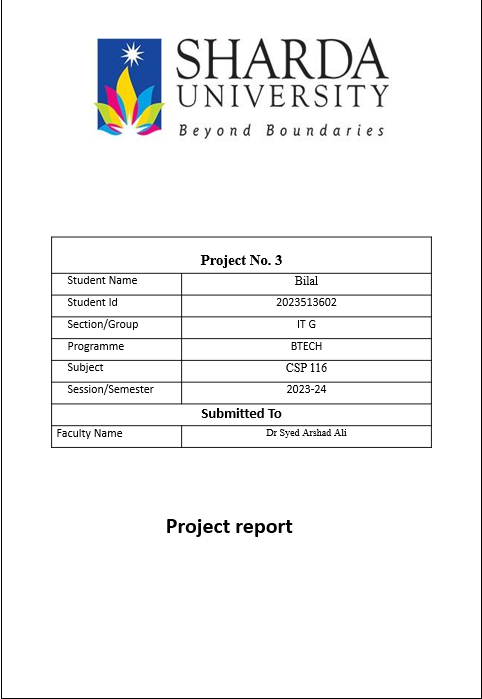
hhsss

TroubleTrace SolutionSphre

I am completing the frontend of Header, Footer and Main content area. Using HTML,CSS and XML

Basic Layout:

Here's a simple HTML/CSS layout that covers the header, footer, and main content areas:

HTML:

html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Basic Layout</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<header>

<h1>Header</h1>

</header>

<div class="content">

<h2>Main Content Area</h2>

<p>This is the main content area. Placeholder text can be inserted here.</p>

</div>

<footer>

<p>Footer</p>

</footer>

</body>

</html>

CSS (styles.css):

css

body {

margin: 0;

font-family: Arial, sans-serif;

}

header {

background-color: #333;

color: #fff;

padding: 20px;

text-align: center;

}

footer {

background-color: #333;

color: #fff;

padding: 10px;

position: fixed;

bottom: 0;

width: 100%;

text-align: center;

}

.content {

padding: 20px;

margin-top: 20px;

}

User Authentication:

User authentication system setup kare jismein users apne accounts create kar sakein, login kar sakein, aur logout kar sakein. Iske liye aapko registration form, login form, aur logout button shamil karne honge.

HTML:

html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>User Authentication</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div id="registrationForm" class="form-container">

<h2>Register</h2>

<form id="registerForm">

<input type="text" id="registerUsername" placeholder="Username" required>

<input type="password" id="registerPassword" placeholder="Password" required>

<button type="submit">Register</button>

</form>

</div>

<div id="loginForm" class="form-container">

<h2>Login</h2>

<form id="loginForm">

<input type="text" id="loginUsername" placeholder="Username" required>

<input type="password" id="loginPassword" placeholder="Password" required>

<button type="submit">Login</button>

</form>

</div>

<button id="logoutBtn" class="hidden" onclick="logout()">Logout</button>

<script src="script.js"></script>

</body>

</html>

CSS (styles.css):

css

.form-container {

max-width: 300px;

margin: 0 auto;

padding: 20px;

border: 1px solid #ccc;

border-radius: 5px;

margin-top: 20px;

}

button {

display: block;

width: 100%;

padding: 10px;

margin-top: 10px;

background-color: #4CAF50;

color: white;

border: none;

border-radius: 3px;

cursor: pointer;

}

.hidden {

display: none;

}

JavaScript (script.js):

javascript

// Dummy user data for demonstration purposes

const users = [

{ username: "user1", password: "password1" },

{ username: "user2", password: "password2" }

];

const registrationForm = document.getElementById("registerForm");

const loginForm = document.getElementById("loginForm");

const logoutBtn = document.getElementById("logoutBtn");

registrationForm.addEventListener("submit", register);

loginForm.addEventListener("submit", login);

function register(event) {

event.preventDefault();

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

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

// Check if username already exists

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

alert("Username already exists. Please choose a different one.");

return;

}

// Add new user

users.push({ username, password });

alert("Registration successful. Please login.");

}

function login(event) {

event.preventDefault();

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

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

// Check if username and password match

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

alert("Login successful.");

loginForm.reset();

showLogoutButton();

} else {

alert("Invalid username or password.");

}

}

function logout() {

hideLogoutButton();

}

function showLogoutButton() {

logoutBtn.classList.remove("hidden");

}

function hideLogoutButton() {

logoutBtn.classList.add("hidden");

}

Develop a backend API that will handle user authentication. This will include endpoints for user registration, login, and logout.

Using Express framework with Node.js you can develop a backend API that will handle user authentication. Below is a basic example that includes endpoints of user registration, login, and logout:

javascript

const express = require('express');

const bodyParser = require('body-parser');

const bcrypt = require('bcrypt');

const jwt = require('jsonwebtoken');

const app = express();

const PORT = 3000;

// Middleware to parse JSON bodies

app.use(bodyParser.json());

// Dummy database to store user data

const users = [];

// Secret key for JWT

const secretKey = 'your\_secret\_key';

// Endpoint for user registration

app.post('/register', async (req, res) => {

try {

const { username, password, email, full\_name } = req.body;

// Check if username already exists

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

return res.status(400).json({ error: 'Username already exists. Please choose a different one.' });

}

// Hash the password

const hashedPassword = await bcrypt.hash(password, 10);

// Store user data

users.push({ username, password: hashedPassword, email, full\_name });

res.status(201).json({ message: 'Registration successful. Please login.' });

} catch (error) {

res.status(500).json({ error: 'Internal server error.' });

}

});

// Endpoint for user login

app.post('/login', async (req, res) => {

try {

const { username, password } = req.body;

// Find user by username

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

if (!user) {

return res.status(401).json({ error: 'Invalid username or password.' });

}

// Compare passwords

const passwordMatch = await bcrypt.compare(password, user.password);

if (!passwordMatch) {

return res.status(401).json({ error: 'Invalid username or password.' });

}

// Generate JWT token

const token = jwt.sign({ username: user.username }, secretKey);

res.status(200).json({ token });

} catch (error) {

res.status(500).json({ error: 'Internal server error.' });

}

});

// Endpoint for user logout (not necessary with JWT)

app.listen(PORT, () => {

console.log(`Server is running on http://localhost:${PORT}`);

});

\*Frontend Integration:\*

Integrate frontend and backend so that user authentication works seamlessly. Send requests from frontend to backend API endpoints and complete the user authentication process.

To integrate the frontend with the backend, you can send API requests to the frontend using JavaScript.

html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>User Authentication Frontend</title>

</head>

<body>

<div id="registrationForm" class="form-container">

<h2>Register</h2>

<input type="text" id="registerUsername" placeholder="Username" required>

<input type="password" id="registerPassword" placeholder="Password" required>

<input type="email" id="registerEmail" placeholder="Email">

<input type="text" id="registerFullName" placeholder="Full Name">

<button onclick="register()">Register</button>

</div>

<div id="loginForm" class="form-container">

<h2>Login</h2>

<input type="text" id="loginUsername" placeholder="Username" required>

<input type="password" id="loginPassword" placeholder="Password" required>

<button onclick="login()">Login</button>

</div>

<button id="logoutBtn" class="hidden" onclick="logout()">Logout</button>

<script>

async function register() {

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

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

const email = document.getElementById("registerEmail").value;

const fullName = document.getElementById("registerFullName").value;

try {

const response = await fetch('/register', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({ username, password, email, full\_name: fullName })

});

const data = await response.json();

alert(data.message);

} catch (error) {

console.error('Error:', error);

}

}

async function login() {

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

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

try {

const response = await fetch('/login', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({ username, password })

});

const data = await response.json();

const token = data.token;

if (token) {

localStorage.setItem('token', token);

showLogoutButton();

alert('Login successful.');

} else {

alert('Invalid username or password.');

}

} catch (error) {

console.error('Error:', error);

}

}

async function logout() {

localStorage.removeItem('token');

hideLogoutButton();

}

function showLogoutButton() {

document.getElementById('logoutBtn').classList.remove('hidden');

}

function hideLogoutButton() {

document.getElementById('logoutBtn').classList.add('hidden');

}

</script>

</body>

</html>

Dashboard:

Dashboard for users to show their sued issues and their status.

HTML (index.html):

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>User Dashboard</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<header>

<h1>User Dashboard</h1>

</header>

<main>

<section>

<h2>Submitted Issues</h2>

<ul id="submittedIssues"></ul>

</section>

<section>

<h2>Issue Status</h2>

<ul id="issueStatus"></ul>

</section>

</main>

<footer>

<p>&copy; 2024 Your Company</p>

</footer>

<script src="script.js"></script>

</body>

</html>

CSS (styles.css):

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

}

header {

background-color: #333;

color: #fff;

padding: 20px;

text-align: center;

}

main {

padding: 20px;

}

section {

margin-bottom: 20px;

}

footer {

background-color: #333;

color: #fff;

text-align: center;

padding: 10px;

}

JavaScript (script.js):

// Sample data for demonstration purposes

const submittedIssues = ['Issue 1', 'Issue 2', 'Issue 3'];

const issueStatus = ['Resolved', 'In Progress', 'Pending'];

document.addEventListener('DOMContentLoaded', function() {

const submittedIssuesList = document.getElementById('submittedIssues');

const issueStatusList = document.getElementById('issueStatus');

// Display submitted issues

submittedIssues.forEach(issue => {

const listItem = document.createElement('li');

listItem.textContent = issue;

submittedIssuesList.appendChild(listItem);

});

// Display issue status

issueStatus.forEach(status => {

const listItem = document.createElement('li');

listItem.textContent = status;

issueStatusList.appendChild(listItem);

});

});

Advanced Search:

Must have advanced search feature to identify problems quickly.

HTML (index.html):

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Advanced Search</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<header>

<h1>Advanced Search</h1>

</header>

<main>

<form id="searchForm">

<input type="text" id="searchInput" placeholder="Search...">

<button type="submit">Search</button>

</form>

<div id="searchResults"></div>

</main>

<footer>

<p>&copy; 2024 Your Company</p>

</footer>

<script src="script.js"></script>

</body>

</html>

CSS (styles.css):

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

}

header {

background-color: #333;

color: #fff;

padding: 20px;

text-align: center;

}

main {

padding: 20px;

}

form {

margin-bottom: 20px;

}

input[type="text"] {

width: 70%;

padding: 10px;

margin-right: 10px;

}

button {

padding: 10px;

background-color: #4CAF50;

color: #fff;

border: none;

cursor: pointer;

}

#searchResults {

border: 1px solid #ddd;

border-radius: 5px;

padding: 10px;

}

JavaScript (script.js):

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

event.preventDefault();

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

// Sample search functionality

const searchResults = performSearch(searchTerm);

// Display search results

displaySearchResults(searchResults);

});

function performSearch(searchTerm) {

// Sample search function, replace with actual search logic

const searchResults = [

'Result 1 for ' + searchTerm,

'Result 2 for ' + searchTerm,

'Result 3 for ' + searchTerm

];

return searchResults;

}

function displaySearchResults(results) {

const searchResultsDiv = document.getElementById('searchResults');

searchResultsDiv.innerHTML = ''; // Clear previous search results

results.forEach(result => {

const resultDiv = document.createElement('div');

resultDiv.textContent = result;

searchResultsDiv.appendChild(resultDiv);

});

}

## Thank you

# Bilal 2023513602