

- ◆ 1. Tell us about a project where you worked as a full stack developer. What challenges did you face and how did you solve them?

**Answer:**

*"I developed an e-commerce platform using the MERN stack. One major challenge was ensuring efficient cart handling for both guests and logged-in users. I solved it by implementing a Redis-based temporary cart for guests and syncing it with the MongoDB user cart after login."*

---

- ◆ 2. What is the difference between MongoDB's **populate()** and **\$lookup**? When do you use which?

**Answer:**

- **populate()** is a Mongoose method that helps populate referenced documents easily within the app.
  - **\$lookup** is an aggregation operator used for joining collections in MongoDB natively.  
Use **populate()** when using Mongoose and dealing with small datasets.  
Use **\$lookup** for complex joins or performance-optimized queries in aggregation pipelines.
- 

- ◆ 3. How do you secure your MERN application? Mention at least 3 practices.

**Answer:**

1. Use **JWT tokens** stored in **httpOnly cookies** to prevent XSS.
  2. Implement **rate limiting** and **helmet.js** for Express security.
  3. Validate inputs using tools like **express-validator** or custom validation to prevent injection attacks.
- 

- ◆ 4. How would you optimize performance in a React application?

**Answer:**

- Use **React.memo**, **useMemo**, and **useCallback** to prevent unnecessary re-renders.
- Lazy load components with **React.lazy** and **Suspense**.
- Paginate and debounce data fetching.
- Optimize large lists with **react-window** or **react-virtualized**.

- 
- ◆ 5. What happens when you send a request to a secured route without a valid JWT token?

Answer:

- The backend middleware detects the invalid or missing token and returns a **401 Unauthorized** error.
- The frontend should handle this by redirecting the user to the login page or showing a message.

- 
- ◆ 6. What is the difference between `useEffect(() => {}, [])` and `useEffect(() => {}, [value])`?

Answer:

- `useEffect(() => {}, [])` runs only once after the component mounts.
- `useEffect(() => {}, [value])` runs every time `value` changes. It helps in reacting to prop or state changes.

- 
- ◆ 7. What are some best practices for structuring a large Node.js project?

Answer:

- Use MVC or feature-based folder structure.
- Separate **routes**, **controllers**, **models**, **middlewares**, and **services**.
- Use **dotenv** for environment config.
- Implement **logging** and **error handling** properly.
- Use **async/await** with proper try-catch.

- 
- ◆ 8. How would you design a scalable system for real-time chat?

Answer:

- Use **Socket.IO** for real-time communication.
  - Store messages in **MongoDB** with indexing.
  - For scale: Use **Redis Pub/Sub** for multi-instance syncing.
  - Deploy with **Load Balancer** and use **horizontal scaling**.
- 

♦ 9. What are the differences between SQL and NoSQL? Why did you choose MongoDB for your project?

Answer:

- **SQL** is relational, structured schema, great for complex joins.
  - **NoSQL (MongoDB)** is document-based, schema-less, more flexible and scalable.  
I chose MongoDB because the project required fast iterations and flexibility in data structure (e.g., product variants, dynamic attributes).
- 

♦ 10. How do you handle file uploads in MERN stack?

Answer:

- Frontend: Use **FormData** and **axios** to send files.
  - Backend: Use **multer** middleware in Express to handle file uploads.
  - Store in local filesystem, cloud (like AWS S3), or database as needed.
- 

 **Bonus Tips for Viva:**

- Be honest if you don't know something — say *"I haven't worked with it yet, but I'm willing to learn."*
- Be ready to show **GitHub projects**, **code snippets**, or discuss **real-world experience**.
- Maintain **clarity in explanation**, not just buzzwords.