All Empty

Creating package.json = npm init -y (all yes)

Done required changes in package.json

Video 2

Done More required changes in package.json

Install express = npm i express

Dot env = npm i dotenv

Morgan –api req konsa hit hua h = npm i morgan

Video 3

Create folder config = file db.js for database connection export the function

And use it in server.js

Video 4

Create folder routes,helpers,models,controllers, middlewares

Models/create file userModel.js = skeleton for user database

Now we will create routes

Routes/create file authRoute.js

AuthController/create file authController.js

We use MVC pattern file structure for our project

**Model-View-Controller(MVC) architecture for Node applications**

npm i bcrypt

helpers/create authHelper.js = password hashing

video 5

authController.js = registering user

video 6

npm i jsonwebtoken (see documentation if needed) for login route

video 7

create protected routes for user in middleware using user login token

in database = role 0 = normal user

role 1= admin

tested all the routes with middleware

video 8

working in react

cmd = npx create-react-app client

npm i react-router-dom

VS Code select all

  <li className="nav-item">

Select className do ctrl+dddddd……… for selecting all className variables at once

cmd= npm install react-icons

Video 9

SEO

React by default have no support for SEO it depends on 3rd party apps

Cmd = npm i react-helmet

Removed this by myself not necessary

Video 10 IMP

Pages/create folder Auth/ file Register.js

Have to avoid page refresh on submit =

*// avoid refresh of page on submit*

const handleSubmit = (e) => {

e.preventDefault();

console.log(name,email,password,phone,address);

}

npm i axios == handles http network request in frontend and backend

npm i react-hot-toast = notify icons

for frontend protection create .env file in root

In env file = REACT\_APP\_API = <http://localhost:8080>

Writing “REACT\_APP”\_youname is the format

npm i concurrently cors == to run two servers concurrently here client and server

Add cors in server js

cmd = PS D:\GitHub Desktop\Ecommerce-App> **npm run dev** == runs both together

working on register completed

client/src/pages/auth/create Login.js == for login form

video 12

Use **context api** for dynamic Navbar feature using global state = it is also use for large scale apps

Client/src/folder Context/ file auth js

Video 13

Private routes

Client/folder user/file Dashboard js

Client/src/components/ file Private js

Loading Spinner added

userLink location history = after login get back to user left page

video 14

done

video 15

dashboard for user and admin

**security**

if a normal user tries to access dashboard admin then it will direct the user again to the homepage

components/layout/AdminMenu js

video 16

for products we will create api in backend

**API designing is done here for products**

npm i slugify = for whitespace type of link eg user/**category-shirts**

These types of APIs are commonly referred to as \*\*RESTful APIs\*\* (Representational State Transfer APIs).

RESTful APIs adhere to the principles of REST, which is an architectural style for designing networked applications. RESTful APIs use standard HTTP methods (such as GET, POST, PUT, DELETE) to perform CRUD (Create, Read, Update, Delete) operations on resources.

RESTful APIs are widely used due to their simplicity, scalability, and flexibility, making them a popular choice for building web services and integrating different systems.