

**Міністерство освіти і науки України**  
**Львівський національний університет імені Івана Франка**

Факультет електроніки та комп'ютерних технологій

**Звіт**

Про виконання

Індивідуальної роботи №1

З курсу «Системи опрацювання даних»

«ВІ завдання: побудувати дашборд та написати історію-пояснення до його змісту»

Виконав:

Студент групи Фес-21

Осадчук Дмитро

Львів-2025

## Вступ:

У даній роботі було використано бібліотеки Python — Dash та Plotly для створення інтерактивного дашборду з аналізу поширення пандемії COVID-19 на основі відкритих даних від Johns Hopkins University та Our World in Data.

Метою аналізу є дослідження динаміки захворюваності, смертності та вакцинації в різних країнах світу. Особлива увага приділяється порівнянню епідеміологічних показників між різними регіонами, впливу вакцинації на динаміку захворюваності та смертності, а також виявленню різних хвиль пандемії в хронологічному порядку.

Проведений аналіз дозволяє виявити ключові закономірності поширення COVID-19, оцінити ефективність протиепідемічних заходів, а також сформулювати загальне уявлення про взаємозв'язки між різними показниками пандемії за допомогою візуалізації.

## Джерела даних:

1. Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE) - один з найбільш авторитетних та актуальних джерел даних про пандемію COVID-19. Репозиторій містить щоденні дані про кількість підтверджених випадків, смертей та одужань по всьому світу з початку пандемії.
  - Посилання на репозиторій: <https://github.com/CSSEGISandData/COVID-19>
2. Our World in Data - надійне джерело даних про вакцинацію проти COVID-19, яке містить інформацію про відсоток вакцинованого населення, кількість введених доз та інші показники, пов'язані з вакцинацією.
  - Посилання на репозиторій: <https://github.com/owid/covid-19-data>

## Пояснення графіків

- Динаміка нових випадків COVID-19 (лінійний графік) – відображає 7-денне ковзне середнє нових випадків захворювання для обраних країн. Згладжування дозволяє нівелювати щоденні коливання та виявити загальні тренди захворюваності.
- Динаміка смертей від COVID-19 (лінійний графік) – показує 7-денне ковзне середнє нових смертей для обраних країн. Дозволяє відстежувати зміни в летальності та порівнювати їх з динамікою захворюваності.
- Рівень вакцинації (лінійний графік) – візуалізує відсоток повністю вакцинованого населення в обраних країнах з плином часу. Цей графік допомагає оцінити прогрес у вакцинації та порівняти його з епідеміологічними показниками.
- Порівняння країн (стовпчикова діаграма) – відображає загальну кількість смертей та підтверджених випадків для обраних країн на певну дату. Дозволяє порівняти абсолютний вплив пандемії на різні країни.

## Питання, на які відповідає дашборд

1. Яка динаміка нових випадків і смертей від COVID-19 у різних країнах та регіонах?
2. Як змінювалась інтенсивність пандемії з часом і які хвилі захворюваності можна виділити?
3. Як вплинула вакцинація на динаміку захворюваності та смертності?
4. Які країни найбільш та найменш постраждали від пандемії відносно їх населення?
5. Чи є кореляція між рівнем вакцинації та кількістю нових випадків або смертей?
6. Як відрізняється смертність від COVID-19 у різних країнах?
7. Чи спостерігається синхронність у хвилях захворюваності між різними країнами?
8. Як швидко країни досягли високого рівня вакцинації і як це співвідноситься з динамікою захворюваності?
9. Які країни найбільш ефективно боролися з пандемією, якщо порівнювати показники захворюваності та смертності?
10. Чи можна виявити сезонні закономірності у поширенні COVID-19?

## Код роботи:

```
import dash
from dash import dcc, html
from dash.dependencies import Input, Output
import plotly.express as px
import plotly.graph_objects as go
import pandas as pd
import numpy as np

# Завантаження даних
# Використовуємо прямі посилання на репозиторії з даними
url_confirmed = 'https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_gis/covid_19_data/covid_19_timeseries.csv'
url_deaths = 'https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_gis/covid_19_data/covid_19_timeseries_deaths.csv'
url_recovered = 'https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_gis/covid_19_data/covid_19_timeseries_recovered.csv'
url_vaccinations = 'https://raw.githubusercontent.com/owid/covid-19-data/master/github-data/covid19-vc.csv'

# Функція для завантаження та попередньої обробки даних
def load_data():
    # Завантаження даних про підтверджені випадки, смерті та одужання
    df_confirmed = pd.read_csv(url_confirmed)
    df_deaths = pd.read_csv(url_deaths)
    df_recovered = pd.read_csv(url_recovered)

    # Підготовка даних для аналізу
    # Перетворення з широкого формату в довгий
    dates_confirmed = df_confirmed.columns[4:]

    # Підготовка даних про підтверджені випадки
    confirmed_long = pd.melt(
        df_confirmed,
        id_vars=['Province/State', 'Country/Region', 'Lat', 'Long'],
        value_vars=dates_confirmed,
        var_name='date',
        value_name='confirmed'
    )

    # Підготовка даних про смерті
    deaths_long = pd.melt(
        df_deaths,
        id_vars=['Province/State', 'Country/Region', 'Lat', 'Long'],
        value_vars=dates_confirmed,
        var_name='date',
        value_name='deaths'
    )
```

```

# Об'єднання даних
covid_data = confirmed_long.merge(
    deaths_long,
    on=['Province/State', 'Country/Region', 'Lat', 'Long', 'date']
)

# Перетворення дати в правильний формат
covid_data['date'] = pd.to_datetime(covid_data['date'])

# Агрегація даних по країнах
country_data = covid_data.groupby(['Country/Region', 'date']).agg({
    'confirmed': 'sum',
    'deaths': 'sum'
}).reset_index()

# Обчислення нових випадків і смертей
country_data = country_data.sort_values(['Country/Region', 'date'])
country_data['new_cases'] = country_data.groupby('Country/Region')['confirmed'].diff().fillna(0)
country_data['new_deaths'] = country_data.groupby('Country/Region')['deaths'].diff().fillna(0)

# Виправлення негативних значень
country_data['new_cases'] = country_data['new_cases'].clip(lower=0)
country_data['new_deaths'] = country_data['new_deaths'].clip(lower=0)

# Обчислення 7-денних рухомих середніх
country_data['new_cases_7day_avg'] = country_data.groupby('Country/Region')['new_cases'].rolling(7).mean().reset_index(0, drop=True)
country_data['new_deaths_7day_avg'] = country_data.groupby('Country/Region')['new_deaths'].rolling(7).mean().reset_index(0, drop=True)

# Завантаження даних про вакцинацію
try:
    df_vaccinations = pd.read_csv(url_vaccinations)
    df_vaccinations['date'] = pd.to_datetime(df_vaccinations['date'])

    # Вибір найбільш важливих колонок
    vac_cols = ['location', 'date', 'total_vaccinations_per_hundred', 'people_vaccinated_per_hundred', 'people_fully_vaccinated_per_hundred']
    vac_data = df_vaccinations[vac_cols].copy()

    # Перейменування для узгодження з нашими даними
    vac_data = vac_data.rename(columns={'location': 'Country/Region'})

    # Об'єднання даних про випадки і вакцинацію
    combined_data = pd.merge(
        country_data,
        vac_data,
        on=['Country/Region', 'date'],
        how='left'
    )
except Exception as e:
    print(f"Помилка при завантаженні даних вакцинації: {e}")
    combined_data = country_data

return combined_data

# Завантаження даних
df = load_data()

# Список країн для вибору
countries = sorted(df['Country/Region'].unique())

# Ініціалізація додатку Dash
app = dash.Dash(__name__, meta_tags=[{"name": "viewport", "content": "width=device-width"}])
server = app.server

# Заголовок додатку
app.title = "COVID-19 Дашборд"

# Створення макету дашборду
app.layout = html.Div([
    # Заголовок
    html.Div([
        html.H1("COVID-19 Глобальний Дашборд", style={"text-align": "center"}),
        html.P("Аналіз поширення COVID-19, смертності та вакцинації у світі",
            style={"text-align": "center", "font-size": "1.2em"}),
    ], className="header"),

    # Вибір країн і періоду часу
    html.Div([
        html.Div([
            html.Div([
                html.Label("Виберіть країни:"),
                dcc.Dropdown(
                    id='country-dropdown',
                    options=[{'label': country, 'value': country} for country in countries],
                    value=['US', 'Ukraine', 'Italy', 'Germany', 'India'],
                    multi=True
                ),
            ], className="six columns"),

```

```

            ], className="six columns"),
        ], className="six columns"),
    ], className="six columns"),

    # Вибір країн і періоду часу
    html.Div([
        html.Div([
            html.Div([
                html.Label("Виберіть країни:"),
                dcc.Dropdown(
                    id='country-dropdown',
                    options=[{'label': country, 'value': country} for country in countries],
                    value=['US', 'Ukraine', 'Italy', 'Germany', 'India'],
                    multi=True
                ),
            ], className="six columns"),

```

```

        html.Div([
            html.Label("Виберіть період:"),
            dcc.DatePickerRange(
                id='date-picker',
                min_date_allowed=df['date'].min(),
                max_date_allowed=df['date'].max(),
                start_date=df['date'].max() - pd.DateOffset(months=6),
                end_date=df['date'].max(),
                display_format='YYYY-MM-DD'
            ),
        ], className="six columns"),
    ], className="row filters"),

    # Графіки для підтверджених випадків
    html.Div([
        html.H2("Динаміка підтверджених випадків COVID-19"),
        dcc.Graph(id='cases-graph'),
    ], className="row graph-container"),

    # Графіки для смертей
    html.Div([
        html.H2("Динаміка смертей від COVID-19"),
        dcc.Graph(id='deaths-graph'),
    ], className="row graph-container"),

    # Дані про вакцинацію
    html.Div([
        html.H2("Рівень вакцинації"),
        dcc.Graph(id='vac-graph'),
    ], className="row graph-container"),

    # Порівняння країн
    html.Div([
        html.H2("Порівняння країн"),
        dcc.Graph(id='country-comparison'),
    ], className="row graph-container"),

    # Інформаційний розділ
    html.Div([
        html.H2("Аналіз даних COVID-19"),
        html.P([
            "Цей дашборд показує динаміку поширення COVID-19, смертності та вакцинації в різних країнах світу. ",
            "Дані оновлюються регулярно і отримані з репозиторію ",
            html.A("Johns Hopkins University CSSE"),
        ])
    ])

```

```

        " і бази даних вакцинації від ",
        html.A("Our World in Data",
            href="https://github.com/owid/covid-19-data",
            target="_blank"),
        ", "
    ]),
    ], className="row", style={"margin-top": "20px"}),

    # Футер
    html.Footer([
        html.P("Дані актуальні на: " + str(df['date'].max().strftime('%Y-%m-%d')),
            style={"text-align": "center"})
    ])
])

# Зворотні виклики для оновлення графіків

# Графік підтверджених випадків
@app.callback(
    Output('cases-graph', 'figure'),
    [Input('country-dropdown', 'value'),
     Input('date-picker', 'start_date'),
     Input('date-picker', 'end_date')]
)
def update_cases_graph(selected_countries, start_date, end_date):
    filtered_df = df[
        (df['Country/Region'].isin(selected_countries)) &
        (df['date'] >= start_date) &
        (df['date'] <= end_date)
    ]

    fig = px.line(
        filtered_df,
        x='date',
        y='new_cases_7day_avg',
        color='Country/Region',
        title='Нові випадки COVID-19 (7-денне середнє)',
        labels={'new_cases_7day_avg': 'Нові випадки (7-денне середнє)', 'date': 'Дата', 'Country/Region': 'Країна'})

    fig.update_layout(
        xaxis_title='Дата',
        yaxis_title='Нові випадки (7-денне середнє)',
        legend_title='Країна',
        hovermode='x unified'
    )

```

```

# Графік смертей
@app.callback(
    Output('deaths-graph', 'figure'),
    [Input('country-dropdown', 'value'),
     Input('date-picker', 'start_date'),
     Input('date-picker', 'end_date')]
)
def update_deaths_graph(selected_countries, start_date, end_date):
    filtered_df = df[
        (df['Country/Region'].isin(selected_countries)) &
        (df['date'] >= start_date) &
        (df['date'] <= end_date)
    ]

    fig = px.line(
        filtered_df,
        x='date',
        y='new_deaths_7day_avg',
        color='Country/Region',
        title='Нові смерті від COVID-19 (7-денне середнє)',
        labels={'new_deaths_7day_avg': 'Нові смерті (7-денне середнє)', 'date': 'Дата', 'Country/Region': 'Країна'})

    fig.update_layout(
        xaxis_title='Дата',
        yaxis_title='Нові смерті (7-денне середнє)',
        legend_title='Країна',
        hovermode='x unified'
    )

    return fig

# Графік вакцинації
@app.callback(
    Output('vac-graph', 'figure'),
    [Input('country-dropdown', 'value'),
     Input('date-picker', 'start_date'),
     Input('date-picker', 'end_date')]
)
def update_vac_graph(selected_countries, start_date, end_date):
    filtered_df = df[
        (df['Country/Region'].isin(selected_countries)) &
        (df['date'] >= start_date) &
        (df['date'] <= end_date)
    ]

    if 'people_fully_vaccinated_per_hundred' in filtered_df.columns:

```

```

# Фільтруємо дані
filtered_df = df[
    (df['Country/Region'].isin(selected_countries)) &
    (df['date'] == latest_date)
]

# Створюємо порівняльну діаграму
fig = px.bar(
    filtered_df,
    x='Country/Region',
    y='deaths',
    color='Country/Region',
    title=f'Загальна кількість смертей на {latest_date.strftime("%Y-%m-%d")}',
    labels={'deaths': 'Загальна кількість смертей', 'Country/Region': 'Країна'})

# Додаємо другу вісь з кількістю підтверджених випадків
fig.add_trace(
    go.Scatter(
        x=filtered_df['Country/Region'],
        y=filtered_df['confirmed'],
        mode='markers',
        marker=dict(size=15, color='red'),
        name='Загальна кількість випадків',
        yaxis='y2'
    )
)

# Налаштування макету
fig.update_layout(
    xaxis_title='Країна',
    yaxis_title='Загальна кількість смертей',
    yaxis2=dict(
        title='Загальна кількість випадків',
        titlefont=dict(color='red'),
        tickfont=dict(color='red'),
        anchor='x',
        overlaying='y',
        side='right'
    ),
    legend_title='Країна',
    hovermode='closest'
)

```

## COVID-19 Глобальний Дашборд

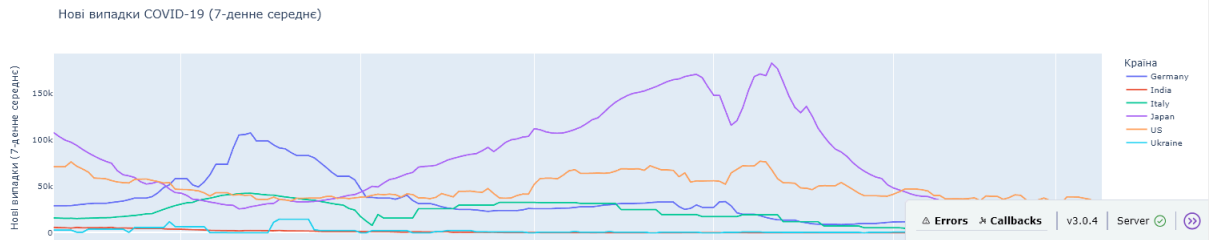
Аналіз поширення COVID-19, смертності та вакцинації у світі

Виберіть країни:

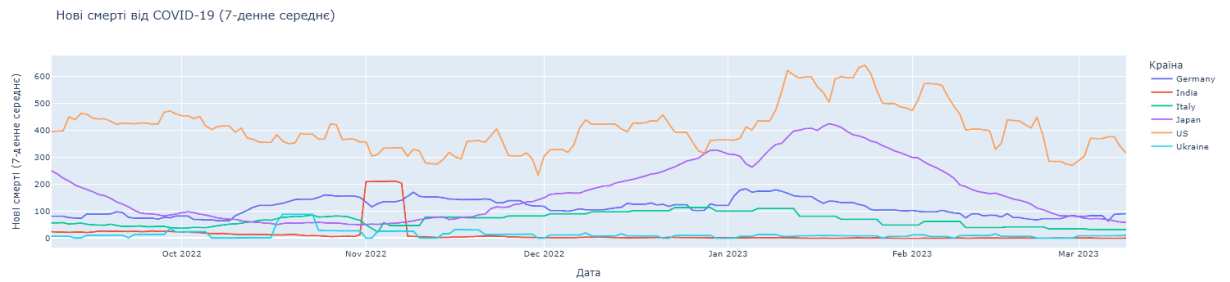
US × Ukraine × Italy × Germany × India × Japan ×

Виберіть період: 2022-09-09 → 2023-03-09

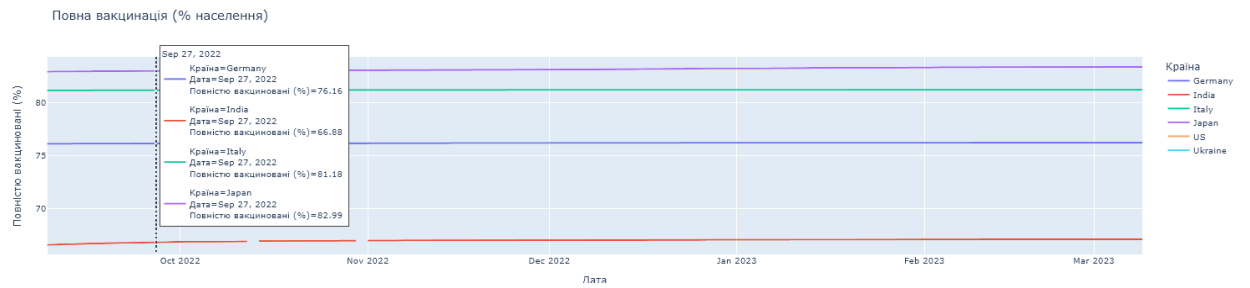
### Динаміка підтверджених випадків COVID-19



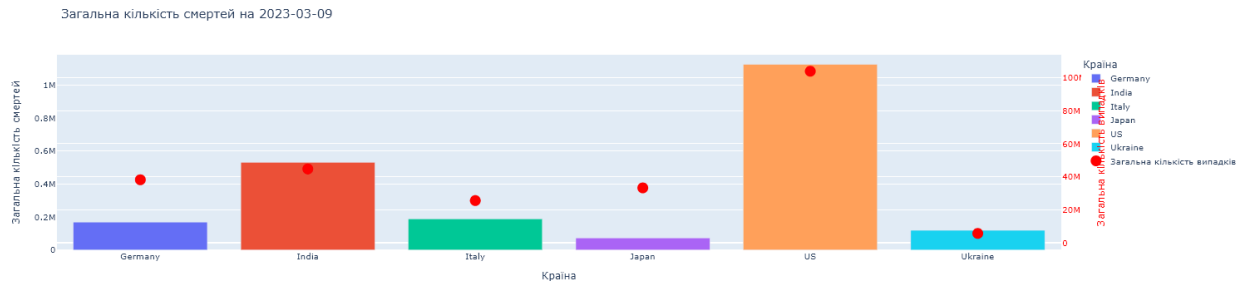
### Динаміка смертей від COVID-19



### Рівень вакцинації



### Порівняння країн



**Висновок:** У результаті виконання лабораторної роботи було створено інтерактивний дашборд, який дозволяє проаналізувати перебіг епідемії COVID-19 в різних країнах залежно від різних факторів. Візуалізація даних допомогла виявити, наприклад, що чим більший рівень вакцинації серед населення, тим менше нових випадків спалення хвороби.