Question Answering Bonus

1. What are the differences between Rest and gRPC ?

Key Differences Between REST and gRPC

Protocol: REST uses HTTP 1.1, a text-based protocol, while gRPC uses HTTP/2, which is binary and multiplexed.

Data Format: REST typically uses JSON, a text-based and human-readable format, whereas gRPC uses Protocol Buffers, a binary, serialized, and compressed format.

Communication Model: REST follows a request-response model using HTTP verbs and URLs, while gRPC uses a client-server model with function calls and supports unary as well as bi-directional streaming.

Browser Support: REST has universal browser support due to its use of HTTP 1.1 and JSON; gRPC has limited browser support and often requires proxy layers to work in browsers.

Performance: gRPC offers higher performance and lower latency because of its binary format and HTTP/2 multiplexing, while REST has slower transmission due to larger JSON payloads and HTTP 1.1 constraints.

Flexibility: REST is flexible in data formats and endpoint design; gRPC provides strongly typed interfaces with built-in code generation and validation.

Use Cases: REST is popular for public web APIs needing broad browser compatibility. gRPC is better suited for internal microservices, mobile apps, IoT, and scenarios requiring high performance and streaming.

Tooling and Complexity: REST benefits from mature tooling and simplicity. gRPC requires a more complex setup but offers excellent language support with automatic code generation.

Caching: REST supports caching through HTTP headers and separation of endpoints with URLs. gRPC generally sends POST requests which are not cacheable.

When to Use Each

- REST is ideal for public-facing APIs needing broad browser compatibility, flexible data interchange, and straightforward implementation.
- gRPC excels in internal systems, microservices communication, mobile apps, and IoT where performance, efficiency, and streaming are critical.

In essence, REST is widely supported and simpler for public APIs, while gRPC offers speed and stronger typing advantages for internal and performance-sensitive scenarios.