Experiment No. 1.2

Student Name: Rishav Kumar **UID: 22MCC20039**
Branch: MCA - CCD **Section/Group: MCD-1/ Grp A**
Semester: II **Date of Performance: 28th Feb 23**
Subject Name: Advanced Internet Programming Lab **Subject Code: 22CAP-686**

1. Aim/Overview of the practical:

Create a servlet that describes how to use the HttpSession object to find out the creation time and the last-accessed time for a session. We would associate a new session with the request one does not already exist

2. Code for practical:

- Create a new Java project in your preferred IDE.
- Create a new servlet class by right-clicking on the project folder, selecting New -> Servlet.
- In the servlet class, import the following packages:

```
import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.io.IOException;

import java.io.PrintWriter;
```

- Add the @WebServlet annotation to the servlet class to define the servlet URL mapping:

```
@WebServlet("/session-info")

public class SessionInfoServlet extends HttpServlet {

  // servlet code goes here

}
```

- Override the doGet() method to handle GET requests:

```
@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {

    // servlet code goes here

}
```

- Inside the doGet() method, retrieve the HttpSession object from the HttpServletRequest:

```
HttpSession session = request.getSession();
```

- Check if the session is new by calling the isNew() method on the HttpSession object. If the session is new, set a session attribute to store the creation time:

```
if (session.isNew()) {

    session.setAttribute("creationTime", System.currentTimeMillis());

}
```

- Get the last-accessed time of the session by calling the getLastAccessedTime() method on the HttpSession object:

```
long lastAccessTime = session.getLastAccessedTime();
```

- Set a session attribute to store the last-accessed time:

```
session.setAttribute("lastAccessTime", lastAccessTime);
```

- Create a PrintWriter object to write the response:

```
response.setContentType("text/html");

PrintWriter out = response.getWriter();
```

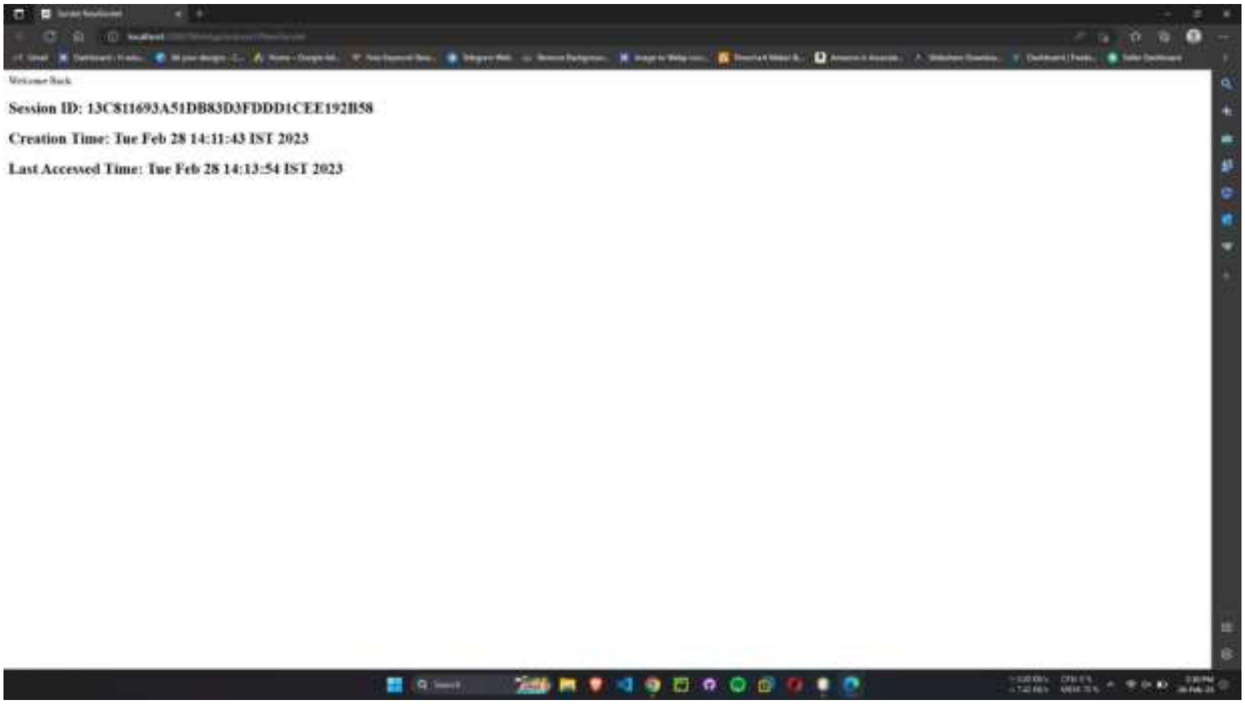
- Write the session information to the response:

```
out.println("<html><body>");

out.println("<h1>Session Information:</h1>");

out.println("<ul>");

out.println("<li>Session ID: " + session.getId() + "</li>");

out.println("<li>Creation Time: " + session.getAttribute("creationTime") + "</li>");

out.println("<li>Last Accessed Time: " + session.getAttribute("lastAccessTime") + "</li>");

out.println("</ul>");

out.println("</body></html>");
```

- Close the PrintWriter object:

```
out.close();
```

- Save the servlet class and deploy the application to a server (e.g. Apache Tomcat).
- Open a web browser and navigate to the URL mapped to the servlet (e.g. http://localhost:8080/myapp/session-info).



```
Welcome Back
Session ID: 13C811693A51DB83D3FDDD1CEE192B58
Creation Time: Tue Feb 28 14:11:43 IST 2023
Last Accessed Time: Tue Feb 28 14:13:54 IST 2023
```

- The servlet should create a new session (if one does not already exist) and display the session information in the browser.

** **THE END** ***