**Instructions on how to deploy on IBM Cloud.**

1. Create an IBM Cloud account.
2. Choose IBM Cloud object storage
3. Install and configure your chosen e-commerce platform.
4. Add products and content, optimize performance, and test thoroughly.
5. Configure your domain and go live.
6. Maintain, update, and ensure compliance with regulations.
7. **Create .env File**

* duplicate .env.example in backend folder and rename it to .env

1. **Setup MongoDB**

* Local MongoDB
* Install it from [here](https://www.mongodb.com/try/download/community)
* In .env file update MONGODB\_URI=mongodb://localhost/auratraffic
* OR Atlas Cloud MongoDB
* Create database at [https://cloud.mongodb.com](https://cloud.mongodb.com/)
* In .env file update MONGODB\_URI=mongodb+srv://auratraffic:auratraffic@aura.komktxw.mongodb.net/auratraffic?retryWrites=true&w=majority&appName=AtlasApp

1. **Run Backend**

$ cd backend

$ npm install

$ npm start

1. **Run Frontend**

# open new terminal

$ cd frontend

$ npm install

$ npm start

1. **Seed Users and Products**

* Run this on browser: http://localhost:5000/api/seed
* It returns admin email and password and 6 sample products

1. **Admin Login**

* Run http://localhost:3000/signin
* Enter admin email and password and click signin

**Step By Step Process**

1. **Create React App**
2. **List Products**
   1. create products array
   2. add product images
   3. render products
   4. style products
3. **Add page routing**
   1. npm i react-router-dom
   2. create route for home screen
   3. create router for product screen
4. **Create Node.JS Server**
   1. run npm init in root folder
   2. Update package.json set type: module
   3. Add .js to imports
   4. npm install express
   5. create server.js
   6. add start command as node backend/server.js
   7. require express
   8. create route for / return backend is ready.
   9. move products.js from frontend to backend
   10. create route for /api/products
   11. return products
   12. run npm start
5. **Fetch Products From Backend**
   1. set proxy in package.json
   2. npm install axios
   3. use state hook
   4. use effect hook
   5. use reducer hook
6. **Manage State By Reducer Hook**
   1. define reducer
   2. update fetch data
   3. get state from usReducer
7. **Add bootstrap UI Framework**
   1. npm install react-bootstrap bootstrap
   2. update App.js
8. **Create Product and Rating Component**
   1. create Rating component
   2. Create Product component
   3. Use Rating component in Product component
9. **Create Product Details Screen**
   1. fetch product from backend
   2. create 3 columns for image, info and action
10. **Create Loading and Message Component**
    1. create loading component
    2. use spinner component
    3. craete message component
    4. create utils.js to define getError fuction
11. **Create React Context For Add Item To Cart**
    1. Create React Context
    2. define reducer
    3. create store provider
    4. implement add to cart button click handler
12. **Complete Add To Cart**
    1. check exist item in the cart
    2. check count in stock in backend
13. **Create Cart Screen**
    1. create 2 columns
    2. display items list
    3. create action column
14. **Complete Cart Screen**
    1. click handler for inc/dec item
    2. click handler for remove item
    3. click handler for checkout
15. **Create Signin Screen**
    1. create sign in form
    2. add email and password
    3. add signin button
16. **Connect To MongoDB Database**
    1. create atlas monogodb database
    2. install local mongodb database
    3. npm install mongoose
    4. connect to mongodb database
17. **Seed Sample Products**
    1. create Product model
    2. create seed route
    3. use route in server.js
    4. seed sample product
18. **Seed Sample Users**
    1. create user model
    2. seed sample users
19. **Create Signin Backend API**
    1. create signin api
    2. npm install jsonwebtoken
    3. define generateToken
20. **Complete Signin Screen**
    1. handle submit action
    2. save token in store and local storage
    3. show user name in header
21. **Create Shipping Screen**
    1. create form inputs
    2. handle save shipping address
    3. add checkout wizard bar
22. **Create Sign Up Screen**
    1. create input forms
    2. handle submit
    3. create backend api
23. **Implement Select Payment Method Screen**
    1. create input forms
    2. handle submit
24. **Create Place Order Screen**
    1. show cart items, payment and address
    2. calculate order summary
25. **Implement Place Order Action**
    1. handle place order action
    2. create order create api
26. **Create Order Screen**
    1. create backend api for order/:id
    2. fetch order api in frontend
    3. show order information in 2 cloumns
27. **Pay Order By PayPal**
    1. generate paypal client id
    2. create api to return client id
    3. install react-paypal-js
    4. use PayPalScriptProvider in index.js
    5. use usePayPalScriptReducer in Order Screen
    6. implement loadPaypalScript function
    7. render paypal button
    8. implement onApprove payment function
    9. create pay order api in backend
28. **Display Order History**
    1. create order screen
    2. create order history api
    3. use api in the frontend