FshionBot (Preloved Clothes)

Documentation

FYP

**INTRODUCTION:**

The website is coded using Vite React. To run:

* Frontend: npm run dev
* Backend:nodemon index.js

**FRONTEND:**

**REQUIRED TO DOWNLOAD:**

* **SOFTWARE**
  + node js from google
* **EXTENSIONS IN VS CODE:**
  + ES7 + react reduce
* **NODE PAKAGE MANAGER MODULES AND PAKAGES:**
  + npm install react-bootstrap (command prompt front-end)
  + npm install boorstrap (command prompt front-end)
  + npm install js-cookie (command prompt front-end)
  + npm add sass (command prompt front-end)
  + npm install axios (command prompt front-end)
  + npm install react-icons (command prompt front-end)
  + npm install react-stars (command prompt front-end)
  + npm instal aos (command prompt front-end)
  + npm install react-svg (command prompt front-end)
  + npm install react-modal (command promt front-end)
  + npm install sweetalert2 (command prompt front-end)
  + npm install mdb-react-ui-kit (command prompt front-end)
  + npm install react-jwt (command prompt front-end)
  + npm install firebase (command prompt front-end)
* **EXTERNAL RESOURCES USED:**
  + UnDraw website for illustrations
  + Coolors website for color theme selection.
  + Wepik for creative images

**FOLDERS:**

we mainly have 4 folders in the fronted:

* **Admin**: this folder has components and pages for Admin managing, admin can create, update and delete category. Routing has been performed on this folder
* **User**: This folder has components and pages that will be shown to the users that are Logged in. Routing has been performed on this folder.
* **Guest**: Initially, when the wesite begins it shows the guest UI which is entirely done in this folder, Routing is used and then components and pages are rendered. Context for Login and Signup has been used here along with Decoding json web token.
* **Common Features**: This folder contains components and pages that needs to be used in the entire website example footer etc. There is no Routing performed.

**BACKEND:**

**REQUIRED TO DOWNLOAD:**

**NODE PAKAGE MANAGER PAKAGES AND MODULES:**

* + npm install express (cmd, back-end)
  + npm install dotenv (command prompt backend)
  + npm install cors (command prompt backend)
  + npm install -g nodemon (command prompt backend)
  + npm install mongodb (command prompt backend)
  + npm install mongoose (command prompt backend)
  + npm install bcryptjs (command prompt backend)
  + npm install jsonwebtoken (command prompt backend)
  + npm install dialogflow (command prompt backend)

***MONGODB:***

Username: mahnooryameen22

Password: Z4TAjtgyySMw4aXr

**.ENV Variable:**

MONGO\_URL:mongodb+srv://mahnooryameen22:<password>@cluster0.b0e0eyz.mongodb.net/?retryWrites=true&w=majority

***FIREBASE:***

Firebase setup is placed in the Admin Foleder by the name FireConfig. It has been used for imgae URLs generation of Ad posting by users. Its Storage Bucket has been implemented.

***API ENDPOINTS:***

***AD:***

|  |  |  |
| --- | --- | --- |
| HTTP METHOD | ENDPOINT | Description |
| POST | /api/create-ad | For creating ad In databse, when user posts ad |
| GET | /api/get-ads-by-AdCategory | Getting ad by category |
| GET | /api/get-ads-by-id | Getting ad by id |
| GET | /api/get-all-ad | Getting all ads from the mongodb |
| PUT | /api/update-ad | Update ad |
| DELETE | /api/delete-ad | Delete ad |
| GET | /api/review-ad | For taking reviews on ads |

***CATEGORY:***

|  |  |  |
| --- | --- | --- |
| HTTP METHOD | ENDPOINT | Description |
| POST | /api/create-category | For creating category In databse only by admin |
| GET | /api/get-category-by-name | Getting category by name |
| GET | /api/get-category-by-id | Getting category by id |
| GET | /api/get-all-categories | Getting all category from the mongodb |
| PUT | /api/update- category | Update category by admin only |
| DELETE | /api/delete- category | Delete category by admin only |

***USER:***

|  |  |  |
| --- | --- | --- |
| HTTP METHOD | ENDPOINT | Description |
| POST | /api/login | For login |
| POST | /api/register | For creating user In databse |
| GET | /api/get-all-user | For getting all users from mongodb |
| GET | /api/getuserbyid | Getting user by id from mongodb |
| GET | /api/delete-user | Delete user by admin only |
| DELETE | /api/getuserbyemail | Get user by email |
| PUT | /api/update-user | Update user details by admin only |