**MERN Multi-vender E-commerce Website**

This project aims to assess the candidate's proficiency in building a front-end application, managing the state, handling complex UI components, and implementing basic e-commerce features. Develop a secure and efficient backend API for an E-commerce Product Catalog. The backend should support user authentication, product management, and shopping cart functionality.

The Requirements we use in frontend:-

1. Product Listing
2. Product Details
3. Shopping Cart
4. Checkout Process
5. State Management
6. Routing:

The Requirements we use in the backend:-

1. Authentication
2. Product Management
3. Shopping Cart
4. Database Integration
5. Middleware

Technology Is Used in this project:-

* FRONTEND:-
  1. HTML
  2. CSS, Tailwind CSS
  3. JAVASCRIPT
  4. React, Json
* BACKEND
  1. NODE JS
  2. YARN
  3. MONGO DB

=============================FRONTEND=============================

Now we start the project assignment.

-:The basics Steps

1) First open the terminal.

2) Then type in cmd npx create-react-app (Your Project name) /\*(Note:- Node shoulb be installed before.)

3)instal some library in vs code extention ES7 & ESLint

4) Go to the MERNTASK file.

5) Now install some libraries.

{

1) Axios (Used for get, post,delete, patch data calls). (npm install axios)

2) Material ui (For UI).(npm i @material-ui/core & @material-ui/icons)

3) Redux (state management tool) (npm redux react-redux redux-thunk (dispatch middleware)) & redux-devtools-Extensions

4) moment (date formatter library)

}

\* For Start this project simple run npm start

app.js header & front

app.css style for home page

================================BACKEND===========================

:- make a directory & open this (use mkdir F & cd F)

1) \*npm init -y (it create package.json file)

2) install library.

{

\*npm cache clean --force

\* npm i express mongoose cors

\* npm i nodemon

\*nodemon (for runserver)

}

3) create server.js file

4) install some more library

{

\*npm install bcrypt cookie-parser jsonwebtoken

1) bcrypt (for password encryption)

2) cokie parser, jsonwebtoken(for token generation)

3)multer (bor multi-part/form data parsing.) (\* npm i multer)

4) dotenv()

}

----------------------------------------------

create .env file

create controller folder

create models folder

create middlewares folder

create routers folder

HOW TO RUN THIS PROJECT LOCALLY

1. Download this source code from git hub.
2. Extract this zip file
3. Open your terminal or CMD
4. Go to the terminal and type CD socket & type npm i
5. Similarly go to the backend & type npm i
6. & go to the frontend then type npm I if it shows any error then you can use YARN
7. Now we connect the database so go to MONGO DB atlas & create a database & link it to the .env file in backend.

Also you fill all the require field (

PORT= "8000"

DB\_URL = " "

JWT\_SECRET\_KEY = " "

JWT\_EXPIRES = 7d

ACTIVATION\_SECRET = ' '

SMPT\_SERVICE = gmail

SMPT\_HOST = smtp.gmail.com

SMPT\_PORT = 465

SMPT\_PASSWORD = " "

SMPT\_MAIL = " "

STRIPE\_API\_KEY = ' '

STRIPE\_SECRET\_KEY = ' '

CLOUDINARY\_NAME = " "

CLOUDINARY\_API\_KEY = " "

CLOUDINARY\_API\_SECRET

1. ) Then go to server.js in frontend & type the link localhost://3000/api/v2
2. Also in app.js in backend use localhost://3000