

Mock-Synoptic-Project

Deployed App

The Application has been deployed to the Google Cloud Platform App Engine Service and can be accessed with the following URL:

<https://synoptic-project-restaurant.nw.r.appspot.com/>

Documentation

The Documentation can be found in the documentation folder at the root of this project.

Configuring the Application

The application is mostly config driven, so much of the text and values like the max number of a single item allowed per user can all be edited easily by editing the following file:

`/src/nuxt/config/app.config.js`

Further files/folders that contain meaningful config

- `/src/nuxt/config/` - This folder also contains further infrastructure config
- `/src/nuxt/nuxt.config.js` - This Folder Contains development config for the application but also contains the color scheme and font to use (Mainly should be left untouched)

Running Locally

This project consists of 2 parts: the Front End App (*Nuxt*) and the API.

Prerequisites

- Node Version 14 or greater and npm

Nuxt/Frontend App

Go to the `/src/nuxt`

Then we install the required dependencies

Run `npm install`

Once that finishes, we can then run the App in development mode by running

`npm run dev`

The Application should then be available at `http://localhost:3000/`

API

Go to the `/src/api`

Then we install the required dependencies

Run `npm install`

Once that finishes, we can then start the App by running

`npm start`

The API should then be available at `http://localhost:3300/`

You can test the API is working by going to `http://localhost:3300/ping` it should reply with pong

Caveats/Limitations

The Nuxt App is closely integrated with **Firestore** (*My Database*) and **Firebase Authentication** (*My Auth Provider*); both services provide their own API for their respective services. Therefore, Authentication is handled directly between the Client (Browser) and Firebase Authentication, and the data fetching is for the most part also managed directly; the database is not fully accessible publicly as I've added security rules to prevent unauthorized access. So, for example, a User can view the menu but it can not write to it. Therefore I created an API that the User (Frontend App) can Interact with so that the User can, for example, place an order, the API is then able to take the User's basket and create a new Order from it and it has permission to edit the menu, which in this case means decrementing the stock by the amount ordered by the User. However, for example, an Admin account, in the security rules has unrestricted access to the database, including the menu; therefore all admin actions can be handled directly between the browser and the Firestore database as on Google's own backend it verifies the User and their Auth Tokens/Permissions and allows them access to more features.

Therefore, because the Local API also requires unrestricted access, it can only run locally when provided with the appropriate service account from the Google Cloud Platform Identity and Access Management Service, which I can provide on request. The Local Nuxt App should work without issue as there is a flag in `/src/nuxt/config/endpoints` that forces it to use the API that I have deployed alongside this App, so it should run without a problem.

Luckily, the API is only responsible for the following features: - Booking a table as a User - Placing an Order (By Converting a User's Basket into an Order and editing the Menu Stock Levels)

This means that even if the API is unavailable, all other features should work without issue

Extra Info

I have provided **Admin Test Logins** in the documentation folder to be used locally or on the deployed App to test the Application further.

I also provided instructions on how to create your **own** firebase project and hook it up to this application (/Documentation/Deployment Instructions.md), where you can acquire your own Keys.

Difference between an Admin Account and a Service Account

- Admin Account - Has unrestricted to the database
- Service Account - Has unrestricted access to the whole firebase project (Which it required to verify auth tokens and edit the database)