Food Order App

Project

</talentlabs>

Introduction:

Food order app is the web application that allow users to select the food item and add them to the cart. The users can also delete the food item from the cart if they accidentally added them. As we know that all kinds of applications are managed by admins. So, in this application I have also implemented the functionality for the admins. Basically, admins can add the new food items and delete them if they are not available for users to buy.

Goals:

The goal of this assignment is to create a website allow:

- Users to select their favorite food items from the available food items.
- After selecting the food item, the item will be added to the cart.
- User can view the cart.
- User can also remove the items from the cart if he accidently added them add can also increase their quantity.
- Admins to add new food items to the menu.
- Admins can also delete the food item from menu, if the item is not available.

Specification:

Our web application is divided into two portions. One portion is for Users and other is for Admins. As I explained above Users can simply select the food items from the available food items and admins are the one who controls our application. The admins can add new items to the menu and can remove the item if it is not available.

Before you start:

Before you start, I would suggest getting familiar with:

1. Some React basics like:
   - React state management using useState and useReducer hook.
   - Handling side effects in react.
   - Context API.
2. Material UI. It is a third-party library that will help use to build the user interface. It basically gives us components so we can use the in our application. So, I suggest to get familiar with material UI components like Grid, Appbar, Toolbar, Card and Button.

Tools and libraries to use:

1. **React**
   - We will be using functional components to build this application.
   - We will be using context API to manage overall state of our application.
   - We will be using some react hooks. (useState, useEffect, useReducer, and UseContext).
   - We will be using react portals (a feature that allow us to create cart modal).
   - We will be using react Fragments.

2. **Material UI:**

   We will be using some of material UI components like Appbar, Toolbar, Grid, Card and Button.

Steps to create the application:

To create the React application you need to install Node.js. You can watch any video on YouTube that will guide you on how to install Node.js.

After installing Node.js you open terminal in the directory where you want to create the application.

Now type

$ npx create-react-app food-order-app

It will take some time to create the application. After it is done navigate to the project's directory and by typing.

$ cd food-order-app

After that open the project in your code editor. You can also type the command below to open the project in your code editor.

$ code .

Now, open the terminal of the code editor and install third party libraries by using the command below. We are going to use version 4 of material UI.

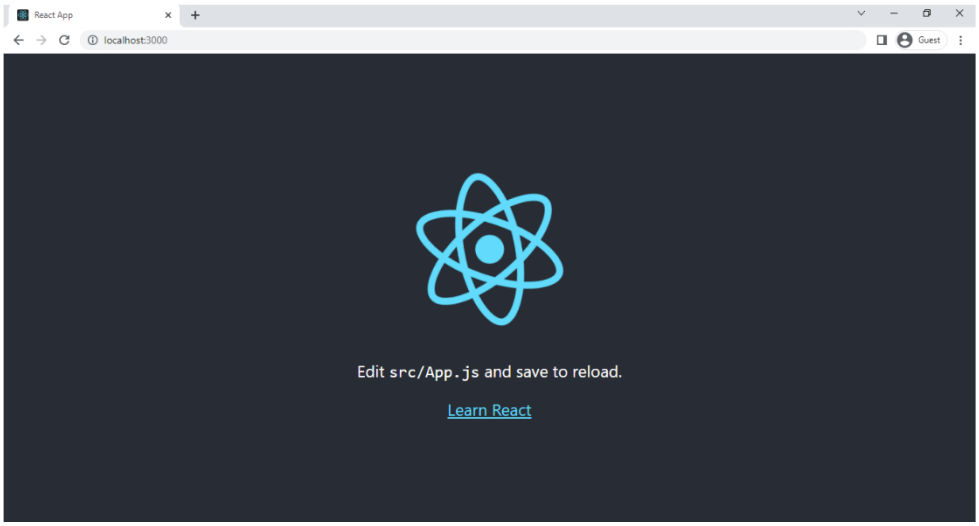
$ npm install @material-ui/core

Now, it is the time to start the application locally. You can start the application by using

$ npm start

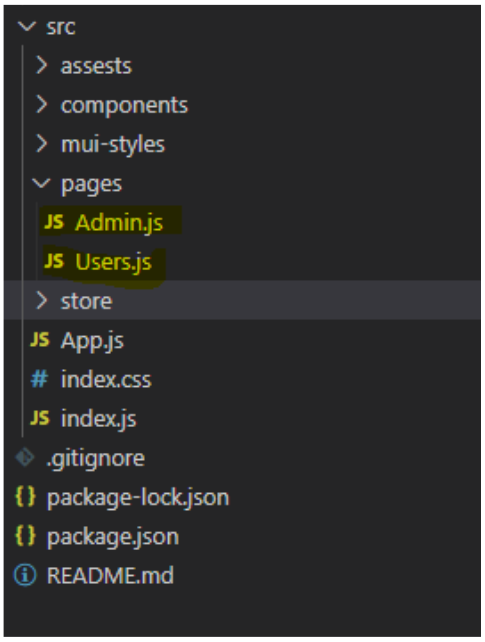
Now you can go to your favorite browser and type localhost:3000 to run the application in your browser.

In the beginning the app will look like this.

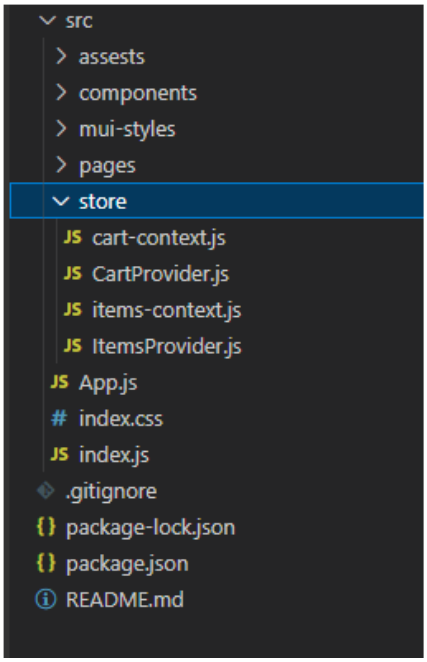


Now, let's start working on creating the food order app.

We will start by removing all the previous code from react the App.js file. Now it's time to create two pages one is for user and other is for admins.



As we know that we will be using Context API in our project. With context API we can pass data to different components without using props.



So we are going to use two context one is cart-context for managing the cart data.

```
import { createContext } from "react";

const cartContext = createContext({
  items: [],
  totalAmount: 0,
  addItem: (item) => {},
  removeItem: (id) => {},
});

export default cartContext;
```

And other is items-context for navigation and for adding and removing items data.

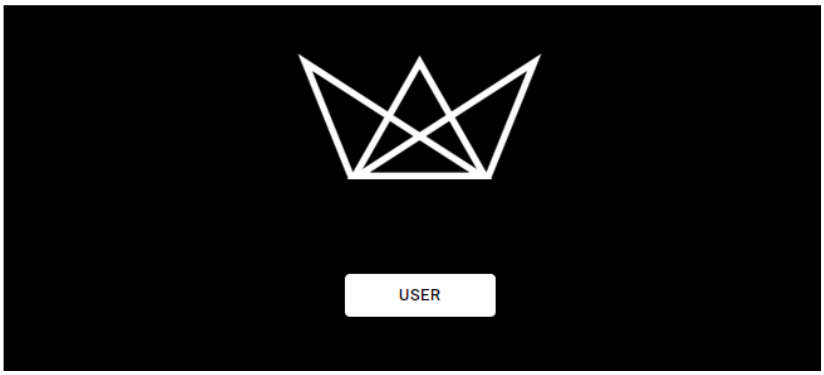
```
> store > JS items-context.js > ...
1   import { createContext } from "react";
2
3   const itemsContext = createContext({
4     itemsData: [],
5     switchPage: null,
6     addNewItem: (item) => {},
7     removeItem: (id) => {},
8     updateItem: (id) => {},
9     tooglePage: () => {},
0   });
1
2   export default itemsContext;
3
```

Now, inside our App.js file we have to use the logic navigation and also the footer component.

```
function App() {
  const itemsCtx = useContext(itemsContext);
  return (
    <Fragment>
      {itemsCtx.switchPage ? <Users /> : <Admin />}
      <Footer />
    </Fragment>
  );
}
```

Footer Component:

The user can navigate between the admin and user portion using the button given in the footer.



User's Page:

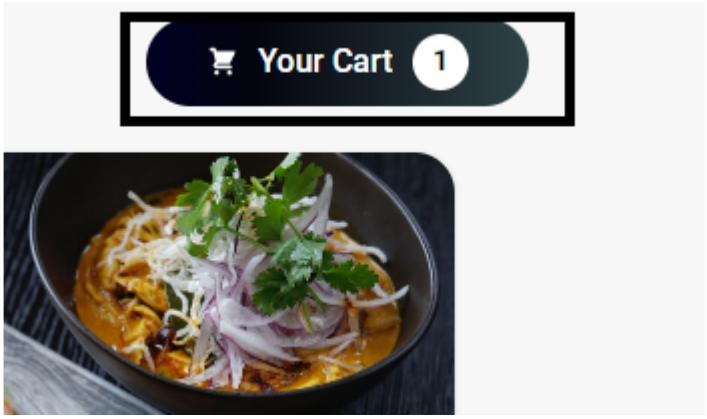
Now, let's explore User's page first.

```
const Users = () => {
  const [Isvalid, setIsVaild] = useState(false);
  const showModalHandler = () => {
    setIsVaild(true);
  };
  const hideModalHandler = () => {
    setIsVaild(false);
  };
  return (
    <CartProvider>
      <Appbar modleHandler={showModalHandler} />
      {Isvalid && <Cart hidemodleHandler={hideModalHandler} />}
      <Banner />
      <section>
        <AboutUs />
        <div style={{ marginTop: "15rem" }} className="top-margin">
          <Meals />
        </div>
      </section>
    </CartProvider>
```

Here we are using Appbar, Banner, AboutUs and Meals components wrapped inside CartProvider component so that all the data related to cart will be accessible inside these components.

Appbar component: The Appbar components contains the cart button. We will be using material UI AppBar component to build our appbar component. For that we need to import AppBar and Toolbar components from material UI.

```
import { AppBar, Toolbar } from "@material-ui/core";
import { useContext } from "react";
```



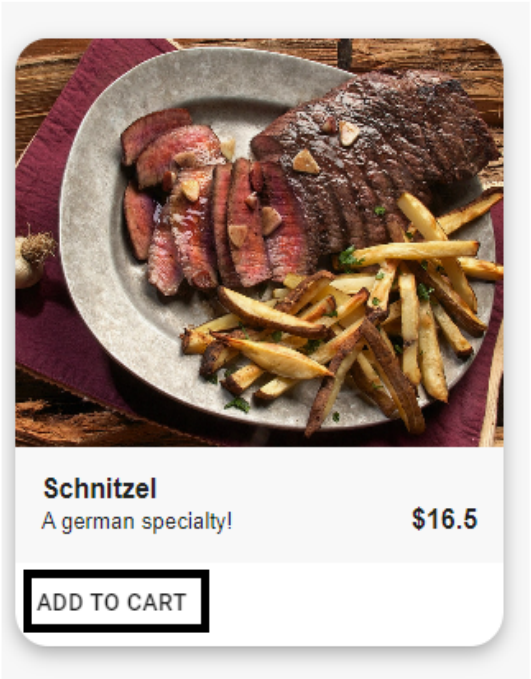
MealsItem Component:

User portion has mealsItem component which show the details about the food. To build this component we will be using material UI card component. So, we will be importing several different components from material UI to complete our card component. The image below shows that components that we are importing from material UI.

```
import {
  Card,
  CardActionArea,
  CardMedia,
  CardContent,
  CardActions,
  Button,
} from "@material-ui/core";
```

The mealsItem component has the “Add to cart” button in it. The user can add new food item to the cart by clicking that button.

When a user clicks on add to cart button a new item is added to the cart. You can see the Add to cart button in the image bellow.



Cart component:

User can open the cart by clicking on the “Your Cart” button. The cart component is model we can build models in react by using react portals.

The image below is the image of the cart. Now, user can delete the item from the cart by clicking on the “negative” button and can also increase their amount by clicking on the “positive” button.

The User can close the cart by clicking on the “Close” button. The “Order” button is just the random button it is not doing anything.

Admin Page:

Admin page has two components Sidebar and Main component.

```
const Admin = () => {
  return (
    <div className="admin-main-container">
      <Sidebar />
      <Main />
    </div>
  );
};
```

Main Component:

Main component has Admin and Meals component.

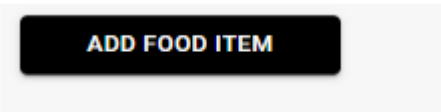
```
const Main = () => {
  const classes = componentStyles();
  const [Isvalid, setIsVaild] = useState(false);
  const showAddItemForm = () => setIsVaild(true);
  const hideAddItemForm = () => setIsVaild(false);

  return (
    <Box style={{ flex: 4 }}>
      {Isvalid ? (
        <AdminForm hideAddItemForm={hideAddItemForm} />
      ) : (
        <Button
          variant="contained"
          onClick={showAddItemForm}
          className={classes.addItemsButton}
        >
          Add Food Item
        </Button>
      )}
      <Meals />
    </Box>
  );
```

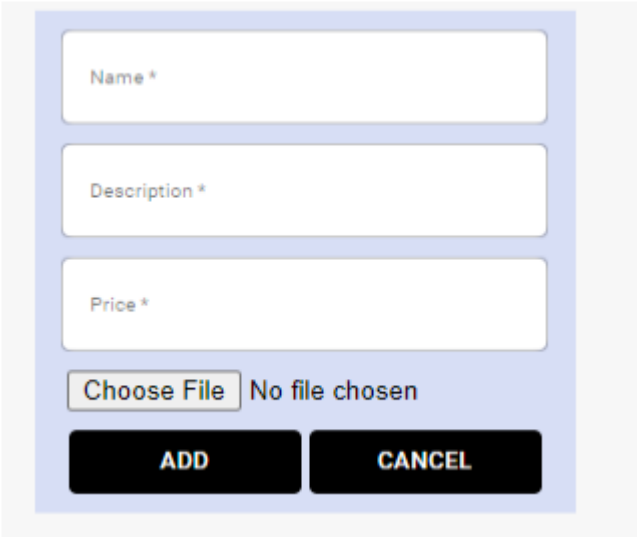
Form component:

In Admin's portion we have "Add food item" button. This button allows admins to open a Form component and add new food items. To create a form component, we will be importing some components from material UI.

```
import { Box, Button, TextField } from "@material-ui/core";
// user imports
```



When admin clicks on that button a form will be opened which allow admins to enter data about the new food item.



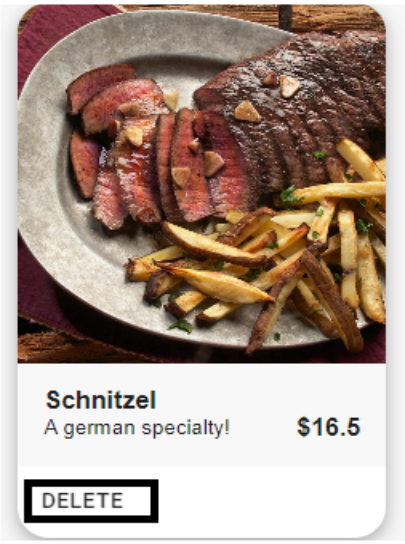
When admin is saving an item with an image, you will need to store the image somewhere online, so you can retrieve it later. We suggest to leverage freeimage.host for this purpose. They provide a free API for uploading image and retrieving images.

To understand the usage of it, you can refer to the links below:

Freeimage.host: https://freeimage.host/
Freeimage.host API: https://freeimage.host/page/api

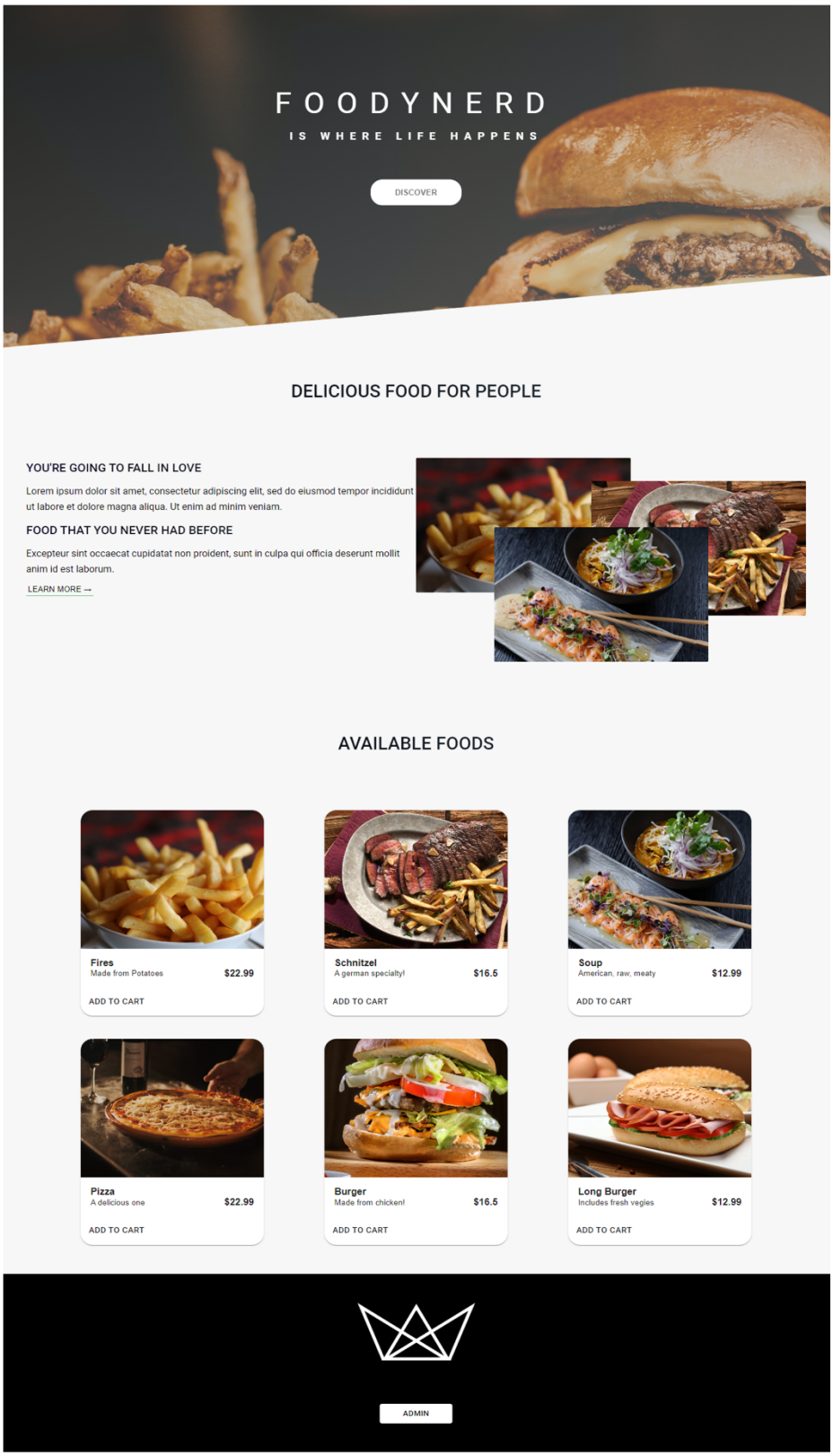
MealsItem Component:

Admin portion also has mealsItem component which show the details about the food. The mealsItem component has the delete button in it. The admin can delete the food item by clicking on the delete button.

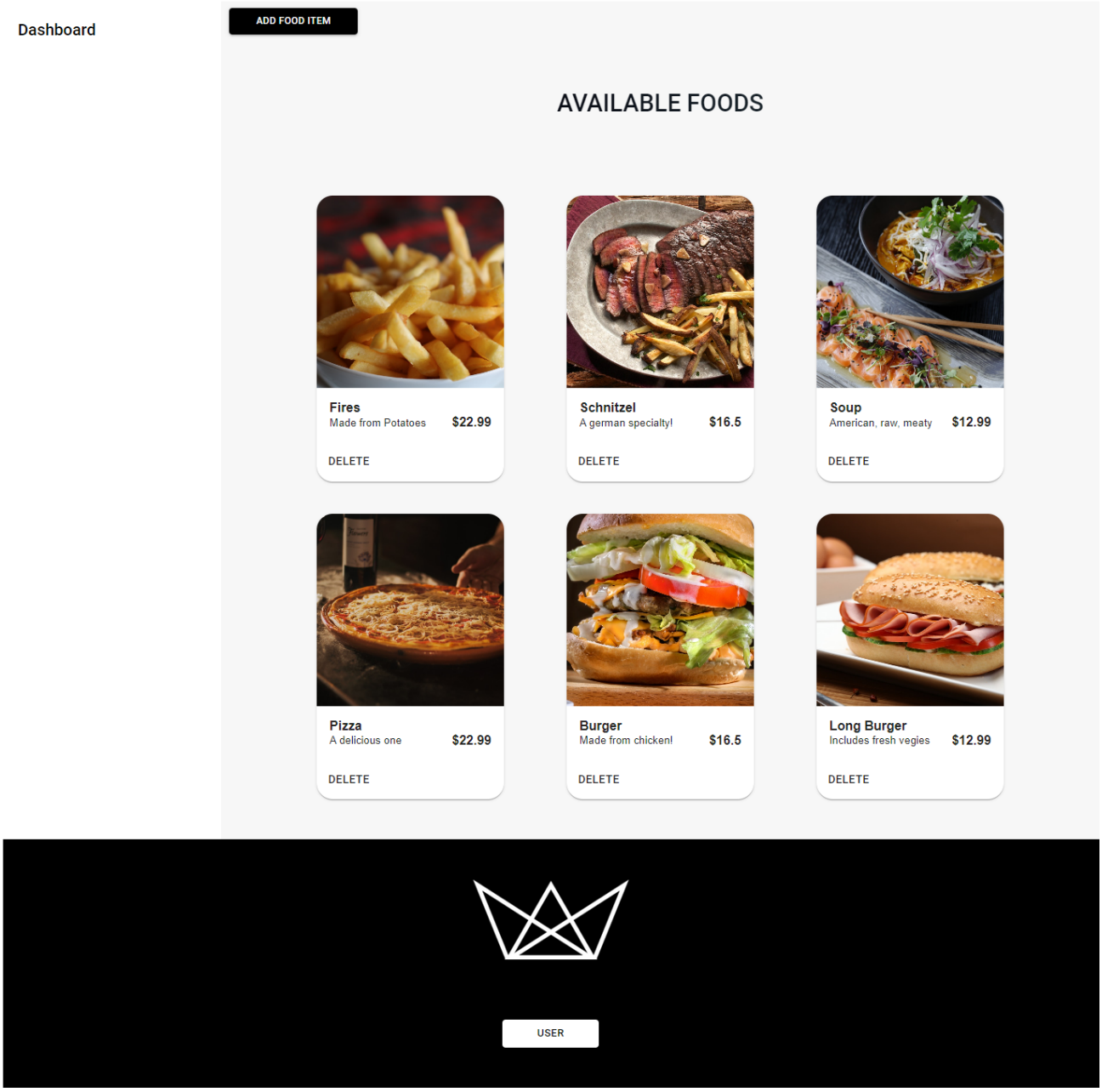


Layout:

User's Portion:

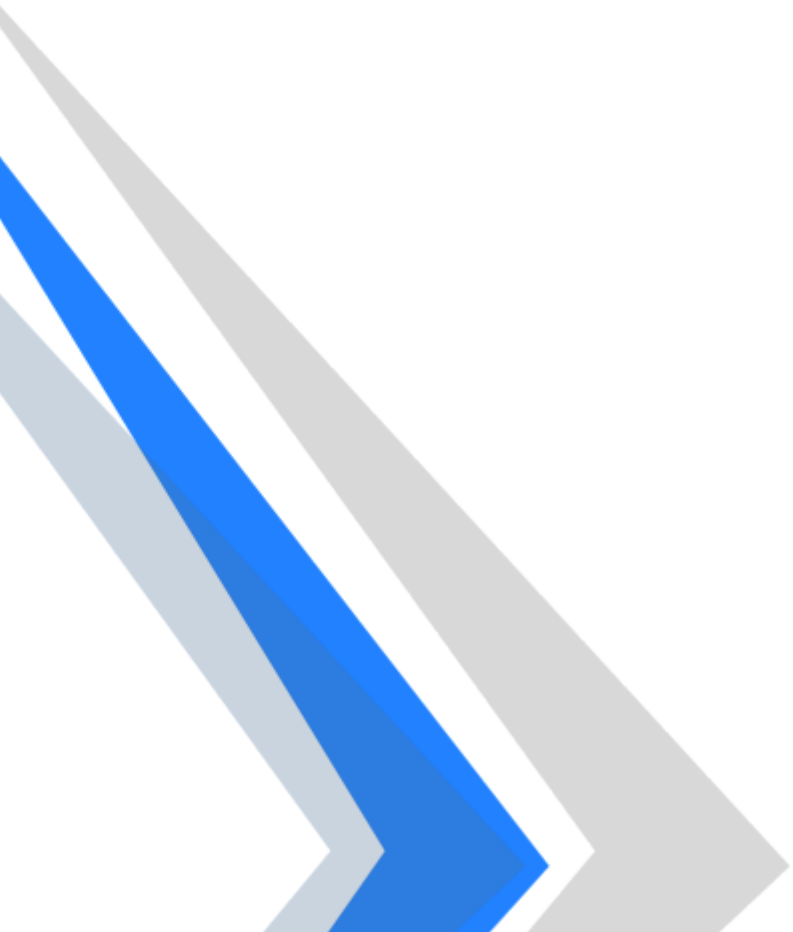


Admin's Portion:



Thank you.

All The Best :)



</talentlabs>