LEMP

Group 20

Links:

- https://github.com/PowellTravis/LEMP-UI
- https://lemp.powellnetworks.net
    - Username: demouser@powellnetworks.net
    - Password: GPWgG7qZkCQ=
    - User Expires Dec. 12
- https://docs.google.com/presentation/d/1IL3nXfyj3XzKIi0Xj9zOFqoQ2wSjApjkh_GjjITavT8/edit#slide=id.g3187cc9001a_0_24

Navbar.tsx

Description:

This component creates a dynamic navigation bar made for authenticated users. The bar displays different navigation options depending on the user's roles and session status.

Key Features:

- **Dynamic Role-Based Navigation:** Displays different options if the user is an admin or a standard user.
- **Session Awareness:** Utilizes `useSession` from NextAuth to determine if the user is logged in.
- **Active State Highlighting:** Highlights the currently active navigation link using a comparison with the `navTarget` prop.

- **Sign In/Out Options:** Displays a login button for unauthenticated users or a logout button for authenticated users.

Props:

- `navTarget` **(string):** Highlights the active navigation item based on the current page.

Systems.tsx

Description:

This component renders a grid of system blocks based on data fetched from a backend API. It also supports expandable views for individual system details.

Key Features:

- **Dynamic Data Fetching:** Retrieves system data from `/api/data/server`.
- **Loading State:** Displays a loading message while data is being fetched.
- **Expandable System Blocks:** Allows toggling the expanded view of each system block.
- **Session Provider:** Wraps the component in a `SessionProvider` to manage session-based functionality.

States:

- `data` (array): Stores the fetched system data.
- `isLoading` (boolean): Tracks if the data is still being loaded.
- `expandedSystemId` (string|null): Keeps track of which system block is expanded.

SystemBlock.tsx

Description:

- Dynamically renders system information.
- Supports collapsible details when `isExpanded` is true.

Props:

- **`system` (object):** The data for the specific system.
- **`isExpanded` (boolean):** Determines if the block should display expanded details.
- **`toggleExpand` (function):** Callback function to toggle the expanded state.

InitializeDatabase.tsx

Description:

This component initializes a database on the client-side by making a call to the `/api/init` endpoint. It runs once when the component mounts.

Key Features:

- **Client-Side Execution:** Ensures the database initialization logic only executes on the client side using `useEffect`.
- **Initialization Flag:** Prevents multiple unnecessary API calls with the `initialized` variable.
- **Error Handling:** Logs errors to the console if the API call fails.

API Endpoints

/api/data/server

- **Method:** `GET`
- **Description:** Fetches the list of systems to be displayed on the `Systems` page.
- **Response:** Returns a JSON array of system data.

/api/init

- **Method:** `GET`
- **Description:** Initializes the database when called.
- **Response:** Returns a JSON object with a message indicating the success or failure of the operation.

Styling

CSS Modules

The components rely on CSS modules for styling. Key styles are stored in:

- `nav.module.css` for the navigation bar.
- `home.module.css` for the homepage and general layout.
- `global.module.css` for global styles.
- `systems.module.css` for the grid layout on the Systems page.

Dependencies

The project uses the following key libraries and tools:

- **React:** Core library for building components.
- **Next.js:** Framework for rendering pages and API routes.
- **NextAuth.js:** Authentication library for managing user sessions.
- **CSS Modules:** Scoped styling for individual components.

Usage

To use these componentst:

1. Ensure your environment is set up with Next.js and NextAuth.js.
2. Place each component in its respective folder, e.g., `/components/`.
3. Import and use the components in your pages or other components as needed.
4. Ensure API endpoints `/api/data/server` and `/api/init` are implemented and functioning.

Creative Objective

Build a platform with the capability to be used to provide group policy like implementation to manage linux clients, by collecting information from systems automatically through a background process.

Tech summary section

The technologies used were NextJS, MYSQL, Cloudflare Tunnels, Active Directory, and Microsoft Entra ID

Individual Member Notes

Colter

I was involved with the styling of the website. I made the header of the website containing the title, and styling of the nav bar. I handled the css and design of the home page, as well as the systems page. I also handled making the slide show as well as the documentation for this project.

Travis

Wrote the backend and parts of the frontend, integrated with the MYSQL database, and wrote the policy and search backend which is integral to the application in python. Deployed the application and database on the hardware stack, then implemented integration of the Microsoft Entra ID which syncs passwords and users from AD On Prem System.

Conclusion

I learned effective strategies for managing the state and structure of the application. One example is the use of useSession from NextAuth in the Navbar.tsx to manage user sessions, ensuring that users are shown the correct navigation links based on their authentication status. Another strategy involved optimizing user interface behavior, such as expanding and collapsing system blocks based on user interaction, which was implemented using the useState hook in the Systems.tsx component. These practices allowed me to not only stay in sync with the project but also create a smoother user experience while maintaining a clean, versioned codebase. On the other hand I encountered difficulty editing the navigation bar due to the way the sign-in button was set up. The button's functionality was tightly integrated with session management using NextAuth, which made it challenging to modify the layout or behavior without affecting the authentication flow. This required careful adjustments to ensure the sign-in button was still functional while implementing the necessary changes to the navigation elements.

References

https://nextjs.org/docs