Web Application Programming

Week 7: Building Dynamic Web Applications(Server-side scripting for generating dynamic web content)

By Elubu Joseph - MSc.IS

Lecturer

Department of Information Technology

Kumi University

Email: josebulinda@gmail.com or jose@kumiuniversity.ac.ug

Agenda

1. Summary of previous lecture
2. Building Dynamic Web Applications(
    i. Introduction
    ii. Creating a Database
    iii. Establish a Database Connection
    iv. Performing CRUD Operations

Summary of previous lecture

1. Introduction to databases(Definition, E.gs of DBMS, Advantages and Disadvantages)
2. Introduction to SQL(Definition, E.gs of sql commands used to create database and minuplate its data)

Intro to Building Dynamic Web Applications

Dynamic web content refers to web pages or elements of web pages that can change and update in real-time based on various factors, such as user interactions, data from external sources, or other dynamic processes.

Unlike static web content, which remains the same until it is manually updated, dynamic web content adapts and refreshes automatically without requiring constant manual intervention.

Dynamic content is a fundamental aspect of modern websites and web applications, and it enables more engaging and interactive user experiences.

Intro to Building Dynamic Web Applications+

Here are some key characteristics and examples of dynamic web content:

1. **Real-Time Updates:** Dynamic web content can change without requiring the user to refresh the page. For example, a news website can display the latest headlines as they become available, and a social media feed can show new posts and updates in real time.
2. **User Interaction:** User interactions, such as submitting a form, clicking a button, or making selections in a dropdown menu, can trigger changes in the content. For instance, when a user submits a search query, the search results page dynamically updates to display relevant results.
3. **Database Integration:** Dynamic content often involves interactions with databases. Web applications can retrieve and display data from a database, allowing users to access and manipulate information. E-commerce sites, for example, use dynamic content to display product listings and manage shopping carts. OpenAI. (2021).

Intro to Building Dynamic Web Applications++

4. **Personalization:** Dynamic content can be tailored to individual users or user groups. This personalization can include recommendations, user profiles, and custom content based on user preferences and behavior.
5. **Automation:** Dynamic web content can be automated to perform various tasks. For instance, an online booking system can check the availability of hotel rooms in real time and allow users to book rooms without manual intervention.
6. **Integration with APIs:** Web applications can communicate with external services and APIs (Application Programming Interfaces) to fetch and display dynamic data. For example, a weather website can display real-time weather conditions by integrating with a weather data API.

Intro to Building Dynamic Web Applications+++

7. **Content Management Systems (CMS):** CMS platforms, like WordPress, Drupal, and Joomla, allow users to manage and update dynamic content on their websites without extensive coding knowledge. These systems enable users to create, edit, and publish content easily.
8. **Web Frameworks:** Many web development frameworks, such as Ruby on Rails, Django, and Node.js, provide tools and libraries for building dynamic web applications. These frameworks simplify the process of creating and managing dynamic content.

Overall, dynamic web content enhances the user experience by providing up-to-date information, interactive features, and personalized content. OpenAI. (2021).

Intro to Building Dynamic Web Applications+++++

PHP is a common choice for generating dynamic web content and interacting with databases.

PHP is a versatile scripting language that allows you to create dynamic web pages and perform CRUD (Create, Read, Update, Delete) operations on databases.

Here's a high-level overview of how you can achieve this:

Introduction to Building Dynamic Web Applications - Steps

Here's a high-level overview of how you can achieve this:

1.Setting Up Your Environment:

1. Install a web server (e.g., Apache, Nginx) and a database server (e.g., MySQL, PostgreSQL) on your server or development machine.
2. Install PHP on your server or development machine. You can use packages like XAMPP or WAMP for local development or configure PHP manually.

2.Creating a Database:

1. Create a database to store your application data using a database management system (DBMS). You can use tools like phpMyAdmin, MySQL Workbench, or the command line to create and manage databases.

Introduction to Building Dynamic Web Applications-Steps+

Here's a high-level overview of how you can achieve this:

3. Establish a Database Connection:

In your PHP code, establish a connection to the database using the appropriate functions. For example, if you are using MySQL, you can use the mysqli to connect to the database.

```php
$server = "localhost";
$username = "wap";
$password = "wap";
$database = "wap";

$conn = new mysqli($server, $username, $password, $database);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

Introduction to Building Dynamic Web Applications-Steps++

4. Performing CRUD Operations:

i. Create (Insert): You can insert data into the database by constructing an SQL query and executing it.

```php
$sql = "INSERT INTO your_table (column1, column2, column3)
VALUES ('value1', 'value2', 'value3')";
if ($conn->query($sql) === TRUE) {
    echo "Record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
```

Introduction to Building Dynamic Web Applications-Steps+++

4. Performing CRUD Operations:

ii. **Read (Select):** Retrieve data from the database using SQL SELECT queries and display it on your web page.

```php
$sql = "SELECT * FROM your_table";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
    // Display the data
    }
} else {
    echo "No records found";
}
```

Introduction to Building Dynamic Web Applications-Steps++++

4. Performing CRUD Operations:

iii. Update: Modify existing records in the database by constructing and executing SQL UPDATE queries.

```php
$sql = "UPDATE your_table SET column1 = 'new_value'
WHERE some_condition";
if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
```

Introduction to Building Dynamic Web Applications-Steps++++

4. Performing CRUD Operations:

vi. **Delete:** Remove records from the database using SQL DELETE queries.

```php
//Delete
$sql = "DELETE FROM your_table WHERE some_condition";
if ($conn->query($sql) === TRUE) {
    echo "Record deleted successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
```

Introduction to Building Dynamic Web Applications-Steps++++

5. Close Database Connection

It's important to close the database connection when you're done with it to free up resources.

```
//Close db connection
$conn->close();
```

6. Security Considerations

i. Sanitize user input and use prepared statements to prevent SQL injection attacks.

ii. Implement proper authentication and authorization to control access to your CRUD operations.

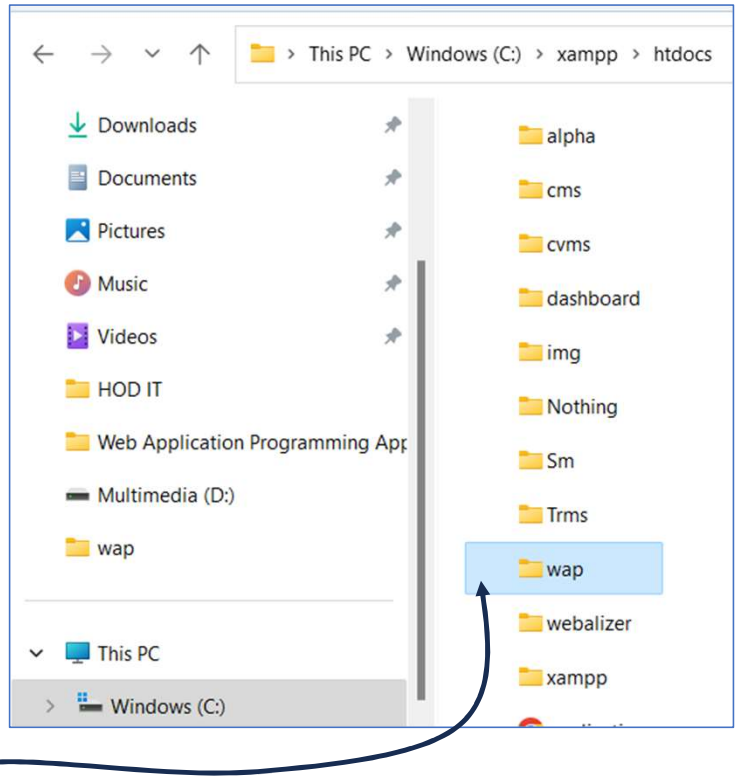
Building Dynamic Web Applications – Steps 1

Here's a high-level overview of how you can achieve this:

1.Setting Up Your Environment:

We finished to work on our environment in last lecture. Where xampp was installed containing everything we need to develop dynamic web applications.

We also created our domain directory called “wap”, placed inside htdocs folder.



Building Dynamic Web Applications – Steps 1

2. Create a Database through the following steps.

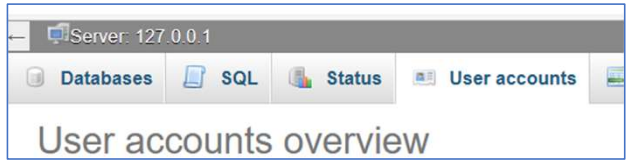
i. Open xampp, **Start** Appache server and SQL

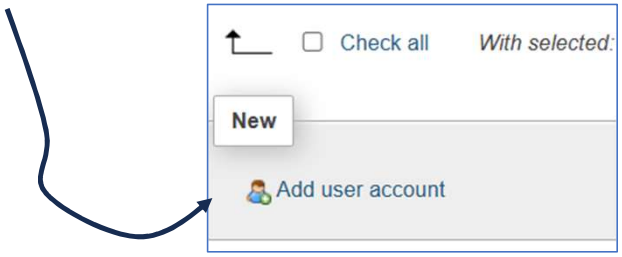
ii. Click Admin Button

iii. Click on phpMyAdmin menu found on the web page displayed

iv. Click User accounts

v. Click Add User accounts





Building Dynamic Web Applications – Steps 1+

vi. Enter User name e.g **wap**

vii. Select Host name as local

viii. Enter password e.g **wap**

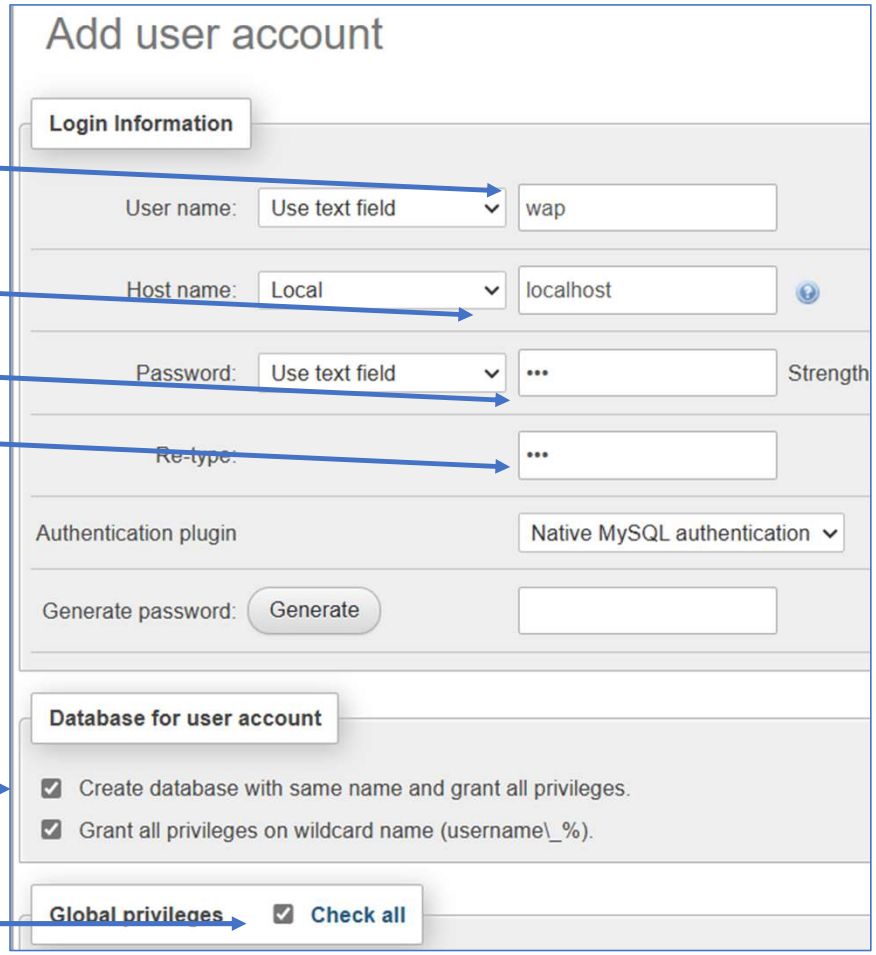
ix. Re-type password e.g **wap**

x. Check the box next to Create database with same name and grant privileges

xi. Check all Global privileges

xii. Click Go

xiii. Done



Creation of a table

Remember we created a database with the following

Name: wap

Username: wap

Password: wap

And **hostname:** localhost

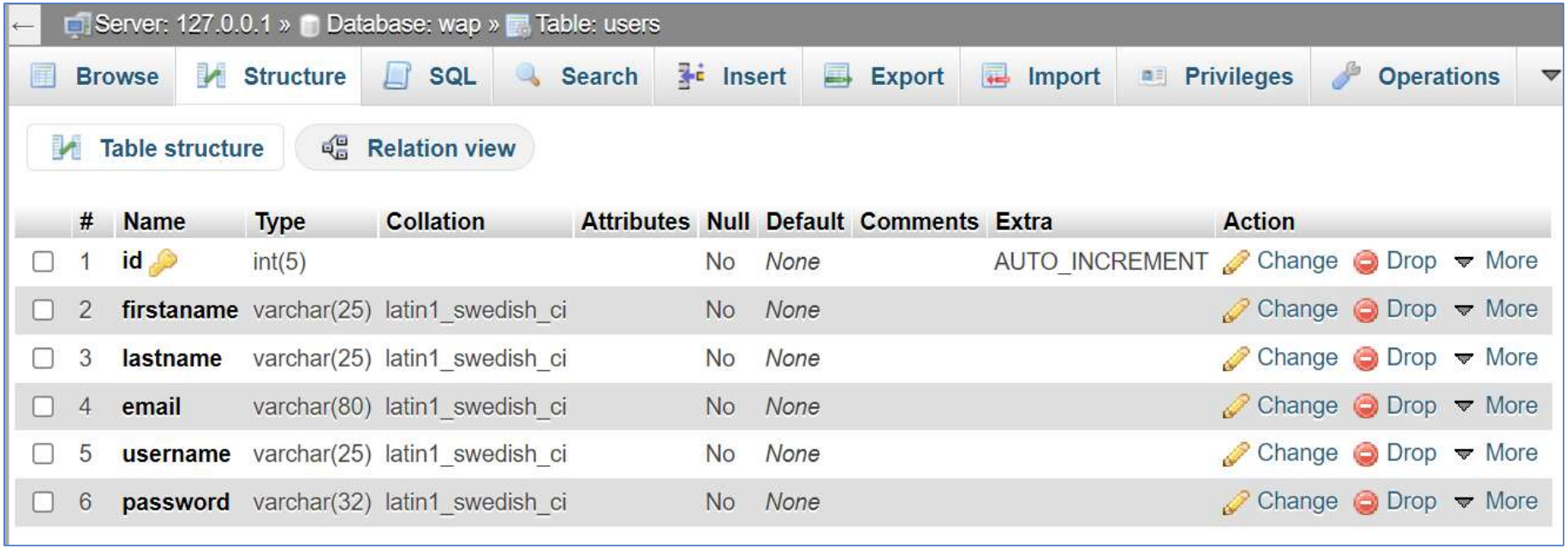
We now need to create a table with following details:

Table name: users

Fields: id, firstname, lastname, email, username and password

Creation of users table

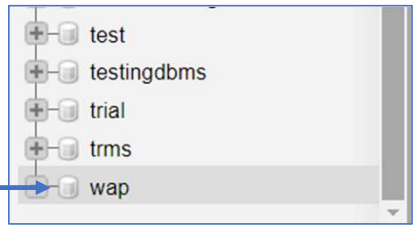
To enable us register and login users, we created a table called users with the following structure and attributes.

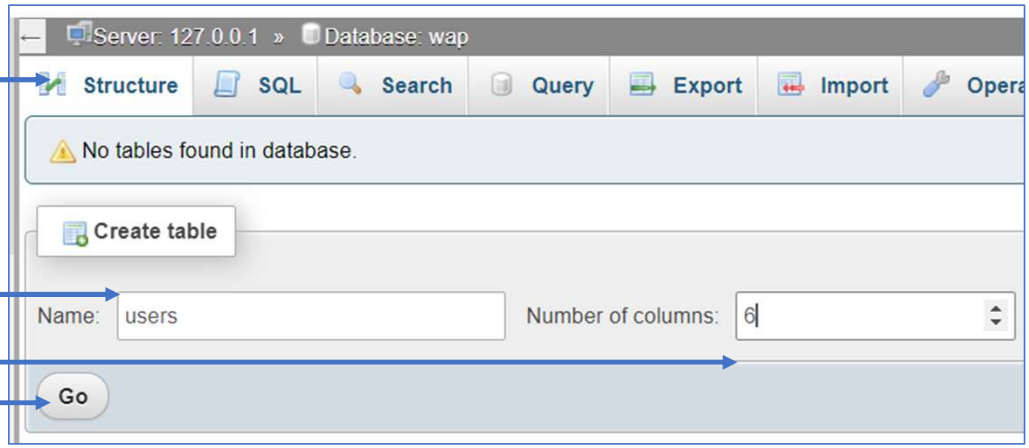


#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(5)			No	None		AUTO_INCREMENT	Change Drop More
2	firstaname	varchar(25)	latin1_swedish_ci		No	None			Change Drop More
3	lastname	varchar(25)	latin1_swedish_ci		No	None			Change Drop More
4	email	varchar(80)	latin1_swedish_ci		No	None			Change Drop More
5	username	varchar(25)	latin1_swedish_ci		No	None			Change Drop More
6	password	varchar(32)	latin1_swedish_ci		No	None			Change Drop More

Creation of users table-Steps

1. Click on the Database you would want to add the table e.g wap
2. Click structure
3. Enter table name
4. Enter number of columns
5. Click go





Introduction to Building Dynamic Web Applications-Steps+

Here's a high-level overview of how you can achieve this:

3. Establish a Database Connection:

In your PHP code, establish a connection to the database using the appropriate functions. For example, if you are using MySQL, you can use the mysqli to connect to the database.

```php
$server = "localhost";
$username = "wap";
$password = "wap";
$database = "wap";

$conn = new mysqli($server, $username, $password, $database);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

Introduction to Building Dynamic Web Applications-Steps++

4. Performing CRUD Operations:

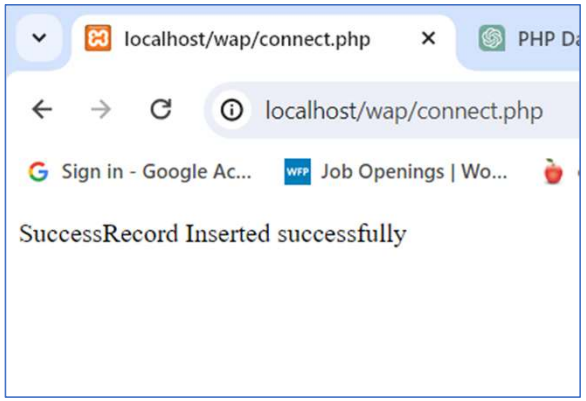
i. **Create (Insert):** You can insert data into the database by constructing an SQL query and executing it. Entre the following code into your php file. Eg. connect.php

```php
//insert
    $sql = "INSERT INTO `users`(`firstname`, `lastname`, `email`, `username`, `password`)
    VALUES ('Okullo','James','james@gmail.com','james','james')";
    if ($conn->query($sql) === TRUE) {
        echo "Record Inserted successfully";
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
```

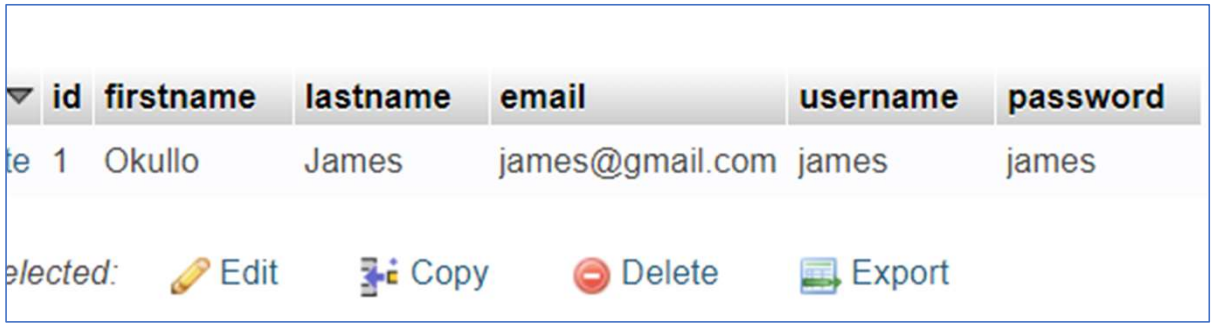
Now open the browser then enter: localhost/wap/connect.php

Introduction to Building Dynamic Web Applications-Steps++

Record inserted successfully



Now refresh your table.



Introduction to Building Dynamic Web Applications-Steps+++

4. Performing CRUD Operations:

ii. Read (Select): Retrieve data from the database using SQL SELECT queries and display it on your web page.

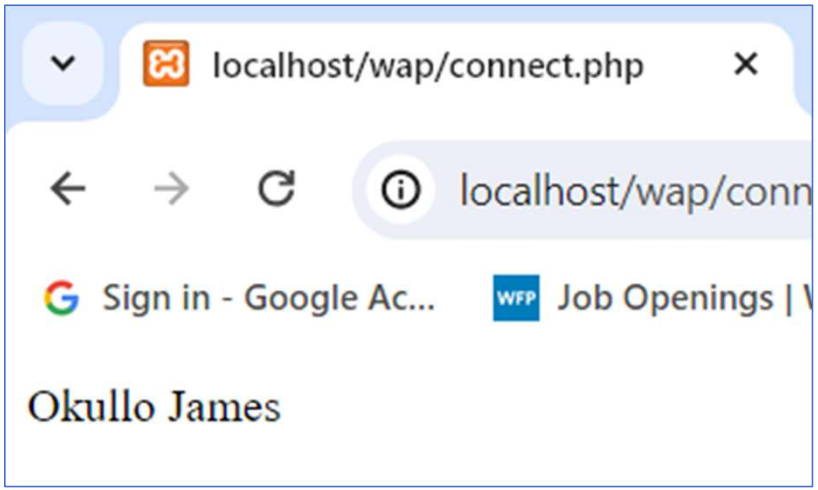
```php
//Read
$sql = "SELECT * FROM users";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
     echo "  " . $row["firstname"] . "  " . $row["lastname"] . "<br>";
    }
} else {
    echo "No records found";
}
```

Introduction to Building Dynamic Web Applications-Steps+++

4. Performing CRUD Operations:

ii. Read (Select): Retrieve data from the database using SQL SELECT queries and display it on your web page.

Output:



Building Dynamic Web Applications-Steps++++

4. Performing CRUD Operations:

iii. Update: Modify existing records in the database by constructing and executing SQL UPDATE queries.

Before we update data, kindly take note of the additional data on the table. We will therefore update record with id = 2 to have firstname as **Kadimba** instead of **Okullo**.

id	firstname	lastname	email	username	password
1	Okullo	James	james@gmail.com	james	james
2	Okullo	James	james@gmail.com	james	james
3	Okullo	James	james@gmail.com	james	kemba

Building Dynamic Web Applications-Steps++++

4. Performing CRUD Operations:

iii. Update: Modify existing records in the database by constructing and executing SQL UPDATE queries.

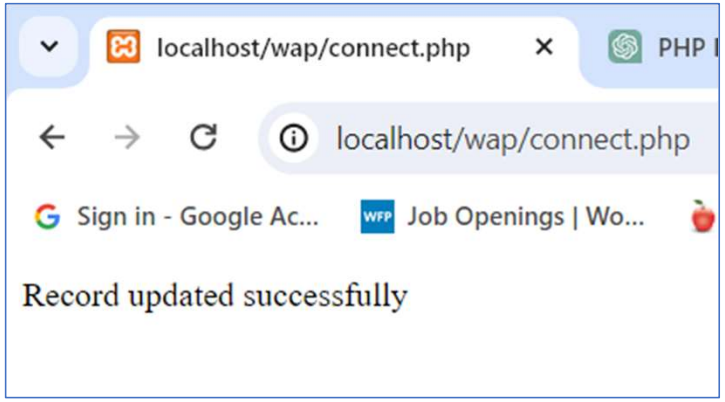
```php
//Update
$sql = "UPDATE users SET firstname = 'Kadimba' WHERE id='2'";
if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
```

Building Dynamic Web Applications-Steps++++

4. Performing CRUD Operations:

iii. Update: Modify existing records in the database by constructing and executing SQL UPDATE queries.

Output:



id	firstname	lastname	email	username	password
1	Okullo	James	james@gmail.com	james	james
2	Kadimba	James	james@gmail.com	james	james
3	Okullo	James	james@gmail.com	james	kemba

Note that firstname: '**Kadimba**' has replaced Okullo for record with id:2.

Building Dynamic Web Applications-Steps++++

4. Performing CRUD Operations:

vi. **Delete:** Remove records from the database using SQL DELETE queries.

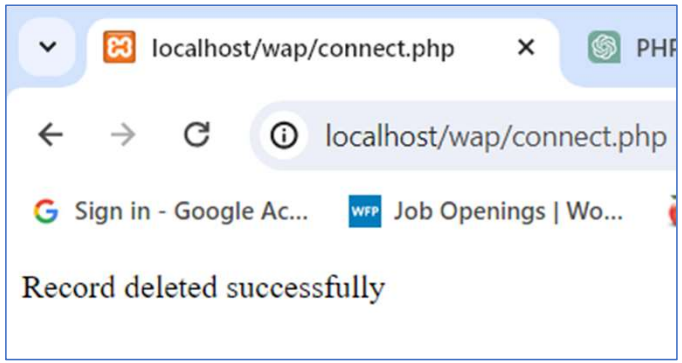
```php
//Delete
$sql = "DELETE FROM users WHERE id='3'";
if ($conn->query($sql) === TRUE) {
    echo "Record deleted successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
```

Building Dynamic Web Applications-Steps++++

4. Performing CRUD Operations:

vi. **Delete:** Remove records from the database using SQL DELETE queries.

Output:



id	firstname	lastname	email	username	password
1	Okullo	James	james@gmail.com	james	james
2	Kadimba	James	james@gmail.com	james	james

Note that record with id:3 has been deleted

Building Dynamic Web Applications-Steps++++

5. Close Database Connection

It's important to close the database connection when you're done with it to free up resources.

```
//Close db connection
$conn->close();
```

6. Security Considerations

i. Sanitize user input and use prepared statements to prevent SQL injection attacks.

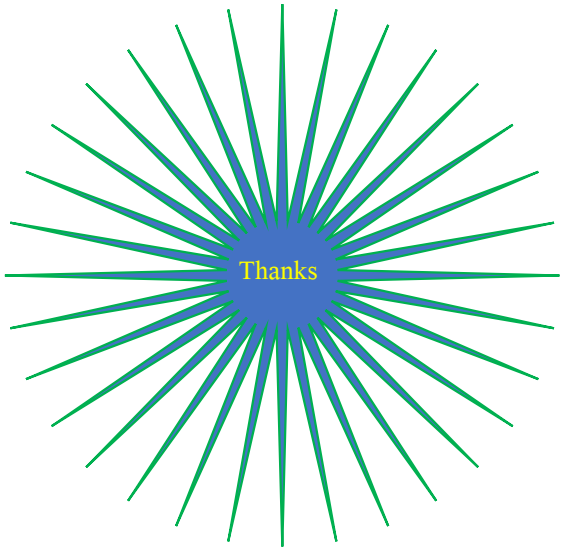
ii. Implement proper authentication and authorization to control access to your CRUD operations. This section will be handled in the next lecture(s)

Summary

Summary

1. Summary of previous lecture
2. Building Dynamic Web Applications(
    i. Introduction(Definition, x-tics and examples of dynamic content)
    ii. Creating a Database(wap) and table(users)
    iii. Establish a Database Connection using connect.php
    iv. Performing CRUD Operations(insert, read, update and delete)

Thank you for Listening



References

JavaScript Data Types. OpenAI Knowledge Base. OpenAI. (2021, September 22).