

Margin - top / Margin - bottom

Margin - right / Margin - left

Assignment-1

You are tasked with creating a web page for a bike showroom. The showroom wants a clean and interactive design where customers can easily navigate through different bike brands and view the specific models and their specifications. The page should be divided into three frames using a frameset: a top row for displaying the showroom name, a left column for listing bike brands with hyperlinks and a right column for showing the corresponding bike models and specifications when a brand is selected.

Step 1:- Create the 'frameset' structure:

```
<!DOCTYPE html>
<html>
<head>
    <title> Bike showroom </title>
</head>
<frameset rows = "100,*">
    <frame name = "header" src = "header.html" scrolling = "no" noresize>
    <frameset cols = "200,*">
        <frameset name = "brands" src = "brands.html" noresize>
        <frameset name = "Content" src = "welcome.html">
            </frameset>
        </frameset>
    </frameset>
</html>
```

Step 2:- Create the Header frame

```
<!DOCTYPE html>
<html>
<head>
    <style>
        body{
            margin: 0;
            font-family: Arial, sans-serif;
        }
    </style>
</head>
```

gradients / none

border - Break : width

CSS - Margin :

```
background-color: #333;  
color: #fff;  
text-align: center;  
padding: 20px;
```

}

<style>

<head>

<body>

<h1> welcome to [Showroom Name] </h1>

</body>

</html>

Step 3 : Create to Brands frame:-

```
<!DOCTYPE html>
```

<html>

<head>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

padding: 10px;

}

a {

text-decoration: none;

color: #333;

display: block;

padding: 10px 0;

}

a:hover {

background-color: #ddd;

<style>

<head>

<body>

<h2> Bike Brands </h2>

Brand 1

UND - IMAGE

Border - Bottom: width
Border - Top: width

```
<a href = "brand2.html" target = "content">Brand 2</a>  
<a href = "brand3.html" target = "content">Brand 3</a>  
</body>  
</html>
```

Step 4: Create the welcome page.

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
    font-family: Arial, sans-serif;  
    background-color: #ffff33;  
    padding: 20px;  
}  
</style>  
</head>  
<body>  
    <h2>welcome to our bike showroom</h2>  
    <p>Select a brand to view available models and specification.</p>  
</body>  
</html>
```

Step 5: Create the Brand - specific page (e.g. "brand1.html")

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
    font-family: Arial, sans-serif;  
    background-color: #ffff33;  
    padding: 20px;  
}  
</style>  
</head>  
<body>
```

BACKGROUND - HTML

<h2> brand & models </h2>

 model 1' specifications here

 model 2' specifications here

 model 3' specifications here

</body>

</html>

2. You are tasked with designing a job application form for a company's career page. The form should be user-friendly and visually appealing. The form needs to capture the application person information, including their name, highest qualified degree and gender. Additionally, it should have a "submit" button to send the application and a "cancel" button to reset the form or go back to previous page.

CREATE TO HTML STRUCTURE:-

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title> Job Application Form </title>

<style>

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

display: flex;

justify-content: center;

Border - Break : width

align-items: center;
height: 100vh;
margin: 0;

}

.form-container {
background-color: #fff;
padding: 20px;
border-radius: 8px;
box-shadow: 0 0 10px rgba(0,0,0,0.1);
width: 100%;
max-width: 400px;
}

}

.form-container h2 {

text-align: center;
margin-bottom: 20px;
color: #333;

}

.form-group {

margin-bottom: 15px;

}

.form-group label {

display: block;
margin-bottom: 5px;
color: #555;

}

.form-group input

.form-group select {

width: 100%;

padding: 10px;

border: 1px solid #ddd;

border-radius: 4px;

font-size: 16px;

BACKGROUND - IMAGE

```
color: #333;
```

```
}
```

```
.form-group input [type="radio"] {
```

```
width: auto;
```

```
margin-right: 10px;
```

```
}
```

```
.form-group input [type="radio"] + label {
```

```
display: inline-block;
```

```
margin-right: 20px;
```

```
color: #555;
```

```
}
```

```
.form-options {
```

```
display: flex;
```

```
justify-content: space-between;
```

```
}
```

```
.form-action button {
```

```
padding: 10px 20px;
```

```
border: none;
```

```
border-radius: 4px;
```

```
font-size: 16px;
```

```
cursor: pointer;
```

```
}
```

```
.form-options .submit-btn {
```

```
background-color: #28a745;
```

```
color: #fff;
```

```
}
```

```
.form-action .cancel-btn {
```

```
background-color: #dc3545;
```

```
color: #fff;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```

<div class = "form-container">
    <h2>Job Application Form </h2>
    <form action = "submit" method = "Post">
        <div class = "form-groups">
            <label for = "name"> Full Name </label>
            <input type = "text" id = "name" name = "name" required>
        </div>

        <div class = "form-groups">
            <label>Gender</label>
            <input type = "radio" id = "male" name = "gender" value = "male">
            <label for = "male"> male </label>
            <input type = "radio" id = "female" name = "gender" value = "female">
            <label for = "female"> female </label>
            <input type = "radio" id = "other" name = "gender" value = "Other">
            <label for = "other"> Other </label>
        </div>

        <div class = "form-actions">
            <button type = "submit" class = "submit-btn"> Submit </button>
            <button type = "reset" class = "cancel-btn"> Cancel </button>
        </div>
    </form>
</div>

```

3. You have been assigned the task of establishing a development environment for a new web application project. The team is considering whether to utilize the LAMP stack or the WAMP stack. They need a clear understanding of the essential stack.

BACKGROUND - IMAGE

Udo /

Bordon

components of these stacks, their functions and the installation procedures on both Linux and Windows system. Additionally they seek to grasp the role of the Apache web server within the stack and how it integrates with the other components.

Overview of LAMP and WAMP stacks:-

LAMP (Linux, Apache, MySQL (MariaDB), PHP | Perl | Python) and WAMP (Windows, Apache, MySQL (MariaDB), PHP | Perl | Python) are both software stacks used for developing and deploying web applications.

Essential components of LAMP and WAMP stacks:-

1. Linux (Windows):- The operating system that forms the base of the stack.

2. Apache:- The web server that handles requests from clients and serves web content.

3. MySQL (MariaDB):- The relational database management system used to store and manage application data.

4. PHP | Perl | Python:- Server-side scripting languages used to develop dynamic web applications.

functions of each component:-

• Operating system (Linux | Windows):- The platform on which all other components are installed and executed. Linux is open-source widely used in server environments, while Windows is proprietary and commonly used in desktop environments.

• Apache web server:- The core of both stacks, Apache is responsible for handling HTTP requests, serving static content and forwarding dynamic requests to the

appropriate handler.

MySQL/MariaDB: These are database engines to store, retrieve and manage application data. They are queried by the server-side scripts to perform CRUD operations.

PHP/Perl/Python: These are programming languages that run server-side scripts to generate dynamic content.

Installation Procedures:

1. Install Apache

`sudo apt update`

`sudo apt install apache2`

Verify installation!

`sudo systemctl status apache2`

2. Install MySQL/MariaDB

`sudo apt install mysql-server`

Secure the installation!

`sudo mysql_secure_installation`

3. Install PHP

`sudo apt install php libapache2-mod-php php-mysql`

Restart Apache to load PHP module!

`sudo systemctl restart apache2`

4. Test PHP: Create a test PHP file in the web root

```
echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php
```

Access it via

`http://your-server-ip/info.php`

BACKGROUND -

WAMP Installation on windows

1. Download WAMP: download the WAMP installer from the official website.
2. Run the Installer: follow the installation wizard. It will install Apache, MySQL, and PHP automatically.
3. Start WAMP: Launch the WAMP server from the desktop shortcut. You can access the WAMP control panel from the system tray.
4. Test the installation: open a browser and navigate to '<http://localhost/>'. The WAMP home page should appear.

Role of Apache within the stack:-

1. Apache receives the request and determines whether the content is static or dynamic.
2. The PHP script connects to MySQL/MariaDB to perform database operations.
3. The result is then processed by Apache and returned to the user's browser as HTML.

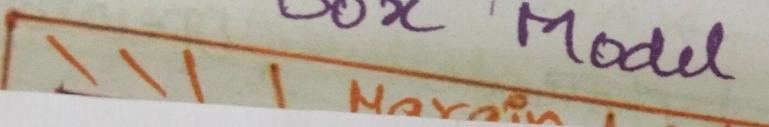
Integration of Apache with other components:-

- Apache and PHP: Apache uses the 'mod_php' module to interpret PHP scripts and generate dynamic content.
- Apache MySQL / MariaDB: Through PHP scripts Apache data, which is then embedded into the web pages served to the client.

Top / Margin - bottom
Margin - right / Margin - left

CSS

Box Model



Conclusion :

- LAMP is ideal for Linux environments and is known for its stability and performance in Server environment.
- WAMP is more suited for windows users, providing a straight forward setup and integration for local.