



MedicoCare

By Shubham Singh and Prajwal Machado

Overview

Built an Ecommerce application that can be used both by the user and a pharmacist who are willing to sell their products.

Goals

1. Build the application as a practice and understand the concepts of the react and redux in depth.
2. Build a backend with different schemas and also query the products and orders made by the users.

3. Implement a UI which is elegant and easier to understand.

Models

Schemas that are used for the backend and the property of each of the schemas

1. User Schema
Fields: name, email, password, image, address, mobileNo, type
2. Seller Schema (Shopkeeper Schema)
Fields: name, email, password, image, address, mobileNo, type, shopName
3. Medicine (Product schema)
shopkeeperId, product name, quantity, date, price, picture
4. Order Schema
buyerId, medicineId, buy, date, cart, quantity, deliveryPlace

Controllers

The functionality of each request based on each of the route parameters is segregated into User controllers and Seller controllers. The corresponding functionality is performed such as fetching the details from the database.

The controllers for the User routes are separated in the controller folder under User. Few of the controllers for user routes are as follows:

PostUserSignUp
ShowProfile
AddMedicineCart
MedicineDetails

The controllers for the Shopkeeper routes are separated in controller folder in Admin Folder. Some of the Shopkeeper or seller routes are as follows

shopkeeperLogin
detailOfMedicine
editMedicine

Packages Used

Express-fileupload

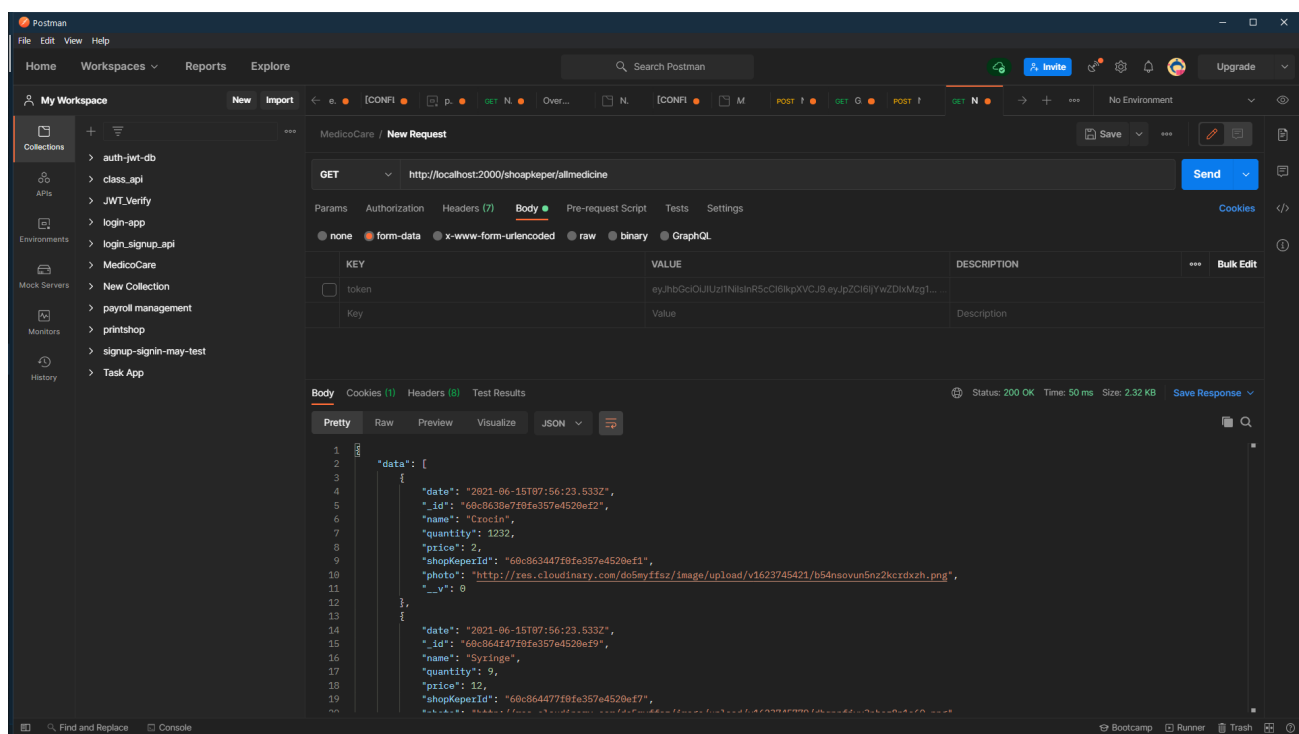
cloudinary

jsonwebtoken

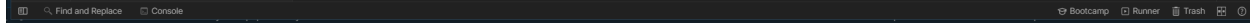
bcrypt

Screenshots

shopkeeper/allmedicine route while testing in postman



shopkeeper/signup route while testing in postman



The front-end part involves react-redux and for the component based library Material-UI. The actions performed based on a particular state change are captured using a global state manager redux. The api requests to the backend are also done through these actions and few of the internal state changes are also managed through action such as the loader state.

On Login the user can choose whether he is a user or a seller of the product. The UI has a forms for each of the specific inputs such as the user signup, seller signup, adding of the product, updating the product.

The user can also buy the product from the cart or directly buy it from the product description page. The cart items are retained for the specific user.

The user after making a purchase is able to download the generated bill for the ordered items.

react

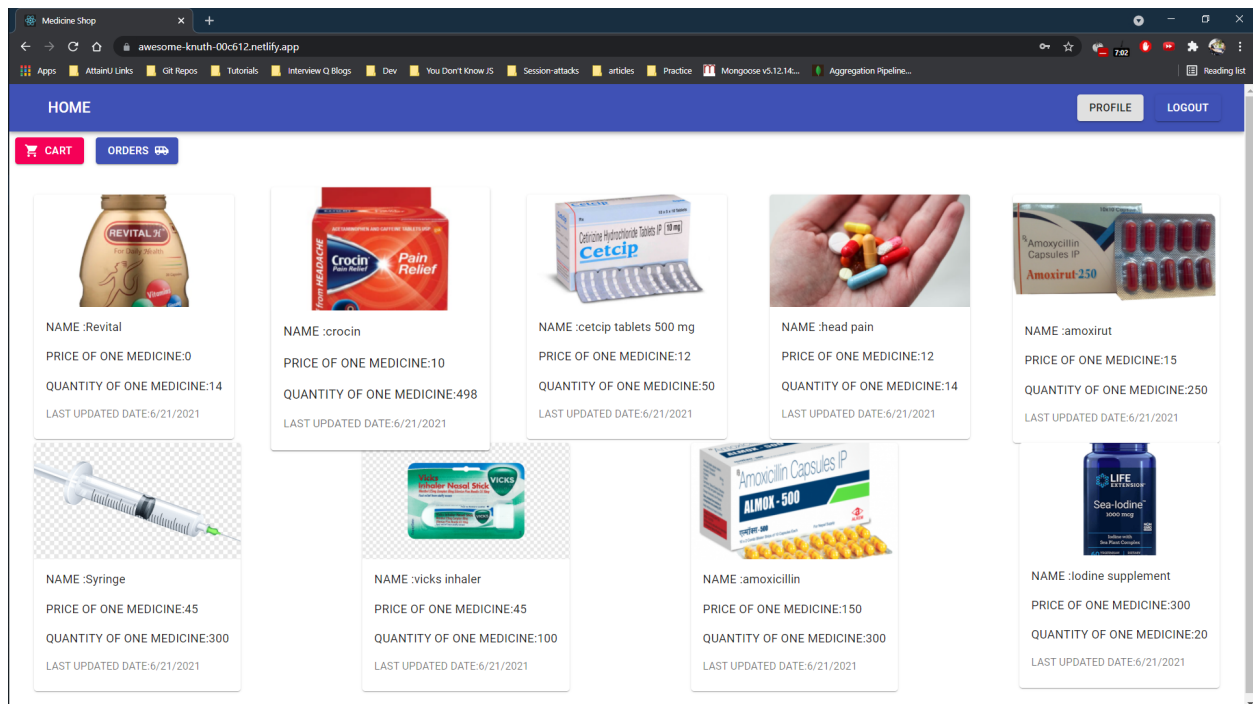
react-redux

react-to-pdf

react-router-dom

Screenshots

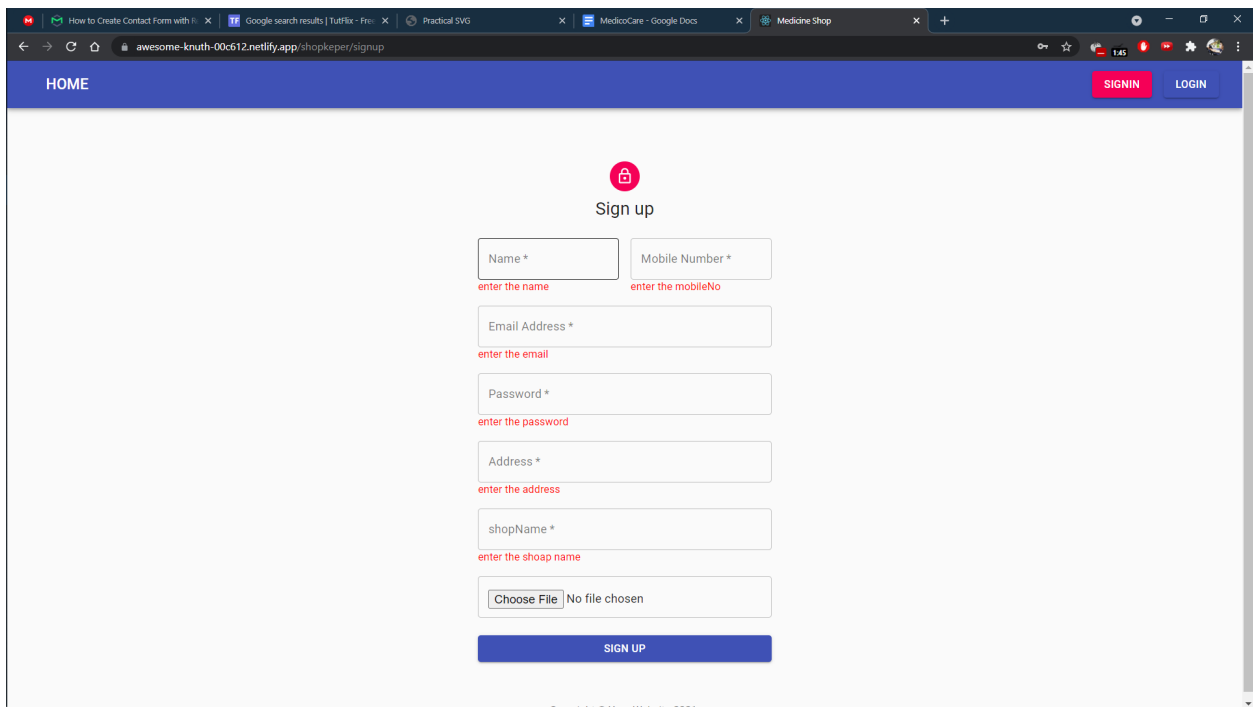
All products the user is able to buy.



Seller Pages

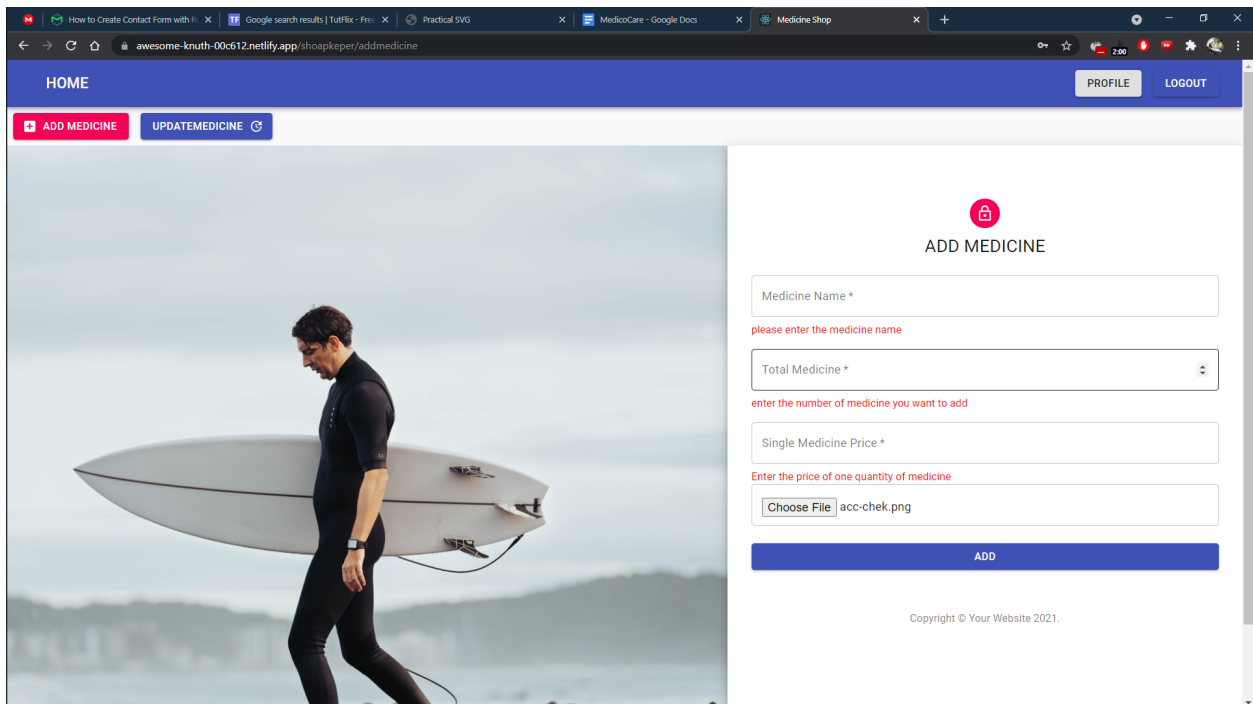
The seller who is willing to sign up to and sell his product on the website needs to sign up.

The screenshot is taken to show the validation error of that particular seller.



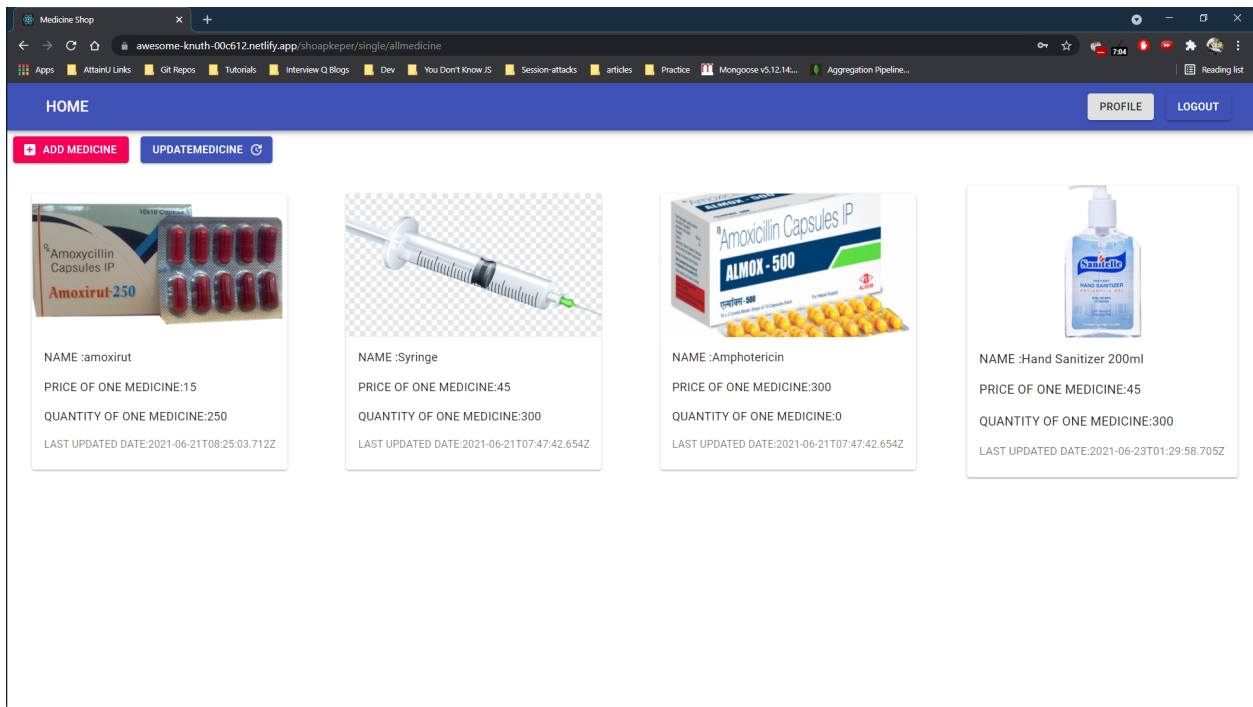
The screenshot shows a web browser window with the URL `awesome-knuth-00c612.netlify.app/shopkeeper/signup`. The page has a blue header with "HOME" on the left and "SIGNIN" and "LOGIN" buttons on the right. The main content area is titled "Sign up" with a red lock icon. Below the title are several input fields with red error messages: "Name *" (enter the name), "Mobile Number *" (enter the mobileNo), "Email Address *" (enter the email), "Password *" (enter the password), "Address *" (enter the address), "shopName *" (enter the shoap name), and a file upload section with "Choose File" and "No file chosen". A blue "SIGN UP" button is at the bottom. The footer says "Copyright © Your Website 2021".

The seller has who adds the particular product has to validate each of the fields and only then he is able to sign up into the application. The screenshot shows the same use case here.



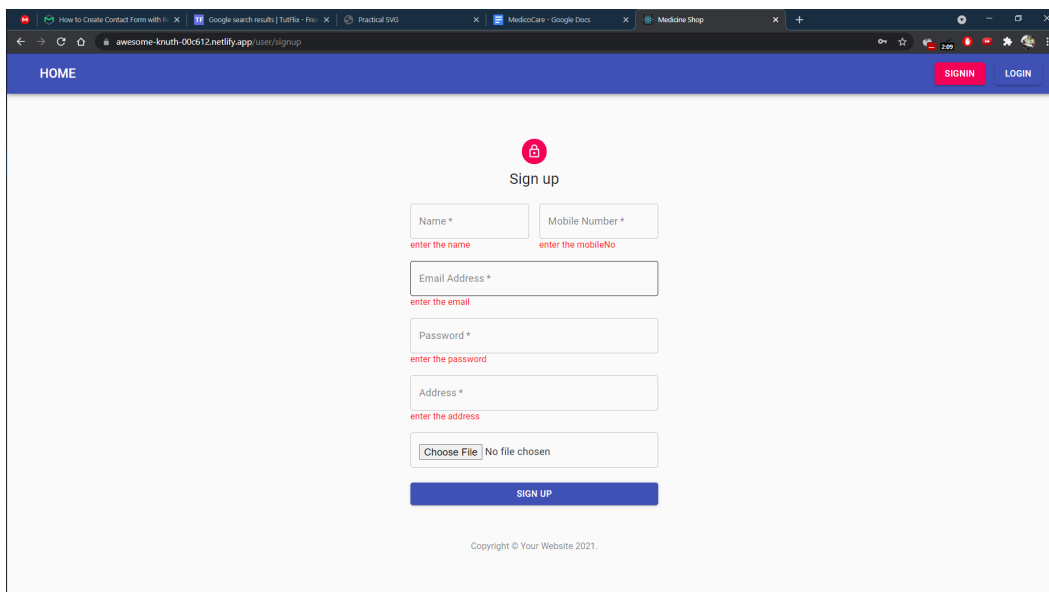
The screenshot shows a web browser window with the URL `awesome-knuth-00c612.netlify.app/shopkeeper/addmedicine`. The page has a blue header with "HOME" on the left and "PROFILE" and "LOGOUT" buttons on the right. Below the header are two buttons: "ADD MEDICINE" (red) and "UPDATERMEDIATE" (blue). The main content area is titled "ADD MEDICINE" with a red lock icon. Below the title are several input fields with red error messages: "Medicine Name *" (please enter the medicine name), "Total Medicine *" (enter the number of medicine you want to add), "Single Medicine Price *" (Enter the price of one quantity of medicine), and a file upload section with "Choose File" and "acc-chek.png". A blue "ADD" button is at the bottom. The footer says "Copyright © Your Website 2021".

All the products added by a particular seller are available in this route. He can add or update the products.

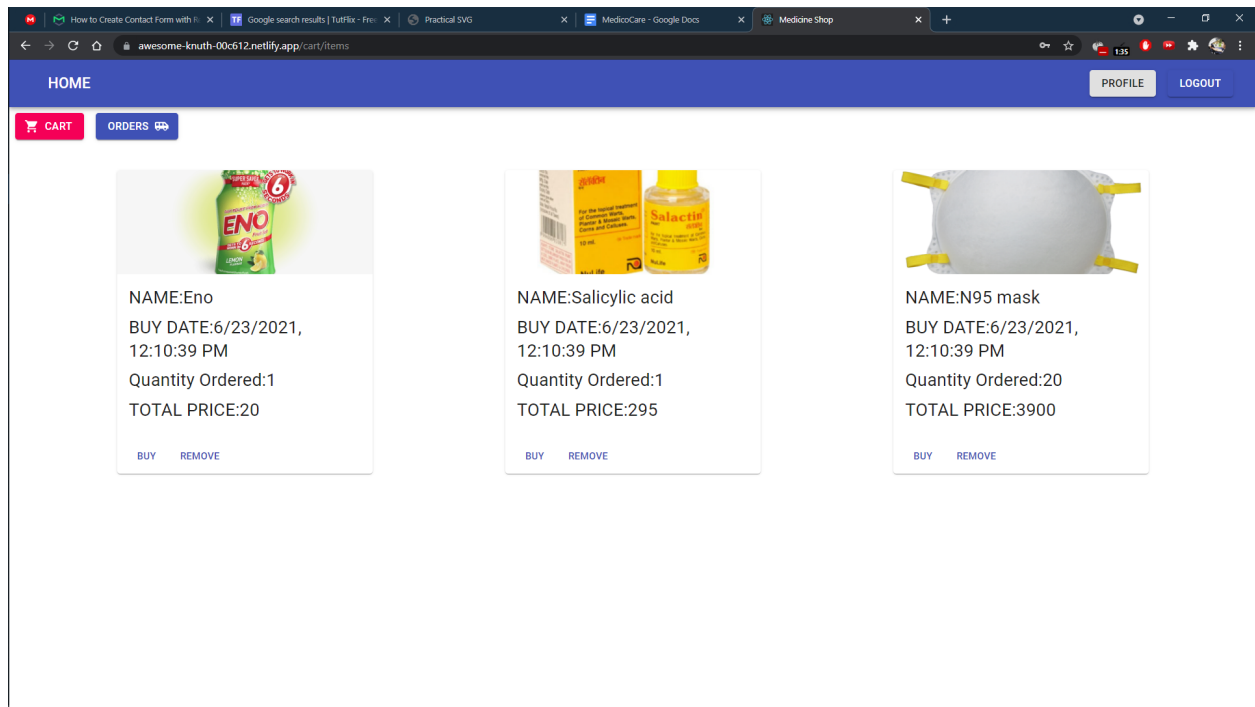


User Pages

User has to login and only then he will be able to perform buy functionality or adding into the cart. The route for the same is user/signup.



Cart functionality is available for a particular user, he can add it to his cart and then checkout. Screenshot here shows the cart of the particular user who is logged in.



The orders made by the particular buyer reflect in his order page for later use. The route for orders is `all/orders`

