# 

**פרויקט גמר לתואר הנדסאי מחשבים**

**שם הפרויקט:**

**Small World**

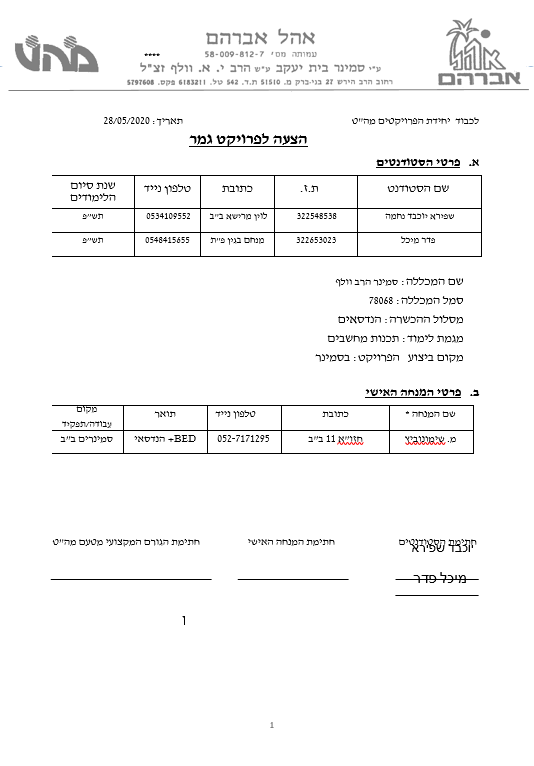
**מוגש ע"י**: יוכבד נחמה שפירא

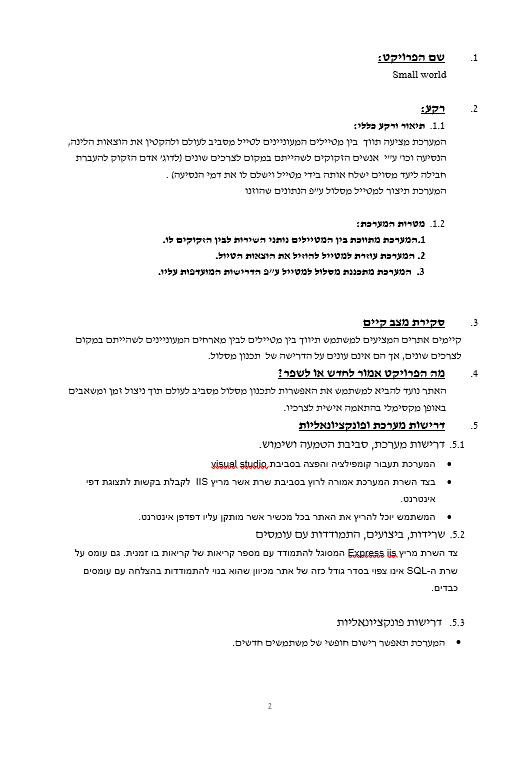
מיכל פדר

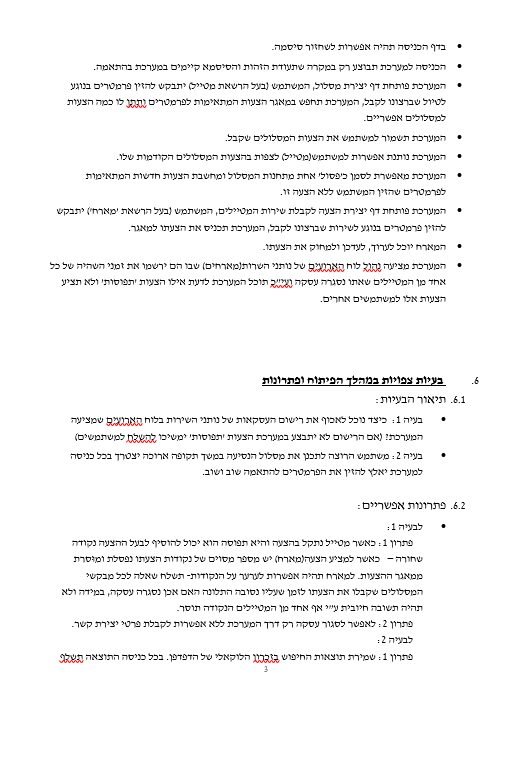
מגמת תכנות

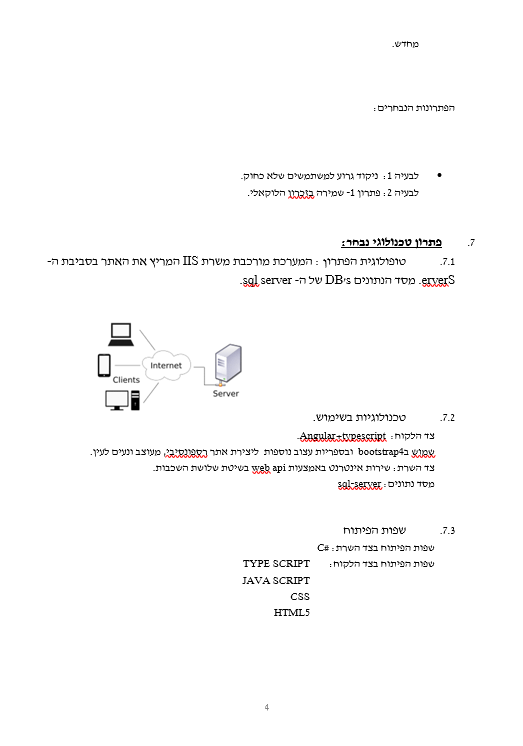
**שם המנחה**:

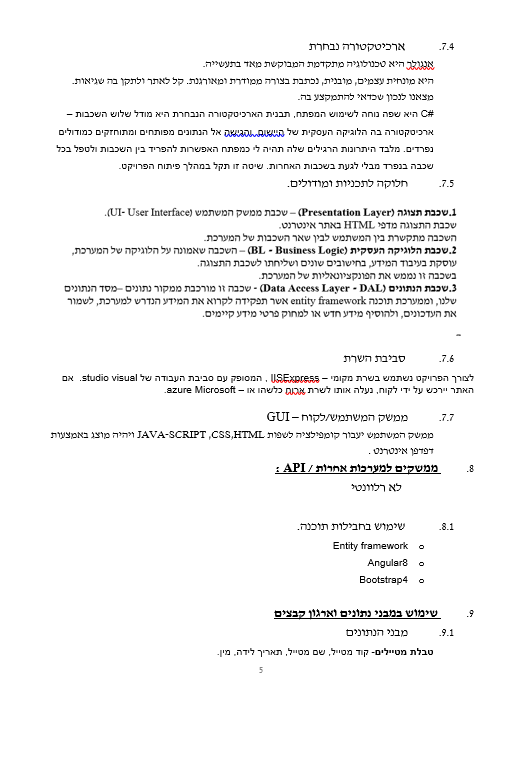
2020

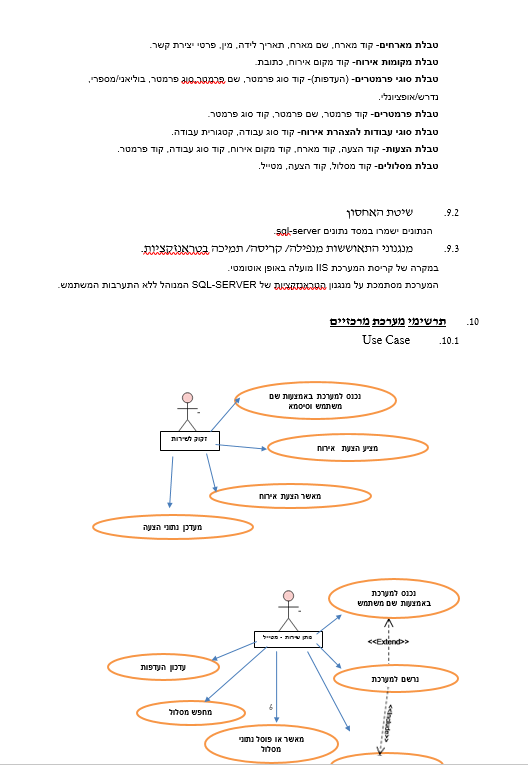


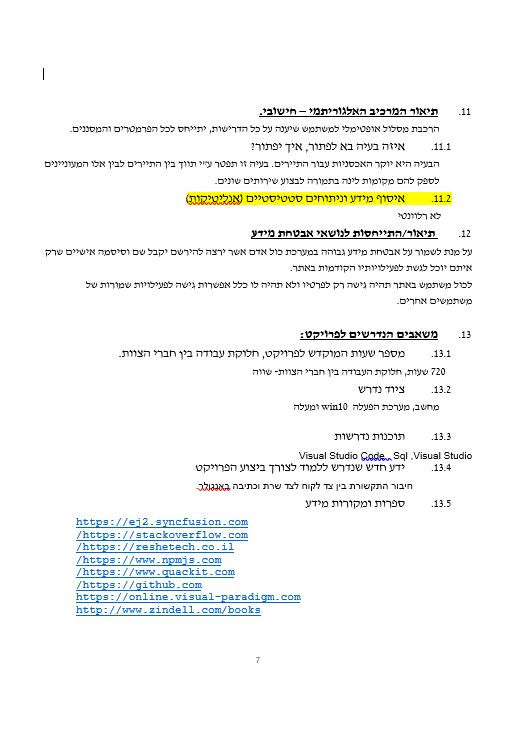


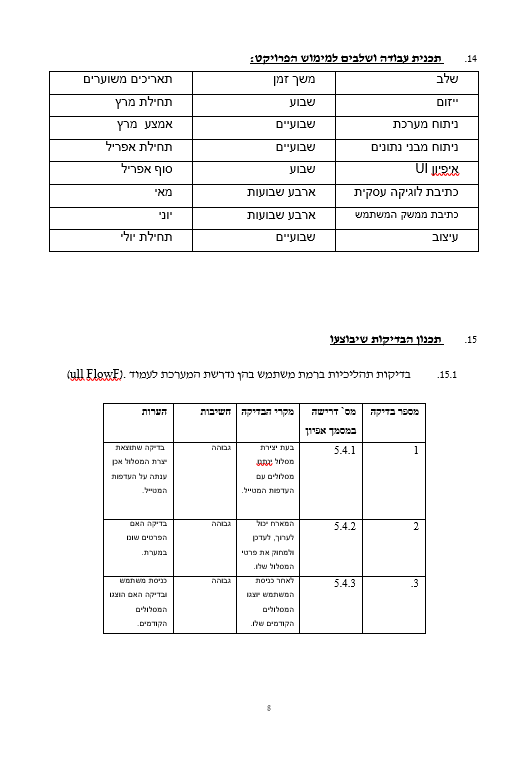


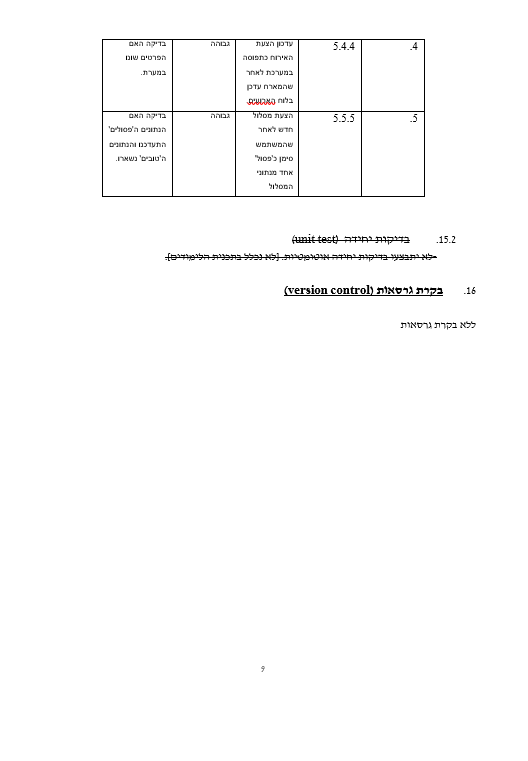


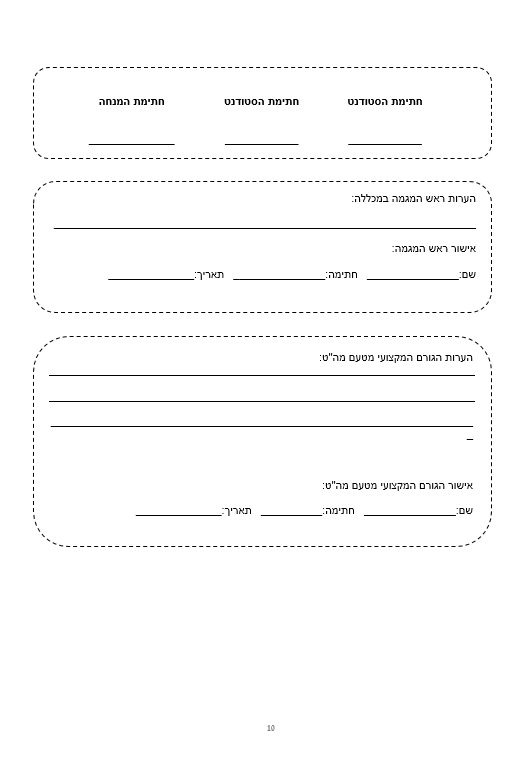












# תודות

מתוך שבח והודיה לבורא עולם, החונן לאדם דעת,

אנו מתכבדות להגיש ספר זה המאגד בתוכו את פרויקט הגמר בתכנות עליו עמלנו רבות.

**התודות והברכות**

* להורינו היקרים ולמשפחתנו על העידוד והתמיכה בכל עת ובכל שעה, החל מתקופת הלי­­­מודים ועד לפרויקט הגמר.
* למנהלת המגמה על לימוד, יחס ואוזן קשבת בכל עת.
* למורות ולמנחות היקרות, שסייעו ותמכו לאורך כל הדרך, משלב התכנון הראשוני ועד לשלב הביצוע. ייעצו, בדקו ונתנו חוות דעת, כדי שנוכל להגיע להישגים המקסימליים, ולפרויקט ראוי ומושלם.
* לצוות חדר המחשבים על התמיכה הטכנית בכל שלב,

על עזרה מושטת תמיד בעין יפה וברצון, על נתינת פתרונות מקצועיים לכל פנייה.

* לסמינר על ההשקעה, המקצועיות והנכונות לעזור בכל מצב

# הצהרה

**הצהרת סטודנט**

שם הסטודנט: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ת.ז. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

שם הסטודנט: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ת.ז. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

שם המכללה בה לומד הסטודנט:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

אני הח"מ, מצהיר בזאת כי פרויקט הגמר וספר הפרויקט המצ"ב נעשו על ידי בלבד.

פרויקט הגמר נעשה על סמך הנושאים שלמדתי במכללה ובאופן עצמאי.

פרויקט הגמר וספר הפרויקט נעשו על בסיס הנחייתו של המנחה האישי.

מקורות המידע בהם השתמשתי לביצוע פרויקט הגמר מצוינים ברשימת המקורות המצוינים בספר הפרויקט.

אני מודע לאחריות שהנני מקבל על עצמי על ידי חתימתי על הצהרה זו שכל הנאמר בה אמת ורק אמת.

חתימת הסטודנט: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ תאריך: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

חתימת הסטודנט: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ תאריך: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**אישור המנחה האישי**

הריני מאשר שהפרויקט בוצע בהנחייתי, קראתי את ספר הפרויקט ומצאתי כי הוא מוכן לצורך הגשת

הסטודנט להגנה על פרויקט גמר.

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

**אישור ראש המגמה**

הריני מאשר שספר הפרויקט מוכן לצורך הגשת הסטודנט להגנה על פרויקט הגמר.

שם ראש המגמה: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ חתימה \_\_\_\_\_\_\_\_\_\_\_\_\_\_ תאריך: \_\_\_\_\_\_\_\_\_\_\_

# תוכן העניינים

[דף כריכה I](#_Toc526859710)

[דף שער II](#_Toc526859711)

[הצעה לפרויקט גמר III](#_Toc526859712)

[אישור הצעת הפרויקט ממה"ט X](#_Toc526859713)

[תודות 1](#_Toc526859714)

[הצהרה 2](#_Toc526859715)

[תוכן העניינים 3](#_Toc526859716)

[1. מבוא 4](#_Toc526859717)

[2. מדריך למתכנת: 7](#_Toc526859718)

[2.1. אסטרטגיות טכנולוגיות: 7](#_Toc526859719)

[2.2. תיאור מבנה הפרויקט: 7](#_Toc526859720)

[2.3. עקרונות הפיתוח: 8](#_Toc526859721)

[2.4. עקרונות התכנון/ הבניה/ הניתוח 10](#_Toc526859722)

[2.5. תרשימים 11](#_Toc526859723)

[2.6. מבני נתונים מאוחסן 15](#_Toc526859724)

[2.7. תוכן הפרויקט 17](#_Toc526859725)

[3. מדריך למשתמש: 34](#_Toc526859726)

[3.1. הוראות כלליות לשימוש באתר: 34](#_Toc526859727)

[3.2. מסכים: 35](#_Toc526859728)

[4. סיכום ומסקנות: 36](#_Toc526859729)

[5. נספחים: 37](#_Toc526859730)

[7. ביבליוגרפיה: 44](#_Toc526859731)

## מבוא

האתר מציע תווך בין מטיילים המעוניינים לטייל מסביב לעולם ולהקטין את הוצאות הלינה, הנסיעה וכו' ע"י אנשים הזקוקים לשהייתם במקום לצרכים שונים (לדוג' אדם הזקוק להעברת חבילה ליעד מסוים ישלח אותה בידי מטייל וישלם לו את דמי הנסיעה) .

המערכת תיצור למטייל מסלול ע"פ הנתונים שהוזנו

הכניסה לאתר למשתמשים שירשמו לאתר.

האתר מציע את האפשרויות הבאות למשתמשים :

* לאפשר למשתמשים להכניס את פרטי ההצעה שלהם ומה רוצים בתמורה.
* לאפשר חיפוש נוח למשתמשים שמחפשים לקבל על מנת לתת
* לאפשר למשתמשים שמוכנים לתת לקשר בין מי שמוכן לתת על מנת לקבל

מטרתו של הספר היא להציג בפניכם הקוראים את הפרויקט מהבטוי השונים

להתרשם מהנעשה בו ולדעת כיצד להפיק ממנו את המירב.

הספר מכיל מדריך לתכנת המחולק לעקרונות התכנון ועקרונות הבניה.

העקרונות התכנון המתכנת יכול לראות א תיאור הפרויקט מבחינה תיכנותית, האלגוריתמים הנכללים בפרויקט, התהליכים שישי במערכת, תיאור בסיסי הנתונים והבעיות שליוו אותנו במהלך פיתוח התוכנה עם דרכי הפתרון שהשתמשנו בהם .

בעקרונות הבניה תיארנו את המחלקות הקיימות בפרויקט ואת הפונקציות העיקריות בצירוף הקוד שלהן כולל הסברים, לפי סדר המחלקות או המסכים כמו כן מכיל הספר מדריך למשתמש שבו יש את צילומי המסכים והסברים כיצד להשתמש במגוון האפשרויות הכלולות בתוכנה.

אנו תקוה שתערב לכם הקריאה בספר ותפיקו ממנו תועלת רבה.

מודל הפרויקט מחולק ל4 חלקים:

1. בנית classes מתאימים ויצירת השדות והפונקציות הנדרשות בתוכם.

יצירת controllers המנהלים את קריאות הclient ומתחברים לSql כדי לשמור את הנתונים

2. קוד Typescript הנכתב בפלטפורמת angular

ובו גם תקשורת עם שרתים השונים חישובים אלגוריתמים שונים ותקשורת עם הserver

3. תצוגת html המוגשת כאתר בממשק חדשני ונח לשימוש.

שלב אפיון וניתוח הפרויקט היווה משימה לא פשוטה ומרכזית בפיתוח המערכת, ודרש ידע רב ואיסוף רחב של מידע. משימת הפיתוח הייתה מורכבת ומאתגרת, ודרשה חשיבה יצירתית בראש גדול והרבה למידה עצמאית.

## מדריך למתכנת:

### אסטרטגיות טכנולוגיות:

בכתיבת פרויקט זה שמנו דגש רב על התאמה טכנולוגית מדויקת ושאיפה לטכנולוגיות חדשניות,

ויחד עם זאת כאלה שלא תפגענה באיכות הפרויקט ותתאמנה לדרישותיו.

צד שרת –server side כתבנו בשפת c# בפלטפורמת Asp.net Web API .

צד לקוח– מנהל חברה/משתמש – client side כתבנו בשפות Html5, Typescript, SCSS בפלטפורמת .Angular

### תיאור מבנה הפרויקט:

#### מבנה הServer side:

מבנה Server Side השרת נבנה בהתאם למודל השכבות וכולל את הפרויקטים הבאים:

א. DAL- פרויקט זה משמש כשמו Data Access Layer כאחראי על המידע, כלומר על מסד הנתונים. פרויקט זה מתחבר למסד הנתונים הנדרש לפרויקט (SQL)

ב. BLL- פרויקט זה מכיל את עיקר הלוגיקה של צד השרת, ומטפל בעיקר בשליפות ושמירת נתונים.

ג. WebApi- חשיפת API עימו מתממשק האתר, כגון: התחברות משתמש, הוספת משתמש , שליפת נתונים, ועוד.

כל שכבה מכירה את השכבה שמעליה בלבד.

#### מבנה הClient side :

מבנה ה-client מורכב משלושה אלמנטים: דפי SCSS דפיHTML ודפי TS.

דפי ה- SCSS אחראיים על העיצוב והמראה הכללי,

דפי ה-HTML מהווים את התצוגה,

ודפי ה-TS כוללים בתוכם פונקציות לוגיות שונות, וכן שירותים שונים

ההתחברות לצד השרת ע"י קריאות Http

### עקרונות הפיתוח:

מאחורי קוד המערכת עומדים עקרונות תכנותיים רבים שחלקם לא היו מוכרים לנו. בסעיף זה ננסה

להבהיר את העקרונות העיקריים שעל פיהם פעלנו לאורך כתיבת הקוד.

**SQl**

היא מערכת לניהול בסיס נתונים במודל היחסי (RDBMS) של חברת מיקרוסופט., שפת הפיתוח שבאמצעותה מועברות הפקודות למערכת (על ידי כתיבת פקודות או שימוש בממשק גרפי ששולח פקודות באופן סמוי) היא Transact-SQL ‏(TSQL) שהיא מימוש תקן ANSI של שפת SQL משמשת לתשאול וטיפול בנתונים (DML), יצירת טבלאות והיחסים ביניהן (DDL) ותחזוקת המערכת תוך שימוש בתוכניות שירות שונות.

SQL הוא מסד נתונים בו נשמרים הנתונים בטבלאות בעלות קשרי גומלין, וניתן לשלוף ממנו מידע ע"י שאלתות SQL.

יש לו יכולות רבות נוספות כמו view, פרוצדורות, פונקציות, טרנזקציות ועוד.

זהו מסד הנתונים הנפוץ ביותר.

**ASP.net** **Web API**

Iמה זה API?

API הוא קיצור של .Interface Program Application במונחים פשוטים, זהו הממשק של פיסת

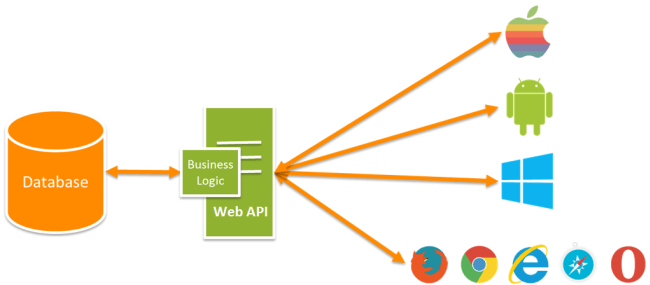
תוכנה עם העולם החיצוני, הקובע איך ישתמשו בה. אלה החוקים שנקבעו עבור האינטראקציה

שלה עם העולם הרחב, אלה יקבעו אילו חלקים של התוכנה יכולות לדבר עם תוכנות אחרות, ואיך

היא תגיב.

מה זה Web API ?

ASP.NET Web API הוא מסגרת המקלה על בניית שירותי HTTP המגיעים למגוון רחב של לקוחות, כולל דפדפנים והתקנים ניידים. ASP.NET Web API הוא פלטפורמה אידיאלית לבניית יישומים RESTful ב- NET Framework.



מה זה REST?

Rest( ראשי תיבות Transfer State Representational )היא סגנון ארכיטקטוני לכתיבת צד שרת.

התפיסה הארכיטקטונית ב- Rest היא תפיסת שרת-לקוח. תפיסה זו מחייבת קיום לקוח ושרת.

לקוחות יוזמים פניות המכילות בקשות לשרתים. השרת מעבד את הפנייה, ומחזיר תגובות

מתאימות.

בכל מצב נתון הלקוח יכול להיות בתהליך של שינוי מצב או במצב מנוחה (rest .(במצב של מנוחה

הלקוח יכול להיות באינטראקציה עם המשתמש, אבל אינו תופס משאבים בשרת. הלקוח שולח פניות

כאשר הוא מוכן לעבור למצב חדש. כאשר קיימת פנייה אחת או יותר שטרם הסתיים הטיפול בהן,

הלקוח נמצא במצב של מעבר ממצב למצב.

REST הוא לא פרוטוקול )כמו SOAP )אלא יותר קונבנציה שבה אנחנו משתמשים. Rest מבוסס ברב

המקרים על פרוטוקול HTTP ,למרות זאת REST היא ארכיטקטורה כללית הניתנת למימוש גם

בסביבות אחרות ולא רק תחת HTTP.

עבודה ב REST-מחייבת התחשבות באילוצים ובמגבלות של ארכיטקטורה זו, נפרט את חלקם:

• שרת-לקוח: עובד רק בתפיסת שרת לקוח, כאשר האחד אינו מושפע ממה שמתרחש באחר,

למעט המסרים העוברים ביניהם.

• Stateless :ההקשר (context (של הלקוח בפניה לשרת, אינו נשמר בשרת. מגבלה זו נועדה

לשפר את מדרגיות (Scalability (השרת.

כשאנו יוצרים שירות שממלא אחר הקונבנציות של REST אנחנו יוצרים שירות RESTful.

:Rest API - לסיכום

ממשק API של REST מגדיר קבוצה של פונקציות אשר מפתחים יכולים לבצע בקשות ולקבל תגובות

באמצעות פרוטוקול HTTP כגון GET ו- POST.

**Typescript**

היא שפת תכנות חינמית ומבוססת קוד פתוח המפותחת ומתוחזקת על ידי מיקרוסופט .היא מכילה

את קבוצת כל פקודות ותחביר JavaScript הפופולרית, כלומר כל קוד

JavaScriptהוא גם קוד Typescript תקין, ומוסיפה עליה טיפוסים סטטיים ותכנות מונחה עצמים

מבוסס מחלקות. בין התכונות שהיא מוסיפה: typing static ,תמיכה במחלקות ותמיכה במודולים

ובדקורטורים.

Typescript היא סופר-סט (superset) של JavaScript כלומר, הקוד מבוסס על JavaScript וחייב

לעבור קומפילציה (תרגום) ל JavaScript-כדי שהדפדפנים יבינו אותו מפני שדפדפנים

מבינים JavaScript בלבד. בסופו של דבר קוד Typescript מעובד לכדי קוד JavaScript ולכן ניתן

לומר "באחריות" שהשפה נתמכת בכל דפדפן שתומך ב- JavaScript.

קבצי Typescript יישמרו בסיומת ts.

### עקרונות התכנון/ הבניה/ הניתוח

הפרדת שכבות:

כל תוכנה/ אתר שנפגוש בעולם מבוססת על ארכיטקטורת שכבות הנקראת בשם Three Tier- -שלש שכבות במבנה שלBL-UI Application DAL-. זוהי תבנית עיצוב בסיסית שמגדירה הפרדת האפליקציה ל: שכבת נתונים, שכבת לוגיקה, ושכבת ממשק משתמש.

לתבנית עיצוב זו יתרונות רבים:

1.תחזוקה:

ניתן להחליף או לתקן מימוש פנימי של שכבה אחת בארכיטקטורה בלי לשנות שכבה אחרת.

2.נוחות פיתוח:

אדם אחד עובד על רכיב בתכנה, אדם אחר עובד על רכיב אחר, כל עוד שהחתימות זהות ניתן לשלב כוחות ולייעל זמני פיתוח.

3. בדיקות:

תקלה כלשהיא מבודדת בכל שכבה בנפרד, לדוגמא אם לא קיבלנו רשימת נתונים לתצוגה נבדוק קודם את שלב הבינים (שכבת ה BL) אם הנתונים שם תקינים נדע בוודאות שגם ברמת שכבת ה DAL הנתונים תקינים וכל שנותר הוא לפתור את התקלה ברמת ה UI .

4. שימוש חוזר:

נניח שהרכיבים BL+ DAL אהובים עלינו ועובדים היטב, ניתן להחליף את פלטפורמת UI לטכנולוגיה אחרת במינימום מאמץ.

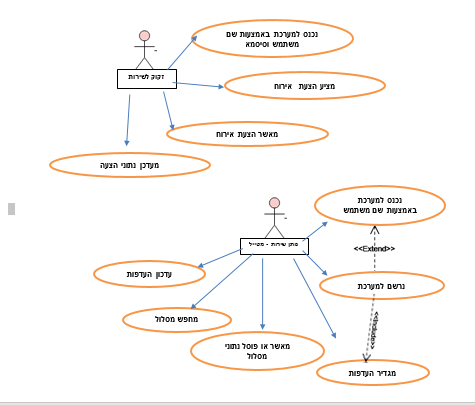
5. אבטחה:

נוח יותר להגדיר Interface API לשכבה מסוימת בלי לחשוף מבני נתונים Logic או Data שלא רלוונטי למשתמש, כמו גם חסימה בפני האקרים (רלוונטי יותר בטכנולוגיות Web ) ברמות שונות.

הפרויקט של צד שרת- מכיל את שלושת השכבות הנ"ל.

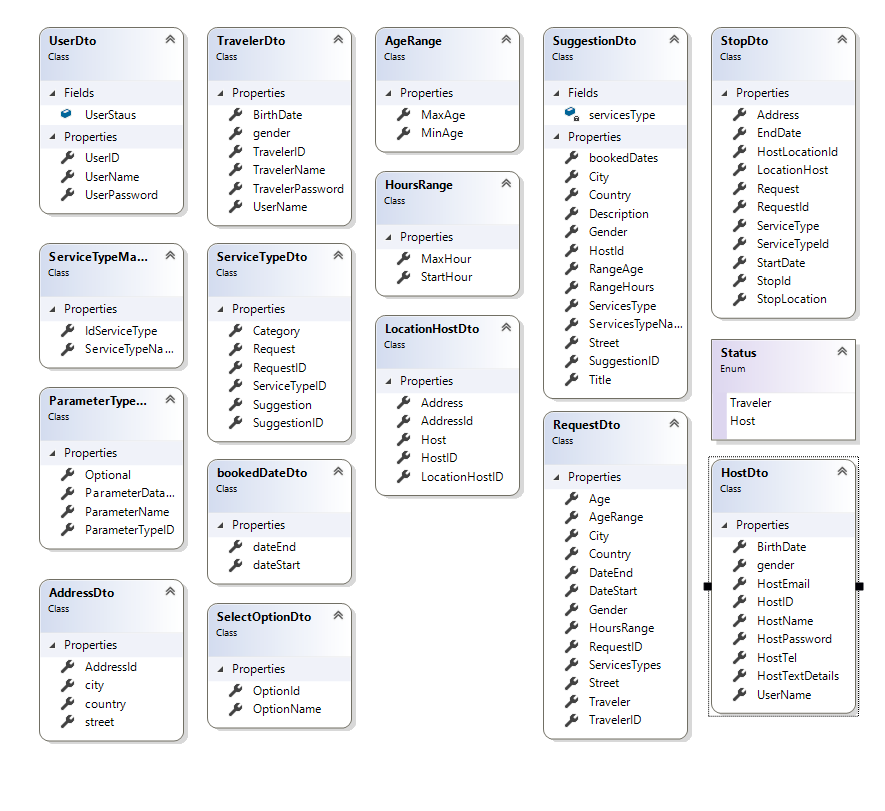
### תרשימים

#### תרשים זרימה

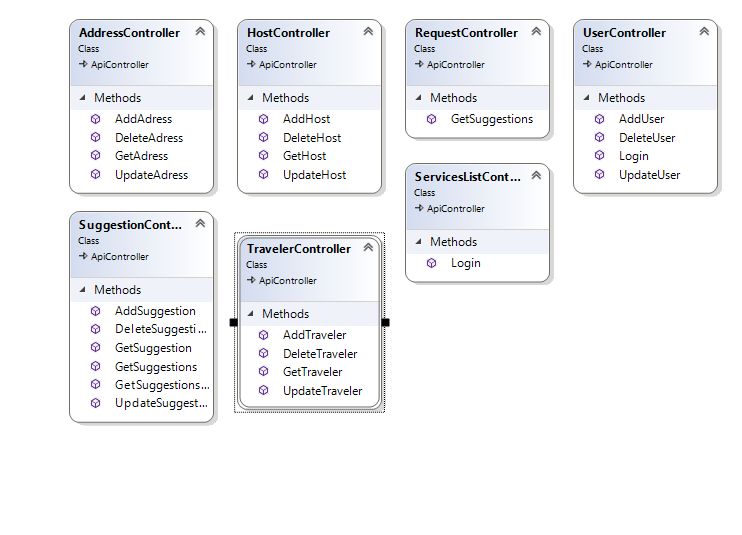


#### תרשים מראה מחלקות

* פרויקט הDal



פרויקט Controller



#### מבנה קבצים ותיקיות

צד client :

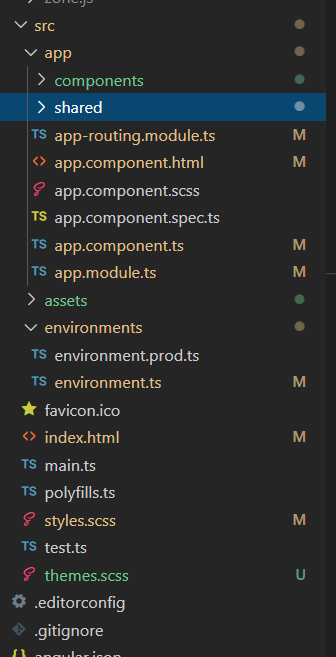
בתקיה זו נשמרים כל התמונות והקבצים שאנו משתמשים באתר

בתוך תיקיה זו נמצאים כל הקבצים שמגדירים דברים לכל האתר

1. קובץ app.scss מכיל עיצוב גלובלי לכל האתר
2. קובץ app.html

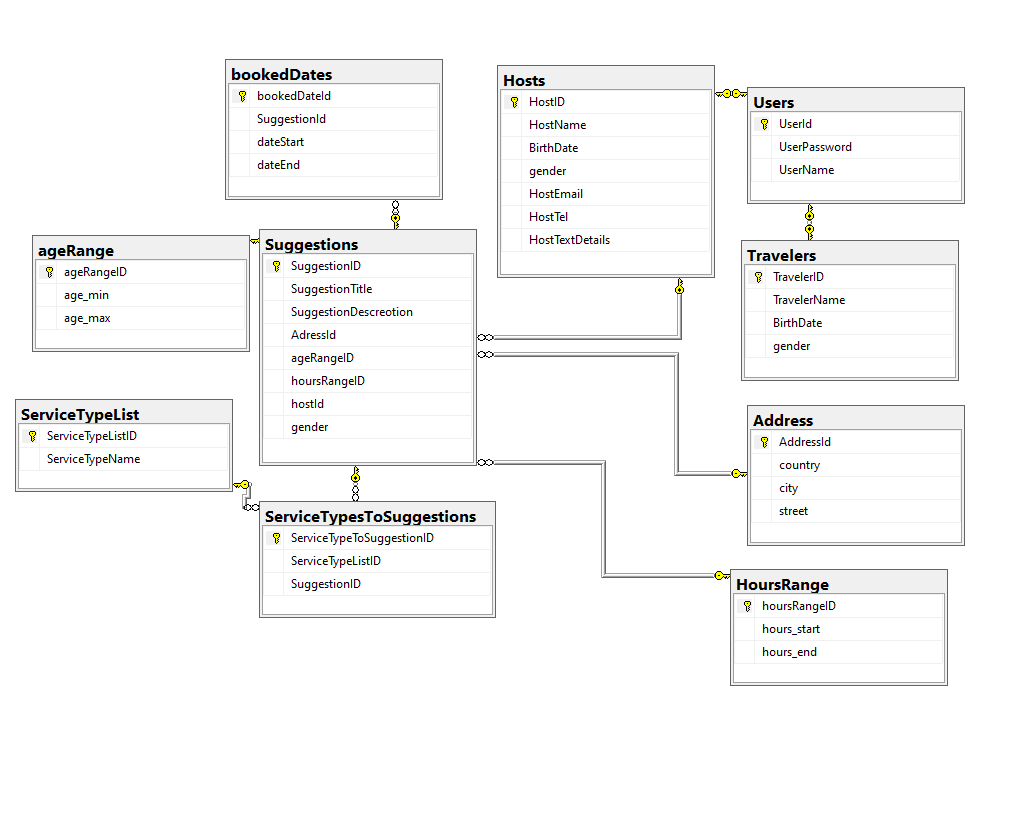
בתיקיות אלו נמצאים כל דפי האתר מסודרים בתיקיות.

כל תיקיה מכילה שלושה קבצים:html, scss, ts



### מבני נתונים מאוחסן

#### מבנה SQl

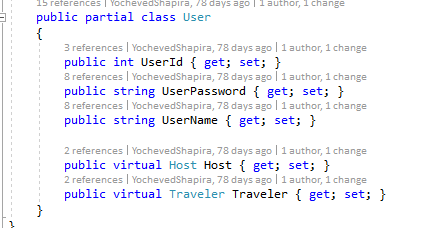


### תוכן הפרויקט

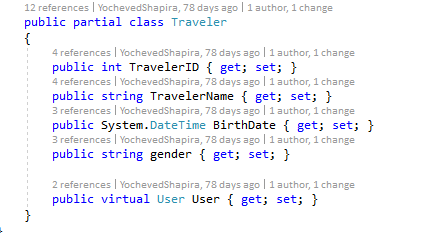
#### תיאור מחלקות

מחלקות ב Dal

**User**–משתמש



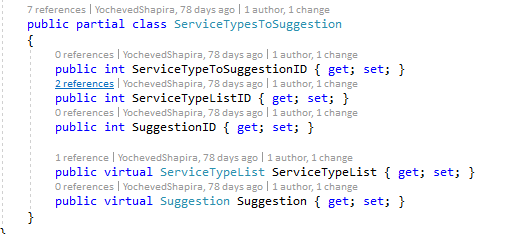
**Traveler**- מטייל



**Suggest** -הצעה



**ServiceTypeToSuggestion –**מחבר בין הצעה לסוג שרות



**ServiceType-**סוג שרות



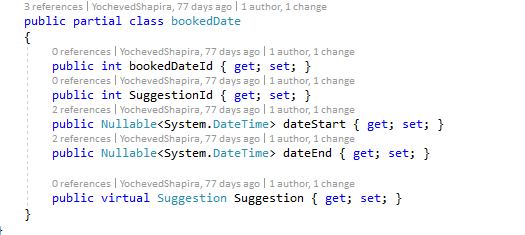
**HourRange-**טווח שעות



**Host-**מארח



**BookDate-**תאריכים תפוסים



**AgeRange-**טווח גלאים



**Address-**כתובת



**BL**

**:** **ManageHost**



**ServiceListBl**



**ManageSuggestion**

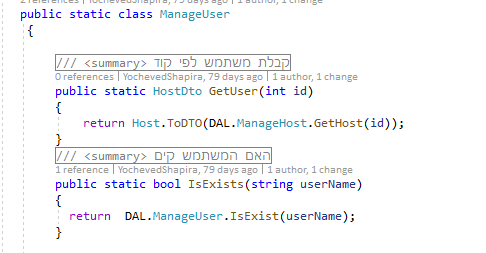




**ManageTravelBl**



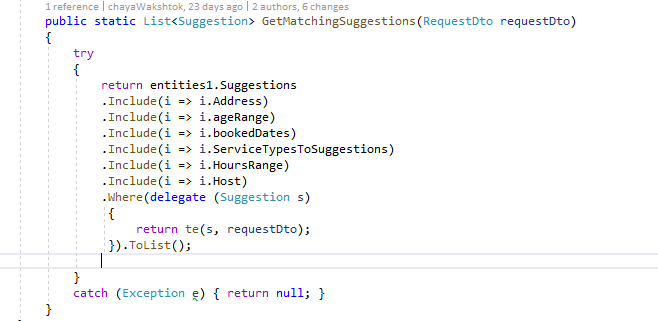
**ManageUser**



#### תאור הפנקציות

**פונקציות בצד שרת:**

1. פונקציה **להחזרת כל ההצעות שמתאימות לשרות**



1. פונקציה **להתאמה בין הצעה לבקשה**



## מדריך למשתמש:

### הוראות כלליות לשימוש באתר:

#### מדריך למארח

הכנס לאתר.

כאשר אתה רשום-היכנס לכניסה מלא את שדה שם משתמש וסיסמה וכנס למערכת.

כאשר כניסתך פעם ראשונה מלא את כל פרטיך והרשם לאתר

במסך ראשי צפה בכל ההצעות שלך, ובאפשרותך גם להוסיף הצעה חדשה ולערוך את כל ההצעות שלך ולצפות בכל הבקשות המשתמשים האחרים.

#### מדריך למחפש

הכנס לאתר.

כאשר אתה רשום-היכנס לכניסה מלא את שדה שם משתמש וסיסמה וכנס למערכת.

כאשר אתה נכנס פעם ראשונה- מלא את פרטיך והירשם לאתר.

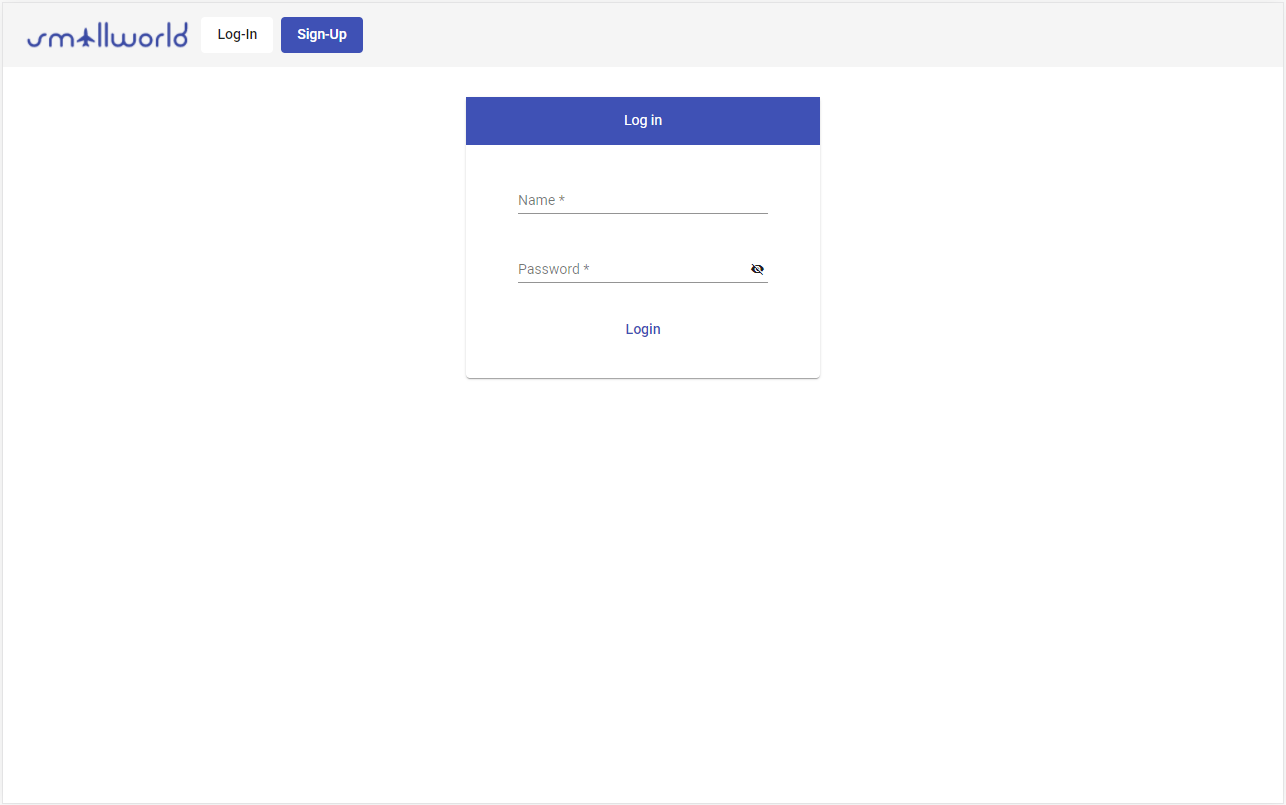
במסך ראשי באפשרותך לעדכן את פרטיך .

במסך בקשות צפה בכל ההצעות שרלוונטיות לגביך, באפשרותך להוסיף בקשה חדשה ולערוך את בקשותיך.

במסך חיפוש באפשרותך לחפש הצעה לפי פרמטרים מורחבים.

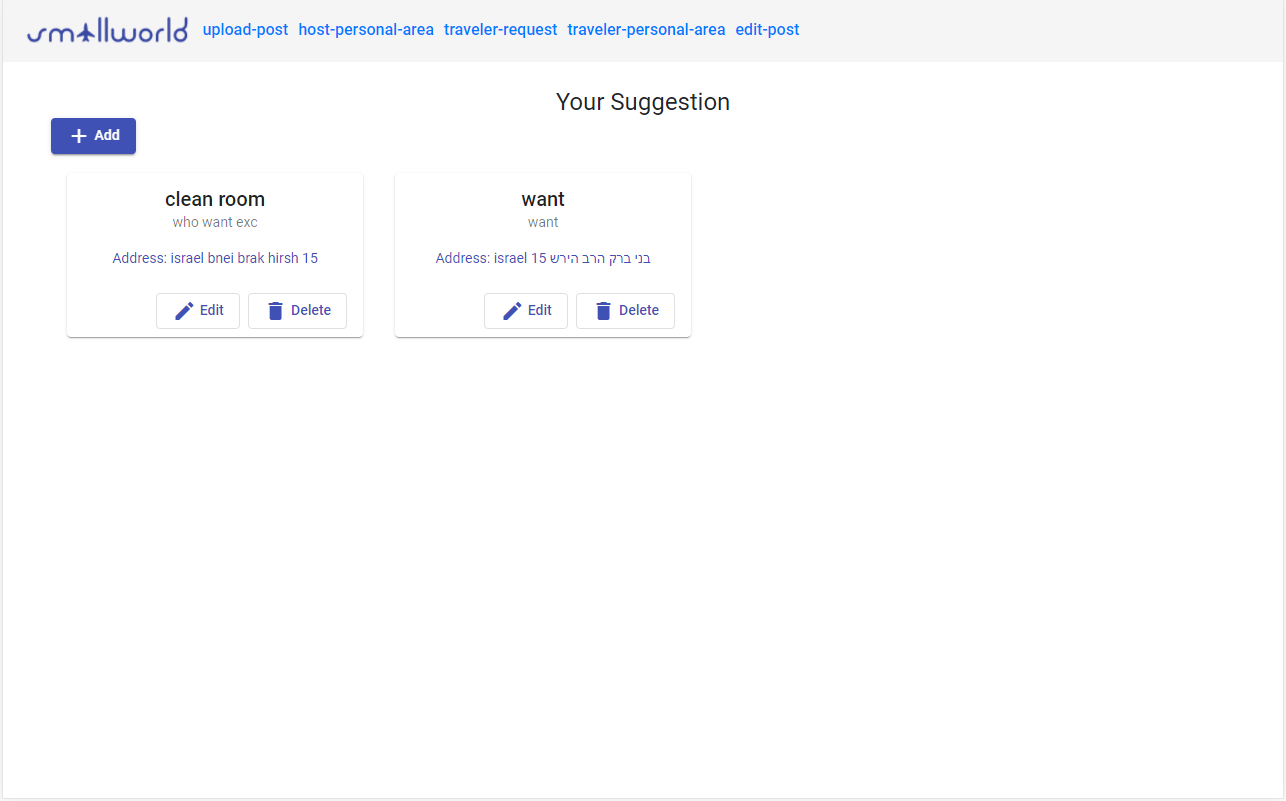
### מסכים:

#### כניסה

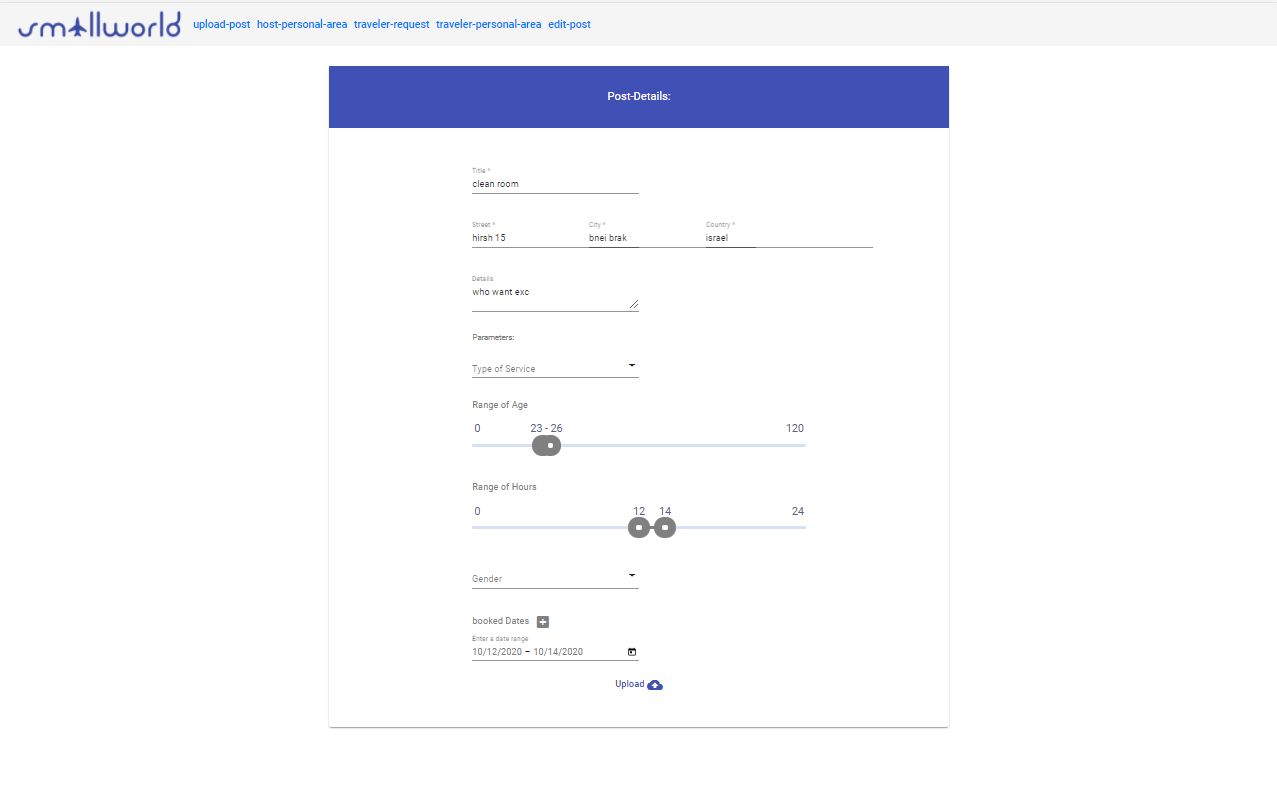


**מארח**:

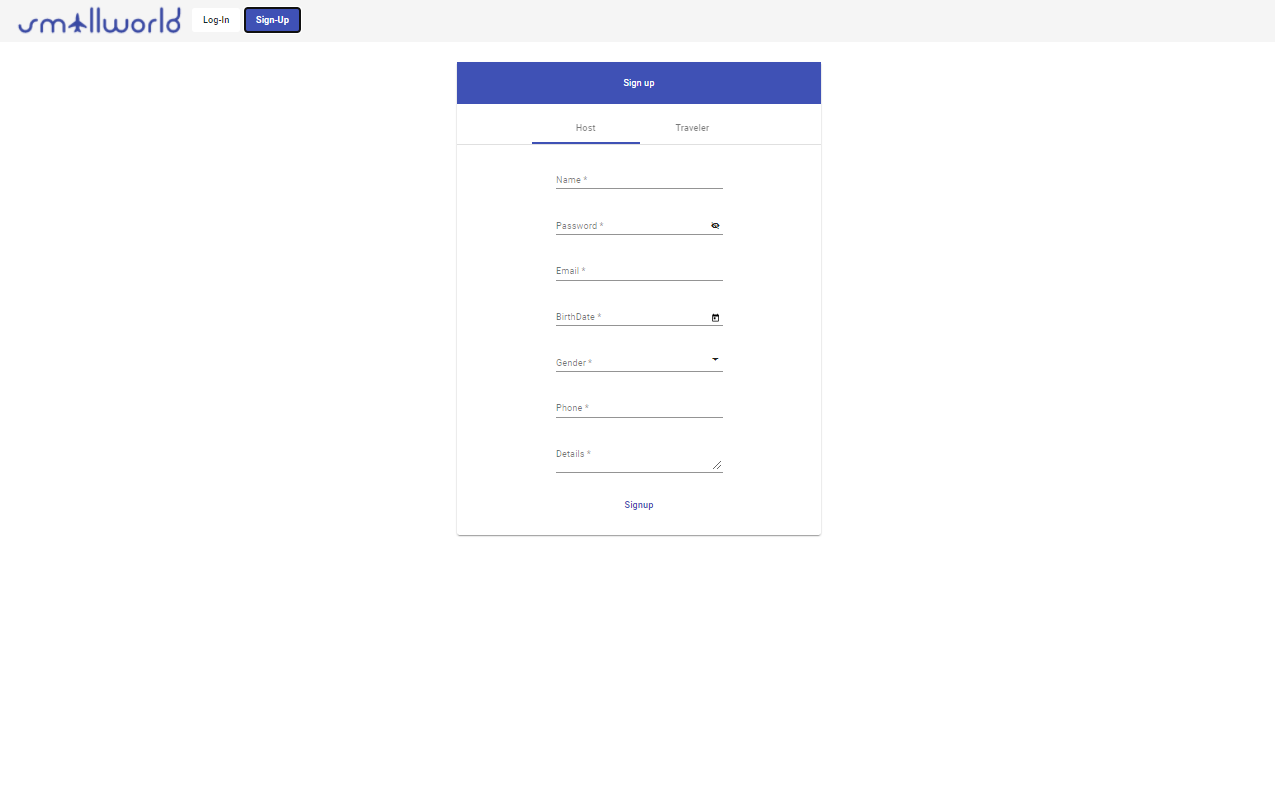
#### הצעות –צפה בהצעות שהציע



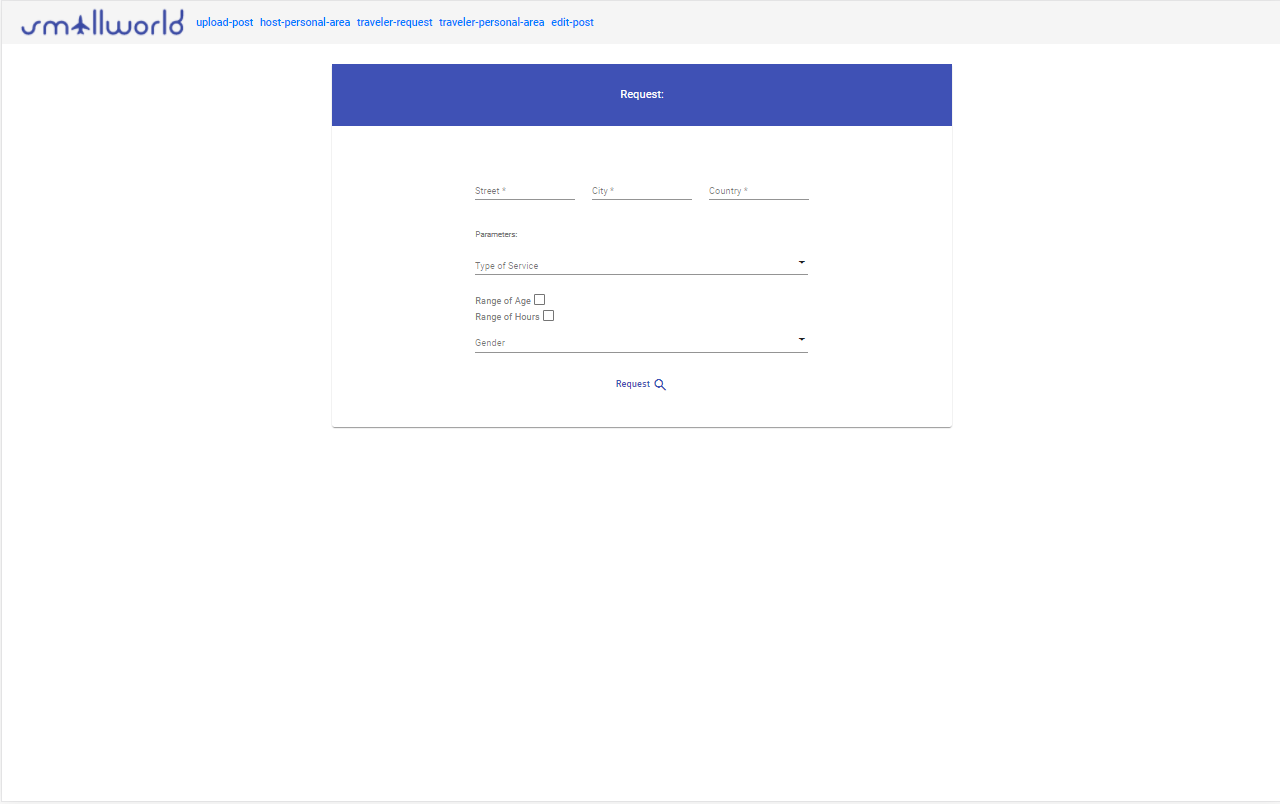
#### הוספת\עריכת הצעה



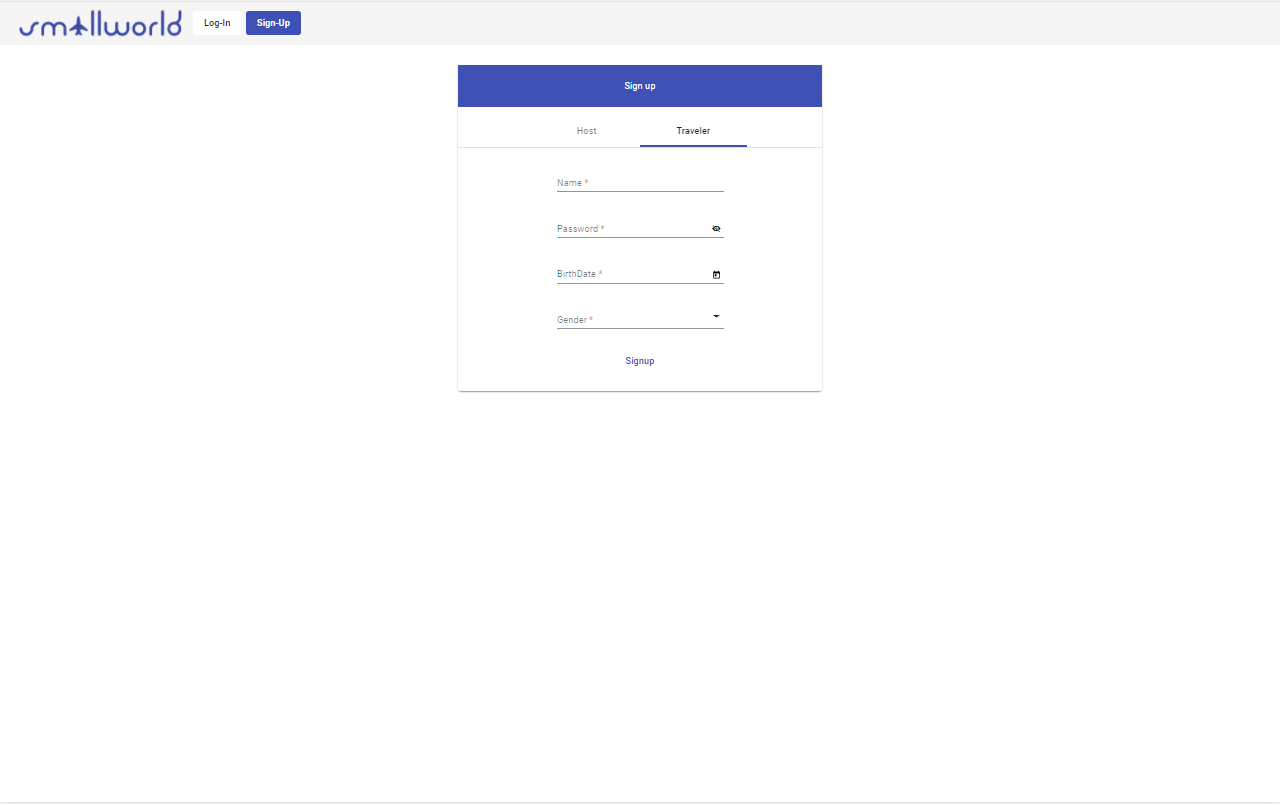
#### הרשמת מארח

 **מבקש:**

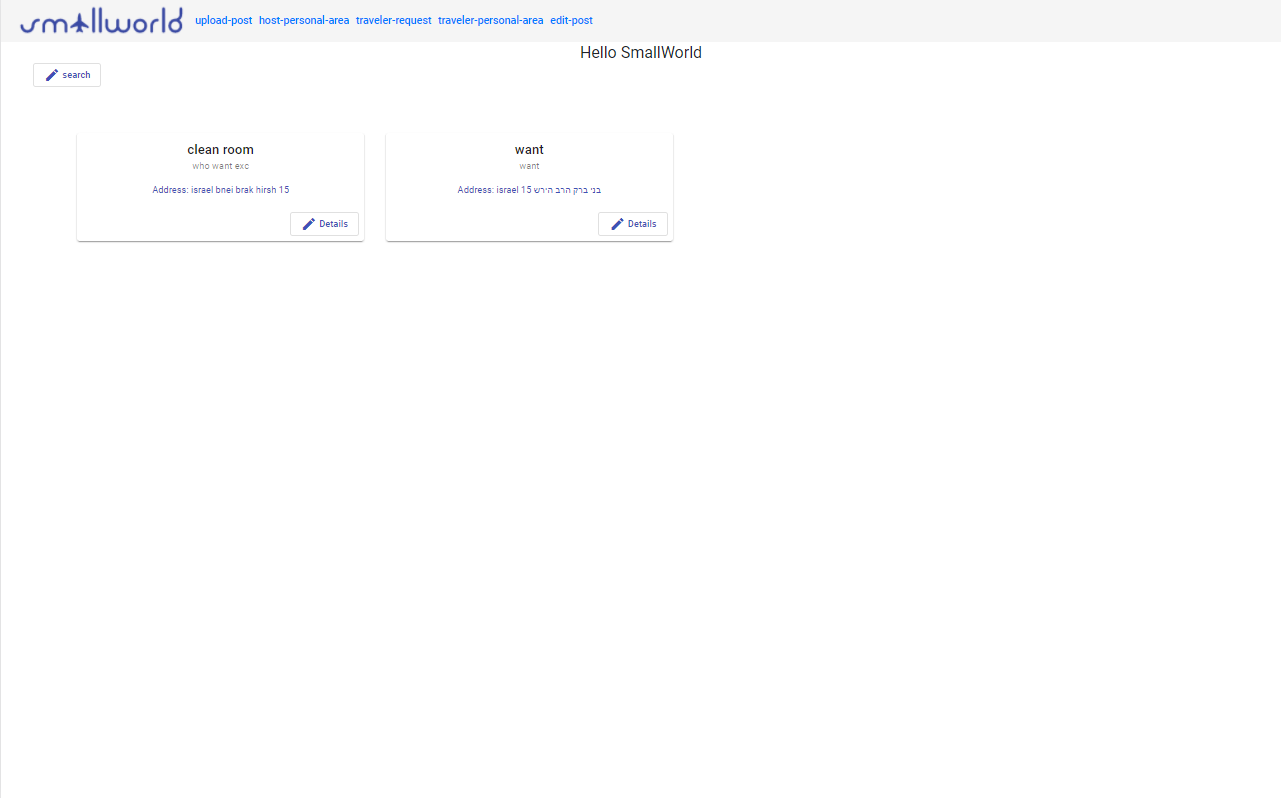
#### הוספת בקשה



#### הרשמת מבקש



#### צפה ברשימת ההצעות



#### חפש לפי קריטריונים הצעה מתאימה

## 

## סיכום ומסקנות:

השקענו רבות בבניה נכונה של המערכת ועל תקשורת נכונה עם השרת, ראשית הפרויקט הקנה לנו מרחב הסתכלות ופרספקטיבה רחבה על פרויקט מושלם, משלב ההצעה ועד לתוצר המוגמר תוך דרך ארוכה הכוללת באגים ופתרונם ויצירתיות רבה על מנת להפוך את האתר לאתר יעיל ,חווייתי ומקצועי ככל האפשר. במהלך הפרויקט למדנו הרבה ,ואנו מרגישות כי הוא תרם לנו רבות . תחומים שהכרנו באופן שטחי ,נעשו ידידותיים בהרבה ממה שחשבנו .השקענו זמן ומאמץ מרובים בלמידת נושאים וטכנולוגיות חדשים .גילינו שמהנה גם אם לא קל ללמוד ולהכיר תחומים חדשים ולנסות דרכי חשיבה מגוונות ומפתיעות שלא חשבנו עליהם ממבט ראשון. למדנו לחשוב בהגיון וביעילות על מנת להשיג תוצאות במהירות. למדנו לעבוד עם angular

ומעל הכול הפנמנו שסבלנות והתמדה-אין כמותם בכל משימה הנראית קשה עד בלתי אפשרית...

## נספחים:

**Dependences- פרוייקט:**

## 

## ביבליוגרפיה:

אתרים בנושא תכנות:

* stackoverflow.com
* webmaster.org.il
* [.c-sharpcorner.com](https://www.c-sharpcorner.com/article/web-api-in-asp-net)
* [.djamware.com](https://www.djamware.com/post/5a48517280aca7059c142972/ionic-3-angular-5-firebase-and-google-maps-location-tracking)
* .joshmorony.com
* .dotnetcurry.com
* .msdn.microsoft.com