Top Backend Interview Questions and Answers

1. What is REST API?

REST (Representational State Transfer) is an architectural style for designing networked applications. REST APIs use HTTP methods like GET, POST, PUT, DELETE to perform CRUD operations. They are stateless and communicate over standard URLs and HTTP responses.

2. What is MVC architecture?

MVC stands for Model-View-Controller. In backend:

- Model handles data and database logic
- Controller handles the business logic
- Route connects incoming requests to the controller

This separation makes the code cleaner and more manageable.

3. What is middleware in Express.js?

Middleware functions in Express.js are functions that have access to the request, response, and next middleware. They can modify the request/response or end the request-response cycle. Common uses include logging, authentication, and error handling.

4. What is JWT?

JWT (JSON Web Token) is a compact and secure way to transmit information between parties as a JSON object. It is often used for authentication. It has three parts: Header, Payload, and Signature. Access tokens and refresh tokens are often implemented using JWTs.

5. Difference between Access Token and Refresh Token

Access Token: Short-lived token used to access protected routes.

Refresh Token: Long-lived token used to generate new access tokens without requiring the user to log in again.

Access tokens are usually sent in headers; refresh tokens are stored securely (e.g., in HTTP-only cookies).

6. What is CORS?

CORS (Cross-Origin Resource Sharing) is a browser security feature that restricts cross-origin HTTP requests. Servers can enable CORS for specific domains using headers. In Express.js, you can use the 'cors' package to configure CORS.

7. How do you handle errors in Express.js?

You can handle errors using custom middleware with four parameters: (err, req, res, next). You can create centralized error handling logic to catch exceptions and send meaningful error responses.

8. How do you secure a Node.js backend?

Some ways include:

- Use HTTPS
- Sanitize inputs
- Use Helmet for setting secure HTTP headers
- Use JWT for auth and bcrypt for password hashing
- Implement rate limiting and logging

9. What is the difference between SQL and NoSQL?

SQL databases (like MySQL) are relational and use structured schemas.

NoSQL databases (like MongoDB) are non-relational and store data as documents (e.g., JSON).

NoSQL is more flexible and scalable for unstructured or rapidly changing data.

10. What is the purpose of models, controllers, and routes?

- Models: Define the schema and interact with the database
- Controllers: Contain business logic and interact with models
- Routes: Define endpoints and map them to controller functions

Together, they follow the MVC structure and make code modular and scalable.