SAVEETHA SCHOOL OF ENGINEERING
SIMATS, CHENNAI

Engineer

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

ASSIGNMENT 3 QUESTIONS

SET -5

S.No	Questions	Marks	CO	B
1	Implementing a feature in a web application that tracks the number of accesses by a client within a single session. You need to use Java Servlets to manage and monitor session data. The application should count how many times the client accesses the application during their session and retrieve information about the session, such as the session ID, creation time, and last accessed time.	20	CO4	
2	Write a scenario where you had to use JSTL to solve a complex problem and how you went about it. Also, elaborate the function library in JSTL and how to create custom functions.	20	CO4	
3	A page of stock market quotes uses script to refresh the page every five minutes in order to ensure the latest statistics remain available. 20 seconds before the five minute period expires, a confirm dialog appears asking if the user needs more time before the page refreshes. This allows the user to be aware of the impending refresh and to avoid it if desired.	20	CO5	
4	You are developing an e-commerce application that needs to integrate with an external payment gateway service. This service is described using a WSDL file. How would you use the WSDL file to integrate the payment gateway service into your e-commerce application? Describe the steps involved in generating the client code, invoking the service, and handling any potential errors	20	CO6	

Assignment -3

1) Implementing a feature in a web application that tracks number of accesses by a client within a session using Java.

servlet code:-

```
import java.io.IOException;
import java.io.print write;
import java.util.Date;
import javax. servlet. servlet exception;
import javax. servlet. http. Http servlet;
import javax. servlet. http. Http session;
@ web servlet ("/session Tracker").
        Http servlet Response response throws servlet
                exception IO exception{
    response. set content Type ("text/html");
    Print writer out = response. get writer();
    String session Id = session. get (Id ();
    long last Access Time = session. get last.
        Accessed Time();
Integer visit count = (Integer) session. get
            Attribute ("visit count");

    if (visit count == null) {
            visit count = 0;
  }
  visit count ++;
out. println ("<html><body>");
out. println ("<h1> session Tracking example</h
out.println("<P> session ID:" + sessionId + "</P>");
out.println("<P> No. of accesses in this session:"
                                    visit count "</P>");
  out.println("</body></html>");
}
}
```

Output:-

```
Session ID: 12345ABCDE
Session Created: Mon Sep 09 12:00:00 IST 2024
Last Accessed: Mon Sep 09 12:01:05 IST 2024
No. of access in this session: 1
```

2) write a scenario where you had to use JSTL to solve a complex problem and how you went about it.

JSP code using JSTL

```
<%@ taglib uri="http://java.sum.com/jsp/jst// code"
                          prefix="c" %>
<%@ taglib uri="http://java.sam.com/jsplust/
                functions" prefix="fn" %>
<html>
<head>
      <title> order Managment </title>
<head>
<body>
```

```
<h2> order List</h2>
<form method ="GET" action = "ordor..jsp">
    <select name = "status" id = "status">
        <option value = "All" > All </option>
        <option value="Delievered" > Delivered
                                         </option>
    </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>
    </tr>
</thread>
<tbody>

    <c: for each var = order items =${orders}

    <c:choose>

        <tr>

            <td> $ {order.id} </td>
            <td> $ {order. date} </td>
            <td> $ {order. amount} </td>
```

```
              </tr>
               </c:when>
             </c:choose>
           </c: for choose>
          <c: for each>
        </tbody>
      </table>
  </body>
</html>
```

O/P :-

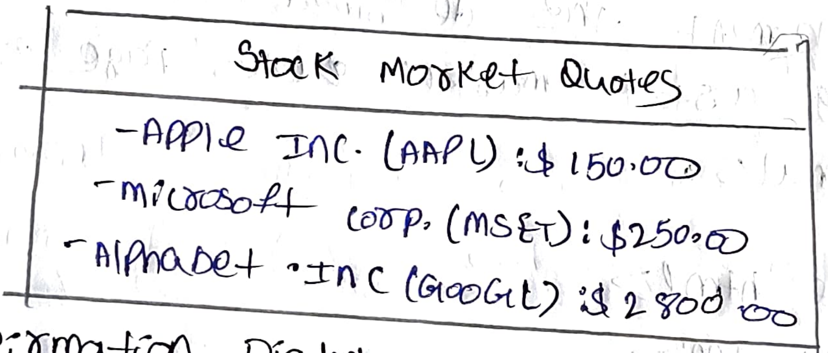
order ID	Date	Status	Amount
1002	2024-09-08	pending	160.00
1003	2024-09-09	pending	300.00

3) To implement the described functionality for refreshing a stock market quotes page ever five minutes.

```
<!DOCTYPE html>
<html lang ="en">
<head>
      <meta charset ="UTF-8">
      <title> Stock market Quotes </title>
      <scrip>
           function refres Page(){
                       location.relod();
           }
```

```
Set Time out () =>{
    if(!confirm Refresh){
            refresh Page();
    }else{
        a lert("Page refresh canceled");
    }
}
280000);
</script>
<body>
    <h1> Stock Market Quotes </h1>
</body>
</html>
```

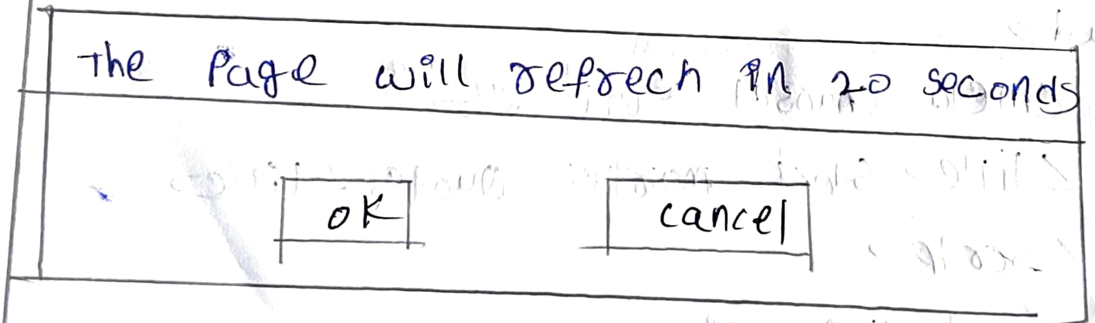
O/P:-



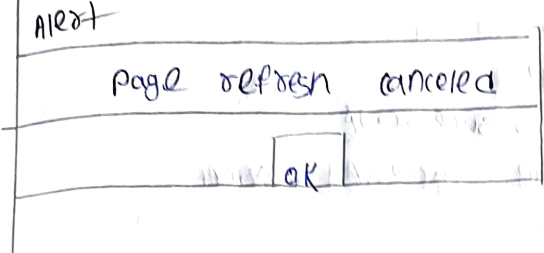
Stock Market Quotes

- Apple Inc. (AAPL) :$ 150.00
- Microsoft Corp. (MSFT): $250.00
- Alphabet .Inc (GOOGL) :$ 2800.00

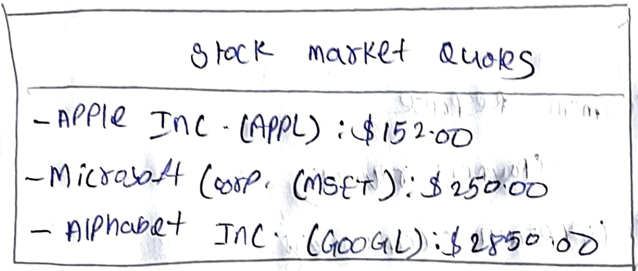
Confirmation Dialog



The Page will refrech in 20 seconds



Page Refresh



- Apple Inc. (APPL) : $152.00
- Microsoft Corp. (MSFT) : $250.00
- Alphabet Inc. (GOOGL) : $2850.00

4) To integrate an external payment gate way service into your e-commerce application using a WSDL file.

i) Generate cliend code from WSDL

```
wsimport -keep -s sre -d bin -p com.example
                         payment -verbose http://
                    example.com/paymentgateway
```

ii) Integrate generates code into Application.

- Include Generated code
- Configure service endpoint

iii) Invoke Payment service

```
Paymentservice service = new Payment service();
Payment Response = Port.Process Payment (Payment
```

4) Handle Response and errors

```
check Response
if (response.is sucess()){
        // Handle successful paymant
}else{
      // Handle payment failure
}
try{
    payment Response
            (payment request);
}catch (SOAPFault exception e){
} catch (web service ){
}
```

O/P:-

Payment sucessful Tranaction ID: 19875326

Payment fail (eg:- Invalid credit card details.

Payment failed due to a SOAP fault: Invalid Request format