WEB TECHNOLOGIES LAB

**WEEK -1**

**1. Create a web page with advanced layouts and positioning with CSS and HTML.**

<!DOCTYPE html>

<html lang="en">

<head>

<title>CSS Website Layout</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

\* {

box-sizing: border-box;

}

body {

margin: 0;

}

.header {

background-color: #f1f1f1;

padding: 20px;

text-align: center;

}

.topnav {

overflow: hidden;

background-color: #333;

}

.topnav a {

float: left;

display: block;

color: #f2f2f2;

text-align: center;

padding: 14px 16px;

text-decoration: none;

}

.topnav a:hover {

background-color: #ddd;

color: black;

}

.column {

float: left;

width: 33.33%;

padding: 15px;

}

.row:after {

content: "";

display: table;

clear: both;

}

</style>

</head>

<body>

<div class="header">

<h1>SNIST</h1>

<p>Sreenidhi Institute of Science and Technology</p>

</div>

<div class="topnav">

<a href="#">Home</a>

<a href="#">About</a>

<a href="#">Departmental Info</a>

</div>

<div class="row">

<div class="column">

<h2>College Info</h2>

<p>Sreenidhi Institute of Science and Technology–Hyderabad, sponsored by the Sree Educational Group was established in the year 1997 by the Chairman, Dr. K.T. Mahhe, an extraordinary educationist, a pragmatic leader, and a dynamic entrepreneur with a rich experience in Academics and Industry. A genuine commitment to the mission and dedicated leadership make SNIST one of the finest and most well-recognized higher educational institutions in India.</p>

</div>

<div class="column">

<h2>Branches Info</h2>

<p>CSE,IT,ECE,EEE,MECH,CIVIL.....</p>

</div>

<div class="column">

<h2>Address</h2>

<p>GHATKESAR,YAMNAMPET,HYDERABAD</p>

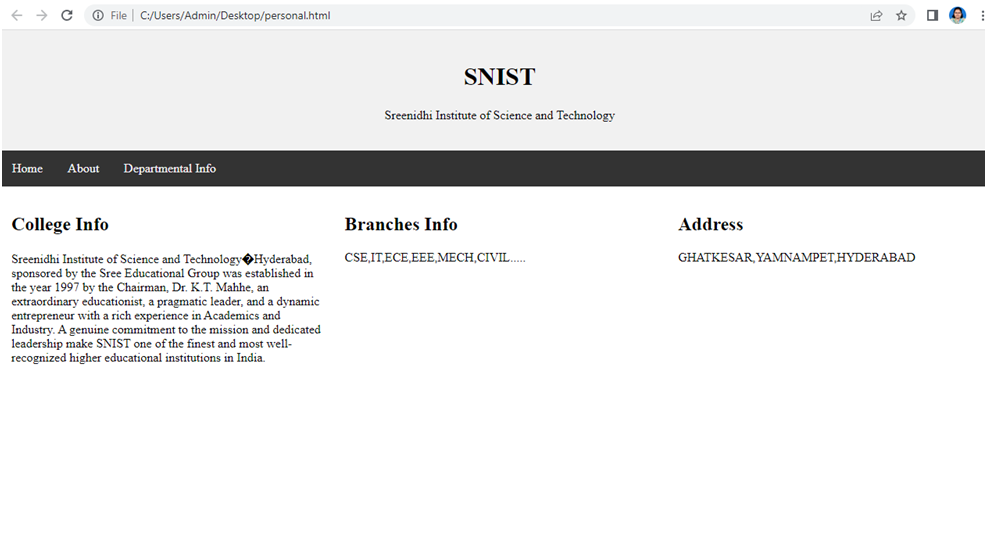
</div>

</div>

</body>

</html>

**Output:**



**2. Design a website with different methods of embedding CSS in a web page.**

**Inline CSS:**

<!DOCTYPE html>

<html>

<body>

<h1 style="color:blue;">A Blue Heading</h1>

<p style="color:red;">A red paragraph.</p>

</body>

</html>

**OUTPUT:**

A Blue Heading

**A red paragraph.**

**Internal CSS:**

<!DOCTYPE html>

<html>

<head>

<style>

body {background-color: powderblue;}

h1 {color: blue;}

p {color: red;}

</style>

</head>

<body>

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

</body>

</html>

**OUTPUT:**

This is a heading

**This is a paragraph.**

**Extenal CSS:**

**Html File:**

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

</body>

</html>

**CSS File:**

body {

background-color: powderblue;

}

h1 {

color: blue;

}

p {

color: red;

}

**OUTPUT:**

This is a heading

**This is a paragraph.**

**3. Create a static web page which displays your personal details.**

<!DOCTYPE html>

<html>

<head>

<title>Personal Information</title>

<style>

\* {

margin: 0;

padding: 0;

}

.firstsection {

background-color: green;

height: 400px;

}

.secondsection {

background-color: blue;

height: 400px;

}

.box-main {

display: flex;

justify-content: center;

align-items: center;

color: black;

max-width: 80%;

margin: auto;

height: 80%;

}

.firsthalf {

width: 100%;

display: flex;

flex-direction: column;

justify-content: center;

}

.secondhalf {

width: 30%;

}

.text-big {

font-family: 'Piazzolla', serif;

font-weight: bold;

font-size: 35px;

}

.text-small {

font-size: 18px;

}

.section {

height: 400px;

display: flex;

align-items: center;

justify-content: center;

max-width: 90%;

margin: auto;

}

</style>

</head>

<body>

<section class="firstsection">

<div class="box-main">

<div class="firsthalf">

<h1 class="text-big" id="web">Personal Information</h1>

<br><br>

<p class="text-small">

Name-Moiz

<br>

Roll Number-21311A6601 <br>

SECTION-IT A

</p>

</div>

</div>

</section>

<section class="secondsection">

<div class="box-main">

<div class="firsthalf">

<h1 class="text-big" id="program">

Contact Details

</h1>

<br>

<br>

<p class="text-small">

E-Mail - xyz@gmail.com

<br>

Mobile Number - 99999999999

</p>

</div>

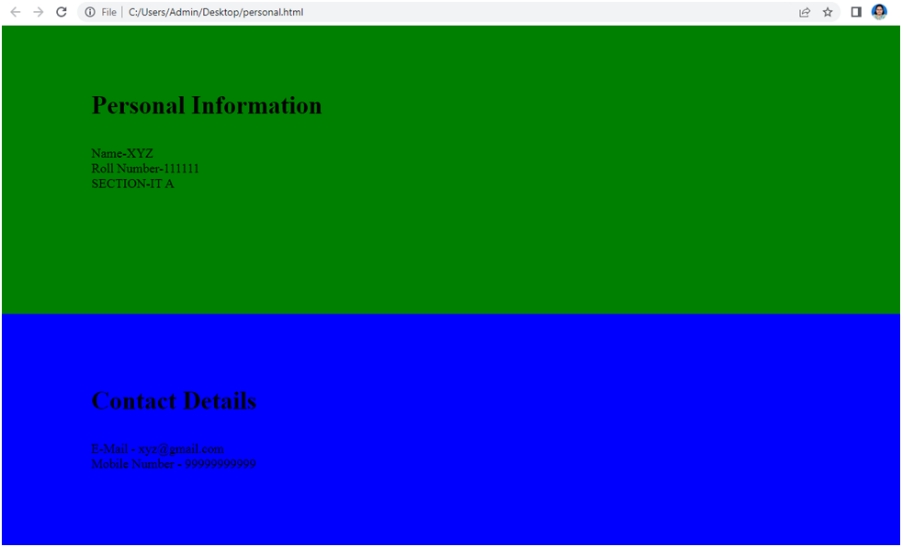
</div>

</section>

</body>

</html>

**OUTPUT:**



**4. Create a web page through which the user can enter his / her details to become an authenticated user of that page.**

**HTML File:**

<form action="action\_page.php" style="border:1px solid #ccc">

<div class="container">

<h1>Sign Up</h1>

<p>Please fill in this form to create an account.</p>

<hr>

<label for="email"><b>Email</b></label>

<input type="text" placeholder="Enter Email" name="email" required>

<label for="psw"><b>Password</b></label>

<input type="password" placeholder="Enter Password" name="psw" required>

<label for="psw-repeat"><b>Repeat Password</b></label>

<input type="password" placeholder="Repeat Password" name="psw-repeat" required>

<label>

<input type="checkbox" checked="checked" name="remember" style="margin-bottom:15px"> Remember me

</label>

<p>By creating an account you agree to our <a href="#" style="color:dodgerblue">Terms & Privacy</a>.</p>

<div class="clearfix">

<button type="button" class="cancelbtn">Cancel</button>

<button type="submit" class="signupbtn">Sign Up</button>

</div>

</div>

</form>

CSS File:

\* {box-sizing: border-box}

input[type=text], input[type=password] {

width: 100%;

padding: 15px;

margin: 5px 0 22px 0;

display: inline-block;

border: none;

background: #f1f1f1;

}

input[type=text]:focus, input[type=password]:focus {

background-color: #ddd;

outline: none;

}

hr {

border: 1px solid #f1f1f1;

margin-bottom: 25px;

}

button {

background-color: #04AA6D;

color: white;

padding: 14px 20px;

margin: 8px 0;

border: none;

cursor: pointer;

width: 100%;

opacity: 0.9;

}

button:hover {

opacity:1;

}

.cancelbtn {

padding: 14px 20px;

background-color: #f44336;

}

.cancelbtn, .signupbtn {

float: left;

width: 50%;

}

.container {

padding: 16px;

}

.clearfix::after {

content: "";

clear: both;

display: table;

}

@media screen and (max-width: 300px) {

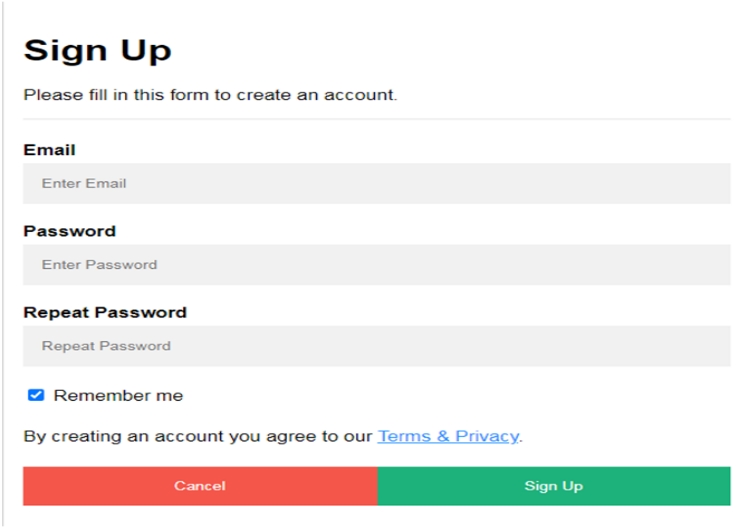
.cancelbtn, .signupbtn {

width: 100%;

}

}

**OUTPUT:**

****

**WEEK-2**

**1. Create a web page that shows different methods of embedding JavaScript with validation.**

JavaScript in HTML document

There are two general areas in HTML document where JavaScript can be placed. First is between <head>......</head> section, another is specific location in <body>......</body> section. If you want to display a message 'Good Morning' (through the JavaScript alert command) at the time of page loading then you must place the script at the <head>......</head> section. In the following examples you will see the different location of <script>.....</script> tags in a HTML document.

Scripts in the Head and Body

<!DOCTYPE html>

<head>

<meta charset="utf-8" />

<title> Script in head and body section </title>

<script type = "text/javascript">

JavaScript statements.......

</script>

</head>

<body>

<script type = "text/javascript">

JavaScript statements.......

</script>

</body>

</html>

Two Scripts in the Body

<!DOCTYPE html>

<head>

<meta charset="utf-8" />

<title> Two Scripts in the Body </title>

</head>

<body>

<script type = "text/javascript" scr="jsexample.js" >

</script>

<script type = "text/javascript">

JavaScript statements.......

</script>

</body>

</html>

**2. Create a web page with rollover menus. Rollover menus should be created using JavaScript.**

**Dropdown**

A dropdown menu is a toggleable menu that allows the user to choose one value from a predefined list:

**Create a Clickable Dropdown**

Create a dropdown menu that appears when the user clicks on a button.

**Step 1) Add HTML:**

<div class="dropdown">

<button onclick="myFunction()" class="dropbtn">Dropdown</button>

<div id="myDropdown" class="dropdown-content">

<a href="#">Link 1</a>

<a href="#">Link 2</a>

<a href="#">Link 3</a>

</div>

</div>

**CSS:**

.dropbtn {

background-color: #3498DB;

color: white;

padding: 16px;

font-size: 16px;

border: none;

cursor: pointer;

}

.dropbtn:hover, .dropbtn:focus {

background-color: #2980B9;

}

.dropdown {

position: relative;

display: inline-block;

}

.dropdown-content {

display: none;

position: absolute;

background-color: #f1f1f1;

min-width: 160px;

box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);

z-index: 1;

}

.dropdown-content a {

color: black;

padding: 12px 16px;

text-decoration: none;

display: block;

}

.dropdown-content a:hover {background-color: #ddd;

}

.show {display:block;}

**JavaScript:**

function myFunction() {

document.getElementById("myDropdown").classList.toggle("show");

}

window.onclick = function(event) {

if (!event.target.matches('.dropbtn')) {

var dropdowns = document.getElementsByClassName("dropdown-content");

var i;

for (i = 0; i < dropdowns.length; i++) {

var openDropdown = dropdowns[i];

if (openDropdown.classList.contains('show')) {

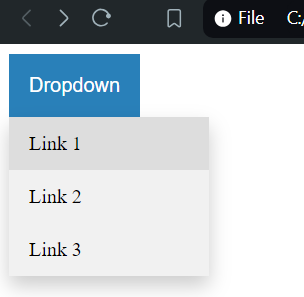
openDropdown.classList.remove('show');

}

}

}}

**Output:**

****

**3. Create a simple calculator, which can perform the basic arithmetic operations**

function calculate(operator, number1, number2) {

  switch (operator) {

    case '+':

      return number1 + number2;

    case '-':

      return number1 - number2;

    case '\*':

      return number1 \* number2;

    case '/':

      return number2 !== 0 ? number1 / number2 : "Division by zero error";

    default:

      return "Invalid operator";

  }

}

const operator = prompt('Enter operator ( +, -, \*, / ): ');

const number1 = parseFloat(prompt('Enter first number: '));

const number2 = parseFloat(prompt('Enter second number: '));

const result = calculate(operator, number1, number2);

console.log(`${number1} ${operator} ${number2} = ${result}`);

Steps to execute:

1. Open a web browser (Google Chrome, Microsoft Edge, etc.)

2. Press Ctrl+Shift+I to open the developer tools window

3. Select the “Console” tab. Paste the JavaScript code from above in the console window and press Enter.

4. The prompt for user input will be displayed in the browser, and output after input will be displayed in the Console

**Output:**

**Enter operator ( +, -, \*, / ):** +

**Enter first number:** 12

**Enter second number:** 24

12+24=36

**Enter operator ( +, -, \*, / ):** \*

**Enter first number:** 5

**Enter second number:** 4

5\*4=20

**Enter operator ( +, -, \*, / ): /**

**Enter first number:** 7

**Enter second number:** 0

7 / 0 = Division by zero error

**WEEK-3**

**1. Write an XML file which will display the Book information which includes the following:**

**1) Title of the Book**

**2) Author Name**

**3) ISBN number**

**4) Publisher name**

**5) Edition**

**6) Price**

<bookstore>

  <book>

    <title>web programming</title>

    <author>chrisbates</author>

    <ISBN>123-456-789</ISBN>

    <publisher>wiley</publisher>

    <edition>3</edition>

    <price>350</price>

  </book>

  <book>

    <title>internet worldwideweb</title>

    <author>ditel&amp;ditel</author>

    <ISBN>123-456-781</ISBN>

    <publisher>person</publisher>

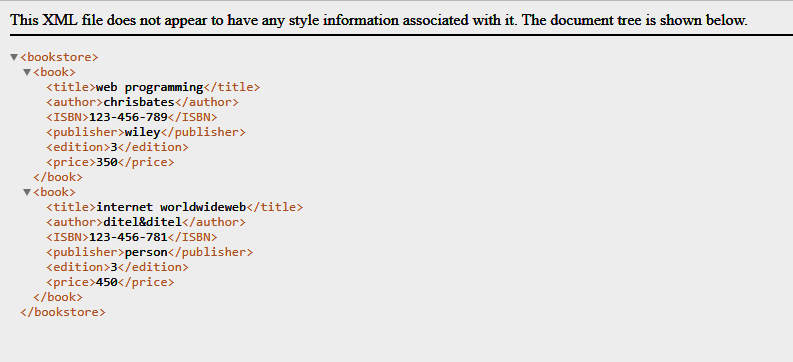
    <edition>3</edition>

    <price>450</price>

  </book>

</bookstore>

**Output:**



**2. Write a Document Type Definition (DTD) or XML Schema Definition (XSD) to validate the above XML file.**

**Program:**

**Bookstore.xml**

<!DOCTYPE bookstore SYSTEM "C:\wt\bookstore.dtd">

<bookstore>

            <book>

                        <title>web programming</title>

                        <author>chrisbates</author>

                        <ISBN>123-456-789</ISBN>

                        <publisher>wiley</publisher>

                        <edition>3</edition>

                        <price>350</price>

            </book>

            <book>

                        <title>internet worldwideweb</title>

                        <author>ditel&amp;ditel</author>

                        <ISBN>123-456-781</ISBN>

                        <publisher>person</publisher>

                        <edition>3</edition>

                        <price>450</price>

            </book>

</bookstore>

**XML document Validation using DTD**

**DTD document (bookstore.dtd)**

<?xml version="1.0" encoding="UTF-8"?>

<!ELEMENT bookstore (book+)>

<!ELEMENT book (title,author,ISBN,publisher,edition,price)>

<!ELEMENT title (#PCDATA)>

<!ELEMENT author (#PCDATA)>

<!ELEMENT ISBN (#PCDATA)>

<!ELEMENT publisher (#PCDATA)>

<!ELEMENT edition (#PCDATA)>

<!ELEMENT price (#PCDATA)>

**Validation using XSD:**

**Bookstore.xml**

<bookstore xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="C:\wt\bookstore.xsd">

            <book>

                        <title>web programming</title>

                        <author>chrisbates</author>

                        <ISBN>123-456-789</ISBN>

                        <publisher>wiley</publisher>

                        <edition>3</edition>

                        <price>350</price>

            </book>

            <book>

                        <title>internet worldwideweb</title>

                        <author>ditel&amp;ditel</author>

                        <ISBN>123-456-781</ISBN>

                        <publisher>person</publisher>

                        <edition>3</edition>

                        <price>450</price>

            </book>

</bookstore>

**XML Schema (bookstore.xsd)**

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">

            <xs:element name="bookstore">

                        <xs:complexType>

                        <xs:sequence>

<xs:element name="book" maxOccurs="unbounded">

<xs:complexType>

            <xs:sequence>

<xs:element name="title"       type="xs:string"></xs:element>

<xs:element name="author"  type="xs:string"></xs:element>

<xs:element name="ISBN"    type="xs:string"></xs:element>

<xs:element name="publisher"          type="xs:string"></xs:element>

<xs:element name="edition"  type="xs:int"></xs:element>

<xs:element name="price"     type="xs:decimal"></xs:element>

            </xs:sequence>

</xs:complexType>

/xs:element>

                        </xs:sequence>

                        </xs:complexType>

            </xs:element>

</xs:schema>

**WEEK 4**

**1. Prepare a JSON file with Student information and display the content in HTML Table format**

[

{

name:'arun',

gender:'Male',

physics:88,

maths:87,

english:78

},

{

name:'rajesh',

gender:'Male',

physics:96,

maths:100,

english:95

},

{

name:'moorthy',

gender:'Male',

physics:89,

maths:90,

english:70

},

{

name:'raja',

gender:'Male',

physics:30,

maths:25,

english:40

},

{

name:'usha',

gender:'Female',

physics:67,

maths:45,

english:78

},

{

name:'priya',

gender:'Female',

physics:56,

maths:46,

english:78

},

{

name:'Sundar',

gender:'Male',

physics:12,

maths:67,

english:67

},

{

name:'Kavitha',

gender:'Female',

physics:78,

maths:71,

english:67

},

{

name:'Dinesh',

gender:'Male',

physics:56,

maths:45,

english:67

},

{

name:'Hema',

gender:'Female',

physics:71,

maths:90,

english:23

},

{

name:'Gowri',

gender:'Female',

physics:100,

maths:100,

english:100

},

{

name:'Ram',

gender:'Male',

physics:78,

maths:55,

english:47

},

{

name:'Murugan',

gender:'Male',

physics:34,

maths:89,

english:78

},

{

name:'Jenifer',

gender:'Female',

physics:67,

maths:88,

english:90

}

]

var table = "<table border=1>";

table += `<tr>

<th>Name</th>

<th colspan="4">Marks</th>

</tr>`;

table += `<tr>

<th></th>

<th>Math</th>

<th>English</th>

<th>Chemistry</th>

<th>Physics</th>

</tr>`;

var tr = "";

for(let i = 0; i < data.result.length; i++) {

tr += "<tr>";

tr += `<td>${data.result[i].name}</td>`;

for (var key in data.result[i].marks) {

tr += `<td>${data.result[i].marks[key]}</td>`;

}

tr += "</tr>"

}

function showTable(){

fetch("./lib/examples/students.json")

.then(response => response.json())

.then(data => createTable(data));

}

function createTable(data) {

var table = "<table border=1>";

table += `<tr>

<th>Name</th>

<th colspan="4">Marks</th>

</tr>`;

table += `<tr>

<th></th>

<th>Math</th>

<th>English</th>

<th>Chemistry</th>

<th>Physics</th>

</tr>`;

var tr = "";

for(let i = 0; i < data.result.length; i++) {

tr += "<tr>";

tr += `<td>${data.result[i].name}</td>`;

for (var key in data.result[i].marks) {

tr += `<td>${data.result[i].marks[key]}</td>`;

}

tr += "</tr>"

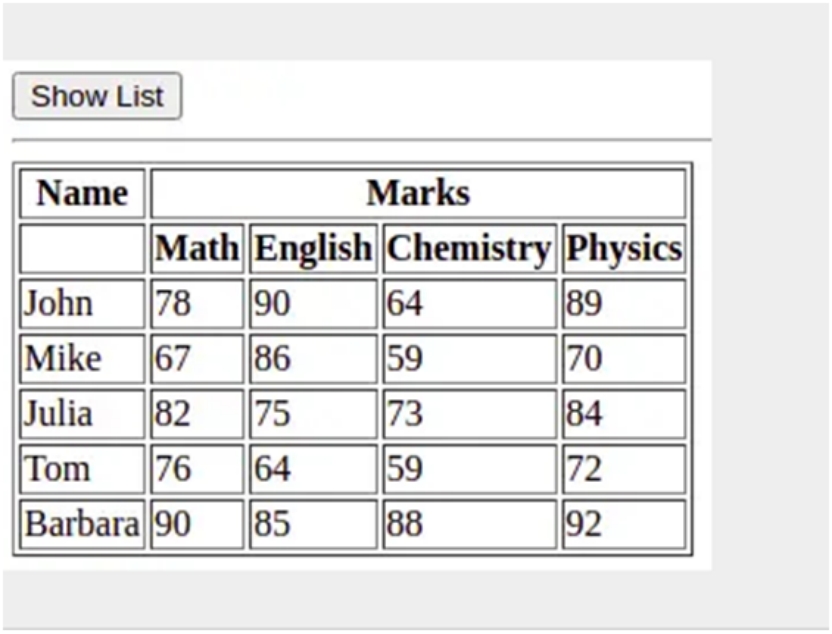
}

table += tr + "</table>";

document.body.innerHTML += table;

}

**Output:**



**WEEK 5**

**1. Write a PHP code to connect the database, creation of the database and the table creation in the database.**

**Database Connection**

<?php

echo "welcome to connect to db";

$servername = "localhost"'

$username = "root";

$password = "";

$conn = mysqli\_connect($servername,$username,$password);

echo "connection was successfull";

?>

**Database Creation**

<?php

$servername = "localhost"'

$username = "root";

$password = "";

$conn = mysqli\_connect($servername,$username,$password);

$sql = "CREATE DATABASE dbsam";

mysqli\_query($conn,$sql);

echo "connection was successfull";

?>

**Database Table Creation**

$servername = "localhost"'

$username = "root";

$password = "";

$database = "dbsam";

$conn = mysqli\_connect($servername,$username,$password,$database);

echo "connection was successfull";

$sql = "create table 'table name' ('sno' int (5), 'name' varchar(12), primary key ('sno'))";

?>

**Output:**



**WEEK 6**

1. **Write a PHP to insert values form HTML to database(registration Page)**

**HTML PAGE**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Add Records Form</title>

</head>

<body>

<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="phone\_no">Phone number:</label>

<input type="text" name="ph\_no" id="ph\_no"> </p>

<p> <label for="emailAddress">Email Address:</label>

<input type="text" name="email" id="emailAddress"> </p>

<p> <label for="gender">Gender:</label>

<input type="text" name="gender" id="genderder"> </p>

<p>

<label for="Address">Address:</label>

<input type="text" name="address" id="Address">

</p>

<p>

<label for="DOB">DOB:</label>

<input type="text" name="dob" id="dob">

</p>

<p>

<label for="Language">Language Known:</label>

<input type="text" name="lang" id="lang">

</p>

<input type="submit" value="Add Records">

</form>

</body>

</html>

**PHP to insert values**

<?php

$link = mysqli\_connect("localhost", "username", "password", "demo");

if($link === false){

die("ERROR: Could not connect. " . mysqli\_connect\_error());

}

$first\_name = mysqli\_real\_escape\_string($link, $\_REQUEST['first\_name']);

$phone\_no = mysqli\_real\_escape\_string($link, $\_REQUEST['phone\_no']);

$email = mysqli\_real\_escape\_string($link, $\_REQUEST['email']);

$gender = mysqli\_real\_escape\_string($link, $\_REQUEST['gender']);

$address = mysqli\_real\_escape\_string($link, $\_REQUEST['address']);

$dob = mysqli\_real\_escape\_string($link, $\_REQUEST['dob']);

$lang = mysqli\_real\_escape\_string($link, $\_REQUEST['lang']);

// attempt insert query execution

$sql = "INSERT INTO registration VALUES ('$first\_name', '$phone\_no', '$email','$gender','$address','$dob','$lang')";

if(mysqli\_query($link, $sql)){

echo "Records added successfully.";

} else{

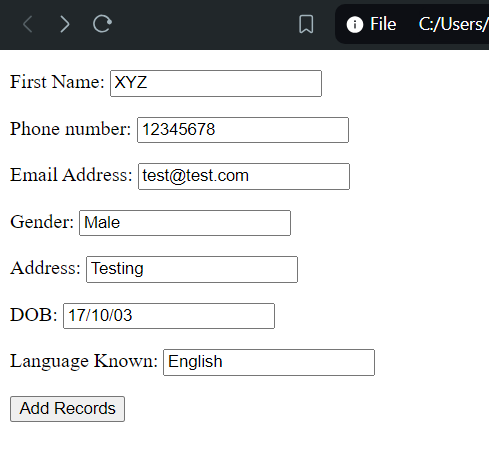
echo "ERROR: Could not able to execute $sql. " . mysqli\_error($link);

}

mysqli\_close($link);

?>

**Output:**



2. **Write a PHP to select values form database table.**

<?php

$link = mysqli\_connect("localhost", "username", "password", "demo");

if($link === false){

die("ERROR: Could not connect. " . mysqli\_connect\_error());

}

$sql = "SELECT \* FROM registration";

if($result = mysqli\_query($link, $sql)){

if(mysqli\_num\_rows($result) > 0){

echo "<table>";echo "<tr>";echo "<th>id</th>";echo "<th>first\_name</th>";

echo "<th>phone\_no</th>";

echo "<th>email</th>";

echo "<th>gender</th>";

echo "<th>ADDRESS</th>";

echo "<th>Dob</th>";

echo "<th>lang</th>";

echo "</tr>";

while($row = mysqli\_fetch\_array($result)){

echo "<tr>";

echo "<td>" . $row['id'] . "</td>";

echo "<td>" . $row['first\_name'] . "</td>";

echo "<td>" . $row['phone\_no'] . "</td>";

echo "<td>" . $row['email'] . "</td>";

echo "<td>" . $row['gender'] . "</td>";

echo "<td>" . $row['address'] . "</td>";

echo "<td>" . $row['dob'] . "</td>";

echo "<td>" . $row['lang'] . "</td>";

echo "</tr>";

}

echo "</table>";

mysqli\_free\_result($result);

} else{

echo "No records matching your query were found.";

}

} else{

echo "ERROR: Could not be able to execute $sql. " . mysqli\_error($link);

}

mysqli\_close($link);

?>

**Output:**

****

3. **Write a PHP to update existing records of a database table.**

<?php

$link = mysqli\_connect("localhost", "username", "password", "demo");

if($link === false){

die("ERROR: Could not connect. " . mysqli\_connect\_error());

}

$sql = "UPDATE registration SET email='xyz@abcdmail.com' WHERE id=1";

if(mysqli\_query($link, $sql)){

echo "Records were updated successfully.";

} else {

echo "ERROR: Could not able to execute $sql. " . mysqli\_error($link);}

mysqli\_close($link);

?>

**Output:**

****

4. **Write a PHP to validate user login**

<html>

<title>login page</title>

<form method="post" action="login.php">

Username:<input type="text" name="username" placeholder="Enter username"><br>

password:<input type="password" name="password" placeholder="Enter the password"><br>

<input type="submit" name="submit">

<input type="reset" name="reset">

</form>

</html>

<?php

$con=mysqli\_connect('localhost','username','password','database');

if(isset\_POSt['submit']){

$name=$\_POST['username'];

$password=$\_POST['password'];

$sql=mysqli\_query("select name,password from table name where name=$name and password=$password");

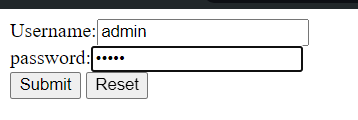
$result=mysqli\_query($con,$sql);

if(mysqli\_num\_rows($result)>0)

{

header("location: home.php"); }else { echo "login denied"; }}?>

**Output:**



**WEEK 7**

**1. Create angular project with name MY\_First\_Angular\_App and write simple Angular program to display the name SNIST.**

**Installation of Angular:**

1. Download and Install Node.js and NPM from Node.js website

2. Verify Node installation with the following commands:

**node -v**

**npm -v**

3. Install Angular CLI globally using the following command:

**npm install -g @angular/cli**

4. Verify Angular installation using the following:

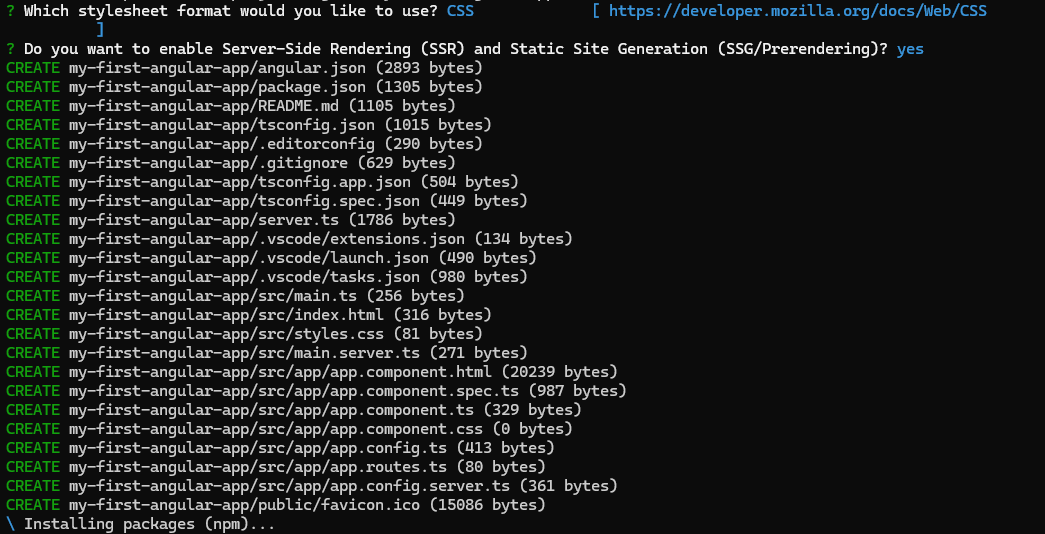
**ng version**

**Creation of Angular project:**

1. Create a new Angular project with the following command:

**ng new my-first-angular-app**

**Output:**

****

2. Navigate to the project directory:

**cd my-first-angular-app**

3. In the “app.component.html” file, add the following code:

<!DOCTYPE html>

<html>

<head>

  <title>AngularJS First App</title>

  <script src="scripts/angular.js"></script>

</head>

<body>

  <div ng-app="myApp">

    <h1>SNIST angular application</h1>

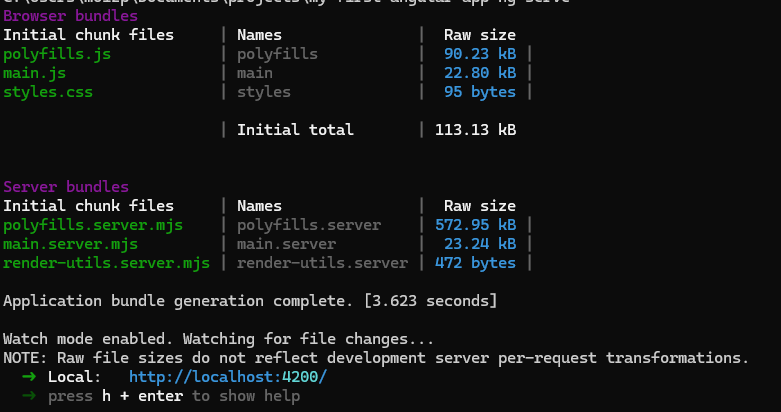
  </div>

</body>

</html>

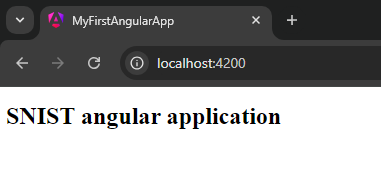
4. Run the Angular development server:

**ng serve**

****

5. Navigate to “localhost:4200” to view the output

**Output:**



**WEEK 8**

**1. Create servlet program to display ‘HELLO WORLD’ message using tomcat server.**

**Steps to execute servlet program:**

1. Open XAMPP Control Panel and start the Apache Tomcat service

2. In the XAMPP Tomcat directory, navigate to the “Webapps” folder

3. Create a new folder to host the servlet application “helloworld”

4. In the new folder, create a “WEB-INF” folder

5. In the “WEB-INF” folder, create a file “web.xml” with the following contents:

**Web.xml**

<web-app>

<servlet>

<servlet-name>

XYZ

</servlet-name>

<servlet-class>

Test

</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>

XYZ

</servlet-name>

<url-pattern>

/demo

</url-pattern>

</servlet-mapping>

</web-app>

6. Create a new folder called “classes” within the “WEB-INF” folder

7. In the classes folder, create a file “Test.java” with the following contents:

**Test.java**

import javax.servlet.\*;

import java.io.\*;

public class Test extends GenericServlet{

public void service(ServletRequest req, ServletResponse res) throws IOException, ServletException{

PrintWriter o = res.getWriter();

o.print("Hello World");

}

}

8. Save the file and compile it using the following command:

**javac -classpath “C:\xampp\tomcat\lib\servlet-api.jar” Test.java**

9. Open the browser and navigate to the url “localhost:8080/helloworld/demo” to view the output.

**Output:**

