UTM

UNIVERSITI TEKNOLOGI MALAYSIA

WEB TECHNOLOGY SCSJ3483 02

Prof. Madya Dr. Yazid

Assignment 4

nAvi Team

Ahmad Firdaus Bin Abd Rashid
A18CS0013

1. Video Conference - Ahmad Firdaus

This is an assignment 4 where I need to use the Slim Framework as backend.

Here I use the methods which are PUT, DELETE, GET and POST.

Image Arrangement

1. PUT/DELETE Method Display
    - Image 1.4
2. PUT method code
    - Image 1.9
3. DELETE method code
    - Image 1.8
4. GET method display
    - Image 1.1
    - Image 1.3
5. GET method code
    - Image 1.6
    - Image 1.7
6. POST method display
    - Image 1.5
7. POST method code
    - Image 1.10
    - Image 1.11
8. JavaScript code to fetch data
    a. GET
        - Image 1.12
    b. POST from Form
        - Image 1.13

1. Image 1.1 – display – GET method
2. Image 1.2 – Example database
3. Image 1.3 – display – GET method
4. Image 1.4 – display – PUT/DELETE method
5. Image 1.5 – display – POST method
6. Image 1.6 – code – GET method
7. Image 1.7 – code – GET method
8. Image 1.8 – code – DELETE method
9. Image 1.9 – code – PUT method
10. Image 1.10 – code – POST method
11. Image 1.11 – code – POST method
12. Image 1.12 – code – JavaScript GET
13. Image 1.13 – code – JavaScript POST form

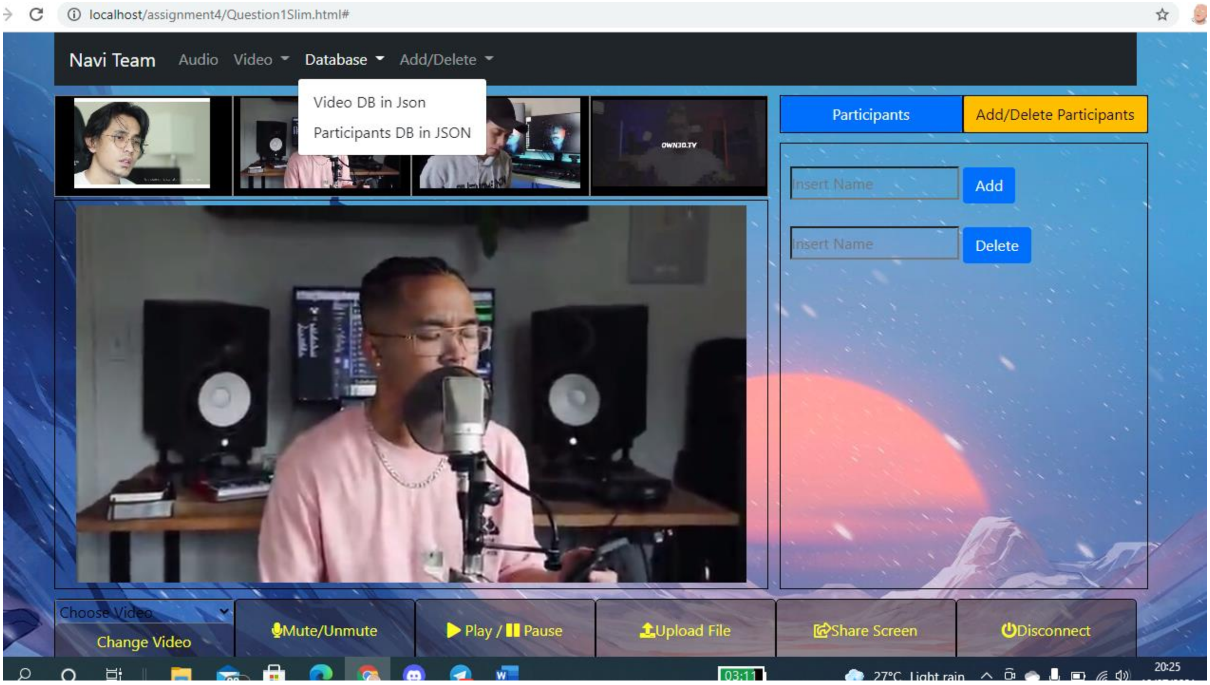
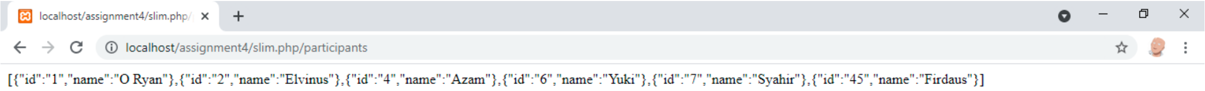


Image 1.1: Database navbar to see the database fetch from SLIM


localhost/assignment4/slim.php/
localhost/assignment4/slim.php/participants

```
[{"id":"1","name":"O Ryan"},{"id":"2","name":"Elvinus"},{"id":"4","name":"Azam"},{"id":"6","name":"Yuki"},{"id":"7","name":"Syahir"},{"id":"45","name":"Firdaus"}]
```

Image 1.2: I replace the name index.php to slim.php, Database data for GET method

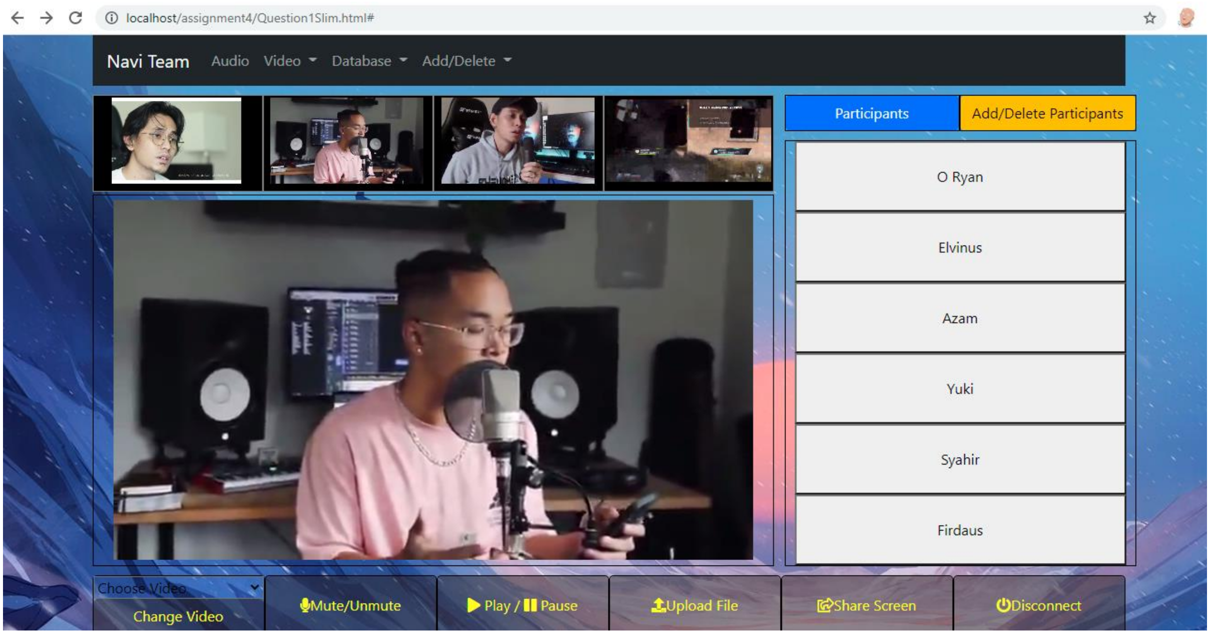


Image 1.3: All data fetch from slim.php

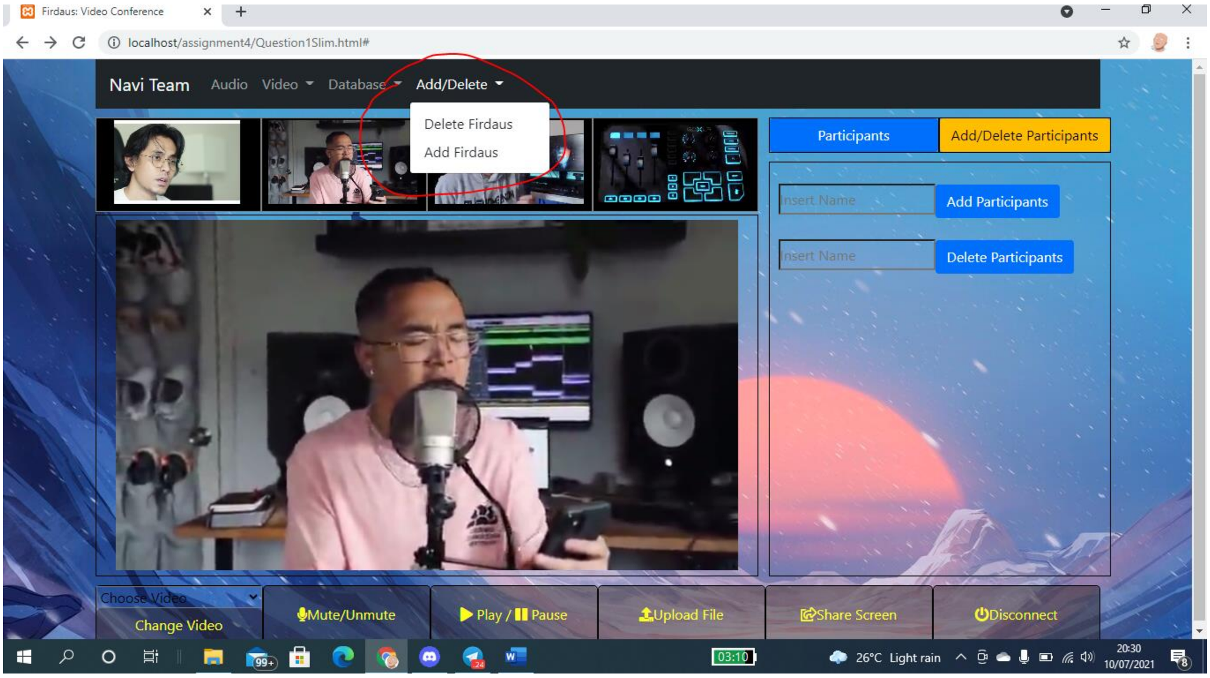


Image 1.4: Using navbar to add/delete Firdaus as participants using PUT/DELETE method

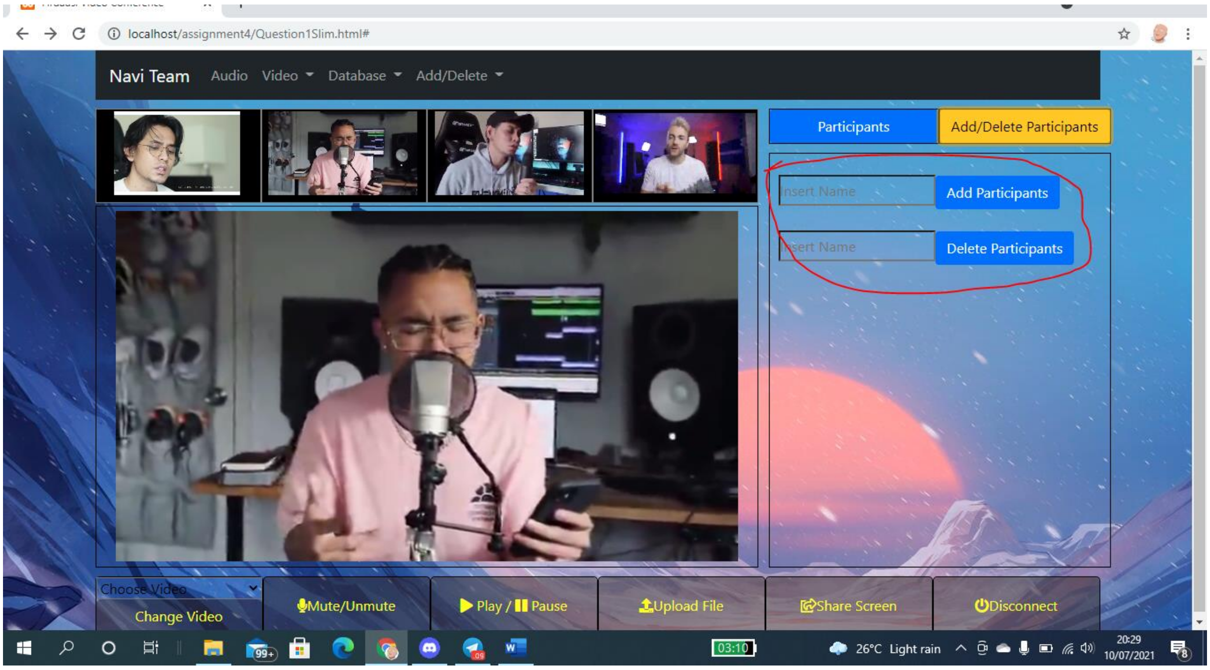


Image 1.5: Using POST method to insert and delete data from database

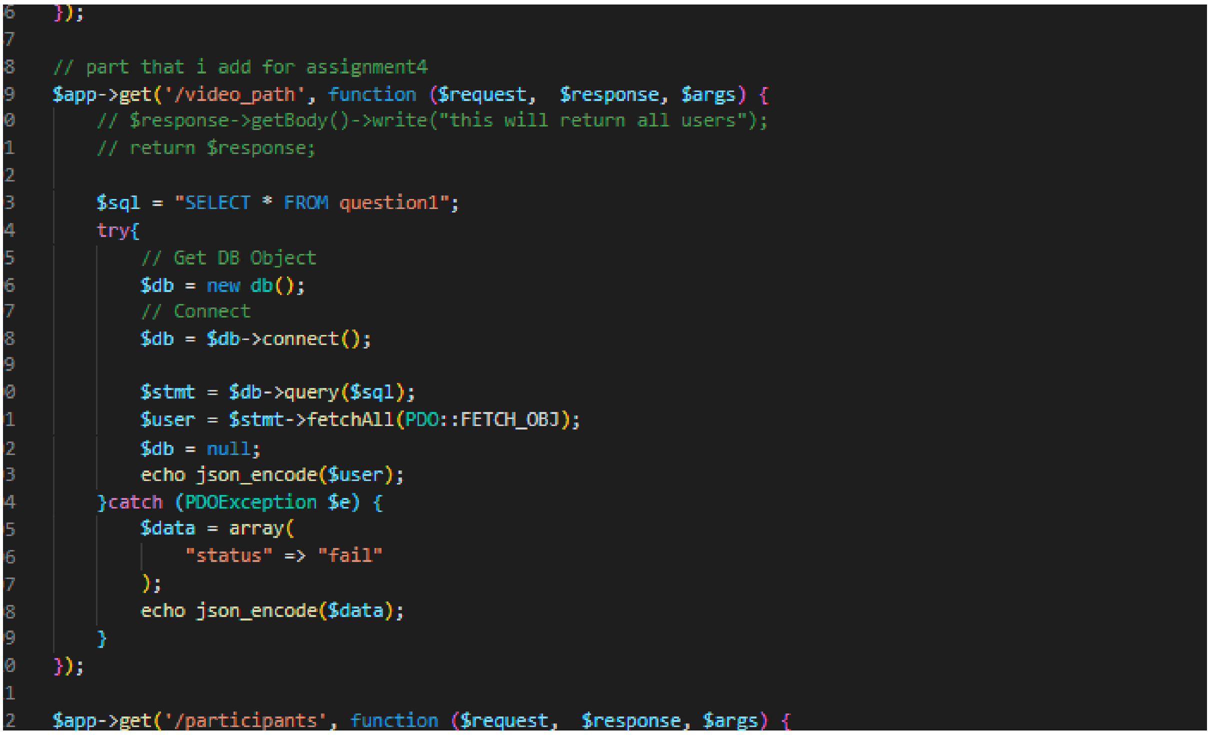
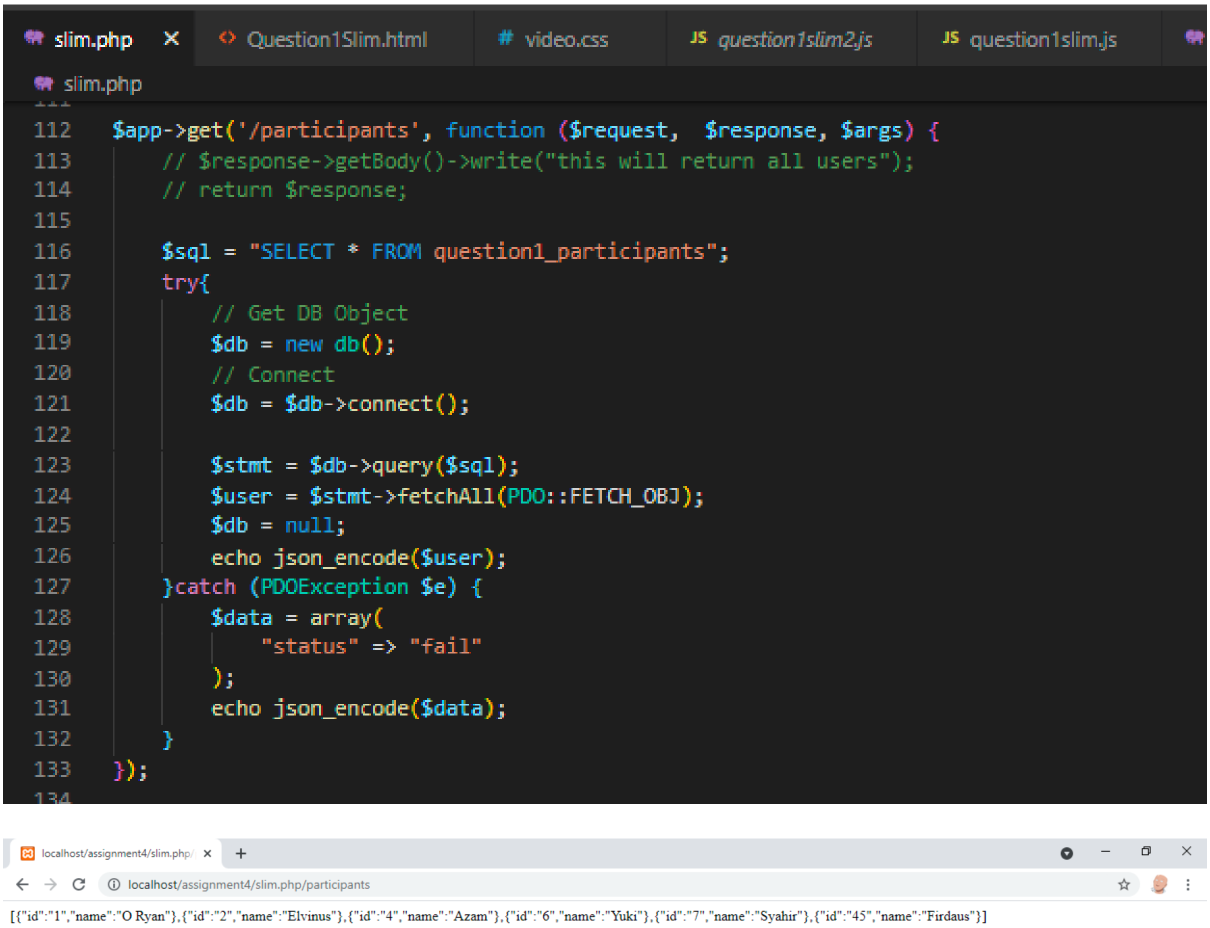

```
});

// part that i add for assignment4
$app->get('/video_path', function ($request,  $response, $args) {
    // $response->getBody()->write("this will return all users");
    // return $response;

    $sql = "SELECT * FROM question1";
    try{
        // Get DB Object
        $db = new db();
        // Connect
        $db = $db->connect();

        $stmt = $db->query($sql);
        $user = $stmt->fetchAll(PDO::FETCH_OBJ);
        $db = null;
        echo json_encode($user);
    }catch (PDOException $e) {
        $data = array(
            "status" => "fail"
        );
        echo json_encode($data);
    }
});

$app->get('/participants', function ($request,  $response, $args) {
```

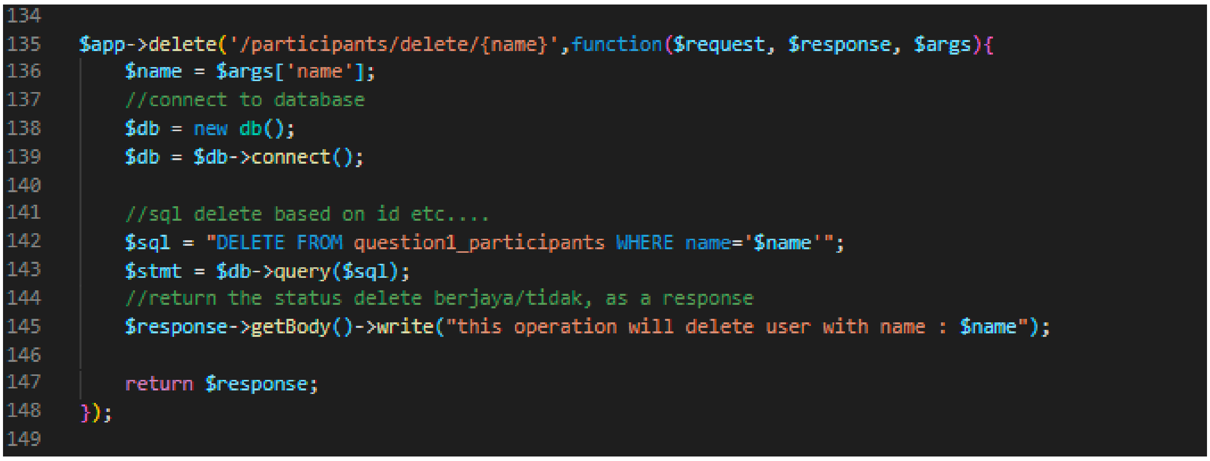
Image 1.6: GET method in SLIM to get video_path



```php
$app->get('/participants', function ($request,  $response, $args) {
    // $response->getBody()->write("this will return all users");
    // return $response;

    $sql = "SELECT * FROM question1_participants";
    try{
        // Get DB Object
        $db = new db();
        // Connect
        $db = $db->connect();

        $stmt = $db->query($sql);
        $user = $stmt->fetchAll(PDO::FETCH_OBJ);
        $db = null;
        echo json_encode($user);
    }catch (PDOException $e) {
        $data = array(
            "status" => "fail"
        );
        echo json_encode($data);
    }
});
```

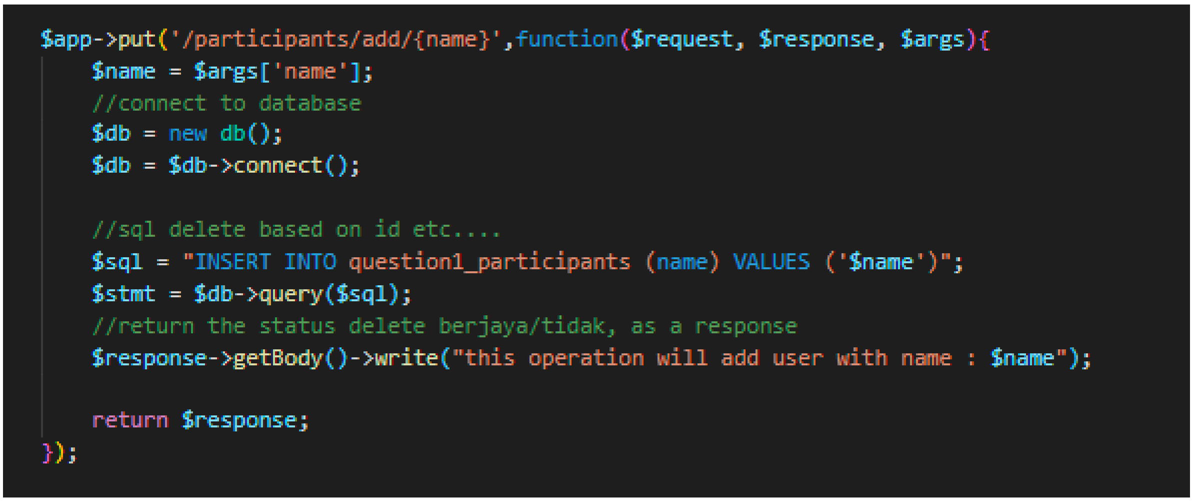
```
[{"id":"1","name":"O Ryan"},{"id":"2","name":"Elvinus"},{"id":"4","name":"Azam"},{"id":"6","name":"Yuki"},{"id":"7","name":"Syahir"},{"id":"45","name":"Firdaus"}]
```

Image 1.7: GET method in SLIM to view all participants



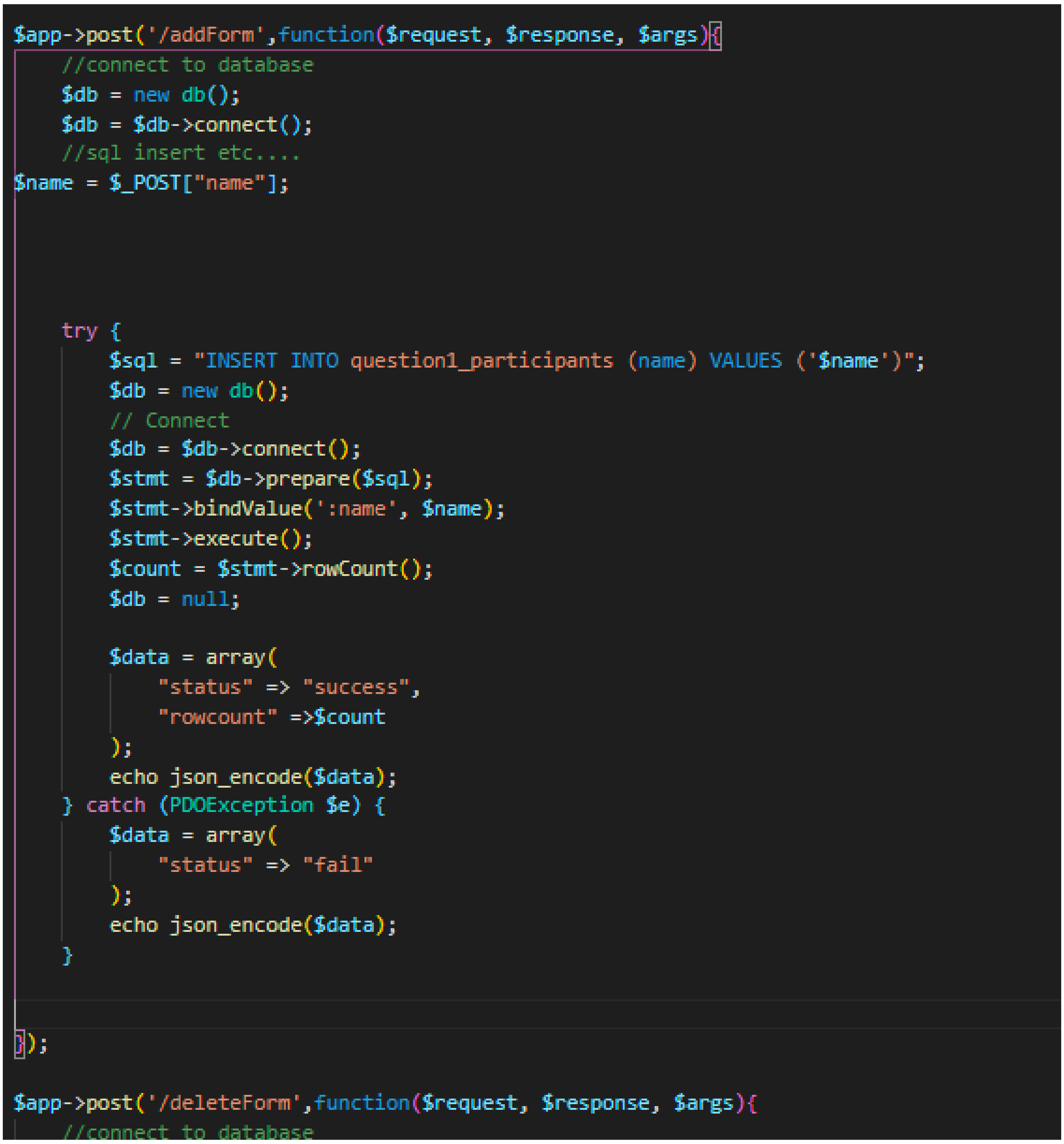
```php
$app->delete('/participants/delete/{name}',function($request, $response, $args){
    $name = $args['name'];
    //connect to database
    $db = new db();
    $db = $db->connect();

    //sql delete based on id etc....
    $sql = "DELETE FROM question1_participants WHERE name='$name'";
    $stmt = $db->query($sql);
    //return the status delete berjaya/tidak, as a response
    $response->getBody()->write("this operation will delete user with name : $name");

    return $response;
});
```

Image 1.8: DELETE method to delete a participant



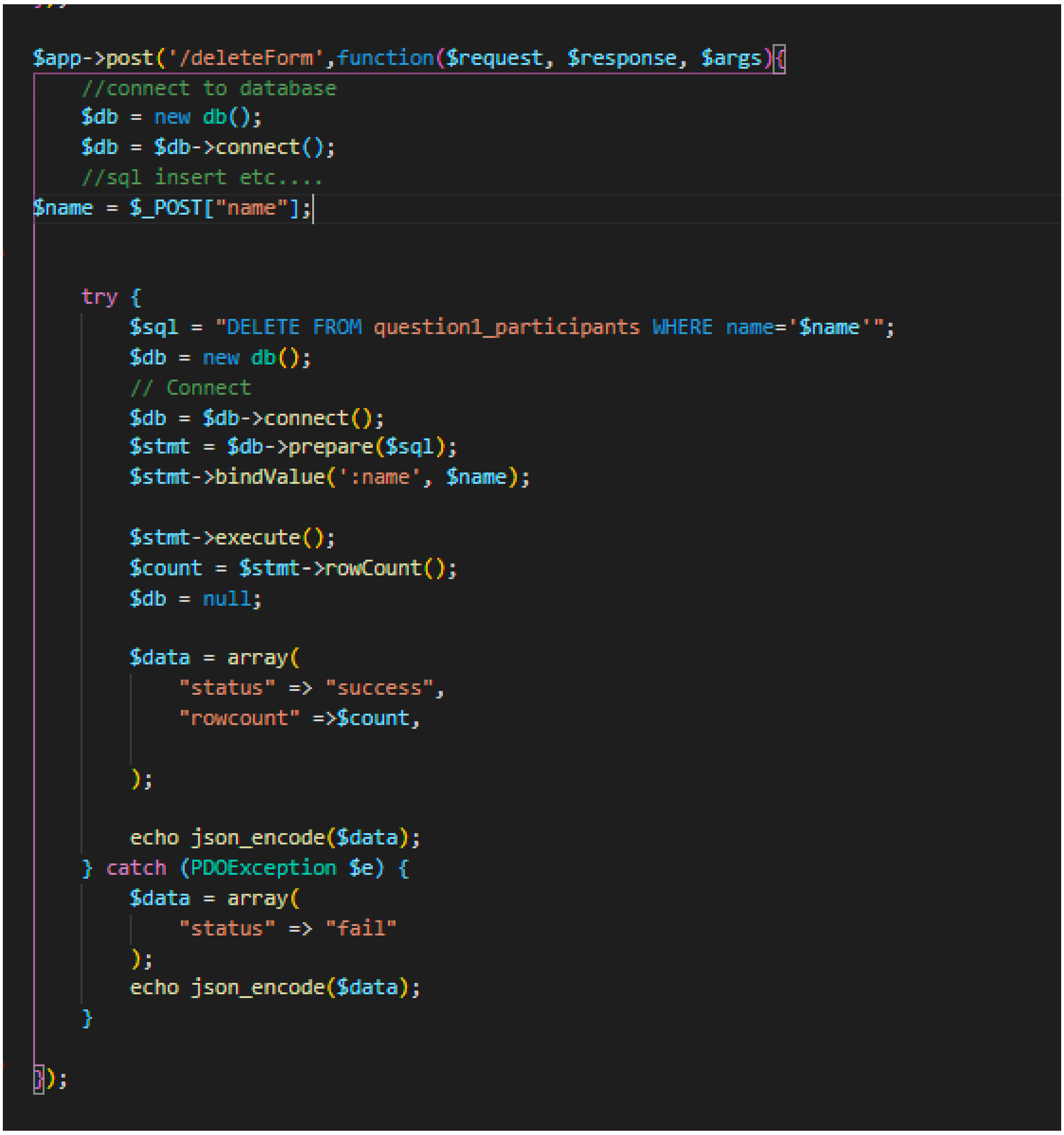
```php
$app->put('/participants/add/{name}',function($request, $response, $args){
    $name = $args['name'];
    //connect to database
    $db = new db();
    $db = $db->connect();

    //sql delete based on id etc....
    $sql = "INSERT INTO question1_participants (name) VALUES ('$name')";
    $stmt = $db->query($sql);
    //return the status delete berjaya/tidak, as a response
    $response->getBody()->write("this operation will add user with name : $name");

    return $response;
});
```

Image 1.9: PUT method to add participants



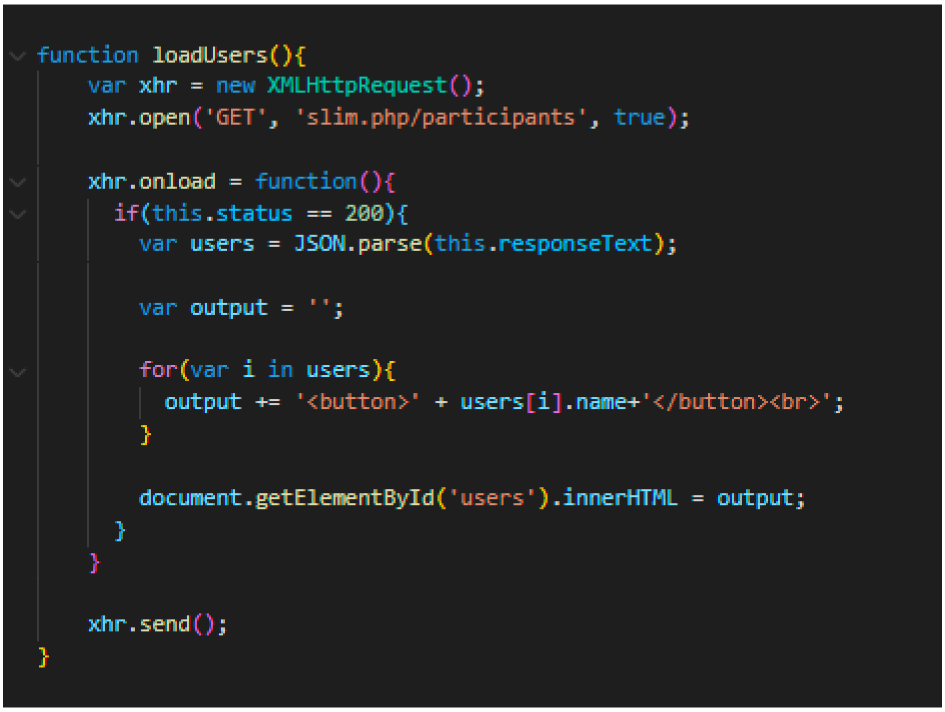
```php
$app->post('/addForm',function($request, $response, $args){
    //connect to database
    $db = new db();
    $db = $db->connect();
    //sql insert etc....
$name = $_POST["name"];




    try {
        $sql = "INSERT INTO question1_participants (name) VALUES ('$name')";
        $db = new db();
        // Connect
        $db = $db->connect();
        $stmt = $db->prepare($sql);
        $stmt->bindValue(':name', $name);
        $stmt->execute();
        $count = $stmt->rowCount();
        $db = null;

        $data = array(
            "status" => "success",
            "rowcount" =>$count
        );
        echo json_encode($data);
    } catch (PDOException $e) {
        $data = array(
            "status" => "fail"
        );
        echo json_encode($data);
    }


});

$app->post('/deleteForm',function($request, $response, $args){
    //connect to database
```

Image 1.10: POST method to add participants



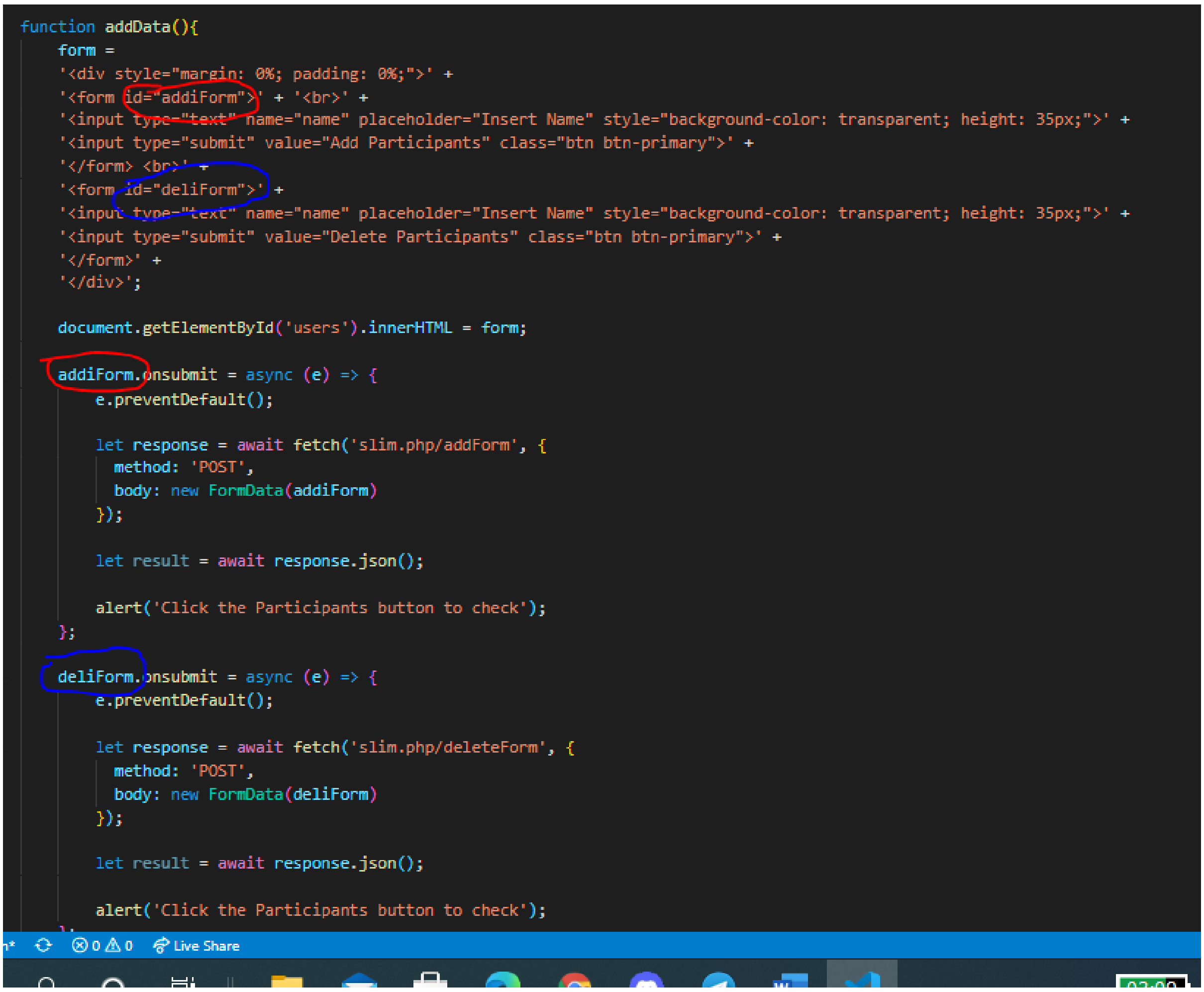
```
});

$app->post('/deleteForm',function($request, $response, $args){
    //connect to database
    $db = new db();
    $db = $db->connect();
    //sql insert etc....
$name = $_POST["name"];


    try {
        $sql = "DELETE FROM question1_participants WHERE name='$name'";
        $db = new db();
        // Connect
        $db = $db->connect();
        $stmt = $db->prepare($sql);
        $stmt->bindValue(':name', $name);

        $stmt->execute();
        $count = $stmt->rowCount();
        $db = null;

        $data = array(
            "status" => "success",
            "rowcount" =>$count,

        );

        echo json_encode($data);
    } catch (PDOException $e) {
        $data = array(
            "status" => "fail"
        );
        echo json_encode($data);
    }

});
```

Image 1.11: POST method to delete participants



```
function loadUsers(){
    var xhr = new XMLHttpRequest();
    xhr.open('GET', 'slim.php/participants', true);

    xhr.onload = function(){
      if(this.status == 200){
        var users = JSON.parse(this.responseText);

        var output = '';

        for(var i in users){
          output += '<button>' + users[i].name+'</button><br>';
        }

        document.getElementById('users').innerHTML = output;
      }
    }

    xhr.send();
}
```

Image 1.12: using JavaScript to call slim framework, GET method



```
function addData(){
    form =
    '<div style="margin: 0%; padding: 0%;">' +
    '<form id="addiForm">' + '<br>' +
    '<input type="text" name="name" placeholder="Insert Name" style="background-color: transparent; height: 35px;">' +
    '<input type="submit" value="Add Participants" class="btn btn-primary">' +
    '</form> <br>' +
    '<form id="deliForm">' +
    '<input type="text" name="name" placeholder="Insert Name" style="background-color: transparent; height: 35px;">' +
    '<input type="submit" value="Delete Participants" class="btn btn-primary">' +
    '</form>' +
    '</div>';

    document.getElementById('users').innerHTML = form;

    addiForm.onsubmit = async (e) => {
        e.preventDefault();

        let response = await fetch('slim.php/addForm', {
          method: 'POST',
          body: new FormData(addiForm)
        });

        let result = await response.json();

        alert('Click the Participants button to check');
    };

    deliForm.onsubmit = async (e) => {
        e.preventDefault();

        let response = await fetch('slim.php/deleteForm', {
          method: 'POST',
          body: new FormData(deliForm)
        });

        let result = await response.json();

        alert('Click the Participants button to check');
```

Image 1.13: use JavaScript to manipulate html file, and use form to send POST data