Assignment 1

b. Write a JavaScript Program to get the user registration data and push to array/local storage with AJAX POST method and data list in new page.

Registration.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Latest compiled and minified CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

    <!-- jQuery library -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>

    <!-- Latest compiled JavaScript -->
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
    <link rel="stylesheet" href="style.css">
    <title>Registration Form</title>
</head>
<body>
    <div class="container">
        <h2 class="text-center">Registration Form</h2>
        <form>
            <div class="form-group">
                <label for="name">Name</label>
                <input type="text" class="form-control" id="name" placeholder="Enter your name">
            </div>

            <div class="form-group">
                <label for="email">Email ID</label>
                <input type="email" class="form-control" id="email" placeholder="Enter your email">
            </div>

            <div class="form-group">
                <label for="password">Password</label>
                <input type="password" class="form-control" id="password" placeholder="Enter your
password" >
            </div>
            <button type="submit" class="btn btn-primary btn-block"
onclick="Register()">Register</button>
        </form>
    </div>
</body>
```

```html
</html>
<script>
    function Register() {
        let name = document.getElementById("name").value;
        let email = document.getElementById("email").value;
        let password = document.getElementById("password").value;

        let users = JSON.parse(localStorage.getItem("user")) || [];

        if (users.some(user => user.email === email)) {
            alert("Duplicate data found! Email already registered.");
        } else {
            users.push({
                "name": name,
                "email": email,
                "password": password
            });

            // Store the updated users array back to localStorage
            localStorage.setItem("user", JSON.stringify(users));

            alert("Registration successful!");
        }

        console.log("Data saved to localStorage:");
        console.log(users);
    }
//.some method=it can check atleast one element in the array
//localStorage.setItem("name", name);
//localStorage.setItem("email", email);
//localStorage.setItem("password", password);
</script>
```

Styles.css

```css
body {
    background-color: #f0f8ff;
    font-family: 'Arial', sans-serif;
}

.container {
    max-width: 600px;
    margin-top: 50px;
    padding: 30px;
    background-color: #ffffff;
    border-radius: 8px;
    box-shadow: 0 4px 15px rgba(0, 0, 0, 0.1);
```

```css
    }

    h2 {
        color: #333;
        font-size: 2em;
        margin-bottom: 30px;
        font-weight: bold;
    }

    .form-group label {
        color: #555;
        font-size: 1.1em;
    }

    .form-control {
        border-radius: 4px;
        border: 1px solid #ccc;
        padding: 12px;
        margin-bottom: 20px;
    }

    .btn-primary {
        background-color: #5cb85c;
        border-color: #4cae4c;
        font-size: 1.1em;
        text-transform: uppercase;
        border-radius: 4px;
        transition: all 0.3s ease;
    }

    .btn-primary:hover {
        background-color: #4cae4c;
        border-color: #398439;
    }

    .btn-block {
        width: 100%;
    }
    @media screen and (max-width: 768px) {
        .container {
            margin-top: 20px;
            padding: 20px;
        }
        h2 {
            font-size: 1.5em;
        }
    }
```

Output:

Registration Form

Name

Sonali Patil

Email ID

sonali@gmail.com

Password

•••••••••

REGISTER

Elements   Console   Sources   Network   Performance   Memory   **Application**   Security

Application
  Manifest
  Service workers
  Storage

Storage
  ▼ Local storage
      file://
  ▶ Session storage
  Extension storage
  IndexedDB
  ▶ Cookies
  Private state tokens
  Interest groups
  ▶ Shared storage
  Cache storage
  Storage buckets

Background services
  Back/forward cache
  Background fetch
  Background sync
  Bounce tracking mitigati...
  Notifications
  Payment handler
  Periodic background sync
  ▶ Speculative loads
  Push messaging
  Reporting API

Frames
  ▶ top

file://

Origin file://

Key	Value
user	[{"name":"Sonali Patil","email":"sonali@gmail.com","...

Select a value to preview