

# INTERVIEW QUESTION-ANSWERS OF ADVANCE FULL STACK DEVELOPMENT

## 1. What is Full-Stack Development?

Full-stack development involves working on both the frontend (client-side) and backend (server-side) of an application. Advanced full-stack development also covers optimization, deployment, and scalability.

## 2. Explain the difference between Monolithic and Microservices architectures.

- **Monolithic Architecture:** A single unified codebase where all functionalities are tightly coupled.
  - Pros: Simpler to develop and deploy for small-scale apps.
  - Cons: Difficult to scale and maintain as the app grows.
- **Microservices Architecture:** An application is divided into smaller, independent services that communicate via APIs.
  - Pros: Scalability, easier debugging, and independent deployment.
  - Cons: Complexity in setup and communication overhead

## 3. How do you manage state in modern frontend applications?

State management can be done using:

- **Redux:** For predictable state updates in complex applications.
- **Context API:** Built-in React feature for managing state globally.
- **MobX:** For observable state management.
- **Hooks:** React's `useState` and `useReducer` for local and global state.

## 4. How do you secure a full-stack application?

- Use **HTTPS** to encrypt data in transit.
- Implement **JWT** or **OAuth** for secure authentication.
- Sanitize user inputs to prevent **SQL Injection** and **XSS** attacks.
- Apply **rate-limiting** to prevent brute force attacks.
- Use **Content Security Policy (CSP)** to prevent unauthorized resource loads.

## 5. Explain the MVC architecture.

- **Model:** Manages application data and business logic.
- **View:** Handles the UI and displays data to users.
- **Controller:** Interprets user input and updates the model or view.

## 6. What are the benefits of using TypeScript in full-stack development?

- **Type Safety:** Reduces runtime errors.
- **Better Tooling:** Enhanced IntelliSense and autocomplete in IDEs.
- **Scalability:** Easier to manage large codebases.
- **Compatibility:** Works seamlessly with JavaScript libraries.

## 7. What are WebSockets, and when should you use them?

WebSockets provide full-duplex communication over a single TCP connection. They are ideal for real-time applications like chat apps, stock price monitoring, or multiplayer games.

## 8. What is CI/CD, and how does it benefit full-stack development?

- **CI (Continuous Integration):** Automatically tests and integrates code changes.
- **CD (Continuous Deployment):** Deploys code to production environments automatically.
- **Benefits:**
  - Reduces time to deployment.
  - Ensures consistent and reliable builds.
  - Automates testing, reducing errors.

## 9. What are HTTP status codes, and what do common ones signify?

- **200:** Success.
- **301:** Resource permanently moved.
- **400:** Bad request.
- **401:** Unauthorized.
- **403:** Forbidden.
- **404:** Not found.
- **500:** Internal server error.

## 10. How do you implement authentication in a full-stack application?

- **Backend:** Use OAuth2, JWT, or session-based authentication.
- **Frontend:** Store tokens securely (e.g., HttpOnly cookies) and manage user sessions.
- **Example:**
  - Use **Passport.js** or **Auth0** for Node.js.
  - Implement login flows with APIs for session management.

