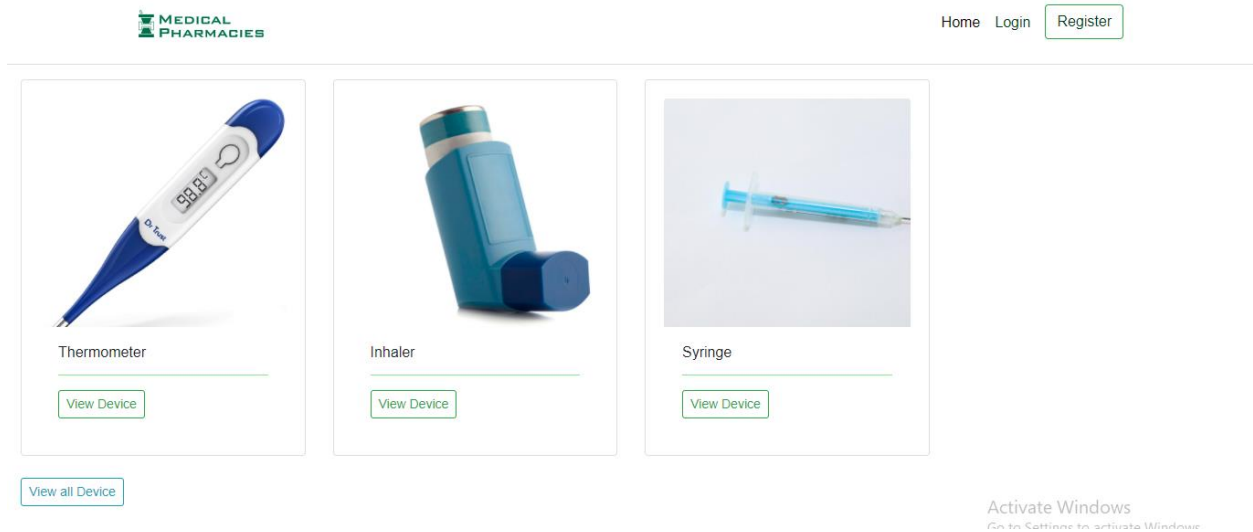


# Medical Device project

Using React, Node, Express, MongoDB and Typescript

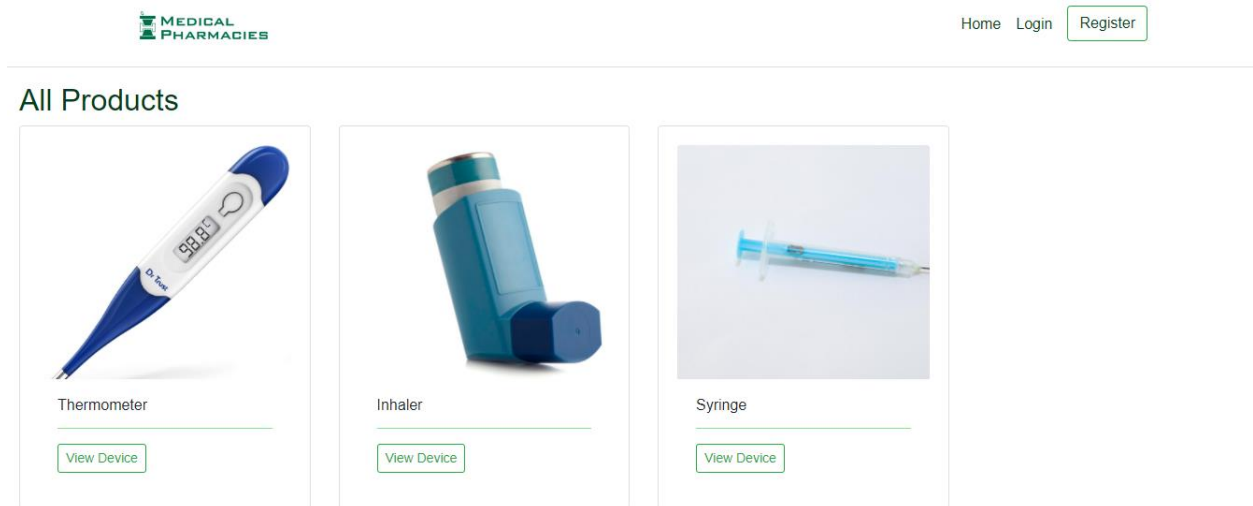
## Instruction

Home Page:



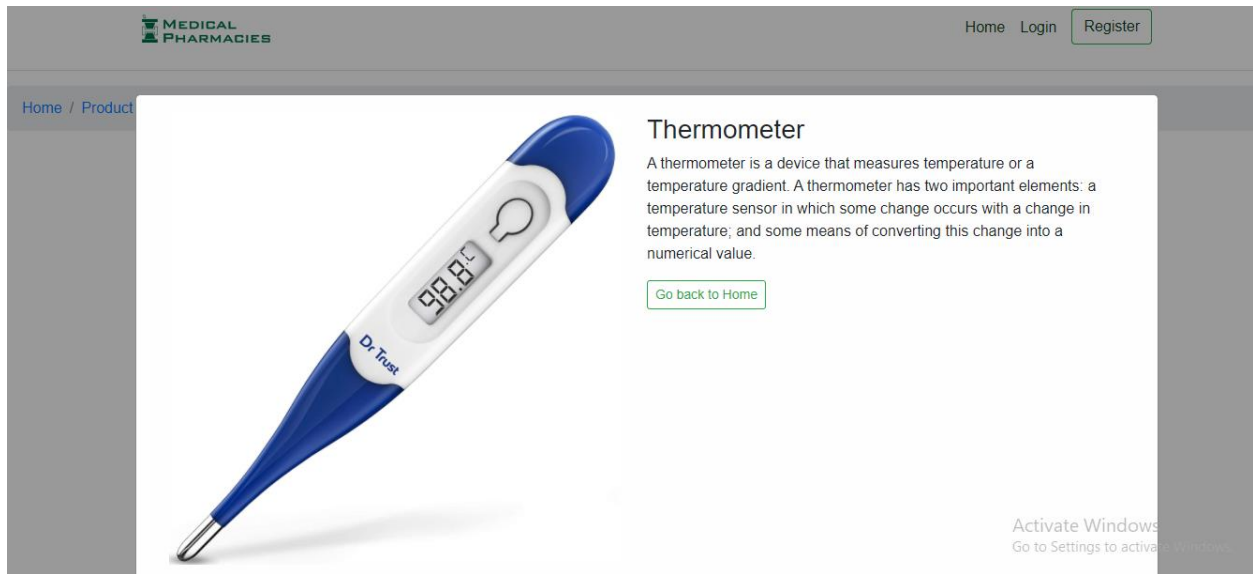
Click 'View all Device':

**URL: /modeltype**



Click 'View Device':

**URL:** /modeldata/{deviceId}



Login: URL: /login

A screenshot of a web application interface for a login page. The header is identical to the previous screenshot, with the 'MEDICAL PHARMACIES' logo and 'Home', 'Login', and 'Register' links. The main heading is 'Login Here'. Below this, there are two input fields: one for 'Email:' and one for 'Password:'. A 'Login' button is positioned below the password field. The page has a clean, minimalist design with a light gray background.

Role: Admin Email: <u>khan@gmail.com</u> Password: 123456
Role: User Email: <u>khanmashrur03@gmail.com</u> Password: 123456

Register as a user: URL: /register



[Home](#) [Login](#) [Register](#)

## Register Here

Name:

Email:

Password:

Create Account

User dashboard:

URL: /user/dashboard



[Home](#) [Dashboard](#) [Log Out](#)

## User Information


**Name:** Mashrur khan

**E-mail:** khanmashrur03@gmail.com

user

Admin dashboard:

URL: /admin/dashboard



Home Dashboard Log Out

User Links

Create Category

Create Device

User Information

**Name:** mashrur


**E-mail:** khan@gmail.com

admin

*Artists Window*

Click 'create category'

URL: /devicetype



Home Dashboard Log Out

Name

Create Category

Go to Dashboard

Click 'create category'

URL: /devicemodel



[Home](#) [Dashboard](#) [Log Out](#)

Photo:

No file chosen

Name:

Description:

Category:

----Select Category----

[Create Med Device](#)

[Go to Dashboard](#)

Activate Windows  
Go to Settings to activate Windows

I have used Nodejs, express JS and mongo DB for the backend of the project.

Why node JS?

- Node and react both are JavaScript language.
- Reactjs code directly in the Nodejs environment.
- Fast and scalable network applications.
- Lightweight and efficient.

Why mongo DB?

- free through the open source community edition
- Stores data in JSON format.

## File Structure

### Backend:

```
controllers/  
  categoryControllers.js  
  categoryControllers.ts  
  productController.js  
  productController.ts  
  userControllers.js  
  userControllers.ts  
middlewares/  
  admin.js  
  admin.ts  
  authorize.js  
  authorize.ts  
  error.js  
  error.ts  
models/  
  category.js  
  category.ts  
  product.js  
  product.ts  
  user.js  
  user.ts  
routers/  
  categoryRouter.js  
  categoryRouter.ts  
  productRouter.js  
  productRouter.ts  
  userRouter.js  
  userRouter.ts  
.env  
App.js  
App.ts  
Package-lock.json  
Package.json  
Server.js  
Server.ts  
Tconfig.json
```

## Frontend:

```
public/  
  favicon.ico  
  index.html  
  logo192.png  
  logo512.png  
  manifest.json  
  
src/  
  api/  
    apiAdmin.js  
    apiAdmin.ts  
    apiAuth.js  
    apiAuth.ts  
    apiProduct.js  
    apiProduct.ts  
  assets/  
    logo.svg  
  components/  
    admin/  
      AdminDashboard.js  
      CreateCategory.js  
      CreateProduct.js  
    home/  
      AllProduct.js  
      Card.js  
      Home.js  
      ProductDetails.js  
  protectRoutes/  
    AdminRoutes.js  
    PrivateRoute.js  
  user/  
    Dashboard.js  
    Login.js  
    Register.js  
  Layout.js  
  Main.js  
  Menu.js  
  utils/  
    auth.js  
    config.js  
    messages.js  
  App.js  
  index.js
```

react-app-env.d.ts

.env  
Package-lock.json  
Package.json  
Tconfig.json