Name: Muskan Kumari **Sap ID: 1000016871**

Instructions

- There are **2** questions in this assignment.
- Email/paper/other modes of submissions will not be accepted.
- Upload a word **version** of this document.
- Submit the assignment by the due date and time.

Due Date: 22/10/24, 12pm

Submitting this Assignment

You will submit (upload) this assignment in MS Teams. Name this document as NGLA1_AJPODD2024_John_Doe.doc in case your name is John Doe, and this non-graded lab assignment is no. 1 of course whose acronym is AJP, and offered in ODD 2024. Paste your code after each question, paste the screenshot of output, save and upload the document.

Grading Scheme: This assignment has 0 Marks. However, students must submit the complete assignment by the due date and time.

Question 1:

Create a student table and insert new students details and delete old student details.

```
CREATE TABLE Student (
   StudentID INT PRIMARY KEY,
   FirstName VARCHAR(50),
   LastName VARCHAR(50),
   DateOfBirth DATE,
   EnrollmentDate DATE
);
INSERT INTO Student (StudentID, FirstName, LastName, DateOfBirth, EnrollmentDate)
VALUES
   (1, 'John', 'Doe', '2000-01-15', '2021-09-11'),
   (2, 'Jane', 'Smith', '1999-05-23', '2021-10-12'),
   (3, 'Emily', 'Jones', '2001-11-11', '2021-11-13');


SELECT*
FROM Student;
```

StudentID	FirstName	LastName	DateOfBirth	EnrollmentDate
1	John	Doe	2000-01-15	2021-09-11
2	Jane	Smith	1999-05-23	2021-10-12
3	Emily	Jones	2001-11-11	2021-11-13

Delete old student details.

```
DELETE FROM Student
WHERE EnrollmentDate < '2021-01-01';
```

StudentID	FirstName	LastName	DateOfBirth	EnrollmentDate
2	Jane	Smith	1999-05-23	2021-10-12
3	Emily	Jones	2001-11-11	2021-11-13

Question 2:
Create database with User Information and Item information. The Item catalog should be dynamically loaded from the database.

Step 1: Create the Database and Tables

```
-- Create the database
CREATE DATABASE ShopDB;

-- Use the database
USE ShopDB;

-- Create the Users table
CREATE TABLE Users (
    UserID INT PRIMARY KEY,
    UserName VARCHAR(50),
    Email VARCHAR(100)
);

-- Create the Items table
CREATE TABLE Items (
    ItemID INT PRIMARY KEY,
    ItemName VARCHAR(100),
    ItemDescription TEXT,
    Price DECIMAL(10, 2)
);
```

Step 2: Insert Data

```
-- Insert sample users
INSERT INTO Users (UserID, UserName, Email)
VALUES
(1, 'Alice', 'alice@example.com'),
(2, 'Bob', 'bob@example.com');
```

-- Insert sample items
INSERT INTO Items (ItemID, ItemName, ItemDescription, Price)
VALUES
(1, 'Laptop', 'A high-performance laptop', 999.99),
(2, 'Smartphone', 'A latest model smartphone', 699.99),
(3, 'Headphones', 'Noise-cancelling headphones', 199.99);

Step 1: Set Up the Server
Using Node.js and Express to serve the item data. If you don't have Node.js installed, you can download it from nodejs.org.

JavaScript

```
const express = require('express');
const sqlite3 = require('sqlite3').verbose();
const app = express();
const port = 3000;

// Connect to the database
let db = new sqlite3.Database('./ShopDB.db');

// Serve static files
app.use(express.static('public'));

// Endpoint to get items
app.get('/items', (req, res) => {
    db.all("SELECT ItemID, ItemName, ItemDescription, Price FROM Items", [], (err, rows) =>
{
        if (err) {
            throw err;
        }
        res.json(rows);
    });
});

// Start the server
app.listen(port, () => {
    console.log(`Server running at http://localhost:${port}/`);
});
```

Step 2: Create the HTML and JavaScript

1. Create a directory named public in your project directory.

2. Inside the public directory, create an index.html file and add the following code:

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Item Catalog</title>
</head>
<body>
    <h1>Item Catalog</h1>
    <div id="itemCatalog"></div>

    <script>
        // Function to load items from the server
        async function loadItems() {
            const response = await fetch('/items');
            const items = await response.json();
            const itemCatalog = document.getElementById('itemCatalog');

            items.forEach(item => {
                const itemDiv = document.createElement('div');
                itemDiv.innerHTML = `
                    <h2>${item.ItemName}</h2>
                    <p>${item.ItemDescription}</p>
                    <p>Price: $${item.Price.toFixed(2)}</p>
                `;
                itemCatalog.appendChild(itemDiv);
            });
        }

        // Load items when the page loads
        window.onload = loadItems;
    </script>
</body>
</html>
```

OUTPUT;

Name: Muskan Kumari **Sap ID: 1000016871**

```
Item Catalog:
ID: 1, Name: Laptop, Description: A high-performance laptop, Price: $999.99
ID: 2, Name: Smartphone, Description: A latest model smartphone, Price: $699.99
ID: 3, Name: Headphones, Description: Noise-cancelling headphones, Price: $199.99
```