­­

# דף כריכה



# C:\Users\rivkasa\Documents\My Received Files\97EBD8E3.PNG

**מגישות** : רבקה ויזל, רחל טננולד

**שם המנחה** : הגב' מ. שמעונוביץ'

מסלול הנדסאי – מגמת תוכנה

# 2020

# דף שער

# לוגו

**שמות המגישות** :רחל טננולד ורבקה ויזל

חתימה\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**שם המנחה** : הגב' מוריה שילר

חתימה\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**שם רכז המגמה:** הגב' ח ברגמן

חתימה\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 2020

# C:\Users\rivkasa\Documents\My Received Files\97EBD8E3.PNG

# הצעה לפרויקט גמר

תאריך: דצמבר 2019

לכבוד: יחידת הפרויקטים מה״ט

**הצעה לפרויקט גמר**

**א. פרטי הסטודנטים**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **שנת סיום הלימודים** | **טלפון נייד** | **כתובת** | **ת.ז. 9 ספרות** | **שם הסטודנט** |
| 2019 | 0504168448 | אור החיים 45 | 207086844 | רבקה ויזל |
| 2019 | 0548527277 | תרשיש 9 אופקים | 315121178 | רחל טננולד |

שם המכללה: סמינר וולף – שלוחת המכללה למנהל ראשל"צ

סמל המכללה:

מסלול ההכשרה: הנדסאים

מגמת לימוד: תכנות מחשבים

מקום ביצוע הפרויקט: בבית ובסמינר

**ב. פרטי המנחה**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מקום עבודה/תפקיד | תואר | טלפון נייד | כתובת | שם המנחה |
|  |  |  |  | מוריה שילר |

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

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

# שם הפרויקט :

Ten10

# רקע

## תיאור ורקע כללי

הפרויקט מציג אתר למידה באמצעות משחק תחרותי בין שני קבוצות הכולל צבירת נקודות, ניצחונות וכן מאפשר להתאים רמה שאלות בכל נושא לימודי .

## מטרות המערכת

* המערכת תספק משחק חווייתי לחזרה על החומר הלימודי
* המערכת תאפשר למורה להכניס שאלות משחק בהתאם לחומר הלימודי או להשתמש בשאלונים מהמאגר
* המערכת תאפשר לקבוצות לצפות בתורות בניקוד שנצבר ובמספר הניצחונות
* המערכת תאפשר לשמור שאלות לשימוש מורות אחרות

# סקירת מצב קיים בשוק

אתרי למידה קיימים בנושאים ספציפיים ללא אפשרות לבחירת נושא רצוי ורמה לימודית מתאימה

## מה הפרויקט אמור לחדש או לשפר

הפרויקט פותח עבור לקוח ונמצא בשימוש ומאפשר הכנסת שאלות טריוויה בכל נושא לימודי ומחדד את הידע הנדרש אצל התלמידים ע"י משחק חוויתי.

# דרישות מערכת ופונקציונאליות

## דרישות מערכת, סביבת הטמעה ושימוש.

* המערכת תעבור קומפילציה והפצה בסביבת visual studio
* המערכת אמורה לרוץ בסביבת שרת אשר מריץ IIS  לקבלת בקשות לתצוגת דפי אינטרנט.
* המשתמש יוכל להריץ את האתר בכל מכשיר אשר מותקן עליו דפדפן אינטרנט.

## שרידות, ביצועים, התמודדות עם עומסים

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

גם עומס על שרת ה-SQL אינו צפוי בסדר גודל כזה של אתר מכיוון שהוא בנוי להתמודדות  בהצלחה עם עומסים כבדים.

## דרישות פונקציונאליות

### בדף הכניסה תהיה למשתמש מורה אפשרות כניסה והזנת שמות הקבוצות המשחקות משך זמן לענית תשובה והודעות שיוצגו לתשובה נכונה, שגויה, זמן תם

### משתמש מורה יוכל להזין שאלות למערכת ולשמור שאלון לשימוש מורות אחרות

### משתמש תלמיד יוכל חזור על החומר הלימודי בצורה חווייתית שמשקפת את ידיעותיו.

### משתמש תלמיד יוכל לצפות בניהול תורים בין הקבוצות

### משתמש תלמיד יוכל לצפות במספר הנקודות שצבר ובמספר הניצחונות עד סוף המשחק

### משתמש תלמיד יוכל לצפות במספר הנקודות וניצחונות שצברה הקבוצה היריבה

# בעיות צפויות במהלך הפיתוח

## הבעיות:

### **בעיה 1**: איך למנוע מהמשתמש לשנות תשובה שגויה

### **בעיה 2**: כיצד למנוע מהמשתמש לענות על שאלה שכבר ענה עליה

## פתרונות אפשריים:

### **לבעיה 1**:

* פתרון 1: להפוך את בחירת התשובות ללא פעיל לאחר בחירת תשובה אחת
* פתרון 2: לסגור את חלון השאלה מיד לאחר בחירת תשובה לפני שנגמר הזמן הקצוב
* פתרון 3: להמתין מספר שניות לבחירת תשובה ולסגור את החלון

### **לבעיה 2**:

* פתרון 1: למחוק את השאלה שהופיע ממערך השאלות
* פתרון 2: לציין ליד שאלה שכבר ענה שעל שאלה זו לא יצברו נקודות

## הפתרון הנבחר עבור כל אחת מהבעיות:

### **לבעיה 1**: פתרון 2 פתרון יותר תקין לא תהיה אפשרות לתלמיד לשנות את דעתו לאחר בחירת התשובה (חלון השאלה ייסגר לאחר המתנה שאנימציית בחירת התשובה תסתיים)

### **לבעיה 2**:פתרון 1 הגיוני ונח מבחינת המשתמש לא יוגרלו שאלות שהופיעו כבר במשחק

# פתרון טכנולוגי נבחר

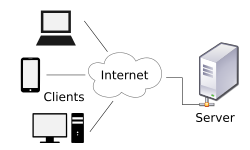
## טופולוגית הפתרון

המערכת מורכבת משרת  IIS המריץ את האתר בסביבת ה- server.

מסד נתונים - DB’s   של sql-server.

ממשק משתמש בצד הלקוח - דפדפן אינטרנט כלשהו: chrome, firefox  , internet explorer.

## דיאגרמה



## טכנולוגיות בשימוש:

כתיבת צד הלקוח: Html +Css +javaScript

כתיבת צד השרת: c#

מסד נתונים באמצעות Sql-Server

## שפות הפיתוח:

### בצד השרת: #C - שפה זו היא שפת תכנות פופולארית שמיועדת לפיתוח כללי של מגוון אפליקציות בכל התחומים. התחביר והעקרונות שלה פשוטים מצד אחד, אך עשירים ביכולות מצד שני. #C הוא תנאי הכרחי לשימוש במגוון טכנולוגיות.

### בצד הלקוח:

**HTML5** - היא שפת בניית אתרים אשר נותנת הנחיות לדפדפנים בנוגע לאופן הצגתו של דף האינטרנט מבחינת טקסט, תמונות, טבלאות עיצוב ועוד. זוהי שפת הקוד הבסיסית לסביבת ה-web עבור בניית אתרים . HTML5 הוא גרסה נוספת ומתקדמת הרבה יותר של HTML שכוללת הרבה כלים חדשים שבאו להקל על עבודת המפתחים.

**CSS** - כידוע, שפת HTML מגדירה את תוכן ומבנה הדפים באינטרנט, אך היא אינה מגדירה את העיצוב הויזואלי שלהם. CSS קיצור של Sheets Styles Cascading או בשמה העברי, גליונות עיצוב מדורגים, היא שפה המאפשרת להגדיר מאפייני עיצוב על מסמכי HTML.

**JavaScript**- היא שפת תכנות דינמית מומחית-עצמים המותאמת לשילוב באתרי אינטרנט ורצה על ידי דפדפן האינטרנט בצד הלקוח. השפה מרחיבה את יכולות שפת התגיות הבסיסית HTML ומאפשרת בכך ליצור יישומי אינטרנט מתוחכמים יותר.

## תיאור הארכיטקטורה הנבחרת

תבנית הארכיטקטורה הנבחרת היא **מודל שלוש השכבות** – ארכיטקטורה בה הלוגיקה העסקית של היישום והגישה אל הנתונים מפותחים ומתוחזקים כמודולים נפרדים. מלבד היתרונות הרגילים שלה תהיה לי כמפתח האפשרות להפריד בין השכבות ולטפל בכל שכבה בנפרד מבלי לגעת בשכבות האחרות. שיטה זו תקל במהלך פיתוח הפרויקט.

## חלוקה לתכניות ומודולים:

חלוקה לתוכניות לפי שלוש השכבות:

* שכבת התצוגה - שכבה הדואגת לניראות ממשק המשתמש.
* שכבת התחום העיסקי - שכבה אשר תפקידה לעבד ולהגיב לאירועים המתרחשים.
* שכבת הנתונים - שכבה המטפלת בגישה אל הנתונים.

# סביבת השרת

## ממשק המשתמש/לקוח – GUI

ממשק המשתמש יתקבל כ- HTML ויהיה מוצג באמצעות דפדפן אינטרנט .

## ממשקים למערכות אחרות / API

לא רלוונטי

## שימוש בחבילות תוכנה

* Entity framework
* webApi

# מבני נתונים וארגון קבצים

## שיטת האחסון

הנתונים ישמרו במסד נתונים של sql-server.

## מבני הנתונים

**מורות**- קוד מורה , שם מורה, מקצוע, נושא, כיתה, מס שאלון.

**שאלות** - קוד שאלה ,טקסט שאלה, טקסט תשובה1 , טקסט תשובה 2, טקסט תשובה 3, מס תשובה נכונה, מס שאלון.

## מנגנוני התאוששות מנפילה/ קריסה/ תמיכה בטראנזקציות.

במקרה של קריסת המערכת IIS מועלה באופן אוטומטי, כמו כן המערכת מסתמכת על מנגנון

הטראנזקציות של SQL-SERVER המנוהל ללא התערבות המשתמש.

# תרשימי מערכת מרכזיים

## Use Case

# C:\Users\rivkasa\Desktop\useCase.PNG תיאור המרכיב האלגוריתמי – חישובי

## איזה בעיה בא לפתור, איך יפתור?

משתמש מורה מזין שאלות למערכת, השאלות יתורגמו לקוד ג'אווה סקריפט ויופיעו במשחק ונשמרים במאגר.  
מורה יכול להשתמש אם שאלות שנשמרו במאגר המערכת  
לפני המשחק המורה מגדיר:  
1. זמן לעניית תשובה.   
2.שמות השחקנים.  
3.הודעות שיופיעו במהלך המשחק.  
4.תווח המספרים שיוגרל הניקוד.

הלוח נותן תמונת מצב אמתית של המשחק כולל: שחקן פעיל מספר נקודות וניצחונות לכל קבוצה, מספר הנקודות שצבר על כל תשובה נכונה וניקוד שהצטבר לניצחונות.

במהלך המשחק מוגרלות שאלות שלא נענו או שלא נענתה התשובה הנכונה.

במענה לתשובה נכונה לפני הזמן הקצוב:

1. יוגרלו נקודות או בונוסים.
2. כפתור השאלה הופך לבלתי פעיל אם הטקסט שהוגרל.
3. בתור שהצטבר 10 נקודות או יותר יתאפסו הנקודות ויעלה הניצחונות.
4. ויזואלית הדבש של השחקן יעלה.

לאחר ענית תשובה שגויה:

1.הודעה של תשובה שגויה.

2.מודל השאלה יאלם בלי המתנה עד לסוף הזמן הקצוב לעניית תשובה כדי לא לאפשר לסמן תשובה אחרת.

לאחר ענית תשובה בין נכונה בין שגויה:

1.התורות מתחלפים, עכשיו הדבורה שמסמלת את הקבוצה היריבה קופצת

ניצחון:

1.כאשר נגמר השאלות או אין משושים שלא הוגרל עליהם בונוסים

2. לוח המשחק מתחלף עם לוח ניצחון.

3.השחקן עם מספר הניצחונות הרב ביותר צנצנת הדבש שלו תתמלא עד סופה ,הדבורה המנצחת קופצת עם הצגת הודעה מתאימה.

## איסוף מידע וניתוחים סטטיסטיים (אנליטיקות)

לא רלוונטי

# תיאור/התייחסות לנושאי אבטחת מידע

יש לדאוג לאבטחת השרת מעומסים מופרזים הנגרמים יל ידי גורמים זדוניים, ואת שרת ה- server sql מ - sql injection.

# משאבים הנדרשים לפרויקט:

## מספר שעות המוקדש לפרויקט: 360

## חלוקת עבודה בין חברי הצוות

## ציוד נדרש: מחשב עם אינטרנט

## תוכנות נדרשות: Studio ,Visual Server  SQL ,  express iis ,דפדפנים

## ידע חדש שנדרש ללמוד לצורך ביצוע הפרויקט: חיבור בין בc# ל javascript

## ספרות ומקורות מידע

[https://github.com](https://github.com/fullcalendar/fullcalendar/releases)

<https://stackoverflow.com>

https://W3schools.com

# תכנית עבודה ושלבים למימוש הפרויקט:

* ייזום הרעיון –יולי.
* ניתוח מערכת – נובמבר.
* ניתוח מבנה נתונים – נובמבר.
* אפיון UI – UX - עד סוף נובמבר.
* כתיבת הלוגיקה העסקית -עד סוף דצמבר.
* כתיבת ממשק המשתמש -עד סוף דצמבר.
* עיצוב - עד סוף דצמבר.
* בדיקות התוכנה -עד סוף ינואר.
* התקנה והטמעה -עד סוף ינואר.

# תכנון הבדיקות שיבוצעו

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **מספר בדיקה** | **מקרי הבדיקה** | **ידנית/ אוטומטית** | **חשיבות** | **הערות** |
| 1 | שמירת שאלון | ידנית | גבוהה | לאחר מכן בחירת שאלון מהמאגר ולראות שהתוסף שאלון |
| 2 | משחק עד לניקוד שחקן 10 או יותר | ידנית | גבוהה | אם בתור הניקוד שהוגרל על תשובה נכונה התוסף ל10 או יותר עולה בניצחונות מתאפס הניקוד ותור השחקו הבא |
| 3 | משתמש עונה תשובה נכונה לשאלה | ידנית | גבוהה | כפתור השאלה הופך לבלתי פעיל לאחר תשובה נכונה |
| 4 | משחק עד לניצחון תוך וידוא שאין שאלות שחזרו על עצמן | ידנית | גבוהה | השאלות במשחק הן רק שאלות שהמשתמש לא נשאל או לא ענה עליהם תשובה נכונה |

# בדיקות יחידה (unit test)

בדיקות יחידה לא נכללו בתוכנית הלימודים

# בקרת גרסאות (version control)

**חתימת הסטודנט חתימת המנחה האישי**

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

הערות ראש המגמה במכללה:

אישור ראש המגמה במכללה:

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

הערות הגורם המקצועי מטעם מה"ט:

אישור הגורם המקצועי מטעם מה"ט:

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

# אישור הצעת הפרויקט ממה"ט

# תודות

רצינו להביע את תודתנו לכל מי שסייע לנו במהלך העבודה על הפרויקט.

לגב' מוריה שילר מנחת הפרויקט שהדריכה אותנו במהלך העבודה ולימדה אותנו את עקרונות השפה והטכנולוגיה שבה עבדנו.

לגב' ברגמן ,רכזת מגמת מחשבים אשר דאגה שנקבל את כל התנאים שהיינו צריכות בשביל

הפרויקט ונתנה לנו עצות בעת הצורך.

להורים שלנו שעמדו לצדינו, תמכו ועודדו אותנו לאורך הפרויקט.

וכמובן, אין אנחנו מספיקין להודות. .. לריבונו של עולם שראינו הרבה סיעתא דשמיא במהלך

העבודה.

# הצהרה

שם הסטודנט:

רחל טננולד ת.ז. 315121178

רבקה ויזל ת.ז. 207086844

שם המכללה בה לומד הסטודנט: סמינר ע"ש הרב וולף

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

פרויקט הגמר נעשה על סמך הנושאים שלמדתי במכללה ובאופן עצמאי.

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

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

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

תאריך: חתימת הסטודנט:

**אישור המנחה האישי**

הריני מאשר שהפרויקט בוצע בהנחייתי, קראתי את ספר הפרויקט ומצאתי כי הוא מוכן לצורך הגשת הסטודנט להגנה על פרויקט גמר .

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

**אישור ראש המגמה**

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

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

.

# תקציר

האתר 10TEN מהווה דרך חווייתית לרכישת ולתרגול יסודות המקצוע פותח עבור בית ספר לפי דרישותיו ונמצא בשימוש.

מתכונת המשחק היא כזו: הכיתה מתחלקת לשתי קבוצות משחקים במשחק טרויה מקורי ומאתגר צוברים בונוסים וניקוד צופים בניהול בטורים ובלוח הנכבש ובעיקרון צבירת ניצחונות הייחודית, משך זמן נותר וכו' וכך מתרגלים את החומר.

המורה יכול ולערוך ולהוסיף שאלות לחידון ולשומרם במערכת או לחילופין להשתמש ממאגר הקיים. להגדיר הודעות שיוצגו, תווך הניקוד שיוגרל, משך זמן לעניית תשובה נכונה, שמות המשחקים, והכל להתאמה מקסימלית של החומר הלימודי לרמת התלמידים

האתר מעוצב בצורה מעניינת ומושכת לעין על מנת לגרות לעניין את התלמידים לשחק שוב ושוב עד לשליטה מלאה בחומר הנלמד(הגרפיקה החדשנית במימון הלקוח).

# תוכן העניינים

[דף כריכה 1](#_Toc25058805)

[דצמבר 2019 1](file:///C:\Users\shraibern\Downloads\תבנית%20לספר%20-%20בסיס%20לבניה%20עליו%20(1).docx#_Toc25058807)

[דף שער 2](#_Toc25058808)

[הצעה לפרויקט גמר 2](#_Toc25058809)

[דצמבר 2019 2](file:///C:\Users\shraibern\Downloads\תבנית%20לספר%20-%20בסיס%20לבניה%20עליו%20(1).docx#_Toc25058810)

[אישור הצעת הפרויקט ממה"ט 2](#_Toc25058811)

[תודות 3](#_Toc25058812)

[הצהרה 4](#_Toc25058813)

[תקציר 4](#_Toc25058814)

[תוכן העניינים 6](#_Toc25058815)

[1. מבוא 8](#_Toc25058816)

[2. מדריך למתכנת: 9](#_Toc25058817)

[2.1. אסטרטגיות טכנולוגיות: 9](#_Toc25058818)

[2.2. תיאור מבנה הפרויקט: 10](#_Toc25058819)

[צד ה server 10](#_Toc25058820)

[2.3. עקרונות התכנון/ הבניה/ הניתוח: 11](#_Toc25058821)

[2.3.1. עקרונות תיאורטיים: 11](#_Toc25058822)

[2.4. תרשימים: 13](#_Toc25058823)

[2.4.1. עץ תהליכים: 13](#_Toc25058824)

[2.4.2. תרשים Uml: 14](#_Toc25058825)

[תרשים מראה המחלקות: 15](#_Toc25058826)

[2.5. מבנה נתונים מאוכסנים: 16](#_Toc25058827)

[2.5.1. sql: 16](#_Toc25058828)

[2.5.2. מבנה קבצים ותיקיות: 17](#_Toc25058829)

[Client: 17](#_Toc25058830)

[2.6 תוכן הפרויקט: 18](#_Toc25058831)

[2.6.1 תיאור המחלקות: 18](#_Toc25058832)

[1.6.2 תיאור הפונקציות 21](#_Toc25058833)

[3. מדריך למשתמש: 28](#_Toc25058834)

[3.1. הוראות כלליות לשימוש באתר: 28](#_Toc25058835)

[3.2. מסכים: 29](#_Toc25058836)

[3.2.1. מסך כניסה: 29](#_Toc25058837)

[3.2.2. מסך הרשמה: 29](#_Toc25058838)

[3.2.3. המסך ראשי - תלמיד: 30](#_Toc25058839)

[3.2.3 מסך המשחק: 30](#_Toc25058840)

[3.2.4 מסך הלימוד: 31](#_Toc25058841)

[3.2.5 מסך חידון הטרויה: 31](#_Toc25058843)

[3.2.6 מסך עריכת חידון 32](#_Toc25058844)

[3.2.7 מסך תלמידים: 32](#_Toc25058846)

[4. סיכום ומסקנות: 33](#_Toc25058848)

[מה הקנה לנו הפרויקט? 33](#_Toc25058849)

[5. נספחים: 34](#_Toc25058850)

[7. ביבליוגרפיה: 35](#_Toc25058852)

## מבוא

בתיווך הסמינר פנה אלינו בית ספר שביקש פיתוח משחק שיתאים לכל גיל ולכל מקצוע.

ראינו לנכון לבוא ולתת דרך חווייתית יותר למורה לבדוק את ידיעות תלמידיו, לא רק למידה קונבנציונלית של שינון – חזרה- בחינה, אלא בדרך מהנה ומגוונת.

ולכן פרויקט הגמר שלנו הוא אתר לחזרה על החומר הלימודי לכיתות בית הספר היסודי בצורה חווייתית וכייפית בהנחיית המורה ובהתאמה למקצוע ולגיל.

האתר מכיל ממשק עבור מורה: המורה עורך את שאלות הטריוויה והתשובות ושומרם במערכת אופציה למשחק עם שאלונים במאגר אם אפשרות חיפוש פי מקצוע. כמו כן הוא יכול לצפות ברשימת התלמידים שתחתיו ובנקודות שצברו במהלך המשחק ובחידון.

תלמידים המשחקים באתר, לומדים שיעור קליל וקצר המקנה להם חומר לימודי, ולאחר מכן הם יכולים לשחק במשחק ולתרגל את החומר, עד שישלטו בו ישר והפוך. במהלך המשחק הם צוברים נקודות ויכולים להעלות את הנקודות.

בנוסף, הם יכולים לשחק בחידון הטריוויה ולבחון את ידיעותיהם בחומר.

הן המשחק והן החידון מקנית חוויה לימודית החשובה כל כך ברכישת ידע במקצועות בית הספר היסודי.

## מדריך למתכנת:

### אסטרטגיות טכנולוגיות:

בחרנו לכתוב את האתר בטכנולוגית WEB API כך שצד server נכתב בשפת C# וצד client- בשפת javaScript תוך שימוש בטכנולוגיות נוספות כמו: Entity Framework, ועוד.

וזאת בכדי להפיק את המירב ולהנות מגישה נוחה וברורה שמציע לנו web api ובנוסף הנחות שמציעה שפת ה javaScript

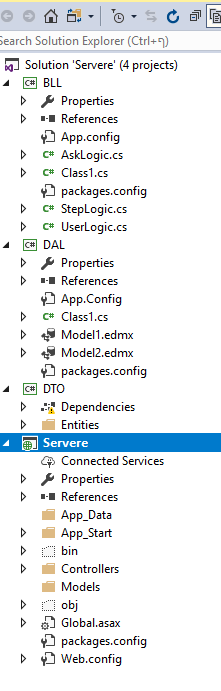
את האתר עצמו כתבנו תוך שימוש בסביבת העבודה של 2017Visual Studio

ואילו את בסיס הנתונים שעומד מאחוריו יצרנו בעזרת Sql Server

השתמשנו בEntity Framework Model כדי לקשר בין בסיס הנתונים לתכנה.

השתמשנו בשפות הבאות בכתיבת הפרויקט

* C#: שפת c# היא שפת תכנות שפותחה ע"י מיקרוסופט ונחשבת לאחת משפות התכנות הפופולריות בעולם. היא מיועדת לפיתוח כללי של מגוון אפליקציות בכל התחומים מאתרי web , דרך משחקים, מאפליקציו למכשירי מובייל וטאבלטים ועד לשירותי ענן. התחביר והעקרונות שלה הם פשוטים מצד אחד אך עשירים ביכולת מצד שני
* Html: זהוי השפה הטבעית ליצירת דפי אינטרנט ברשת. זו שפה פשוטה, אוניברסלית המאפשרת לעורכי אתרים ליצור דפים מורכבים שמכילים טקסט ותמונות, שיכולים להראות בידי כל המשתמשים ברשת האינטרנט ללא תלות בסוג המחשב או בסוג הדפדפן
* javaScript: javaScript היא שפת תכנות של html והאינטרנט מאפשרת למפתחים להטביע קטעי סקריפט ("תוכניות קטנות") בתוך מסמכי html ובכך להפוך את הדפים לדינמיים יותר
* css: css היא שפת עיצוב המגדירה את תבנית העיצוב של מסמכי ה html. Css למשל מטפלת בגופנים,צבעים,שורות,גובה,רוחב,תמונות רקע, מיקום מתקדם ודברים רבים נוספים



### תיאור מבנה הפרויקט:

### צד ה server

ה- Solution שלנו כולל 4פרויקטים:

* DAL- פרויקט זה מורכב ממקור נתונים –מסד הנתונים שלנו, וממערכת תוכנה entity framework אשר תפקידה לקרוא את המידע הנדרש למערכת, לשמור את העדכונים ,ולהוסיף מידע חדש או למחוק פרטי מידע קיימים. מכיל את המחלקות התואמות לבסיס הנתונים, נבנה בשיטת DBFirst
* DTO- מכיל מחלקות המקבילות לDAL העוזרות בשליחה וקבלת אובייקטים מה client ל- server.

BL- הפרויקט שאחראי על הלוגיקה של המערכת, עוסקת בעיבוד המידע, בחישובים שונים ושליחתו לשכבת התצוגה.

בפרויקט זה נממש את הפונקציונאליות של המערכת.

מסד הנתונים והממשק משתמש מתקשרים

דרך השכבה הזו

* API - זוהי שכבת הAPI- התקשורת עם ה client מכילה פונקציות אליהם יפנה ה client בקריאות http הפונקציות מאוחסנות ב Controllers שונים המחולקים לפי נושאים

### עקרונות התכנון/ הבניה/ הניתוח:

#### עקרונות תיאורטיים:

* **הפרדת שכבות**

ישנם אתרים המבוססים על ארכיטקטורת שכבות הנקראת –.Tree-Tier- Architecturez בארכיטקטורה זו קיימת הפרדה בין השכבות במבנה של -. זוהי תבנית DAL-BL-GUI זוהי תבנית עיצוב בסיסית שמגדירה הפרדת האפליקציה לשכבת נתונים, שכבת לוגיקה ושכבת ממשק משתמש. לתבנית עיצוב זו יתרונות רבים:

* **תחזוקה**:

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

* **נוחות פיתוח**:

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

* **בדיקות**:

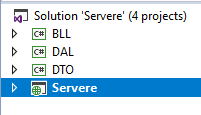
כאשר ישנה תקלה היא מבודדת בשכבה שאליה היא שייכת. לדוגמא אם לא קיבלנו רשימת נתונים לתצוגה, נבדוק קודם בשכבת הנתונים (DAL)האם הנתונים שם תקינים. במידה שהם אכן תקינים, נעבור לבדיקת שכבת הביניים, שכבת הלוגיקה BL)) ואם גם שם הנתונים תקינים, נדע בוודאות שהתקלה היא בשכבת התצוגה (GUI) ונפתור אותה שם.

* **שימוש חוזר**:

במידה והשכבות הDAL וה BLעובדים היטב וברצוננו להחליף את פלטפורמת הUI , ניתן לעשות זאת במינימום מאמץ.

* **אבטחה**:

נוח יותר להגדיר Interface-API Logic - לשכבה מסוימת בלי לחשוף מבני נתונים, LOGIC או DATA שלא רלוונטיים למשתמש, כמו גם חסימה בפני האקרים (רלוונטי יותר בטכנוליגה WEB ) ברמות שונות

**מימוש:**

הפרויקט שלנו מחולק ל-2 חלקים client ו- server צד ה server מחולק לפי מודל שלושת השכבות –DAL,BL השכבה השלישית היא web-api שלמעשה משמשת כapi ובנוסף יש לנו את שכבת התצוגה המשתמשת בשכבת ה API

שלושת השכבות

* **הורשה**

בהורשה משתמשים בתכנות מונחה-עצמים, ירושה היא דרך לבסס יחס "סוג-של" (is a ) בין עצמים. המימוש הנפוץ של הגדרה זו היא בעזרת מחלקות. מחלקה יכולה לקבל בירושה תכונות והתנהגות של מחלקת-האם שלה (נקראת גם "מחלקת-על", "מחלקת אב" או "מחלקת בסיס"). היחסים בין מחלקות מגדירים היררכיה של מחלקות. מנגנון הירושה מסייע במידול של תחומים בדרך דומה לזאת המוגדרת באופן "טבעי" על ידי העוסקים בהם.

לדוגמה: אם ריבוע הוא סוג-של (או מקרה פרטי של) מצולע, ניתן להגדיר יחס ירושה בין המחלקה מרובע למחלקה מצולע; על פי רוב, אובייקט של המחלקה היורשת (מרובע) יתמוך בכל הפעולות המוגדרות על מצולע כלשהו, ובנוסף יתמוך בפעולות נוספות, המיוחדות למרובע. בדומה, ניתן להגדיר יחס ירושה בין המחלקה ריבוע למחלקה מרובע.

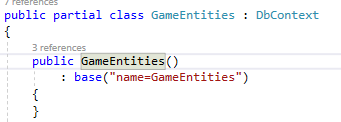
שימושים:

ירושה מאפשרת להשיג מספר מטרות:

1. שימוש חוזר בקוד. בעזרת ירושה ניתן לכתוב מחלקות בעלות טווח התנהגות נרחב, תוך כתיבת קטעי קוד קטנים יחסית שמרחיבים מחלקה קיימת.
2. מידול של העולם האמתי באופן דומה לזה המוגדר על ידי האדם. למשל, ניתן לכתוב היררכיה של מחלקות המקבילה לטקסונומיה המקובלת בביולוגיה (יען היא סוג של ציפור, האדם הוא סוג של יונק, וכן הלאה).

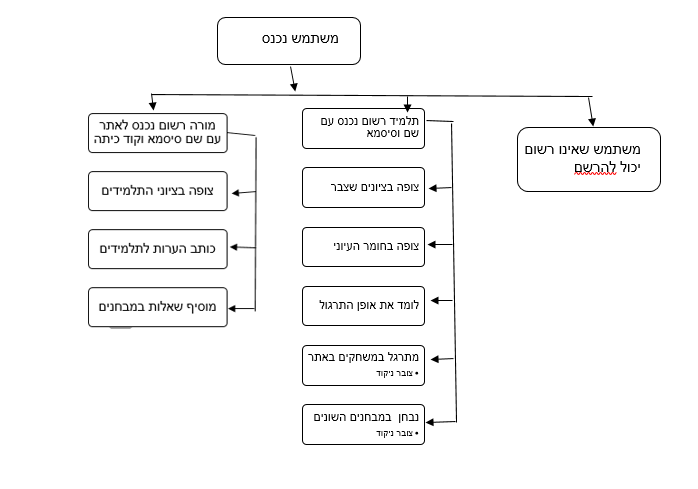
פולימורפיזם (רב צורתיות) של זמן-ריצה. משתנה מטיפוס (סטטי) מסוים יכול להתנהג בצורות שונות, לפי הטיפוס של האובייקט שהוא מתייחס אליו בזמן ריצה. למשל משתנה מהמחלקה בעל חיים שנדרש לבצע פעולה מסוג "השמע קול" יבצע פעולות שונות לחלוטין במקרה שהאובייקט הוא מהמחלקה כלב או חתול. ניתן לבצע זאת גם ללא ירושה (בעזרת פקודות תנאי), אך ירושה מאפשרת מימוש מודולרי, גמיש וקל להרחבה.

בפרויקט השתמשנו בהורשה במקרים הבאים:

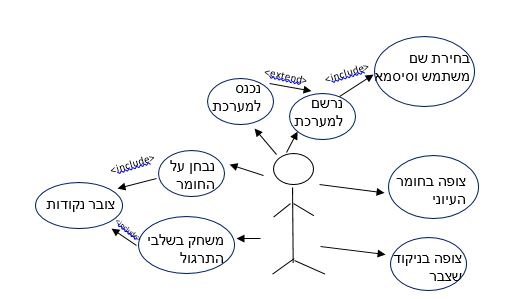
* + כל controller יורש ממחלקת ApiController לדוגמה : 
  + השימוש ב EntityFrameWork כולל ירושה:המחלקהgameEntities יורשת ממחלקת DbContext. המקושרת ל System.Data.Entity ומאפשרת תקשורת עם מסד הנתונים בצורה חלקה ונוחה.

### תרשימים:

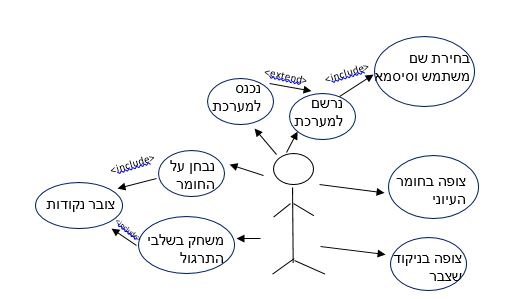
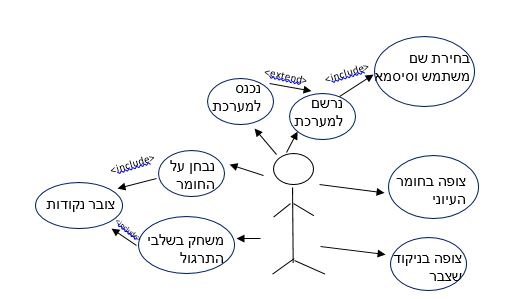
#### עץ תהליכים:



#### תרשים Uml:



מורה



תלמיד

כותב הערות

צופה בציונים

#### תרשים מראה המחלקות:



### מבנה נתונים מאוכסנים:

#### sql:

בבניית מסד הנתונים נתנו את דעתנו על מבנה הטבלאות והקשרים ביניהם ע"מ ליצור בסיס מערכתי איתן.

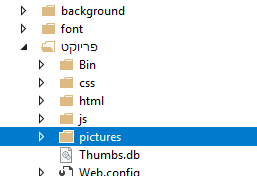
מסד הנתונים נבנה באופן מקיף שילבנו טבלאות ושדות שנועדו לצורך הרחבת הפרויקט בהמשך . נפרט כאן מעט על כל טבלה:

* **טבלת user –** המכילה את פרטי המשתמש.
* **טבלת userAnswer –** המכילה את פרטי הניקוד שצבר המשתמש בבחינה.
* **טבלת step –** המכילה את פרטי הניקוד שצבר כל משתמש בכל שלב.
* **טבלת answer–** המכילה את התשובות האפשריות לשאלה .
* **טבלת ask –** המכילה את השאלות למבחן.

#### מבנה קבצים ותיקיות:

##### Client:

תיקיית זו מכילה את קבצי הcss- דפי העיצוב לאתר



תיקיית זו מכילה את קבצי הJavaScript- דפי הקוד לתפעול האתר

תיקיית זו מכילה את קבצי הhtml- דפי תוכן האתר

תיקיית זו מכילה תמונות בהם השתמשנו לצורך עיצוב האתר וממוקמת בפרויקט client בתיקיית ה pictures

#### 2.6 תוכן הפרויקט:

#### 2.6.1 תיאור המחלקות:

**צד server**

**מחלקות ב- Dal:**

User **- משתמש**

public class User

{

public int Id { get; set; }

public string UserName { get; set; }

public string Password { get; set; }

public virtual ICollection<Step> step { get; set; }

public Nullable<int> teacherId { get; set; }

public static User ToDto(DAL.Users a)

{

return new User() { Id = a.Id, UserName = a.UserName, Password = a.Password, teacherId=a.teacherId };

}

public static DAL.Users ToDal(User a)

{

return new DAL.Users() { Id = a.Id, UserName = a.UserName, Password = a.Password, teacherId=a.teacherId};

}

}

Step**- שלבי המשחק**

public class Step

{

public int Id { get; set; }

public Nullable<int> userId { get; set; }

public int step { get; set; }

public int dots { get; set; }

public static Step ToDto(DAL.step a)

{

return new Step() {Id=a.Id, userId = a.userId, step=a.step1, dots=a.dots };

}

public static DAL.step ToDal(Step a)

{

return new DAL.step() { Id=a.Id, userId=a.userId, step1=a.step, dots=a.dots };

}

// public List<Albom> Albom { get; set; }

}

Ask- שאלות

public class Ask

{

public int Id { get; set; }

public string Text { get; set; }

public int TrueAnswer { get; set; }

public Nullable<int> teacherId { get; set; }

public static Ask ToDto(DAL.ask a)

{

return new Ask() { Id = a.Id,Text=a.Text, TrueAnswer=a.trueAnswer, teacherId=a.teacherId };

}

public static DAL.ask ToDal(Ask a)

{

return new DAL.ask() { Id = a.Id, Text=a.Text, trueAnswer=a.TrueAnswer, teacherId=a.teacherId};

}

}

Answer- תשובות

public class Answer

{

public int Id { get; set; }

public int AskId { get; set; }

public string Text { get; set; }

public static Answer ToDto(DAL.answer a)

{

return new Answer() { Id = a.Id , AskId=a.askId,Text=a.Text };

}

public static DAL.answer ToDal(Answer a)

{

return new DAL.answer() { Id = a.Id,askId=a.AskId, Text=a.Text };

}

}

StudentScore- נקודות למשתמש

public class StudentScore

{

public string StudentName { get; set; }

public int Lavel { get; set; }

public int Score { get; set; }

}

**צד client**

* **פרויקט האנגולר**

פרויקט זה מייצג את צד ה client ומכיל תקיות js,css,html

###### 1.6.2 תיאור הפונקציות

1.6.2.1 הפונקציה מזיזה את מיקום השחקן על הלוח

//========================= שינוי מיקום==============================

function doo(evt) {

girlh = 8;

girlw = 3;

remove = [2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2];

idgirl = []

if (checkallkod(evt) == true) {

var i = 0;

interval = setInterval(function () {

if (i == idgirl.length) {

point = 3;

clearInterval(interval);

setTimeout(function () { points(p); }, 500);

}

if (remove[i] != 2)

{

var poo = remove[i] + 1;

var r = "ballon" + poo;

document.getElementById(r).style.display = "none";

}

place(idgirl[i], "girl");

i++;

}, 2000)

place(idgirl[i], "girl");

}

}

1.6.2.2 הפונקציה בודקת האם הקוד שהוקלד תקין

//========================== בדיקת קוד==============================

function checkallkod(evt)// להורידevt

{

var str = document.getElementById("try").value;

if (str.indexOf(';') == -1)//אם חסר ;

{

al = "; הקוד לא תקין יתכן שחסר ";

alertdiv();

return false;

}

var ceanValue = document.getElementById("try").value.replace(/\n/g, "");

var main = ceanValue.split(";");

for (var i = 0; i < main.length - 1; i++) {

idCheck = Check(main[i]);

if (idCheck == false)

return false;

idgirl[i] = idCheck;

}

p=main.length-1;

return finish();

}

1.6.2.3 הפונקציה בודקת את תקינות הקוד

**//========================= בדיקת תקינות==============================**

**function Check(mainn) {**

**var res = mainn.split(" ");**

**//res.splice(" ");//אם אין 3 מילים**

**for (var i = 0; i < res.length; i++) {**

**if (res[i] == "") {**

**res.splice(i, 1);**

**i--;**

**}**

**}**

**if (res.length != 3) {**

**al = "משהו בתחביר לא נכון ";**

**alertdiv();**

**return false;**

**}**

var fgo = go(res);

var fnum = num1(res);

if (fgo == false || fnum == false)

return false;

var fdir = 0;

fdir = dir(res, fnum, girlh, girlw);

if (fdir == false)

return false;

return fdir;

}

1.6.2.4 הפונקציה בודקת אם הבלונים הסתיימו

//========================= בדיקת סיום בלונים ==========================

function finish() {

for (var i = 0; i < idgirl.length; i++) {

for (var k = 0; k < ballon.length; k++) {

if (ballon[k] == idgirl[i])

{

ballon[k] = 0;

remove[i] = 0;

}

}

}

if (ballon[0] != 0)

{

al = "משהו בקוד לא תקין";

alertdiv();

}

return ballon[0] == 0 ;

}

1.6.2.5 הפונקציה שומרת את השאלות שהמורה הוסיף

//---------------שמירת שאלות

function saveQuestion() {

user = localStorage.getItem('user');

userId = JSON.parse(user)["Id"];

document.getElementById('wrapperForm').style.visibility = "hidden";

var ans = [];

ans.push(document.forms["myForm"]["Ans1"].value);

ans.push(document.forms["myForm"]["Ans2"].value);

ans.push(document.forms["myForm"]["Ans3"].value);

ans.push(document.forms["myForm"]["Ans4"].value);

var data = {

'ask': document.forms["myForm"]["ask"].value,

'ans':ans,

'currentAns': document.forms["myForm"]["currentAns"].value,

'teacherId': userId,

}

$.ajax({

type: "POST",

headers: {

'Content-Type': 'application/json; charset=utf-8'

},

url: "http://localhost:50665/api/Ask/addQuestion",

data: JSON.stringify(data),

dataType: "json",

success: function (data) {

location.reload();

}

});

}

1.6.2.6 הפונקציה מחזירה את רשימת התלמידים

//---------------------רשימת התלמידים

function onload() {

user = localStorage.getItem('user');

var table = document.getElementById('studentTable');

$.ajax({

type: "GET",

headers: {

'Content-Type': 'application/json; charset=utf-8'

},

url: "http://localhost:50665/api/Users/getStudents/" + JSON.parse(user)["Id"],

dataType: "json",

success: function (data) {

data.forEach(d => {

let newRow = table.insertRow(-1);

let name = newRow.insertCell(0);

let newText = document.createTextNode(d['Score']);

name.appendChild(newText);

let lavel = newRow.insertCell(0);

newText = document.createTextNode(d['Lavel']);

lavel.appendChild(newText);

let score = newRow.insertCell(0);

newText = document.createTextNode(d['StudentName']);

score.appendChild(newText);

})

}

});

}

1.6.7.7 הפונקציה מחשבת את מספר הנקודות שהשחקן יקבל

//-------------------------חישוב הנקודות

function points(p, lavel)

{

if (p == point)

{

dot = "10 :מספר הנקודות";

m = 10;

}

else if (p > point && p <= point + 2)

{

dot = "6 :מספר הנקודות";

m = 6;

}

else

{

dot = "3:מספר הנקודות";

m = 3;

}

localStorage.setItem("sum", +localStorage.getItem("sum") + m);

var data = {

'sum': m,

'userId': JSON.parse(user)["Id"] ,

'lavel': lavel

}

$.ajax({

type: "POST",

headers: {

'Content-Type': 'application/json; charset=utf-8'

},

url: "http://localhost:50665/api/step/updateDot",

data: JSON.stringify(data),

dataType: "json",

success: function (data) {

alertdivv(data);

}

});

}

1.6.2.8 הפונקציה מחזירה את רשימת השאלות

//------------------------רשימת השאלות

function getAsks() {

user = localStorage.getItem('user');

userId = JSON.parse(user)["Id"];

return $.ajax({

type: "GET",

headers: {

'Content-Type': 'application/json; charset=utf-8'

},

data: '',

url: "http://localhost:50665/api/ask/getAsks/" + userId,

success: function (data) {

var table = document.getElementById('AskTable');

data.forEach(d => {

let newRow = table.insertRow(-1);

d['answer'].forEach(a => {

let ans = newRow.insertCell(0);

let ansText = document.createTextNode(a['Text']);

ans.appendChild(ansText);

})

let text = newRow.insertCell(0);

newText = document.createTextNode(d['Text']);

text.appendChild(newText);

let remove = newRow.insertCell(0);

remove.addEventListener("click", function () {

$.ajax({

type: "GET",

headers: {

'Content-Type': 'application/json; charset=utf-8'

},

url: "http://localhost:50665/api/Ask/removeAsk/" + d["Id"] ,

dataType: "json",

success: function (data) {

location.reload();

}

})

});

newText = document.createTextNode('X');

remove.appendChild(newText);

})

}

})

}

1.6.2.9 הפונקציה בודקת אם הפרטים נכונים

//-----------------------בדיקת תקינות הפרטים

function checkPassword() {

document.getElementById("userNameError").innerText = "";

document.getElementById("passwordError").innerText = "";

document.getElementById("confirmPasswordError").innerText = "";

var userName = document.forms["myForm"]["userName"];

var password = document.forms["myForm"]["password"];

var confirmPassword = document.forms["myForm"]["confirmPassword"];

if (userName.value == "") {

document.getElementById("userNameError").innerText = "הכנס שם משתמש.";

userName.focus();

return false;

}

if (password.value == "") {

document.getElementById("passwordError").innerText = "הכנס סיסמא.";

password.focus();

return false;

}

if (confirmPassword.value == "") {

document.getElementById("confirmPasswordError").innerText = "הכנס אימות סיסמא.";

confirmPassword.focus();

return false;

}

if (confirmPassword.value != password.value) {

document.getElementById("confirmPasswordError").innerText = "אימות סיסמא חייב להיות זהה לסיסמא";

confirmPassword.focus();

return false;

}

return true; }

## מדריך למשתמש:

### הוראות כלליות לשימוש באתר:

**מדריך למורה:**

* היכנס לדף הגדרות המשחק בכפתור הראשי בדף הבית
* הזן שאלות למשחק או בחר שאלות ממאגר קיים
* הגדר את שמות השחקנים ושאר הגדרות המשחק
* מלא את פרטיך ושמור את השאלות בדף השמירה
* הנך יכול למחוק שאלות
* הנך יכול להוסיף שאלה ע" לחיצה על הוסף שאלה

**מדריך לתלמיד:**

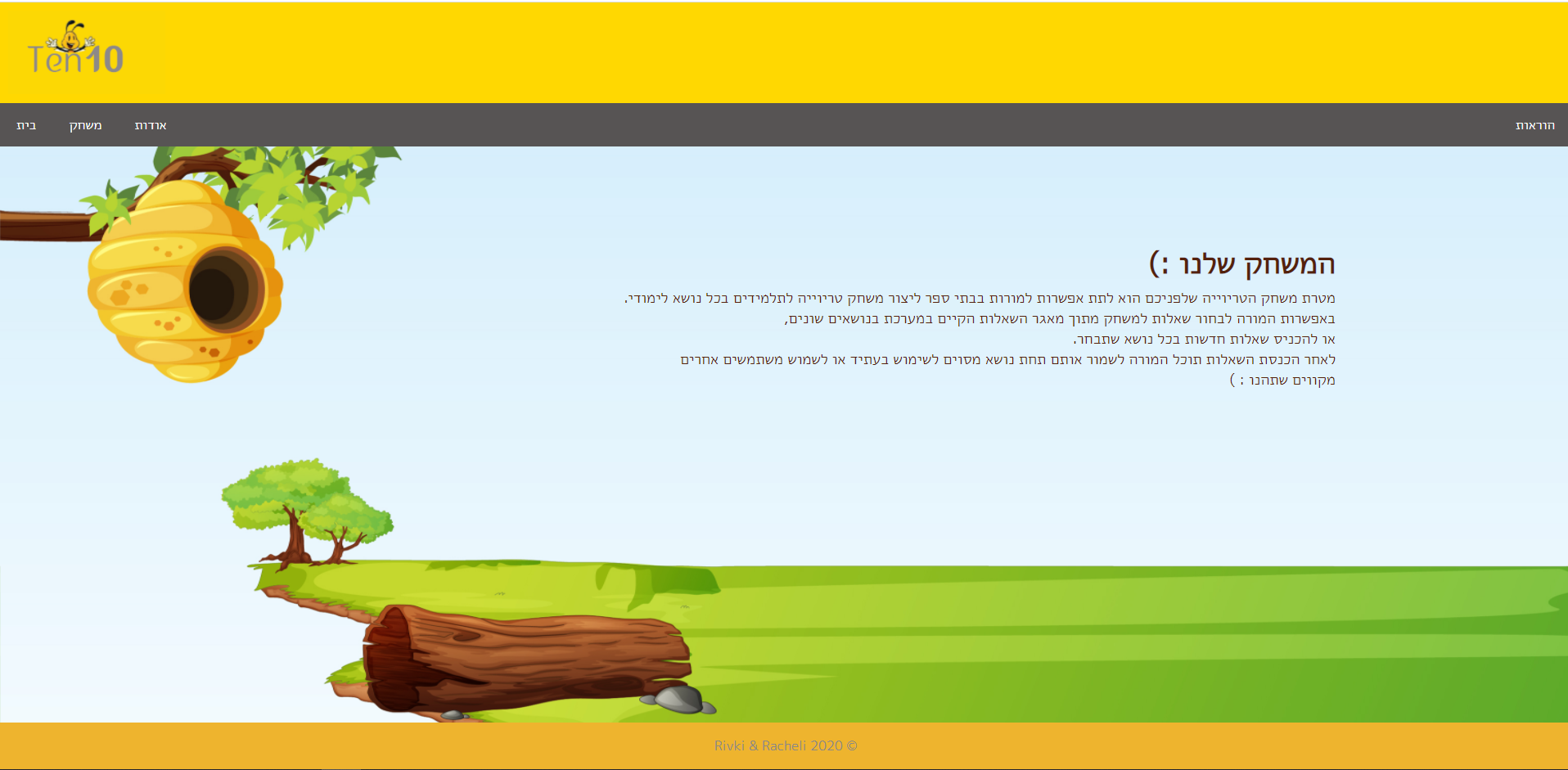
* היכנס לדף המשחק בלחיצה על הטאב 'משחק' בתפריט הכניסה
* לחץ על אחד מהמשושים שבלוח המשחק
* ענה על השאלה שתופיע במסך
* העבר את התור לשחקן השני
* בהצלחה!!

### מסכים:

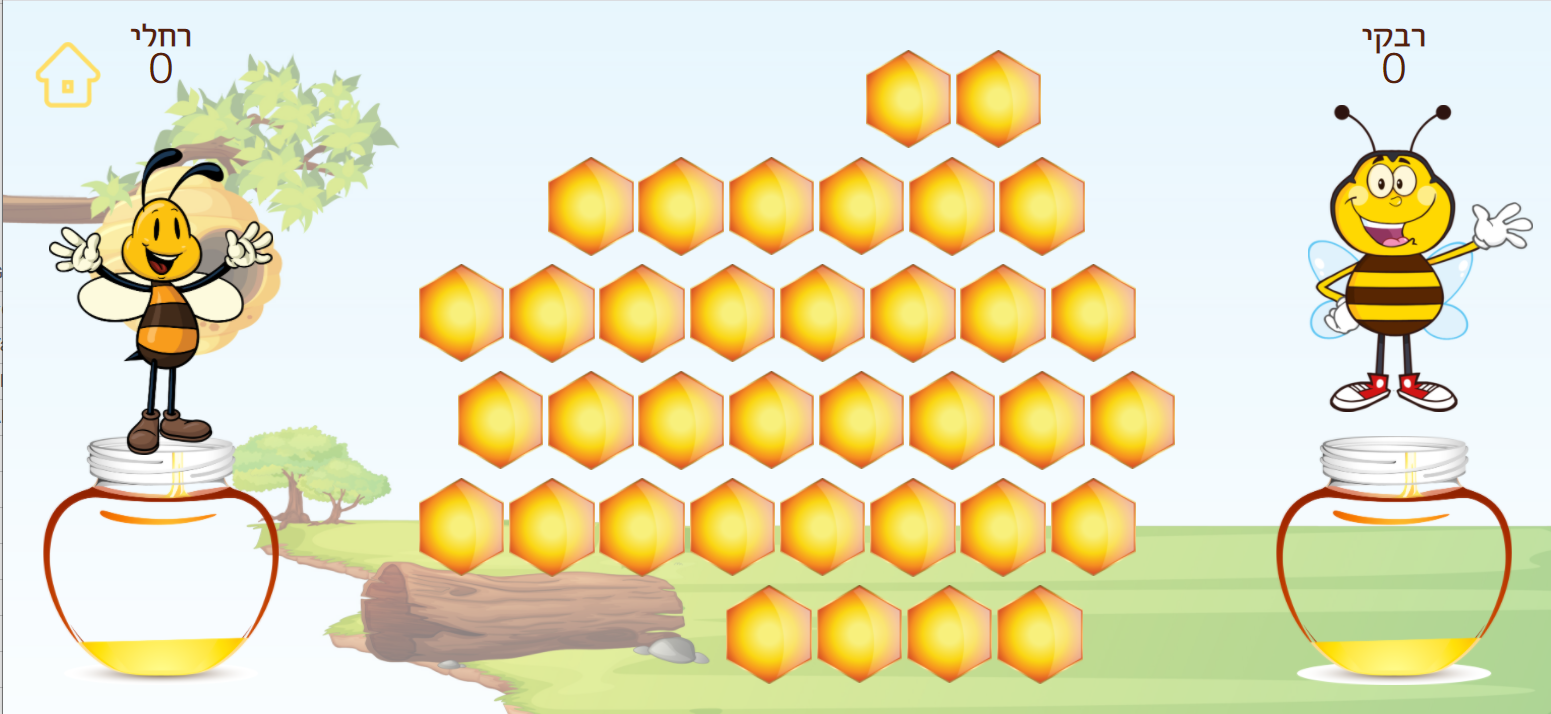
#### מסך כניסה:

#### מסך הוראות:

#### מסך אודות:



#### 3.2.3 מסך המשחק:



#### 3.2.4 מסך הגדרת שאלות:

#### 

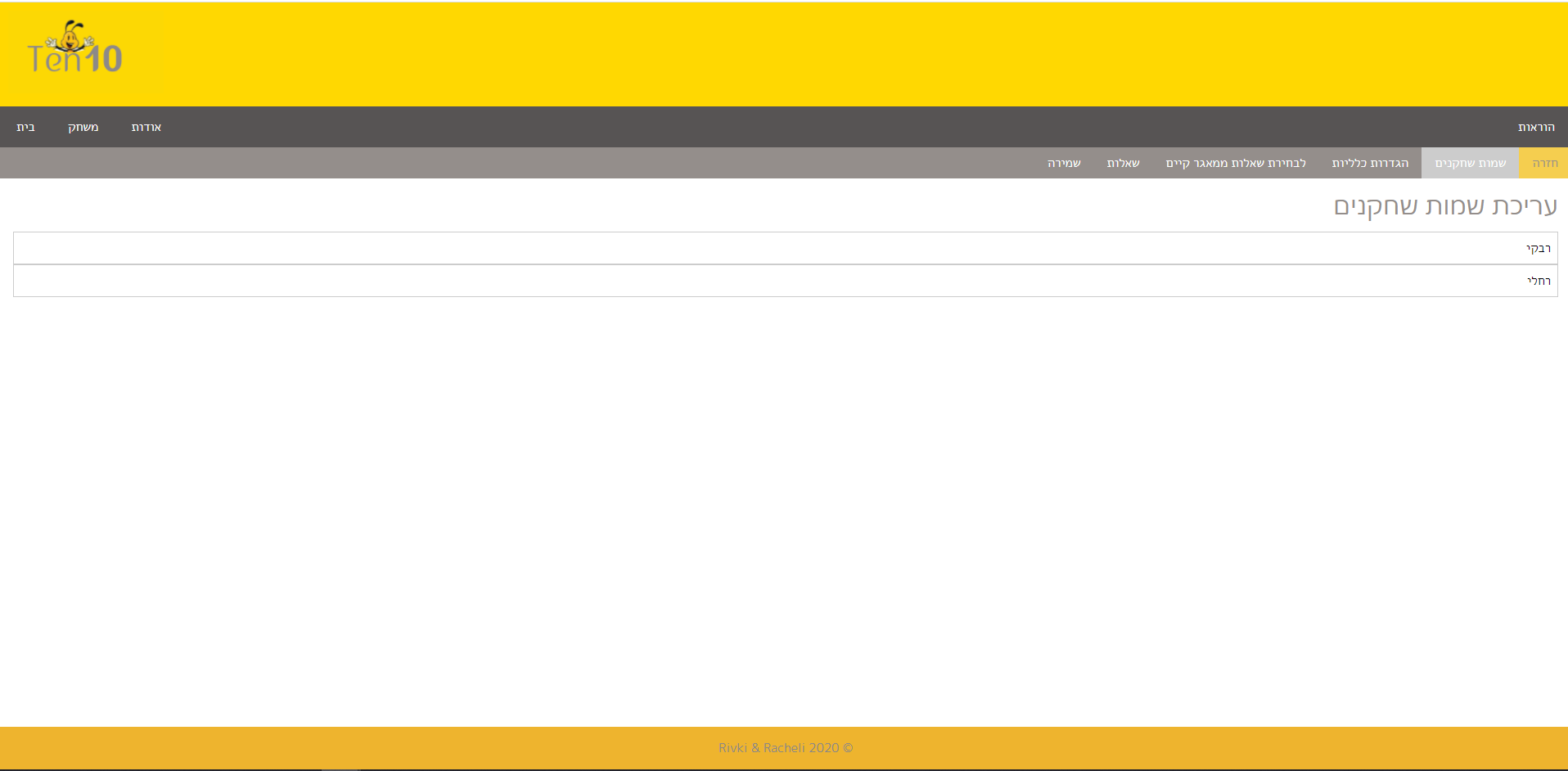
#### 3.2.5 מסך הגדרות כלליות:



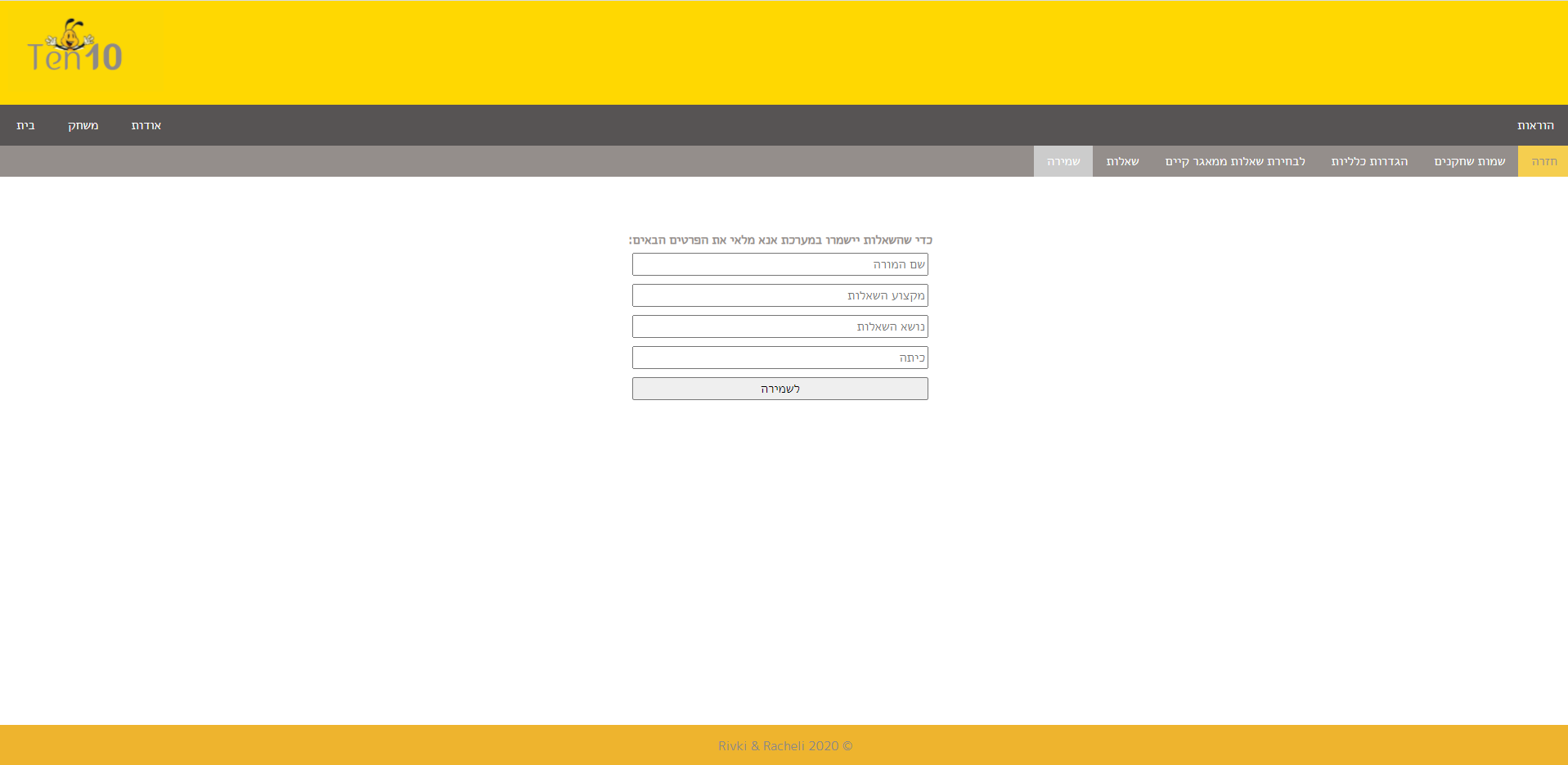
#### 3.2.6 מסך בחירת שאלות ממאגר:

#### 

#### מסך הגדרת שמות השחקנים:



* + 1. מסך שמירת שאלון:



1. סיכום ומסקנות:

בפיתוח פרויקט זה היו מספר מטרות:

* לעזור למורים ולתלמידים ללמוד ולהכיר את עולם התכנות בדרך חווייתית ונוחה,
* להעמיק ידע ושליטה ב js וב c#, כמו גם ההתנהלות עם web api.

פיתוח פרויקט זה ערך במשך ארבעה חודשים שבמהלכן נתקלנו בבעיות שונות והיה עלינו למצוא פתרונות חכמים כדי לתת להם את המענה הטוב ביותר.

לאחר עבודה רבה הגענו לתוצר המוגמר שענה לכל הדרישות וזכה לתגובות נלהבות.

מה הקנה לנו הפרויקט?

* למידה עצמית של נושאים רבים בהיקף נרחב מתוך אתרי אינטרנט.
* התמודדות עם תכנון מסדי נתונים.
* תרגול בניתוח מערכות יעיל.
* חשיבה יצירתית ומקורית לפתרון בעיות.
* עמידה בלוחות זמנים.

## נספחים:

אתרים בנושא תכנות:

1. stackoverflow.com
2. w3schools.com
3. codepen.com

## 6.ביבליוגרפיה:

אתרים בנושא תכנות:

* stackoverflow.com
* w3schools.com

### 

משתמש יכול להכניס שמות שחקנים

משתמש יוכל לבחור שאלות ממאגר קיים בחיפוש עפ"י נושא

משתמש יוכל להכניס שאלות למשחק

משתמש יכול להכניס מהירות זמן לעניית תשובה נכונה