חוזר מנהל מה"ט 51-4-11 – נספח מס '1 (הצעה לפרויקט גמר)

תאריך:\_\_\_\_\_\_\_\_\_\_\_

לכבוד

יחידת הפרויקטים מה"ט

הצעה לפרויקט גמר

יש להדפיס את כל הנתונים הנדרשים

### פרטי הסטודנטים

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| תאריך סיום  הלימודים | טלפון נייד | כתובת | ת.ז .9 ספרות | שם הסטודנט |
| …. | 0539315561 | רחוב 120 | 325968071 | פאיז שלאבנה |
|  |  |  |  |  |

שם המכללה:  **המכללה הטכנולוגית נוף** הגליל סמל המכללה: **72209**

מסלול ההכשרה: **הנדסאים**

מגמת לימוד: **תוכנה בהתמחות FS**   
מקום ביצוע הפרויקט: המכללה הטכנולוגית נוף הגליל

### פרטי המנחה האישי

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מקום עבודה/תפקיד | תואר | טלפון נייד | כתובת | שם המנחה \* |
| מכללת נוף הגליל | Bsc/MBA | 0523203282 | יקינטון 23 חיפה | להב רון |
|  |  |  |  |  |

\* עבור מנחה אישי חדש יש לצרף קורות חיים, ניסיון מקצועי ותעודות השכלה לאישור מה"ט.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
| חתימת הסטודנט | חתימת המנחה האישי | חתימת הגורם המקצועי מטעם מה"ט |

דרך מנחם בגין 86 תל אביב ת.ד .36049 מיקוד 67138 טלפון: 7347521-03 פקס: 7347644-03

# שם הפרויקט

"טעינה קרובה" – איתור עמדות טעינה קרובות לפי מיקום

# רקע

## תיאור ורקע כללי

מערכת המאפשרת למשתמשים לאתר תחנות טעינה קרובות לרכבים חשמליים. המשתמשים יוכלו למצוא תחנות לפי המיקום הנוכחי שלהם, בין אם מדובר בעזרת GPS או בעזרת כתובת IP בחיפוש ממחשב רגיל) ולקבל מידע על זמינות תחנות, סוגי חיבורים ועלויות הטעינה. קהל היעד כולל נהגים פרטיים, חברות עם ציי רכבים חשמליים, ובעלי תחנות .טעינה המעוניינים למשוך משתמשים נוספים

## מטרות המערכת

מטרת המערכת היא לאפשר לנהגים של רכבים חשמליים לאתר תחנות טעינה קרובות בקלות, לקבל מידע על זמינות תחנות, סוגי חיבורים ועלויות טעינה, ולספק נתונים בזמן אמת. המערכת גם תעזור לבעלי תחנות .טעינה להציע את שירותיהם ולהגיע ליותר משתמשים

# סקירת מצב קיים בשוק, אילו בעיות קימות

[open map charge :](https://openchargemap.org/site/profile/applications)  
המערכת מאפשרת למשתמשים לחפש תחנות טעינה קרובות לרכב חשמלי באמצעות ממשק API, כולל אפשרויות חיפוש לפי מיקום GPS.  
**בעיה**: למרות התמיכה בחיפוש מבוסס מיקום, אין אינטגרציה מלאה עם מפה אינטראקטיבית שמאפשרת עדכונים בזמן אמת ומידע על זמינות תחנות.

[Google Maps API:](https://developers.google.com/maps/apis-by-platform)

המערכת מציעה אפשרות ניווט ואיתור תחנות טעינה קרובות, כולל הנחיות נסיעה מדויקות ואינטגרציה עם מפות.

בעיה: המערכת אינה מספקת מידע מלא על זמינות תחנות טעינה בזמן אמת או נתונים כמו עלות ושירותים נוספים בתחנה.

[OpenRouteService API (ORS):](https://openrouteservice.org/dev/" \l "/home)

המערכת מספקת הנחיות נסיעה מפורטות, כולל חישוב זמן נסיעה, מרחק בין תחנת הטעינה למיקום המשתמש, ותכנון מסלול אופטימלי.  
**בעיה**:ה-API מספק נתוני ניווט בלבד ואינו תומך בגישה ישירה לנתונים עדכניים ומרוכזים על תחנות טעינה, כגון זמינות, עלויות וסוגי חיבורים, מה שמגביל את חוויית המשתמש הכוללת

.

# מה הפרויקט אמור לחדש או לשפר

הפרויקט מאפשר למשתמשים לאתר תחנות טעינה על המפה בהתאם למיקומם הנוכחי. בנוסף, המשתמש יכול להזין מיקום באופן ידני ולקבל מידע על תחנות טעינה באותו מיקום. כל זאת באמצעות מערכת .מתקדמת שמשלבת מפה, נתונים בזמן אמת על זמינות תחנות, וסוגי חיבורים

# דרישות מערכת ופונקציונאליות

## דרישות מערכת

 **חומרה**:

* שרת ייעודי או שירות ענן עם חיבור מהיר.
* מינימום: מעבד Quad-core 2.4GHz, זיכרון 8GB RAM, אחסון SSD 256GB.
* תמיכה במכשירי אנדרואיד 10/ iOS 13 ומעלה ודפדפני אינטרנט מודרניים.

 **תוכנה**:

* צד לקוח: פיתוח ב-React.js או דומה.
* צד שרת: Node.js עם Express.
* בסיס נתונים: MongoDB.
* מפות: Google Maps API או OpenStreetMap.
* עדכונים בזמן אמת: WebSocket או Firebase.

 **אבטחה**:

* אחסון נתונים מוצפן (AES-256).
* תקשורת מאובטחת (HTTPS).
* בקרת גישה מבוססת תפקידים.

## דרישות פונקציונאליות

רשימת דרישות המשתמש מהמערכת, מהן הפעולות בהן נדרשת המערכת לתמוך.

|  |  |  |  |
| --- | --- | --- | --- |
| מסי מזהה | דרישה | FR | NFR |
| 1 | **כניסה ורישום למערכת** |  |  |
| 1.1 | **המערכת תאפשר רישום משתמשים חדשים באמצעות מילוי טופס דינמי הכולל את הפרטים הבאים:**   * שם מלא. * מספר טלפון * כתובת דואר אלקטרוני – * סיסמה * סוג הפייה המתאימה לרכב – לדוגמה: סוג 1, סוג 2, צ'אדמו, CCS. המערכת תציג רק אפשרויות רלוונטיות בהתאם לסוג הרכב שנבחר. * סוג הרכב – בחירה מתוך רשימה נפתחת, כגון רכב חשמלי מלא, רכב היברידי נטען, רכב מסחרי ועוד. * מספר רישוי רכב. | v |  |
| 1.2 | כתובת האימייל תאומת על ידי שליחת הודעת אימות לדוא"ל שהמשתמש הזין. | v |  |
| 1.3 | המערכת תאפשר למשתמש לבצע רישום ראשוני תחת התנאים הבאים:   1. הסיסמה תכלול לפחות אות אחת גדולה. 2. הסיסמה יכולה להיות משולבת מכל שפה. 3. הסיסמה תכלול לפחות מספר אחד. | v |  |
| 2 | **פונקציונליות עיקרית** |  |  |
| 2.1 | המערכת תאפשר חיפוש תחנות טעינה לפי מיקום בזמן אמת באמצעותGPS או IP. | v |  |
| 2.2 | המערכת תציג עמדות טעינה קרובות על גבי מפה.. | v |  |
| 2.3 | המערכת תאפשר חיפוש תחנה ע"י הזנת מיקום ידני / כתובת**;** | v |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 2.4 | תוצאות חיפוש תחנה יוצגו ב GRID או על המפה ויכללו את הפרטים הבאים: (ילקחו מ API)   1. שם התחנה. 2. כתובת התחנה. 3. סוג המטענים הזמינים. 4. שעות פעילות התחנה. 5. מחיר לשימוש אם קיים 6. זמן המתנה משוער אם המידע זמין בזמן אמת | v |  |
| 2.5 | בבחירת תחנה המשתמש המערכת תציג את הפרטים הבאים   1. שם התחנה. 2. כתובת מלאה של התחנה. 3. כיוון וזמן נסיעה משוער מהמשתמש לתחנה. (API ) 4. סוגי המטענים הזמינים בתחנה. 5. שעות פעילות התחנה. 6. מחיר לשימוש (אם קיים) 7. סטטוס התחנה בזמן אמת (פעילה/לא פעילה). | v |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | **אבטחת מידע** |  | v |
| 3.1 | שמירה על נתוני משתמשים מוצפנים (AES-256) ותקשורת מאובטחת (HTTPS**).** |  | v |
| 3.2 | המערכת תאפשר למנהל להוסיף, לערוך או להסיר תחנות טעינה באמצעות ממשק ייעודי |  | v |
| 3.3 | המערכת תאפשר קליטת נתוני תחנות טעינה ממערכת חיצונית, כאשר המידע יתקבל מ  Open Charge Map API | v |  |
| 3.4 | המערכת תאפשר מיקום על גבי המפה תוך שימוש ב  Goole Maps API | v |  |
| 3.5 | המערכת תאפשר שימוש בניווט אל התחנה תוך סיוע ב ORS API. | v |  |
| 4 | **חיפוש ואיתור** |  |  |
| 4.1 | המערכת תאפשר חיפוש תחנות טעינה בזמן אמת לפי מיקום המשתמש (IP או GPS באמצעות), בהתבסס על הנתונים הזמינים. | v |  |
| 4.2 | המערכת תציג תוצאות חיפוש תחנה על גבי המפה וב Grid מתחת למפה. |  |  |
| 4.3 | המערכת תאפשר חיפוש מותאם אישית על ידי הזנת כתובת ידנית עיר, רחוב או מיקוד) לאיתור תחנות רלוונטיות. | v |  |
| 5 | **ממשק מנהל** |  |  |
| 5.1 | המערכת תאפשר למנהל המערכת להוסיף ולערוך תחנות טעינה באופן ידני או לשלב תחנות קיימות באמצעות Open Charge Map API.   * הוספת תחנה ידנית – הזנת פרטים כמו מיקום, סוג הפייה וזמינות התחנה. * עדכון נתונים שוטף על מיקומים, זמינות וסוגי טעינה   שילוב אוטומטי באמצעות  Open Charge Map API  ממאגר התחנות העולמי. | v |  |
| 5.2 | המערכת תאפשר התאמת תצוגת המפה – תצוגת המפה תתעדכן אוטומטית, כולל סימון המיקום החדש והתאמת נקודות עניין באמצעות. (Google Maps API) |  | v |
| 5.3 | **המערכת תאפשר חישוב מסלול וזמני נסיעה מעודכנים** – לאחר עדכון מיקום תחנה, המערכת תחשב מחדש את זמני הנסיעה והמסלול לטובת המשתמשים, בהתבסס על נתוני **ORS API**. | v |  |
| 6 | **תאימות** |  | v |
| 6.1 | המערכת תאפשר **תמיכה בדפדפנים נפוצים** המערכת תפעל בצורה תקינה ב Google Chrome.1  Microsoft Edge.2  Safari.3 |  | v |
| 6.2 | המערכת תתאים אוטומטית את התצוגה לפי גודל המסך באמצעות  **window.innerWidth JavaScript**, **CSS, Media Queries**  ותתמוך בטלפונים, טאבלטים ומחשבים ניידים |  | v |
| 7 | **תחזוקה וניהול** |  |  |
| 7.1 | המערכת תאפשר ניטור ובקרה של פעילות התחנות – מנהל המערכת יוכל לעקוב אחר נתוני פעילות התחנות בזמן אמת, כגון מצב התחנה וזמינותה, ולקבל התראות בסיסיות במקרה של תקלות משמעותיות או אי-זמינות מתמשכת. | v |  |
| 7.2 | המערכת תאפשר ניטור ובקרת שגיאות  תקלות יתועדו בלוגים באמצעות  **ELK Stack** או **CloudWatch**  עם התראות בזמן אמת |  | v |

# בעיות צפויות במהלך הפיתוח ופתרונות:

|  |  |
| --- | --- |
| **בעיה צפויה (6.1)** | **פתרון אפשרי (6.2)** |
| אינטגרציה מורכבת בין מספר APIs חיצוניים (Google Maps, ORS, OpenChargeMap) עלולה לגרום לשגיאות או לאי־התאמה בנתונים | שימוש במודול **APIHandler** לבדיקת תקינות הנתונים, הטמעת מנגנון **Retry** במקרה של כישלון זמני, ואיחוד הנתונים למבנה אחיד לפני הצגתם למשתמש. |
| זמינות חלקית או חוסר מידע בזמן אמת על תחנות טעינה ממערכות חיצוניות. | הצגת הודעת fallback ידידותית (כגון "מידע לא זמין"), שימוש ב־**Cache** לשמירה זמנית של נתונים עדכניים, והתראות למנהל במקרה של חזרות מרובות על שגיאות. |
| בעיות תאימות בהצגת הממשק על מסכים בגדלים שונים (מחשב, טאבלט, סמארטפון). | שימוש ב־**CSS Media Queries** וב־**JavaScript (window.innerWidth)** להתאמה אוטומטית של התצוגה, וביצוע בדיקות **QA** במספר דפדפנים ומכשירים. |
| עומס על השרת עקב ריבוי חיפושים או גידול בכמות המשתמשים. | פריסת המערכת על שרת **ענן סקיילבילי**, שימוש באינדקסים במסד הנתונים **MongoDB**, והוספת **Load Balancer** לשיפור הביצועים והזמינות. |
| סיכוני אבטחת מידע כגון גניבת נתונים אישיים או שימוש לא מורשה במפתחות API. | שימוש ב־**HTTPS** לתקשורת מאובטחת, הצפנת נתונים עם **AES-256**, הטמעת **JWT** לאימות משתמשים, ושמירה מאובטחת של מפתחות API בצד השרת בלבד. |
| סיכון לנפילות מערכת או אובדן נתונים. | שימוש ב־**Replica Set** ב־MongoDB להבטחת זמינות גבוהה, גיבוי יומי אוטומטי, והפעלת **Retryable Writes** לטיפול בכשלי כתיבה. |

# פתרון טכנולוגי נבחר:

## טופולוגית הפתרון - deployment

## טכנולוגיות בשימוש

Visual Studio Code – סביבת פיתוח מתקדמת לעריכת וכתיבת קוד בצורה נוחה.

MongoDB – מערכת לניהול בסיסי נתונים המבוססת על מסמכים.

## שפות הפיתוח:

**צד לקוח:**

JSX – תחביר המשלב קוד JavaScript עם רכיבי תצוגה, ליצירת ממשקים דינמיים.

CSS – שפת עיצוב המשמשת לעיצוב והצגת דפי אינטרנט בצורה ויזואלית.

**צד שרת:**

JavaScript – שפת תכנות ליצירת אינטראקטיביות ולוגיקה בצד השרת.

Node.js – פלטפורמה להרצת קוד JavaScript מחוץ לדפדפן.

בסיס נתונים:

MongoDB – מסד נתונים מבוסס מסמכים (Documents) בפורמט JSON, המאפשר אחסון גמיש ודינמי.

## תיאור הארכיטקטורה הנבחרת-

שכבת הנתונים (Database):

אחסון מאובטח ואמין של כל נתוני המערכת – משתמשים, ספרים, השכרות ודוחות.

שכבת הגישה לנתונים (DAL – Data Access Layer):

קישור מאובטח בין השרת למסד הנתונים, כולל פעולות קריאה, כתיבה, עדכון ומחיקה.

מודולים ייעודיים לניהול השאלות, חיפושים ובדיקות תנאים.

ממשק המשתמש (UI):

תצוגה אינטראקטיבית מותאמת לכל סוגי המכשירים, עם חוויית שימוש נוחה.

דוחות (Reporting):

הפקת דוחות דינמיים מתוך הנתונים, כגון סטטיסטיקות השכרה והחזרות.

## חלוקה לתכניות ומודולים

Class User – מייצג משתמש במערכת. כולל פרטי התחברות (אימייל, סיסמה), פרטי רכב (מספר רישוי, סוג שקע טעינה), ומיקום נוכחי. אחראי על חיפוש תחנות טעינה וקבלת נתונים מותאמים אישית.

Class Station – מייצג תחנת טעינה. כולל שם תחנה, כתובת, סוגי חיבורים זמינים, שעות פעילות, עלות שימוש, זמינות בזמן אמת.

Class APIHandler – מחלקה שאחראית על התקשורת מול שירותי צד שלישי:

Open Charge Map API – לשליפת תחנות טעינה.

Google Maps API – להצגת מפה, חישוב מיקום והצגת מסלול.

ORS API – חישוב מסלולי נסיעה וזמני הגעה.

Class Manager – מנהל מערכת. מאפשר הוספה, עדכון ומחיקה של תחנות טעינה באופן ידני או דרך ה־API. כולל ממשק בקרה לניטור מצב התחנות, זמינות, ושגיאות מערכת.

Class Report – אחראית ליצירת דוחות סטטיסטיים: מספר משתמשים רשומים, כמות חיפושי תחנות, זמני טעינה ממוצעים, ותקלות תחנה.

Class DatabaseHandler – מנהל את הקשר מול מסד הנתונים (MongoDB). כולל שמירת משתמשים, תחנות טעינה שהוזנו ידנית, נתוני חיפושים ודוחות.

## סביבת השרת:

React.js עם JSX:

פיתוח צד לקוח באמצעות ממשק אינטראקטיבי, כולל שימוש ב-API וחיבור לשרת דרך Node.js.

MongoDB:

מסד נתונים מבוסס מסמכים, מנוהל באמצעות Mongoose, תומך בגמישות ובסקלאביליות גבוהה.

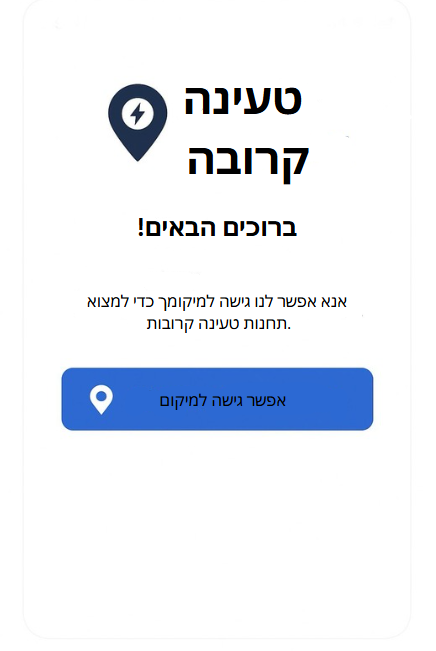
אבטחה:

שימוש בפרוטוקול HTTPS להצפנת התקשורת בין לקוח לשרת.

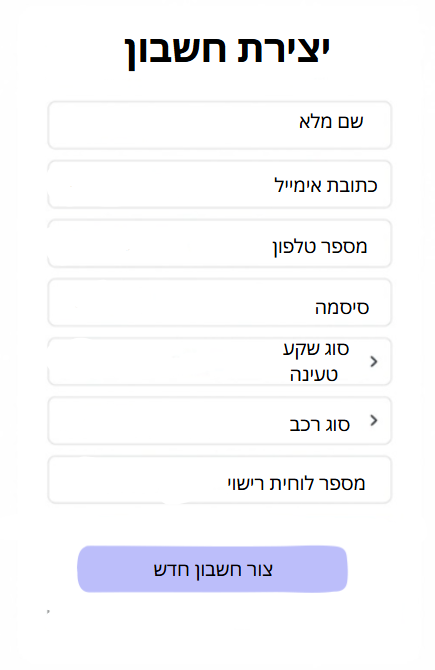
שימוש ב-JSON Web Token (JWT) לאימות משתמשים בצורה מאובטחת וללא שמירת מצב (stateless).

## ממשק המשתמש/לקוח - GUI :

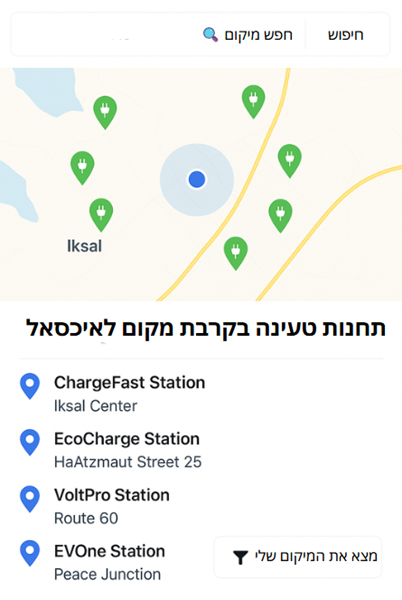
**מסך ברוך הבא עם גישת מיקום**



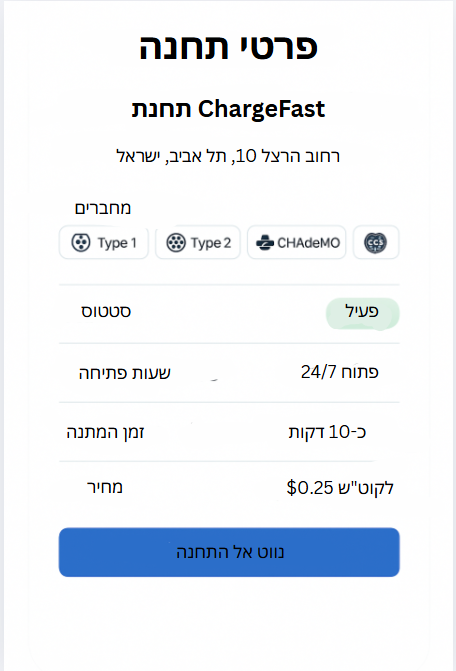
**מסך התחברות / יצירת חשבון**



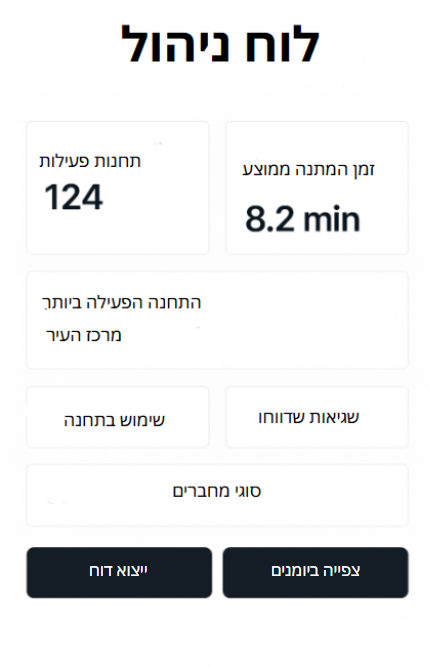
**מפת תחנות טעינה קרובות**



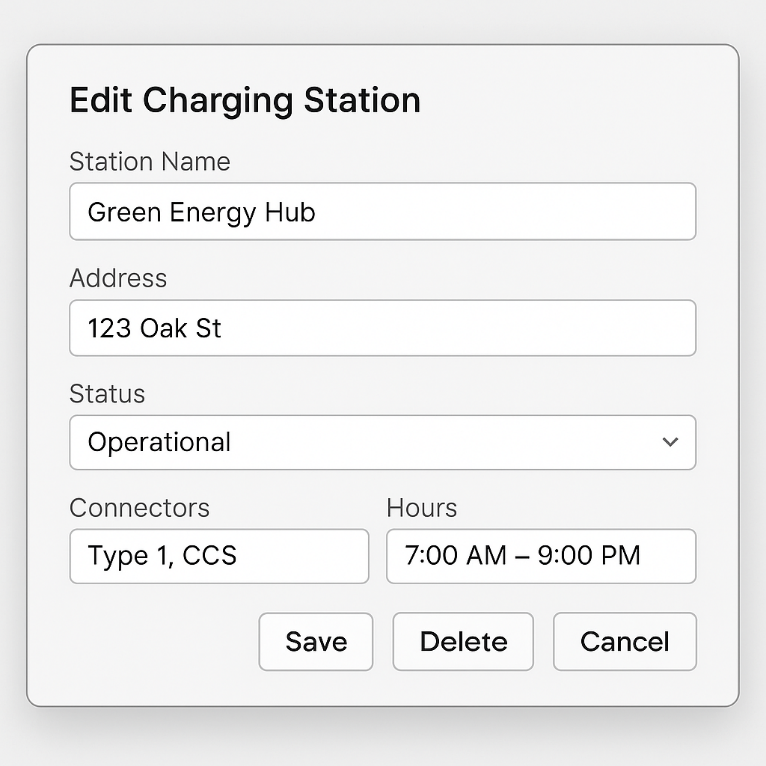
**פרטי תחנת טעינה**



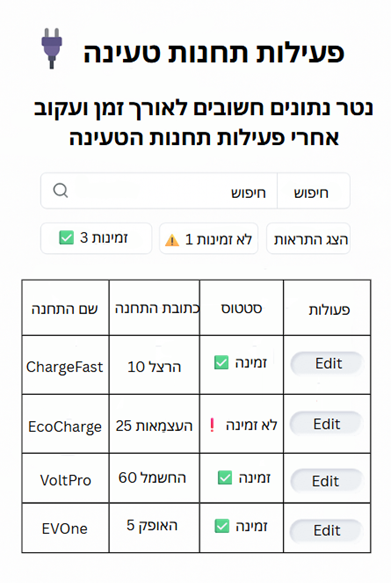
**לוח ניתוחים למנהל**



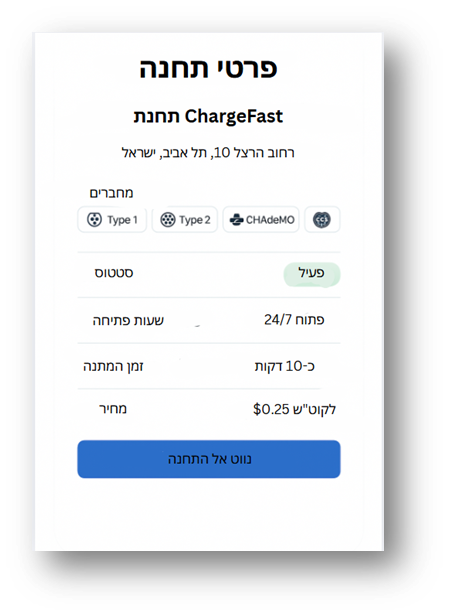
**עריכת תחנת טעינה**



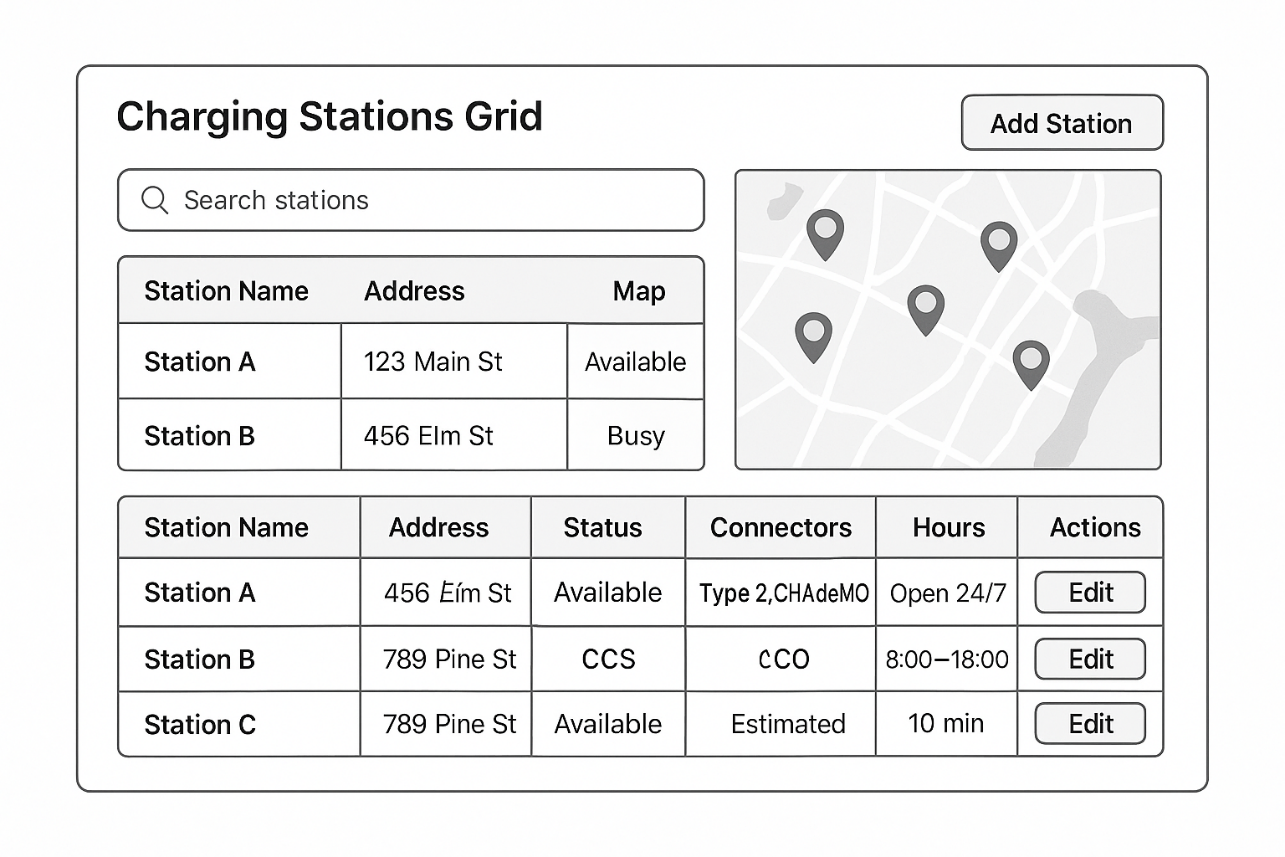
**פעילות תחנות טעינה – לוח בקרה**

****

**פרטי תחנת טעינה**

****

**טבלת תחנות טעינה עם מפה**



## ממשקים למערכות אחרות/ API:

Open Charge Map API.1 – משמש לקבלת מידע על תחנות קיימות.

Google Maps API.2 – להצגת התחנות על מפה, ניווט, וחישוב מיקום.

MongoDB Atlas.3 – ממשק למסד נתונים בענן, דרך ספריית Mongoose.

## שימוש בחבילות תוכנה:

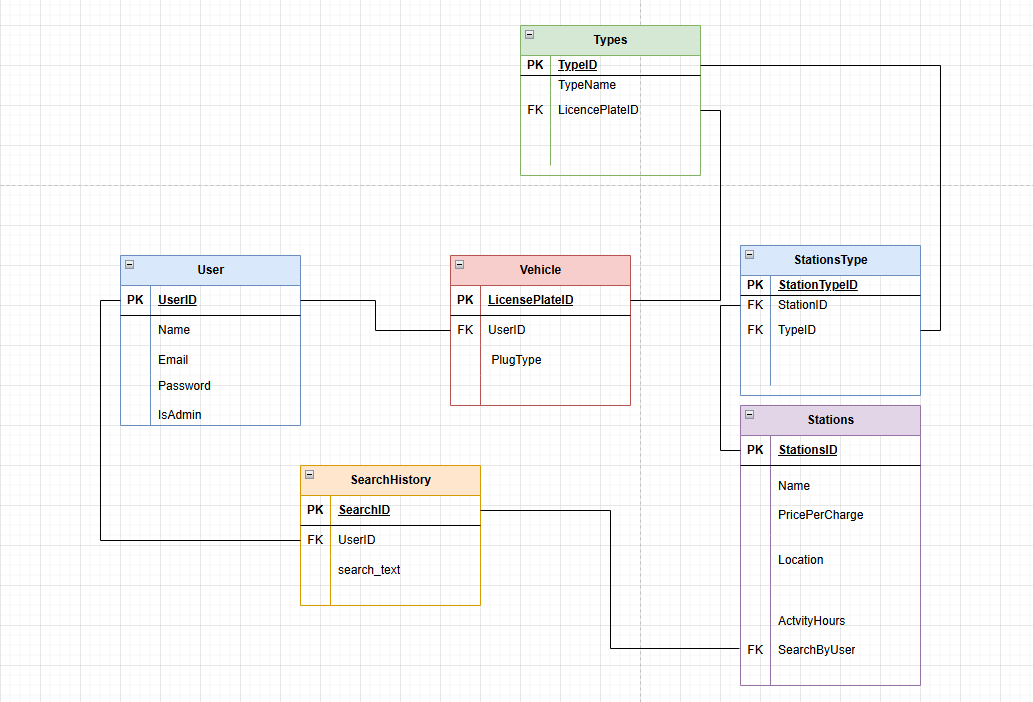
MVC (Model-View-Controller).1 – צד השרת מחולק למודלים, בקרים וראוטים.

Singleton.2 – התחברות למסד נתונים מתבצעת פעם אחת בלבד.

Separation of Concerns.3 – הפרדת אחריות בין רכיבים שונים במערכת

# שימוש במבני נתונים וארגון קבצים

## ERD.



## נא פרט את מבני הנתונים.

Data Structures – מבני נתונים

Users Collection – אוסף משתמשים הכולל: email, passwordHash, fullName, phone, vehicleId, plugType, createdAt.

Vehicles Collection – אוסף רכבים, כאשר licensePlate משמש כמזהה ייחודי (ID), כולל שדות: make, model, year, plugType.

Stations Collection – אוסף תחנות טעינה הכולל: name, address, location { type, coordinates }, openingHours, priceInfo, status (פעילה/לא פעילה), source (api/manual).

Types Collection – אוסף סוגי טעינה (Type1 / Type2 / CHAdeMO / CCS).

StationTypes Collection – טבלת קשר many-to-many בין תחנה ↔ סוג טעינה, עם שדות: stationId, typeId (ייחודי למניעת כפילויות).

Logs/Imports Collection – יומן יבוא מ־Open Charge Map ושגיאות מניטור המערכת.

## נא פרט את שיטת האחסון :

MongoDB Atlas – בסיס נתונים בענן, שמירת מסמכי JSON/BSON.

2dsphere Index – אינדקס מרחבי על Stations.location לחיפושי קרבה גיאוגרפיים.

Unique Index – אינדקס ייחודי על Vehicles.licensePlate ועל צירוף stationId + typeId.

TTL Cache – אחסון זמני (Time To Live) לנתוני API להאצת ביצועים.

## נא ציין מנגנוני התאוששות מנפילה\קריסה\תמיכה בטראנזקציות.

Replica Set – אשכול שכפול ב־Atlas לזמינות גבוהה.

Daily Backup – גיבוי יומי אוטומטי ושחזור נקודתי לפי צורך.

Write Concern (majority) – כתיבה מאובטחת עם אימות מרוב צמתים.

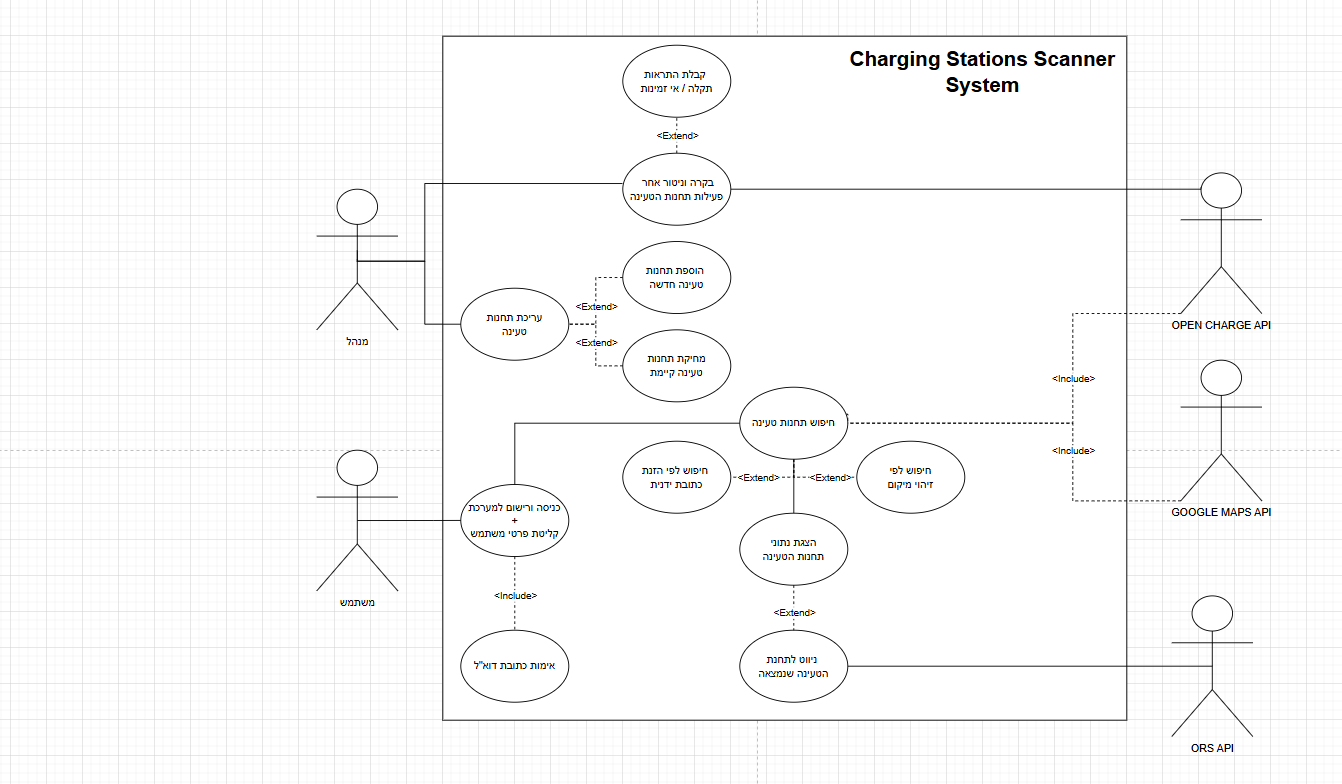
Retryable Writes – ניסיון חוזר אוטומטי במקרה של ניתוק או כישלון זמני.

Transactions – שימוש בטרנזקציות לעדכון בו־זמני של כמה אוספים.

Error Logs – רישום שגיאות והתראות בזמן אמת לניטור תקלות מערכת.

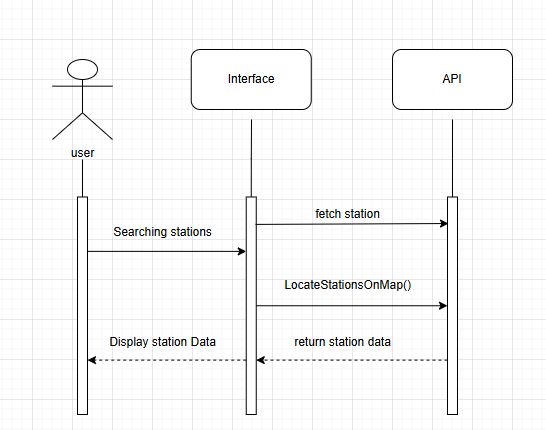
# תרשימי מערכת מרכזיים

## Use Case



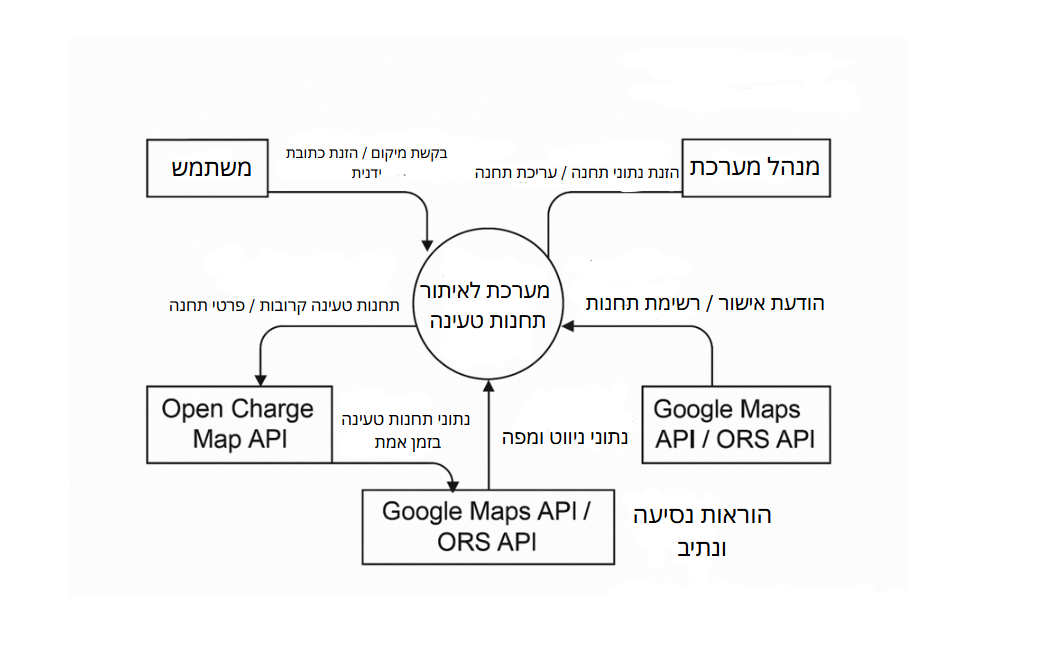
## Sequence Diagram

Sequence digram for p2 station searching data

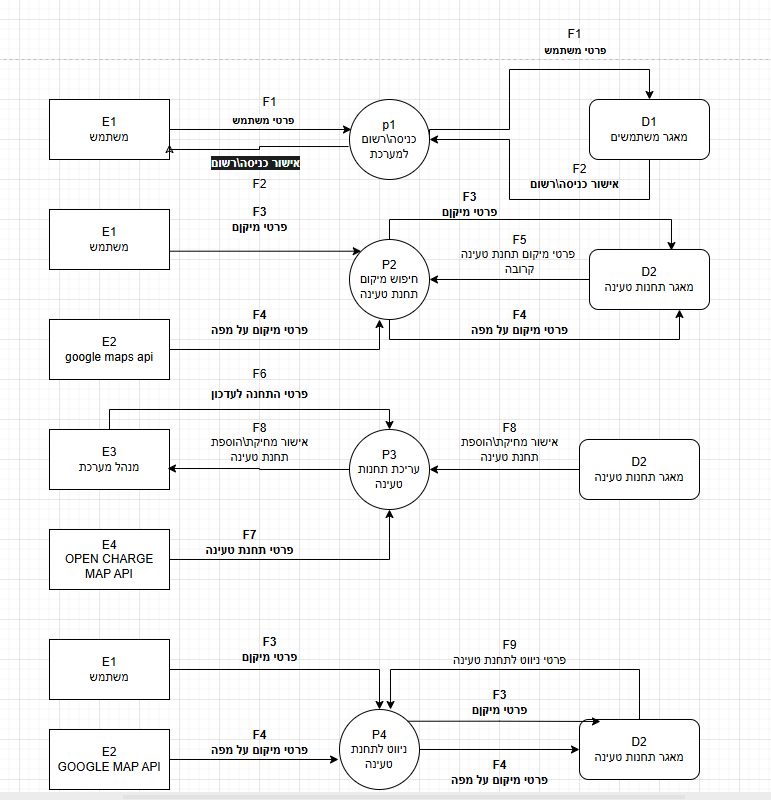


## Data flow

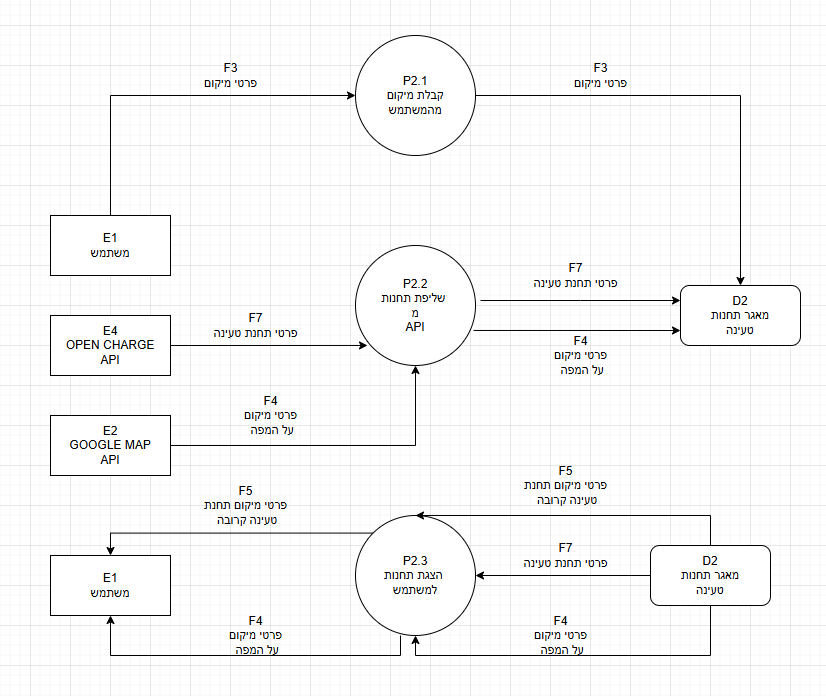
Context Diagram



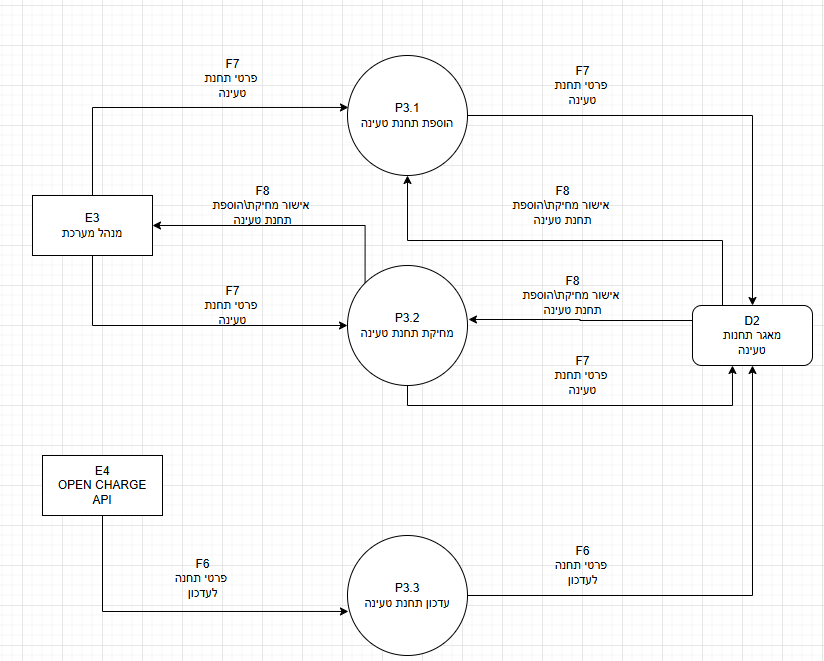
**DFD Level 0 – Electric Vehicle Charging Station Management System**



Dfd1 for P2 – Station searching details



Dfd1 for P3 – Station editing details



**מילון**

תהליכים (P)

|  |  |  |
| --- | --- | --- |
| **תיאור** | **שם** | **קוד** |
| תהליך רישום משתמש חדש | רישום למערכת | P1 |
| חיפוש תחנות טעינה לפי מיקום | חיפוש תחנת טעינה | P2 |
| ניהול, הוספה או עדכון תחנה ע"י המנהל | עריכת תחנת טעינה | P3 |
| ניווט לתחנת טעינה שנבחרה | ניווט לתחנת טעינה | P4 |
| שלב של קבלת המיקום מהמשתמש | קבלת מיקום מהמשתמש | P2.1 |
| שליפת תחנות מה־API החיצוני | שליפת תחנות מ־API | P2.2 |
| הצגת התחנות בממשק משתמש | הצגת התחנות למשתמש | P2.3 |
| הוספת תחנה חדשה | הוספת תחנת טעינה | P3.1 |
| מחיקת תחנה קיימת | מחיקת תחנת טעינה | P3.2 |
| עדכון פרטי תחנה קיימת | עדכון תחנת טעינה | P3.3 |

מאגרי נתונים (D)

|  |  |  |
| --- | --- | --- |
| **תיאור** | **שם** | **קוד** |
| בסיס נתונים המכיל את כל המשתמשים | מאגר משתמשים | D1 |
| בסיס נתונים של תחנות טעינה | מאגר תחנות טעינה | D2 |

ישויות חיצוניות (E)

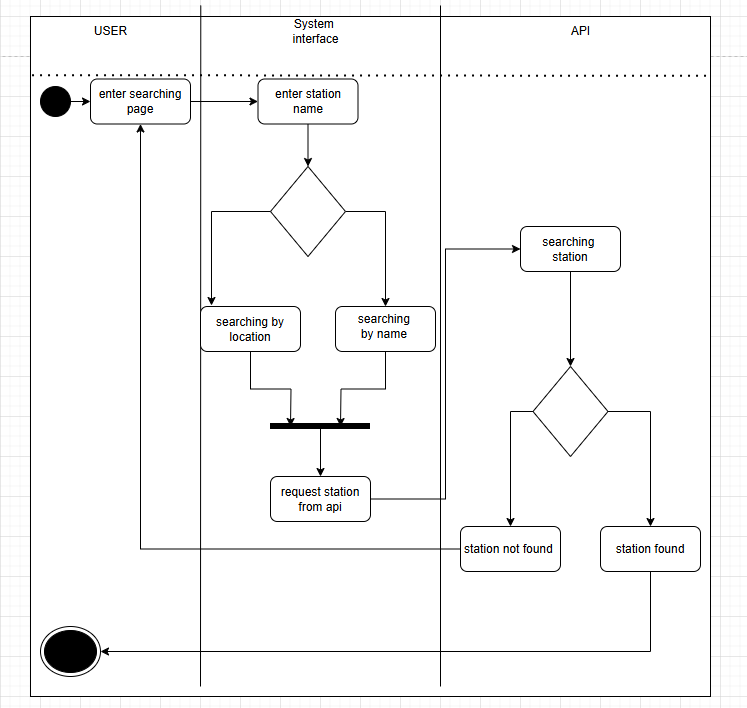
|  |  |  |
| --- | --- | --- |
| **תיאור** | **שם** | **קוד** |
| האדם שמשתמש במערכת | משתמש | E1 |
| שירות מפות להצגת מיקומים | GOOGLE MAP API | E2 |
| אדם שמנהל את המערכת | מנהל מערכת | E3 |
| API ציבורי המכיל תחנות טעינה | OPEN CHARGE MAP API | E4 |

זרימות נתונים (F)

|  |  |  |
| --- | --- | --- |
| **תיאור** | **שם** | **קוד** |
| שם, אימייל, מיקום וכו’ | פרטי משתמש | F1 |
| אישור כניסה או רישום | אישור כניסה/רישום | F2 |
| נתוני מיקום של המשתמש | פרטי מיקום | F3 |
| מיקום גאוגרפי להצגה על המפה | פרטי מיקום על מפה | F4 |
| תוצאה של חיפוש תחנה קרובה לפי מיקום | פרטי מיקום תחנת טעינה קרובה | F5 |
| פרטים שיש לעדכן בתחנה | פרטי תחנה לעדכון | F6 |
| נתוני תחנה שלמה | פרטי תחנת טעינה | F7 |
| קלט של מחיקה או הוספה של תחנה | אישור מחיקה/הוספת תחנת טעינה | F8 |
| נתוני תחנה + מסלול ניווט | פרטי ניווט לתחנת טעינה | F9 |

# תיאור המרכיב האלגוריתמי – חישובי

**ACTVITY DIGRAM FOR SEARCHING STATIONS**



## איזו בעיה בא הפרוייקט לפתור, ואיך יפתור?

Problem Definition – נהגים של רכבים חשמליים מתקשים למצוא תחנות טעינה קרובות פעילות, לדעת את זמינותן, את סוגי השקע המתאימים לרכבם, ואת זמן ההגעה המשוער.

Solution Approach – המערכת מבצעת חישוב מבוסס מיקום (GPS / IP), שולפת נתוני תחנות בזמן אמת מ־Open Charge Map API, מחשבת מסלול וזמן הגעה בעזרת ORS API, ומציגה על המפה תחנות רלוונטיות בלבד לפי סוג החיבור של המשתמש.

## איסוף מידע וניתוחים סטטיסטיים (אנליטיקות)

Data Collection – המערכת אוספת מידע על חיפושי משתמשים, תחנות נבחרות, זמינות תחנות בזמן אמת, וזמני נסיעה מחושבים.

Analytics – הנתונים משמשים ליצירת סטטיסטיקות כגון: תחנות מבוקשות ביותר, ממוצע זמני המתנה, דפוסי שימוש לפי שעות/ימים, וזיהוי תחנות בעייתיות.

Reports – מנהל המערכת יכול להפיק דוחות מסכמים המבוססים על הנתונים הללו כדי לשפר את השירות, להוסיף תחנות חדשות או לשנות הגדרות קיימות.

# תיאור/התייחסות לנושאי אבטחת מידע

HTTPS Protocol – שימוש בחיבור מאובטח בין הלקוח לשרת, הצפנת כל התקשורת.

JWT (JSON Web Token) – אימות משתמשים בצורה מאובטחת, ללא שמירת מצב (stateless).

AES-256 Encryption – הצפנת סיסמאות ונתוני משתמשים בבסיס הנתונים.

Role-Based Access Control – בקרת גישה לפי תפקיד (משתמש רגיל, מנהל מערכת).

API Security – שימוש במפתחות API מוגנים ובקרה על גישה לשירותי צד שלישי (Open Charge Map, Google Maps, ORS).

Error Logging & Monitoring – ניטור שגיאות ותיעודן ביומן מאובטח למניעת דליפת מידע רגיש.

# משאבים הנדרשים לפרויקט:

## מספר שעות המוקדש לפרויקט, חלוקת עבודה בין חברי הצוות

סה"כ כ 300 שעות עבודה, מתוכן: 270 שעות לפיתוח צד שרת 130 שעות לפיתוח צד לקוח.

## ציוד נדרש

מחשבים ניידים עם מערכת הפעלה Windows/Mac, חיבור אינטרנט יציב, וסמארטפון אנדרואיד/iOS לבדיקות בשטח.

## תוכנות נדרשות

Visual Studio Code – סביבת פיתוח.

Node.js + Express – צד שרת.

React.js – צד לקוח.

MongoDB Atlas – בסיס נתונים בענן.

Postman – בדיקות API.

GitHub – ניהול גרסאות.

## ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט

עבודה עם MongoDB ומודלים ב־Mongoose.

שימוש ב־Open Charge Map API לשליפת תחנות טעינה.

שילוב Google Maps API להצגת מפות וניווט.

אינטגרציה עם ORS API לחישוב מסלולים.

אבטחת מידע באמצעות JWT ו־HTTPS.

## ספרות ומקורות מידע

1. **-Flanagan, D. (2020). *JavaScript: The Definitive Guide*. O’Reilly Media**ספר יסוד להבנת JavaScript ולפיתוח צד לקוח וצד שרת.
2. **-React – Official Documentation** (https://react.dev) תיעוד רשמי של, React כולל מדריכים, דוגמאות ושיטות עבודה מומלצות.
3. **-Node.js – Official Documentation** (https://nodejs.org) תיעוד מלא לפיתוח צד שרת באמצעות Node.js.
4. **-MongoDB – Official Documentation** ([https://www.mongodb.com/docs](https://www.mongodb.com/docs?utm_source=chatgpt.com)) מסד נתונים מבוסס מסמכים, כולל שימוש ב־Atlas ו־Mongoose.
5. **Google Maps API – Developers Documentation** (https://developers.google.com/maps) – תיעוד רשמי לשילוב מפות, מיקום, ניווט וחישוב מסלולים במערכות ווב ומובייל.
6. **Open Charge Map API – Developer Documentation** (https://openchargemap.org/site/develop/api) – תיעוד השירות הציבורי לשליפת נתוני תחנות טעינה בזמן אמת.
7. **OpenRouteService (ORS) – API Documentation** (https://developer.openrouteservice.org/) – תיעוד רשמי לחישוב מסלולים, זמני נסיעה, איזוקרונות וניווט.
8. **Stallings, W. (2018). *Cryptography and Network Security: Principles and Practice*.**

**Pearson** –   
ספר אקדמי בנושא אבטחת מידע והצפנה, תומך בהטמעת HTTPS, JWT והצפנת נתונים.

# תכנית עבודה ושלבים למימוש הפרויקט

|  |  |  |
| --- | --- | --- |
| 1. | קבלת החלטה על נושא הפרויקט והתנעה ראשונית | 12/2024 |
| 2. | כתיבת הצעה/ ניתוח פרויקט וממשקים | 12/09/2025 |
| 3. | Code Review 1 + מסירת ספרי פרויקט | 12/09/2025 |
| 4. | Code Review 1 + ועדה פנימית | 15/10/2025 |

# תכנון הבדיקות שיבוצעו

|  |  |
| --- | --- |
| **תיאור בדיקה** | **תוצאות** |
| רישום משתמש חדש ואימות דואר אלקטרוני | אם מצליח: המשתמש נרשם בהצלחה, מתקבלת הודעת אימות בדואר האלקטרוני, ההפעלה מתבצעת והחשבון מאומת  ────────────────────  אם נכשל: ההרשמה נכשלת או שהנתונים אינם תקינים, מופיעה הודעת שגיאה ברורה ואין יצירת חשבון |
| כניסה למערכת לאחר הקלדת דואר אלקטרוני וסיסמה ולחיצה על התחבר | אם מצליח: המשתמש נכנס למערכת, נוצר אסימון זיהוי פנימי ויש מעבר למסך הראשי  ────────────────────  אם נכשל: מופיעה הודעה דואר אלקטרוני או סיסמה שגויים, אין כניסה ואין יצירת אסימון |
| חיפוש תחנות לפי מיקום המכשיר | אם מצליח: מתקבלת הרשאת מיקום והמכשיר מספק נקודת מיקום, תחנות מופיעות במפה ובטבלה בתוך זמן קצר עד שתי שניות  ────────────────────  אם נכשל: ההרשאה נדחתה או שהמיקום אינו זמין, מוצגת הודעה ידידותית והצעה לחיפוש לפי זיהוי רשת או הזנת כתובת ידנית |
| חיפוש תחנות לפי כתובת ידנית | אם מצליח: המפה מתמרכזת לכתובת והתחנות המתאימות מוצגות במפה ובטבלה  ────────────────────  אם נכשל: הכתובת אינה מזוהה, מופיעה הודעה לא נמצאה כתובת והמערכת ממשיכה לפעול כרגיל |
| הצגת תוצאות תחנות במפה ובטבלה כולל פרטי תחנה | אם מצליח: לכל תחנה מוצגים שם, כתובת, סוגי חיבור, שעות פעילות, מחיר אם קיים ומצב פעילות  ────────────────────  אם נכשל: חלק מן המידע חסר מן השירות החיצוני, במקום שבירה מוצג לא זמין והמסך נשאר יציב |
| פתיחת פרטי תחנה וחישוב מסלול נסיעה | אם מצליח: נפתח מסך פרטים ומתקבל זמן ומרחק משירות המסלולים, קו המסלול מצויר על המפה  ────────────────────  אם נכשל: שליפת הפרטים או חישוב המסלול נכשל, מופיעה הודעת לא ניתן לבצע כעת והמסך נשאר מגיב |
| יבוא תחנות ממאגר ציבורי | אם מצליח: התחנות מיובאות ומאוחדות ללא כפילויות ומופיעה התראת הצלחה עם הכמות  ────────────────────  אם נכשל: היבוא נכשל עקב תקלה בשירות או במבנה הנתונים, מופיעה הודעת כישלון ואין שינוי בנתונים |
| הוספת תחנה חדשה על ידי מנהל מערכת | אם מצליח: התחנה נשמרת בבסיס הנתונים ומופיעה במפה ובטבלה  ────────────────────  אם נכשל: השמירה נכשלת או שהשדות אינם תקינים, מופיעה הודעת שגיאה והתחנה אינה מתווספת |
| עריכת תחנה קיימת על ידי מנהל מערכת | אם מצליח: השינויים נשמרים ומופיעים מיד במפה וברשימה  ────────────────────  אם נכשל: השמירה נכשלת או שיש התנגשות נתונים, מופיעה הודעת שגיאה והערכים הקודמים נשמרים |
| ניטור ותיעוד תקלות | אם מצליח: שגיאה יזומה נרשמת ביומן עם חומרה, זמן ומודול ומתקבלת התראה אם הוגדרה  ────────────────────  אם נכשל: אין רישום ביומן או אין התראה, נדרש תיקון הגדרות הניטור |

|  |  |
| --- | --- |
|  |  |
|  |  |
| חתימת הסטודנט | חתימת המנחה האישי |

### הערות ראש המגמה במכללה

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### אישור ראש המגמה

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
| שם | חתימה | תאריך |

### הערות הגורם המקצועי מטעם מה"ט

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### אישור הגורם המקצועי מטעם מה"ט

שם: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימה: \_\_\_\_\_\_\_\_\_\_\_\_ תאריך:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_