

Next.js Guided Project

Introduction:

Build Foodie, a simple, responsive food menu browser using Next.js with TypeScript. This application allows users to explore a variety of dishes and view their details. The app includes login as a standalone feature.

Styling (global CSS), layout (card-based and centered views), and TypeScript setup (interfaces, tsconfig.json with strict mode) are already provided in the source code. Your task is to complete the application logic and page functionality.

Features to Implement

1. Menu Page (/pages/menu/index.tsx)

- Fetch and display a list of food items using Server-Side Rendering (getServerSideProps).
- Render each dish as a card showing its:
 - Name
 - Price
 - Thumbnail image (use `<Image>` from `next/image`)
- Clicking a dish should take the user to the detail page of that dish.

2. Dish Detail Page (/pages/menu/[id].tsx)

- Use dynamic routing to render details for a specific dish based on its ID.
- Fetch data using `getServerSideProps`.
- Display:
 - Dish name
 - Full-sized image
 - List of ingredients

- Price
- Include a link to go back to the Menu page.

3. Login Page (/pages/login)

- Standalone login functionality with a form to enter username and password.
- Authenticate using hardcoded credentials:
username: admin, password: foodie123
- On successful login:
 - Store a dummy JWT token ("fake-jwt-token") in localStorage.
 - Redirect to the menu page.
- Login is not required to view the menu or details.

Technical Requirements

Already Provided:

- styles.css with global styles
- Centered layout and responsive card-based design
- TypeScript interfaces
- tsconfig.json with strict type checking enabled

You Must Implement

- Functional pages with routing:
 - /menu (SSR)
 - /menu/[id] (dynamic SSR)
 - /login (client-side auth)
- Use <Image> from next/image for all images.