Full Stack Questions

Q1.What is the Difference between JWT authentication and session based authentication?

* JWT->Data is Stored in token itself on client-side.
* Sessions->server stores user data and cookie holds session Id.

Q2. Differnce Between Require and import?

* Require->Used in Common JS ,synchronous and also used in older Node.js syntax.
* Import-> Used in ES6 modules,asynchronous and it used with modern JS Syntax.

Q3. What are react hooks?

Ans3-> React hooks are functions to use React features in functional components.

* Coomon Hooks are : useState,useEffect,useContext,useReducer,useRef

Q4.What is Context Api and when to use it?

Ans4->Context Api allows Sharing data across components without passing props at every level. EG.User Authentication,theme-toogling used when many components need the same data.

Q5.Differnce between Props and State?

Ans 5-> Props->Read-only and are passed from parent to child.

State-> Local to the component and are mutable via useState or setState.

Q6. Differnce Between controlled and uncontrolled components?

Ans6-> .Controlled-> React controls the input value which we can update via onChange. It is used when we need real time validation or dynamic UI.

.Uncotrolled->The DOM controls the input value we can read via ref when needed. It is used for simple forms or quick inputs.

Q7->What is Middleware in Node+Express? Explain different types of middleware and when would you use each?

Ans7->Middleware is a function that sits between request and response.It can modify the request,respond or , end the request.

Types-:

1.Application level ->>runs for all routes.

2.Router-level ->> runs for specific routes Egample (Admin,Auth).

3.Error Handling->>handles error globally (req,res,err,next).

4.Built-in ->>powered by express (express.json()).

5.Thired party from npm egample(CORS).

Q8->In Node.js what is the difference between synchronous and asynchronous code execution? How does the event loop handle async tasks?

Ans8-> .Synchronous-> code runs line by line. And one task must finish before the next start. Example {fs.readfilesync()}->blocks everything until the code is read.

.Asynchronous->Node.js starts a task and when it finishes, a callback promise handles the result meanwhile other code keeps running does not blocks the code.

.Event loop ->It is like a traffic manager IT firsts listens for completed async tasks. And when it’s done it puts the callback promise into the queue to be executed.

This makes Node.js Non-blocking and handle thousands of request efficiently.

Q9->In Express.js why would you need a error handling middleware ?

Ans9-> It is a special type of middleware that catches errors from routes or other middlewares and sends a proper response instead of crashing the server.Designes with 4 Parameters(req,res,err,next).

Q10->Difference between Params,query and body in a request?

Ans10 .Req.Params ->URL path variables. And it is used when the value is part of the route itself. Example /users/:id ->req.params.id

.Req.Query ->URL Query Strings.It is used for optional filters or options.Egample/users?role=admission&sort=asc ->req.query.role,req.query.sort.

.Req.Body ->Data is sent in POST/PUT requests.It is used when client sends data in request body usually it is of JSON format.

Q11.What is the difference between app.use() and app.get()?

Ans11-> app.use()->>Defines middleware.It runs for all HTTP methods on the given path.Typically used for things like body parsing,authentication,logging or serving static files.

app.get()->>handles get requests to a specific route and is used to send data or a page when the client requests it.

Q12.In MongoDB what are the indexes and why do we use them .Explain one drawback of using too many indexes?

Ans12->Indexes are special data structure that make searching better, similar to an index in a book Even Collection has default \_id as in index. B ut we can also create indexes on the other fields to optimize queries

The downside is that indexes use extra memory and slows down insert and updates operations because the index must also need to be maintained.

Q13How Would you Protect your Api Routes in Express using Jwt Authentication and middleware?

Ans 13- > In Express I would protect routes using a jwt middleware .The middleware checks ho the authorization headers for a token, verifies it using jsonWeb token and if its valid then it attaches the userInfo to the request and allows it to continue. If it’s invalid or missing then it returns error 401 or unauthorized This Ensures only logged in users with a valid token can access protected API’S.

Q14how does a MVC architecture work in a web application and why is it useful?

Ans14->MVC(Model View Controller) is a way to organize web app into three parts. The model is about the data and business logic, bacially how information is stored and manged in the database.  
 .The View is what the user sees the interface.

.The controller is the middle It takes user requests talks to the model and often returns the right view or response .

.The benefit is that Each part has it’s own responsibilities so the code is cleaner, easier to maintain. Example In a mern app MongoDB with a mongoose is the model and React is the view and Express routes act as a controller.

Q15.What is the difference between an access token and a refresh token in authentication and why do we need both?

Ans15->.Access Token 🡪 is short lived(like 15 minutes) and is sent with every api request in the header to validate the new user and their roles.

.Refresh Token 🡪 is long lived(could be days or week) and is stored securely usually in http only cookie.

When the access token expires , the frontend sends a refresh token to get a new access token without asking the user to log in again.

Q16.What is REST API in Brief?

Ans 16🡪 A REST(Representational State transfer )API is away for the frontend and backend to talk to each other over HTTP. The Frontend (like react app sends request and the backend responds with data, usually in JSON format.  
 .It mainly uses HTTP methods 🡪

\*GET ->used to fetch data

\*Post -> used to create new Data.

\*PUT ->update existing data completely.

\*Patch ->update part of existing data.

\*Delete ->Remove data.

REST API’S also use status codes to indicate results.

.200->ok .201->created .400->bad request .401->unauthorized ,.403->forbidden ,404->Not Found ,500->internal server Error  
Testing Tools like Postman and thunderclient help developers test these api’s.

Q17. Difference between Server-Side-rendering(SSR) and Client-Side-rendering(CSR) in web applications and when would you use one over the other?

Ans 17 🡪 In CSR the server sends the HTML and JS file to the browser.The Browser then executes the JS builds the DOM,and renders the page.

\*Example: React by default uses CSR which gives the smoother app experience but has slower initial load and weaker SEO since the HTML is mostly empty first.

SSR 🡪the server generates the full html page and sends it to the client. The browser can display content immediately.

\*EXample:Next.js supports SSR SSR increases SEO and first-page load speedly but increases server load.

When to use-> .CSR for dynamic apps where SEO isn’t crucial(like dashboard,SAAS tools)

.SSR when SEO and fast first loads are important(blogs,e-commerce)

Q18why JWT Authentication ?

Ans 18 🡪 JWT tokens are stateless,secure and easy to use across frontend and backend. They help protect routes and manage user sessions.

Q19Difference Between Local Storage and Session Storage and cookies?

Ans 19🡪.Local Storage ->stores data permanently in browser.

.Session Storage🡪stores data only for a tab session.

.Cookies are small pieces of data stored and sent with each HTTP Request, usually for authentication.

Q20. Difference Between Flexbox VS Grid?

Ans 20🡪.Flexbox ->it is one dimensional useful for aligning items in a row or column

.Grid is two dimensional it Is used when we need to arrange elements in both rows and columns.

Q21 Difference Between HTML vs HTML5?

Ans 21🡪 HTML is the older version used for structuring content.

HTML5 --> introduced modern features like semantic tags video/audio support local and session storage.

Q22-> What is CORS and How to solve it?

Ans 22🡪 Cross -origin-Resource-Sharing issue occurs when frontend and backend both runs on different domains.solved by enabling CORS middleware in express backend.

Q23. What is reconciliation?

Ans23🡪 Process by which react updates the Dom Efficiently by comparing the Virtual Dom with previous Dom.

Q24. What is Client side routing?

Ans 24🡪Instead of fetching a new page from server ,React Router Swaps components for different paths ->SPA(SinglePage Application).(React just changes the components on the Screen using JS instead of loading HTML from server) so the page doesnot realod only the updated parts reload.

Q25. What is code splitting in React?

Ans25 🡪 Code Splitting is breaking apps into smaller bundles.for faster initial loads Achieved with React.lazy & suspense.

Q26. What is Lazy Loading?

Ans26🡪Components get Loaded Only when needed(in E-learn Lecture pages loaded only when users open a lecture).

Q27. How did you handle API Calls in frontend?

Ans27🡪used fetch/axios inside useEffect .Data is stored in context or components.

Q28. How do you handle Authentication on frontend?

Ans28🡪When user logs in ,I store the JWT token in local storage.Then on every API call ,I attach it in headers for backend validation.

Q29. Difference Between Pages and components?

Ans29🡪Pages --> full screens mapped to routes(Home,courses,dashboard).

Components -->Smaller UI building blocks resused across pages(Header,footer).

Q30.Differnce Between Authentication and Authorization?

Ans30🡪 Authentication --> is the process of verifying the identity of a user ,or system to ensure they are who they claim to be. It involves credentials like usernames,passwords,OTP’s and biometrics. IT Helps Prevents unauthorized access.

\*Authorization --> is the process of granting access rights to a unauthenticated user or system .It defines what resouces a user can access and what actions they are allowed to perfrom.

Q31. What is Express js ?

Ans31🡪Express.js is a lightweight Node.js framework that helps build API’s and backend services easily with routing,middleware and HTTP request handling.

Q32.Differnence Between REST API and GRAPHQL?

Ans 32🡪REST->multiple endpoints,fixed structure.

GraphQL 🡪Single endpoint,client defines the shape of the data.

Q33.How do you test React components?

Ans33🡪Using JEST and sometimes using REACT testing Library (enzyme).

Q34.What are Error boundaries in React?

Ans34 🡪 Error Boundaries are components that catch JS erorrs and display fallback UI.

Q35.What are portals in React?

Ans35🡪Think of portals as a secret door –they let you render a component outside it’s normal parent DOM but still keeps the react connection.

Q36.What are react fragments?

Ans36 🡪<> </> used to group children without extra DOM nodes.

Q37. What is Hydration?

Ans37🡪Process of attaching React events to server rendered HTML.

Q38What are Higher Order Components(HOC)?

Ans38 🡪Function that take a component and return a new component with added props.

Q39 How does React Router works?

Ans39 🡪Performs Client side Routing using <Browser Router>,<Route>,<Link>.

Q40How do you Optimize React performance?

Ans40 🡪use React.memo, lazy loading, code-splitting and avoiding unnecessary re-renders.

Q41.what is diffing?

Ans41🡪It how React figures out what changed in the UI.

Q42 What is Redux ?

Ans42 🡪Redux is like a central storage box for your app’s data. Instead of passing props everywhere all components can get or update data from this one box.

Q43Difference Between Context API and Redux?

Ans43🡪Context API -->simple way to share data between components. Best for small apps.

Redux -->more powerful, manages big complex state with clear control(actions,reducers,store).

Q44.what is Virtual Dom ?

Ans44🡪It is a lightweight copy of real Dom used for efficient updates.

Q45. Explain React Lifecycle methods?

Ans45🡪Mounting(Component DIDMount),updating (Component DIDUPDATE),unmounting(ComponentWILLUNMOUNT).

Q46.Difference between useMemo and useCallback?

Ans46🡪 useMemo -->memoizes values.

useCallback -->memoizes functions.

Q47.What is useContext?

Ans47 🡪used for global state management without prop drilling.

Q48.Differnce between functional and class components?

Ans48🡪Functional-->just a function that returns JSX. Use hooks for state.

Class -->old-style class with this.state and lifecycle methods.

Q49.What is React and why did you choose it?

Ans49🡪React is a Javascript library for building user interfaces using reusable components. I choose It because it improves performance with the virtual DOM and makes UI development modular and maintainable.