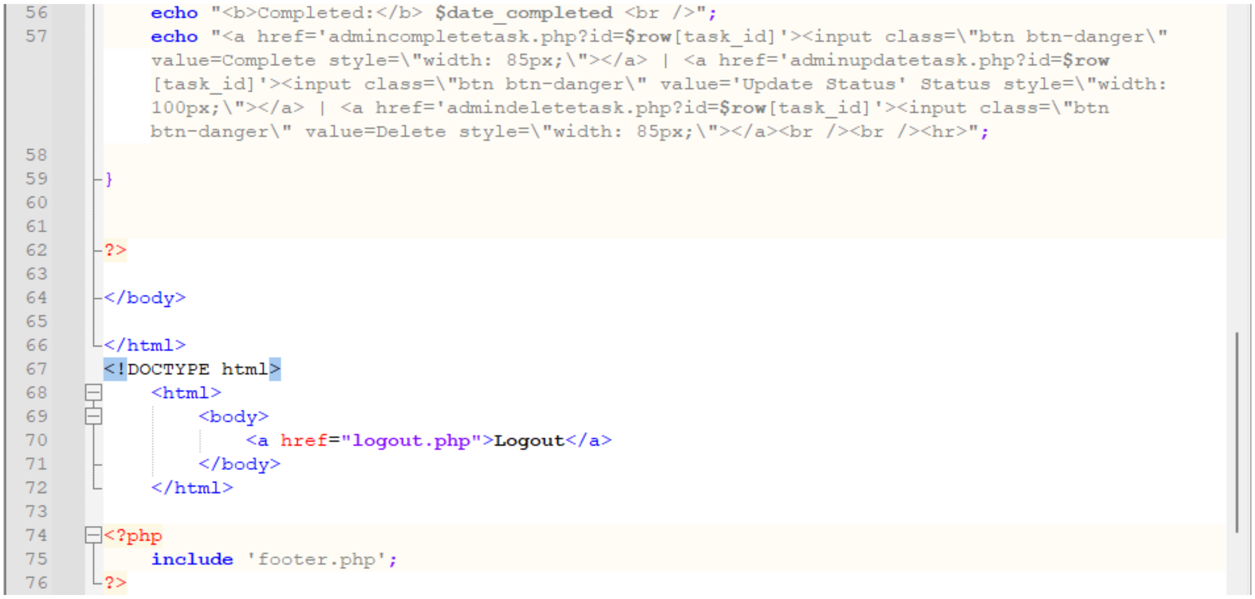


```
<!DOCTYPE html>
<html>
<head>
<title>Update Task</title>
</head>
<body>

<?php
    include'mysqli_connect.php';
    include'headeradmin.php';

    echo "<br />Please enter the update! <br /><br /><br />";

    $user_id = $_GET['id'];

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

    $status_flag = $_POST['adminstatus'];

    $query = "UPDATE users SET status_flag= '$status_flag' WHERE user_id = '$user_id'";
    $results = mysqli_query($dbc, $query);
}

echo '<form action="adminupdatestatus.php?id=' . $user_id . '" method="POST">
    <label for="adminstatus"><b>Status Flag (1 = active or 0 = inactive)</b></label>
    <input list="adminstatus_list" id="adminstatus" name="adminstatus" placeholder = "Status"
    required>
    <datalist id="adminstatus_list">
        <option value="1">
        <option value="0">
    </datalist><br><br>
<button type="submit" name="submit">Update Status</button>
<button type="submit" name="submit">Update Status</button>
</form>';

include('footer.php');
?>
</body>
</html>
```

adminplanner.php: Allows the administrator to look up a specific user’s tasks by entering the User ID and Date.

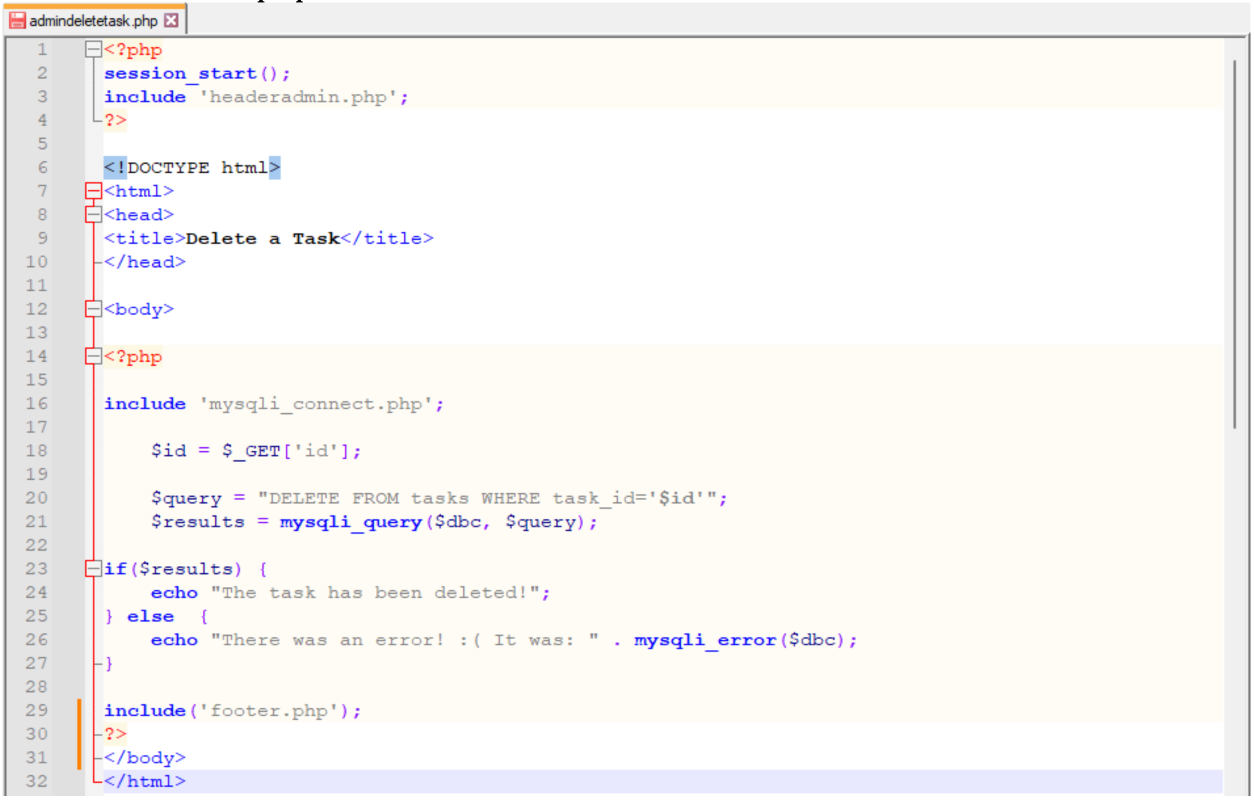


```
<!DOCTYPE html>
<html>
<head>
<title>Today's Tasks</title>
</head>

<body>

<?php
ini_set('display_errors', 0);
include 'mysqli_connect.php';
include 'header.php';

?>
    <br>
    <form action = "adminplanner.php" method = "post">
    <label for="User ID"><b>Select a User ID: </b></label>
    <input type="text" id="user_id" name="user_id"  placeholder="User ID" required>
    <label for="date due"><b>Select a Due Date (YYYY-MM-DD): </b></label>
    <input type="text" id="select_date_due" name="select_date_due"  placeholder="YYYY-MM-DD" required>
    <br>
    <br>
    <button type="submit" name="submit">Submit Query</button>
    </form>
    <hr>
    <br>

<?php

echo "Here are the requested Tasks: <br /><br /><br />";
$user_id = $_POST['user_id'];
$user_id = $_POST['user_id'];
$pickdate = $_POST['select_date_due'];
$query = "SELECT * FROM tasks, priority, categories WHERE tasks.priority_id = priority.priority_id && tasks.categories_id = categories.categories_id && date_created='$pickdate' && user_id ='$user_id'";

$results = mysqli_query($dbc, $query);
mysqli_close($dbc);

while($row = mysqli_fetch_array($results, MYSQLI_ASSOC)) {

    $task_id = $row['task_id'];
    $task_name = $row['task_name'];
    $task_description = $row['task_description'];
    $task_notes = $row['task_notes'];
    $priority_name = $row['priority_name'];
    $category_name = $row['category_name'];
    $date_created = $row['date_created'];
    $date_due = $row['date_due'];
    $date_completed = $row['date_completed'];
    $task_status = $row['task_status'];

    echo "<b>Date Created: </b> $date_created | <b> Date Due: </b> $date_due <br>";
    echo "<b> Priority: </b> $priority_name | <b> Category: </b> $category_name | <b> Status: </b> $task_status <br>";
    echo "<b>Task name:</b> $task_name <br>";
    echo "<b>Task Description: </b>$task_description</b> <br >";
    echo "<b>Task notes:</b> $task_notes <br />";
    echo "<b>Completed:</b> $date_completed <br />";
```



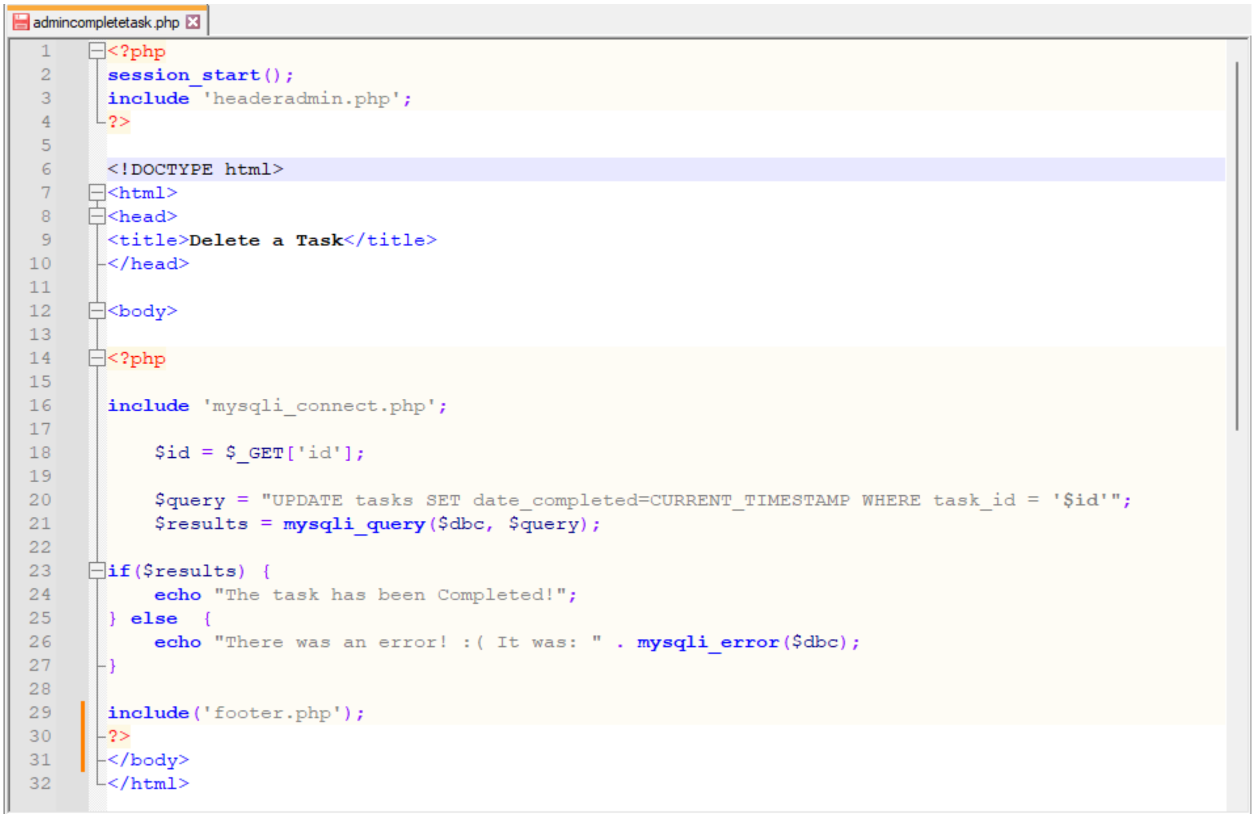
```
    echo "<b>Completed:</b> $date_completed <br />";
    echo "<a href='admincompletetask.php?id=$row[task_id]'><input class=\"btn btn-danger\" value=Complete style=\"width: 85px;\"></a> | <a href='adminupdatetask.php?id=$row[task_id]'><input class=\"btn btn-danger\" value='Update Status' Status style=\"width: 100px;\"></a> | <a href='admindeletetask.php?id=$row[task_id]'><input class=\"btn btn-danger\" value=Delete style=\"width: 85px;\"></a><br /><br /><hr>";

}


?>

</body>

</html>
<!DOCTYPE html>
    <html>
        <body>
            <a href="logout.php">Logout</a>
        </body>
    </html>

<?php
    include 'footer.php';
?>
```

admindeletetask.php: Allows the administrator to delete the selected task.



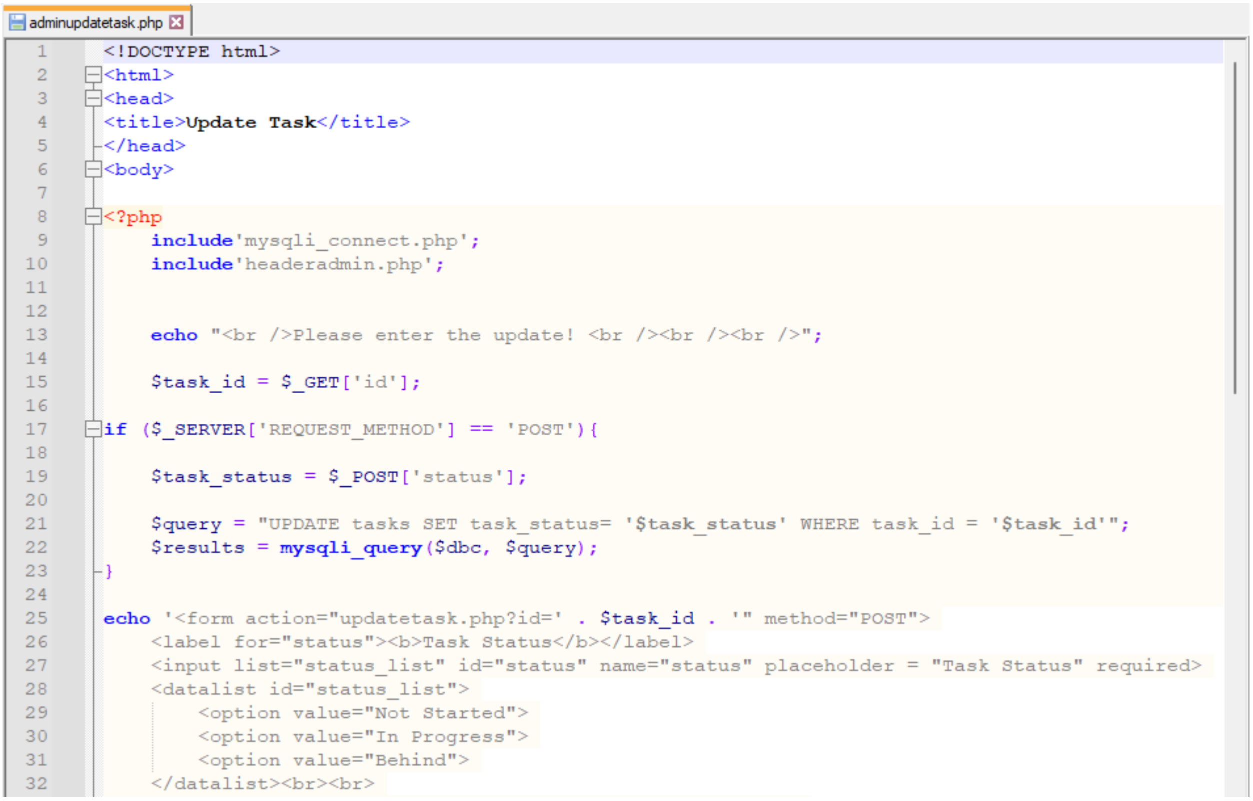
```
<?php
session_start();
include 'headeradmin.php';
?>

<!DOCTYPE html>
<html>
<head>
<title>Delete a Task</title>
</head>

<body>

<?php

include 'mysqli_connect.php';

    $id = $_GET['id'];

    $query = "DELETE FROM tasks WHERE task_id='$id'";
    $results = mysqli_query($dbc, $query);

if($results) {
    echo "The task has been deleted!";
} else  {
    echo "There was an error! :( It was: " . mysqli_error($dbc);
}

include('footer.php');
?>
</body>
</html>
```

admincompletetask.php: Allows the administrator to mark the selected task as complete.

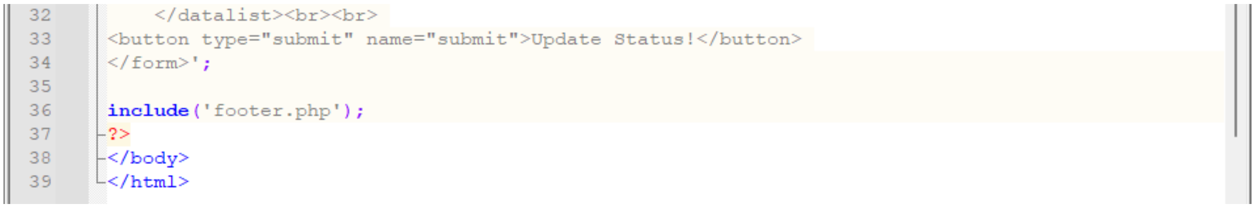


```
<?php
session_start();
include 'headeradmin.php';
?>

<!DOCTYPE html>
<html>
<head>
<title>Delete a Task</title>
</head>

<body>

<?php

include 'mysqli_connect.php';

    $id = $_GET['id'];

    $query = "UPDATE tasks SET date_completed=CURRENT_TIMESTAMP WHERE task_id = '$id'";
    $results = mysqli_query($dbc, $query);

if($results) {
    echo "The task has been Completed!";
} else  {
    echo "There was an error! :( It was: " . mysqli_error($dbc);
}

include('footer.php');
?>
</body>
</html>
```

Adminupdatetask.php: Allows the administrator to update the status of the selected task.



```
<!DOCTYPE html>
<html>
<head>
<title>Update Task</title>
</head>
<body>

<?php
    include'mysqli_connect.php';
    include'headeradmin.php';


    echo "<br />Please enter the update! <br /><br /><br />";

    $task_id = $_GET['id'];

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

    $task_status = $_POST['status'];

    $query = "UPDATE tasks SET task_status= '$task_status' WHERE task_id = '$task_id'";
    $results = mysqli_query($dbc, $query);
}

echo '<form action="updatetask.php?id=' . $task_id . '" method="POST">
    <label for="status"><b>Task Status</b></label>
    <input list="status_list" id="status" name="status" placeholder = "Task Status" required>
    <datalist id="status_list">
        <option value="Not Started">
        <option value="In Progress">
        <option value="Behind">
    </datalist><br><br>
```



```
    </datalist><br><br>
<button type="submit" name="submit">Update Status!</button>
</form>';

include('footer.php');
?>
</body>
</html>
```

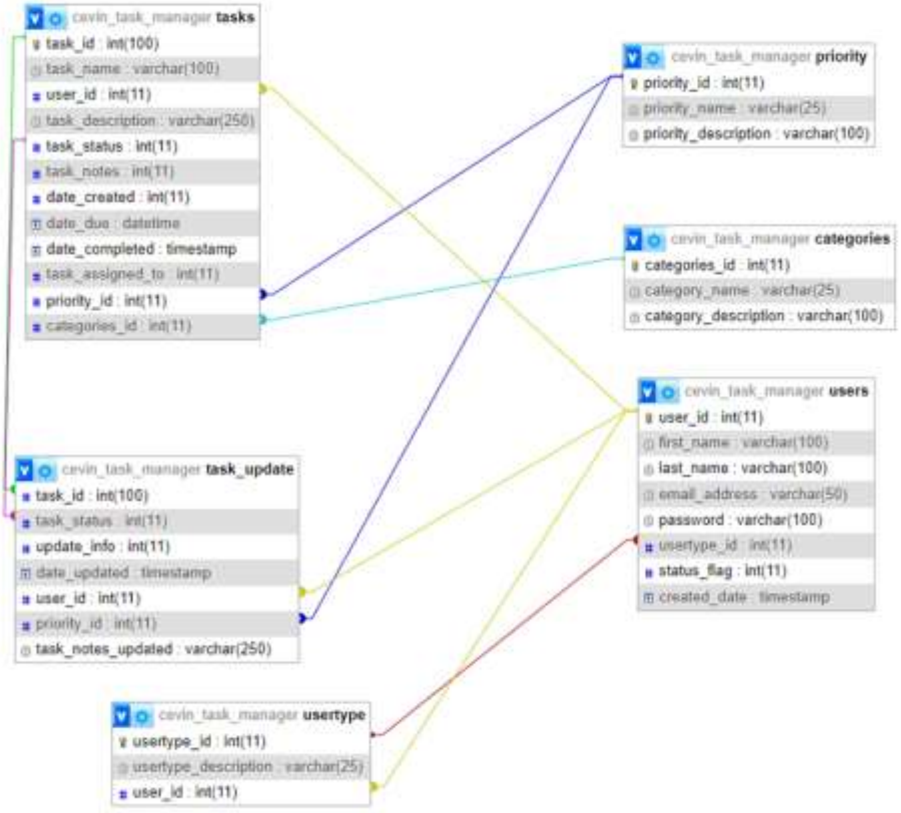
Database Design

1. Overview

The Task Genius website database is hosted on MySQL for maximum compatibility with PHP, the chosen web programming language for the site. It consists of four tables now with one-to-one relationships. In earlier designs it had 6 tables with a couple of one-to-many relationships. To focus on the main scope of the project a couple of tables were dropped because they did not fit the criteria of what we were trying to accomplish. The tables can be associated into two main focus areas. One is the Tasks and the other is the Users, with the other smaller tables (categories & priority) supporting both main tables. Only the main tables of Users and Tasks had auto-incrementing IDs that uniquely identifies each record in the tables. The tables utilize the INNODB engine so that foreign key relationships could be created to enforce referential integrity and overall data integrity throughout. The design is simple and flexible enough to support continuously changing business requirements.

The following section outlines the Entity-Relationship Diagram (ERD), a conceptual and representational model of data used to represent the entity framework infrastructure.

2. Entity-Relationship Diagram (ERD) 1:N Prototype #1



cevin_task_manager **tasks**
task_id : int(100)
task_name : varchar(100)
user_id : int(11)
task_description : varchar(250)
task_status : int(11)
task_notes : int(11)
date_created : int(11)
date_due : datetime
date_completed : timestamp
task_assigned_to : int(11)
priority_id : int(11)
categories_id : int(11)

cevin_task_manager **priority**
priority_id : int(11)
priority_name : varchar(25)
priority_description : varchar(100)

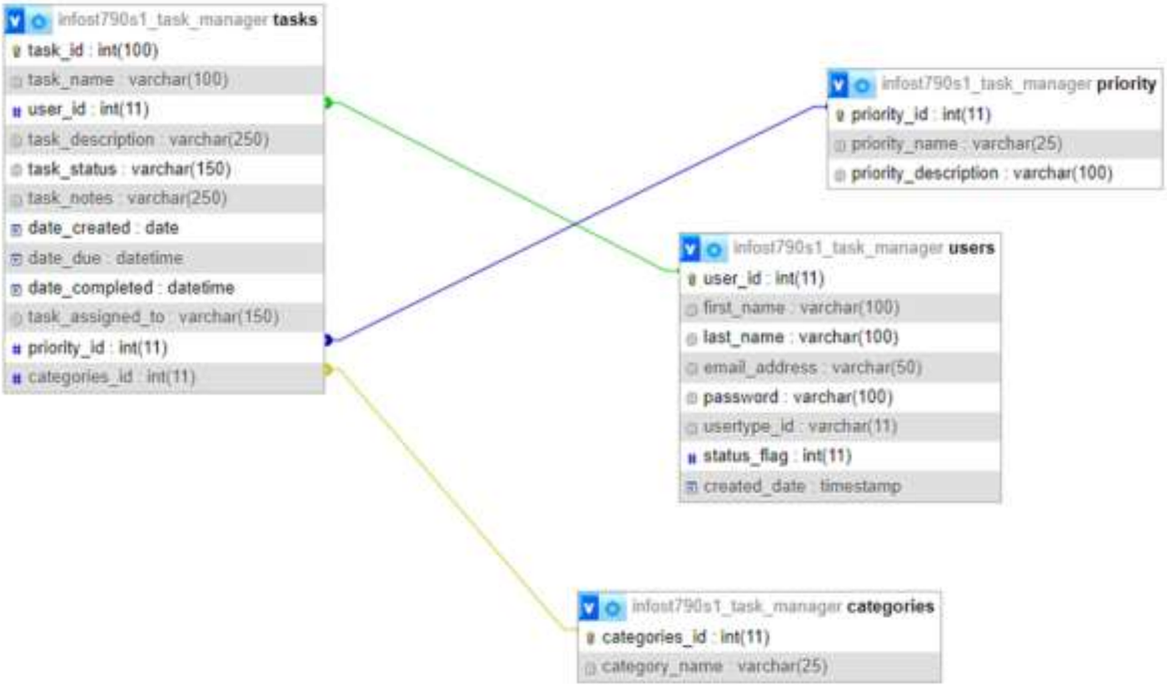
cevin_task_manager **categories**
categories_id : int(11)
category_name : varchar(25)
category_description : varchar(100)

cevin_task_manager **users**
user_id : int(11)
first_name : varchar(100)
last_name : varchar(100)
email_address : varchar(50)
password : varchar(100)
usertype_id : int(11)
status_flag : int(11)
created_date : timestamp

cevin_task_manager **task_update**
task_id : int(100)
task_status : int(11)
update_info : int(11)
date_updated : timestamp
user_id : int(11)
priority_id : int(11)
task_notes_updated : varchar(250)

cevin_task_manager **usertype**
usertype_id : int(11)
usertype_description : varchar(25)
user_id : int(11)

3. Entity-Relationship Diagram (ERD) 1:1 Prototype Final


infost790s1_task_manager tasks
task_id : int(100)
task_name : varchar(100)
user_id : int(11)
task_description : varchar(250)
task_status : varchar(150)
task_notes : varchar(250)
date_created : date
date_due : datetime
date_completed : datetime
task_assigned_to : varchar(150)
priority_id : int(11)
categories_id : int(11)
infost790s1_task_manager priority
priority_id : int(11)
priority_name : varchar(25)
priority_description : varchar(100)
infost790s1_task_manager users
user_id : int(11)
first_name : varchar(100)
last_name : varchar(100)
email_address : varchar(50)
password : varchar(100)
usertype_id : varchar(11)
status_flag : int(11)
created_date : timestamp
infost790s1_task_manager categories
categories_id : int(11)
category_name : varchar(25)

4. Data Definition Language (DDL)

Prototype #1 MYSQL code

```
--
-- Database: `cevin_task_manager`
--
--
-- Table structure for table `categories`
--

CREATE TABLE `categories` (
  `categories_id` int(11) NOT NULL,
  `category_name` varchar(25) NOT NULL,
  `category_description` varchar(100) NOT NULL COMMENT 'leisure, work, home'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------


--
-- Table structure for table `priority`
--

CREATE TABLE `priority` (
  `priority_id` int(11) NOT NULL DEFAULT '2' COMMENT '1=low 2=medium, 3=high',
  `priority_name` varchar(25) NOT NULL COMMENT 'low, medium, high',
  `priority_description` varchar(100) NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `tasks`
--

CREATE TABLE `tasks` (
  `task_id` int(100) NOT NULL,
  `task_name` varchar(100) NOT NULL,
  `user_id` int(11) NOT NULL,
  `task_description` varchar(250) NOT NULL,
  `task_status` int(11) NOT NULL,
  `task_notes` int(11) NOT NULL,
  `date_created` int(11) NOT NULL,
  `date_due` datetime NOT NULL,
  `date_completed` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `task_assigned_to` int(11) NOT NULL,
  `priority_id` int(11) NOT NULL COMMENT '1=low,2=medium,3=high',
  `categories_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `task_update`
--

CREATE TABLE `task_update` (
  `task_id` int(100) NOT NULL,
  `task_status` int(11) NOT NULL COMMENT 'In Progress, Behind, Not started',
  `update_info` int(11) NOT NULL,
  `date_updated` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `user_id` int(11) NOT NULL,
  `priority_id` int(11) NOT NULL COMMENT '1=low 2=medium, 3=high',
  `task_notes_updated` varchar(250) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `users`
--

CREATE TABLE `users` (
  `user_id` int(11) NOT NULL,
  `first_name` varchar(100) NOT NULL,
  `last_name` varchar(100) NOT NULL,
  `email_address` varchar(50) NOT NULL,
  `password` varchar(100) NOT NULL,
  `usertype_id` int(11) NOT NULL COMMENT 'admin or user',
  `status_flag` int(11) NOT NULL COMMENT 'disabled or active',
  `created_date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------


--
-- Table structure for table `usertype`
--

CREATE TABLE `usertype` (
  `usertype_id` int(11) NOT NULL,
  `usertype_description` varchar(25) NOT NULL COMMENT 'admin or user',
  `user_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Indexes for dumped tables
--


--
-- Indexes for table `categories`
--
ALTER TABLE `categories`
  ADD PRIMARY KEY (`categories_id`);

--
-- Indexes for table `priority`
--
ALTER TABLE `priority`
  ADD PRIMARY KEY (`priority_id`);

--
-- Indexes for table `tasks`
--
ALTER TABLE `tasks`
  ADD PRIMARY KEY (`task_id`),
  ADD KEY `priority_id` (`priority_id`),
  ADD KEY `categories_id` (`categories_id`),
  ADD KEY `user_id` (`user_id`) USING BTREE,
  ADD KEY `task_status` (`task_status`);

--
-- Indexes for table `task_update`
```

```
--
ALTER TABLE `task_update`
  ADD KEY `task_id` (`task_id`),
  ADD KEY `user_id` (`user_id`),
  ADD KEY `priority_id` (`priority_id`),
  ADD KEY `task_status` (`task_status`);

--
-- Indexes for table `users`
--
ALTER TABLE `users`
  ADD PRIMARY KEY (`user_id`),
  ADD KEY `usertype_id` (`usertype_id`);

--
-- Indexes for table `usertype`
--
ALTER TABLE `usertype`
  ADD PRIMARY KEY (`usertype_id`),
  ADD KEY `user_id` (`user_id`) USING BTREE,
  ADD KEY `usertype_description` (`usertype_description`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `categories`
--
ALTER TABLE `categories`
  MODIFY `categories_id` int(11) NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT for table `tasks`
--
ALTER TABLE `tasks`
  MODIFY `task_id` int(100) NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT for table `users`
--
ALTER TABLE `users`
  MODIFY `user_id` int(11) NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT for table `usertype`
--
ALTER TABLE `usertype`
```

```
  MODIFY `usertype_id` int(11) NOT NULL AUTO_INCREMENT;

--
-- Constraints for dumped tables
--

--
-- Constraints for table `tasks`
--
ALTER TABLE `tasks`
  ADD CONSTRAINT `tasks_ibfk_1` FOREIGN KEY (`categories_id`) REFERENCES `categories` (`categories_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `tasks_ibfk_2` FOREIGN KEY (`priority_id`) REFERENCES `priority` (`priority_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `tasks_ibfk_3` FOREIGN KEY (`user_id`) REFERENCES `users` (`user_id`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `task_update`
--
ALTER TABLE `task_update`
  ADD CONSTRAINT `task_update_ibfk_1` FOREIGN KEY (`task_id`) REFERENCES `tasks` (`task_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `task_update_ibfk_2` FOREIGN KEY (`priority_id`) REFERENCES `priority` (`priority_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `task_update_ibfk_3` FOREIGN KEY (`user_id`) REFERENCES `users` (`user_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `task_update_ibfk_4` FOREIGN KEY (`task_status`) REFERENCES `tasks` (`task_status`);

--
-- Constraints for table `users`
--
ALTER TABLE `users`
  ADD CONSTRAINT `users_ibfk_1` FOREIGN KEY (`usertype_id`) REFERENCES `usertype` (`usertype_id`) ON DELETE CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `usertype`
--
ALTER TABLE `usertype`
  ADD CONSTRAINT `usertype_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`user_id`) ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;
```

Prototype Final MYSQL code

```
--
-- Database: `infost790s1_task_manager`
```

```
--

-- --------------------------------------------------------

--
-- Table structure for table `categories`
--

CREATE TABLE `categories` (
  `categories_id` int(11) NOT NULL,
  `category_name` varchar(25) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


--
-- Dumping data for table `categories`
--

INSERT INTO `categories` (`categories_id`, `category_name`) VALUES
(1, 'leisure'),
(2, 'work'),
(3, 'home');

-- --------------------------------------------------------

--
-- Table structure for table `priority`
--

CREATE TABLE `priority` (
  `priority_id` int(11) NOT NULL DEFAULT '2' COMMENT '1=low 2=medium, 3=high',
  `priority_name` varchar(25) NOT NULL COMMENT 'low, medium, high',
  `priority_description` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `priority`
--

INSERT INTO `priority` (`priority_id`, `priority_name`, `priority_description`) VALUES
(1, 'low', ''),
(2, 'medium', ''),
(3, 'high', '');

-- --------------------------------------------------------

--
-- Table structure for table `tasks`
```

```
--

CREATE TABLE `tasks` (
  `task_id` int(100) NOT NULL,
  `task_name` varchar(100) NOT NULL,
  `user_id` int(11) NOT NULL,
  `task_description` varchar(250) NOT NULL,
  `task_status` varchar(150) NOT NULL COMMENT 'In Progress, Behind, Not started',
  `task_notes` varchar(250) NOT NULL,
  `date_created` date NOT NULL,
  `date_due` datetime NOT NULL,
  `date_completed` datetime DEFAULT CURRENT_TIMESTAMP,
  `task_assigned_to` varchar(150) NOT NULL COMMENT 'name assigned to',
  `priority_id` int(11) NOT NULL COMMENT '1=low,2=medium,3=high',
  `categories_id` int(11) NOT NULL COMMENT 'Is this needed?'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `tasks`
--

INSERT INTO `tasks` (`task_id`, `task_name`, `user_id`, `task_description`, `task_status`, `task_notes`,
`date_created`, `date_due`, `date_completed`, `task_assigned_to`, `priority_id`, `categories_id`) VALUES
(17, 'test', 1, 'test', 'In Progress', '', '2023-04-28', '0000-00-00 00:00:00', '2023-04-28 18:57:43', '1', 1, 1),
(18, 'test2', 1, 'test2', 'In Progress', '', '2023-04-28', '2023-04-29 00:00:00', '2023-04-28 21:55:00', '1', 1, 1),
(19, 'test task1', 1, 'This is a test', 'Not Started', 'This is a test', '2023-04-29', '2023-04-29 00:00:00', '0000-00-00
00:00:00', '1', 1, 1),
(20, 'test task2', 1, 'This is a test', 'Not Started', 'This is a test', '2023-04-29', '2023-04-29 00:00:00', '0000-00-00
00:00:00', '1', 1, 1),
(26, '', 12, 'Another test task', 'In Progress', 'Wow such test', '2023-04-29', '2023-04-30 07:30:00', '2023-04-29
20:51:14', '12', 2, 3),
(28, '04/30 test', 12, 'test task', 'Not Started', 'test', '2023-04-30', '2023-04-30 07:30:00', '0000-00-00 00:00:00',
'12', 2, 3);

-- --------------------------------------------------------

--
-- Table structure for table `users`
--

CREATE TABLE `users` (
  `user_id` int(11) NOT NULL,
  `first_name` varchar(100) NOT NULL,
  `last_name` varchar(100) NOT NULL,
  `email_address` varchar(50) NOT NULL,
  `password` varchar(100) NOT NULL,
  `usertype_id` varchar(11) NOT NULL COMMENT 'admin or user',
  `status_flag` int(11) NOT NULL COMMENT '1=active 0=inactive ',
  `created_date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `users`
--

INSERT INTO `users` (`user_id`, `first_name`, `last_name`, `email_address`, `password`, `usertype_id`, `status_flag`,
`created_date`) VALUES
(1, 'admin', '', 'admin', 'd033e22ae348aeb5660fc2140aec35850c4da997', 'admin', 1, '2023-04-22 04:08:11'),
(2, 'Bridgette', 'Antonat', 'bantonat1@mediafire.com', 'ac92fb8b3e0275f9415836badcab3ae7068ec031', 'user', 1,
'2023-02-04 06:00:00'),
(3, 'Smitty', 'Kelloway', 'skelloway2@google.ru', '62926f9f584838a1cf6953aff4244e67e68254dc', 'user', 1, '2023-
03-03 06:00:00'),
(4, 'Corene', 'Paintain', 'cpaintain3@xinhuanet.com', 'de7d41b60611d47e82a4d72111fe9875ed5d5083', 'user', 1,
'2023-03-31 05:00:00'),
(5, 'Serene', 'Grelak', 'sgrelak4@springer.com', '0bb4e633d813cd1fa18f133464077472920a1f96', 'user', 0, '2023-
03-12 06:00:00'),
(6, 'Barbaraanne', 'Wand', 'bwand5@instagram.com', '4ea55462111b3de4e97496e55c83232d0bd1400f', 'user', 1,
'2023-03-11 06:00:00'),
(7, 'Tom', 'Baker', 'tom@uwm.com', '5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8', 'user', 1, '2023-04-22
04:15:02'),
(8, 'Holly', 'Cricket', 'hcricket6@nbcnews.com', 'd672ca8c3151d2f0897fa011265df6efd1b49b1e', 'user', 0, '2023-02-
15 06:00:00'),
(9, 'Gennie', 'Mougeot', 'gmougeot7@vimeo.com', '5e754d2a4c6a473028026ff79a9ed0c61e781d42', 'user', 1,
'2023-02-15 06:00:00'),
(10, 'Dorey', 'Pettie', 'dpettie8@cam.ac.uk', 'bd3bbfb29404471baa354b62355f9e9f54176401', 'user', 1, '2023-04-
01 05:00:00'),
(11, 'Tersina', 'Seath', 'tseath9@artisteer.com', '10b2eb76749cd414cb4f2da446e1f4932390d822', 'user', 0, '2023-
04-07 05:00:00'),
(12, 'Andrew', 'Lukowski', 'test', 'a94a8fe5ccb19ba61c4c0873d391e987982fbbd3', 'user', 1, '2023-04-26 22:45:48'),
(13, 'Andrew', 'Lukowski', 'test2', 'a94a8fe5ccb19ba61c4c0873d391e987982fbbd3', 'user', 1, '2023-04-29
12:48:27'),
(14, 'Cassandra', 'Franco', 'clfranco@uwm.edu', 'd7cd56f2a2a3f47830760edfb89946eb7b9e2cd1', 'user', 1, '2023-
04-30 18:15:05'),
(15, 'Andrew', 'Lukowski', 'lukowsk2@uwm.edu', '9d4e1e23bd5b727046a9e3b4b7db57bd8d6ee684', 'user', 1,
'2023-04-30 18:20:15');

--
-- Indexes for dumped tables
--


--
-- Indexes for table `categories`
--
ALTER TABLE `categories`
  ADD PRIMARY KEY (`categories_id`);

--
-- Indexes for table `priority`
```

```
--
ALTER TABLE `priority`
  ADD PRIMARY KEY (`priority_id`);

--
-- Indexes for table `tasks`
--
ALTER TABLE `tasks`
  ADD PRIMARY KEY (`task_id`),
  ADD KEY `priority_id` (`priority_id`),
  ADD KEY `categories_id` (`categories_id`),
  ADD KEY `user_id` (`user_id`) USING BTREE;

--
-- Indexes for table `users`
--
ALTER TABLE `users`
  ADD PRIMARY KEY (`user_id`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `categories`
--
ALTER TABLE `categories`
  MODIFY `categories_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT for table `tasks`
--
ALTER TABLE `tasks`
  MODIFY `task_id` int(100) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=29;

--
-- AUTO_INCREMENT for table `users`
--
ALTER TABLE `users`
  MODIFY `user_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=16;

--
-- Constraints for dumped tables
--

--
-- Constraints for table `tasks`
--
```

```
ALTER TABLE `tasks`
  ADD CONSTRAINT `tasks_ibfk_3` FOREIGN KEY (`user_id`) REFERENCES `users` (`user_id`) ON DELETE CASCADE ON
UPDATE CASCADE,
  ADD CONSTRAINT `tasks_ibfk_4` FOREIGN KEY (`priority_id`) REFERENCES `priority` (`priority_id`) ON DELETE
CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `tasks_ibfk_5` FOREIGN KEY (`categories_id`) REFERENCES `categories` (`categories_id`) ON
DELETE CASCADE ON UPDATE CASCADE;
COMMIT;
```

Encrypted pre-existing passwords on database.

```
UPDATE `users` SET `password` = SHA1('i8Ai4mjM9NZi') WHERE `users`.`user_id`
= 2; UPDATE `users` SET `password` = SHA1('VaZGWG') WHERE `users`.`user_id` =
3; UPDATE `users` SET `password` = SHA1('Rx5arhDXF') WHERE `users`.`user_id`
= 4; UPDATE `users` SET `password` = SHA1('gVb9gV') WHERE `users`.`user_id` =
5; UPDATE `users` SET `password` = SHA1('2ttWEDfm') WHERE `users`.`user_id` =
6; UPDATE `users` SET `password` = SHA1('password') WHERE `users`.`user_id` =
7; UPDATE `users` SET `password` = SHA1('kZpeRbJYi3r') WHERE
`users`.`user_id` = 8; UPDATE `users` SET `password` = SHA1('XPWEJ4ZIh')
WHERE `users`.`user_id` = 9; UPDATE `users` SET `password` = SHA1('Jyisyob1')
WHERE `users`.`user_id` = 10; UPDATE `users` SET `password` = SHA1('8RuDcE')
WHERE `users`.`user_id` = 11;
```

5. MySQL Information

The following section outlines the table schemas which give the values in a column, constraints on the value, and the expected format of the data, stored procedures which the website uses to provide scheduling information, and the Data Definition Language (DDL) which creates the tables and the relationships between them.

Table Schemas

“tasks” Table

Key	Fieldname	Datatype	Length	Optional	Other
PK	task_id	int	100	N	auto_increment
	task_name	varchar	100	N	
FK	user_id	int	11	N	
	task_description	varchar	250	N	
	task_staus	varchar	150	N	In Progress, Behind, Not Started
	task_notes	varchar	250	N	

	date_created	date		N	
	date_due	datetime		N	
	date_completed	datetime		Y	CURRENT_TIMESTAMP
	task_assigned_to	varchar	150	N	Name assigned to
FK	priority_id	int	11	N	1=low,2=medium,3=high
FK	categories_id	int	11	N	

“users” Table

Key	Fieldname	Datatype	Length	Optional	Comments/Other
PK	user_id	int	11	N	auto_increment
	first_name	varchar	100	N	
	last_name	varchar	100	N	
	email_address	varchar	50	N	
	password	varchar	100	N	
	usertype_id	varchar	11	N	Admin or user
	status_flag	int	11	N	1=active 0-inactive
	created_date	timestamp		N	CURRENT_TIMESTAMP

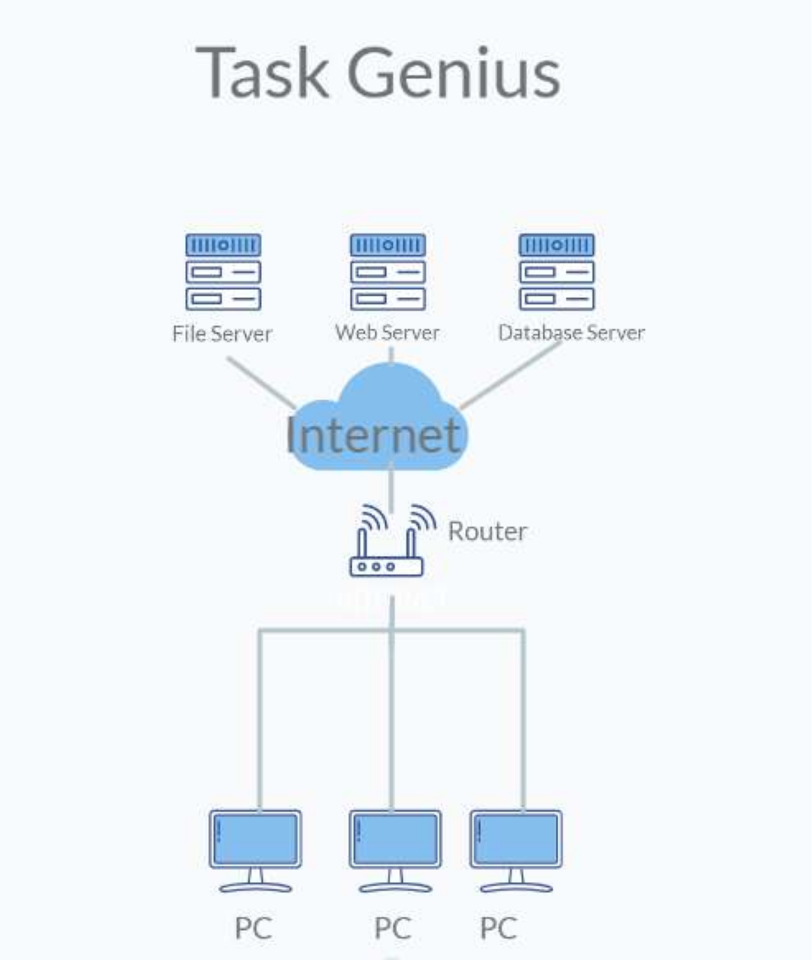
“priority” Table

Key	Fieldname	Datatype	Length	Optional	Other
PK	priority_id	int	11	N	1=low,2=medium,3=high
	priority_name	varchar	25	N	low, medium, high
	priority_description	varchar	100	N	

“categories” Table

Key	Fieldname	Datatype	Length	Optional	Other
PK	user_id	int	11	N	auto_increment
	first_name	varchar	25	N	

Network Diagram



Conclusion:
In conclusion, the personal task manager prototype is a web-based application that helps users manage their daily tasks and events. The system is designed using modern web development tools and technologies and follows the Model-View-Controller architecture. The system's functionality includes the ability to create, edit, and delete tasks and events, set task priorities, reminders, and due dates, and search for specific tasks and events.