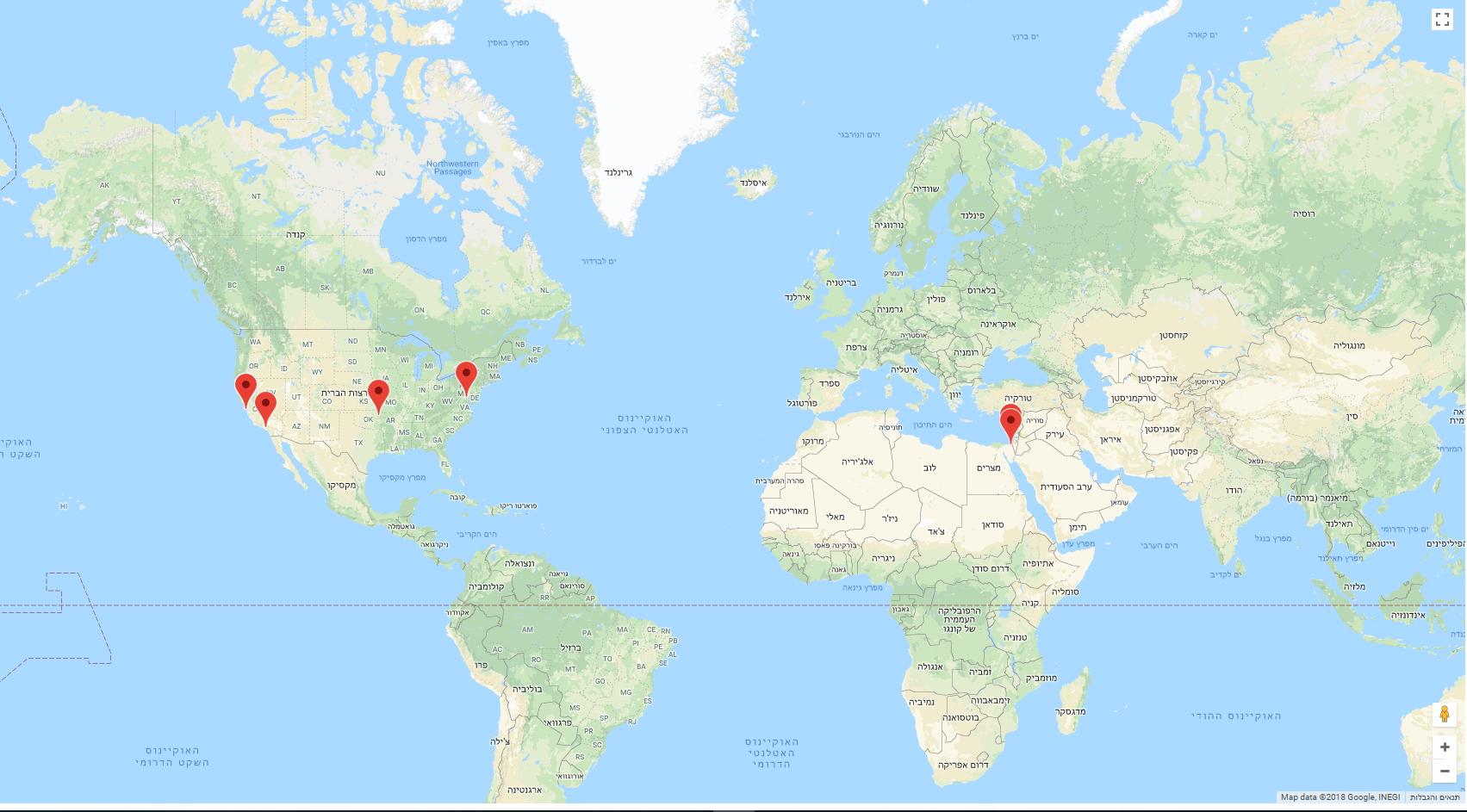


מסמך אפיון

תומר גרוס

<גרסה סופית>

<27/5>





**1. הקדמה**

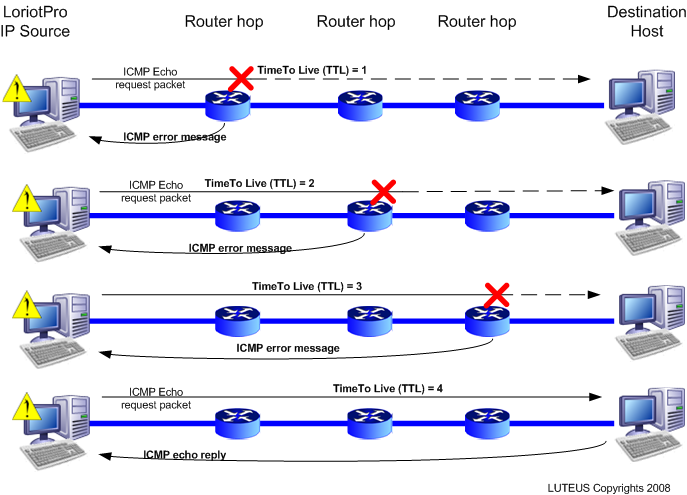
בחרתי להכין בפרויקט את כלי הtraceroute ולהציגו באופן ויזואלי על גבי מפת העולם, בשילוב סטטיסטיקות על נקודות וקווי תקשורת, מדידת ביצועים ועוד.

* 1. מטרה

ע"י כלי זה, מנהל רשת יכול לפענח היכן נמצאים "צווארי הבקבוק" של הרשת, איזה מהנתבים הנמצאים במסלול החבילה טובים יותר ואילו פחות, כלומר מאפשר מדידת ביצועים. בגדול, מאפשר תחושה מפורטת וניתוח של בעיות ברשת האינטרנט. נוסף על כך, הכלי הויזואלי מאפשר לראות את המיקום הגיאוגרפי שלהם.

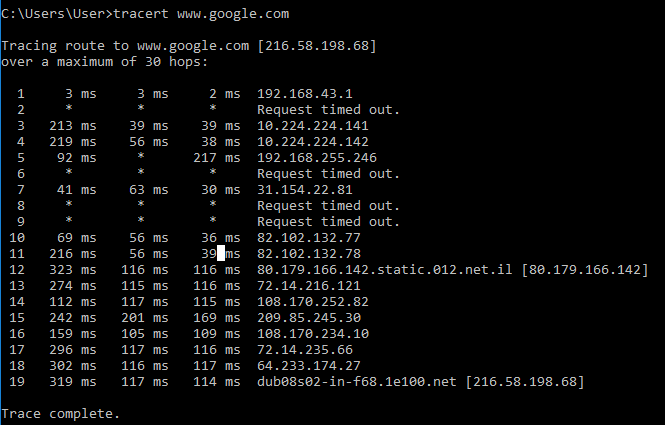
* 1. המוצר

הערה: מומלץ לקרוא את המושגים המתוארים בסעיף 1.3 לפני הסברת המוצר.

במחשבים, המונחtraceroute הוא כלי ניתוח לרשת מחשבים המציין את המסלול שעוברת חבילת מידע מהרשת שלך אל רשת ספציפית המצוינת בפקודה tracert(כתובת של אתר או כתובת IP כלשהי). איך זה עובד? הנתיב נמצא על ידי שליחה של שלוש הודעותICMP  אל היעד עם ערכי שדה Time to Live ‏(TTL) נמוכים. כל נתב בדרך אל היעד מפחית את הערך בשדה זה ב-1 בזמן שהוא מעביר את ההודעה הלאה. אם הנתב רואה כי ערך השדה הוא אפס, הוא שולח הודעתICMP  מסוג שגיאה בחזרה אל השולח. השולח מעלה כל פעם את הערך ההתחלתי של ה TTL (מתחיל מ-1) כך שבסופו של דבר הפעולה נפסקת כאשר הוא מקבל תשובת ICMP מסוג הצלחה מכתובת היעד. סכום ממוצע הזמנים בכל הופ, זהו הזמן שלקח עד שהצלחנו לתקשר עם כתובת היעד. Geo-traceroute לא מראה את כתובות ה- IP של הראוטרים דרכה עוברת החבילה כמו כלי ה -traceroute , אלא הוא משתמש בכלי זה על מנת להפוך את המסלול ממסלול בצורת כתובותIP למסלול ויזואלי שמאפשר לראות את מיקומם הגיאוגרפי של כתובות אלה על גבי מפת העולם, כך שבסופו של דבר מסלול זה הוא המסלול שעברה החבילה עד הגעתה ליעדה.

1.3 הגדרות

* IP – כתובת פרוטוקול אינטרנט (internet protocol) היא תווית מספורית המוקצית לכל התקן המחובר לרשת מחשבים המשתמשת בפרוטוקול האינטרנט לצורך תקשורת. כתובת זו משרתת שתי פונקציות עיקריות: זיהוי מארח או ממשק רשת ומיקום כתובת.
* Packet (חבילה) - היא יחידה בסיסית של נתונים המועברים ברשת מחשבים. חבילת מידע מורכבת משני חלקים עיקריים: ראש החבילה, הheader שמכיל את המידע הנחוץ על מנת להעביר את החבילה מהמקור אל היעד, בהתאם לפרוטוקול המשמש להעברת החבילה, לדוגמא כתובת IP של היעד, וסוג המידע שנשלח. החלק השני הוא החלק בו מופיעים הנתונים, הdata.
* קובץ מסוג JSON - ראשי תיבות של JavaScript Object Notation , הוא פורמט טקסטואלי, הקריא לאדם, המיועד להעברת מבני מידע המורכבים מזוגות של מפתח-ערך. השימוש העיקרי של הפורמט הוא להעברת מידע בין שרת לצרכן. אני השתמשתי בו כדי לקבל מידע מאתר אינטרנט בצורה של json שאפשר לי סידור נוח של המידע שלי.
* Tracert (פקודה) – הפקודה tracert בwindows- וtraceroute- ב-Linux וMac מקבלת כפרמטר חובה כתובת יעד. הפקודה בעצם מראה לנו את המסלול שעוברת חבילת מידע מהמחשב שעליו הופעלה הפקודה עד להגעתה לכתובת היעד שהוזנה לה, היא מראה זמנים שלקח לה לעבור מנתב תקשורת אחד לאחר והאם היו נתבים שדרכה נכשלה לעבור מספר פעמים. לדוגמה נזין את הפקודה בצירוף כתוב היעד של גוגל:



* 1. תקציר – תוכן עניינים

עמוד 2 – הסבר כללי על המוצר, שימושיו ,מטרותיו ואופן פעולתו.

עמוד 3 – הגדרות כלליות שיאפשרו להבין את תהליך והמרכיבים בעבודה.

עמוד 5 – תיאור כללי של מה התוכנית מסוגלת לעשות, מי ירצה להשתמש בה ואילוצים עיקריים.

עמוד 6 – הסבר על אופן השימוש בפרויקט ואילוצים שישנו לי את זמן החישוב של המסלול.

1. תיאור כללי

2.1 פונקציונליות

הפונקציות העיקריות של המערכת:

* הצגת מסלול שחבילת מידע עוברת על גבי מפת העולם - ויזואלית.
* המערכת שומרת תחנות ביניים בצורת כתובות IP ומבצעת סטטיסטיקה על פרמטרים שונים כגון: שם הרשת, זמן שלקח בממוצע לחבילה להגיע אל כתובת זאת, מיקום גיאוגרפי שלה, מספר ביקורים בתחנה זו במסלולים השונים ועוד. בסופו של דבר יישמר לנו מידע על כתובות IP שמייצגות נתבים ונוכל לפענח בעיות תעבורה ולמדוד ביצועים של רשת האינטרנט הנ"ל.

2.2 קהל היעד

קהל היעד של המוצר הוא מנהלי רשתות אינטרנט, להם כלי זה משמש כלי מקצועי שבעזרתו הם יכולים לפענח תעבורה ולמצוא היכן נמצאים צווארי הבקבוק ברשת באופן מהיר ופשוט מאוד. ניתן לראות מתוך שלושת החבילות שנשלחות כמה מהן הצליחו לחזור כתגובה ובאיזה זמן. נוסף על כך, כל אחד שברצונו להשתמש בכלי זה על מנת לדעת היכן עוברת חבילה שהוא שולח למקום כלשהו יכול להשתמש בו ללא דרישה של ידע קודם ב-"רשתות".

2.3 אילוצים עיקריים

שני האילוצים שהתוכנה צריכה בשביל לרוץ על מחשב מסוים:

* על מערכת ההפעלה של המחשב להיות מסוג windows.
* המחשב נדרש להיות מחובר לאינטרנט

1. דרישות מפורטות

3.1 דרישות פונקציונליות

* פיצ'ר ראשון – על מנת לראות את המפה הוויזואלית שמתקבלת עבור כתובת ip נתונה, צריך להזין את כתובת הip כמוסבר במדריך המשתמש ולהריץ את קוד הפייתון. ייווצר קובץ פלט מסוג html שמציג את המפה, אותו ניתן לפתוח באופן רגיל או דרך האלקטרון. כברירת מחדל הוא נפתח באלקטרון.
* פיצ'ר שני –הסטטיסטיקות נשמרות ומוצגות באותו קובץ html. הערה: הסטטיסטיקות המוצגות הם לכתובת הip האחרונה שחיפשת דרכה. בפועל בDB נשמרות כל תחנות הביניים ע"י סיווג של איזה כתובת חיפשתי. לדוגמה הרצתי מספר פעמים על יוטיוב אז אני אוכל לראות דרך קובץ הjson את כל הכתובות שהחבילה שלי עברה עד הגעתה לשרתי יוטיוב בכל ההרצות שלי.

3.2 דרישות של ממשקים חיצוניים

**3.2.1 קיים** GUI **מסוג אלקטרון.**

**3.2.3 ממשקי תוכנה** – התוכנה "מתקשרת" בין קוד הpython לקוד הJavaScript באמצעות אובייקט מסוג json. קובץ הpython שומר את הנתונים באובייקט מסוג JSON, וקוד הJavaScript אחראי על הצגתו בדפדפן/בממשק המשתמש. בעת הרצת הסטטיסטיקות מפענח אותן ומציגן בצורה נוחה וקריאה.

3.3 דרישות לא פונקציונליות

**3.3.1 דרישות ביצועים (**performance**)** – אינטרנט איטי יביא לזמן חישוב גדול יותר של המסלול שהחבילה עוברת.

**3.3.6 דרישות ניידות (**portability**)** – עובד גם על מחשב נייד, בתנאי שעונה לאילוצים העיקריים.

3.4 דרישות בסיס נתונים

* **במבנה הנתונים יישמרו נתונים על כל כתובות ה**IP **שדרכם עברנו, לכל כתובת** IP **יישמר שם אזור גיאוגרפי, זמן להגעת חבילה, מספר ביקורים בה כתחנת ביניים וכו'.**
* **קובץ מסוג** JSON **שמאחסן את כל המידע, על מנת שאם נרצה להציג מידע עבור התחנה המסוימת לא רק בהרצה הנוכחית**.
* **נפח אחסון נדרש, מקסימום כ-4 מגה בייט לפי הנחה שלא יישמרו נתונים על יותר מ-1000 כתובות** IP**.**



מסמך עיצוב

<תומר גרוס>

<גרסה סופית>

<27/5>





**1. הקדמה**

1.1 מטרה

*המטרה של מסמך העיצוב היא להסביר בדיוק איך הפרויקט בנוי, איך לקוח יוכל להשתמש בו, ואיך בנוי ממשק המשתמש.*

1.2 המוצר

*שם המוצר* – Geo Traceroute Statistics

*הכלי מאפשר תצוגה ויזואלית של נקודות התקשורת על גבי מפת העולם. בנוסף, המוצר מאפשר פיענוח של תעבורת הרשת.*

1.3 קישור למסמכים קודמים

*מסמך האפיון*

1.4 הגדרות

* GUI – ראשי תיבות של graphic user interface או בעברית ממשק משתמש גרפי. הוא ממשק משתמש לתוכנה או לאתר אינטרנט, המבוסס על עיצוב גרפי של המסך המוצג למשתמש בלבד.
* קובץ מסוג JSON - ראשי תיבות של JavaScript Object Notation , הוא פורמט טקסטואלי, הקריא לאדם, המיועד להעברת מבני מידע המורכבים מזוגות של מפתח-ערך כלומר במקרה זה כתובת ip ומידע עליה. השימוש העיקרי של הפורמט הוא להעברת מידע בין שרת לצרכן. אני השתמשתי בפורמט זה כדי לקבל מידע מאתר אינטרנט בצורה של json שאפשר לי סידור נוח של המידע שלי, שמירתו וקריאתו לאחר מכן.

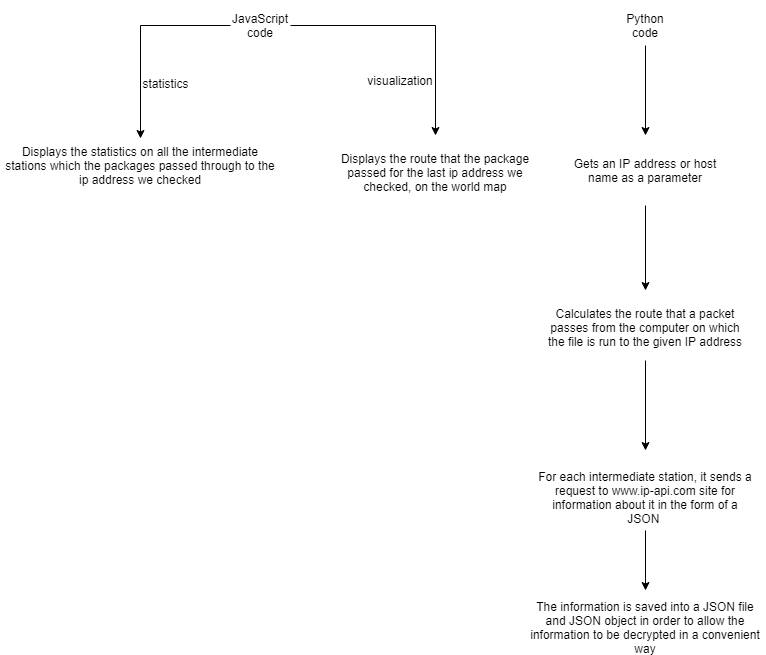
1. ארכיטקטורת המערכת

2.1 מבט על

ישנם שני פעולות עיקריות במערכת:

* מסלול החבילה – קוד הPython אחראי לחישוב מסלול החבילה, אסיפת מידע עליה, יצירת מאגר המידע מסוג JSON ועדכונו מפעם לפעם.
* הצגת מסלול החבילה האחרונה, הצגת סטטיסטיקות – קוד הJavaScript יודע לבנות את מפת העולם שעליה מוצגות כל כתובות הip שהחבילה האחרונה שבדקתי עברה דרכם. בנוסף ישנו עוד קוד JavaScript שמאפשר הצגה של הסטטיסטיקות כלומר מידע על כל המקומות שכל החבילות שלי עברו דרכם עד כה – פיענוח מאגר המידע. המידע מוצג בממשק משתמש גרפי מסוג אלקטרון או בדפדפן.

תרשים זרימה המתאר בצורה נוחה את תפקיד כל קוד:

****

2.2 פירוט רכיבי המערכת

Python functions

validate\_ip

input: gets an ip address.

output: return true if the address is valid, otherwise false.

route\_by\_dicts  
input: gets a list of ips.

output: for each ip in the list it sends request to "http://ip-api.com" for information about it in JSON format and returns list of dictionaries. In python - json converted into dictionary using json.loads.

route\_by\_ips

input: gets a goal address.

output: returns a list of router addresses we passed through, until we reached the goal address.

current\_json

input: gets a route

output: creates and returns a json object from the route information in python syntax

update\_json\_db

input: history of all ip addresses

output: update the history by adding the current calculation

remove\_duplicates

input: gets a list

output: removes all the duplicates

main

the main uses the above functions to save all the information we have collected to the json file and json object that will be used by the JavaScript code to generate the statistics or to display the last package path on the world map using electron GUI.

JavaScript functions:

showMap

this is the most important function in the file, this function managed the display of the map. All the google maps api is written in JavaScript.

2.3 דיון בנושא העיצוב הנבחר

**בחלוקות התפקידים בעצם תיכנתתי את כל מה שקשור לטיפול בקבצים שלי ו"דיבור" עם שרת ב**Python **ואת יצירת המפה והצגתה ב**JavaScript **מהסיבה שב**Python **הרבה יותר נוח לי לעבוד ובגלל שה**api **קיים רק ב**JavaScript.

**הבעיה העיקרית שבה נתקלתי והחלטתי בעצם לטפל בקבצים שלי ב**Python **היא שהיו לי שני אופציות כדי לקרוא את קובץ ה-** JSON**שפייתון יצר, ע"י המרתו לטקסט ואז לאובייקט מסוג** JSON. **אופצייה שנייה הייתה לפתוח שרת על המחשב או שרת חיצוני, להעלות את כל הקבצים הדרושים בפרויקט בפרט קובץ ה**JSON **לשם ואז לבקש אותו מהשרת, זוהי הדרך בה לא בחרתי מהסיבה שלא רציתי להתעסק עם פתיחת שרת, אלא רציתי להתרכז ביצירת המרכיבים העיקריים בפרוייקט שהם להציג את הסטטיסטיקות ולהראות את מפת העולם עבור חבילה שעוברת במסלול כלשהו.**

1. עיצוב נתונים ופרוטוקולים

* **מועבר אובייקט** JSON **שמבקש קובץ ה**Python **מאתר בשם** www.ip-api.com**:**

**מטרה: הקובץ מבקש מהאתר אובייקט מסוג** JSON **שמכיל שדות מידע על כתובת ה**ip **שנשלחה בבקשה.**

**השדות הם: מדינה, עיר, אזור זמן, קיצור של השם של המדינה, קו אורך וקו רוחב(מכאן משיגים את מיקום כתובת ה**ip **על פני מפת העולם), ספק אינטרנט ועוד. אני בחרתי להציג רק חלק מהשדות מהסיבה שאני חושב שלא כל השדות רלוונטים.**

* **מידע שנשמר לוקלית בתיקייה בה נמצא הקובץ:**
* **קובץ מסוג** JSON **שמכיל את כל הסטטיסטיקות כלומר כל ההיסטוריה של כתובות ה**ip**, וכל המידע עליהם.**

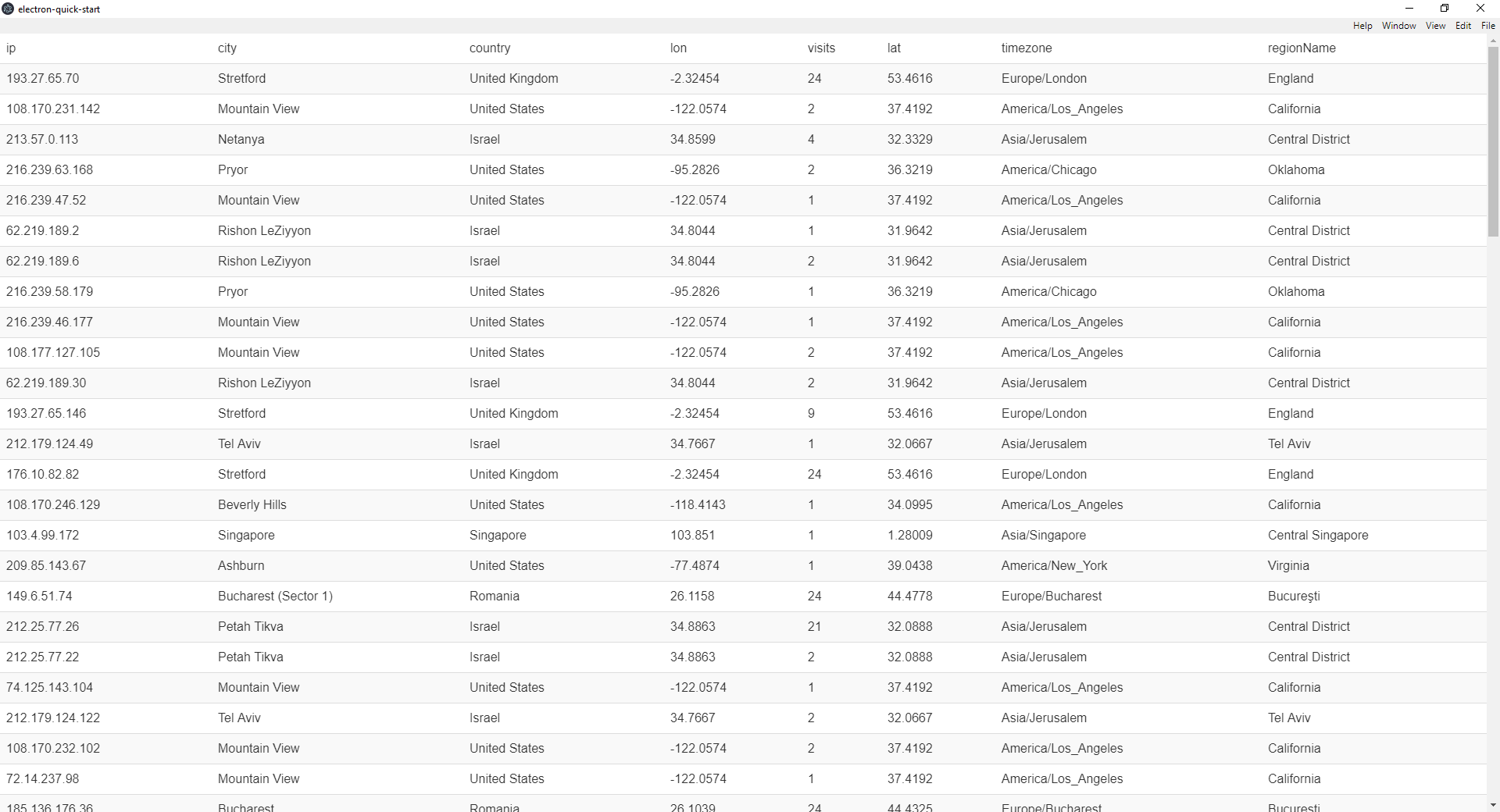
**מטרה: לקובץ הזה מותאם קוד** JavaScript **שיודע לקרוא אותו ולהשתמש בו כדי ליצור את טבלת הסטטיסטיקות, אך אנו בפרוייקט נציג רק את המידע עבור החבילה האחרונה.**

* **קובץ** html**.**

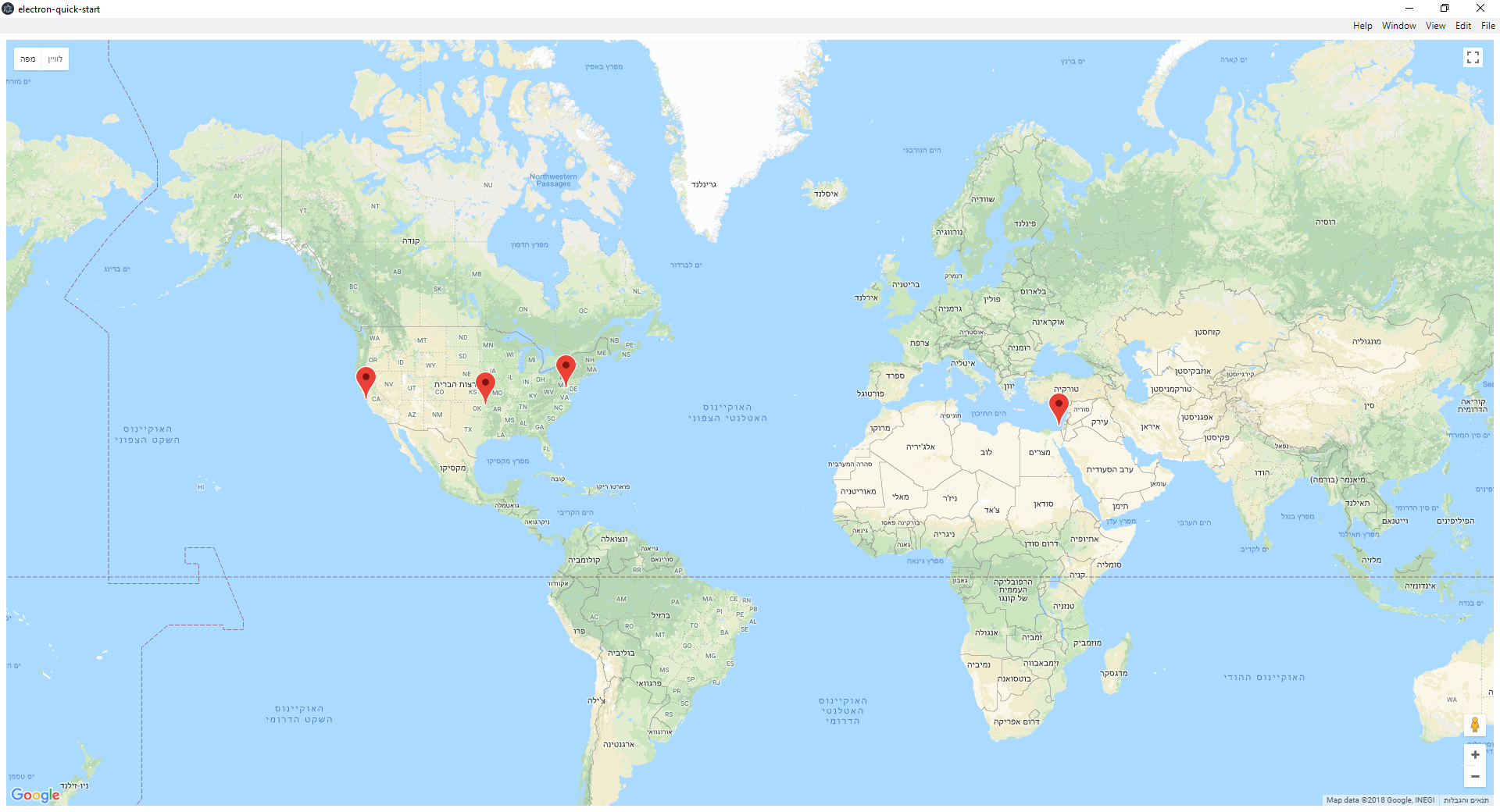
**מטרה: זהו המידע שאותו ה**gui **מציג.**

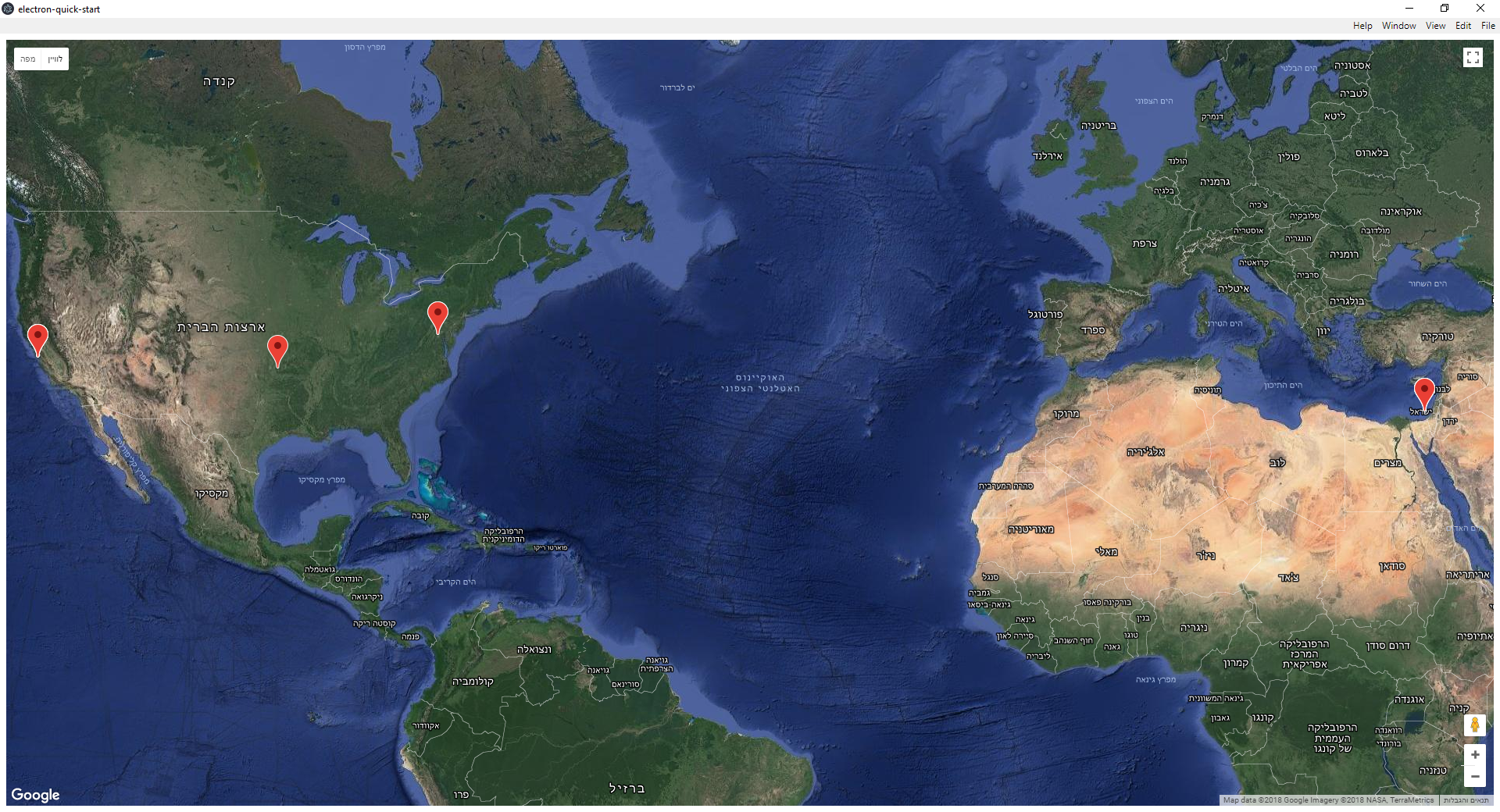
4. ממשק משתמש

תצוגת הסטטיסטיקות:



**תצוגת המפה (מצב רגיל ומצב לווין):**





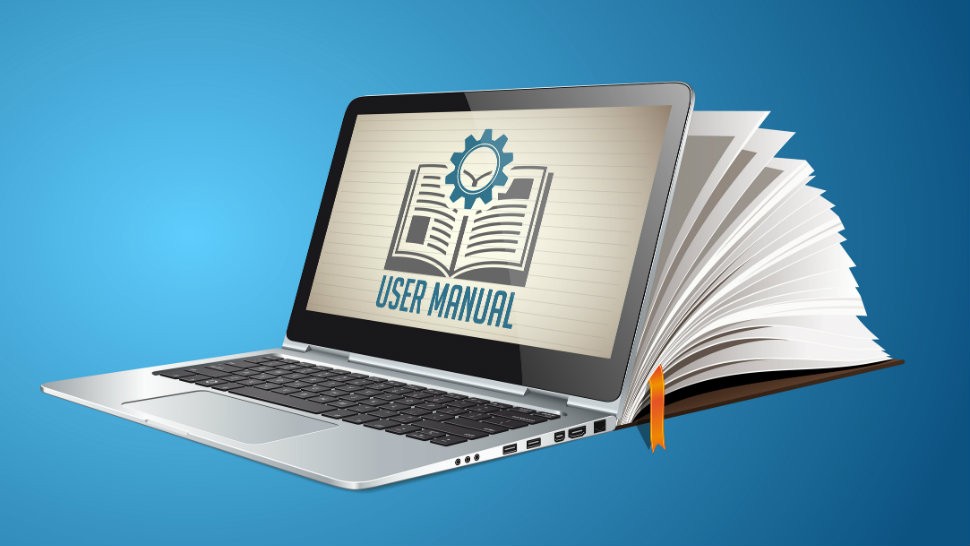


מדריך למשתמש

תומר גרוס

<גרסה סופית>

<27/5>





Assumptions

1. Python is in the windows path
2. The computer has a network connection

Installation guide

1. Run the node-v8.11.3-x64 file and follows instructions.
2. Open a new directory in your preferred folder.
3. Download project\_engine.py and project\_preparation.py files to your chosen folder.
4. Open command line.
5. Navigate to the project directory.
6. Run project\_preparation.py with python.
7. Copy the directory Google Maps Markers to the directory: electron-quick-start

User manual

1. Open command line.
2. Navigate to your folder.
3. Enter the following command: python project\_engine.py [param]  
   in the param write an ip address that you want to see a route to. Wait for the program to end.

מה למדתי ורפלקציה אישית

**הדברים העיקריים שלמדתי בפרויקט זה הם שפת תכנות חדשה –** JavaScript**, לקחתי קורס באתר בשם** codecademy **בנושא** JavaScript **כי כאשר חקרתי את הנושא בשלבים הראשונים של הפרויקט, הבנתי שעליי לדעת את השפה הזאת על מנת לעבוד עם** googlemapsapi **וגם כי התכנון היה לכתוב את מרבית הפרוייקט ב-**JavaScript**. אני חושב שבכל מקרה חשוב ללמוד את השפה הזאת כיוון שזוהי השפה העיקרית כיום בה משתמשים להקמת שרתים, בניית אתרים ועוד. בנוסף, למדתי איך לעבוד עם אובייקט מסוג** JSON **בפייתון, דבר שנותן יתרון מבחינת סדר וארגון נתונים, אני חושב שדבר זה ישמש אותי הרבה מאוד בעתיד, באופן שמירת המידע לאחר איסוף מידע גדול על נתונים כלשהם. אני חושב שהפרויקט הזה הצליח לאתגר אותי מבחינת למידה של דברים חדשים, אני מאוד שמח שהוספתי לידע שלי שפת תכנות חדשה, ופורמט כתיבה/מבנה נתונים שמשתמשים בו כיום בתדירות גבוהה בתעשיית ההייטק.**