

Frontend

Home Component

The `Home` component serves as the main page for a task management application. It includes features for displaying user information, managing tasks, and interacting with the backend API.

Dependencies

- `React`
- `Next.js`
- `Axios`
- `TaskForm` component
- `TaskList` component
- `Navbar` component
- `useRouter` hook

State

- `name` (`useState`): Represents the name of the authenticated user.
- `email` (`useState`): Represents the email of the authenticated user.
- `tasks` (`useState`): Represents the list of tasks fetched from the backend.

useEffect Hook

The `useEffect` hook fetches user details and tasks when the component mounts. If the user is not authenticated, they are redirected to the login page.

Functions

`addTask(task: Object)`

Adds a new task to the tasks list.

`updateTaskStatus(taskId: string, newStatus: string)`

Updates the status of a task both in the backend and frontend.

Components

- `Navbar` Component: Displays user information in the navigation bar.
- `TaskForm` Component: Allows users to add new tasks.
- `TaskList` Component: Displays a list of tasks based on their status.

Styling

The component uses Tailwind CSS for styling, providing a responsive and visually appealing layout.

Additional Notes

- Relies on the presence of an authentication token (`localStorage.getItem("authorization")`) for user authentication.
- Tasks are categorized into three lists: Todo, In Progress, and Done.

Error Handling

Any errors that occur during API requests are logged to the console for debugging purposes.

Future Improvements

- Implement proper error handling and user feedback.
- Enhance user interface and styling.
- Add additional features, such as task details and due dates.

Login Component

The `Login` component handles user authentication. Users provide their username and password to log in.

State

- `username (useState)`: Represents the entered username.
- `password (useState)`: Represents the entered password.

useEffect Hook

Redirects the user to the home page if already authenticated.

Functions

`loginHandler()`

Authenticates the user by sending a request to the backend API.

Styling

The component is styled using Tailwind CSS for a clean and responsive design.

Error Handling

Displays an alert for any authentication errors.

Signup Component

The `signup` component allows users to register for the task management application.

State

- `username (useState)`: Represents the entered username.
- `email (useState)`: Represents the entered email.
- `password (useState)`: Represents the entered password.

Functions

`signupHandler()`

Registers the user by sending a request to the backend API.

Styling

Styled using Tailwind CSS for a modern and user-friendly interface.

Error Handling

Displays an alert for any registration errors.

URL Configuration

The `URL.js` file exports the backend URL for the application.

```
export const backend_url = "localhost:3020";
```