Web Development Notes-

1. Vite- Vite (French word for "quick", pronounced /vit/, like "veet") is a build tool that aims to provide a faster and leaner development experience for modern web projects. Vite. js has a significantly faster development server, better hot-reloading experience, and built-in support for modern JavaScript features like ES modules and TypeScript, resulting in a more efficient and enjoyable React development process.

Command-npm create vite@latest <folder name>

1. Tailwind CSS- A utility-first CSS framework for rapidly building custom designs ,steps to install in website.
2. Use tailwind intelli-sense extension.
3. React icons for putting icons in frontend.
4. Create separate frontend and backend folders in the project.
5. For header, use header component for SEO.
6. For backend folder,create package.json in the project directory for deployment using “npm init -y”.
7. Node.js-Node.js is a cross-platform, open-source JavaScript runtime environment that can run on Windows, Linux, Unix, macOS, and more. Node.js runs on the V8 JavaScript engine, and executes JavaScript code outside a web browser.
8. Express-Express.js is a fast, flexible and minimalist web framework for Node.js. It’s effectively a tool that simplifies building web applications and APIs using JavaScript on the server side.
9. Use nodemon to run server-side code effectively rather than node <filename>.

"scripts": {

"dev":"nodemon api/index.js",

"start":"node api/index.js"

},

This can be added in package.json of project so that we only do “npm run dev”.

1. For git to move to project folder when adding backend, use “mv .git ../” and then move the gitignore file from frontend folder to project directory.
2. For passwords or API key/Mongo url use dotenv package.

import dotenv from 'dotenv';

dotenv.config();

Create a .env file in project directory.

1. For api routes create a folder called api in backend folder and then put the routes using express.Router();
2. Create separate contollers for each api route rather than writing logic inside app.get().
3. In the index.js,put ‘app.use(express.json()) to allow json form data to perform POST operation while testing api route.
4. For database CRUD operations, always give try catch block.

Eg:-const newUser=new User({username,email,password:hashedPassword});

try {

await newUser.save();

res.status(201).json("User added successfully");

} catch (error) {

res.status(404).json(error);

}

1. To prevent refresh of form on submit,in handleSubmit() function put e.preventDefault().
2. JWT is used for authentication in sign-in purposes by creating a cookie token with unique id and a SECRET key.
3. Destructuring is done in the following way:-

const {password:pass,...rest}=validUser.\_doc;

1. Redux is a state management library that allows you to manage the state of your JavaScript applications more efficiently and predictably. Let's say you're building an e-commerce site. You may need to keep track of the items in a user's cart, their payment information, and their shipping details. Instead of passing this information from component to component using props, Redux allows you to store them in one central location where they can be easily accessed and updated. This makes it easier to manage complex states and keep your application organized.
2. Use redux-persist package to persist with the state value on refreshing also.
3. Use the useSelector() hook to use redux global state variables.
4. useNavigate() is a hook while Navigate is a React component.
5. useEffect() is a React Hook that lets you synchronize a component with an external system.
6. Use cookie-parser package to parse the cookie token.
7. useParams() hook is used to get the parameters from an endpoint url.

Eg:- <http://localhost:5173/update-listing/6688e7ad4598cdb072a5f139>

To get the listing-id, we can use params.listingId.

1. For getting react functional component add ES7+ React extension.
2. For console log in VS Code use Console Ninja.
3. Prettier extension for easy code formatting.
4. Tailwind CSS Intellisense for css.

Git steps: -

1. Git init
2. Git status
3. Git add .
4. Git commit -m “message”
5. Git remote add origin “github\_repo\_url”
6. Git branch -M main
7. Git push -u origin main