

# COVID-19

HEALTH AND  
INFORMATION  
WORLD WIDE APP



## תיעוד תוכנה

מטרת האפליקציה היא לעזור למשתמש לקבל מידע לגבי נגיף הקורונה במדינות ברחבי העולם, בהתאם לתאריך מסוים. האפליקציה מציגה נתונים סטטיסטיים, ונותנת אינפורמציה לגבי קצב התקדמות הנגיף ביחס למדינות נוספות.

### Data Base

השתמשנו בשני מאגרי מידע שמכילים נתונים על הנגיף לפי תאריך ומדינה. המאגר הראשון מכיל עבור כל תאריך את נתוני התחלואה של כ-200 מדינות, כולל מידע בין התאריכים 1.1.20, עד אמצע נובמבר 2020. המאגר השני מכיל נתונים נלווים על מצב מוסדות הלימוד במדינה, כולל מידע בין מרץ עד אמצע נובמבר 2020.

- מאגר מידע על בתי ספר - <https://en.unesco.org/covid19/educationresponse>
- מאגר מידע על נתוני התחלואה - <https://ourworldindata.org/coronavirus-source-data>

בחלק מהכנסת המידע, ערכנו מספר פעולות תיקון על מאגרי הנתונים כמו:

- ניקוי עמודות מיותרות.
- עיצוב תאריכים בפורמט תאריך מתאים.
- הגבלת תווים.
- הורדת מידע לא רלוונטי תוך כדי בדיקת ערכיות המידע.

## מבנה ה- Data Base

תיעוד של הטבלאות:

באפליקציה שלנו קיימות 3 טבלאות, המפתחות מסומנים בכחול. **מפתחות זרים** - יש לנו טבלה ראשית (covid country), ובכל טבלה אחרת שהיא לא הטבלה הראשית הזאת יש בה שימוש ב iso code ולכן הגדרנו אותו כמפתח זר (הוא המפתח של הטבלה הראשית, וככה ניתן לשמור על הקשרים). מבחינת הקישור של מפתח זר- הטבלאות אצלנו לא מתעדכנות (אין מחיקה או הוספה של ערכים), וגם אם היה היינו רוצים לשמור על הטבלה העיקרית בשלמותה. ולכן לא הכרחנו התנהגות במקרים אלו.

### Covid Country

בטבלה זו קיימות יש לנו 3 עמודות:

- **Iso\_code** – קוד המדינה. עמודה זו היא מסוג- varchar(10) PK.
- **Location** – שם המדינה. עמודה זו היא מסוג- varchar(45)

### Covid data

בטבלה זו קיימות 7 עמודות:

- **Iso\_code** – קוד המדינה. עמודה זו היא מסוג- varchar(10) PK
- **Date** – תאריך. עמודה זו היא מסוג- date PK
- **total\_cases** – מספר מקרים כללי. עמודה זו היא מסוג- int
- **new\_cases** – מספר מקרים חדשים. עמודה זו היא מסוג- int
- **total\_death** – מספר נפטרים כולל. עמודה זו היא מסוג- int
- **new\_death** – מספר נפטרים חדשים. עמודה זו היא מסוג- int

\*מבחינת הנתונים, הם לא עוברים את הגודל של int, ולכן בחרנו להכניס int. במקרה שהטבלה הייתה מתעדכנת היינו מכניסות big int.

### education

בטבלה זו קיימות 3 עמודות:

- **Iso\_code** – קוד המדינה. עמודה זו היא מסוג- varchar(10) PK

Covid19- Health and information app  
מגישות: שירה תורג'מן, דורין דומין ונעה אלישמרני

- **Date** – תאריך.  
עמודה זו היא מסוג -date PK.
- **Status education** - סטטוס לגבי מוסדות החינוך במדינה.  
עמוד זו היא מסוג varchar(25).

## Code

האפליקציה שלנו רשומה בשפת פייתון 3.  
הקוד שלנו מחולק ל 4 קבצים:

הקוד שלנו בנוי באופן מודולרי:  
שני קבצים אחראים על הגישה לmysql.  
קובץ אחד על הUI.  
קובץ אחד על שכבת הלוגיקה, יש בה את פונקציית main.  
\* כל המידע של השאילתות נשמר במילונים, הקוד של UI לא ניגש ל-mysql בעצמו אלא מקבל את כל המידע מהמילונים.

create\_db - בקובץ זה קיימות 4 פונקציות, ליצירה והכנסת המידע לטבלאות.

- Create\_db\_first\_time  
פונקציה זו אחראית על יצירת מאגר המידע בפעם הראשונה, ועל יצירת הסכמה והטבלאות (באופן ריק).
- create\_scheme\_and\_table  
יצירת הטבלאות והגדרת העמודות של כל טבלה, יחד עם הגדרת סוגי הטבלאות.
- Initialization\_covid\_country\_and\_covid\_data\_tables  
בפונקציה זו אנחנו מכניסים את המידע לשתי הטבלאות covid\_country, covid\_data.
- Initialization\_education\_table  
בפונקציה זו אנחנו מכניסים את המידע לטבלה education\_table.

## Corona\_app

קובץ זה הוא main של האפליקציה.  
בשורה 8 אנו קוראים לקוד create\_db יוצרים את מאגר הנתונים, ואנו עושים זאת רק פעם אחת, לאחר מכן נכניס שורה זו להערה.  
לאחר מכן ניצור את האפליקציה עצמה בעזרת קריאה לפונקציה create\_window\_app בקובץ UI\_corona\_app.  
ולבסוף נסגור את MySQL.

## mysql queries

בקובץ זה אנחנו יוצרים את השאילתות של האפליקציה.

- Connect\_mysql  
נתחבר לשרת ע"י הכנסת שם משתמש וסיסמא.
- Close\_mysql  
סגירה של השרת.

- Run\_all\_queries

פונקציה זו מריצה את כל השאילתות לפי הסדר, ומחזירה את המידע הנחוץ כדי שנוכל להציג אותו בנתונים גרפיים בעמוד השני.

### השאילתות:

- Iso\_query

הבאת iso coden של המדינה שנבחרה (נוצר בשביל שיהיה לנו את iso coden עבור שאר השאילתות).

```
SELECT iso_code FROM team09.covid_country WHERE location=%s
```

- Query1

כמות החולים במדינה שנבחרה, עד התאריך שנבחר.

```
"SELECT DISTINCT total_cases FROM team09.covid_data WHERE date=%s  
AND iso=%s
```

- Query2

כמות חולים חדשים ביממה האחרונה, של התאריך שנבחר, במדינה שנבחרה.

```
SELECT DISTINCT new_cases FROM team09.covid_data WHERE date=%s AND  
iso=%s
```

- Query3

מספר הנפטרים במדינה מסוימת, עד התאריך שנבחר.

```
SELECT DISTINCT total_death FROM team09.covid_data WHERE date=%s  
AND iso=%s
```

- Query4

מספר המאומתים החדשים ביממה האחרונה, במדינה מסוימת.

```
SELECT DISTINCT new_death FROM team09.covid_data WHERE date=%s AND  
iso=%s
```

Query5 •

החזרת הסטטוס של מוסדות החינוך במדינה מסוימת בתאריך מסוים- אחת מ4 האופציות:  
.fully open, partially open, academic break, closed due to covid19

```
SELECT DISTINCT status_education FROM team09.education WHERE  
date=%s AND iso=%s
```

Query7 •

סה"כ חולים במדינה הנבחרת בכל אחד מהימים בשבוע האחרון.

```
SELECT date,total_cases FROM team09.covid_data WHERE date >= (%s -  
INTERVAL 7 DAY) AND date <= %s AND iso = %s;
```

Query10 •

בדיקת הסטטוס של מוסדות החינוך במדינה שנבחרה לפי התאריך שנבחר.

```
SELECT status_education FROM team09.education WHERE iso = %s AND  
date = %s
```

Prev\_query10 •

בדיקה האם המדינה שנבחרה נמצאת בטבלת מוסדות החינוך.

```
select Count(*) from team09.education where iso = %s
```

Sub\_query10 - שאילתא מורכבת. •

בדיקה האם אנחנו מעל או מתחת לממוצע שחושב בשאילתא query10\_get\_avg.

```
SELECT COUNT(*) FROM team09.covid_data WHERE iso = %s AND date =  
%s AND total_cases <=( SELECT AVG(" \  
"distinct total_cases) FROM team09.covid_data AS  
cd,team09.education AS ed WHERE cd.iso = \  
ed.iso AND cd.date = ed.date AND ed.iso = %s AND  
ed.status_education <> 'Fully open');
```

- Query10\_get\_avg

אם אין סטטוס בתאריך הנבחר, או שהסטטוס לא וודאי (חופשה\פתוח חלקית) נחשב לפי הממוצע של סה"כ החולים בזמן שהמדינה לא הייתה פתוחה באופן מלא.

```
SELECT AVG(" \
    "distinct total_cases) FROM team09.covid_data AS
cd,team09.education AS ed WHERE cd.iso = \
ed.iso AND cd.date = ed.date AND ed.iso = %s AND
ed.status_education <> 'Fully open
```

- Query11- שאילתא מורכבת.

סכימת כמות החולים בכל חודש בכל המדינות, מחודשים ינואר עד נובמבר.  
\*הערה- בחודש דצמבר אין נתונים ולכן לא מציגים אותו, בחודשים ינואר ופברואר יש נתונים אך זה היה בתחילת המגיפה ולכן ביחס לחודשים אחרים הנתון שלהם נמוך, ולא מספיק משמעותי כדי שיראו את זה בגרף.

```
SELECT MONTH(date),YEAR(date), SUM(total_cases) FROM
team09.covid_data group by MONTH(date)+'-'+YEAR(date)
```

- Get\_countries\_list

מחזירה רשימה של כל המדינות הקיימות (בשביל הבחירה בעמוד הפתיחה).

```
SELECT location FROM covid_country
```

- Query12- שאילתא מורכבת.

מציגה את כמות החולים בכל העולם בתאריך הנבחר.

```
SELECT date,SUM(total_cases) FROM team09.covid_data group by date
= %s;
```

### חישובים מעניינים שביצענו בקוד:

חישוב סיכוי להיכנס לסגר- בהתחלה בודקים האם המדינה נמצאת בכלל בטבלת מוסדות החינוך (אם לא, אין לנו איך לחשב בכלל אחוזים-שאילתא prev\_query10). אם היא כן, נבדוק את הסטטוס שלה לפי התאריך בטבלה (שאילתא 10). אם אין סטטוס בתאריך הנבחר או שהסטטוס הוא לא וודאי- חופשה או פתוח חלקית- נחשב יחסית לפי הממוצע בו המדינה לא הייתה פתוחה באופן מלא וסה"כ חולים (שאילתאות query1, query10\_get\_avg)  
(כמובן שאם אין כזה ממוצע או סה"כ חולים נחזיר שאי אפשר לחשב)  
אם אנחנו מתחת לממוצע-נחזיר חלק יחסי (בדיקת מעל/מתחת לממוצע זו שאילתא sub\_query10) ואם מעל- זה מראה על הסתברות גבוהה להיכנס לסגר וכך נציג. אחרת, המידע וודאי- פתוחים לגמרי או סגורים, נחזיר בהתאם (0 או 100) לכל שגיאה כזו או אחרת נאמר שאין באפשרותנו לחשב.

## UI

### UI corona app

- `Create_window_app`  
פונקציה זו יוצרת לנו את החלון משתמש, כלומר היא מאתחלת את החלון של האפליקציה.
- `Create_first_window`  
פונקציה זו יוצרת את החלון הראשון של האפליקציה. היא מכילה את כל הפרטים היוזואלים של החלון.



- `Moved`  
פונקציה זו אחראית על הגרפיקה של הכפתור מעבר לחלון הבא.
- `Add_combobox_country`  
פונקציה זו אחראית על יצירה החלופית שנפתחת בעת בחירת מדינה.
- `Clear_combo`  
פונקציה זו אחראית על אתחול החלופית.
- `On_keypress`  
פונקציה זו אחראית על האירוע של לחיצה באפליקציה.
- `Listbox_update`  
מעדכנת את החלופית של בחירת המדינה.
- `Add_calender`  
פונקציה זו אחראית על פתיחת לוח השנה כאשר בוחרים תאריך.
- `Image_background`  
אחראית על תמונת הרקע של החלון הראשון.



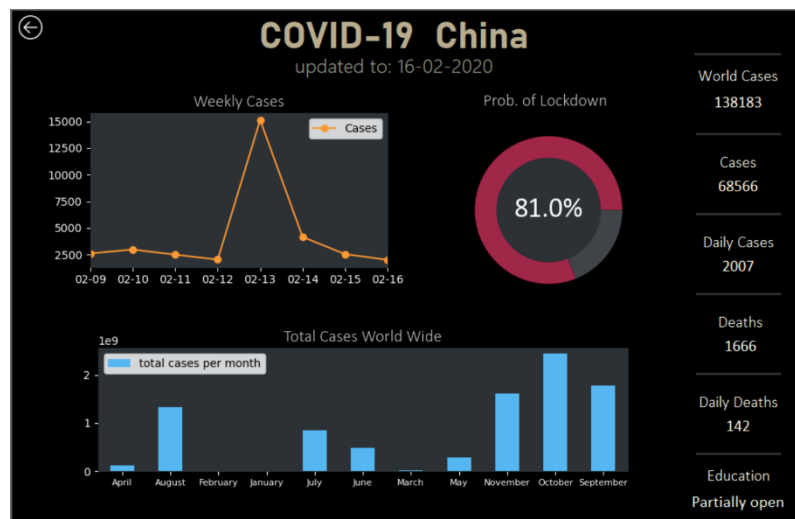
- Add\_button\_choose  
אחראית על יצירת לחצן הבחירה בעמוד הראשון.

- Clicked\_button\_choose  
המאורע של בחירה בחלון הראשון (לחיצה על כפתור הבחירה).

- Second\_window  
יצירת החלון השני של האפליקציה המכיל את הנתונים הרלוונטיים.

- Show\_all\_in\_second\_window  
אחראית על הצגת כל הנתונים הרלוונטיים בחלון השני.

- Show\_back\_butoon  
הוספת לחצן בחירה של חזרה לעמוד הראשון.

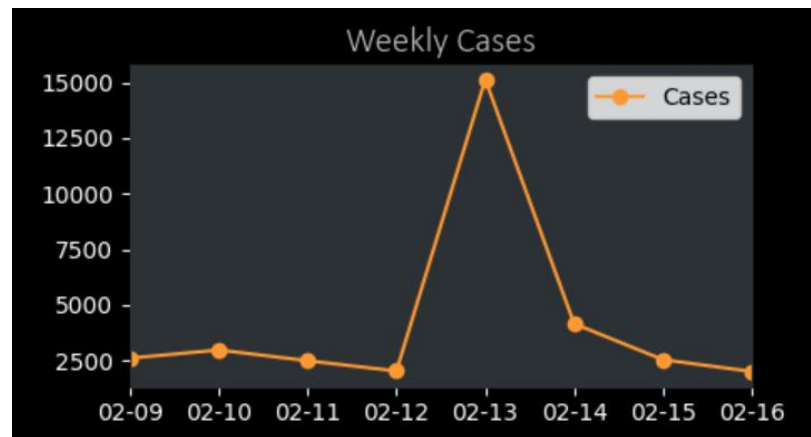


- Show\_first\_screen\_again  
חזרה לעמוד הראשון.

- Show\_rel\_data  
הצגה של המידע הרלוונטי על המדינה שנבחרה בחלון הראשון:

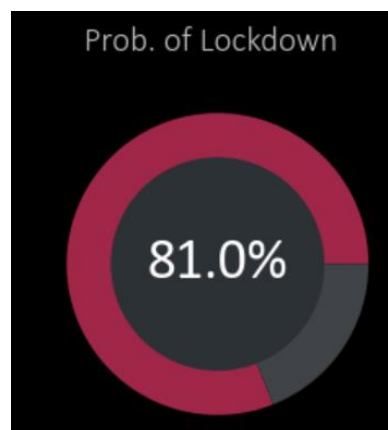
Draw\_new\_cases\_g •

הצגה של הגרף עם מקרי המאומתים החדשים במדינה שנבחרה:



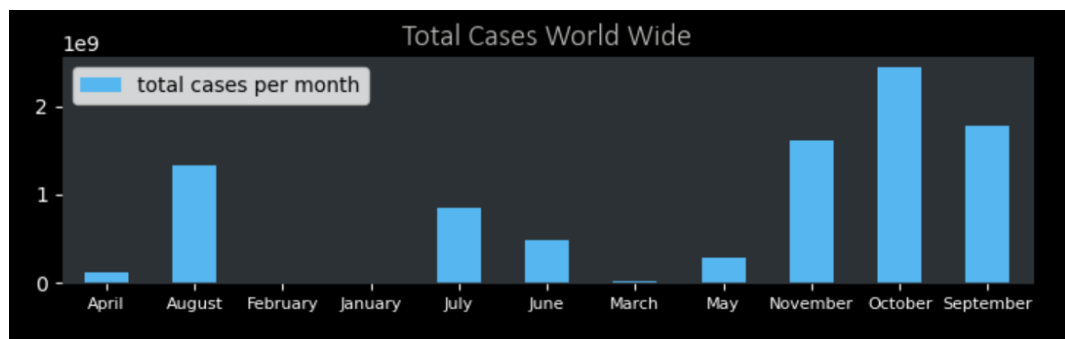
Draw\_chances\_g •

הצגה של הסיכויים להיכנס לסגר במדינה שנבחרה.



Show\_column\_graph\_for\_query11 •

הצגה של מקרי המאומתים הכללי בכל העולם, בהתאם לשאילתא מורכבת 11.



### טיפול בשגיאות

- עטפנו את הקוד כדי שיוכל להתמודד עם שגיאות שונות ובפרט להדפיס הודעה למסך כשאפשר. בין היתר:
  - import מסוים לא עבד, ספרייה חסרה (נדפיס למסך).
  - אם לא ניתן להתחבר לmysql נקבל שגיאה שתודפס למסך.
  - אם אנחנו לא מקבלים בשאילתא את כל המידע, נחזיר שגיאה (במידה והמשתמש בחר תאריך שלא נמצא בנתונים, נציג על מסך הודעה מתאימה).
  - הממשק שלנו יודע להציג הודעות על שגיאות מהשאילתות ולהתמודד איתן (לדוגמא- אם שאילתא לא הצליחה, או שהחזירה מידע חלקי, אם חלק מהשאילתא לא צלח ועוד).
- דאגנו להתמודד בצורה ויזואלית וגם שלא, כלומר להדפיס למסך את השגיאה ולאפשר לשאר המרכיבים שלא חלק מהשגיאה להמשיך לרוץ.