Sara Royde 2022

RENT**ALL**

תודה

\* לבורא עולם, החונן לאדם דעת, שממנו כל הכוחות, כל הכלים וכל היכולות.

\* לגב' צביה נוביק, רכזת המסלול, על דאגה לרמה לימודית ומקצועית גבוהה ולתכנים מתקדמים ועדכניים.

\* לגב' צביה קראוס על הליווי וההכוונה במשך הכנת הפרויקט.

\* לכל האנשים הטובים שבדרך שתמכו, עודדו והשקיעו מזמנם ומרצם כדי לתת לנו את המקסימום.

תוכן

הגדרת דרישות ותאור כללי ............................................................. 14

ממשקים חיצוניים ......................................................................... 16

ממשק אדם – מכונה ..................................................................... 17

מבנה נתונים וארגון קבצים ............................................................ 23

תכנון .......................................................................................... 25

מה הקנה הפרויקט ....................................................................... 28

ביבליוגרפיה ................................................................................. 28

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

תאריך:28/04/2021

לכבוד

יחידת הפרויקטים

מה"ט

**הצעה לפרויקט גמר**

\* יש להדפיס את כל הנתונים הנדרשים

1. **פרטי הסטודנטים**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| תאריך סיום  הלימודים | טלפון נייד | כתובת | ת.ז .9 ספרות | שם הסטודנט |
| /7/202117 | 0583270119 | מרומי שדה 36 מודיעין עילית | 323774224 | שרה רויד |

שם המכללה סמינר הליכות בית יעקב סמל המכללה:\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

מסלול ההכשרה: הנדסאים

מגמת לימוד: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

מקום ביצוע הפרויקט:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **פרטי המנחה האישי**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מקום עבודה/תפקיד | תואר | טלפון נייד | כתובת | שם המנחה \* |
| הליכות בית יעקב | B.A מדעי המחשב | 0527676885 | מסילת יוסף 40  מודיעין עילית | נוביק צביה |

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

\_\_\_\_Sara\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

[Type here]

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

1. **שם הפרויקט**

RENTALL

1. **רקע** 
   1. תיאור ורקע כללי

פלטפורמה דיגיטלית המתווכת בין פריטים להשכרה לבין שוכרים פוטנציאליים. המערכת מאפשרת הצעת פריטים להשכרה, בדגש על פרמטרים ייחודיים לפריטים להשכרה, ברמה מותאמת אישית, לבין לקוחות המעוניינים לשכור את הפריט המתאים ביותר בעבורם, ממגוון פרמטרים, ע"י חיפוש פריטים חכם במערכת.

* 1. מטרות המערכת  
     פיתוח אפליקציית אינטרנט המאפשרת למשכיר להציע פריטים העומדים לשכירות, ולשוכר פוטנציאלי למצוא את הפריט המתאים ביותר עבורו.

1. **סקירת מצב קיים בשוק, אילו בעיות קימות**   
   כיום קיימים אתרי תיווך רבים בין מוכר לקונה, בקטגוריות שונות. בין הידועים היא יד2. אבל בפועל אין התייחסות מספקת לפעילות של שכירות, שבה קיימים פרמטרים מיוחדים כמו מחיר ליום, לחודש וכו'. אדם המעוניין להציע פריט להשכרה צריך ליצור לעצמו אתר המציע את שרותיו, או להציג את הפריט שלו כפוסט פרסומת, מה שמקשה על מנועי החיפוש למצוא את פרטי הפריט שהוא מעוניין להשכיר. גם לשוכר פוטנציאלי אין מעטפת נוחה לחיפוש פריט לשכירות, לפי פרמטרים הרלוונטיים בד"כ לחיפוש פריט לשכירות.
2. **מה הפרויקט אמור לחדש או לשפר**   
   הפרויקט בא לתת פלטפורמה פשוטה, קלה ונגישה להצעת פריטים לשכירות באופן מותאם אישית לכל פריט, וכן לשכירת פריטים על ידי חיפוש חכם ונוח למשתמש.

1. **דרישות מערכת ופונקציונאליות** 
   1. דרישות מערכת

סביבת הטמעה ושימוש. שרידות, ביצועים\התמודדות עם עומסים.  
המערכת תפותח בסביבת web על גבי .net כאשר שפת פיתוח צד השרת הוא c# , השרת יהיה שרת iis של Microsoft שעליו רץ asp .

הגישה למערכת ע"י דפדפן סטנדרטי.

* 1. דרישות פונקציונאליות

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

רשימת דרישות המשתמש מהמערכת הם:

* שוכר
  + חיפוש פריט נצרך לפי פרמטרים רווחים כמו מקום, מחיר ליום/ לשבוע/ לטווח זמן, שם פריט, קטגוריה וכן לפי טקסט חופשי
* משכיר
  + הצעת פריט למכירה באופן נוח , נגיש וברור
  + הפריטים יוצגו למשתמש באופן ברור, עם כל הפרטים הנצרכים לו
  + המשתמש יוכל למצוא את הפריט הנוכחי כאשר יתאים לפרמטרים.

1. **בעיות צפויות במהלך הפיתוח ופתרונות (תפעוליות, טכנולוגיות, עומס ועוד):** 
   1. תיאור הבעיות- הללו כפועל יוצא של דרישות המשתמש מהתוכנה.  
      1. עקב ריבוי סוגי ההתקנים שדרכם המשתמשים ישתמשו באפליקציה, יגרם מצב בו נראות האתר יהיה בגודל ספציפי כאשר לבעלי מסכים גדולים התצוגה תהיה קטנה מדי ולבעלי מסכים קטנים התצוגה לא תיכנס במלואה למכשיר.

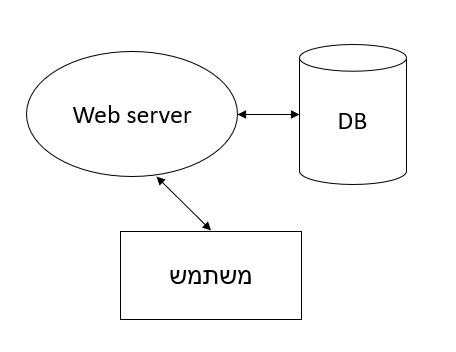
2. ריבוי פריטים במערכת יגרום להאטת פעילות החיפוש והסינון

* 1. פתרונות אפשריים. (נא ציין פתרונות אפשריים וחלופות ארכיטקטוניות)   
     1. פיתוח רספונסיבי של הקוד כך שיותאם לפי גודל מסך.

2. ניתן להגדיל את זיכרון הRAM ואת זיכרון הHARD DISC בכל שלב כך שבמקרה ויהיו עומסים תהיה אפשרות להרחיב את הזיכרון. ניתן לשקול שימוש בשרתי CDN , פעולה המפזרת את העומס מהשרת הנוכחי לשרתים רבים.

1. **פתרון טכנולוגי נבחר:** 
   1. טופולוגית הפתרון- כלומר: פרישת המערכת , היכן יתבצע יישום המערכת) deployment(, מרכיבי הפרישה .הנ"ל ברמת מערכת) לדוג' פרויקט פיתוח אתר אינטרנט:

המערכת מורכבת ממשק משתמש בצד הלקוח על סביבת .NET core ומסד נתונים לשמירת מידע מסוג SQL server .



* 1. טכנולוגיות בשימוש.(איזה ומדוע בכמה מילים)   
     .NET Core - כיון שהיא גמישה, יכולה לעבור בין מחשבים שונים ומערכות הפעלה שונות. היא 'רזה' (אפשר לצרף לה רק את הספריות הנצרכות)
  2. שפות הפיתוח: (איזה שפות ומדוע בכמה מילים)

C#, JavaScript, html, sql

את הserver בשפת C# כיוון שזו שפה עדכנית עם מגוון כלים בסביבת הפיתוח.

Sql מתממשקת היטב עם C# ולכן בחרתי לעבוד איתה

JavaScript לצד הclient. השתמשתי בספרייה react כיוון שזו ספריה שמשפרת את העבודה עם JS

* 1. תיאור הארכיטקטורה הנבחרת- הסבר בכמה מילים מדוע   
     עבודה תהיה במודל שרת-לקוח כמקובל באפליקציות אינטרנטיות. פעולות שמירה ועדכון הנתונים יהיו בצד השרת, והתצוגה והסינון בצד הלקוח.
  2. חלוקה לתכניות ומודולים   
     מתבסס על מודל שלש השכבות.

1. שכבת האפליקציה (UI)
2. שכבת לוגיקה עסקית.
3. שכבת גישה לנתונים.
   1. סביבת השרת (מקומי, וירטואלי, ענן, שירות אירוח)  
      כרגע על שרת מקומי שלiis ובעתיד יתכן ואעלה את הפרויקט לשרת אירוח
   2. ממשק המשתמש/לקוח - GUI   
      הממשק יהיה ממשק web המבוסס על html ויתווסף לו עיצוב בפורמט css כאשר ההרצות בצד הלקוח יהיו כתובים בjava script (שימוש בReact).
   3. ממשקים למערכות אחרות/ API:   
      כרגע אין, ייתכן ובשביל חיפוש לפי מרחק אוסיף חיבור לGoogle Maps
   4. שימוש בחבילות תוכנה:   
      React  
      axios  
      Entity Framework  
      ספריות עיצוב נוספות.
4. **שימוש במבני נתונים וארגון קבצים** 
   1. נא פרט את מבני הנתונים.

שימוש בבסיס נתונים טבלאי של מיקרוסופט SQL-Server .

הטבלאות המתוכננות הם:

**השכרות** שיכלול את פרטי ההשכרות: שם, פרטים כלליים, מקום, פרטי מנהל, קישור לטבלת פריטים, ועוד פרטים אודות ההשכרה

**פריטים:** שיכלול את פרטי המוצרים להשכרה: שם פריט, קישור לטבלת מחירים לפריט, קישור לטבלת תגיות לפריט,

**תגיות לפריט:** קישור לטבלת פריטים, טקסט לחיפוש חופשי.

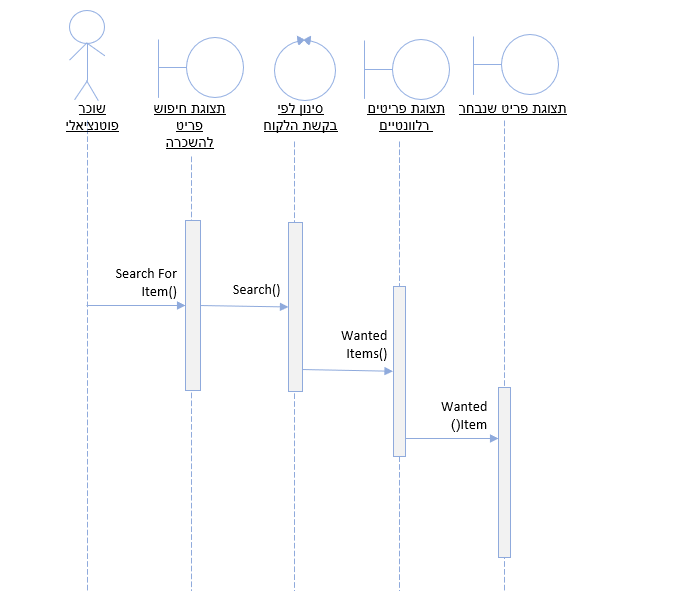
**מחירים לפריט:** מחיר לפי יום/ חודש/ שנה כמות ימים/חודשים/ שנים וכו

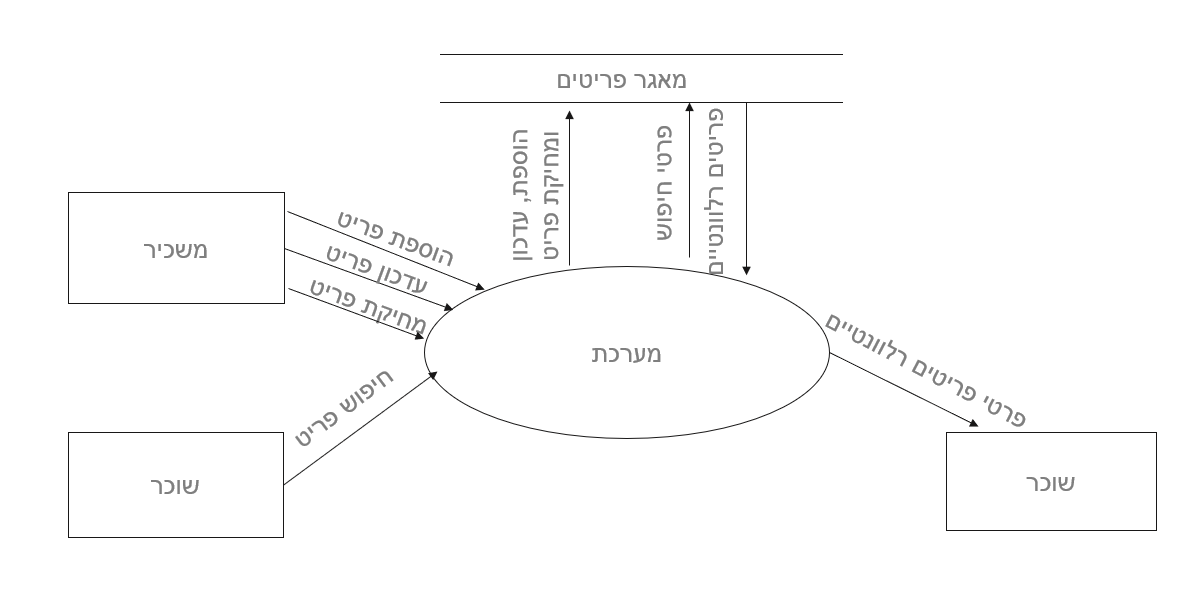
* 1. נא פרט את שיטת האיחסון (מאגר, קבצים, ובאיזה טכנולוגיה).

אחסון בDB בסיסי נתונים טבלאיים עם SQL server של מיקרוסופט.

* 1. נא ציין מנגנוני התאוששות מנפילה\קריסה\תמיכה בטראנזקציות.

יתבצע שימוש במנגנוני השרידות והתמיכה בעומסים המובנים בתוך השרת ה- SQL .

1. **תרשימי מערכת מרכזיים** 
   1. Use Case  
      
   2. Sequence diagram - רצף קריאות פונקציות מרכזיות בלוגיקה העיסקית המרכזית של הפרוייקט   
      
   3. Data flow



1. **תיאור המרכיב האלגוריתמי – חישובי** 
   1. איזה בעיה בא לפתור, איך יפתור?   
      מציאת הפריט המתאים ביותר עבור השוכר, ע"פ הפרמטרים המבוקשים.

האלגוריתם יערוך חיפוש במאגר הפריטים לפי קטגוריות החיפוש של המשתמש או לפי טקסט חופשי, האלגוריתם יזהה טקסט דומה בין החיפוש של המשתמש לבין פרטי הפריט הרלוונטיים, ולפי הגדרות המשכיר.

* 1. איסוף מידע וניתוחים סטטיסטיים (אנליטיקות)   
     סינון ומיון לפי חיפושים שכיחים, חוות דעת משתמשים על המערכת

1. **תיאור/התייחסות לנושאי אבטחת מידע**

נא לציין אזורים הדורשים אבטחה, כגון: שרת, בקרת גישה לאתר, חשבונות משתמשים, מאגרי מידע וכיצד ניתן מענה.

נא ציין מס' מקרים ותגובות להם ניתן מענה אבטחתי.   
שכבה ראשונה היא חומת אש של השרת.  
השכבה הנוספת היא תוכנת הגנה המותקן על גבי שרת האינטרנט המקומי  
בנוסף, יתבצע שימוש במנגנוני האבטחה הבאים:  
1. שימוש בפרוטוקול https לצורך הצפנה של הנתונים בתקשורת בין ה-client ל-server   
2. ביצוע בקרת גישה לממשק הניהול המבוססת כתובות ip ייעודיות שרק להן תתאפשר גישה לממשקים אלו  
3. בקרת גישה מבוססת שם משתמש וסיסמה לממשקי המשתמש האישיים.

1. **משאבים הנדרשים לפרויקט:** 
   1. מספר שעות המוקדש לפרויקט, חלוקת עבודה בין חברי הצוות   
       כ-350 שעות.
   2. ציוד נדרש   
      מחשב עם מעבד i5 .8 ג'יגה בייט זיכרון הכולל חיבור לאינטרנט.
   3. תוכנות נדרשות   
      visual studio מגרסת 2017 ומעלה  
      Microsoft sql server  
      .net core מגרסה 2.1 ומעלה
   4. ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט   
      שליטה היקפית בc#

עבודה עם שרתי אינטרנט והכרת פרוטוקולים והאובייקטים בצד שרת.

למידה מעמיקה של ספרית axios

שליטה היקפית בשפת JS ובספריית REACT.

* 1. ספרות ומקורות מידע   
     React

C# document

Npm

Stackoverflow

1. **תכנית עבודה ושלבים למימוש הפרויקט**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 03/  2021 | 04/  2021 | 05/  2021 | 06/  2021 | 07/  2021 | 08/  2021 | 09/  2021 | 10/  2021 | 11/  2021 | 12/  2021 |
| \* | \*\*\* |  |  |  |  |  |  |  |  |  |
| ניתוח ואפיון |  | \*\*\* | \*\*\* |  |  |  |  |  |  |  |
| הגדרת סביבת הפיתוח |  | \*\*\* | \*\*\* |  |  |  |  |  |  |  |
| פיתוח מודליים מרכזיים |  | \*\*\* | \*\*\* | \*\*\* | \*\*\* | \*\*\* |  |  |  |  |
| בדיקות וניפוי שגיאות |  |  | \*\*\* | \*\*\* | \*\*\* | \*\*\* | \*\*\* |  |  |  |
| התקנת המערכת בסביבת העבודה |  |  |  |  |  |  |  | \*\*\* | \*\*\* |  |
| תיעוד והגשה |  |  |  |  |  |  |  |  |  | \*\*\* |

1. **תכנון הבדיקות שיבוצעו** 
   1. נא פרט בטבלה ,בדיקות תהליכיות ברמת משתמש בהן נדרשת המערכת לעמוד

|  |  |  |
| --- | --- | --- |
|  | הוספת פריט | חיפוש פריט |
| בדיקה 1 | אימות סיסמא | האם חיפוש לפי קטגוריות מחזיר ערכים נכונים |
| בדיקה 2 | אימות פרטי יצירת קשר | האם חיפוש לפי טקסט חופשי מחזיר ערכים נכונים |
| בדיקה 3 | בדיקת הוספת שדות מותאמים אישית |  |
| בדיקה 4 | ולידציה על שדות הוספת פריט |  |

.)full Flow(

* 1. נא פרט בטבלה, מס מייצג של בדיקות יחידה למודולים המרכזיים בהן נדרשת המערכת לעמוד) .unit test(

|  |  |  |
| --- | --- | --- |
|  | חיפוש פריט | הוספת פריט |
| בדיקה 1 | נפח נתונים חריג בגודלו | האם יש גישה לדף עדכון הפריט |
| בדיקה 2 | חיפוש עם טקסט שגוי | הוכנסו ערכים שגוים |

1. **בקרת גרסאות (Version control)**נשתמש במערכת git השומרת בכל שנוי גרסה חדשה של המערכת כך שבמידה ותהיה תקלה נוכל לשנמך לגרסה הקודמת

\_\_\_\_\_\_Sara\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימת הסטודנט חתימת המנחה האישי

1. **הערות ראש המגמה במכללה**

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

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

1. **אישור ראש המגמה**

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

1. **הערות הגורם המקצועי מטעם מה"ט**

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

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

1. **אישור הגורם המקצועי מטעם מה"ט**

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

1. הגדרות דרישות ותאור כללי

1.1 תאור כללי

פלטפורמה דיגיטלית המתווכת בין פריטים להשכרה לבין שוכרים פוטנציאליים. המערכת מאפשרת הצעת פריטים להשכרה, בדגש על פרמטרים ייחודיים לפריטים להשכרה, ברמה מותאמת אישית, לבין לקוחות המעוניינים לשכור את הפריט המתאים ביותר בעבורם, ממגוון פרמטרים, ע"י חיפוש פריטים חכם במערכת.

1.1.1 מטרות האפליקציה

פיתוח אפליקציית אינטרנט המאפשרת למשכיר להציע פריטים העומדים לשכירות, ולשוכר פוטנציאלי למצוא את הפריט המתאים ביותר עבורו.

1.1.2 היקף הפרויקט

כ- 350 שעות

1.1.3 מבנה וארגון

האתר מורכב מ:

DB: שמירת הנתונים ע"י שימוש במבנה נתונים טבלאי של Microsoft SQL-server.

SERVER: גישה לשרת באמצעות API (Application Programming Interface)

CLIENT: מסכי המערכת, שימוש בספרית React

1.1.4 משימות המערכת

שוכר:

חיפוש פריט נצרך לפי פרמטרים רווחים כמו מקום, מחיר ליום/ לשבוע/ לטווח זמן, שם פריט, קטגוריה וכן לפי טקסט חופשי – חיפוש חכם

משכיר:

הצעת פריט למכירה באופן נוח , נגיש וברור

הפריטים יוצגו למשתמש באופן ברור, עם כל הפרטים הנצרכים לו

המשתמש יוכל למצוא את הפריט הנוכחי כאשר יתאים לפרמטרים.

1.2 תאור חומרה

מחשב אישי- PC, Mobile או Tablet, עם גישה לדפדפן של Google Chrome.

1.3 תאור תוכנת המערכת

1.3.1 כללי

DB: מסד הנתונים SQL Server

:SERVERהשרת מנוהל ע"י Web API והגישה למסד הנתונים נעשית באמצעות Entity Framework – DB First. השליפות נעשות על ידי שאילתות LINQ.

CLIENT: צד הלקוח נכתב ב-JavaScript, עם שימוש בספריית React. קריאות שרת מנוהלות על ידי axios.

1.3.2 תאור ה- SYSTEM

Visual Studio, Visual Studio Code, SQL Server

1.3.3 כלי התוכנה לפיתוח המערכת

HTML5, CSS, JavaScript, React, C#

1.4 פונקציות המערכת

שוכר:

חיפוש פריט במערכת לפי פרמטרים

חיפוש פריט במערכת לפי חיפוש חכם

משכיר:

רישום למערכת

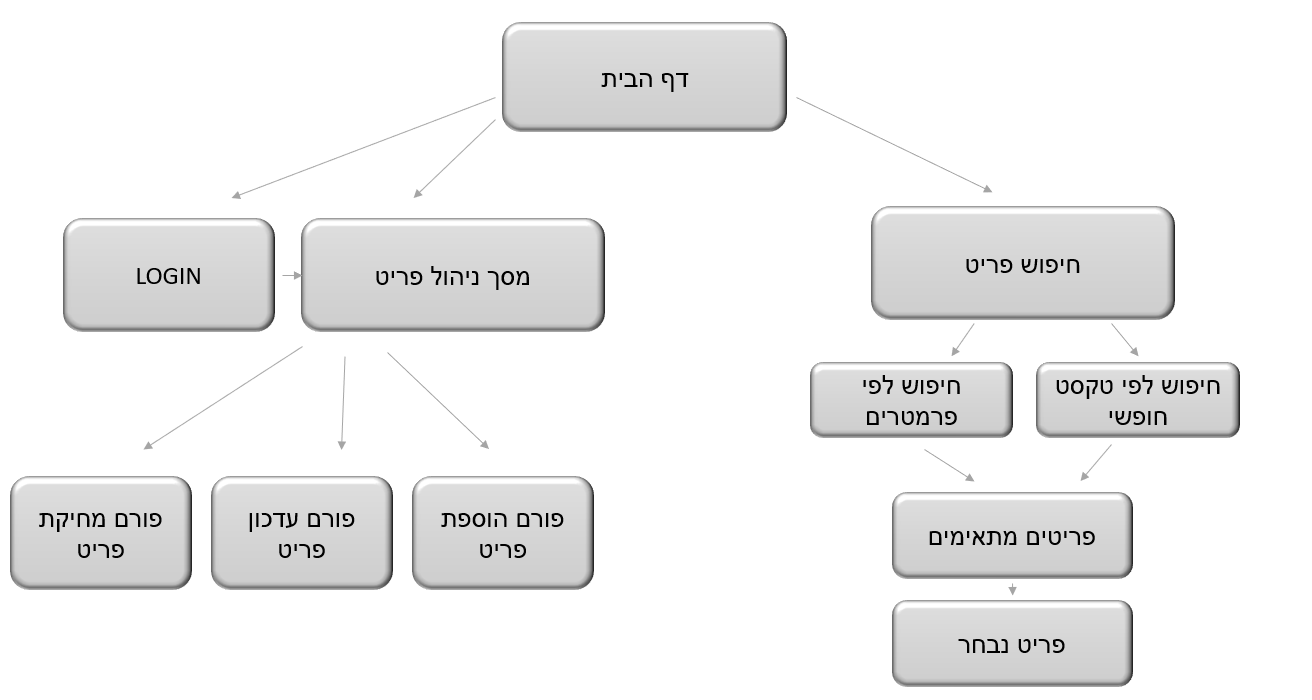
הצעת פריט להשכרה

עדכון פרטי הפריט

1.5 זרימת המידע במערכת

זרימת מידע דו כיוונית. מידע מגיע מהשרת ללקוח (כמו פרטי פריט) ומהלקוח לשרת (לדוג' עדכון פריט)

תרשים זרימת המידע:



2. ממשקים חיצוניים

התממשקות לAPI של Google Maps , בכדי לתת לשוכר אופציה של חיפוש פריט שנמצא במרחק סביר, לפי הגדרותיו.

פעולת ההתממשקות דרשה למידת האופציות שהAPI מקנה למְפַתח, השגת מַפְתח דרכו ניתן להתחבר לAPI, וכתיבת פונקציה להתחברות לAPI של Google Maps.

המערכת שומרת עבור כל פריט את מיקומו, וכשהשוכר הפוטנציאלי מחפש פריט לפי מיקום, המערכת מחזירה לו את הפריטים שנמצאים בטווח המתאים, ע"פ תוצאת חישוב המרחק ע"י Google Maps.

3. ממשק אדם- מכונה (GUI)

3.1 כללי

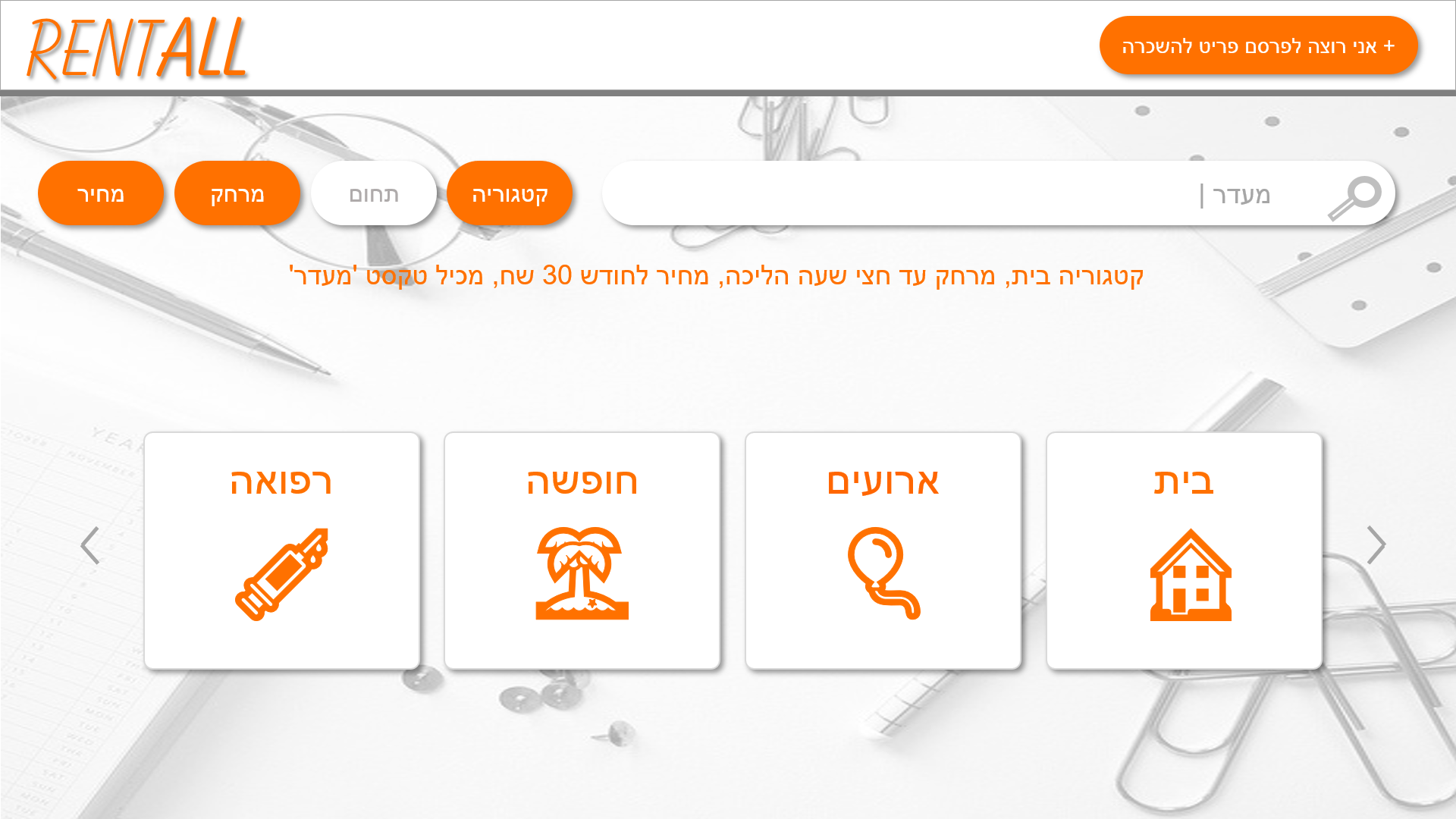
בעת בניית ממשק המשתמש הושם דגש על בהירות מקסימלית, וכן על מהירות ופשטות בשימוש. המערכת פשוטה ונוחה לתפעול, והביצועים מהירים וחוסכים בקריאות שרת מיותרות המאטות את פעולתה של המערכת, ע"י ביצוע חלק מהחיפוש בצד הלקוח.

3.2 תאור המסכים

המערכת מנהלת 2 סוגי פעולות: השכרת פריט וחיפוש פריט לשכירות.

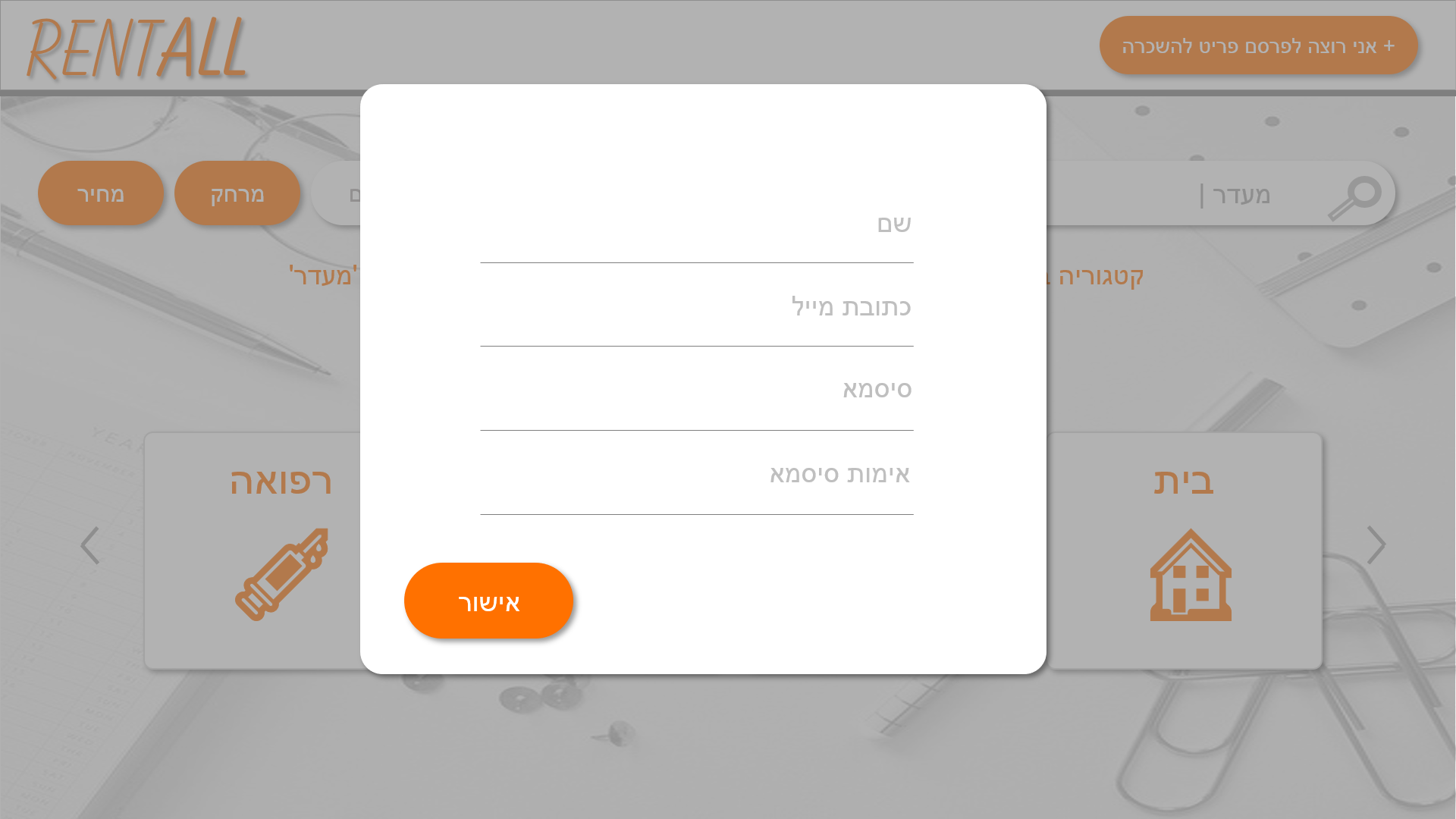
כל הפעולות הקשורות בניהול דורשות רישום למערכת, אך חיפוש במערכת אינו דורש כניסה.

בכניסה למערכת ישנה אפשרות של צפיה בפריטים, חיפוש, וניהול פריטים, אך אם לא בוצעה הרשמה למערכת, המערכת תדרוש Login.



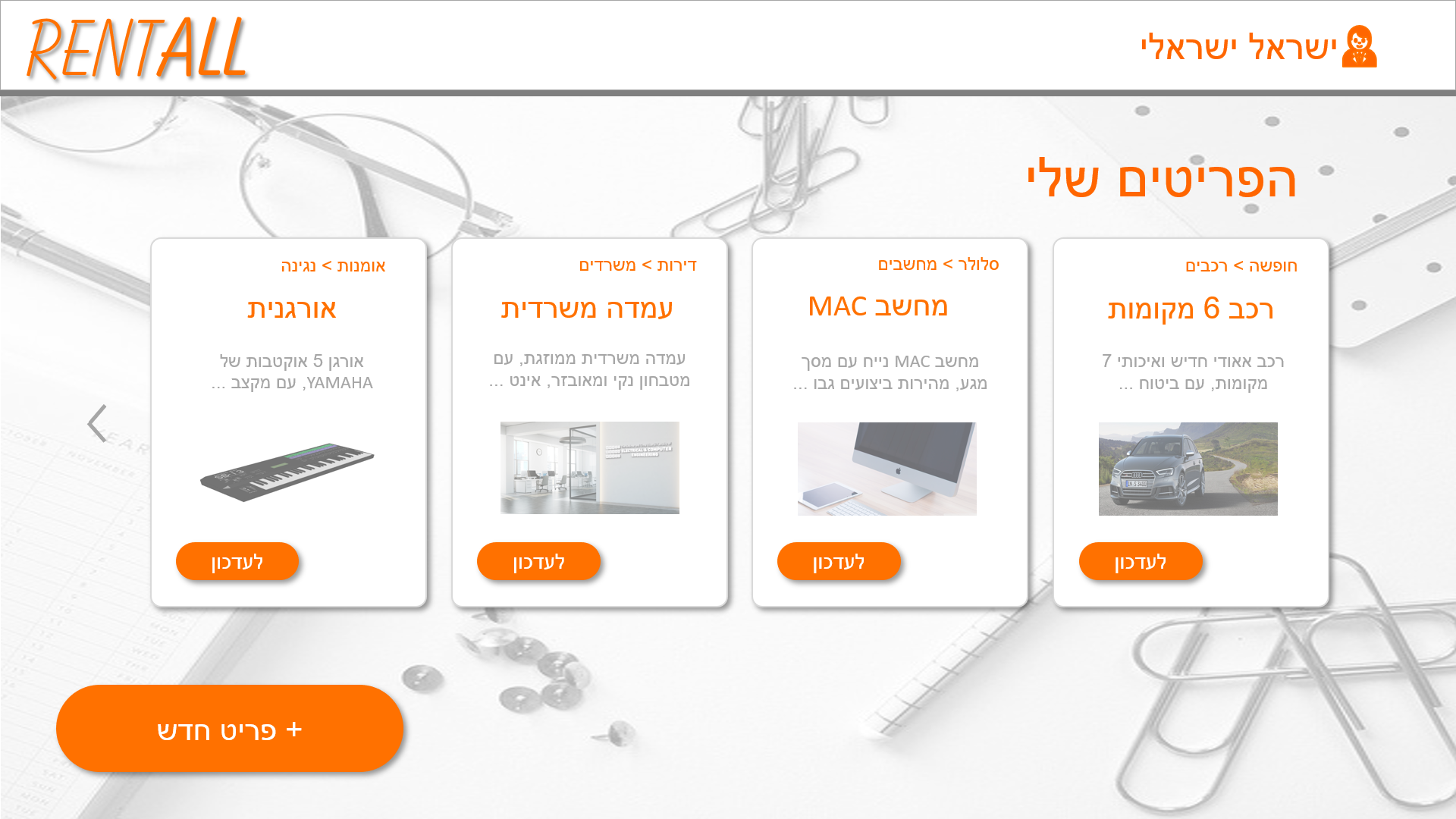
משכיר (משתמש רשום)

מסך Login:



כשהמשתמש רשום הוא יוכל להגיע למסך ניהול פריטים

המסך יציג למשתמש את כל הפריטים שהוא הוסיף בעבר ואקטיביים כרגע, כך שיוכל לעדכן, וכן אופציה של הוספת פריט חדש.

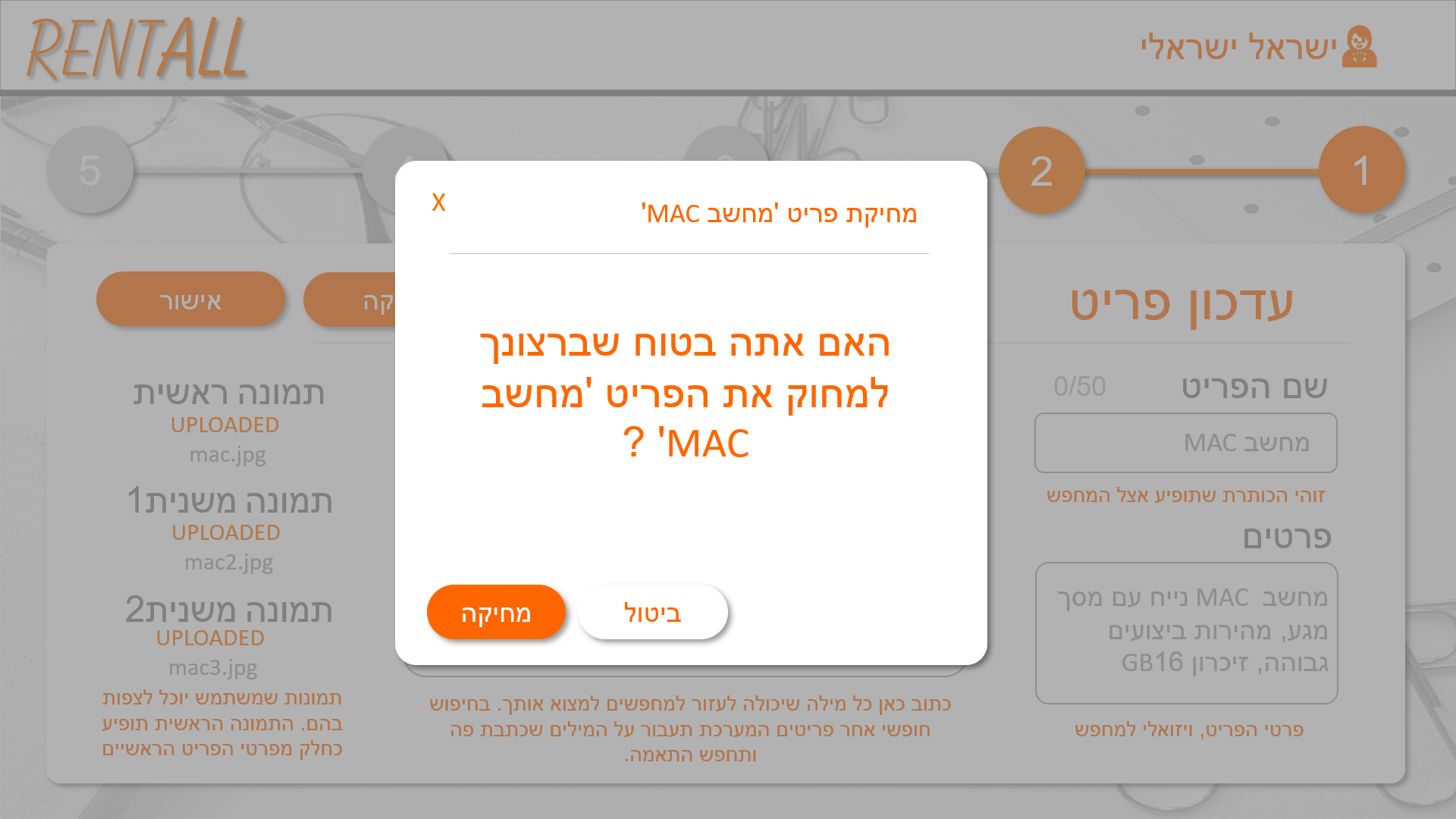


פורם הוספת פריט :

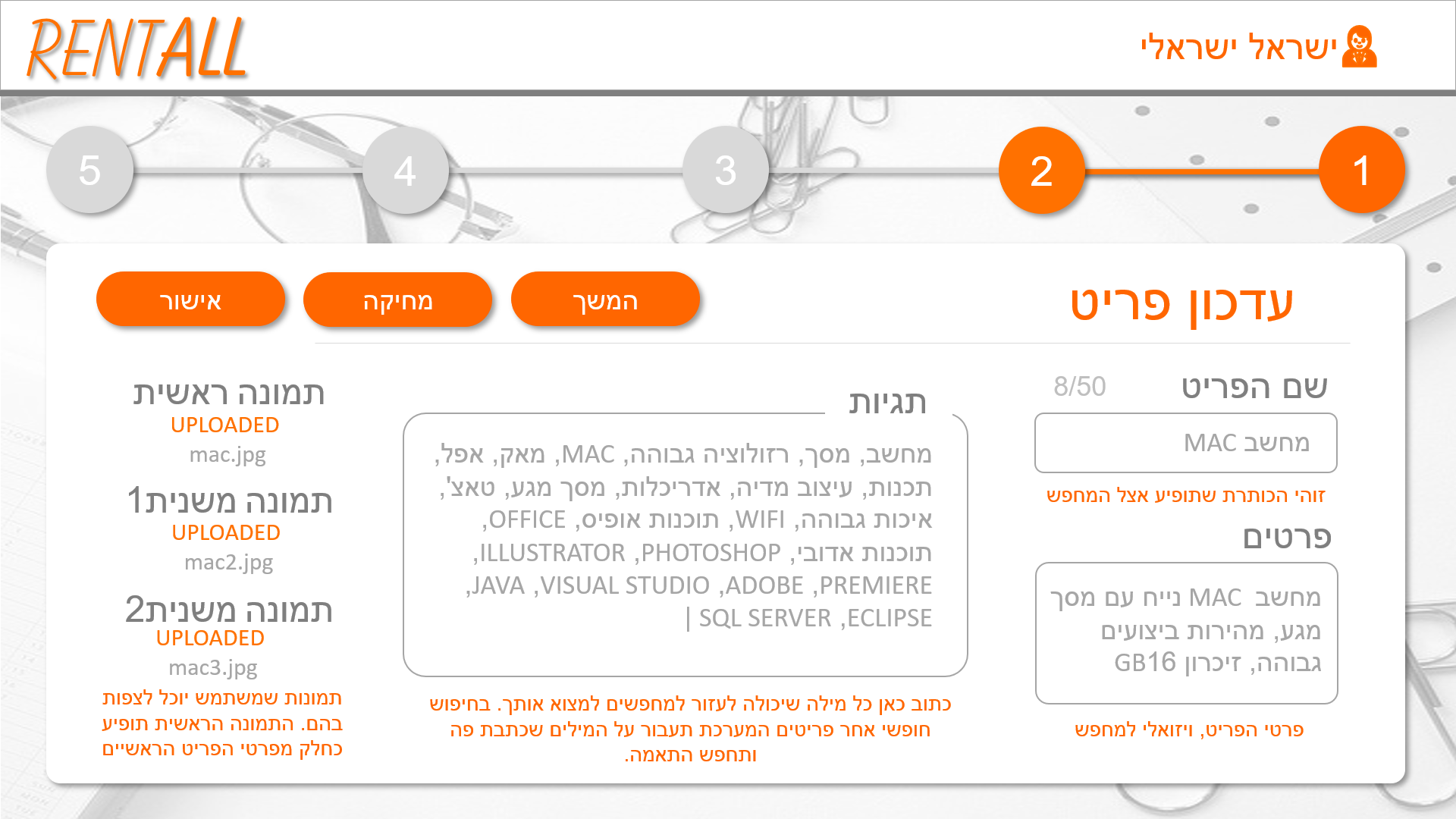


על כל פריט יש אופציות של עדכון ומחיקה.

מחיקת פריט :



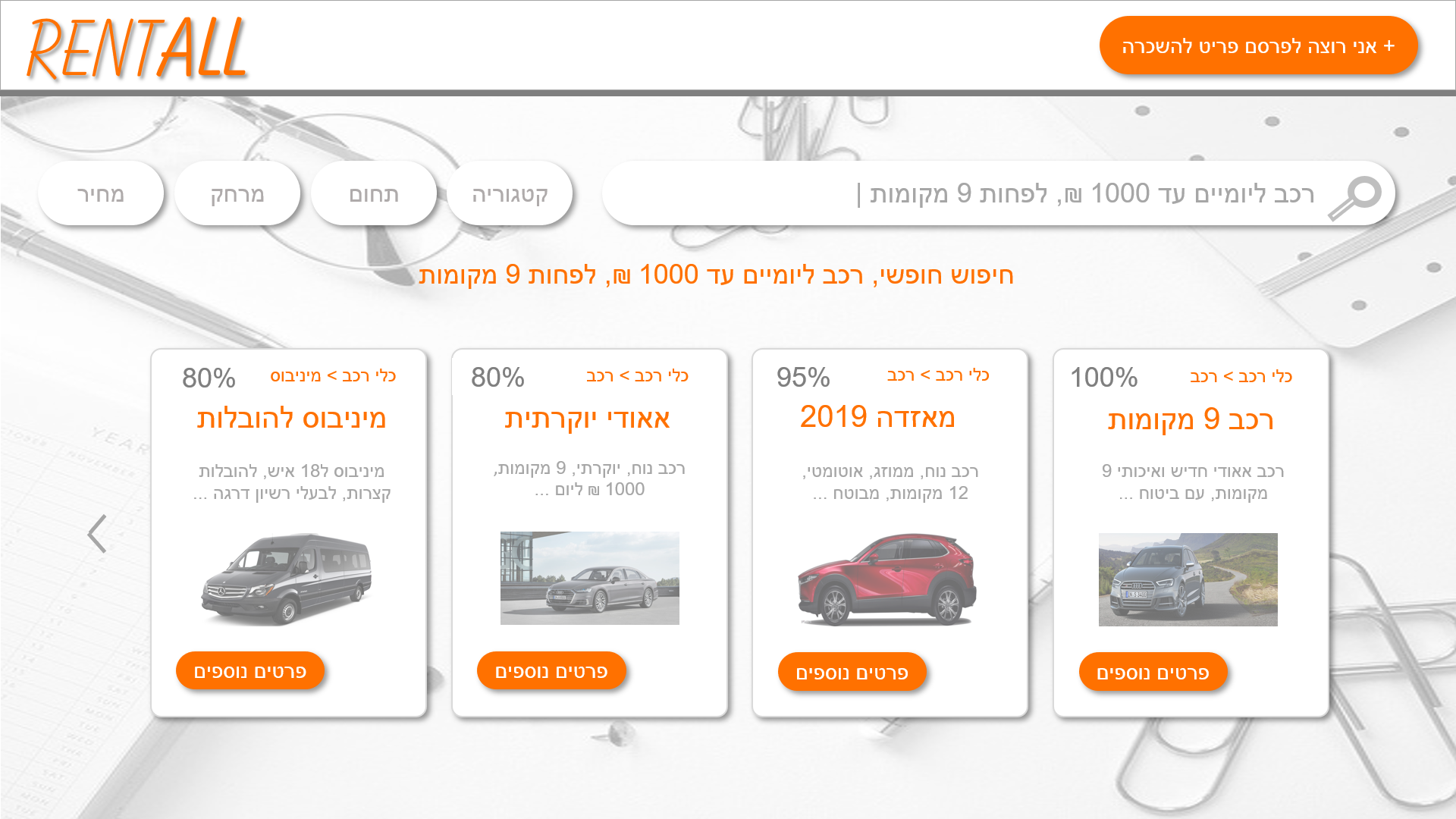
טופס עדכון פריט :



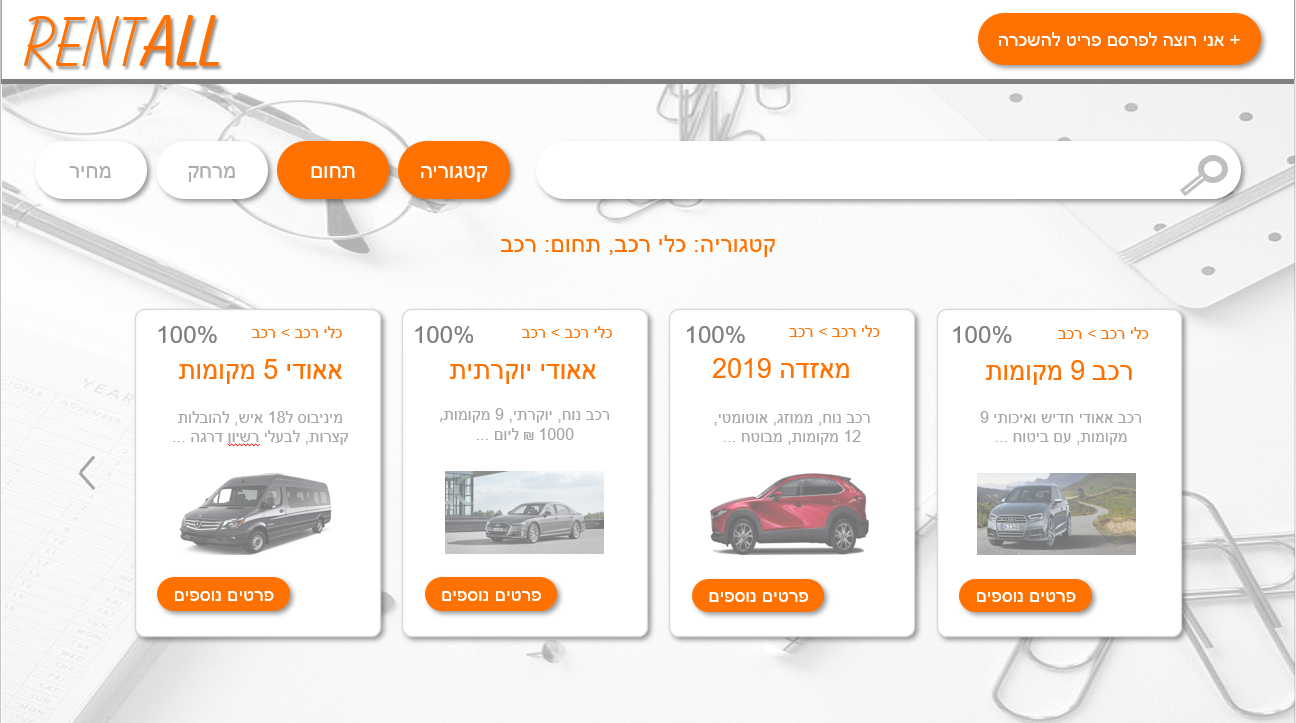
שוכר (חיפוש פריט במערכת)

חיפושים שונים:

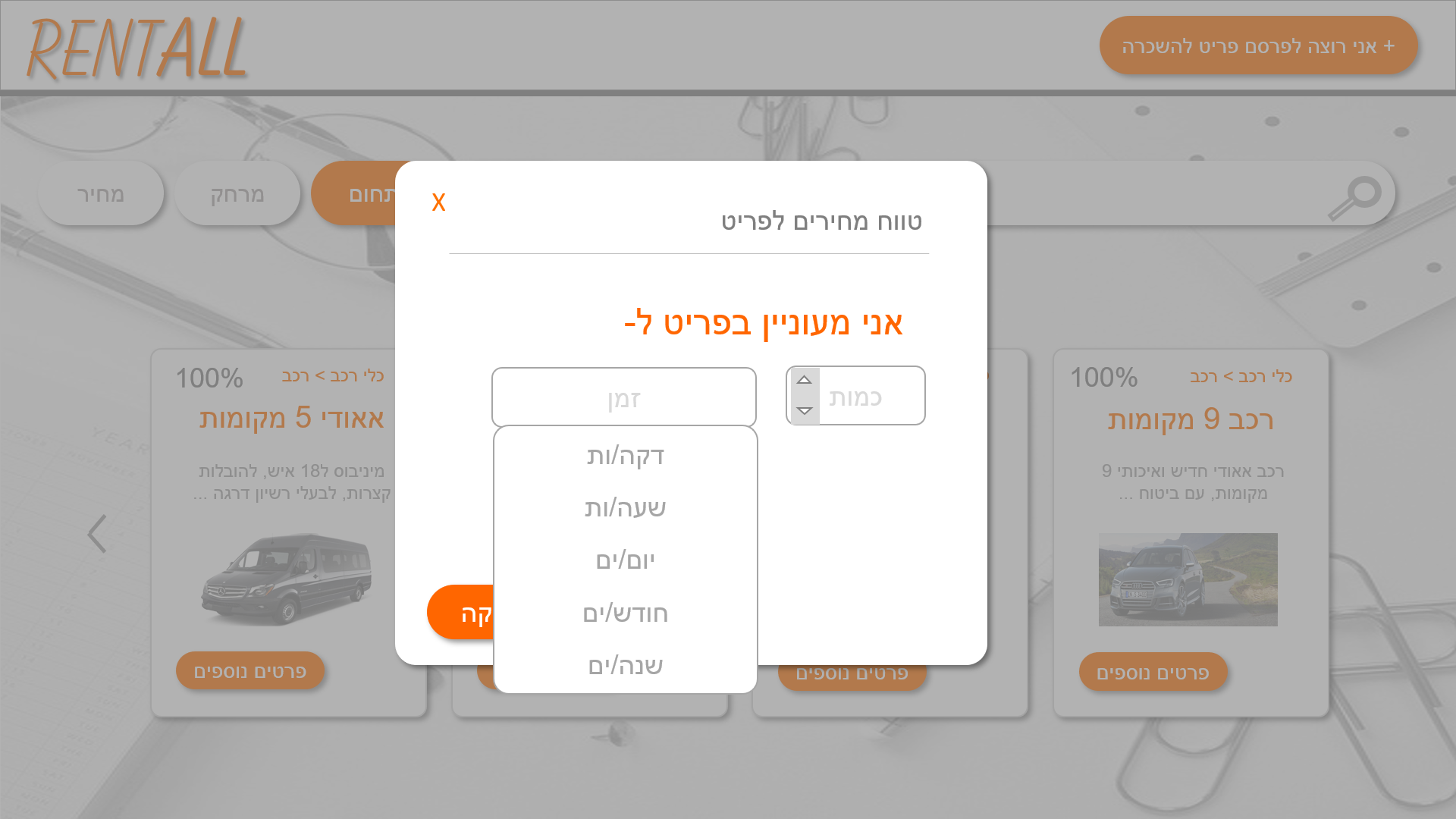
חיפוש לפי טקסט חופשי



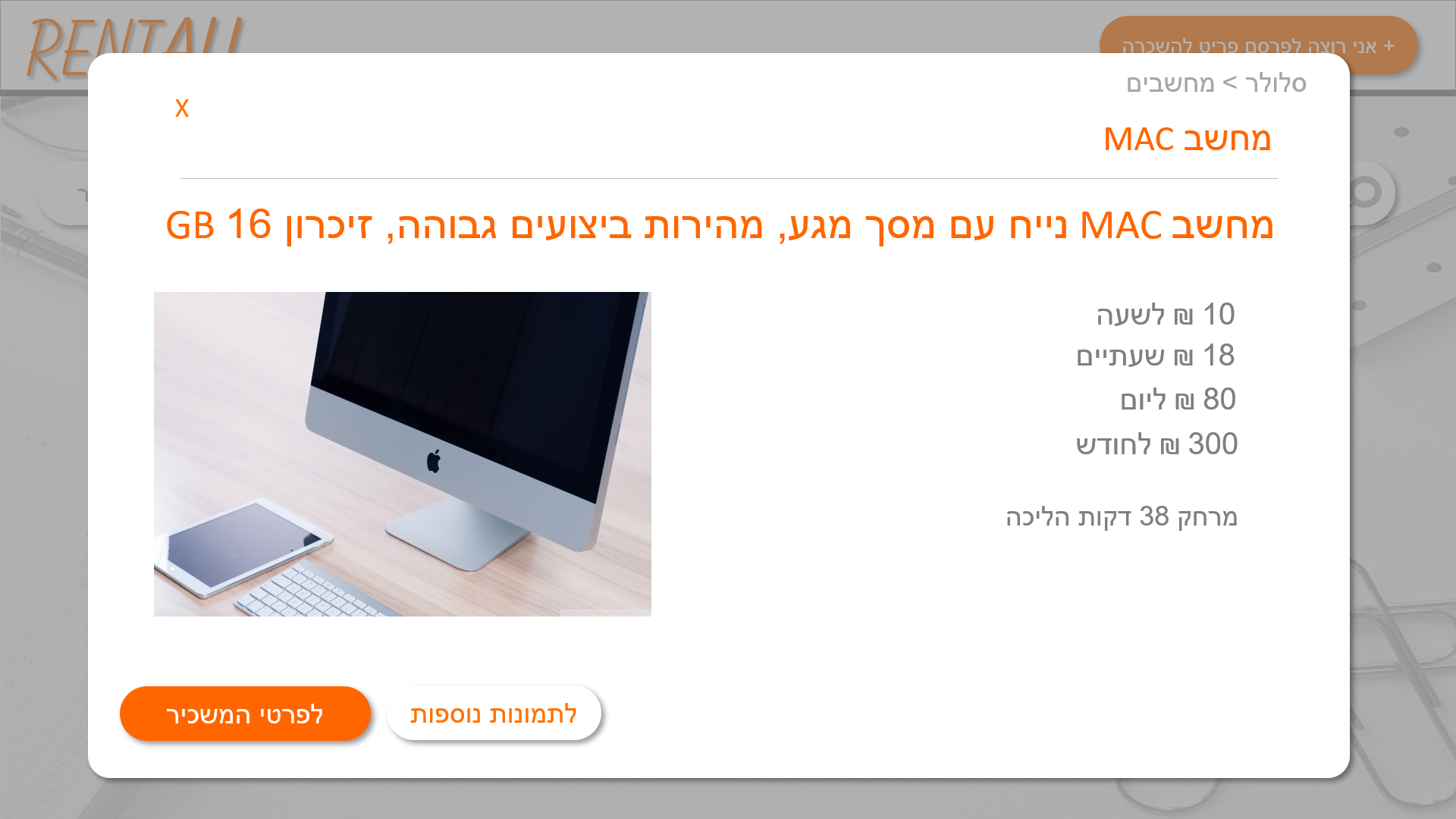
תוצאות חיפוש לפי קטגוריה ותחום



חלון בחירת מחיר (לאחר בחירת קטגוריה ותת קטגוריה)



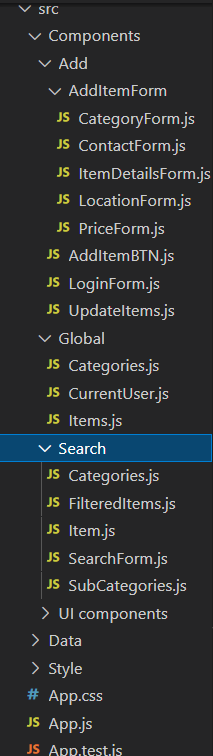
בחירת פריט מסוים



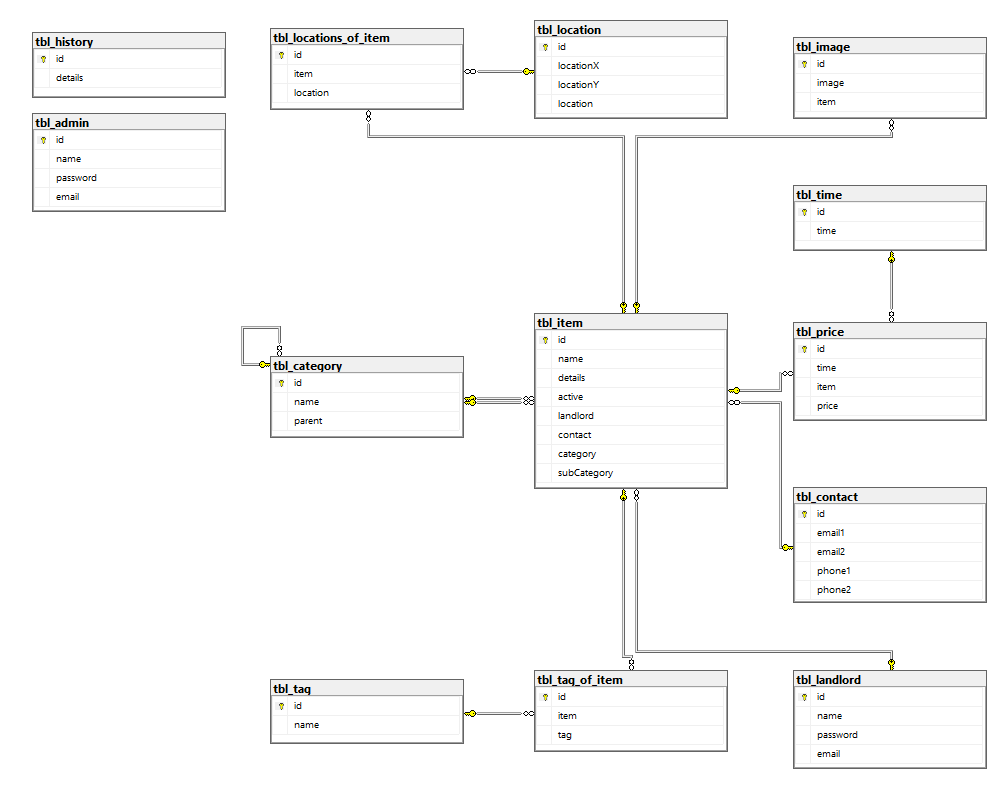
4. מבנה נתונים וארגון הקבצים

4.1 ארגון קבצים

מבנה התיקיות והקבצים בצד CLIENT :



4.2 מבנה נתונים



5. תכנון

5.1 כללי

מאחורי בחירת הארכיטקטורה המתאימה והיעילה ביותר למימוש הפונקציונאליות של פרויקט זה עומדים מספר פרמטרים:

מבחינה מקצועית –

שפת שרת: C# הינה שפת תכנות עילית מרובת פרדיגמות, מונחית עצמים בעיקרה, המשלבת רעיונות כמו טיפוסיות חזקה, אימפרטיביות, הצהרתיות, פונקציונליות, פרוצדורליות וגנריות. #C הינה שפת תכנות הנמצאת בשימוש נפוץ, ולכן קהילת המתכנתים שלה רחבה במיוחד, מה שנותן מענה לשאלות או בעיות, אם יעלו.

ארכיטקטורת שרת: הפרויקט נבנה בגישה של מודל השכבות. בארכיטקטורה זו קיימת הפרדה בין השכבות השונות בפרויקט. (Tiers-N)

היתרון העיקרי בחלוקה לשכבות הוא תחזוקה קלה, במיוחד במערכות גדולות. כאשר ייווצר צורך לערוך שינויים בתוכנית ניתן יחסית בקלות לעדכן כל שכבה בלי לגרום לשינוי בשכבות האחרות.

השכבות מחולקות לשכבת הUI, שכבת הBL ושכבת הDAL.

שכבת הUI פונה דרך הWeb API לפונקציונאליות המתאימה בשכבת הBL (business layer).שכבת הBL פונה לשכבת הDAL(Data Layer) שמתחברת למסד הנתונים ומחזירה את הנתונים לשכבת הBL, או מעדכנת את הDB, על פי הדרישה.

לצורך ייעול ההפרדה מחד, וטעינת הנתונים באופן יעיל מה-Data Base מאידך, המערכת משתמשת ב- - EntityFramwork טכנולוגית עבודה מתקדמת של Microsoft.

EntityFramwork מספק קוד שנוצר אוטומטית, מקטין את זמן הפיתוח ומוזיל את עלותו. כמו כן מאפשרת למפות את בסיס הנתונים ולעשות השמה ישירה ואוטומטית לתוך אובייקטים בקוד.

השימוש ב-EntityFramwork נעשה באמצעות הגדרת קובץ מודל להגדרת המחולל classes לעבודה מול ה- Data Base ,כך שנוצרת ישות לכל טבלה. הישות הנוצרת מכילה אתחולים, מאפיינים וקשרים עבור הטבלה הספציפית. הודות לישויות הללו שנוצרו על ידי קובץ המודל ניתן להגדיר מופע מסוג הטבלה .

כמו כן ניתן גם לייבא פונקציות ממסד הנתונים לשכבת הDAL ולהשתמש בהן במחלקות השונות.

שפת לקוח: הפרויקט משתמש בספריה REACT. REACT זוהי ספריית הקליינט החזקה בשוק, ובעלת שימוש נרחב. מיתרונותיה הבולטים הם חיסכון במשאבים, מהירות ביצוע, מניעת חזרה על קוד, נוחות ופשטות השימוש, רוב העבודה מתבצעת במד הלקוח ולא בצד השרת, מה שמפחית תקורה. קהילת המשתמשים בReact רחבה במיוחד.

מבחינת חווית המשתמש-

על מנת למקסם את הנוחות ובהירות למשתמשי קצה, האתר מעוצב באופן נקי ומאורגן, ע"י שימוש ב- Material UI. ספריה זו היא ספריית קומפוננטות המספקת עיצוב קלאסי ונעים לעין לפקדי המערכת. בנוסף הסבר ברור ועם זאת מינימלי, לשימוש נוח ופשוט ביותר.

5.2 עקרונות התכנות

הפרויקט נבנה בגישת OOP (Object Oriented Programming) תכנות מונחה עצמים.

זוהי פרדיגמת תכנות המשתמשת ב'עצמים' אובייקטים לשם תכנון תכניות מחשב ויישומים.

הפרדיגמה מספקת מספר כלי הפשטה וטכניקות ובהן הורשה, מודולריות, פולימורפיזם וכימוס.

בנוסף שימוש במודול השכבות, חלוקה בין שכבת הגישה לנתונים לשכבת הלוגיקה, ובין שכב הלוגיקה לבין שכבת הלקוח.

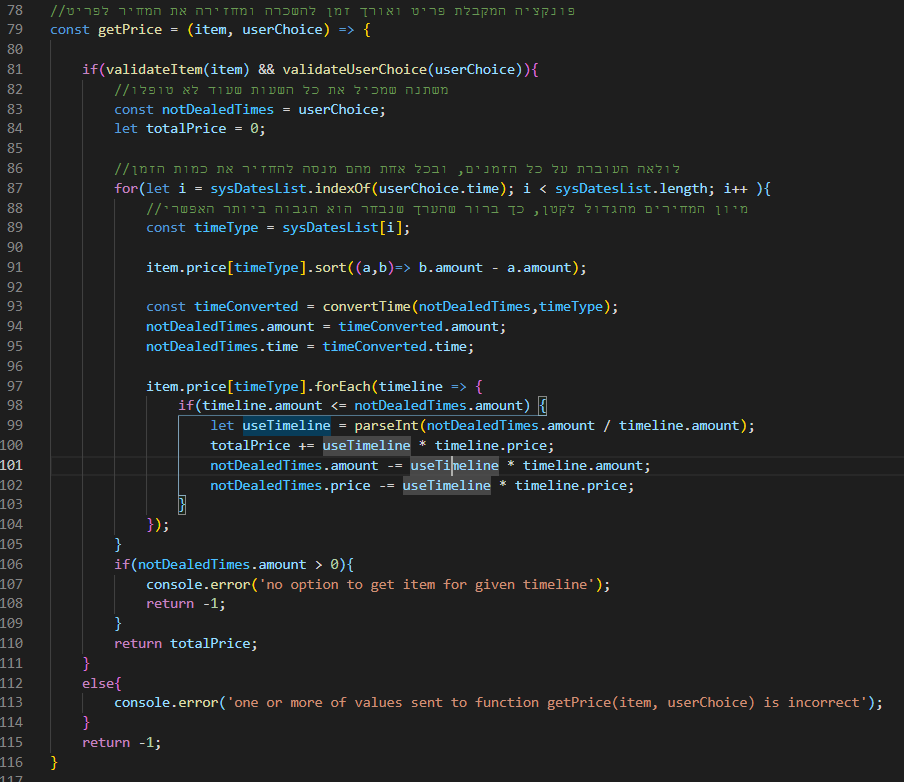
5.3 תאור האלגוריתמים

אלגוריתם החיפוש עובד בעיקרון על צד הלקוח ולא בצד השרת כדי לחסוך עלויות. אמנם ישנם חיפושים ראשוניים שקיימים כבר בצד השרת, המונעים העברה מיותרת של פריטים לא רצויים.

אלגוריתם החיפוש עובד באופן ממודר מאוד. כל סוג חיפוש בפונקציה אחרת ובלוגיקה אחרת, בכדי לחסוך עלויות של חיפוש לא יעיל לסוגי חיפושים מסוימים.

פונקציית הסינון הראשית שולחת את מערך הפריטים לפונקציות הסינון. לכל פרמטר סינון מותאמת פונקציה שתפקידה לבצע את הסינון באופן היעיל ביותר לפרמטרים הרלוונטיים.

להלן פונקציה המחזירה לכל פריט את מחירו הזול ביותר, בהתייחס לטווח הזמן המבוקש. זוהי תת פונקציה של הפונקציה לסינון פריט לפי מחיר:



הפונקציה משתמשת בפונקציה להמרת זמנים:

5.4 בדיקות מערכת

תכנון המערכת תוך כדי התייחסות למקרי קצה אפשריים, כשכל רכיב נבדק גם ביחס למערכת הכללית, וגם בבדיקות יחידה. אנשי UAT משתמשים נסיונית במערכת ומעירים הערות לשיפור מתוך השטח, ובנוסף CODE REVIEW מעמיק.

6. מה הקנה הפרויקט

\* שליטה היקפית בSQL

\* שליטה היקפית בC#

\* שליטה היקפית בJavaScript

\* היכרות מעמיקה עם ספריית React

\* היכרות מעמיקה עם Material UI, CSS וספריות עיצוב נוספות

7. ביבליוגרפיה

<https://www.javascript.com/>

<https://reactjs.org/>

<https://stackoverflow.com/>

<https://mui.com/>