

Project Documentation: Subdomain-Based Customization in Next.js

Overview

This project demonstrates how to build a Next.js web application that utilizes subdomain-based routing for customization. Each subdomain corresponds to a distinct set of configurations, including color schemes, content, and images. The application dynamically renders content based on the accessed subdomain.

Components

1. Pages

- **Page Component:** Renders customizable content based on the subdomain, including title, description, and image.
- **RootLayout Component:** Defines the layout structure for the entire application, including HTML structure and customization loading.

2. Configuration

- **subdomainConfig.ts:** Defines configurations for different subdomains, including color schemes, content, and images.
- **next.config.js:** Configures custom rewrites for subdomain-based routing.

3. Utilities

- **LoadCustomizations Function:** Loads customizations based on the accessed subdomain.

Implementation Details

Subdomain Routing

- Custom server middleware parses incoming requests to extract the subdomain.
- Next.js rewrites are configured to handle subdomain-based routing.

Customization Loading

- The `LoadCustomizations` function fetches customization data based on the subdomain.
- Customization data includes color schemes, content, images, and other styling properties.

Page Rendering

- The `Page` component dynamically renders content based on the accessed subdomain.
- Color schemes, content, and images are applied dynamically based on the loaded customization data.

Deployment

1. **Deployment Platform:** Choose a hosting provider that supports subdomains, such as Vercel, Netlify, or AWS.
2. **Deployment Process:**
 - Deploy the Next.js application to the chosen hosting provider.
 - Configure DNS settings to point subdomains to the deployed application.
3. **Testing:** After DNS settings have propagated, test the application by accessing different subdomains in a web browser.

Usage

1. Clone the project repository from GitHub.
2. Install dependencies using `npm install`.
3. Start the development server using `npm run dev`.
4. Access the application in a web browser using subdomains defined in the `subdomainConfig.ts` file.
5. Customize configurations in the `subdomainConfig.ts` file as needed.