דו"ח הפרויקט בקורס "מעבדה בייצור משולב מחשב(Cim) "

הפקולטה להנדסה, אוניברסיטת ת"א

מגישים:

יותם דרעי	311551170
יובל אפל	311545669
רפאל שוכהנדלר	311402200

תאריך הגשה: 14/01/2021

מוגש לידי סגל הקורס:

פרופ' עירד בן גל, גד הלוי, יואב זיו, תומר גונן, מהראן ג'אזי

תיאור מבנה הפרויקט

הפרויקט אותו בחרנו ליישם הינו אפליקציה בעלת ממשק אינטראקטיבי בו יכול המשתמש לקבל מידע מעודכן אודות מצב התחלואה בישראל ו/או ביישובים וערים ספציפיות במדינה, בכל הנוגע לוירוס הקורונה.
הפרויקט נכתב ומומש בתקופת הגל השלישי, ונבחר בעקבות רצונם של חברי הקבוצה לבצע פרויקט בעל משמעות חברתית וערכית. אי לכך, בחרנו (צוות הפרויקט) לשתף את הקוד כמקובל בעידן הOpen-source העולה והצומח. ניתן למצוא את החומרים בקישור הזה.
בפרויקט זה יושמו שתי מעבדות אותן למדנו. מעבדת "אנדרואיד" יושמה במידה רבה, ומעבדת "web-scarping" יושמה אף היא. יישום מעבדת אנדרואיד בא לידי ביטוי בהקמת ממשק המשתמש והצגת הנתונים בו, בעוד שיישום מעבדת הweb-scarping בא לידי ביטוי בעבודה שנעשית אל מול הAPI של משרד הבריאות, לשם השגת הdata הגולמי של מצב התחלואה העדכני בישראל.
בהתאם לכך, הפרויקט מכיל את הקבצים הבאים:

- קובץ פייתון – health_data.py
  מטרת הסקריפט הינה לדלות את המידע באמצעות API שמספק משרד הבריאות ולבצע עליו עיבוד מקדים אשר כולל ניקוי של הדאטה, אגרגציה של הדאטה לכדי ערים ויישובים, יצירת עמודות חישוביות חדשות מהדאטה הקיים והכנה של הדאטה לכדי הכנסתו לבסיס הנתונים.
  הסקריפט משתמש בחבילות הבאות: Numpy, Pandas, urllib.request, datetime
- קובץ פייתון – corona_main.py
  זה הוא הסקריפט העיקרי אשר מממש את הממשק האינטראקטיבי, בו אנו מכניסים הלכה למעשה את הדאטה של התחלואה לבסיס הנתונים MySQL DB, יוצרים מחלקות ופונקציות אשר מתקשרות עם קובץ הkivy ומתשאלים את בסיס הנתונים לקבלת התוצאות הרצויות.
  הסקריפט משתמש בחבילות הבאות: Pandas, kivy.app, kivy.uix.boxlayout, kivy.properties, mysql.connector, sqlalchemy, pymysql, matplotlob.
  הפעלתו של הקובץ מפעילה הן את קובץ הפייתון המצוין לעיל, והן את קובץ הkv. עליו נפרט להלן.

- קובץ הקיוי - corona_kivy.kv

קובץ זה אחראי על העיצוב של הממשק כפי שהוא נראה למשתמש. בקובץ זה אנו מממשים את החיבור לבסיס התונים, את הפונקציונליות של האפליקציה ואת קבלת הקלטים אשר מזין המשתמש לפי רצונו.

קובץ הקיוי מהווה מאין "מתווך" בין המשתמש לקובץ הפייתון הראשי, לאור העובדה לפיה לחיצות מסוימות של המשתמש מפעילות מאחורי הקלעים פונקציות מסוימות מקובץ הפייתון הראשי. כך למעשה מתבצע ה-"flow" באפליקציה עד לקבלת המידע המבוקש על ידי המשתמש.

- קבצי קלט נוספים:

א. תיאור קובץ התחלואה:

קובץ התחלואה בו השתמשנו הינו publicly available באתר של משרד הבריאות, תחת מאגרי המידע הממשלתיים. את הקובץ אותו משרד הבריאות מספק ואיתו אנו עובדים ניתן למצוא בקישור הזה.

הקישור מספק תיאור של הפיצ'רים הקיימים בקובץ הגולמי (טרם הוספת העמודות החישוביות על ידינו). כאמור, אנו ניגשים לקובץ באמצעות API ולכן הוא איננו שמור אצלנו מקומית במחשב.

חשוב מאוד לציין:

הקובץ של משרד הבריאות מתעדכן מדי כמה ימים, ולפעמים משרד הבריאות מכיל בו שינויים מבניים. לכן, במידה ועולה שגיאה מסקריפט הפייתון health_data.py, יש לטפל בה בהתאם לשינוי המבני שהתרחש ע"י משרד הבריאות ושאיננו בשליטתנו.

ב. sociodemographic_stat.csv – קובץ הציונים הסוציאל-דמוגרפים:

קובץ זה מפרט את הציון הסוציאל-דמוגרפי לכל תת אזור של כל עיר. לכן, ביצענו אגרגציה לרמת העיר/יישוב ע"י ביצוע ממוצע בין כל הציונים של האזורים המוכלים באותה העיר/יישוב.

הציון הסוציאל-דמוגרפי נמצא כפיצ'ר בבסיס הנתונים. את הקובץ עליו עבדנו ניתן למצוא באתר של הלשכה המרכזית לסטטיסטיקה.

ג. towns_hebrew_english.csv – קובץ המפרט את שמות הערים בעברית ואנגלית:

נעזרנו בקובץ זה כדי להמיר את שמות הערים מעברית לאנגלית, מאחר וממשק

הקיוי לא יודע להתמודד עם קלטים בעברית (גם עבור גופן מותאם ומסוג שונה).

ד. xlsx.סהכ-אוכלוסייה-לפי-אס-גיל-ומין-סוף-2017 – קובץ המפרט את גודל האוכלוסיה לכל יישוב, נכון לסוף שנת 2017. אנו, חברי הצוות, רואים חשיבות רבה בשילוב גודל האוכלוסיה כאשר משווים בין נתוני תחלואה של ערים/יישובים שונים. גם את הקובץ המתואר ניתן למצוא באתר הלשכה המרכזית לסטטיסטיקה, (קישור מצורף בסעיף ב').

תיאור שלבי הרצת הקוד ומהלך התכנית:

הרצת התכנית:

להרצת האפליקציה יש לקבץ יחדיו את קובץ הקיוי וקבצי הפייתון. בנוסף, יש להתקין את סביבת העבודה Pycharm או לחילופין כל תכנת הרצה אחרת, ולהתקין את MySQL וסביבת הפיתוח שלו – MySQL Workbench. על מריץ האפליקציה לשנות את הcredentials שמופיעים בקובץ corona_main.py בכל מקום בו מתקיים חיבור לבסיס הנתונים לערכים התואמים את החשבון שלו.

לאחר ביצוע כל האמור לעיל, יש להפעיל את הסקריפט corona_main.py ולהמשיך דרך ממשק הקיוי.

מהלך התכנית:

תחילה, הסקריפט הראשי מייבא את החבילות הדרושות, מזהה את קובץ הקיוי ובין היתר מפעיל את הסקריפט health_data.py. במהלכו, אנו פונים דרך הAPI של משרד הבריאות בבקשה לקבלת הדאטה הגולמי, מורידים ערכים קטגוריאלים מעמודות התחלואה השונות ומבצעים המרות לסוגי המשתנים הרצויים.
נציין כי עבור ערכים חסרים בעמודה של הציון הסוציו אקונומי בחרנו להגריל ערך רנדומלי בין 1 ל10, בהתאם לסולם הערכים של הלשכה המרכזית לסטטיסטיקה.
בנוסף, השמתנו ערים אשר יש להם פחות משבוע דאטה מתועד (לאחר הניקוי), זאת מכיוון שאי אפשר יהיה לחשב עבורן את הציון והצבע אשר נקבעים לפי מודל רמזור (מפורט בהמשך).
לאחר סיום הניקוי של הדאטה, אנו יוצרים שלוש עמודות חדשות אשר מציינות את מדדי התחלואה היומיים באופן אבסולוטי (ולא כערך נצבר, כפי שעושה משרד הבריאות). באמצעות אותן עמודות אנחנו מחשבים עמודות חדשות (כמו למשל

כמות חולים פעילים, שהינה כמות האבחונים הפעילים עד כה פחות כמות המחלימים עד כה, לכל תאריך).
בהמשך, אנו משלבים בדאטה הבריאותי גם את גודל האוכלוסייה לכל עיר/יישוב ואת הציון הסוציואקונומי המתאים. נוסף על כך, החלטנו לחשב עבור כל עיר את הצבע המתאים שלה לפי מודל "רמזור" שהוצע ע"י הפרויקטור לשעבר פרופ' רוני גמזו. לשם כך, חיפשנו באינטרנט אודות הנוסחה לחישוב צבעה של כל עיר, ועל סמך התוצאות שמצאנו (מצ"ב קישור), יצרנו עמודה חישובית עבור כל מרכיב בנוסחה ולבסוף יצרנו את העמודה המציינת את צבעה של כל עיר לפי הציון המחושב לה ולפי ציוני הסף של המודל. טבלה זו מועברת לסקריפט הראשי (corona_main.py), אותה אנו מכניסים לבסיס הנתונים שלנו.
דוגמה לטבלה ולעמודותיה מופיעה בנספח מס' 1 תחת נספחים.

לאחר שמסתיימת הפרוצדורה המתוארת לעיל, אנו חוזרים לסקריפט הראשי כאשר כעת אנו ניצבים אל מול ממשק הקיוי שנפתח לנו במסך הראשי.
כעת נרצה לבצע חיבור ראשוני לבסיס הנתונים ולהכניס לתוכו את הטבלה שקיבלנו כקלט מהפרוצדורה במלואה באמצעות הכפתור 'Update DB' (ראה נספחים 2 ו3 בהתאמה).

בנוסף, המשתמש יכול לבחור במסך הראשי לקבל "טעימה" מהדאטה (אשר כוללת שלוש עמודות עיקריות – תאריך, סמל הישוב ושם הישוב), לקבל דאטה ברמת העיר/ישוב, לקבל דאטה ברמת המדינה, או לצאת בבטחה מהאפליקציה.

בהנחה שהמשתמש בחר לקבל דאטה ברמת העיר/ישוב, הוא יועבר למסך אחר בו הוא יוכל לבחור את מקום המגורים שלו (או כל אופציה יחידה אחרת) ולקבל עליה מדדים שונים, יחד עם גרף המייצג את כמות החולים הפעילים לפי זמן (ראה נספח מס' 4).

בהנחה שהמשתמש בחר לקבל דאטה ברמת המדינה, הוא יועבר למסך נפרד בו הוא יוכל בלחיצת כפתור לקבל מידע כמותי וגרפי אודות מצב התחלואה במדינה (ראה נספח מס' 5). זאת ועוד, יוכל המשתמש בכל שלב ולכל בחירה לצאת בבטחה באמצעות כפתור ה'Finish' שנמצא בכל רגע נתון, או לחילופין לחזור לעמוד הבית באמצעות הכפתור 'Go home' כדי להתחיל חיפוש מחדש.

יש לציין כי כדי להציג גרף מסוים כפי שבחר המשתמש, התכנית שומרת את הגרף הנוצר כקובץ מסוג jpeg. וקוראת את אותו הקובץ לאחר מעבר למסך נפרד כדי להציג אותו למשתמש.

כל הפונקציונליות התאפשרה ולא יכלה להתקיים ללא קובץ הקיוי corona_kivy.kv אשר באמצעותו אנו יכולים לקבל את הקלט מהמשתמש ולבצע את כל העיצוב הגרפי מתחילתו ועד סופו (כולל הצגת התוצאות הנדרשות למשתמש). באמצעות הידע אותו רכשנו במעבדת "אנדרואיד" הרחבנו רבות את האפליקציה שלנו והוספנו מספר רב של מחלקות ופונקציות, כך שהמידע אותו יוכל לקבל המשתמש יהיה כמה שיותר מגוון ועשיר. הקוד נכתב לפי העקרונות של שפת הקיוי, תוך שמירה על כמות קוד מינימלית בקובץ הפייתון באמצעות הורשה (כולל הורשה ספציפית של מתודות מסוימות).

<u>קשיים ומסקנות:</u>

במהלך העבודה על הפרויקט, נתקלנו בקשיים רבים, הן במישור היצירתי והן במישור היישומי של האפליקציה. במשך זמן לא מבוטל חשבנו איך נוכל לחדש ולהביא ערך מוסף בפרויקט אותו נבחר לעשות, וגם כאשר בחרנו להנגיש מידע אודות נגיף הקורונה, השקענו מאמץ ומחשבה אודות כיצד נוכל להביא את הנתונים המעניינים ביותר במסגרת הזמן שהוקצב לנו לסיום הפרויקט.

בנוסף, התמודדנו עם האתגר שטמון בלמידת שפת הקיוי, אשר בה אין לנו ניסיון מעבר לביצוע מעבדת האנדרואיד במסגרת הקורס.

לעניות דעתנו הרחבנו רבות את הידע שלנו. צברנו ניסיון בכתיבת שורות רבות ורציפות של קוד, העשרנו את הידע שלנו בשפת קיוי ובהקמת בסיס נתונים עם עשרות אלפי שורות בדרך נוחה ואלגנטית, ולמדנו כיצד לעבוד עם APIs.

נרצה לציין כי אנו שבעי רצון מהתוצאות ובעיקר מהויזואליזציה אשר תוצאותיה מתיישבות עם ההיגיון ועם תזמון הסגרים. יחד עם זאת, נרצה לסייג ולומר כי ברוב המקרים התוצאות מלוות ברעש גדול מאוד, ולכן אם היה לנו זמן נוסף למקצה שיפורים, היינו מבצעים אגרגציה לאזורים גדולים יותר, כדי לנטרל במידה רבה יותר את הרעש אשר קיים בנתונים כעת.

אנו מקווים כי עבודתנו תוכל להיות לעזרם של כל החפצים בכך, בין אם כדי לקבל מידע אודות התפשטות נגיף הקורונה, ובין אם כדי לקחת את החומרים והקוד שיצרנו ולשכללו למשהו נרחב אף יותר.

נרצה בזאת להודות לצוות הקורס על הליווי, התמיכה ומתן העצות לאורך העבודה הרציפה בקורס בכלל ועל הפרויקט בפרט.

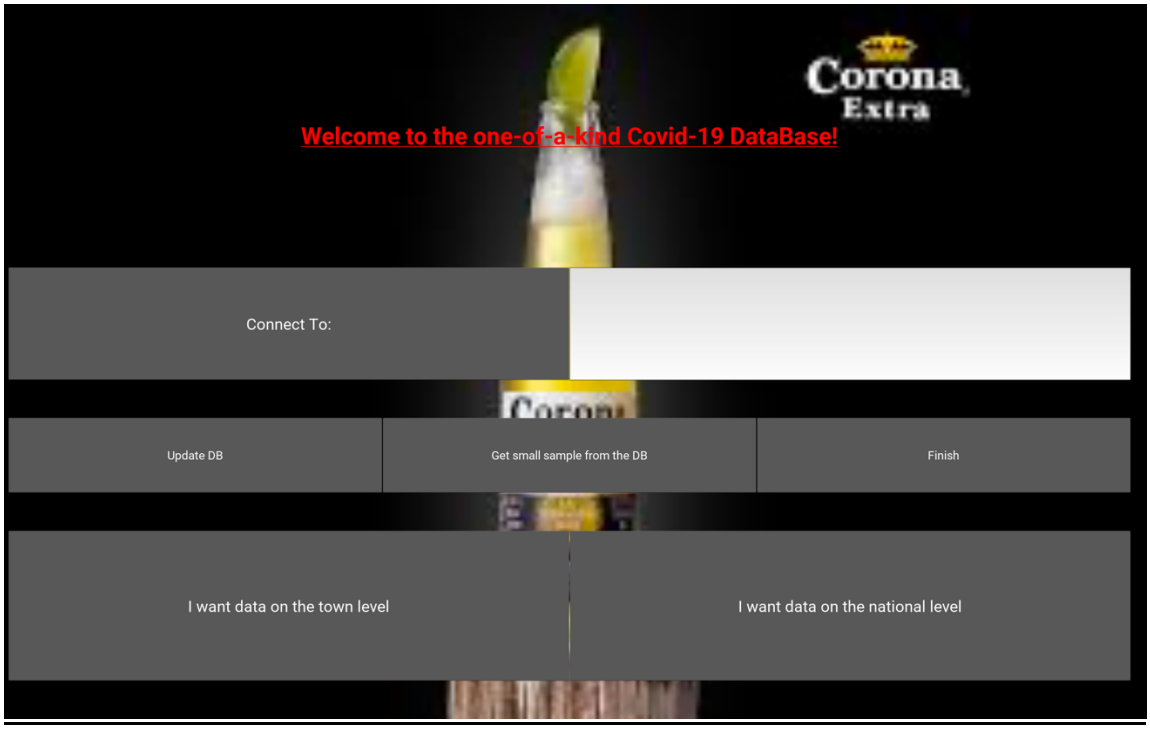
נספחים

נספח 1 – תקציר הטבלה הסופית מקובץ health_data.py :

A1 | fx | date

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	date	town_code	accumulated_cases	accumulated_recoveries	accumulated_tested	active_cases	town	count_tests	count_positi	count_active_cases	positivity_rate	population	Poverty_De	sick_for_10,(	growth_rate	positivity_rat	traffic_light_	town_color
2	30/09/2020	26	35	16	697	19	ROSH PINNA	9	1	0	11.11	3061	5.43	1	1	29.3	5.38	yellow
3	01/10/2020	26	35	17	728	18	ROSH PINNA	31	0	-1	0	3061	5.43	1	1	29.3	5.38	yellow
4	02/10/2020	26	35	19	741	16	ROSH PINNA	13	0	-2	0	3061	5.43	1	1	29.3	5.38	yellow
5	03/10/2020	26	35	20	745	15	ROSH PINNA	4	0	-1	0	3061	5.43	1	1	29.3	5.38	yellow
6	04/10/2020	26	35	21	747	14	ROSH PINNA	2	0	-1	0	3061	5.43	1	1	29.3	5.38	yellow

נספח 2 – המסך הראשי של האפליקציה :



נספח 3 – בסיס הנתונים כפי שהוא קיים בסביבת העבודה (workbench):

Limit to 50000 rows

```
use corona_db;
SELECT * FROM corona_data;


SELECT town,accumulated_tested,accumulated_cases,accumulated_recoveries,active_cases,positivity_rate FROM corona_data
```

100% 27:2

Result Grid Filter Rows: Search Export:

index	date	town_code	accumulated_cases	accumulated_recoveri...	accumulated_test...	active_cases	town	count_tests	count_positive
42724	2020-12-10 00:00:00	9800	179	173	6908	6	BINYAMINA	107	1
42725	2020-12-11 00:00:00	9800	179	173	7045	6	BINYAMINA	137	0
42726	2020-12-12 00:00:00	9800	181	173	7088	8	BINYAMINA	43	2
42727	2020-12-13 00:00:00	9800	183	173	7195	10	BINYAMINA	107	2
42728	2020-12-14 00:00:00	9800	188	173	7279	15	BINYAMINA	84	5
42729	2020-12-15 00:00:00	9800	190	174	7539	16	BINYAMINA	260	2
42730	2020-12-16 00:00:00	9800	190	175	7752	15	BINYAMINA	213	0
42731	2020-12-17 00:00:00	9800	194	175	7877	19	BINYAMINA	125	4
42732	2020-12-18 00:00:00	9800	195	175	8015	20	BINYAMINA	138	1
42733	2020-12-19 00:00:00	9800	197	177	8181	20	BINYAMINA	166	2
42734	2020-12-20 00:00:00	9800	199	179	8281	20	BINYAMINA	100	2
42735	2020-12-21 00:00:00	9800	203	179	8418	24	BINYAMINA	137	4
42736	2020-12-22 00:00:00	9800	211	181	8619	30	BINYAMINA	201	8
42737	2020-12-23 00:00:00	9800	216	188	8740	28	BINYAMINA	121	5
42738	2020-12-24 00:00:00	9800	217	188	8903	29	BINYAMINA	163	1
42739	2020-12-25 00:00:00	9800	217	190	8996	27	BINYAMINA	93	0
42740	2020-12-26 00:00:00	9800	218	190	9122	28	BINYAMINA	126	1
42741	2020-12-27 00:00:00	9800	220	197	9202	23	BINYAMINA	80	2
42742	2020-12-28 00:00:00	9800	225	198	9356	27	BINYAMINA	154	5
42743	2020-12-29 00:00:00	9800	226	198	9449	28	BINYAMINA	93	1
42744	2020-12-30 00:00:00	9800	230	204	9580	26	BINYAMINA	131	4
42745	2020-12-31 00:00:00	9800	233	212	9701	21	BINYAMINA	121	3
42746	2021-01-01 00:00:00	9800	236	214	9835	22	BINYAMINA	134	3
42747	2021-01-02 00:00:00	9800	239	215	9947	24	BINYAMINA	112	3
42748	2021-01-03 00:00:00	9800	242	218	10023	24	BINYAMINA	76	3

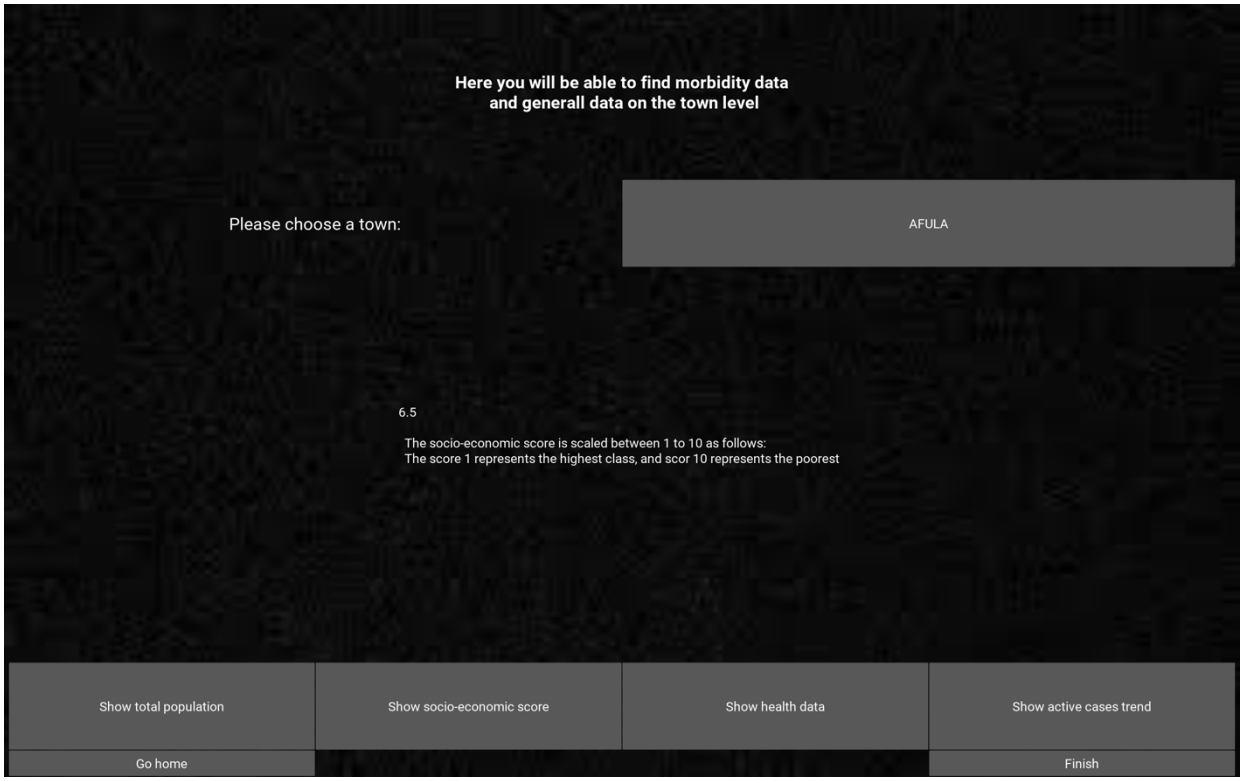
Result Grid, Form Editor, Field Types, Query Stats, Execution Plan

corona_data 1 Read Only

Action Output

	Time	Action	Response	Duration / Fetch Time
1	23:21:49	SELECT * FROM corona_data LIMIT 0, 50000	42749 row(s) returned	0.0010 sec / 0.095 sec

נספח 4 – בחירה בקבלת נתוני תחלואה ברמת העיר/ישוב:



נספח 5 – בחירה בקבלת נתוני תחלואה ברמת המדינה: