|  |  |
| --- | --- |
|  | **BAHRIA UNIVERSITY,**  **(Karachi Campus)**  *Department of Software Engineering*  **ASSIGNMENT#04 – Fall 2022**  **Problem Based Learning** |

Course Title: **Web Engineering** Course Code: **SEN-310**

Class: **BSE-5(A,B)** Shift: **Morning**

Course Instructor: **Engr. M. Adnan Ur Rehman** Time Allowed:  **01 Week.**

Date: **09-Jan-2023** Max. Marks: **20 Points**

**Instructions:**

* This assignment is based on Complex Engineering Problem.
* Submit this assignment in a group consist up to 3 members. Submission is permitted on ***LMS***.
* **All** members of the group **MUST** upload the same solution in LMS.
* You have to submit a compressed file containing the following:
  1. ***This title page file*** **.docx** after filling the group names and output of your program
  2. ***Source files*** May include **.html** and **.js** files

|  |  |  |
| --- | --- | --- |
| **GROUP MEMBERS** | | |
| **S.No.** | **Student Name** | **Enrolment#** |
|  |  |  |
|  |  |  |
|  |  |  |

**Task:** Create a web application having three pages with the following properties. ***[CLO-3; 20 Points]***

1. **Page1.html:**

* This page must contain a button by which value of a shared integer counter can be increased.
* Value of the integer counter must be shared among all the users of your web application i.e. if any user increase the counter, the counter value will be increased for all users.

1. **Page2.html:**

* This page must contain a button by user can increase the value of his/her integer counter.
* Value of the integer counter must be private for all the users i.e. increment of one user’s counter must not affect another.

1. **Page3.html:**

* Initially on first visit, page must show a time on which it will be expired.
* This page must be expired after specific time (after 5 seconds) if user revisited.
* User must be displayed “page expired” message.

**Code:**

import express from 'express'

import cookieparser from "cookie-parser";

import cors from "cors";

const app = express()

app.use(cookieparser())

app.use(cors());

let shared\_counter = 0;

let counters = {};

app.get('/page1', function (req, res) {

    shared\_counter++

    return res.send(

      `

      <form method="get">

      <p>Shared\_Value :${shared\_counter}</p>

      <button type="submit">Update Counter</button>

      </form>

      `

    )

});

app.get('/page2', function (req, res) {

  let ID = req?.cookies?.ID || Math.floor(Math.random()\*999).toString()

  console.log("User\_ID",ID)

  let Private\_counter = counters[ID] || 0

  console.log(counters)

  counters[ID] = (parseInt(Private\_counter) + 1).toString()

  res.cookie('ID', ID)

  ;

  return res.send(

    `

    <form method="get">

    <p>Private\_Value :${Private\_counter}</p>

    <button type="submit">Update Value</button>

    </form>

    `

  )

});

app.get('/page3', (req, res) => {

    var expire = req.cookies.ExpiredPage;

    if (!expire || expire < new Date()) {

        var date = new Date();

        date.setSeconds(date.getSeconds() + 5);

        res.cookie('ExpiredPage', date);

        res.send(`<p>Page will expire after ${ date }</p>`);

    } else {

        res.send("Page Expired...!!!");

    }

  });

app.listen(8000, function () {

    console.log("Listening on port 8000 ...");

});

**Output**:

Graphical user interface, text, application, chat or text message

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

