|---> Index.html

| ---> Admin.html

|---> User.html

| ---> User-result.html

| ---> Result.html

| ---> Script.js

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Online Voting System</title>

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

    <style>

        body {

            background-image: url('voting\_img1.jpeg');

            background-size: cover;

            background-position: center 5%;

            background-repeat: no-repeat;

        }

        #loginForm,

        #registerForm,

        #adminLoginForm {

            display: none;

            margin-top: 20px;

            width: 400px;

            background-color: rgba(255, 255, 255, 0.8);

            padding: 20px;

            border-radius: 10px;

            opacity: 0;

            transform: translateY(20px);

            transition: all 0.5s ease;

        }

        .form-container {

            display: flex;

            justify-content: flex-start;

            align-items: center;

            height: 100vh;

            margin-top: -50px;

            margin-left: 30px;

        }

        .form-container.active {

            display: block;

            opacity: 1;

            transform: translateY(0);

        }

        /\* Add sliding animation \*/

        .slide-in {

            animation: slideIn 0.8s forwards;

        }

        @keyframes slideIn {

            from {

                transform: translateY(30px);

                opacity: 0;

            }

            to {

                transform: translateY(0);

                opacity: 1;

            }

        }

        /\* Navigation bar color \*/

        .navbar {

            background-color: rgb(23, 49, 68);

            height: 80px;

            padding: 20px;

        }

        .navbar-brand {

            font-size: 1.8rem;

            color: rgb(255, 255, 255);

        }

        .navbar-nav .nav-link {

            font-size: 1.2rem;

            color: white;

        }

        .navbar-nav .nav-link:hover {

            color: #d3d3d3;

        }

        .form-group {

            margin-bottom: 15px;

        }

        .form-footer {

            margin-top: 15px;

            font-size: 0.9rem;

            text-align: center;

        }

        .form-footer a {

            color: #007bff;

            text-decoration: none;

        }

        .form-footer a:hover {

            text-decoration: underline;

        }

        .btn-center {

            display: block;

            margin: 0 auto;

        }

    </style>

</head>

<body>

    <nav class="navbar navbar-expand-lg">

        <a class="navbar-brand" href="#">Online Voting System</a>

        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

            <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse" id="navbarNav">

            <ul class="navbar-nav ml-auto">

                <li class="nav-item">

                    <a class="nav-link" href="#" onclick="showLoginForm()">Login</a>

                </li>

                <li class="nav-item">

                    <a class="nav-link" href="#" onclick="showRegisterForm()">Register</a>

                </li>

                <li class="nav-item">

                    <a class="nav-link" href="#" onclick="showAdminLogin()">Admin</a>

                </li>

                <li class="nav-item">

                    <form class="form-inline" id="searchForm">

                        <input class="form-control mr-2" type="search" placeholder="Search" id="searchInput" aria-label="Search">

                        <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>

                    </form>

                </li>

            </ul>

        </div>

    </nav>

    <div class="form-container">

        <div id="loginForm">

            <h3>User Login</h3>

            <form id="userLoginForm">

                <div class="form-group">

                    <label for="loginUsername">Username:</label>

                    <input type="text" class="form-control" id="loginUsername" placeholder="Enter your username" required>

                </div>

                <div class="form-group">

                    <label for="loginPassword">Password:</label>

                    <input type="password" class="form-control" id="loginPassword" placeholder="Enter your password" required>

                </div>

                <button type="submit" class="btn btn-primary btn-center">Login</button>

                <div id="loginMessage" class="text-danger mt-2"></div>

                <div class="form-footer">

                    Don't have an account? <a href="#" onclick="showRegisterForm()">Register here</a>

                </div>

            </form>

        </div>

        <div id="registerForm">

            <h3>User Register</h3>

            <form id="userRegisterForm">

                <div class="form-group">

                    <label for="registerUsername">Username:</label>

                    <input type="text" class="form-control" id="registerUsername" placeholder="Enter your username" required>

                </div>

                <div class="form-group">

                    <label for="registerPassword">Password:</label>

                    <input type="password" class="form-control" id="registerPassword" placeholder="Enter your password" required>

                </div>

                <div class="form-group">

                    <label for="confirmPassword">Confirm Password:</label>

                    <input type="password" class="form-control" id="confirmPassword" placeholder="Confirm your password" required>

                </div>

                <button type="submit" class="btn btn-primary btn-center">Register</button>

                <div id="registerMessage" class="text-success mt-2"></div>

                <div class="form-footer">

                    Already have an account? <a href="#" onclick="showLoginForm()">Login here</a>

                </div>

            </form>

        </div>

        <div id="adminLoginForm">

            <h3>Admin Login</h3>

            <form id="adminLogin">

                <div class="form-group">

                    <label for="adminUsername">Username:</label>

                    <input type="text" class="form-control" id="adminUsername" placeholder="Enter admin username" required>

                </div>

                <div class="form-group">

                    <label for="adminPassword">Password:</label>

                    <input type="password" class="form-control" id="adminPassword" placeholder="Enter admin password" required>

                </div>

                <button type="submit" class="btn btn-primary btn-center">Login</button>

                <div id="adminLoginMessage" class="text-danger mt-2"></div>

            </form>

        </div>

    </div>

    <script>

        // Simulated user database

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

        const adminCredentials = {

            username: 'admin',

            password: 'admin123'

        };

        // Show login form by default

        showLoginForm();

        // Show forms based on user actions

        function showLoginForm() {

            hideAllForms();

            const form = document.getElementById('loginForm');

            form.classList.add('active', 'slide-in');

            form.style.display = 'block';

        }

        function showRegisterForm() {

            hideAllForms();

            const form = document.getElementById('registerForm');

            form.classList.add('active', 'slide-in');

            form.style.display = 'block';

        }

        function showAdminLogin() {

            hideAllForms();

            const form = document.getElementById('adminLoginForm');

            form.classList.add('active', 'slide-in');

            form.style.display = 'block';

        }

        function hideAllForms() {

            const forms = document.querySelectorAll('#loginForm, #registerForm, #adminLoginForm');

            forms.forEach(form => {

                form.classList.remove('active', 'slide-in');

                form.style.display = 'none';

            });

        }

        document.getElementById('userLoginForm').addEventListener('submit', function (event) {

            event.preventDefault();

            const username = document.getElementById('loginUsername').value;

            const password = document.getElementById('loginPassword').value;

            // Dummy login logic

            if (users.some(user => user.username === username && user.password === password)) {

                localStorage.setItem('username', username); // Store the username in localStorage

                window.location.href = "user.html"; // Change to your user dashboard page

            } else {

                document.getElementById('loginMessage').innerText = 'Invalid username or password!';

            }

        });

        document.getElementById('userRegisterForm').addEventListener('submit', function (event) {

            event.preventDefault();

            const username = document.getElementById('registerUsername').value;

            const password = document.getElementById('registerPassword').value;

            const confirmPassword = document.getElementById('confirmPassword').value;

            if (password === confirmPassword) {

                // Generate a new User ID

                const userId = `UID${String(users.length + 1).padStart(3, '0')}`;

                users.push({ username, password, userId });

                localStorage.setItem('users', JSON.stringify(users)); // Save to localStorage

                document.getElementById('registerMessage').innerText = `Registration successful!`;

                document.getElementById('registerUsername').value = '';

                document.getElementById('registerPassword').value = '';

                document.getElementById('confirmPassword').value = '';

            } else {

                document.getElementById('registerMessage').innerText = 'Passwords do not match!';

            }

        });

        document.getElementById('adminLogin').addEventListener('submit', function (event) {

            event.preventDefault();

            const username = document.getElementById('adminUsername').value;

            const password = document.getElementById('adminPassword').value;

            if (username === adminCredentials.username && password === adminCredentials.password) {

                window.location.href = "admin.html"; // Change to your admin dashboard page

            } else {

                document.getElementById('adminLoginMessage').innerText = 'Invalid admin username or password!';

            }

        });

        // Prevent the default action for the search form

        document.getElementById('searchForm').addEventListener('submit', function (event) {

            event.preventDefault();

            const query = document.getElementById('searchInput').value;

            console.log('Search query:', query);

            // Implement search functionality here

        });

    </script>

</body>

</html>

**Admin.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Admin Page</title>

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

    <style>

        /\* Navigation bar color \*/

        .navbar {

            background-color: rgba(1, 70, 120, 255);

            height: 80px;

            padding: 20px;

        }

        .navbar-brand {

            font-size: 1.8rem;

            color: rgb(255, 255, 255);

        }

        .navbar-nav .nav-link {

            font-size: 1.2rem;

            color: rgb(255, 255, 255);

        }

        .navbar-nav .nav-link:hover {

            color: #d3d3d3;

        }

    </style>

</head>

<body>

    <nav class="navbar navbar-expand-lg">

        <a class="navbar-brand" href="#">Online Voting System</a>

        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

            <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse" id="navbarNav">

            <ul class="navbar-nav ml-auto">

                <li class="nav-item">

                    <a class="nav-link" id="logoutLink" href="#">Logout</a>

                </li>

            </ul>

        </div>

    </nav>

    <div class="container mt-5">

        <div class="row">

            <div class="col-md-6">

                <h3>Add Participants</h3>

                <form id="addParticipantForm">

                    <div class="form-group">

                        <label for="participantName">Participant Name:</label>

                        <input type="text" class="form-control" id="participantName" required>

                    </div>

                    <div class="form-group">

                        <label for="participantId">Participant ID:</label>

                        <input type="text" class="form-control" id="participantId" required>

                    </div>

                    <button type="submit" class="btn btn-primary">Add Participant</button>

                </form>

            </div>

            <div class="col-md-6 mt-0">

                <h3>Search Participants</h3>

                <div class="form-group">

                    <label for="searchInput">Participant Name or ID:</label>

                    <input type="text" class="form-control" id="searchInput" placeholder="Search by name or ID">

                    <button id="searchButton" class="btn btn-secondary mt-3">Search</button>

                </div>

            </div>

        </div>

        <h3 class="mt-4">Participants List</h3>

        <table class="table">

            <thead>

                <tr>

                    <th>S.No</th>

                    <th>Participant Name</th>

                    <th>Participant ID</th>

                    <th>Votes</th>

                    <th>Percentage</th>

                    <th>Actions</th>

                </tr>

            </thead>

            <tbody id="participantsTableBody">

                <!-- Participants will be dynamically added here -->

            </tbody>

        </table>

        <!-- Start and Stop Voting Buttons -->

        <button id="startVoting" class="btn btn-success mt-3">Start Voting</button>

        <button id="stopVoting" class="btn btn-danger mt-3">Stop Voting</button>

        <button id="publishResults" class="btn btn-warning mt-3" disabled>Publish Results</button>

        <!-- User Accounts Section -->

        <h3 class="mt-4">User Accounts</h3>

        <table class="table">

            <thead>

                <tr>

                    <th>S.No</th>

                    <th>Username</th>

                    <th>Password</th>

                    <th>Actions</th>

                </tr>

            </thead>

            <tbody id="userAccountsTableBody">

                <!-- User accounts will be dynamically added here -->

            </tbody>

        </table>

    </div>

    <script>

        let participants = JSON.parse(localStorage.getItem('participants')) || []; // Retrieve stored participants

        let users = JSON.parse(localStorage.getItem('users')) || []; // Retrieve stored user accounts

        let votingActive = false; // Voting status

        let resultsPublished = false; // Publish status

        // Function to populate participants table

        function populateParticipantsTable(filteredParticipants = null) {

            const participantsTableBody = document.getElementById('participantsTableBody');

            participantsTableBody.innerHTML = ''; // Clear existing content

            // Use filtered participants if provided, otherwise use the full list

            const participantsToDisplay = filteredParticipants || participants;

            participantsToDisplay.forEach((participant, index) => {

                const totalVotes = participants.reduce((sum, p) => sum + p.votes, 0); // Calculate total votes

                const percentage = totalVotes > 0 ? ((participant.votes / totalVotes) \* 100).toFixed(2) + '%' : '0.00%'; // Calculate percentage

                const row = document.createElement('tr');

                row.innerHTML = `

                    <td>${index + 1}</td>

                    <td>${participant.name}</td>

                    <td>${participant.id}</td>

                    <td>${participant.votes}</td>

                    <td>${percentage}</td>

                    <td><button class="btn btn-danger btn-sm" onclick="deleteParticipant(${index})">Delete</button></td>

                `;

                participantsTableBody.appendChild(row);

            });

        }

        // Function to add a new participant

        document.getElementById('addParticipantForm').addEventListener('submit', function (event) {

            event.preventDefault(); // Prevent default form submission

            const name = document.getElementById('participantName').value;

            const id = document.getElementById('participantId').value;

            // Add participant

            participants.push({ id, name, votes: 0 }); // Initialize votes to 0

            localStorage.setItem('participants', JSON.stringify(participants)); // Store in localStorage

            // Clear the input fields

            document.getElementById('participantName').value = '';

            document.getElementById('participantId').value = '';

            // Refresh the participants table

            populateParticipantsTable();

        });

        // Function to delete a participant

        function deleteParticipant(index) {

            if (confirm("Are you sure you want to delete this participant?")) {

                participants.splice(index, 1); // Remove the participant from the array

                localStorage.setItem('participants', JSON.stringify(participants)); // Update localStorage

                populateParticipantsTable(); // Re-render the participants table

            }

        }

        // Function to populate user accounts table

        function populateUserAccounts() {

            const userAccountsTableBody = document.getElementById('userAccountsTableBody');

            userAccountsTableBody.innerHTML = ''; // Clear existing content

            users.forEach((user, index) => {

                const row = document.createElement('tr');

                row.innerHTML = `

                    <td>${index + 1}</td>

                    <td>${user.username}</td>

                    <td>${user.password}</td>

                    <td><button class="btn btn-danger btn-sm" onclick="deleteUser('${user.username}')">Delete</button></td>

                `;

                userAccountsTableBody.appendChild(row);

            });

        }

        // Function to delete a user

        function deleteUser(username) {

            users = users.filter(user => user.username !== username);

            localStorage.setItem('users', JSON.stringify(users)); // Update localStorage

            populateUserAccounts(); // Refresh the table

        }

        // Search functionality

        document.getElementById('searchButton').addEventListener('click', function () {

            const searchTerm = document.getElementById('searchInput').value.toLowerCase();

            // Filter participants based on search input

            const filteredParticipants = participants.filter(participant =>

                participant.name.toLowerCase().includes(searchTerm) || participant.id.toLowerCase().includes(searchTerm)

            );

            // Populate the table with filtered participants

            populateParticipantsTable(filteredParticipants);

        });

        // Start Voting Functionality

        document.getElementById('startVoting').addEventListener('click', function () {

            votingActive = true;

            localStorage.setItem('votingStatus', 'open'); // Update voting status in localStorage

            alert("Voting has started!");

        });

        // Stop Voting Functionality

        document.getElementById('stopVoting').addEventListener('click', function () {

            votingActive = false;

            localStorage.setItem('votingStatus', 'closed'); // Update voting status in localStorage

            document.getElementById('publishResults').disabled = false; // Enable publish results button

            alert("Voting has stopped!");

        });

        // Publish Results Functionality

        document.getElementById('publishResults').addEventListener('click', function () {

            if (!resultsPublished) { // Check if results are already published

                localStorage.setItem('publishStatus', 'published'); // Update publish status in localStorage

                alert("Results have been published!");

                resultsPublished = true;

                window.location.href = 'result.html'; // Redirect to result.html

            }

        });

        // Populate participants and user accounts tables on page load

        window.onload = function () {

            populateParticipantsTable();

            populateUserAccounts();

        };

        // Logout functionality

        document.getElementById('logoutLink').addEventListener('click', function () {

            window.location.href = 'index.html'; // Redirect to login page

        });

    </script>

</body>

</html>

**User.html**

<!-- This is your updated user.html -->

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>User Page</title>

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

    <style>

        /\* Navigation bar color \*/

        .navbar {

            background-color: rgba(1, 70, 120, 255);

            height: 80px;

            padding: 20px;

        }

        .navbar-brand {

            font-size: 1.8rem;

            color: rgb(255, 255, 255);

        }

        .navbar-nav .nav-link {

            font-size: 1.2rem;

            color: rgb(255, 255, 255);

        }

        .navbar-nav .nav-link:hover {

            color: #d3d3d3;

        }

        .user-profile {

            margin-bottom: 20px;

        }

    </style>

</head>

<body>

    <nav class="navbar navbar-expand-lg">

        <a class="navbar-brand" href="#">Online Voting System</a>

        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

            <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse" id="navbarNav">

            <form class="form-inline ml-auto" id="searchForm">

                <input class="form-control mr-sm-2" type="search" placeholder="Search" id="searchInput" aria-label="Search">

                <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>

            </form>

            <ul class="navbar-nav ml-2">

                <li class="nav-item">

                    <a class="nav-link" href="index.html">Logout</a>

                </li>

            </ul>

        </div>

    </nav>

    <div class="container mt-5">

        <div class="user-profile">

            <h5>User Profile</h5>

            <p id="usernameDisplay"></p>

            <p id="userIdDisplay"></p> <!-- User ID display -->

        </div>

        <h3>Vote for Participants</h3>

        <table class="table">

            <thead>

                <tr>

                    <th>S.No</th>

                    <th>Participant Name</th>

                    <th>Participant ID</th>

                    <th>Vote</th>

                </tr>

            </thead>

            <tbody id="participantsTableBody">

                <!-- Participants will be dynamically added here -->

            </tbody>

        </table>

        <!-- View Results Button -->

        <button id="viewResultsButton" class="btn btn-success">View Results</button>

    </div>

    <script>

        let participants = JSON.parse(localStorage.getItem('participants')) || [

            { name: 'Participant 1', id: 'P1', votes: 0 },

            { name: 'Participant 2', id: 'P2', votes: 0 },

            { name: 'Participant 3', id: 'P3', votes: 0 },

            { name: 'Participant 4', id: 'P4', votes: 0 },

            { name: 'Participant 5', id: 'P5', votes: 0 },

            { name: 'Participant 6', id: 'P6', votes: 0 },

        ];

        let userVotes = 0; // Track user votes

        const username = localStorage.getItem('username');

        const users = JSON.parse(localStorage.getItem('users')) || [];

        const user = users.find(u => u.username === username);

        document.getElementById('usernameDisplay').innerText = username ? `Username: ${username}` : 'No user logged in';

        document.getElementById('userIdDisplay').innerText = user ? `User ID: ${user.userId}` : '';

        function renderParticipants(filteredParticipants = participants) {

            const tableBody = document.getElementById('participantsTableBody');

            tableBody.innerHTML = ''; // Clear the table before rendering

            filteredParticipants.forEach((participant, index) => {

                const row = document.createElement('tr');

                row.innerHTML = `

                    <td>${index + 1}</td>

                    <td>${participant.name}</td>

                    <td>${participant.id}</td>

                    <td><button class="btn btn-primary" onclick="vote(${index})">Vote</button></td>

                `;

                tableBody.appendChild(row);

            });

        }

        function vote(index) {

            const votingStatus = localStorage.getItem('votingStatus'); // Get voting status from localStorage

            if (votingStatus === 'open') { // Check if voting is open

                if (userVotes < 5) {

                    participants[index].votes++; // Increment votes for the participant

                    localStorage.setItem('participants', JSON.stringify(participants)); // Update localStorage

                    userVotes++; // Increment user's vote count

                    renderParticipants(); // Re-render the participants to show updated votes

                } else {

                    alert("You can only vote for up to 5 participants.");

                }

            } else {

                alert("Voting is currently closed."); // Display alert if voting is closed

            }

        }

        // Search functionality

        document.getElementById('searchForm').addEventListener('submit', function(event) {

            event.preventDefault();

            const searchTerm = document.getElementById('searchInput').value.toLowerCase();

            const filteredParticipants = participants.filter(participant =>

                participant.name.toLowerCase().includes(searchTerm)

            );

            renderParticipants(filteredParticipants);

        });

        // View Results button functionality

        document.getElementById('viewResultsButton').addEventListener('click', function() {

            const publishStatus = localStorage.getItem('publishStatus'); // Check publish status

            if (publishStatus === 'published') {

                window.location.href = "user-result.html"; // Redirect to results page

            } else {

                alert("Results are not published yet. Please check back later."); // Display alert if results are not published

            }

        });

        // Initial render of participants

        renderParticipants(participants);

    </script>

</body>

</html>

**User-result.html**

<!-- This is your updated user-result.html -->

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Results Page</title>

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

    <style>

        /\* Navigation bar color \*/

        .navbar {

            background-color: rgba(1, 70, 120, 255);

            height: 80px;

            padding: 20px;

        }

        .navbar-brand {

            font-size: 1.8rem;

            color: rgb(255, 255, 255);

        }

        .navbar-nav .nav-link {

            font-size: 1.2rem;

            color: rgb(255, 255, 255);

        }

        .navbar-nav .nav-link:hover {

            color: #d3d3d3;

        }

        .result {

            font-weight: bold;

            font-size: 1.5rem;

        }

        .winner {

            margin-top: 20px;

            font-size: 1.5rem;

            color: green;

        }

    </style>

</head>

<body>

    <nav class="navbar navbar-expand-lg">

        <a class="navbar-brand" href="#">Online Voting System</a>

        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

            <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse" id="navbarNav">

            <ul class="navbar-nav ml-auto">

                <li class="nav-item">

                    <a class="nav-link" href="user.html" id="backLink">Back</a>

                </li>

                <li class="nav-item">

                    <a class="nav-link" href="index.html" id="logoutLink">Logout</a>

                </li>

            </ul>

        </div>

    </nav>

    <div class="container mt-5">

        <h3 class="text-center">Voting Results</h3>

        <table class="table">

            <thead>

                <tr>

                    <th>Participant Name</th>

                    <th>Participant ID</th>

                    <th>Total Votes</th>

                    <th>Percentage</th>

                </tr>

            </thead>

            <tbody id="resultsTableBody">

                <!-- Results will be dynamically added here -->

            </tbody>

        </table>

        <div id="winnerAnnouncement" class="text-center winner"></div>

    </div>

    <script>

        // Function to calculate percentages and determine the winner

        function calculateResults() {

            const participants = JSON.parse(localStorage.getItem('participants')) || [];

            const totalVotes = participants.reduce((sum, participant) => sum + participant.votes, 0);

            const resultsTableBody = document.getElementById('resultsTableBody');

            resultsTableBody.innerHTML = ''; // Clear the table before rendering results

            let winner = null;

            participants.forEach(participant => {

                const percentage = totalVotes > 0 ? ((participant.votes / totalVotes) \* 100).toFixed(2) : 0;

                const row = document.createElement('tr');

                row.innerHTML = `

                    <td>${participant.name}</td>

                    <td>${participant.id}</td>

                    <td>${participant.votes}</td>

                    <td>${percentage}%</td>

                `;

                resultsTableBody.appendChild(row);

                // Determine winner

                if (!winner || participant.votes > winner.votes) {

                    winner = participant;

                }

            });

            // Display the winner

            if (winner) {

                document.getElementById('winnerAnnouncement').innerText = `Winner: ${winner.name} (Votes: ${winner.votes})`;

            } else {

                document.getElementById('winnerAnnouncement').innerText = 'No votes cast yet.';

            }

        }

        // Check publish status and calculate results

        const publishStatus = localStorage.getItem('publishStatus');

        if (publishStatus === 'published') {

            calculateResults(); // Calculate and display results

        } else {

            document.getElementById('resultsTableBody').innerHTML = '<tr><td colspan="4">Results are not published yet.</td></tr>';

        }

        // Event listeners for back and logout links

        document.getElementById('backLink').addEventListener('click', function () {

            window.location.href = "user.html"; // Redirect to user.html

        });

        document.getElementById('logoutLink').addEventListener('click', function () {

            localStorage.clear(); // Clear local storage on logout

            window.location.href = "index.html"; // Redirect to login page

        });

    </script>

</body>

</html>

**Result.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Results Page</title>

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

    <style>

        /\* Navigation bar color \*/

        .navbar {

            background-color: rgba(1, 70, 120, 255);

            height: 80px;

            padding: 20px;

        }

        .navbar-brand {

            font-size: 1.8rem;

            color: rgb(255, 255, 255);

        }

        .navbar-nav .nav-link {

            font-size: 1.2rem;

            color: rgb(255, 255, 255);

        }

        .navbar-nav .nav-link:hover {

            color: #d3d3d3;

        }

        .result {

            font-weight: bold;

            font-size: 1.5rem;

        }

        .winner {

            margin-top: 20px;

            font-size: 1.5rem;

            color: green;

        }

    </style>

</head>

<body>

    <nav class="navbar navbar-expand-lg">

        <a class="navbar-brand" href="#">Online Voting System</a>

        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

            <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse" id="navbarNav">

            <ul class="navbar-nav ml-auto">

                <li class="nav-item">

                    <a class="nav-link" id="backLink" href="#">Back</a>

                </li>

                <li class="nav-item">

                    <a class="nav-link" id="logoutLink" href="#">Logout</a>

                </li>

            </ul>

        </div>

    </nav>

    <div class="container mt-5">

        <h3>Voting Results</h3>

        <table class="table">

            <thead>

                <tr>

                    <th>Participant Name</th>

                    <th>Participant ID</th>

                    <th>Votes</th>

                    <th>Percentage</th>

                </tr>

            </thead>

            <tbody id="resultsTableBody">

                <!-- Results will be dynamically added here -->

            </tbody>

        </table>

        <div class="winner" id="winnerAnnouncement"></div>

    </div>

    <script>

        window.onload = function() {

            const participants = JSON.parse(localStorage.getItem('participants')) || []; // Retrieve stored participants

            const resultsTableBody = document.getElementById('resultsTableBody');

            const totalVotes = participants.reduce((sum, p) => sum + p.votes, 0); // Calculate total votes

            let winner = null; // Initialize winner variable

            participants.forEach(participant => {

                const percentage = totalVotes > 0 ? ((participant.votes / totalVotes) \* 100).toFixed(2) + '%' : '0.00%'; // Calculate percentage

                const row = document.createElement('tr');

                row.innerHTML = `

                    <td>${participant.name}</td>

                    <td>${participant.id}</td>

                    <td>${participant.votes}</td>

                    <td>${percentage}</td>

                `;

                resultsTableBody.appendChild(row);

                // Determine the winner

                if (!winner || participant.votes > winner.votes) {

                    winner = participant; // Update winner if current participant has more votes

                }

            });

            // Display the winner if available

            if (winner) {

                document.getElementById('winnerAnnouncement').innerHTML = `Winner: ${winner.name} with ${winner.votes} votes!`;

            }

        };

        // Logout functionality

        document.getElementById("logoutLink").addEventListener("click", function (event) {

            event.preventDefault(); // Prevent default anchor behavior

            localStorage.removeItem("loggedInUser"); // Clear the logged-in user

            alert("You have been logged out.");

            window.location.href = "index.html"; // Redirect to the login page

        });

        // Back button functionality

        document.getElementById("backLink").addEventListener("click", function (event) {

            event.preventDefault(); // Prevent default anchor behavior

            window.location.href = "admin.html"; // Redirect to admin page

        });

    </script>

</body>

</html>

**Script.js**

let users = JSON.parse(localStorage.getItem('users')) || []; // Load users from localStorage or initialize empty array

// Login

document.getElementById("loginForm")?.addEventListener("submit", function (event) {

    event.preventDefault(); // Prevent the default form submission

    const username = document.getElementById("username").value;

    const password = document.getElementById("password").value;

    const user = users.find(user => user.username === username && user.password === password);

    if (user) {

        localStorage.setItem("loggedInUser", username);

        // Initialize vote history for the new user if not present

        let userVotesList = JSON.parse(localStorage.getItem("userVotes")) || {};

        if (!userVotesList[username]) {

            userVotesList[username] = []; // Initialize empty vote list for new user

            localStorage.setItem("userVotes", JSON.stringify(userVotesList));

        }

        alert("Login successful! Welcome, " + username);

        window.location.href = "dashboard.html"; // Redirect to the dashboard

    } else {

        alert("Invalid username or password!");

    }

});

// Registration

document.getElementById("registerForm")?.addEventListener("submit", function (event) {

    event.preventDefault(); // Prevent the default form submission

    const username = document.getElementById("username").value;

    const password = document.getElementById("password").value;

    const newUser = { username: username, password: password };

    users.push(newUser); // Add new user to users array

    localStorage.setItem('users', JSON.stringify(users)); // Save users to localStorage

    alert("Registration successful! You can now log in.");

    window.location.href = "index.html"; // Redirect to the login page

});