Brains Kart Application (MERN)

Client-side Technologies

React + Redux + Bootstrap + Typescript

Server-Side Technologies

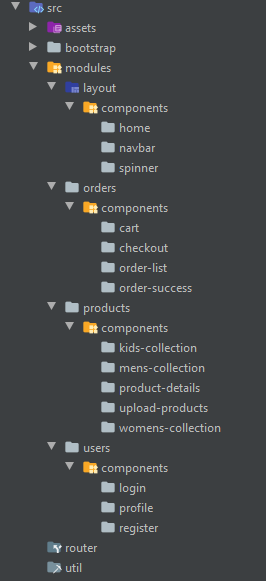
Node JS + Express JS + MongoDB (Typescript)

Folder Structure

Brains-kart-react

-> client

-> Client Code



-> server Code

Client Configuration

-> Create an React Application "client"

-> npx create-react-app client

Client Side Npm packages

|  |  |
| --- | --- |
| Modules | Details |
| Modules | axios  react-router-dom  redux  react-redux  redux-thunk  redux-logger  redux-devtools-extension  mdbootstrap |
| npm install axios react-router-dom redux react-redux redux-thunk redux-logger redux-devtools-extension mdbootstrap | |

React Modules

products , orders , users

-> each module have its own things

components

-> Users Module :

components : login , register , profile

-> Products module :

components : mens-wear , kids-wear , women-wear, upload

product-details

-> Orders module :

components : cart , checkout , order-list , order-success

Redux Setup

* User
  + user.actoinTypes
  + user.actions
  + user.reducer
* Order
  + order.actoinTypes
  + order.actions
  + order.reducer
* Product
  + product.actoinTypes
  + product.actions
  + product.reducer

Server Configuration (Express + MongoDB)

|  |  |
| --- | --- |
| Modules | Details |
| Modules | bcryptjs ,  cors ,  dotenv ,  express ,  express-validator ,  gravatar ,  jsonwebtoken,  mongoose,  stripe |
| npm install bcryptjs cors dotenv express express-validator gravatar jsonwebtoken mongoose stripe  npm install -D concurrently | |

REST API Configuration

Routers : userRouter , productRouter , paymentRouter , orderRouter

User Router

|  |  |
| --- | --- |
| 1. Register a User |  |
| URL | /users/register |
| Fields | Name , email , password |
| Method | POST |
| Access | PUBLIC |

|  |  |
| --- | --- |
| 2. Login a User |  |
| URL | /users/login |
| Fields | email , password |
| Method | POST |
| Access | PUBLIC |

|  |  |
| --- | --- |
| 3. Get User Info |  |
| URL | /users/ |
| Fields | No-fields |
| Method | GET |
| Access | PRIVATE |

|  |  |
| --- | --- |
| 4. Create / Update Address |  |
| URL | /users/address |
| Fields | flat , street , landmark , city , state , country , pin , mobile |
| Method | POST |
| Access | PRIVATE |

Product Router

|  |  |
| --- | --- |
| 1. Upload a Product |  |
| URL | /products/upload |
| Fields | name , brand , price , qty , image , category , description , usage |
| Method | POST |
| Access | PRIVATE |

|  |  |
| --- | --- |
| 2. Get Men’s Collection |  |
| URL | /products/men |
| Fields | No-fields |
| Method | GET |
| Access | PUBLIC |

|  |  |
| --- | --- |
| 3. Get Women’s Collection |  |
| URL | /products/women |
| Fields | No-fields |
| Method | GET |
| Access | PUBLIC |

|  |  |
| --- | --- |
| 4. Get Kid’s Collection |  |
| URL | /products/kids |
| Fields | No-fields |
| Method | GET |
| Access | PUBLIC |

|  |  |
| --- | --- |
| 5. Get a Product |  |
| URL | /product/:product\_id |
| Fields | No-fields |
| Method | GET |
| Access | PUBLIC |

Order Router

|  |  |
| --- | --- |
| 1. To Place an Order |  |
| URL | /orders/place |
| Fields | Items , tax , total |
| Method | POST |
| Access | PRIVATE |

|  |  |
| --- | --- |
| 2. To Get All Orders |  |
| URL | /orders |
| Fields | No-fields |
| Method | GET |
| Access | PRIVATE |

Payment Router

|  |  |
| --- | --- |
| 1. Accept Stripe Payments |  |
| URL | /payment/pay |
| Fields | Product , token |
| Method | POST |
| Access | PRIVATE |

Database Configuration (MongoDB)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Database Name | BRAINS-KART-05-RCJS-EVNG |  |  |
| Tables | Orders | Products | Users |
| Orders Table | Name , email , mobile , items [ name, brand, qty , price] , tax , total , created | | |
| Products Table | Name , brand , price , qty , image , category , description , usage , created | | |
| Users Table | name , email , password , avatar , isAdmin , created ,  Address : {flat , street , landmark , city , state , country , pin , mobile } | | |