

Learning Management System (LMS) Frontend

This is the frontend application for a Learning Management System built with Next.js, TypeScript, and Chakra UI.

Features

- User authentication and authorization
- Course management and enrollment
- Responsive design with Chakra UI and Tailwind CSS
- Real-time updates using SWR
- Type-safe development with TypeScript

Prerequisites

- Node.js (Latest LTS version recommended)
- npm or yarn package manager
- Backend API running (default: <http://localhost:8000>)

Installation

1. Clone the repository:

```
git clone [repository-url]
cd lms-frontend
```

2. Install dependencies:

```
npm install
# or
yarn install
```

3. Create a `.env.local` file in the root directory with the following variables:

```
NEXT_PUBLIC_API_URL=http://localhost:8000
```

4. Start the development server:

```
npm run dev
# or
yarn dev
```

The application will be available at <http://localhost:3000>

Project Structure

```
lms-frontend/
├── app/           # Next.js app directory
```

```
| components/      # Reusable React components
| context/        # React context providers
| lib/            # Utility functions and API clients
| types/          # TypeScript type definitions
| utils/          # Helper functions
| styles/         # Global styles and Tailwind CSS
| public/         # Static assets
```

Tech Stack

- **Framework:** Next.js
- **Language:** TypeScript
- **UI Libraries:**
 - Chakra UI (v2.8.2)
 - Tailwind CSS (v3.4.1)
- **State Management:** React Context + SWR
- **HTTP Client:** Axios
- **Authentication:** JWT

Testing

Refer to [TESTING.md](#) for detailed testing procedures and test cases.

Authentication

The application uses JWT-based authentication. Tokens are automatically handled by the middleware and stored securely.

Responsive Design

The application is fully responsive and tested across:

- Desktop browsers
- Tablet devices
- Mobile devices

Development Workflow

1. Create a new branch for your feature/fix
2. Make your changes
3. Test thoroughly using the test cases in TESTING.md
4. Submit a pull request

Available Scripts

- `npm run dev` - Start development server
- `npm run build` - Build production bundle
- `npm start` - Start production server
- `npm test` - Run tests (when implemented)

Contributing

1. Fork the repository
2. Create your feature branch
3. Commit your changes
4. Push to the branch
5. Open a pull request



License

ISC License



Support

For support, please open an issue in the repository or contact the development team.