**מדריך קבצי JavaScript - מעודכן על בסיס הקוד הקיים**

**קבצים עיקריים (שורש)**

**calendar-integration.js**

**מטרה בפועל:** קובץ מרכזי המשמש כמתאם בין כל המודולים ואוסף נתונים לשליחה סופית

**פונקציות שזוהו בקוד:**

* **collectFormData()** - פונקציה מרכזית ומורכבת שאוספת את כל הנתונים מהטופס
  + מטפלת בלוגיקת רכב שני (hasSecondCar) ושיתוף כתובות
  + אוספת נתוני vehicle data מ-dataset attributes (צבע, גיר, סוג מנוע וכו')
  + אוספת נתוני תקלות וסוגי גרר מהמערכת החדשה
  + יוצרת מבנה מורכב של formData עם כל הנתונים הדרושים
  + מחברת נתונים מ-PricingManager ו-VehicleManager
  + **כוללת נתוני שטחים (isOutskirts) ופרטי תשלום**
* **collectPricingData()** - אוסף נתוני תמחור מ-PricingManager
  + קורא את המצב הנוכחי של התמחור (טיר נבחר, מחירים מחושבים)
  + מטפל במחירים ידניים מול אוטומטיים
  + **כולל מחיר סופי ונתוני שטחים לשליחה לגיליון**
* **collectDefectDetails()** - חדש - אוסף תקלות נבחרות מהמודל האינטראקטיבי
  + קורא תגיות תקלות מהקונטיינר
  + מחזיר מחרוזת מפורמטת של התקלות
* **collectTowSelection()** - חדש - אוסף סוגי גרר נבחרים מהמודל
  + קורא תגיות גרר מהקונטיינר
  + מחזיר מחרוזת מפורמטת של סוגי הגרר
* **processAddress()** - מטפל בכתובות מורכבות עם טקסט מקורי וכתובת פיזית
* **resetFormKeepUserData()** - איפוס מתקדם של הטופס
  + מנקה לחלוטין את כל נתוני הכתובות והרכבים
  + מאפס בחירות תקלות וגרר
  + משמר רק פרטי משתמש (אימייל, חברה, מחלקה)
  + מאפס את מערכת התמחור למצב התחלתי

**חשיפה גלובלית של פונקציות:**

* window.collectDefectDetails - גישה לפונקציית איסוף התקלות
* window.collectTowSelection - גישה לפונקציית איסוף הגרר

**תיקיית js/modules/**

**FormManager.js**

**מטרה בפועל:** מנהל את האינטראקציות והלוגיקה של הטופס

**פונקציות חדשות שנוספו:**

* **setupDefectSelector()** - מאתחל מערכת בחירת התקלות
* **showDefectModal()** - מציג מודל בחירת התקלות
* **createDefectModal()** - יוצר את המודל האינטראקטיבי לתקלות
* **setupDefectOptions()** - מגדיר אירועי בחירה לתקלות (בחירה מרובה)
* **confirmDefectSelection()** - מאשר ושומר בחירת תקלות
* **updateSelectedDefects()** - מעדכן תצוגת התקלות הנבחרות
* **setupTowSelector()** - מאתחל מערכת בחירת הגרר
* **showTowModal()** - מציג מודל בחירת הגרר
* **createTowModal()** - יוצר את המודל האינטראקטיבי לגרר
* **setupTowOptions()** - מגדיר אירועי בחירה לגרר (בחירה מרובה)
* **confirmTowSelection()** - מאשר ושומר בחירת הגרר
* **updateSelectedTow()** - מעדכן תצוגת הגרר הנבחר

**פונקציות קיימות:**

* ניהול מעברים בין סוגי גרירה
* הצגה/הסתרה של חלקי טופס
* ניהול רכב שני (הוספה והסרה)
* ניהול שיתוף כתובות בין רכבים
* אימות טופס לפני שליחה
* ניהול מצבי UI (loading, success, error)

**PricingManager.js**

**מטרה בפועל:** מחלקה מלאה וחכמה לניהול תמחור דינמי

**פונקציות מרכזיות שזוהו בקוד:**

* **init()** - מאתחל את כל המערכת כולל Google Maps
* **calculateTotalPrice()** - פונקציה מרכזית המחשבת מחיר כולל
* **calculateDistance()** - משתמשת ב-Google Maps Distance Matrix API
* **setupAutomaticCalculation()** - מערכת חכמה למעקב אחר שינויים
* **getRecommendedTier()** - לוגיקה מתקדמת להמלצת מחיר לפי זמן
* **updatePriceDisplay()** - מעדכן את כל התצוגות של המחירים
* **recalculatePrices()** - מחשב מחדש בעת שינוי הגדרות
* **handleOutskirtsToggle()** - מטפל בשינוי מצב שטחים (+25%)

**עדכונים במערכת Google Apps Script**

**שינויים בגיליון הנתונים:**

**סדר העמודות החדש:**

1. תאריך יצירה
2. תאריך ביצוע
3. שעת ביצוע
4. מספר הזמנה
5. חברה
6. מחלקה
7. שם לקוח (אם קיים)
8. משתמש יוצר
9. מספר רכב
10. סוג רכב/דגם
11. צבע רכב
12. סוג גיר
13. סוג דלק
14. הנעה
15. קוד רכב
16. מאגר מידע (מקור הנתונים)
17. פירוט התקלות
18. סוג גרר מתאים
19. כתובת מוצא
20. כתובת יעד
21. **שטחים** ← עמודה חדשה (כן/לא)
22. מרחק (ק״מ)
23. זמן נסיעה
24. איש קשר מוצא
25. טלפון מוצא
26. איש קשר יעד
27. טלפון יעד
28. הערות
29. **מחיר** ← עמודה חדשה
30. **סוג תשלום** ← עמודה חדשה (מזומן/אשראי/חשבונית)
31. **טלפון בעל הכרטיס** ← עמודה חדשה

**פונקציות חדשות ב-Google Apps Script:**

function getPaymentType(paymentType) {

const types = {

'cash': 'מזומן',

'credit': 'אשראי',

'invoice': 'חשבונית סוף חודש'

};

return types[paymentType] || paymentType;

}

function getAddressText(addressObj) {

// טיפול מתקדם בכתובות - מחזיר רק הכתובת הנקייה

// ללא כפילויות טקסט

}

**שינויים בתיאור האירוע (יומן):**

* ב-createDefectiveDescription(): הוספת חלק "🚛 גרר מתאים" אחרי פירוט התקלה
* ב-createExchangeDescription(): הוספת חלק "🚛 גרר מתאים" אחרי פירוט התקלה של הרכב התקול

**מבנה המידע החדש ביומן:**

🔧 פירוט התקלה:

────────────────────

[רשימת התקלות שנבחרו]

🚛 גרר מתאים:

────────────────────

[רשימת סוגי הגרר שנבחרו]

**רשימת תקלות וסוגי גרר נתמכים**

**תקלות זמינות:**

1. אין חשמל
2. פנצ'ר
3. תאונה
4. גלגל עקום או שבור
5. נזילת מים/שמן
6. לא נדלק/לא מניע
7. חניון תת קרקעי
8. מניע/נדלק ונוסע
9. אחר - מילוי ידני

**סוגי גרר זמינים:**

1. מובילית
2. רמ-סע
3. משקפיים
4. דולי

**תיקיית js/utils/**

* **formatters.js, validators.js, helpers.js** - פונקציות עזר שנשארו ללא שינוי מהמערכת המקורית.

**תיקיית js/config/**

* **constants.js** - נשאר ללא שינוי מהמערכת המקורית.

**סדר הטעינה המומלץ:**

1. constants.js - קבועים בסיסיים
2. formatters.js, validators.js, helpers.js - כלי עזר
3. כל המודולים (ApiManager.js, AddressManager.js, VehicleManager.js, PricingManager.js, FormManager.js)
4. calendar-integration.js - פונקציות איסוף נתונים (תלוי בכל האחרים)
5. main.js - חיבור הכל יחד

**הערות חשובות מעודכנות:**

**שינויים חדשים במערכת:**

* **נוספה תמיכה מלאה בשטחים** - כולל חישוב מחיר (+25%) ושמירה בגיליון
* **נוספו נתוני תשלום מורחבים** - מחיר, סוג תשלום וטלפון בעל כרטיס
* **שופרה מערכת הכתובות** - ללא כפילויות טקסט בגיליון
* **הוסרו מזהי יומן** - מהגיליון לפי בקשה

**מאפיינים קיימים:**

* המערכת מורכבת יותר ממה שהמדריך המקורי תיאר
* נוספו מודלים אינטראקטיביים לבחירת תקלות וסוגי גרר
* יש שימוש מתקדם ב-data attributes לשמירת מידע זמני
* המערכת תומכת בכתובות מורכבות עם שמירת טקסט מקורי ופיזי
* ה-PricingManager הוא מחלקה מלאה עם state management
* יש אינטגרציה מתקדמת עם Google Maps לחישוב מרחקים
* המערכת מנהלת רכבים מרובים עם שיתוף נתונים ביניהם
* המערכת כוללת מודלים רספונסיביים עם עיצוב מותאם למובייל וטאבלט
* כל מודול משתמש ב-window object לחשיפת API ציבורי
* המערכת משתלבת בצורה חלקה עם מערכת איסוף הנתונים והיומן הקיימת

**מצב פיתוח וייצור:**

* **מצב פיתוח:** שמירה מקומית בתיקיית mock-data
* **מצב ייצור:** שליחה לגוגל Apps Script (יומן + גיליון)
* ניתן להחליף בין מצבים באמצעות משתני .env
* מערכת לוגים מפורטת לזיהוי בעיות