**Backend development**

**Backend folder :** create server.js

Run command >npm init

**{** npm init is a **command used to create a new Node.js project** and set up a file called **package.json**, which contains all the details about your project.

**📘 Full Form:**

**npm** → Node Package Manager  
**init** → Initialize

So, **npm init means “initialize a new Node project.” }**

Run command for install dependancies>npm i cors dotenv express jsonwebtoken mongoose multer nodemon razorpay stripe validator cloudinary bcrypt

| **Package** | **Purpose** |
| --- | --- |
| express | Create server & handle routes |
| mongoose | Connect MongoDB database |
| dotenv | Store secret keys in .env |
| cors | Allow frontend-backend communication |
| jsonwebtoken | Authentication (login tokens) |
| multer | Upload images/files |
| cloudinary | Store images in the cloud |
| bcrypt | Encrypt passwords |
| nodemon | Auto restart server on save |
| validator | Validate inputs (email, etc.) |
| razorpay | Indian payment gateway |
| stripe | Global payment gateway |

**CORS -It allows your frontend (React) and backend (Express) to communicate even if they’re on different ports.**

**Folder structure :**

**📁 1️⃣ config/**

**Purpose:  
To store configuration and setup files used throughout the project.**

**Includes:**

* **Database connection (db.js)**
* **Cloud services setup (like cloudinary.js)**
* **Environment setup using .env**

**Example:  
Connect MongoDB or setup Cloudinary image storage.**

**⚙️ 2️⃣ middleware/**

**Purpose:  
To store functions that run between the request and the response.  
They control access, handle errors, and manage file uploads.**

**Includes:**

* **authMiddleware.js → verifies JWT token for authentication**
* **errorMiddleware.js → handles app errors**
* **uploadMiddleware.js → handles image upload using multer**

**Example:  
Check if a user is logged in before accessing a protected route.**

**🧱 3️⃣ models/**

**Purpose:  
To define the structure of your database collections using Mongoose schemas.**

**Includes:**

* **userModel.js → defines fields like name, email, password**
* **productModel.js → defines product details (title, price, image, etc.)**
* **orderModel.js → defines order details (user, product, status, etc.)**

**Example:  
A “User” schema tells MongoDB how each user document should look.**

**🧠 4️⃣ controllers/**

**Purpose:  
To store the main logic or brain behind each API route.  
They handle what happens when an API is called.**

**Includes:**

* **userController.js → login, signup, profile logic**
* **productController.js → add, update, delete products**
* **orderController.js → create and track orders**

**Example:  
When you hit /api/users/login, the controller checks credentials and returns a token.**

**🛣️ 5️⃣ routes/**

**Purpose:  
To define all the available API endpoints (URLs) in your project.  
Each route connects to a specific controller function.**

**Includes:**

* **userRoutes.js**
* **productRoutes.js**
* **orderRoutes.js**

**{** "scripts": {

    "start": "node server.js", -------------------🡪//this command while deployment

    "server":"nodemon server.js"-------------🡪//for local server

}

**}**

**"type":"module",---------------🡪 for using import statement in the files**

**Purpose of .env File**

**🔒 1️⃣ To keep secrets safe**

**It stores sensitive data like:**

* **Database URLs**
* **API keys**
* **Cloud service credentials**
* **JWT secret keys**

**Server = Device + Program**

1. **Server as a device (hardware)**

* **A physical computer or machine that runs continuously to handle requests.**
* **Example: A computer in a data center hosting your website or backend.**

1. **Server as a program (software)**

* **A program running on that machine which manages incoming requests and sends responses.**

**Database -**Search mogodb atlas 🡪login--🡪new project-🡪create cluster(free version cluster 0)

***Do the database connection in config/mongodb.js file and import it in server .js***

**Create a one more file in config folder cloudnary.js for storing the images**

For that create account on cloudinary.com

Create cloudinary api key and add it to .env

Copy api secreate key and add it to .env

and establish a cloudinary connection for image storage

**🌩️ 1. What is Cloudinary?**

**Cloudinary** is an **online cloud storage service** specially made for **images and videos**.  
It allows developers to:

* Store files online (instead of saving them in your local computer).
* Get URLs for uploaded images.
* Automatically compress, resize, and optimize images.
* Deliver images fast to users worldwide.

👉 Think of it like **Google Drive**, but made for **web apps** and with extra features for developers.

**=🡺 In models folder create a schema for product and user details**

**--------------------------------------------------------------------------------------------**

**In the controller folder create a usercontroller file to add logic of user login and sign in and admin login**

**in router create the routes for this register ,login ,admin**

| **Part** | **Meaning** |
| --- | --- |
| express.Router() | Creates a separate route section for user APIs |
| UserRouter.post('/register', registerUser) | When frontend sends /register, call registerUser |
| import { loginUser... } | Import controller logic |
| export default UserRouter | Use these routes in main server |

**🔒 bcrypt**

* A library used to **securely hash passwords** before saving them in the database.
* It converts a plain password (like "Tushar123") into an unreadable string.
* Even if hackers access your database, they **can’t read real passwords**.
* Used in: **User registration & login authentication**

**✅ validator**

* A library used to **check and validate input data** like email, URL, etc.
* Helps ensure user inputs are **correct and safe** before storing or using them.
* Prevents invalid or harmful data from entering your system.

**JWT:**

**1️⃣ Purpose**

* **Authenticate users after login.**
* **Acts as a digital ID card to prove identity without sending password again.**
* **Makes backend requests secure and convenient.**

**==================================================================================*Checking the api respons :***we have vs code build support thunderapi to test it