Tech Stack

For the purpose of this app, we have chosen the MERN stack with the integration of the firebase that we thought would be the best one, enabling us to speed up the development process, Cut down on testing time and bugs. We Have had these basic requirements to keep in mind, while selecting the Tech Stack

The main Technology used here is React-JS for the front-end with a nodejs + google firebase backend. This decision was taken keeping in mind the hosting, Database management needs for the app, and a secure, easy and trusted Authentication protocols that have been in the trade for long and have been tested.

We will implement the Stripe/Razorpay functionality with either the cloud functions (nodeJS) or the Google Firebase protocols

The Database we are using is a NoSQL database that is provided by Firebase (CLOUD FIRESTORE). The reason for this decision is we need a fast, secure and easy navigating database, also the Realtime feature enhances the speed of the app and user experience.

For the purposes of the Tech Stack explanation,we have kept in mind 2 things about the MVP Requirements :

- The WHAT and,
- The HOW

User Registration / User Log-In:

- Requirement : As a new user to the site I should be able to register my self and as a returning user I should be able to login to my account

- Solution : For the Purposes of User Registration and use Log in, we have decided to use the Google Firebase Authentication Protocols coupled with our custom authentication. This will undergo in the following steps:
    1. User will only be prompted to signIn or Register with the App if he/she decides to actually interact with the app (Interaction : Ordering, CRUD Operations)
    2. In that case, The user has two options :
        - Hit the Button on the header component to directly sign In/ Register
        - Can also register via the User specific page
    3. After that, User can login via either the Email/Pass, Google/Facebook Authentication or via the phone number (OTP login)
    4. This will be achieved via the Firebase Authentication which will enable us to integrate all three with minimum development time in thus minimizing the Bugs.

User Profile:

- Requirement : As a user i should be able to add/update/delete my profile details (Address/communications)
- Solution : The user profile will lie on its specific page which will be coded in ReactJS with its own individual Component lying on the '/user' route. The user profile should have the following details on the page:
    1. The basic information about the user that is registering, including the name, address, phone numbers and default payment method
    2. Also it should have the orders component, which will contain the past orders/ orders in progress
- This All will be integrated into the App with Reactjs with using the Context API, in order to have easy access of the data throughout the App.

Payment Gateway:

- As this is an E-commerce app, a safe, secure and fast payment gateway system is almost and basic and true MVP requirement. To tackle the issue of making a secure payment system, we will use the Payment functionalities of Stripe.
- This will be done via the cloud functions and integration of the Firebase functions to process the customer payments

Product Searching:

- Requirement : As a user I want to find a particular products as per my requirement
- Solution : As any e-commerce App should have a search functionality, we have decided to tackle this with the " React-fuzzy-search" library, that enables users to search for the items, co-related to the search term.

User Details/Cart Functionality/Order Management:

1. Cart Functionality :
    - Cart Page
    - CRD operation on cart page
    - Add product to cart from various pages (search, category etc)
    - Update cart no in header
    - Add product to cart when user is not logged in

2. Order Management:
    - Order History Page
        - Previous order Details
            - Product basic details
            - Address
            - Order Delivery Status

- The cart Functionality and Order-Management, are two aspects of the app that adds to the user experience. We want these two aspects to be as accurate, fast and easy-to-use as possible.
- Hence, we will be using the React Context Api, for data management of the App, wherein we every component can have Insert and Update access to the Data, which negates the need for Prop Drilling and therefore reducing the number of bugs.