# Project: Social Media Dashboard using Instagram API

*Source : Google and Chatgpt.AI*
*Location : Github*

## Goal:

Create a **Social Media Dashboard** to analyze user engagement on Instagram posts by fetching, storing, and visualizing data.

## Key Skills You'll Learn:

✅ API Authentication (OAuth2)
✅ Fetching Data from Instagram API
✅ Parsing JSON Data
✅ Storing Data in a Database (SQLite/PostgreSQL)
✅ Data Analysis & Visualization (Tableau/Power BI)

# Step 1: Set Up the Environment

**Tools Required:**

✔ Python (Install: `pip install requests pandas sqlalchemy`)
✔ Instagram API Developer Account (Social technologies | Meta for Developers)
✔ Database (SQLite/PostgreSQL)
✔ Tableau/Power BI for Visualization

## 1️⃣ Install Python

### What is Python?

Python is a programming language used for automation, data analysis, APIs, and more. You need Python installed to run this project.

**Verify installation** by opening **Command Prompt (Windows)** or **Terminal (Mac/Linux)** and running:

```
python --version
```

## 2️⃣ Install Required Python Libraries

Python needs additional libraries to fetch and process data. We will install:

- `requests` – To call APIs and get data from Instagram.
- `pandas` – To store and analyze data.
- `sqlalchemy` – To save data in a database.

### Install Libraries Using pip

1. Open **Command Prompt** (Windows) or **Terminal** (Mac/Linux).

   Run the following command:
   ```
   pip install requests pandas sqlalchemy
   ```

2. After installation, verify by running:

   ```
   python -c "import requests, pandas, sqlalchemy;
   print('Libraries installed successfully!
   ```

# 3️⃣Create an Instagram API Developer Account

## What is an API Developer Account?

Instagram restricts access to its data. To use its API, you need a **Facebook Developer Account** and an **Instagram App**.

## Step-by-Step Guide to Create an Instagram API Account

1. **Go to Facebook Developers** and **Log in** with your Facebook account.
2. Click on "**Create App**" (top right corner).
3. Select "**For Business**" and click **Continue**.
4. Enter **App Name** (e.g., "My Instagram Dashboard").
5. Under **App Settings**, select **Instagram Graph API**.
6. **Generate an Access Token** using OAuth (Follow the on-screen instructions).
7. **Copy your App ID, App Secret, and Access Token** – You will need them for API calls.

# 4️⃣ Set Up the Database (SQLite/PostgreSQL)

Instagram data needs to be stored for later analysis. We will use **SQLite**, a lightweight database.

## How to Install SQLite(Windows):

1. Download SQLite from the official site.
2. Extract the file and add it to the system **PATH**.

## How to Create a SQLite Database :

Open Python **Command Prompt** and run:(jupyter notebook)

```
from sqlalchemy import create_engine

engine = create_engine("sqlite:///instagram_data.db")

print("Database created successfully!")
```

This will create a file named **instagram_data.db** where your data will be stored.

---

# Step 2:Register App w/Instagram API

## 1️⃣ Go to Facebook for Developers

1. Open your browser and visit: Facebook for Developers
2. **Log in** using your Facebook account.
3. Click on **"Get Started"** if this is your first time.
4. 

## 2️⃣ Create a New App

Instagram's API is managed by Facebook, so you must create an app to access it.

**Steps to Create an App:**

1. Click on **"My Apps"** (top-right corner).
2. Click on **"Create App"**.
3. Select **"For Business"** and click **Continue**.
4. Enter:
    - **App Name:** (e.g., "My Instagram Dashboard")
    - **App Contact Email**
5. Click **Create App** and enter your Facebook password if asked.

## 3️⃣ Enable Instagram Graph API

1. In your newly created app, go to **"Add a Product"**.
2. Find **Instagram Graph API** and click **Set Up**.
3. Under **App Settings**, go to **Basic Display** and click **Create New App**.
4. Fill in the required details:
    - **Valid OAuth Redirect URIs** → Use `https://localhost/` for testing.
    - **Deauthorize Callback URL** → Enter a placeholder URL (`https://example.com`).
5. Click **Save Changes**.

## 4️⃣ Generate an Access Token Using OAuth2

**What is OAuth2?**

OAuth2 is a secure authentication method that allows your app to access Instagram data without exposing your password.

**Steps to Get Access Token:**

1. Go to **Instagram Graph API > Basic Display**.
2. Click on **"Generate Access Token"**.
3. Follow the on-screen instructions to grant permissions.
4. After approval, **copy the Access Token**.

🔹 **Important:** Instagram Access Tokens expire. You might need to refresh them periodically.

## 5️⃣ Copy Your Credentials

Once you have successfully generated credentials, **save them** securely:

- **App ID** → Used for API requests.
- **App Secret** → Used for authentication.
- **Access Token** → Grants access to Instagram data.

For security, **DO NOT** hardcode these in your code. Instead, store them as environment variables.

---

# Step 3: Fetch Instagram Data using API

## 1️⃣ Install Required Libraries

Before running the script, make sure you have the required Python libraries installed.

### Run this command in Jupyter Notebook or Terminal:

```
pip install requests pandas
```

📌 Libraries Used:

- `requests` → To make API calls
- `pandas` → To organize data in tabular form (DataFrame)
- `json` → To process JSON responses from the API
- `os` → To securely handle environment variables

## 2️⃣ Set Your Instagram API Credentials

To interact with the Instagram API, you need an Access Token.(OAuth Token)

- How to Securely Store Your Access Token?

⚠ DO NOT hardcode your credentials in the script. Instead, store them in environment variables. Set Environment Variable (One-Time Setup)

**For Windows (Command Prompt):**
set INSTAGRAM_ACCESS_TOKEN=your_access_token_here

**For Windows (PowerShell):**
`$env:INSTAGRAM_ACCESS_TOKEN="your_access_token_here"`

## 3️⃣ Python Code to Fetch Instagram Data

### Step i: Import Required Libraries

```
import requests

import os

import json

import pandas as pd
```

`requests` → Used to make HTTP requests to the Instagram API.
`os` → Used to get API credentials from environment variables.
`json` → Helps process API responses in JSON format.
`pandas` → Converts API response into a DataFrame for easy analysis.

### Step ii: Set API Credentials

```
ACCESS_TOKEN = os.environ.get("INSTAGRAM_ACCESS_TOKEN")
USERNAME = "your_instagram_username"
```

`ACCESS_TOKEN` → Retrieves the Instagram access token securely.
`USERNAME` → Your Instagram username (not used in API calls but useful for reference).

### Step iii: Get Your Instagram User ID

```
user_url =
f"https://graph.instagram.com/v18.0/me?fields=id,username&acce
ss_token={ACCESS_TOKEN}"

response = requests.get(user_url)

user_data = response.json()

user_id = user_data["id"]
```

**API URL Explanation:**

- `https://graph.instagram.com/v18.0/me` → Fetches your Instagram user details.
- `fields=id,username` → Requests **user ID and username**.
- `access_token={ACCESS_TOKEN}` → Sends the API token for authentication.

🔹 **How It Works:**

- Makes an HTTP GET request to fetch **user data**.
- Converts the API response to a **Python dictionary (`json()`)**.
- Extracts the **user ID** (needed to fetch posts).

## Step iv: Fetch Instagram Posts Data

```
posts_url = f"https://graph.instagram.com/v18.0/{user_id}/media?fields=id,caption,media_type,media_url,like_count,comments_count,timestamp&access_token={ACCESS_TOKEN}"

posts_response = requests.get(posts_url)

posts_data = posts_response.json()
```

**API URL Explanation:**

- `https://graph.instagram.com/v18.0/{user_id}/media` → Fetches all posts for the given user.
- `fields=` → Specifies **which data fields to retrieve**:
  - `id` → Post ID
  - `caption` → Text content of the post
  - `media_type` → Type of content (image, video, carousel)
  - `media_url` → Direct link to the post image/video
  - `like_count` → Total likes on the post
  - `comments_count` → Number of comments
  - `timestamp` → Date and time of the post

**How It Works:**

- Makes an API request to fetch **all posts** by the user.
- Converts the JSON response to a Python dictionary.

**Step 4: Store Data in a Database**

**Using SQLite with SQLAlchemy**

python

CopyEdit

```
from sqlalchemy import create_engine


# Create SQLite database
engine =
create_engine("sqlite:///instagram_data.db")
posts_df.to_sql("instagram_posts", engine,
if_exists="replace", index=False)


print("Data stored in database successfully!")
```

✅ This stores Instagram data in an SQLite database for further analysis.

---

**Step 5: Create a Dashboard in Tableau/Power BI**

**1. Load Data:**

- Open **Tableau/Power BI**
- Connect to **CSV File** or **Database (SQLite/PostgreSQL)**

**2. Create Visualizations:**

📊 **Engagement Analysis:** Bar chart for likes/comments per post

📈 **Post Frequency:** Line chart for post activity over time
📌 **Most Engaging Post:** Highlight top-performing posts

---

## Final Deliverables

📁 **Python Script** – Fetches Instagram data
📂 **Database (SQLite/PostgreSQL)** – Stores historical data
📊 **Tableau/Power BI Dashboard** – Visualizes social media engagement

---

## Next Steps

✔ Expand to multiple accounts
✔ Integrate other social media platforms (Twitter, LinkedIn)
✔ Apply ML models for sentiment analysis