

רות זילבר 212332514

א

BLD

[כותרת המסמך]

[כותרת משנה של מסמך]

שלומית

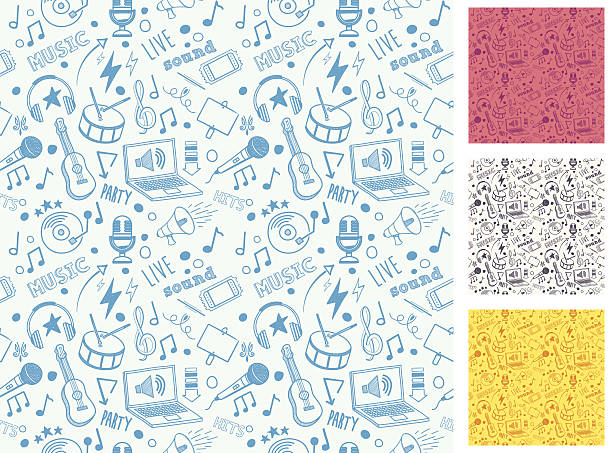
[שנה]





הבלדה שלך | בלעדי.

פרויקט גמר תש"פא 2021







תוכן עניינים

[שלמי תודה 3](#_Toc42618784)

[מבוא 4](#_Toc42618785)

[תקציר 5](#_Toc42618786)

[הכנסת נתונים: 5](#_Toc42618787)

[שיבוץ אוטומטי: 5](#_Toc42618788)

[האתר: 5](#_Toc42618789)

[תיאור המערכת 7](#_Toc42618790)

[קליטת נתונים: 7](#_Toc42618791)

[האלגוריתם: 7](#_Toc42618792)

[הצגת דו"חות: 8](#_Toc42618793)

[מפרט טכני 9](#_Toc42618794)

[תיחום המערכת 10](#_Toc42618795)

[מטרות ויעדים 11](#_Toc42618796)

[מילון נתונים 12](#_Toc42618797)

[ניתוח המערכת 13](#_Toc42618798)

[Use Case Diagram 13](#_Toc42618799)

[קליטת נתונים: 15](#_Toc42618800)

[שיבוץ**:** 15](#_Toc42618801)

[הצגת תוצאות השיבוץ 15](#_Toc42618802)

[האלגוריתם המרכזי 16](#_Toc42618803)

[Data base 23](#_Toc42618804)

[צילום מסד הנתונים 23](#_Toc42618805)

[טבלאות מסד הנתונים 24](#_Toc42618806)

[ניתוח DB 27](#_Toc42618807)

[מבנה המערכת 32](#_Toc42618808)

[ארגון קבצים 34](#_Toc42618809)

[UI – מדריך למשתמש 37](#_Toc42618810)

[פירוט המסכים: 38](#_Toc42618811)

[מסך כניסה – 38](#_Toc42618812)

[כניסת מנהל – 39](#_Toc42618813)

[כניסת הורה – 44](#_Toc42618814)

[פיתוח עתידי 46](#_Toc42618815)

[סיכום ומסקנות 48](#_Toc42618816)

[ביבליוגרפיה: 49](#_Toc42618817)

? **רישום עם ראוטינגס עדכון גודל במסדר נתונים, מקס אן ורצ'ר. לוודא שהמשתמש מקיש מילים שלמות.**

**הודעות המגיעות מהמשתמש והצגתן במסך רק נותר בתחם של שלי**

**ו.... ע"י הקלדת אנטר, איפשור הלחיצה...**

**ההודעה "לא נמצאה התאמה בחיפוש המשופר" ובדיקה מדוע לא.**

**דוחות.**

**שלמי תודה**

תודה לחונן לאדם דעת, המלמד לאנוש בינה.

על הכח, הרעיונות, הבצועים, השמחה, המוטיבציה וכל הכלים שנותן לי חינם. תודה.

אמי היקרה תחי' , תודה! על התמיכה והעדוד, הנתינה והאשיות שבי. כל ששלי-שלך הוא! לתמר אחותי - על האמון, הטוב שבך שמלטף אותי, ההכלה שבך!

תודה לכל האנשים הטובים שנתנו מעצותיהם וידיעותיהם להעשיר את פרויקט הגמר שלי.

תודה לכולכן, צוות המורות. העומד לצידי בכל זמן ונותן את הכוונותיו ועצותיו, במסירות ובנעימות.

גב' גילה לוי, שסיעה בפיתוח האלגוריתם ובשימת לב לכל הפרטים.

גב' מוריה וויס, שסיעה בפיתוח הרעיון ובתמיכה אינסופית בREACT ובWEB API.

גב' רבקה אדלשטיין, שסיעה בבחירת הרעיון ובקריאת ושפור הקוד.

גב' חנה סירוקה, על הידע וההעשרה.

גב' שרה ברלין, שעומדת מאחורי הכל. מרכזת בתבונה ובמסירות.

הצעת הפרויקט | בסעיפים רבים הדיוק עובד בתיק הפרויקט, והורחב באופן נהיר.

סמל מוסד:

541508

שם מכללה :

סמינר שערי דעת בני ברק

שם הסטודנט :

רות זילבר

ת"ז הסטודנט :

212332514

שם הפרויקט :

BLD ballad

**תיאור הפרויקט:**

ריתמיקאיות, מפעילות ואנשים פרטיים רבים משתמשים בשירים שלא קימים במאגרי רשת.

לעיתים רבות, שם השיר לא מופיע ובמקומו קים הטקסט "רצועה 1" או לחילופין: "מחבר לא ידוע" וכד'.

בנוסף, הצורך בשיתוף שירים במאגר ממוסד, תיוגם וסיווגם הכרחי.

מטרת הפרויקט היא לתייג שירים, לאפשר איתור מהיר לפי התיוג ולשיים אותם.

שיתופם באתר הגלובלי, יוביל לאיתור יעיל, מהיר , איכותי ומדויק.

תיאור ברור ומפורט לפי שלבים, בהמשך הספר.

**הגדרת הבעיה האלגוריתמית:**

האלגוריתם ימצא את תגיות השיר לפי פזמון, מילים חוזרות, תדירות מילים ופירוק המלל לפי כללים וחוקים בשפה העברית, חוקי מבנה משפט ושיר ובהתאם למספר ההופעות, חשיבות המילים וניתוחן. בהמשך, עם חיפושי המשתמש, התגיות ידורגו וינוקדו. מיקומן הנכון יוביל לאיתור נכון, מהיר , מדויק ויעיל.

הגדרה נכונה ומדויקת בהרחבה בהמשך הספר.

**רקע תיאורטי בתחום הפרויקט:**

הפרויקט יפצל את טקסט השירים לתגיות לפי אלגוריתם המזהה את הפזמון וקטעים משמעותיים בשיר, משפטים ותגיות מתאימות. עם העלאת השיר לאתר , תנתן אפשרות לאתרו לפי התגיות שהוגדרו בתחילה. בהמשך, עם איתורי המשתמשים האלגוריתם ישתפר.

הרקע התיאורטי הנכון מפורט ומדויק בהרחבה רבה בהמשך הספר.

**תהליכים עיקריים בפרויקט:**

1. העלאת שירים לאתר ע"י משתמש. כדי לשפר ולהבטיח אלגוריתם תקין ומדויק, יש לאפשר שיתוף של הקובץ במאגר הגלובלי הפתוח ברשת.
2. טכנולוגיה שאינה מטופלת כרגע בפרויקט: הפרדת השיר מהמנגינה ותמלול השיר.
3. יצירת תגיות מתוכן השיר ואיתור פזמון ומילות מפתח בעזרת טכנולוגיות לניתוח טקסט והכרת מאפייני מבני המשפטים בשפה העברית. בינה מלאכותית להבנת משמעות התגיות בהתבסס על מחקר שנערך על שירי ילדים.
4. שפור החיפושים והעלאת רמת התגית-הלקוח מאתר ומוריד את השיר הדרוש, ובכך משפיע על חשיבות התגית.
5. שיום סופי של השיר לאחר זמן ונתונים מוגדרים מראש.

**תיאור הטכנולוגיה :**

מחשב: Lenovo

GB 8 RAM

5I Core

מערכת הפעלה: Windows 10

שפת תכנות: # C

**צד שרת :**

צד השרת יחולק לשכבות:

שכבת ה DAL-

שכבת ה BLL-

שכבת ה - Web Api

**שפת תכנות בצד השרת :**

צד השרת ייכתב בשפת C# וייעזר בטכנולוגיית – WEB ABI

**צד לקוח :**

צד הלקוח- UI – יבנה כממשק נוח למשתמש וישים דגש על:

חווית משתמש גבוהה, אבטחת מידע, הימנעות מקריאות שרת מיותרות ובנייה נכונה של האתר עצמו.

**שפת תכנות בצד הלקוח:**

צד הלקוח יפותח ב-React .

**מסד נתונים :**

מסד הנתונים ייבנה בשפת Sql, בסביבת ה-Sql SERVER.

ויכלול טבלאות עבור:

לקוחות, שירים ותגיות.

**מבוא**

בשעות הרבות בהן עמלתי לבחור רעיון, עלו במוחי עשרות אפשרויות. חלקן מוכרות וקימות, חלקן דלות אלגוריתם: אפילו כאלה שדרשו API מורכב ובלתי יעיל, וחלקן פשוט לא ענו לדרישות. תרתי אחר רעיון שימצה את הידע והלמידה הרבה שקניתי במהלך שנות הלימוד והוצאת הדיפלומה.

כשגב' אתי קזן, חברת משפחה, ריתמיקאית ומפעילה, פנתה אלי בבקשת עזרה, הבנתי שאכן. יש לי תשתית ורעיון לפיתוח הפרויקט.

כמו רבות כמוה, מפעילות גני ילדים, ברשותה מאגר ענק של שירים. רבים מהם הועתקו מדיסקים, ובאופן אוטומאטי שוימו כ"שיר לא ידוע Y" או לחילופין "רשימת השמעה X " וכד'. נושאי השירים מגוונים, ועבור נושא מסוים יתכן והשירים שהמפעילה תצטרך כדי להכין שיעור אחד ישולבו ממספר רב של תיקיות. סיווגן לעיתים לפי סוגי שעורים, שנים, נושאים ואופני חלוקה נוספים.

הדלקת השיר ושמיעת המילים, הינה משימה שגוזלת זמן רב בהכנת השיעור מאחר ונדרשת האזנה לכמחצית מאורך רצועת השמע. בממוצע, מבלות הריתמיקאיות כ-60% מזמן הכנת השעור בשיטוטים בין התקיות במחשב ושמיעת שירים רבים, חלקם אף עד סופם, כדי להגיע לשיר הרצוי.

כל שיר יכול להיות מקוטלג בכמה תקיות ותתי תקיות, ושמות השיר דינאמיים. לעיתים המפעילה לא זוכרת את שם השיר וזוכרת רק חלק מתוכו, ובשל זאת זמן רב מתכלה על איתורי השירים.

ערכתי מאות חיפושים בgoogle - ומצאתי אפליקציות ואתרים מגוונים בנושא, כמו: <https://www.musicgrotto.com/what-song-is-this/>, ,<https://support.google.com/googleplaymusic/>, <https://www.midomi.com/> <https://blogs.opera.com/mobile/2016/04/what-song-is-this/>, <https://www.shazam.com/> , <https://apps.apple.com/us/app/shazam-music-discovery/id284993459>, <https://speechlogger.appspot.com/he/>, [https://vocalremover](https://vocalremover/).org/.

אתרים אלו מיודעים ל: זהוי שירים, איתור מלל לשיר, הפרדת המלל מהמנגינה, תמלול קבצי שמע, איתור קבצי טקסט ועוד רבים אחרים, שלא ספקו אותי בשל 3 מגבלות עיקריות: א. שירים רבים הינם פרטיים ובשפה העברית שגוגל לא מכירה חלק גדול מהם ואינם זמינים ברשת. ב. האפליקציות לא מונגשות כמאגר אישי ולא מאפשרות איתור ממוקד או שיתוף שירים בנושאים ספייציפים ג. העלות של כלל השירותים גבוהה, ועבור משתמש פרטי איננה כדאית.

ישנם אתרים רבים, כפי שיפורט בהמשך, המיועדים לניתוח טקסט. לכל אחד מאפיינים שונים שמנעו ממני להשתמש בהם, ודרשו פיתוח קוד חדש. כמו: ניתוח כללי ולא מילה מילה, בשפות שונות, חיובי או שלילי, ללא קוד פתוח, או אפילו העלות הגבוהה!

וכך, לאחר שהתוועדתי למצוקה הרבה של מפעילות הגנים ולחסרון הכלים בתחום, החלטתי ליצור פרויקט בנושא זה.

בתחילה, חשבתי לפצל את מילות השיר לתגיות באופן שרירותי, ובהתאם לחיפושי המשתמשים, לבחור שם סופי לשיר. עם ההתקדמות בכתיבת הקוד, הבנתי שיש צורך לפצל לפי משמעות. על המחשב להבין מהי משמעות של כל מילה ומה חלקה בתגית, ורק כך הפיצול יתבצע באופן מיטבי.

בצעתי תחקיר מקיף סקרתי למעלה מ-1000 שירים מגוונים ובחנתי את תוכנם. במקביל להגשת פרויקט התיכנות האחרון, למדתי כתיבה אריסטוטאלית אצל הסופרת גב' אורית הראל, וכיום אני משוררת בכתב עת לנשות רוח. וכך, עם האהבה שלי לשפה ולאוצר המילים שבה, גמעתי חומרים במשך שבועות שלמים. מנוע החיפוש של גוגל חרק שיניים בלי הפסק, ובעזרת הבורא הגעתי לתוצאות מירביות בכתיבת אלגוריתם הפרויקט.

היו שלבים בהם רציתי לאפשר למשתמש לבחור קטגוריות בעת ההלאת השיר, אך הסקתי במהלך הפיתוח, שדבר זה מלאה את המתמשים ואינו תורם לחוויה גבוהה.

התמקדתי בהמשך בהנגשת פלטרפורמת איתור שירים לפי תגיות נוחה ושיום השירים.

במהלך כתיבת הקוד, אני חשתי נשאבת לתוך מחצב תכנותי עשיר ומרתק. הלהטוטים שבכימוס, המעקבים האינסופיים, השפורים, האפשרויות---

בכל השלבים, השתדלתי להעניק דינמיות למשתמש, ולהעניק חווית שימוש מעולה. המהירות שבפענוח הטקסט, השפור ולכידת השגיאות הוו גם הן אתגר לא פשוט.

שיקול הדעת במיליוני האפשרויות הניצבות , קריטי ומרתק!

התחקירים הרבים, אסופת המסקנות וניתוחן שוב ושוב.

נתתי אמון במשתמש, שיעזור הלאה בשיפור הקוד.

האומנות שהתחברתי אליה ביותר, זו האפשרות לתת לאלגוריתם ללמוד את עצמו ולשפר את שיום השירים עד הבחירה הסופית, באי-תלות מצד המשתמש.

# מטרות ויעדים

היעד שהצבתי לפני היה:

לשיים את השירים ולפצלם לתגיות באופן מיטבי ומהיר, בעזרת אלגוריתם יעיל ככל האפשר, הן מבחינת אופטימליות הפיצול והן מבחינת סיבוכיות זמן הריצה.

המטרה שלי דרכו: רכישת ידע והכרה היקפית בטכנולוגית React שמזנקת ברמת ההתפשטות בשנים האחרונות, בטכנולוגיות נוספות כגון: Entity Framework ,Web Api, data-base, ועוד.

מטרות נוספות מהמערכת:

* חיסכון בזמן בעת איתור השיר.
* פיצול השיר לתגיות בצורה הנכונה ביותר.
* בנית אתר יפה ונוח לשימוש.
* נגישות לכלל המשתמשים הרשומים לשירים המורשים.

יעדי המערכת:

* הצגת השירים ושיומם באופן הנכון והמהיר ביותר.
* המערכת תאפשר לערוך ולעדכן שינויים לפני אתחולה מחדש.
* יצירת ממשק משתמש המאפשר קליטת נתונים בצורה קלה ונוחה עם חווית משתמש.
* הצגת המידע בצורה ברורה ונוחה.

# אתגרים

במהלך חודשי הקמת הפרויקט, עמדו לפני אתגרים, שעם הרבה סיעתא דשמיא צלחתי אותם בגאון.

* החומרים ברשת בשפה העברית מוגבלים.
* כמעט ואין תחקירי שפה על שירי ילדים.
* למידת שפת צד הלקוח נעשתה במקביל לפיתוח.
* צלחתי 4 מבחנים חיצוניים במהלך חודשי הפיתוח.
* על כל צעד ושעל נזקקתי להכרעות משמעותיות, וכתיבת הפרויקט נעשתה באופן עצמאי.
* המחקרים האינסופיים לגבי מבני המשפטים בשירי הילדים תפסו זמן רב.
* למדתי במהלך כתיבת הקוד עשרות קורסים, כך שהיה עלי ללהטט בין המשימות.
* חשובה לי אסטטיות, כך שגם צד הלקוח דרש את שעותיו הרבות.
* הכרת הספריה לניתוח השפה והשמוש בה, תוך יצירת עצמים רבים לצורך השמוש.
* הדיוק בוולידציות ארך שעות ארוכות.

והנה אני היום, מגישה את הפרויקט, בתחושת ספוק ותודה לבוראי!

# מדדי הצלחה:

במהלך החודשים הבאים, לאחר העלאת הפרויקט לאתר ורכישת שרת מתאים, אזמין כ-100 עובדות ריתמיקה להעלות את קבציהן לאתר.

מדי פעם, אבחון את מספר השירים המעודכנים במערכת ואת מספר התגיות הממוצע, וכך אדע האם המוצר ספק את דרישות הלקוח וממלא את מטרתו בהוותו פלטפורמה נוחה ואיתור מהיר לשירים.

# רקע תאורטי :

אלגוריתם פיצול התגיות דרש הבנת משמעות מילות הטקסט ופירוקן לתגיות לפי זה. אתר זה פותח עבור דוברי עברית, שהיא שפה שמית.

שפות שמיות מאוד מורכבות מבחינה לשונית, מורכבות זו מקשה מאוד על הבנת משמעות המילים. לכל מילה בשפה שמית קיימות לעיתים מאות ואלפי נטיות שונות, מנועי חיפוש לא יזהו את כל המופעים של מילת החיפוש כשהיא מופיעה בנטייה לשונית. המילה ”אישה“.לשם המחשה, באחד השירים מופיעה המילה ”נשים“. מנועי חיפוש אשר אינם עושים שימוש בניתוח מורפולוגי לא יבינו את משמעות המילה "נשים" ולא יתיחסו אליה בהתאם, במהלך הניתוח. בנוסף, בשפה שמית קימות מילים רבות בעלות "דו משמעות" ('סלט טעים יותר מפיצה' לעומת 'אני מפיצה את שירי').

קימות מילים בעלות כתיב מלא וחסר, מילים נרדפות ולועזיות, (כורסא בה' ובא' ), ניקוד, רב משמעויות.

חיפוש המילה: "עיר": המילים: "צעיר, עירנות, עיר, עירי , ערים, העיר, העיר, ועירי, ערים, וכלעיר, בעריהם, שערי, מהעיר, הערים, הערות, עירית, עיראק" ועוד אלף צורות יקושרו אליה, כשחלקן מתאימות ממש ולחלקן אין קשר כלל. מה גם שמילים מתאימות לא ימצאו כלל וכך, חיפוש המילה עיר בלי נטיות יחמיץ כ- %80 מהתוצאות.

# מצב קים, ניתוח חלופות מערכתי ותוספת רקע תאורטי :

* הפתרון של **מלינגו** ה - Search Concept של מלינגו מתחבר למנועי חיפוש ארגוניים ומסייע להם להתגבר על בעיות השפה ובכך משפרים משמעותית את תוצאות החיפוש ומיצוי הטקסט. הרכיב מנתח את הטקסט ומנרמל כל מילה לצורת היסוד שלה בהתאם להקשר של כל מילה בטקסט. מופיMEINGO : חברה. משיקים MORFIX BOT המאפשר להבין ולנתח שאלות של משתמשים. **אך פתרון זה מיועד עבור שפת דיבור ולא שפת שירה.**
* vaderSentiment -מנתח טקסט ומסיק האם הרגש המובע בו הוא שלילי או חיובי על פי השוואה ללקסיקון מילים. הקישור: <https://reshetech.co.il/machine-learning-tutorials/sentiment-analysis-in-hebrew-kind-of>. מוביל את המשתמש לקבל את הכלי לצורך כך. **אך פתרון זה מיועד עבור הבנת הרגש מתוך משפט ולא מנתח מילה מילה.**
* גוגל פיתחה מספר כלים לניתוח מכונה. הקישור אליהם:

<https://reshetech.co.il/machine-learning-tutorials/sentiment-analysis-in-hebrew-kind-of>.. כדי להבין היטב מהי הטכנולוגיה שגוגל פיתחה, יש להבין מה פירוש NLP :

NLP היא סוג של [טכנולוגיית בינה מלאכותית](https://learn.g2.com/what-is-artificial-intelligence) שמטרתה להבין את השפה האנושית. NLP מסייע למחשבים להבין, לפרש ולשכפל מאפייני שפה אנושית. בעידן הנשלט על ידי טכנולוגיה, חשוב שמחשבים יוכלו להבין אותנו. [NLP](https://www.g2.com/categories/natural-language-processing-nlp) הוא ניסיון לקחת את תשומות השפה הזולות שלנו ולהפוך אותן למשהו שמחשבים שימושיים יכולים להבין. המשימה הזאת לא תהיה קשה אם כולם היו מדברים כמו רובוטים, אך לא. מרבית האנשים משתמשים בסלנג, יש ניבים שונים, דקדוק שימוש לרעה, ולהשאיר את סימני פיסוק. מנקודת מבט של מחשב, זה כמו להיות מתחיל בשפה חדשה ולהעז למרכז העיר רק כדי להיתקל בביטויים שמעולם לא שמעת לפני כן. אתה לא יודע איך להגיב כראוי.

מאז שהוכנסו אלגוריתמים [של למידת מכונה,](https://learn.g2.com/machine-learning) מחשבים יכולים כעת לעבד כמויות עצומות של מידע כדי לזהות דפוסים ולהבין טוב יותר שפות אנושיות. זה יש מקרי שימוש רבים עבור הייטק, בתחומים כמו: מנועי חפוש, זהוי דואר זבל, עוזרות קוליות וסנון תוכן. [15% מהשאילתות של גוגל](https://searchengineland.com/google-reaffirms-15-searches-new-never-searched-273786) הן חדשות לגמרי. במילים אחרות, מנוע החיפוש מעולם לא ראה את השילוב של מילים המרכיבות 15% מכלל החיפושים.זה לא בגלל שמחפשים נושאים שגוגל מעולם לא ראתה לפני כן. זה בגלל הדרכים השונות שבהן המשתמשים משלבים מילים ושואלים שאלות, הן בכתב והן [בחיפוש קולי](https://learn.g2.com/voice-search-statistics). גוגל ומנועי חיפוש אחרים מסוגלים לענות על השאלות שלא נוצרות כראוי. גוגל מנסה כל הזמן להשתפר בהבנת שאילתות על מנת לשרת את תוצאות החיפוש הרלוונטיות, ועיבוד שפה טבעית ממלא תפקיד גדול במאמץ זה. גוגל הציגה ב 2018 דגם ייצוג מקודד דו כיווני של רובוטריקים, בשם BERT , העוזר למנועי חיפוש להבין טוב יותר את הכוונה, בעיקר עבור חיפושים ארוכים יותר המכילים מילות יחס מרובות. לדוגמה, גוגל כינתה את השאילתה "נוסע ברזיל 2019 לארה"ב צריך ויזה" בפוסט שפורסם לאחרונה. המילה "אל" היא קריטית כאן. לפני BERT, גוגל הייתה חוזרת תוצאות על אזרחי ארה"ב הנוסעים לברזיל. לאחר ברט, גוגל יכולה לזהות את הניואנס הזה ולהחזיר תוצאה רלוונטית ומועילה יותר.

## ה-API של השפה הטבעית של גוגל

גוגל נותנת כמה תובנות מועילות באמצעות מוצר API ענן המציע קבוצה של דגמי NLP מתקדמים. API זה זמין דרך Google Cloud ומשמש מספר חברות למגוון יישומים. גוגל מציעה הדגמה חינמית מקוצרת של ה-API שלה, במגוון שפות שעברית לא נכללת בהן. ישנם ארבעה היבטים המרכזיים ל-API של השפה הטבעית של Google: ישות, סנטימנט וניתוח תחביר וקטגוריות.

### ישויות

ישות יכולה להיות מקום, אדם, ארגון, רעיון או מושג. כל דבר שהוא! ישויות מטפלות ביחסים בין דברים ועוזרות למנועי חיפוש כמו גוגל להבין את הקשרים ביניהם. נקח כדוגמא את הישויות "ראש הממשלה" , "קנדה" , "ג'סטין טרודו". מנועי חיפוש מחפשים [מופע משותף](https://patents.google.com/patent/US5675819A/en) כדי ליצור את קשרי הגומלין בין ישויות. שלוש הישויות המפורטות לעיל קיימות יחד לעתים קרובות מספיק באינטרנט שמנועי חיפוש מסוגלים להגיש לך בביטחון תוצאה בודדת לשאילתה. בנוסף להתרחשות משותפת, גוגל בוחנת גורמים אחרים כמו הסבירות והסמכות של התורם כדי לאמוד את החשיבות של היחסים בין ישויות. ה-API של גוגל בשפה הטבעית מודד את ההבלטה של ישויות שנמצאות בתוכן. Salience הוא ציון של כמה חשובה הישות בהקשר של הטקסט כולו. ככל שהציון גבוה יותר, כך הישות בולטת יותר, ומשמפיעה על מהות המילה בטקסט.

### סנטימנטים וסנטימנט

ניתוח סנטימנטים הוא תחום משנה נוסף של NLP המנסה לזהות דעות ורגשות על ישות בתוך טקסט. סנטימנט הוא הבקיע מ **-1 (רגשות שליליים מאוד)** ל **1 (רגשות חיוביים מאוד)**. Magnitude הוא מספר חיובי המודד את התחושה הכללית של טקסט, בין אם חיובי או שלילי.

בהתבסס על הדגמה גוגל מספקת, זה לא מפתיע הטקסט היחיד שבלט הוא בהתייחסות טלפונים אנדרואיד, אשר סונדאר פיצ'אי ציין משתמשים אהבו. סדר הגודל של הצהרה זו היה 1 והציון היה 0.5.

### syntaxתחביר

ניתוח תחביר מפרק את כל המילים של טקסט ומחזיר את הסוג, מצב הרוח, הקול ועוד.

התחביר נותן תובנות לגבי האופן שבו גוגל מסתכלת על המשפטים ומקטלגות מילים להבנה. התחביר באנגלית ובשפות רבות נוספות, מורכב ברמה סבירה ופתירה, כך שגוגל פיתחה כלי לניתוחו.

### סיווג קטגוריה

קטגוריות חושפות כיצד Google מסווגת טקסט. בתוך הדגמת ה-API שלהם, תראה את הקטגוריה יחד עם ציון ביטחון המסומן בטווח 0-1 . כמו בישויות, לא סביר שיש תהליך ניסה ואמיתי למיטוב לסיווג. עם זאת, קטגוריות יכולות לשמש גם כמדריך נהדר. לדוגמה, כאשר נתון מאמר על גורים שקיבל ציון ביטחון של 0.99 עבור /Autos & Vehicles/Motor Vehicles (לפי סוג)/אופנועים, כדאי לבקר שוב בנושאים ובנושאי המשנה שכיסית (או אולי רק להתחיל את כל הדרך).

* ישנו אתר מעולה לניתוח מילים, אך אין ברשת API פתוח עבורו. <https://hlp.nite.org.il/WebMorphAnalyzer.aspx>
* גוגל השיקה ב2016 שני שירותים,**Cloud Natural Language API** ו-**Cloud Speech API** -Google Cloud Platform. כפי שניתן להבין משמם, שני ה-API-ים החדשים הסובבים בעיקר סביב אלמנטים בתחום השפה שיועדו לסייע לחברות ליצור אינטראקציות מבוססות שיחה בצורה טובה יותר. CLOUD NATURAL LANGUAGE API הינו כלי המאפשר למפתחים להבין בצורה עמוקה יותר את הטקסט ואף לגזור ממנו את המשמעויות הנסתרות. לעומתו, SENTIMENT ANALYSIS זו הבנה כללית של המשמעות הנרחבת של הטקסט שהוזן. אפשרות זו מעניקה למפתחים הבנה מהירה ויחסית שטחית של הנושאים הכללים בפסקה מסוימת. מידע היכול לעזור בסיווג הטקסט, הקצאתו לגורמים הרלוונטים וציון למידת החיוביות או השליליות של הטקסט.
  + **Entity Recognition**: : פירוק הטקסט לישויות נפרדות, אותן ניתן לקטלג לפי קטגוריות שונות כגון אדם, ארגון, מיקום, אירוע, מוצר, מדיה ועוד. חלק זה משמעותי ובעל אינספור מימושים אפשריים. מעבר לכך, הוא יכול לחולל מהפכה של ממש בתחום אחר שצובר תאוצה לאחרונה: בוטים. היכולת לפרק משפטים שלמים לכדי ישויות נפרדות תהווה זריקת מרץ למפתחים בתחום אשר עכשיו יכולים להבין בצורה עמוקה יותר את הדו שיח שמתנהל בין המשתמש לרובוט.
  + **Syntax Analysis**: זיהוי חלקי המשפט וניתוח מורכב של היחסים ביניהם לטובת פירוש והבנה של הטקסט. גוגל מאפשרת לעשות את התהליך בצורה אוטומטית וברמה מרשימה למדי. היכולת הזו מאפשרות לקבל תמונה מקיפה ומלאה של הטקסט והמסרים שמועברים בו, וכפועל יוצא מכך לבצע החלטות עסקיות שונות על סמך המידע המפולטר. ה-API נמצא עודנו בשלב בטא כך שלא הכל עובד מושלם.
  + נכון לכתיבת שורות אלה, Cloud Natural Language API תומך בשלוש שפות בלבד: **אנגלית, ספרדית ויפנית ולא בעברית! יש המתרגמים את הטקסט לאחת השפות הנ"ל ומנתחים אותו, אך אין זה פתרון מספק דיו עבור ניתוח מילה מילה. בנוסף, עלות הAPI של גוגל גבוהה ונדרשת מידי חודש.**
* [ברוקס מנלי](https://learn.g2.com/author/brooks-manley) הוא מומחה לשיווק דיגיטלי ומוביל קידום אתרים [ב-Engenius](https://engeniusweb.com/), המעדכנת את האלגוריתמים שלה מידי תקופה. ברוקס דורש תשלום גבוה על שירותיו שלא מספיק די הצורך את ניתוח השירים, בשל התחום העסקי הרחב בו הוא מתמחה.
* מילון וניתוח מורפולוגי:

# פרויקט השפה העברית הושק על ידי המרכז הארצי לבחינות ולהערכה (מאל"ו) בשנת 2000 והוא עוסק בפיתוח כלים ממוחשבים לניתוח לשוני של טקסטים בעברית ובפיתוח תכנה להערכה ממוחשבת של תוצרי כתיבה (חיבורים). עד כה פותחו במסגרת הפרויקט כלים ויישומים רבים, והם מצויים בתהליך מתמיד של עדכון ושיפור.

## המילון המורפולוגי שנבנה במסגרת הפרויקט מהווה בסיס לניתוח טקסטים בעברית. המילון נוצר באמצעות הזנה ידנית של ערכים מילוניים (שורשים וצורות בסיס) ושל מידע לשוני על כל ערך, לרבות תבניות הנטייה וצורות הכתיב השונות של הערך. המילון מכיל 21,665 ערכים בסיסיים, מתוכם 3,221 שורשים פועליים, 17,074 שמות עצם, 3,289 שמות תואר, 349 תוארי פועל, 8,214 שמות פרטיים ו-953 מילות דקדוק ומילים מקטגוריות מצומצמות נוספות (מילות שלילה, מילות קיום וכיו"ב). המילים מופיעות במילון הן בכתיב חסר הן בכתיב מלא. סך הכול כולל המילון 1,102,830 צורות מנותחות מורפולוגית (נכון לאוגוסט 2019). המילון דינאמי ומתעדכן באופן שוטף באמצעות תוכנה ייעודית. מילון זה ישמש **לפיתוח עתידני** של הפרויקט בחלוקה לקטגוריות. הניתוח המתאפשר בקישור: [מנתח מורפולוגי - מרכז ארצי לבחינות ולהערכה (nite.org.il)](https://hlp.nite.org.il/WebTextAnalyzer.aspx) מספק רק חלק מסוים מצרכי המערכת ובנוסף אינו מאפשר גישה לקוד המקור.

* ספרית HEBREW NLP, <https://hebrew-nlp.co.il/>, הינה ספרית קוד פתוח למפתחים, ומספקת מידע רב אודות ניתוח מלל, כמו: פעלים, עצמים, תארים, מילות חיבור, שמות, מילים עם הופעות רבות, מילת יחס, מספרים, מילות גוף, מיליות, תארי הפועל, פועל התואר, זמנים, פעלי פעול, גופים, סמיכויות, ועוד ניתוחים רבים אחרים. ספריה זו אמנם מספקת חלוקה לתתי משפטים אך רק כאשר מופיעים סימני פיסוק, שאלו כמובן לא יסופקו על ידי תוכנת התימלול. לספריה מספר חסרונות:
  + אין מידע לגבי רצפי משפטים החוזרים על עצמם בטקסט.
  + אין מידע מהן המילים הפותחות / סוגרות / באמצע משפט.
  + אין מידע לגבי משמעות המילה במשפט.
  + האות וו לעיתים מחברת ולעיתים מפרידה. תלוי במילים הסמוכות לה.
  + אין מאגרי מילים עבור מילות קישור, פתיח וסיום.
  + המידע הבא מניתוח כל מילה, מופיע כENUM לא מאורגן. (אינו בינרי).
  + אין חומר לגבי ניתוח טקסט, ובטח שלא על שירי ילדים.

תיאור החלופה הנבחרת והנתונים לבחירתה

כאמור, החלטתי לחלק את מילות השיר לתגיות לפי תוכן כל השיר, בהתאם לאלגוריתם פיצול אפשרי לכל מילות השיר ל"תגיות" באופן מתאים.

לאור הנ"ל, בחרתי בשמוש בספריה HEBREW NLP בשילוב אינם בינארי, שישמש לצורך המרת תוצאות הניתוח ושמוש ידידותי בהן. להלן חלק מהקוד:

בנוסף, השתמשתי במאגרי מילים שלקטתי מהרשת, ובשלוב עם מחקרים ועזרים רבים.



כגון מאגרי המילים הנ"ל:

לקטתי כאמור מהרשת מאגרי מילים בינהם:

>מילים המוכרחות לפתח/לסיים/להיות באמצע תגית.

> מילים היכולות לפתח/לסיים/להיות באמצע תגית.

> מילים שאסור להן לפתח/לסיים/להיות באמצע תגית.

(חלק המילים הן מילות גוף על הטיותיהן, מילות קישור על הטיותיהן, וכד'.)

מילים כמו "אחריך" , "אצלך" , "אותך" , "תחתך" , "מאחורך" , "שלך", "לפניך", "והם" , "זאת" , "לגבי" , "עליך", "בגללך" , "בלעדיך" , "בתוכך" , "הזאת" , "אל", "אשר", "ליד", "תחת", "בתוך", "את", "על", "אצל", "ללא", "הכל", "מכל", "בכל", "לכל", "אל", "כבר", "את", "אשר", "רק", "על", "כן", "לא", "כל", "של", "בתוך", "ליד" , "הזה ", "מעל" , "בין", "בשביל", "אחר", "מן", "כמו", "המה" ועוד מילים רבות.

פיתחתי אלגוריתם לאיתור מילות הפזמון הארוך ביותר בשיר שמש אותי לצורך קבלת 2 מילים, האחת שבדרך כלל תפתח תגית בשיר, והשניה בדרך כלל תחתום תגית בשיר.

בנוסף, חקרתי עקרונות במבנה שירי ילדים כמו:

מילות צווי הפותחות משפטים, אותיות מש"ה ול"ב שבדרך כלל באמצע תגית: וו לעיתים דווקא פותחת, תלוי במשמעות. אותיות חיבור מורכבות יותר כמו "כש. לכש. מש. " , מילות שאלה הפותחות תגית.

כך, לצד ספרי דקדוק (אין חומר די מדויק בנושא שירי הפעלה, בשל היותו תחום חדש ובלתי נפוץ=>) תחקירים וסטטיסטיקות, מיפיתי את הנתונים בתיעוד מפורט.

פונקציות מורכבות רבות שפתחתי, הסיקו על סמך הנתוח הראשוני מימדים רבים, ולפי המילים השכנות למילה, פיצלו / אגדו קבוצות מילים ל'תגית'.

במהלך כתיבת הקוד התייעצתי עם מנחי לשון מקצועיים ואנשי חקר המילים, שהעשירו את הידע שלי וענו על שאלותי. המסע היה מרתק, מה גם שכאמור, אין ברשת מידע בנושא זה: הן בשל השפה, הן בגלל מבנה שירי ההפעלה לילדים, השונה מכל מבנה טקסטי שהוא.

# אפיון המערכת

ניתוח דרישות המערכת

סביבת פיתוח

עמדת פיתוח: מחשב DELL i5

מערכת הפעלה: Windows10

שפת תכנות: c# תוך שימוש בטכנולוגיית REACT .

צד הלקוח נכתב ב – HTML,typeScript

בתוספת ספריות bootstrap - ,jQueri ,React Material

כלי התכנה לפיתוח צד השרת: Microsoft Visual Studio 2017

כלי התכנה לפיתוח האתר: Microsoft Visual Studio Code 2019

sql server :מסד נתונים

עמדת משתמש מינימאלית:

Core i3 or higher , RAM 1024 :חומרה

מערכת הפעלה: Windows10

חיבור לאינטרנט: נדרש.

הסביבה הארגונית:

תוכנה זו פותחה ע"מ לאפשר שיתוף שירים, שיומם, והנגשתם לריתמיקאיות ומפעילות ילדים.

# תיחום המערכת | מודול

המערכת תכלול:

* מאגר שירים גדול, המכיל פרטי מידע רבים עבור כל שיר.
* אוספי מילים קשיחים ועצמים נדיפים השמישים בזמן העלאת השיר בעת הרצת אלגוריתם פיצול התגיות.
* קליטת שירים למאגר.
* ניתוח כל מילה בשיר ועל פיו : אלגוריתם המפצל את מילות השיר לתגיות לפי משמעות המילה, רצפים מוכרחים, כורחי פתיחת תגית וכורחי סיום .
* העלאת ערך התגיות לפי החיפושים וסידור רשימת התגיות תואמות הטקסט שהוקש בחיפוש לפי רמת (ערך) התגית ובהתאמת טקסט מיטבית לטקסט שהוקלד.
* שיום השיר לפי התגית בעלת הערך הגבוה ביותר החל מערך מסוים.
* שיפור התגיות, איחודן או פיצולן, בהתאם לדרישות המשתמש ושביעות רצונו מהשירים שאותרו באופן המסוים.
* הצגת נתונים השירים בצורה טבלאית: שם השיר, 3 תגיות עקריות, וחלונית המכילה את טקסט השיר ברצף בתצוגת תגיות.

נושאים שלא באחריות המערכת:

* המערכת מפצלת את התגיות לפי משמעות, ותקינה בלפחות 70% מהפעמים. לשם כך, פותח אלגוריתם שיפור התגיות ע"י המשתמש כפי שיפורט בהמשך, בהסבר האלגוריתם.
* בשלב זה, המערכת אינה מתממשקת ל API המתמלל את השירים, אלא מתבססת על העלאת קבצי טקסט. ההתממשקות תתבצע בפועל בעת השקת האתר, בשל העלויות הגבוהות.
* האחריות על העלאת שירים ללא זכויות יוצרים על המשתמשים.
* לוגיות ניתוח השירים ופירוקם לתגיות מתבססת על כך שהשירים המועלים לאתר הינם שירי הפעלה לילדים בעברית.
* אלגוריתם "שיפור התגיות על ידי המשתמש" סומך על הנ"ל שלא יבצע שמוש לרעה באפשרות זו.
* כאמור, בפיתוח העתידני :
  + כל שיר משויך לקטגוריה וכל קטגוריה הינה תת קטגוריה לתת קטגוריה אחרת, עד הקטגוריה בעלת הרמה הנמוכה ביותר.
  + לכל שיר ניתוח המנגינה לפי כלי תיכנות המתממשקים למערכת זהוי צלילים.
  + במערכת ניתן לאתר מערכי שיעורי ריתמיקה והפעלה עם רשימת שירים מתאימה. המערכים יערכו ע"י זהוי צרכי המשתמש והבנת הקשרי הנושאים אלה לאלה.

? אפיון פונקציונאלי

יכולות המערכת:

המערכת מורכבת ממספר מודולים מרכזיים:

* מודול התקשרות בין הלקוח לשרת- מודול זה אחראי על מעבר המידע בין צד הלקוח לצד השרת. הלקוח יוזם בקשת HTTP המכילה קובץ JSON לשרת. זאת כדי להתחיל את השידור ובמקרה הצורך, להפסיקו.
* מודול תמלול מלל בממשק מובנה – מודול זה אחראי להמיר את קבצי השמע המועלים לאתר לקבצי טקסט בשפה העברית. מודול זה יהיה שמיש בפועל בעת פיתוח עתידי של המוצר.
* מודול הורדת קבצים למחשב הביתי-מודול זה אחראי להוריד קובץ מאורגן ובו התכנים הנדרשים בהתאם לבקשת הלקוח, מהשרת או הדפדפן לתקית ה"הורדות" במחשב הביתי.

ביצועים עקריים

אופי פיצול השיר לתגיות:

בתחילה, המערכת תבצע חלוקה של מילות השיר לתגיות בהתאם למשמעות המילים והקשרן בשיר.

כדי לפצל את תגיות השיר באופן המיטבי ביותר, האלגוריתם מצמיד לכל מילה את ניתוחה בENUM בינארי, (חלק מהמידע מספרית HEBREW.NLP מותאם אישית וחלקו הרב יותר, מתחקירים שבצעתי בשילוב כללים בשפה העברית, ומאגרי מילים קשיחים שבניתי)

המערכת מפצלת את השיר לתגיות הגיוניות בגודלן (לא יותר מ8 מילים בתגית ולא פחות מ2 מילים) ..

המערכת משתמשת בקבצי טקסט , מנקה אותם מרווחים מיותרים ותווים שאינם אותיות הא"ב / רווחים. קבצים אלו מייצגים את הפלט של העלאת קבצי אודיו תוך התממשקות ל API המתמלל את מילות השיר ומפריד בינו לבין המנגינה.

המערכת תומכת בשירים בשפה העברית בלבד.

אופי איתור השירים

החיפוש מתאפשר לפי מנגנון ההשלמה אוטומטית (שליפה לפי התאמה תבניתית, ואחר כך בסדר לקסיקוגרפי. כל בחירה בתגית, מעלה את ערך ניקוד התגית לשיר זה.

אופי שיפור התגיות:

המערכת תאתר את הטקסט הנתון הנמצא ברצף בין 2 תגיות סמוכות (מתוכן השיר ממש) , ותציג רשימה של שירים בהם נמצא רצף זה.

כאשר המשתמש מבצע הורדה של השיר, מתהווה תגית חדשה בהתאם לתוכן תיבת הטקסט (זאת כמובן בתנאי שיש יותר ממילה אחת, ונמצאו התאמות) . המילים שלפניה / אחריה בתגית מתלכדות לאחת התגיות שסביבן בהתאם לתנאיהן.

כאשר התגית החדשה המתהווה, מורכבת מתגית ראשונה/אחרונה בשיר, או כאשר נותרה מילה אחת בודדה בתחילת/בסוף התגית, יש "לבנות" את התגית בהתאם לנתונים ולפי אלגוריתם מורכב, המיוצג בהמשך בעץ בינארי.

אופי שיום השירים:

כאשר תגית נבחרת עבור איתור שיר מסוים, ומספר התגיות בה הגדול ביותר (כמובן שמטווח כל שהוא) , תגית זו נבחרת להיות כשם השיר.

תקציר

## **לפרויקט זה כמה חלקים:**

## העלאת השירים:

שלב ראשון הוא איפשור שיתוף שירים במאגר. כל משתמש רשום יכול להעלות את שיריו למאגר השירים הגלובאלי או ל"אזור האישי".

## תיוג אוטומטי:

שלב שני הוא פיצול התגיות. אלגוריתם זה מנתח את משמעות המילים ותפקידו להבין את הקשרי מילות השיר ובהתאם לכך- לפצלו לתגיות קריאות וברורות.

אפשור חיפוש:

שלב שלישי הוא הצגת מילות השיר המפוצלות לתגיות בטבלה, כך שכל התגיות ממוקמות ברציפות לפי סדר הופעתן בשיר, וקריאתן ברצף מהווה את תוכן מלל השיר. לכל שיר יבחר שם ושלש תגיות ראשיות באופן די-שרירותי. בהמשך, האלגוריתם ילמד את עצמו לפי בחירות המשתמשים וההתאמה תהיה נכונה.

ניקוד התגיות:

שלב רביעי הוא ניקוד כל תגית בהתאם לבחירתה ע"י המשתמש. בהקלדת טקסט, נשלפת רשימת תגיות בציון מקורן עם אופצית השלמה אוטומאטית לתגית. התגית המתאימה ביותר לטקסט עם הנקוד הגבוה ביותר מוצגת בשורה הראשונה.

שפור התגיות:

שלב חמישי הוא אפשור שיפור התגיות עד להתאמה המדויקת. כאשר המשתמש לא מוצא את השיר הדרוש לו בהקלדת הטקסט בהשלמה האוטומאטית , ויתכן בשל פיצול הטקסט לתגית בתוך הטקסט שהוקלד, ניתנת אפשרות איתור משופרת יותר. המשתמש מקליד בתיבת טקסט את המילים הדרושות, ומתבצע איתור בתוכן השיר ממש. לרשות המשתמש רשימת שירים מתאימה, וביכולתו להתנווט בדף לפי הרשימה. כאשר תתבצע הורדת השיר לאחר איתור זה, התגיות שבשיר משופרות לפי ה"תגית החדשה" שהוקלדה ע"י המשתמש.

שיום סופי של השירים:

שלב שישי הוא התאמת שם לכל שיר, לפי התגית בעלת הנקוד הגבוה ביותר בשיר. שלב זה מתבצע מידי יום ??? בשעה 00:00 . לכל שיר, מותאמת התגית בעלת הנקוד הגבוה ביותר בתנאי שערך הנקודות שנצברו בה גדול מממוצע הניקוד הכללי.

## האתר:

האתר הוקדם עבור מפעילות ילדים וריתמיקאיות.

כל משתמש רשום, יכול להעלות שירים למאגר. במהלך ההלאה, יציין האם ברצונו לשתף את שירו/יו במאגר הגלובאלי ובכך לשפר את אלגוריתם השיום.

במקביל, הלקוח יכול לאתר שירים לפי תוכן, תגיות ואף חלקי תגיות.

## קליטת נתונים:

קליטת פרטי הלקוח בזמן הרישום, לפי התקינויות הנ"ל:

1. שם משתמש שלא קים במערכת בעל מילה אחת.

2. שם פרטי ושם משפחה בעברית/באנגלית.

3. כתובת דוא"ל תקינה ויחודית.

4. סיסמא המתחילה ב@ או ב# ואחריו עוד 4 תווים לפחות, שאינה חוזרת על עצמה.

6. תאריך הרישום מאותחל להיות התאריך הנוכחי.

בעת ביצוע ה LOGIN , על המשתמש להקיש את כתובת הדוא"ל שלו ואת הסיסמא.

בפיתוח העתידי תהיה אפשרות להכנסת קטגוריות, לאפיון המנגינה ולקליטת

האלגוריתם:

**ישנם 3 אלגוריתמים עקריים בפרויקט.**

האלגוריתם המרכזי הינו אלגוריתם פיצול התגיות, האחראי לפרק את כלל מילות השיר לתגיות באופן נכון, קריא והגיוני ככל האפשר. הוא מתבסס על בינה מלאכותית וניתוח טקסט. אלגוריתם זה פורק לתתי פונקציות רבות הנעזרות בENUM בינארי.

האלגוריתם המשני, יועד לשפור המערכת. הוא מקבל טקסט מחרוזתי ומאתר אותו לפי הדרישה בבסיס הנתונים. כעת, אפשרויות פיצול התגית לפי דרישת המשתמש מורכבת מאד, מאחר ויתכן שחלק מהטקסט בתגית הראשונה/האחרונה, יתכן והוא מסיים / פותח תגית, או שנותרת מילה בודדה שאינה תגית. אלגוריתם מורכב זה מיוצג בהמשך בעזרת עץ בינארי בעל 4 רמות מטופלות, בצרוף דוגמאות.

האלגוריתם הנוסף מעלה את ניקוד התגיות בהתאם להקשות המשתמש ובהקשת טקסט, מציג את רשימת השירים המתאימים לפי דירוג התגיות, עם שיפור עתידני ולמידת המשתמש.

## הצגת דו"חות:

במערכת זו, אין צורך בהפקת דו"חות מאחר והאתר אינו שיך לארגון המיועד לייעל את עצמו או לנטר את הנתונים, אלא נועד לספק שיתוף ושיום שירים. ?

הספר:

המעיין בספר זה ילמד על אופן פעילות האתר, האלגוריתם, המחלקות ,הפונקציות, מבנה מסד הנתונים והנתונים שהובילו לאופן הפתרון, הפתרון בעצמו ומדריך למשתמש.

אילוצים

למערכת מספר הגבלות:

* אלגוריתם פיצול השיר לתגיות מוכרח להתרחש תוך כמה שניות, בזמן העלאת השיר. כאשר מספר משתמשים מעלים בו זמנים עשרות שירים, לעיתים נוצר "אפקט השירה" , בו משתמשים שלהם מספר רב של שירים להעלאה נאלצים להמתין זמן רב.
* בעת שיפור התגיות, יש לשמור על סדר לקסיקוגרפי של מספורי התגיות בהתאם למיקומן בשיר. מאחר והמספרים הבאים נתפסים במרבית המקרים ע"י שירים אחרים, נמצא הפתרון הבא: הסרת התגיות מציר מסוים והחזרתן למספרים החדשים שיווצרו.
* כאשר משתמש מוחק את השיר שהעלה, ובמקביל משתמש אחר בוחר בו, המערכת יכולה לבצע הורדה של השיר למרות שהוא נמחק, מאחר והוא כבר הועלה במהלך טעינת הדף.
* אחוזים ספורים של תגיות שלא פוצלו כראוי, והמשתמש מאתר אותם. לשם כך, פותח אלגוריתם לאיתור מתקדם ולשיפור התגיות.
* על המערכת לסנן בזמן אמת את נתוני המאגר, ולאפשר הורדה מהירה: בעת כניסת מספר משתמשים במערכת, יתכן והטעינה תקח כמה רגעים א ויותר , מאחר ונוצרת גישה למסד הנתונים מכמה מקומות בו זמנית בנתונים גדולים.

# מילון נתונים ?סדר לקסיקוגרפי

**שיר-**

**תגית-**

**שיום-**

**פיצול-**

**שיפור תגית-**

**השלמה אוטומאטית-**

**איתור-**

**התחלה-**

**סוף-**

**שיך לפותח-**

**שיך לסוגר-**

**תוכן השיר-**

**אותיות השמוש-**

**מילות יחס-**

**מיליות-**

**מילות גוף-**

**תואר הפועל-**

**שם פועל-**

**וו החיבור-**

**וו מפרידה-**

**ניסמך-**

**מילה היכולה/חיבת/אסור לפתח/לסגר/להיות באמצע של: תגית-**

# תיאור הארכיקטורה

# מבנה המערכת:

הפרויקט פותח בסביבת עבודה NET בשיתוף טכנולוגיית WebApiהמאפשרים בניית אתר של צד **שרת - לקוח** בצורה מובנית וברורה.

צד השרת נכתב בשפת c# בשימוש טכנולוגית Entity Framework הפונקציות בו יבצעו את השיבוץ תוך חלוקה נכונה ומסודרת של הקוד.

האתר – צד הלקוח נבנה בפלטפורמת REACT נכתב ב-HTML ו-Typescript בתוספת ספריות Bootstrap, REACT Material ונעזר בפונקציות רבות לצורך יצירת חווית משתמש גבוהה הן מצב מהירות השירות והן מצד נוחות המשתמש.

באופן כללי הושקעה מחשבה רבה הן ביצירת האתר שיהיה נח וקל לתפעול והן בכתיבת הקוד שיהיה מהיר ויעיל ככל האפשר.

תיאור צד שרת:

צד שרת מחולק כמקובל לשכבות:

* שכבת ה- DAL.
* שכבת ה- BLL.
* שכבת ה- WebApi.

החלוקה לשכבות נועדה כדי להפריד באופן מוחלט בין הלוגיקה של הפרויקט, לבין הנתונים עצמם, ומלבד היתרונות הרגילים בתוכנה [מודולרית](https://www.hamichlol.org.il/%D7%9E%D7%95%D7%93%D7%95%D7%9C_(%D7%AA%D7%95%D7%9B%D7%A0%D7%94)) בעלת [ממשקים](https://www.hamichlol.org.il/%D7%9E%D7%9E%D7%A9%D7%A7_(%D7%AA%D7%9B%D7%A0%D7%95%D7%AA)) מוגדרים היטב, ארכיטקטורת שלוש השכבות מאפשרת גם החלפה או שדרוג של כל אחת מהשכבות באופן בלתי תלוי וללא זעזועים בשכבות האחרות, בהתאם לדרישות או לשינויים ב[טכנולוגיה](https://www.hamichlol.org.il/%D7%98%D7%9B%D7%A0%D7%95%D7%9C%D7%95%D7%92%D7%99%D7%94) ונוחה לאיתור באגים.

הארכיטקטורה של פתרון זה נעזרה בפורמט המכונה "TOP DOWN LEVEL" . עיצוב 'מעלה ומטה' זה, הינו המקור העיקרי של שפות תכנות פרוצדורליות מסורתיות. הטכניקה היא לכתב פונקציה ראשית המתפצלת לתת פונקציות , ותהליך זה מתמשך בצורה רקורסיבית עד לכתיבת פעולות בסיסיות. כך ניתן להיות ממוקדים במטרה והקוד קריא וקל למעקב.

פירוט הרכיבים:

**DAL:** שכבת ה- DAL היא השכבה דרכה ניגשים לנתונים היושבים ב- DB שנכתב ב-SQL. שכבה זו מכילה מחלקות המייצגות את בסיס הנתונים. פעולות ההתקשרות עם בסיס הנתונים נעשו בטכנולוגית Entity Framework. טכנולוגיה זו הינה קוד פתוח שנכתב עבור ישומי NET . הנתמכים ע"י Microsoft. היא מאפשרת לעבוד עם הנתונים באמצעות אובייקטים של מחלקוח ספייציפיות בלי להתמקד בטבלאות ובעמודות המשמשות בסיס למסד הנתונים. כך אפשר לעבוד באופן אבסטרקטי ולתחזק את הישום באופן מונחה עצמים. מקובלת בתחזוקת בסיס נתונים כאשר היא מנתחת את בסיס הנתונים, בונה מחלקות לייצוג הטבלאות ומספקת רשימות מלאות בנתוני בסיס הנתונים. היא ממפה יחסי אובייקטים וכך הגישה לנתונים וכל ממשק הפתוח נוחים ונעימים.

**BLL:** בשכבה זו כתובה הלוגיקה של הפרויקט, ובה יופיע אלגוריתם הפיצול והשיום.

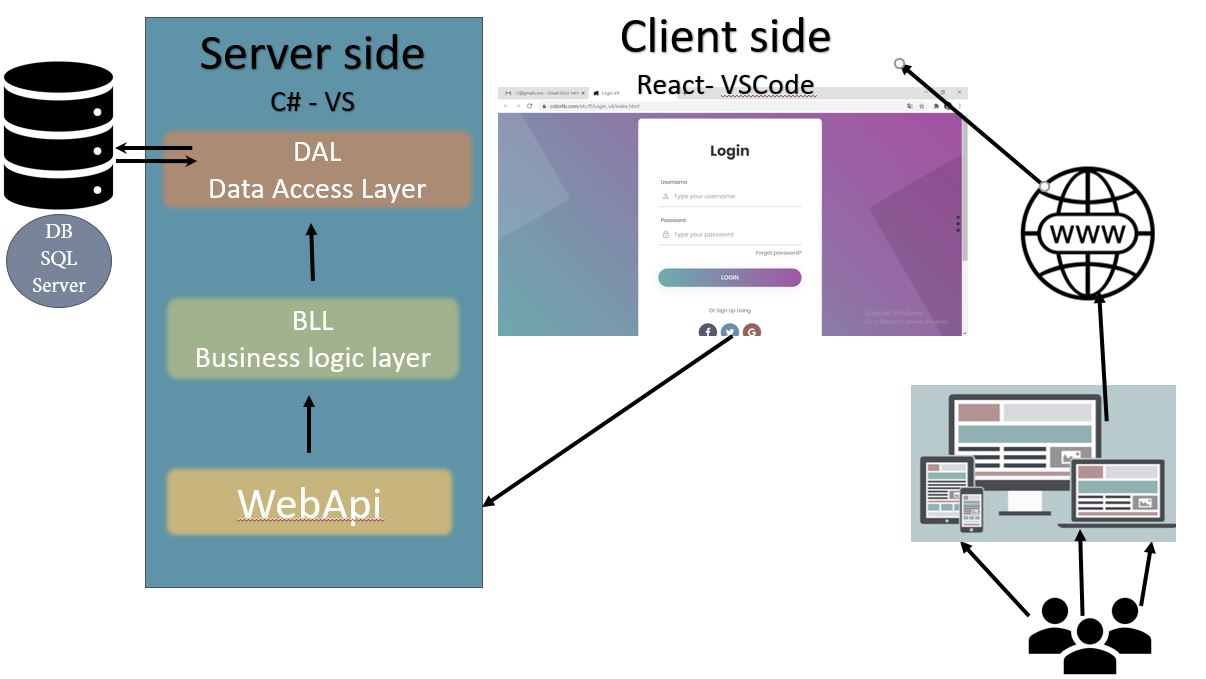
שכבת ה- BLL מכילה פונקציות המרה שנמצאות במחלקת Converts הממירות אובייקטים מטיפוס הנתונים של מסד הנתונים לטיפוס הנתונים של ה-BLL בשם DTO ולהיפך, מה שבפרויקט זה נכתב בשפת #C.

**WebApi:** מתודות השרת המחצינות את הפעולות שניתן לבצע בשרת. מתודות אלו משתמשות ב- BLL ומופעלות ע"י שכבת ה- GUI בצד הלקוח.

**GUI:** צד הלקוח נכתב ב REACT ומקבל דרך שכבת ה WEBAPI את הנתונים מהשרת.

מודל זה אמנם גורם לטרחה טכנית לא מעטה, בכתיבת הקוד ובתכנון הפונקציות אבל מספק קוד נקי, קל להבנה ונוח לשינויים ושדרוגים.

בפרויקט הושקעה עבודה רבה בכתיבה נכונה ויעילה כמקובל בעולם התכנות, דבר שדורש אמנם זמן רב למדי אך הקנה לנו ניסיון אמתי מעין כמוהו.

****תרשים 3 השכבות:

ארכיטקטורת רשת

ארכיטקטורת הרשת של האינטרנט, המשמשת את הפרויקט, מתבטאת בעיקר בשימוש בחבילת פרוטוקול האינטרנט , ולא במודל ספציפי לחיבור רשתות או צמתים ברשת או שימוש בסוגים ספציפיים של קישורי חומרה. מודל OSI מציג את הפעולות השונות הנדרשות על-מנת להעביר נתונים ברשת תקשורת. הוא מגדיר ומגדיר את המושג ארכיטקטורת רשת שכבתית. שכבות הפשטה משמשות לחלוקה נוספת של מערכת תקשורת לחלקים קטנים יותר לניהול. (שכבה היא אוסף של פונקציות דומות המספקות שירותים לשכבה שמעליה ומקבלת שירותים מהשכבה שמתחתיה. בכל שכבה, מופע מספק שירותים למופעים בשכבה שלמעלה ומבקש שירות מהשכבה למטה) :

נתונים שעוברים ברשת עוברים תהליך של עטיפה, (encapsulation) בצד השולח כל שכבה מוסיפה נתוני בקרה שונים למידע המועבר, ובצד המקבל נתונים אלו מוסרים, שלב אחר שלב. במהלך שלבי העטיפה נהוג לכנות את הנתונים בשלבים שונים. -

* בשלוש השכבות העליונות הפרוטוקולים השונים מוסיפים פתיח ו/או סוגר לנתונים, והם מכונים "רצף נתונים . (data stream) "
* בשכבת התעבורה הנתונים מחולקים למקטעים (segments) כדי שאפשר יהיה לבצע בקרה על ההעברה שלהם ברשת.
* בשכבת הרשת כל מקטע מחולק ל[חבילות](https://www.hamichlol.org.il/%D7%97%D7%91%D7%99%D7%9C%D7%AA_%D7%9E%D7%99%D7%93%D7%A2) (packets) ולכל חבילה מוצמדת הכתובת הלוגית של היעד.
* בשכבת הקו (data link) לכל חבילה מוצמדת הכתובת הפיזית של היעד, והחבילות מכונות [מסגרות](https://www.hamichlol.org.il/%D7%9E%D7%A1%D7%92%D7%A8%D7%AA_(%D7%A8%D7%A9%D7%AA)) (frames).
* השכבה הפיזית עוסקת רק בייצוג ה[בינארי](https://www.hamichlol.org.il/%D7%91%D7%A1%D7%99%D7%A1_%D7%91%D7%99%D7%A0%D7%90%D7%A8%D7%99) של המסגרות, ולכן מכונים - רצף בינארי .

תיאור פרוטוקולי התקשורת

**פרוטוקול HTTP** (( Protocol Transfer Hypertext הינו פרוטוקול תקשורת המאפשר העברת דפי HTML ברשת האינטרנט. הפרוטוקול פועל בשכבת היישום (Application) של מודל ה ,OSI השכבה השביעית. אופן העבודה של תקשורת HTTP מתחיל בהיווצרות שיחה בין השרת ללקוח באמצעות פרוטוקול TCP) בשכבת התעבורה), וממשיך בשליחה של בקשות ותשובות בין שני הצדדים (צד אחד שולח בקשה, הצד השני מחזיר תשובה). לבקשת HTTP סטנדרטית הרכיבים הבאים ● שיטת הבקשה - לפי גרסה 1.1 קיימות 8 שיטות בקשה שונות: ,GET .HEAD, POST, PUT, DELETE OPTIONS, TRACE, CONNECT .

,GET לשם הדגמה, הינה סוג הבקשה הנפוץ ביותר באינטרנט, ומיועדת לקבלת אובייקט הנמצא בשרת לפי כתובת שניתנה בתחילת ההודעה. POST - בקשה המכילה גוף הודעה, בד"כ לשליחת מחרוזות ארוכות לשרת לצורכי עיבוד. גרסת הפרוטוקול לפיו הבקשה בנויה משדות כותרת המתייחסים לבקשה, ללקוח, או לתוכן בגוף הבקשה ,request headers = גוף הבקשה (אופציונלי) , והרכב תשובת .HTTP : לאחר קבלת הבקשה מהלקוח, השרת בונה תשובה שמורכבת מגרסת הפרוטוקול לפיו התשובה בנויה, קוד מצב - מספר המציין את תוצאת הניסיון של השרת למלא את בקשת הלקוח, שדות כותרת המכילים מידע על הודעת התשובה ועל השרת - response headers. , גוף ההודעה. ו. HTTP headers להודעות (בקשות ותשובות) בפרוטוקול HTTP קיימים שדות כותרת המתווספים לאחר השורה הראשונה בהודעה. שדות אלה מייצגים הגדרות ומידע על ההודעה , ומופיעים בצורת זוג של מפתח-ערך המופרדים ע"י נקודותיים. רוב שדות הכותרת הם אופציונאליים בשליחת הודעת HTTP, אך ישנם שדות מסוימים אשר הכרחיים בהודעה, כמו השדה host עבור בקשות ( מציין את שם ה-domain של השרת והפורט בו הוא מאזין), ושדות המציינים את אורך גוף ההודעה במידה ולא ריק. רשימה מפורטת של שדות הכותרת בבקשות ותשובות HTTP ניתן למצוא בקישור הבא: https://en.wikipedia.org/wiki/List\_of\_HTTP\_header\_fields#Request\_fiel ds HTTPS: ○

בנוסף, קיים הפרוטוקול HTTPS ,שלמעשה אינו פרוטוקול כשלעצמו, אלא השימוש בפרוטוקול HTTP בצורה מאובטחת ע"י שימוש בשכבת. TLS/SSL אלו פרוטוקולי אבטחה ברשת האינטרנט שמשתמשים בשיטות קריפטוגרפיות חזקות על המידע המועבר בין השרת והלקוח. 3 העקרונות עליהם פרוטוקולים אלו אמורים לתת מענה הם: פרטיות, אימות ואותנטיות ע"י קוד מסרים.

URL הינו רצף תווים המסודרים לפי מבנה מסוים, ומשמש לייצוג מיקום דפי אינטרנט וקבצים ברשת האינטרנט. למעשה URL מהווה מקרה פרטי ל-URI) ר"ת של Identifier Resource Uniform ), המשמש כנתיב חד חד ערכי לזיהוי כל מקור מידע שהוא, שלא בהכרח דף אינטרנט, ובכך שונה מ URL ,ומהווה בפועל הרחבה שלו. המבנה הכללי של כתובת = http / URL: scheme://authority/path?query - scheme : הפרוטוקול בו נעשה שימוש (לדוגמא HTTP או .( authority :כתובת ה-IP של השרת עליו נמצא המשאב המבוקש (או כתובת מילולית שתתורגם ע"י .( DNS קיימת אפשרות להוסיף נקודתיים ואחריהם מספר פורט ה-TCP בשרת path : הנתיב של המשאב במערכת הקבצים בשרת. אם הנתיב ריק, המשמעות היא ששם הנתיב הוא כללי, כמו: .asp.default, html.index . query לצורך העברה של פרמטרים נוספים למשאב (בשיטת GET ,ב-POST לא משתמשים ב-URL לצורך העברת הנתונים) בצורה של: ,value=parameter התו '&' משמש להפרדה בין פרמטרים מרובים.

דוגמא מהפרויקט: בפרויקט יש שימוש בכתובת URL בצד הלקוח לצורך ציון מיקום הלקוח באתר.

: http://localhost:8080/transmit\_image? הפירוק של כתובת ה-URL: ה-scheme הוא http ה-authority הוא localhost:8080 ,המציין את הכתובת של המחשב הנוכחי בפורט 8080 . transmit\_image. הוא path-ה אין חלק query משום שבפרויקט יש שימוש בשיטת POST ,ולכן הנתונים לא מועברים דרך כתובת ה-URL ,אלא בגוף ההודעה בפורמט JSON .

שרת-לקוח, טכנולוגית ההתקשרות

הלקוח משתמש בדפדפן המשמש כתוכנת לקוח על מנת לשלוח בקשה לשרת HTTP שירכש בהמשך הפיתוח. שרת ה-HTTP יכול בתורו להעביר את הבקשה לתוכנת לקוח בסיס נתונים שהוא מריץ, אשר שולחת בקשה לשרת בסיס נתונים שנמצא בחוות שרתים, על מנת לאחזר את פרטי השיר הדרוש. הפרטים מוחזרים לתוכנת לקוח בסיס הנתונים, שמחזירה את הנתונים לדפדפן (תוכנת הלקוח של המשתמש), אשר לבסוף מציג את הנתונים למשתמש.

בשלב זה של הפיתוח, הלקוח נגש לנתונים מ, localhost  (כלומר "מחשב זה") זוהי כתובת ה-hostname הניתנת לממשק הרשת הפונה אל תוך הציוד עצמו (loopback device). שם זה הוא גם דומיין-רמה עליונה.

יש להבחין כי ברשתות מחשבים מודרניות, ב-[IPv4](https://www.hamichlol.org.il/IPv4) הכתובת localhost מתורגמת לבלוק כתובות 127.0.0.1 / 8, או ל [IPv6](https://www.hamichlol.org.il/IPv6) .

מבני הנתונים ששימשו את הפרויקט:

במהלך פיתוח הקוד וכתיבתו, נוצר הצורך להשתמש במבנה נתונים לאחסון התגיות.

בתהליך העלאת השיר לאתר, מתבצעת החלוקה לתגיות.

נדרשה הכרעה למבנה נתונים מתאים לאיחסון התגיות. מבנה הנתונים LIST הינו MUTABLE , ניתן להסיר, להוסיף, למיין, לשנות, לפצל, להמיר ולעדכן בקלות. הוא גמיש יותר בפני עצמו, והקוד עליו קריא בשל היכולת שלו לעבוד על כל מבנה נתונים אחר (מערכים, רשימות מקושרות, משפטים וכד'). הפונקציונאליות והסריקה גרמו לבחירה בו, למרות הכבדת זמן הריצה. שכן, תהליך פיצול התגיות מורכב ממספר פונקציות מורכבות, שטפלו בתגיות בדינאמיות ארוכה. כך הקוד קריא, מבוצע מדיויק ונח לפיתוח ושנויים עתידיים.

12? ניתוח ותרשים cases Use / UML של המערכת המוצעת 1.12

תיאור ה-UC העיקריים של המערכת 2.12

הצגת case use עבור כל הפונקציות העיקריות בפרויקט. 3.12 מבנה נתונים בהם השתמשת )רשימה, מחסנית, תור וכו'( – נמקי את הבחירה דווקא במבנה זה 4.12 הקשרים בין היחידות השונות . 5.12 עץ מודולים Use case Diagram 12.6 Use cases רשימת 12.7 UML 12.8 9.12 תרשים של כל המחלקות בפרויקט )ב studio Visual – לחיצה ימנית על הsolution >- view >- diagram class ולהעתיק בצילום מסך את כל המחלקות שנוצרו( יש להציג את המחלקות של כל שכבה בנפרד ולפרט ולהסביר דברים הדורשים הסבר. 11.12 תיאור המחלקות המוצעות - הסבר מפורט על כל מחלקה/ יחידת מחלקות בפרויקט • מה תפקידה של כל יחידה • מה הקלטים שלה • מה הפלטים שלה הבהרה - קלט יכול להיות מהמשתמש / מהרשת / מבסיס הנתונים או מכל יחידה אחרת, כנ"ל גם פלט. יש להראות את זרימת המידע בין היחידות השונות בסעיף זה אל תתמקדי בתיאור האלגוריתם אלא במחלקות ניהוליות 13 רכיבי ממשק 14 תיכון המערכת 1.14 ארכיטקטורת המערכת.

# **ממשק משתמש גרפי**

[קפיצה לניווט](https://www.hamichlol.org.il/%D7%9E%D7%9E%D7%A9%D7%A7_%D7%9E%D7%A9%D7%AA%D7%9E%D7%A9_%D7%92%D7%A8%D7%A4%D7%99#mw-head)[קפיצה לחיפוש](https://www.hamichlol.org.il/%D7%9E%D7%9E%D7%A9%D7%A7_%D7%9E%D7%A9%D7%AA%D7%9E%D7%A9_%D7%92%D7%A8%D7%A4%D7%99#p-search)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Incomplete-document-purple.svg | **יש להשלים ערך זה**: בערך זה חסר תוכן מהותי. | |
| הנכם מוזמנים להשלים את החלקים החסרים ולהסיר הודעה זו. שקלו ליצור כותרות לפרקים הדורשים השלמה, ולהעביר את התבנית אליהם. | [עריכה](https://www.hamichlol.org.il/w/index.php?title=%D7%9E%D7%9E%D7%A9%D7%A7_%D7%9E%D7%A9%D7%AA%D7%9E%D7%A9_%D7%92%D7%A8%D7%A4%D7%99&action=edit) |

**ממשק משתמש גרפי** (באנגלית: **G**raphical **U**ser **I**nterface - **GUI**או **M**an-**M**achine**I**nterface - **MMI**) הוא [ממשק משתמש](https://www.hamichlol.org.il/%D7%9E%D7%9E%D7%A9%D7%A7_%D7%9E%D7%A9%D7%AA%D7%9E%D7%A9) ל[תוכנה](https://www.hamichlol.org.il/%D7%AA%D7%95%D7%9B%D7%A0%D7%94) או ל[אתר אינטרנט](https://www.hamichlol.org.il/%D7%90%D7%AA%D7%A8_%D7%90%D7%99%D7%A0%D7%98%D7%A8%D7%A0%D7%98), המבוסס על [עיצוב גרפי](https://www.hamichlol.org.il/%D7%A2%D7%99%D7%A6%D7%95%D7%91_%D7%92%D7%A8%D7%A4%D7%99) (להבדיל מעיצוב טקסטואלי בלבד) של המסך המוצג למשתמש.

בעבר, כאשר מהירות המחשבים הייתה נמוכה, נהגו להשתמש בממשק משתמש טקסטואלי, כזה המבוסס רק על אותיות, ובנוסף להן כפתורי פעולה ב[מקלדת](https://www.hamichlol.org.il/%D7%9E%D7%A7%D7%9C%D7%93%D7%AA_%D7%9E%D7%97%D7%A9%D7%91) (ראה: [מעטפת פקודה](https://www.hamichlol.org.il/%D7%9E%D7%A2%D7%98%D7%A4%D7%AA_%D7%A4%D7%A7%D7%95%D7%93%D7%94)). מסוף [שנות ה-80](https://www.hamichlol.org.il/%D7%A9%D7%A0%D7%95%D7%AA_%D7%94-80_%D7%A9%D7%9C_%D7%94%D7%9E%D7%90%D7%94_%D7%94-20) נפוץ ממשק משתמש גרפי, שבו משלבים אלמנטים גרפיים כחלונות, [צלמיות](https://www.hamichlol.org.il/%D7%A6%D7%9C%D7%9E%D7%99%D7%AA_(%D7%A1%D7%9E%D7%9C_%D7%97%D7%96%D7%95%D7%AA%D7%99)), פקדים כדוגמת תפריטים וכפתורים ואמצעי הצבעה (WIMP). הממשקים הגרפיים מאפשרים תצוגה גמישה יותר, במיוחד בצבעים ו[גופנים](https://www.hamichlol.org.il/%D7%92%D7%95%D7%A4%D7%9F), הבהרת כוונת הממשק והשימוש בה, באמצעות סמלים וצלמיות, ואינטראקציה קלה ושימושית יותר במחשב.

ממשק משתמש גרפי העושה שימוש ב[עכבר](https://www.hamichlol.org.il/%D7%A2%D7%9B%D7%91%D7%A8_(%D7%9E%D7%97%D7%A9%D7%91)), מוטמע ברוב התוכנות שפותחו מאמצע שנות ה-90 ומוכר לכל משתמש ב[מחשב האישי](https://www.hamichlol.org.il/%D7%9E%D7%97%D7%A9%D7%91_%D7%90%D7%99%D7%A9%D7%99). הטכנולוגיה פותחה ב"[זירוקס פארק](https://www.hamichlol.org.il/%D7%96%D7%99%D7%A8%D7%95%D7%A7%D7%A1_%D7%A4%D7%90%D7%A8%D7%A7)" - מרכז המחקר של חברת זירוקס, ושולבה מחשבי ה[מקינטוש](https://www.hamichlol.org.il/%D7%9E%D7%A7%D7%99%D7%A0%D7%98%D7%95%D7%A9). תפוצתו הנרחבת של ממשק זה החלה עם מערכת ההפעלה [Windows](https://www.hamichlol.org.il/Windows) של חברת [מיקרוסופט](https://www.hamichlol.org.il/%D7%9E%D7%99%D7%A7%D7%A8%D7%95%D7%A1%D7%95%D7%A4%D7%98). מאמצי פיתוח רבים לצד סכומי כסף גדולים הושקעו מצד חברות רבות בשיפור ממשקי משתמש גרפיים, כאשר המילה האחרונה כיום היא ממשקים גרפים תלת ממדיים כדוגמת ה-[WPF](https://www.hamichlol.org.il/WPF) של חברת מיקרוסופט.

ב[מחשבי לוח](https://www.hamichlol.org.il/%D7%9E%D7%97%D7%A9%D7%91_%D7%9C%D7%95%D7%97) וב[סמארטפונים](https://www.hamichlol.org.il/%D7%A1%D7%9E%D7%90%D7%A8%D7%98%D7%A4%D7%95%D7%9F) נעשה שימוש נרחב בממשק משתמש גרפי, המתאפיין ב[מסך מגע](https://www.hamichlol.org.il/%D7%9E%D7%A1%D7%9A_%D7%9E%D7%92%D7%A2), שבו ההצבעה נעשית באמצעות אצבעות כף היד (ולא באמצעות [עכבר](https://www.hamichlol.org.il/%D7%A2%D7%9B%D7%91%D7%A8_%D7%9E%D7%97%D7%A9%D7%91)). ב[עשור השני של המאה ה-21](https://www.hamichlol.org.il/%D7%94%D7%A2%D7%A9%D7%95%D7%A8_%D7%94%D7%A9%D7%A0%D7%99_%D7%A9%D7%9C_%D7%94%D7%9E%D7%90%D7%94_%D7%94-21) נוספה גם למחשבים אופציה של מסך מגע.

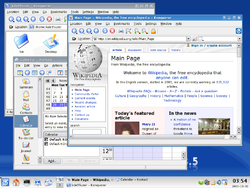
ב[הנדסת תוכנה](https://www.hamichlol.org.il/%D7%94%D7%A0%D7%93%D7%A1%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) [עקרונות הנדסה טובה](https://www.hamichlol.org.il/w/index.php?title=%D7%A2%D7%A7%D7%A8%D7%95%D7%A0%D7%95%D7%AA_%D7%94%D7%A0%D7%93%D7%A1%D7%94_%D7%98%D7%95%D7%91%D7%94&action=edit&redlink=1) קובעים שיש להפריד בין רכיבי ממשק המשתמש בתוכנה לבין שאר החלקים, ובפרט רכיבי הלוגיקה העסקית. דרך זו מאפשרת להקטין את ה[צימוד](https://www.hamichlol.org.il/%D7%A6%D7%99%D7%9E%D7%95%D7%93_(%D7%AA%D7%95%D7%9B%D7%A0%D7%94)) בין חלק התוכנה, ולפיכך להקל על עריכת שינויים בממשק המשתמש.



## תוכן עניינים

* [1 היסטוריה של ממשקי משתמש גרפיים](https://www.hamichlol.org.il/%D7%9E%D7%9E%D7%A9%D7%A7_%D7%9E%D7%A9%D7%AA%D7%9E%D7%A9_%D7%92%D7%A8%D7%A4%D7%99#.D7.94.D7.99.D7.A1.D7.98.D7.95.D7.A8.D7.99.D7.94_.D7.A9.D7.9C_.D7.9E.D7.9E.D7.A9.D7.A7.D7.99_.D7.9E.D7.A9.D7.AA.D7.9E.D7.A9_.D7.92.D7.A8.D7.A4.D7.99.D7.99.D7.9D)
* [2 התאמה אישית של ממשק המשתמש](https://www.hamichlol.org.il/%D7%9E%D7%9E%D7%A9%D7%A7_%D7%9E%D7%A9%D7%AA%D7%9E%D7%A9_%D7%92%D7%A8%D7%A4%D7%99#.D7.94.D7.AA.D7.90.D7.9E.D7.94_.D7.90.D7.99.D7.A9.D7.99.D7.AA_.D7.A9.D7.9C_.D7.9E.D7.9E.D7.A9.D7.A7_.D7.94.D7.9E.D7.A9.D7.AA.D7.9E.D7.A9)
* [3 ראו גם](https://www.hamichlol.org.il/%D7%9E%D7%9E%D7%A9%D7%A7_%D7%9E%D7%A9%D7%AA%D7%9E%D7%A9_%D7%92%D7%A8%D7%A4%D7%99#.D7.A8.D7.90.D7.95_.D7.92.D7.9D)
* [4 קישורים חיצוניים](https://www.hamichlol.org.il/%D7%9E%D7%9E%D7%A9%D7%A7_%D7%9E%D7%A9%D7%AA%D7%9E%D7%A9_%D7%92%D7%A8%D7%A4%D7%99#.D7.A7.D7.99.D7.A9.D7.95.D7.A8.D7.99.D7.9D_.D7.97.D7.99.D7.A6.D7.95.D7.A0.D7.99.D7.99.D7.9D)

## **היסטוריה של ממשקי משתמש גרפיים**

[](https://www.hamichlol.org.il/%D7%A7%D7%95%D7%91%D7%A5:Kde35.png)

GUI (מסוג [KDE](https://www.hamichlol.org.il/KDE)) במערכת ההפעלה [לינוקס](https://www.hamichlol.org.il/%D7%9C%D7%99%D7%A0%D7%95%D7%A7%D7%A1)

* אפריל [1973](https://www.hamichlol.org.il/1973) יצירת ממשק משתמש גרפי למחשב "אלטו" (Alto) במעבדות [זירוקס](https://www.hamichlol.org.il/%D7%96%D7%99%D7%A8%D7%95%D7%A7%D7%A1) (שימוש בעכבר מחשב בעל שלושה לחצנים, תצוגת מפת סיביות, תמיכה בתקשורת אתרנט)
* [1980](https://www.hamichlol.org.il/1980) תחנת עבודה גרפית "פרק" (Perq) של חברת Three Rivers Computer Corporation יוצאת לשוק.
* [1981](https://www.hamichlol.org.il/1981) זירוקס סטאר, המחשב המסחרי הראשון עם ממשק גרפי.
* [1983](https://www.hamichlol.org.il/1983) מחשב "ליסה" של אפל יוצא לשוק, עם חלונות חופפים וסרגל תפריט ראשי.
* 1983 חברת ויזי מוציאה את ויזי און (Visi On) סביבת העבודה הגרפית המוכללת הראשונה לסביבת [MS-DOS](https://www.hamichlol.org.il/MS-DOS).
* [1984](https://www.hamichlol.org.il/1984) ה[מקינטוש](https://www.hamichlol.org.il/%D7%9E%D7%A7%D7%99%D7%A0%D7%98%D7%95%D7%A9) הראשון יוצא לדרך.
* [1985](https://www.hamichlol.org.il/1985) - הגרסה הראשונה של [Windows](https://www.hamichlol.org.il/Windows), גרסה 1.0, יצאה לשוק תחת השם "Interface manager", עשתה שימוש בצלמיות וממשק משתמש גרפי. מספר חברות נוספות הציגו ממשק משתמש גרפי ל-DOS, ביניהן [IBM](https://www.hamichlol.org.il/IBM) עם [Topview](https://www.hamichlol.org.il/w/index.php?title=Topview&action=edit&redlink=1), וקוורטרדק עם [DESQview](https://www.hamichlol.org.il/w/index.php?title=DESQview&action=edit&redlink=1) (האחרונה עם ריבוי משימות וחלונות בלי גרפיקה). בנוסף ל-PC ואפל, הופיעו [אטארי](https://www.hamichlol.org.il/%D7%90%D7%98%D7%90%D7%A8%D7%99) ST עם מערכת הפעלה גרפית GEM של [דיגיטל ריסרץ'](https://www.hamichlol.org.il/w/index.php?title=%D7%93%D7%99%D7%92%D7%99%D7%98%D7%9C_%D7%A8%D7%99%D7%A1%D7%A8%D7%A5%27&action=edit&redlink=1), ו[קומודור אמיגה](https://www.hamichlol.org.il/%D7%A7%D7%95%D7%9E%D7%95%D7%93%D7%95%D7%A8_%D7%90%D7%9E%D7%99%D7%92%D7%94) עם [מערכת הפעלה](https://www.hamichlol.org.il/%D7%9E%D7%A2%D7%A8%D7%9B%D7%AA_%D7%94%D7%A4%D7%A2%D7%9C%D7%94) גרפית קניינית.

## **התאמה אישית של ממשק המשתמש**

ממשק המשתמש למוצרי [תוכנה](https://www.hamichlol.org.il/%D7%AA%D7%95%D7%9B%D7%A0%D7%94) מאפשר לעיתים התאמה אישית שלו, לצרכים והעדפות ספציפיים של המשתמש. ה[דפדפן](https://www.hamichlol.org.il/%D7%93%D7%A4%D7%93%D7%A4%D7%9F) למשל, מאפשר התאמה אישית של גודל ה[גופן](https://www.hamichlol.org.il/%D7%92%D7%95%D7%A4%D7%9F), אופן הצגת הקישורים ועוד. פרטי ההתאמה האישית נשמרים בדרך כלל תחת [פרופיל המשתמש](https://www.hamichlol.org.il/%D7%A4%D7%A8%D7%95%D7%A4%D7%99%D7%9C_%D7%9E%D7%A9%D7%AA%D7%9E%D7%A9) המתאים.

לעיתים מתאפשר תמרון בצבעים ובסימנים הגרפיים של ממשק המשתמש, ובמקרה זה קרויות הווריאציות השונות בשם "מעטפת" (ב[אנגלית](https://www.hamichlol.org.il/%D7%90%D7%A0%D7%92%D7%9C%D7%99%D7%AA): Skin, זהו ה"עור" של התוכנה, להבדיל מהפונקציונליות הנמצאת בתוכה).

דוגמה: שתי מעטפות שונות לתוכנה [VLC media player](https://www.hamichlol.org.il/VLC_media_player), המשמשת להפעלת קובצי מדיה:

# **תיכון (הנדסה)**

[קפיצה לניווט](https://www.hamichlol.org.il/%D7%AA%D7%99%D7%9B%D7%95%D7%9F_(%D7%94%D7%A0%D7%93%D7%A1%D7%94)#mw-head)[קפיצה לחיפוש](https://www.hamichlol.org.il/%D7%AA%D7%99%D7%9B%D7%95%D7%9F_(%D7%94%D7%A0%D7%93%D7%A1%D7%94)#p-search)

פירוש נוסף ערך זה עוסק בתהליך הנדסי. אם התכוונתם למושג מתחום הנדסת המישור, ראו [תיכון (גאומטריה)](https://www.hamichlol.org.il/%D7%AA%D7%99%D7%9B%D7%95%D7%9F_(%D7%92%D7%90%D7%95%D7%9E%D7%98%D7%A8%D7%99%D7%94)).

Disambig RTL.svg המונח "עיצוב הנדסי" מפנה לכאן. לערך העוסק בהסבר על עיצוב מוצר, ראו [עיצוב](https://www.hamichlol.org.il/%D7%A2%D7%99%D7%A6%D7%95%D7%91).

**תיכוּן** (**תכן**) הוא תהליך הנדסי לפיתוח מוצר. תהליך התיכון והפיתוח של [מוצר](https://www.hamichlol.org.il/%D7%9E%D7%95%D7%A6%D7%A8) כולל סדרת צעדים:

* זיהוי [צורך](https://www.hamichlol.org.il/%D7%A6%D7%95%D7%A8%D7%9A)
* קביעת המטרה והדרישות מהמוצר - שלב ה-SRR
* העלאת אפשרויות התכן של המוצר ובחירה באחד מהם - שלב ה-SDR
* תכנון המוצר, פיתוחו ובניית [אב טיפוס](https://www.hamichlol.org.il/%D7%90%D7%91_%D7%98%D7%99%D7%A4%D7%95%D7%A1) (רעיונות ליישום, בחירת החומרים שישמשו בייצורו, קביעת צורתו של המוצר וקביעת תהליכי העיבוד והייצור) - שלבי ה-PDR וה-CDR
* הערכת המוצר הקיים ביחס לדרישות שהיו ממנו בשלב התכנון. לפעמים נדרש שיפור המוצר, כלומר מתחיל סבב חדש של [מחזור חיים](https://www.hamichlol.org.il/%D7%9E%D7%97%D7%96%D7%95%D7%A8_%D7%97%D7%99%D7%99%D7%9D) של המוצר.

הצלחת תיכון נמדדת במידת שביעות רצונו של הלקוח, או ביכולת למכור את המוצר ברווח.



## תוכן עניינים

* [1 זיהוי הצורך/הבעיה](https://www.hamichlol.org.il/%D7%AA%D7%99%D7%9B%D7%95%D7%9F_(%D7%94%D7%A0%D7%93%D7%A1%D7%94)#.D7.96.D7.99.D7.94.D7.95.D7.99_.D7.94.D7.A6.D7.95.D7.A8.D7.9A.2F.D7.94.D7.91.D7.A2.D7.99.D7.94)
* [2 הגדרת דרישות המוצר](https://www.hamichlol.org.il/%D7%AA%D7%99%D7%9B%D7%95%D7%9F_(%D7%94%D7%A0%D7%93%D7%A1%D7%94)#.D7.94.D7.92.D7.93.D7.A8.D7.AA_.D7.93.D7.A8.D7.99.D7.A9.D7.95.D7.AA_.D7.94.D7.9E.D7.95.D7.A6.D7.A8)
* [3 בחירת פתרון](https://www.hamichlol.org.il/%D7%AA%D7%99%D7%9B%D7%95%D7%9F_(%D7%94%D7%A0%D7%93%D7%A1%D7%94)#.D7.91.D7.97.D7.99.D7.A8.D7.AA_.D7.A4.D7.AA.D7.A8.D7.95.D7.9F)
  + [3.1 בדיקת והערכת הפתרון](https://www.hamichlol.org.il/%D7%AA%D7%99%D7%9B%D7%95%D7%9F_(%D7%94%D7%A0%D7%93%D7%A1%D7%94)#.D7.91.D7.93.D7.99.D7.A7.D7.AA_.D7.95.D7.94.D7.A2.D7.A8.D7.9B.D7.AA_.D7.94.D7.A4.D7.AA.D7.A8.D7.95.D7.9F)
* [4 מימוש הפתרון - תכנון מפורט](https://www.hamichlol.org.il/%D7%AA%D7%99%D7%9B%D7%95%D7%9F_(%D7%94%D7%A0%D7%93%D7%A1%D7%94)#.D7.9E.D7.99.D7.9E.D7.95.D7.A9_.D7.94.D7.A4.D7.AA.D7.A8.D7.95.D7.9F_-_.D7.AA.D7.9B.D7.A0.D7.95.D7.9F_.D7.9E.D7.A4.D7.95.D7.A8.D7.98)
* [5 אבן דרך](https://www.hamichlol.org.il/%D7%AA%D7%99%D7%9B%D7%95%D7%9F_(%D7%94%D7%A0%D7%93%D7%A1%D7%94)#.D7.90.D7.91.D7.9F_.D7.93.D7.A8.D7.9A)
* [6 ראו גם](https://www.hamichlol.org.il/%D7%AA%D7%99%D7%9B%D7%95%D7%9F_(%D7%94%D7%A0%D7%93%D7%A1%D7%94)#.D7.A8.D7.90.D7.95_.D7.92.D7.9D)
* [7 קישורים חיצוניים](https://www.hamichlol.org.il/%D7%AA%D7%99%D7%9B%D7%95%D7%9F_(%D7%94%D7%A0%D7%93%D7%A1%D7%94)#.D7.A7.D7.99.D7.A9.D7.95.D7.A8.D7.99.D7.9D_.D7.97.D7.99.D7.A6.D7.95.D7.A0.D7.99.D7.99.D7.9D)

## **זיהוי הצורך/הבעיה**

תהליך התיכון מתחיל מרגע זיהוי צורך שאינו בר-פתרון באמצעים הקיימים או שפתרונו אינו יעיל באופן מספק. הצורך יכול לבוא לידי ביטוי בתהליכים שבוצעו עד כה על ידי האדם או מערכת חיצונית אחרת, על מנת לשפר ביצועים והוא גם יכול לבוא לידי ביטוי בתהליך שלא היה קיים קודם לכן.

דוגמאות לצורך:

* צורך לחצות נהר שמימיו קרים מאד
* צורך לשוחח עם אדם שאינו בטווח העין או הקול
* צורך לחסוך כוח עבודה גדול בשימוש של מערכת אחת

שלב זיהוי הצורך כולל את הבנת הבעיה, זיהוי הנתונים, המגבלות והאילוצים, ניסוח דרישות מפורטות ומדויקות מהפתרון, זיהוי אילוצים המכתיבים את דרך הפתרון לבעיה וזיהוי [פלח השוק](https://www.hamichlol.org.il/%D7%A4%D7%9C%D7%97_%D7%A9%D7%95%D7%A7) הרלוונטי למוצר, בין הדרישות ישנן דרישות **הכרחיות** עליהן חייב הפתרון לענות ואי אפשר להתפשר עליהן. לעומת זאת ישנן דרישות **רצויות** שאותן מדרגים לפי דרגת חשיבות, עליהן אפשר להתפשר במהלך בחירת הפתרון המתאים.

דוגמאות למגבלות שיש להתחשב בהן:

* מגבלות תקציב: עלות פיתוח [אב טיפוס](https://www.hamichlol.org.il/%D7%90%D7%91_%D7%98%D7%99%D7%A4%D7%95%D7%A1), עלות [ייצור](https://www.hamichlol.org.il/%D7%99%D7%99%D7%A6%D7%95%D7%A8) המוני, הוצאות ה[אחזקה](https://www.hamichlol.org.il/w/index.php?title=%D7%90%D7%97%D7%96%D7%A7%D7%94&action=edit&redlink=1) של המוצר, אורך החיים של המוצר.
* מגבלות זמן: [זמן](https://www.hamichlol.org.il/%D7%96%D7%9E%D7%9F) הנדרש להשלמת תהליך הפתרון לעומת הזמן המצוי.
* מגבלות של [גודל](https://www.hamichlol.org.il/%D7%92%D7%95%D7%93%D7%9C), [משקל](https://www.hamichlol.org.il/%D7%9E%D7%A9%D7%A7%D7%9C_(%D7%A4%D7%99%D7%96%D7%99%D7%A7%D7%94)) או צורה.
* [ארגונומיה](https://www.hamichlol.org.il/%D7%90%D7%A8%D7%92%D7%95%D7%A0%D7%95%D7%9E%D7%99%D7%94) : קשר בין צורתו של מוצר, גודלו ומשקלו, לבין נוחיות השימוש בו, בטיחותו והשפעתו על בריאות המשתמש לטווח הארוך.
* מגבלות [אסתטיקה](https://www.hamichlol.org.il/%D7%90%D7%A1%D7%AA%D7%98%D7%99%D7%A7%D7%94).
* מגבלות הנוגעות לזמינות חומרים נדרשים.
* מגבלות הנובעות מטכניקות של הרכבה ועיבוד חומרים.
* מגבלות של תפעול או של מימוש הפתרון: נצילות, אמינות, בטיחות – בתהליך הייצור של המוצר ובשימוש בו, מגבלות אקולוגיות – זיהום סביבה.
* אילוצים סוציאליים ופוליטיים: התחשבות בקבוצות מיעוט, דרישה לרכישת מוצר כחול-לבן וכדומה.

## **הגדרת דרישות המוצר**

בתהליך התיכון נהוג להבחין בין אילוצים לדרישות המוצר. אילוץ הוא מגבלה אשר אינה מאפשרת פתרון אחר (לדוגמה: מידות מוצר שתפקידו לנקות צינור בקוטר 1 מטר). דרישה מתחלקת בין דרישה הכרחית - עליה חייב הפתרון לענות (לדוגמה: ניקוי הצינור) לבין דרישה רצויה (לדוגמה: עלות המערכת) עליה אפשר להתפשר בהתאם לדרגת החשיבות.

תוצר שלב זה הוא מפרט, הכולל תיאור מפורט של הדרישות מהפתרון והתוצאות הרצויות כולל המגבלות והאילוצים הכרוכים בפתרון הבעיה.

תוצרי שלב זה מוצגים לבעלי העניין בפרויקט (פורום הפרויקט) ומקובעים להמשך הפרויקט בסקר תיכון הנקרא בשפה המקצועית **סקר דרישות מערכת (SRR)**. בהשאלה, כל הפעילויות לקראת ה-SRR נקראות שלב ה-SRR.

## **בחירת פתרון**

בשלב זה מתבצע איסוף ה[מידע](https://www.hamichlol.org.il/%D7%9E%D7%99%D7%93%D7%A2) מהיבטים שונים היכולים לסייע לפתרון הבעיה. המידע נאסף ממקורות שונים: [ספרים](https://www.hamichlol.org.il/%D7%A1%D7%A4%D7%A8), [אנציקלופדיות](https://www.hamichlol.org.il/%D7%90%D7%A0%D7%A6%D7%99%D7%A7%D7%9C%D7%95%D7%A4%D7%93%D7%99%D7%94), [אנשים](https://www.hamichlol.org.il/%D7%90%D7%93%D7%9D) באמצעות [סקרים](https://www.hamichlol.org.il/%D7%A1%D7%A7%D7%A8), [ראיונות](https://www.hamichlol.org.il/%D7%A8%D7%90%D7%99%D7%95%D7%9F_%D7%A2%D7%95%D7%9E%D7%A7), מומחים, [חנויות](https://www.hamichlol.org.il/%D7%97%D7%A0%D7%95%D7%AA), [סרטים](https://www.hamichlol.org.il/%D7%A1%D7%A8%D7%98_%D7%A7%D7%95%D7%9C%D7%A0%D7%95%D7%A2), [עיתונות](https://www.hamichlol.org.il/%D7%A2%D7%99%D7%AA%D7%95%D7%9F), מאגרי מידע, רשת ה[אינטרנט](https://www.hamichlol.org.il/%D7%90%D7%99%D7%A0%D7%98%D7%A8%D7%A0%D7%98) ועוד.

לאחר איסוף המידע, נהוג להעלות את כל הרעיונות האפשריים לפתרון הבעיה, בהתחשב באילוצים ובדרישות שהועלו. שיטה הנהוגה בשלב זה היא שיטת [סיעור מוחות](https://www.hamichlol.org.il/%D7%A1%D7%99%D7%A2%D7%95%D7%A8_%D7%9E%D7%95%D7%97%D7%95%D7%AA), במהלכו משתפים אחד את השני ברעיונות ומרכזים אותם על דף אחד. שיטה נוספת הנהוגה בשלב הזה היא תיאור כל הפתרונות באמצעות טבלה מורפולוגית, אשר מכילה את כל הצירופים האפשריים לפתרון, בהתחשב באילוצים ובדרישות שהוגדרו קודם לכן. פתרון נוצר על ידי חיבור אמצעים שונים בכל הפונקציות. דירוג הפתרונות מגלה את הפתרון הטוב ביותר.

בסופו של השלב, בוחרים את הרעיון המתאים ביותר.

תוצרי שלב זה מוצגים לבעלי העניין בפרויקט ומקובעים להמשך הפרויקט בסקר תיכון הנקרא בשפה המקצועית **סקר תיכון מערכת (SDR)**. בהשאלה, כל הפעילויות לקראת ה-SDR נקראות שלב ה-SDR.

### בדיקת והערכת הפתרון

האם הוא נוח, יעיל, עונה על דרישות המפרט, עומד במגבלות ובאילוצים הכרוכים בפתרון הבעיה, עומד בתקציב הייצור ובתקציב התפעול, מהי השפעתו על הסביבה, מהו אורך חייו ומהי התדירות בה יש להחליף חלקים.

## **מימוש הפתרון - תכנון מפורט**

* הכנת תיק שרטוטים: בשלב זה מכינים תיק שרטוטים מפורט של הדגם על כל חלקיו, בחירת החומרים שמהם יהיה עשוי הדגם, הכלים שבעזרתם יבוצע הדגם והמנגנונים בהם יעשה שימוש. תיק השרטוטים, בעידן המודרני, מתבצע בעיקר באמצעות תוכנות [CAD](https://www.hamichlol.org.il/CAD) שונות המגדירות את כל הפרמטרים למוצר.

תוצרי שלב זה מוצגים לבעלי העניין בפרויקט ומקובעים להמשך הפרויקט בסקר תיכון הנקרא בשפה המקצועית **סקר תיכון ראשוני (PDR)**. בהשאלה, כל הפעילויות לקראת ה-PDR נקראות שלב ה-PDR.

* ייצור דגם המוצר: במידה וכל השלבים עד כה עברו בהצלחה, יש לסיים את תהליך התיכון ולגשת לייצור המוצר, לרוב, יפותח [אב טיפוס](https://www.hamichlol.org.il/%D7%90%D7%91_%D7%98%D7%99%D7%A4%D7%95%D7%A1) המשמש כ[דגם](https://www.hamichlol.org.il/%D7%93%D7%92%D7%9D) של המוצר קודם לייצורו.

תוצרי שלב זה מוצגים לבעלי העניין בפרויקט ומקובעים להמשך הפרויקט בסקר תיכון הנקרא בשפה המקצועית **סקר תיכון קריטי (CDR)**. בהשאלה, כל הפעילויות לקראת ה-CDR נקראות שלב ה-CDR.

## **אבן דרך**

בשלב זה מתבצעת הערכה מחודשת של כל הפרויקט על כל הבטיו. נבדקת היעילות המעשית של אב הטיפוס לצורך, בהתאם לדרישות והאילוצים. במידה והמוצר עובר את תהליך זה הוא יעבור לשלב היצור הסידרתי והשיווק. השלב הבא (שלב היצור הסידרתי והשיווק) הוא שלב שנמצא כבר מחוץ לשלב התיכון ומגדיר, למעשה, את סיומו של התהליך.

# **רכיטקטורת תוכנה**

[קפיצה לניווט](https://www.hamichlol.org.il/%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94#mw-head)[קפיצה לחיפוש](https://www.hamichlol.org.il/%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94#p-search)

|  |
| --- |
| [**הנדסת תוכנה**](https://www.hamichlol.org.il/%D7%94%D7%A0%D7%93%D7%A1%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) |
| ערך זה הוא חלק מקטגוריית [הנדסת תוכנה](https://www.hamichlol.org.il/%D7%A7%D7%98%D7%92%D7%95%D7%A8%D7%99%D7%94:%D7%94%D7%A0%D7%93%D7%A1%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94)  [Coding Shots Annual Plan high res-5.jpg](https://www.hamichlol.org.il/%D7%A7%D7%95%D7%91%D7%A5:Coding_Shots_Annual_Plan_high_res-5.jpg) [מתכנת](https://www.hamichlol.org.il/%D7%9E%D7%AA%D7%9B%D7%A0%D7%AA) בעבודתו |
| **פעילויות ושלבים** |
| [דרישות](https://www.hamichlol.org.il/%D7%93%D7%A8%D7%99%D7%A9%D7%94_(%D7%94%D7%A0%D7%93%D7%A1%D7%94)) • [ניתוח](https://www.hamichlol.org.il/%D7%A0%D7%99%D7%AA%D7%95%D7%97_%D7%9E%D7%A2%D7%A8%D7%9B%D7%95%D7%AA_%D7%9E%D7%99%D7%93%D7%A2) • [אפיון](https://www.hamichlol.org.il/%D7%90%D7%A4%D7%99%D7%95%D7%9F_%D7%9E%D7%A2%D7%A8%D7%9B%D7%AA_%D7%9E%D7%99%D7%93%D7%A2) • ארכיטקטורה • [עיצוב](https://www.hamichlol.org.il/%D7%A2%D7%99%D7%A6%D7%95%D7%91_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) • [תכנות](https://www.hamichlol.org.il/%D7%AA%D7%9B%D7%A0%D7%95%D7%AA_%D7%9E%D7%97%D7%A9%D7%91%D7%99%D7%9D) • [דיבוג](https://www.hamichlol.org.il/%D7%A0%D7%99%D7%A4%D7%95%D7%99_%D7%A9%D7%92%D7%99%D7%90%D7%95%D7%AA) • [בדיקה](https://www.hamichlol.org.il/%D7%91%D7%93%D7%99%D7%A7%D7%95%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) • [אימות](https://www.hamichlol.org.il/%D7%90%D7%99%D7%9E%D7%95%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) • [בנייה](https://www.hamichlol.org.il/%D7%91%D7%A0%D7%99%D7%99%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) • [פריסה](https://www.hamichlol.org.il/%D7%A4%D7%A8%D7%99%D7%A1%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) • [תפעול](https://www.hamichlol.org.il/w/index.php?title=%D7%AA%D7%A4%D7%A2%D7%95%D7%9C_%D7%AA%D7%95%D7%9B%D7%A0%D7%94&action=edit&redlink=1) • [תחזוקה](https://www.hamichlol.org.il/%D7%AA%D7%97%D7%96%D7%95%D7%A7%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) |
| [**מתודולוגיות**](https://www.hamichlol.org.il/%D7%9E%D7%AA%D7%95%D7%93%D7%95%D7%9C%D7%95%D7%92%D7%99%D7%99%D7%AA_%D7%A4%D7%99%D7%AA%D7%95%D7%97_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) |
| [זריזות](https://www.hamichlol.org.il/%D7%A4%D7%99%D7%AA%D7%95%D7%97_%D7%AA%D7%95%D7%9B%D7%A0%D7%94_%D7%96%D7%A8%D7%99%D7%96) • [מפל המים](https://www.hamichlol.org.il/%D7%9E%D7%95%D7%93%D7%9C_%D7%9E%D7%A4%D7%9C_%D7%94%D7%9E%D7%99%D7%9D) • [תכנת ותקן](https://www.hamichlol.org.il/%D7%9E%D7%AA%D7%95%D7%93%D7%95%D7%9C%D7%95%D7%92%D7%99%D7%99%D7%AA_%D7%AA%D7%9B%D7%A0%D7%AA_%D7%95%D7%AA%D7%A7%D7%9F) • [Crystal Clear](https://www.hamichlol.org.il/w/index.php?title=Crystal_Clear&action=edit&redlink=1) • [Scrum](https://www.hamichlol.org.il/Scrum) • [Unified Process](https://www.hamichlol.org.il/Unified_Process) • [Extreme Programming](https://www.hamichlol.org.il/Extreme_Programming) • [אינטגרציה רציפה](https://www.hamichlol.org.il/%D7%90%D7%99%D7%A0%D7%98%D7%92%D7%A8%D7%A6%D7%99%D7%94_%D7%A8%D7%A6%D7%99%D7%A4%D7%94) • [DevOps](https://www.hamichlol.org.il/DevOps) |
| **תחומים תומכים** |
| [ניהול פרויקטים](https://www.hamichlol.org.il/%D7%A0%D7%99%D7%94%D7%95%D7%9C_%D7%A4%D7%A8%D7%95%D7%99%D7%A7%D7%98%D7%99%D7%9D) • [ניהול תצורה](https://www.hamichlol.org.il/%D7%A0%D7%99%D7%94%D7%95%D7%9C_%D7%AA%D7%A6%D7%95%D7%A8%D7%94) • [תיעוד](https://www.hamichlol.org.il/%D7%AA%D7%99%D7%A2%D7%95%D7%93_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) • [הבטחת איכות](https://www.hamichlol.org.il/%D7%94%D7%91%D7%98%D7%97%D7%AA_%D7%90%D7%99%D7%9B%D7%95%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) • [Profiling](https://www.hamichlol.org.il/w/index.php?title=Profiling&action=edit&redlink=1) |
| **כלים** |
| [מהדר](https://www.hamichlol.org.il/%D7%9E%D7%94%D7%93%D7%A8) • [מקשר](https://www.hamichlol.org.il/%D7%9E%D7%A7%D7%A9%D7%A8_(%D7%AA%D7%95%D7%9B%D7%A0%D7%94)) • [מפרש](https://www.hamichlol.org.il/%D7%9E%D7%A4%D7%A8%D7%A9_(%D7%AA%D7%95%D7%9B%D7%A0%D7%94)) • [IDE](https://www.hamichlol.org.il/%D7%A1%D7%91%D7%99%D7%91%D7%AA_%D7%A4%D7%99%D7%AA%D7%95%D7%97_%D7%9E%D7%A9%D7%95%D7%9C%D7%91%D7%AA) • [ניהול גרסאות](https://www.hamichlol.org.il/%D7%A0%D7%99%D7%94%D7%95%D7%9C_%D7%92%D7%A8%D7%A1%D7%90%D7%95%D7%AA) • [אוטומציית בנייה](https://www.hamichlol.org.il/%D7%90%D7%95%D7%98%D7%95%D7%9E%D7%A6%D7%99%D7%94_%D7%A9%D7%9C_%D7%91%D7%A0%D7%99%D7%99%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) |

**ארכיטקטורה** היא התחום העוסק בתכנון מערכות [תוכנה](https://www.hamichlol.org.il/%D7%AA%D7%95%D7%9B%D7%A0%D7%94). המונח ארכיטקטורה בהנדסת תוכנה פירושו ייצוג היבטים שונים של התוכנה באופן [מופשט](https://www.hamichlol.org.il/%D7%90%D7%91%D7%A1%D7%98%D7%A8%D7%A7%D7%A6%D7%99%D7%94_(%D7%9E%D7%93%D7%A2%D7%99_%D7%94%D7%9E%D7%97%D7%A9%D7%91)). ארכיטקטורה של מערכות תוכנה היא לפיכך תכנון מופשט של ההיבטים השונים של התוכנה, היחסים בין המרכיבים השונים של התוכנה והחוקים החלים עליהם.



## תוכן עניינים

* [1 חשיבותה של ארכיטקטורה](https://www.hamichlol.org.il/%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94#.D7.97.D7.A9.D7.99.D7.91.D7.95.D7.AA.D7.94_.D7.A9.D7.9C_.D7.90.D7.A8.D7.9B.D7.99.D7.98.D7.A7.D7.98.D7.95.D7.A8.D7.94)
* [2 דיסצפלינות הארכיטקטורה](https://www.hamichlol.org.il/%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94#.D7.93.D7.99.D7.A1.D7.A6.D7.A4.D7.9C.D7.99.D7.A0.D7.95.D7.AA_.D7.94.D7.90.D7.A8.D7.9B.D7.99.D7.98.D7.A7.D7.98.D7.95.D7.A8.D7.94)
* [3 סגנונות בארכיטקטורה](https://www.hamichlol.org.il/%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94#.D7.A1.D7.92.D7.A0.D7.95.D7.A0.D7.95.D7.AA_.D7.91.D7.90.D7.A8.D7.9B.D7.99.D7.98.D7.A7.D7.98.D7.95.D7.A8.D7.94)
* [4 הסמכה](https://www.hamichlol.org.il/%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94#.D7.94.D7.A1.D7.9E.D7.9B.D7.94)
* [5 ראו גם](https://www.hamichlol.org.il/%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94#.D7.A8.D7.90.D7.95_.D7.92.D7.9D)
* [6 קישורים חיצוניים](https://www.hamichlol.org.il/%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%AA_%D7%AA%D7%95%D7%9B%D7%A0%D7%94#.D7.A7.D7.99.D7.A9.D7.95.D7.A8.D7.99.D7.9D_.D7.97.D7.99.D7.A6.D7.95.D7.A0.D7.99.D7.99.D7.9D)

## **חשיבותה של ארכיטקטורה**

מחקרים ראשונים בתחום זה נעשו כבר בשנות ה-60 של המאה ה-20, אבל חשיבותו עלתה מאד החל משנות ה-90 בשל הגודל והמורכבות של מערכות התוכנה (ראו גם [משבר התוכנה](https://www.hamichlol.org.il/%D7%9E%D7%A9%D7%91%D7%A8_%D7%94%D7%AA%D7%95%D7%9B%D7%A0%D7%94)). ארכיטקטורה של מערכת תוכנה עוסקת לעיתים גם בהגדרת מבנה התוכנה בקנה מידה בינוני וקטן, אך לרוב מתמקדת בהגדרת המבנים העיקריים, הממשקים החיצוניים והתהליכים העיקריים של התוכנה. עדיין אין הסכמה בתעשייה באשר להיבטים השונים של התוכנה הנדרשים להכלל כחלק מהארכיטקטורה, אם כי יש דרך תקנית לתיאור חלק מההיבטים באמצעות שפת המידול המאוחדת [UML](https://www.hamichlol.org.il/UML).

## **דיסצפלינות הארכיטקטורה**

* **ארכיטקטורת יישומים** היא הדיסצפלינה העוסקת במבנה הפנימי של התוכנה, תתי-המערכות והרכיבים המרכיבים אותה, היחסים השונים ביניהם והממשקים החיצוניים של התוכנה.
* **ארכיטקטורת**[**אינטגרציה**](https://www.hamichlol.org.il/%D7%A9%D7%99%D7%9C%D7%95%D7%91_%D7%9E%D7%A2%D7%A8%D7%9B%D7%95%D7%AA) היא הדסצפלינה העוסקת בחיבור וקישוריות של מערכות תוכנה, הפעלה-הידודית, הגדרה ותיחום שירותים וניהול תהליכים עסקיים.
* **ארכיטקטורה ארגונית**
* **ארכיטקטורת תקשורת**
* **ארכיטקטורת אבטחת מידע**
* **ארכיטקטורה תפעולית**

## **סגנונות בארכיטקטורה**

* [ארכיטקטורה מוכוונת שירותים](https://www.hamichlol.org.il/%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%94_%D7%9E%D7%95%D7%9B%D7%95%D7%95%D7%A0%D7%AA_%D7%A9%D7%99%D7%A8%D7%95%D7%AA%D7%99%D7%9D) (SOA).
* [ארכיטקטורה רב-שכבתית](https://www.hamichlol.org.il/%D7%90%D7%A8%D7%9B%D7%99%D7%98%D7%A7%D7%98%D7%95%D7%A8%D7%94_%D7%A8%D7%91-%D7%A9%D7%9B%D7%91%D7%AA%D7%99%D7%AA_(%D7%AA%D7%95%D7%9B%D7%A0%D7%94))

תיכון מפורט: טכנולוגית הפתרון, ולפני כן 3 שורות אדומות. להתמקד במחלקות ניהוליות ולא בתיאור האלגוריתם.

חלופות לתיכון המערכת: ? טכנולוגיות אחרות שהיו יכולות לבא בחשבון.

תיאור התוכנה:

מרבית הקוד בצד השרת, כאמור, נכתב בשפת #C בסביבת העבודה של MICROSOFT הנוחה והמפותחת, VISUAL STUDIO בטכנולוגיית WebApi .

קצת על השפה:

שפת C #היא שפת תכנות עילית, אשר התחילה את הפיתוח שלה בשנות ה-2000 על ידי חברת Microsoft כחלק מפרויקט ה NET שלה. שפה זו מתבססת על שפות התכנות Java++ , C ,ו- Visual Basic ודומה מאוד מבחינה תחבירית ל-++C ו .Java שפת התכנות C #היא שפת תכנות מונחית עצמים, כאשר המונח עצמים מייצג למעשב מבנה תכונתי המוכר יותר כמחלקה. כל מחלקה עשויה מטיפוס מסוים, מוחשי או מופשט, כאשר כל מחלקה יכולה להיות דבר המכיל נתונים מאותו סוג. בשפה זו כל המשתנים הם עצמים כולל גם הטיפוסים הנחשבים לפשוטים. כמו יתר שפות התכנות מונחות העצמים, גם C #כוללת שימוש בעקרונות כמו ירושה, כמוס ורב צורתיות. עקרון חשוב מאוד של תכנות בשפת C #הוא השימוש בממשקים מרובים. בדומה לשפות ותיקות אחרות תהליך ההורשה ב-C #יכול להתבצע טיפוס בסיסי אחד, אולם C #מעניקה למתכנת יכולת לרשת ממספר רב של ממשקים שמהווים יתרון גדול מאוד לפרויקטים בתחום התוכנה שעובדים בצוותים כאשר על צוות עובד על מאפיין או יכולת אחרת של התוכנית. כחלק מטכנולוגיית ה .NET גם C #מוגדרת כשפה בעלת קוד מנוהל. זה אומר שהשפה עושה הרבה עבודה חשובה מאחורי הקלעים ובכך משחררת את המתכנת מהאילוץ להתמודד עם נושאים כמו פינוי זיכרון או זליגה של זיכרון. כל הדברים הללו הופכים את שפת התכנות C #לשפה פופולרית, נוחה וידידות למשתמש.

הDB נכתב בסקריפטים בשפת SQL SERVER ; והמידע נאגר שם.

קצת על מסד הנתונים:

**Microsoft SQL Server** : (ההגייה המקובלת ב[ישראל](https://www.hamichlol.org.il/%D7%99%D7%A9%D7%A8%D7%90%D7%9C) היא אֶס-קְיוּ-אֶל סֶרְוֶר וב[ארצות הברית](https://www.hamichlol.org.il/%D7%90%D7%A8%D7%A6%D7%95%D7%AA_%D7%94%D7%91%D7%A8%D7%99%D7%AA) סִיקְוֶול סֶרְוֶר) היא [מערכת לניהול בסיס נתונים](https://www.hamichlol.org.il/w/index.php?title=%D7%9E%D7%A2%D7%A8%D7%9B%D7%AA_%D7%9C%D7%A0%D7%99%D7%94%D7%95%D7%9C_%D7%91%D7%A1%D7%99%D7%A1_%D7%A0%D7%AA%D7%95%D7%A0%D7%99%D7%9D&action=edit&redlink=1) במודל ה[יחסי](https://www.hamichlol.org.il/%D7%91%D7%A1%D7%99%D7%A1_%D7%A0%D7%AA%D7%95%D7%A0%D7%99%D7%9D_%D7%99%D7%97%D7%A1%D7%99) (RDBMS) של חברת [מיקרוסופט](https://www.hamichlol.org.il/%D7%9E%D7%99%D7%A7%D7%A8%D7%95%D7%A1%D7%95%D7%A4%D7%98).

שפת הפיתוח שבאמצעותה מועברות הפקודות למערכת (על ידי כתיבת פקודות או שימוש בממשק גרפי ששולח פקודות באופן סמוי) היא [Transact-SQL](https://www.hamichlol.org.il/Transact-SQL) ‏(TSQL) שהיא מימוש תקן ANSI . שפת  [SQL](https://www.hamichlol.org.il/SQL) משמשת לתשאול וטיפול בנתונים (DML), יצירת טבלאות והיחסים ביניהן (DDL) ותחזוקת המערכת תוך שימוש בתוכניות שירות שונות.

המתחרות העיקריות של SQL Server בשוק בסיסי הנתונים הרלציוניים הן תוכנות כדוגמת [Oracle](https://www.hamichlol.org.il/%D7%90%D7%95%D7%A8%D7%A7%D7%9C_(%D7%91%D7%A1%D7%99%D7%A1_%D7%A0%D7%AA%D7%95%D7%A0%D7%99%D7%9D)), [MySQL](https://www.hamichlol.org.il/MySQL) ועוד; SQL Server ניתנת להתקנה רק במערכות הפעלה ממשפחת [Windows](https://www.hamichlol.org.il/Windows) של מיקרוסופט.

מבסיס הנתונים, המידע מועבר ל#C ע"י טכנולוגית EF.

הGUI נכתב ב- React הכולל html, css, Typescript תוך שימוש בספריות Material ו- BootStrap ונכתב בסביבת עבודה VISUALCODE.

קצת על הספריה:

React , (לעיתים נכתבת כ-React.js או ( ReactJS הינה [ספריית](https://www.hamichlol.org.il/%D7%A1%D7%A4%D7%A8%D7%99%D7%99%D7%94_(%D7%AA%D7%9B%D7%A0%D7%95%D7%AA)) [קוד פתוח](https://www.hamichlol.org.il/%D7%A7%D7%95%D7%93_%D7%A4%D7%AA%D7%95%D7%97) בשפת [ג'אווה סקריפט](https://www.hamichlol.org.il/%D7%92%27%D7%90%D7%95%D7%95%D7%94_%D7%A1%D7%A7%D7%A8%D7%99%D7%A4%D7%98) המשמשת לפיתוח [ממשק משתמש](https://www.hamichlol.org.il/%D7%9E%D7%9E%D7%A9%D7%A7_%D7%9E%D7%A9%D7%AA%D7%9E%D7%A9).

React פותחה על ידי ג'ורדן וולקה, [מפתח תוכנה](https://www.hamichlol.org.il/%D7%9E%D7%A4%D7%AA%D7%97_%D7%AA%D7%95%D7%9B%D7%A0%D7%94) בחברת [פייסבוק](https://www.hamichlol.org.il/%D7%A4%D7%99%D7%99%D7%A1%D7%91%D7%95%D7%A7) שהושפע מ-[AngularJS](https://www.hamichlol.org.il/AngularJS) ו-XHP, שהייתה ספריית  [PHP](https://www.hamichlol.org.il/PHP) לניהול רכיבי [HTML](https://www.hamichlol.org.il/HTML). הספרייה שימשה לראשונה בדף הבית של פייסבוק בשנת [2011](https://www.hamichlol.org.il/2011)  ובהמשך ב[אינסטגרם](https://www.hamichlol.org.il/%D7%90%D7%99%D7%A0%D7%A1%D7%98%D7%92%D7%A8%D7%9D) ב-[2012](https://www.hamichlol.org.il/2012). הספרייה שוחררה בקוד פתוח ב[מאי](https://www.hamichlol.org.il/%D7%9E%D7%90%D7%99) [2013](https://www.hamichlol.org.il/2013). בתחילת [2015](https://www.hamichlol.org.il/2015)  פרסמה פייסבוק ספרייה נוספת בשם React Native  המאפשרת פיתוח ממשקי משתמש ל[אנדרואיד](https://www.hamichlol.org.il/%D7%90%D7%A0%D7%93%D7%A8%D7%95%D7%90%D7%99%D7%93_(%D7%9E%D7%A2%D7%A8%D7%9B%D7%AA_%D7%94%D7%A4%D7%A2%D7%9C%D7%94)) ול-[Microsoft Windows](https://www.hamichlol.org.il/Microsoft_Windows) (UWP) .

אחת התכונות המרכזיות של React היא השימוש ב-Virtual Document Object Model  או Virtual DOM. : הספרייה יוצרת מבנה נתונים בזיכרון, מחשבת את השינויים הנדרשים ובכך מעדכנת את ה-[DOM](https://www.hamichlol.org.il/DOM) ביעילות.

רכיבי React נכתבים בשפת  ,JSX הרחבה של [ג'אווה סקריפט](https://www.hamichlol.org.il/%D7%92%27%D7%90%D7%95%D7%95%D7%94_%D7%A1%D7%A7%D7%A8%D7%99%D7%A4%D7%98) המאפשרת תגיות [HTML](https://www.hamichlol.org.il/HTML)  בתוך הקוד.  תגיות ה-HTML מתרגמות לקריאות לפונקציות ג'אווה סקריפט של React ומעדכנות את ה-Virtual DOM לפי הצורך.

קצת על שפת HTML:

שפת HTML ראשי תיבות של Language Markup Text Hyper ,היא שפת תגיות המיועדת עבור עיצוב דפי אינטרנט והצגת תוכן בדפדפן. השפה נחשבת לשפת סימון, דהיינו שפת מחשב אשר משמשת לתיאור מידע נלווה לטקסט, תמונות או מידע חזותי אחר. דוגמא לכך היא הגדרת צבע, גופן או גודל לטקסט. בין היתר השפה מאפשרת הצגת נתונים שונים בדפדפן כגון טקסט, צליל, תמונה, וידאו ועוד. היא משמשת תקן ברשת האינטרנט וכלליה מוגדרים ע"י ארגון התקינה הבינלאומי שנקרא W3C . במהלך השנים יצאו גרסאות שונות לשפה, כאשר העדכנית ביותר היא 5HTML ,

תחביר התגיות: ניתן לתאר בצורה כללית את התגיות שנמצאות בשימוש בשפה ע"י: content כל תגית מתחילה עם סוגריים משולשים <>, כוללת שם ולפעמים תכונות נוספות, מכילה באמצע את התוכן (אם קיים) ומסתיימת ב- </br.< HTML: כל מסמך HTML בנוי לפי מבנה היררכי של עץ, מה שנותן משמעות לקינון של התגיות.

הפורמט הנבחר לשימוש בפרויקט הוא פורמט .JSON מבנה אובייקט ה-JSON שנשלח מהלקוח לשרת (במצב של שידור) JSON : ר"ת של AJAX And JavaScript Asynchronous . המונח AJAX אינו מתייחס לטכנולוגיה העומדת בפני עצמה, אלא לשימוש במספר טכנולוגיות מסוימות יחדיו. הטכנולוגיות ש-AJAX עושה בהן שימוש: HTML ו-CSS לסימון ועיצוב מידע. שפת JavaScript - לצורך מימוש הלוגיקה של היישום, תקשורת עם השרת ולאינטראקציה עם המשתמש ע"י שימוש ב-DOM.JavaScript . הפורמט המשומש לצורך העברת המידע בין השרת ללקוח הינו באמצעות XML או, JSON אך פורמטים אחרים כגון HTML או CSV גם יעבדו . בתחילת הספר הרחבתי על טכניקה זו וכיצד מתבצעת התקשורת בין השרת ללקוח תוך שימוש ב-API של XMLHttpRequest . .

פורמט הJSON קריא לאדם המיועד להעברת מידע שמורכב מזוגות של מפתח-ערך (מופרדים ע"י התו ':'), לרוב בין שרת וצרכן. למרות שבמקור הפורמט פותח לעבודה ב-:JavaScript הוא לא תלוי שפה, ושפות תכנות רבות תומכות בהמרה לפורמט זה. קבצי JSON הם בעלי הסיומת " json. ". טיפוסי הנתונים הבסיסיים הקיימים בפורמט JSON הם: ● ערך מספרי ● מחרוזת ● ערך בוליאני - ע"י ערכי אמת/שקר (false/true) . ● מערך - רשימה ממוינת של ערכים התחומה ע"י סוגריים מרובעים: [ ]. 21 ● אובייקט ',' . ● ערך null \* כל אובייקט JSON תחום ע"י סוגריים מסולסלים.

# תיאור המסכים:

המסכים הם:

* דף ה"לוגין".
* הצטרפות לקוח חדש.
* עדכון לקוח קים.
* המסך הראשי, בו מופיעה טבלת השירים, תיבות האיתור , התגיות, ואפשרות העלאת שירים ומחיקתם. בנוסף, מסך זה מכיל מספר תתי מסכים:
  + אודות
  + הוראות שימוש.
  + פרוסמות.

היררכית המסכים והמעברים ביניהם:

? תרשים זרימה המתאר את תהליך העברת המידע ממסך למסך.

צילום של דף הלוגין, שאיבת הפרטים מהד.ב. של השירים, ואז מילוי הטבלה בדף הבא,

עדכון לקוח, לוקח את הפרטים ומדביק בדף של עדכון פרטי לקוח. אחר כך שנית חוזר.

המתג מחליף בין המסכים.

עלי ידי פונציות או רק על ידי צלומי מסך, מה בדיוק?

פירוט המסכים:

? צילומי מסך של 4 המסכים+ 3 תתי המסכים. פירוט קצר. סימון בעיגול אדום ומספר את האלמנטים כמו תיבת טקסט והסבר התפקיד.

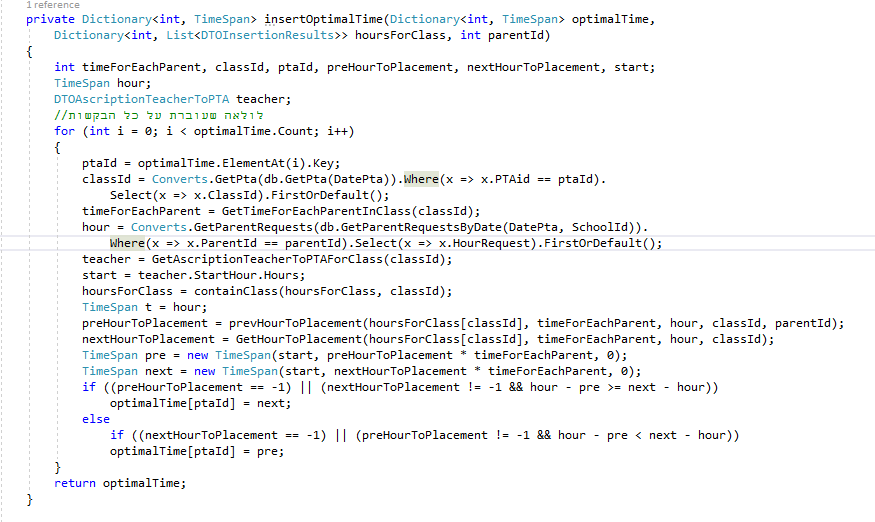
# האלגוריתם המרכזי | קוד התוכנית ותיעוד ?

ששתיהם מחזירים 1- כי חישבנו שיהיה מספיק זמן לכל ההורים בכיתה להגיע). אחרת בודקים מה ההפרש הקטן ביותר מהשעה שההורה ביקש.

פונקציה זו דואגת לכך שזמן הגעת ההורה יקבע לזמן האפשרי הקרוב ביותר לבקשתו.

כמו שכתוב לעיל זה יכול לחסוך, שאם יש זמן פנוי חמש דקות לפני השעה המבוקשת או שעה וחצי אח"כ אז המערכת תשבץ את השעה המוקדמת יותר.

אך הורה צריך לדעת שיכול להיות שישתבץ הרבה לפני/אחרי שביקש,

ויש לקחת בחשבון שדבר כזה עלול לקרות.

**findMinimum()**

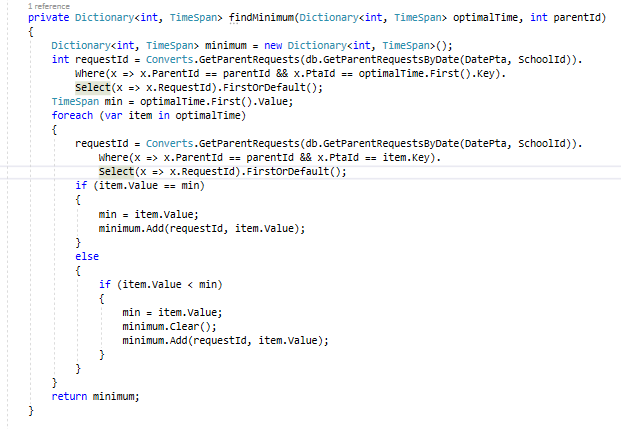
Dictionary ומכניסים ל- optimalTimeבשם dictionaryעוברים בלולאה על

Minimumחדש בשם

שבמקרה הפשוט ביותר מכניסים איבר אחד – אספה אחת, ששעת האספה הפנויה היא הכי מוקדמת – הכי קרובה לשעת הבקשה.

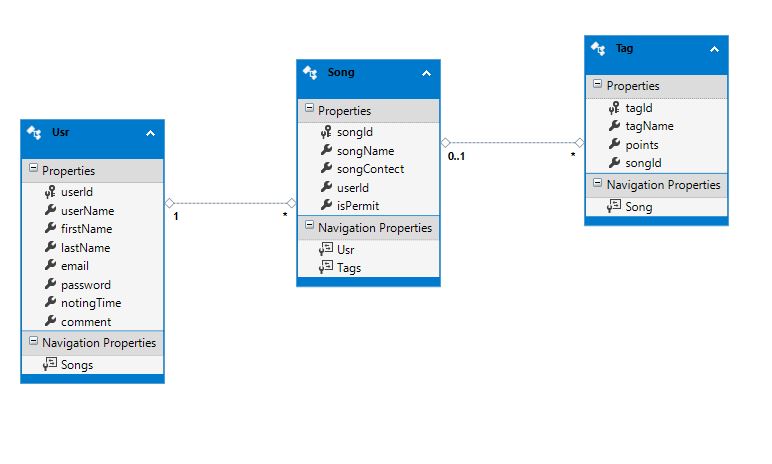
בכל מקרה אחר בו ישנן כמה אספות ששעת האספה הפנויה היא הכי מוקדמת שוות אז נכניס את כל האספות האלה למילון.

את המילון הזה שולחים לפונקציה בשם chooseMinimum() שהיא מחליטה איזה אספה לשבץ ראשונה במקרה ויש כמה אספות עם מינימום שווה.



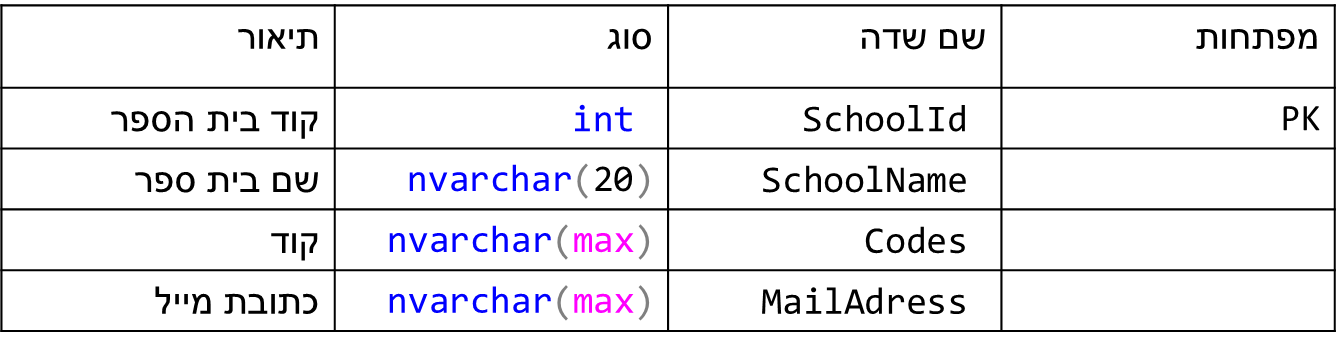
# Data base

## צילום מסד הנתונים

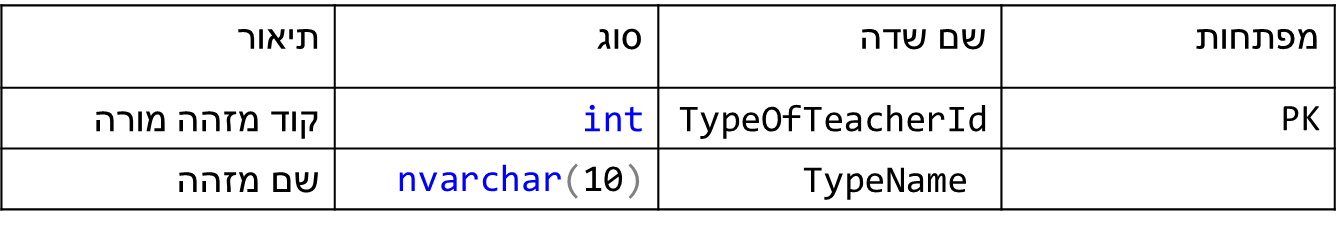


טבלאות מסד הנתונים / טבלאות קימות

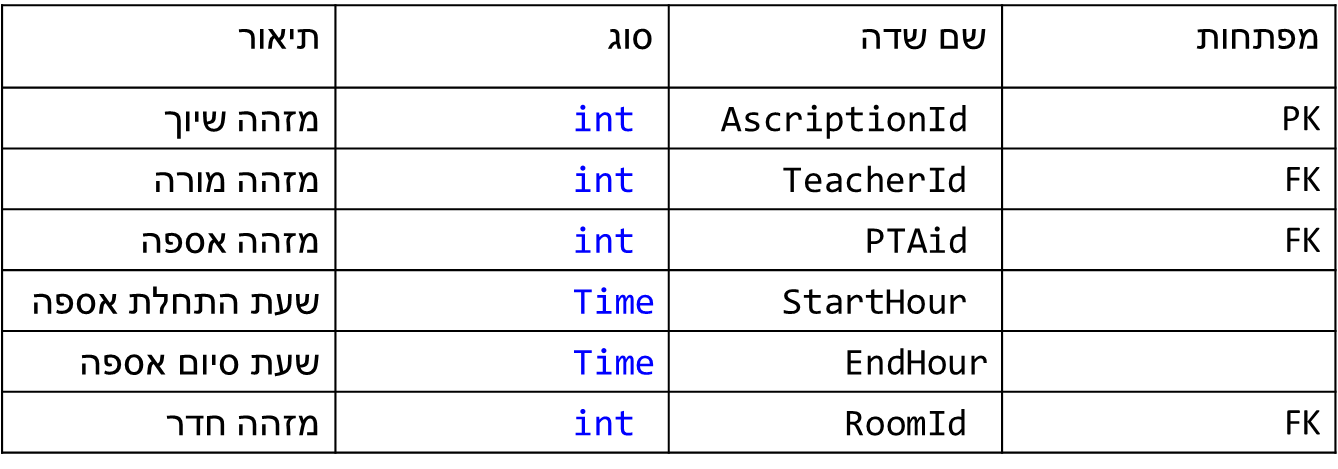
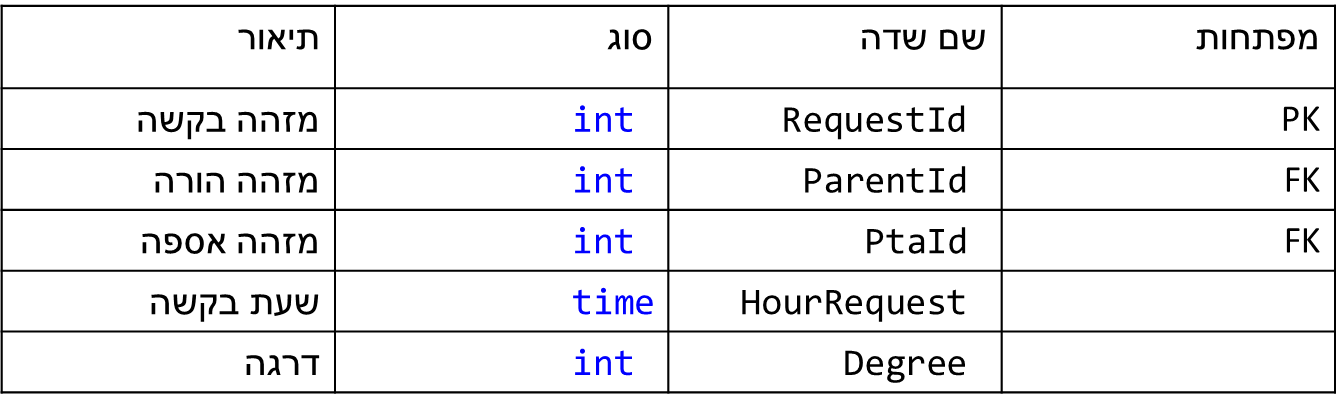
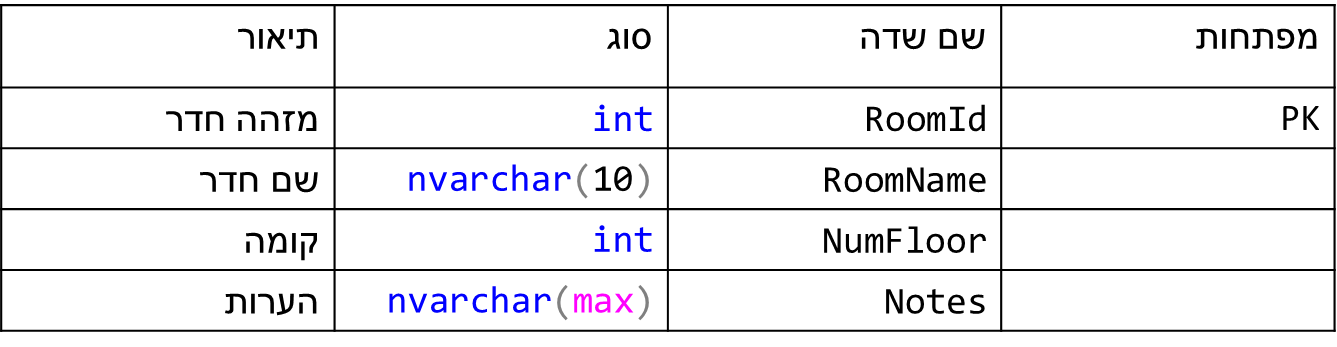
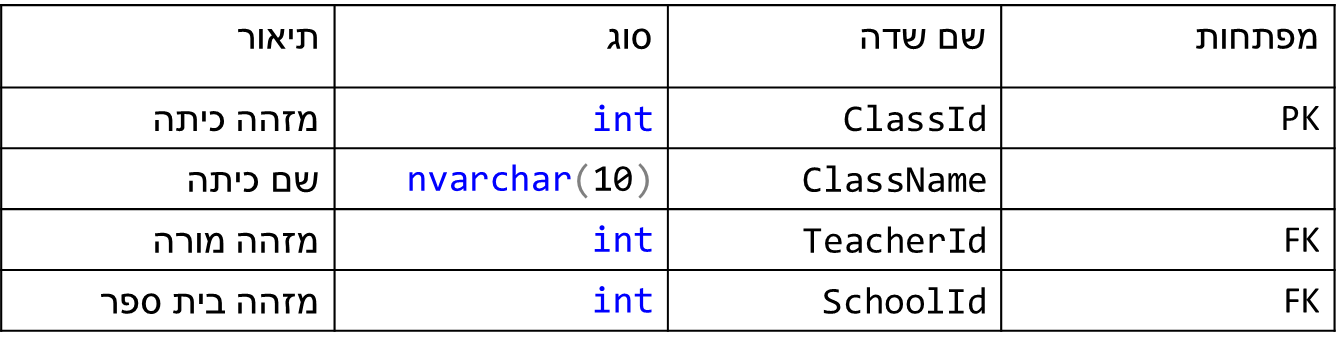
Usr משתמשים:



Songs - שירים

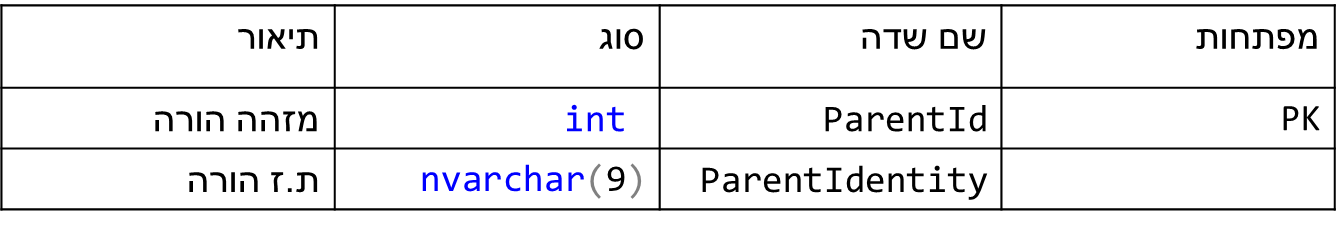


Tags- תגיות

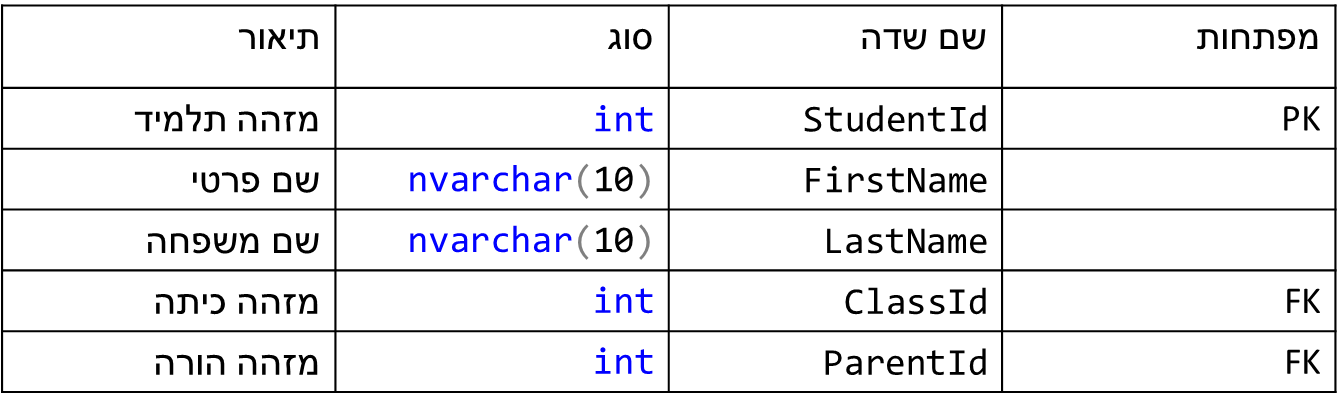


טבלאות לפיתוח עתידני:

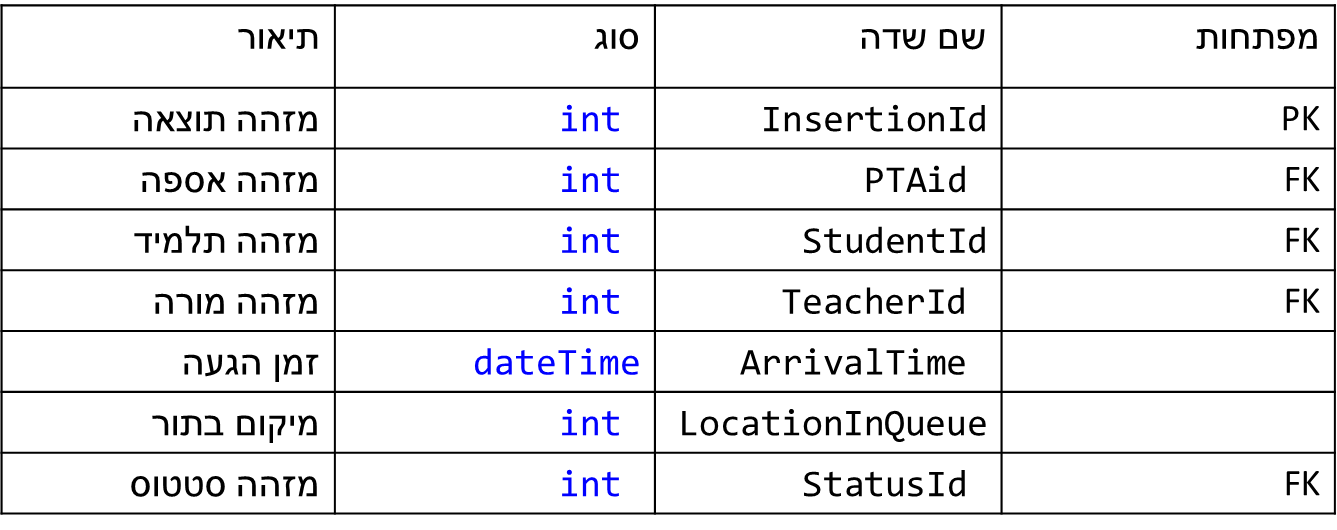
טבלת תאור מנגינה TuneDesc



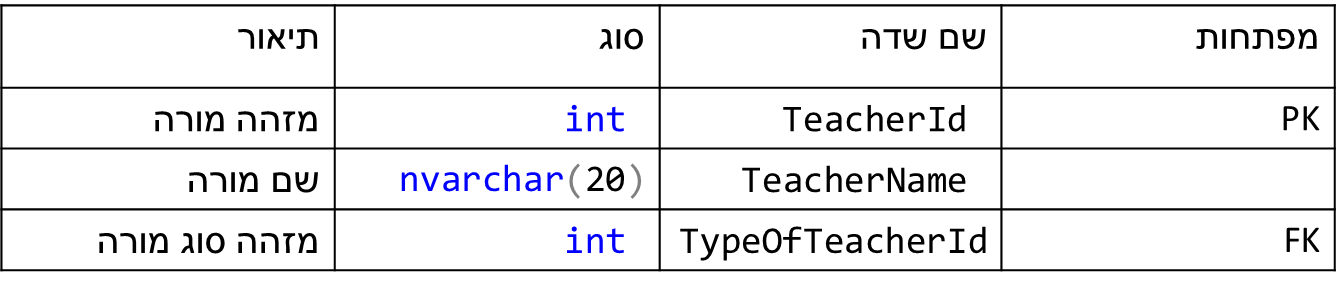
טבלת קטגוריה מקושרת לעצמה רקורסיבית CategoryLable :



טבלת מערכי הפעלות RhythmicsArrays :



. טבלת מערכי הפעלות לשירים RhythmicsArraysSong :



## ניתוח DB

האתר כאמור, מיועד לשיום, שיתוף וניהול שירים. כדי שהשימוש וניהול הנתונים יהיה בצורה היעילה ביותר, ובמקביל, שהפתוח העתידני לא ידרוש שנוי משמעותי במבנה בסיס הנתונים הנוכחי, הושקעה חשיבה במהלך ניתוח ואפיון הנתונים.

אפיון בסיסי הנתונים לא היה מורכב מאד, מאחר והאלגוריתם הראשי מפענח את נתוני השיר באופן חד פעמי במבני נתונים נדיפים, אך התכנון העתידני דרש עירנות היקפית ומודעות לסנכרון.

מבנה ה- data Base עבר מספר גרסאות עד שהגעתי לשלב הרצוי בהתאמת קשרי הגומלין הנוחים והבטוחים ביותר עבור הקוד.

כיוון שתהליך בנייתו של ה- Data Base נעשה עם מחשבה לעתיד כך שמבנהו יהיה מותאם גם להמשך הפיתוח, היה עלי להתחשב בכלל הנתונים ולבכר את המבנה שיענה על מירב הדרישות.

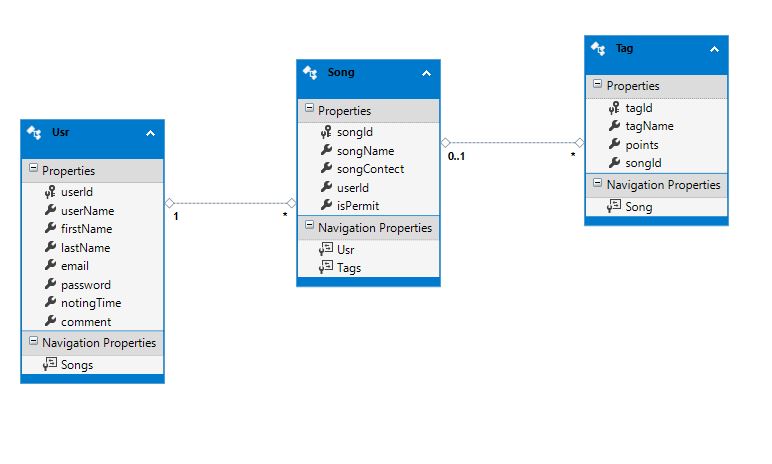
במהלך כתיבת הקוד שיניתי מספר נתונים בעזרת הגדרת היעדים והתמקדות בסיס נתונים יעיל ונכון.

**נושאים במסד הנתונים:**

בפרק זה אפרט כיצד בניתי את מסד הנתונים על מנת שישרת את המערכת באופן הנכון והיעיל ביותר. נרחיב על חלק מטבלאות ה- Data Base הצריכות ביאור מפורט ומעמיק יותר תוך חלוקה לפי סוגי תהליכים הקיימים במערכת. יש לציין, שהיקף המסד קטן יחסית להיקף הפרויקט, מאחר וחלק מהמידע לניתוח סטטי, וחלקו מתבסס על הרשת. הניתוח המורכב של התגיות מפושט למחרוזות, וכל המורכבות הינה חד פעמית, בזמן העלאת השיר, ואינה נשמרת בזכרון.

טבלת משתמשים Usrs:

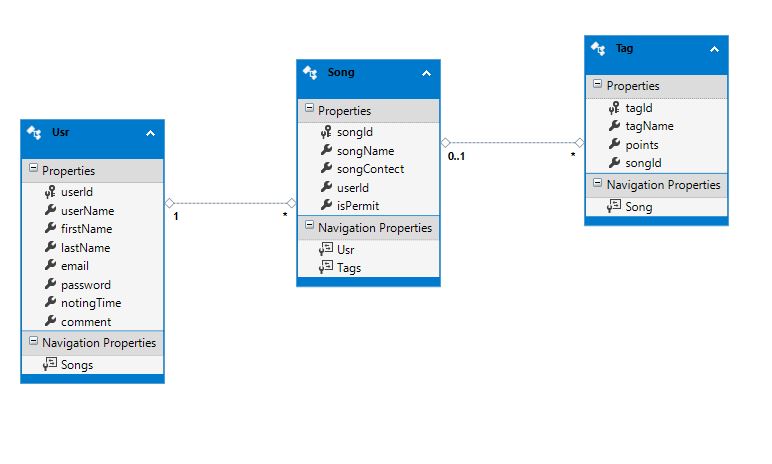
בטבלה זו מופיעים כל המשתמשים באתר, ומכילה את השדות הבאים: מזהה משתמש, שם פרטי, שם משפחה, שם משתמש, כתובת דוא"ל, סיסמא, תאריך הצטרפות, הערות. טבלה זו מקושרת בקשרי גומלין לטבלת שיר, כדי להוות בעלות משתמש לשיר.



טבלת שירים Songs:

טבלה זו מכילה את השדות הבאים: מזהה שיר, שם נוכחי של שיר, תוכן השיר (מילים) , האם מורשה להצגה באתר הגלובאלי.

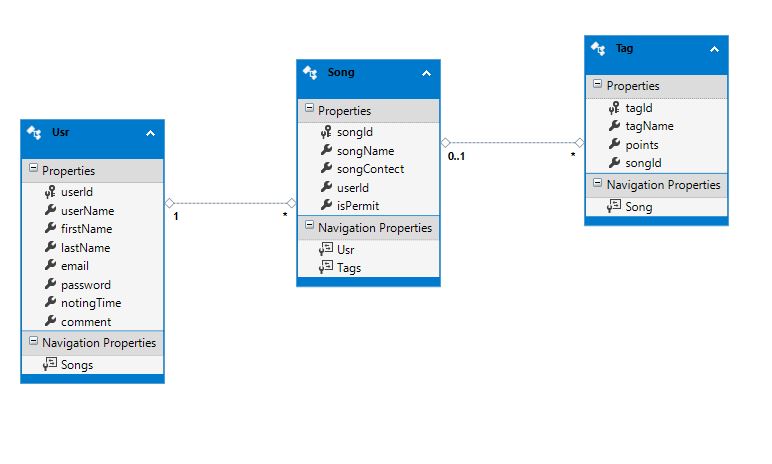
ומקושרת בקשרי גומלין לטבלאות: משתמש-כדי לאפשר הצגת השיר באתר הגלובאלי או באופן אישי למשתמש, ולטבלת תגיות: כדי לאפיין קבוצת תגיות לשיר.



**טבלת שירים:**

טבלת תגיות Tags:

טבלה זו מכילה את השדות הבאים: מזהה תגית, שם תגית, ניקוד נוכחי, שם השיר. טבלה זו מקושרת בקשרי גומלין לטבלת שירים, כדי להציג את השירים אליה משוייכת.



**טבלאות לפיתוח עתידי:**

טבלת תאור מנגינה TuneDesc:

טבלה זו תשמש לתאור סוגי המנגינות ואפיונן.

טבלת קטגוריה מקושרת לעצמה רקורסיבית CategoryLable :

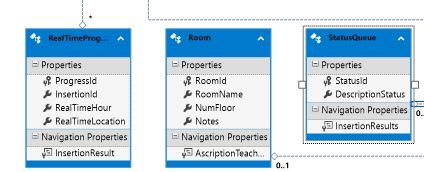
טבלה זו תספק את קטגורית השיר בהתאם למלל שלו, ולאתורי המשתמש הרציפים בסמוך למציאתו. טבלה זו מכילה תתי קטגוריות רקורסיבית כדי לאפשר סיווג מדויק.

טבלת מערכי הפעלות RhythmicsArrays :

טבלה זו תכיל אוסף שירים ותהווה כלי עזר להכנת מערכי הפעלות.

טבלת מערכי הפעלות לשירים RhythmicsArraysSong :

טבלה זו תבטא את קשר "רבים לרבים" בין טבלת מערכי הפעלות לטבלת שירים.



# UI – מדריך למשתמש

צד הלקוח פותח בריאקט, ובו כלול ממשק המשתמש עבור העלאה, הורדה, איתור וצפיה בטקסט השירים. על המשתמש להרשם למערכת לפני הכניסה, או לבצע כניסה חוזרת בדף ה"LOGIN" .

ההתנהלות באתר עבור המשתמש :

1. רישום או כניסה.

2. איתור שירים.

3. העלאת שירים.

4. הורדת שירים.

5. שיפור התגיות.

פיתוח עתידי: סיווג שירים לפי קטגוריות ותתי קטגוריות.

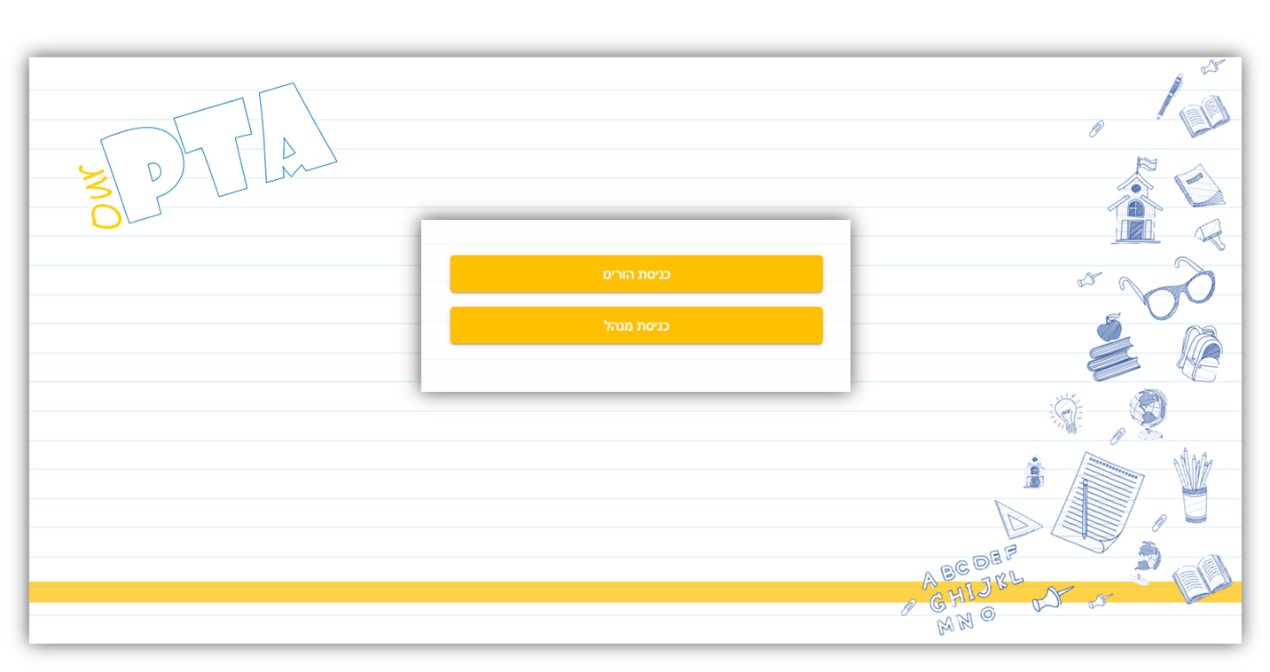
יצירת והורדת מערכי שעורי ריתמיקה והפעלה.

מסכים:

* מסך ראשי – כניסת משתמש למערכת.
  + התחברות / הרשמה למערכת.
  + מידע,
  + הוראות
  + פרסומות.
  + העלאת שירים.
  + מתג לצפיה במאגר האישי / הגלובלי.
  + טבלת השירים לפי תגיות.

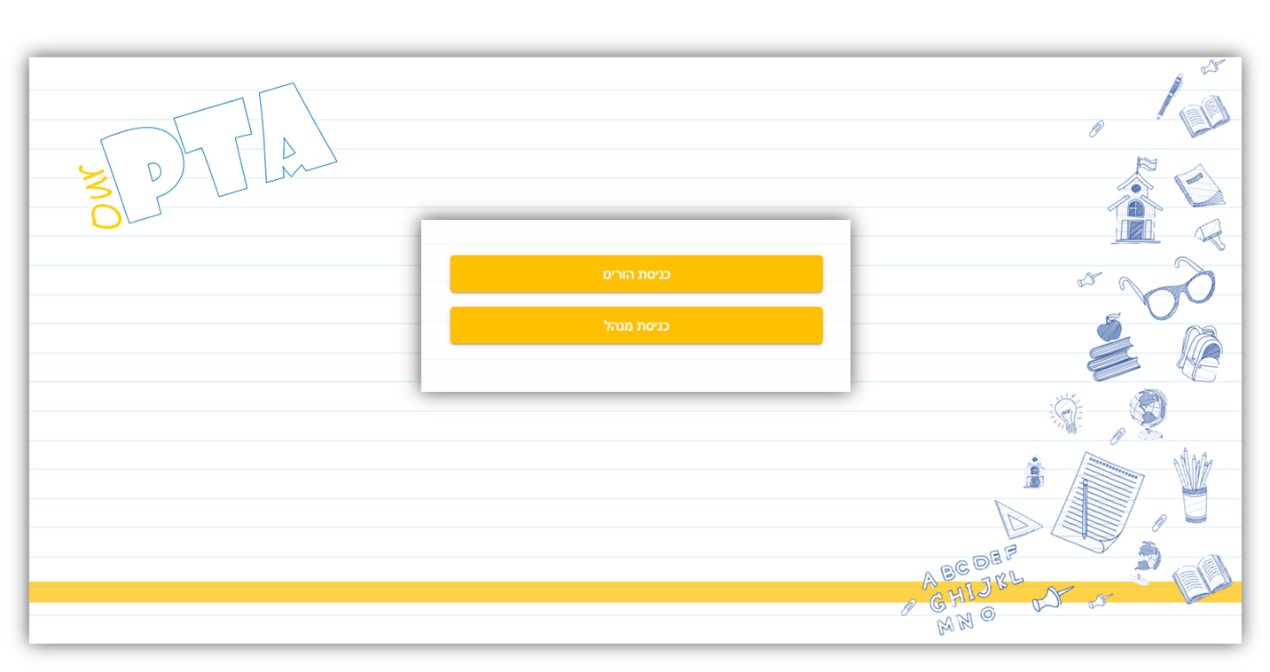
# פירוט המסכים:

## מסך כניסה –

כאשר משתמש נכנס למערכת עליו להקיש שם משתמש וסיסמא.

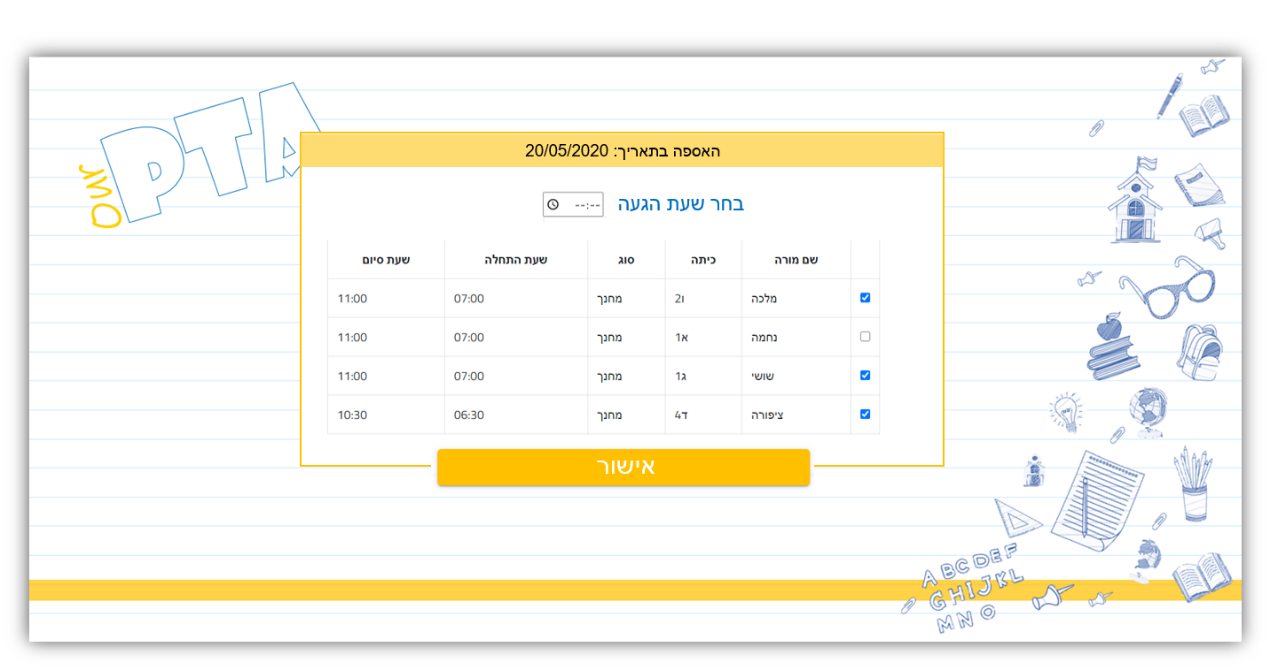
במידה והמשתמש אינו רשום במערכת, ניתן ללחוץ על כפתור "צור חשבון" ולמלא את הפרטים בדף הבא:

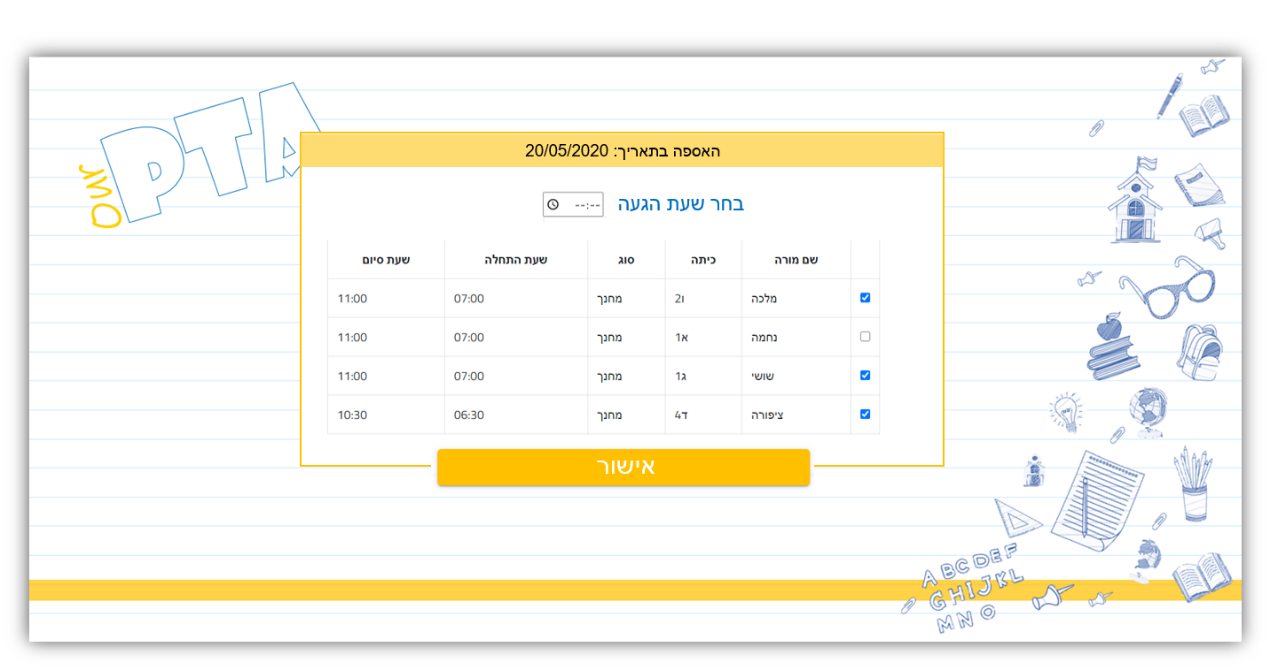
דף רישום משתמש חדש:



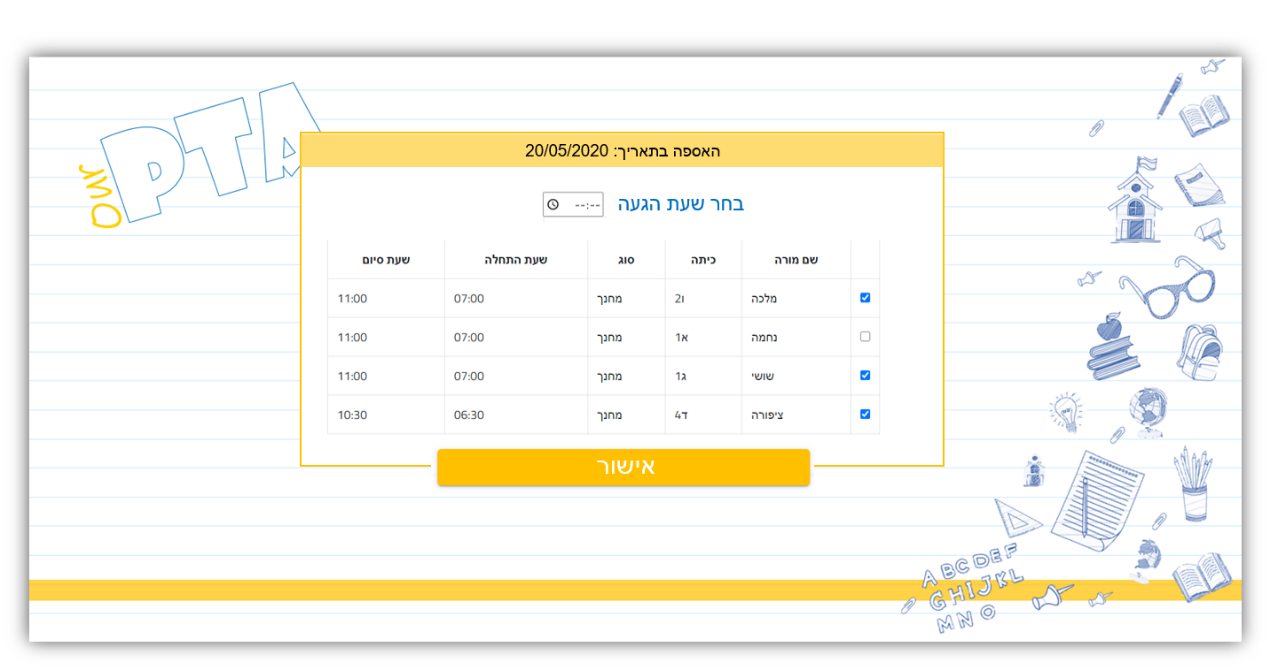
יש למלא את הפרטים לפי ההוראות ולאשר התחברות.

המסך הראשי בפרויקט:

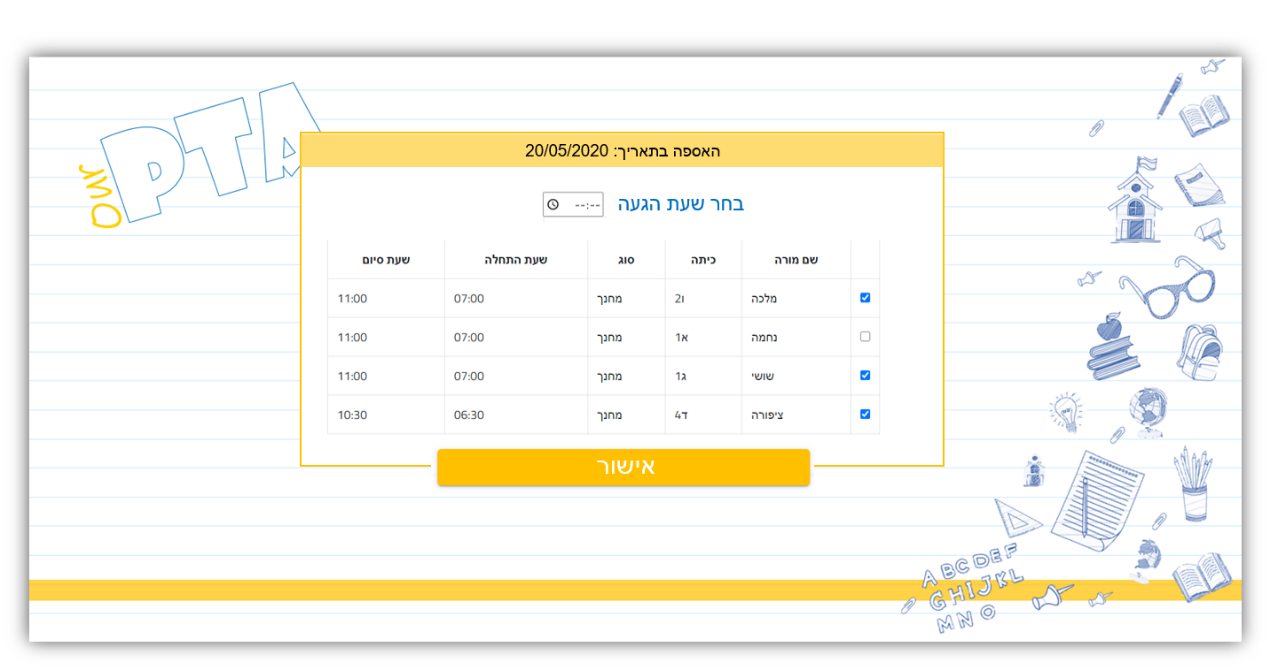


בעת הכניסה למערכת מוצגים השירים המורשים לצפיה, עם מתג לבחירה : מאגר גלובאלי / אזור אישי. המשתמש יכול להעלות שירים למאגר ע"י בחירה במקש ה +. המשתמש יכול לאתר את השיר המבוקש לפי תגיות ההשלמה האוטומאית, המופיעות בשטח העליון במסך. 

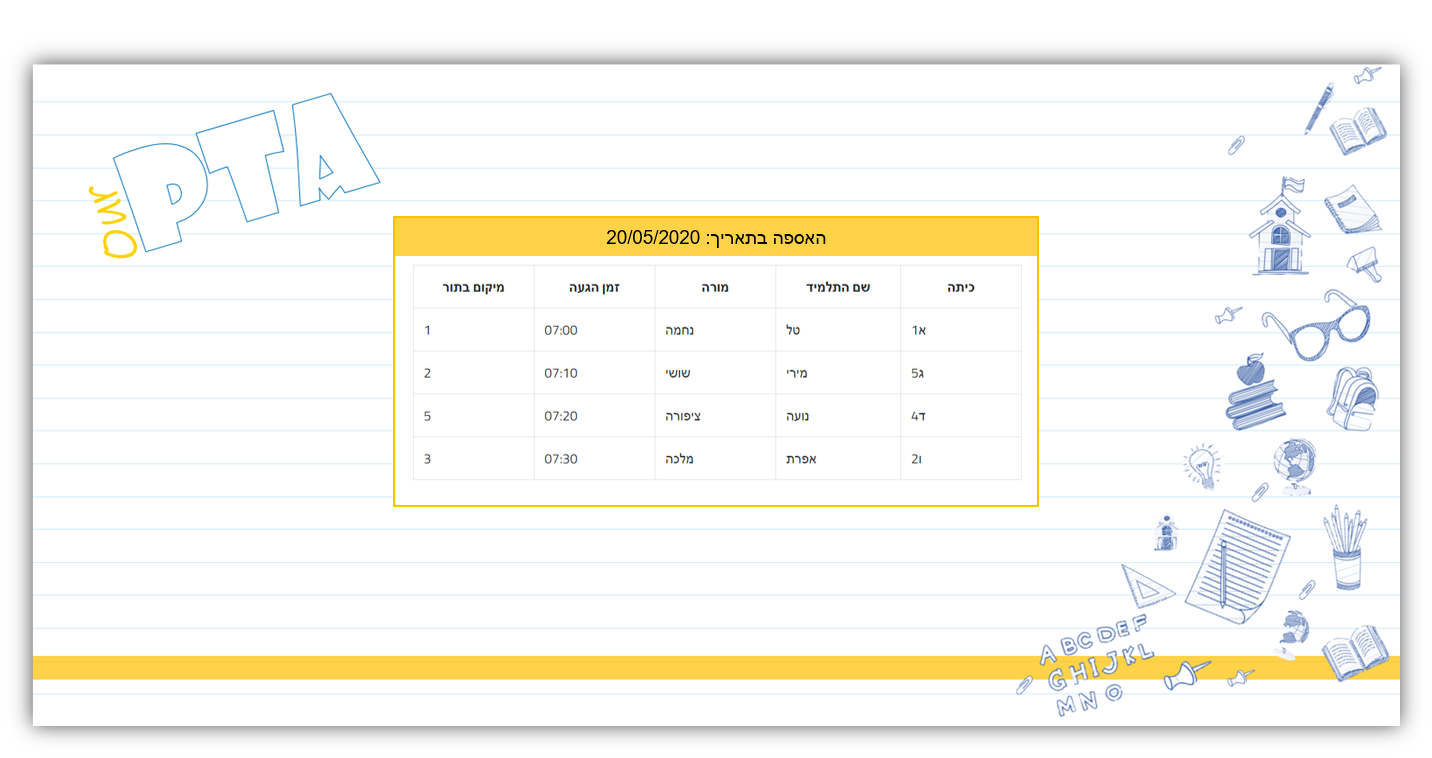
אם השיר הדרוש לא נימצא, ניתן להקליד את המילים המדויקות שבו וכך לאתרו בתיבת הניווט המופיעה לאחר לחיצה על כפתור החיפוש.



ניתן לבצע הורדה של שירים נבחרים ישירות מהטבלה או את כל השירים שאותרו ע"י התגיות, בלחיצה על הכפתור הסמוך לשורת התגיות בחלק העליון של המסך. המשתמש יוכל לעדכן את פרטיו ע"י הקשה על הכפתור המתאים בבר.



ניתן להעלות שירים בלחיצה על כפתור ה+ . על המשתמש להחליט אם ברצונו לאשר את שיתוף השיר באתר הגלובלי בכדי לאפשר שיפור ודיוק מיטבי בנתונים, בהתאם לזכויות היוצרים.



בעת העלאת השיר, המשתמש ממתין כך:

# 

# בדיקות והערכה

אחר כשנה של עבודה אינסטנסיבית, רכשתי ידע בטכנולוגיות רבות והתנסיתי בפתרון באגים ואיתורם ברשת.

אכן, ניתן דרכו לאתר את השיר הרצוי במהירות גבוהה ובקלות. מרבית התגיות ברורות וקולעות לתוכן השיר ולמשמעות, ובמקרים בהם לא : נעזרתי באיתור המשופר.

הפלטפורמה נעימה לעין כל כך, החסכון בזמן עצום, והשיתוף נעשה בקלות וביעלות.

גם באתגרים אותם הצבתי לעצמי הצלחתי לעמוד, ולהגיש פרויקט זה בספוק רב.

אמנם, יש אילוצים כמו במרבית התחומים בחיים, אך המערכת משרתת את מטרותיה ביעליות ובנוחות.

# ניתוח יעילות

מרבית תהליכים בפרויקט שולפים מידע מבסיס הנתונים ומנגישים אותו לצפיה, ויתבצעו בניתוח יעילות של כאשר = מספר הנתונים לטבלה בבסיס הנתונים , (מספר המשתמשים, מספר השירים, מספר התגיות לכל שיר).

יש להבחין שבמקרים מסוימים, כמו בעת כניסת משתמש למערכת, סבוכיות זמן הריצה תהא מורכבת מחבור שני גורמים, כמו: כאשר מספר המשתמשים הרשומים במערכת, ו = מספר השירים במאגר. בניתוח יסודי יותר : כאשר = מספר התגיות הממוצע לשיר.

יש להבחין שהטבלאות מאונדקסות , כך שזמן הריצה יהיה קצר יותר, וישאף ל .

לסבוכיות זמן ריצה זו, מתווספים חשובים נוספים.

* בעת העלאת שיר: מילות השיר מנותחות אחת אחת, ומתבצעות מספר סריקות עליהן כדי לפצלן בצורה הטובה ביותר. במהלך הסריקה, כמות התגיות הנסרקות מצטמצמת מאחר ומילים מתלכדות לתגיות באופן סדרתי, כפי דוגמא זו:

n ⊥ n/2⊥n/4⊥n/8=n(1) => בתחילה הסריקה מתבצעת על תגיות, עד להתלכדותן. אך במקרה הגרוע ביותר תתבצע הסריקה בכל פעם גם על כל מילות התגיות, וממילא סבוכיות זמן הריצה עלולה להיות במקרה זה , כאשר הינו מספר המילים בשיר.

* בעת איתור שיר: הנתונים כבר נשלפו, וסוננו. (על פי שירים אישיים ושירים מורשים לשיתוף) , מנגנון ההשלמה האוטומטית מובנה, והינו בסבוכיות עדינה יותר בשל הנתונים הקטנים יחסית: ?
* בעת איתור מתקדם: הנתונים כבר נשלפו, וסוננו. (על פי שירים אישיים ושירים מורשים לשיתוף) , הסבוכיות עדינה יותר בשל הנתונים הקטנים יחסית, והיא: כאשר מספר האותיות בתכני השירים שנשלפו בעת הכניסה לאתר.
* בעת שיפור באיתור המתקדם: השיר כאמור נמצא, ויש לאתרו בבסיס הנתונים על פי המפתח שלו, סבוכיות זמן הריצה הינה מאחר ויש לכל שיר מספור אוטומטי.

# אבטחת מידע

כדי לשמור על פרטיות המשתמשים ואבטחת המידע, נדרש כל משתמש לבחור סיסמא בעת ההרשמה למערכת. בכל כניסה לאתר, תדרש הכנסת הסיסמא.

על הסיסמא להתחיל בתו מיוחד (#\*&) ולאחריה 5 מספרים/אותיות.

לקוח יכול למחק מהמאגר רק שירים שהוא העלה, ומחיקת שירים שמשתמשים אחרים העלו לא תתאפשר כלל.

המידע לגבי השירים במאגר הגלובאלי החשוף לעיני כל משתמש בעת ההרשמה, הינו לצפיה בלבד.

ביצוע החפושים המתקדמים על שירים מהמאגר הגלובאלי, נותן אימון בהגינות המשתמשים.

כאשר נעשה במנגנון זה שמוש חריג לרעה, (פגימה במשמעות התגית) המשתמש יתבע חוקית מאת מנהלי המערכת. מעקב אחר פעילויות אלו, יתבצע בפיתוח העתידני.

# פיתוח עתידני

במהלך בניית הפרויקט עלו רעיונות לשדרוג הפרויקט.

ראשית: מימוש API בתמלול הטקסים.

כדי לאפשר העלאת קבצי אודיו. בשלב פיתוח זה, הקבצים הינם קבצי טקסט בסיומת TXT, ואלגוריתם הפרויקט מתייחס אליהם כאל קבצי אודיו מתומללים. זאת מאחר וAPI לתמלול קבצי שמע בעברית בעלות חודשית, מה שלא התאפשר במסגרת התקציב.

העלות של ה- API של חברת "?" לחודש הינה "?" . זו עתידה להתכסות ע"י תשלום המשתמשים.

גוגל גם היא מספקת אפשרות לשימוש ב"GOOGLE DOC" ע"י התקנת התקן שמע ווירטואלי בצד השרת, התקנתו כמנהל המערכת והשמוש בו, באמצעות הורדה מהאתר : https://www.vb-audio.com/Cable/

יש להבחין, כי זמן העלאת קבצי השמע ותמלולם יאריך את העלאת הקבצים לאתר במספר שניות נוספות, אך סבוכיות זמן הריצה לא תגדל משמעותית.

**שנית, מספר רעיונות לשדרוג שעלו במהלך בניית הפרויקט :**

1.       שיפור האתר ותוספים.

2.      אפיון מנגינה לכל שיר, כגון : קצבי, סוער, מנגינת משפטים, מנגינה קופצנית, מנגינת הפסקות, מנגינת חמש, נגינה מזרחית, ועוד.

3. קטגוריות משותפות עבור שירים, ובהמשך: פיתוח אלגוריתם המזהה הקשות סמוכות לקטגוריות.

# סיכום ומסקנות

עם הקלדת השורה הראשונה בקוד, היה ברור לי שאני לפני משהו גדול. מיום ליום התעצמה ההבנה. ואכן, הפרויקט דרש השקעה , טיפוח אינסטנטיבי, תחקירים ועבודה היקפית רבה.

בכל שלב מכתיבתו, הקפדתי לתעד כל שורת קוד, אפילו בלתי משמעותית.

את האלגוריתמים פיתחתי מיום ליום, שפרתי, תקנתי, ערכתי ועקבתי ....

העמדתי מול עיני את יעדי הפרויקט, העקרונות והדינמיות. השתדלתי להסיט ענינים שלא חשובים לצורך האלגוריתם ממש, וטפלתי בהם בהמשך. בכל מאות השעות של ההקלדה והבניה, התמקדתי במטרה:

שרטטתי דפים, עקבתי אחר מסד הנתונים ועקבתי בלי סוף.

למדתי לחפש ברשת באופן מקצועי, לאתר באגים סמויים ולכתב קוד באופן קריא ונכון.

ואכן, חשתי כי ההשקעה בבסיס נכון וחזק הינה השקעה נכונה לטווח ארוך.

טעמתי מעולם הבינה המלאכותית ??? ואני מקווה שכתיבת הפרויקט תהווה עבורי שער להטמעות בתחום.

כתיבת העצמים, הלמידה העצמית והבנת דרישות האלגוריתם, פתחה את היצירתיות שבי ואת החשיבה מחוץ לקופסה.

לאחר ההשקעה והעמל הרב, אני חשה ספוק עצום ומבינה מה רבה התועלת שהפקתי מכתיבת פרויקט הגמר, בתפילה שתסייע לי בעתיד הלא רחוק.

# ביבליוגרפיה:

* https://stackoverflow.com
* [https://getbootstrap.com](https://getbootstrap.com/)
* https://www.w3schools.com
* [https://icons8.com](https://icons8.com/)
* ?
* [(react-israel.co.il)](https://www.react-israel.co.il/react-router/)
* ויקיפדיה : מאגרי מילים
* JKL
* JKL
* תמלול טקסטים של גוגל, API <https://cloud.google.com/speech-to-text>

קוד התוכנית+תיעוד (האלגוריתם).

פיצול, איתור, שפור, שיום.

* פיצול: המחלקות: תגית. (מילה+ניתוח) . תגית סופית (מלל התגית, ליסט של ניתוחים). האינם הבינארי: לשם המחשה:

// The example displays the following output:

// All possible combinations of values without FlagsAttribute:

// 0 - None

// 1 - Black

// 2 - Red

// 3 - 3

// 4 - Green

// 5 - 5

// 6 - 6

// 7 - 7

// 8 - Blue

// 9 - 9

// 10 - 10

// 11 - 11

// 12 - 12

// 13 - 13

// 14 - 14

// 15 - 15

// 16 - 16

//

// All possible combinations of values with FlagsAttribute:

// 0 - None

// 1 - Black

// 2 - Red

// 3 - Black, Red

// 4 - Green

// 5 - Black, Green

// 6 - Red, Green

// 7 - Black, Red, Green

// 8 - Blue

// 9 - Black, Blue

// 10 - Red, Blue

// 11 - Black, Red, Blue

// 12 - Green, Blue

// 13 - Black, Green, Blue

// 14 - Red, Green, Blue

// 15 - Black, Red, Green, Blue

* // 16 - 16

כדי לפצל את השיר משתמשים ב3' פונקציות+תתי פונקציות לעבוד. פירוט:

ניתוח האינם, פונקציות הפיצול, פונקציות הטיהור.

* איתור: כדי לאתר לפי רצף מילים ולא לפי תגיות, פותח אלגוריתם לכך.
* שפור: העץ הבינארי המפורט.
* שיום: מניה פשוטה, צלום הפונקציה.
* בריאקט: מיון לפי התאמה ואז לפי ניקוד, הורדה.

צלומי מסך ותיעודים-מחר בעזרת ד' יתברך.