**Shark**

**Cloud Project**

**מגישים :**

**שם: ת.ז:**

דימה צ'ודנובסקי 322175894

דוד אסולין 205908569

בר מלכה 313393324

אור בנימין 208605535

נטלי נניקשוילי 208744094

דינה קקיאשוילי 318296936

**Design Thinking:**

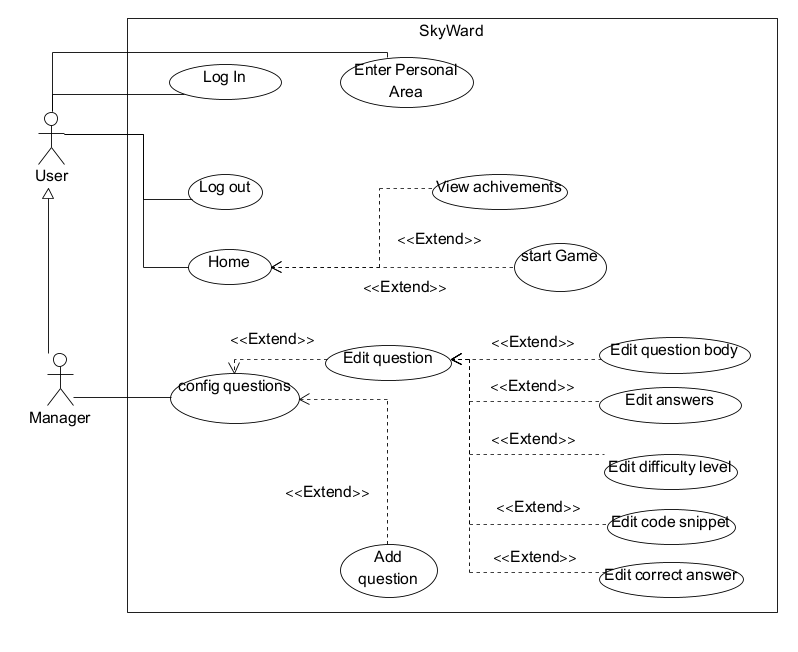
1. שם האתר: **SkyWard**  
   המערכת תהיה משחק שבו יהיה ניתן לענות על שאלות בנושא קוד Python. למשחק תהיה גישה עם שם משתמש וסיסמא כאשר לכל משתמש יישמרו הניקוד, הרמה וההישגים שלו. למשתמש האפשרות להתקדם ברמה במשחק ולעלות שלב. בנוסף ניתן לפתוח ולהשיג הישגים שונים ולעלות בטבלת הניקוד של המשתמשים.

דרישות פונקציונליות:

1. המערכת תאפשר צפייה בנקודות.
2. המערכת תאפשר צבירה של נקודות.
3. המערכת תאפשר התקדמות ברמות המשחק.
4. המערכת תאפשר התחברות לאזור אישי.
5. המערכת תאפשר זכייה בהישגים שונים במשחק.

דרישות לא פונקציונליות:

1. **אבטחה-** המערכת צריכה להיות מאובטחת, להגן על המידע של המשתמש ולמנוע כניסה של משתמשים לא מורשים. בנוסף המערכת צריכה למנוע זליגת מידע של משתמש אחד למשתמש אחר.
2. **תאימות-** המערכת צריכה להתאים לכמה סוגי מכשירים (לפחות 3) ומערכות הפעלה (windows/linux/mac).
3. **ביצועים**- המערכת צריכה לדעת להתמודד עם מספר גדול של משתמשים (כ 10,000 משתמשים) ומספר גדול של משחקים פעילים (כ 1500) ללא כל ירידה בביצועים.  
   המערכת צריכה להגיב במהירות (תוך 0.2 שניות) ולספק חווית משתמש טובה.
4. **תחזוקה**- המערכת צריכה להיות קלה לתחזוקה ועדכון
5. **זמינות**- המערכת חייבת להיות זמינה בעת הצורך.
6. תרשים Use Case:



**איטרציות עבודה-SDP:**

**עבור איטרציה 1:**

|  |  |  |  |
| --- | --- | --- | --- |
| Executor | Task | Requirement | Iteration |
| Frontend developer – David | Show Profile screen, including score and achievements | Start the application | 1 |
| Backend developer – Dima | Fetch the questions from the database and save them locally |
| QA – Natalie | Implement user authentication |
| Backend developer – Dima | Handle adding questions | Save questions and answers in the database |
| Handle adding answers |
| Scrum master-Bar | Implement a description screen for the game | Start the game |
| Frontend developer – David +  UI developer - Dina | Implement a screen that shows the instructions of the game |
|  |
| Show a massage to the player after he answers each question, that informs him that his answer is correct/wrong | Implement the game |
| Design and implement a screen board that displays a question and four optional answers |
| Show final score of the player | End the game |
| Display a graph that summarizes the game |
| QA – Natalie | Create an acceptance test | Test the application |
| Scrum master-Bar | Search for information about python to write more complex questions | information gathering |
| *product manager-Or* | Search for existing games to see how things work there and have some inspiration. |

**עבור איטרציה 2:**

|  |  |  |  |
| --- | --- | --- | --- |
| Executor | Task | Requirement | Iteration |
| Backend developer-Natalie | Create a database of users in firebase | Build and manage a database for users | 2 |
| Backend developer-Natalie | Handle adding game sessions to the database |
| Backend developer-Natalie | Add users to the database |
| UI-David | Plan a register screen |
| Backend developer-Natalie  Frontend developer-Dina | Integrate with frontend developer to check the functionality of the API that adds users |
| QA-Or | Add admin data to the database |
| Product manager-Bar | Implement user authentication |
| Backend developer-Natalie | Handle adding questions from admin interface | Add admin interface to the game |
| Handle editing questions from admin interface |
| UI-David | Plan an admin interface for adding and editing questions |
| Frontend developer-Dina | Implement a screen for the operations of admin |
| QA-Or | Perform acceptance tests for the added functionality | Test the new functionality |

**הפרויקט המלא:**

1. **SUS:**

לאחר ביצוע הסטודיו ולאחר קבלת המשוב הציון והציון הממוצע שקיבלנו הוא:

**A picture containing text, screenshot, number, parallel

Description automatically generated**

1. **תיק מתכנת:**

**קבצים מרכזיים:**

1) קובץ MAIN\_HTML – קובץ זה מכיל את המבנה של המסך וכן את המסך ההתחלתי שבו נתחיל את המשחק. בנוסף קובץ זה מכיל את החיבורים והקישורים הבסיסיים שדרושים על מנת שהמשחק יקבל את התצורה שלו (רקע, לוגו, חיבור למסד נתונים – firebase וכן קישור מיוחד שקשור להצגה של סטטיסטיקות ומודלים חישוביים – גרפים (גרף סיכום בסוף המשחק).

2) קובץ CSS\_STYLE – קובץ זה מכיל את כל העיצובים של כל האלמנטים השונים שנמצאים במסכים השונים במשחק בין אם זה עיצוב של כפתור רקע כותרת במצב רגיל ובין אם זה עיצוב של כפתור בעת לחיצה. בנוסף קובץ זה מעצב את המסכים השונים שמחלוקים בתוכם למספר אלמנטים שונים (כדוגמה: מסך המכיל 3 רכיבים בתוכו) כאשר לכל רכיב עיצוב מותאם.

3) קובץ MAIN\_JS – קובץ זה מכיל את כל הלוגיקה והפונקציונליות של המשחק. בקובץ זה נמצאת הלוגיקה שמאחורי כל לחיצה של כפתור ביצוע מענה על שאלות הוספת שאלות וכדו'. בנוסף בקובץ זה נמצא המידע לגבי הישגים שבעת משחק המשתמש יכול לפתוח ולזכות בהם.

בנוסף ישנם מספר חבילות שימושיות שאנחנו משתמשים בהם בפרויקט ולכן עושים להם import בתחילת הפרויקט והם:

-time

-requests

-IPython

-thinker

-asyncio

-matplotlib

-ipywidgets

-pyinputsplus

-firebase

-json

-os

-google.colab

**מסד נתונים:**

-במהלך הפרויקט אנו מבצעים שימוש ב firebase בתור המסד נתונים שלנו. במסד הנתונים אנו מאחסנים את כל המידע הנדרש למשחק: השאלות שמורכבות מ-גוף השאלה, קטע קוד, 4 תשובות אפשריות, אינדקס תשובה נכונה ורמת קושי של השאלה. בנוסף במסד הנתונים אנחנו מאחסנים את המידע של המשתמשים הכולל: שדה מפתח, שדה המתריע האם המשתמש הוא מסוג שחקן או מנהל, הרמה שבה המשתמש נמצא, שם המשתמש והסיסמא, הניקוד וההישגים שהמשתמש זכה בהם. כתוצאה משימוש ב- firebase אנחנו נדרשים להשתמש בפקודה !pip install firebase על מנת שהמשך נוכל להחזיק חיבור למסד הנתונים ולכתוב/ לקרוא מידע ממנו.

**פונקציות מרכזיות:**

1) handleLoginButton – פונקציה זו מקבלת event והיא על הלוגיקה מאחורי לחיצה על כפתור ההתחברות. במשחק שלנו כפתור ההתחברות מייצג שני מקרים: מקרה ראשון שבו המשתמש מתחבר וכן הכפתור משמש ככפתור התחברות, ומצב שני שבו המשתמש רוצה להתנתק ולוחץ עליו לכן במקרה זה הכפתור משמש כהתנתקות. פונקציה זו אחראית על מקרים אלה ועל המעבר וההתאמה בין שני המצבים.

2) handleLoginView – פונקציה זו מקבלת event והיא זו שאחראית על מבנה מסך ההתחברות. בפונקציה זו מוגדרים כל האלמנטים ההכרחיים כגון מקום להכנסת שם משתמש וסיסמא וכפתור התחברות על מנת שמסך ההתחברות יהווה כמסך התחברות.

3) handleuserLogin – פונקציה זו אחראית על התחברות המשתמש. בפונקציה זו תהיה השוואה בין שם המשתמש והסיסמא שהוכנסו ע"י המשתמש לבין אלה שנמצאים במסד הנתונים על מנת לבצע בדיקת תקינות קלט. במידה ונמצאה השוואה מתאימה יוחזר המבנה של המשתמש ובמידה ולא יוחזר null.

4) handleMenuButtonClick – פונקציה זו מקבלת event והיא אחראית על תפקוד כפתורי התפריט. לדוגמה במקרה שנלחץ כפתור אחד מכפתורי הבית נמצא את אותו כפתור ונבצע נעילה כך שלא תוכל להתבצע לחיצה נוספת.

5) fetchQuestionsFromDB – פונקציה זו מקבלת רמת קושי שנבחרה ע"י המשתמש, שולפת מהמסד הנתונים את כל השאלות המתאימות לרמת קושי שנבחרה, מבצעת עירבוב של כל השאלות שהוצאו ולאחר מכן לוקחת את ה-7 שאלות הראשונות בערבוב (בסיבוב משחק ישנם 7 שאלות שיש לענות עליהם).

6) handleConfigView – פונקציה זו מקבלת event והיא אחראית על כל הנראות של מסך הכנסת השאלות. בפונקציה זו נטפל בכל האלמנטים השונים הדרושים להכנסת שאלה (גוף שאלה, קטע קוד, 4 תשובות נכונות, אינדקס שאלה נכונה, רמת קושי וכפתור submit). בפונקציה זו נכניס את כל הקלטים שהוכנסו ע"י המשתמש לאובייקט על מנת שנוכל לבדוק אותם בעת לחיצה על כפתור ה-submit.

7) handleAddQuestion – פונקציה זו מקבלת event והיא ההמשך של הפונקציה מסעיף 6.בפונקציה זו נבצע בדיקה שכל הנתונים שהוכנסו ע"י המשתמש במסך הכנסת השאלה הם נתונים תקינים כגון בדיקה של חסר גוף שאלה 4 תשובות הוכנסו, סומן האינדקס של השאלה הנכונה ונבחרה רמת קושי. במידה והבדיקה עברה נבצע הכנסה של השאלה למסד נתונים כאשר נצמיד לה question\_id מיוחד שנקבע אוטומטית בפונקציה.

8) checkInputs – פונקציה זו אחראית על בדיקה של תקינות הקלט שהוכנס ע"י המשתמש. למעשה נשתמש בפונקציה זו בתוך הפונקציה מסעיף 7 על מנת להכניס שאלה למסד נתונים.

9) clearImputs – פונקציה זו אחראית על ניקוי כל השדות שמולאו ע"י המשתמש בעת הכנסת שאלה חדשה למסד נתונים. השדות יימחקו רק כאשר השאלה הוכנסה בהצלחה למסד נתונים ולאחר מספר שניות מהלחיצה על כפתור ה-submit.

10) handleGameInfoView – פונקציה זו מקבלת event והיא אחראית על המסך לפני התחלת המשחק. בפונקציה זו נטפל בהסבר על המשחק וכן נציג אפשרות של בחירת רמת קושי. רק לאחר בחירת רמת קושי יהיה ניתן להתחיל במשחק. בנוסף המשתמש יכול לבחור רק את רמת הקושי שמתאימה לשלב שבו הוא נמצא או רמת קושי מתחת לרמה הנוכחית שלו.

11) checkChooseDifficulty – פונקציה זו מקבלת event והיא אחראית על הבדיקה שנבחרה רמת קושי לפני לחיצה על כפתור התחלת המשחק. במידה ולא נבחרה רמת קושי תוצג הודעת שגיאה המתריעה על כך.

12) handlePlayView – פונקציה זו מקבלת event והיא אחראית על כל המבנה של מסך המשחק. בפונקציה זו מתאימים את הכפתורים השונים (דילוג על שאלה חזרה אחרונה לשאלה קודמת) הצגת נתוני השאלה (מבנה השאלה, קטע הקוד של השאלה ו-4 תשובות אפשריות) ובנוסף מופיע גם התקדמות של המשתמש במהלך המענה על השאלות במשחק.

13) displayProgressNode – פונקציה זו מקבל אינדקס ומטפלת בכל המשבצות במסך המשחק שמהוות אינדיקציה על התשובות שענינו נכונה לא נכונה או שדילגנו.

14) handleGameOver – פונקציה זו אחראית על המסך סיכום משחק לאחר שהמשתמש סיים לענות/ לעבור על כל השאלות במשחק. במסך זה יופיע גרף סיכום של התשובות הנכונות מול הלא נכונות שהמשתמש ענה במהלך סבב המשחק הנוכחי.

15) handleUserUpdate – פונקציה זו אחראית על עידכון מבנה המשתמש על השאלות שהוא ענה נכון/לא נכון. בפונקציה זו מבצעים את חישוב בניקוד שמקבל המשתמש עבור תשובות נכונות וכן הישגים שהוא יכול לקבל.

16) handleAnswerClick – פונקציה זו אחראית לביצוע שינוי השאלה ברגע שהמשתמש ענה על שאלה. בפונקציה זו נבצע עידכון של המענה על השאלה אם ענה נכון או לא נכון וכן כעבור מספר שניות נעבור לשאלה הבאה.

17) trackProgress – פונקציה זו מעדכנת את המשבצות של השאלות בצבעים המתאימים במידה והמשתמש ענה נכון או לא נכון על שאלה.

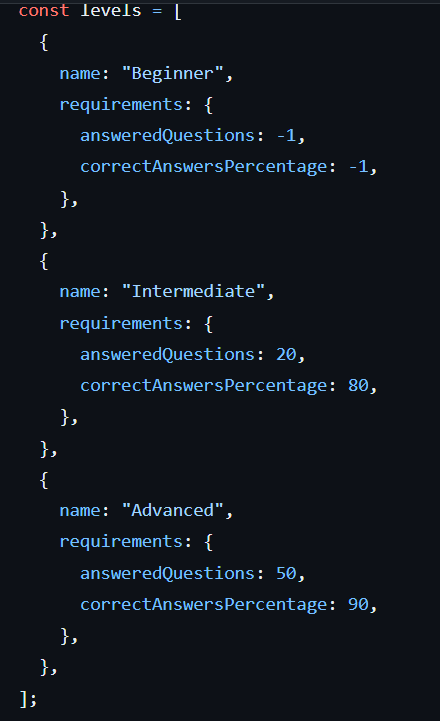
18) handleProgNode – פונקציה זו אחראית על קידום/ החזרה של הסמן על השאלה הנוכחית במשבצת של השאלות בתחתית המסך.

19) handleHomeView – פונקציה זו מקבלת event ואחראית על התצוגה של מסך הבית עם כל האלמנטים שנמצאים בה בין אם זה טבלת ההישגים של המשתמש, סטטיסטיקות מתאימות למשתמש וטבלת משתמשים מדורגים לפי ניקוד.

20) handleUserUpdate\_DB – פונקציה זו אחראית על עידכון המשתמש בתוך מסד הנתונים עם כל השינויים שבוצעו במבנה שלו.

**קטעי קוד/תבניות עיצוב:**

1. קטע קוד מרכזי ראשון שיש לנו במשחק הוא הקטע שבו נקבעים השלבים והדרגים במשחק:



כפי שניתן לראות קבענו כי במשחק ישנם 3 שלבים עיקריים:

-שלב מתחיל: שלב ראשון שבו המשתמש נמצא לפני שהתחיל

לשחק משחק ראשון. מכיוון שזהו השלב ההתחלתי אין לו

דרישות מסוימות לכן אין דרישה לכמות שאלות שנענו

ולאחוז מענה על תשובה נכונה גדול.

-שלב ביניים: שלב שני שבו המשתמש יכול להיות. משתמש

יגיע למצב זה רק לאחר שענה על 20 שאלות וכאשר יש לו

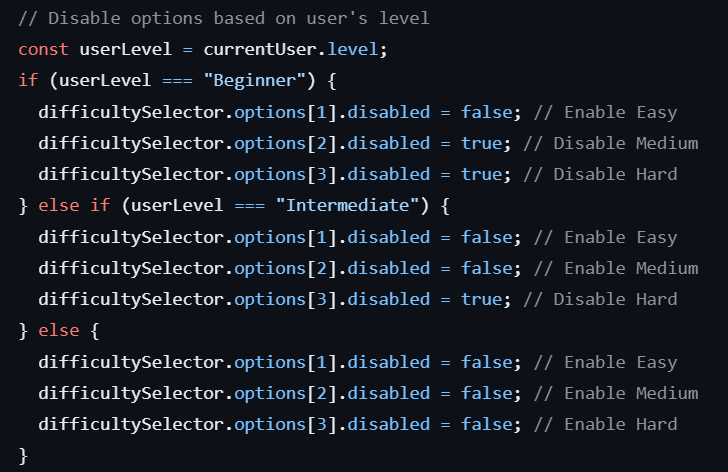
80 אחוז מענה נכונה על השאלות.

-שלב מתקדם: שהו השלב הגבוה ביותר והסופי שבוא אפשר

להיות במשחק. משתמש יתקדם לשלב זה לאחר שענה על

50 שאלות כאשר יש לו 90 אחוז מענה נכונה על השאלות.

קוד מרכזי נוסף שיש לנו הוא בהמשך למצוין בסעיף הראשון. בחלק קוד זה המשתמש נמצא במסך טרום משחק שבו הוא צריך לבחור רמת שאלות שיופיעו לו במשחק. בחלק קוד זה אנו מתאימים את אפשרויות הבחירה לרמת המשתמש. כלומר משתמש ברמה הראשונה לא יוכל לבחור רמת קושי גבוהה יותר מרמת הקושי ההתחלתית. משתמש ברמת ביניים יוכל לבחור שאלות מרמת קושי קלה או בינונית ואילו משתמש ברמה הגבוהה ביותר יוכל לבחור רמת קושי מבין 3 הרמות האפשריות. בצורה כזו אנחנו מבטיחים כי משתמש לא יוכל לענות על שאלות שלא מותאמות לרמה שלו.



1. **תיק משתמש:**

A picture containing screenshot

Description automatically generated

המסך הראשון- מסך LOGIN:

במסך זה על המשתמש להזין את הusername שלו ואת הpassword שלו וללחוץ LOGIN.

A screenshot of a computer

Description automatically generated with medium confidence

לאחר ההתחברות, מופיע מסך הבית של המשתמש.  
מסך הבית כולל הישגים של המשתמש, כמו הנקודות שיש לו, השלב בו הוא נמצא ומה היעד שהוא צריך להגיע אליו כדי לעבור לשלב הבא.  
בנוסף, מציג טבלה של השחקנים המובילים במשחק ואת הדירוג שלהם.

A picture containing text, screenshot, multimedia software

Description automatically generated

המשתמש לחץ על PLAY! והגיע למסך הראשוני של המשחק.

במסך זה יש הסבר על המשחק ועל מטרתו ואפשרות לבחור את רמת הקושי.

לאחר מכן, המשתמש ילחץ על start game להתחלת המשחק.

A screenshot of a computer

Description automatically generated with medium confidenceתמונה שמכילה טקסט, צילום מסך, תוכנה, תכונות מולטימדיה

התיאור נוצר באופן אוטומטיתמונה שמכילה טקסט, צילום מסך, מולטימדיה, תכונות מולטימדיה

התיאור נוצר באופן אוטומטי

לאחר בחירת הרמה ולחיצה על start game נפתח למשתמש מסך המשחק: שאלה ו-4 תשובות אפשריות.

לאחר מענה על השאלה, התשובה שנבחרה תצבע באדום אם היא לא נכונה ובירוק אם היא נכונה.

A screenshot of a computer

Description automatically generated with medium confidence

בתום המשחק, יוצג לשחקן גרף המתאר את כמות השאלות בהן צדק ואת כמות השאלות בהן טעה במהלך המשחק.

A screenshot of a computer

Description automatically generated

בתום המשחק, יוכל השחקן לחזור לעמוד הבית שלו ולראות את ההישגים והניקוד שלו מתעדכן.

למשל: בעת ההתחברות הניקוד של השחקן היה 56 נקודות, ולאחר המשחק הניקוד עלה ל59 נקודות.

**תוספת למשתמש מסוג ADMIN:**

A screenshot of a computer login form

Description automatically generated with medium confidence

מסך הוספת שאלות.

מסך זה כולל אופציה של הוספת תוכן לשאלה, הוספת קוד, 4 תשובות אפשריות, רמת קושי ובחירת התשובה הנכונה.

A screenshot of a computer

Description automatically generated

מסך עדכון שאלות.

מסך זה כולל אופציה של בחירת שאלה ועריכה שלה.

1. **קישורים:**

קישור לגיט: [GitHub](https://github.com/DinaKakiashvili/test)

קישור ל Firebase: [FireBase](https://trivia-game-6a531-default-rtdb.firebaseio.com/)

**אתגרים שנתקלנו בהם:**

1) האתגר המרכזי שבו נתקלנו הסמסטר היה עם סביבת העבודה ב- Google colab. הדבר התאפיין בכך שלא יכלנו לעבוד במקביל בקובץ המשותף.

2) עבודה עם קבצים חיצוניים שאיתם אנו עובדים ב colab הייתה כרוכה בזמני המתנה על מנת שהקבצים יתעדכנו ויעבדו בצורה הנכונה עם הקוד ב colab.

3) חוסר טכניקה ב text formatting כמו שקיים בסביבות עבודה אחרות, דבר שגרר לדוגמה עריכה פיזית בקוד של שורה שורה.

4) לא היה ניתן להשתמש בהרחבות שיכלו להקל על העבודה שלנו כגון: prettier ב VS Code שמקבל על הכתיבה של הקוד.

5) בכל פעם שהיינו צריכים להפעיל את הקוד ולהשתמש בחבילות מיוחדות הדבר היה כרוך בזמן המתנה שהחבילות יותקנו. הדבר נכון כאשר כל פעם היינו פותחים את הקובץ מחדש והיה צריך לבצע את השלבים מההתחלה.