Good Morning and welcome to my project 3 Ecommerce MERN stack presentation. I will be presenting an application that will allow any store, regardless of size or product sold, the ability to provide an online experience for consumers in spite of the current or future situations. Having seen so many small retail stores close when quarantine began, my focus was driven to attempt to provide a flexible, low cost, self-manageable e-Commerce Product.

MondoDB, Express, React and Node are the 4 key technologies that makeup the MERN stack in turn allowing us to easily construct a 3-tier architecture to combine the frontend, backend and database.

The **Client-Side** tier or frontend is JavaScript’s framework of React.js in conjunction with HTML and CSS, it allows us to create a dynamic web application contributing to a more enjoyable user experience with greater functionality. Though there was not much time spend on the UI, I would like to mention that I incorporated a grid css layout which is a 2 dimensional layout structure rather than the hacky 1 dimensional flex-box (demo)

The Logic tier which runs inside a NodeJS server allows for ExpressJS to provide for routing and handling of requests and responses through the use of its powerful modules.

This then leads us to the Database tier which is where our application will store all its information regarding inventory, pricing and user profiles.

Upon arriving at the site, a user will be provided the ability to create a profile which will allow them to shop at your store. **(signup):** We then click sign up and are immediately taken to the shopping list. While in the signup process, if I enter a duplicate email, I will be notified that the email is already in use. Should I enter an invalid email, I will be advised of the email requirements as well.

(Login) Once I am a registered user, I can log in with the use of the personal email and the password previously entered to access the shopping dashboard. Keeping in mind that the application is a MERN Application, the appearance along with colors, logo and menus provided can be adjusted without impacting the performance and functionality of the site.

(Dashboard) As a registered customer and while in the dashboard a list of items available are displayed where we will see a photo of the item, along with detail pertaining to the item. You will also see a warm greeting addressing the user which allows them to easily identify their logged-on status.  Should I wish to see more details on a specific item I simply click on the image for more detail including whether it is in stock.  By clicking the “Add to Cart” button, (Add to cart) we add the item to our shopping cart and we are provided an alert indicating that it has been added. The number on the cart link will then be updated as you can see here.  We discussed some of the dynamic features provided by MERN, allowing me to demonstrate the mobile responsiveness of the site. You can see how well the application easily adjusts to the various screen sizes.

(Cart) When the cart option is displayed, you will see a list of items previously selected being displayed in the cart. If I added several quantities of an item, that will also be reflected in the quantity displayed.

**(future)** Let me tell you a little about what is to come: I hesitantly entered into this project alone so that I might see what I was truly capable of after our boot camp experience and I was pleased to see that I managed to complete a large portion of the application which includes the use of ReactJS, Node and Express Servers as well as MongoDB. You witness the administration of GET route functionality when seeing the display of items available for example as well as a POST route when adding items to carts. You also witnessed the authentication of users and a somewhat decent UI. In truth, I did not spend too much time on the UI so that I might have more time for the server side.

That being said, in the future, I hope to return to this project to address the UI in more detail. It is also my hope to be able to interact with a public api allowing for online payment as well as providing an administrative interface which will allow the organization to add detail on items kept in stock. On the user side, it is my intention to provide a menu, once logged in, so that the user may select a category from which to shop allowing a more refined search.

Addressing the requirements of introducing two new technologies, I tried to utilize Auth0 for authentication but it didn’t go to well. I also used RXJS as an extension’s library which utilizes observables to make it easier to compose asynchronous or callback-based code.

It is a work in progress and as my skill continues to develop so will this app.