We implemented a mobile order system for ordering boxes of specific varieties of foods. The system allows customers to order these foods and stores those subscription-orders in a database. The system will also retrieve data on the order so the user can see what they've ordered and the details for that order, similar to the concept of a receipt.

An order placed by a customer can either be a one-time purchase or a subscription. The subscription details can be set by the user, such as the frequency of delivery and size of the box accounting for the varying amounts of people it will be serving. A utility for managing a user's subscription will also be provided.

Within the backend there were 2 main files that were used. Server.js encapsulated the implementation of Express and Mongoose (for use with MongoDB). It connected to the database and created routes for the front end and the queries.js file to use. Queries.js contains queries to the mongoDB database (such as login and signup) to be used throughout the application. For the backend, we had to install cors and mongoose and node dependencies to avoid issues running the frontend and backend concurrently and connecting to the database.

There were numerous new React components we had to build for this application. As any application with a user functionality, there needed to be a login and signup page, both of which were separate components that interacted with the backend and the database to authenticate users. The home page encapsulated the component "Product List" in order to hide the functionality of the app unless the user was logged/signed in. The subscription and orders components were used to let the user add a subscription and see their orders respectively. The subscription page hit an api endpoint that created a new entry into the subscription database in our MongoDB cluster. The orders page hit the same endpoint but with a get request to see the orders that a given user had made. Finally the profile page displays the information about the user and allows the functionality to logout of the application. On the frontend, we opted to use the bootstrap framework as it seemed like the most simple and widely used. It made it very easy to create the structure of our navigation bar and other styling.