Venkata Praneeth Mummaneni
UML ID: 01662636
UML Email: venkatapraneeth_mummaneni@student.uml.edu

BURGER BUILDER

Venkata Praneeth Mummaneni
UML ID: 01662636
UML Email: venkatapraneeth_mummaneni@student.uml.edu

Abstract

This Project involves creating a Web App based on the Internet and Web Systems 1 course which would allow people to order a burger with certain ingredients. People can make profiles and can also view their past orders. This Web App is optimized for mobiles too.

UML ID: 01662636
UML Email: venkatapraneeth_mummaneni@student.uml.edu

Contents

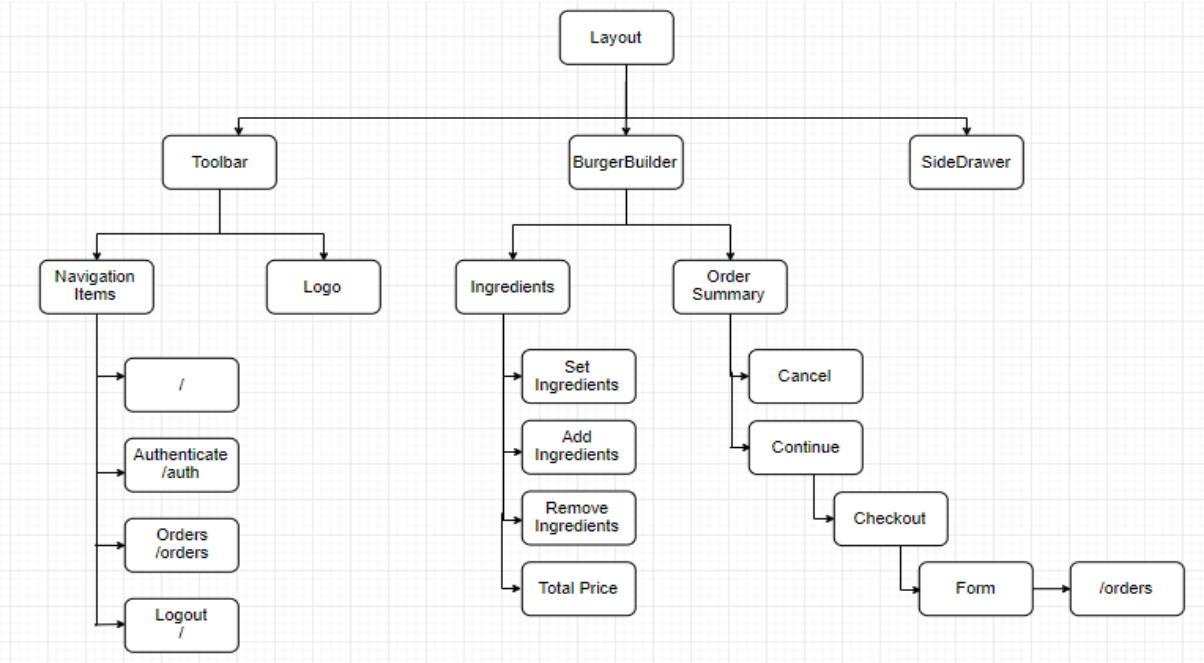
Venkata Praneeth Mummaneni
UML ID: 01662636
UML Email: venkatapraneeth_mummaneni@student.uml.edu

Introduction

This project uses HTML, CSS, ReactJS, Redux and Google Firebase. HTML, CSS and ReactJS were used for Frontend development. Redux was used to store states (of BurgerBuilder, Authentication and Orders) in a common container and make them available for different components of the app. Google Firebase was used to provide Authentication and used to store data in JSON format in its Real Time Database. The app is hosted on Firebase too. Some packages were also used to perform various functions in React. Axios package was used to send GET/POST requests to Firebase end points in order to get data from its database or authenticate a user. React-router-dom was used for dynamic routing.

Link to project: https://react-my-burger-4d77a.firebaseapp.com/

Working



The Layout of this project consists of 3 main components: Toolbar, BurgerBuilder and SideDrawer. The Toolbar consists of a horizontal bar where the burger Logo and Navigation items are present. The Navigation items are buttons and are clickable. Only the Authenticate and the BurgerBuilder buttons are present if the User is not authenticated. Authenticate button leads to /auth when pressed where you can enter your email and password to Log in or Signup. Axios is used to send a POST request to a Firebase URL for Log in/Signup. If the user is already authenticated, in place of the Authenticate button, Orders and Logout buttons appear. Each user has a unique UserID and gets a token when he/she logs in. This token lasts for 1 hour. Orders leads you to the previous orders of the User, if any. Axios sends a GET request to Firebase to fetch these orders for that particular User. Logout button safely logs you out when pressed.

The BurgerBuilder contains the Ingredients and OrderSummary components. You have a layout for top part of a bun and the bottom part of the bun. There are buttons called more and less to add ingredients (salad, cheese, bacon and meat) to the burger or remove ingredients respectively. There is also a total price element which dynamically reflects the price of the burger when components are

added or removed. If the user is not authenticated, he/she will be greeted with a “SIGN UP TO ORDER” button which will taken them to the /auth form to Signup or login. If the user is already authenticated, the button turns to “ORDER NOW”. This button is only active if ingredients are added to the burger. Once the “ORDER NOW” button is clicked, the Order Summary component comes up which shows a confirmation for the Price and ingredients present in your burger. If the “CANCEL” button is clicked on the Order Summary component, we are routed back to the BurgerBuilder, but, if the “CONTINUE” button is clicked, it leads us to the /checkout page where the User can see the image of the burger, he/she has ordered. This page has “CANCEL” and “CONTINUE” buttons beneath the burger. If the “CANCEL” button is clicked, we are routed back to the BurgerBuilder, but, if the “CONTINUE” button is clicked, it leads us to the contact-data form where the User must fill up details such as Name, Street, Zipcode, Country, Email and Delivery Method (Cheapest or Fastest). If the “ORDER” button (present below the form) is clicked, the order and contact data are sent through Axios POST request and stored in the Firebase Database. Each order has a unique key and is associated with the User’s userID. You can see this new order in the Orders tab of the Toolbar.

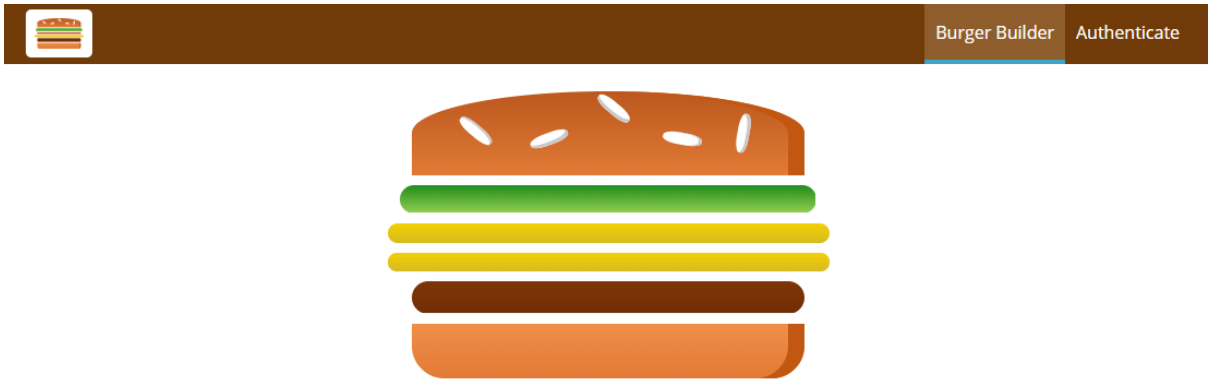
SideDrawer is a functionality for mobile devices. The SideDrawer component becomes active on screens below 500px width. When the SideDrawer is active, buttons on the toolbar vanish and three ‘-’ appear horizontally on the Toolbar on its left. When those are clicked, the SideDrawer opens and the original buttons present on the Toolbar are present on the SideDrawer.

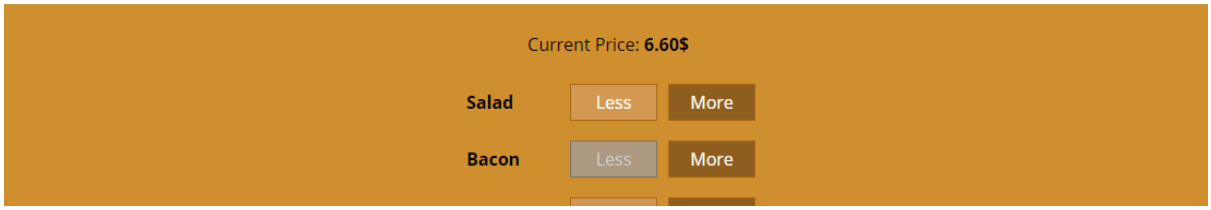
Progress

Weekly Progress Reports:
https://github.com/praneethmv17/Food_Builder/blob/master/Progress_Report.txt

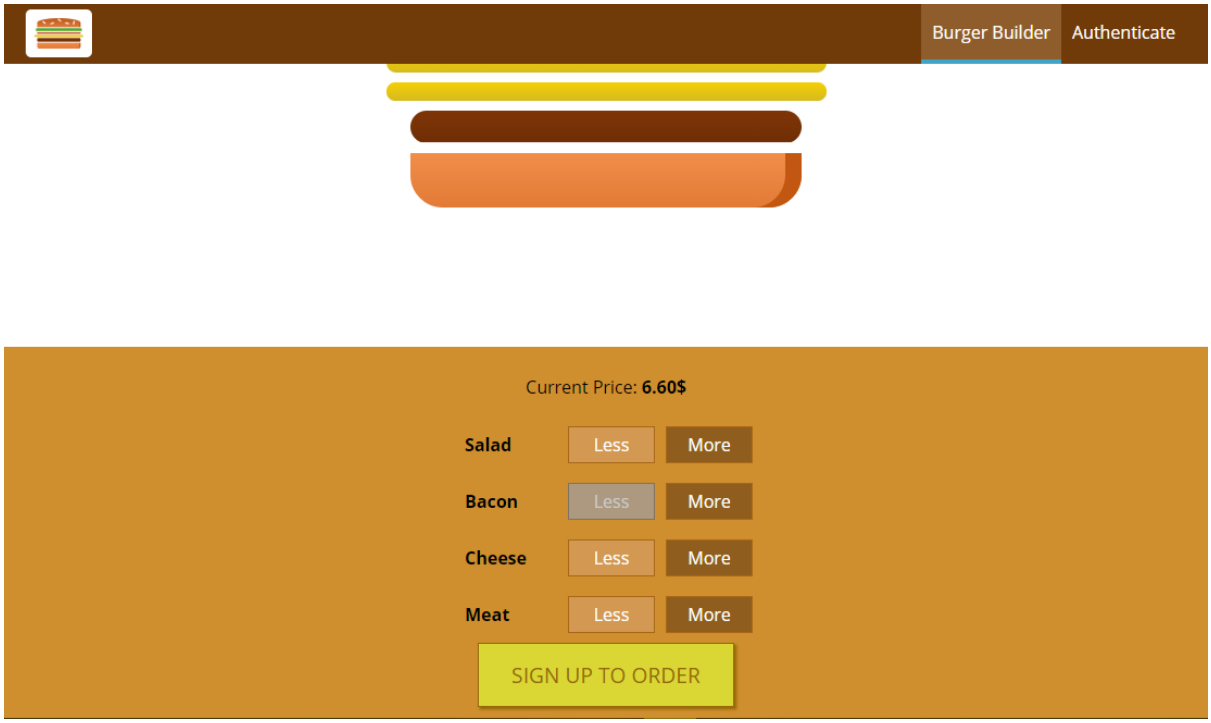
Results

BurgerBuilder page:

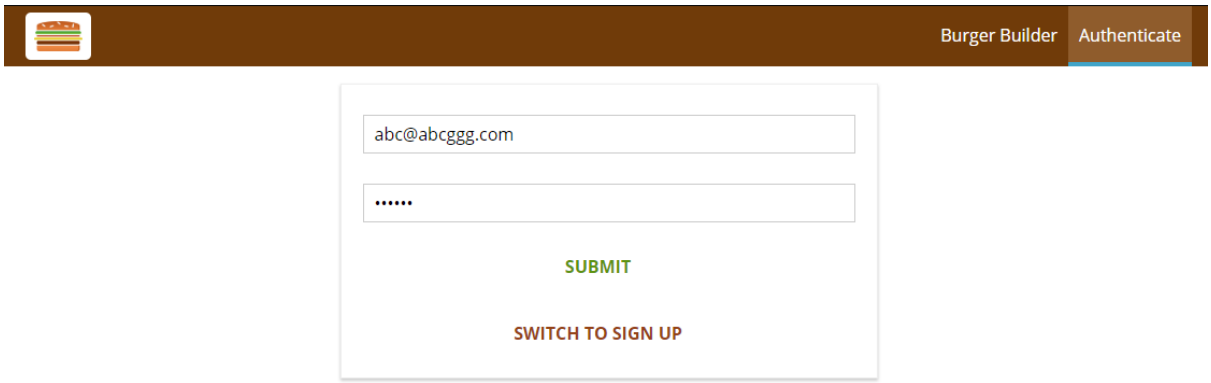




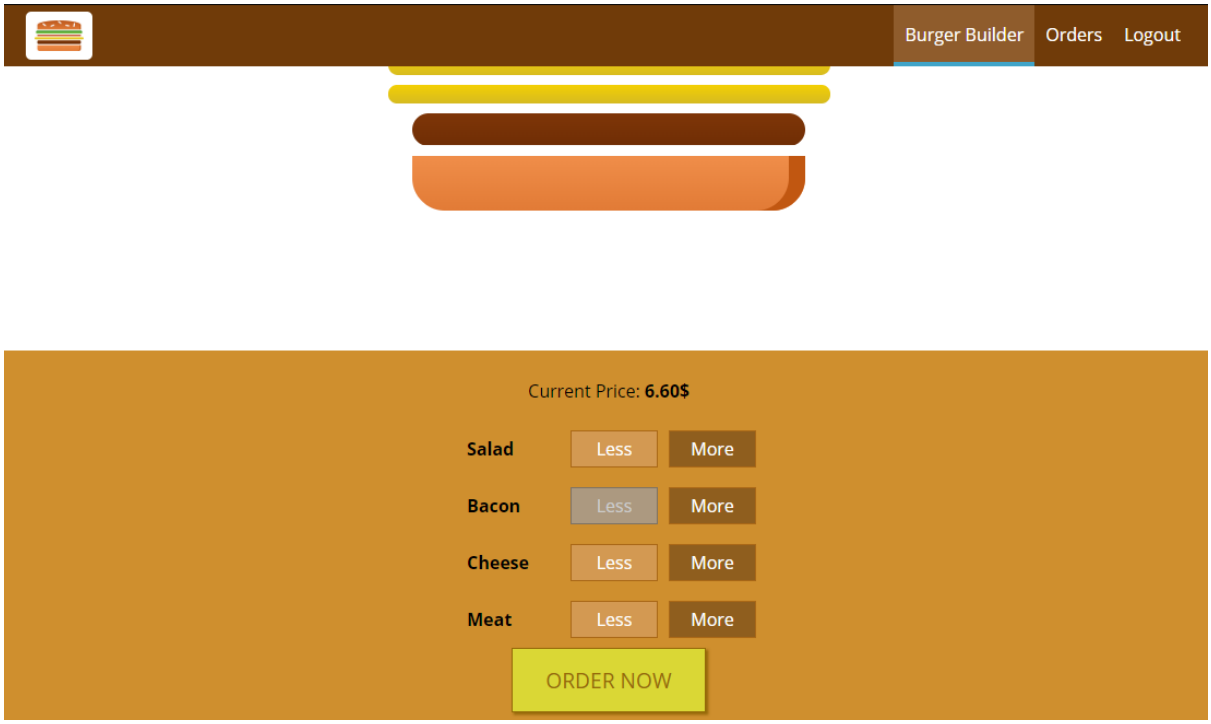
BurgerBuilder page(continued):



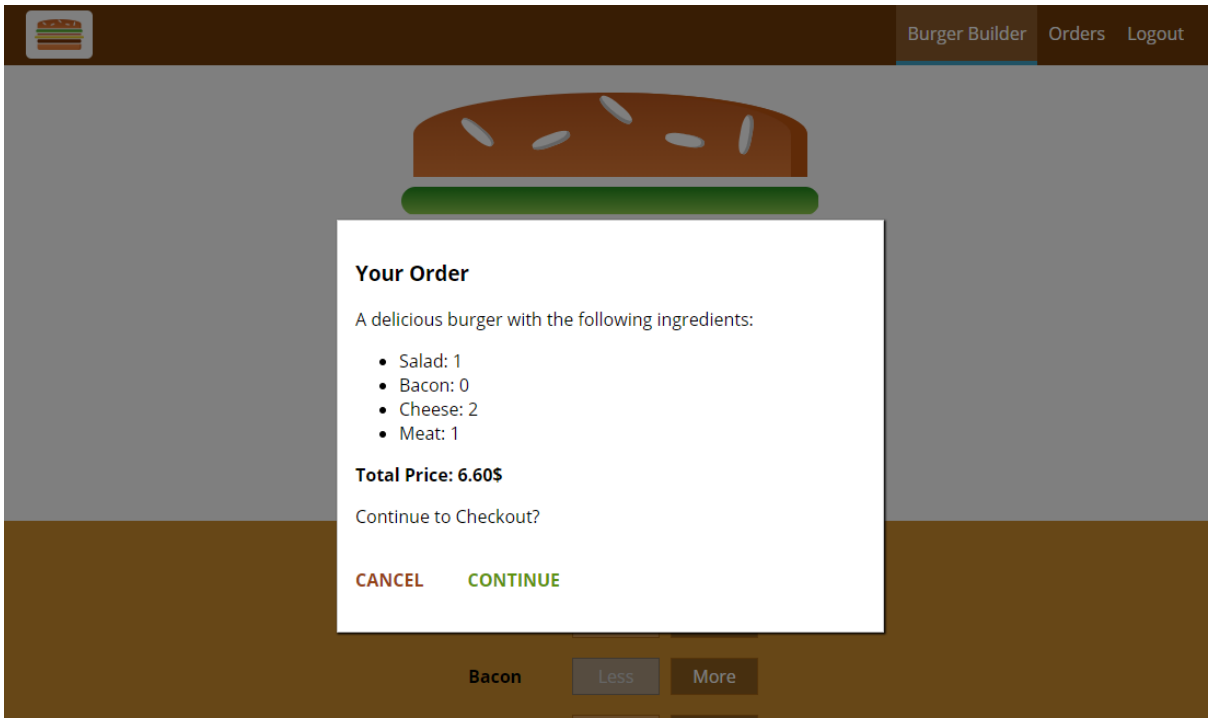
Authenticate page:



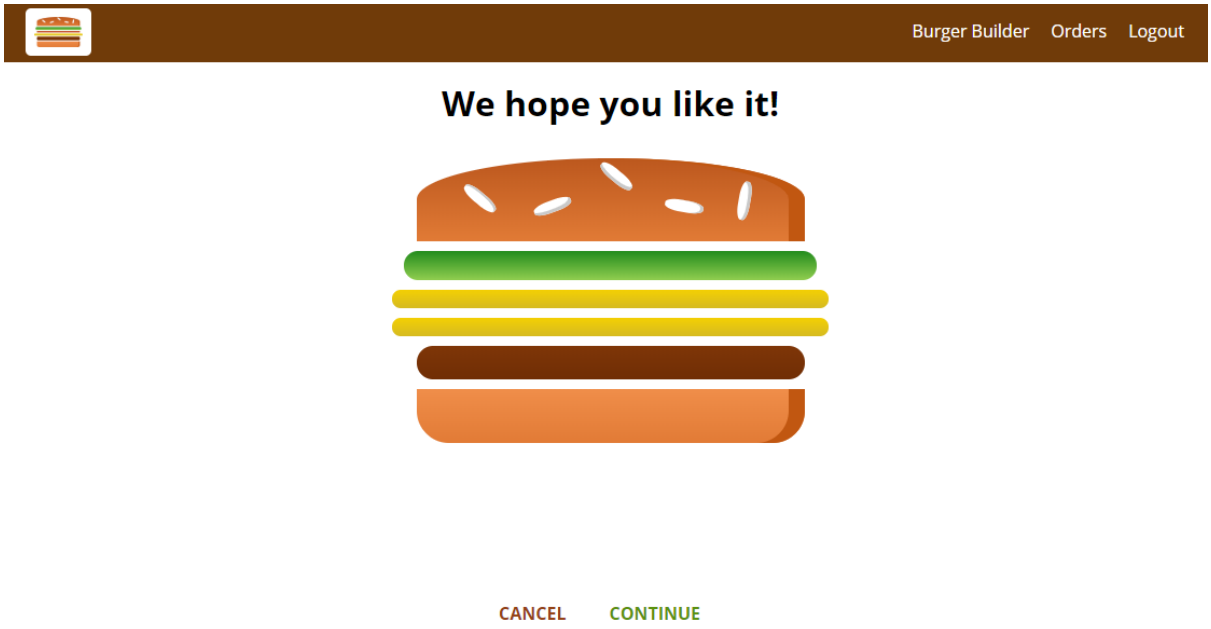
BurgerBuilder after User is authenticated:



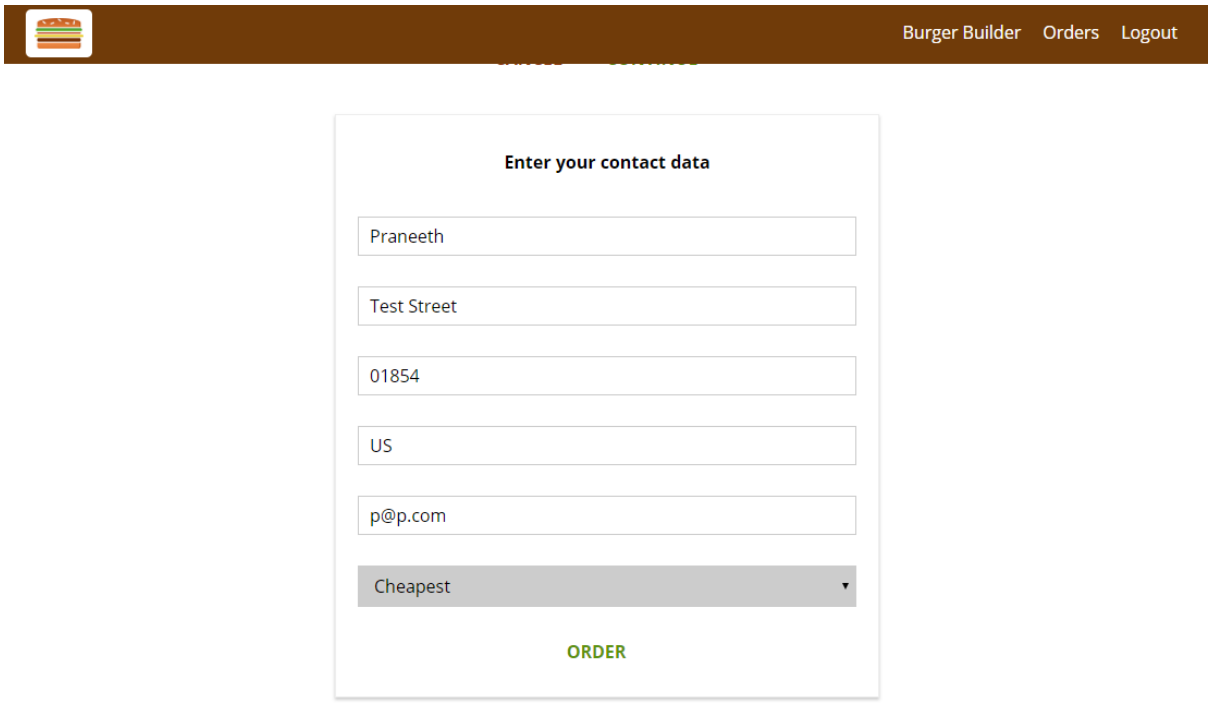
OrderSummary:



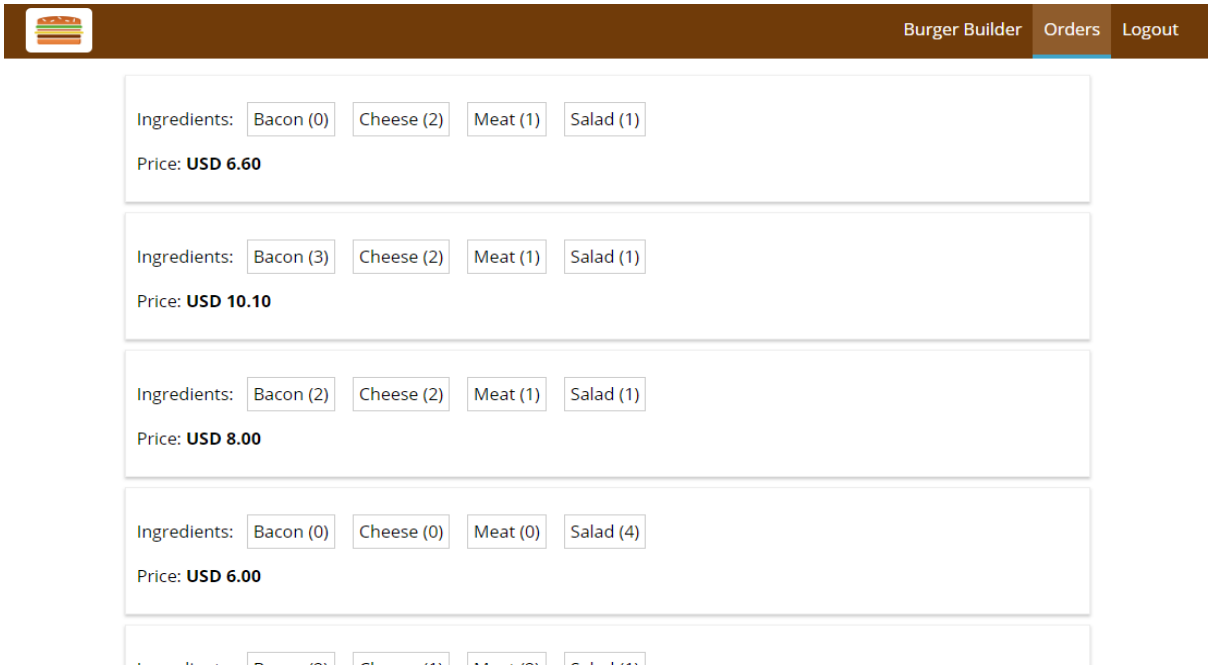
Checkout page:



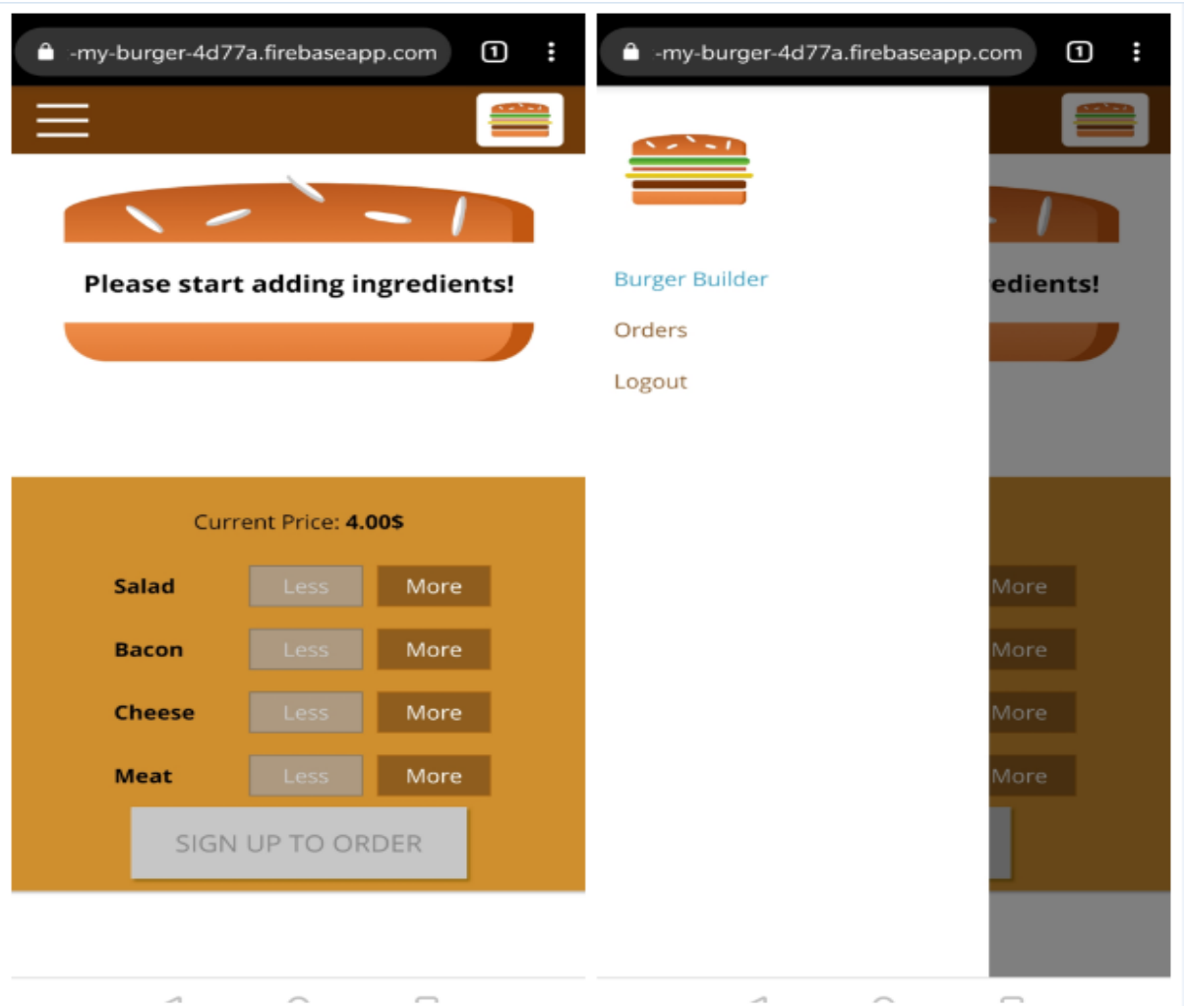
Contact-data page:



Orders page:


Burger Builder
Orders
Logout
Ingredients: Bacon (0) Cheese (2) Meat (1) Salad (1)
Price: USD 6.60
Ingredients: Bacon (3) Cheese (2) Meat (1) Salad (1)
Price: USD 10.10
Ingredients: Bacon (2) Cheese (2) Meat (1) Salad (1)
Price: USD 8.00
Ingredients: Bacon (0) Cheese (0) Meat (0) Salad (4)
Price: USD 6.00

BurgerBuilder and SideDrawer on mobile device:



Discussion

Through this project, I was able to understand the concepts of Internet and Web Systems a lot better. I learnt new technologies such as ReactJS and Redux. Learning Front end technologies was a new challenge for me. It took me a lot longer than I expected to learn React.

Conclusion and Future Work

I was able to create a well-organized ReactJS Web App. I hope to use my knowledge of ReactJS that I have achieved to make a full stack application in the future.
I want to further improve the Burger Builder Project by including my own Authentication instead of relying on Firebase and also, I would want to be able let Users delete their orders as well as their profiles.

Acknowledgements

The work described in this paper was conducted as part of a Fall 18 Internet and Web Systems 1 course, taught in the Computer Science department of the University of Massachusetts Lowell by Prof. Haim Levkowitz.

References

- https://reactjs.org/
- https://medium.freecodecamp.org/all-the-fundamental-react-js-concepts-jammed-into-this-single-medium-article-c83f9b53eac2
- https://www.udemy.com/react-the-complete-guide-incl-redux/
- https://firebase.google.com/docs/
- https://www.npmjs.com/package/axios
- https://hackernoon.com/tutorial-how-to-make-http-requests-in-react-part-3-daa6b31b66be
- https://redux.js.org/basics/usagewithreact
- https://medium.freecodecamp.org/understanding-redux-the-worlds-easiest-guide-to-beginning-redux-c695f45546f6
- https://reacttraining.com/react-router/web/guides/quick-start
- https://www.npmjs.com/package/react-router-dom
- https://projects.lukehaas.me/css-loaders/

(NOTE: I was not able to deploy my App on Weblab server even after numerous tries due to configuration issues, so I have deployed it on Google Firebase)