**Refer FB + LinkedIn API's :: Try some of the API's in Postman**

**Which API referred ?**

**Facebook -** Graph API

**LinkedIn -** Rest API or OAuth-based API

**What did you understand from the API's**

The Facebook Graph API allows developers to interact with and retrieve data from the Facebook platform. It provides access to various types of data, such as user profiles, posts, photos, pages, groups, and more.

The LinkedIn API allows developers to access LinkedIn platform data, including user profiles, connections, posts, messages, and more. It enables integration with LinkedIn's professional networking features.

**What are the parameters + endpoints**

**Facebook API -** The API endpoints and parameters depend on the specific data you want to retrieve or interact with. For example, you might use endpoints like /me to retrieve the user's own profile, /user-id/posts to fetch a user's posts, or /page-id/feed to retrieve a page's feed.

**LinkedIn API -** Similar to Facebook, the API endpoints and parameters depend on the specific data you want to access. Examples include /v2/me to retrieve the authenticated user's profile, /v2/people/id to fetch a user's profile by ID, or /v2/shares to post content to LinkedIn.

**What are the business utilities of the API's**

**Facebook API -** The Facebook Graph API enables businesses to integrate Facebook features into their applications. This can be used for purposes such as:

* Authenticating users using their Facebook accounts.
* Displaying a user's Facebook feed within an app.
* Posting content to a user's Facebook timeline.
* Retrieving user data to personalize user experiences.
* Integrating social sharing and engagement features.

**LinkedIn API -** The LinkedIn API provides various opportunities for businesses:

* Integrating LinkedIn authentication for user login.
* Displaying LinkedIn profile information within a business application.
* Posting professional content to a user's LinkedIn feed.
* Extracting professional network insights for analytics.
* Facilitating communication between users through messages.

**DB Schema:**

A database schema is a logical blueprint or plan that defines the structure and organization of a database. It represents the way data is organized, stored, and related within the database. A schema provides a framework for the database, specifying tables, fields, data types, constraints, and relationships between different elements. It acts as a roadmap for how the data should be structured and accessed by applications or users.

**Endpoint :**

**Queue Endpoint:**

Queue endpoint refers to the specific address or location where messages are sent and received within a messaging infrastructure. It represents the destination where messages are stored temporarily until they are consumed by the intended recipient or consumer application.

**Topic Endpoint:**

Topic endpoint is a destination or address where messages are sent and received in a publish-subscribe messaging pattern. Topics are used in pub/sub messaging systems to allow multiple consumers to receive messages from a single producer or publisher.

**What is High Availability Architecture?**

High Availability Architecture is an approach to designing and implementing systems that are highly reliable and constantly available, minimizing downtime and ensuring uninterrupted service even in the event of hardware or software failures. The main goal of high availability architecture is to enhance system resilience and reduce the impact of outages on end-users and critical business processes.