פנינה בולבין ובת שבע ריץ'

ספר פרויקט

מתנ"ס מעלה רמות

[הצעה לפרויקט גמר 5](#_Toc44926398)

[פרטי הסטודנט 5](#_Toc44926399)

[פרטי המנחה האישי 5](#_Toc44926400)

[1.שם פרויקט הגמר 5](#_Toc44926401)

[2.מטרת פרויקט הגמר 5](#_Toc44926402)

[שלמי תודה 10](#_Toc44926403)

[מבוא 12](#_Toc44926404)

[1. הגדרת דרישות ותיאור כללי 14](#_Toc44926405)

[1.1 תיאור כללי: 14](#_Toc44926406)

[1.1.1 מטרות המערכת 14](#_Toc44926407)

[1.1.2 היקף הפרויקט 14](#_Toc44926408)

[1.1.3 מבנה המערכת 14](#_Toc44926409)

[1.1.4 משימות המערכת 15](#_Toc44926410)

[1.1.4.1 אתר 15](#_Toc44926411)

[1.1.4.2 שרת 15](#_Toc44926412)

[1.2 תיאור חומרת המערכת 15](#_Toc44926413)

[1.2.1 כללי 15](#_Toc44926414)

[1.2.2 מרכיבי המערכת 15](#_Toc44926415)

[1.3 תיאור תוכנת המערכת 15](#_Toc44926416)

[1.3.1 כללי 15](#_Toc44926417)

[1.3.2 כלי התוכנה לפיתוח המערכת 15](#_Toc44926418)

[1.3.3 טכנולוגיות בשימוש: 16](#_Toc44926419)

[1.4 תיאור פונקציות המערכת 16](#_Toc44926420)

[1.4.1 צד לקוח 16](#_Toc44926421)

[1.4.1.1 פונקציות CRUD 16](#_Toc44926422)

[1.4.1.2 פונקציות לוגיות 17](#_Toc44926423)

[1.4.1.3 פונקציות תקשורת עם השרת 17](#_Toc44926424)

[1.4.1.4 פונקציות יבוא ויצוא נתונים 17](#_Toc44926425)

[1.4.2 שרת 17](#_Toc44926426)

[1.4.2.1 פונקציות ניהול מאגר נתונים 17](#_Toc44926427)

[1.5 זרימת המידע במערכת 18](#_Toc44926428)

[1.5.1 זרימת המידע - כללי: 18](#_Toc44926429)

[1.5.2 תאור תהליך הזרימה: 18](#_Toc44926430)

[1.5.2.1 כניסה ראשונית לאתר והעלאת קובץ אקסל לצורך אתחול נתונים: 18](#_Toc44926431)

[1.5.2.2 פעילות שוטפת באתר: 18](#_Toc44926432)

[1.5.2 תרשים תהליכים – User Cases: 18](#_Toc44926433)

[1.5.2.1 כניסת מנהלת מערך המתנדבים לאתר: 19](#_Toc44926434)

[2. ממשקים חיצוניים 21](#_Toc44926435)

[3. ממשק אדם מכונה 23](#_Toc44926436)

[3.1 כללי: 23](#_Toc44926437)

[3.1.1 עקרונות בניית הממשק: 23](#_Toc44926438)

[3.1.2 הממשק הגרפי כולל: 23](#_Toc44926439)

[3.2 תיאור מסכים: 23](#_Toc44926441)

[3.2.1 דף בית: 23](#_Toc44926442)

[3.2.2 מסכי צפיה בנתוני מרכיבי ההתנדבות: 24](#_Toc44926443)

[3.2.3 הודעות בקרה למשתמש: 26](#_Toc44926447)

[4 מבנה נתונים וארגון קבצים 29](#_Toc44926448)

[4.1 ארגון קבצים 29](#_Toc44926449)

[4.1.1 כללי: 29](#_Toc44926450)

[4.1.2 תיאור קבצים: 29](#_Toc44926451)

[4.1.2.1 צד שרת 29](#_Toc44926452)

[4.1.2.2 צד לקוח 30](#_Toc44926453)

[4.1.3 מבנה נתונים 33](#_Toc44926454)

[4.1.2.1 מבנה הטבלאות: 34](#_Toc44926455)

[5 תכנות המערכת 38](#_Toc44926456)

[5.1 תיאור כללי: 38](#_Toc44926461)

[5.1.1 צד שרת: 38](#_Toc44926462)

[5.1.2 צד לקוח: 38](#_Toc44926463)

[5.2. עקרונות תכנות: 38](#_Toc44926467)

[5.3. תיאור אלגוריתמים ודוגמאות קוד 38](#_Toc44926468)

[5.3.1. צד שרת 38](#_Toc44926469)

[5.3.1.1. שכבת הנתונים- DAL: מחלקות המנהלות את הטבלאות במסד הנתונים, מבצעות שליפה, עדכון, מחיקה והוספה (CRUD). 38](#_Toc44926470)

[5.3.1.2. שכבת ה-Common: מכילה מחלקות של אוביקטים שמשותפים ל-DAL ול-BLL. 42](#_Toc44926471)

[5.3.1.3. שכבת ה-BLL: מכילה את הפונקציות העסקיות לכל מרכיב, פונקציות אלו זמינות ללקוח ע"י הפעלתן באמצעות ה-WebApi. 42](#_Toc44926472)

[*5.3.1.4.* WebApi: מכיל את ה-controllers שבאמצעותן צד הלקוח ניגש לצד השרת. 43](#_Toc44926473)

[5.3.2 צד לקוח 44](#_Toc44926474)

[5.3.1.5. מודולים בשימוש: 44](#_Toc44926475)

[ AppModule- המודול הראשי, מכיל את הקומפוננטות המטפלות בנתונים ובמרכיבי ה-UI. 44](#_Toc44926476)

[ MaterialModule- מודול המרכז את הקומפוננטות שנלקחות מהספריה Angular Material ומיצא אותם אל המודול המרכזי- AppModule. 46](#_Toc44926477)

[ RoutingModule- מודול המטפל בנושא הניווט בין המסכים בדף, מיצא את הגדרות הניווט אל המודול הראשי. 47](#_Toc44926478)

[5.3.1.6. דף HTML ראשי, ע"פ מתודולוגית SPA- single page application. 48](#_Toc44926479)

[5.3.1.7. דף CSS ראשי, המכיל עיצובים כללים עבור אלמנטים במערכת. 49](#_Toc44926480)

[5.3.1.8. קומפוננטות לעיצוב מראה הדף: 49](#_Toc44926481)

[ Header: קומפוננטה לעיצוב תפריט עליון, כולל שימוש בצורה רספונסיבית למכשירים צרים. 49](#_Toc44926482)

[ Footer: קומפוננטה לעיצוב תחתית הדף, נדבק לתחתית העמוד. 50](#_Toc44926483)

[ Home: קומפוננטת דף הבית, מכיל אלמנטים המציגים את ההתראות החכמות. 50](#_Toc44926484)

[ קומפוננטות להצגת טבלה של מרכיב התנדבות, וכן פרטים על התנדבות בודדת: 52](#_Toc44926485)

[5.3.1.9. סרביסים לניהול קריאות לשרת ע"פ מרכיבי התנדבות. 56](#_Toc44926486)

[5.3.1.10. מודלים להוספה/עריכה של מרכיבי התנדבות וכן הוספת פעולת התנדבות 57](#_Toc44926487)

[5.3.1.11. פונקציות לטיפול ביבוא וביצוא של הנתונים לאקסל 62](#_Toc44926488)

[5.3.1.12. הודעות והתראות למשתמש: 65](#_Toc44926489)

[5.3.1.13. עיצוב הגדרות טבלת מרכיבי ההתנדבות והתאמתה לעברית: 66](#_Toc44926490)

[5.3.1.14. שימוש ב-Pipes: 68](#_Toc44926491)

[5.3.1.15. שימוש ב-Directives: 68](#_Toc44926492)

[5.3.1.16. הצגת ההתראות החכמות בתפריט: 69](#_Toc44926493)

[5.3.1.17. מחלקות שמיצגות את מרכיבי ההתנדבות: 69](#_Toc44926494)

[5.4 בדיקות המערכת 70](#_Toc44926496)

[6. מה הקנה הפרויקט 71](#_Toc44926497)

[7. ביבליוגרפיה 73](#_Toc44926498)

[7.1. אתרי לימוד/תיעוד קוד: 73](#_Toc44926499)

[7.2. עזרה מקוונת: 73](#_Toc44926500)

שלמי תודה

**הצעה לפרויקט גמר**

**פרטי הסטודנט**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| שם הסטודנט | ת.ז. 9 ספרות | כתובת | טלפון נייד | תאריך סיום הלימודים |
| פנינה בולבין | 209428655 | שירת הים 12 רמות א' ירושלים | 052-7143489 | 2020 |
| בת שבע ריץ | 206995805 | ששת הימים 23, ירושלים | 050-418-0122 | 2020 |

שם המכללה **מכון בית יעקב ירושלים** סמל המכללה: **76084**

מסלול ההכשרה: הנדסאים

**פרטי המנחה האישי**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| שם המנחה | כתובת | טלפון נייד | תואר | מקום עבודה/תפקיד |
| שולמית ברלין |  | 052-711-1483 |  |  |

1. **שם פרויקט הגמר**

פרויקט מתנדבים מתנס מעלה רמות

1. **מטרת פרויקט הגמר**

ארגון, ניהול ומחשוב מערך המתנדבים במנהל קהילתי "מעלה רמות" שהוא ארגון גג לכל פעילויות החסד בשכונת רמות, תוך מתן מענה לפניות נזקקות ומציאת מתנדב מתאים.

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

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

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

חתימת הסטודנט

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

חתימת הסטודנט חתימת המנחה האישי

ג. הערות ראש המגמה במכללה

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

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

ד. אישור ראש המגמה

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

ה. הערות הגורם המקצועי מטעם מה"ט

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

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

ו. אישור הגורם המקצועי מטעם מה"ט

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

1. **פרויקט מתנדבים מתנס מעלה רמות**
2. **רקע** 
   1. **תיאור ורקע כללי**

מערך המתנדבים במנהל הקהילתי "מעלה רמות" מסייע ופועל לרווחת כלל תושבי שכונת רמות. המערך כולל סיוע למשפחות במצוקה, ארגון ביקורים אצל קשישות, עזרה ליולדות וכן פרויקטים מיוחדים כגון עזרה בנקיון בערב פסח וביצוע מכירות בגדים מוזלות למשפחות ברוכות ילדים.

הפרויקט בא לתת מענה למחשב את מערך המתנדבים הקיים ולהוסיף עליו באופן יעיל.

* 1. **מטרות המערכת**

לאפשר למנהלת מערך המתנדבים עבודה ממוחשבת ויעילה, ארגון ושמירת הנתונים על המתנדבים ועל הנתרמים, וכן ביצוע התאמות בין מתנדבת לנזקקת. כמן כן קבלת מידע עדכני לגבי סטטיסטיקות של התנדבויות ואירועים שהתקיימו.

1. **סקירת מצב קיים בשוק, אילו בעיות קיימות**

במצב הקיים כיום מתנהל המערך בצורה ידנית, ללא שימוש במחשוב כלל, מה שמצריך אנרגיות מיותרות של דפדוף וחיפוש במאגר הנתונים בדרך ידנית ואיטית, כמו כן ביצוע ההתאמות מסתמך על טלפונים אקראיים, במקום להעזר בקריטריונים ברורים שיאפשרו מיקוד יעיל של התאמה בין צרכים של הנעזרים לתחומי התנדבות אפשריים של המתנדבות.

בנוסף לכך אין שום מידע על סטטיסטיקות של ההעזרות ועל היסטוריית התנדבות וקבלת עזרה של כל מקרה פרטי.

1. **מה הפרוייקט אמור לחדש ולשפר**

* הקמת בסיס נתונים אמין ומדויק שישמור נתוני אמת על הנעזרים ועל המתנדבים.
* מתן מענה לתיווך בין עוזרת לנעזרת ע"י קריטריונים מותאמים מראש.
* ביצוע סטטיסטיקות ודוחות על כמות ואופי העזרה שניתנת במערך.

**5. דרישות מערכת ופונקציונליות**

**5.1. דרישות המערכת**

* הסביבה שבה המערכת תוטמע הינה מחשב אישי מבוסס חלונות, כולל תקשורת בסיסית ברשת האינטרנט.
* הפרויקט יכתב בסביבת Visual Studio בטכנולוגית web API והנתונים ישמרו ב- DB SQL .
* קוד הקליינט יהיה כתוב על פי תקן הכתיבה שמקובל ב- Angular7.
* לפני כל מימוש של פונקציה יינתן תיעוד על פעולתה, וייכתב תיעוד במקומות לא ברורים ו/או קריטיים בקוד.
* המידע המוצג למשתמש יהיה מעודכן על פי הנתונים בשרת הנתונים. שרת הנתונים יתעדכן מיד בסיום עדכון המידע על ידי המשתמש, ובדיקת נכונותו על ידי המערכת.
* ממשק המשתמש יהיה קל ונוח לשימוש, ויהיו מוצגים הסברים על אופן השימוש במערכת.
* זמן התגובה של המערכת יהיה קצר מאוד עבור כל פעולת משתמש, גם עבור פעולות חישוביות מורכבות יותר.

**5.2. דרישות פונצקיונליות**

* הוספה, עריכה והסרה של מתנדבים ונעזרים.
* ביצוע התאמות בין נעזרת לעוזרת.
* תכנון אירועים מיוחדים.
* מעקב אחר סטטיסטיקות ודוחות שימוש.

1. **בעיות צפויות במהלך הפיתוח ופתרונות**

**6.1. בעיות אפשריות**

נפילה של המערכת (חומרה, תוכנה, תקשורת וכדומה).

בעיות תוכנה שונות.

**6.2. פתרון אפשרי**

גיבוי של הנתונים בענן, וכן אחסון של המערכת בשרת נוסף.

תמיכה של המפתחים גם לאחר סיום הפרוייקט.

**6.3. בעיות בפיתוח התוכנה**

בחירת כלי פיתוח שיתאימו למערכת הנדרשת.

בחירת בסיס הנתונים שיתאים למערכת הנדרשת.

1. **פתרון טכנולוגי נבחר**

המערכת תיבנה כאתר, ותורכב משרת, וממשק משתמש בצד הלקוח, וכן DB SQL לשמירת נתונים.

כתיבת הפרויקט בשפת C# (שפה עילית מונחית עצמים, נוחה לשימוש, עם הרבה יכולות תכנותיות) – תוך שימוש בטכנולוגית Entity FrameWork.

צד client – ב-Angular7 - שימוש ב CSS , Bootstrap4, ובטכנולגית web API.

הארכיטקטורה הנבחרת היא רב שכבתית- שכבת תצוגה שכבת יישום ושכבת נתונים, תוכנה ותבנית עיצוב.

המערכת מחולקת לקטגוריות בצורה חכמה ומסודרת כך שהקוד יהיה מסודר וקל לתפעול.

החלוקה:

צד client – בשפת Angular – בטכנולוגית web API

צד Server – BLL – business Logic – חישובים לוגיים, מקשר בין ה-Web ל–Dal- שליפה והכנסה של הנתונים ל – DB. נקראת ע''י ה BLL.

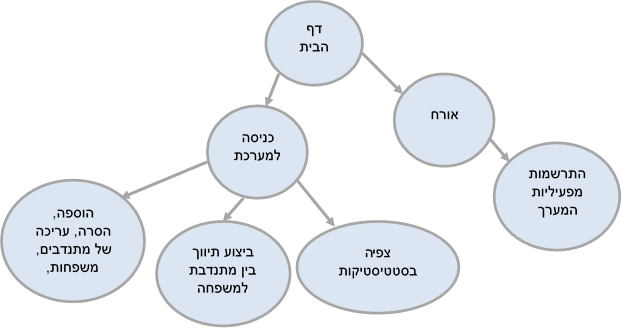
1. **שימוש במבני נתונים וארגון קבצים**

במערכת יעשה שימוש במבנה נתונים המתאים לתכנות מונחה עצמים, מחלקות ואובייקטים.

הנתונים יאוכסנו בשיטת Entity FrameWork ב – DB SQL.

בכל פרק זמן מסוים יתקיים תהליך של גיבוי אוטומטי לקבצים הקיימים במערכת.

1. **תרשימים**

****

1. **תיאור המרכיב האלגוריתמי- חישובי**

* בנית מסד הנתונים ב SQL.
* בנית האלגוריתמים עם החישובים הנדרשים והפונקציות המתאימות ב C#.
* בנית שכבת ה BL הכוללת את כל הפונקציות הלוגיות והשרותים הנדרשים בשפת C#.
* בנית שכבת ה DAL – הגישה לנתונים והפעלת המערכת עם נתונים מה – DB.
* חיבור של כל השכבות בטכנולוגיית web API לדפי html הכתובים ב – Angular7.

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

\* אבטחת מידע של מסד הנתונים כמערכת קבצים (NTFS).

\* כניסת מנהלת המערך – באמצעות סיסמאות – לצורך מעקב ואבטחה.

1. **משאבים נדרשים לפרויקט**
   1. **מספר שעות נדרשות לפרויקט**: 720 שעות.
   2. **ציוד נדרש:** מחשב ואינטרנט.
   3. **תוכנות נדרשות**: visual studio, visual code, sql
   4. **ספרות ומקורות מידע**:

[https://material.angular.io](https://material.angular.io/)

<https://github.com/>

<https://docs.angularjs.org/guide>

[https://www.w3schools.com/angular](https://www.w3schools.com/angular/)

<http://stackoverflow.com/questions>

<https://docs.angularjs.org/api>

<http://getbootstrap.com/getting-started>

1. **תכנית עבודה ושלבים למימוש הפרויקט**

* אפיון ותכנון.
* כתיבת צד לקוח, כתיבת צד שרת ועיצוב צד לקוח.
* בדיקות של התוכנה.
* כתיבת ספר פרויקט.

**14. תכנון הבדיקות שיבוצעו**

* בדיקות תוכניתן – בדיקות שמבוצעות ע''י המפתח, לדמות סימולציות של input ו output.
* בדיקות יחידה – בדיקת פונקציונליות של מודל בודד (מסך / דו''ח)
* בדיקות אינטרגרציה – בדיקת פנוקציונליות של שילוב מודלים בתת מערכת.
* בדיקות ממשקים – בדיקת ממשקים בין תת המערכת לתת מערכת אחרת או למערכת חיצונית.
* בדיקות קבלה – ביצוע בדיקות מהיבט הלקוח – הלקוח בודק ומאשר שקיבל את המערכת לשביעות רצונו.

**15. בקרת גרסאות**

שימוש בכלי בקרת תצורה כדי לאפשר ביצוע שינויים, ושמירת כל הגרסאות האפשריות.

שלמי תודה

להנהלת "מכון בית יעקב למורות" על ההשקעה במשאבים, על חשיבה נכונה והתאמצות להגיע למצויינות.

למרכזת מגמת מדעי המחשב הגב' טובה הרשלר תחי', על סבלנות, התחשבות והענות לכל בקשה ויוזמה. על היחס החם, האוזן הקשבת בכל עת, העזרה, העידוד והתמיכה לכל אורך שנות הלימוד, וכמובן, מעל הכל על בחירת הצוות והמרצים, על עריכת תכניות לימודים מתוכננות ובנויות כהלכה ואוירה נעימה ונינוחה.

לכל צוות המורות והמרצים במגמה, על המקצועיות, האנושיות ורמת הלימוד המרשימה.

למנחת הפרויקט הגב' שולמית ברלין על העזרה, הייעוץ, ההכוונה, התכנון בסבלנות וביעילות ועל הסיוע בכל בעיה שהיא לכל אורך הדרך. תודה רבה!

לגב' אסתר ביקסר– מנהלת מערך המתנדבים במנהל קהילתי מעלה רמות- ששלנו משלה הוא. תודה על יחס אדיב וסובלני, על המידע, המקצועיות, הנעימות, ההסבר הברור אודות מערך המתנדבים, דרישות הפרויקט, מתן תשובות לשאלות שעלו במהלך הפיתוח, אורך רוח וסבלנות עד שזכינו להגיע למוצר המושלם.

ומעל הכל לריבונו של עולם שחונן אותנו בדעה, בינה והשכל ובכוח להתקדם.

וכמובן שלא נשכח להודות למשפחותינו היקרות שהיו חלק בלתי נפרד מה"פרויקט", העלו רעיונות, פסלו ועודדו, שבחו והגיהו, תרמו מניסיונם וחוכמתם עד לשלב הגמר.

מבוא

מבוא

המנהל הקהילתי שבשכונת רמות – השכונה הגדולה ביותר בירושלים מתפעלת מערך מתנדבים רחב היקף, שנועד לתת מענה של עזרה וסיוע למשפחות נזקקות מתוך הקהילה.

מערך הסיוע כולל שליחת מתנדבות לבתים בהם עקרת הבית אינה יכולה לנהלו כראוי, בהם גם בתי יולדות ונשים קשות יום. כמו כן ניתן תייוך ועזרה בפניה לארגוני חסד מוכרים, וכן מתקיימים אירועים ופרויקטים מיוחדים – מכירות חסד של בגדים, סיוע מיוחד בערב פסח וכו'.

המערך התנהל עד עתה בצורה ידנית – שימוש בקלסרים, מחברות ודפים לצורך אחסון המידע, וההתאמות נעשו על בסיס הכרות אישית בלבד. כמו כן לא נשמר מידע מספק על היסטוריות של התנדבויות, ומעקב אחר משפחה ממסוימת או מתנדבת הצריך חיפוש נרחב ברישומים.

מחשוב מערך ההתנדבות יאפשר למנהלת המערך הוספה, עריכה והסרה של מתנדבים, משפחות, ארגונים וארועים בצורה ידידותית ופשוטה.

כמו כן הוא ממקד את ההתאמה עפ"י קטגוריות מובנות שניתן להוסיף עליהם קטגוריות מותאמות אישית.

כמו כן מוצג מידע מעודכן וממוקד על היסטוריות של התנדבויות והעזרויות במערך.

הוספנו גם מערכת חכמה של התראות שתפקידה להזכיר לאחראית על משפחות שטרם ניתן להן מענה, על מתנדבות פנויות, אירועים קרובים וכו.

במהלך כתיבת התוכנית צברנו ניסיון בשפות החשובות בעולם התוכנה, הכרנו טכנולוגיות חדשניות ולמדנו לעומק את הנושאים החמים בשוק כמו תקשורת שרת-לקוח, אבטחה, ניהול מסד נתונים, התממשקות לאתרים שונים ועוד.

השקענו רבות בפרויקט. עמדנו מול אתגרים ומצאנו להם פתרונות יצירתיים, חתרנו לכתיבת קוד חזק, יעיל ונקי ובעיקר – העשרנו את ידיעותינו בכל נושאי ה-design והפיתוח. בסיעתא דשמיא הגשמנו את המטרות שהצבנו לעצמנו ועל כך תודתנו נתונה. כולנו תקווה שגם אתם תיהנו מהשימוש במערכת ותפיקו ממנה תועלת.

לשאלות או הערות על ספר זה ניתן לפנות אל:

פנינה בולבין [pnini489@gmail.com](mailto:pnini489@gmail.com) 052-714-3489

בת שבע ריץ [batsheva7424@gmail.com](mailto:batsheva7424@gmail.com) 050-418-0122

הגדרת דרישות ותיאור כללי

1. הגדרת דרישות ותיאור כללי

# 1.1 תיאור כללי:

ניהול מערך מתנדבים, הוספה עריכה והסרה של מתנדבים, נעזרים והתנדבויות בצורה ידידותית וקלה, שימוש בהתאמות חכמות עפ"י קטגוריות, הצגת מידע על היסטוריות התנדבויות ומתן התראות על צורך בתיאום.

## 1.1.1 מטרות המערכת

השימוש במערכת עונה על צורך חיוני של משתמשים באתרים השונים:

* מחשוב ואחסון המידע אודות המתנדבים, הנעזרים והתיאום ביניהם
* ביצוע התאמות חכמות עפ"י קטגוריות
* הצגת מידע על התנדבויות שבוצעו בעבר בצורה רחבה וממצה
* הצגת התראות ותזכורות למענה למשפחות ועל אירועים קרובים ופרויקטים מיוחדים

## 1.1.2 היקף הפרויקט

700 שעות.

(350 שעות לכל אחת).

## 1.1.3 מבנה המערכת

שרת המנהל את ה-Database המכיל את נתונים המתנדבים, המשפחות הנעזרות, הארגונים המסייעים והארועים שמתבצעים.

צד לקוח המקשר בין האתר לשרת, מקבל את דרישת המשתמש, שולח בקשות לשרת ומאזין לו.

ממשק המשתמש: אתר שמאפשר שליטה ובקרה על כל מערך המתנדבים.

Sql Server Database

שרת

לקוח- אנגולר/ אלקטרון

## 1.1.4 משימות המערכת

### 1.1.4.1 אתר

* אפשור עריכה, הוספה והסרה של נתונים
* הצגת הצעות מתאימות להתנדבויות בהתבסס על קטגוריות
* הצגת התראות חכמות

### 1.1.4.2 שרת

* ניהול Database של מתנדבים, משפחות ארגונים וארועים המשתתפים במערך ההתנדבות.
* האזנה וביצוע בקשות הלקוח לשמירה, שליפה או עדכון של הנתונים.
* שליחת הצעות חכמות להתנדבויות.
* שליחת התראות לגבי ההתנדבויות ואירועים קרובים

# 1.2 תיאור חומרת המערכת

## 1.2.1 כללי

* שרת אינטרנטי

## 1.2.2 מרכיבי המערכת

* מחשב תואם – IBM PC.
* זיכרון פנימי
* כונן קשיח

# 1.3 תיאור תוכנת המערכת

## 1.3.1 כללי

* מערכת הפעלה windows 7 ומעלה.
* דפדפן אינטרנט

## 1.3.2 כלי התוכנה לפיתוח המערכת

* שפות תכנות c#, TypeScript , css, HTML, JQuery.
* Database SqlServer.
* Visual studio code
* Microsoft Visual Studio 2019.
* שימוש בכלי הדיבוג של google chrome

לצורך כתיבת צד השרת בחרנו להשתמש בשפת c#- שפה מונחית עצמים, בעלת יכולות המתאימות לפרויקט זה.

היתרון המשמעותי בשימוש בC# הוא ריבוי ספריות עשירות בנושאים של שליפת מידע וקישור ל Database.

## 1.3.3 טכנולוגיות בשימוש:

* *webAPI*-

​ ממשק תכנות יישומים לשרתי אינטרנט ו/או לדפדפני אינטרנט. מטרתו– לתווך בין צד הלקוח (client) לצד השרת (server) על מנת לבצע את הפונקציונליות הנדרשת לצד הלקוח.

עיקר תפקודו הוא בקבלת נתונים מצד הלקוח והעברתם לשרת, ולהפך.

LINQ- שיטת כתיבה חדשנית מובנית ב.NET מיועדת לריצה על Lists – אוספים של נתונים ומאפשרת לבצע שאילתות בצורה פשטנית, חכמה וקלילה.

* *Entity framework –*

טכנולוגית עבודה מתקדמת מול DB. יצירת אובייקטים ומיפוי מול טבלאות ה-DB, כאשר אופן העבודה ועיבוד הנתונים בשפת LINQ המוזכרת לעיל.

* *Angular10-*

טכנולוגיה מתקדמת המאפשרת פיתוח מקצה לקצה בצד הלקוח, כוללת התנהגויות שונות שהתבצעו בצד השרת והועברו לביצוע בצד הלקוח, בצורה מהירה וקלה יותר (Injectable, services וכו').

Angular תומכת ב-design pattern של single page. אנגולר נכתבת בשפת TypeScript - שפה חדשה המבוססת על JavaScript – עם מאפיינים של OOP וטיפוסיות של האובייקטים.

* *Angular material, bootstrap –*

שימוש בספריות מוכנות לעיצוב ובניה חכמה של הקומפוננטות בצד הקליינט.

* *SqlServer Database*

כמו כן, השתמשנו במסד נתונים SqlServer המתקשר בצורה קלה עם שפת C#, אמין ומאובטח.

* *Electron-*

ממיר אתר הכתוב בטכנולוגית web לאפליקציה שולחנית.

# 1.4 תיאור פונקציות המערכת

## 1.4.1 צד לקוח

### 1.4.1.1 פונקציות CRUD

* פונקציה להוספת מתנדבת/משפחה/ארגון/ארוע (להלן "מרכיבי התנדבות")
* פונקציית עריכת מרכיב התנדבות קיים.
* פונקציה מחיקת רכיב התנדבות.
* פונקציה לצפיה בנתונים קיימים, הסטוריות התנדבויות שבוצעו.

### 1.4.1.2 פונקציות לוגיות

* פונקציה למציאת התנדבות מתאימה ע"פ קטגוריה.
* פונקציות חיפוש מתקדם בנתונים.
* פונקציות התראות על פעולות נדרשות, תזכורת על ארועים צפויים.
* הצגת סטטיסטיקות על הנתונים.

### 1.4.1.3 פונקציות תקשורת עם השרת

* פונקציות לקבלת הנתונים מהמערכת
* פונקציות להוספת נתונים למערכת.
* פונקציות לעדכון נתונים במערכת.
* פונקציות להסרת נתונים מהמערכת.
* פונקציות לקבלת התראות חכמות.
* פונקציות חיפוש מתקדמות.
* פונקציות לקבלת סטטיסטיקות.

### 1.4.1.4 פונקציות יבוא ויצוא נתונים

* פונקצית יבוא מאקסל.
* פונקצית יצוא לאקסל.

## 1.4.2 שרת

### 1.4.2.1 פונקציות ניהול מאגר נתונים

* יצירת Database על השרת בפעם הראשונה של הפעלת המערכת.
* יצירת טבלאות מתאימות לבסיס הנתונים
* יצירת קלאסים מקבילים לבסיס הנתונים לניהול פונקציונליות
* פונקציה להוספת רשומה של מרכיבי ההתנדבות
* פונקציה לעדכון מרכיבי ההתנדבות
* פונקציה לשליפה של מרכיבי ההתנדבות.
* פונקציה למחיקה של מרכיבי ההתנדבות.

#### 1.4.2.2. פונקציות תקשורת שרת - לקוח

* פונקציה המאזינה לבקשות לקוח
* פונקציה לפענוח בקשות הלקוח והמרתן לאובייקט מערכת
* פונקציה המחזירה את התוצאות ללקוח
* פונקציה להמרה בין הפורמטים השונים של נתונים(json, JArray string ועוד)

# 1.5 זרימת המידע במערכת

## 1.5.1 זרימת המידע - כללי:

המערכת מורכבת ממסד נתונים, שרת ואתר/אפליקצית אלקטרון. כל זרימת המידע במערכת היא הליך חוצה של אתר-שרת-מסד כך שהשרת מקשר בין המסד לאתר ומאפשר את ביצוע הפונקציונאליות הנדרשת.

## 1.5.2 תאור תהליך הזרימה:

### 1.5.2.1 כניסה ראשונית לאתר והעלאת קובץ אקסל לצורך אתחול נתונים:

בעת כניסה לאתר בפעם הראשונה מנהלת מערך ההתנדבות שעבורה מיועד האתר יכולה להתחיל בתהליך מחשוב הנתונים הקיימים (בניירת) ע"י העלאה של קובץ אקסל או ע"י הוספה ידנית באמצעות הטופס הקיים באתר.

לאחר ההוספה של הנתונים יש אפשרות להוסיף קטגוריות מתאימות לכל מרכיב התנדבותי ע"מ ליעל את תהליך ההתאמה וכן שמירת מידע ממוקד יותר לגבי ההתנדבות.

### 1.5.2.2 פעילות שוטפת באתר:

המנהלת יכולה לתאם התנדבות בין משפחה למתנדבת, לקשר בין ארגון למשפחה, לקבל מידע עדכני אודות ההתנדבויות האחרונות שבוצעו, לערוך פרטים של מרכיב ההתנדבות בקלות, להוריד קובץ אקסל עם הנתונים לצורך שימוש במערכות אחרות.

בכל כניסה חדשה לאתר מתבצעת טעינה של התראות חכמות שמזכירות אודות משפחות שטרם נמצא להן מתנדבת וכן אודות ארועים קרובים.

כמו כן ניתן להעזר במערכת לצורך תאום מהיר בין מרכיבי התנדבות ע"פ קטגוריה ולבצע חיפושים מיוחדים לצרכי בקרה ותפעול.

## 1.5.2 תרשים תהליכים – User Cases:

### 1.5.2.1 כניסת מנהלת מערך המתנדבים לאתר:

**שליחת הנתונים המעודכנים לשרת ושמירתם, עדכון הטבלאות במסד הנתונים, במקרה של שגיאה בקרת שגיאות.**

**כניסה לדף הבית**

**מעבר לפעולות נדרשות ע"פ ההתראות (הוספת מתנדבת למשפחה...)**

**מעבר לדף סטטיסטיקות**

**שליחת בקשה לשרת לקבלת התראות**

**הצגת ההתראות**

**כניסה לדף הכולל רשימת מרכיב התנדבותי.**

**עריכת הנתונים (CRUD)**

**שליחת בקשה לשרת לקבלת הנתונים והצגתם.**

**כניסה לדף ההתנדבויות**

**שליחת בקשה לקבלת ההתנדבויות והצגתם.**

ממשקים חיצוניים

2. ממשקים חיצוניים

מערכת זו אינה זקוקה לממשקים חיצוניים. כל הלוגיקה והשימוש בה הוא על ידי שימוש בספריות קוד בשפות השונות.

ממשק אדם - מכונה

3. ממשק אדם מכונה

# **3.1 כללי:**

## **3.1.1 עקרונות בניית הממשק:**

ממשק המשתמש הוא ממשק גרפי המותאם למשתמש הפשוט. המערכת ברורה מאד ונוחה לתפעול וכוללת מסכים ברורים, הוראות והודעות שגיאה מובנות. המערכת מאוד אינטואטיבית, כוללת שימוש באייקונים ויזואלים והודעות קופצות, מודלים וטפסים נטענים.

שם דגש על צבעוניות, נוחות, שימוש ב-loader ואנימציות בעת טעינה של מסכים ונתונים והודעות שגיאה ממוקדות.

המעבר בין המסכים מתבצע בצורה קלילה ומהירה, גם התקשורת עם השרת מהירה במיוחד, ותורמת לחוית המשתמש.

## **3.1.2 הממשק הגרפי כולל:**

* אתר/אפליקצית אלקטרון הכולל דף בית, תפריט ובו ניווט לדפים נוספים.
* מסכים המציגים למשתמש אפשרויות עריכה של נתונים, צפיה בנתונים קודמים, הוספת נתונים חדשים, הסרת נתונים קיימים, צפיה בסטטיסטיקות והתראות.
* הודעות שגיאה המוצגות למשתמש, מודיעות לו על שגיאה ומנחות אותו לפתרון.
* סימון גרפי של שדות גרפים שאינם תקינים, תצוגה ויזואלית של כפתורים זמינים ושאינם.
* מודלים ומסכים קופצים לחוית משתמש ידידותית.
* שימוש באייקונים גרפיים במקום טקסט לממשק משתמש ידידותי יותר.

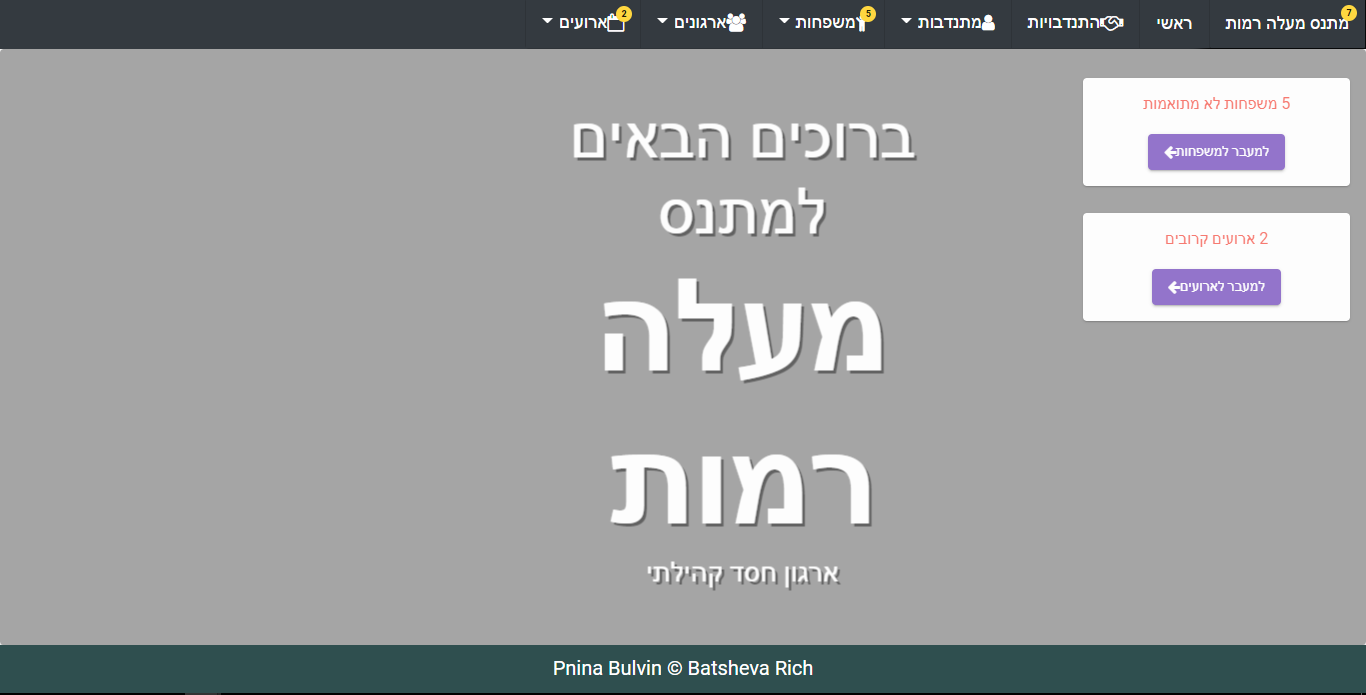


# תיאור מסכים:

המסכים במערכת מתחלקים ל:

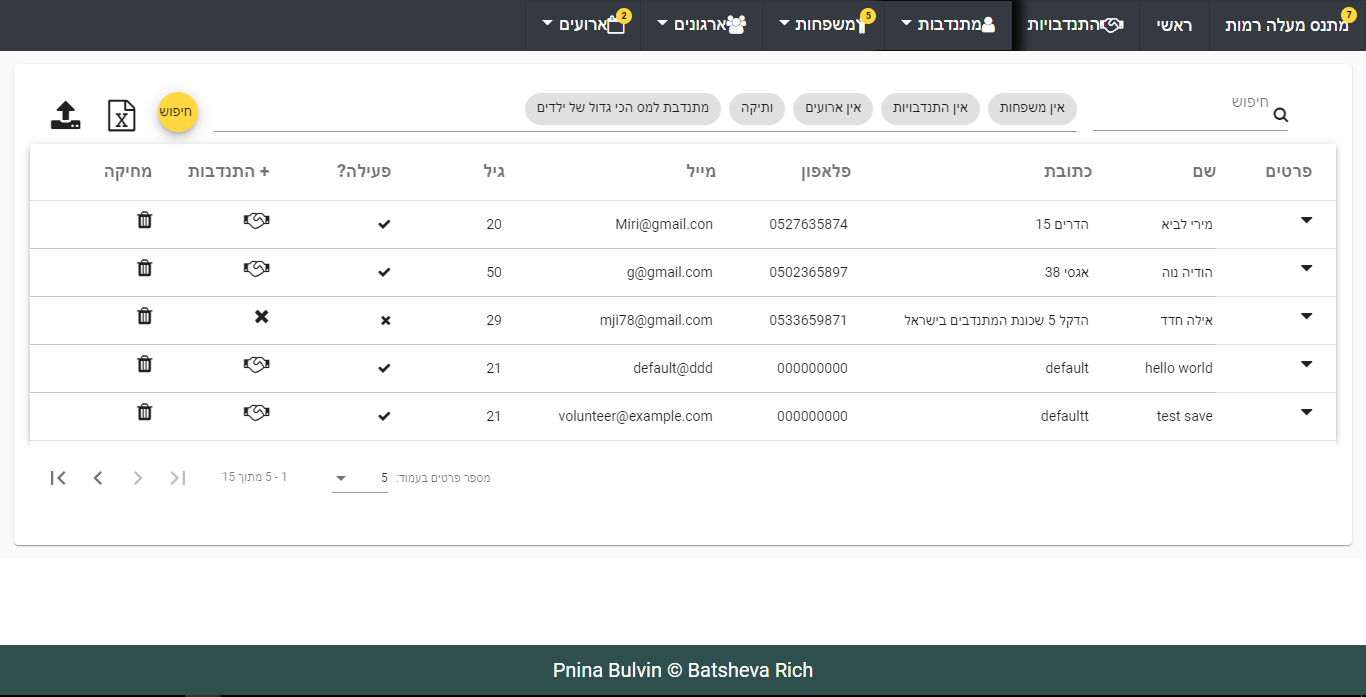
## דף בית:

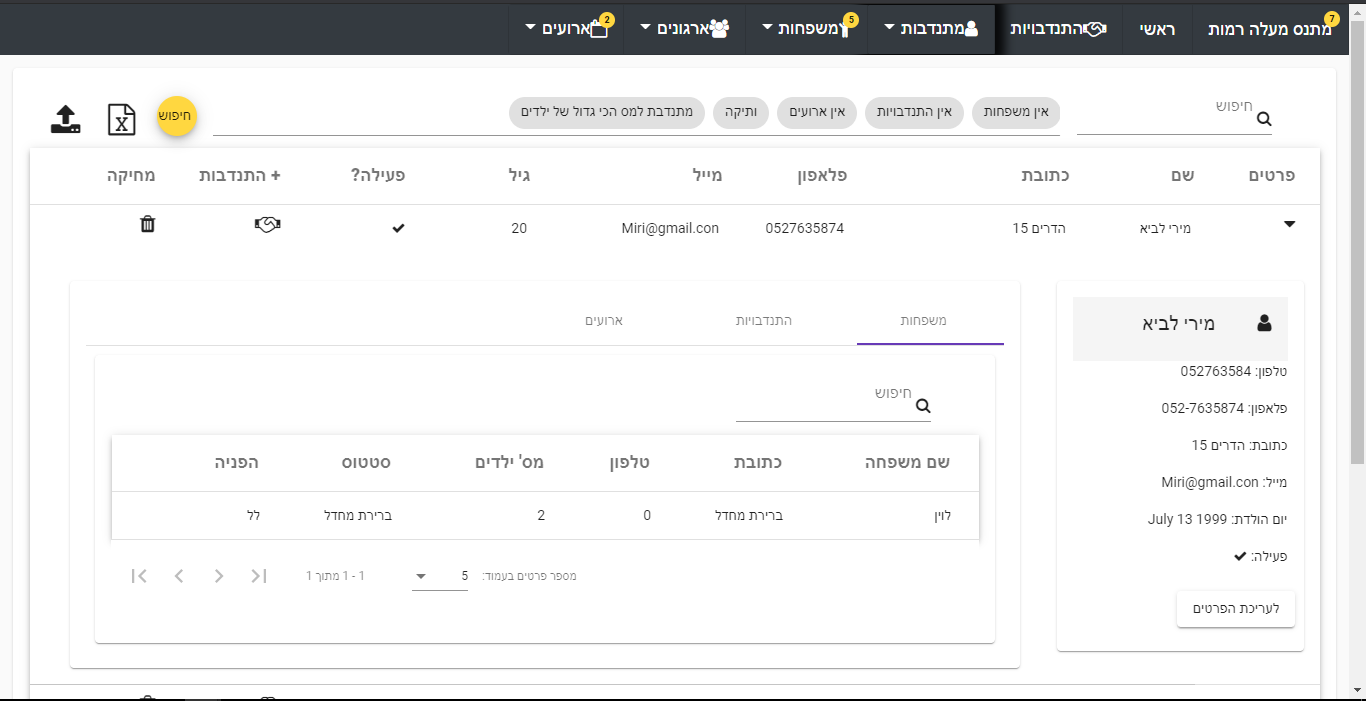
הכולל בתוכו התראות על משפחות שטרם נמצא להן מתנדבת וכן ארועים קרובים.



## מסכי צפיה בנתוני מרכיבי ההתנדבות:

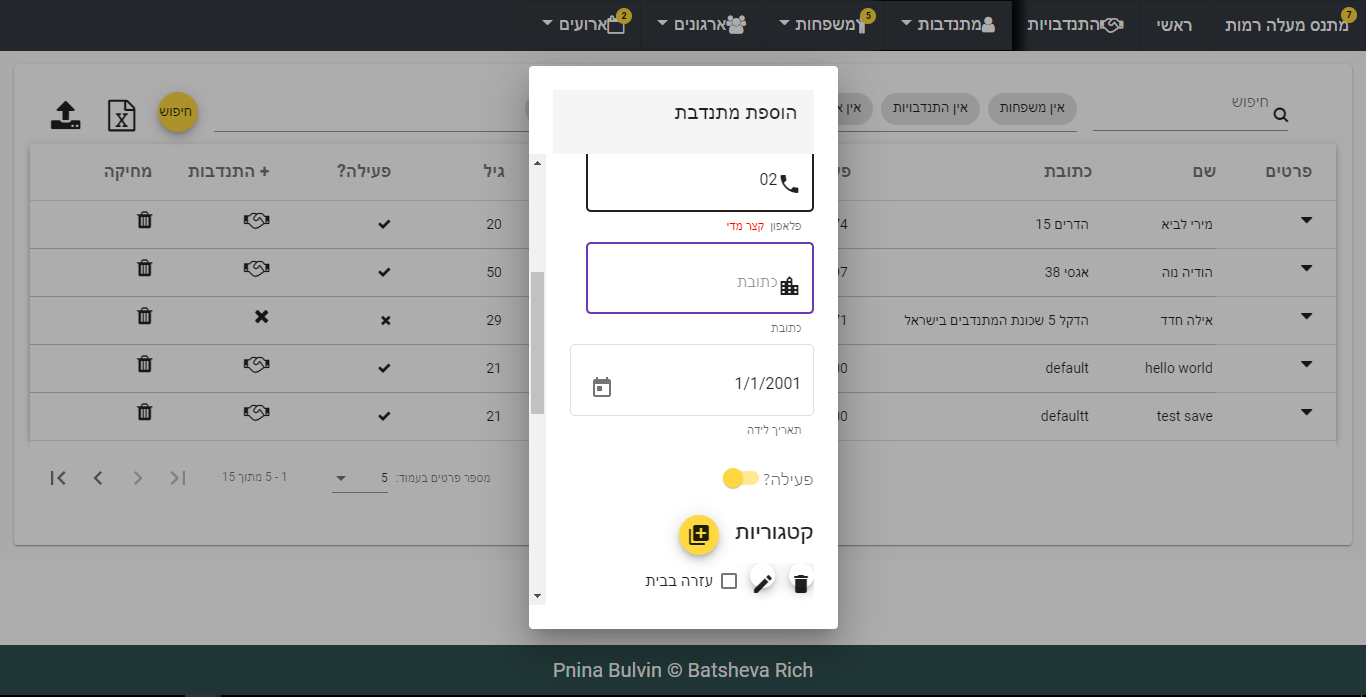
* כל מסך כולל טבלה מרכזית ובה נתוני המרכיב, בלחיצה על שורה בטבלה נפתח מסך שכולל נתונים הניתנים לעריכה, וכן צפיה בנתוני התנדבות קודמים מסווגים לפי מרכיבי התנדבות אחרים.



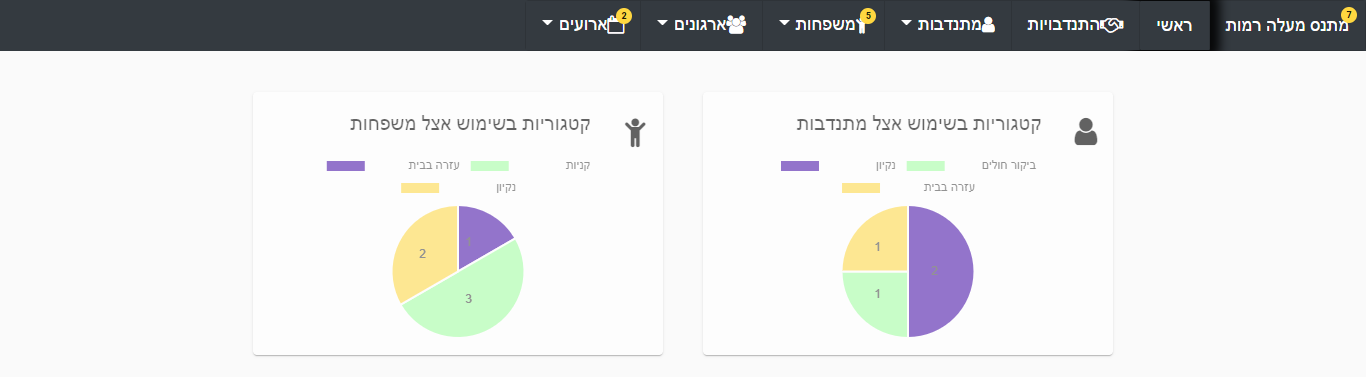


המסך כולל אפשרויות חיפוש וחיפוש מתקדם וכן פונקציות יצוא ויבוא הנתונים לאקסל.

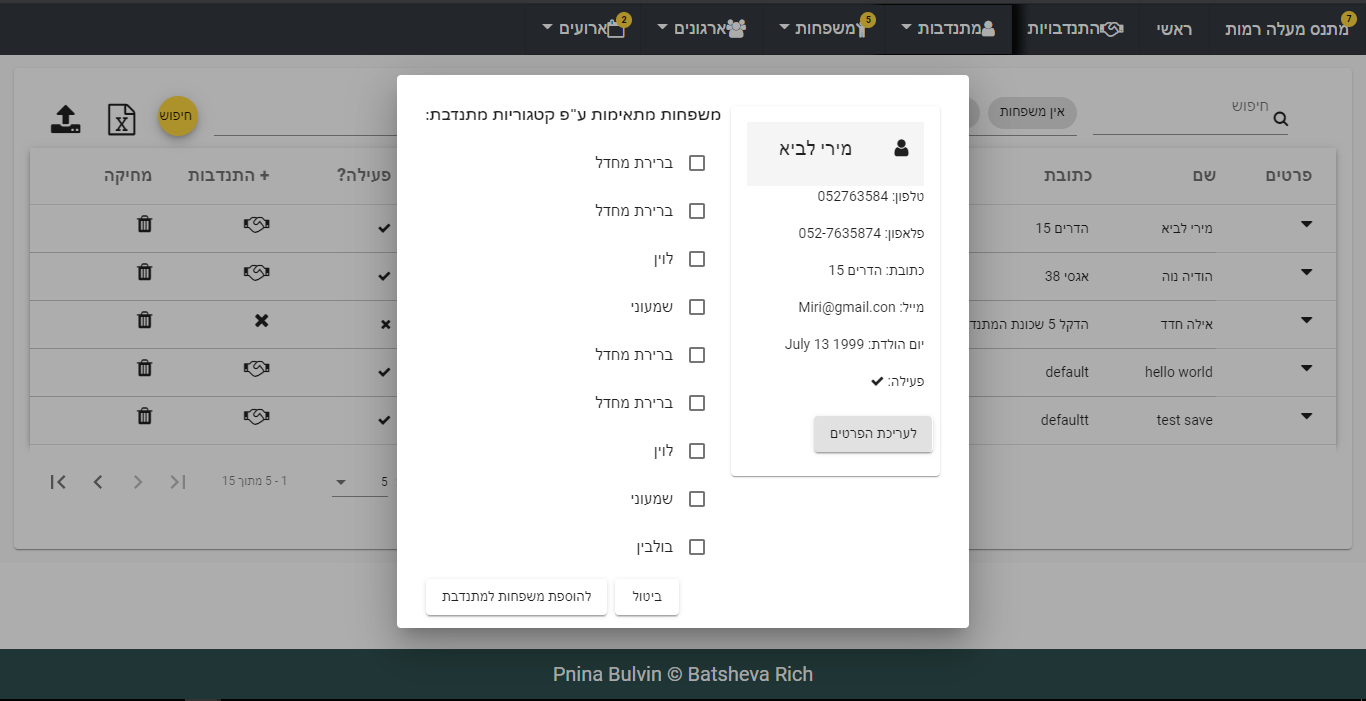
* מסך הוספה של נתונים כולל אייקונים ידידותיים לכל שדה וכן בדיקות מידע עם אינדיקציה למשתמש לגבי תקינות הנתונים. כמו כן, הקטגוריות ניתנות להוספה בצורה אינטואטיבית.



* מסך סטטיסטיקות בו ניתן לקבל מידע שימושי אודות נתונים כלליים של התנדבויות וקטגוריות.



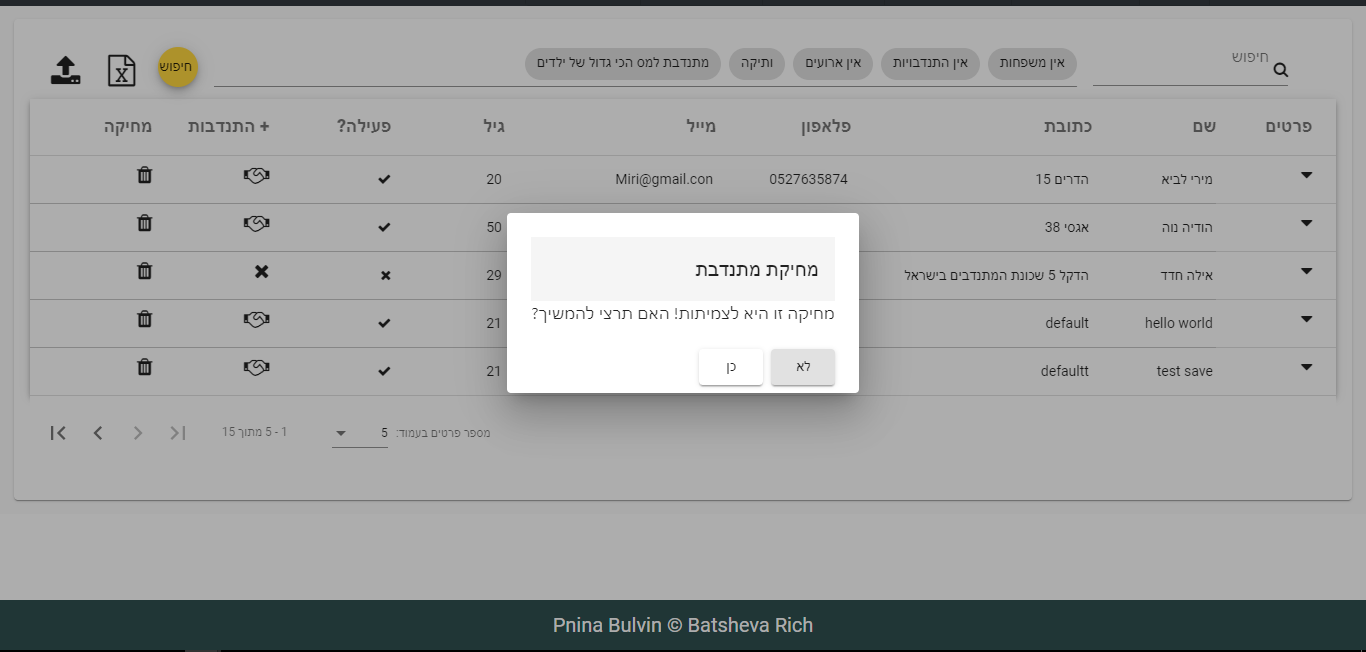
* מסך להוספת התנדבות למרכיב ספציפי ע"פ קטגוריה



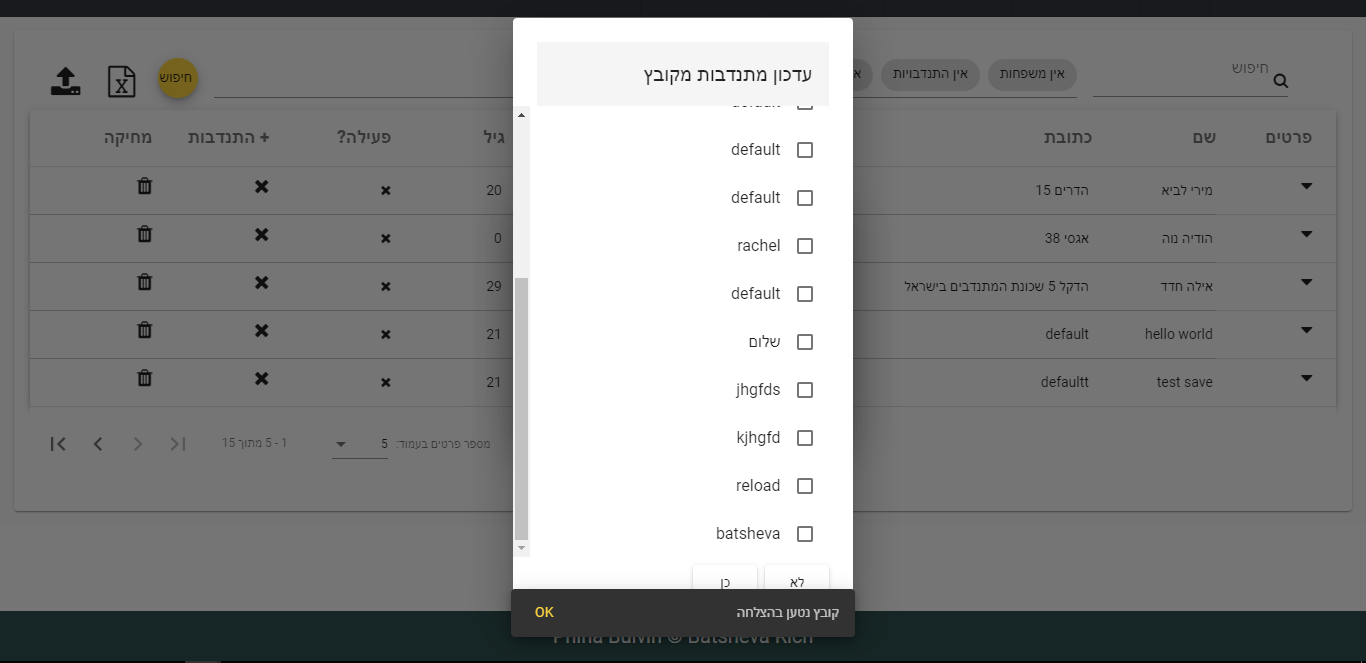


## הודעות בקרה למשתמש:

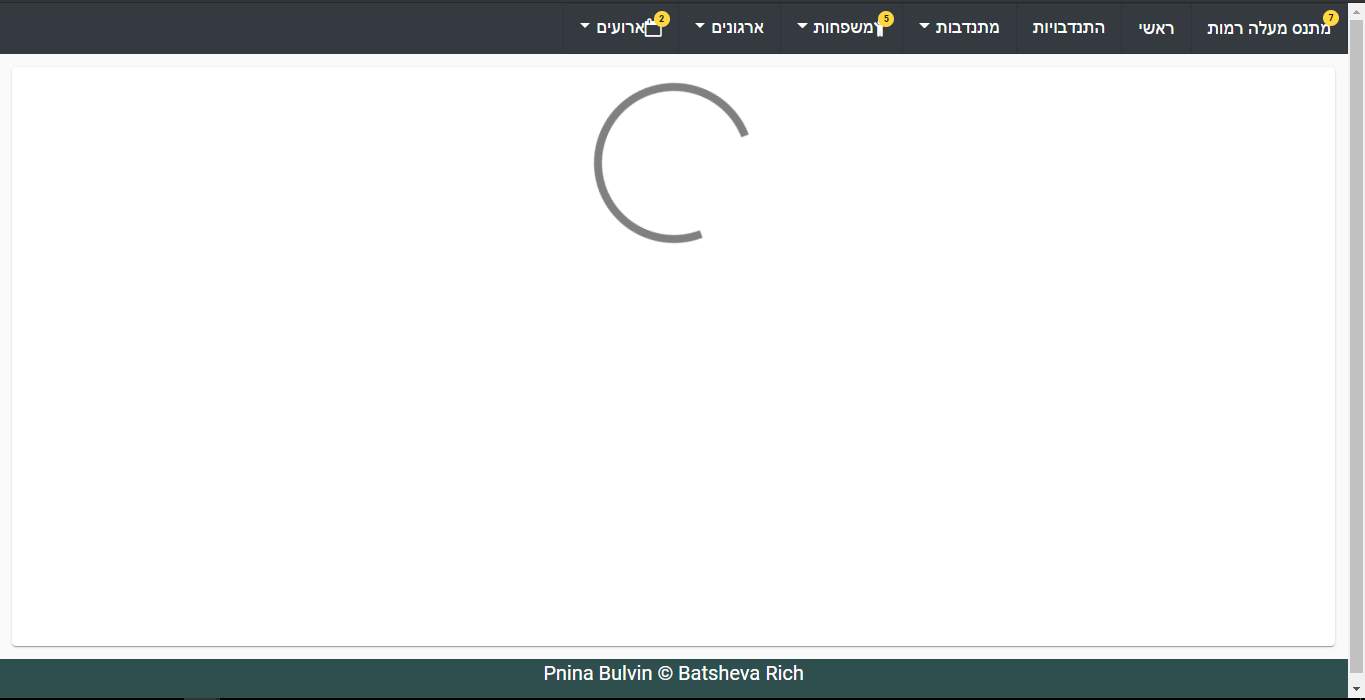
* הודעה מוצגת לפני מחיקה של מרכיב התנדבותי



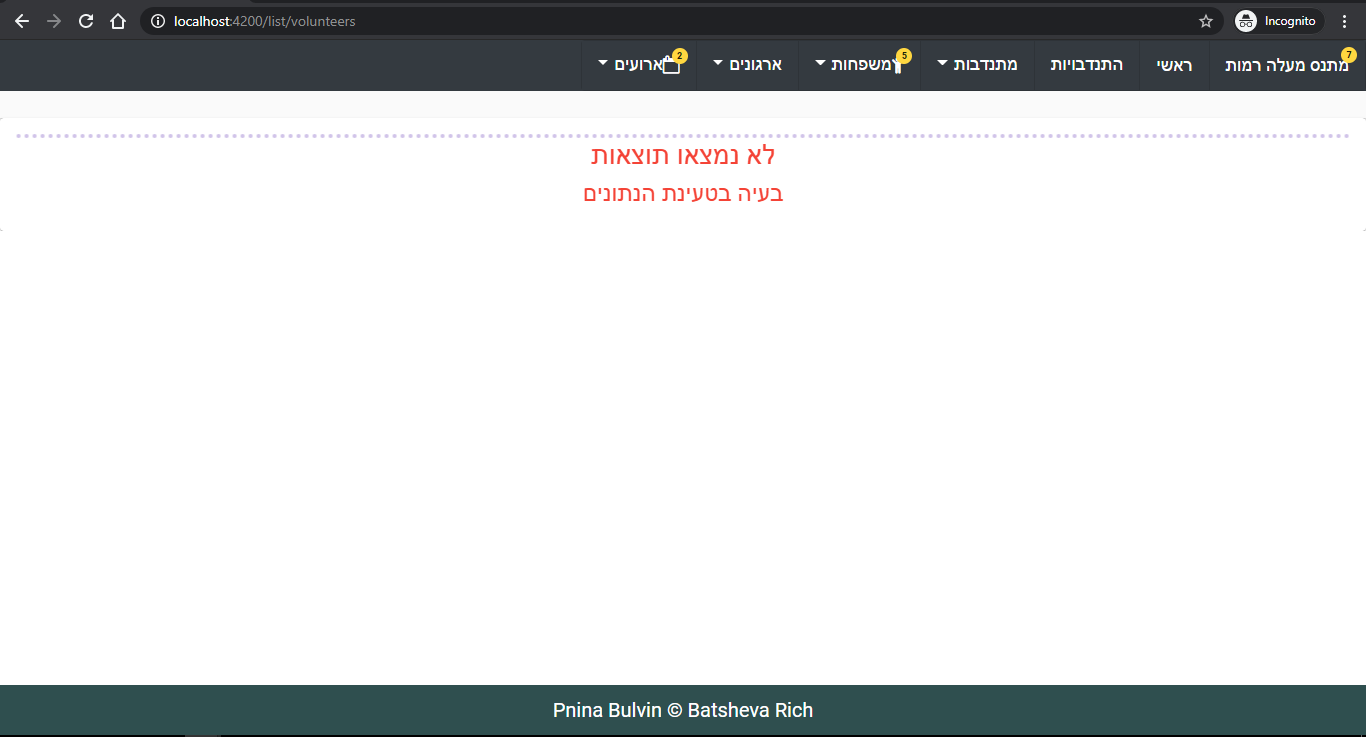
* הודעה על אישור טעינת נתונים מקובץ אקסל



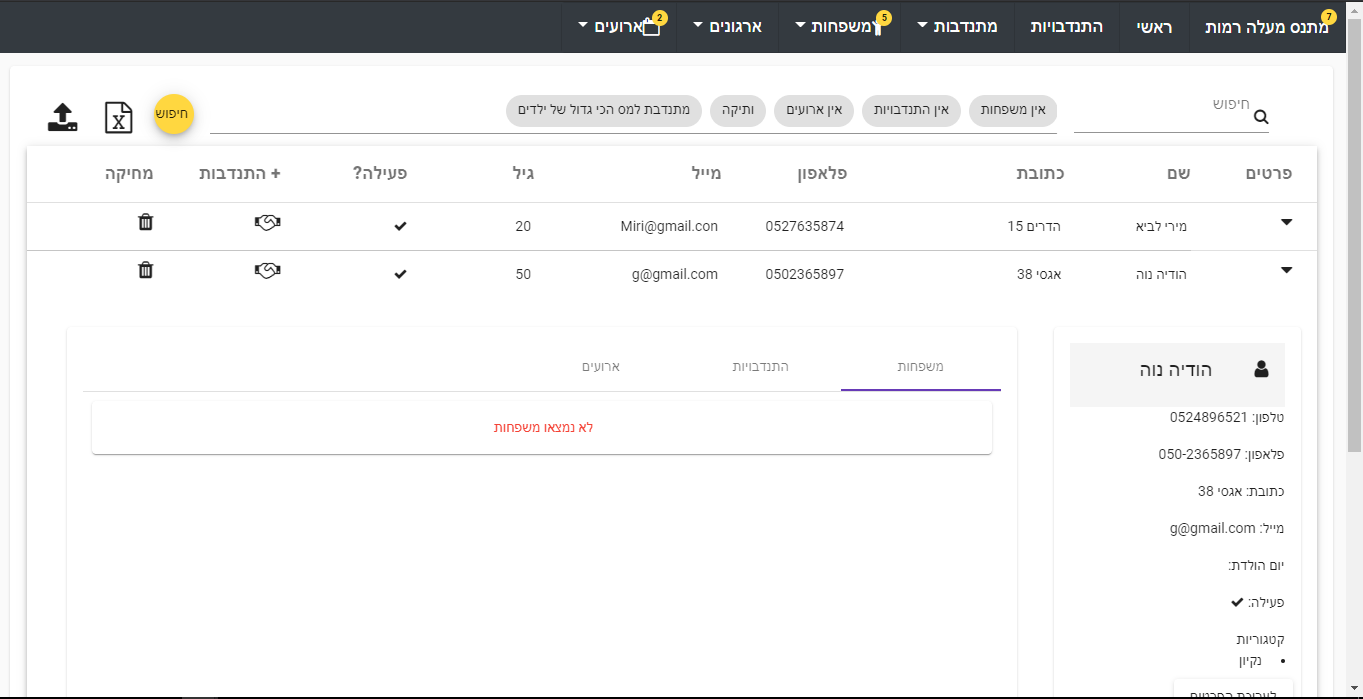
* Loader בעת טעינת הנתונים.



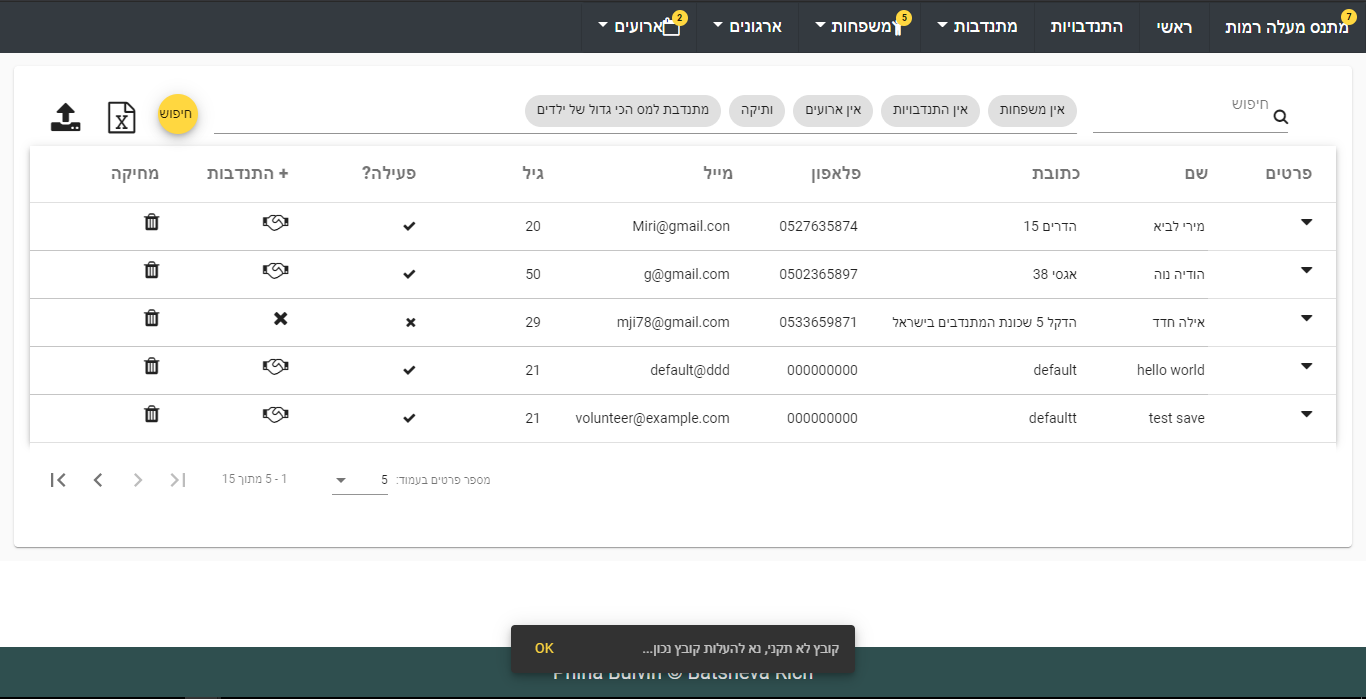
* הודעת שגיאה במקרה שהנתונים לא חזרו מהשרת.



* הודעה במקרה ולמרכיב ההתנדבותי לא נמצא התנדבות משותפת עם מרכיב אחר.



* שגיאה במקרה של טעינת קובץ אקסל לא תקינה.



מבנה נתונים וארגון קבצים

1. מבנה נתונים וארגון קבצים

## ארגון קבצים

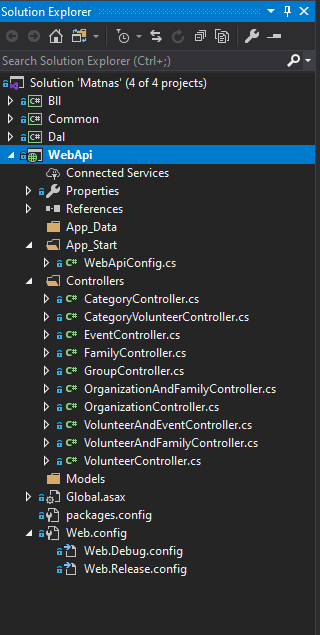
## כללי:

המערכת מורכבת משני חלקים: אתר/אפליקצית אלקטרון ושרת. האתר מנהל את ממשק המשתמש ואת האפשרויות השונות מול הדפדפן והשרת אחראי על ניהול מסד הנתונים וכל הפונקציונאליות הנלווית.

## תיאור קבצים:

### צד שרת

צד השרת כתוב בשפת C# המציעה ספריות לטיפול במסד נתונים, המרת אובייקטים, תקשורת עם לקוח ואבטחה. התכנית מחולקת לקבצים- מחלקות על פי יסודות ה-OOP. חלוקה זו מאפשרת הפרדה לוגית בין הישויות השונות בתכנית, כך שכל מחלקה מטפלת בתחום מסוים השייך לטיפולה (עקרון ה-encapsulation). כך נמנעות בעיות אופייניות לתכנות הפרוצדוראלי.

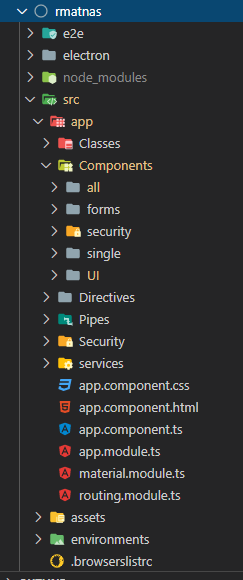


**הסבר:**

* התיקייה references מכילה קבצי dll לצורך שימוש בספריות קוד חיצוניות לתכנית.
* הקובץ web.config הוא קובץ המכיל הגדרות שונות ברמת התכנית.
* הקובץ webApi.config הוא קובץ הגדרות התחברות לצד לקוח.
* צד השרת בנוי במודל MVC- מודל השכבות. כולל
  + שכבת DAL- data access layer, המטפלת בגישה ישירה למסד הנתונים וטיפול בנתונים.
  + שכבת BLL- business layer logic, המיצגת את הדרישות העסקיות.
  + שכבת ה-Common שכוללת מחלקות של אוביקטים משותפים לשכבה העסקית ולשכבת הנתונים.
  + WebApi- ממשק המאפשר ללקוח לגשת לפונקציות השרת.

### צד לקוח

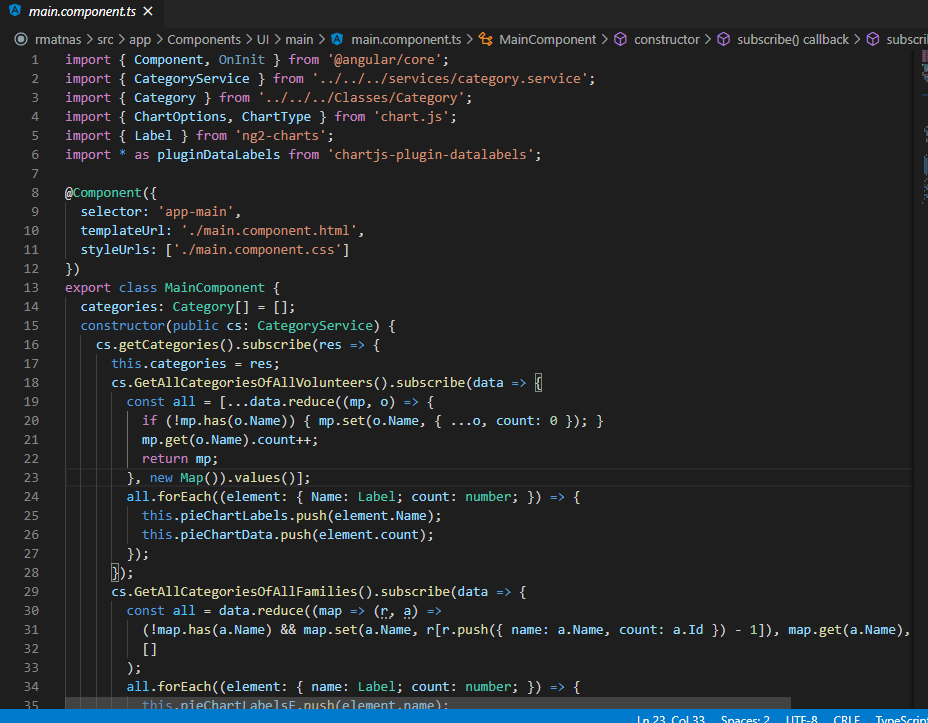
צד הלקוח כתוב באנגולר 10 (בשפת typescript), בשילוב שימוש ב-Angular Material, Bootstrap, JQuery, fontawesome, xlsx, webfontloader, ng2-charts, saturn-datepicker, flex-layout כדי לאפשר קוד גמיש, ביצועים מורכבים ויצירת UX מותאם באופן מושלם למשתמש.

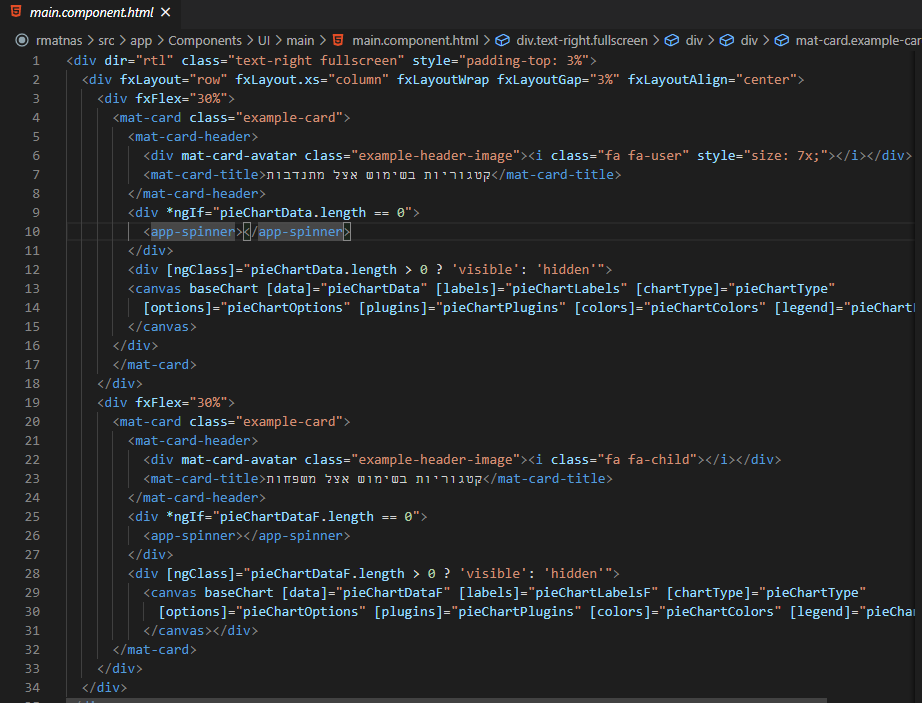


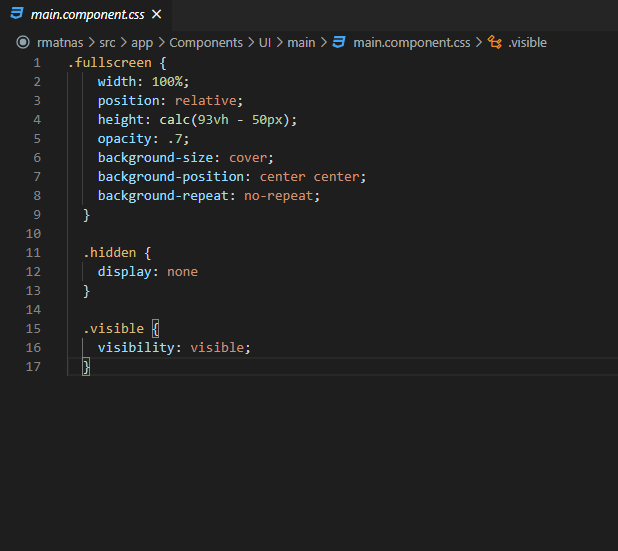
#### שימוש בקומפוננטות וחלוקה לוגית ע"פ מתודולוגיות של אנגולר:

* + חלוקה לקומפוננטות ע"פ האוביקטים השימושיים במערכת. כל קומפוננטה כוללת קובץ ts, שמכיל את הלוגיקה שמנהלת את הקומפוננטה, קובץ html שמיצג את הצד הויזואלי וקובץ css שקובע את העיצוב היחודי לכל קומפוננטה.

לדוג', בקומפוננטה main שמציגה את הסטטיסטיקות, קימים 3 קבצים כנ"ל:

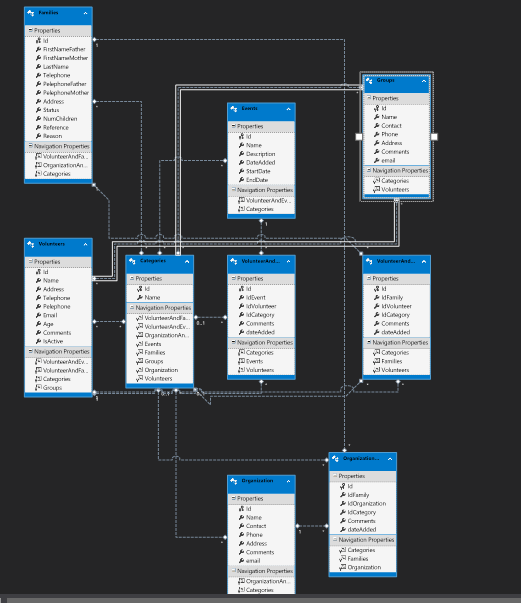






## מבנה נתונים

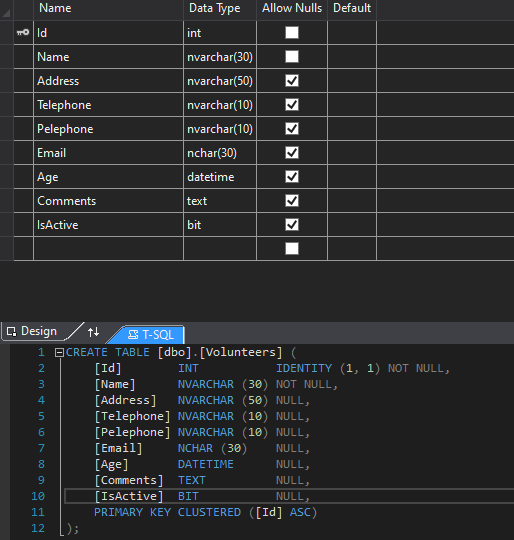
בסיס הנתונים הוא מסוג SqlServer. מסד הנתונים מכיל טבלה לכל מרכיב התנדבותי וכן טבלאות של התנדבויות בין מרכיבים. בנוסף כולל טבלה עבור הקטגוריות וטבלאות של תאום בין מרכיב לקטגוריה.



### **מ**בנה הטבלאות:

* מרכיב התנדבותי:

לדוג', טבלת מתנדבות

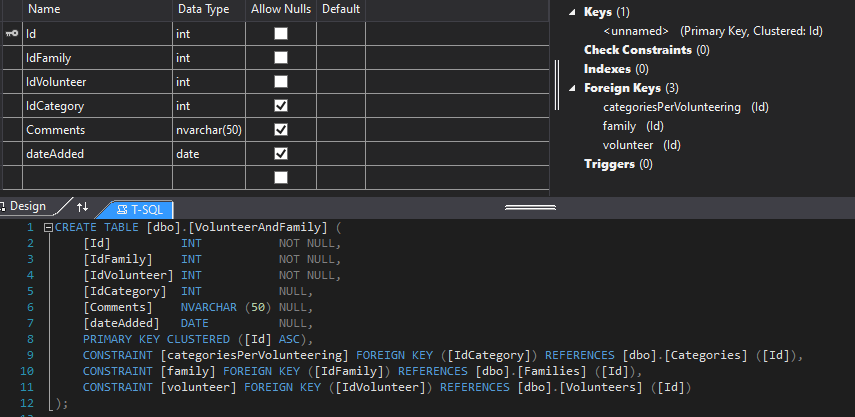


הטבלה מכילה את השדות הבאים:

* **Id**: שדה מסוג INTEGER המכיל מזהה ייחודי של השדה. שדה זה משמש כמפתח ראשי והוא מתווסף באופן אוטומטי לשדה בסדר עולה.
* **Name:** שדה מחרוזת, מיצג שם מתנדבת. שדה מנדטורי.
* **Address:** שדה מחרוזת, מיצג כתובת. שדה לא מנדטורי.
* **Telephone:** שדה מחרוזת, מיצג טלפון. שדה לא מנדטורי.
* **Pelephone:** שדה מחרוזת, מיצג פלאפון. שדה לא מנדטורי.
* **Email:** שדה מחרוזת, מיצג מייל. שדה לא מנדטורי.
* **Age:** שדה תאריך/שעה, מיצג תאריך לידה. שדה לא מנדטורי.
* **Comments:** שדה מחרוזת, מיצג הערות. שדה לא מנדטורי.
* **IsActive:** שדה בינארי, מיצג האם המתנדבת פעילה או לא. שדה לא מנדטורי.

טבלת קישור בין 2 מרכיבי התנדבות:

לדוג' טבלה בין מתנדבות למשפחות:



הטבלה מכילה את השדות הבאים:

* **Id**: שדה מסוג INTEGER המכיל מזהה ייחודי של השדה. שדה זה משמש כמפתח ראשי והוא מתווסף באופן אוטומטי לשדה בסדר עולה.
* **IdFamily:** שדה מסוג INTEGER, מיצג מפתח זר למשפחה קיימת. שדה מנדטורי.
* **IdVolunteer:** שדה מסוג INTEGER, מיצג מפתח זר למתנדבת קיימת. שדה מנדטורי.
* **IdCategory:** שדה מסוג INTEGER, מיצג מפתח זר לקטגוריה קיימת. שדה מנדטורי.
* **Comments:** שדה מחרוזת, מיצג הערות. שדה לא מנדטורי.
* **dateAdded:** שדה תאריך/שעה, מיצג תאריך התנדבות. שדה לא מנדטורי.

תכנות המערכת

1. תכנות המערכת

# תיאור כללי:

בכתיבת התוכנית ניתן דגש על כתיבת קוד יעיל, נקי ואלגנטי וניצול מרכיבים לשימוש חוזר. לאורך הדרך הקפדנו על שימוש מינימלי במשאבים חיוניים ונעזרנו בטכנולוגיות חדשניות בכדי למקסם יכולות.

### צד שרת:

צד השרת נכתב בשפת C#, שפה חזקה ובעלת אפשרויות נרחבות המתאימות למטרות הפרויקט. כדי לבצע פונקציונאליות מורכבת בפרויקט השתמשנו בספריות שונות כמוSystem.Collections, System.Linq, System.Data, System.Diagnostics, ועוד.

ג"כ התשמשנו ב-entityFramework לצורך ניהול נכון של מסד הנתונים.

כמו כן השתמשנו ב Newtonsoft למעבר חכם של אובייקטים בין השרת לאתר ובחזרה.

### צד לקוח:

צד הלקוח הוא אתר או אפליקצית אלקטרון. בכתיבת האתר נעזרנו באתר angular.io בהדרכה איך לבנות את האתר בצורה נכונה ובספריות קוד מגוונות ליישום התהליכים הנדרשים. האתר נכתב ברובו בשפת TS ברמה מתקדמת וכמובן HTML ו-CSS. נעשה שימוש בפורמט Json בהעברת האובייקטים לשרת וחזרה בצורה האופטימלית.



## עקרונות תכנות:

* תכנות מונחה עצמים – Pure OOP
* חלוקה נכונה למחלקות ולפונקציות
* תקשורת דו כיוונית בין שרת ללקוח הדורשת בניית מחלקות מקבילות לנתונים המתקבלים מהשרת
* קוד קריא, מסודר וברור

## תיאור אלגוריתמים ודוגמאות קוד

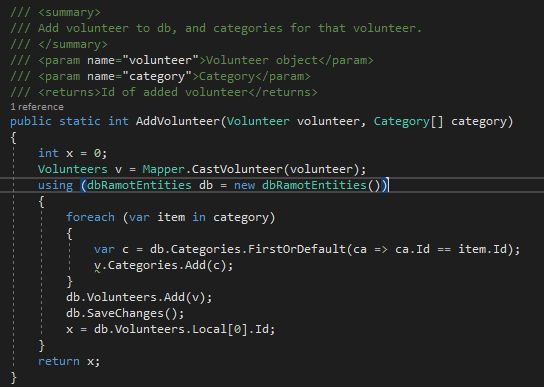
## צד שרת

## **שכבת הנתונים- DAL:** מחלקות המנהלות את הטבלאות במסד הנתונים, מבצעות שליפה, עדכון, מחיקה והוספה (CRUD).

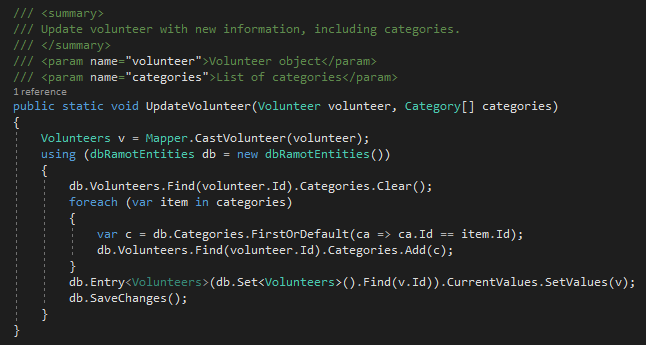
**לדוג' מחלקת VolunteerManager:**

***דוגמאות לפונקציות במחלקה:***

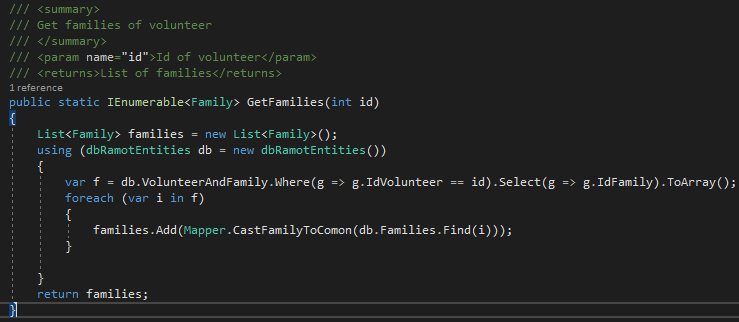
* **AddVolunteer**: פונקציה המשמשת להוספת מתנדבת חדשה למסד הנתונים. הפונקציה מקבלת כפרמטר אוביקט ובו הנתונים של המתנדבת וכן מערך קטגוריות המשויך אליה. הפונקציה מעדכנת את טבלת המתנדבות וכן את טבלת הקטגוריות למתנדבת ומחזירה את ה-id של המתנדבת שנוצרה באופן אוטומטי ע"י ה-SQL לשימוש בצד הלקוח.



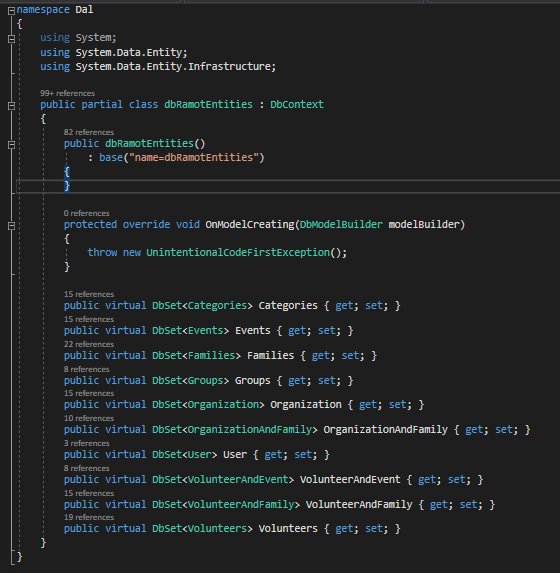
* **UpdateVolunteer**: פונקציה לעדכון נתונים של המתנדבת. הפונקציה מקבלת כפרמטר אוביקט ובו הנתונים של המתנדבת וכן מערך קטגוריות המשויך אליה. הפונקציה מעדכנת את טבלת המתנדבות וכן את טבלת הקטגוריות למתנדבת.



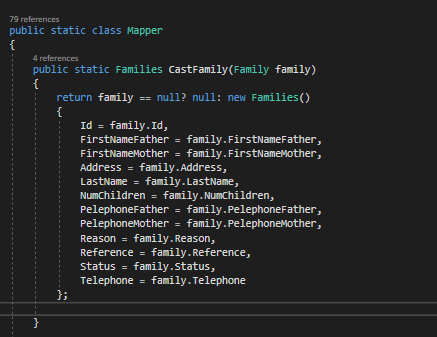
* כמו כן קיימות פונקציות עבור מחיקה ושליפה של הנתונים.
* **GetFamilies:** פונקציה המחזירה את המשפחות שהמתנדבת עבדה אצלן. מקבלת id של מתנדבת ומחזירה מערך של משפחות. במקרה ולמתנדבת אין עדין משפחות שהתנדבה אצלן, הפונקציה מחזירה מערך ריק. קימות פונקציות זהות עבור שאר מרכיבי ההתנדבות, בשינויים מתבקשים.



**דוג' למחלקה שמשמשת את Entity Framework:**

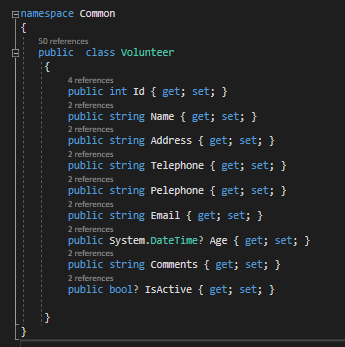
****

**Mapper: מחלקה שתפקידה להמיר אוביקטים בין ה-Common ל-DAL**

****

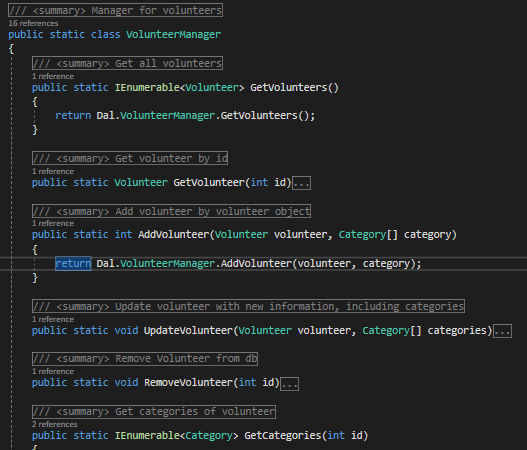
## שכבת ה-Common: מכילה מחלקות של אוביקטים שמשותפים ל-DAL ול-BLL.

***מחלקה לדוג': מחלקת Volunteer***

****

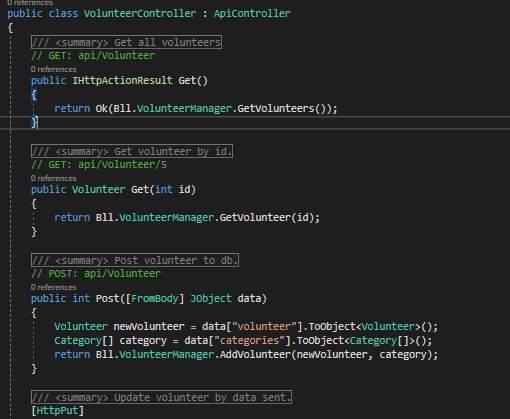
## שכבת ה-BLL: מכילה את הפונקציות העסקיות לכל מרכיב, פונקציות אלו זמינות ללקוח ע"י הפעלתן באמצעות ה-WebApi.

***מחלקה לדוג': מחלקת VolunteerManager***

******

## WebApi: מכיל את ה-controllers שבאמצעותן צד הלקוח ניגש לצד השרת.

***מחלקה לדוג': VolunteerController***

******

## 5.3.2 צד לקוח

## מודולים בשימוש:

## AppModule- המודול הראשי, מכיל את הקומפוננטות המטפלות בנתונים ובמרכיבי ה-UI.

...

import { DragDropModule } from '@angular/cdk/drag-drop';

import { ChartsModule } from 'ng2-charts';

import { MatIconRegistry } from '@angular/material/icon';

import \* as WebFont from 'webfontloader';

import { NotificationService } from './services/notification.service';

import { LoginComponent } from './Components/security/login/login.component';

WebFont.load({

  custom: { families: ['Material Icons', 'Material Icons Outline'], }

});

@NgModule({

  imports: [

    ReactiveFormsModule,

    BrowserModule,

    FormsModule,

    RoutingModule,

    BrowserAnimationsModule,

    HttpClientModule,

    FormsModule,

    MaterialModule,

    FlexLayoutModule,

    ChartsModule,

    ReactiveFormsModule,

    SatDatepickerModule,

    SatNativeDateModule,

    ScrollingModule,

    DragDropModule,

    MatBadgeModule

  ],

  declarations: [

    AppComponent,

    VolunteerComponent,

    VolunteerFComponent,

    AllVolunteersComponent,

    FamilyComponent,

    FamilyFComponent,

    AllFamiliesComponent,

    EventComponent,

    AllEventsComponent,

    EventFComponent,

    OrganizationComponent,

    OrganizationFComponent,

    AllOrganizationsComponent,

    HomeComponent,

    FooterComponent,

    ChooseCategoryComponent,

    BorderDirective,

    PhoneMaskDirective,

    PhonePipe,

    FilterPipe,

    AddVFComponent,

    AddVEComponent,

    AddVGComponent,

    AddFOComponent,

    AllToVolunteersComponent,

    ToVolunteerComponent,

    CategoryFComponent,

    ConfirmDialogComponent,

    HeaderComponent,

    MainComponent,

    RangesFooter,

    SpinnerComponent,

    ErrorComponent,

    LoginComponent

  ],

  entryComponents: [RangesFooter],

  providers: [DatePipe, { provide: MatPaginatorIntl, useValue: getHebrewPaginatorIntl() }, NotificationService],

  bootstrap: [AppComponent]

})

export class AppModule {

  constructor(matIconRegistry: MatIconRegistry) {

    matIconRegistry.registerFontClassAlias('fontawesome', 'fa');

  }

}

## MaterialModule- מודול המרכז את הקומפוננטות שנלקחות מהספריה Angular Material ומיצא אותם אל המודול המרכזי- AppModule.

import { NgModule } from '@angular/core';

import { CdkTableModule } from '@angular/cdk/table';

import { MatAutocompleteModule } from '@angular/material/autocomplete';

import { MatButtonModule } from '@angular/material/button';

import { MatButtonToggleModule } from '@angular/material/button-toggle';

import { MatCardModule } from '@angular/material/card';

import { MatCheckboxModule } from '@angular/material/checkbox';

import { MatChipsModule } from '@angular/material/chips';

import { MatNativeDateModule, MatRippleModule } from '@angular/material/core';

import { MatDatepickerModule } from '@angular/material/datepicker';

import { MatDialogModule } from '@angular/material/dialog';

import { MatExpansionModule } from '@angular/material/expansion';

import { MatFormFieldModule } from '@angular/material/form-field';

import { MatGridListModule } from '@angular/material/grid-list';

import { MatIconModule } from '@angular/material/icon';

import { MatInputModule } from '@angular/material/input';

import { MatListModule } from '@angular/material/list';

import { MatMenuModule } from '@angular/material/menu';

import { MatPaginatorModule } from '@angular/material/paginator';

import { MatProgressBarModule } from '@angular/material/progress-bar';

import { MatProgressSpinnerModule } from '@angular/material/progress-spinner';

import { MatRadioModule } from '@angular/material/radio';

import { MatSelectModule } from '@angular/material/select';

import { MatSidenavModule } from '@angular/material/sidenav';

import { MatSlideToggleModule } from '@angular/material/slide-toggle';

import { MatSliderModule } from '@angular/material/slider';

import { MatSnackBarModule } from '@angular/material/snack-bar';

import { MatSortModule } from '@angular/material/sort';

import { MatStepperModule } from '@angular/material/stepper';

import { MatTableModule } from '@angular/material/table';

import { MatTabsModule } from '@angular/material/tabs';

import { MatToolbarModule } from '@angular/material/toolbar';

import { MatTooltipModule } from '@angular/material/tooltip';

const modules = [

  CdkTableModule,

  MatAutocompleteModule,

  MatButtonModule,

  MatButtonToggleModule,

  MatCardModule,

  MatCheckboxModule,

  MatChipsModule,

  MatStepperModule,

  MatDatepickerModule,

  MatDialogModule,

  MatExpansionModule,

  MatGridListModule,

  MatIconModule,

  MatInputModule,

  MatListModule,

  MatMenuModule,

  MatNativeDateModule,

  MatPaginatorModule,

  MatProgressBarModule,

  MatProgressSpinnerModule,

  MatRadioModule,

  MatRippleModule,

  MatSelectModule,

  MatSidenavModule,

  MatSliderModule,

  MatSlideToggleModule,

  MatSnackBarModule,

  MatSortModule,

  MatTableModule,

  MatTabsModule,

  MatToolbarModule,

  MatTooltipModule,

];

@NgModule({

  imports: modules,

  exports: modules,

})

export class MaterialModule {

}

## RoutingModule- מודול המטפל בנושא הניווט בין המסכים בדף, מיצא את הגדרות הניווט אל המודול הראשי.

import { NgModule } from '@angular/core';

import { RouterModule, Routes } from '@angular/router';

import { HomeComponent } from './Components/UI/home/home.component';

import { AllToVolunteersComponent } from './Components/all/all-to-volunteers/all-to-volunteers.component';

import { AllFamiliesComponent } from './Components/all/all-families/all-families.component';

import { AllOrganizationsComponent } from './Components/all/all-organizations/all-organizations.component';

import { AllEventsComponent } from './Components/all/all-events/all-events.component';

import { AllVolunteersComponent } from './Components/all/all-volunteers/all-volunteers.component';

import { MainComponent } from './Components/UI/main/main.component';

import { LoginComponent } from './Components/security/login/login.component';

export const routeList: Routes = [

  { path: 'login', component: LoginComponent },

  { path: '', component: HomeComponent },

  { path: 'home', component: HomeComponent },

  { path: 'main', component: MainComponent },

  { path: 'list', children: [

      { path: 'vaf', component: AllToVolunteersComponent },

      { path: 'volunteers', component: AllVolunteersComponent },

      { path: 'families', component: AllFamiliesComponent },

      { path: 'events', component: AllEventsComponent },

      { path: 'organizations', component: AllOrganizationsComponent }    ]

  }

];

@NgModule({

  imports: [RouterModule.forRoot(routeList)],

  exports: [

    RouterModule

  ]

})

export class RoutingModule { }

## דף HTML ראשי, ע"פ מתודולוגית SPA- single page application.

<!doctype html>

<html lang="he">

<head>

  <meta charset="utf-8">

  <title>מתנ"ס מעלה רמות</title>

  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="icon" type="image/x-icon" href="favicon.ico">

</head>

<body>

  <app-root></app-root>

</body>

</html>

## דף CSS ראשי, המכיל עיצובים כללים עבור אלמנטים במערכת.

דוג' לעיצובים:

@import "~@angular/material/prebuilt-themes/deeppurple-amber.css";

@import '~font-awesome/css/font-awesome.css';

@import '~material-design-icons/iconfont/material-icons.css';

@import '~saturn-datepicker/bundle.css';

\*:focus {

  outline: none !important

}

body {

  direction: rtl;

}

.fullscreen {

  width: 100%;

  position: relative;

  /\* height: calc(93vh - (50px + 50px)); \*/

  /\* background-color: gray; \*/

  background-size: cover;

  background-position: center center;

  background-repeat: no-repeat;

}

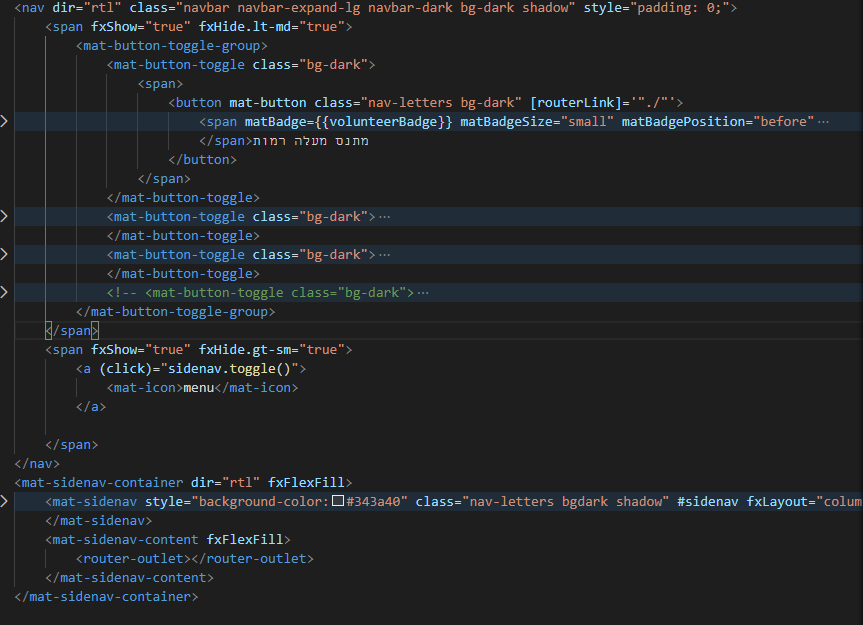
.wf-loading mat-icon {

  display: none;

}

## קומפוננטות לעיצוב מראה הדף:

## Header: קומפוננטה לעיצוב תפריט עליון, כולל שימוש בצורה רספונסיבית למכשירים צרים.



## Footer: קומפוננטה לעיצוב תחתית הדף, נדבק לתחתית העמוד.

<footer>

  <mat-toolbar class="footer">

    <div layout="row" layout-align="center center" flex>

      Pnina Bulvin © Batsheva Rich

    </div>

  </mat-toolbar>

</footer>

.footer {

    position: fixed;

    bottom: 0px;

    justify-content: center;

    align-items: center;

    background-color: darkslategray;

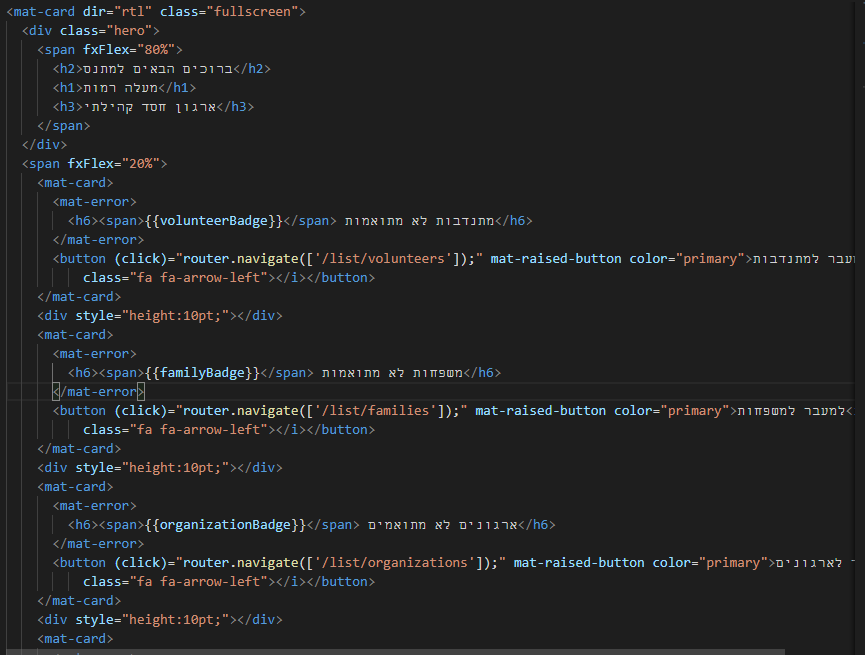
    width: 100%;

    color: #fff;

    height: 50px;

}

## Home: קומפוננטת דף הבית, מכיל אלמנטים המציגים את ההתראות החכמות.





## קומפוננטות להצגת טבלה של מרכיב התנדבות, וכן פרטים על התנדבות בודדת:

לדוג', קומפוננטה להצגת טבלת המתנדבות: *all-volunteers*

*בנית הטבלה:*

          <table mat-table [dataSource]="dataSource" matSort multiTemplateDataRows class="mat-elevation-z4">

            <ng-container matColumnDef="showDetails">

              <th mat-header-cell \*matHeaderCellDef>פרטים </th>

              <td mat-cell \*matCellDef="let element">

                <h5><i class="fa fa-caret-down"></i>

                </h5>

              </td>

            </ng-container>

            <ng-container matColumnDef="Name">

              <th mat-header-cell \*matHeaderCellDef mat-sort-header> שם </th>

              <td mat-cell \*matCellDef="let element"> {{element.Name}} </td>

            </ng-container>

            <ng-container matColumnDef="Address">

              <th mat-header-cell \*matHeaderCellDef> כתובת </th>

              <td mat-cell \*matCellDef="let element"> {{element.Address}} </td>

            </ng-container>

            <ng-container matColumnDef="Pelephone">

              <th mat-header-cell \*matHeaderCellDef> פלאפון </th>

              <td mat-cell \*matCellDef="let element"> {{element.Pelephone}} </td>

            </ng-container>

            <ng-container matColumnDef="Email">

              <th mat-header-cell \*matHeaderCellDef> מייל </th>

              <td mat-cell \*matCellDef="let element"> {{element.Email}} </td>

            </ng-container>

            <ng-container matColumnDef="Age">

              <th mat-header-cell \*matHeaderCellDef mat-sort-header> גיל </th>

              <td mat-cell \*matCellDef="let element"> {{CalculateAge(element)}} </td>

            </ng-container>

            <ng-container matColumnDef="IsActive">

              <th mat-header-cell \*matHeaderCellDef mat-sort-header> פעילה? </th>

              <td mat-cell \*matCellDef="let element">

                <div \*ngIf="element.IsActive">

                  <i class="fa fa-check"></i>

                </div>

                <div \*ngIf="!element.IsActive">

                  <i class="fa fa-times"></i>

                </div>

              </td>

            </ng-container>

            <ng-container \*ngIf="!vId" matColumnDef="columnadd">

              <th mat-header-cell \*matHeaderCellDef> + התנדבות</th>

              <td mat-cell \*matCellDef="let element">

                <h5> <button

                    (click)="$event.stopPropagation();addVolunteering(element)" [hidden]="!element.IsActive"><i

                      class="fa fa-handshake-o"></i>

                  </button>

                  <span mat-card-avatar matTooltip="לא ניתן להוסיף התנדבות כיון שהמתנדבת לא פעילה"

                    \*ngIf="!element.IsActive">

                    <button \*ngIf="false"></button>

                    <i class="fa fa-times"></i>

                  </span>

                </h5>

              </td>

            </ng-container>

            <ng-container \*ngIf="!vId" matColumnDef="columndelete">

              <th mat-header-cell \*matHeaderCellDef>מחיקה </th>

              <td mat-cell \*matCellDef="let element">

                <h5 (click)="$event.stopPropagation();delete(element)"><i class="fa fa-trash"></i>

                </h5>

              </td>

            </ng-container>

            <ng-container matColumnDef="expandedDetail">

              <td mat-cell \*matCellDef="let element" [attr.colspan]="displayedColumns.length">

                <div class="example-element-detail" [@detailExpand]="

                      element == expandedElement ? 'expanded' : 'collapsed'

                    ">

                  <div \*ngIf="element.show" class="example-element-description">

                    <app-volunteer \*ngIf="!vId" (addedVolunteer)="updateTable($event)" [vId]="element.Id">

                    </app-volunteer>

                    <span class="example-element-description-attribution">

                    </span>

                  </div>

                </div>

              </td>

            </ng-container>

            <tr mat-header-row \*matHeaderRowDef="displayedColumns"></tr>

            <tr mat-row \*matRowDef="let element; columns: displayedColumns" class="example-element-row"

              [class.example-expanded-row]="expandedElement === element" (click)="showDetails(element);vId ? null:

                  expandedElement = expandedElement === element ? null : element

                "></tr>

            <tr mat-row \*matRowDef="let row; columns: ['expandedDetail']" class="example-detail-row"></tr>

          </table>

הגדרות הטבלה ואתחולה:

  constructor(public vs: VolunteerService,

              public dialog: MatDialog,

              private datePipe: DatePipe,

              private elementRef: ElementRef,

              private snackBar: MatSnackBar) {

    // this.dataSource.filterPredicate =

    //   (data: Details, filter: string) => data.Name.indexOf(filter) !== -1;

  }

  @ViewChild(MatSort, { static: true }) sort: MatSort;

  @ViewChild(MatTable, { static: false }) table: MatTable<any>;

  @ViewChild(MatPaginator, { static: false }) paginator: MatPaginator;

  displayedColumns = ['showDetails', 'Name', 'Address', 'Pelephone', 'Email', 'Age', 'IsActive', 'columnadd', 'columndelete'];

  expandedElement: Details | null;

  volunteers: any;

  dataSource = new MatTableDataSource([]);

this.vs.getVolunteers().subscribe((volunteers: Volunteer[]) => {

        volunteers = this.vs.trimResultsFromDB(volunteers);

        this.volunteers = volunteers;

        this.dataSource.data = volunteers;

        this.resultsLength = this.dataSource.data.length;

        this.loaded = true;

        this.error = false;

      }, err => { this.error = true; this.loaded = true; });

*דוג' להצגת מידע של מרכיב בודד:*

לדוג', פרטים של מתנדבת *volunteer*

<div dir="rtl" class="text-right" style="padding-top: 1%">

  <div fxLayout="row" fxLayout.xs="column" fxLayoutWrap fxLayoutGap="3%" fxLayoutAlign="right">

    <div [fxFlex]="toV ? large : smallest">

      <mat-card>

        <mat-toolbar>

          <mat-card-header>

            <div mat-card-avatar class="example-header-image"><i class="fa fa-user" style="size: 7x;"></i></div>

            <mat-card-title>{{ myvolunteer.Name }}</mat-card-title>

          </mat-card-header>

        </mat-toolbar>

        <mat-card-content>

          <p>טלפון: {{ myvolunteer.Telephone }}</p>

          <p>פלאפון: {{ myvolunteer.Pelephone | phone }}</p>

          <p>כתובת: {{ myvolunteer.Address }}</p>

          <p>מייל: {{ myvolunteer.Email }}</p>

          <p>יום הולדת: {{ myvolunteer.Age | date:"MMMM dd yyyy" }}</p>

          <p>פעילה:

            <span \*ngIf="myvolunteer.IsActive">

              <i class="fa fa-check"></i>

            </span>

            <span \*ngIf="!myvolunteer.IsActive">

              <i class="fa fa-times"></i>

            </span>

          </p>

          <header \*ngIf="categories.length > 0">קטגוריות</header>

          <div \*ngFor="let c of categories">

            <li>

              {{c.Name}}</li>

          </div>

          <mat-card-actions>

            <button mat-raised-button (click)="VolunteeropenDialog()">לעריכת

              הפרטים</button>

          </mat-card-actions>

        </mat-card-content>

      </mat-card>

    </div>

    <mat-card \*ngIf="toV != true" [fxFlex]="toV? largest: small">

      <button \*ngIf="false" (click)="tabs.selectedIndex = 0">RESET</button>

      <mat-tab-group dir="rtl" [selectedIndex]=1 (selectedTabChange)="onTabChange($event)" #tabs>

        <mat-tab></mat-tab><!--  ADDITIONAL TAB -->

        <mat-tab label="משפחות">

          <app-all-families dir="rtl" \*ngIf="showFamily" [where]=1 [vId]="vId"></app-all-families>

        </mat-tab>

        <mat-tab label="התנדבויות">

          <app-all-to-volunteers dir="rtl" \*ngIf="showVolunteerings" [vId]="vId" [where]=1></app-all-to-volunteers>

        </mat-tab>

        <mat-tab label="ארועים">

          <app-all-events dir="rtl" \*ngIf="showEvent"></app-all-events>

        </mat-tab>

      </mat-tab-group>

    </mat-card>

  </div>

</div>

אתחול הנתונים עבור המתנדבת נשלח מהטבלה הראשית כך:

 <app-volunteer \*ngIf="!vId" (addedVolunteer)="updateTable($event)" [vId]="element.Id">

                    </app-volunteer>

ובקומפוננטה מאותחל:

  @Input() vId: number;

  @Input() toV: boolean;

ngAfterViewInit() {

    this.vs.getVolunteer(this.vId).subscribe((v: Volunteer) => {

      this.myvolunteer = this.vs.trimResultFromUpdate(v);

      this.vs.getCategoriesOfVolunteer(this.vId).subscribe((c: Category[]) => {

        this.categories = c;

      });

    });

  }

## סרביסים לניהול קריאות לשרת ע"פ מרכיבי התנדבות.

*לדוג' VolunteerService*

@Injectable({

  providedIn: 'root'

})

export class VolunteerService {

  path = environment.baseURL + 'volunteer';

  constructor(private http: HttpClient) { }

  getVolunteers(): Observable<Volunteer[]> {

    return this.http.get<Volunteer[]>(this.path);

  }

  getVolunteer(id: number): Observable<Volunteer> {

    return this.http.get<Volunteer>(

      this.path + '/' + id

    );

  }

  addVolunteer(myvolunteer: Volunteer, categories: Category[]) {

    // returns id to be updated in table

    const myData = {} as any;

    myData.volunteer = myvolunteer;

    myData.categories = categories;

    return this.http

      .post(this.path, myData).toPromise().then(res => res);

  }

  updateVolunteer(myvolunteer: Volunteer, categoriesSelected: Category[]) {

    const myData = {} as any;

    myData.volunteer = myvolunteer;

    myData.categories = categoriesSelected;

    this.http.put(

      this.path, myData

    ).subscribe();

  }

דוג' לשימוש בסרביס בקומפוננטת *volunteer-f* (קומפוננטה להוספת/עריכת מתנדבת)

submitForm() {

    if (this.data.update) {

      this.newVolunteer.Id = this.data.id;

      this.vs.updateVolunteer(this.newVolunteer, this.categoriesSelected);

      this.dialogRef.close(this.newVolunteer);

    } else {

      this.vs.addVolunteer(this.newVolunteer, this.categoriesSelected)

        .then((t: number) => {

          this.token = t;

          this.newVolunteer.Id = this.token;

          this.categoriesSelected = [];

          this.dialogRef.close(this.token);

        });

      this.snackBar.open('שמירת מתנדבת מבוצעת...', 'OK', {

        duration: 2000,

        direction: 'rtl'

      });

    }

  }

## מודלים להוספה/עריכה של מרכיבי התנדבות וכן הוספת פעולת התנדבות

לדוג', טופס הוספת/עריכת מתנדבת:

*Volunteer-f*

<span dir="rtl">

  <mat-toolbar style="width: 100% !important; z-index: 2;">

    <span \*ngIf="!data.update" mat-dialog-title>הוספת מתנדבת</span>

    <span \*ngIf="data.update" mat-dialog-title>עדכון מתנדבת</span>

  </mat-toolbar>

  <mat-dialog-content>

    <form (ngSubmit)="submitForm()" class="text-right" [formGroup]="myForm">

      <div class="form-group">

        <mat-form-field appearance="outline">

          <mat-hint>שם &nbsp;

            <span style="color: red;"

              \*ngIf="myForm.get('name').invalid && (myForm.get('name').dirty || myForm.get('name').touched)">

              <span \*ngIf="myForm.get('name').errors.required">

                שדה חובה

              </span>

              <span \*ngIf="myForm.get('name').errors.minlength">

                קצר מדי

              </span>

            </span>

          </mat-hint>

          <input matInput placeholder="שם" (focus)="$event.target.select()" maxlength="20" minlength="5"

            [(ngModel)]="newVolunteer.Name" name="name" [ngModelOptions]="{standalone: true}"

            (change)="myForm.get('name').setValue(newVolunteer.Name); myForm.get('name').valueChanges; myForm.get('name').markAsTouched()"

            required>

          <mat-icon matPrefix>person</mat-icon>

        </mat-form-field>

      <mat-hint>פעילה? </mat-hint>

        <mat-slide-toggle name="isActive" [ngModelOptions]="{standalone: true}" [ngModel]="newVolunteer.IsActive"

          [checked]="newVolunteer.IsActive" (ngModelChange)="newVolunteer.IsActive = !newVolunteer.IsActive">

        </mat-slide-toggle>

        <br />

        <br />

        <app-choose-category [chosenC]="categoriesOfVolunteer" (selectc)="selectCategories($event)">

        </app-choose-category>

      </div>

      <mat-form-field appearance="outline" class="example-full-width">

        <mat-hint>הערות</mat-hint>

        <textarea matInput (focus)="$event.target.select()" [ngModelOptions]="{standalone: true}"

          [(ngModel)]="newVolunteer.Comments" name="Comments"></textarea>

        <mat-icon matSuffix>comment</mat-icon>

      </mat-form-field>

      <mat-dialog-actions align="end">

        <button mat-button mat-raised-button mat-dialog-close>ביטול</button>

        <button mat-button mat-raised-button [mat-dialog-close]="token" type="submit" [disabled]="!myForm.valid">

          <span \*ngIf="!data.update">הוספה</span>

          <span \*ngIf="data.update">עדכון</span>

        </button>

      </mat-dialog-actions>

    </form>

    <br />

  </mat-dialog-content>

</span>

@Component({

  selector: 'app-volunteer-f',

  templateUrl: './volunteer-f.component.html',

  styleUrls: ['./volunteer-f.component.css']

})

export class VolunteerFComponent implements OnInit, OnDestroy {

  categories: Category[] = [];

  categoriesOfVolunteer: Category[] = [];

  token = 0;

  categoriesSelected: Category[] = [];

  newVolunteer: Volunteer = new Volunteer('', '', '', '', '', '2001-01-01', true);

  @Output() createdVolunteer: EventEmitter<Volunteer> = new EventEmitter<Volunteer>();

  myForm: FormGroup;

  constructor(public vs: VolunteerService,

              cs: CategoryService,

              private dialogRef: MatDialogRef<VolunteerFComponent>,

              @Inject(MAT\_DIALOG\_DATA) public data: any,

              public snackBar: MatSnackBar,

              private elementRef: ElementRef,

              private formBuilder: FormBuilder) {

    cs.getCategories().subscribe((res: Category[]) => {

      this.categories = res;

    });

  }

  ngOnInit() {

    this.myForm = this.formBuilder.group({

      name: new FormControl('', [

        Validators.required,

        Validators.minLength(5),

        Validators.maxLength(20)

      ]),

      email: new FormControl('', [

        Validators.minLength(5),

        Validators.maxLength(25),

        Validators.pattern('^[^\\s@]+@[^\\s@]+\\.[^\\s@]{2,}$')

      ]),

      phone: new FormControl('', [

        Validators.minLength(9),

        Validators.maxLength(9),

        Validators.pattern('^[0-9]\*$')

      ]),

      cellphone: new FormControl('', [

        Validators.required,

        Validators.minLength(10),

        Validators.maxLength(10)

      ])

    });

    if (this.data.update) {

      this.newVolunteer = this.vs.trimResultFromUpdate(this.data.dataKey);

      this.myForm.get('name').setValue(this.newVolunteer.Name);

      this.myForm.get('email').setValue(this.newVolunteer.Email);

      this.myForm.get('phone').setValue(this.newVolunteer.Telephone);

      this.myForm.get('cellphone').setValue(this.newVolunteer.Pelephone);

      this.categoriesOfVolunteer = this.data.chosenC;

    }

  }

  submitForm() {

    if (this.data.update) {

      this.newVolunteer.Id = this.data.id;

      this.vs.updateVolunteer(this.newVolunteer, this.categoriesSelected);

      this.dialogRef.close(this.newVolunteer);

    } else {

      this.vs.addVolunteer(this.newVolunteer, this.categoriesSelected)

        .then((t: number) => {

          this.token = t;

          this.newVolunteer.Id = this.token;

          this.categoriesSelected = [];

          this.dialogRef.close(this.token);

        });

      this.newVolunteer = new Volunteer('', '', '', '', '', '2001-01-01', true);

      this.snackBar.open('שמירת מתנדבת מבוצעת...', 'OK', {

        duration: 2000,

        direction: 'rtl'

      });

    }

  }

  ngOnDestroy(): void {

    this.elementRef.nativeElement.remove();

  }

  selectCategories(e: { checked: boolean, id: number, name: string }[]) {

    this.categoriesSelected = [];

    e.forEach((element: { checked: boolean, id: number, name: string }) => {

      if (element.checked) {

        this.categoriesSelected.push(new Category(element.name, element.id));

      }

    });

  }

}

עוד דוג', טופס לעריכת והוספת קטגוריות למרכיב התנדבות המאפשר בו זמנית לערוך, להוסיף ולמחוק קטגוריות בצורה אינטואטיבית.

<span dir="rtl">

    <h6 style="width: 100% !important; z-index: 2;">

        <span \*ngIf="!vId" mat-dialog-title>הוספת קטגוריה</span>

        <span \*ngIf="vId" mat-dialog-title>עדכון קטגוריה</span>

    </h6>

    <mat-dialog-content>

        <form (ngSubmit)="submitForm(categoryForm)" class="text-right" #categoryForm="ngForm">

            <div class="form-group">

                <mat-form-field appearance="outline">

                    <mat-hint>שם קטגוריה</mat-hint>

                    <input matInput (focus)="$event.target.select()" maxlength="20" [(ngModel)]="newCategory.Name"

                        name="Name" required>

                    <mat-icon \*ngIf="!vId" matPrefix>add\_to\_photos</mat-icon>

                    <mat-icon \*ngIf="vId" matPrefix>edit</mat-icon>

                </mat-form-field>

                <br />

            </div>

            <mat-dialog-actions align="end">

                <button mat-button mat-raised-button mat-dialog-close>ביטול</button>

                <button mat-button mat-raised-button type="submit">

                    <span \*ngIf="!vId">הוספה</span>

                    <span \*ngIf="vId">עדכון</span>

                </button>

            </mat-dialog-actions>

        </form>

        <br />

    </mat-dialog-content>

</span>

@Component({

  selector: 'app-category-f',

  templateUrl: './category-f.component.html',

  styleUrls: ['./category-f.component.css']

})

export class CategoryFComponent implements OnInit, OnDestroy {

  @ViewChild('categoryForm') mytemplateForm: NgForm;

  newCategory: Category = new Category('...', 0);

  token = 0;

  @Input() vId: number;

  @Output() changedCategory: EventEmitter<Category> = new EventEmitter<Category>();

  constructor(private cs: CategoryService,

              private dialogRef: MatDialogRef<CategoryFComponent>,

              @Inject(MAT\_DIALOG\_DATA) public data: any,

              public snackBar: MatSnackBar,

              private elementRef: ElementRef) {

  }

  ngOnInit() {

    if (this.vId) {

      this.cs.getCategory(this.vId).subscribe((res: Category) => {

        this.newCategory = res;

      });

    }

  }

  submitForm() {

    if (this.vId) {

      this.newCategory.Id = this.vId;

      this.cs.updateCategory(this.newCategory).subscribe(res => {

        this.changedCategory.emit(this.newCategory);

      });

    } else {

      this.cs.addCategory(this.newCategory)

        .then((t: number) => {

          this.token = t;

          this.newCategory.Id = this.token;

          this.changedCategory.emit(this.newCategory);

        });

      this.mytemplateForm.resetForm();

      this.newCategory = new Category('default', 0);

      this.snackBar.open('שמירת קטגוריה מבוצעת...', 'OK', {

        duration: 2000,

        direction: 'rtl'

      });

    }

  }

  ngOnDestroy(): void {

    this.elementRef.nativeElement.remove();

  }

}

## פונקציות לטיפול ביבוא וביצוא של הנתונים לאקסל

<div mat-card-avatar matTooltip="הורדה לקובץ אקסל" (click)="exportTableToExcel()" \*ngIf="!vId">

            <i class="fa fa-file-excel-o" aria-hidden="true"></i>

          </div>

          <span class="example-spacer"></span>

          <button style="visibility: hidden;"></button>

          <div mat-card-avatar matTooltip="העלאת קובץ אקסל" \*ngIf="!vId"

            onclick="document.getElementById('fileToUpload').click()"><i matPrefix class="fa fa-upload"

              aria-hidden="true"></i></div>

          <input mat-input type="file" id="fileToUpload" style="display:none;" (change)="uploadedFile($event)"

            placeholder="Upload file" accept=".xlsx">

public exportTableToExcel() {

    const data = this.volunteers.map((x: Volunteer) => ({

      Id: x.Id,

      שם: x.Name,

      כתובת: x.Address,

      טלפון: x.Telephone,

      פלאפון: x.Pelephone,

      מייל: x.Email,

      תאריך\_לידה: this.datePipe.transform(x.Age, 'MM/dd/yyyy'),

      פעילה: x.IsActive === true ? 'כן' : 'לא',

      הערות: x.Comments

    }));

    const wb = XLSX.utils.book\_new();

    const ws = XLSX.utils.json\_to\_sheet(data);

    ws['!cols'] = []; // hide id column

    ws['!cols'][0] = { hidden: true };

    XLSX.utils.book\_append\_sheet(wb, ws, 'מתנדבות');

    XLSX.writeFile(wb, `מתנדבות.xlsx`);

  }

  uploadedFile(event) {

    this.fileUploaded = event.target.files[0];

    this.readExcel();

  }

  readExcel() {

    const readFile = new FileReader();

    readFile.onload = (e) => {

      this.arrayBuffer = readFile.result;

      const data = new Uint8Array(this.arrayBuffer);

      const arr = new Array();

      for (let i = 0; i !== data.length; ++i) { arr[i] = String.fromCharCode(data[i]); }

      const bstr = arr.join('');

      const workbook = XLSX.read(bstr, { type: 'binary' });

      const firstSheetName = workbook.SheetNames[0];

      const worksheet = workbook.Sheets[firstSheetName];

      XLSX.utils.sheet\_to\_json(worksheet, { raw: true });

      let js = XLSX.utils.sheet\_to\_json(worksheet, { raw: true });

      js = js as Details[];

      if (!js[1]['שם']) {

        this.snackBar.open('קובץ לא תקני, נא להעלות קובץ נכון...', 'OK', {

          duration: 5000,

          direction: 'rtl'

        });

      } else {

        const newData = js.map((x) => ({

          Id: x['Id'] as number,

          Name: x['שם'],

          Address: x['כתובת'],

          Telephone: x['טלפון'],

          Pelephone: x['פלאפון'],

          Email: x['מייל'],

          Age: x['תאריך\_לידה'] ? this.datePipe.transform(new Date(x['תאריך\_לידה']).toDateString(), 'MM/dd/yyyy') : new Date(),

          Active: x['פעילה?'] === true ? 'true' : 'false',

          Comments: x['הערות']

        }));

        newData.forEach((element) => {

          if (element.Id) {

            this.toUpdate.push(Object.assign(element));

          } else {

            this.toSave.push(Object.assign(element));

          }

        });

        this.confirmDialogAdd().subscribe(res => {

          if (res === false){

            this.snackBar.open('לא מתבצעת הוספה', 'OK', {

              duration: 2000,

              direction: 'rtl'

            });

          } else {

            if (res) {

            res.forEach(element => {

             this.vs.addVolunteer(element, null);

            });

          }

          }

        });

        this.confirmDialogUpdate().subscribe(res => {

          if (res === false){

            this.snackBar.open('לא מתבצעת הוספה', 'OK', {

              duration: 2000,

              direction: 'rtl'

            });

          } else {

            if (res) {

            res.forEach(element => {

             this.vs.updateVolunteer(element, null);

            });

          }

          }

        });

        this.dataSource.data = newData;

        this.resultsLength = this.dataSource.data.length;

        console.log(newData);

        this.snackBar.open('קובץ נטען בהצלחה', 'OK', {

          duration: 5000,

          direction: 'rtl'

        });

      }

    };

    readFile.readAsArrayBuffer(this.fileUploaded);

  }

## הודעות והתראות למשתמש:

לדוג', בקשת אישור למחיקת מרכיב התנדבות:

*Confirm-dialog*

<div dir="rtl" style="text-align: right;">

  <mat-toolbar>

    <span>{{title}}</span>

  </mat-toolbar>

  <mat-dialog-content>

    <p>{{message}}</p>

  </mat-dialog-content>

  <mat-dialog-actions>

    <button mat-button mat-raised-button mat-dialog-close (click)="onDismiss()">לא</button>

    <button mat-button mat-raised-button mat-dialog-close (click)="onConfirm()">כן</button>

  </mat-dialog-actions>

</div>

@Component({

  selector: 'app-confirm-dialog',

  templateUrl: './confirm-dialog.component.html',

  styleUrls: ['./confirm-dialog.component.css']

})

export class ConfirmDialogComponent implements OnInit {

  title: string;

  message: string;

  constructor(public dialogRef: MatDialogRef<ConfirmDialogComponent>,

              @Inject(MAT\_DIALOG\_DATA) public data: ConfirmDialogModel) {

    // Update view with given values

    this.title = data.title;

    this.message = data.message;

  }

  ngOnInit() {

  }

  onConfirm(): void {

    this.dialogRef.close(this.listOfTypeSelected);

  }

  onDismiss(): void {

    this.dialogRef.close(false);

  }

}

/\*\*

 \* Class to represent confirm dialog model.

 \*

 \* It has been kept here to keep it as part of shared component.

 \*/

export class ConfirmDialogModel {

  constructor(public title: string, public message: string, public listOfType?: object[], public type?: string) {

  }

}

שימוש בקומפוננטה:

  delete(elm: Details) {

    this.confirmDialog().subscribe(res => {

      this.result = res;

      if (res) {

        this.vs.removeVolunteer(elm.Id);

        this.dataSource = new MatTableDataSource(Object.values(this.dataSource)

          .filter(i => i !== elm));

      }

    });

  }

  confirmDialog(): Observable<any> {

    const message = `מחיקה זו היא לצמיתות! האם תרצי להמשיך?`;

    const dialogData = new ConfirmDialogModel('מחיקת מתנדבת', message);

    const dialogRef = this.dialog.open(ConfirmDialogComponent, {

      maxWidth: '75%',

      data: dialogData

    });

    return dialogRef.afterClosed();

  }

## עיצוב הגדרות טבלת מרכיבי ההתנדבות והתאמתה לעברית:

*Ranges-footer*

<div class="mat-calendar-footer">

    <button mat-fab

        \*ngFor="let range of ranges"

        (click)="setRange(range.key)">{{range.label}}</button>

</div>

@Component({

  selector: 'app-ranges-footer',

  templateUrl: './ranges-footer.component.html',

  styleUrls: ['./ranges-footer.component.css']

})

export class RangesFooter<Date> implements SatCalendarFooter<Date> {

  public ranges: Array<{ key: string, label: string }> = [

    { key: 'today', label: 'Today' },

    { key: 'thisWeek', label: 'This Week' },

  ];

  private destroyed = new Subject<void>();

  constructor(

    private calendar: SatCalendar<Date>,

    private datePicker: SatDatepicker<Date>,

    private dateAdapter: DateAdapter<Date>,

    cdr: ChangeDetectorRef

  ) {

    calendar.stateChanges

      .pipe(takeUntil(this.destroyed))

      .subscribe(() => cdr.markForCheck());

  }

  setRange(range: string) {

    switch (range) {

      case 'today':

        this.calendar.beginDate = this.dateAdapter.deserialize(new Date());

        this.calendar.endDate = this.dateAdapter.deserialize(new Date());

        this.calendar.activeDate = this.calendar.beginDate;

        break;

      case 'thisWeek':

        const today = moment();

        this.calendar.beginDate = this.dateAdapter.deserialize(today.weekday(0).toDate());

        this.calendar.endDate = this.dateAdapter.deserialize(today.weekday(6).toDate());

        break;

    }

    this.calendar.activeDate = this.calendar.beginDate;

    this.calendar.beginDateSelectedChange.emit(this.calendar.beginDate);

    this.calendar.dateRangesChange.emit({ begin: this.calendar.beginDate, end: this.calendar.endDate });

    this.datePicker.close();

  }

}

וב-appModule:

providers: [DatePipe, { provide: MatPaginatorIntl, useValue: getHebrewPaginatorIntl() }

## שימוש ב-Pipes:

*לדוג', Date*

<ng-container matColumnDef="StartDate">

    <th mat-header-cell \*matHeaderCellDef> תאריך התחלה </th>

     <td mat-cell \*matCellDef="let element"> {{element.StartDate | date:"MMMM dd yyyy"}} </td>

            </ng-container>

            <ng-container matColumnDef="EndDate">

              <th mat-header-cell \*matHeaderCellDef> תאריך סיום </th>

              <td mat-cell \*matCellDef="let element"> {{element.EndDate | date:"MMMM dd yyyy"}} </td>

</ng-container>

## שימוש ב-Directives:

לדוג', Directive שכתבנו עבור ולידציה של שדה טקסטואלי שאמור להכיל מספר טלפון/פלאפון:

*Phone-mask*

@Directive({

  selector: '[appPhoneMask]'

})

export class PhoneMaskDirective implements OnInit {

  constructor(private el: ElementRef, private red: Renderer2) {}

  ngOnInit() {}

  @HostListener('input')

  onkeypress() {

    const x = this.el.nativeElement.value;

    if (isNaN(x) || x.length > 10) { this.el.nativeElement.value = x.slice(0, -1); }

  }

}

דוג' לשימוש ב-Directive:

<input matInput (focus)="$event.target.select()"appPhoneMask placeholder="02-525-2525" [(ngModel)]="newFamily.Telephone"

            name="Telephone" [ngModelOptions]="{standalone: true}">

## הצגת ההתראות החכמות בתפריט:

<button mat-button class="nav-letters" [routerLink]="'/list/families'"

                        routerLinkActive="activeLink">

  <i matBadge={{familyBadge}} matBadgeSize="small" matBadgePosition="before"

     matBadgeColor="accent" class="fa fa-child" aria-hidden="true"></i>משפחות

</button>

  ngAfterViewInit() {

    this.familyBadge = this.ns.Families.length;

    this.eventBadge = this.ns.Events.length;

    this.homeBadge = this.familyBadge + this.eventBadge;

  }

הסרביס שטוען את הנתונים:

public get Families(): Family[] {

    return this.familiesToConnect;

  }

  public get Events(): Eventt[] {

    return this.EventsToConnect;

  }

getAllFamiliesToConnect() {

    return this.http.get<Family[]>(environment.baseURL + '/getfamiliestoconnect').subscribe((res: Family[]) => {

      this.familiesToConnect = res;

    });

  }

  getAllEventsToConnect() {

    return this.http.get<Eventt[]>(environment.baseURL + '/geteventstoconnect').subscribe((res: Eventt[]) => {

      this.EventsToConnect = res;

    });

  }

  }

## מחלקות שמיצגות את מרכיבי ההתנדבות:

לדוג', מחלקת מתנדבת:

*Volunteer*

export class Volunteer {

    Id: number;

    Name: string;

    Telephone: string;

    Pelephone: string;

    Address: string;

    Email: string;

    Age: string;

    Comments: string;

    IsActive: boolean;

    show: boolean;

    constructor(id: number, Name: string, telephone: string, pelephone: string, email: string, address: string, birthdate: string, isActive: boolean) {

        this.Id = id;

        this.Name = Name;

        this.Telephone = telephone;

        this.Pelephone = pelephone;

        this.Address = address;

        this.Email = email;

        this.Age = birthdate;

        this.IsActive = isActive;

        this.show = false;

    }

    public CalculateAge() {

        const birthday = new Date(this.Age);

        const timeDiff = Math.abs(Date.now() - birthday.getTime());

        return Math.floor((timeDiff / (1000 \* 3600 \* 24)) / 365.25);

    }

}



## בדיקות המערכת

במהלך כתיבת התוכנית ולאחריה ביצענו במערכת בדיקות קפדניות. בדקנו תסריטים של ה-use cases השונים, ערכנו דימויים של מקרי קצה ווידאנו שבכל מצב מגיבה המערכת כמצופה ממנה.

מה הקנה הפרויקט

1. מה הקנה הפרויקט

* ניהול גרסאות ע"י שימוש ב-Git ו-Github.
* לימוד ושליטה בסביבת .NET ובשפת C#
* לימוד טכנולוגית אנגולר ושימוש באוביקטים מובנים תוך שימת לב למחזור החיים של האלמנטים השונים.
* לימוד ושליטה בשפת TS
* ניסיון מקיף באופנים שונים של תקשורת שרת-לקוח
* התנסות בשימוש בספריות קוד של הסביבות השונות
* הכרת מחלקות חדשות
* הכרת נושאים שונים וסוגיות מורכבות בעולם התכנות
* ניסיון בטכנולוגיות חדשניות
* התנסות בתכנון ובאפיון מערכות
* התנסות בהקמת תוכנה מראשית ועד אחרית
* פיתוח הלוגיקה
* ניסיון בתכנון נתונים ועיבוד יעיל
* עמידה בפני בעיות תכנותיות ומציאת פתרונות.
* חיפוש ולמידה דרך הרשת
* כלים חדשים ללימוד עצמי איכותי
* ניסיון בבדיקות מקיפות
* ייעול עבודת צוות
* שימוש בספריות ודריסה של התנהגות מובנית, כמו angular material, xlsx, etc…

ביבליוגרפיה

1. ביבליוגרפיה

## אתרי לימוד/תיעוד קוד:

* <https://docs.microsoft.com/en-us/>
* <https://material.angular.io/>
* <https://angular.io/>
* <https://stackoverflow.com/>
* <https://getbootstrap.com/>
* <https://www.electronjs.org/>
* <https://medium.com/>
* <https://www.npmjs.com/package/xlsx>
* <https://www.npmjs.com/package/@angular/flex-layout>
* <https://github.com/SaturnTeam/saturn-datepicker>
* <https://fontawesome.com/>
* <https://valor-software.com/ng2-charts/#/>

## עזרה מקוונת:

* <https://stackoverflow.com/questions>
* <https://github.com/>
* [www.tutorialspoint.com](http://www.tutorialspoint.com)
* <https://docs.microsoft.com/en-us/dotnet/csharp/>
* ועוד...