SAVEETHA SCHOOL OF ENGINEERING
SIMATS, CHENNAI

**Course Code /Title: CSA4399 – Internet Programming**
**Programme : Computer Science and Engineering**

**ASSIGNMENT QUESTIONS**
**SET - 1**

Assignment - 1

| S.No | Questions | Marks | CO | BTL |
|---|---|---|---|---|
| 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. | 10 | CO1 | 3 |
| 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 applicant's personal 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 the previous page. | 10 | CO1 | 3 |
| 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 components of these stacks, their functions, and the installation procedures on both Linux and Windows systems. Additionally, they seek to grasp the role of the Apache web server within the stack and how it integrates with the other components. | 10 | CO1 | 2 |

# Assignment - 1

① you are tasked with creating a webpage for a bike showroom. The showroom wants a clean and interactive design where customers can easily navigate through different brands and view specific models and their specifications. The page should be divided into 3 frames using a frameset a top row for displaying the showroom name, a left column for showing the corresponding bike models and specifications when a brand is selected.

index.html:

```
<!DOCTYPE html>
<html>
<head>
        <title> Bike Showroom </title>
</head>
<frameset rows="15%, 85%">
      <frame src="header.html" name="header
                    frame" noresize scrolling="no">
      <frameset cols="20%, 80%">
            <frame src="brands.html" name=
              "brands" noresize>
                  frame
            <frame src="models.html" name=
              "models frame">
    </frameset>
</frameset>
</html>
```

header.html:

```
<!DOCTYPE html>
<html>
<head>
    <title> Bike showroom </title>
    <style>
        body{
            background colour: Blue;
            colour: white;
            text allign : centre;
            font_family: Arial;
        }
    </style>
</head>
<body>
    <h1> welcome to Reva Showroom </h1>
</body>
</html>
```

brands.html:

```
<!DOCTYPE html>
<html>
<head>
    <title> Bike Brands </title>
    <style>
        body{
            font family: Arial;
            background colour: green;
            colour: white;
        }
        a{
            colour: Pink;
            text decoration: none;
            display: black;
        }
```

```
    </style>
</head>
<body>
    <h2> B
    <a h
    <a h
</body>
</html>
```

models.ht

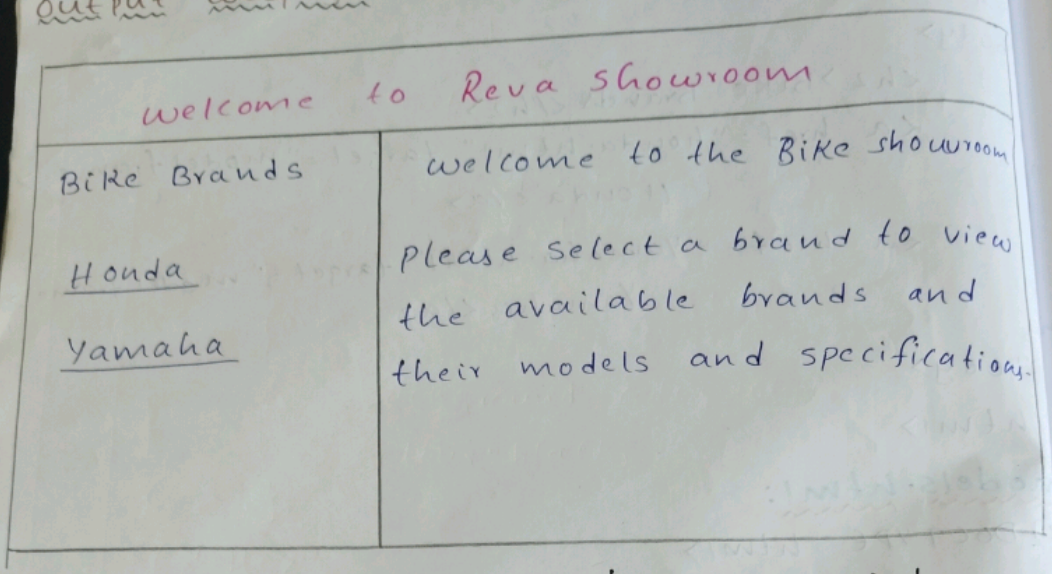
```
<!DOCTYPE
<html>
<head>
    <ti
    <st
    </
</head>
<body>
    <
</body
</htm
```

```
    </style>
</head>
<body>
    <h2>Bike Brands</h2>
    <a href="honda.html" target="model frame">
        Honda</a>
    <a href="yamaha.html" target="model frame">
        Yamaha</a>

</body>
</html>
```

models.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>Bike Models</title>
    <style>
        body{
            background colour: pink;
            font-family: Arial;
        }
    </style>
</head>
<body>
    <h2>welcome to the Bike showroom</h2>
    <p>Please select a brand to view the
     available models and specifications.</p>

</body>
</html>
```

Output webpage:




② You are tasked with designing a job application form for a Company's Carrer Page. The form should be user-friendly and usually appealing. The form needs to capture the applicant's personal 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 to previous Page.

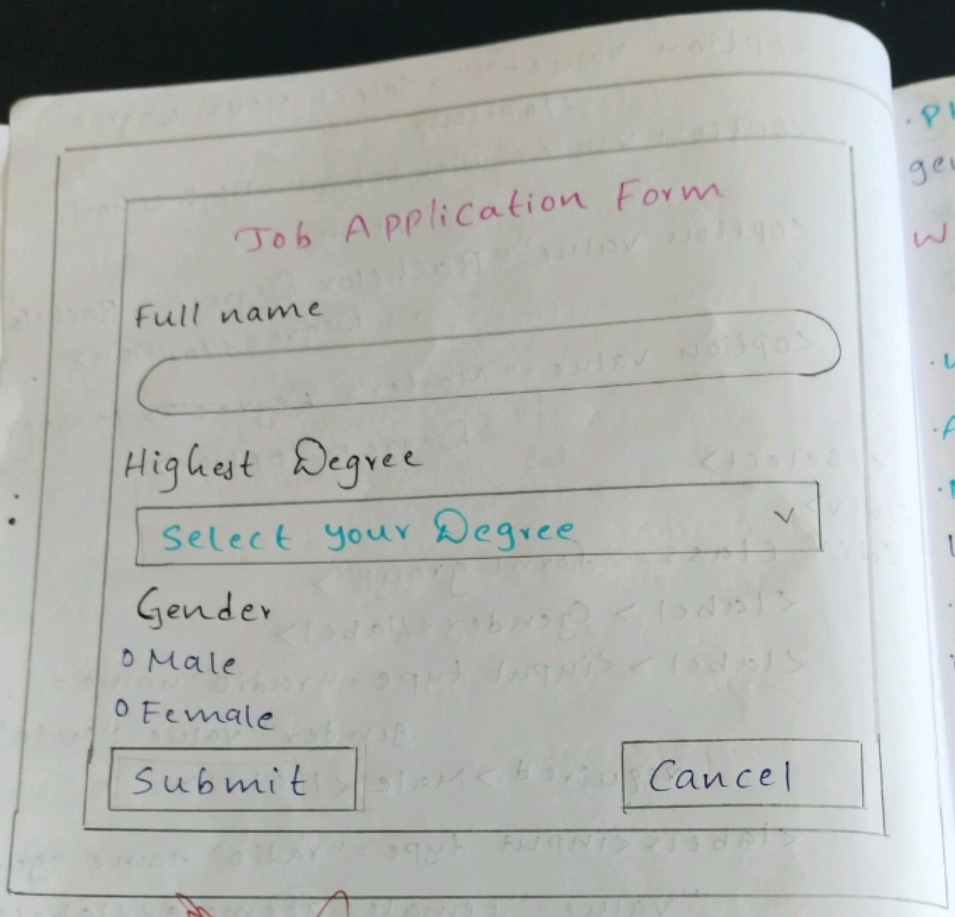
```
<!DOCTYPE html>
<html>
<head>
    <title> Job Application </title>
    <style>
    body {
        font-family = Arial;
        background colour = white;
```

```
    display: flex;
    justify-content: centre;
    align-item: centre;
    height: 100vh;
    margin: 0;
}
.form Container {
        background colour: white;
        padding: 20px
        border-radius: 8px;
        width: 100%
    }
.for Containter h2{
        text align: centre
        colour: black
    }
.form group{
        margin-bottom: 15px;
    }
.form grouplabel{
        display: block;
        margin-bottom: 5px;
        colour: black;
    }
.form group input [type="radio"] {
        margin-right: 10px;
    }
.form group select{
        width: 100%
        Padding: 10px
        border: 1px solid white
        border-radius: 5px;
    }
.form buttons {
```

```
        display: flex;
        justify content: centre;
    }
    .form-buttons button[type="submit"]
    {
        background colour: green;
        colour: white;
    }
    .form buttons button[type="reset", "butt
                                        ]{
        background color: white;
        colour: black;
    }
</style>
</head>
<body>
    <div class="form-container">
        <h2> Job Application Form</h2>
        <form action="/submit.application"
                    method="post">
            <div class="form-group">
                <label for="name">Full name
                    </label>
                <input type="text" id="name"
                    required>
            </div>
            <div class="form-group">
                <label for="degree">Highest Degre
                    </label>
                <select id="degree" name="degre
                    required>
```

```
<option va
<option va
<option
<option
</select>
</div>
<div clas
<lab
<lab
<la
</div>
<divcl
</div>
</form
</div>
</body>
</html>
```

```
<option value="">Select your Degree
</option>
<option value="High school">High school
</option>
<option value="Bachelor Degree">Bachelo-
r's Degree</option>
<option value="Master's Degree">Master
Degree</option>
</select>
</div>
<div class="form-group">
<label>Gender</label>
<label><input type="radio" name=
gender" value="Male"
required>Male</label>
<label><input type="radio" name="gender
" value="Female">Female</label>
</div>
<divclass="form-buttons">
<button type="submit">submit
</button>
<buttontype="reset">Cancel</button>
</div>
</form>
</div>
</body>
</html>
```




③ Compare LAMP vs WAMP stacks. Explain components, functions and installation on Linux/windows. Discuss Apache's role in web Server integration.

LAMP stack (Linux, Apache, MySQL/MariaDB, PHP/Perl):

- Linux: The operating system that provides the foundation for the stack.
- Apache: The web server that handles requests and servers web content.
- MySQL/Maria DB: The database management system for storing and managing data.

- PHP/Perl/Python: Ser
generating dynamic c
WAMP stack (Windo
DB, PHP/Per
- Windows: The opera
- Apache: same role
- MySQL/Maria DB:
like in LAMP.
- PHP/Perl/Python
interaction with
Component Funct
Operating Syst
- provides th
other component
- Handles reso
security and u
Apache web s
- Apache is
dynamic web
- It listens
appropriate r
MySQL/Ma
- These are
systems (RD
tables.
- They han
PHP to

·PHP/Perl/Python: Server-side languages for generating dynamic content.

WAMP stack (Windows, Apache, MySQL/Maria DB, PHP/Perl Python):

·Windows: The operating system for the stack
·Apache: same role as in LAMP, serving web content
·MySQL/Maria DB: Handles database operations, like in LAMP.
·PHP/Perl/Python: For server-site logic and interaction with databases.

Component Functions:

Operating System (LINUX/windows):
- provides the underlying environment for all other components.
- Handles resource managements, file system security and user management.

Apache web server:
- Apache is responsible for serving static and dynamic web pages is Users.
- It listens to incoming requests and sends the appropriate response.

MySQL/Maria DB:
- These are relational database management systems (RDBMS) that store and manage data in tables.
- They handle CRUD operations and work with PHP to retrieve and store data.

PHP/perl/python:
- These scripting languages process server-side code, handle logic and interact with the database.
- PHP generates HTML dynamically from MySQ

Installation steps:

LAMP stack on Linux (ex: Ubuntu):

1. Update system packages
2. Install Apache
3. Install MYSQL/MariaDB
4. Secure installation
5. Install PHP
6. Restart Apache
7. Test PHP

Appache's Role in stack:
- → core web server
- → PHP integration
- → Database Interaction
- → configuration

WAMP stack on windows:

1. Download WAMP
2. Run the installer
3. Test Apache
4. Test PHP
5. Visit 'http://localhost/test.php:"
6. Manage MYSQL: use PHP Admin at 'http://localhost:

Conclusion: Choosing between LAMP and WAMP depends on the team's OS, preferences. Both stack after robust web development environments with Apache acting as a bridge blw web server, server side scripting and database management.

4. Explain H
understand HT
effective web
shooting.
HTTP Reques
structure an
HTTP Reques
1. Request Li
- Method:
- URI/UR
- HTTP ve
2. Headers:
- Provide
- Comm
3. Blank L
- Sepa
4. Body:
- Contai
in a 'P
HTTP R
1. Statu
- HT
- St
- R

(4) Explain HTTP request/response structure. understand HTTP response status codes for effective web application development and trouble-shooting.

Assignment-2

HTTP Request and Response Messages: Structure and Key Concepts:

HTTP Request Message Structures:

1. Request Line:
   - Method: Defines the action.
   - URI/URL: specifies the resources
   - HTTP version: protocol version

2. Headers:
   - provide metadata about the request
   - common headers-
     * Host
     * User-Agent
     * Accept

3. Blank line:
   - separates headers from the body.

4. Body:
   - Contains data sent to the server (eg. form data in a 'POST' requests)

HTTP Response Message Structures:

1. Status line:
   - HTTP version
   - Status code
   - Reason phrase

| Assignment Rubrics | ~~Marks~~ split up | Mark score | Total marks |
|---|---|---|---|
| Assignment 1 | | | |
| Question-1 | form design | 4 | 4 |
| | button design | 4 | 4 |
| | layout look & field | 2 | 2 |
| Question-2 | form design | 4 | 4 |
| | button design | 4 | 4 |
| | layout look & field | 2 | 2 |
| Question-3 | My SQL database | 3 | 3 |
| | PHP script | 3 | 3 |