

## Q1. What is an API? Give an example, where an API is used in real life.

- An API (Application Programming Interface) is a set of rules and specifications that define how software components should interact with each other. APIs allow developers to build applications that can communicate with each other and share data.
- For example, the Google Maps API allows developers to embed maps in their own applications. The Twitter API allows developers to build applications that can post tweets and retrieve user information.
- Real-life example of API:
  - When you use a weather app on your phone, the app uses an API to retrieve weather data from a weather service.
  - When you book a flight online, the website uses an API to communicate with the airline's reservation system.
  - When you use a social media app, the app uses an API to post updates and retrieve your friends' updates.

## Q2. Give advantages and disadvantages of using API.

- Advantages of using APIs:
  - Reduced development time: APIs can save developers time by providing them with pre-built functionality that they can use in their own applications.
  - Improved scalability: APIs can help applications scale by allowing them to access data and resources from other servers.
  - Increased flexibility: APIs can make applications more flexible by allowing them to be easily integrated with other applications and services.
- Disadvantages of using APIs:
  - Increased complexity: APIs can add complexity to applications, especially if they are not well-designed.
  - Security risks: APIs can introduce security risks if they are not properly implemented.
  - Reliance on third-party services: APIs can make applications reliant on third-party services, which can lead to problems if the services are unavailable or change their APIs.

## Q3. What is a Web API? Differentiate between API and Web API.

- A Web API is an API that is exposed over the web. Web APIs are typically accessed using HTTP requests and responses.

- The main difference between an API and a Web API is that a Web API can be accessed by any application over the web, while an API can only be accessed by applications that are running on the same server or local network.

## Q4. Explain REST and SOAP Architecture. Mention shortcomings of SOAP.

- REST (Representational State Transfer) is a lightweight architectural style for designing web APIs. REST APIs are based on the following principles:
  - Resources: REST APIs expose resources, which are data objects that can be created, read, updated, and deleted.
  - Representations: REST APIs represent resources using standard data formats, such as XML and JSON.
  - HTTP methods: REST APIs use standard HTTP methods, such as GET, POST, PUT, and DELETE, to perform operations on resources.
- SOAP (Simple Object Access Protocol) is a heavyweight XML-based protocol for designing web APIs. SOAP APIs are typically more complex to develop and consume than REST APIs.
- Shortcomings of SOAP:
  - Complexity: SOAP APIs are typically more complex to develop and consume than REST APIs.
  - Verbosity: SOAP messages are typically more verbose than REST messages.
  - Performance: SOAP APIs can be slower than REST APIs.

## Q5. Differentiate between REST and SOAP.

Characteristic	REST	SOAP
Architectural style	Lightweight	Heavyweight
Data format	Standard data formats, such as XML and JSON	XML
HTTP methods	Standard HTTP methods, such as GET, POST, PUT, and DELETE	Custom XML messages
Complexity	Less complex	More complex
Verbosity	Less verbose	More verbose
Performance	Faster	Slower

In [ ]:

