



1420-7001

By

Dr. Abdul Majeed (조교수)

1st Semester, 2025

웹 프로그래밍

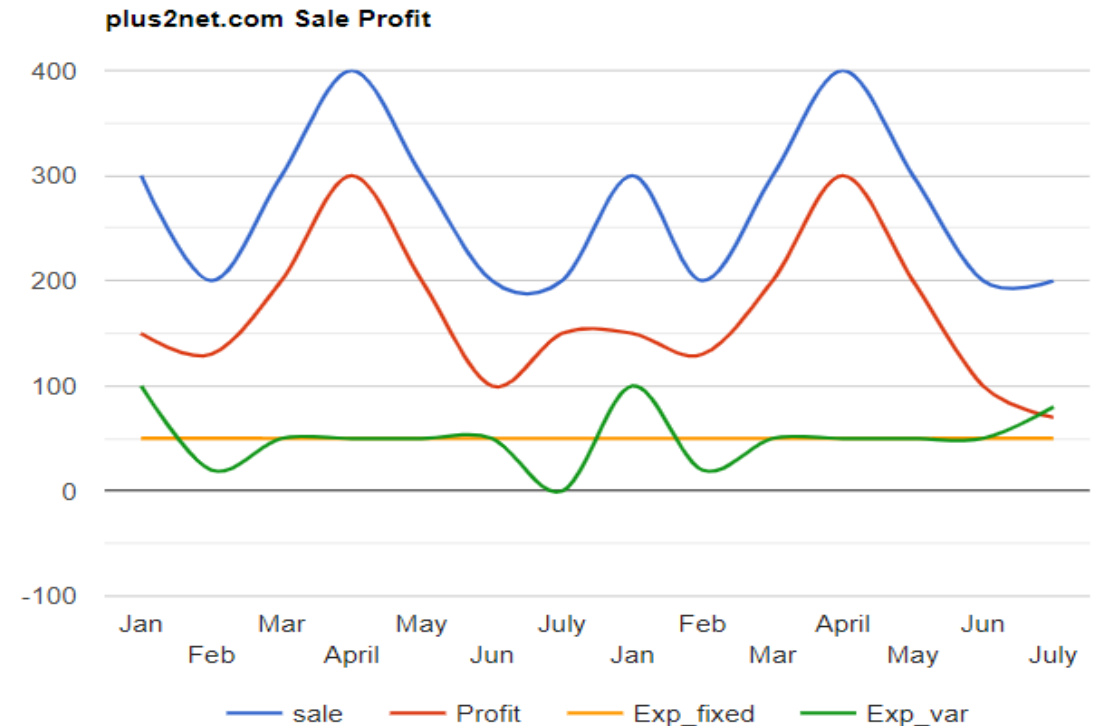
Web Programming



IT융합대학 컴퓨터공학부(컴퓨터공학전공)

Summary of the Today's Lesson

- Dynamic data visualizations
 - Database setup
 - PHP code setup
 - Front-end code setup
 - Display a bar graph in the Browser
- Dynamic data visualizations
 - Area graph
 - Line graph
 - Tools & libraries
 - Four examples
- Data visualization by reading data from a file
- Form data insertion to the DB [Next Class]





Part-I

Inserting Data into DB

Focus of the Today's Class-Overall Goal

- 2nd major objective of this course
- Designing the front side of data collection
- Backend side will be covered in last session of the class

(a) front-end design

A registration form with the following fields: First name, Last name, E-mail, and Gender (Male/Female radio buttons). At the bottom are Submit and Reset buttons. A red label '1st key concept' points to the form.

Html

First name: Abdul
Last name: Majeed
E-mail: ab09@gachon.ac.kr
☒ Male
☐ Female
Submit Reset

php

Abdul
Majeed
ab09@gachon.ac.kr
Male


2nd key concept

DB

Abdul
Majeed
ab09@gachon.ac.kr
Male

(b) back-end design

2nd Major Goal-{Data Storage}



Text:

Password:

Select:

Select 2:

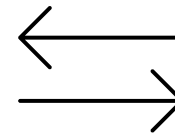
Radio: ☐ Yes ☐ No ☐ Maybe So

Radio 2: ☐ Yes ☐ No ☐ Maybe So

Checkbox: ☐

Checkbox 2: ☐

Buttons:

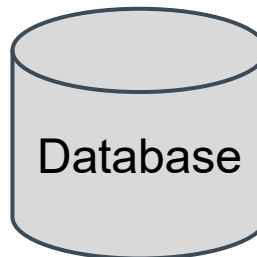
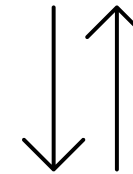


php scripts

Your Input:

Name: Abdul Majeed
Email: ab09@gachon.ac.kr
Mobile No: 1095039597
Website: <https://www.facebook.com/>
Gender: male

Collecting
right data



Data Insertion- Easy Case [one item at a time]

Registration Form

Name

Email

Create

ID	Name	Email
1	John Doe	johndoe@yahoo.com
2	Hans Duke	hans212@gmail.com
3	Super Mario	supermario@gmail.com
4	James Garvey	jamesgarvey@hotmail.cfr

Action

Read/retrieve

Update

Delete



Example 1

Form Data Insertion to XAMP Database

- Setting up database
 - ▣ Database creation
 - ▣ Table creation
 - # of columns
 - Label and datatypes of columns
- Designing server-side code
- Designing client-side code

<https://www.geeksforgeeks.org/how-to-insert-form-data-into-database-using-php/>

Overall Process of data Insertion

Storing Form data in Database

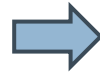
First Name:

Last Name:

Gender:

Address:

Email Address:



Storing Form data in Database

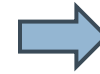
First Name:

Last Name:

Gender:

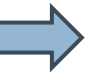
Address:

Email Address:



Entered data

sravan
kumar
male
Peddapalalakaluru Road
sravankumar8128@gmail.com



Data in
destination

localhost/phpmyadmin/sql.php?db=staff&table=college&pos=0

phpMyAdmin

Server: 127.0.0.1 » Database: staff » Table: college

Browse Structure SQL Search Insert Export Import

⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are disabled.

✓ Showing rows 0 - 0 (1 total, Query took 0.0026 seconds.)

```
SELECT * FROM `college`
```

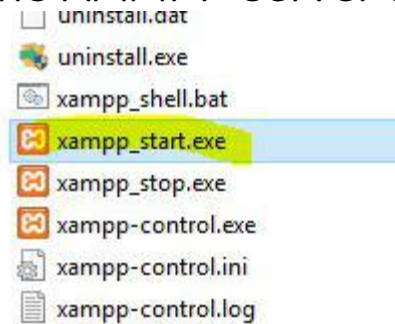
☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

first_name	last_name	gender	address	email
sravan	kumar	male	Peddapalalakaluru Road	sravankumar8128@gmail.com

Data insertion to DB- Database Setup

⌘ Open XAMPP to start the XAMPP server and click Start XAMPP.

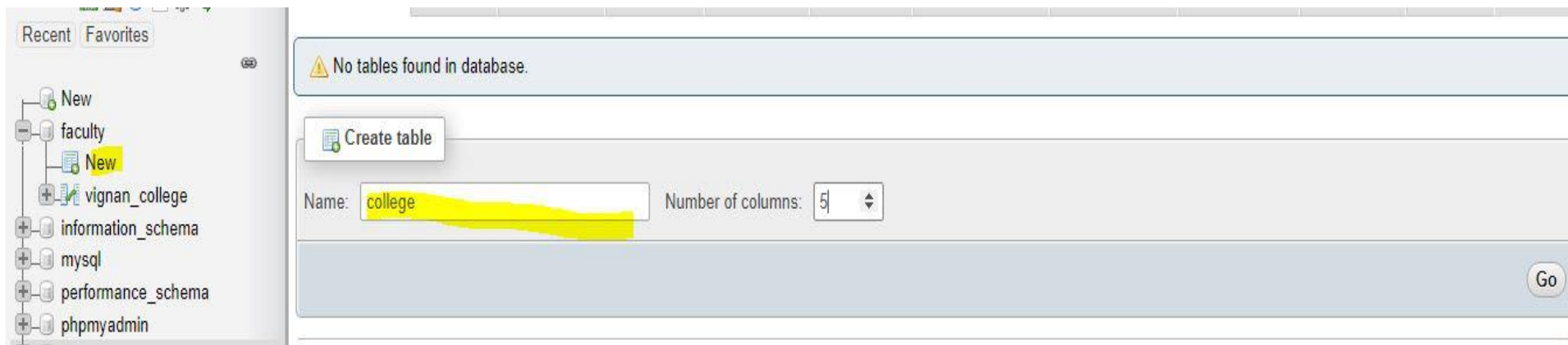


⌘ Open *localhost/phpmyadmin* in your web browser, name the database **staff** and click 'Create'.



Data insertion to DB- Database Setup

⌘ Then create a table named college .



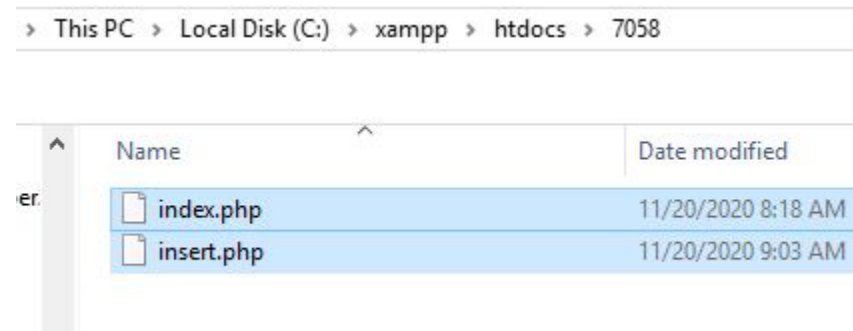
Data insertion to DB- Database Setup

⌘ Enter the columns and click Save.

Table name: college		Add	1	column(s)	Go					
first_name	VARCHAR	255	None							
Pick from Central Columns										
last_name	VARCHAR	255	None							
Pick from Central Columns										
gender	VARCHAR	255	None							
Pick from Central Columns										
address	VARCHAR	255	None							
Pick from Central Columns										
email	VARCHAR	255	None							
Pick from Central Columns										
Table comments:		Collation:		Storage Engine:						
				InnoDB						
PARTITION definition:										

Data insertion to DB- Client Side & Server-Side Code

- ⌘ Now open a notepad and write your PHP code and save it as index.php, then open another notepad and save it as insert.php. Save both files in a folder named htdocs .



Data insertion to DB- Client Side

⌘ This is the client side code.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>GFG- Store Data</title>
  </head>
  <body>
    <center>
      <h1>Storing Form data in Database</h1>
      <form action="insert.php" method="post">

<p>
      <label for="firstName">First Name:</label>
      <input type="text" name="first_name" id="firstName">
    </p>

<p>
      <label for="lastName">Last Name:</label>
      <input type="text" name="last_name" id="lastName">
    </p>

<p>
      <label for="Gender">Gender:</label>
      <input type="text" name="gender" id="Gender">
    </p>

<p>
      <label for="Address">Address:</label>
      <input type="text" name="address" id="Address">
    </p>

<p>
      <label for="emailAddress">Email Address:</label>
      <input type="text" name="email" id="emailAddress">
    </p>

      <input type="submit" value="Submit">
    </form>
  </center>
</body>
</html>
```

Data insertion to DB-Server Side

⌘ This is the server-side code.

```
<!DOCTYPE html>
<html>

<head>
  <title>Insert Page page</title>
</head>

<body>
  <center>
    <?php

      // servername => localhost
      // username => root
      // password => empty
      // database name => staff
      $conn = mysqli_connect("localhost", "root", "", "staff");

      // Check connection
      if($conn === false){
        die("ERROR: Could not connect. "
          . mysqli_connect_error());
      }

      // Taking all 5 values from the form data(input)
      $first_name = $_REQUEST['first_name'];
      $last_name = $_REQUEST['last_name'];
      $gender = $_REQUEST['gender'];
      $address = $_REQUEST['address'];
      $email = $_REQUEST['email'];

      // Performing insert query execution
      // here our table name is college
      $sql = "INSERT INTO college VALUES ('$first_name',
        '$last_name', '$gender', '$address', '$email')";

      if(mysqli_query($conn, $sql)){
        echo "<h3>data stored in a database successfully."
          . " Please browse your localhost php my admin"
          . " to view the updated data</h3>";

        echo nl2br("\n$first_name\n $last_name\n "
          . "$gender\n $address\n $email");
      } else{
        echo "ERROR: Hush! Sorry $sql. "
          . mysqli_error($conn);
      }

      // Close connection
      mysqli_close($conn);
    ?>
  </center>
</body>
</html>
```


Data insertion to DB-Run Code

When you type `localhost/7058/index.php` in your browser, the form will be displayed. After you submit the form, the form data will be submitted to the database.



Storing Form data in Database

First Name:

Last Name:

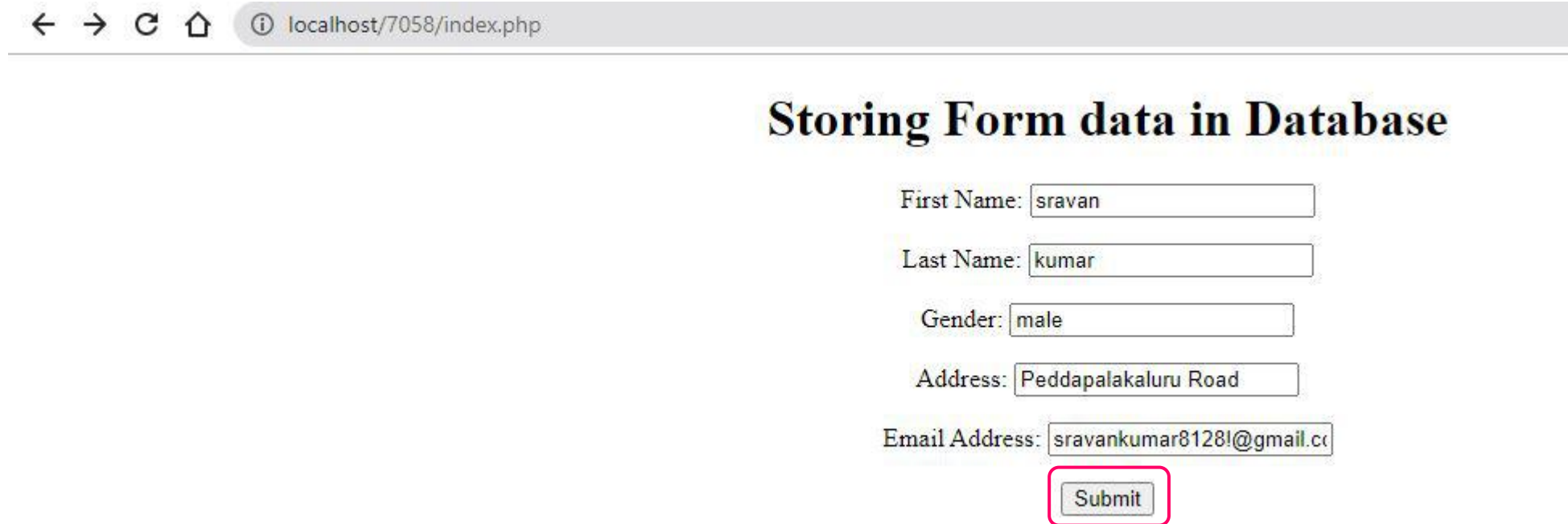
Gender:

Address:

Email Address:

Data insertion to DB-Run Code

Fill the relevant fields with your data & click submit.



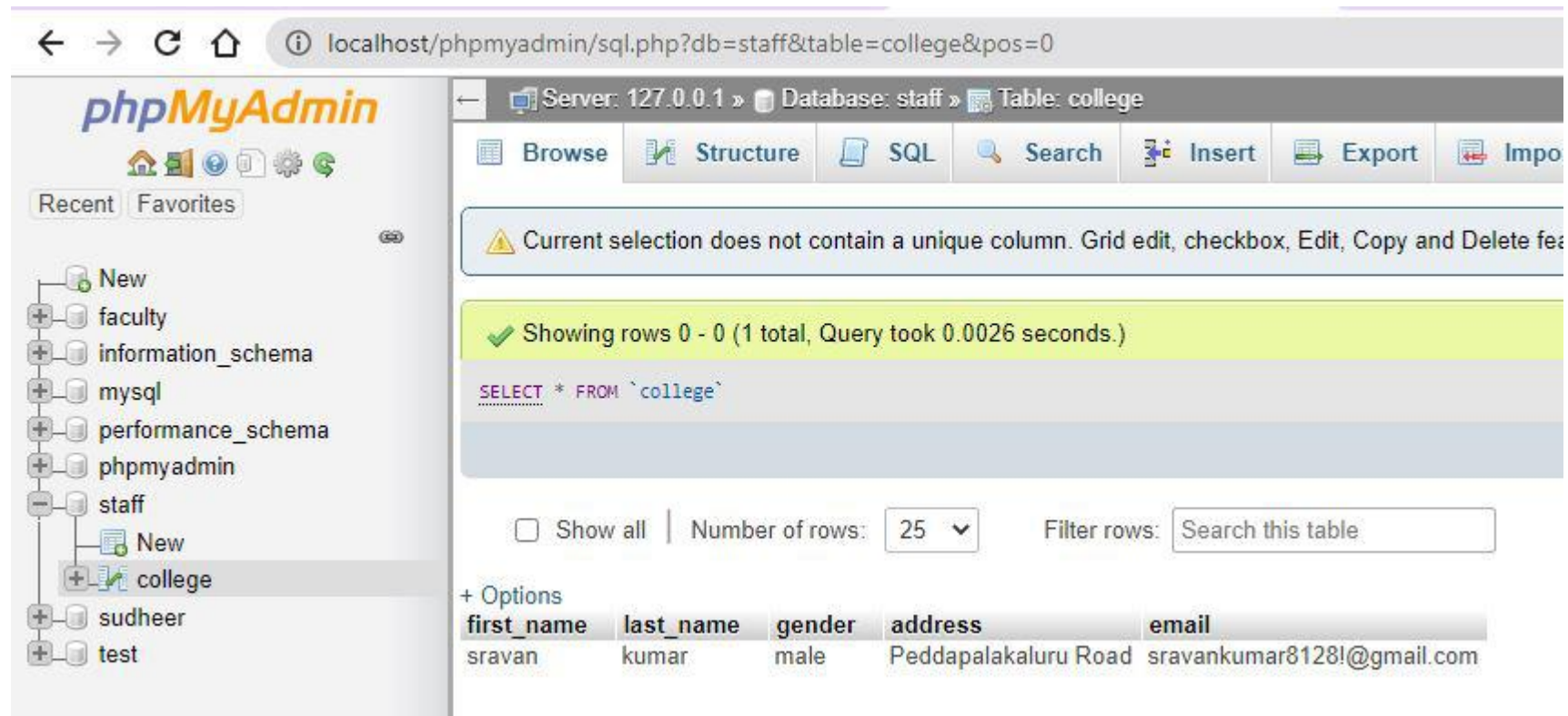
The screenshot shows a web browser window with the address bar displaying 'localhost/7058/index.php'. The page title is 'Storing Form data in Database'. The form contains the following fields and values:

Field	Value
First Name:	sravan
Last Name:	kumar
Gender:	male
Address:	Peddapalalakaluru Road
Email Address:	sravankumar8128!@gmail.co

A red rectangular box highlights the 'Submit' button at the bottom of the form.

Data insertion to DB-Run Code

The data will be added into the xamp database.



The screenshot displays the phpMyAdmin web interface. The left sidebar shows a tree view of databases, with 'staff' expanded and 'college' selected. The main panel shows the 'college' table structure with columns: first_name, last_name, gender, address, and email. A single row of data is displayed:

first_name	last_name	gender	address	email
sravan	kumar	male	Peddapalalakaluru Road	sravankumar8128!@gmail.com

The interface also shows a warning message: "Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are disabled." and a status message: "Showing rows 0 - 0 (1 total, Query took 0.0026 seconds.)". The SQL query bar contains: `SELECT * FROM `college``.

Data insertion to DB-Putting everything together

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>GFG- Store Data</title>
</head>
<body>
<center>
<h1>Storing Form data in Database</h1>
<form action="insert.php" method="post">

<p>
<label for="firstName">First Name:</label>
<input type="text" name="first_name" id="firstName">
</p>

<p>
<label for="lastName">Last Name:</label>
<input type="text" name="last_name" id="lastName">
</p>

<p>
<label for="Gender">Gender:</label>
<input type="text" name="gender" id="Gender">
</p>

<p>
<label for="Address">Address:</label>
<input type="text" name="address" id="Address">
</p>

<p>
<label for="emailAddress">Email Address:</label>
<input type="text" name="email" id="emailAddress">
</p>

<input type="submit" value="Submit">
</form>
</center>
</body>
</html>
```

Front-end



```
<!DOCTYPE html>
<html>
<head>
<title>Insert Page page</title>
</head>
<body>
<center>
<?php
// servername => localhost
// username => root
// password => empty
// database name => staff
$conn = mysqli_connect("localhost", "root", "", "staff");

// Check connection
if($conn == false){
die("ERROR: Could not connect. "
. mysqli_connect_error());
}

// Taking all 5 values from the form data(input)
$first_name = $_REQUEST['first_name'];
$last_name = $_REQUEST['last_name'];
$gender = $_REQUEST['gender'];
$address = $_REQUEST['address'];
$email = $_REQUEST['email'];

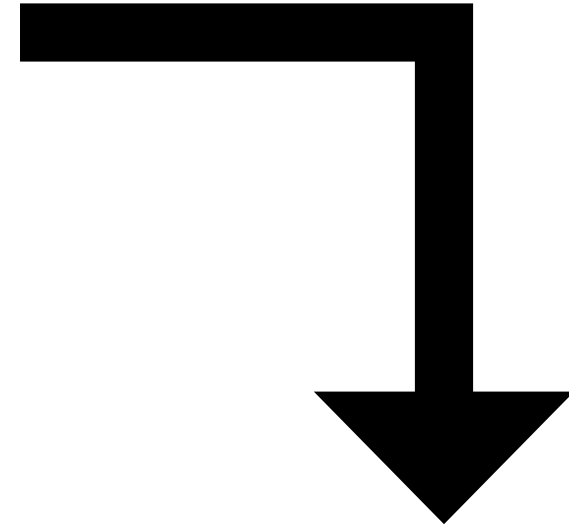
// Performing insert query execution
// here our table name is college
$sql = "INSERT INTO college VALUES ('$first_name',
'last_name','$gender','$address','$email')";

if(mysqli_query($conn, $sql)){
echo "<h3>data stored in a database successfully."
. " Please browse your localhost php my admin"
. " to view the updated data</h3>";

echo nl2br("\n$first_name\n $last_name\n "
. "$gender\n $address\n $email");
} else{
echo "ERROR: Hush! Sorry $sql. "
. mysqli_error($conn);
}

// Close connection
mysqli_close($conn);
?>
</center>
</body>
</html>
```

Back-end



Database

first_name	last_name	gender	address	email
sravan	kumar	male	Peddapalakaluru Road	sravankumar81281@gmail.com

Data insertion to DB-Putting everything together

Storing Form data in Database

First Name:
Last Name:
Gender:
Address:
Email Address:

(a) Front-end

Storing Form data in Database

First Name:
Last Name:
Gender:
Address:
Email Address:



data stored in a database successfully. Please browse your localhost php my admin to view the updated data

Abdul
Majeed
Male
Gachon University, Korea
ab09@gachon.ac.kr

(b) Back-end



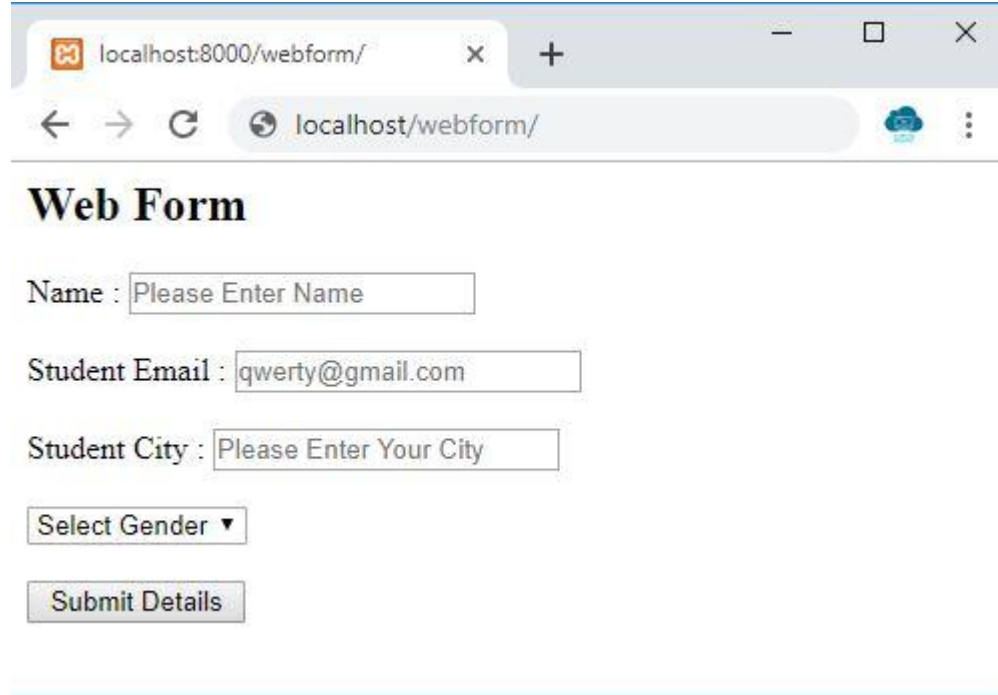
(c) Database

first_name	last_name	gender	address	email
Abdul	Majeed	Male	Gachon University, Korea	ab09@gachon.ac.kr



Example 2

Data Insertion- Complex Case



localhost:8000/webform/ x + - □ ×

localhost/webform/ ☁ ⋮

Web Form

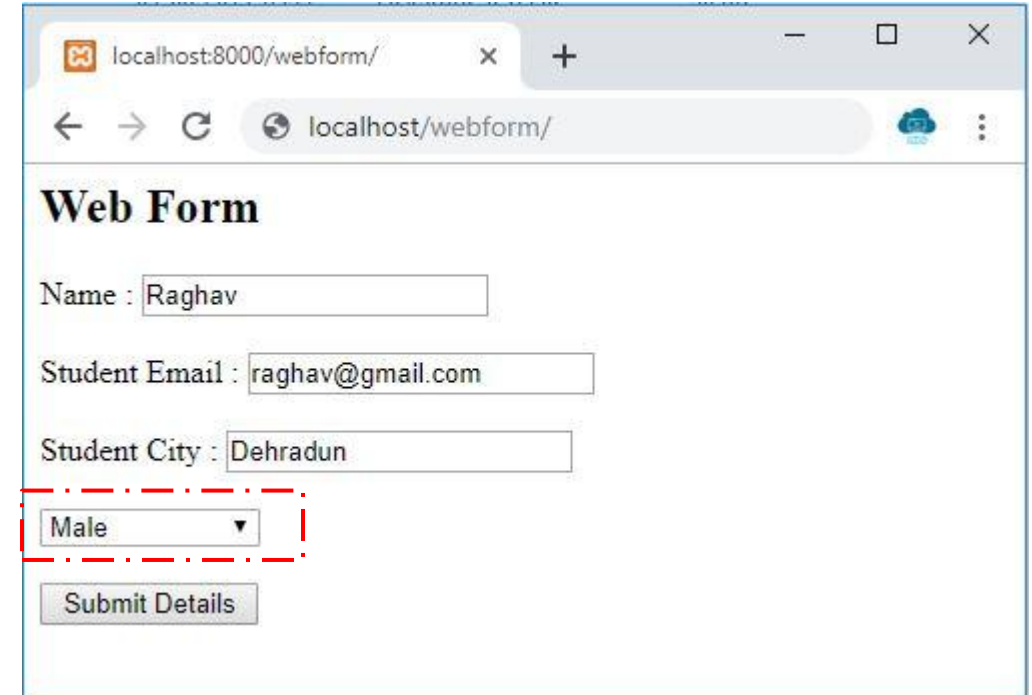
Name :

Student Email :

Student City :

Select Gender ▼

Submit Details



localhost:8000/webform/ x + - □ ×

localhost/webform/ ☁ ⋮

Web Form

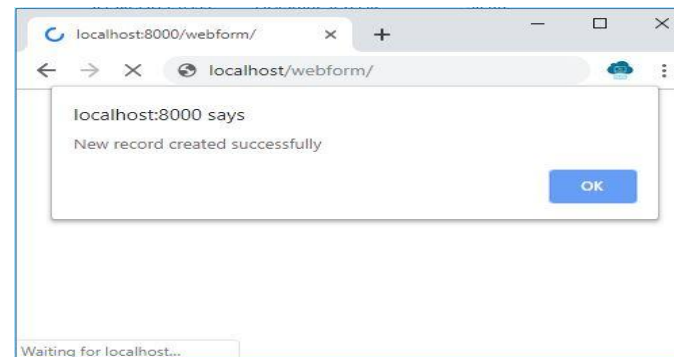
Name :

Student Email :

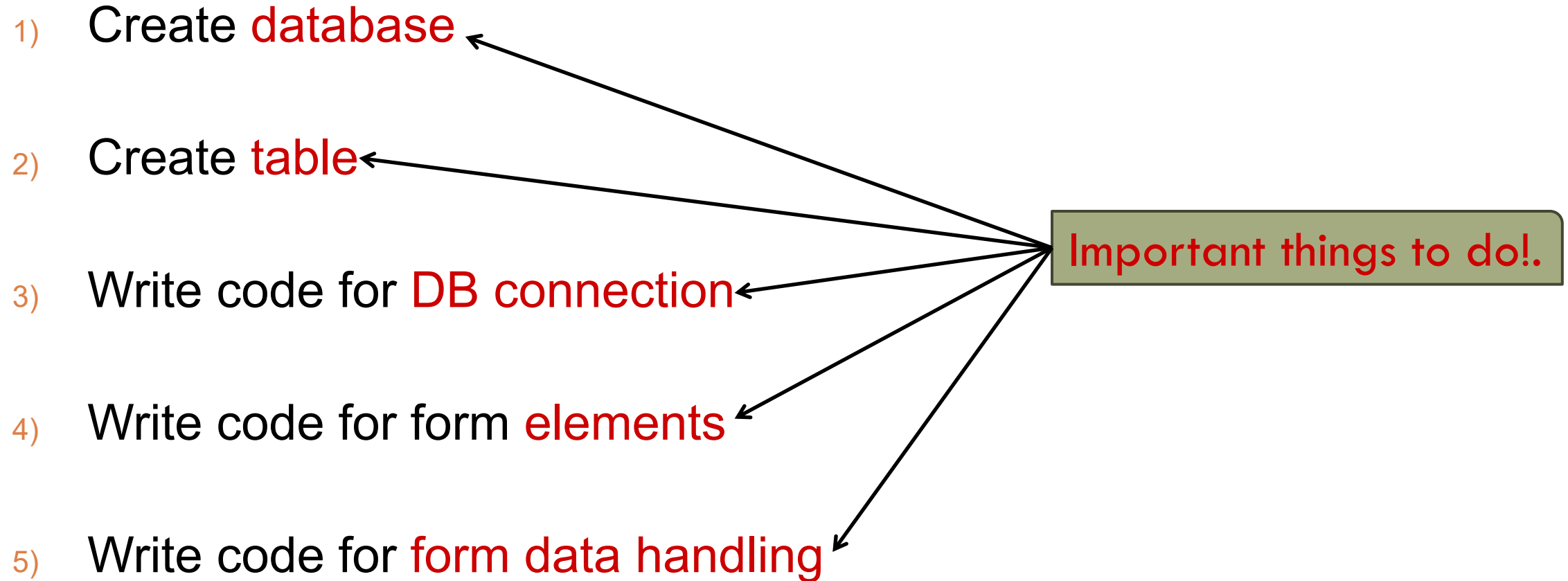
Student City :

Male ▼

Submit Details

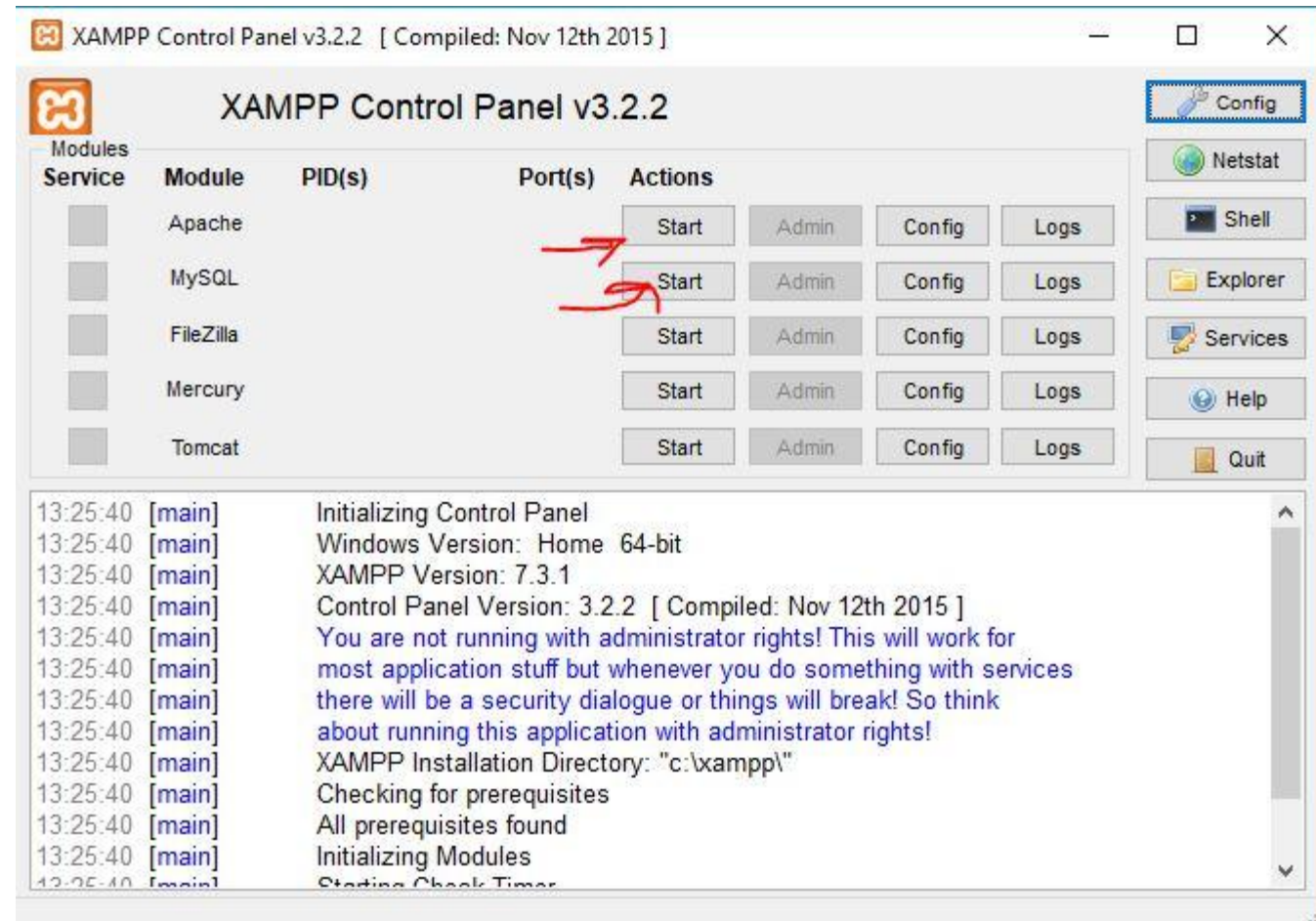


Steps to Perform Data Insertion



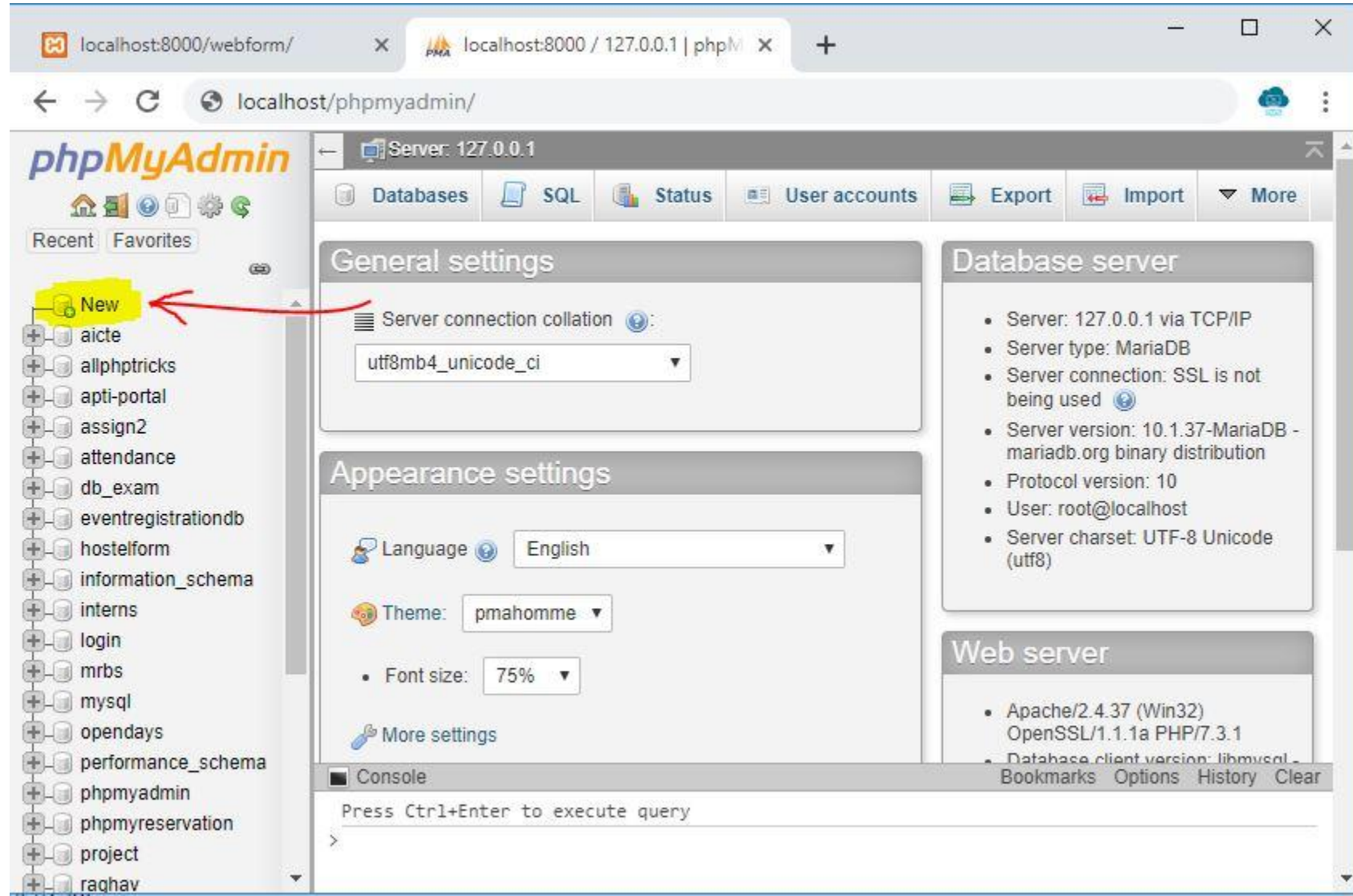
Step 1: Creating Database

- ✓ Start the Apache and MySQL servers in XAMPP.



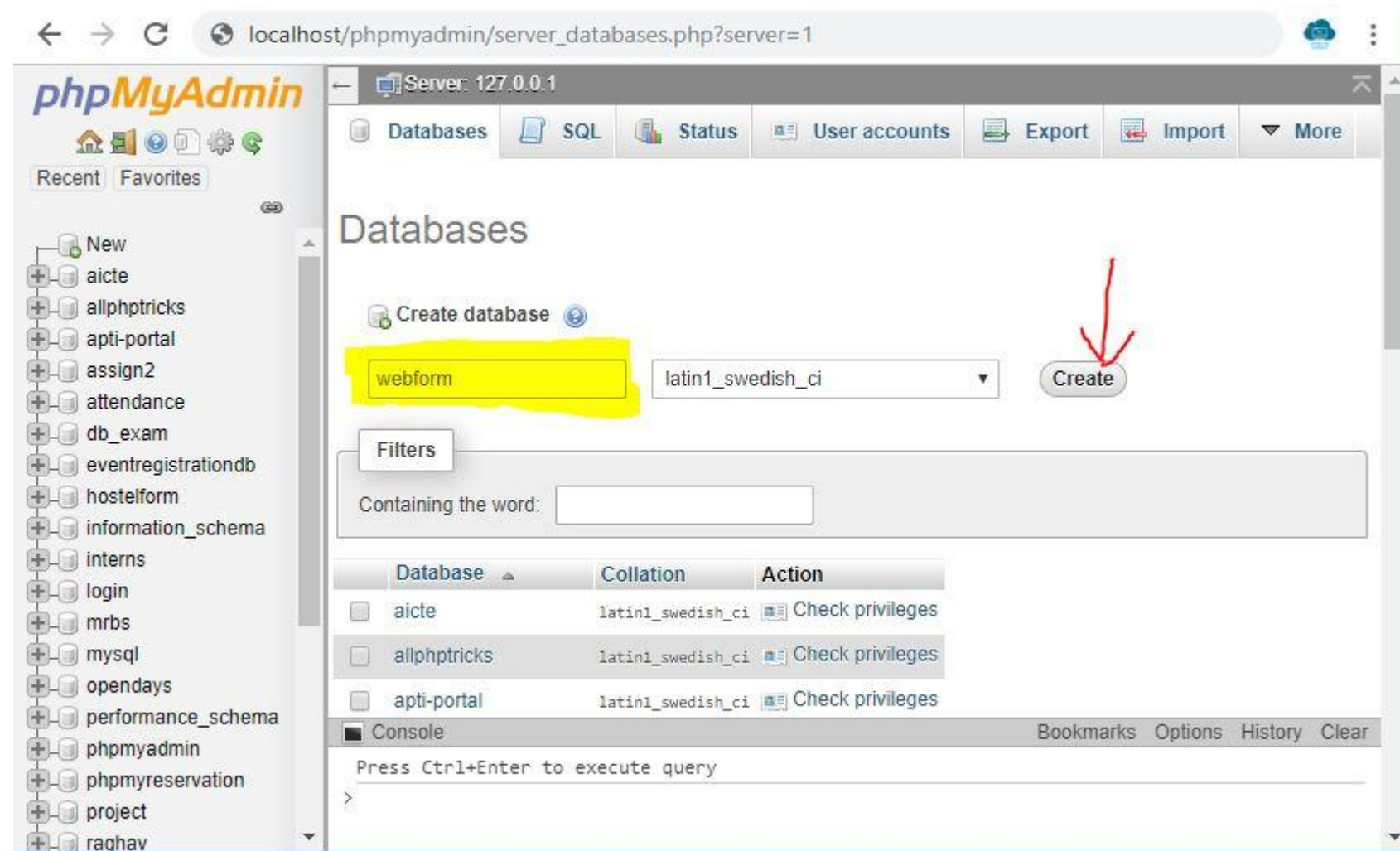
Step 1: Creating Database

- ❑ Now, go to your browser and open <http://localhost/phpmyadmin>
- ❑ In the left, go to New



Step 1: Creating Database

- Name the database **webform**.



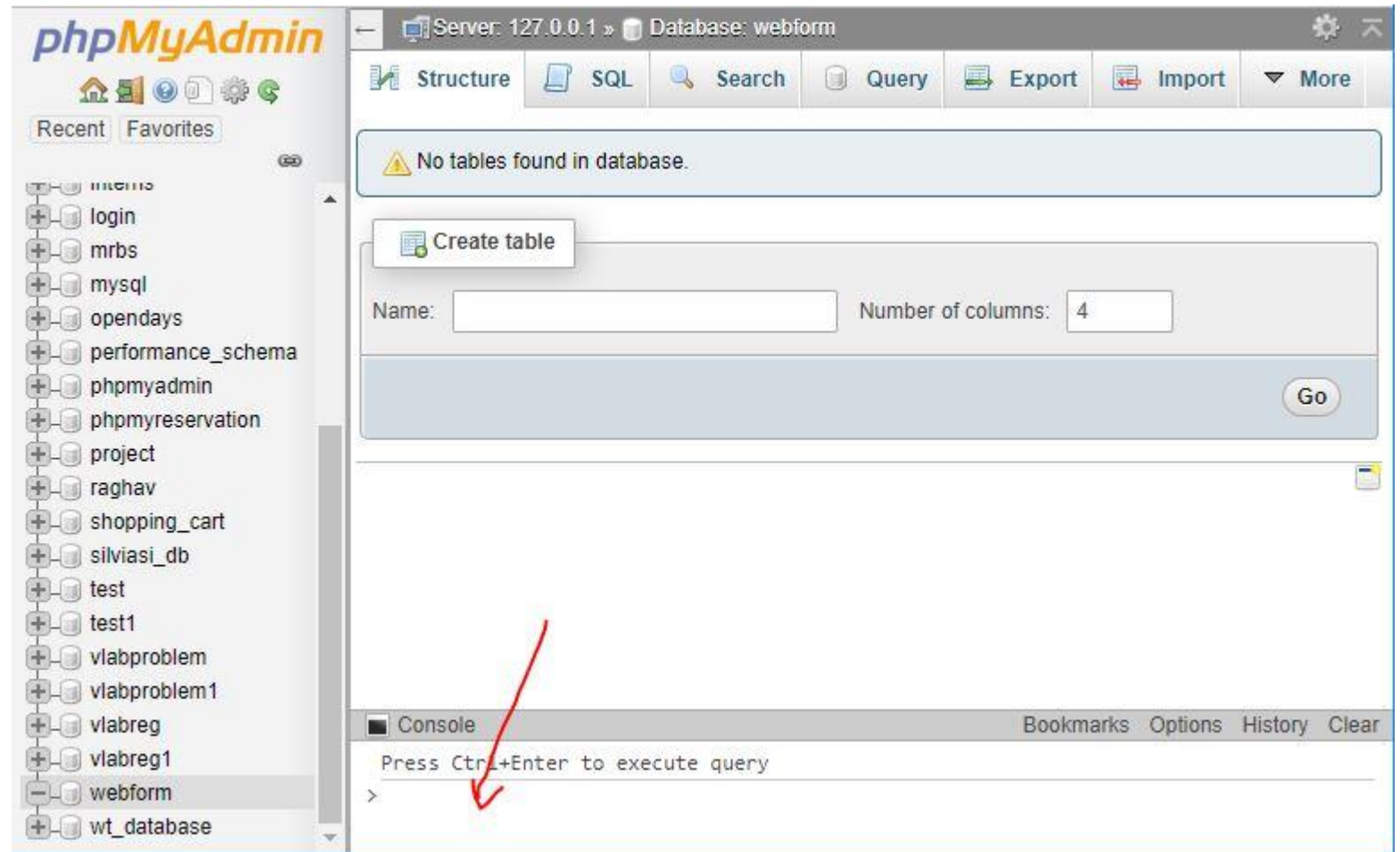
The screenshot shows the phpMyAdmin interface for creating a new database. The browser address bar indicates the URL is `localhost/phpmyadmin/server_databases.php?server=1`. The left sidebar lists various databases, including 'New', 'aicte', 'allphptricks', 'apti-portal', 'assign2', 'attendance', 'db_exam', 'eventregistrationdb', 'hostelform', 'information_schema', 'interns', 'login', 'mrbs', 'mysql', 'opendays', 'performance_schema', 'phpmyadmin', 'phpmyreservation', 'project', and 'raghav'. The main panel displays the 'Databases' section with a 'Create database' form. The form has a text input field containing 'webform' (highlighted in yellow), a collation dropdown menu set to 'latin1_swedish_ci', and a 'Create' button (indicated by a red arrow). Below the form is a 'Filters' section with a search box. At the bottom, there is a table listing existing databases and a console area.

Database	Collation	Action
<input type="checkbox"/> aicte	latin1_swedish_ci	Check privileges
<input type="checkbox"/> allphptricks	latin1_swedish_ci	Check privileges
<input type="checkbox"/> apti-portal	latin1_swedish_ci	Check privileges

Console: Press Ctrl+Enter to execute query

Step 1: Creating Database

- You can see the **webform** database now in your XAMP.



Step 2: Creating Table in a Database

- Create the table with names entries in XAMP.

```
CREATE TABLE entries
( id int(11) NOT NULL
AUTO_INCREMENT,
  name varchar(255) NOT NULL,
  email varchar(255) NOT NULL,
  city varchar(255) NOT NULL,
  gender varchar(255) NOT NULL,
PRIMARY KEY (id) )
```

The screenshot displays the phpMyAdmin web interface. On the left, a sidebar lists various databases, with 'webform' selected. The main panel shows the 'Database: webform' view. A message states 'No tables found in database.' Below this, the 'Create table' form is visible, with the 'Name' field empty and the 'Number of columns' set to 4. A 'Go' button is at the bottom right of the form. At the bottom of the interface, the 'Console' tab is active, showing the SQL query: `CREATE TABLE entries (id int(11) NOT NULL AUTO_INCREMENT, name varchar(255) NOT NULL, email varchar(255) NOT NULL, city varchar(255) NOT NULL, gender varchar(255) NOT NULL, PRIMARY KEY (id))`. The query is highlighted in yellow.

Step 3: Writing Code to link application with DB

- Configuration file to connect with the database.

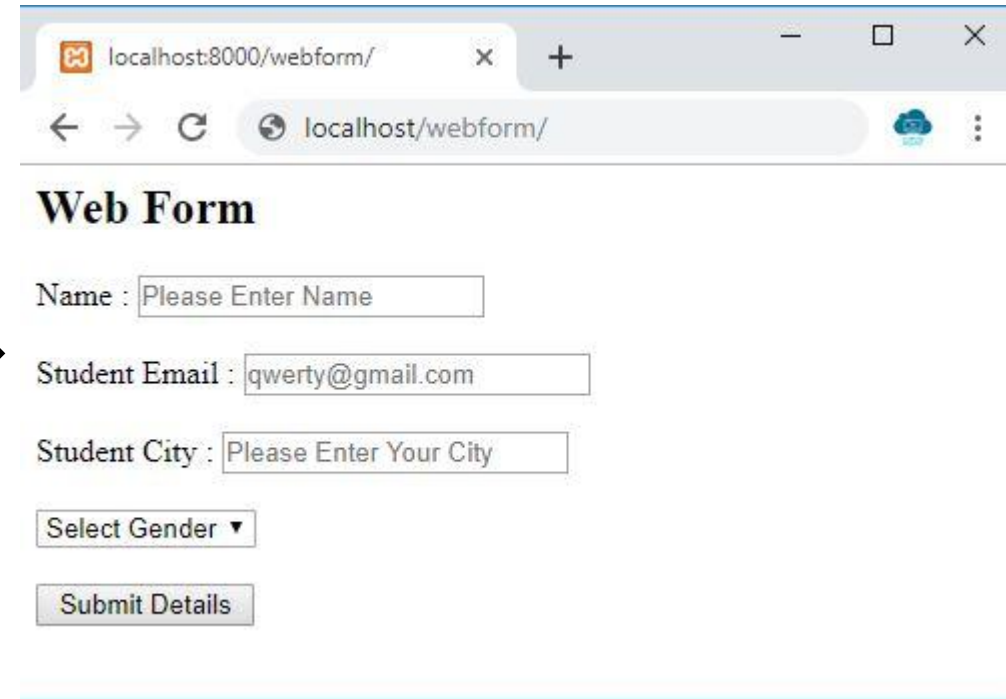
```
//Connecting to the database
<?php
$servername = "localhost";
$username = "root"; //edit if you have set a username for MySQL
$password = ""; // edit if you have set a password
$dbname = "webform";

// Create connection syntax
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```


Step 4: Writing Code to link application with DB

- Creating form to acquire data from the users.

```
<div id="webform">
<h2>Web Form</h2>
<form action="" method="post">
<label>Name :</label>
<input type="text" name="name" required placeholder="Please Enter Name"/><br><br>
<label>Student Email :</label>
<input type="email" name="email" required placeholder="qwerty@gmail.com"/><br><br>
<label>Student City :</label>
<input type="text" name="city" required placeholder="Please Enter Your City"/><br><br>
<select name="gender" required>
  <option value=""> Select Gender </option>
  <option value="M"> Male </option>
  <option value="F"> Female </option>
</select>
<br><br>
<input type="submit" value=" Submit Details " name="submit"/><br />
</form>
</div>
```



localhost:8000/webform/

localhost/webform/

Web Form

Name :

Student Email :

Student City :

Select Gender ▼

Submit Details

Step 5: Writing Code to Handle forms data

- ⌘ Till now, we have written the code for the form. Now, we will come to the main part, which is handling the data entered into this form.
- ⌘ In the above code, we have set the method of the form to **post**, and we have left the action attribute empty. By doing this, the form will post the data on the same page. If we had written the address to some other file/link in the action attribute, the form would have posted data to that particular address. I hope this makes sense.
- ⌘ Now take a close look at the following code

```
<?php
if(isset($_POST["submit"])){
include 'dbconfig.php';

$sql = "INSERT INTO entries (name, email, city, gender)
VALUES ('".$_POST["name"]."', '".$_POST["email"]."', '".$_POST["city"]."', '".$_POST["gender"]."')";

if ($conn->query($sql) === TRUE) {
echo "
    <script type= 'text/javascript'>
        alert('New record created successfully');
    </script>";
}
else
{
    echo
    "<script type= 'text/javascript'>
        alert('Error: " . $sql . "<br>" . $conn->error."');
    </script>";
}

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

Step 5: SQL Query to Insert forms data into DB

```
$sql = "INSERT INTO entries (name, email, city, gender)
VALUES ('".$_POST["name"]."', '".$_POST["email"]."', '".$_POST["city"]."', '".$_POST["gender"]."')";
```


Putting all things together

← → ↻ ⓘ http://localhost/webform/index.php

Web Form

Name :

Student Email :

Student City :

Select Gender ▼

Submit Details

1

Web Form

2

Name :

Student Email :

Student City :

Female ▼

Submit Details

//Connecting to the database

3

<https://iq.opengenus.org/html-form-database-insertion-php-mysql/>

	id	name	email	city	gender
<input type="checkbox"/> Edit Copy Delete	2	Hiba	hb@gmail.com	Guri	F



Getting Right Data

Data Validation

Data Validation

- Required field will check whether the field is filled or not in the proper way. Most of cases we will use the * symbol for required field.

What is Validation ?

Validation means check the input submitted by the user.

- **Client-Side Validation** – Validation is performed on the client machine web browsers.
- **Server Side Validation** – After submitted by data, The data has sent to a server and perform validation checks in server machine.

Validation for Input Field

- Some of Validation rules for field are given below.

Field	Validation Rules
Name	Should required letters and white-spaces
Email	Should required @ and .
Website	Should required a valid URL
Radio	Must be selectable at least once
Check Box	Must be checkable at least once
Drop Down menu	Must be selectable at least once

Client-Side Validation-Empty String

- The code below checks that the field is not empty.
- If the user leaves the required field empty, it will show an error message.

```
if (empty($_POST["name"])) {  
    $errMsg = "Error! You didn't enter the Name.";  
    echo $errMsg;  
} else {  
    $name = $_POST["name"];  
}
```

Client-Side Validation-**Validate String**

- The code below checks that the field will contain only alphabets and whitespace, for example - name.
- If the name field does not receive valid input from the user, then it will show an error message:

```
$name = $_POST ["Name"];  
if (!preg_match ("/^[a-zA-Z]*$/", $name) ) {  
    $ErrMsg = "Only alphabets and whitespace are allowed.";  
    echo $ErrMsg;  
} else {  
    echo $name;  
}
```

Client-Side Validation-**Validate Number**

- The below code validates that the field will only contain a numeric value. For example - Mobile no.
- If the Mobile no field does not receive numeric data from the user, the code will display an error message:

```
$mobilenno = $_POST ["Mobile_no"];  
if (!preg_match ("/^[0-9]*$/", $mobilenno) ){  
    $ErrMsg = "Only numeric value is allowed.";  
    echo $ErrMsg;  
} else {  
    echo $mobilenno;  
}
```

Client-Side Validation-**Validate Email**

- A valid email must contain @ and . symbols. PHP provides various methods to validate the email address.
- Here, we will use regular expressions to validate the email address.

```
$email = $_POST ["Email"];  
$pattern = "^[_a-z0-9-]+(\\.[_a-z0-9-]+)*@[a-z0-9-]+(\\.[a-z0-9-]+)*(\\.[a-z]{2,3})$^";  
if (!preg_match ($pattern, $email) ){  
    $ErrMsg = "Email is not valid.";   
    echo $ErrMsg;  
} else {  
    echo "Your valid email address is: " . $email;  
}
```


Client-Side Validation-Input Length Validation

- The input length validation restricts the user to provide the value between the specified range, for Example - Mobile Number. A valid mobile number must have 10 digits.

The given code will help you to apply the length validation on user input

```
$mobilenno = strlen ($_POST ["Mobile"]);  
$length = strlen ($mobilenno);  
  
if ( $length < 10 && $length > 10) {  
    $ErrMsg = "Mobile must have 10 digits.";  
    echo $ErrMsg;  
} else {  
    echo "Your Mobile number is: " .$mobilenno;  
}
```

Client-Side Validation-**Button Click Validation**

- The below code validates that the user click on submit button and send the form data to the server one of the following method - get or post.

```
if (isset ($_POST['submit'])) {  
    echo "Submit button is clicked.";  
    if ($_SERVER["REQUEST_METHOD"] == "POST") {  
        echo "Data is sent using POST method ";  
    }  
} else {  
    echo "Data is not submitted";  
}
```

<https://www.javatpoint.com/form-validation-in-php>

Client-Side Validation-Putting Everything Together

localhost/program/registration.p x +

← → ↻ 🏠 ⓘ localhost/program/registra... ☆ 🛑 📦

Apps IP http://192.168.254....

Registration Form

* required field

Name: *

E-mail: *

Mobile No: *

Website:

Gender: ☐ Male ☐ Female ☐ Other *

Agree to Terms of Service: ☐ *

Registration Form

* required field

Name: * Name is required

E-mail: * Email is required

Mobile No: * Mobile no is required

Website:

Gender: ☐ Male ☐ Female ☐ Other * Gender is required

Agree to Terms of Service: ☐ * Accept terms of services before submit.

You didn't filled up the form correctly.

Case I: Press submit
without data entry

<https://www.javatpoint.com/form-validation-in-php>

Client-Side Validation-Putting Everything Together

Case II: half data is correct
& half is incorrect.

Registration Form

* required field

Name: *

E-mail: *

Mobile No: *

Website:

Gender: ☒ Male ☐ Female ☐ Other *

Agree to Terms of Service: ☐ *



Registration Form

* required field

Name: *

E-mail: * Invalid email format

Mobile No: * Mobile no must contain 10 digits.

Website:

Gender: ☐ Male ☐ Female ☐ Other *

Agree to Terms of Service: ☐ * Accept terms of services before submit.

You didn't filled up the form correctly.

Client-Side Validation-Putting Everything Together

Case III: All data is correct.

Registration Form

* required field

Name: *

E-mail: *

Mobile No: *

Website:

Gender: ☒ Male ☐ Female ☐ Other *

Agree to Terms of Service: ☒ *



Registration Form

* required field

Name: *

E-mail: *

Mobile No: *

Website:

Gender: ☐ Male ☐ Female ☐ Other *

Agree to Terms of Service: ☐ *

You have successfully registered.

Your Input:

Name: Abdul Majeed

Email: ab09@gachon.ac.kr

Mobile No: 1095039597

Website: https://www.facebook.com/

Gender: male

Client-Side Validation-Putting Everything Together

```
<?php
// define variables to empty values
$nameErr = $emailErr = $mobilenoErr = $genderErr = $websiteErr = $agreeErr = "";
$name = $email = $mobileno = $gender = $website = $agree = "";

//Input fields validation
if ($_SERVER["REQUEST_METHOD"] == "POST") {

//String Validation
if (empty($_POST["name"])) {
    $nameErr = "Name is required";
} else {
    $name = input_data($_POST["name"]);
    // check if name only contains letters and whitespaces
    if (!preg_match("/^[a-zA-Z ]*$/", $name)) {
        $nameErr = "Only alphabets and white space are allowed";
    }
}

//Email Validation
if (empty($_POST["email"])) {
    $emailErr = "Email is required";
} else {
    $email = input_data($_POST["email"]);
    // check that the e-mail address is well-formed
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $emailErr = "Invalid email format";
    }
}

//Number Validation
if (empty($_POST["mobileno"])) {
    $mobilenoErr = "Mobile no is required";
} else {
    $mobileno = input_data($_POST["mobileno"]);
    // check if mobile no is well-formed
    if (!preg_match("/^[0-9]*$/", $mobileno)) {
        $mobilenoErr = "Only numeric value is allowed";
    }
    //check mobile no length should not be less and greater than 10
    if (strlen($mobileno) != 10) {
        $mobilenoErr = "Mobile no must contain 10 digits";
    }
}

//Gender Validation
if (empty($_POST["gender"])) {
    $genderErr = "Gender is required";
} else {
    $gender = input_data($_POST["gender"]);
    // check if gender is well-formed
    if (!preg_match("/^(male|female|other)$/i", $gender)) {
        $genderErr = "Invalid gender";
    }
}

//Website Validation
if (empty($_POST["website"])) {
    $websiteErr = "Website is required";
} else {
    $website = input_data($_POST["website"]);
    // check if website is well-formed
    if (!preg_match("/^[a-zA-Z0-9-_.]*$/", $website)) {
        $websiteErr = "Invalid website";
    }
}

//Agree to Terms of Service
if (empty($_POST["agree"])) {
    $agreeErr = "You must agree to the terms of service";
} else {
    $agree = input_data($_POST["agree"]);
    // check if agree is well-formed
    if ($agree != "yes") {
        $agreeErr = "You must agree to the terms of service";
    }
}
}
```

Registration Form

* required field

form method="post" action="<?php echo htmlspecialchars(\$_SERVER["PHP_SELF"]); ?>"

Name:

* <?php echo \$nameErr; ?>

E-mail:

* <?php echo \$emailErr; ?>

Mobile No:

* <?php echo \$mobilenoErr; ?>

Website:

* <?php echo \$websiteErr; ?>

Gender:

☐ Male

☐ Female

☐ Other

* <?php echo \$genderErr; ?>

Agree to Terms of Service:

☐

* <?php echo \$agreeErr; ?>

</form>

```
<?php
if(isset($_POST['submit'])) {
    if($nameErr == "" && $emailErr == "" && $mobilenoErr == "" && $genderErr == "" && $websiteErr == "" && $agreeErr == "") {
        echo "<h3 color = #FF0000> <b>You have successfully registered.</b>";
        echo "<h2>Your Input:</h2>";
        echo "Name: " . $name;
        echo "<br>";
        echo "Email: " . $email;
        echo "<br>";
        echo "Mobile No: " . $mobileno;
        echo "<br>";
        echo "Website: " . $website;
        echo "<br>";
        echo "Gender: " . $gender;
    } else {
        echo "<h3> <b>You didn't filled up the form correctly.</b> </h3>";
    }
}
?>
```

Summary of the Today's Lesson

- Form data insertion to DB
 - 1) Create **database**
 - 2) Create **table**
 - 3) Write code for **DB connection**
 - 4) Write code for form **elements**
 - 5) Write code for **form data handling**
- Getting right data
 - 1) Empty string
 - 2) Email validation
 - 3) Other input validations



Summary/Coverage of this course-Web Programming

- Familiarity with six web development languages & Tools
 - Html
 - CSS
 - JavaScript
 - PHP
 - SQL
 - Databases
- Three-tier web app development
 - Front end, backend, databases
- Understanding of design patterns
 - MVC
- Data visualization
 - Static scenario
 - Dynamic scenario
- Examples of right data processing
 - Data validation
- Examples of data insertion/retravel in databases
 - Data validation- getting the correct data
- Real-world examples of top websites
 - Things board
 - Coupang
 - Gachon university



- Six Assignments
- Topics Presentations
- Project Presentations
- Discussions
- Demos
- Six Class Activities

Final Exam Schedule & Course Included

Code	Subject	Students	Professor/Tutor	Date (Day)	Time	Place
1420-7001	Web Prog.	31	MAJEED ABDUL	13. 06 (FRI)	13:30 ~	AI공학관-408

Please be on time.
Please thoroughly check all assignments,
PPT, and slides.
Off course, you need preparation as well.

Exam Syllabus
Lecture Notes 8~13 along with Lab Practices.
HomeWorks 5~7

Exam style: Multiple choice questions, and short questions (Same as mid term exams).