

Action 1

ASSIGNMENT-1

SET-1

index.html

```
<!DOCTYPE html>
<html>
<frameset rows="10%, 90%">
    <!-- Top row for showroom name -->
    <frame src="Showroom_name.html" frameborder="0" scrolling="no">
    <frameset cols="20%, 80%">
        <!-- left column for bike brands -->
        <frame src="Bike-brands.html" frameborder="0" scrolling="auto" Name="Brands">
        <!-- right column for bike models and specs -->
        <frame src="Bike-models.html" frameborder="0" scrolling="auto" Name="models">
    </frameset>
</frameset>
</html>
```

Showroom Name.html

```
<!DOCTYPE html>
<html>
<body>
    <h1> Bike Showroom </h1>
    </body>
</html>
```

bike brand .html :-

```
<!DOCTYPE html>
<html>
<body>
<ul>
<li> <a href = "brand.html" target = "models">
    Brand 1, <a></a>
<li> <a href = "brand 2.html" target = "models">
    Brand 2 <a></a>
<ul>
<li>
    <ul>
        <li>
            <ul>
                <li>
                    <ul>
                        <li>
                            <ul>
                                <li>
                                    <ul>
                                        <li>
                                            <ul>
                                                <li>
                                                    <ul>
                                                        <li>
                                                            <ul>
                                                                <li>
                                                                    <ul>
                                                                        <li>
                                                                            <ul>
                                                                                <li>
                                                                                    <ul>
                                                                                        <li>
                                                                                            <ul>
                                                                                                <li>
                                                                                                 &nbsp;

```

Bike models.html

brand1.html

```
<!DOCTYPE html>
<html>
<body>
<h2> Brand 1 models </h2>
<ul>
<li> Model 1: <span> Specs: ... </span> /li>
<li> Model 2: <span> Specs: ... </span> /li>
<li> ... Add more models. </li>
</ul>
<body>
</html>
```

Question 8

HTML code for job application

```
<!DOCTYPE html>
<html>
  <body>
    <form>
      <label> full name: </label>
      <input type = "text" required>
      <label> highest qualified degree! </label>
      <select required>
        <option value = ""> Select Degree </option>
        <option value = "high_school"> Highschool </option>
        <option value = "bachelors"> bachelors </option>
        <option value = "masters"> Masters </option>
        <option value = "phd"> PhD </option>
      </select>
      <label> Gender: </label>
      <select required>
        <option value = ""> Select Gender </option>
        <option value = "male"> male </option>
        <option value = "female"> female </option>
        <option value = "other"> other </option>
      </select>
```

```
<input type = "submit" value = "submit"  
application>  
<input type = "reset" value = "cancel">  
</form>  
</body>  
</html>
```

Question 3

LAMP and WAMP stacks

- LAMP : Linux , Apache , MySQL , PHP

- WAMP : Windows , Apache , MySQL , PHP

Apache web server

- Handles HTTP requests

- Integrates with PHP & MySQL

Installation

- LAMP : Install Apache , MySQL , PHP on Linux

- WAMP : Install WAMP server on windows

Component function

Operating System (Linux / windows)

- Provides the underlying environment for all other components

- Handles resource management , file system
Security and user management

Apache

- Apache

dynam

BHPI PC

For S

daake

Install

1. Upda

2. Inst

3. Inst

4. Se

5. In

6. Re

7. Te

WAM

1. R

2. R

3. T

4. T

5.

6.

Apache webserver

→ Apache is responsible for serving static and dynamic

PHP/SQL python

for server side logic and interaction with databases

Installation steps

1. update system package

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

2. Run The installer

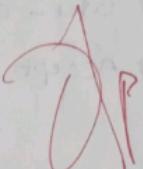
3. Test apache

4. Test PHP

5. Visit `http://localhost/test.php`

6. Manage SQL use PHP admin at

“`http://localhost`”



Question 4

HTTP request and respond messages

1. Request line

- * Method : Define the action
- * URI/FULL : specifies the resource
- * HTTP Version : Protocol version

2. Headers

- * provide metadata about the request
- * common headers
 - * Host
 - * User-Agent
 - * Accept

3. Block line

- * Separates headers from body

4. Body

- * Contains dataset to the server
- HTTP response message structures

1. Status line

- * HTTP Version
- * Status code
- * Reason phrase

2. Headers

- * PROXY
- * Content-Type
- * Content-Length
- * Set-Cookie

3. Blank

- * Separation

4. Body

- * Content

Key

1. 1XX
2. 2XX
3. 3XX
4. 4XX
5. 5XX

error

of

and

error

2. Headers

- * provides metadata about response
- * content type
- * content length
- * set cookie

3. Blank Line

- * separates headers from the body

4. Body

- * contain the data sent to client

Key HTTP response status codes

1. 1xx (Informational)

2. 2xx (Success)

3. 3xx (Redirection)

4. 4xx (Client error)

5. 5xx (Server error)

DPR

Conclusion :- Understanding the structure

of HTTP request and response message

and managing HTTP Status code is
estimated for building web application

Question 5

Home page (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  
product </p>
</div>
</div>
```

< footer>

<p> © copy : 2024, small business. All

right reserved </p>

</ footer>

< /body>

</ html>

4. css About Page (about.html):

<!DOCTYPE html>

<html>

<head>

<title> small business </title>

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

</head>

<body>

<header>

<h1> small business </h1>

<nav>

 Home

 About

 Contact

</nav>

</ header>

<div class="container">

<div class="content">

<h2> about us </h2>

</div>
</div>
<foot>
<p> & copy: 2024 small business </p>

</foot>

</body>

</html>



1.

Output

Welcome to Showroom	
Bike Brand	Welcome to bike Showroom
Honda	Please select a brand to the available brand and their model and specification

5

Job application form

Full name:

Highest degree:

Select your degree

Gender:

- Male
- Female

Submit

cancel

Small business

[Home](#)

[About](#)

[Contact](#)

Welcome to our business

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

© 2024 Small Business All rights reserved

Assignment Rubrics	Marks Spotted	mark score	Total marks
Assignment 1	Form design	4	
	Button design	4	
	Layout look & field	2	
Question 1	Form design	4	
	Button design	4	
	Layout look & field	2	
Question 2	MySQL database	3	
	PHP script	3	
	Output	4	
Assignment 2	HTML & PHP code	5	
	Output	5	
	HTML Structure	5	
Question 4	Output	5	
	Output	5	
Question 5	Output	5	
	Output	5	

Server code

```
import java
```

@websev!

public class

service &
protected

HTTP &

exception &
response

HTTP &

accuse

Integer

att

if

?

Assignment - 3

1. Servlet code

```
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Date;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet ("/Session Tracker")
public class Session Tracker extends HttpServlet
{
    protected void doGet (HttpServletRequest request,
                         HttpServletResponse response) throws ServletException
    {
        response.setContentType ("text/html");
        HttpSession session = request.getSession();
        long time = session.getLastAccessedTime();
        int visitCount = (Integer) session.getAttribute ("visit count");
        if (visitCount == null)
            visitCount = 0;
        visitCount++;
        session.setAttribute ("visit count", visitCount);
        response.getWriter().println("Last Accessed Time : " + time);
        response.getWriter().println("Visit Count : " + visitCount);
    }
}
```

Output

Session tracking example

Session ID: 1234567890CDE

Session Created: Mon Sep 09 12:00:00 IST 2004

Last Accessed: Mon Sep 09 12:01:05 IST 2004

No. of access in this session: 1

2 JSP code using JSTL

```
<%@ taglib uri = "http://java.sun.com/jsp/jstl/core" prefix = "c" %>
<%@ taglib uri = "http://java.sun.com/jsp/jstl/functions" prefix = "fn" %>
```

```
<html>
```

```
<head>
```

```
    <title> Order management </title>
```

```
</head>
```

```
<body>
```

```
    <h2> Order list </h2>
```

```
    <form method = "GET" action = "order.jsp">
```

```
        <label for = "Status" > Filter by Status </label>
```

```
        <select name = "Status" id = "Status">
```

```
            <option value = "All" > All </option>
```

```
            <option value = "Delivered" > Delivered </option>
```

```
            </option> <option value = "Pending" > Pending </option>
```

<| Select >

<input type

</form>

<table>

<thead>

<th> Date

<th> Status

<th> Product

<body>

<c:

<

<tr>

Output

</options>

</select>
<input type="submit" value="filter">

</form>

<table border="1">

<thead>

<tr>

<th> Order ID / th>

<th> Date / th>

<th> Status / th>

<th> Amount / th>

<tbody>

<tr> for each var "Order" items = 7 & so . . .

<td> choose / td>

<td>

<td> when

<td> choose

<td> for choose

<td> choose

</td>

Output

Order ID	Date	Status	Amount
1002	24/10/23	Pending	150
1003	26/10/23	Pending	300

Creating custom functions in JS

```
<taglib> min: "http://java.sun.com/xml/ns/javaee"
<function>
    <function-name>custom</function-name>
    <function-class>http://example.com/custom</function-class>
    <function>
        <function-name>example</function-name>
        <function-class>com.example.custom.functions.Function</function-class>
    </function>
</functions>
```

Outputs

Custom function example

Original : hello world

Reversed : dlrow olleH

Java Script code snippet

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title> Stock Market Quotes </title>
```

```
<Script>
    function refreshPage() {
        location.reload();
    }

    // Set Timeout (in seconds)
    const confirmRefresh = confirm("There will
        refresh in 30 seconds");

    if (!confirmRefresh) {
        refreshPage();
    } else {
        alert("Page refresh cancelled");
    }
}

<Script>
```

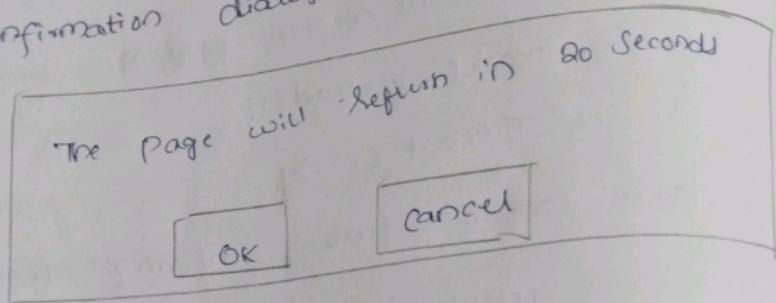
```
<body>
    <h1> Stock market quotes </h1>
</body>
</html>
```

Output

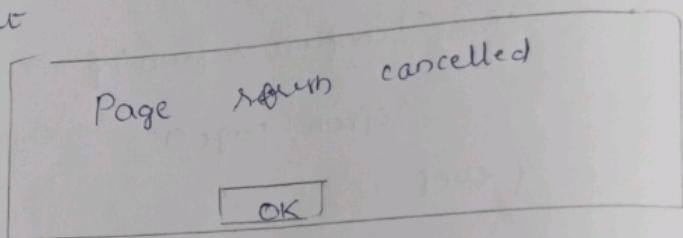
Page display

Stock Market Quotes	
- Apple Inc (AAPL)	\$150.00
- Microsoft Corp (MSFT)	\$250.00
- Alphabet Inc (GOOGL)	\$580.00

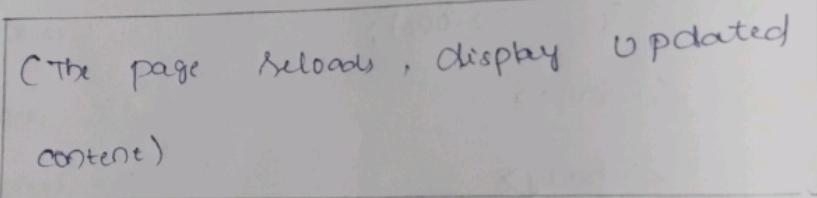
Confirmation dialogue



Alert



Page refresh



Stock market quotes

3.

- Apple Inc (AAPL) : \$150.00
- Microsoft corp. (MSFT) : \$250.00
- Alphabet Inc. (Google) : \$2850.00

1. Generated
import

example

2. Integrate

- Include
- Config

3. Invoke

- Create
- Payment

Invoke

Payment

4. Handle

* Check
if (

? u

* EXC

try {

P

4.

1. Generated client code from WSDL

wsimport -keep-s gae-d bin-p com.exam
payment -verbose http://
example.com/payment/gateway

2. Integrate generated code into application
 - Include generated code
 - Configure service endpoint

3. Invoke the payment service

- Create service instance

PaymentServiceService : new Payment

Invoke Method

PaymentResponse = port . processPayment

4. Handle response and errors

* Check response

if (!response.isSuccess())

} else {

}

* Exception handling

try {

PaymentResponse

Output

Successful payments

Payment transaction ID: 198765321

Payment failure

Payment failed