That's eat

Manual

Marios Tsioutsis

01 TECHNOLOGIES

02 GETTING STARTED

03 FILE STRUCTURE

04 CUSTOM HOOKS

05 CONTEXT

TECHNOLOGIES 01

TECHNOLOGIES USED

1. React A JavaScript library for building user interfaces.
2. Vite a new kind of build tool for frontend web development.
3. Material UI MUI provides a robust, customizable, and accessible library of foundational and advanced component.
4. React Router a standard library for routing in React.
5. SASS is the most mature, stable, and powerful professional grade CSS extension language in the world.
6. Swiper is the most modern free mobile touch slider with hardware accelerated transitions.
7. Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.
8. MongoDB a classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

GETTING STARTED

02

GETTING STARTED

Installing all dependencies:

Yarn: yarn or yarn install.

Npm: npm i or npm install.

Add the PORT and the DATABASE_URI on the .env file of the server

Start the client side: yarn start_client or npm run start_client

Start the server side: yarn start_server or npm run start_server

Visit http://localhost:3100 on your browser.

FILE STRUCTURE

03

FILE STRUCTURE

This is a monorepo workspace. Contains the server side and client side on the corresponding folders.

On the server side there is a basic structure containing some database actions and a few routes.

On the client side the App.js is the main application, and the index.scss and variables.scss affect the global styles.

Every components is in a separate folder with its own scoped style file.

In the helpers folder there are some simple helping functions

In the screens folder we have the page views of the application, such as “Homepage”, “Stores List” etc.

In the hooks folder we have the custom hooks we created to achieve some functionalities.

In the store folder we have everything about the context of the application. Such as the “shopping cart”, “stores” etc.

CUSTOM HOOKS

04

CUSTOM HOOKS

The only custom hook used so far is a hook to fetch the data from the database.

A nice future addition will be a custom hook to save some information on the local or session storage of the browser.

CONTEXT 05

CONTEXT

Currently we are using 3 providers with react context hook.

A simple one that keeps the address of the user in the state.

A more complex one that fetches and manages the stores from the database.

And an another one that has all the functionality for the shopping cart. In this one we have the state of the cart and it’s core functions such as: add to cart, update item, remove from cart etc.

THANKS!