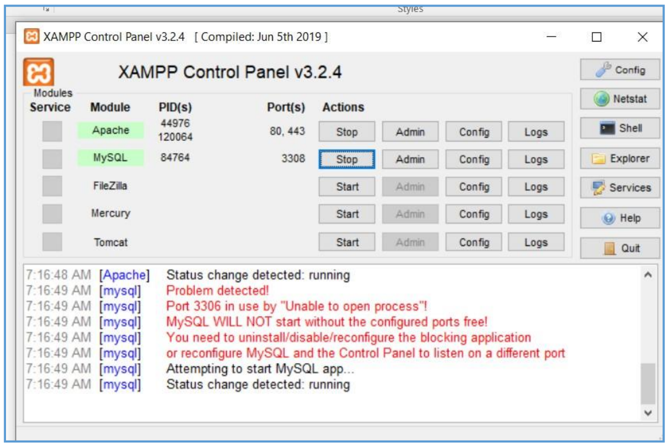
<u>KH4065CEM Lab 8 (P2): CRUD Applicaation</u>

Start Apache and MySQL


XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]
XAMPP Control Panel v3.2.4
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
	Apache	44976 120064	80, 443	Stop	Admin	Config	Logs
	MySQL	84764	3308	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

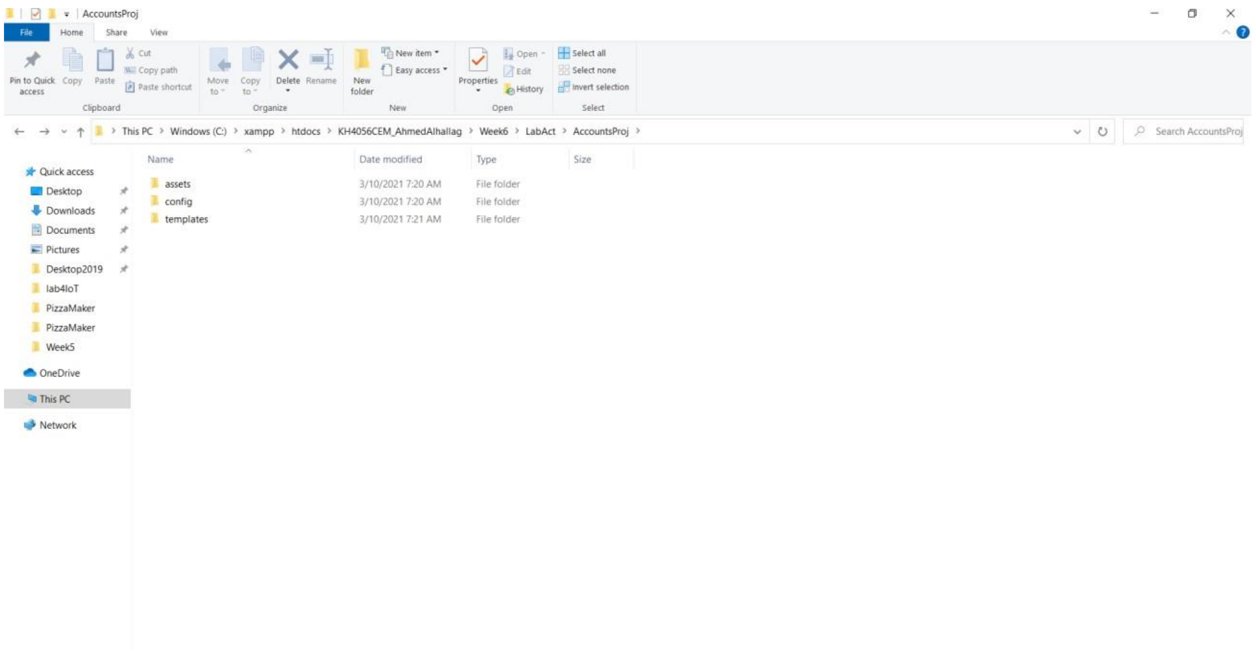
```
7:16:48 AM  [Apache]  Status change detected: running
7:16:49 AM  [mysql]   Problem detected!
7:16:49 AM  [mysql]   Port 3306 in use by "Unable to open process"!
7:16:49 AM  [mysql]   MySQL WILL NOT start without the configured ports free!
7:16:49 AM  [mysql]   You need to uninstall/disable/reconfigure the blocking application
7:16:49 AM  [mysql]   or reconfigure MySQL and the Control Panel to listen on a different port
7:16:49 AM  [mysql]   Attempting to start MySQL app...
7:16:49 AM  [mysql]   Status change detected: running
```

Make sure to have the following path as a project folder:

xampp → htdocs → KH4056CEM_<YourName> → Week 6 →LabAct → AccountsProj

Inside the **AccountsProj** folder create the project folders structure we discussed from before:

- assets
- templates
- config



Name	Date modified	Type	Size
assets	3/10/2021 7:20 AM	File folder	
config	3/10/2021 7:20 AM	File folder	
templates	3/10/2021 7:21 AM	File folder	

Copy the css files from the “boiler_plate_w6” folder (after you extract) on Moodle’s Week 6. Paste them in the assets folder you created.

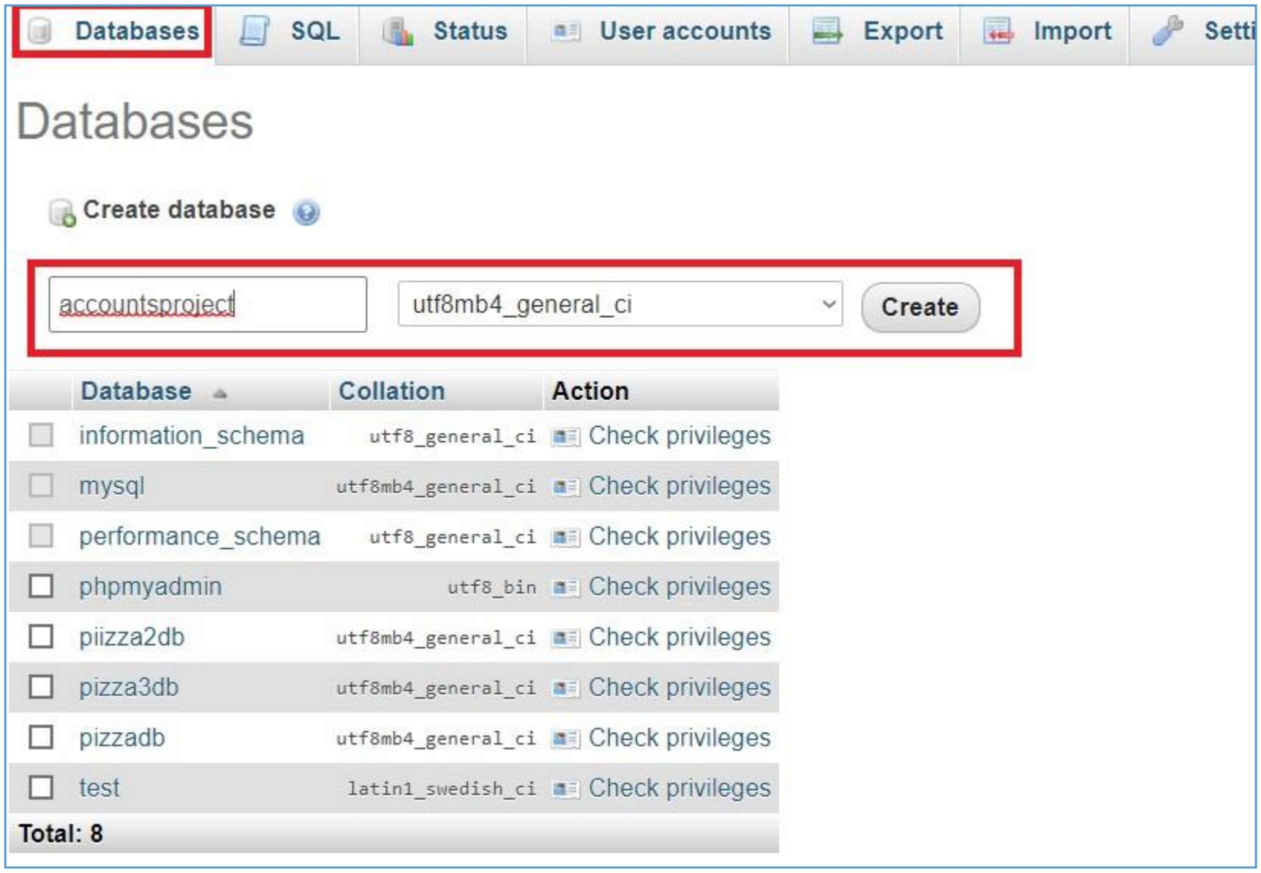
Open the project folder, **AccountsProj**, in Atom. Browse to its location in the xampp directory.

Open ‘**XAMPP Control Panel**’, start the Apache server to serve the php files for use in the browser.

Start the MySQL server to host our databases and tables.

Click on ‘Admin’ beside the MySQL channel to open the database management system (DBMS) interface, or ‘**phpmyadmin**’.

From ‘**Databases’**, Create a database and call it “***accountsproject***”



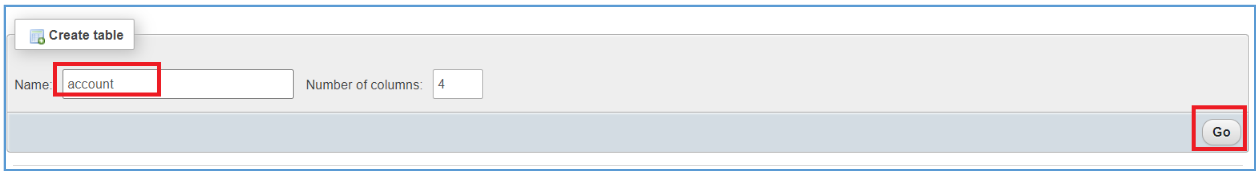
Databases

Create database

accountsproject	utf8mb4_general_ci

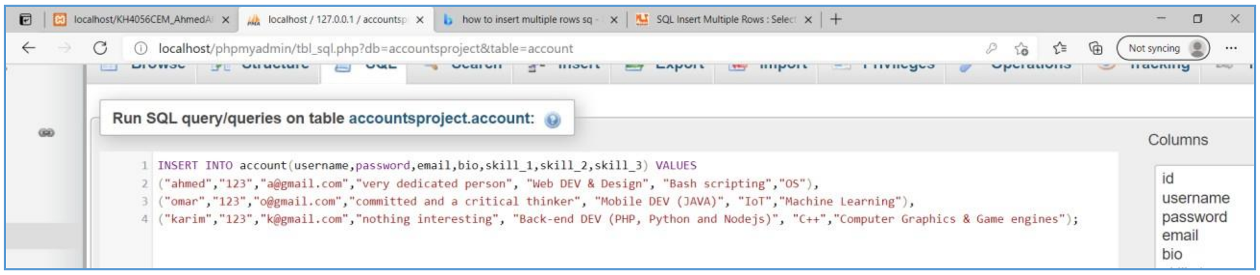
	Database	Collation	Action
☐	information_schema	utf8_general_ci	Check privileges
☐	mysql	utf8mb4_general_ci	Check privileges
☐	performance_schema	utf8_general_ci	Check privileges
☐	phpmyadmin	utf8_bin	Check privileges
☐	piizza2db	utf8mb4_general_ci	Check privileges
☐	pizza3db	utf8mb4_general_ci	Check privileges
☐	pizzadb	utf8mb4_general_ci	Check privileges
☐	test	latin1_swedish_ci	Check privileges
Total: 8			

Create a table and call it ‘**account’** let it have 8 columns



Name: account Number of columns: 4

Got the SQL tab and insert the following values:



localhost/KH4056CEM_AhmedA
localhost / 127.0.0.1 / accountsp
how to insert multiple rows sq
SQL Insert Multiple Rows : Select
localhost/phpmyadmin/tbl_sql.php?db=accountsproject&table=account
Not syncing

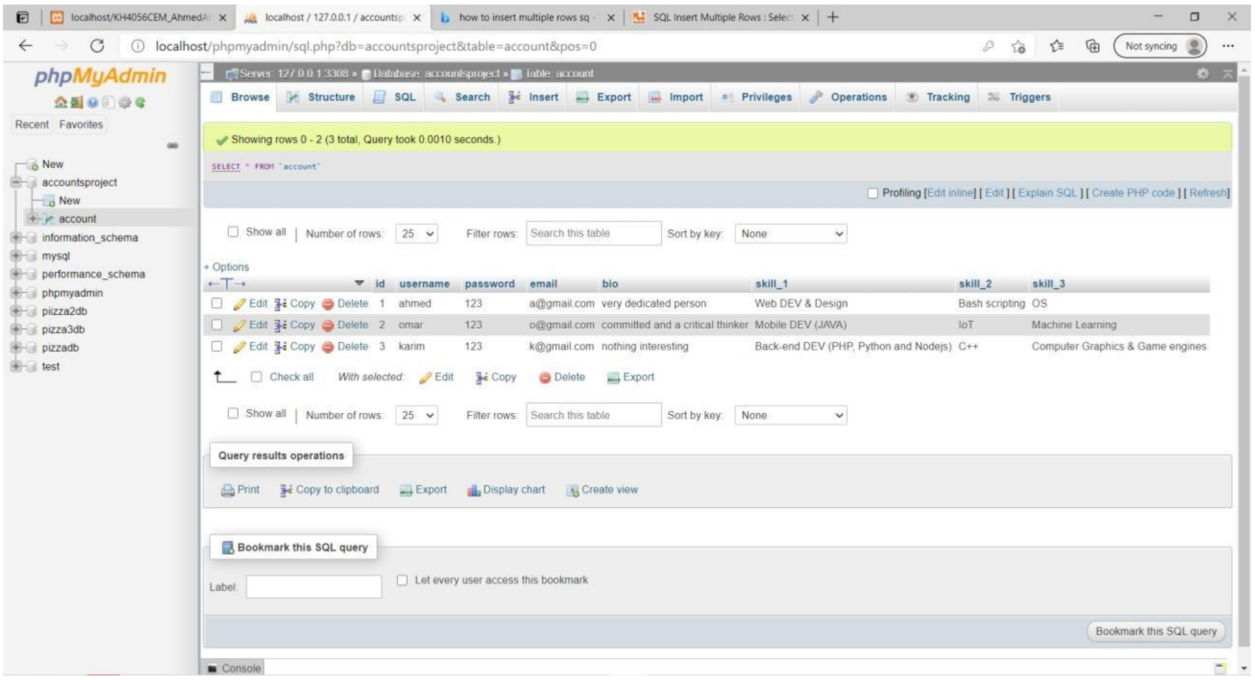
Run SQL query/queries on table accountsproject.account:

```
INSERT INTO account(username,password,email,bio,skill_1,skill_2,skill_3) VALUES
("ahmed","123","a@gmail.com","very dedicated person", "Web DEV & Design", "Bash scripting","OS"),
("omar","123","o@gmail.com","committed and a critical thinker", "Mobile DEV (JAVA)", "IoT","Machine Learning"),
("karim","123","k@gmail.com","nothing interesting", "Back-end DEV (PHP, Python and Nodejs)", "C++","Computer Graphics & Game engines");
```

<u>SQL Query:</u>

```
INSERT INTO account(username,password,email,bio,skill_1,skill_2,skill_3)
VALUES

("ahmed","123","a@gmail.com","very dedicated person", "Web DEV & Design",
"Bash scripting","OS"),
("omar","123","o@gmail.com","committed and a critical thinker", "Mobile DEV
(JAVA)", "IoT","Machine Learning"),

("karim","123","k@gmail.com","nothing interesting", "Back-end DEV (PHP,
Python and Nodejs)", "C++","Computer Graphics & Game engines");
```

Go To **Browse**, you should see the values added.



localhost/phpmyadmin/sql.php?db=accountsproject&table=account&pos=0

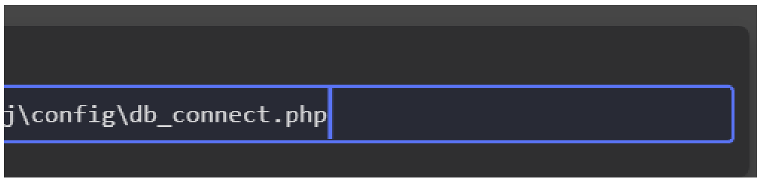
Server: 127.0.0.1:3308 » Database: accountsproject » Table: account

Showing rows 0 - 2 (3 total, Query took 0.0010 seconds.)

```
SELECT * FROM `account`
```

id	username	password	email	bio	skill_1	skill_2	skill_3
1	ahmed	123	a@gmail.com	very dedicated person	Web DEV & Design	Bash scripting	OS
2	omar	123	o@gmail.com	committed and a critical thinker	Mobile DEV (JAVA)	IoT	Machine Learning
3	karim	123	k@gmail.com	nothing interesting	Back-end DEV (PHP, Python and Nodejs)	C++	Computer Graphics & Game engines

In Atom, inside the **config** folder, Create a file called **db_connect.php**

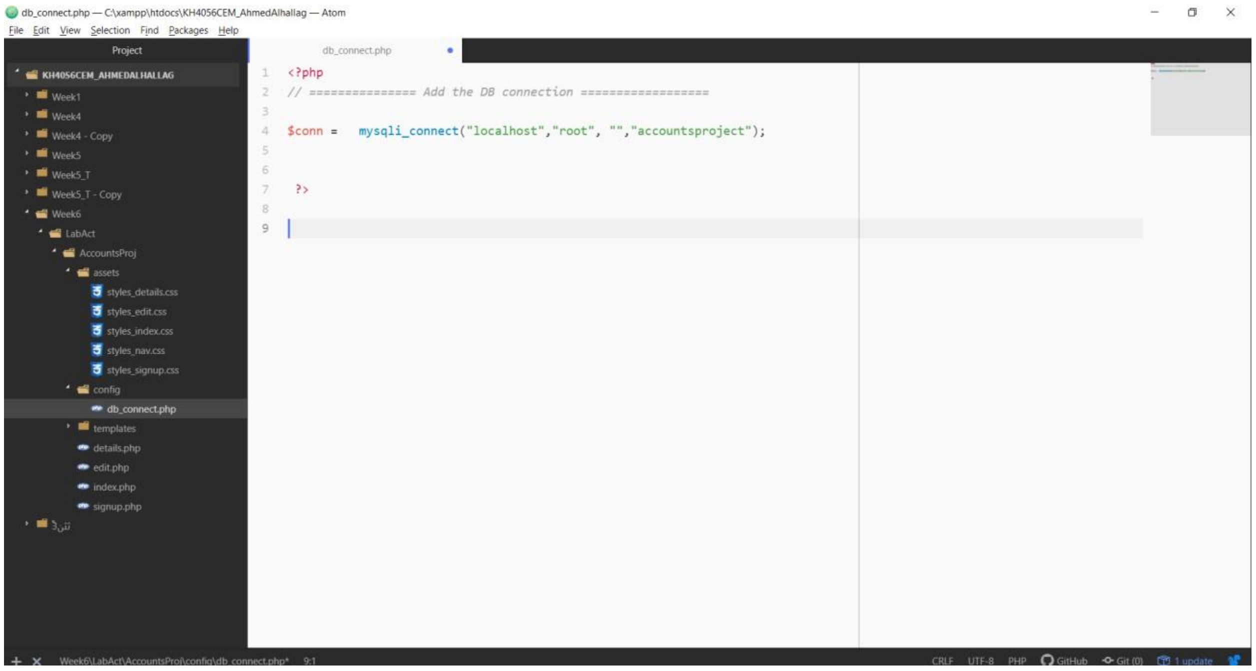


```
j\config\db_connect.php
```

Write the following line to create the connection to the MySQL server and to the database we created.

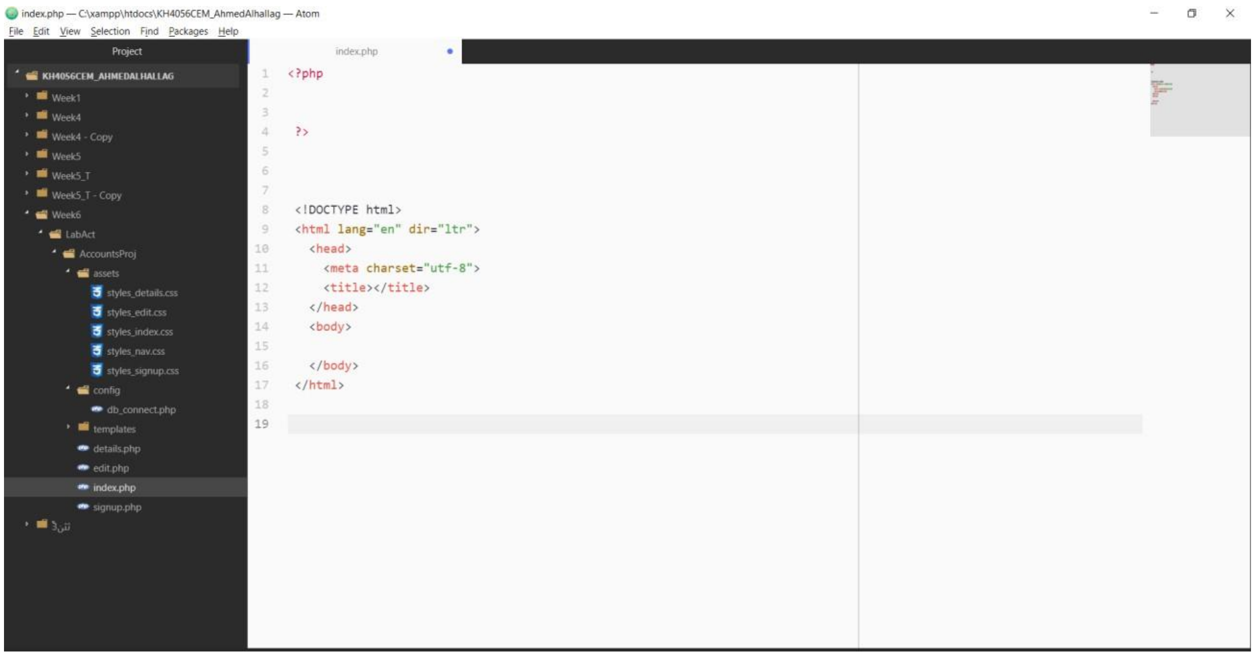
<u>PHP statement:</u>

```
$conn =   mysqli_connect("localhost","root", "","accountsproject");
```



```php
<?php
// ============== Add the DB connection =================

$conn =   mysqli_connect("localhost","root", "","accountsproject");


 ?>

```

Create an **index.php** directly under the project's folder directory. Create the php and html portions in the document.



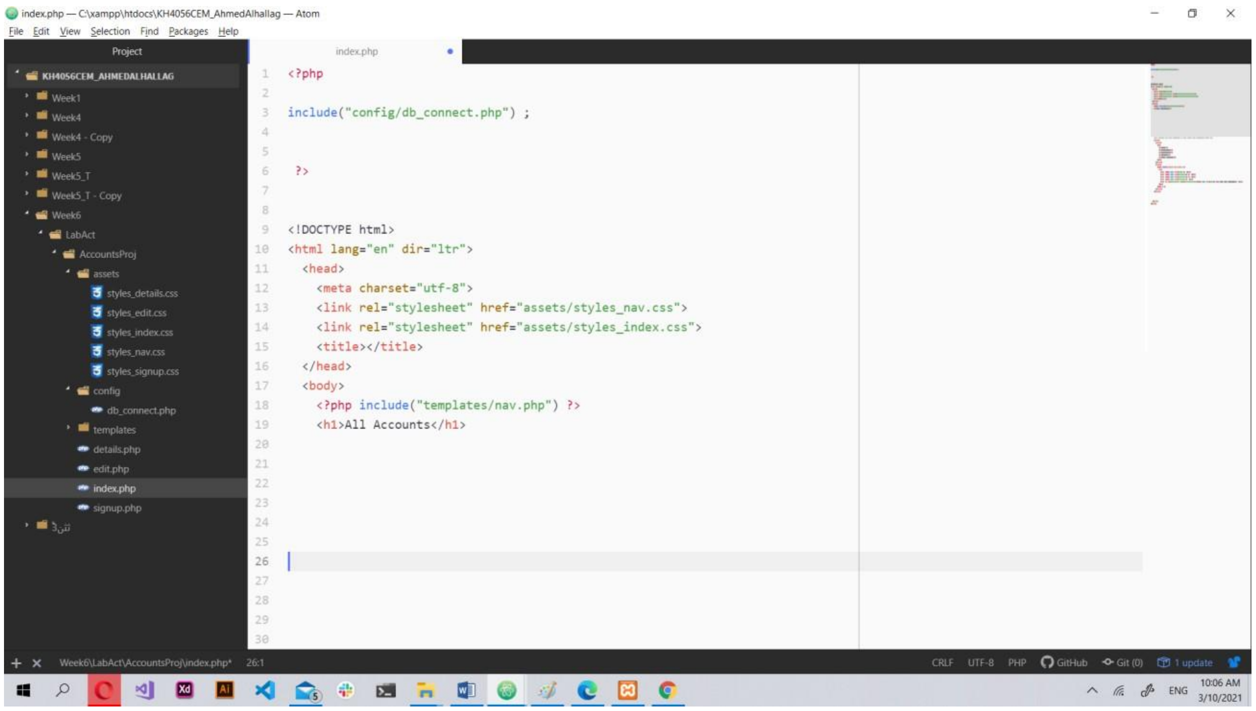
```html
<?php


 ?>



 <!DOCTYPE html>
 <html lang="en" dir="ltr">
   <head>
     <meta charset="utf-8">
     <title></title>
   </head>
   <body>

   </body>
 </html>

```

In the PHP portion, include the connection file from config.

In the HTML portion, Include the navigation bar file, **nav.php** from the **templates** folder (will be created shortly).

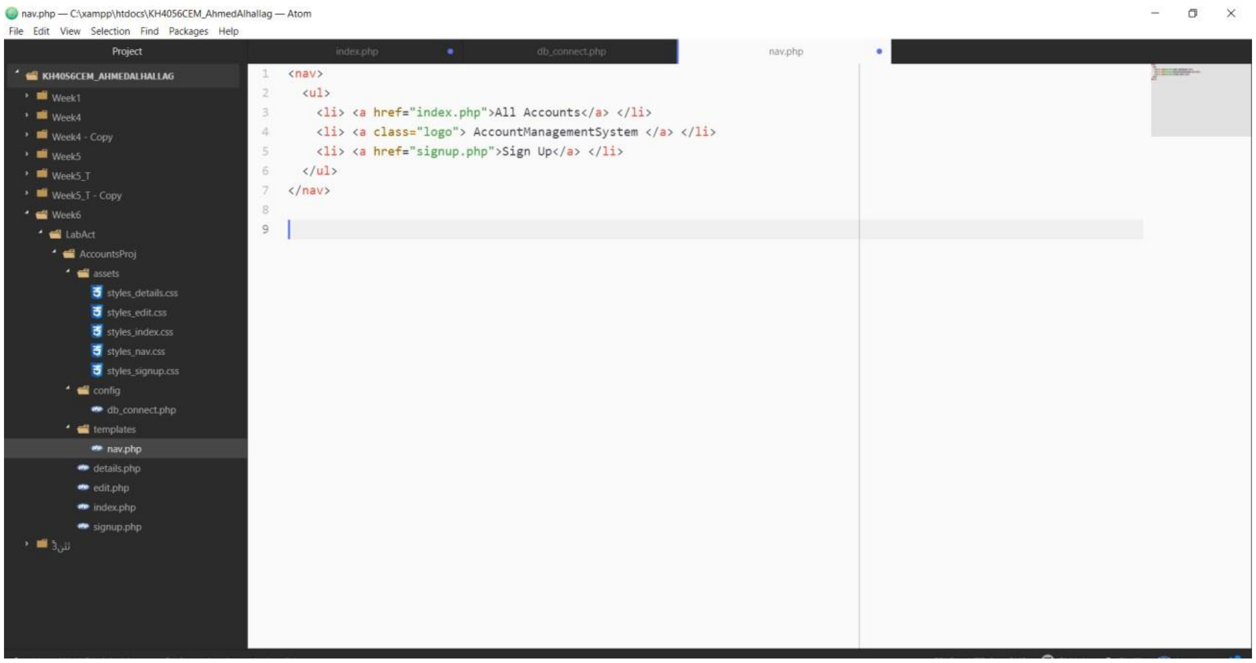
Write down the links needed for styling from the **assets** folder.

Add a heading.



```
<?php

include("config/db_connect.php") ;


 ?>


<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="assets/styles_nav.css">
    <link rel="stylesheet" href="assets/styles_index.css">
    <title></title>
  </head>
  <body>
    <?php include("templates/nav.php") ?>
    <h1>All Accounts</h1>
```

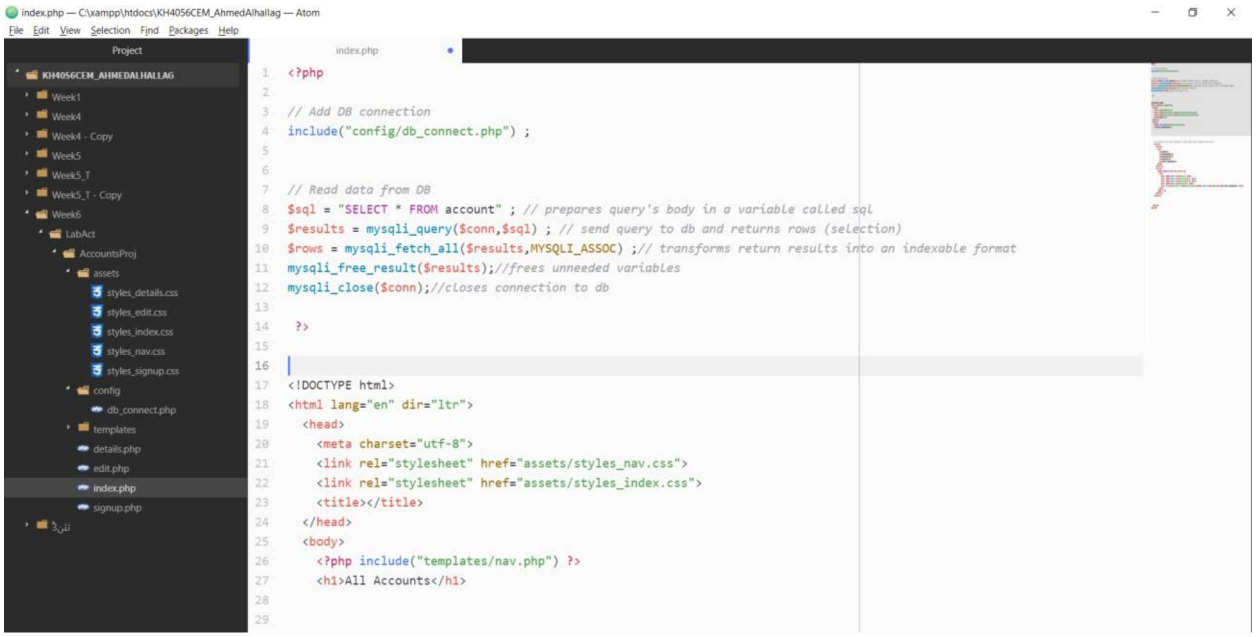
Create a file called ‘**nav.php’** in the **templates** folder. Write the following lines in it:



```
<nav>
  <ul>
    <li> <a href="index.php">All Accounts</a> </li>
    <li> <a class="logo"> AccountManagementSystem </a> </li>
    <li> <a href="signup.php">Sign Up</a> </li>
  </ul>
</nav>
```

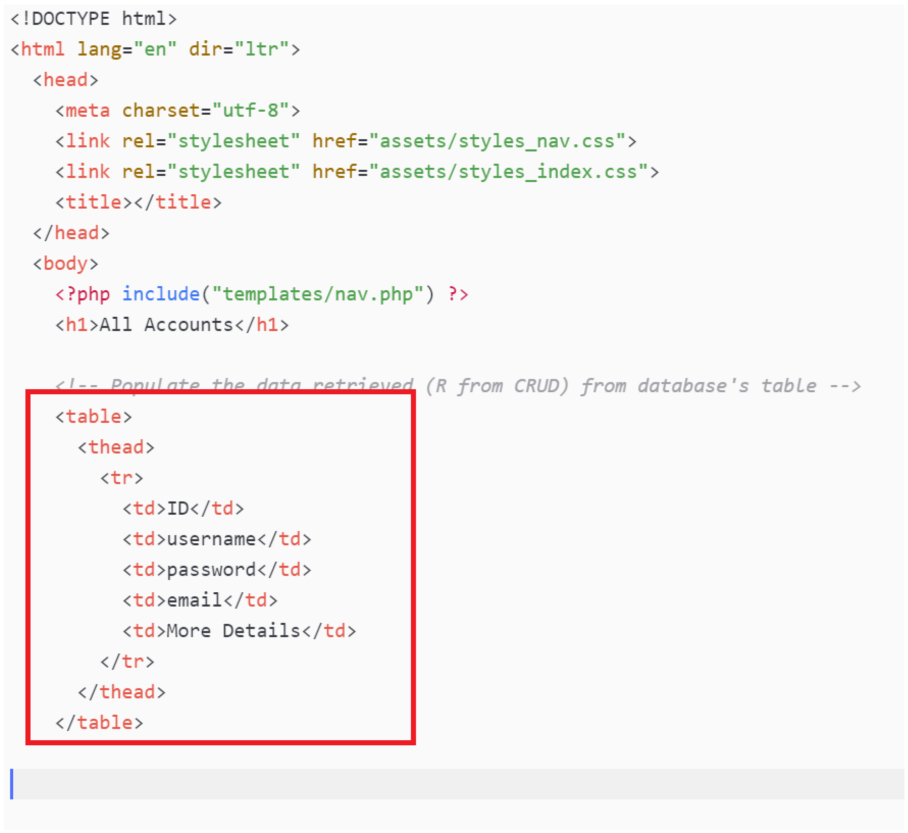
Perform the Read operation from CRUD:

Go back to the **index.php**, And write the following lines. (You don't have to write down the comments)



```php
<?php

// Add DB connection
include("config/db_connect.php") ;


// Read data from DB
$sql = "SELECT * FROM account" ; // prepares query's body in a variable called sql
$results = mysqli_query($conn,$sql) ; // send query to db and returns rows (selection)
$rows = mysqli_fetch_all($results,MYSQLI_ASSOC) ;// transforms return results into an indexable format
mysqli_free_result($results);//frees unneeded variables
mysqli_close($conn);//closes connection to db

 ?>


<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="assets/styles_nav.css">
    <link rel="stylesheet" href="assets/styles_index.css">
    <title></title>
  </head>
  <body>
    <?php include("templates/nav.php") ?>
    <h1>All Accounts</h1>
```

In the HTML portion again, create a table with five headers or columns as follows:

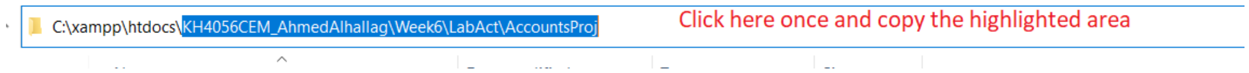


```html
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="assets/styles_nav.css">
    <link rel="stylesheet" href="assets/styles_index.css">
    <title></title>
  </head>
  <body>
    <?php include("templates/nav.php") ?>
    <h1>All Accounts</h1>

    <!-- Populate the data retrieved (R from CRUD) from database's table -->
    <table>
      <thead>
        <tr>
          <td>ID</td>
          <td>username</td>
          <td>password</td>
          <td>email</td>
          <td>More Details</td>
        </tr>
      </thead>
    </table>
```

Check how the page looks so far on the browser.

To figure out the working path or URL, In Atom right click on the index.php and choose “Show in File Explorer”

Once the folder which the file is at has opened, copy the following path (NO WHITE SPACES SHOULD BE PRESENT IN ALL YOUR FILES OR FOLDERS NAMES)


C:\xampp\htdocs\KH4056CEM_AhmedAlhallag\Week6\LabAct\AccountsProj

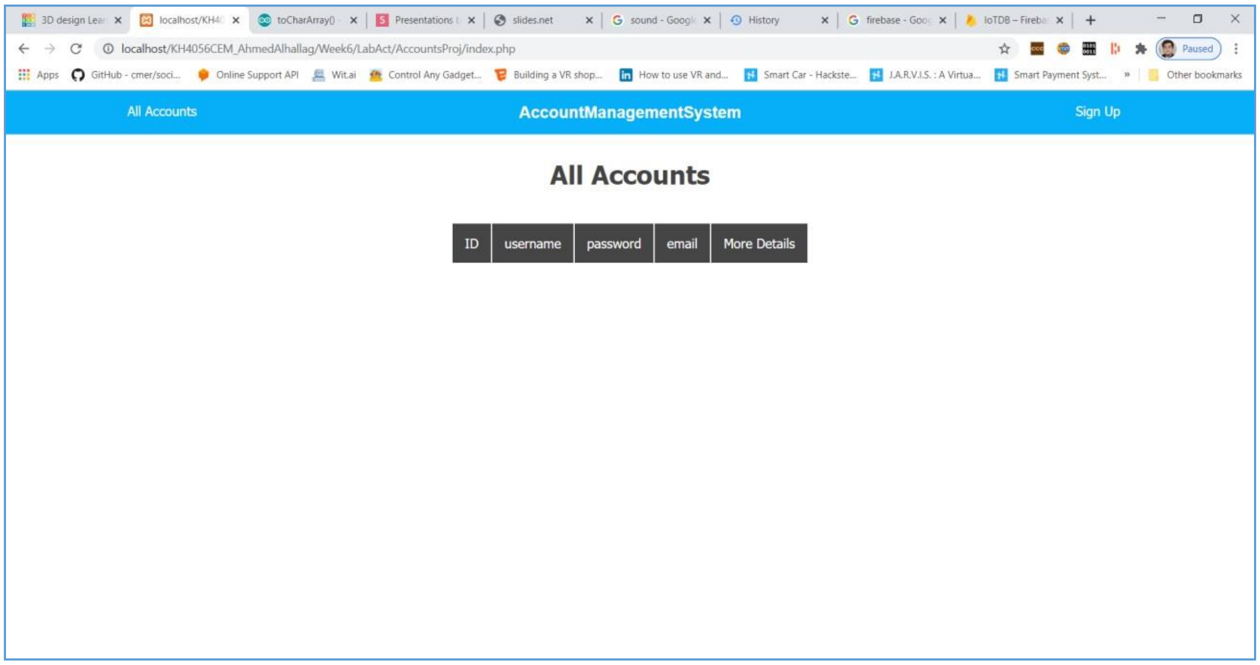
In your browser, paste the copied segment after localhost/.

Example:

localhost/KH4056CEM_AhmedAlhallag\Week6\LabAct\AccountsProj/

And hit enter.

You should see:

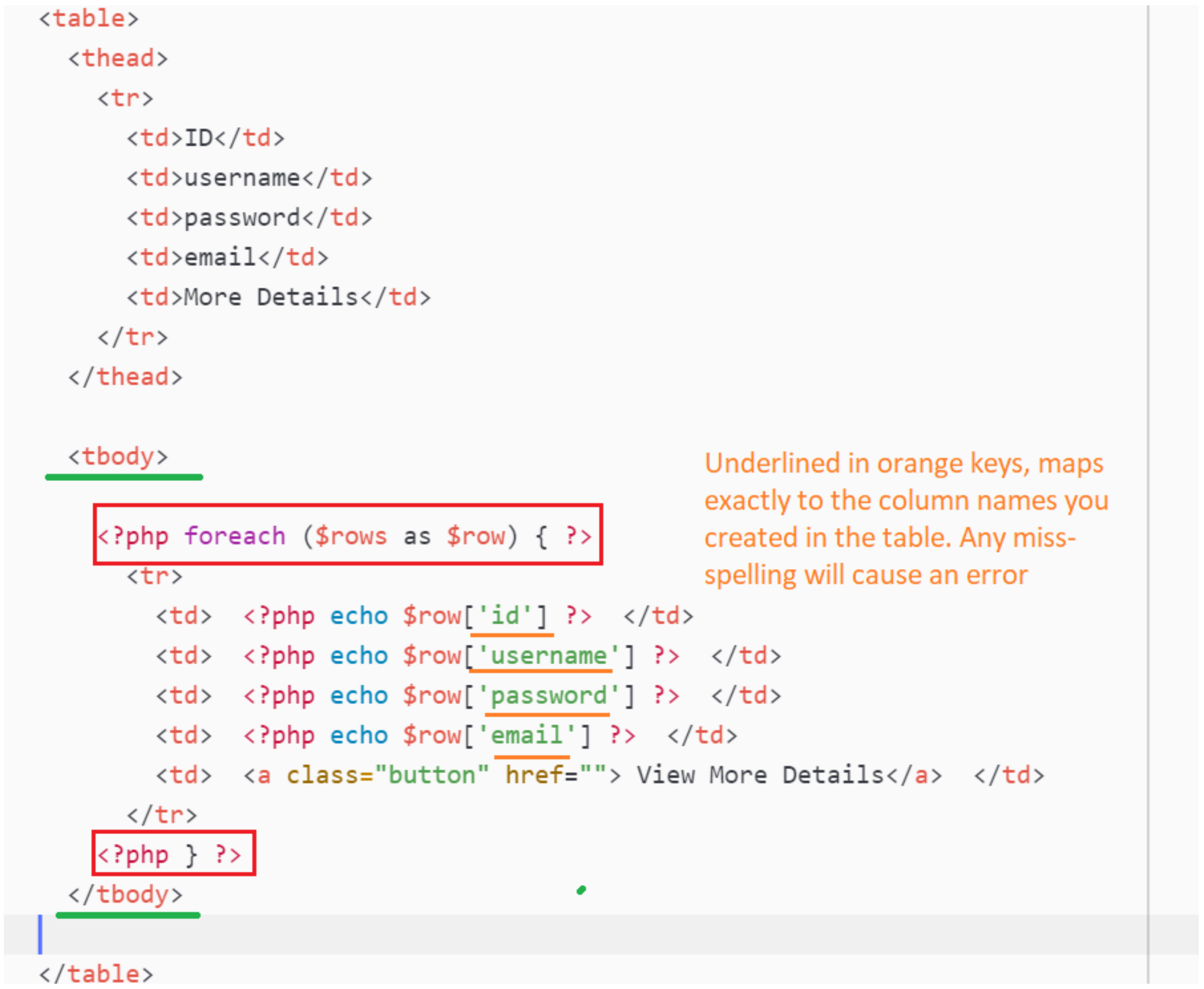

localhost/KH4056CEM_AhmedAlhallag/Week6/LabAct/AccountsProj/index.php

All Accounts

ID	username	password	email	More Details

Now Populate the data using a foreach loop. The loop should generate a table row (tr), every time It executes. Within every row created, we will create 5 data cells (td), respectively for every column.

The first 4 data cells, will map to the value under each column in the database’s ‘account’ table. This fifth will be a link that will take us to each individual’s account page.



```
<table>
  <thead>
    <tr>
      <td>ID</td>
      <td>username</td>
      <td>password</td>
      <td>email</td>
      <td>More Details</td>
    </tr>
  </thead>

  <tbody>

    <?php foreach ($rows as $row) { ?>
      <tr>
        <td>  <?php echo $row['id'] ?>  </td>
        <td>  <?php echo $row['username'] ?>  </td>
        <td>  <?php echo $row['password'] ?>  </td>
        <td>  <?php echo $row['email'] ?>  </td>
        <td>  <a class="button" href=""> View More Details</a>  </td>
      </tr>
    <?php } ?>
  </tbody>

</table>
```

Underlined in orange keys, maps exactly to the column names you created in the table. Any miss-spelling will cause an error

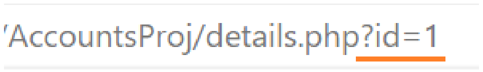
Notice that the link for the anchor tag (href) is still empty. We need to create a details.php page for every account. It does not make sense to "literally" create a details.php for every account/row in database like this:

Details_id_1.php

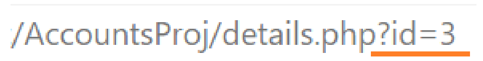
Details_id_2.php

And so on …

Instead, we will rely on "**Query Strings**", where only a dingle details page is created, and it has it's content rendered dynamically from a small attached **parameter** in from the URL. For example:



'AccountsProj/details.php?id=1



/AccountsProj/details.php?id=3

/AccountsProj/details.php?id=2

Notice how a **query string** is written:

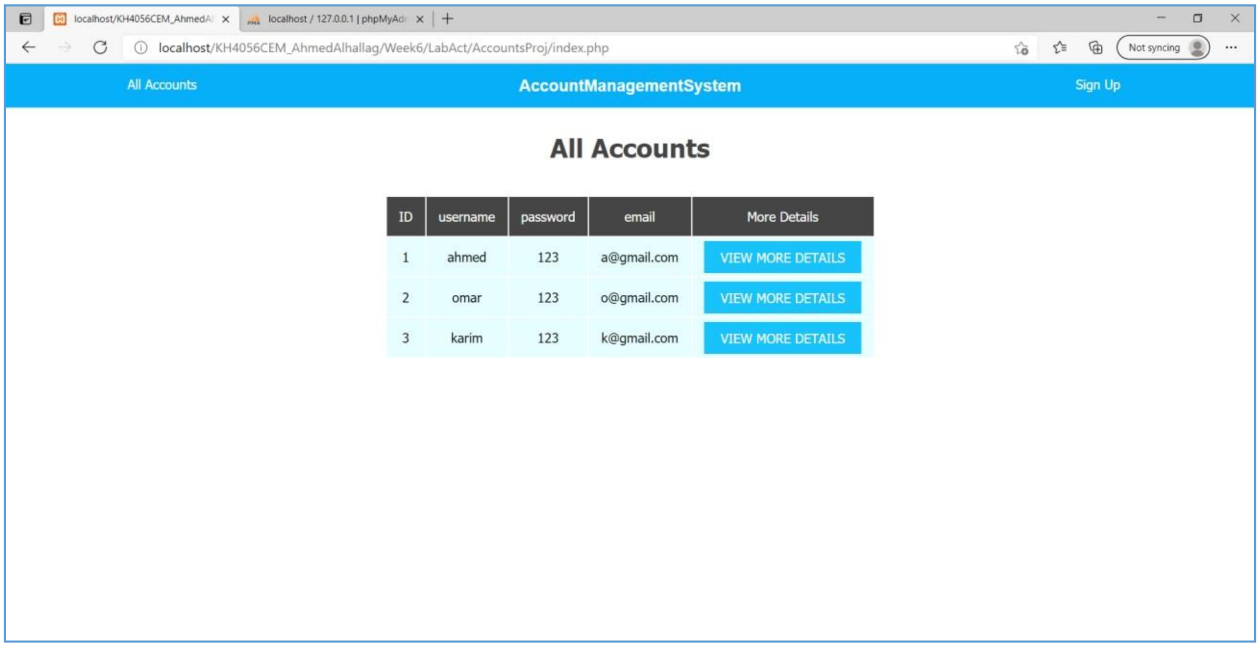
/<filename>.*php*?<parameter of your choice> = <value>

```
<tbody>

  <?php foreach ($rows as $row) { ?>
    <tr>
      <td>  <?php echo $row['id'] ?>  </td>
      <td>  <?php echo $row['username'] ?>  </td>
      <td>  <?php echo $row['password'] ?>  </td>
      <td>  <?php echo $row['email'] ?>  </td>
      <td>  <a class="button" href="details.php?id=<?php echo $row['id']; ?>">View More Details</a>  </td>
    </tr>
  <?php } ?>
```

The parameter name after the question mark, id, can be named to your liking, as long as it stays consistent across all pages/routes.

Since we already have access to ids retrieved from db, attach the value of the id grabbed from the database after the equal sign.

You Should see the following: (if you click on the links nothing will happen since we didn't create the details.php page yet)


localhost/KH4056CEM_AhmedAl
localhost / 127.0.0.1 | phpMyAdr
localhost/KH4056CEM_AhmedAlhallag/Week6/LabAct/AccountsProj/index.php
Not syncing
All Accounts
AccountManagementSystem
Sign Up

All Accounts

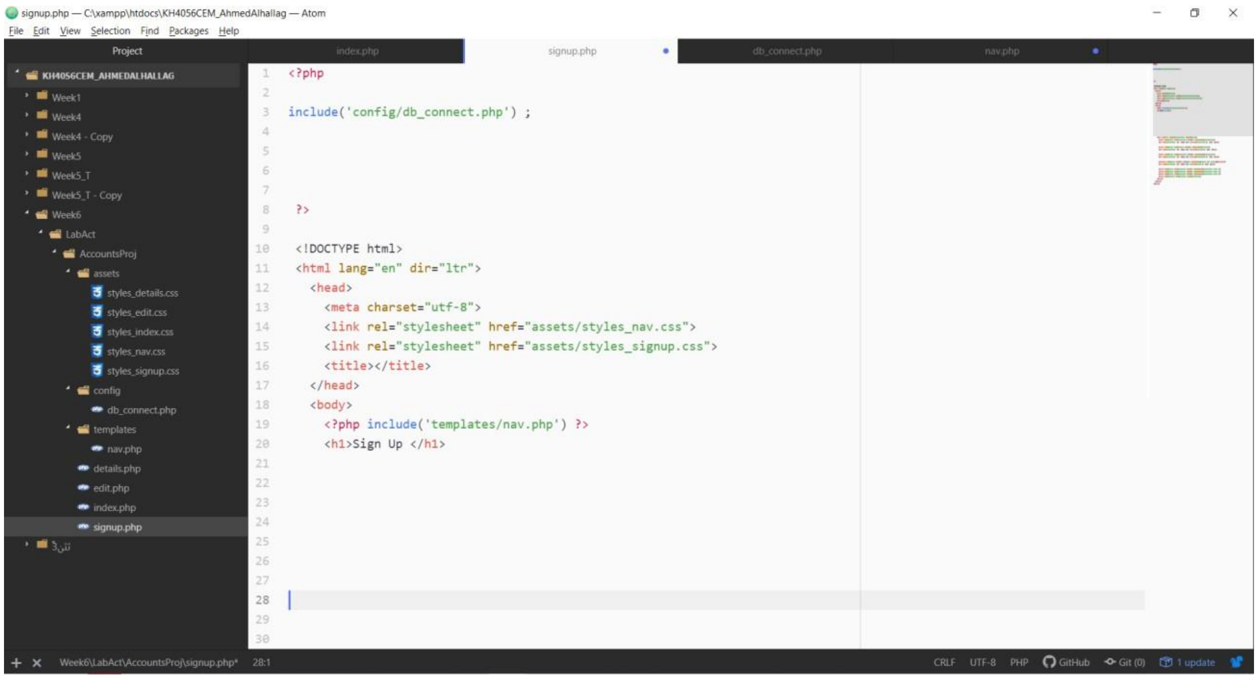
ID	username	password	email	More Details
1	ahmed	123	a@gmail.com	VIEW MORE DETAILS
2	omar	123	o@gmail.com	VIEW MORE DETAILS
3	karim	123	k@gmail.com	VIEW MORE DETAILS

Perform the Create operation from CRUD:

In the project's directory, create a signup.php page.

Create the php portion and html portions in the document like from before.

Add the db connections, nav bar, styles and a heading.

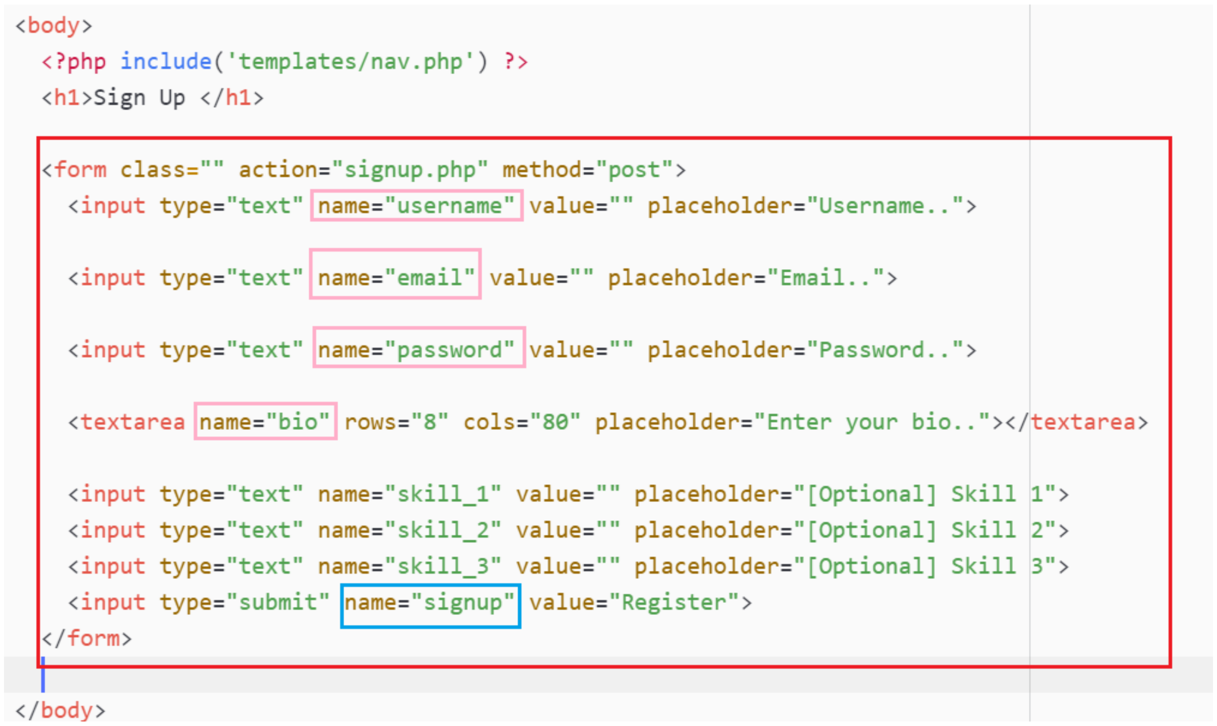


```
<?php

include('config/db_connect.php') ;




 ?>

 <!DOCTYPE html>
 <html lang="en" dir="ltr">
   <head>
     <meta charset="utf-8">
     <link rel="stylesheet" href="assets/styles_nav.css">
     <link rel="stylesheet" href="assets/styles_signup.css">
     <title></title>
   </head>
   <body>
     <?php include('templates/nav.php') ?>
     <h1>Sign Up </h1>
```

Create the following form.

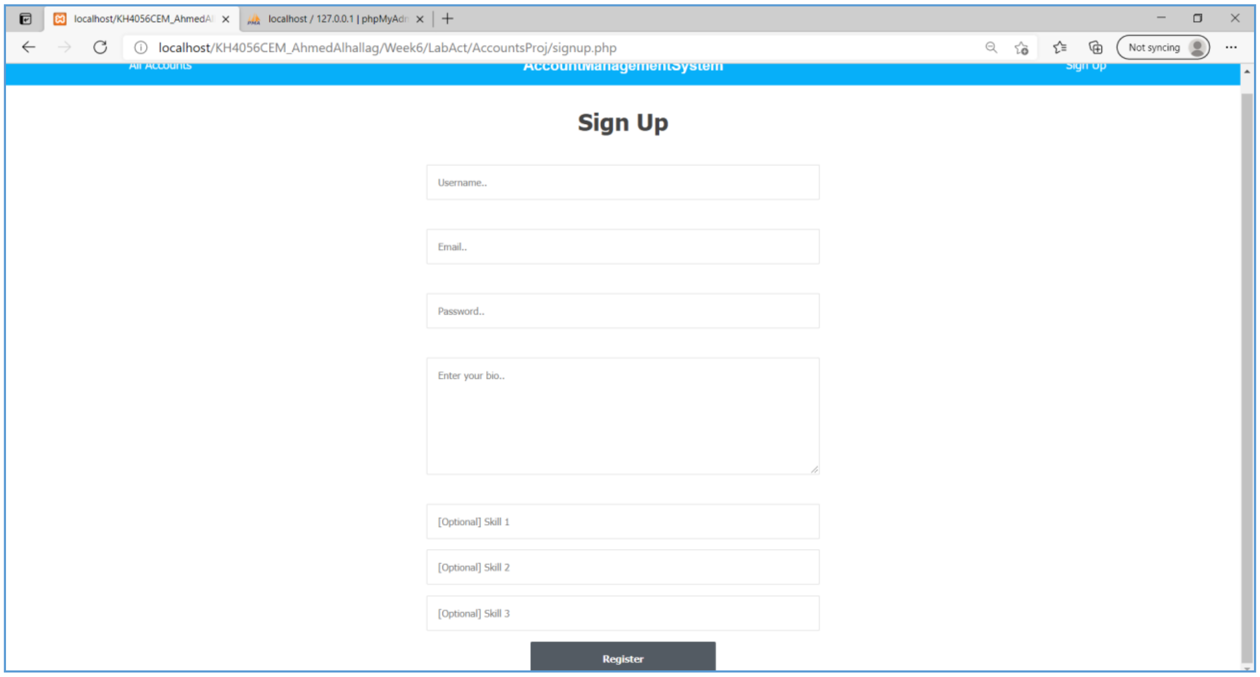
Notice the name of every input field; these will be used to fetch the values from the HTML input fields which the user will enter.

Notice the name if the submit-type-input, '**signup**', this is what will be used **to send the data from all input fields values and redirect them back to the php portion of the same signup.php page (hence the action is set to the same page's name) that we are in once we click in the submit button.**

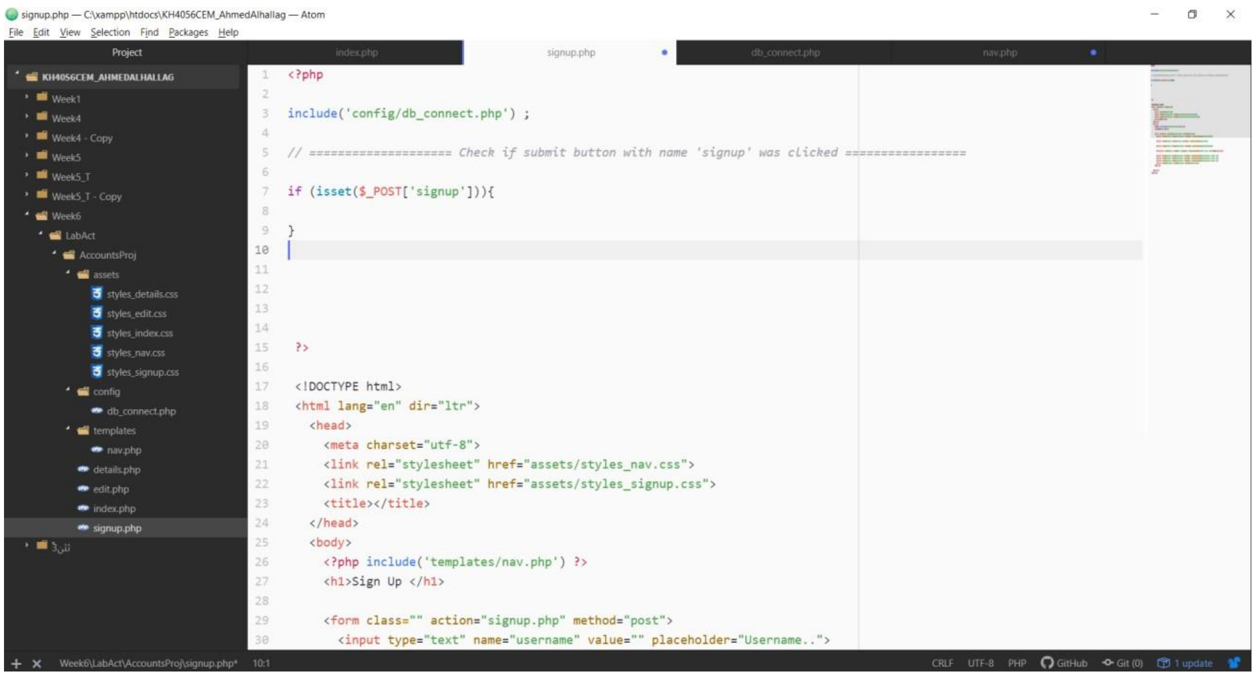


```
<body>
  <?php include('templates/nav.php') ?>
  <h1>Sign Up </h1>

  <form class="" action="signup.php" method="post">
    <input type="text" name="username" value="" placeholder="Username..">

    <input type="text" name="email" value="" placeholder="Email..">

    <input type="text" name="password" value="" placeholder="Password..">

    <textarea name="bio" rows="8" cols="80" placeholder="Enter your bio.."></textarea>

    <input type="text" name="skill_1" value="" placeholder="[Optional] Skill 1">
    <input type="text" name="skill_2" value="" placeholder="[Optional] Skill 2">
    <input type="text" name="skill_3" value="" placeholder="[Optional] Skill 3">
    <input type="submit" name="signup" value="Register">
  </form>

</body>
```

You should see this if you refreshed in browser on the signup.php page:



In the php portion, check if the submit button that has the name ‘signup’ that we gave in the HTML part was clicked using the isset function. Note the protocol or the medium we are using to forward values from a page to the other is the POST method.

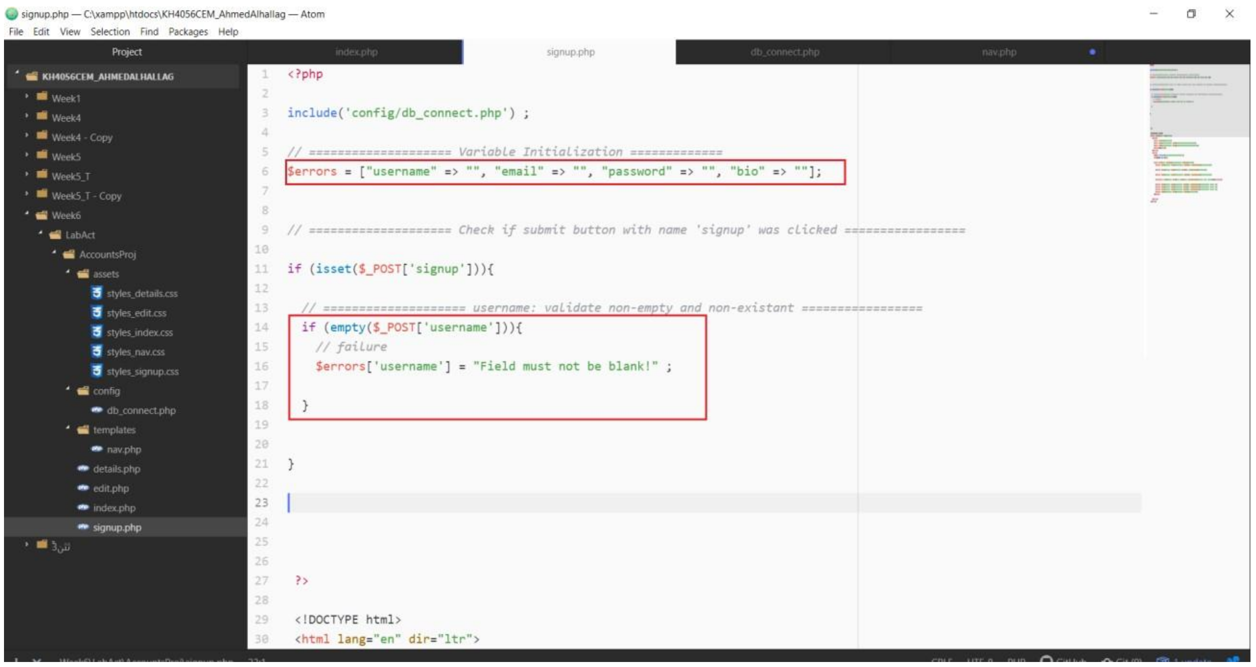


```php
<?php

include('config/db_connect.php') ;

// =================== Check if submit button with name 'signup' was clicked ================

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

}





 ?>

 <!DOCTYPE html>
 <html lang="en" dir="ltr">
   <head>
     <meta charset="utf-8">
     <link rel="stylesheet" href="assets/styles_nav.css">
     <link rel="stylesheet" href="assets/styles_signup.css">
     <title></title>
   </head>
   <body>
     <?php include('templates/nav.php') ?>
     <h1>Sign Up </h1>

     <form class="" action="signup.php" method="post">
       <input type="text" name="username" value="" placeholder="Username..">
```

Since the insertion method used in the SQL languages inserts whole rows, we need to make sure that every value fetched from the post method by the page once we hit on the submit button, is valid.

Start by validating the username.

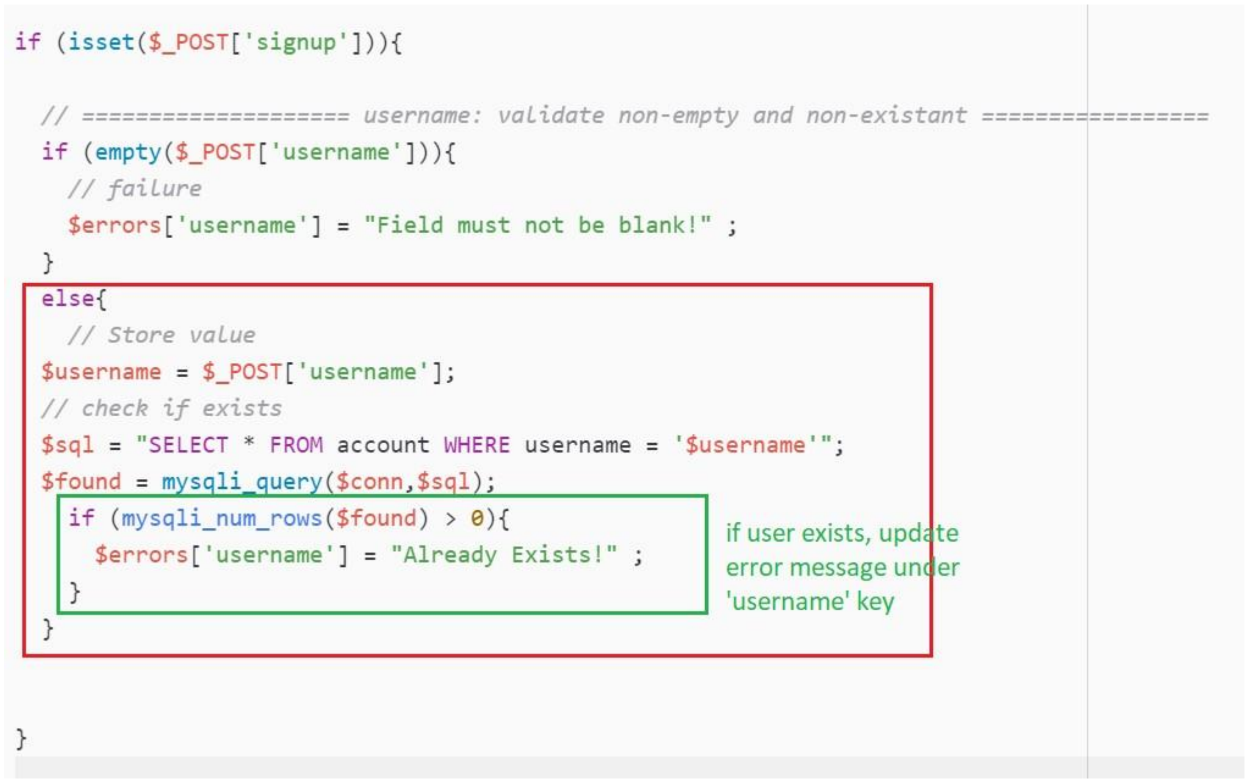
Validation used in this tutorial are for emptiness and existence. Elaborate validation levels based on your project.



```php
<?php

include('config/db_connect.php') ;

// =================== Variable Initialization ============
$errors = ["username" => "", "email" => "", "password" => "", "bio" => ""];


// =================== Check if submit button with name 'signup' was clicked ================

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

  // =================== username: validate non-empty and non-existant ================
  if (empty($_POST['username'])){
    // failure
    $errors['username'] = "Field must not be blank!" ;

  }


}





 ?>

 <!DOCTYPE html>
 <html lang="en" dir="ltr">
```

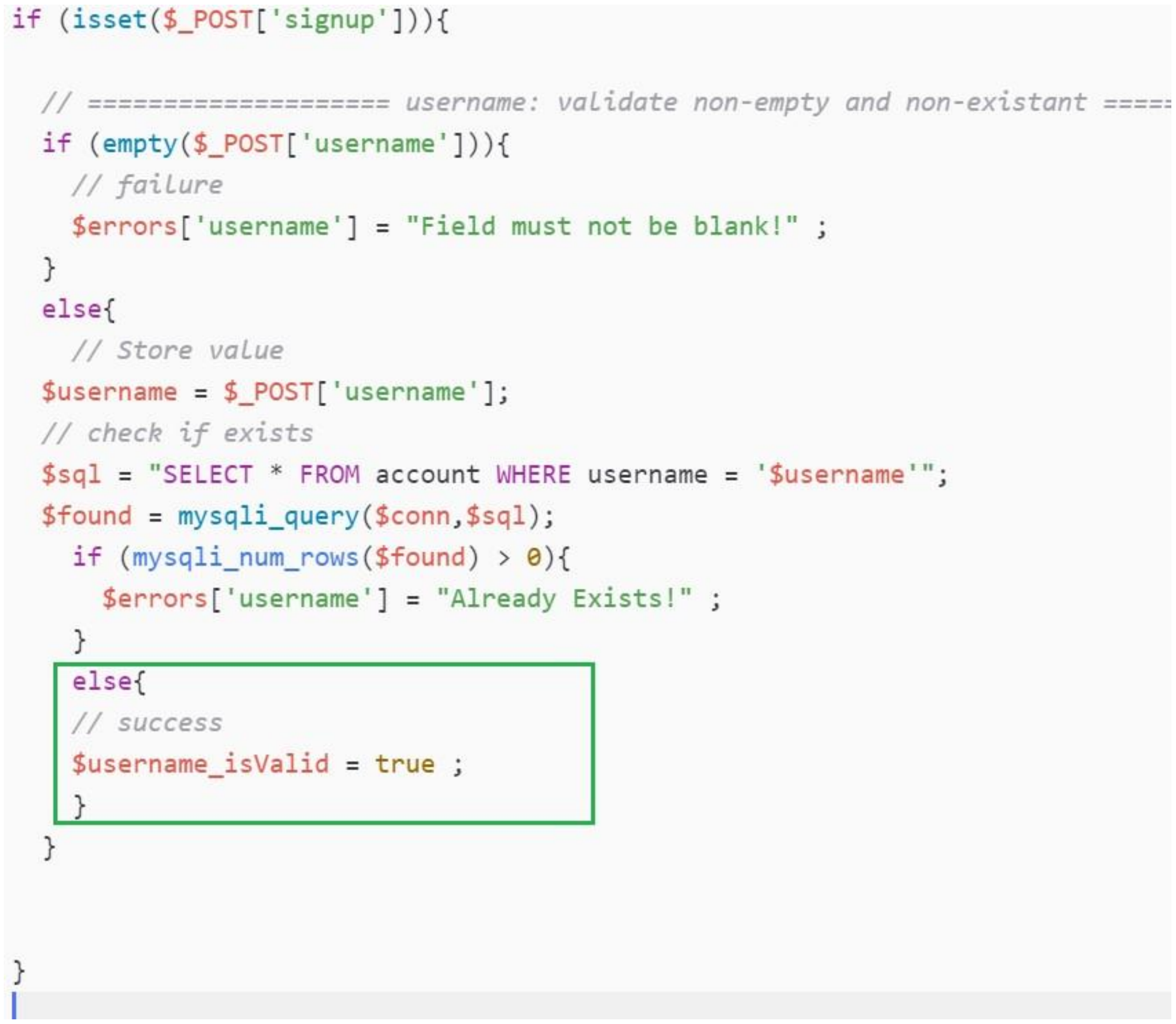
What if it was not empty? Next validation check: User should not be already existent in database.

Remember we created our username column to have the ‘UNIQUE’ index, so inserting a redundant value will cause an error. We need to avoid that.



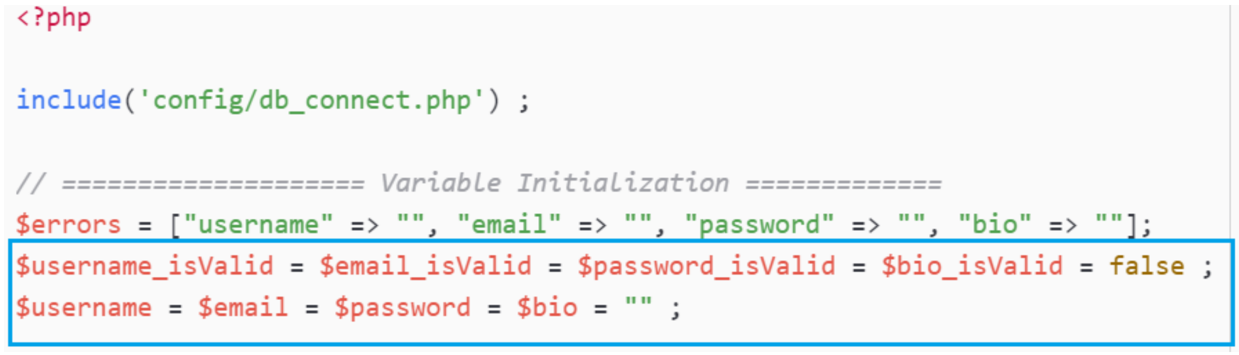
```php
if (isset($_POST['signup'])){

  // ==================== username: validate non-empty and non-existant =================
  if (empty($_POST['username'])){
    // failure
    $errors['username'] = "Field must not be blank!" ;
  }
  else{
    // Store value
  $username = $_POST['username'];
  // check if exists
  $sql = "SELECT * FROM account WHERE username = '$username'";
  $found = mysqli_query($conn,$sql);
    if (mysqli_num_rows($found) > 0){
      $errors['username'] = "Already Exists!" ;
    }
  }


}
```

What If it was not empty and user is not in the database? Then the user value should be valid.



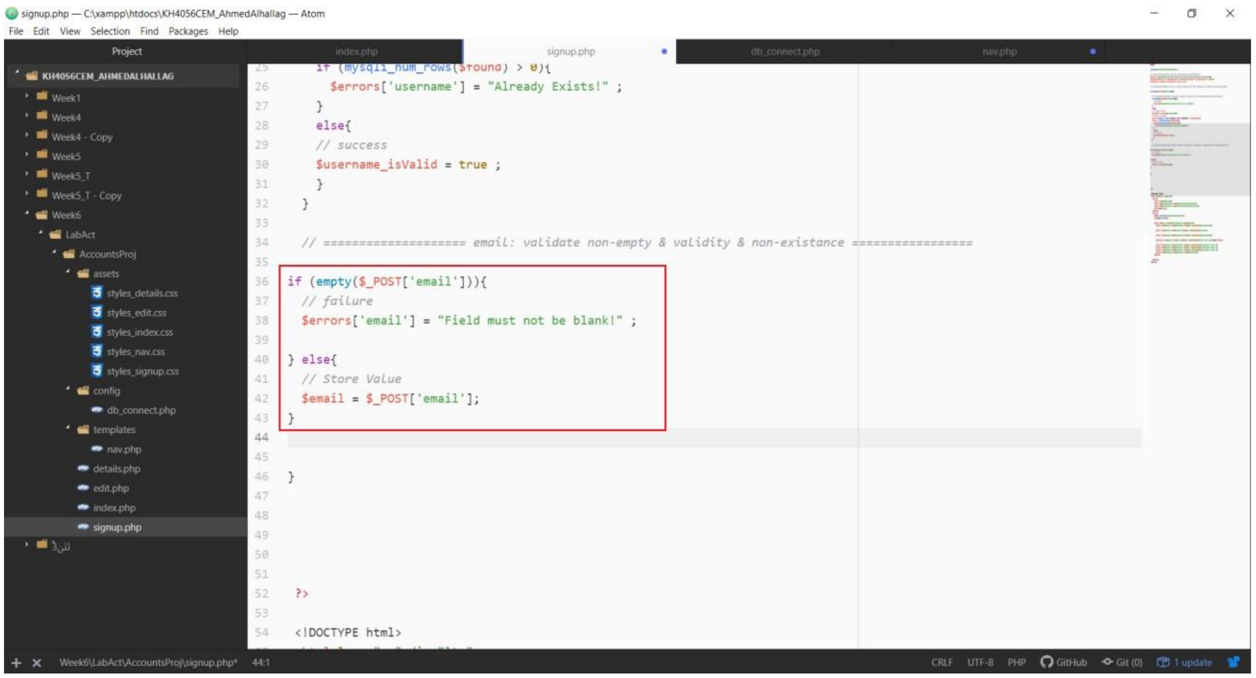
```
if (isset($_POST['signup'])){

  // ==================== username: validate non-empty and non-existant =====
  if (empty($_POST['username'])){
    // failure
    $errors['username'] = "Field must not be blank!" ;
  }
  else{
    // Store value
  $username = $_POST['username'];
  // check if exists
  $sql = "SELECT * FROM account WHERE username = '$username'";
  $found = mysqli_query($conn,$sql);
    if (mysqli_num_rows($found) > 0){
      $errors['username'] = "Already Exists!" ;
    }
    else{
    // success
    $username_isValid = true ;
    }
  }


}
```

Make sure to initialize the following variables at the start:



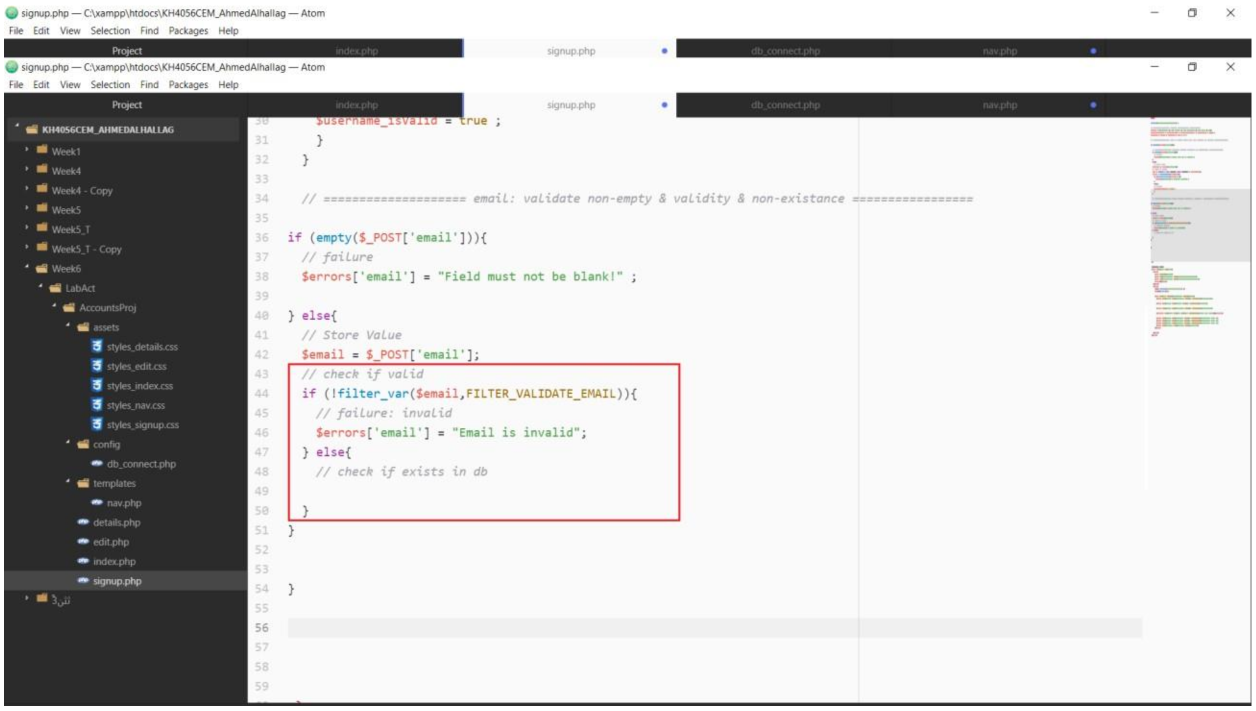
```
<?php

include('config/db_connect.php') ;

// ==================== Variable Initialization =============
$errors = ["username" => "", "email" => "", "password" => "", "bio" => ""];
$username_isValid = $email_isValid = $password_isValid = $bio_isValid = false ;
$username = $email = $password = $bio = "" ;
```

Next validate the email. Start by making sure the email value in the input field named “email” is not fetched by the post method when submit button is clicked is not empty.



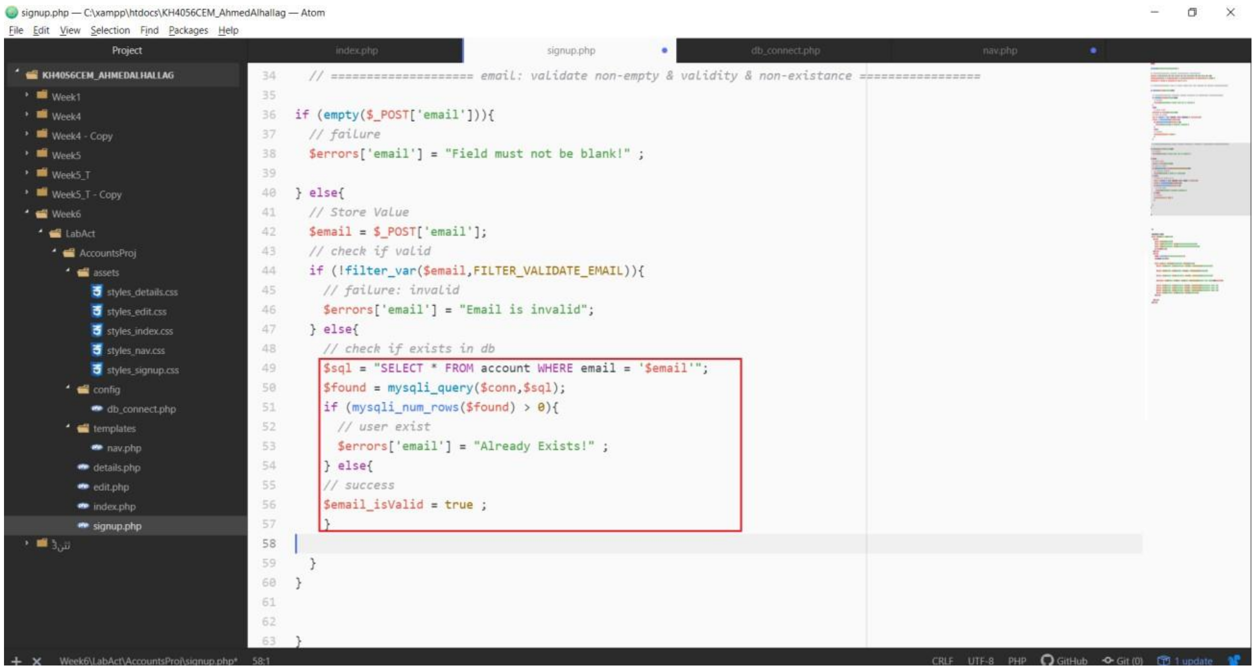
```
    if (mysqli_num_rows($found) > 0){
      $errors['username'] = "Already Exists!" ;
    }
    else{
    // success
    $username_isValid = true ;
    }
  }

  // =================== email: validate non-empty & validity & non-existance ================

if (empty($_POST['email'])){
  // failure
  $errors['email'] = "Field must not be blank!" ;

} else{
  // Store Value
  $email = $_POST['email'];
}


}





 ?>

 <!DOCTYPE html>
```

If not empty, then check if the pattern entered is valid:



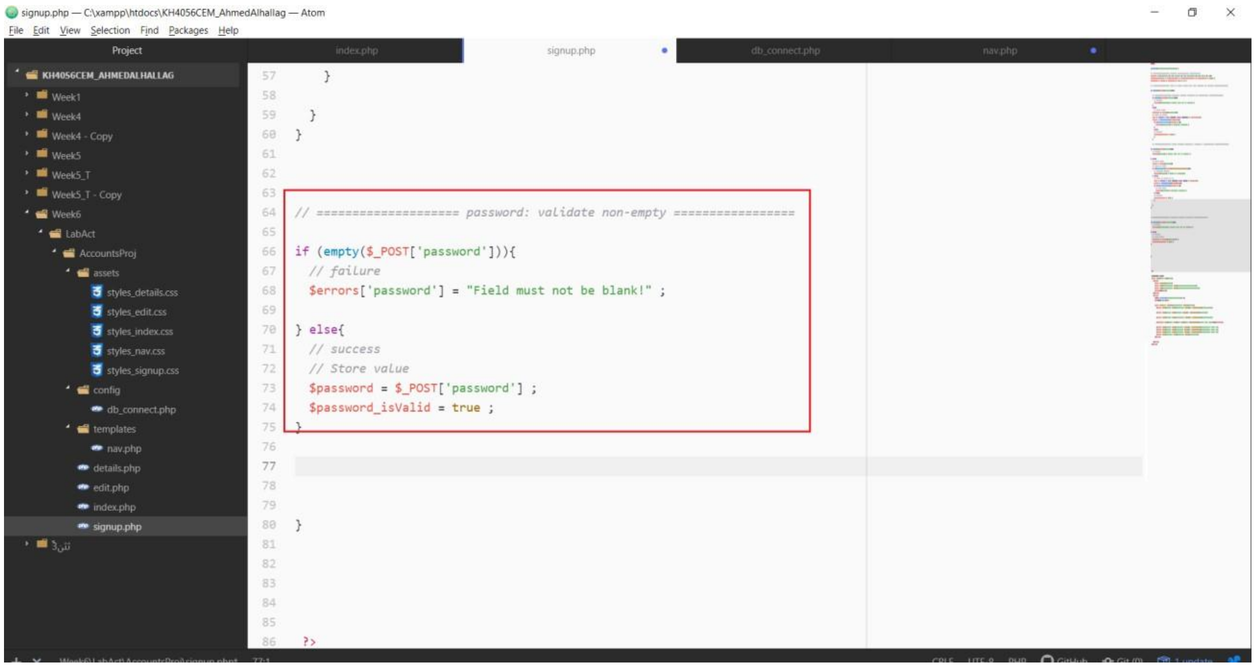
```
    $username_isValid = true ;
    }
  }

  // =================== email: validate non-empty & validity & non-existance ================

if (empty($_POST['email'])){
  // failure
  $errors['email'] = "Field must not be blank!" ;

} else{
  // Store Value
  $email = $_POST['email'];
  // check if valid
  if (!filter_var($email,FILTER_VALIDATE_EMAIL)){
    // failure: invalid
    $errors['email'] = "Email is invalid";
  } else{
    // check if exists in db

  }
}


}
```

If valid, lastly perform the same existence check since it has an index of “UNIQUE” as well in database.



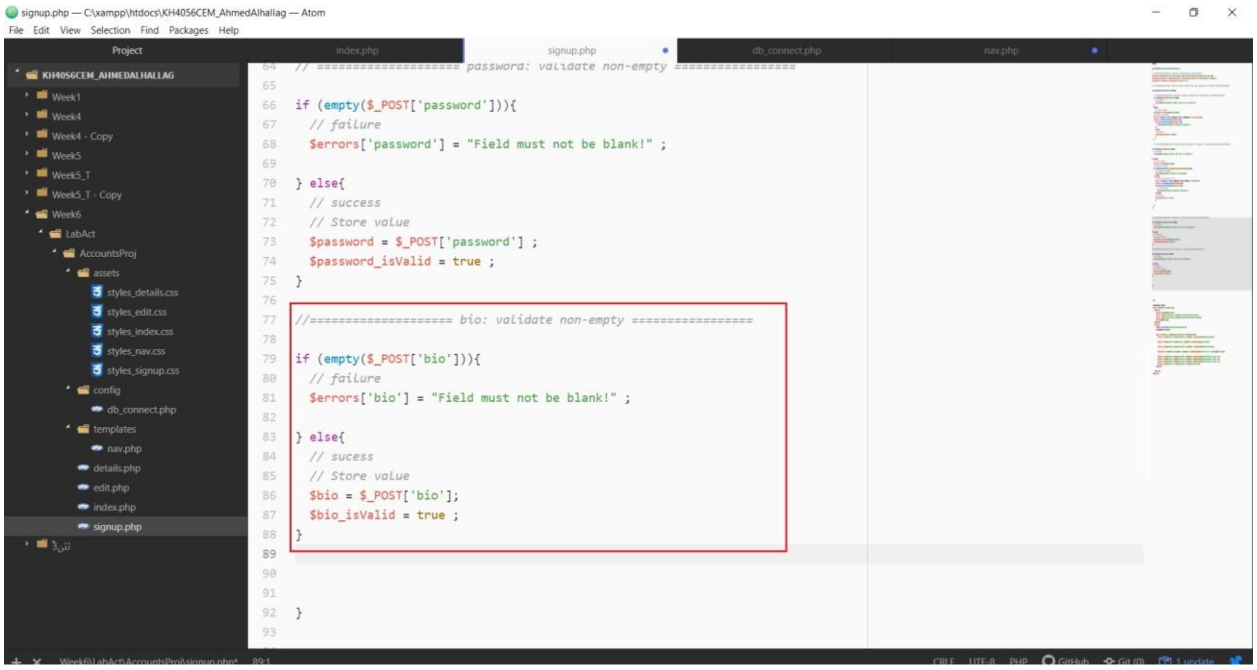
```
  // =================== email: validate non-empty & validity & non-existance ================

if (empty($_POST['email'])){
  // failure
  $errors['email'] = "Field must not be blank!" ;

} else{
  // Store Value
  $email = $_POST['email'];
  // check if valid
  if (!filter_var($email,FILTER_VALIDATE_EMAIL)){
    // failure: invalid
    $errors['email'] = "Email is invalid";
  } else{
    // check if exists in db
    $sql = "SELECT * FROM account WHERE email = '$email'";
    $found = mysqli_query($conn,$sql);
    if (mysqli_num_rows($found) > 0){
      // user exist
      $errors['email'] = "Already Exists!" ;
    } else{
    // success
    $email_isValid = true ;
    }

  }
}


}
```

For the password we need just to make sure that it is not empty:



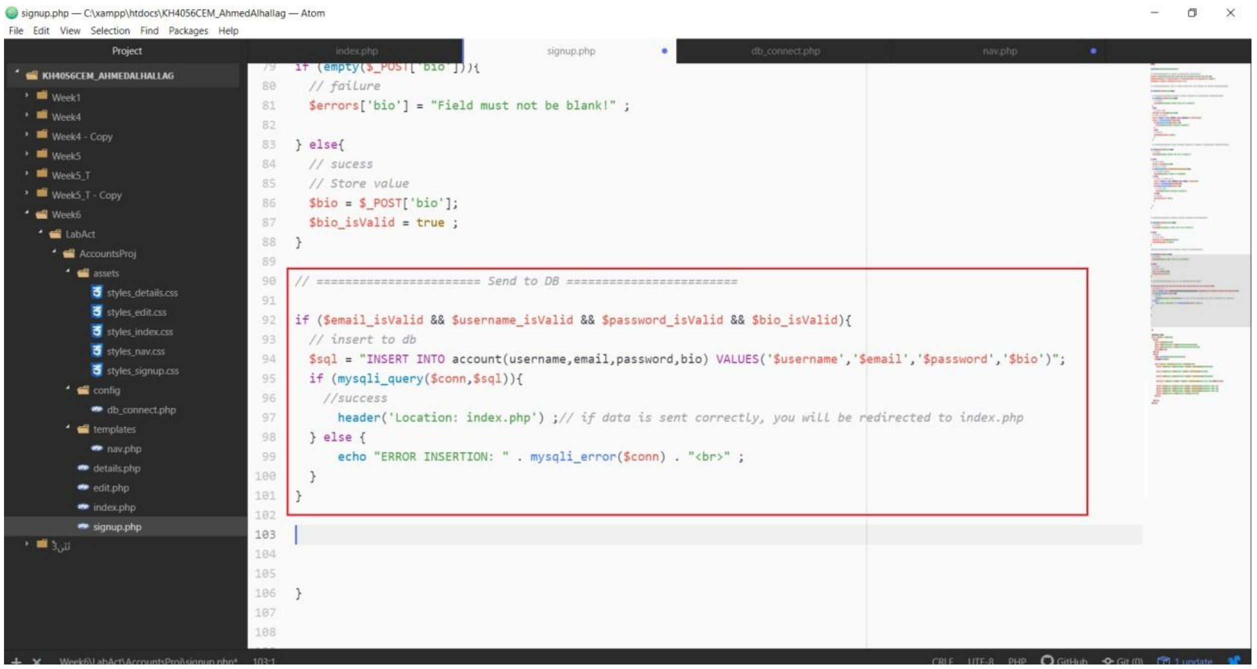
```
    }

  }
}


// =================== password: validate non-empty ================

if (empty($_POST['password'])){
  // failure
  $errors['password'] = "Field must not be blank!" ;

} else{
  // success
  // Store value
  $password = $_POST['password'] ;
  $password_isValid = true ;
}




}






 ?>
```

Same for the bio:



```
// ==================== password: validate non-empty =================

if (empty($_POST['password'])){
  // failure
  $errors['password'] = "Field must not be blank!" ;

} else{
  // success
  // Store value
  $password = $_POST['password'] ;
  $password_isValid = true ;
}

//==================== bio: validate non-empty =================

if (empty($_POST['bio'])){
  // failure
  $errors['bio'] = "Field must not be blank!" ;

} else{
  // sucess
  // Store value
  $bio = $_POST['bio'];
  $bio_isValid = true ;
}



}
```

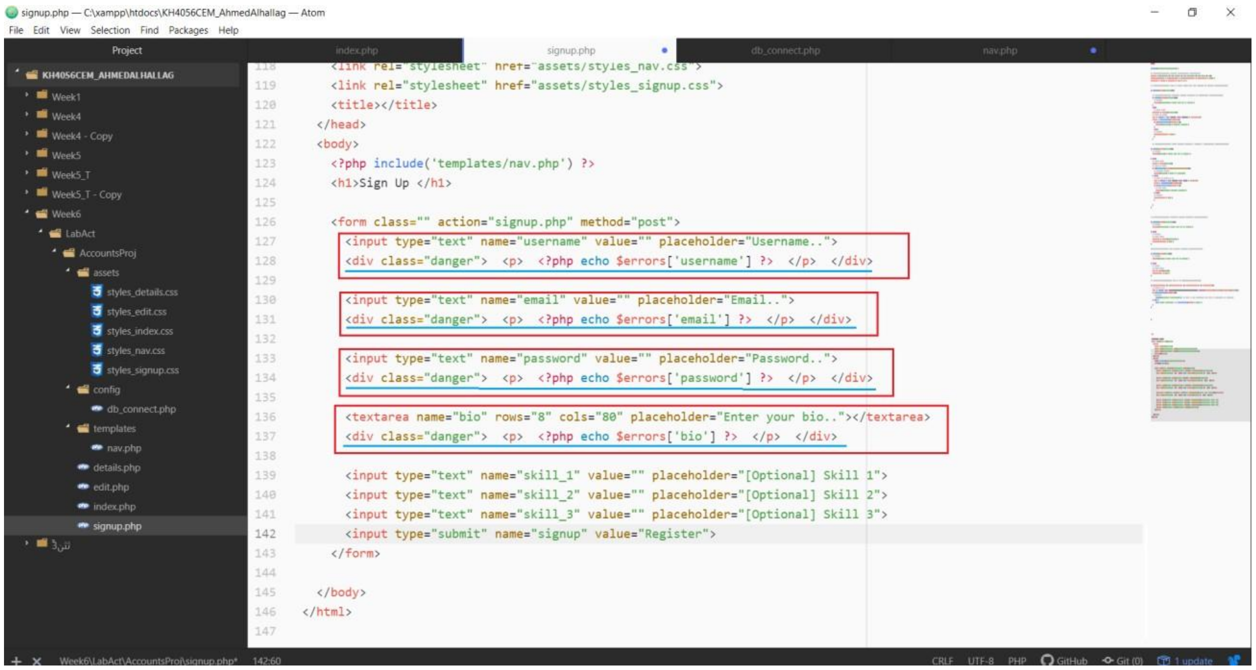
Now send to database:



```
if (empty($_POST['bio'])){
  // failure
  $errors['bio'] = "Field must not be blank!" ;

} else{
  // sucess
  // Store value
  $bio = $_POST['bio'];
  $bio_isValid = true ;
}

// ======================= Send to DB =======================

if ($email_isValid && $username_isValid && $password_isValid && $bio_isValid){
  // insert to db
  $sql = "INSERT INTO account(username,email,password,bio) VALUES('$username','$email','$password','$bio')";
  if (mysqli_query($conn,$sql)){
    //success
      header('Location: index.php') ;// if data is sent correctly, you will be redirected to index.php
  } else {
      echo "ERROR INSERTION: " . mysqli_error($conn) . "<br>" ;
  }
}



}
```

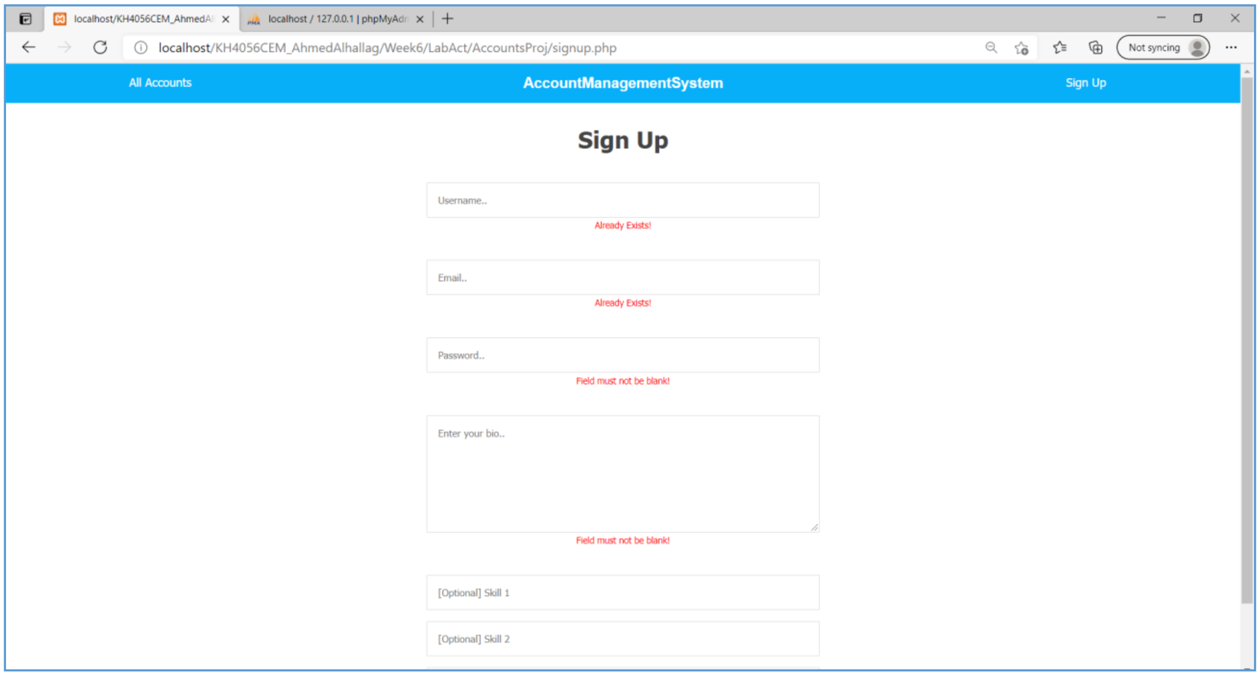
To see the error messages we created for each edge case, we need to display it in the HTML document.

Under each input field from the 4 that cater for, create a div with a class of ‘danger’, then inside of it create a <p> tag that holds the $error key-value pair for each input



```
    <link rel="stylesheet" href="assets/styles_nav.css">
    <link rel="stylesheet" href="assets/styles_signup.css">
    <title></title>
  </head>
  <body>
    <?php include('templates/nav.php') ?>
    <h1>Sign Up </h1>

    <form class="" action="signup.php" method="post">
      <input type="text" name="username" value="" placeholder="Username..">
      <div class="danger">  <p>  <?php echo $errors['username'] ?>  </p>  </div>

      <input type="text" name="email" value="" placeholder="Email..">
      <div class="danger">  <p>  <?php echo $errors['email'] ?>  </p>  </div>

      <input type="text" name="password" value="" placeholder="Password..">
      <div class="danger">  <p>  <?php echo $errors['password'] ?>  </p>  </div>

      <textarea name="bio" rows="8" cols="80" placeholder="Enter your bio.."></textarea>
      <div class="danger">  <p>  <?php echo $errors['bio'] ?>  </p>  </div>

      <input type="text" name="skill_1" value="" placeholder="[Optional] Skill 1">
      <input type="text" name="skill_2" value="" placeholder="[Optional] Skill 2">
      <input type="text" name="skill_3" value="" placeholder="[Optional] Skill 3">
      <input type="submit" name="signup" value="Register">
    </form>

  </body>
</html>
```

Check the validations you made on the sign up page and see if it works or not.


localhost/KH4056CEM_AhmedAl
localhost / 127.0.0.1 | phpMyAdr
localhost/KH4056CEM_AhmedAlhallag/Week6/LabAct/AccountsProj/signup.php
Not syncing
All Accounts
AccountManagementSystem
Sign Up

Sign Up

Username..

Already Exists!

Email..

Already Exists!

Password..

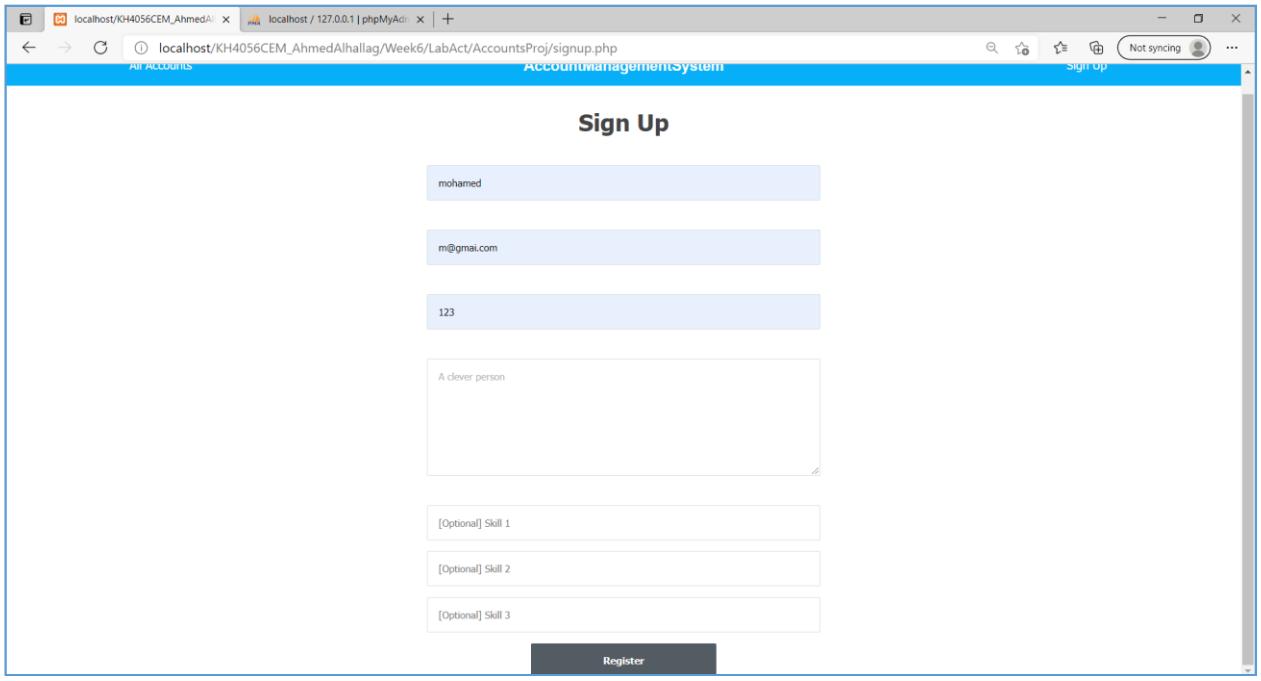
Field must not be blank!

Enter your bio..

Field must not be blank!

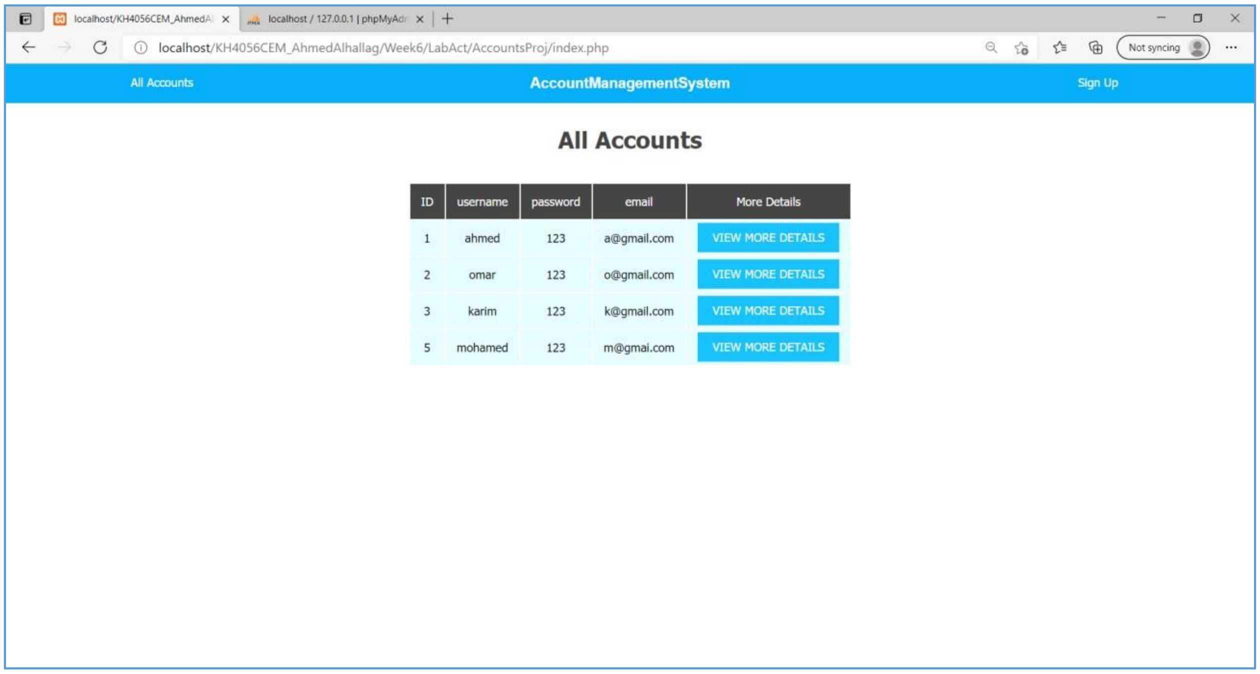
[Optional] Skill 1

[Optional] Skill 2


localhost/KH4056CEM_AhmedAlhallag/Week6/LabAct/AccountsProj/signup.php

Sign Up

mohamed
m@gmai.com
123
A clever person
[Optional] Skill 1
[Optional] Skill 2
[Optional] Skill 3

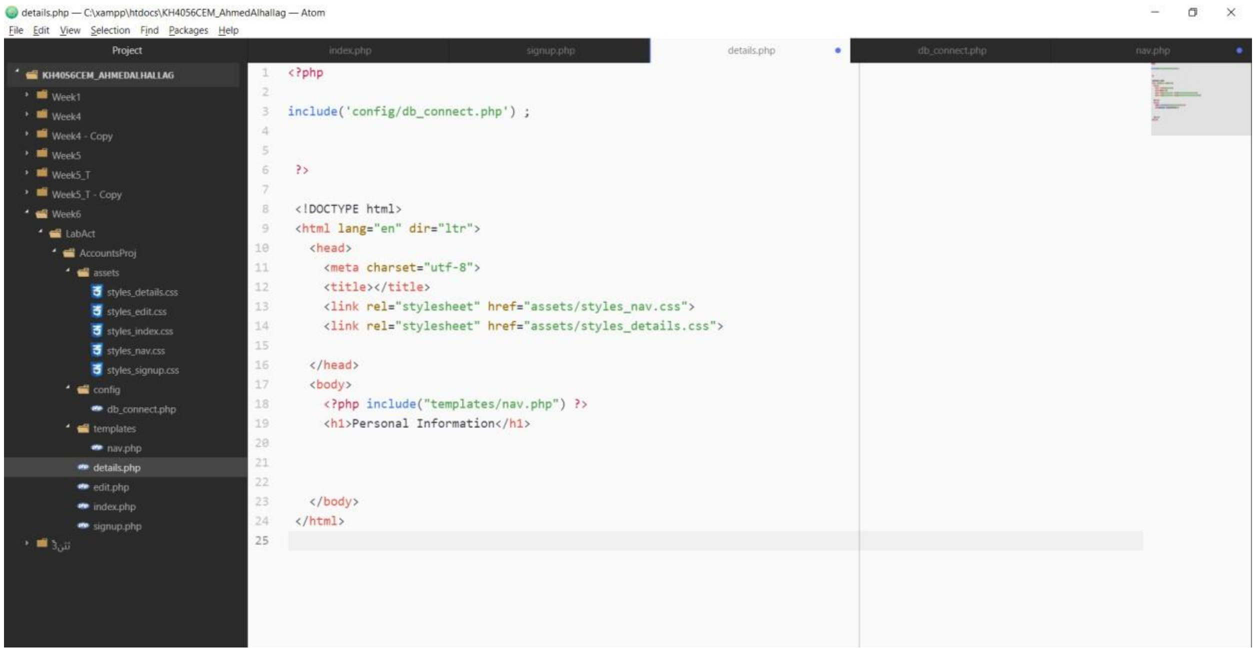

localhost/KH4056CEM_AhmedAlhallag/Week6/LabAct/AccountsProj/index.php

All Accounts

ID	username	password	email	More Details
1	ahmed	123	a@gmail.com	VIEW MORE DETAILS
2	omar	123	o@gmail.com	VIEW MORE DETAILS
3	karim	123	k@gmail.com	VIEW MORE DETAILS
5	mohamed	123	m@gmai.com	VIEW MORE DETAILS

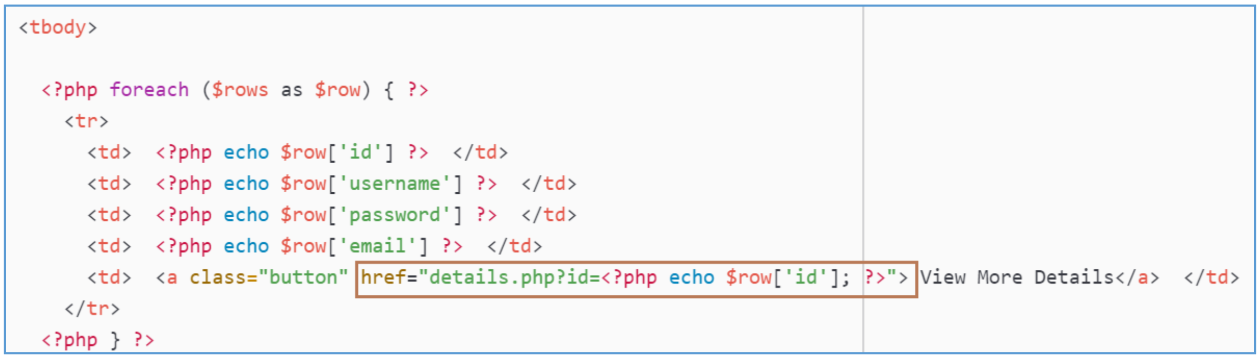
Now create the details.php page under your project directory so we can see each indivudual's personal data.

Set it up as follows:



```
<?php

include('config/db_connect.php') ;


 ?>

 <!DOCTYPE html>
 <html lang="en" dir="ltr">
   <head>
     <meta charset="utf-8">
     <title></title>
     <link rel="stylesheet" href="assets/styles_nav.css">
     <link rel="stylesheet" href="assets/styles_details.css">

   </head>
   <body>
     <?php include("templates/nav.php") ?>
     <h1>Personal Information</h1>



   </body>
 </html>

```

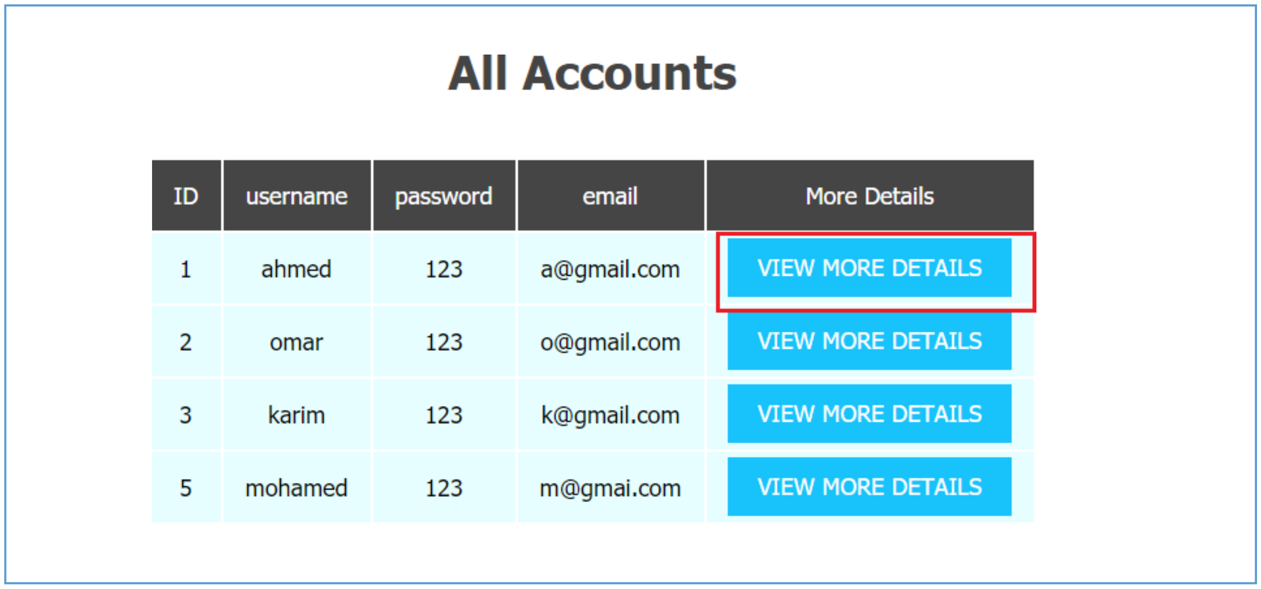
So remember from the index.php we had each anchor tag's href having a format of:



```
<tbody>

  <?php foreach ($rows as $row) { ?>
    <tr>
      <td>  <?php echo $row['id'] ?>  </td>
      <td>  <?php echo $row['username'] ?>  </td>
      <td>  <?php echo $row['password'] ?>  </td>
      <td>  <?php echo $row['email'] ?>  </td>
      <td>  <a class="button" href="details.php?id=<?php echo $row['id']; ?>">View More Details</a>  </td>
    </tr>
  <?php } ?>
```

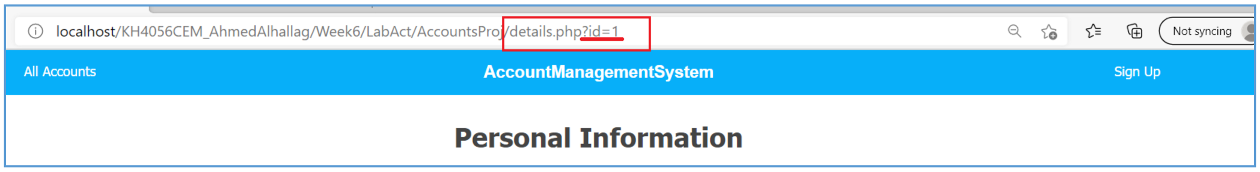
This means that the query string, ?id=<value>, will be sent to the detail.php page every time we visit it by clicking on that link.

Go to the home page and try clicking on the link now.


All Accounts

ID	username	password	email	More Details
1	ahmed	123	a@gmail.com	VIEW MORE DETAILS
2	omar	123	o@gmail.com	VIEW MORE DETAILS
3	karim	123	k@gmail.com	VIEW MORE DETAILS
5	mohamed	123	m@gmai.com	VIEW MORE DETAILS

Check the URL:


localhost/KH4056CEM_AhmedAlhallag/Week6/LabAct/AccountsProj/details.php?id=1

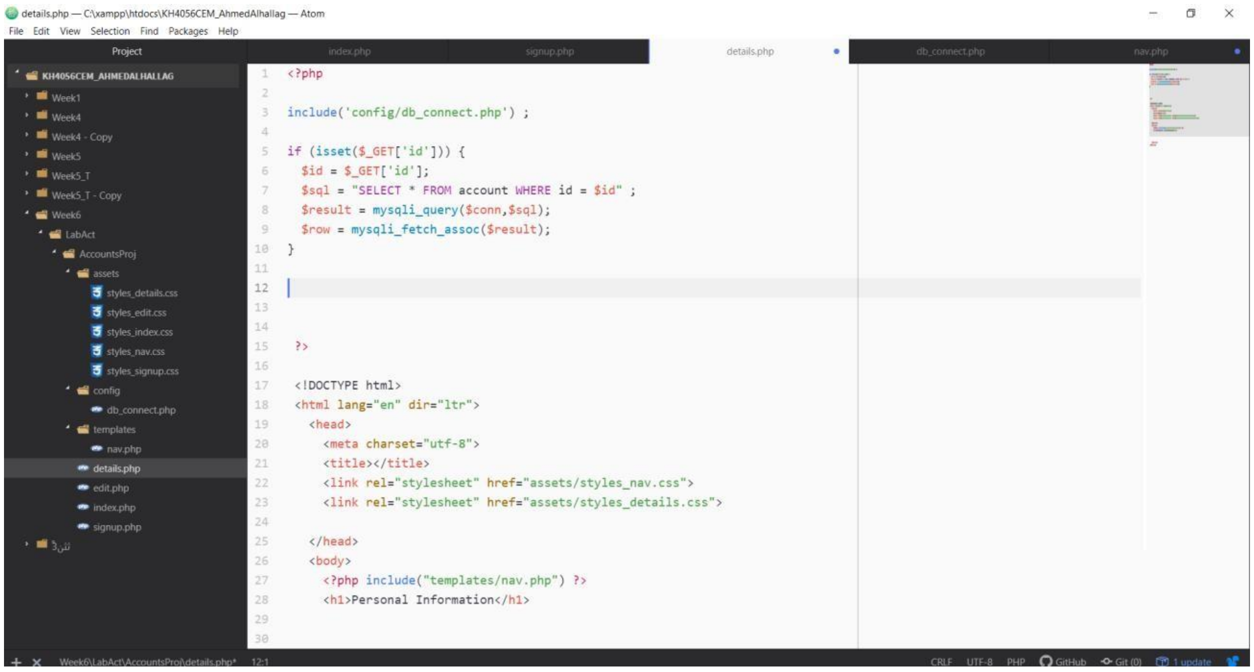
Personal Information

If we can grab the id value from the url to our php portion in the details.php page, then we can easily query the db for the rest of information using the id.

The question is, how to fetch that id form the query string?

Using the $_GET['id'] array (entering and going to URLs or links are just GET requests by the end of the day!)

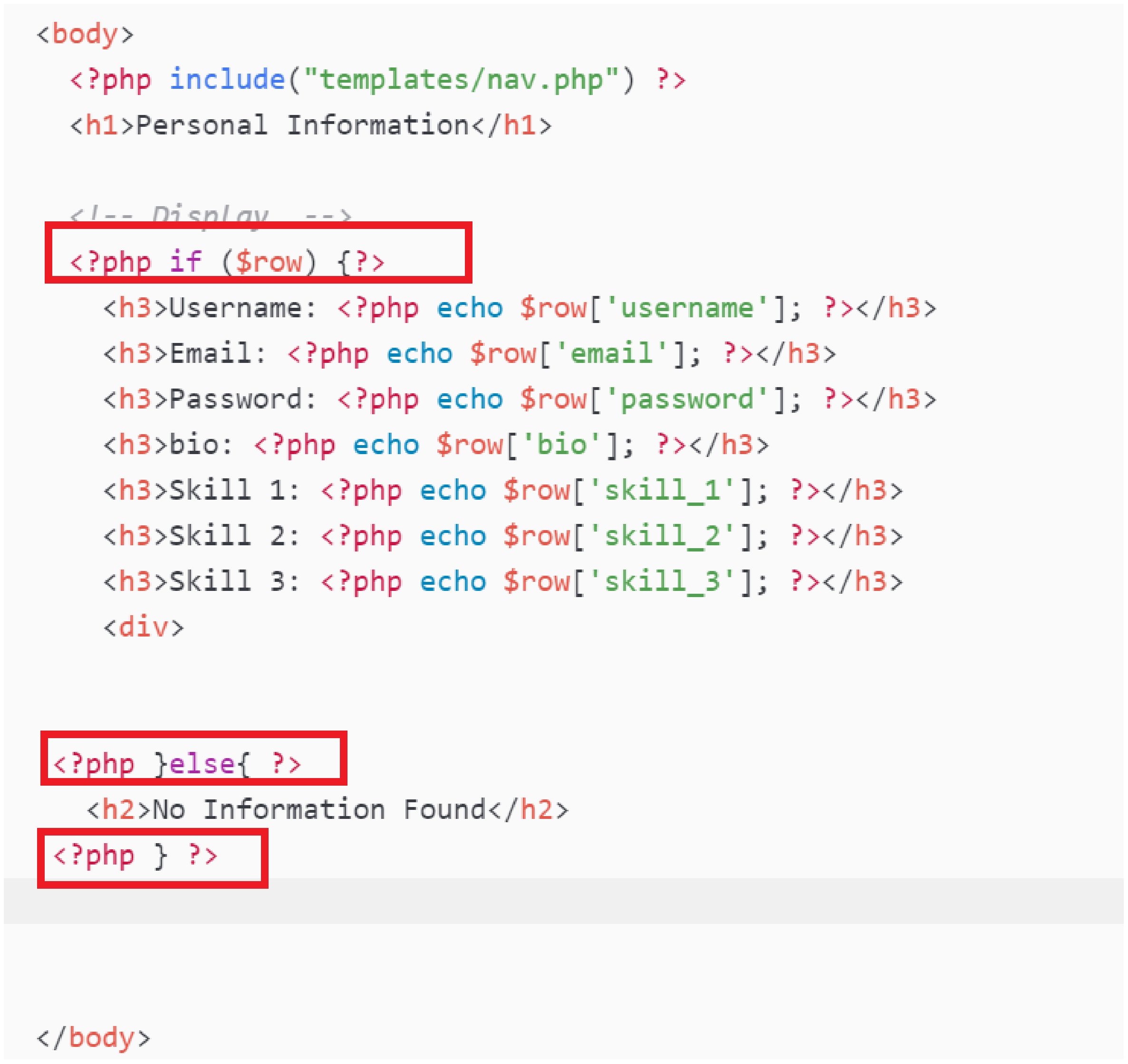
Note: the name you gave the query string parameter, id in this case, from the index page, should be the index used for the $_GET associative array.



```php
<?php

include('config/db_connect.php') ;

if (isset($_GET['id'])) {
  $id = $_GET['id'];
  $sql = "SELECT * FROM account WHERE id = $id" ;
  $result = mysqli_query($conn,$sql);
  $row = mysqli_fetch_assoc($result);
}




 ?>

 <!DOCTYPE html>
 <html lang="en" dir="ltr">
   <head>
     <meta charset="utf-8">
     <title></title>
     <link rel="stylesheet" href="assets/styles_nav.css">
     <link rel="stylesheet" href="assets/styles_details.css">

   </head>
   <body>
     <?php include("templates/nav.php") ?>
     <h1>Personal Information</h1>
```

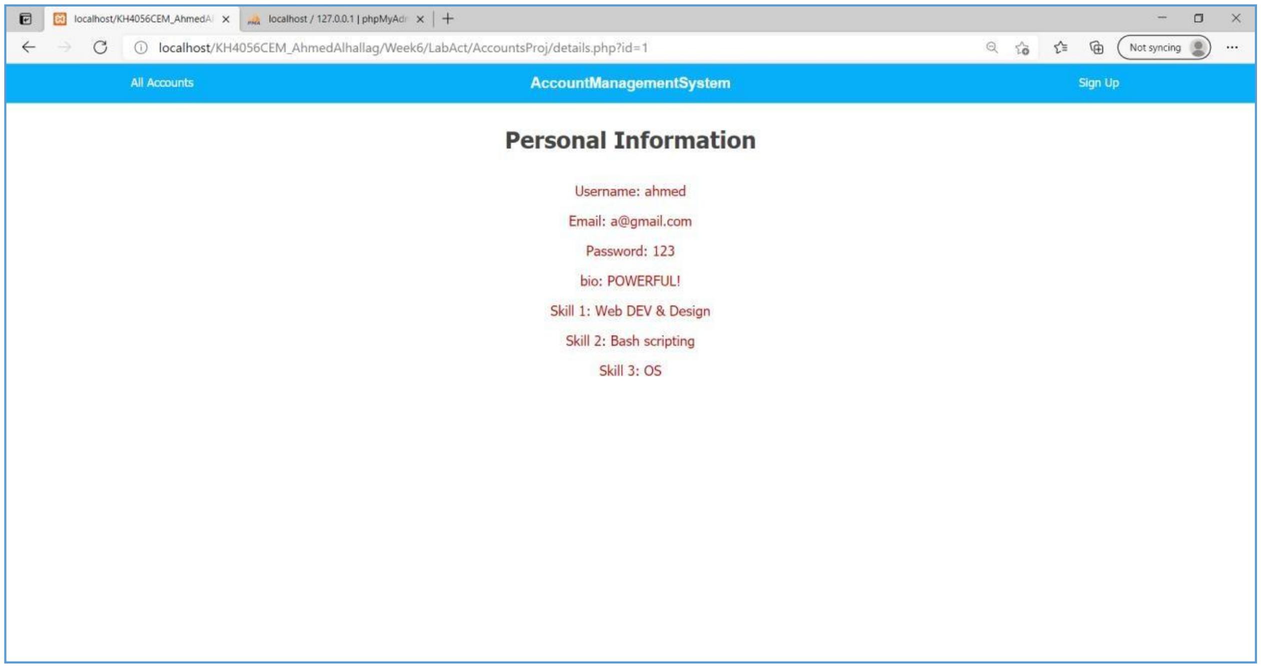
Render the values in h3 tags.

Notice the conditional, this is to make sure that the returned row actually in the database and actually contains something.



```
<body>
  <?php include("templates/nav.php") ?>
  <h1>Personal Information</h1>

  <!-- Display  -->
  <?php if ($row) {?>
    <h3>Username: <?php echo $row['username']; ?></h3>
    <h3>Email: <?php echo $row['email']; ?></h3>
    <h3>Password: <?php echo $row['password']; ?></h3>
    <h3>bio: <?php echo $row['bio']; ?></h3>
    <h3>Skill 1: <?php echo $row['skill_1']; ?></h3>
    <h3>Skill 2: <?php echo $row['skill_2']; ?></h3>
    <h3>Skill 3: <?php echo $row['skill_3']; ?></h3>
    <div>


<?php }else{ ?>
   <h2>No Information Found</h2>
<?php } ?>




</body>
```

Refresh the page:


localhost/KH4056CEM_AhmedAl
localhost / 127.0.0.1 | phpMyAdr
localhost/KH4056CEM_AhmedAlhallag/Week6/LabAct/AccountsProj/details.php?id=1
Not syncing
All Accounts
AccountManagementSystem
Sign Up

Personal Information

Username: ahmed

Email: a@gmail.com

Password: 123

bio: POWERFUL!

Skill 1: Web DEV & Design

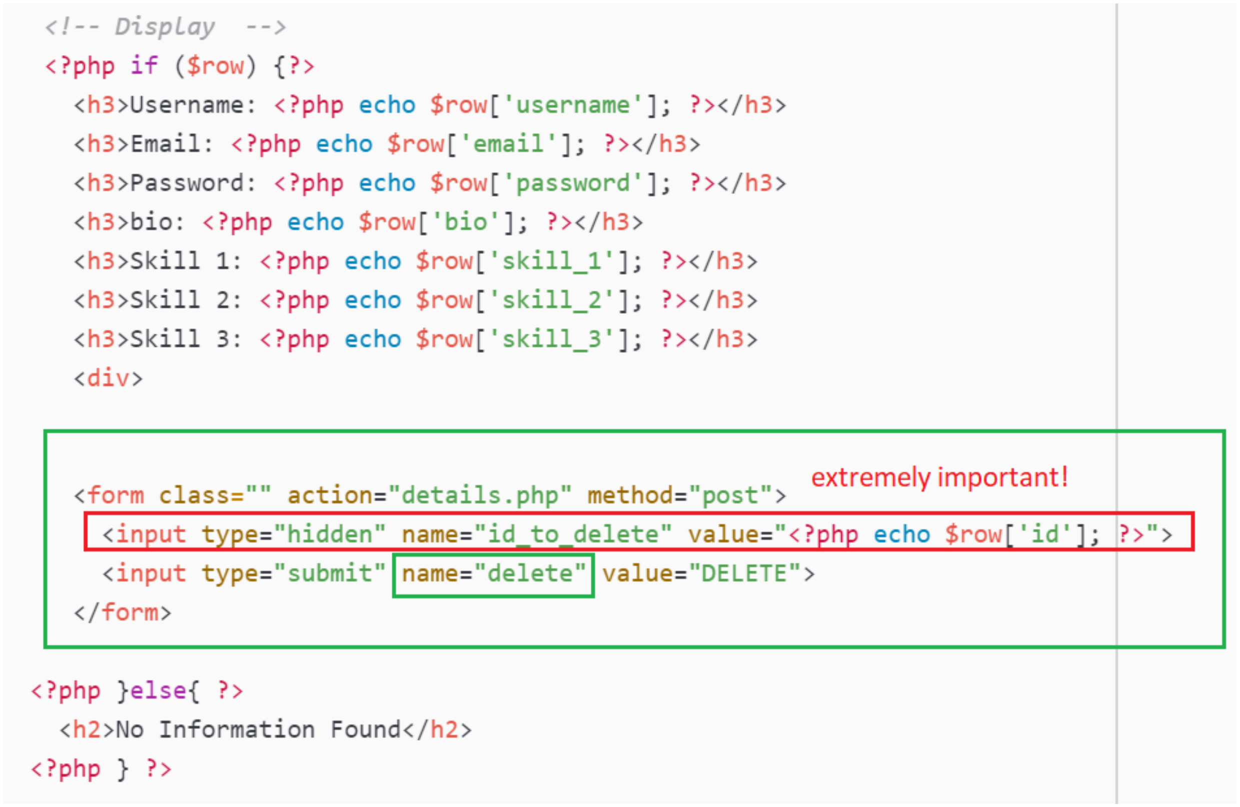
Skill 2: Bash scripting

Skill 3: OS

<u>Perform the Delete operation from CRUD:</u>

Create a form specifically for deleting as follows:

General Rule: the action should always be the page your at.



```
  <!-- Display  -->
  <?php if ($row) {?>
    <h3>Username: <?php echo $row['username']; ?></h3>
    <h3>Email: <?php echo $row['email']; ?></h3>
    <h3>Password: <?php echo $row['password']; ?></h3>
    <h3>bio: <?php echo $row['bio']; ?></h3>
    <h3>Skill 1: <?php echo $row['skill_1']; ?></h3>
    <h3>Skill 2: <?php echo $row['skill_2']; ?></h3>
    <h3>Skill 3: <?php echo $row['skill_3']; ?></h3>
    <div>


    <form class="" action="details.php" method="post">
      <input type="hidden" name="id_to_delete" value="<?php echo $row['id']; ?>">
      <input type="submit" name="delete" value="DELETE">
    </form>

 <?php }else{ ?>
   <h2>No Information Found</h2>
 <?php } ?>
```

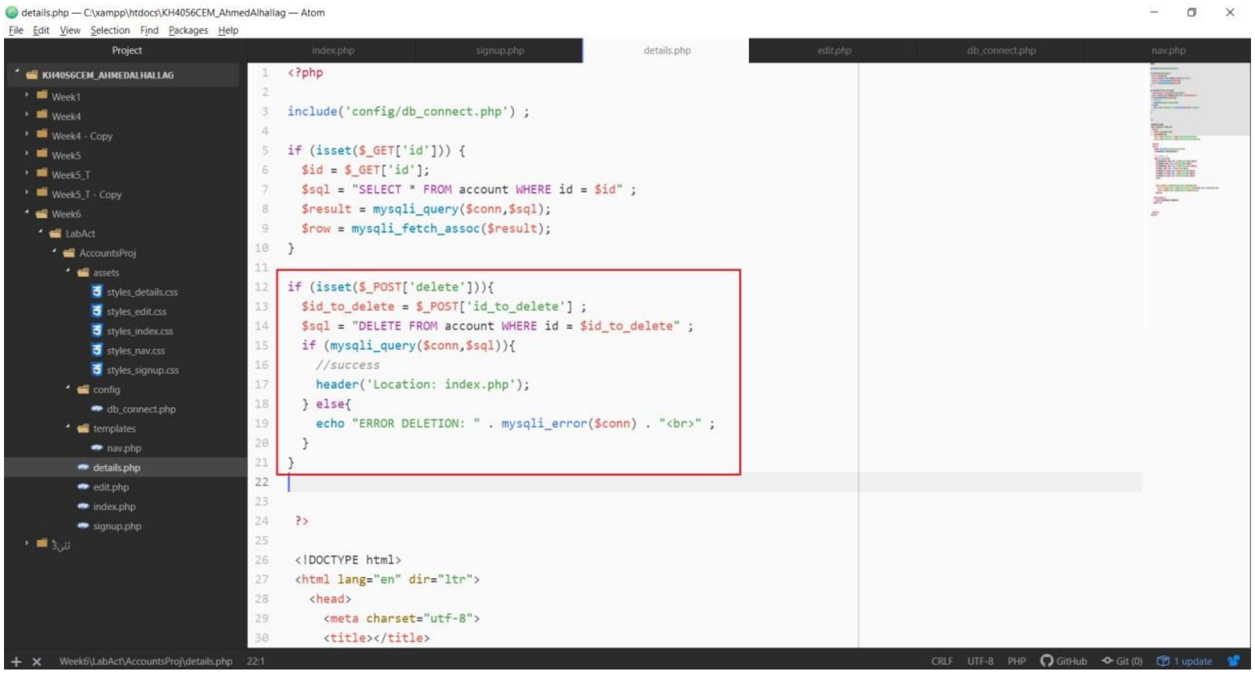
Since the details.php is not really good on it's own without a query string, it makes more sense to add the id parameter of the query string after the details.php right?

Answer is yes BUT when the purpose is to delete, then there is no value of adding the id to the details page, since the whole route of, for example ?id=2, would be eventually delete; so you don't really need to be specific in this case.

IMPORTANT:

- The id fetched from the get request and saved into the variable $id will be used to populate the hidden input field value from the above image, because once you trigger the delete form by clicking the type submit button, that id will be needed again to be used to figure out which row is that row from the db.
- The id fetched from the get request and saved into the variable $id, will eventually die due to scope. That's why we can no longer use it.
- So as we said, we populate it into the field, then we go back and fetch it once the post request is fired due to clicking the submit button named 'delete'

Now add the logic for sending the delete query on submission.



```php
<?php

include('config/db_connect.php') ;

if (isset($_GET['id'])) {
  $id = $_GET['id'];
  $sql = "SELECT * FROM account WHERE id = $id" ;
  $result = mysqli_query($conn,$sql);
  $row = mysqli_fetch_assoc($result);
}

if (isset($_POST['delete'])){
  $id_to_delete = $_POST['id_to_delete'] ;
  $sql = "DELETE FROM account WHERE id = $id_to_delete" ;
  if (mysqli_query($conn,$sql)){
    //success
    header('Location: index.php');
  } else{
    echo "ERROR DELETION: " . mysqli_error($conn) . "<br>" ;
  }
}


 ?>

 <!DOCTYPE html>
 <html lang="en" dir="ltr">
   <head>
     <meta charset="utf-8">
     <title></title>
```

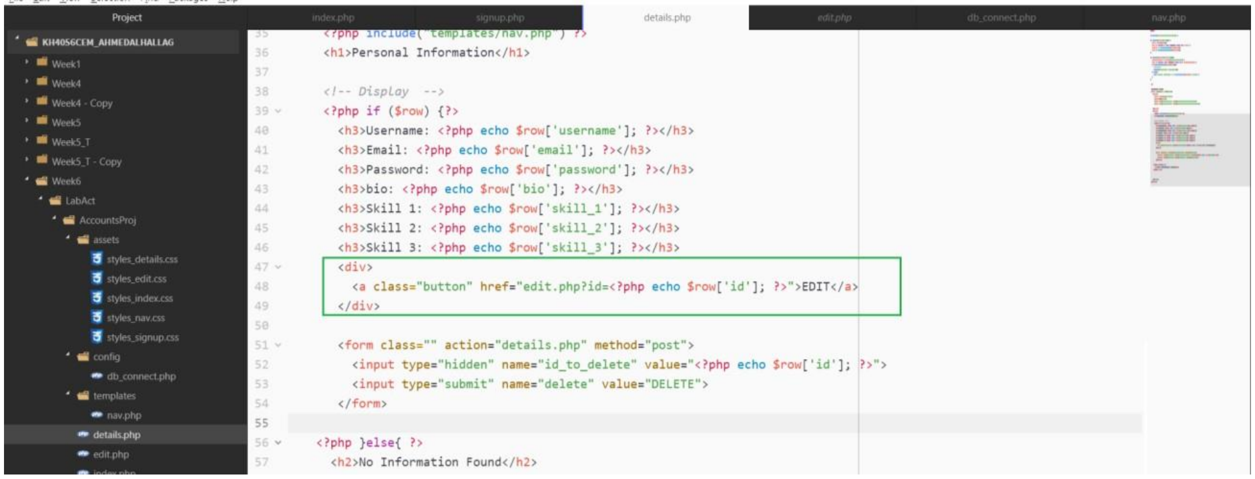
Go to the page and try to delete the last record. It should work.

Activity:

Add an “Edit” button that, using the same fashion, should to take you to an ’edit.php’ page with the current accountid you are using in the for the details.php?id=2.

So the href in that edit button should be as follows:

Hint: put it inside a div.



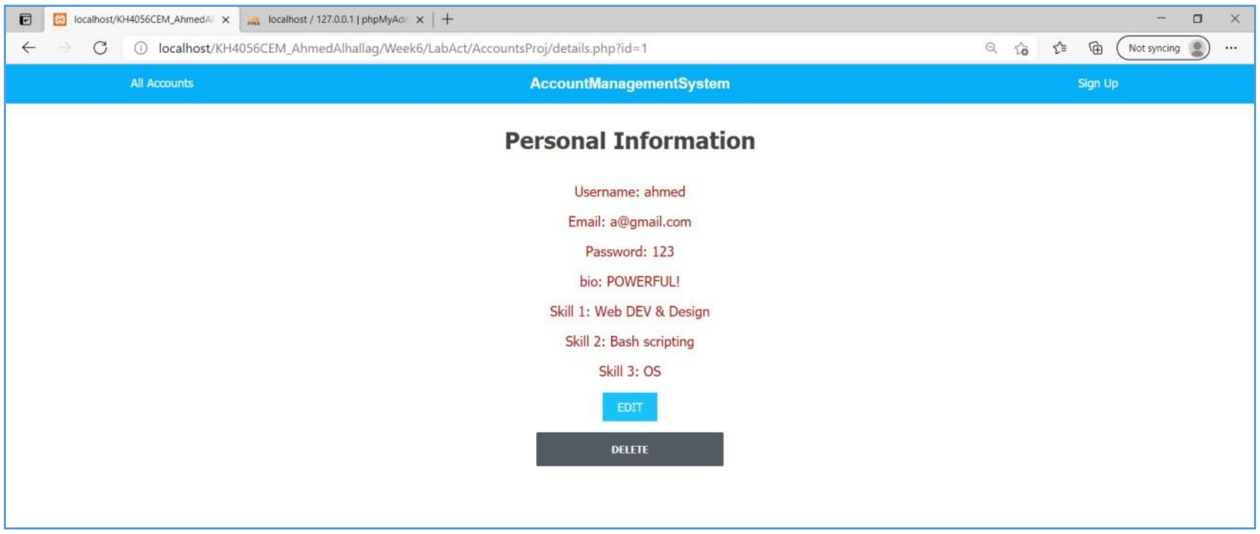
```
<?php include('templates/nav.php') ?>
<h1>Personal Information</h1>

<!-- Display  -->
<?php if ($row) {?>
  <h3>Username: <?php echo $row['username']; ?></h3>
  <h3>Email: <?php echo $row['email']; ?></h3>
  <h3>Password: <?php echo $row['password']; ?></h3>
  <h3>bio: <?php echo $row['bio']; ?></h3>
  <h3>Skill 1: <?php echo $row['skill_1']; ?></h3>
  <h3>Skill 2: <?php echo $row['skill_2']; ?></h3>
  <h3>Skill 3: <?php echo $row['skill_3']; ?></h3>
  <div>
    <a class="button" href="edit.php?id=<?php echo $row['id']; ?>">EDIT</a>
  </div>

  <form class="" action="details.php" method="post">
    <input type="hidden" name="id_to_delete" value="<?php echo $row['id']; ?>">
    <input type="submit" name="delete" value="DELETE">
  </form>

<?php }else{ ?>
 <h2>No Information Found</h2>
```

Create the edit.php page accordingly.

- **The edit page should display username, email, password as disabled inputfields.**
- **User should be able to update ANY of the following fields: bio, skill_1, skill_2, skill_3 and hit on update.**
- **The edit page should have a cancel button that take you back to index.php when clicked.**

The pages should look likes this:

- **details.php (notice the edit button)**



localhost/KH4056CEM_AhmedAl
localhost / 127.0.0.1 | phpMyAdm
localhost/KH4056CEM_AhmedAlhallag/Week6/LabAct/AccountsProj/details.php?id=1
Not syncing
All Accounts
AccountManagementSystem
Sign Up
EDIT
DELETE

Personal Information

Username: ahmed

Email: a@gmail.com

Password: 123

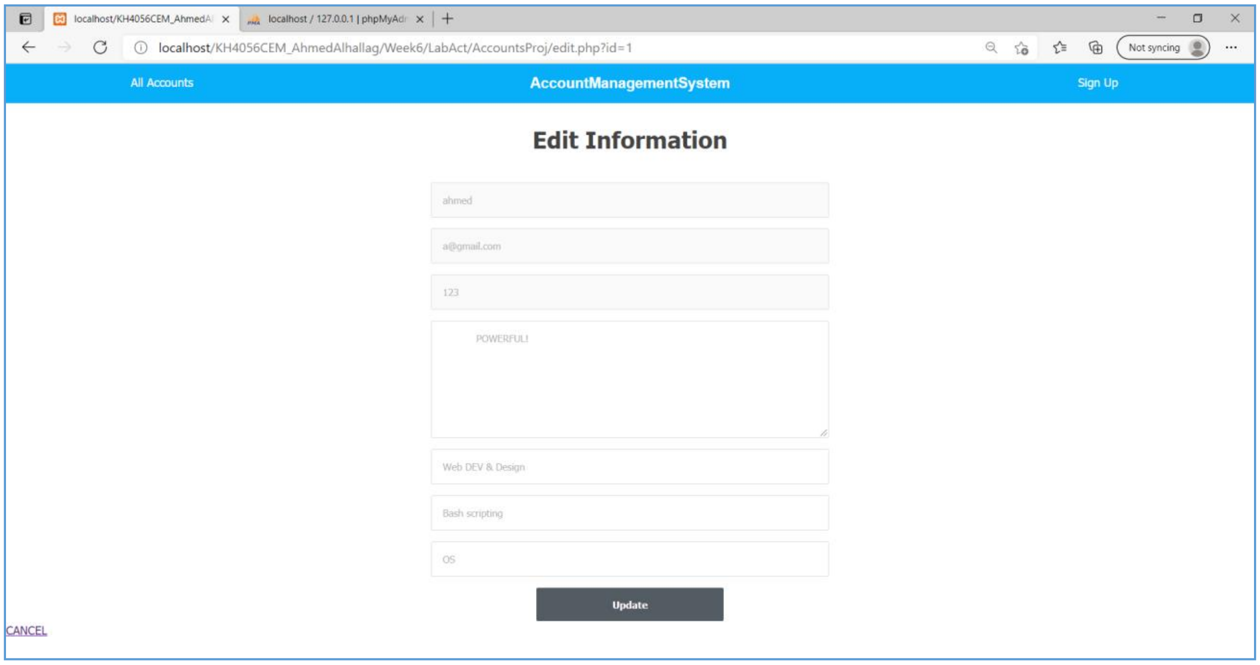
bio: POWERFUL!

Skill 1: Web DEV & Design

Skill 2: Bash scripting

Skill 3: OS

- **edit.php**


localhost/KH4056CEM_AhmedAlhallag/Week6/LabAct/AccountsProj/edit.php?id=1

All Accounts AccountManagementSystem Sign Up

Edit Information

ahmed

a@gmail.com

123

POWERFUL!

Web DEV & Design

Bash scripting

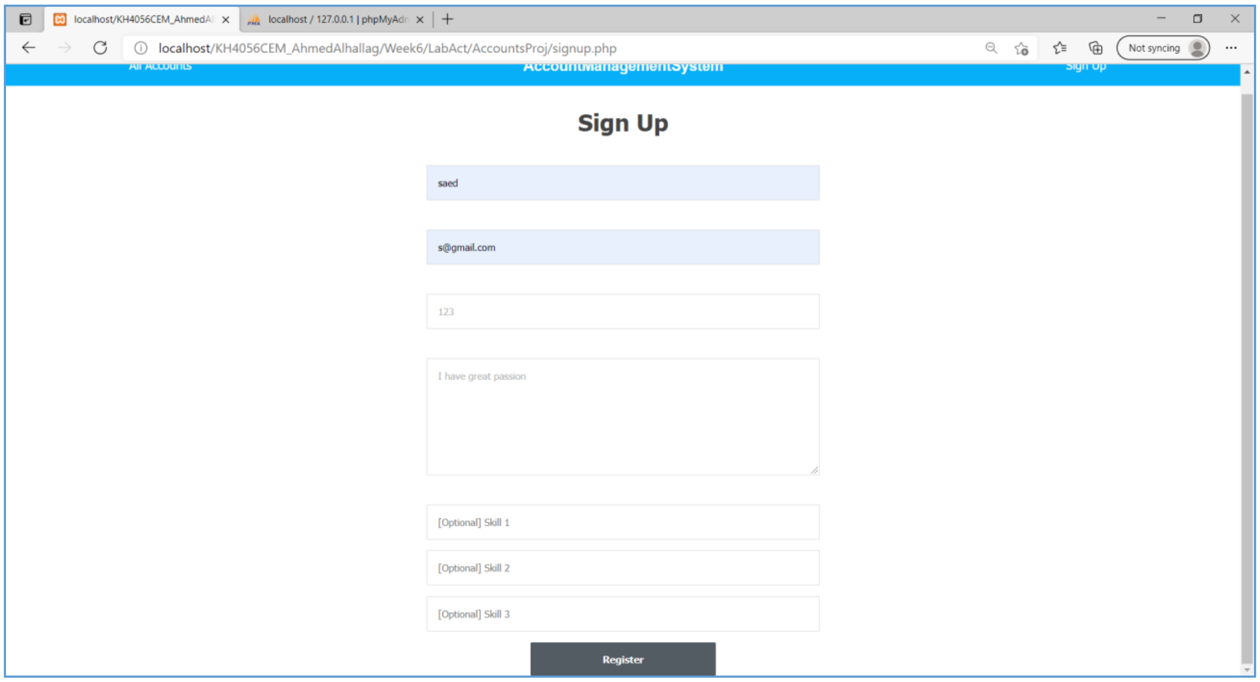
OS

Update

CANCEL

Testing:

Create a new account and leave the skills empty.


localhost/KH4056CEM_AhmedAlhallag/Week6/LabAct/AccountsProj/signup.php

Sign Up

saed

s@gmail.com

123

I have great passion

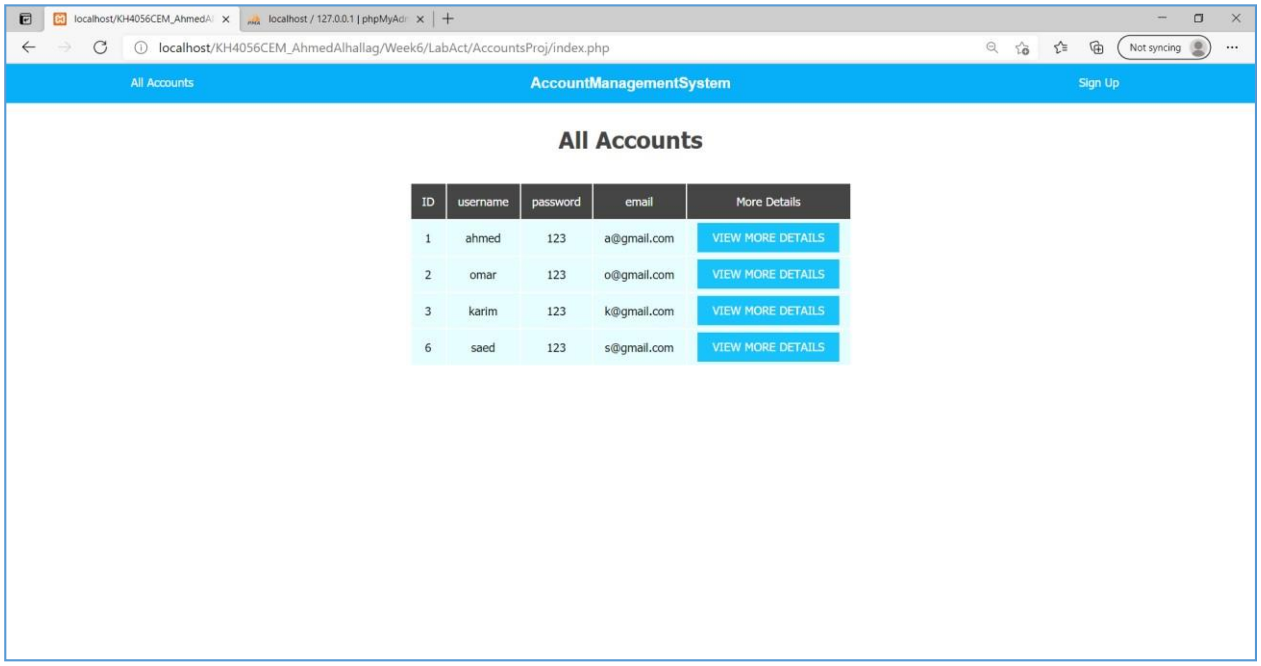
[Optional] Skill 1

[Optional] Skill 2

[Optional] Skill 3

Register

You should see the entry

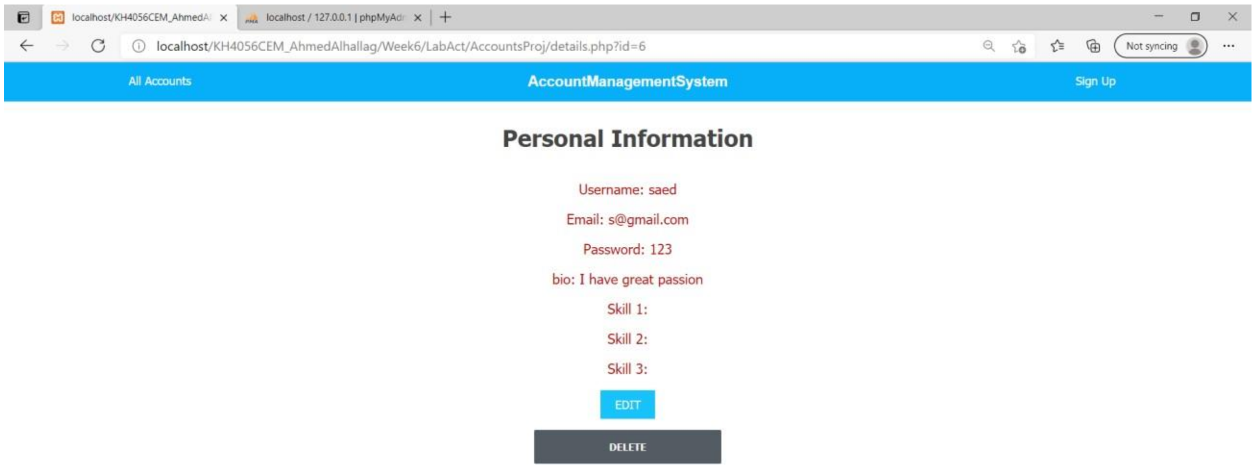

localhost/KH4056CEM_AhmedAlhallag/Week6/LabAct/AccountsProj/index.php

All Accounts | AccountManagementSystem | Sign Up

All Accounts

ID	username	password	email	More Details
1	ahmed	123	a@gmail.com	VIEW MORE DETAILS
2	omar	123	o@gmail.com	VIEW MORE DETAILS
3	karim	123	k@gmail.com	VIEW MORE DETAILS
6	saed	123	s@gmail.com	VIEW MORE DETAILS

Go to his page details.


localhost/KH4056CEM_AhmedAlhallag/Week6/LabAct/AccountsProj/details.php?id=6

All Accounts | AccountManagementSystem | Sign Up

Personal Information

Username: saed

Email: s@gmail.com

Password: 123

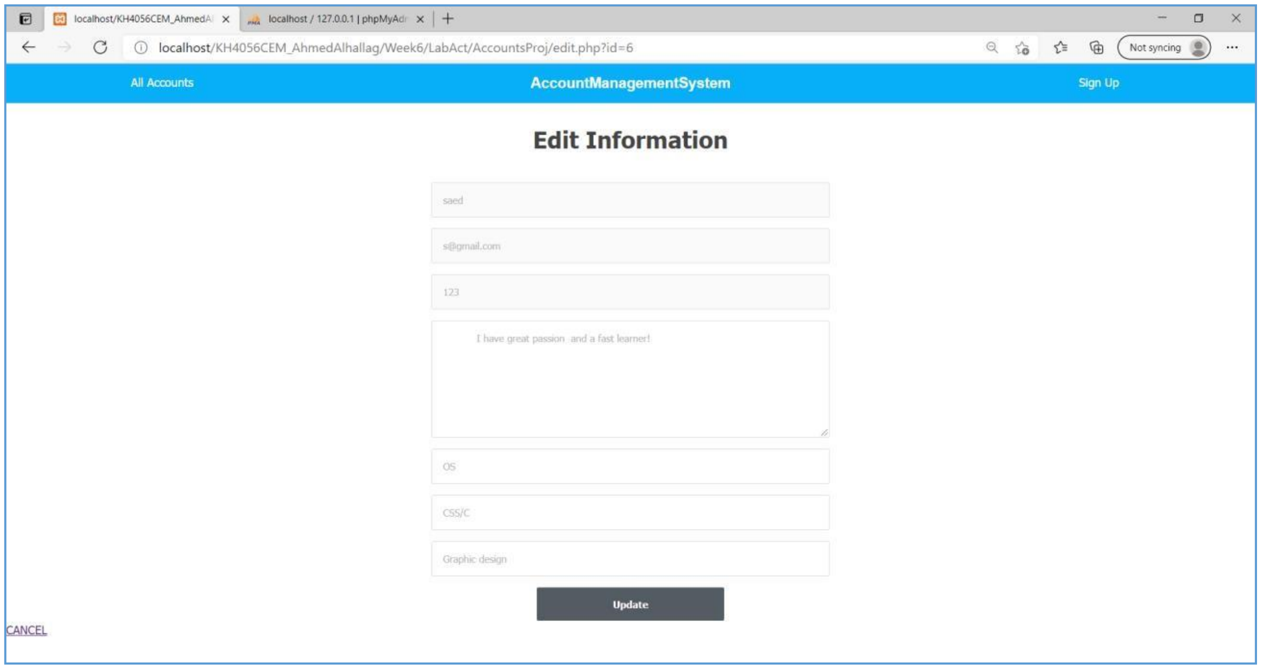
bio: I have great passion

Skill 1:

Skill 2:

Skill 3:

Click on Edit, enter any new values and hot update:



localhost/KH4056CEM_AhmedAlhallag/Week6/LabAct/AccountsProj/edit.php?id=6

Edit Information

saed

s@gmail.com

123

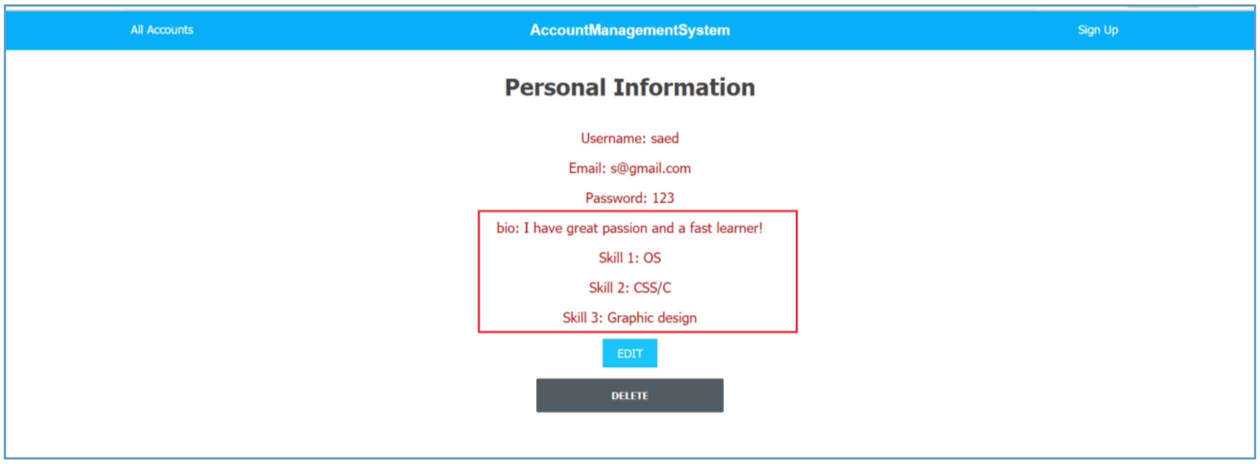
I have great passion and a fast learner!

OS

CSS/C

Graphic design

You should see:



Personal Information

Username: saed

Email: s@gmail.com

Password: 123

bio: I have great passion and a fast learner!

Skill 1: OS

Skill 2: CSS/C

Skill 3: Graphic design