Full-Stack Learning Path (Aapke "ASSEMBLE" Project ke liye)

Yeh guide aapko 0 se batayegi ki hamari website mein use hui har technology ke andar aapko kya-kya topics padhne hain.

Part 1: Backend (Website ka Engine ⚙️)

Backend woh hissa hai jo server par chalta hai. Iska kaam hai data ko manage karna, user ko verify karna, aur frontend ko data bhejna.

1. Node.js

Node.js ek environment hai, language nahi. Isne JavaScript ko browser se bahar nikal kar server par chalne ki power di.

- **Zaroori Topics:**
  - **npm (Node Package Manager):** Yeh Node.js ka "Play Store" hai.
    - npm init: Naya Node.js project shuru karne ke liye. Isse package.json file banti hai.
    - npm install <package-name>: Naye tools (packages) jaise Express, Mongoose install karne ke liye.
    - **package.json file kya hai?** Yeh project ki "kundli" hai. Ismein project ka naam, version, aur saare zaroori packages ki list hoti hai.
    - **node_modules folder kya hai?** Yeh woh "godown" hai jahan npm saare downloaded packages rakhta hai.
  - **Modules (CommonJS):**
    - require(): Kisi doosre file ya package ko apni file mein use karne ke liye. Jaise const express = require('express');.
    - module.exports: Apni file ke functions ya variables ko doosri files ke liye available karana. Jaise module.exports = router;.

2. Express.js

Yeh Node.js ka ek framework hai jo server banana aasan kar deta hai.

- **Zaroori Topics:**
  - **Server Banana:**
    - const app = express(): Express ki application banana.
    - app.listen(PORT, ...): Server ko ek port par start karna taaki woh requests sun sake.
  - **Routing:** Yeh API ke alag-alag address banata hai.
    - app.get('/', ...): Jab koi home URL par jaaye toh kya karna hai.
    - app.use('/api/users', userRoutes): Jab koi /api/users se shuru hone waale URL par jaaye, toh saari requests userRoutes.js file ko bhej do.

        - **express.Router():** Alag-alag routes ko ek file mein group karne ke liye (productRoutes.js dekho).
    - **Request & Response (req, res):** Har route mein do object hote hain.
        - req (Request): Ismein user ki taraf se aayi hui information hoti hai (jaise form ka data req.body mein).
        - res (Response): Isse hum user ko jawab bhejte hain. Jaise res.json(products) (data bhejna) ya res.status(404) (error bhejna).
    - **Middleware:** Yeh ek "security guard" ya "checker" ki tarah hai jo har request ke beech mein aakar kuch check karta hai.
        - app.use(cors()): Yeh check karta hai ki request sahi jagah se aa rahi hai ya nahi.
        - app.use(express.json()): Yeh aane waale JSON data ko JavaScript object mein badalta hai.
        - **Custom Middleware (authMiddleware.js):** Humne protect naam ka middleware banaya jo har request mein check karta hai ki user ne apna token (ID card) bheja hai ya nahi.

3. MongoDB & Mongoose

MongoDB hamara database hai (jahan saara data save hota hai) aur Mongoose us database se baat karne ka aasan tareeka hai.

- **Zaroori Topics:**
    - **Schema (productModel.js):** Yeh data ka blueprint (naksha) hai. Ismein hum batate hain ki har product mein ek name (jo String hoga), ek price (jo Number hoga), etc. hoga.
    - **Model (mongoose.model('Product', ...)):** Yeh us blueprint se bani asli cheez hai. Hum is Product model ka use karke database se saare products nikalte ya naye product daalte hain.
    - **Database Queries (Commands):**
        - Product.find({}): Saare products dhoond kar laao.
        - Product.findById(id): Ek specific ID waala product dhoond kar laao.
        - Product.create(...): Naya product banao.
        - Product.deleteMany(): Saare products delete kar do (humne seeder mein use kiya).

4. Security: bcrypt.js aur JWT

- **Zaroori Topics:**
    - **Password Hashing (bcrypt.js):**
        - bcrypt.hash(password, salt): Yeh function user ke password ko ek unreadable code mein badal deta hai. Hum database mein hamesha yeh hashed password hi save karte hain.
        - user.matchPassword(): Yeh function login ke time user ke daale hue password ko database ke hashed password se compare karta hai.
    - **Tokens (JWT):**

    - jwt.sign({ id }, secret_key): Jab user successfully login kar leta hai, toh hum is function se uske liye ek unique "Token" (digital ID card) banate hain.
    - jwt.verify(token, secret_key): Jab user koi protected page (jaise Profile) dekhna chahta hai, toh woh apna token bhejta hai. Hamara protect middleware is function se us token ko verify karta hai.

Part 2: Frontend (Website ka Showroom 🎨)

Frontend woh hai jo user ko browser mein dikhta hai.

1. React.js

Yeh UI banane ki library hai. Iski main power **Components** hain.

- **Zaroori Topics:**
    - **JSX:** JavaScript ke andar HTML jaisa code likhna.
    - **Components:** UI ke chote-chote, reusable tukde. Jaise hamari website mein Header, Footer, Product alag-alag components hain.
    - **Props:** Parent component se child component ko data bhejna. Jaise HomeScreen product data ko as a prop Product component ko bhejta hai.
    - **State (useState hook):** Component ki apni memory. Jaise HomeScreen mein products ek state hai. Jab setProducts(data) call hota hai, toh component ko pata chalta hai ki data aa gaya hai aur woh screen ko update kar deta hai.
    - **Effects (useEffect hook):** Jab component screen par dikhta hai, tab koi kaam karne ke liye. Humne iska use data fetch karne ke liye kiya hai. useEffect(() => { ... }, []) mein [] ka matlab hai ki "yeh kaam sirf ek baar karo, jab component pehli baar load ho".
    - **Conditional Rendering:** loading ? Loading... : <ProductGrid /> - Iska matlab hai, "agar loading true hai, toh 'Loading...' dikhao, nahi toh product grid dikhao".
    - **Mapping Lists (products.map(...)):** Ek array (jaise products ki list) se ek-ek item utha kar uske liye ek component (jaise <Product />) banana.

2. React Router DOM

Yeh library website mein alag-alag pages banati hai.

- **Zaroori Topics:**
    - **<BrowserRouter>, <Routes>, <Route>:** Yeh App.js mein use hote hain. Yeh batate hain ki kis URL par kaun sa component (screen) dikhana hai.
    - **<Link to="...">:** Yeh <a> tag ki jagah use hota hai. Iska fayda yeh hai ki page refresh nahi hota.
    - **useParams() hook:** URL se ID jaisi cheezein nikalne ke liye. Jaise ProductScreen mein /product/:id se id nikalne ke liye.
    - **useNavigate() hook:** User ko ek page se doosre page par bhejne ke liye. Jaise login hone ke baad home page par bhejne ke liye.

3. Axios

Yeh backend se data laane waala "delivery boy" hai.

- **Zaroori Topics:**
  - **API Call:** Backend se data maangne ke process ko API call kehte hain.
  - **axios.get('url'):** Backend se data laane ke liye.
  - **axios.post('url', data):** Backend ko naya data (jaise register form ka data) bhejne ke liye.
  - **async/await:** Yeh JavaScript ko batata hai ki "API call ke jawab ka wait karo, tab tak aage mat badho".

4. Redux Toolkit

Yeh poori website ka central "notice board" hai.

- **Zaroori Topics:**
  - **Store (store.js):** Yeh woh central notice board hai.
  - **Slice (cartSlice.js):** Notice board ka ek section, jaise "Cart ki Information".
  - **Reducers:** Woh functions jo batate hain ki notice board par information kaise update hogi (jaise addToCart reducer).
  - **useDispatch() hook:** Kisi component se notice board par nayi information likhne (action bhejne) ke liye.
  - **useSelector() hook:** Kisi component se notice board se information padhne ke liye.

5. Framer Motion

Yeh animation library hai.

- **Zaroori Topics:**
  - **motion.div:** Yeh ek normal div ki tarah hai, bas ismein animation ki power hoti hai.
  - **Animation Props:**
    - initial: Animation shuru hone se pehle kaisa dikhega.
    - animate: Animation poora hone par kaisa dikhega.
    - whileHover: Mouse le jaane par kaisa dikhega.
    - variants: Alag-alag animation states (jaise hidden, visible) ko define karne ke liye.