Rubics / Assignment	Marks Split Up	Marks Score	Total Marks
Assignment-1			
Question-1	Form design	4	
	Button design	4	
	Layout look & field	2	
Question-2	form design	4	
	button design	4	
	layout look & field	2	
Question-3	My SQL database	3	
	PHP Script	3	
	Output	4	
Assignment-2			
Question-4	HTML & PHP code	5	
	Output	5	
Question-5	HTML & Structure	5	
	Output	5	

SAVEETHA SCHOOL OF ENGINEERING
SIMATS, CHENNAI

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

ASSIGNMENT QUESTIONS
SET - 1

S.No	Questions	Marks	C
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	CO
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	CO
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	CO
4	You are developing a web application that involves detailed interactions between clients and servers using HTTP. Your task is to ensure that the application can handle various HTTP request and response messages correctly. Additionally, you need to understand and manage common HTTP response status codes to effectively troubleshoot and handle different scenarios Explain the structure of an HTTP request message and an HTTP response message.	10	CO
5	You are tasked with creating a simple multi-page website for a small business that includes a homepage, an about page, and a contact page. The client wants the site to have a cohesive design with appealing visual elements. You need to apply various CSS styling techniques, including color schemes, typography, and layout adjustments, to enhance the overall look and feel of the site.	10	CO

Assignment - 2

Assignment - 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 bike brands and view the specific models and their specifications. The page should be divided into three frames using a frame set : a top row for displaying the showroom name, a left column for listing bike brands with hyperlinks, and right column for showing the corresponding bike models and specifications when a brand is selected.

HTML Code:

```
<!DOCTYPE html>
<html>
<head>
    <title> Bike Showroom </title>
    <style>
        body {
            font-family: Arial, Sans-serif;
            margin: 0;
            Padding: 0;
        }
        .header {
            background-color: #007bff;
            color: white;
            text-align: center;
            padding: 20px;
        }
        .container {
            display: flex;
            height: 100vh;
        }
        .brand-list {
            width: 20%;
            background-color: #f8f9fa;
            Padding: 0;
            list-style: none;
        }
```

```
.brand-list li{
    padding: 10px;
    border-bottom: 1px solid #ccc;
}
.brand-list li a{
    text-decoration: none;
    color: #000;
    display: block;
    padding: 10px;
}
.brand-list a:hover{
    background-color: #007bff;
    color: white;
}
.bike-details{
    flex:1;
    padding: 20px;
}
</style>
</head>
<body>
    <div class: "header">
    Welcome to Bike Showroom
    </div>
    <div class = "container">
        <ul class = "brand_list">
        <li><a href = "#" on click = "showBike Details
                ('yamaha')">Yamaha</a></li>
        <li><a href = "#" on click = "showBike Details
                ('Honda')"> Honda</a></li>
        <li><a href = "#" on click = " show Bike Details
                ('Ducati')">Ducati</a></li>
        </ul>
    <div class = "bike-details" id = "bike details">
</body>
</html>
```

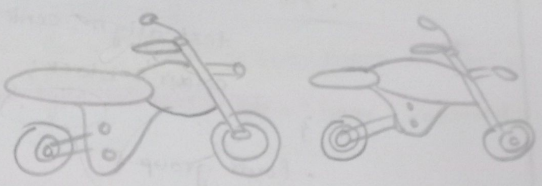
Output:

Welcome to Bike showroom

Yamaha

Honda

Ducati



Yamaha 3.9 | Honda 3.0 | Ducati 3.6

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 & go to previous page.

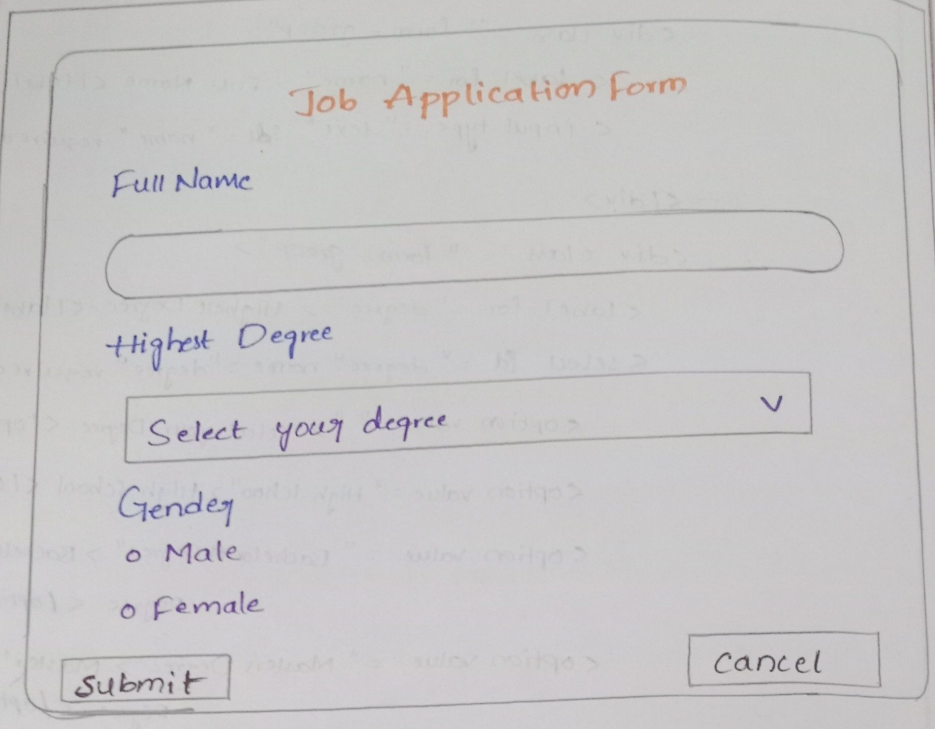
HTML Code:

```
<!DOCTYPE html>
<html>
<head>
    <title> Job Application </title>
    <style>
     body{
          font-family: Aria;
          background-color : white;
          display : flex;
          justify-content : center;
          align-item : center;
          height : 100vh;
          margin : 0;
     }
     .form container{
          background-color: white;
```

```
        padding : 20 px;
        border-radius : 8px;
        width : 100%;
    }
    .for container h2 {
        text align : center;
        colour : black;
    }
    .form group {
        margin bottom : 15px;
    }
    .form group label {
        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 : center;
    }
    .form-button button [type = "submit" ]{
        background colour : green;
        colour : white;
    }
    .form buttons button [type = "reset ", " button" ]{
        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 Degree </label>
                < select id = " degree " name = " degree " required >
                    < option value = " " > Select your Degree </option>
                    < option value = " High School " > High School </option>
                    < option value = " Bachelor Degree " > Bachelor's
                                                Degree </option>
                    < option value = " Masters Degree " > Master's
                                                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>
    < div class = " form-buttons " >
            < button type = " submit " > Submit </button >
            < button type = " reset " > Cancel </button >
    </div>
    </form>
</div>
```

</body>
</html>



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

Overview of LAMP and WAMP Stacks:

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

LINUX: The operating system that provides the foundation for the stack.

Apache: The webserver that handles requests and serves web content.

MySQL / Maria DB: The database management system for storing and managing data.

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, link in LAMP.

PHP / Perl / Python: For server-side logic and interaction.

Component Functions:

Operating System (LINUX / Windows):

- provides the underlying environment for all other components.
- Handles resource management, file system, security and user management.

Apache Web Server:

- Apache is responsible for serving static and dynamic web pages to 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 MySQL.

Installation Steps:

LAMP Stack on LINUX (eg: Ubuntu):

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

WAMP stack on Windows:

1. Download WAMP: WAMP Server.
2. Run the Installer: Follow installation steps and install in the default dicton directory.
3. Test Apache: start WAMP Server.
   open 'http://localhost' in web browser.
4. Test PHP: Create text.PHP in the file.
5. Visit 'http://localhost/test.php:
6. Manage MySQL: Use PhpAdmin at 'http://localhost:'

Apache's Role in the stack:

Core Web Server: Handles incoming HTTP's requests.

PHP integration: Use modules to process PHP scripts.

Database Interaction: Works with PHP scripts that query MySQL, allowing seamless data storage.

Configuration: Apache's configuration files control server behaviour and integration with other servers.

Conclusion:

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

4. You are developing a [illegible] application that involves detailed interaction between clients and servers using HTTP. Your tasks is to ensure that the application can handle various HTTP request and response message correctly. Additionally, you need to understand and manage common HTTP response status codes to effectively troubleshoot and handle different scenarios.

HTTP Request and Response Messages:

Structure and key concepts:

HTTP Request Message Structures:

1. Request Line:

   Method : Defines the action

   URI /URL : Specifies the resource

   HTTP version : protocol version.

2. Headers:

   Provide metadata about the request

   Common headers

   * Host
   * User-Agent
   * Accept

3. Blank Line:

   Seperated 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 : Protocol version

   Status Code : Result of the request

   Reason phrase : Description of the status code

2. Headers:

   Provides metadata about the response

   Common headers:

Content type: formate of response.

Content length: size of response body.

Set-cookie: Information to store cookies.

3. Blank line:

Seperates headers from the body.

4. Body:

Contains the datasent to the client.

Key HTTP Response status code:

1. 1xx (Informational):

'100 continue'

2. 2xx (success):

'200 OK'

'201 created'

3. 3xx (Redirection):

'301 Moved permanently'.

'302 found'.

4. 4xx (client Error):

'400 Bad request'

'401 unauthorized'

'403 forbidden'

'404 Not found'.

5. 5xx (server Error):

'500 Internal server Error'

'502 Bad Gateway'.

'503 Service Unavailable'.

Conclusion:

Understanding the structure of HTTP request and response message and managing HTTP status code is essential for building web applications.

5. You are tasked with creating a simple multi-page for a small business that includes a homepage an about page and a contact page. The client wants the site to have a cohesive design by appealing visual elements. You need to apply various css styling techniques including colour schemes typography, and layout adjustments enhance the overall look and feel of the site.

Homepage (index.html)

```
<!DOCTYPE html>
<html>
<head>
    <title>Small Business</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header>
        <h1>Small Business</h1>
        <nav>
            <a href="index.html">Home</a>
            <a href="about.html">About</a>
            <a href="contact.html">Contact</a>
        </nav>
    </header>
    <div class="container">
        <div class="content">
            <h2>Welcome</h2>
            <p>We are committed to providing exceptional
            service and high quality products</a>
        </div>
    </div>
    <footer>
        <p>&copy; 2024 Small Business. All rights
        reserved</p>
    </footer>
```

</body>
</html>

About page (about.html):

```
<!DOCTYPE html>
<html>
<head>
    <title> Small Business </title>
    <link rel = "style sheet" href = "style.css" >
</head>
<body>
    <header>
        <h1> small Business </h1>
        <nav>
            <a href = "index.html"> Home </a>
            <a href = "about.html"> About </a>
            <a href = "contact.html"> Contact </a>
        </nav>
    </header>
    <div class = "container">
        <div class = "content">
            <h2> About us </h2>
            <p> founded in 2024, our small business has been
            dedicated to serving the community top match
            products and services </p>
        </div>
    </div>
    <footer>
        <p> &copy; 2024 Small Business. All rights are
                                    reserved . </p>
    </footer>
</body>
</html>
```

Contact Page (contact.html):

```
<!DOCTYPE .html>
<html>
<head>
    <title> Contact </title>
    <link rel ="style.sheet" href ="style.css">
</head>
<body>
    <header>
        <h1> Small Business </h1>
        <nav>
            <a href =" index.html" > Home </a>
            <a href =" about.html" > About </a>
            <a href =" contact.html" > Contact </a>
        </nav>
    </header>
        <div class =" container">
            <div class =" content" >
                <h2> Contact us </h2>
                <p> feel free to reach out us: We're
                    here to help! </P>
                <P> email : info@ small business.com</P>
                <P> Phone : (123) 456-7890 </P>
            </div>
        </div>
        <footer>
            <P> &copy; 2024 small Business. All rights are
                reserved. </P>
        </footer>
</body>
</html>
```

Style.CSS:

```
body {
```

```
        font-family : Arial;
        colour : black;
        line-height : 1.6;
}
header {
        padding : 10px;
        text-align : center;
        colour : white;
}
h1, h2 {
        colour : black;
        font-size : 24px;
        margin bottom : 10px;
}
p {
    margin bottom : 15px;
}
footer {
        text-align : center;
        padding : 10px;
        Colour : white;
        Margin top : 20px;
        background colour : black;
}.
```

Small Business

Home About Contact

Welcome to our Business

We are committed to providing exceptional service and high quality products.