**Web Services (REST)**

1. **Q:** How would you version REST APIs, and what are the pros and cons of each approach?  
   **A:** Use URL paths (/v1/resource) or headers (Accept: application/vnd.example.v1+json). URL versioning is simpler but less clean, while header versioning is cleaner but harder to debug.
2. **Q:** What is the difference between PUT and PATCH?  
   **A:** PUT replaces the entire resource, while PATCH updates only specified fields.
3. **Q:** How would you implement rate limiting in a REST API?  
   **A:** Use middleware to track request counts and throttle excessive requests.
4. **Q:** What is the difference between 401 Unauthorized and 403 Forbidden?  
   **A:** 401 indicates missing or invalid authentication, while 403 indicates the user is authenticated but not authorized.
5. **Q:** How would you implement pagination in a REST API?  
   **A:** Use query parameters like page and limit and include metadata in the response.
6. **Q:** What is the difference between SOAP and REST?  
   **A:** SOAP is a protocol with strict standards, while REST is an architectural style using HTTP.
7. **Q:** How would you implement HATEOAS in a REST API?  
   **A:** Include hypermedia links in the response to guide clients through the API.
8. **Q:** What is the difference between JWT and OAuth?  
   **A:** JWT is a token format, while OAuth is an authorization framework.
9. **Q:** How would you implement caching in a REST API?  
   **A:** Use HTTP headers like Cache-Control and ETag to control caching behavior.
10. **Q:** What is the difference between 200 OK and 204 No Content?  
    **A:** 200 includes a response body, while 204 indicates success with no content.
11. **Q:** How would you implement authentication in a REST API?  
    **A:** Use JWT or OAuth tokens and validate them in middleware.
12. **Q:** What is the difference between POST and PUT?  
    **A:** POST creates a new resource, while PUT updates or replaces an existing resource.
13. **Q:** How would you implement error handling in a REST API?  
    **A:** Use consistent error responses with status codes and descriptive messages.
14. **Q:** What is the difference between REST and GraphQL?  
    **A:** REST uses fixed endpoints, while GraphQL allows clients to query for specific data.
15. **Q:** How would you implement versioning in a REST API?  
    **A:** Use URL paths (/v1/resource) or headers (Accept: application/vnd.example.v1+json).
16. **Q:** What is the difference between GET and HEAD?  
    **A:** GET retrieves the resource, while HEAD retrieves only the headers.
17. **Q:** How would you implement file uploads in a REST API?  
    **A:** Use multipart/form-data and handle the file in the backend.
18. **Q:** What is the difference between REST and gRPC?  
    **A:** REST uses HTTP/JSON, while gRPC uses HTTP/2 and Protocol Buffers.
19. **Q:** How would you implement rate limiting in a REST API?  
    **A:** Use middleware to track request counts and throttle excessive requests.
20. **Q:** What is the difference between REST and WebSockets?  
    **A:** REST is request-response, while WebSockets enable real-time, bidirectional communication.