

דרכי אבות מערכת לישיבה ועמותה.

**מגישות:**

גיטי גולדשטוף ורבקי סיטון

**מנחה:**

גב' ורדית לוי

**מקום לימוד:**

סמינר מרכז בית יעקב- "החדש"

**תאריך הגשה:**

יוני 2023

כאן צריך להיות אישור מטעם מה"ט

הצעה לפרויקט גמר תאריך:15/06/2023

1. **פרטי הסטודנטים**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **שם התלמיד** | **ת.ז.** | **כתובת** | **עיר** | **טלפון** | **תאריך סיום הלימודים** |
| גיטי גולדשטוף | 213426695 | מעגלי הרים לוין 21 | ירושלים | 0583201274 | ? |
| רבקי סיטון | 214127649 | הרב עוזיאל 60 | ירושלים | 0527145895 | ? |

**שם המכללה:** סמינר מרכז בית יעקב

**מגמה:** הנדסת תוכנה

**מסלול:** הנדסאים

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **שם המנחה** | **כתובת** | **טלפון נייד** | **תואר** | **מקום עבודה / תפקיד** |
| גב' ורדית לוי |  | 0527617762 |  |  |

1. **פרטי המנחה האישי**:
2. **שם הפרויקט**: מערכת לישיבת דרכי אבות
3. **רקע**
   1. **תיאור ורקע כללי**  
      הלקוח: מנהל עמותת דרכי אבות. עמותה המנהלת ישיבה לחוזרים בתשובה וימי עיון לחיילים וחיילות. פותח יישום להנהלת העמותה, המאפשר ניהול פרטי תלמידים וניהול ימי העיון, וכן טופס דיגיטלי עבור הרשמה לימי עיון.
   2. **מטרות המערכת**  
       o ניהול מאגר תלמידים, כולל ניהול היסטוריית הפרטים  
      o ניהול ימי עיון  
       o טפסי הרשמה מקוונים לימי עיון  
      o קליטת נתוני עבר באופן אוטומטי מקבצי אקסל קיימים
4. **סקירת מצב קיים בשוק**  
   ישנם מספר תוכנות עבור ניהול תלמידים וכד', אולם הן אינן מותאמות אישית לצרכי הלקוח. כמו כן בדרך כלל מנהלי אתרים מסוג זה גובים שכר רב על שינוי והוספת אפשרויות בהתאם ללקוח.
5. **מה הפרויקט אמור לחדש או לשפר**  
   הפרויקט יאפשר ניהול מסודר של פרטי תלמידים, ימי עיון, וכן אפשרות סינכרון נתונים ישנים.

1. **דרישות מערכת ופונקציונאליות**
   1. **דרישות מערכת**  
      **צד השרת**   
      SQL Server כולל מנגנוני התאוששות מקריסה.  
      נפח דיסק פנוי של 30GB וזיכרון מומלץ 8GB.

**צד הלקוח**   
יוכל לפעול על מכשיר או מחשב אישי המריץ דפדפן google chrome (תיתכן תמיכה בדפדפנים נוספים), רוחב פס 100MBPS לפחות.

**שרידות, ביצועים / התמודדות עם עומסים:**

צד השרת מריץ IIS Express המסוגל להתמודד עם מספר קריאות בו זמנית.

גם עומס על שרת ה-SQL אינו צפוי בסדר גודל כזה של אתר, מכיוון שהוא

בנוי להתמודדות בהצלחה עם עומסים כבדים בהרבה.

* 1. **דרישות פונקציונאליות**  
     המערכת תאפשר לגולשים:

להירשם לישיבה באופן מקוון ולקבל הודעה על יום העיון הקרוב ואפשרות הרשמה אליו.   
המערכת תאפשר למנהל:

בניהול פרטי התלמידים:

לצפות ברשימת התלמידים ופרטיהם ,להוסיף תלמיד חדש, לערוך תלמיד קיים, לצפות בהיסטוריית תלמיד, לשלוח קישור להרשמה לישיבה,

וכן אפשרות הורדת רשימת התלמידים ופרטיהם לאקסל או הדפסת רשימת התלמידים ופרטיהם וכן אפשרות להעלות קובץ אקסל ולטענו ברשימה שבמערכת.

בניהול ימי העיון:

לצפות ברשימת ימי העיון ופרטיהם, להוסיף יום עיון חדש, לערוך יום עיון קיים,

לצפות במשתתפי יום העיון הנוכחי, לשלוח קישור להשתתפות ביום העיון

לצפייה בתכניה, וכן הוספת/הסרת משתתף ליום עיון.

1. **בעיות צפויות במהלך הפיתוח ופתרונות**
   1. **תיאור הבעיות**
      1. כיצד הנהלת העמותה תתמודד עם קליטת כמות גדולה של נתוני עבר המנוהלים באקסל?

6.1.2 העמותה תבקש לדעת את סטטוס פרטי התלמיד בזמן נתון, איך נשמור על הנתונים שלא ידרסו בעדכון פרטים?

* 1. **פתרונות אפשריים**
     1. המערכת כוללת אפשרות לטעינת קבצי אקסל ומזריקה את הנתונים מהעבר באופן אוטומטי לרשימת פרטי התלמידים.
     2. ע"מ שפרטי התלמיד יהיו מתועדים בכל זמן נתון ולא ידרסו בעריכה המערכת שומרת בכל עדכון את הפרטים הקודמים ומעבירה אותם לקטגוריית היסטוריית תלמיד שם ניתן לצפות בכל היסטוריית פרטיו.

1. **פתרון טכנולוגי נבחר:**
   1. **טופולוגית הפתרון**  
      המערכת תיושם ברשת האינטרנט, ותהיה מורכבת משרת המנהל מסד נתונים יחסי של SQL SERVER, וממשק משתמש בצד הלקוח. התקשורת ביניהם תתבצע באמצעות WEB API.
   2. **טכנולוגיות בשימוש**Web-API  
      השרת יממש Web API כך שקליינטים יוכלו לפנות אליו ולקבל מידע. התשתית שנשתמש בה לצורך כך היא C# שפותחה בטכנולוגית .NETושימוש ב Entity FrameWork

לצורך הבאת הנתונים מהDB

* 1. **שפות הפיתוח**  
     השרת ייכתב בשפת C#   
     צד הלקוח ייכתב בשפת React - Typescript  
     בחרנו בשפות אלו שהן פופולריות, שימושיות, ונוחות לפיתוח.
  2. **תיאור הארכיטקטורה הנבחרת**  
     שרת SQL Server יאחסן את הנתונים.

שכבת ניהול הנתונים – Data Layer - תהיה אחראית לשליפת הנתונים הנדרשים ולעדכונם.

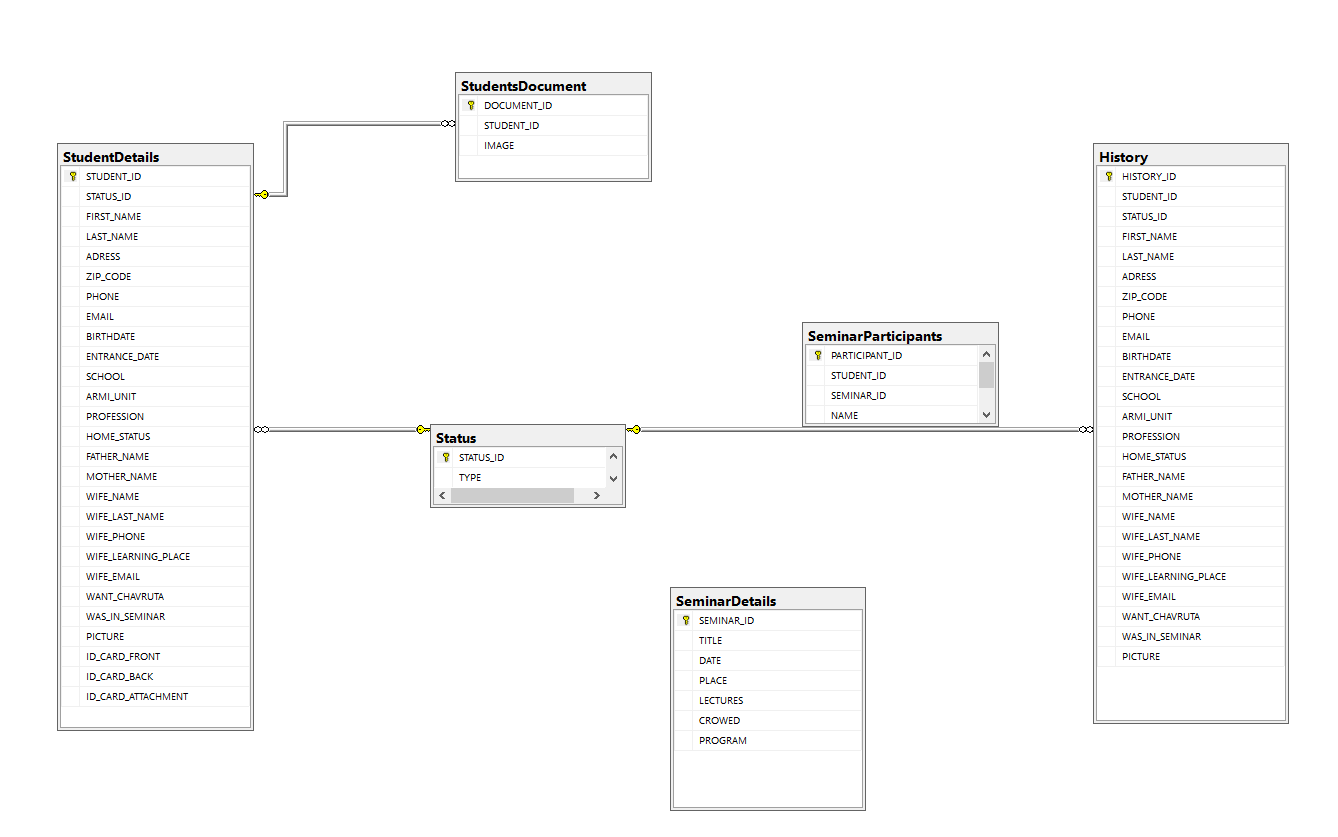
השכבה הלוגית – Business Layer – תנהל את הלוגיקה הנדרשת.

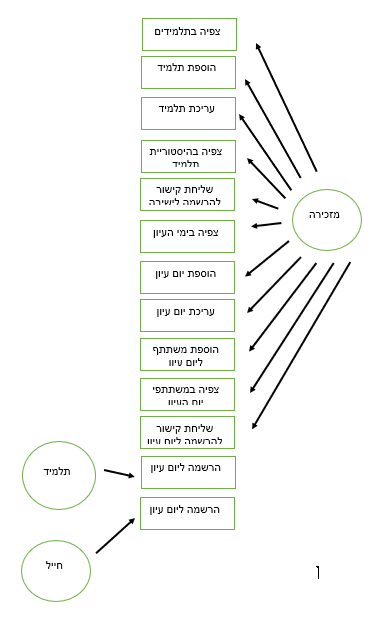
תשתית C# תאפשר תקשורת בין השרת ללקוח.

בשימוש עם Entity Framework

ממשק צד הלקוח ב React יפנה לשרת Web-API לצורך קבלת המידע הנדרש או לצורך עדכונו ויציג את דפי האפליקציה למשתמש.

* 1. **חלוקה לתוכניות ומודולים**  
     חלק א': מכלול תוכנה שרתי, שינהל את מרכיבי הניהול העסקי והאלגוריתם של המערכת.  
     חלק ב': מכלול הלקוח, שיחשוף ממשק משתמש מעל הweb
  2. **סביבת השרת**  
     שרת IIS יארח את הAPI של צד השרת.
  3. **ממשק המשתמש –GUI**  
     בטכנולוגיית React ניצור דפי אינטרנט שיוצגו על גבי דפדפן chrome (תיתכן תמיכה בדפדפנים נוספים).
  4. **ממשקים למערכות אחרות**  
     לצורך שליחת המיילים -התממשקות עם OUTLOOK.
  5. **שימוש בחבילות תוכנה**  
     במהלך הפיתוח נשתמש ב-npm ובNuget Package Manager- להתקנת חבילות תוכנה נוספות באם תידרשנה.

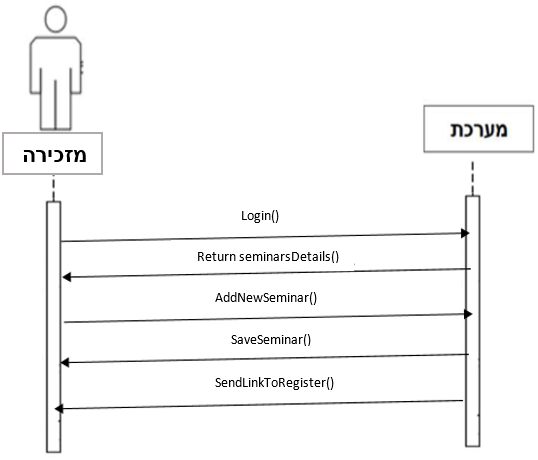
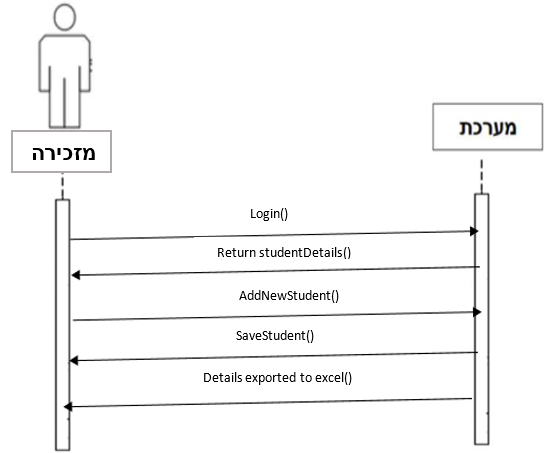
1. **שימוש במבני נתונים וארגון קבצים:**
   1. הישויות העיקריות במערכת: פרטי תלמיד, היסטוריית תלמיד, מסמכים פר תלמיד, פרטי סמינר ומשתתפי סמינר.  
      
   2. הנתונים יאוחסנו במסד נתונים יחסי של SQL SERVER
   3. לצורך התאוששות מקריסה יופעלו מנגנוני הגיבוי והשחזור שלSQL Server



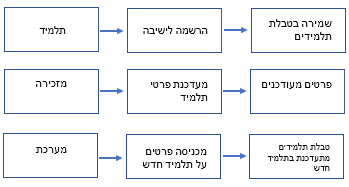
1. **תרשימי מערכת מרכזיים**

9.1 **use cases**

**Sequence Diagram.9.2**



**Data Flow Diagram .9.3**



1. **תיאור המרכיב האלגוריתמי – חישובי**
   1. בעיה רצינית שהפרויקט נותן לה מענה היא שמירת פרטי תלמיד גם לאחר שינויים או עדכונם.

על מנת לשמור על היסטוריית פרטי התלמיד יצרנו פונקציה שבקליטת שינוי היא שולחת את הפרטים הקודמים לקטגוריית היסטוריה ומוסיפה אותם. כך ניתן לצפות בפרטי תלמיד בזמן נתון ולא רק בנוכחיים.

* 1. האפשרות לקלוט נתונים מהעבר אל תוך המערכת נראתה סבוכה וארוכה ועל כן יצרנו פונקציה שתקלוט את כל הפרטים מקובץ אקסל ותעדכן אותם אוטומטית במערכת.

1. **אבטחת מידע**  
   בנושא אבטחת המידע נסתמך על המנגנונים האוטומטיים של שרת הIIS המארח. אין התייחסות מיוחדת נוספת לנושא אבטחת המידע מכיוון שהאפליקציה אינה עוסקת בתכנים רגישים.
2. **משאבים הנדרשים לפרויקט**
   1. **היקף שעות העבודה** כ- 700 שעות, שיחולקו בין המגישות בצורה מאוזנת.
   2. **ציוד נדרש:** מחשב PC עם זיכרון RAM של 8GB, תשתית רשת המאפשרת שליחת הודעות דואר אלקטרוני.
   3. **תוכנות נדרשות:**

Visual Studio Code  
SQL Server

Visual Studio

* 1. **ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט:**  
     העמקת הידע וההכרות עם הטכנולוגיות שבשימוש הפרויקט, הכרת ולימוד התעסקות עם גיט. וכן התמחות בכתיבה ב-react tsx.
  2. **ספריות ומקורות מידע**  
     https://mui.com/  
     https://microsoft.com/en-us/dotnet/csharp/

<https://stackoverflow.com>

[https://quebit.com/askquebit/sql-server /](https://quebit.com/askquebit/sql-server%20/)

1. **תכנית עבודה ושלבים לביצוע הפרויקט**  
   התכנון והפיתוח החל מחודש טבת תשפ"ג:
   1. תכנון פונקציונאליות המערכת (כולל ציורי מסכים) – כשלושה שבועות.
   2. יצירת תרשימי מערכת- כשבועיים.
   3. תכנון בסיס הנתונים- כשבועיים.
   4. כתיבת הצעת הפרויקט- כשבוע.
   5. פיתוח צד השרת- כעשרה שבועות.
   6. פיתוח צד הלקוח- כעשרה שבועות.
   7. כתיבת ספר הפרויקט- לאורך כל תהליך הפיתוח.
2. **תכנון הבדיקות שיבוצעו**
   1. **בדיקות תהליכיות ברמת משתמש – Full Flow**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **מספר בדיקה** | **מהות הבדיקה** | **אופן ביצוע הבדיקה** | **תוצאה רצויה** | **תוצאה בפועל** | **עבר או נכשל** |
| 1 | רישום תלמיד | הזנת פרטי תלמיד ואישור | קבלת הודעה שהתלמיד נוסף למאגר וחזרה לעמוד הבית |  |  |
| 2 | עריכת תלמיד | הזנת פרטים מעודכנים ואישור | קבלת הודעה שהעדכון נשמר וחזרה לעמוד הבית |  |  |
| 3 | רישום תלמיד תוך השמטת פרטים הכרחיים | הזנת פרטים ללא ת.ז. וכד' | השדה הנוכחי יהפוך לצבע אדום וימנע מכפתור השמירה לעבוד |  |  |
| 4 | רישום יום עיון | הזנת פרטי יום העיון ואישור | קבלת הודעה שיום העיון נשמר וחזרה לעמוד ימי העיון |  |  |
| 5 | רישום משתתף ליום עיון | הזנת פרטי משתתף פר יום עיון | קבלת הודעה שהמשתתף נוסף וחזרה לעמוד ימי העיון |  |  |

* 1. **בדיקות יחידה –Unit Test**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **מספר בדיקה** | **מהות הבדיקה** | **אופן ביצוע הבדיקה** | **תוצאה רצויה** | **תוצאה בפועל** | **עבר או נכשל** |
| 1 | שליפת תלמיד לפי id | קריאה לWEB-API לפונקציה GetStudentById עם פרמטר סינון לפי מספר זהות | התלמיד עם תעודת הזהות הנוכחית |  |  |
| 2 | הוספת תלמיד חדש | שליחה לפונקציה AddNewStudent עם פרטי התלמיד | התלמיד החדש יעודכן בDB  ויוצג במערכת |  |  |
| 3 | בדיקה האם קיימת תעודת זהות זהה לזו הנוכחית | שליחת פרטי התלמיד לפונקציה AddStudent בWeb-API ובדיקה האם אין כפילויות | התלמיד נוסף באם תעודת הזהות לא מוכרת ושגיאה באם תעודת הזהות קיימת כבר ע"ש תלמיד אחר |  |  |

1. **בקרת גרסאות**  
   לצורך בקרת הגרסאות נשתמש במנגנון הפופולארי Git ובשירותי האחסון של GitHub.

###### 

###### \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

###### חתימת הסטודנט חתימת המנחה האישי

1. הערות ראש המגמה במכללה

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

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

1. **אישור ראש המגמה**

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

1. **הערות הגורם המקצועי מטעם מה"ט**

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

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

1. **אישור הגורם המקצועי מטעם מה"ט**

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

שלמי תודה:

"אילו פינו מלא שירה... אין אנחנו מספיקים להודות"

בראש ובראשונה לבורא עולם החונן לאדם דעת, שהביאנו עד הלום.

• להנהלת הסמינר על ההשקעה הרבה, ומסירות אין קץ לאורך כל שנות לימודינו בכל הבחינות.

שתמיד קשובה לצרכים ולבקשות בכל נושא המגמה ואינם משיבים ריקם.

• לגב' גרוס, רכזת המגה, על העמל, המסירות וההשקעה הרבה במשך כל שנות לימודינו. שנתנה

לנו מעל ומעבר ושמה דגש על מקסימום בידע התכנותי.

• למנחה גב' ורדית לוי, שהעניקה מזמנה ומכוחה למען הפרויקט ולוותה לאורך כל הדרך, בנועם

ובמסירות אינסופית.

• לצוות חדרי המחשבים ובראשו גב' אלקריף וגב' כי טוב על הנכונות התמידית ועל השמת לילות

כימים למען הצלחתנו ולהעמדת כל עזרה עבורנו.

• להורינו היקרים ולבני משפחותינו אשר עמדו מאחורינו עודדו, העירו והאירו, ודחפו לאורך כל

הדרך .

תוכן העניינים:

. 1 - הצעת פרויקט  
 - שלמי תודה

. 2 מבוא

. 3 הגדרת דרישות ותאור כללי

3.1 מטרות המערכת

3.2 היקף העבודה

3.3 תיאור חומרת המערכת

3.4 תיאור תוכנת המערכת

. 4 תכנות

4.1 תיאור כללי

4.2 טכנולוגיות מתקדמות

4.3 עקרונות התכנות

services 4.4

. 5 בעיות ופתרונן

. 6 ארכיטקטורת המערכת

6.1 תיאור השכבות

6.2 תרשימים

. 7 מדריך למתכנת

7.1 מבנה הנתונים

7.1.1 טבלאות

7.1.2 קשרים

7.2 צד שרת

7.2.1 המחלקות לפי שכבות

Controller 7.2.2

models 7.2.3

BL­ Business Layer 7.2.4

DL­ Data Layer 7.2.5

7.3 צד לקוח

. 8 מדריך למשתמש

. 9 אבטחה

. 10 בדיקות

. 11 בביבליוגרפיה

. 12סיכום

2. מבוא

כאשר חיפשנו רעיון לפרויקט ,הקו המנחה אותנו בבחירת הרעיון היה

לעבוד על פרויקט שיהיה שימושי, תועלתי במיוחד ויענה על צורך אמתי.

ומאידך מבחינה מקצועית ,רצינו פרויקט המשתמש בטכנולוגיות המתקדמות והחדשניות ביותר

בכדי ספק לשימוש נוח למשתמש במערכת ,וכן בכדי לרכוש ניסיון בטכנולוגיות אלו הרווחות בשוק

המפתחים כיום .

חפשנו רעיון שישלב את שתי עקרונות אלו ואכן לקוח עמותת וישיבת דרכי אבות פנה אלינו בבקשה לבניית מערכת שתספק את צרכי המוסד היומיומיים, תקצר תהליכים וכן תכיל טפסים מקוונים שישלחו לנרשמים.

עד היום מערכת הרישום והנתונים לא הייתה אינטרנטית דבר המקשה מאוד על המזכירה שצריכה להתמודד עם כמויות גדולות של ניירת ויצירת קשר טלפוני עם כל הנרשמים שאורכת זמן רב.

לאחר ניתוח הדרישות וחשיבה על כל הצדדים הרלוונטיים קמה לה המערכת שלנו, מערכת לניהול המוסד ורישום תלמידים בצורה מקוונת. נהנינו מאד מהעבודה על הפרויקט, מהרעיון המקורי, התועלתי והביצוע בכלים חדשניים.   
  
 בתקווה שתתרשמו ותיהנו,

מפתחות הפרויקט

h

. 3

הגדרות ותאור כללי :   
  
ניהול המוסד הוא טורח של ממש כאשר הנתונים מוחזקים בדפים, קבצי word וקבצי .excel כל חיפוש של פרטי תלמיד או יום עיון, הוספת חדשים, עריכתם ומחיקתם לוקחת משאבים יקרים. בפרויקט ניצור אתר המאפשר רישום מקוון למוסד הנשמר אוטומטי בעת שליחת הטופס וכן צפייה בפרטי תלמידים או ימי עיון.

האתר נועד לנהל את צרכי המוסד בצורה קלה, נוחה וידידותית למשתמש.

3.1 מטרת המערכת:

o ניהול מאגר תלמידים, כולל ניהול היסטוריית הפרטים  
o ניהול ימי עיון  
 o טפסי הרשמה מקוונים לימי עיון  
o קליטת נתוני עבר באופן אוטומטי מקבצי אקסל קיימים

3.2 היקף העבודה:

מספר השעות שהוקדשו עבור פרויקט זה עומד על כ 700 - שעות.

3.3 תיאור חומרת המערכת:

מחשב pc / מכשיר נייד בעל דפדפן וחיבור לרשת.

3.4 תיאור תוכנת המערכת:

צד שרת בשפת C # שפה מתקדמת ומלאת פונקציונאליות בטכנולוגיית ASP .net core

שפה פופולארית מאוד בשוק כיום הנוחה מאוד למתכנת ולמשתמש.

צד לקוח בשפת React Typescript בטכנולוגית entity framework .

מסד הנתונים SQL server :

4. תכנות:

4.1 תיאור כללי:

המערכת נכתבה בשפת asp.NET C# בסביבת visual studio 2022

כאשר צד ה­ server פותח בטכנולוגיות­ webAPI בשיטות מתקדמות ביותר ש­

NET . ASP מציעה.

וצד הקליינט פותח ב­ React Typescrupt העושה שימוש בטכנולוגיות חדשות ומתקדמות,

בתוכם: Meterial UI , webAPI , Entityframework

כמו כן בפרויקט זה השתמשנו במודל החלוקה ל 4- שכבות פיתוח:

Client-React .1

Service-webAPI .2

BL .3

DAL .4

את המערכת חילקנו למספר חלקים עיקריים:

React. 1 בשכבה זו קיימים כל המסכים של ניהול פרטי המשתמשים, הישויות

והפעילויות.

weAPI. 2 בשכבה זו מתנהלות קריאות ה­ ,GET ה­ ,POST ה-PUT וה-DELETE מה­ Server ואליו.

BL. 3 המרת נתונים ממודל של ­ Client למודל של­ .EntityFramework

DAL. 4 פניה ישירה אל מסד הנתונים.

4.2 טכנולוגיות מתקדמות:

­ web API \* ממשק תכנות יישומים לשרתי אינטרנט ו/או לדפדפני

אינטרנט. מטרתו לתווך בין צד לקוח ( client ) לצד שרת ( server )

על מנת לבצע את הפונקציונליות הנדרשת לצד לקוח, עיקר תפקודו

הוא בקבל נתונים מצד הלקוח והעברתם לשרת, ולהפך.

­ Entity framework \* טכנולוגית עבודה מתקדמת מול ­ DB

יצירת אובייקטים ומיפוי מול טכנולוגית עבודה מתקדמת של מיקרוספט,

ה Entity framework- מאפשר לטעון את הנתונים מ DB ולעשות להם השמה בצורה ישירה ואוטומטית לתוך אובייקטים

בקוד הממפים את מאגר הנתונים בצורה מידית, השימוש ב Entity framework נעשה באמצעות

הגדרת הקובץ tt.ModelDB קובץ זה מחולל classes לעבודה מול Data Base כך שנוצרת ישות לכל טבלה. הישות הנוצרת מכילה אתחולים, מאפיינים וקשרים עבור הטבלה הספציפית וכן שדות חדשים עבור קשרים. הודות לישויות הללו שנוצרו ע"י הקובץ modle ניתן להגדיר מופע( Instance) מסוג הטבלה.

React \* טכנולוגיה מתקדמת המאפשר פיתוח מלא מקצה לקצה בצד לקוח,

ריאקט מאפשרת לבנות ממשקי משתמש המשתלבים לדפים, מסכים ואפליקציות שלמות.

React TypeScript תומך ב-JSX ויכול לדגמן נכון את התבניות המשמשות בבסיסי קוד של React .

­ Meterial UI\* חבילות componens מעוצבות.

4.3 עקרונות התכנות:

\* מבחינה מקצועית – בחירת שפות וטכנולוגיות שתתאמנה לצרכי האפליקציה, התמקדנו בחיפוש

שפות וטכנולוגיות המתקדמות ביותר התומכות ב oop- ומאפשרות מודולריות ופונקציונליות

גבוהה.

\*בנינו את הפרויקט בצורה מודולרית ביותר ע"י חלוקה – מבחינת מבנה הקוד לשכבות

(DL,BL,GUI) בכדי להפוך את הקוד לנהיר ודינאמי כך שניתן יהיה להבין את הקוד בקלות, לנהלו

לתחזקו ולהוסיף על הפרויקט במידת הצורך.

­ מסכי המערכת נכתבו בצורה פשטנית וברורה ביותר לעין, כך שקלים מאד לשימוש, ונותנים

חווית משתמש.

­ הגישה לגבי אפליקציית web היא כמה שיותר כתיבה בצד ה client - מכיוון שכל פניה לשרת

מכבידה על המערכת וגורמת להאטת התהליכים במערכת, בעקבות כך השתדלנו שרוב

הנתונים שלא צריכים לעבור עיבוד בשרת ימומשו בצד ה client . וע"כ השתמשנו בספריית

React כדי לחסוך פניות לשרת.

4.4 Services:

­ שירות שליחת מיילים, המערכת תשלח מייל מ- outlook כאשר משתמש מעונין להירשם לישיבה   
 או לחילופין להירשם ליום עיון מסוים.

.5 בעיות ופתרונן:

בעיה:

כיצד הנהלת העמותה תתמודד עם קליטת כמות גדולה של נתוני עבר המנוהלים באקסל?

פתרון:

המערכת כוללת אפשרות לטעינת קבצי אקסל ומזריקה את הנתונים מהעבר באופן אוטומטי לרשימת פרטי התלמידים.

בעיה:

העמותה תבקש לדעת את סטטוס פרטי התלמיד בזמן נתון, איך נשמור על הנתונים שלא ידרסו בעדכון פרטים?

פתרון:

ע"מ שפרטי התלמיד יהיו מתועדים בכל זמן נתון ולא ידרסו בעריכה המערכת שומרת בכל עדכון את הפרטים הקודמים ומעבירה אותם לקטגוריית היסטוריית תלמיד שם ניתן לצפות בכל היסטוריית פרטיו.

6. ארכיטקטורת המערכת:

6.1 תיאור השכבות:

את כתיבת הקוד כתבנו במודל השכבות, כאשר לכל שכבה יש תפקיד משלה

א. DAL - Data Access Layer ­ פניה ישירה אל מסד הנתונים

ב. BL ­ Business Layer , שכבה זו עוסקת בלוגיקה הקימת בפרויקט.

ג. web API - Application Programming Interface , בשכבה זו מתנהלות קריאות ה­ GET

ה-PUT, ה- DELETE וה­ POST מה­ SERVER ואליו.

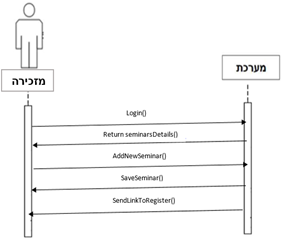
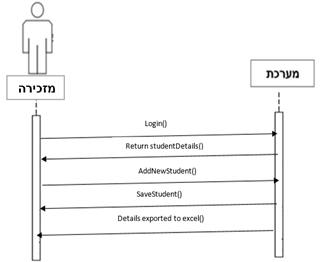
ד. GUI ­ Graphical User Interface , שכבה המספקת ממשק משתמש נוח ונעים לעין,

שכבה זו נכתבה ב .React-

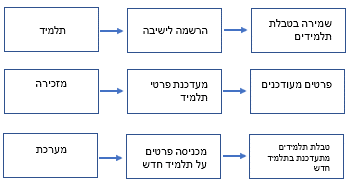
6.2 תרשימים:

אופן זרימת המידע במערכת:

Data Flow Diagram

תרשימי use-case :

Sequence Diagram :

. 7 מדריך למתכנת:

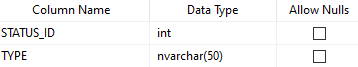
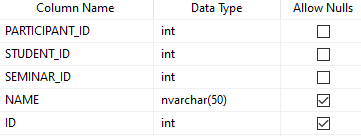
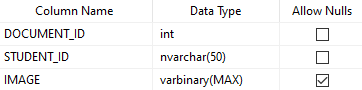
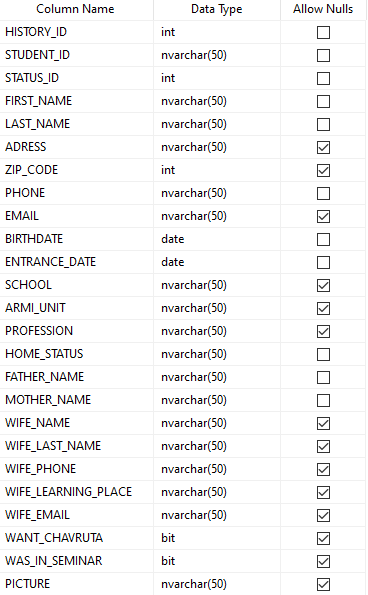
7.1 מבנה הנתונים:

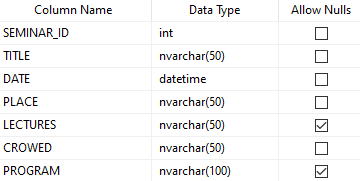
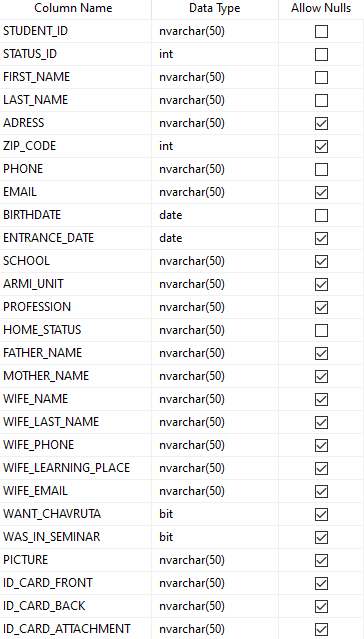
מבנה הנתונים נשמר ב ­ sql server שאליו אנו מתקשרים מהשרת באמצאות

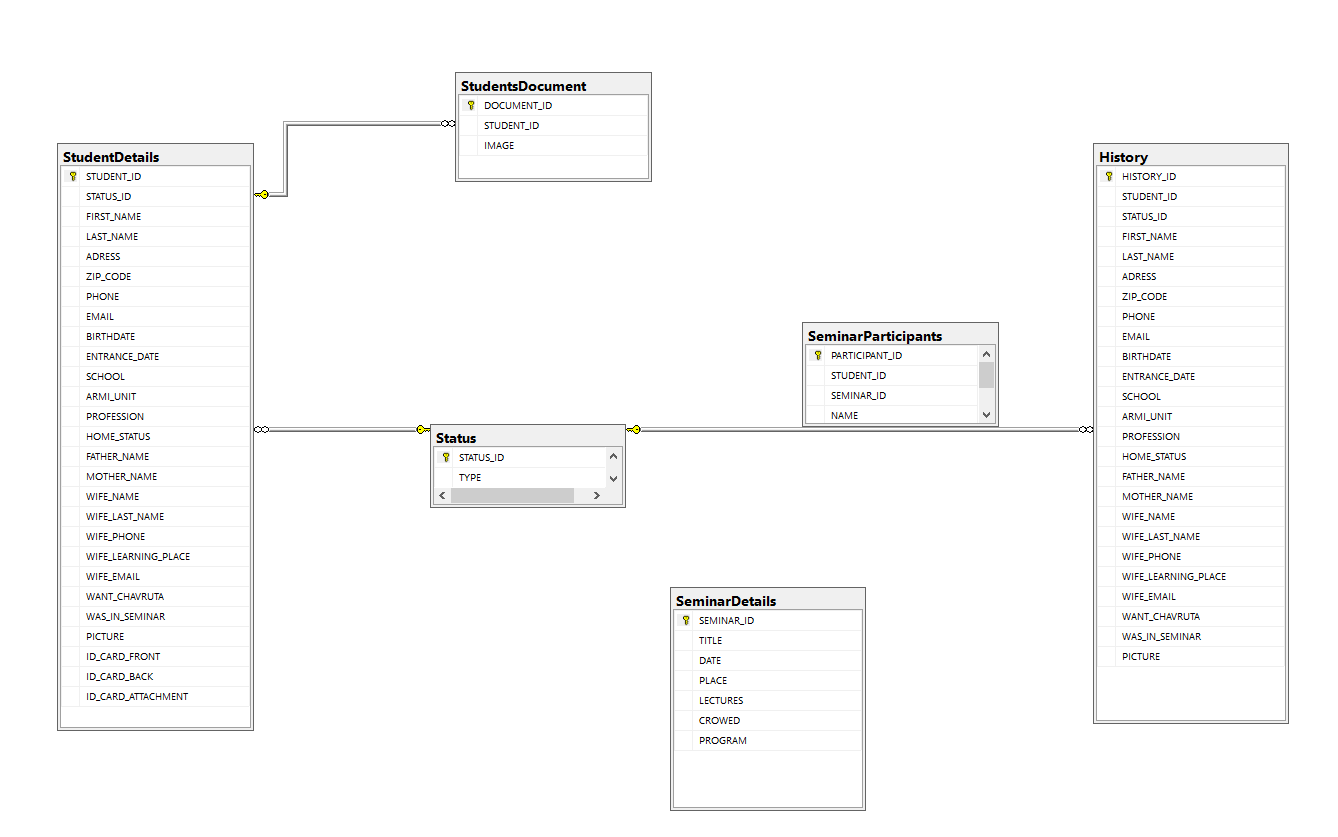
entity framework , סביבת עבודה נוחה ופשוטה הבונה אוטומטית מחלקות

המשתמשות כאובייקטים עבור כל טבלה ב­ DB

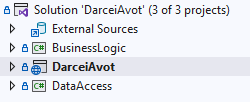
7.1.1 טבלאות:





7.1.2קשרים:

7.1.2 קשרים:

7

7.2 צד שרת:

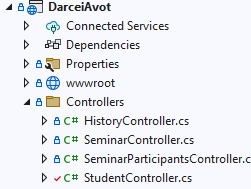
7.2.1: המחלקות לפי שכבות:

7.2.2

:controlleres

DarceiAvot - בשכבה זו ישנו שימוש בשירות webApi המאפשר התממשקות לסביבת הלקוח דרך הדפדפן.

השירות מאפשר קריאות GET,POST,PUT ו-DELETE.

פרוט:

: HistoryController •

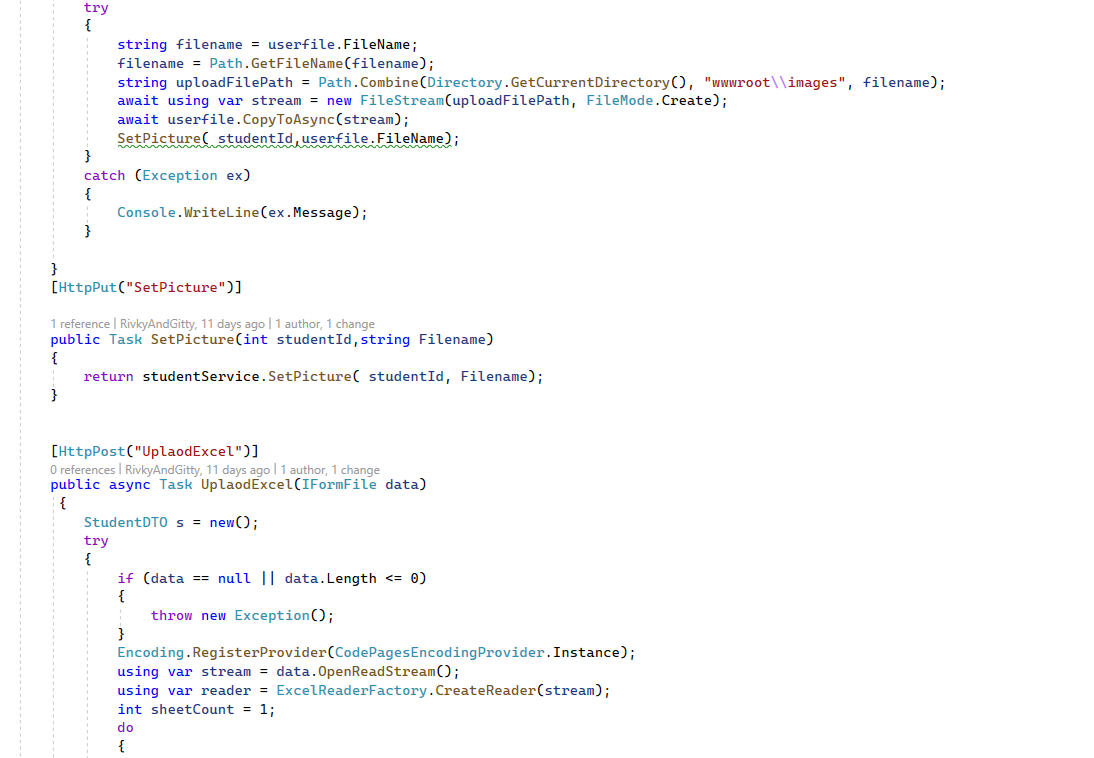
SeminarController •:



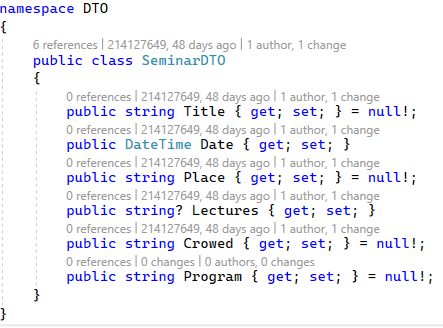
: SeminarParticipantsController•

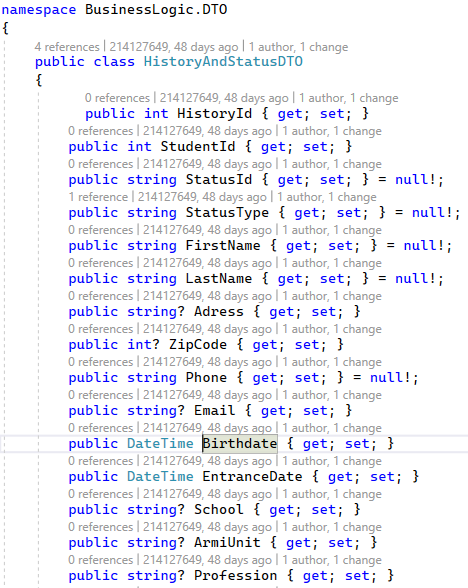


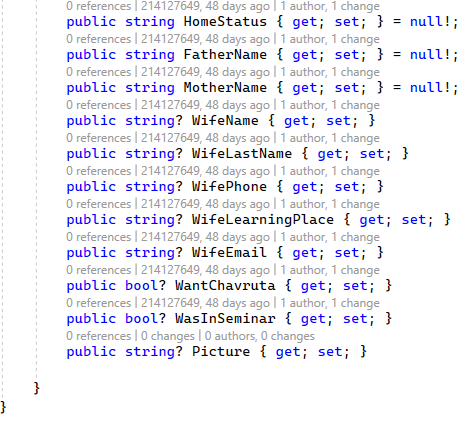
: StudentController•

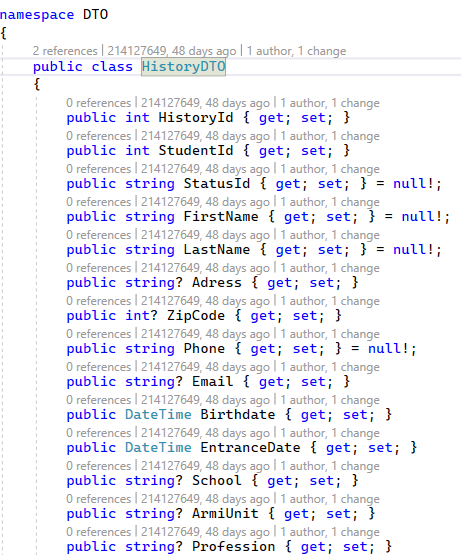


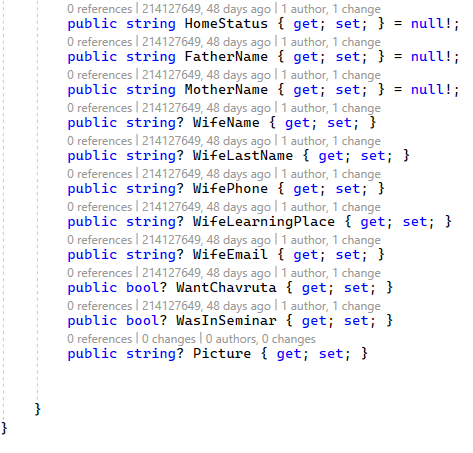
Class SeminarDTO



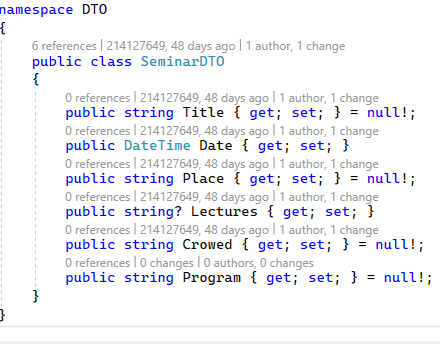
Class HistoryAndStatusDTO

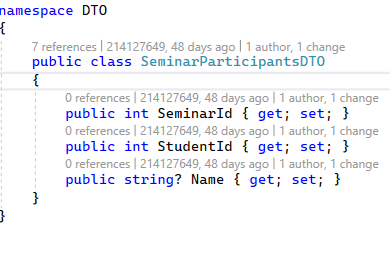


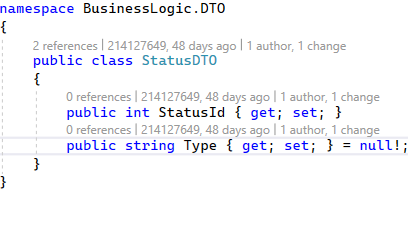
Class HistoryDTO

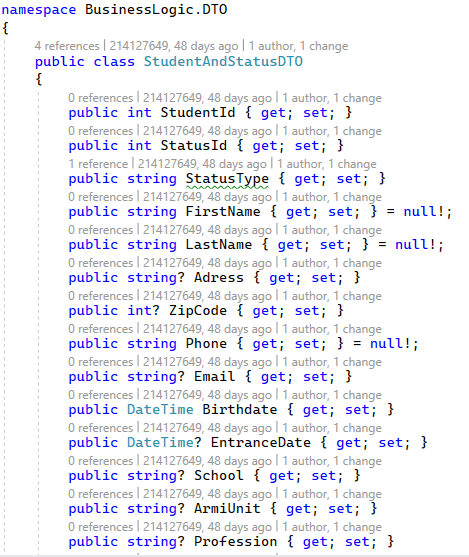


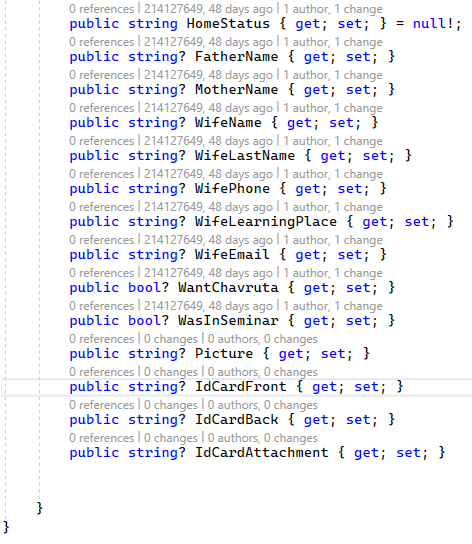
Class SeminarDTO

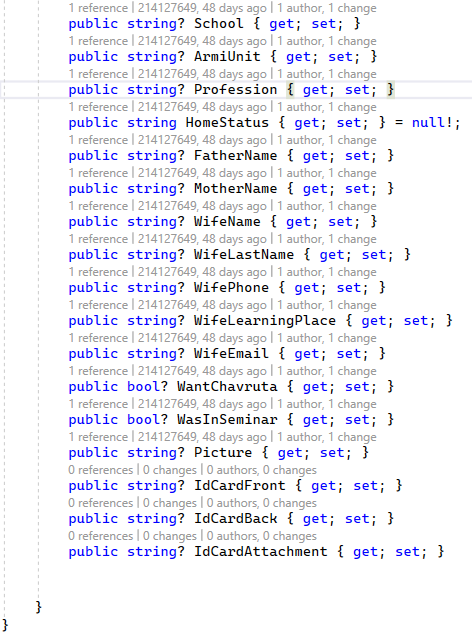


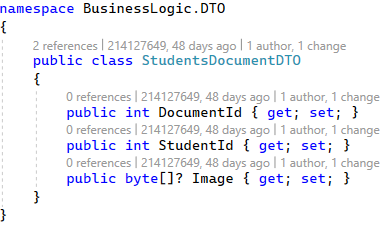
Class SeminarParticipantsDTO

Class StatusDTO

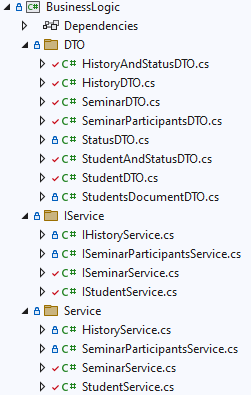
Class StudentAndStatusDTO

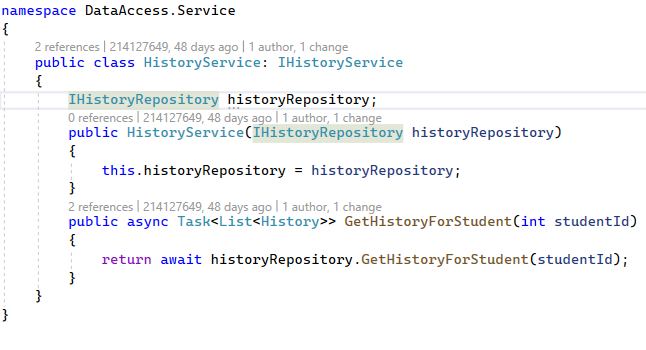


Class StudentDTO

Class StudentsDocumentDTO

Business Layer - BL 7.2.4

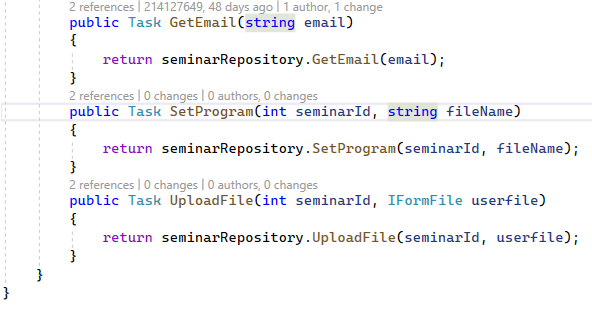


Class HistoryService

Class SeminarParticipantsService



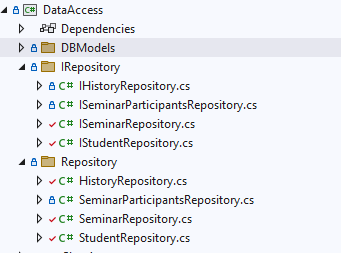
Class SeminarService



Class StudentService

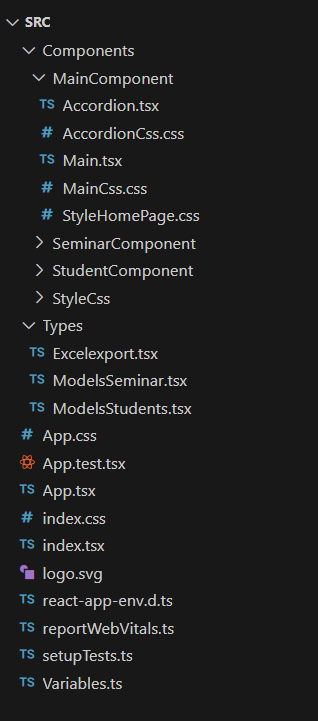


: Data Layer ­ DL 7.2.5



7.3 צד לקוח:

צד הלקוח נכתב בשפת React Typescript   
צד לקוח מספק את ממשקי המערכת והפונקציונליות שמתבצעת אצל הלקוח.  
בניית צד לקוח היוותה חלק חשוב מהפרויקט. את השפה רכשנו ברובה באופן עצמאי עם עזרה מקוונת.  
 הבנייה מורכבת מ- components המכילות קבצי typescript שבתוכם כתובות הפונקציות והפניות לשרת וכן קובץ css לכל קובץ tsx.  
וכן תיקיית types שמכילה את המודלים והממשקים.



דוגמא לקומפוננטה:

קובץ tsx:

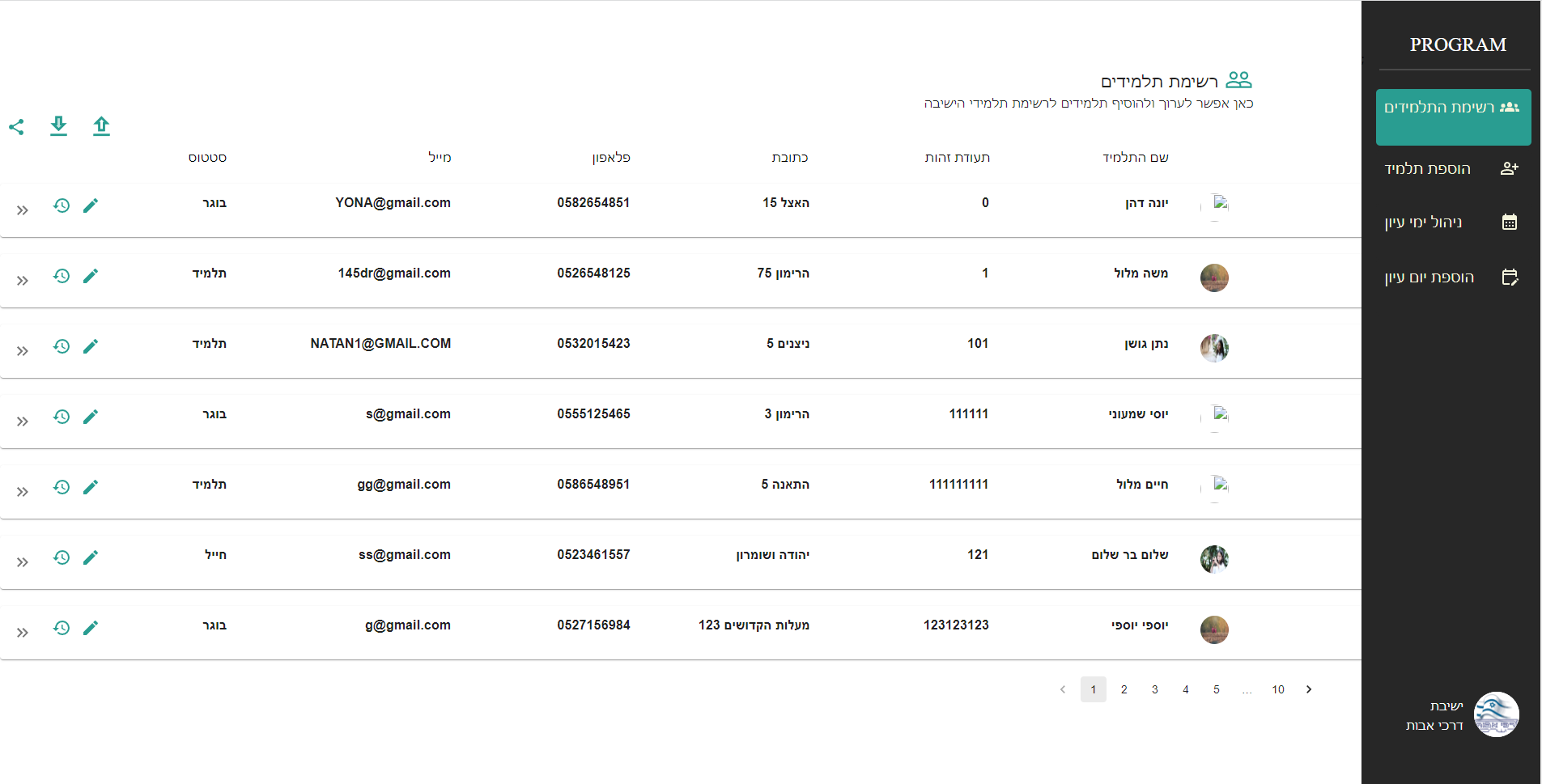
צילום מסך

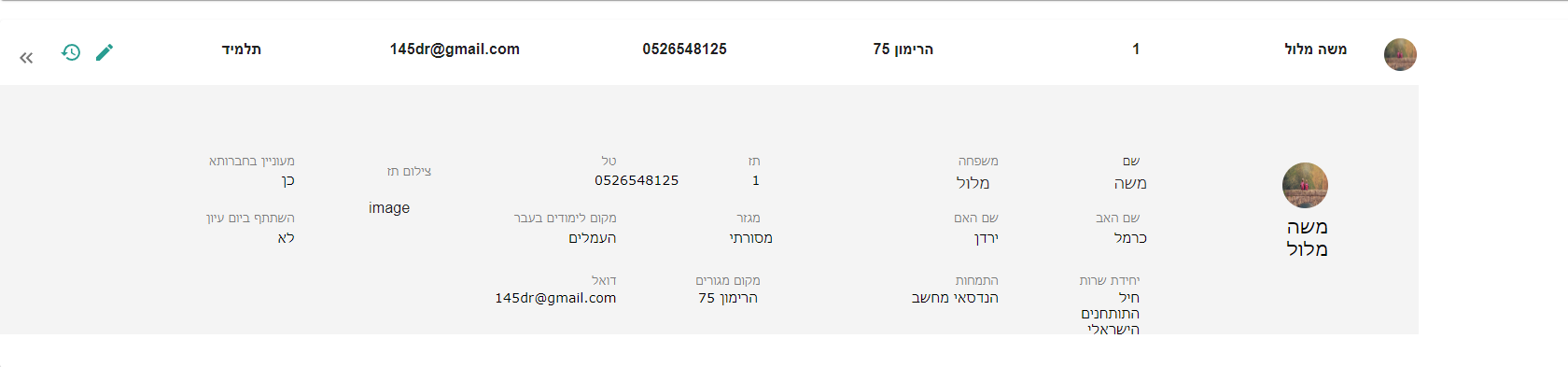
קובץ css:

8. מדריך למשתמש:

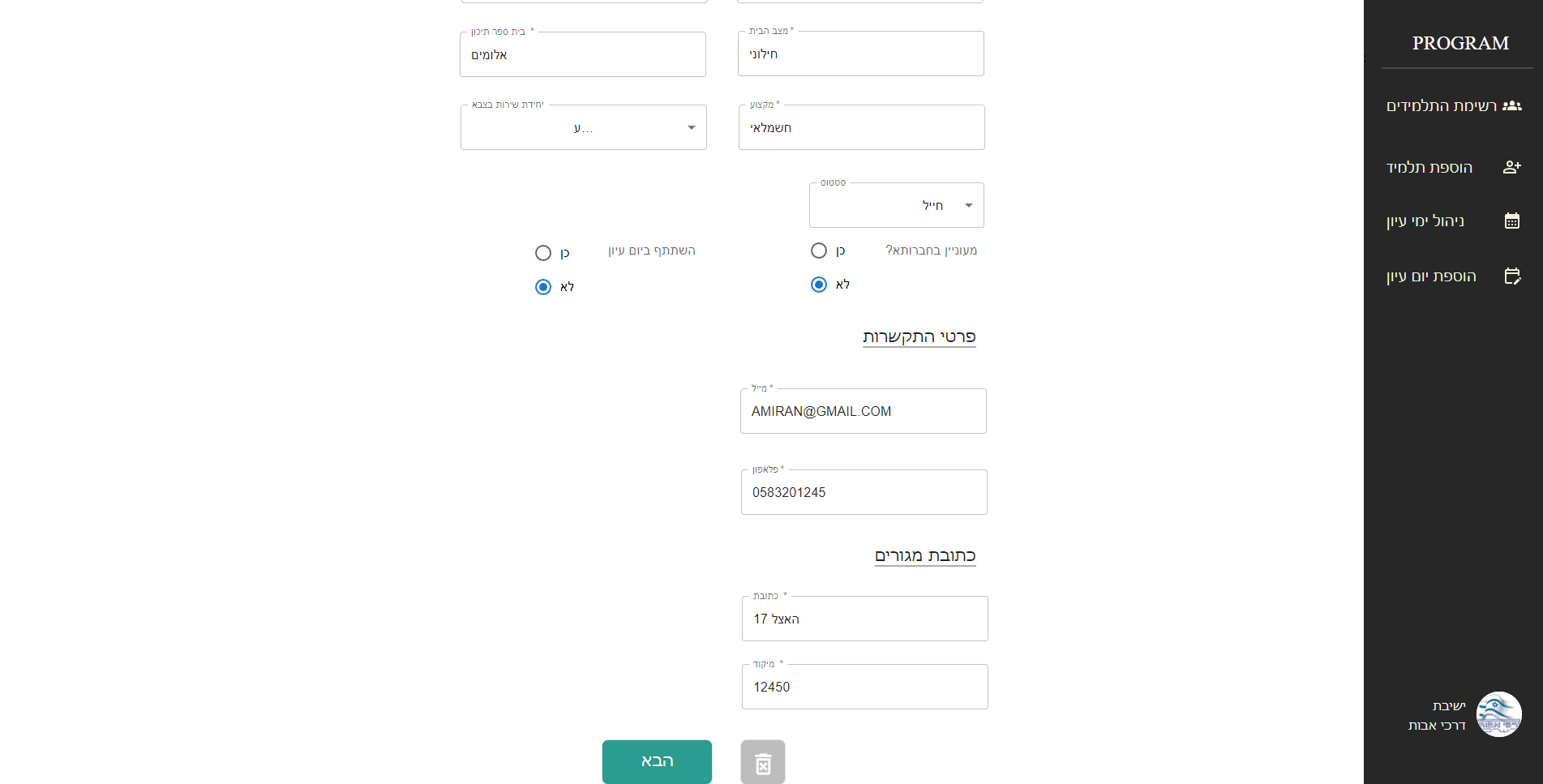
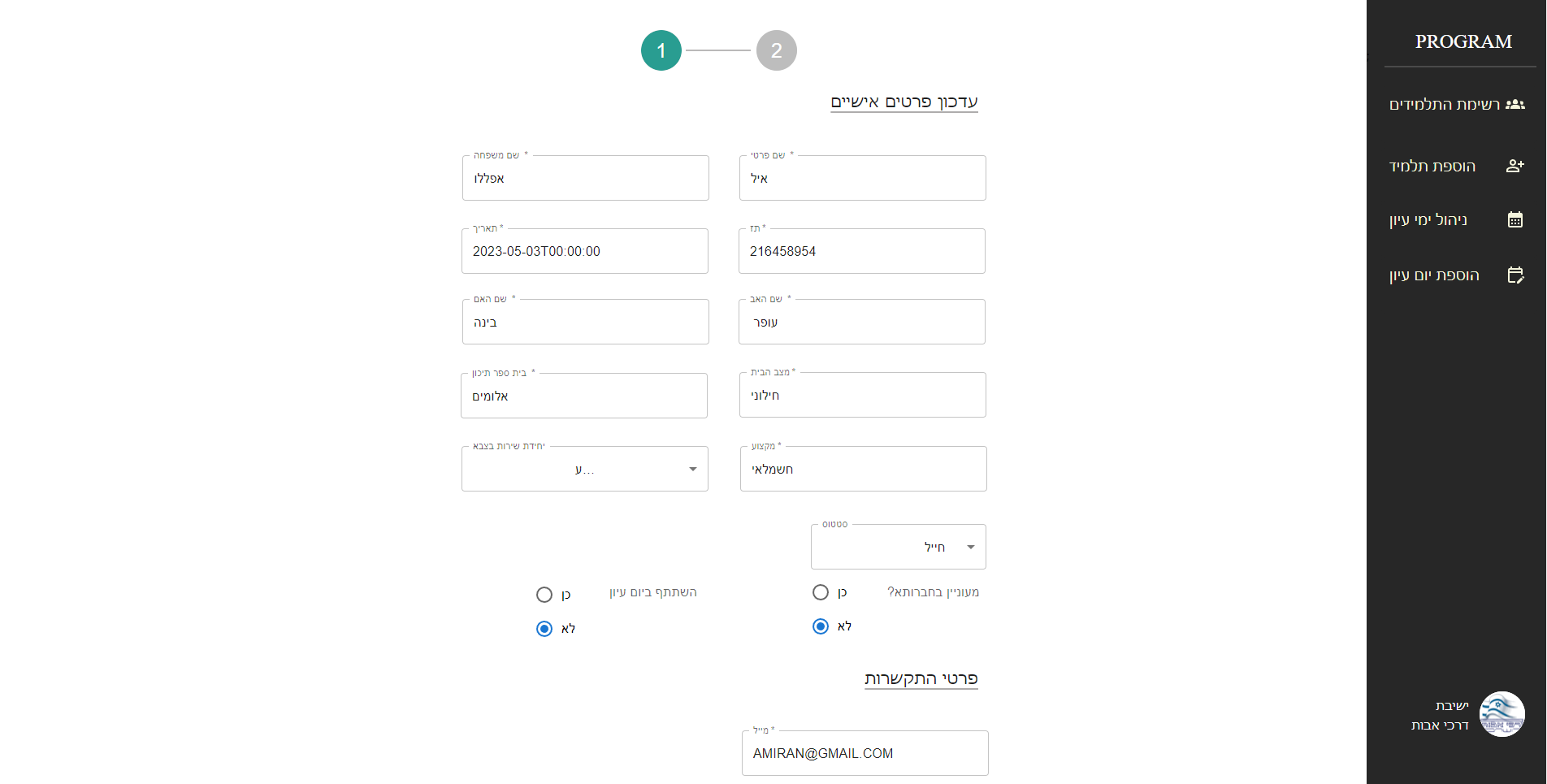
בפרק זה תינתן סקירה המלווה בתצלומי מסך לגבי אופן השימוש באפליקציה והפונקציונאליות שהיא מאפשרת.

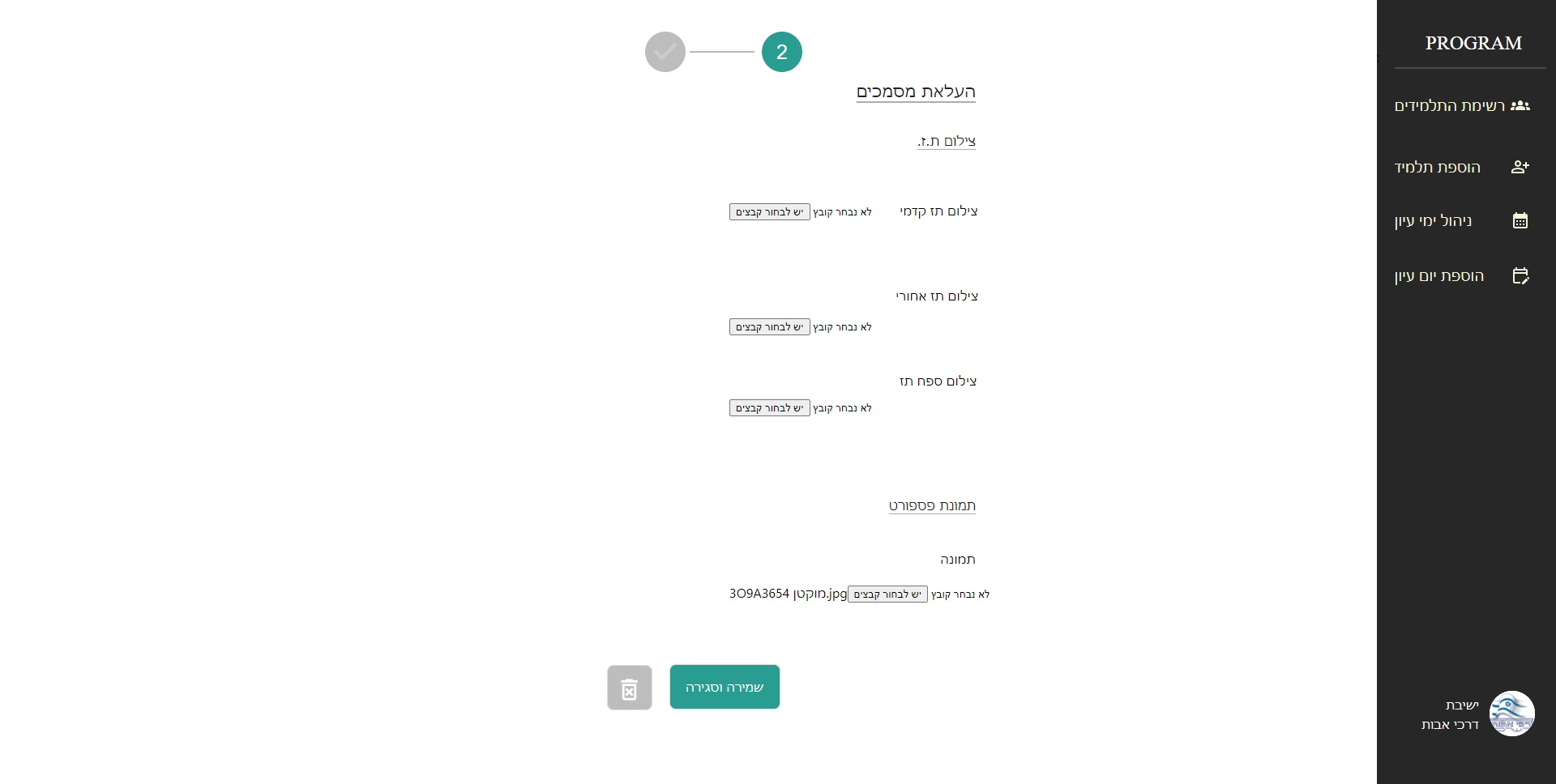
מסך כניסה:  
בכניסתך למערכת יוצג לפניך המסך הראשי שבו יוצגו פרטי התלמידים עם אפשרות עריכת תלמיד, צפייה בהיסטוריית תלמיד, אפשרות שליחת קישור להרשמה בצורה מקוונת, יצוא פרטי התלמידים לאקסל וכן יבוא פרטים מאקסל.  
בכל המסכים יוצג הפוטר הימני שממנו אפשר לצפות ברשימת התלמידים, להוסיף תלמיד, לנהל ימי עיון ולהוסיף יום עיון.

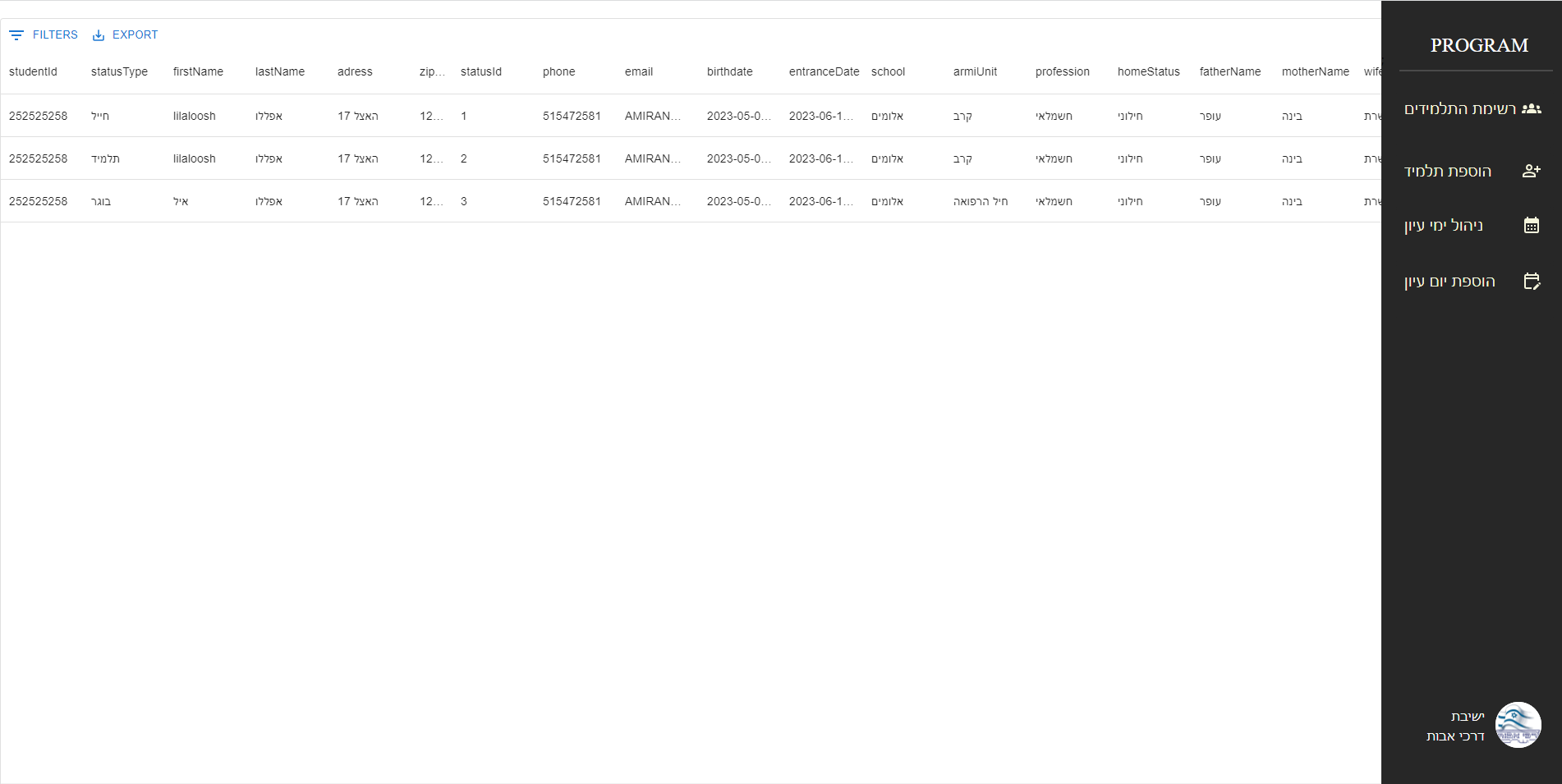


יש אפשרות ללחוץ על החיצים בצד שמאל ולפתוח אקורדיון של הפרטים המלאים

עריכת תלמיד:

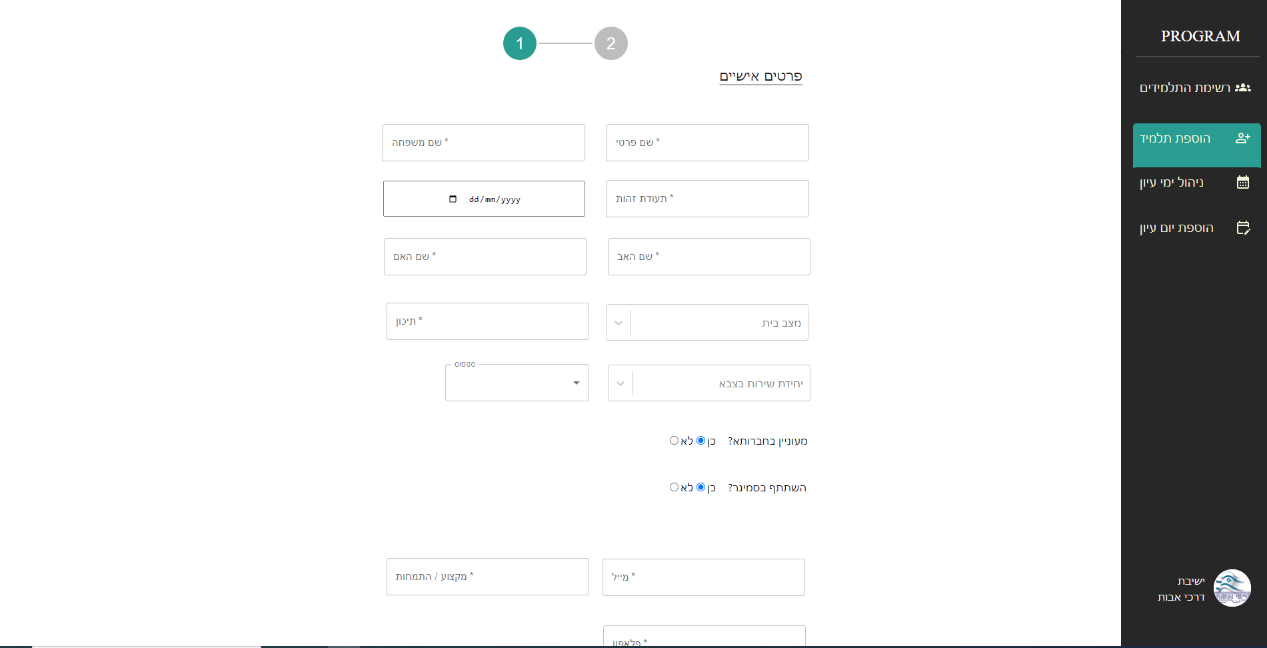


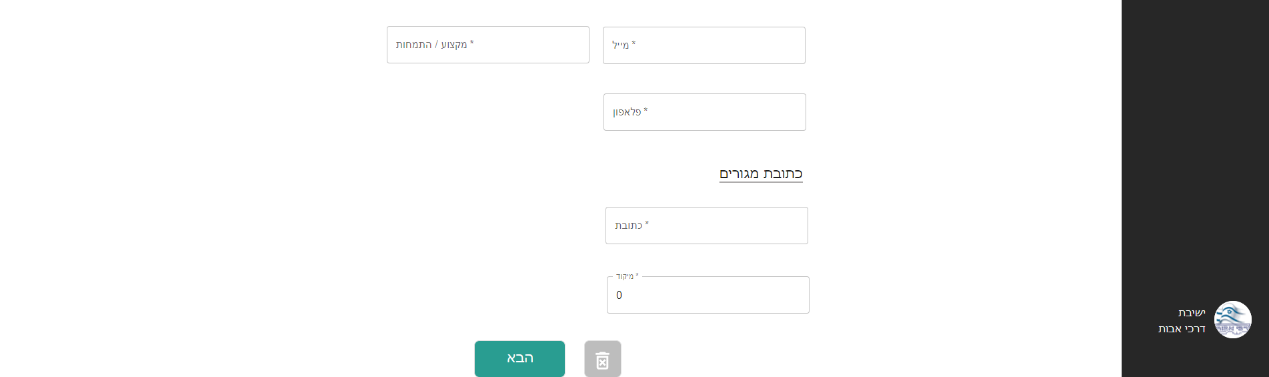
 עריכת תלמיד מסמכים:

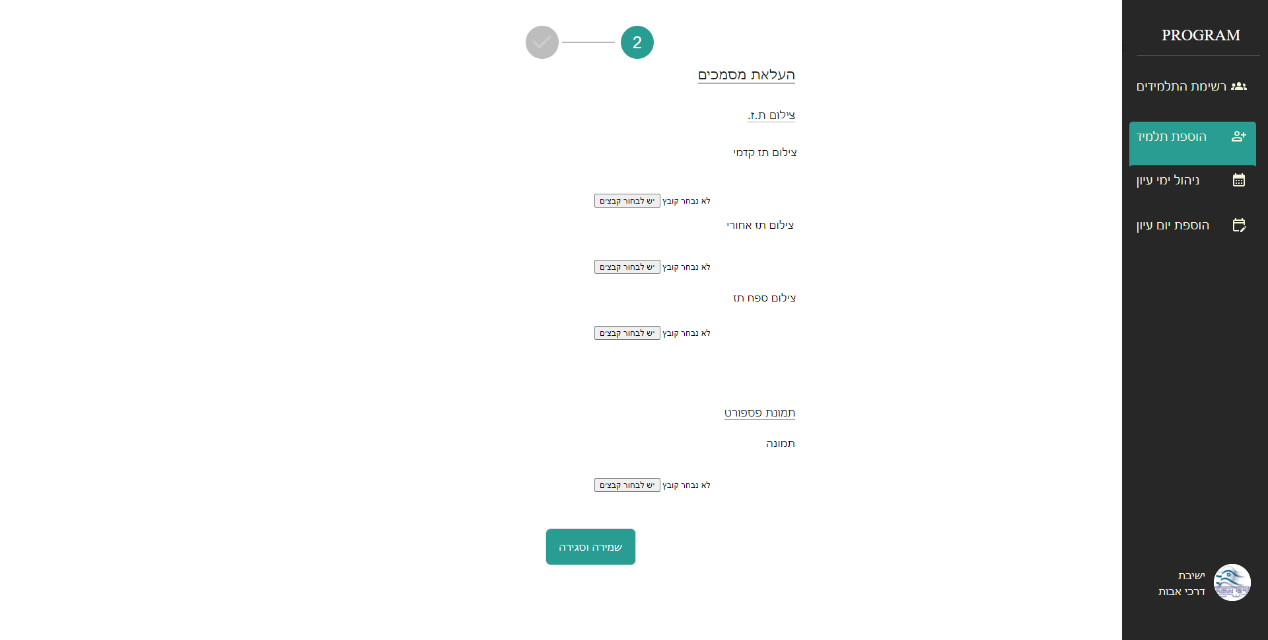
היסטוריית תלמיד:

הוספת תלמיד:

הוספת תלמיד:

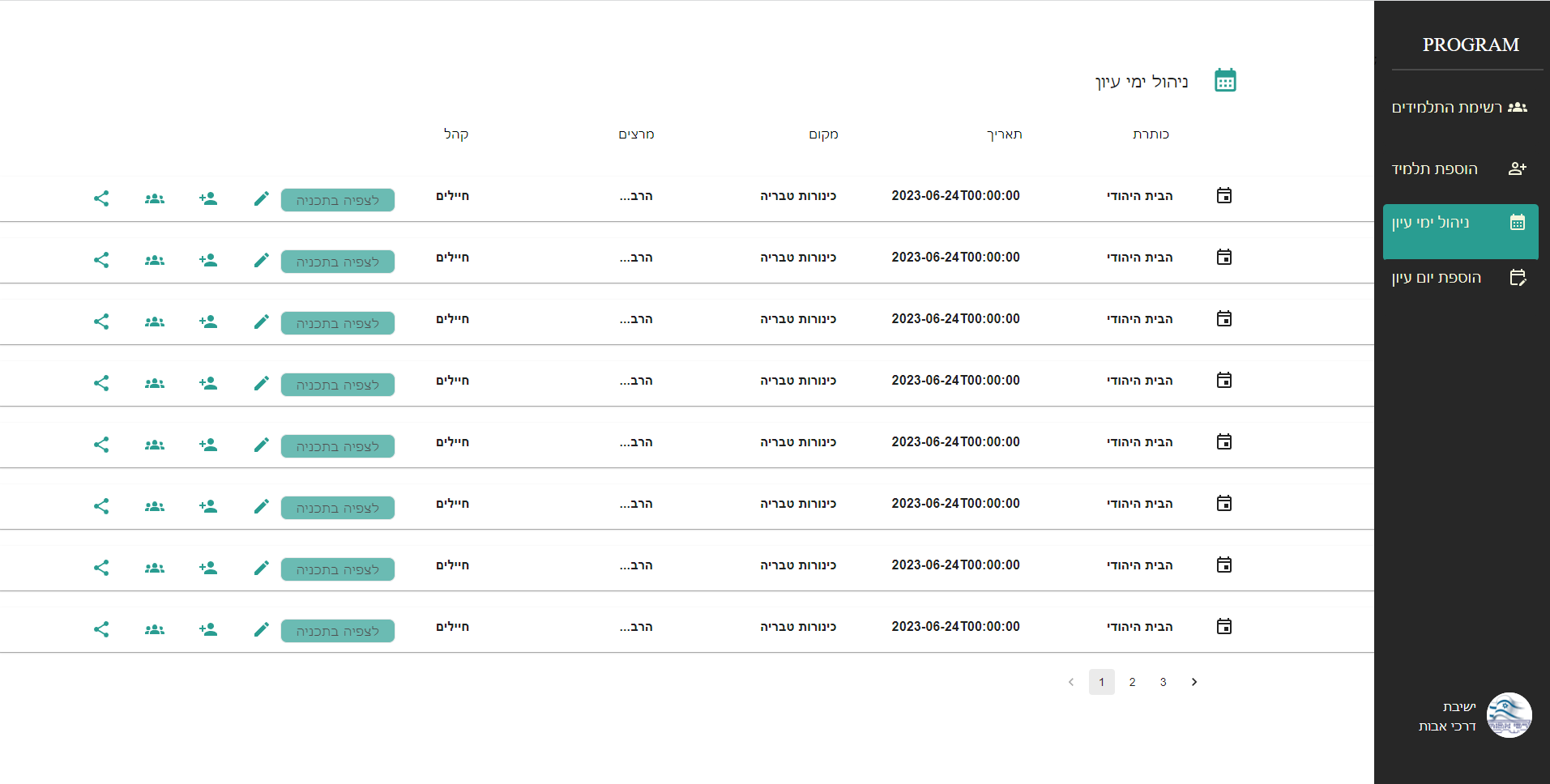


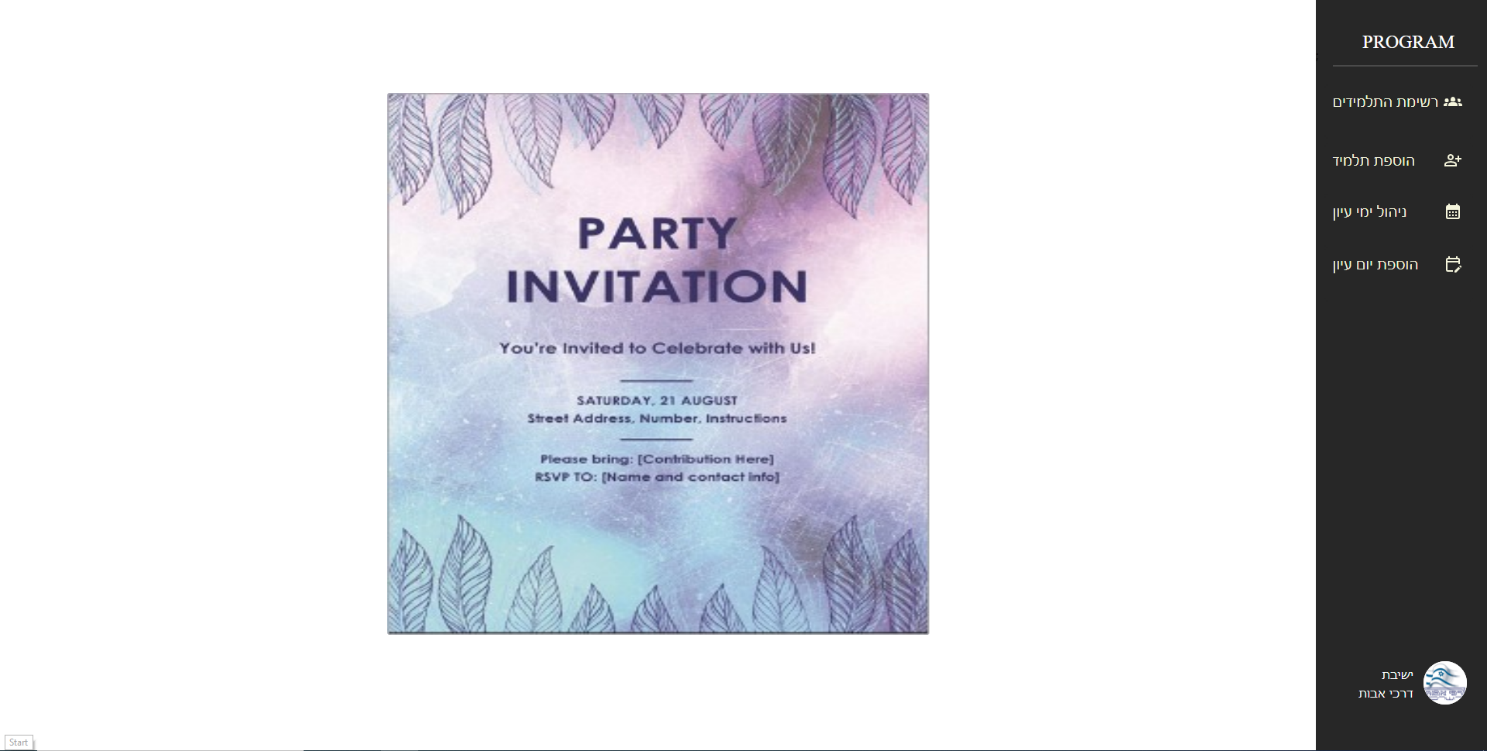


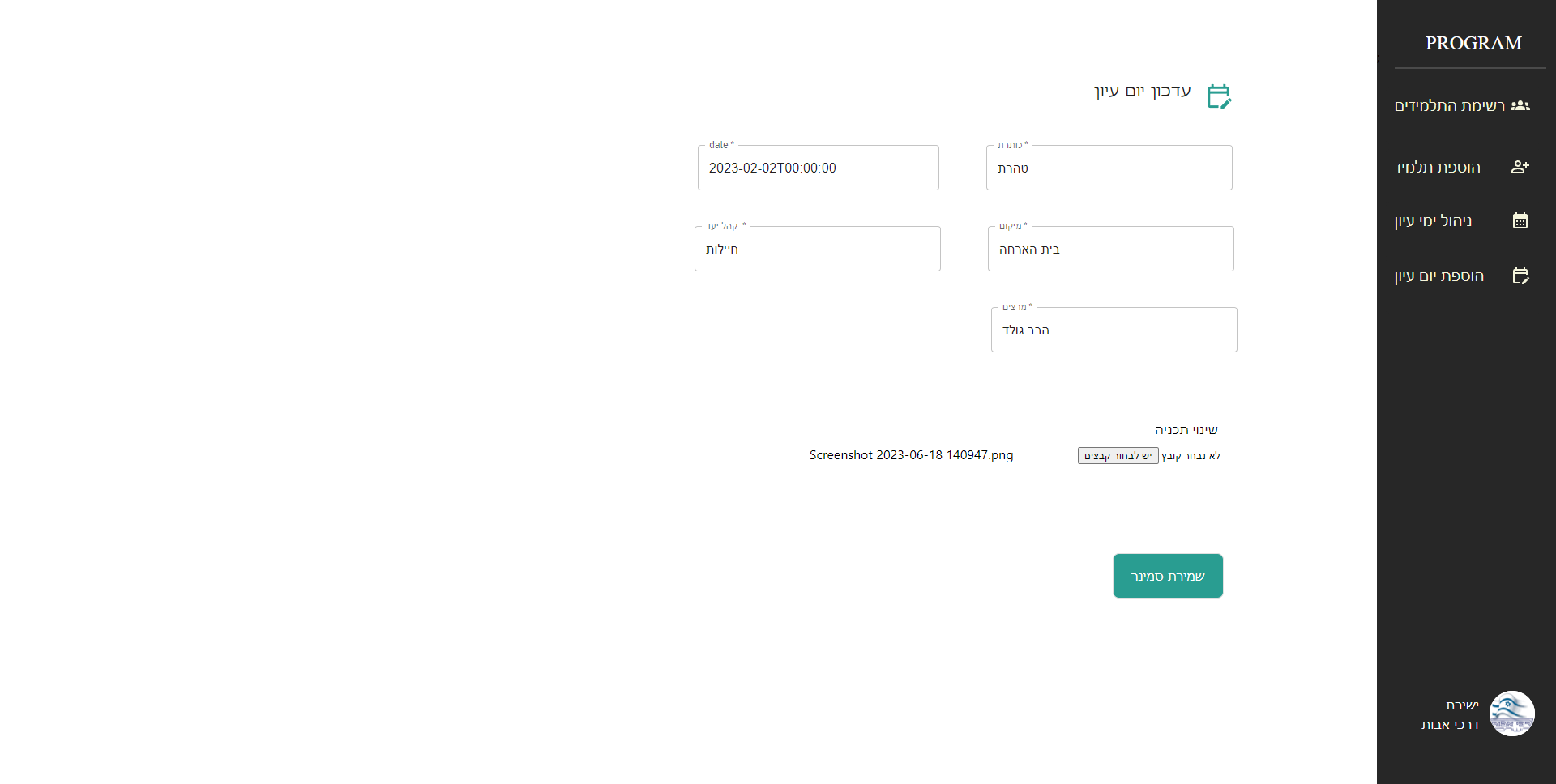
הוספת תלמיד מסמכים

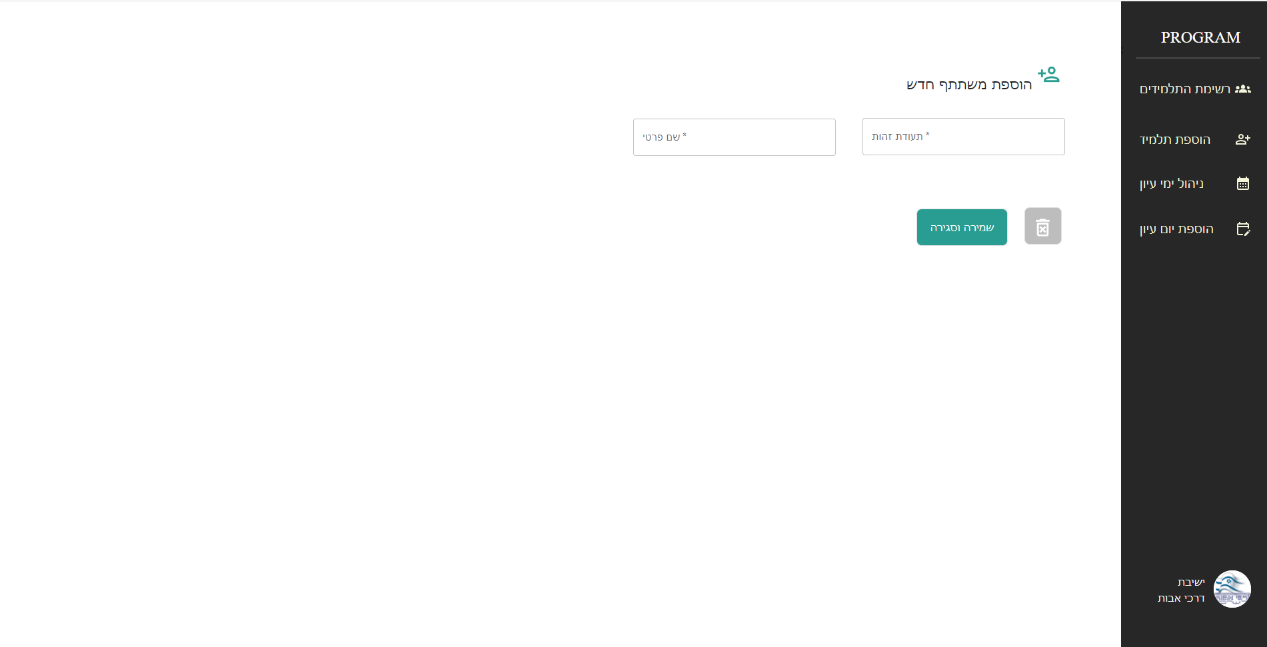
ניהול ימי עיון:

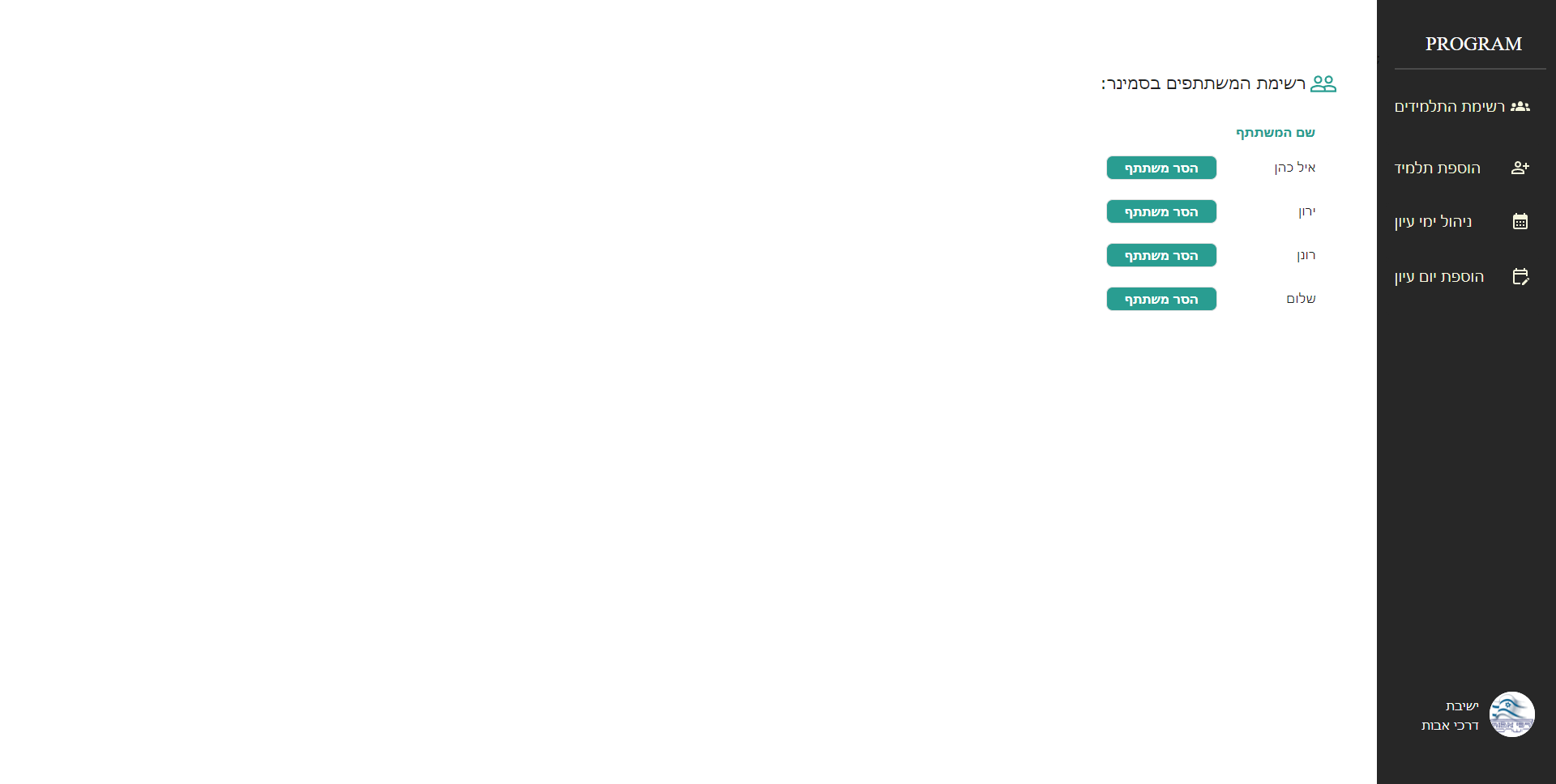
בדף זה ניתן לצפות בכל ימי העיון ופרטיהם, לצפות בתכניה של יום העיון, לערוך אותו, להוסיף משתתף ליום העיון, לצפות במשתתפי יום העיון ולשלוח קישור להשתתפות בו.



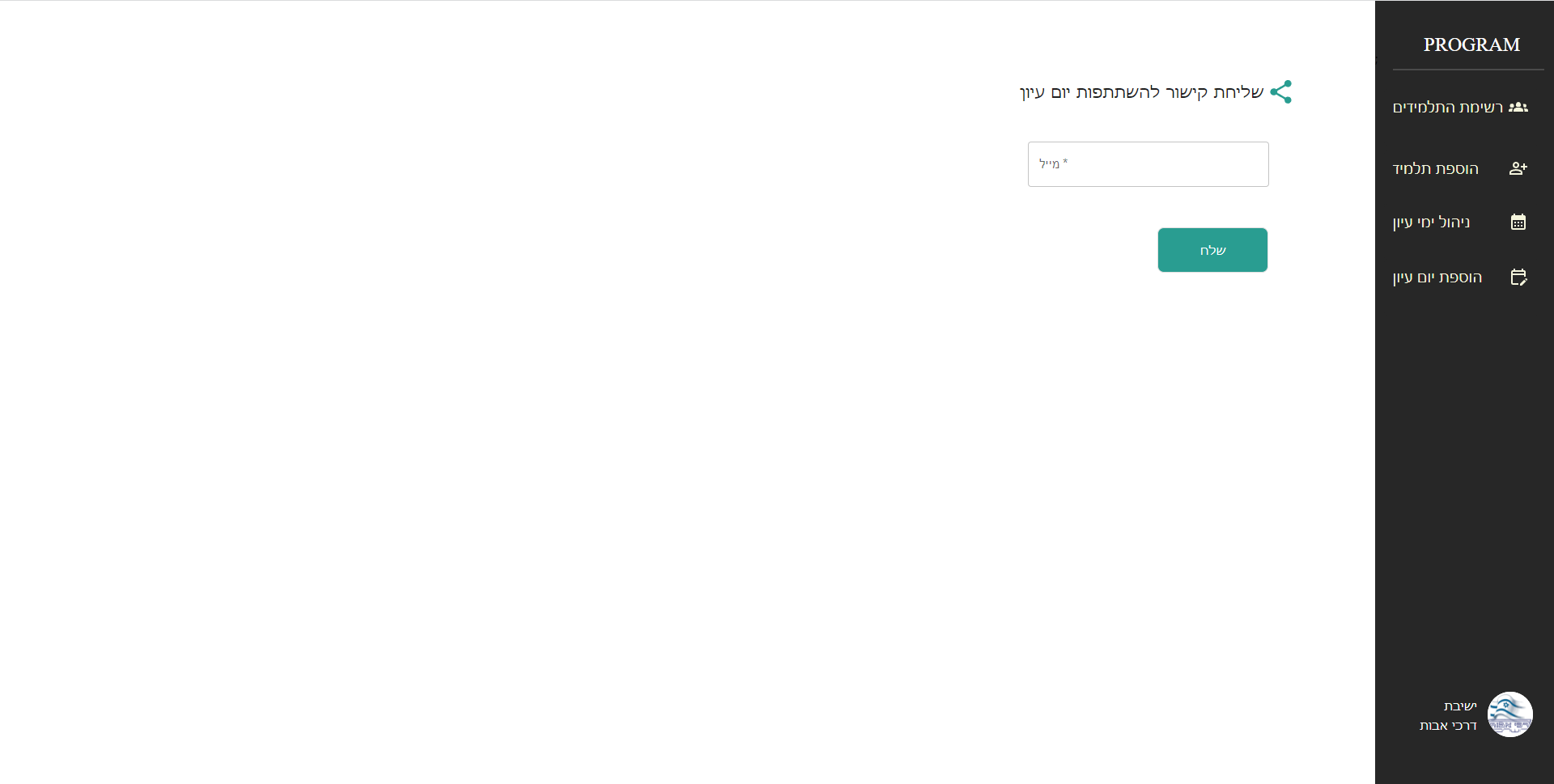
צפייה בתכניה:

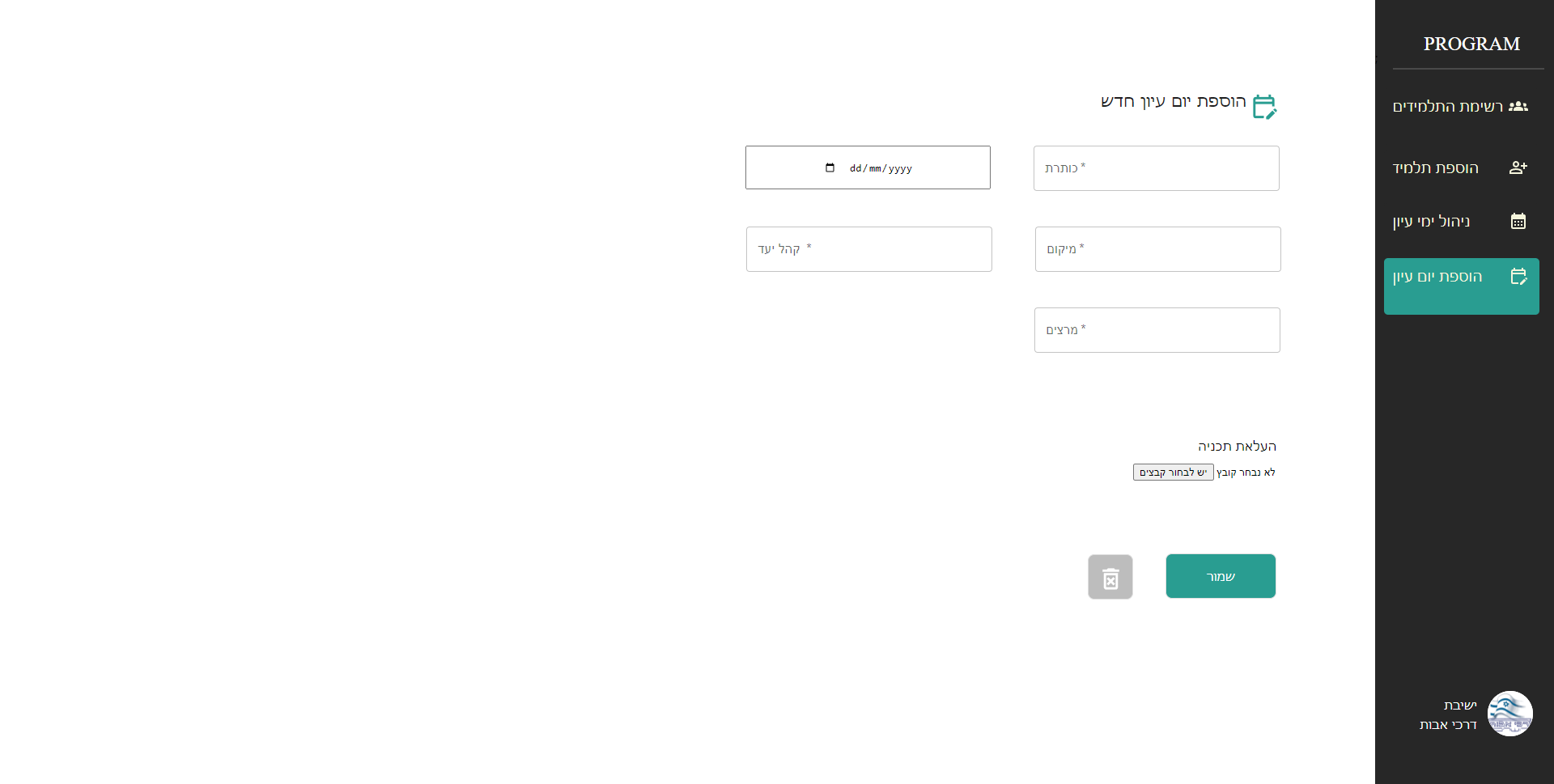
עריכת יום עיון:

הוספת משתתף ליום עיון:

צפייה במשתתפי יום העיון:

כשאין משתתפים:

שליחת קישור להשתתפות ביום העיון:

הוספת יום עיון חדש:

9. אבטחה:  
במהלך הפרויקט שמנו דגש על אבטחת הנתונים.  
לשם כך בחרנו להשתמש בטכנולוגיית Web Api מכיוון שהשימוש בטכנולוגיה זו מחייב שימוש בפרוטוקול https- פרוטוקול בטוח לשימוש ומוכר בעולם התכנות.

כמו כן בשליחת הנתונים מצד לקוח לצד השרת הנתונים מועברים כאובייקט DTO ולא כשרשור גלוי בשורת URL.

10. בדיקות

בדיקה 1- בדיקת שפיות:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| מספר שלב בבדיקה | שלב | תוצאה צפויה | תוצאה בפועל | עבר / נכשל | באגים |
| 1 | כניסה למערכת | נכנס | נפתח העמוד הראשי | עבר | 0 |
| 2 | בחירת תלמיד מסוים | הצליח | הצגת פרטי התלמיד | עבר | 0 |
| 3 | לחץ על רישום תלמיד | הצגת טופס למילוי | הוצג | עבר | 0 |
| 4 | עריכת פרטי תלמיד | נפתח לעריכה | הוצג | עבר | 0 |
| 5 | לחיצה על צפיה בימי עיון | הצליח | נפתח עמוד ימי העיון | עבר | 0 |

בדיקה 2- בדיקת GUI:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| מספר שלב בבדיקה | שלב | תוצאה צפויה | תוצאה בפועל | עבר / נכשל | באגים |
| 1 | הכנסת מייל | נדרשת כתובת באנגלית כולל @ | חייב | עבר | 0 |
| 2 | לחיצה על submit ללא מילוי שדות החובה | לא אפשרי | לא אופשר | עבר | 0 |
| 3 | מילוי שדות הסטטוס ויחידת שירות בצבא | יתאפשר רק מה auto complete | אופשר רק מה auto complete | עבר | 0 |

בדיקה 3- בדיקת פונקציונאליות:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| מספר שלב בבדיקה | שלב | תוצאה צפויה | תוצאה בפועל | עבר / נכשל | באגים |
| 1 | המשתמש נכנס למערכת | יופיע המסך הראשי עם רשימת התלמידים ופרטיהם | הופיע | עבר | 0 |
| 2 | המשתמש לוחץ על החיצים ליד שם הבחור כדי לראות את פרטיו המלאים | יפתח האקורדיון ויופיעו פרטים מלאים של הבחור | הופיע | עבר | 0 |
| 3 | המשתמש לוחץ על עריכת תלמיד | יופיע מסך עם פרטי התלמיד הניתנים לעריכה ואפשרות שמירה | המערכת זיהתה את התלמיד הנבחר ושלפה את פרטיו | עבר | 0 |
| 4 | המשתמש לוחץ על היסטוריית תלמיד | יופיע מסך עם פרטי התלמיד וכל השינויים שנערכו עליו מאז שהוכנס למערכת | המערכת זיהתה את התלמיד הנבחר ושלפה את כל פרטיו | עבר | 0 |
| 5 | המשתמש לוחץ על הורדה לקובץ אקסל | יופיע מסך עם פרטי כל התלמידים יחד ואפשרות הורדה לאקסל | יורד קובץ אקסל עם כל הפרטים | עבר | 0 |
| 6 | המשתמש לוחץ על טעינת קובץ אקסל | יפתח מסך לעיון במחשב ובחירת הקובץ הרצוי להעלאה | פרטי הקובץ הנבחר יטענו אל תוך המערכת | עבר | 0 |
| 7 | המשתמש לוחץ על שליחת קישור להרשמה | יופיע מסך עם שדה למילוי כתובת המייל | תשלח הודעת מייל לנמען | עבר | 0 |
| 8 | המשתמש לוחץ על הוספת תלמיד | יופיע מסך עם שדות למילוי פרטי התלמיד ואפשרות שמירה | הופיע | עבר | 0 |
| 9 | המשתמש לוחץ על ניהול ימי עיון | יופיע מסך עם רשימת ימי העיון ופרטיהם | הופיע | עבר | 0 |
| 10 | המשתמש לוחץ על צפיה בתכניה | יופיע מסך עם התכניה המפורטת של יום העיון הנוכחי | המערכת זיהתה את יום העיון הנבחר ושלפה את תוכנייתו | עבר | 0 |
| 11 | המשתמש לוחץ על עריכת יום העיון | יופיע מסך עם פרטי יום העיון הניתנים לעריכה ואפשרות שמירה | המערכת זיהתה את יום העיון הנבחר ושלפה את פרטיו | עבר | 0 |
| 12 | המשתמש לוחץ על הוספת משתתף ליום העיון | יופיע מסך לרישום המשתתף- מילוי פרטיו הבסיסיים | המערכת מזהה את יום העיון הנבחר ומוסיפה אליו את המשתתף | עבר | 0 |
| 13 | המשתמש לוחץ על צפיה במשתתפי יום העיון | יופיע מסך עם משתתפי יום העיון ואפשרות הסרתם | המערכת זיהתה את יום העיון הנבחר ושלפה את משתתפיו | עבר | 0 |
| 14 | המשתמש לוחץ על שליחת קישור להשתתפות ביום העיון | יופיע מסך עם שדה למילוי כתובת המייל | תשלח הודעת מייל לנמען | עבר | 0 |
| 15 | המשתמש לוחץ על הוספת יום עיון | יופיע מסך עם שדות למילוי פרטי יום העיון ואפשרות שמירה | יתווסף יום עיון חדש לרשימה | עבר | 0 |
| 16 | לחיצה על שמירת יום העיון | הודעה שיום העיון נשמר בהצלחה | הופיעה | עבר | 0 |

בדיקה 4- בדיקת שימושיות

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| מספר שלב בבדיקה | שלב | תוצאה צפויה | תוצאה בפועל | עבר / נכשל | באגים |
| 1 | אחידות בעיצוב | עיצוב אחיד ונעים לעין וכן שמירה על תאימות | בוצע | עבר | 0 |
| 2 | פעולות משתמש | ברירות בפעולות שהמשתמש צריך לבצע | בוצע | עבר | 0 |
| 3 | פעולות משתמש | קבלת התוצאה הרצויה בפועל | בוצע | עבר | 0 |

. 11 בביבליוגרפיה

stackoverflow.com

<https://www.hamichlol.org.il/>

<https://mui.com/>

chatGPT

github.com

12. סיכום:

הפרויקט תרם לנו רבות מכמה בחינות, הן מבחינת הפקה מוגמרת של תוצר, הן מבחינת תכנון המסכים והקלאסים בצורה יעילה ונכונה והן מבחינת הלמידה וההתמודדות עם תחומי ידע חדשים. החכמנו במהלך הפרויקט. נתקלנו בבעיות שדרשו חשיבה יצירתית כדי להגיע לפתרון, מימשנו במהלכו ידע וירטואלי שהגיע דרך הפרויקט לפן המעשי. למדנו דרך הפרויקט שימוש ברכיבים שלא הכרנו, והחיבור ביניהם לבין תוכנה. בנינו את הפרויקט שלב אחרי שלב כפי שמתואר בספר זה. כאשר הפרויקט מתחיל מאפס ואט אט מתחברים כל חלקיו לכדי משהו מושלם. למדנו מטעויות, למדנו משינויים, למדנו מהתכנון המוקדם וכן למדנו להתגמש במהלך הפיתוח. הפרויקט הוסיף לנו רבות. נתרמנו הן כסטודנטיות והן כמתכנתות, וודאי שקבלנו מושגים בהיקף הבניה והפיתוח של פרויקטים גדולים. הפרויקט נכתב בצורה דינאמית ובקלות ניתן לערוך בו שינויים ותוספות. ניתן לסכם שהפרויקט העניק לנו אתגר ועבודה מאומצת כשלצידם גם סיפוק והנאה. ההשוואה בין התכנון הראשוני למוצר הסופי מגלה כי המוצר עומד בדרישות וזהה ברובו לתכנון, הגם שבמהלך הביצוע עלו שאלות רבות לגבי ביצוע טכני ותכנותי בפרויקט. דברים מסוימים שנראו בלתי אפשריים התגלו כברי ביצוע ע"י דרכים יצירתיות ומגוונות ובאמצעות שימוש בספריות וחומרים שהתפרסמו בנושא. במבט לאחור הפרויקט היווה עבורנו התנסות בהתמודדות עם עבודה בסדר גודל, מימוש ופיתוח תכנות מורכב ושימוש ביישומים וטכנולוגיות מתקדמות. את הפרויקט ליוו שעות רבות של תכנון, עמל והשקעה כדי להגיע לרמה תכנותית, מקצועית ומדויקת.