

הערות צד שרת

```
@app.route('/sd1', methods=['GET'])
def getSuggestTicket():
```

בדיקת חיזוי ב-8 מקומות נוספים במפה (בכיוון צפון, צפון מזרח, מזרח וכולי) במרחק כ-200 מטרים מהמיקום הנוכחי. הבדיקה מוודאת שהמיקום החדש המוצע הינו מקום חניה חוקי ע"י מאגר היסטוריית הדוחות ומאגר בקשות מהמשתמשים לחיזוי הדוחות (ההנחה שאם משתמש ביקש חיזוי במקום מסוים כנראה שאפשרי לחנות שם).

```
@app.route('/graph', methods=['GET'])
def getGraphData():
```

החזרת מידע למשתמש על כמות הדיווחים הקיימים במאגר הנתונים על פי הקריטריונים: קאורדינטה, זמן, מזג אוויר. הפונקציה מחזירה את כמות הדוחות שהיו ברדיוס 100 מטר מהמיקום המבוקש המחולק לפי שעות היום.

```
@app.route('/gta', methods=['GET'])
def getTicket():
    # gta - Get Ticket Analysis
```

בדיקת חיזוי לקבלת דוח לפי מיקום וטווח שעות חניה. הפונקציה מוודאת שהקלט תקין, לא תקבל בקשות על חיזוי בעבר (רבע שעה אחורה מקסימום). בנוסף לבדיקת כמות הדוחות בטווח השעות, תמיד ייבדקו בנוסף שעה אחורה ושעה קדימה- כאשר הדוחות בזמנים אלו יהיו במשקל נמוך יותר.

נחלק את הדוחות ל-4 קבוצות רלוונטיות:
Calculate_Percentage.sum_data_to_calculate_percentage

- 1) דוחות על פי אותם תנאים- יום בשבוע, מזג אוויר, טווח שעות והאם חג.
- 2) דוחות על פי סעיף 1 רק שנבדוק במזג אוויר דומה למזג האוויר הנוכחי.
- 3) דוחות על פי סעיף 1 רק שנבדוק על שאר ימי השבוע.
- 4) דוחות בשאר ימי השבוע ובמזג אוויר דומה למזג האוויר הנוכחי.

נבצע את חלוקת הקבוצות לכל אחת מ-3 הטבעות הרלוונטיות לחיזוי הדוח: טבעת ראשונה- עד 100 מטר, טבעת 2- עד 200 מטר וטבעת שלישית עד 300 מטרים.

בנוסף לתהליך קביעת האחוז לקבלת דוח נתחשב בתסריטים מסוימים:
1) אם קיימים דוחות במיקום המבוקש אך בימים אחרים בלבד- כנראה הפקח לא עובר ביום הספציפי שנבחר.

2) אם לא קיימים דוחות במיקום המבוקש בטווח השעות שהמשתמש ביקש אך קיימים דוחות בשאר שעות היום- כנראה הפקחים נוטים פחות לעבור בשעות שהמשתמש ביקש.

3) אם לא קיימים דיווחים במזג האוויר המבוקש ובמזג אוויר דומה אך ישנם דיווחים בסוג מזג אוויר שונה- כנראה הפקחים לא עוברים במזג אוויר מסוים באיזור.

4) בדיקת הדיווח האחרון על מעבר פקח באיזור. השפעת גורם זה תלוי במרחק של הפקח מהמיקום הנוכחי ובזמן שחלף מאז.

5) אם אין דוחות כלל באיזור בכל הוויראציות של התנאים – (מזג אוויר שעות וכו') לא נתחייב באחוז נמוך לקבל דוח.

הערות צד לקוח

AutoCompleteSearchBar

בקומפוננט זה מופיעה הפונקציה searchApi אשר מתחברת לשירות של אתר בשם tom-tom ומקבלת אופציות להשלמה אוטומטית אשר מופיעים בתיבת חיפוש.

לינק למידע לדוגמא שמתקבל מהאתר: tom-tom:

<https://developer.tomtom.com/search-api/search-api-documentation-search/fuzzy-search>

כמו כן הבקשות ל tom-tom נעשות בעזרת קומפוננטה בשם tom-tom שמגדירה מראש הגדרות בסיסיות בעזרת axios.

MapComponent

תפקידה של קומפוננטה זו היא להציג מפה (של גוגל מפיס) וסמן אשר ניתן לגרירה.

GetInfoScreen

קומפוננטה זו מציגה למשתמש את המפה ואת ההשלמה האוטומטית ובנוסף מציעה פונקציונליות של GPS על מנת למרכז את המפה למיקום המשתמש.

ReporterScreen

התפקיד של קומפוננטה זו הינה לאפשר למשתמש לבחור זמן ותאריך ולבצע בקשה של דיווח דו"ח או קבלת חיזוי לדו"ח לצד שרת שלנו ולהציג את תשובת השרת למשתמש.

בנוסף קומפוננטה זו מאפשרת לאחר ביצוע בקשה של קבלת חיזוי לדו"ח, לבקש המלצה למיקום קרוב של עד 300 מטר עם הסתברות נמוכה יותר. במידה והשרת מוצא אופציה כזו אז אנו מאפשרים ממשק עם waze ופותרים ניווט למיקום שהומלץ באפליקציית waze.

TicketDayAnalysis

קומפוננטה זו אחראית לקבל מהמשתמש מידע לגבי מזג אוויר רצוי, יום רצוי, ותקופה רצויה

ומעבירה מידע זה בשילוב מיקום נבחר(שהתקבל מ GetInfoScreen) לקומפוננטה בשם DisplayTicketDayAnalysis והיא מבקשת מהצד שרת כמות דוחות בהתאם לנתונים שהזנו ומציגה מידע זה למשתמש בתצורה של גרף עמודות.