

Data Plan

Group Name/Number:

Feature/Interaction	Purpose/Description	Request Data Source	Request sent to server	Processing Needed/Response Data Source	Response sent to client	Request Type & Path	Other Notes
Dynamic side bar	To provide a user-specific navigation experience where the sidebar content (links, options) dynamically adjusts based on user roles, preferences, or other criteria.	Component: Sidebar that appears when the user clicks the open button. User Input: User interaction to open the sidebar. The system checks if the user is logged in.	AJAX request to fetch user-specific navigation items. { "sessionId": "[session identifier or token]" }	Verify the user's session or token to determine if they are logged in. SELECT user_id, is_logged_in FROM sessions WHERE session_id = ?; SELECT name, url FROM navigation_items WHERE user_role = (SELECT role FROM users WHERE user_id = ?);	JSON formatted AJAX response containing the navigation items, including conditional items based on login status. { { "name": "Home", "url": "/home" }, { "name": "Profile", "url": "/profile" }, { "name": "Settings", "url": "/settings", "condition": "loggedIn" }, ... }	GET Request to /api/sidebar-navigation	
Hidden Sidebar	Provides a mechanism for users to toggle a sidebar on and off, enhancing the user interface by allowing more screen space when the sidebar is hidden and easy access to navigation when needed.	Component: Sidebar with open and close buttons. User Input: User clicks the open button (≡) to show the sidebar and the close button (✕) to hide it.	Execute the Javascript openNav() function to change the CSS properties and display the sidebar. function openNav() { document.getElementById("mySidebar").style.width = "250px"; document.getElementById("main").style.marginLeft = "250px"; } Execute the Javascript closeNav() function to reset the CSS properties and hide the sidebar. function closeNav() { document.getElementById("mySidebar").style.width = "0"; document.getElementById("main").style.marginLeft = "0"; }		Changes are made directly in the DOM via JavaScript, so no data is sent back from a server.		
User Profile Management	Allows users to view and update their personal details such as name, email, password, and other preferences directly through the website interface.	Component: User profile page with editable fields for personal details and preferences. User Input: Users enter or modify their details in the provided fields and submit the changes.	AJAX request containing the updated user details. { "userId": "[user's unique identifier]", "givenName": "[new first name]", "familyName": "[new last name]", "password": "[new password]" // Note: Password should be handled securely }	Authentication Check: Verify the user's identity and authorization to update the profile. Data Validation: Validate the new details to ensure they meet the system's requirements	JSON formatted AJAX response indicating the success or failure of the update operation. { "status": "success", "message": "Profile updated successfully." }	POST Request to /api/update-profile	
User Login	Allows users to securely log into the website using their credentials (username and password). This function is essential for authenticating users and providing access to personalized features and secure areas of the website.	Component: Login form on the website. User Input: Users enter their username and password into the form fields.	AJAX request containing the user's credentials. { "username": "[user's username]", "password": "[user's password]" // Note: Password should be transmitted securely }	Authentication Check: Verify the user's credentials against the database. Database Query: SELECT user_id, password_hash FROM users WHERE username = ?; Password Verification: Compare the provided password with the stored hash using a secure method	JSON formatted AJAX response indicating the success or failure of the login attempt. { "status": "success", "message": "Login successful", "token": "[session token]" // Used for maintaining user session }	POST Request to /api/login	

--	--	--	--	--	--	--	--