Technical documents

1. Source Code Snapshot

Our project is broken down into three repositories:

Smp-frontend:

Front-End

This is the improved version of our Student meal planner that integrates ai for meal generation and finds a recipe that matches with the students ingredients to minimize food wastage. In section 5 you can see how to deploy it.

Contributors:

- Will
- Callum

Django backend:

Back-End

This shows the initial example of how we plan to host our student/house leaderboard based on how many sustainable meals the user has created . It is hosted on Django server alongside the database used to store all the students and their meals made and score.

Contributors:

- Morgan
- Kaylum
- Hector
- Adam

Student mealPlanner:

Student Meal Planner

This is the original prototype we had for generating meals based off of ingredients the user had . In order to make it more efficient we decided to incorporate ai instead of using apis to grab recipes and compare them with the users inventory.

Contributors:

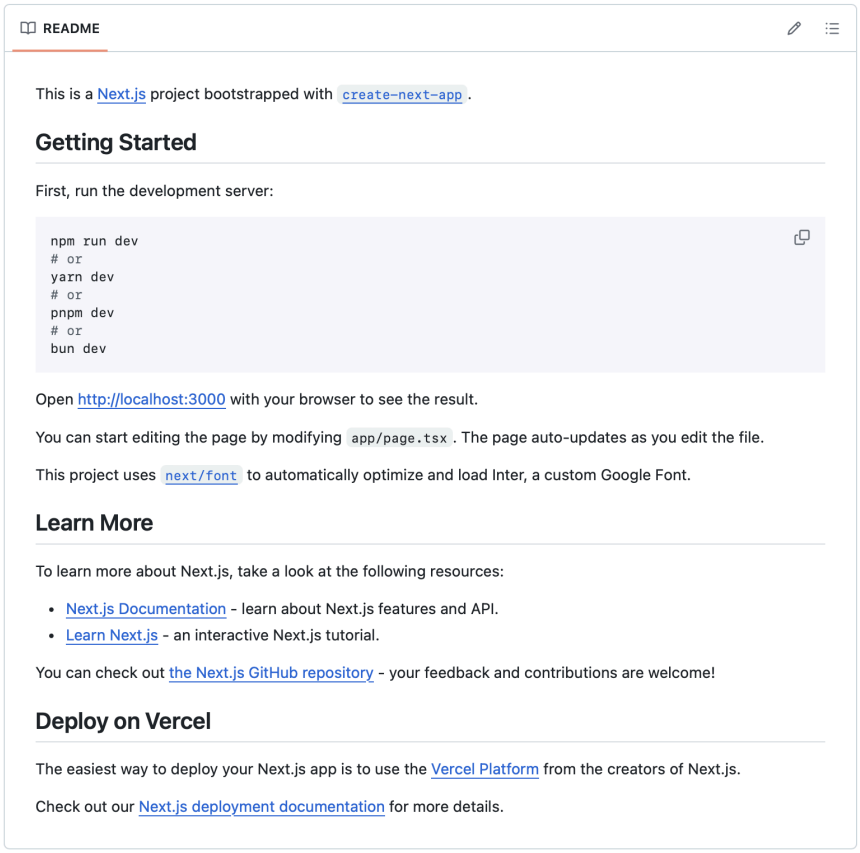
- All

Description:

The source code snapshot includes all the codebase required to build and deploy the web application for promoting sustainability on campus at the University of Exeter.

Contents:

- Link to GitHub repository: https://github.com/PImpliesQ/StudentMealPlanner
- Instructions for deployment:



README

This is a Next.js project bootstrapped with `create-next-app`.

Getting Started

First, run the development server:

```
npm run dev
# or
yarn dev
# or
pnpm dev
# or
bun dev
```

Open http://localhost:3000 with your browser to see the result.

You can start editing the page by modifying `app/page.tsx`. The page auto-updates as you edit the file.

This project uses `next/font` to automatically optimize and load Inter, a custom Google Font.

Learn More

To learn more about Next.js, take a look at the following resources:

- Next.js Documentation - learn about Next.js features and API.
- Learn Next.js - an interactive Next.js tutorial.

You can check out the Next.js GitHub repository - your feedback and contributions are welcome!

Deploy on Vercel

The easiest way to deploy your Next.js app is to use the Vercel Platform from the creators of Next.js.

Check out our Next.js deployment documentation for more details.

2. Coding conventions:

- Coding style: Google JavaScript Style Guide for frontend and PEP 8 for Python backend.
- A standard naming convention, coding convention and commenting was used throughout the project

3. Testing Strategy

To test our project we used a number of techniques such as unit testing, integration testing and general in house testing. We used unit testing to test different code on a smaller level such as django backend and smp-frontend, once each portion was complete we then performed integration tests to make sure that the whole project worked as a whole. We even had a small end user test where one person from outside the group tested the website to give us feedback on the user experience. All the tests were instrumental in refining the user interface and enhancing the overall user satisfaction with our project.

4. Developer documentation

This document shows the insight on how our project architecture, the languages used and coding convention followed. Whether it's on the technical document or on the github repositories, users and developers can find all the information needed to set up their version including necessary dependencies and configuring for the project structure. The document shows the directory structure of the project, showing the organization of frontend and backend code, database-related files, and documentation resources. Coding conventions used in the work are standard used in any coding practice and highlighted with comments to further explain and help readability. For the front-end of our project we used javascript and typescript and the back-end the languages used were python, django. Below i have also added our figma template which shows the the webpage overview helping further show the setup used for our project