

# Dhaka University of Engineering & Technology, Gazipur

Project Name: COVID-19 Data Dashboard

Student Name: Tanvir Ahmed

ID: 2204078

**Description:** Students can create a dashboard that collects and visualizes COVID-19 data using an API. The project could include visualizations of cases, recoveries, and vaccinations over time in different countries.

**Skills Used:** API, Pandas, Matplotlib/Seaborn, File Handling, Data Structures.

## ১. লাইব্রেরি ইম্পোর্ট করা

```
import requests
```

```
import pandas as pd
```

```
import tkinter as tk
```

```
from tkinter import messagebox
```

```
import matplotlib.pyplot as plt
```

- **requests:** এটি একটি প্যাকেজ যা HTTP অনুরোধ পাঠানোর জন্য ব্যবহার করা হয়। এর মাধ্যমে আমরা COVID-19 ডেটা API থেকে তথ্য সংগ্রহ করতে পারি।
- **pandas:** এটি একটি প্যাকেজ যা ডেটা বিশ্লেষণ এবং ম্যানিপুলেশনের জন্য ব্যবহৃত হয়। pandas ডেটা স্ট্রাকচার (যেমন DataFrame) পরিচালনা করতে সাহায্য করে।
- **tkinter:** এটি একটি GUI লাইব্রেরি যা আপনাকে গ্রাফিক্যাল ইউজার ইন্টারফেস তৈরি করতে সাহায্য করে।
- **messagebox:** tkinter এর একটি অংশ যা পপ-আপ মেসেজ দেখানোর জন্য ব্যবহার করা হয়।
- **matplotlib.pyplot:** এটি একটি গ্রাফ প্লটিং লাইব্রেরি যা আমাদের ডেটা প্লট বা গ্রাফ আকারে প্রদর্শন করতে সহায়ক।

## ২. গ্লোবাল ভেরিয়েবল

```
df = None
```

- **df:** এটি একটি গ্লোবাল ভেরিয়েবল যা COVID-19 ডেটা ধারণ করবে। এটি pandas DataFrame এর আকারে থাকবে এবং API থেকে অথবা CSV ফাইল থেকে ডেটা লোড হবে।

### ৩. COVID-19 ডেটা API থেকে ফেচ করা

```
def fetch_country_data(country):
```

```
    url = f"https://disease.sh/v3/covid-19/historical/{country}?lastdays=all"
```

```
    response = requests.get(url)
```

```
    if response.status_code == 200:
```

```
        data = response.json()
```

```
        return data['timeline']
```

```
    else:
```

```
        print(f"Error fetching data for {country}. Status code: {response.status_code}")
```

```
    return None
```

- **fetch\_country\_data:** এই ফাংশনটি একটি দেশ নাম নেয় এবং সেই দেশের COVID-19 হিস্টোরিকাল ডেটা API থেকে ফেচ করে।
  - URL টি API এর পাথ যার মাধ্যমে COVID-19 এর সমস্ত ডেটা ফেচ করা হয়।
  - যদি API সফলভাবে ডেটা ফেরত দেয় (status code 200), তবে এটি ডেটা সংগ্রহ করে এবং "timeline" অংশটি ফিরিয়ে দেয়।
  - যদি কোনো ত্রুটি ঘটে, তবে একটি ত্রুটির মেসেজ প্রদর্শন করে।

### ৪. CSV ফাইলে ডেটা সেভ করা

```
def save_data_to_csv(country, data):
```

```
    cases = pd.DataFrame.from_dict(data['cases'], orient='index', columns=['cases'])
```

```
    deaths = pd.DataFrame.from_dict(data['deaths'], orient='index', columns=['deaths'])
```

```
    recovered = pd.DataFrame.from_dict(data['recovered'], orient='index', columns=['recovered'])
```

```
    vaccinations = pd.DataFrame.from_dict(data.get('vaccinated', {}), orient='index',  
columns=['vaccinations'])
```

```
    if vaccinations.empty:
```

```
        vaccinations = pd.DataFrame(index=cases.index, columns=['vaccinations'])
```

```
        vaccinations['vaccinations'] = 0
```

```
    df = pd.concat([cases, deaths, recovered, vaccinations], axis=1)
```

```
    df.index = pd.to_datetime(df.index)
```

```
    df = df.reset_index().rename(columns={"index": "date"})
```

```
file_name = f"{country}_covid_data.csv"
df.to_csv(file_name, index=False)
print(f"Data for {country} has been saved to {file_name}.")
```

- **save\_data\_to\_csv:** এই ফাংশনটি ফেচ করা COVID-19 ডেটাকে একটি CSV ফাইলে সেভ করে।
  - cases, deaths, recovered, এবং vaccinations এর জন্য আলাদা DataFrame তৈরি করা হয়।
  - যদি vaccinations ডেটা না থাকে, তবে সেটি 0 দিয়ে পূর্ণ করা হয়।
  - সব ডেটাকে একত্রিত করে একটি DataFrame তৈরি করা হয় এবং এটি CSV ফাইলে সেভ করা হয়।

## ৫. CSV ফাইল থেকে ডেটা লোড করা

```
def load_country_data_from_csv(country):
    global df
    try:
        df = pd.read_csv(f"{country}_covid_data.csv")
        df['date'] = pd.to_datetime(df['date'])
        print(f"Data for {country} loaded from CSV.")
        return df
    except FileNotFoundError:
        messagebox.showerror("Error", f"CSV file for {country} not found.")
        return None
```

- **load\_country\_data\_from\_csv:** এই ফাংশনটি একটি CSV ফাইল থেকে ডেটা লোড করে।
  - যদি CSV ফাইলটি পাওয়া যায়, তবে তা DataFrame আকারে লোড করা হয় এবং তারিখের ফরম্যাট সংশোধন করা হয়।
  - যদি ফাইলটি না পাওয়া যায়, তাহলে একটি ত্রুটির মেসেজ প্রদর্শন করা হয়।

## ৬. ডেটা ফিল্টার করা (তারিখের রেঞ্জ অনুযায়ী)

```
def filter_by_date_range(df, start_date, end_date):
    start_date = pd.to_datetime(start_date)
    end_date = pd.to_datetime(end_date)
    return df[(df['date'] >= start_date) & (df['date'] <= end_date)]
```

- **filter\_by\_date\_range:** এই ফাংশনটি তারিখের একটি নির্দিষ্ট রেঞ্জ অনুযায়ী ডেটা ফিল্টার করে।
  - start\_date এবং end\_date কে datetime আকারে রূপান্তরিত করা হয়।
  - ফিল্টার করা ডেটা শুধুমাত্র সেই তারিখগুলির জন্য ফিরে আসে যা রেঞ্জের মধ্যে পড়ে।

## ৭. নির্দিষ্ট তারিখ অনুযায়ী ডেটা ফিল্টার করা

```
def filter_by_specific_date(df, specific_date):
    specific_date = pd.to_datetime(specific_date)
    return df[df['date'] == specific_date]
```

- **filter\_by\_specific\_date:** এই ফাংশনটি একটি নির্দিষ্ট তারিখের ডেটা ফিল্টার করে।
  - specific\_date কে datetime আকারে রূপান্তরিত করা হয় এবং তারপর শুধুমাত্র ওই তারিখটির জন্য ডেটা ফিরে আসে।

## ৮. নির্দিষ্ট বছর অনুযায়ী ডেটা ফিল্টার করা

```
def filter_by_year(df, year):
    df['year'] = df['date'].dt.year
    return df[df['year'] == year]
```

- **filter\_by\_year:** এই ফাংশনটি ডেটার মধ্যে একটি নির্দিষ্ট বছর অনুযায়ী ফিল্টার করে।
  - তারিখ থেকে বছর বের করে সেটি ডেটাফ্রেমের একটি নতুন কলামে রাখা হয় এবং তারপর ওই বছর অনুযায়ী ডেটা ফিল্টার করা হয়।

## ৯. COVID-19 ট্রেন্ড প্লট করা

```
def plot_covid_trends(df, title, country):
    if df.empty:
        messagebox.showinfo("No Data", "No data available to plot.")
        return
    plt.figure(figsize=(12, 6))
    plt.plot(df['date'], df['cases'], label='Cases', color='blue', linestyle='-', linewidth=2)
    plt.plot(df['date'], df['recovered'], label='Recoveries', color='green', linestyle='-', linewidth=2)
    plt.plot(df['date'], df['deaths'], label='Deaths', color='red', linestyle='-', linewidth=2)
    plt.plot(df['date'], df['vaccinations'], label='Vaccinations', color='purple', linestyle='-', linewidth=2)
    plt.title(f"{title} for {country.title()} ", fontsize=16)
    plt.xlabel("Date", fontsize=12)
```

```
plt.ylabel("Counts", fontsize=12)
plt.xticks(rotation=45)
plt.grid(True, linestyle='--', alpha=0.6)
plt.legend()
plt.tight_layout()
plt.show()
```

- **plot\_covid\_trends:** এই ফাংশনটি COVID-19 এর বিভিন্ন ট্রেন্ড (কেস, মৃত্যু, সুস্থতা, টিকাকরণ) একটি গ্রাফে প্লট করে।
  - এটি একটি নতুন চিত্র তৈরি করে এবং প্রতিটি ট্রেন্ড (কেস, সুস্থতা, মৃত্যু, টিকাকরণ) আলাদা রঙে এবং লাইন স্টাইল সহ প্লট করে।
  - গ্রাফের শিরোনাম, এক্স এবং ওয়াই অক্ষের লেবেল সেট করা হয় এবং একটি গ্রিড এবং লেজেন্ড যোগ করা হয়।

## ১০. GUI এর মাধ্যমে ইউজারের সাথে ইন্টারঅ্যাকশন

```
def run_dashboard():
```

```
    def search_country():
```

```
        global df
```

```
        country = country_entry.get().strip().lower()
```

```
        if not country:
```

```
            messagebox.showwarning("Input Error", "Please enter a country name.")
```

```
            return
```

```
        # Logic for searching or fetching country data (from CSV or API)
```

```
        ...
```

```
    # Other GUI components and logic for filtering and plotting the data
```

```
    ...
```

- **run\_dashboard:** এটি একটি GUI ফাংশন যা ইউজারকে একটি দেশের নাম ইনপুট দেওয়ার জন্য একটি উইন্ডো প্রদান করে।
  - ইউজার যখন দেশের নাম প্রবেশ করবে, তখন এটি দেশের ডেটা ফেচ করে বা CSV থেকে লোড করে।
  - ইউজার ডেটা ফিল্টার করার জন্য তারিখের রেঞ্জ বা নির্দিষ্ট তারিখ নির্বাচন করতে পারে।
  - এরপর নির্বাচিত ডেটা অনুযায়ী প্লট তৈরি হবে।

### কোডের ব্যবহারের পদ্ধতি:

1. **স্ক্রিপ্ট রান করুন:** স্ক্রিপ্টটি চালান, একটি উইন্ডো খুলবে যেখানে আপনি দেশের নাম (যেমন "India", "USA") ইনপুট করতে পারবেন।
2. **দেশ অনুসন্ধান করুন:** যখন আপনি একটি দেশ লিখবেন এবং "Search" ক্লিক করবেন, তখন এটি প্রথমে CSV ফাইলে ডেটা খুঁজবে। যদি না থাকে, তবে এটি API থেকে ডেটা ফেচ করবে।
3. **ফিল্টার নির্বাচন করুন:** দেশের ডেটা লোড হওয়ার পরে, আপনি ডেটা ফিল্টার করতে পারবেন (তারিখের রেঞ্জ, নির্দিষ্ট তারিখ, বা বছর)।
4. **গ্রাফ দেখুন:** ফিল্টার করার পরে, নির্বাচিত ডেটা অনুযায়ী গ্রাফ প্রদর্শিত হবে।

### উপসংহার:

এই কোডটি আপনাকে দেশের COVID-19 ডেটা অনুসন্ধান করতে, ফিল্টার করতে, এবং গ্রাফে প্রদর্শন করতে সাহায্য করবে। এটি CSV ফাইলেও ডেটা সেভ করে রাখবে যাতে পরবর্তীতে দ্রুত কাজ করতে পারেন।