**תרגיל בית 3 חלק ב׳ מחשוב ענן**

*מגישים:* **Salamandras’s group**

*שיר כהן ת.ז. 316216340*

*דוד זבונריוב ת.ז. 206440901*

*גל פרפל ת.ז. 206459562*

*עמית מלכה ת.ז. 324130483*

*יותם אהרון ת.ז. 208608448*

*אלדד שניאור ת.ז. 206696833*

**קישור ל-GIT הקבצים נמצאים בברנץ' TCCv2:**

<https://github.com/cohensh96/CloudComputing/tree/TCCv2>

*חלק ראשון: עבודה קבוצתית*

**קבוצה בודקת: Salamandra**

**קבוצה נבדקת: bee**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tests | Test ID | Test Description | Expected Result | Test Result | Pre-Condition |
| 1 | Link to trivia game valid. | Click on “Start Game” button. | Move to “Trivia Game” page. | SUCCSSES | The user is logged in. |
| 2 | Link to “About us” valid. |  |  | NO-MATCH |  |
| 3 | Link to “Our games” valid. |  |  | NO-MATCH |  |
| 4 | Link to “Leaderboard” valid. | Click on “Top Scorer” table tab. | Leader-board table is shown to the user. | SUCCSSES | The user is logged in.  The user finished playing the game. |
| 5 | Link to “Home” valid. | Click on “Home Screen” button. | Move to “Home Screen” page. | SUCCSSES | The user is logged in.  The user finished playing the game. |
| 6 | Link to “Login” valid. | User Enter username. User Enters Password.  User Clicks on “Login” button. | Move to “Home” page. | SUCCSSES | None. |
| 7 | Link to “My profile” valid. |  |  | NO-MATCH |  |
| 8 | Link to “Logout” valid. | Click on “Logout” button. | The user is redirected to “Login screen”. | SUCCSSES | The user is Logged in. |
| 9 | Login successful. | User enters “shir”.  User enter password “123”.  User clicks on “Login” button. | The user is redirected to “Home” page. | SUCCSSES | None. |
| 10 | Login unsuccessful – username is incorrect, and password is correct. | User enters “shirrrrrrr”.  User enter password “123”.  User clicks on “Login” button. | Pop up with login error. | SUCCSSES | None. |
| 11 | Login unsuccessful- username is correct, and password is incorrect. | User enters “shir”.  User enter password “123456789”.  User clicks on “Login” button. | Pop up with login error. | SUCCSSES | None. |
| 12 | Login unsuccessful- username is incorrect, and password is incorrect. | User enters “shirrrrr”.  User enter password “123456789”.  User clicks on “Login” button. | Pop up with login error. | SUCCSSES | None. |
| 13 | Logout successfully. | Click on “Logout” button. | The user is redirected to the “Login” page. | SUCCSSES | The user is logged in. |
| 14 | Display user’s nickname successfully. | The user logs in. | User’s nickname is shown in the “Home” page. | SUCCSSES | None. |
| 15 | update questions successful test. | User clicks on a question from the combo-box.  User changes the fields of the question as he wishes.  User clicks on “Update Question”. | The question is updated in the DB. | SUCCSSES | None. |
| 16 | Filter questions test successfully. |  |  | NO-MATCH |  |
| 17 | Deleting question from the DB successfully. | User clicks on a question from the combo-box.  User clicks on “Remove from Database”. | The question is removed from the DB. | SUCCSSES | None. |
| 18 | Show My profile screen successfully. |  |  | NO-MATCH |  |
| 19 | Adding question successfully. | User fills in all question fields.  User clicks on “Add to Database!”. | The question is added to the DB. | SUCCSSES | None. |
| 20 | Adding question unsuccessfully. | User does not fill all question fields.  User clicks on “Add to Database!”. | The message is shown to the user: “Can’t Add Empty Field in Add Question Form!!” | SUCCSSES | None. |
| 21 | Show Manager screen successfully. | Run admin Cell in the Google Colab. | Show  “Manager” page. | SUCCSSES | None. |
| 22 | Leaderboard successful test. | Click on “Top Scorer” table tab. | Leader-board table is shown to the user. | SUCCSSES | The user is logged in.  The user finished playing the game. |
| 23 | Answer question correct successfully. | Click on correct answer option button. | The button color changes to green. | SUCCSSES | The user has entered the game. |
| 24 | Answer question wrong successfully. | Click on incorrect answer option button. | The button color changes to red. | SUCCSSES | The user has entered the game. |
| 25 | moving to next question successful. | Click on some answer option. | The answer button changes its color accordingly.  User is redirected to the next question. | SUCCSSES | The user has entered the game. |
| 26 | Finish the game successfully. | Click on some answer option in the fifth question. | The user is redirected to the “Statistics” page. | SUCCSSES | The user has entered the game. |
| 27 | Show correct answer successfully. | Click on some answer option. | The correct answer button changes its color to green. | SUCCSSES | The user has entered the game. |
| 28 | Exit game successfully. |  |  | NO-MATCH |  |
| 29 | Register successful test. | User enters “Gal”. User enters his “123”.  User re-enters his password – “123”. | Pop up window showing the message: “Sign Up Successful”. | SUCCSSES | The username does not exist in the DB. |
| 30 | Username already taken successful test. | User enters “shir” as username. User enters his password.  User re-enters his password. | Pop up window showing the message: “Error Sign Up”. | SUCCSSES | The username already exists in the DB. |
| 31 | Nickname already taken successful test. |  |  | NO-MATCH |  |
| 32 | Registered page link to login page successfully. | User enters his username – “John”. User enters his password “tj67”.  User re-enters his password - “tj67”. | The user is redirected to the “Login” page. | SUCCSSES | The username does not exist in the DB. |
| 33 | Statistics-showing successfully. | Click on some answer option in the fifth question. | The user is redirected to the “Statistics” page. | SUCCSSES | The user is logged in.  The user has entered the game. |
| 34 | Statistics With a new highscore record successful test. |  |  | NO-MATCH |  |
| 35 | display all questions on the screen - GET operation successful test. | Run admin Cell in the Google Colab. | Combo-box contains all the questions from the DB. | SUCCSSES | None. |
| 36 | Link button to its page successful test. |  |  | NO-MATCH |  |
| 37 | Button\_add\_a\_question | User fills in all question fields.  User clicks on “Add to Database!”. | The question is added to the DB. | SUCCSSES | None. |
| 38 | Button\_show\_all\_questions |  |  | NO-MATCH |  |
| 39 | Display\_all\_questions. | Run admin Cell in the Google Colab. | Combo-box contains all the questions from the DB. | SUCCSSES | None. |
| 40 | Error 404 page successful. |  |  | NO-MATCH |  |
| 41 | Statistics-showing successfully | Click on some answer option in the fifth question. | The user is redirected to the “Statistics” page. | SUCCSSES | The user is logged in.  The user has entered the game. |
| 42 | Confetti shown successfully |  |  | NO-MATCH |  |
| 43 | Popup show correct answer button successfully |  |  | NO-MATCH |  |
| 44 | Popup show "Next question” button successfully |  |  | NO-MATCH |  |
| 45 | Popup show "Good job! +2" successfully. |  |  | NO-MATCH |  |
| 46 | Popup show "wrong answer" successfully. |  |  | NO-MATCH |  |
| 47 | Display all the sessions successfully. | User plays a full game. | A progress graph with the user’s match history is displayed. | SUCCSSES | The user is logged in.  User plays full game at least thrice. |
| 48 | Display session‘s “view statistics” successfully |  |  | NO-MATCH |  |
| 49 | Saving all the sessions the user successfully. | User plays a full game. | A progress graph with the user’s match history is displayed. | SUCCSSES | The user is logged in.  User plays full game at least thrice. |
| 50 | The user enters to the combo box on top right and clicks on the "My profile"  button and scroll down to the “Match History” section |  |  | NO-MATCH |  |

*חלק שני: עבודה אישית*

כל סטודנט יבצע code review לקוד של המחברת שהתקבלה, לפי הפרמטרים הבאים  
 (מבוסס על :<https://www.michaelagreiler.com/code-review-checklist-2>)

נא לענות על כל קריטריון *ולפרט* :

שם הסטודנט: עמית מלכה

|  |  |
| --- | --- |
| קריטריון | הסבר |
| מימוש | האם הקוד מבצע את הנדרש?  כן לאחר הבדיקות שעשינו בסעיף הקבוצתי כל הדרישות בהם הקבוצה נדרשה לעמוד בוצעו בהצלחה. |
| יעילות | האם ניתן לשפר את יעילות הקוד?  קוד יעיל פשוט וקריא |
| פשטות | האם ניתן לפשט את הקוד?  הקוד היה ברור לקריאה לא נתקלתי בקשיים בהבנת הקוד (בעיקר עקב התיעוד הטוב של חברי הקבוצה) |
| מודולריות | האם הקוד מודולרי מספיק? (חלוקה תקינה לקבצים, פונקציות ומחלקות)  יש חלוקה לכמה cells בנוסף יש שימוש ב2 קבצי html ו js .בתוך הcells הפונקציות מחולקות בצורה טובה |
| תבניות עיצוב | האם ישנו מימוש ברור ונכון של תבניות עיצוב?  אין כל כך שימוש בתבניות עיצוב בקוד |
| OOP | האם ישנו שימוש נכון בעקרונות OOP? האם ניתן לשפר?  אין שימוש מספיק טוב בעקרונות OOP כיוון שאין שימוש במתודות פרטיות לכן הקוד פתוח בפני כל המשתמשים אין היררכיה בין דברים בצורה טובה אין שימוש בירושה ובכלל אין כמעט שימוש בעקרונות אלו כן אולי היה כדאי להשתמש בירושה על מנת ליצור יישויות שונות כמו מנהל ויוזר פשוט |
| באגים וטעויות | האם ישנם מקרים בהם הקוד לא מתנהג כצפוי? אילו?  אני באופן אישי לא מצאתי מקומות בהן הקוד לא מתנהג כצפוי |
| טיפול בשגיאות | האם ישנה התראה על שגיאות משתמש?האם ישנו שימוש במנגנון תפיסת שגיאות?האם הבדיקה ודיבוג הקוד נוחים?האם הודעות המשתמש ברורות וידידותיות למשתמש?  ישנה התראה טובה על טעויות משתמש כמו שגיאה בשם המשתמש או בסיסמה כמו גם בטיפול מבחינת הקוד |
| בדיקות | האם יש בדיקות שניתן להוסיף?  כל הפונקציונליות נבדקה מכל הכיוונים בצורה טובה לכן לדעתי אין עוד בדיקות שצריך להוסיף |
| שימושיות | האם הממשק שימושי? (usability)? האם המשחק מעורר רצון להמשיך לשחק?  כיוון שיש כמות קטנה מאוד של שאלות (7 בלבד) המשחק חוזר על עצמו באופן קבוע מה שלא מעורר אצלי רצון להמשיך לשחק מעבר ל2-3 משחקים מנגד ניתן לראות שימושיות בקוד בכך שלא ניתן לראות סטטיסטיקה לפני שאסיים 3 משחקים לכן ש מוטיבציה לשחק |
| תיעוד | האם הממשק מתועד?  כן הממשק מתועד כהלכה (כל פונקציה בפני עצמה) והערות ביניים חשובות |
| אתיקה | האם יש שימוש בנתוני משתמש אשר מפרים פרטיותו?האם יש ניצול של התנהגות או חולשה אנושית?האם ישנו algorithmic bias כנגד קבוצה או פרטים מסוימים?  מבחינה אתית הקוד עומד בנדרש אין דבר הפוגע בפרטיותו או זכויותיו של אף אדם |
| אבטחה | האם ישנו מידע אבטחה (שמות משתמש, ססמאות) גלויים?  כל המידע מאוחסן בDB בצורה שלא גלויה למשתמש |
| ביצועים | האם לדעתך שינוי עתידי בקוד יכול לגרום לפגיעה בביצועים?  שינוי עתידי לא אמור לפגום בביצועי הקוד |
| ביצועים | האם ניתן לשפר לדעתך את הביצועים?  תמיד ניתן לשפר ביצועים אך עם זאת הקוד רץ בצורה מהירה וחלקה ולכן אין צורך לשפר ביצועים |
| קריאות - readability | האם הקוד מובן בקלות?אילו חלקים בקוד היו לא ברורים עבורך?כיצד ניתן לשפר את קריאות הקוד?  הקוד קל להבנה וברור לגמרי ויש שימוש טוב שמות הפונקציות ובתיעוד שעוזר להבין מה כל דבר בקוד אמור לעשות |
| סעיף פתוח  **עיצוב** | האם התוצאה מעוצבת בצורה המושכת את העין ?  לטעמי כן התוצאה יחסית פשוטה מבחינת נראות ואין יותר מידי משחק של צבעים הייתי אולי משפר את הסטטיסטיקה לצבע שונה כדי שיהיה קל לראות את הגרף בצורה קלה |
| סעיף פתוח  משחקיות | קיימת משחקיות בקוד הן על ידי הleaderboard והן על ידי הhighscore של כל משתמש באשר הוא משתמש הייתי אולי מוסיף אפקטים על שבירת שיא סטטיסטיקה נוספת שלי ביחס לאחרים על מנת לעודד משחקיות |

**שאלות פתוחות:**

**1.כיצד הרגשת במהלך החוויה של מעבר על הקוד?**

הרגיש בסדר קצת מאתגר לעבור על קוד של מישהו אחר ולהבין עד הסוף עצם העובדה שאני יודע פחות או יותר איך אמורה להראות התוצאה עזר לי להתחבר לקוד בצורה טובה יותר ולהבין בקלות יותר "למה התכוון המשורר"

**2.האם לדעתך ניתן לשפר תרגיל זה?**

כן היה ניתן לשפר את התרגיל בכך שהייתם נותנים לנו את הקישורים הנדרשים על מנת לבדוק ולא היינו צריכים להתחיל לחפש מי שייך לאיזה קבוצה ולבקש מנו את הקוד שלו

**3. הערות נוספות**

אין לי הערות נוספות.

**שם הסטודנט: שיר כהן**

|  |  |
| --- | --- |
| קריטריון | הסבר |
| מימוש | האם הקוד מבצע את הנדרש?  לאחר מעבר על הקוד נראה כי הקוד מיישם את כל הפונקציונליות הנדרשת. המשתמש יכול להירשם, להתחבר, לשחק במשחק(כולל מתן חיווי למשתמש אם ענה נכון או לא ובמידת הצורך מראה מהי התשובה הנכונה), לעקוב אחר התוצאות ע״י הצגה של גרף לינארי בסוף המשחק. בנוסף בצד המנהל המשתמש יכול לערוך, להוסיף ולמחוק שאלות. |
| יעילות | האם ניתן לשפר את יעילות הקוד?  לדעתי אין לקוד בעיות יעילות משמעותיות, הקוד מובנה ומאורגן היטב. עם זאת, על מנת לשפר את יעילות הקוד הייתי יוצרת קבצי CSS ,JS , HTML נפרדים במקום לשלב בין כולם בתוך קוד הPython חלוקה זאת תקטין את גודל הקובץ ע״י הורדת תווים מיותרים מה שעשוי לשפר את מהירות טעינת הדף.  בנוסף, במקום להשתמש ב callback functionsניתן להשתמש בטעינה אסינכרונית) ( asynchronous loading זה עשוי לגרום לטעינה מהירה יותר ולגרום לקוד להיות קריא יותר. |
| פשטות | האם ניתן לפשט את הקוד?  ניתן לפשט את הקוד ע״ יצירת קבצים נפרדים עבור קוד ה CSS וה-JavaScript במקום להטמיע אותם ישירות בקוד Python זה יקל על עדכון, ניהול ומעקב אחר העיצוב והסקריפט.  הקוד קל ופשוט להבנה , יש שימוש בשמות ברורים למשתנים ופונקציות (למשל הפונקציה check\_signup בודקת האם הרישום תקין האם שם משתמש פנוי והאם הסיסמא והאימות שלה זהים ובמידת הצורך מוציאה הודעה מתאימה. ניתן לקבל אינדקציה שזה מה שהפונקציה עושה לפי שמה) וחלוקה לפונקציות נפרדות שכל אחת אחראית למשימה ספציפית. לדוגמה, ישנן פונקציות לאחזור נתונים ממסד הנתונים, טיפול בהתחברות או הרשמה וכו׳, כל אלו הופכים את הקוד לקריא יותר. |
| מודולריות | האם הקוד מודולרי מספיק? (חלוקה תקינה לקבצים, פונקציות ומחלקות)  ישנה חלוקה לפונקציות נפרדות שכל אחת מהן אחראית למשימה מסוימת, חלוקה זאת משפרת את ארגון הקוד ומקלה על ההבנה והתחזוקה שלו. בנוסף יש חלוקה לחלקים בקוד שניתן לראות את הקשר ביניהם מה שמקל על התמצאות בקוד.  למשל:  פונקציות loginToggle, check\_signup ו-check\_login מטפלות בפונקציונליות הכניסה וההרשמה. קיימת הפרדה בין טיפול באימות משתמשים ,בדיקת שגיאות והצגת מודלים מתאימים.  פונקציות showAnswer, showAnswerExample ו-endGame מטפלות באספקטים שונים של הצגת תשובות, משוב וסיום המשחק. הם מטמיעים את הלוגיקה לעיבוד רכיבי ממשק משתמש ומתן משוב למשתמשים על סמך הבחירות של המשתמש. |
| תבניות עיצוב | האם ישנו מימוש ברור ונכון של תבניות עיצוב?  לא ראיתי בקוד שימוש בתבניות עיצוב, אני חושבת שהקוד פשוט ומובן לכן אינני רואה צורך להוסיף. עם זאת, כדי למנוע תקלות ויצירת מספר מופעי גישה למסד הנתונים ניתן להשתמש בתבנית עיצוב singleton (ניתן לראות בקוד כי הם יצרו מופע אחד של גישה למסד הנתונים וגם הגישה אליו מתבצעת פעם אחת אך זה לא מימוש של singleton כי הגישה לא מוגבלת לפעם אחת). |
| OOP | האם ישנו שימוש נכון בעקרונות OOP? האם ניתן לשפר?  הקוד אינו משתמש בעקרונות OOP. בקוד אין שימוש במחלקות או אובייקטים כדי לכלול נתונים והתנהגות ולאפשר שימוש חוזר באובייקטים המקיימים אינטראקציה זה עם זה.  ניתן להוסיף כמה מחלקות ,לשפר את האינקפסולציה של הנתונים, וליצור פונקציונליות נוספת המקשרת בין המחלקות, למשל ניתן ליצור מחלקת User אשר תייצג את המשתמש במשחק , תשמור את פרטיו ותכיל מתודות שקשורות לאינטרציות שונות איתו. ניתן ליצור מחלקת Game שתייצג את הלוגיקה של המשחק ואת הזרימה, מחלקת Question שתייצג את השאלות, במחלה זאת ישמרו פרטי השאלה ומתודות שונות ועוד.  הוספת מחלקות אלו יכולה לשפר את יכולת התחזוקה וההרחבה של הקוד. |
| באגים וטעויות | האם ישנם מקרים בהם הקוד לא מתנהג כצפוי? אילו?  לא, הקוד מתנהג כצפוי , המבנה הלוגי ותפעול המשחק עובדים בדיוק כפי שחשבתי שיעבדו בכל שלב ובכל לחיצת כפתור. אך יש מקום אחד שהקוד עבר לא כצפוי, לאחר ביצוע הרשמה על מנת להתחבר למערכת יש לטעון את התא בcolab מחדש על מנת שנוכל להתחבר למערכת עם המשתמש החדש שיצרנו, מצב זה עלול לבלב את המשתמשים ולגרום להם לחשוב שההרשמה לא בוצעה בהצלחה על אף שהיא כן. |
| טיפול בשגיאות | האם ישנה התראה על שגיאות משתמש?האם ישנו שימוש במנגנון תפיסת שגיאות?האם הבדיקה ודיבוג הקוד נוחים?האם הודעות המשתמש ברורות וידידותיות למשתמש?  כן,יש התראה על שגיאות שונות, המשתמש מקבל פידבק בכל פעולה שהוא מבצע, למשל אם בעת ההתחברות המשתמש הזין שם משתמש או סיסמא לא נכונים יקפוץ לו popupp עם הודעת שגיאה שהשם משתמש או הסיסמא לא נכונים, כשהמשתמש מבצע רישום הוא מקבל הודעה שהרישום הצליח או אם יש בעיה באחד מהפרטים שהמשתמש הזין בטופס הרישום תוצג למשתמש הודעה מתאימה וכו׳.  הודעות השגיאה שמוצגות למשתמש ברורות ואינפורמטיביות ועוזרות לו להבין ולפתור בעיות.  הדיבוג בcolab קצת קשה לדעתי במיוחד כשרוב הקוד רשום בcell אחד, לכן על מנת להבין את הקוד לעומק נעזרתי בהדפסות לקונסול. |
| בדיקות | האם יש בדיקות שניתן להוסיף?  הקוד כולל בדיקות קלט מרובות(למשל: הזנת שם משתמש וסיסמא שגויים בשלב ההתחברות והצגת הודעת שגיאה מתאימה למשתמש) אך הוא אינו כולל בדיקות המבטיחות את אמינות הקוד ובכדי להבטיח את אמינות הקוד מומלץ ליישם בדיקות שיסייעו לזהות ולתקן שגיאות ובאגים בקלות ובמהירות (למשל : בדיקות שתופסות חריגות). |
| שימושיות | האם הממשק שימושי? (usability)? האם המשחק מעורר רצון להמשיך לשחק?  ממשק המשתמש ידידותי , הUI מאוד יפה לדעתי, יש פידבק כמעט על כל פעולה של המשתמש(למשל, לחיצה על תשובה לא נכונה – הכפתור יצבע באדום ותוצג גם התשובה הנכונה בירוקת לחיצה על תשובה נכונה- הכפתור יצבע בירוק).  משתמשים יכולים להתחבר או להירשם, להתחיל משחק, לענות על שאלות ולראות את התוצאות וההתקדמות שלהם. הקוד משתמש במודלים כדי להציג הודעות ומידע למשתמשים, מה שמקל עליהם להבין את זרימת המשחק ולספק קלט מתאים במידת הצורך.  כל האלמנטים הללו מגדילים את המוטיבציה והרצון לשחק במשחק. |
| תיעוד | האם הממשק מתועד?  הקוד מתועד ברמה מספקת לדעתי, עבור כל פונקציה רשום מהי היא עושה מה שנותן הבנה כללית טובה על המערכת. |
| אתיקה | האם יש שימוש בנתוני משתמש אשר מפרים פרטיותו?האם יש ניצול של התנהגות או חולשה אנושית?האם ישנו algorithmic bias כנגד קבוצה או פרטים מסוימים?  הקוד הגון ואתי, לא ראיתי שימוש בנתוני המשתמש אשר מפרים את פרטיותו. |
| אבטחה | האם ישנו מידע אבטחה (שמות משתמש, ססמאות) גלויים?  שמות המשתמשים והסיסמאות אינם גלויים, הגישה למידע רגיש כמו למידע במסד הנתונים מאובטח הוא לא עובר הרבה סינונים מה שמגביר את ההגנה על מידע רגיש זה. |
| ביצועים | האם לדעתך שינוי עתידי בקוד יכול לגרום לפגיעה בביצועים?  לדעתי לא, הקוד מאורגן ומסודר היטב , לכן שינוי בקוד לא אמור להיות מסובך מידי או לגרום לפגיעה בביצועים. הגישה למסד הנתונים קלה ומאפשרת לאחסן ולסנכרן נתונים בזמן אמת בקלות, לכן הוספת פיצ'רים חדשים שדורשים שינוי במסד הנתונים לא ישפיעו על הביצועים באופן משמעותי. |
| ביצועים | האם ניתן לשפר לדעתך את הביצועים?  הקוד רץ מהר ועובד באופן חלק לכן, לדעתי אין צורך לשפר את הביצועים, עם זאת תמיד אפשר לנסות לשפר אותם ע״י צמצום חישוביות וזמן ריצה, הוספת פיצ'רים שונים שישדרגו את המערכת וייחדו אותה ,שימוש בתבניות עיצוב שיוסיפו למערכת גמישות ותחזוקה ועוד. |
| קריאות - readability | האם הקוד מובן בקלות? אילו חלקים בקוד היו לא ברורים עבורך? כיצד ניתן לשפר את קריאות הקוד?  הקוד קריא ומובן יחסית בקלות, תיעודי הפונקציות מספקות הבנה כללית טובה של הקוד ומאפשרות התמצאות פשוטה בו.  עם זאת , ניתן לפשט את הקוד ע״ יצירת קבצים נפרדים עבור קוד ה CSS וה-JavaScript במקום להטמיע אותם ישירות בקוד Python זה יקל על עדכון, ניהול ומעקב אחר העיצוב והסקריפט ויגדיל משמעותית את קריאות הקוד.  או שניתן לחלק את הקוד ליותר cells כאשר כל cell יהיה אחראי על פונקציונליות מסוימת באפליקציה (במקום שכל הפונקציונאליות של האפליקציה תהיה באותו הcell ), זה יקל על עדכון, ניהול ומעקב אחר הקוד ויגדיל משמעותית את קריאתו. |
| סעיף פתוח  **Motivation**  **and**  **Gamification** | האם קיים בקוד אלמנט משחקי שיגביר את מוטיבציית המשתמש?  הקוד מכיל אלמנטים משחקיים שונים, המשתמשים צוברים ניקוד עבור כל שאלה, יש צבע לתשובה נכונה ולתשובה שגויה, יש לוח מובילים שמדרג את השחקנים ומכניס אלמנט של תחרות בריאה בין השחקנים, כל אלו הם אלמנטים משחקיים שמגבירים ומחדירים מוטיבציה במשתמשים ובכך עשויים לגרום להגדלת תדירות השימוש במשחק. |
| סעיף פתוח  **scalability** | הוסיפו קטגוריה נוספת לבדיקה ובצעו אותה  האם קיימת מדרגיות בקוד?  שימוש במסד נתונים בענן של Firebase מאפשר הגדלה או הקטנה משאבים בהתאם לצורך בכדי לענות על צרכים משתנים.  כלומר הקוד יכול להסתגל ולהתמודד יחסית ביעילות ובקלות עם עומסי עבודה או דרישות מוגברות (כמו צמיחה במשתמשים, נתונים וכו׳). |

**שאלות פתוחות:**

***1. כיצד הרגשת במהלך החוויה של מעבר על הקוד?***

*החוויה של מעבר על הקוד הייתה מהנה וחיובית, הקוד ברור וקריא מה שמקל על הבנתו. בנוסף, המעבר על הקוד כאשר הדרישות שניתנו לנו היו זהות אך אופן המימוש היה מאוד שונה ביחס למימוש שלנו לימד וחשף בפניי כיווני חשיבה שונים ומגוונים.*

***2.האם לדעתך ניתן לשפר תרגיל זה?***

*אני חושבת שאם ה- code review היה נעשה כאשר שתי הקבוצות נמצאות פיזית ביחד זה היה מאפשר הבנה עמוקה יותר של הקוד וזה היה מוסיף להווי החברתי והופך את התרגיל למהנה יותר. בנוסף, אני חושבת ששיבוץ הקבוצות צריך לכלול מראש את שמות חברי הקבוצה, הקישור למחברת הפרויקט ואת הקישור למסד הנתונים, על מנת לאפשר תקשורת מהירה וקלה ולחסוך את הזמן שנדרש למציאת הקבוצה שעליה אנו צריכים לעשות את הבדיקה.*

***3.הערות נוספות***

*הקורס היה מלמד, מהנה וחוויתי אך אני חושבת שפלטפורמת הcolab הייתה מאתגרת ומגבילה עבור כתיבת הקוד בpython בשילוב עם* CSS ,JS , HTML , לדעתי סביבת פיתוח שונה הייתה עשויה להפחית את הזמן שהשקענו בהתאמת סביבת העבודה. עם זאת , האתגר הנ״ל גרם לי להיחשף לכלים שלא הכרתי לפני כן (כמו ngrok,Flask,Jinja ) שאפשרו לנו לעבוד בצורה נוחה וקלה יחסית.

**שם הסטודנט: יותם אהרון**

|  |  |
| --- | --- |
| קריטריון | הסבר |
| מימוש | האם הקוד מבצע את הנדרש?  כן, הקוד מבצע את הנדרש ועובד טוב. כדרוש הקוד מפעיל את המערכת ועובד בצורה חלקה, מריץ את המשחק ואת החיווי לשאלות בצורה טובה וברורה, ומציג לאחר כל משחק סטטיסטיקה של אותו שחקן בהשוואה למשחקים קודמים שלו. בנוסף, גם צד המנהל עובד כמו שצריך כאשר טעינת כלל השאלות לקומבו בוקס עובדת וניתן לערוך שאלות בהתאם ולעדכן במסד הנתונים. |
| יעילות | האם ניתן לשפר את יעילות הקוד?  לדעתי ניתן היה לשפר את יעילות הקוד בכך שהקוד של צד השחקן וצד המנהל לא פועלים באותה תיבת קוד ולכן כביכול לא רצים ביחד תחת אותה הרצה. |
| פשטות | האם ניתן לפשט את הקוד?  כן ניתן לפשט את הקוד יותר, כעת הקוד רשום בכמה תיבות קוד ב Google Collab אך אחת מהתיבות כוללת הרבה מאד קוד והוא נראה עמוס, ניתן לפשט אותו תוך שימוש בכתיבת הקוד בקבצים חיצוניים וטעינתם בתוך תיבת הקוד ב Collab וכך היה רק הקוד הבסיסי רשום והיה פשוט יותר. |
| מודולריות | האם הקוד מודולרי מספיק? (חלוקה תקינה לקבצים, פונקציות ומחלקות)  הקוד מודולרי, מחולק היטב לפונקציות ולכל פונקציה אכן יש שם משמעותי שעוזר להבין טוב מה תפקידה, בנוסף הקוד מחולק לתיבות קוד ספציפיות בקולאב שלכל תיבה יש את התפקיד שלה ( Html,JavaScript,Css ). אך עדיין כפי שציינתי למעלה הקוד של ה js הוא עמוס בקוד וכן היה אפשר לעשות שימוש בקבצים חיצוניים על מנת לפשט אותו יותר. |
| תבניות עיצוב | האם ישנו מימוש ברור ונכון של תבניות עיצוב?  בקוד לא קיים מימוש של תבנית עיצוב כלשהי, אך כן הקוד מסודר מאד ומובן. אך כדי למנוע תקלות כלשהן ביצירת מספר מופעי גישה למסד הנתונים, ניתן להשתמש בתבנית עיצוב singleton , בקוד שלהם הם יצרו מופע אחד של גישה למסד הנתונים, בנוסף הגישה למסד הנתונים גם כן קורית פעם אחת. אבל, זהו לא מימוש של תבנית עיצוב singleton ולכן הגישה לא מוגבלת בדיוק לפעם אחת. |
| OOP | האם ישנו שימוש נכון בעקרונות OOP? האם ניתן לשפר?  בקוד אין שימוש בעקרונות OOP , אין שימוש במחלקות או באובייקטים. לכן, ניתן לשפר את הקוד תוך שימוש בעקרונות OOP כאשר שימוש זה כולל יצירת מחלקות או אובייקטים אשר יכללו נתונים ופונקציות ויאפשרו שימוש חוזר בהם במקומות שונים בקוד. לדוגמא, ניתן היה ליצור מחלקת User אשר שומרת את כל נתוני המשתמש ואת כל המידע הדרוש עבור משתמש מסוים וכך להעביר מידע באמצעות אובייקט זה. אותו הדבר היה ניתן לעשות עבור Question , באותו אופן ליצור מחלקה עבור שאלה ולשמור את כל הפרטים שלה באובייקט וככה רק לגשת לשדות הרצויים. דבר זה יכול לשפר את תחזוקת הקוד ואת ההבנה שלו. |
| באגים וטעויות | האם ישנם מקרים בהם הקוד לא מתנהג כצפוי? אילו?  לפי כל הבדיקות שביצענו, ראינו כי הקוד מתנהג כצפוי ועובד כמו שצריך ולא זיהינו באגים כלשהם. |
| טיפול בשגיאות | האם ישנה התראה על שגיאות משתמש?האם ישנו שימוש במנגנון תפיסת שגיאות?האם הבדיקה ודיבוג הקוד נוחים?האם הודעות המשתמש ברורות וידידותיות למשתמש?  ישנה התראה על שגיאות המשתמש כמו למשל אם הוא מנסה להתחבר וטועה בשם המשתמש או בסיסמא קופץ לו חלון פופ אפ עם הודעה מתאימה, אותו הדבר קורה אם משתמש מנסה להירשם והשם משתמש תפוס. בנוסף גם כאשר המשתמש טועה/ צודק בשאלה במשחק הטריוויה יש חיווי בהתאם. כל זאת בצורה ידידותית ונוחה למשתמש כך שהכל ברור ומובן וגם נעים לעין. |
| בדיקות | האם יש בדיקות שניתן להוסיף?  לדעתי אין עוד בדיקות שניתן היה להוסיף, כי באמת עברנו על כל הפונקציונליות שדרושה במערכת ובדקנו מצבי כישלון ומצבי הצלחה שלהן, לכן אני לא חושב על בדיקות נוספות עבור פונקציונליות שלא נבדקה במערכת. |
| שימושיות | האם הממשק שימושי? (usability)? האם המשחק מעורר רצון להמשיך לשחק?  הממשק שימושי ומעודד רצון להמשיך לשחק, דבר ראשון אם אתה משתמש חדש אתה לא תראה את הסטטיסטיקה שלך עד שלא תעשה לפחות 3 משחקים, מה שגורם כן לרצון לעשות לפחות שלושה משחקים ולראות את רמתך. בנוסף, יש את טבלת הניקוד ששם אתה רואה את מיקומך ביחס לשחקנים אחרים מה שגם גורם לרצון להתקדם במעלה הטבלה ולשחק עוד. |
| תיעוד | האם הממשק מתועד?  רוב הממשק מתועד, וברור להבנה. כמו כן, כל הפונקציות עם שמות משמעותיים שנותנות לנו אינדיקציה על תפקידן ועל מה שהן אמורות לבצע. |
| אתיקה | האם יש שימוש בנתוני משתמש אשר מפרים פרטיותו?האם יש ניצול של התנהגות או חולשה אנושית?האם ישנו algorithmic bias כנגד קבוצה או פרטים מסוימים?  אין שימוש בנתוני המשתמש כך שמפרים את פרטיותו, כל משתמש רואה רק את הפרטים שלו ואת והשם שלו, ואת נתוני המשחק שלו. הדבר היחיד שמשתמש יכול לראות על משתמשים אחרים זה את מיקומם בטבלת המובילים, וזה לא פוגע בפרטיותם. |
| אבטחה | האם ישנו מידע אבטחה (שמות משתמש, ססמאות) גלויים?  ישנה אבטחה בכך שכאשר אנו מזינים שם משתמש אנו נראה אותו בגלוי, אך כאשר נזין את הסיסמא לא נראה אותה בגלוי אלא כצופן של נקודות כך לא ניתן לדעת מה הסיסמא שהקלדנו, בנוסף, המערכת דואגת שלא יהיו שתי משתמשים בעלי שם משתמש זהה. |
| ביצועים | האם לדעתך שינוי עתידי בקוד יכול לגרום לפגיעה בביצועים?  לדעתי שינוי עתידי בקוד יכול לגרום לפגיעה בביצועים, רק אם יעמיסו ממש על התיבת קוד בקולאב כי לפי מה שאני ראיתי העמסת יתר על תא קוד בקולאב לפעמים גורם לקריסת האפליקציה. |
| ביצועים | האם ניתן לשפר לדעתך את הביצועים?  לדעתי המערכת רצה בצורה מאד טובה וחלקה, כל לחיצה על כפתור או מעבר בין דפים קורה בצורה חלקה ומהירה ולכן הביצועים טובים, לכן לדעתי אין דרך לשפר את הביצועים יותר מכך. |
| קריאות - readability | האם הקוד מובן בקלות?אילו חלקים בקוד היו לא ברורים עבורך?כיצד ניתן לשפר את קריאות הקוד?  הקוד ברור ומובן, כפי שציינתי כל הפונקציות כתובות בשמות משמעותיים כך שקל להבין מה הן עושות לכן הקוד קריא. מה שכן אפשר לשפר, זה להוסיף עוד קצת תיעוד על הדרך שבה פועלת הפונקציה, יותר על הלוגיקה שלה, כך נוכל להבין יותר לעומק את הפונקציה. כמו כן, אם הייתה חלוקה לקבצים היה יותר קל להבין איזה קוד מתקשר למה וגם לעקוב אחרי שגיאות. |
| סעיף פתוח  **Scalability** | הוסיפו קטגוריה נוספת לבדיקה ובצעו אותה  האם הקוד פועל תחת עיקרון המדרגיות? האם יש מדרגיות בקוד?  כן, הקוד בנוי באמצעות פונקציות אשר לכל אחת יש מטרה מסוימת וניתן להרחיב את הקוד בקלות על ידי הוספת פונקציות חדשות או שינוי פונקציות קיימות. בנוסף שימוש במסד הנתונים firebase לטובת אחסון של שאלות,משתמשים. מסד נתונים זה ניתן להרחבה ויכול להתמודד עם מספר רב של משתמשים בו זמנית. |
| סעיף פתוח  **Consistency** | הוסיפו קטגוריה נוספת לבדיקה ובצעו אותה  האם הקוד כתוב באופן עקבי?  כן, הקוד כתוב במבנה קבוע ועקבי. לאורך כל הקוד אנחנו יכולים לראות שמירה על סגנון ומבנה אחיד. ניתן לראות כי לאורך כל הקוד מתבצע מתן שמות עקבי של משתנים ופונקציות לפי עיקרון ה camelCase. בנוסף ניתן לראות שהקוד מאורגן בצורה טובה ועקבית כלומר, ישנו קיבוץ של קוד הקשור אחד לשני ביחד מה שגורם לסדר והקלה בקריאת ובהבנת הקוד. גם מבחינת העיצוב והזחות הקוד עקבי וניתן לראות שמעברי השורות, מרווחים ורמות ההזחה עקביות מה שהופך את הקוד למאורגן ויזואלית וקל להבנה. |

**שאלות פתוחות:**

1. ***כיצד הרגשת במהלך החוויה של מעבר על הקוד?***

*במהלך המעבר על הקוד חשתי עניין רב מכיוון שמאד הסתקרנתי לראות איך קבוצה אחרת פעלה, ואיך הקבוצה השנייה ניגשה לכתוב את הקוד וגם מאד רציתי לראות ולהיפתח לצורות חשיבה שונות וצורת כתיבה שונות.  
בנוסף, במעבר על הקוד רציתי לראות האם אני מצליח להבין במעבר ראשוני אם אני מבין את כל מה שרשום בקוד ואת כל מה שהם עשו.  
לכן, התיעודים ושמות הפונקציות מאד עזרו לי להבין על מה אני מסתכל ומה כל פונקציה מבצעת, וכך עוד לפני שקראתי את תוכן הפונקציה קראתי את התיעוד קודם, וניגשתי לקריאת הפונקציה עם ידע מקדים, וידעתי פחות או יותר מה הפונקציה הולכת לבצע וככה היה לי יותר קל להבין את הקוד במעבר ראשוני.*

1. ***האם לדעתך ניתן לשפר תרגיל זה?***

*לדעתי כן ניתן לשפר תרגיל זה, כאשר אנו נדרשים למצוא את הקבוצה הנבדקת אפשר להוסיף את שמות חברי הקבוצה שנדע את מי אנחנו צריכים לבדוק.*

1. ***הערות נוספות***

באופן כללי אני חושב שהקורס עזר לי להתפתח ולהבין יותר לעומק את כל החשיבה בכתיבת קודים בשפות: Html, JavaScript, Css, Python .  
כמו כן, העבודה עזרה לי להתפתח ולהבין יותר בתחום מחשוב ענן ואפילו בנושאים רבים אחרים אשר למדתי תוך כדי העבודה וכתיבת הקוד ( Flask, ngrok ) , בנוסף העבודה בצוות הייתה טובה מאד ומועילה וגם ההצגות בכיתה היו מהנות וכיפיות ועזרו בסופו של יום להבין את החומר בצורה יותר טובה ומעמיקה.

**שם הסטודנט: דוד זבונריוב**

|  |  |
| --- | --- |
| **קריטריון** | **הסבר** |
| מימוש | האם הקוד מבצע את הנדרש?  כן, הקוד מבצע את כל מה שביקשו במטלות. המימוש מלא, לפי מה שאני ראיתי מהבקשות שלכם יש במימוש את כל מה שצריך. |
| יעילות | האם ניתן לשפר את יעילות הקוד?  יש כמה אספקטים שהיית משפר מבחינת אופטימיזציה, לדוגמא: בחלק מהקוד משתמשים ב-var, ובחלק מהקוד ב-const,let , אז לא מומלץ להשתמש בvar, זה לא משהו שפוגע בקוד אבל מבחינת עקביות עדיף שלא.  המלצה אחת ליעילות קצת יותר טובה של הקוד זה להשתמש ב: async במקום ב-callback functions, דבר זה מקל על הקריאה, ובכללי מאפשר בשימוש של await, לקוד א-סינכרוני, callback זה מאוד כבד על הקוד וגורם לו להיות יותר איטי. |
| פשטות | האם ניתן לפשט את הקוד?  אפשר להשתמש בכל מיני טכניקות של JS על מנת לפשט על הקוד ולעשות אותו יותר קל לקריאה, לדוגמא להשתמש ב- Object.values() על מנת להכניס את המידע למערך, זה מונע את ההשמה הידנית למערך. בנוסף לזאת, חלק מה-inline styles היו לא הכרחיים, זה מאוד עוזר לפשטות הקוד, אפשר גם סטייל שמשתמשים בו הרבה כגון צבע שמשתמשים בו הרבה להכניס לתוך משתנה, שאפשר פשוט לכתוב אותו וזהו, זה גם מאוד משפר את קריאת הקוד. |
| מודולריות | האם הקוד מודולרי מספיק? (חלוקה תקינה לקבצים, פונקציות ומחלקות)  למרות שרוב הקוד כתוב בcell אחד, עדיין ניתן לראות חלוקה לפונקציות, לsections שקשורים זה לזה, ולא קשה לחפש דברים בקוד. אבל עדיין, cell אחד מאוד קשה לקריאה, אבל בgoogle colab אי אפשר כל כך להימנע מזה אפשר רוצים קוד רציף ואחיד. ולמרות זאת, לא התקשיתי בקריאת הקוד, היה לי מאוד מסודר. |
| תבניות עיצוב | האם ישנו מימוש ברור ונכון של תבניות עיצוב?  לפי מה שאני רואה אפשר לראות שימוש אולי אפילו לא במכוון של וריאציה של singleton, שבו אנחנו רואים instance אחד של גישה לDB, אבל זה עדיין לא מספיק בשביל לקרוא לזה singleton בגלל שזה לא מוגבל לגישה אחת, פה ספציפית לא פונים עוד פעם לבקשה לפתיחת גישה לDB.  בנוסף לזה אפשר לראות Factory pattern בפונקציית getFromDataBase, שהיא כמו מפעל שממנו אנחנו מבקשים מידע והוא מחזיר לנו את המידע באובייקט. |
| OOP | האם ישנו שימוש נכון בעקרונות OOP? האם ניתן לשפר?  לא, אבל ניתן להתייחס להערות הבאות: אפשר לשפר את האינקפסולציה של הקוד, כמו שנאמר המודולריות הייתה טובה, למרות המגבלות של ה-google colab הקוד כן היה מודולרי בשבילי לפחות. אפשר לעשות את הקוד יותר אבסטרקטי, לדוגמא להכין קלאס של Question, שבו אנחנו שומרים את כל המידע של השאלה, ואז הגישה והנראות לגישה לשאלות יותר נאה ומסודר. זה לא הפריע לי בכלל, וזה קשה לשמור על כתיבת שמות אחידים כשיש קבוצה גדולה של אנשים שרושמים את הקוד הזה, במיוחד כשהוא כולו כתוב ב-cell אחד. |
| באגים וטעויות | האם ישנם מקרים בהם הקוד לא מתנהג כצפוי? אילו?  אני לא ראיתי מקומות בקוד שלא עובדים כמו שהgui מראה. אבל יש מקומות שאולי הקוד יכול במצב מסוים לעבוד לא כצפוי, וזה כאשר משתמשים בfirebase. בגלל ש-firebase הוא NoSQL, הוא לא מכריח תבנית מסוימת להכנסה ל-collection, מסיבה זאת, אם בקוד עצמו לא בודקים אם השלד של השאלה מלא, אם השאלה לא מלאה והיא תוגרל במשחק עצמו יכולות להיות נפילות בקוד. |
| טיפול בשגיאות | האם ישנה התראה על שגיאות משתמש? האם ישנו שימוש במנגנון תפיסת שגיאות? האם הבדיקה ודיבוג הקוד נוחים? האם הודעות המשתמש ברורות וידידותיות למשתמש?  לפי מה שאני רואה בקוד, ישנם התראות כאשר משתמש לא ממלא פרטים מסוימים, או לא שם סיסמא נכונה או שם משתמש נכון, או כאשר הוא כן נכנס למערכת זה נותן פידבק. ההודעות משתמש כן ברורות, קל להבין מה לעשות כאשר יש הודעת משתמש, והן מאוד ידידותיות. ניסיתי לדבג קצת את הקוד, אני לא ממש מבין איך לדבג קוד של google colab אז השתמשתי בדיבוג שהוא bruteforce, על ידי שימוש בprinting ולוגים לconsole. הדיבוג סיפק לי הרבה מידע והבנה על מבנה הקוד. |
| בדיקות | האם יש בדיקות שניתן להוסיף?  אפשר בכללי להוסיף בדיקות של exceptions שיכולים להיגרם אבל זה לא הכרחי. אולי אפשר להוסיף כל מיני הערות של תחילת משחק, סוף משחק, מעבר בין שאלות, וכו' בגלל שהwidgets, שימוש בcolab, וגישה למסד נתונים הוא שילוב מאוד איטי, אז אולי לשים הערות כאלה יכול לתת הרגשה שהמערכת עובדת יותר מהר. |
| שימושיות | האם הממשק שימושי? (usability)? האם המשחק מעורר רצון להמשיך לשחק?  כן, אני מאוד אוהב את הצבעים, לא משעמם לשחק, הUI מאוד יפה, ישנם שאלות מאתגרות, ויש אינטראקציה בין המשתמש לממשק. |
| תיעוד | האם הממשק מתועד? יש תיעוד של הקוד, לדעתי התיעוד הוא מספק, בחלקים שאולי לא היו מובנים של הקוד היה להם תיעוד מה שהקל בסופו של דבר על ההבנה של הקוד.  על כל פונקציה עיקרית יש תיעוד של מה התפקיד שלה ומה היא עושה, לכן לדעתי התיעוד מתאר יפה את המערכת ומסביר עליה טוב. |
| אתיקה | האם יש שימוש בנתוני משתמש אשר מפרים פרטיותו?האם יש ניצול של התנהגות או חולשה אנושית?האם ישנו algorithmic bias כנגד קבוצה או פרטים מסוימים?  לא ראיתי סממנים לדברים כאלו בקוד. |
| אבטחה | האם ישנו מידע אבטחה (שמות משתמש, ססמאות) גלויים?  גם לזה לא שמתי לב, לפי מה שאני רואה הם ניגשים למסד נתונים שלהם בדרך נכונה, לא נותנים למידע הרגיש לעבור הרבה פילטרים ולכן לפי דעתי המידע הוא יחסית מוגן. |
| ביצועים | האם לדעתך שינוי עתידי בקוד יכול לגרום לפגיעה בביצועים? לדעתי לא, הקוד יחסית גנרי, זאת אומרת קל להכניס כל מיני פיצ'רים חדשים אליו, והגישה למסד נתונים בנויה כך שמאוד קל להכניס נתונים חדשים, לכן אני לא חשוב ששינוי עתידי בקוד יכול לגרום לפגיעה בביצועים. |
| ביצועים | האם ניתן לשפר לדעתך את הביצועים?  כפי שנשאל בשאלות קודמות, לשפר ביצועים ברמת האופטימיזציה אפשר לבצע, אפשר לנסות לגשת למסד נתונים בדרך יעילה יותר, לעבד את הנתונים החוזרים מהמסד נתונים בדרך מהירה יותר, אפשר להשתמש בtag elements יותר סמנטיים למרות שזה לא משנה שום דבר, זה פשוט קל יותר לקריאה. אפשר לאחסן דברים בlocalstorage בשביל גישה מהירה, לייעל אלגוריתמים מסוימים. |
| קריאות - readability | האם הקוד מובן בקלות?אילו חלקים בקוד היו לא ברורים עבורך?כיצד ניתן לשפר את קריאות הקוד?  כפי שכתבתי גם קודם, הקוד עצמו הוא מתועד, קל להבנה, קל לחיפוש, אבל זה כן מקשה כשכל הלוגיקה וכל בניית האפליקציה נמצאת במקום אחד, במיוחד כאשר אין לי מספור שורות, אבל לא היו לי בקוד חלקים שאינם ברורים, והקוד מובן בקלות. את קריאת הקוד ניתן לשפר בזה שאפשר לחלק אותו למקטעים, וכל המקטעים להכניס לגוגל דרייב בתור שמות של חלקים מהאפליקציה, וכאשר אני ארצה לקרוא חלק מסוים אני אגש לתיקיה זו, אבחר באותו חלק שיהיה לו שם משמעותי ואז יהיה לי מאוד קל למצוא דברים כי זה כבר ממוקד הרבה יותר. |
| **Security and Data Privacy** | Are authorization and authentication handled correctly?  יש פה חלק של אוטוריזציה, אבל הוא לא גדול, לכן יכול לקרות מצב שמאוד קל לפרוץ למשתמש אחר, או שמשתמש אחר יקבל גישות למקומות שהוא לא מורשה אליהם. אבל ראיתי בקוד כי כן עוקבים אחר משתמש מחובר, כן יש בדיקת התחברות תקינה, אבל חוץ מזה אין עוד משהו. |
| **Readability** | Would more comments make the code more understandable?  לדעתי התיעוד בקוד הוא מספק, גם תיעוד רב מידי יכול לגרום לזה שקורא את הקוד להתייאש ממספר הרב של תיעוד שהוא צריך לקרוא, רוב הפעמים מספיק להסביר בשורה אחת על בלוק של קוד, הרבה מאוד תיעוד גם יכול לפגוע ביכולת הקריאה של הקוד. |

**שאלות פתוחות:**

1. ***כיצד הרגשת במהלך החוויה של מעבר על הקוד?****במהלך המעבר על הקוד ראיתי שהקוד מאוד זרם, היה לי קל להתמצא בכל מיני דברים שחשבתי בהתחלה שיהיה לי קשה למצוא.  
   כשרק התחלתי לעבור על הקוד הייתה לי קצת תחושת תסכול כי הקוד מאוד מאוד ארוך, הרבה מאוד פונקציות, הרבה מאוד מלל, אבל לאחר זמן מועט הבנתי שהקוד זורם, שהוא מאוד מסודר ומתועד, והתחלתי להבין איך האפליקציה נראית תוך כדי קריאת הקוד, הצלחתי לדבג את הקוד עם גישה נאיבית של פרינטים בקונסול, וכאשר עשיתי זאת גיליתי שהתוצאות שקיוויתי שיצאו באמת יצאו, כתוצאה מזה הרגשתי שאני מבין את הקוד שרשום, וגם חלקים שלא כל כך הבנתי, או חלקים שהיו חדשים עבורי היו מתועדים ברמה גבוהה, קראתי את התיעוד ולרוב הבנתי, אבל כאשר לא הבנתי ניגשתי לאינטרנט ומהר מאוד מצאתי את התשובות לתהיות שלי.**נהניתי לעבור על הקוד, נהניתי לחשוב על דברים שהייתי יכול לשפר, לייעל, נהניתי למצוא דרכים לשבור את הקוד, ועל איך לתקן את אותם שבירות.  
   היה לי חוויה יפה להשוות בין הפרויקט שלהם לבין הפרויקט שלנו, הרי את שלנו עשינו בשיטה שונה לחלוטין, אז מאוד עניין אותי כיצד הם היו יכולים לעשות את המעבר לטכנולוגיה שאנחנו השתמשנו בה וגיליתי שזה בכלל לא קשה.  
   בסך הכול מאוד נהניתי בחוויה של מעבר על הקוד.*
2. ***האם לדעתך ניתן לשפר תרגיל זה?***

*דרך אחת שהייתי משפר תרגיל זה היא לתת טבלה עם פחות שאלות מוכנות, ולתת לסטודנטים לחפש במאמר יותר דברים שמתאימים לפרויקט שהם עוברים עליו ולענות על זה, דבר זה יגרום להם לקרוא יותר את המאמר ואולי להבין יותר את המהות והחשיבות של מעבר על הקוד.*

1. ***הערות נוספות***

אין לי כל כך הערות נוספות חוץ מזה שאני הרגשתי שהקורס עצמו מועבר בצורה טיפה עמוסה, הרבה מאוד דברים לעשות ומהר, אני מאוד שמחתי ללמוד הרבה דברים חדשים, ונהניתי מרוב המטלות, אבל לפעמים היו פעמים שהרגשתי שאני חייב לעשות את זה ולא רוצה, שלי אישית זה טיפה הפריע. בנוסף לזאת אני חושב שה-2 שאלות ששואלים בסוף ההצגה וזה 10 אחוז מהציון הסופי לא היו מתוכננות כראוי, אני הרגשתי לחץ מטורף כשאני פתרתי את השאלות כי אחרי כל ההצגות נשארו בדיוק 7 דקות עד שהשיעור הבא מתחיל, למרות שלדעתי הצלחתי בשאלות כי אני למדתי את החומר ממש הרבה והגעתי לכל הרצאה ותרגול והשתתפתי והתעמקתי בכל הקורס, אני מרגיש שהלחץ גבר עליי, בנוסף לזה שיש לי כתב מאוד גדול ולא הכי קריא, אז אני מנסה לכתוב הכי לאט שאני יכול, לא היה לי כל כך הרבה מקום לכתוב תשובות כמו שאני הייתי רוצה לכתוב אותן.

**שם הסטודנט: גל פרפל**

|  |  |
| --- | --- |
| **קריטריון** | **הסבר** |
| מימוש | האם הקוד מבצע את הנדרש?  הקוד מבצע את הנדרש ובהחלט עונה על הדרישות הפונקציונליות שהוגדרו במטלה. האפליקציה תומכת בהרשמה עבור משתמשים חדשים, התחברות עבור משתמשים רשומים, שיחוק במשחק עבור משתמשים מחוברים, הצגת סטטיסטיקות משחקים של משתמש והתנתקות. עבור המנהל המערכת תומכת בפעולות |
| יעילות | האם ניתן לשפר את יעילות הקוד?  ניתן לשפר מעט את יעילות הקוד באופן הבא: במשיכת כל המשתמשים ממסד הנתונים בקובץ index.js ניתן למשוך רק את שם המשתמש והסיסמה שלו ולשמור אותם בתוך Hash-Map ובכך לשפר את יעילות פונקציה בדיקת פרטי המשתמש במהלך ה-login. |
| פשטות | האם ניתן לפשט את הקוד?  ניתן לפשט את הקוד על ידי יצירת קבצי HTML עבור כל מיני אלמנטים הנטענים באופן דינאמי מתוך index.js ולבצע קריאה מהקבצים הללו במקום להשתמש במחרזות HTML ארוכה בתוך הקובץ עצמו. שיטה זו תקצר משמעותית את אורך הקובץ index.js ותוסיף ל-readability של הקובץ. |
| מודולריות | האם הקוד מודולרי מספיק? (חלוקה תקינה לקבצים, פונקציות ומחלקות)  בהינתן הכלים שנלמדו בהרצאה לתכנות ב-Google Colab הקוד מודולרי ברמה מסויימת, אך בכל זאת ניתן לשפר את החלוקה שלו מעט. ישנה חלוקה בסיסית של הקוד למספר תאים בקולאב, כלומר מספר סקריפטים שרצים אך ביניהם ישנו תא אחד – index.js שבו שמורה הלוגיקה של המשחק עצמו עם טעינת המשתמשים והשאלות, בדיקת תקינות פרטי המשתמש בביצוע התחברות, בדיקת פרטי המשתמש בביצוע הרשמה, הצגת דף הבית והצגת הגרפים. הייתי מציע לפרק את הקוד למספר דפי HTML ולמספר סקריפטים של JS כך שתהיה הפרדה יותר ברורה בין הלוגיקה וההצגה של המשחק, ההרשמה, ההתחברות ובניית הגרפים והסטטיסטיקות. כמו כן, ניתן להפריד בין מחרוזות ה-HTML הנטענות על ידי העברתן לקבצים נפרדים, בשיטה זו גודלו של קובץ index.js קטן משמעותית.  בנוסף, ההפרדה בין קבצי דף המנהל לבין קבצי דף המשתמש הרגיל, מעולה. |
| תבניות עיצוב | האם ישנו מימוש ברור ונכון של תבניות עיצוב?  אין שימוש בתבניות עיצוב. |
| OOP | האם ישנו שימוש נכון בעקרונות OOP? האם ניתן לשפר?  לא בדיוק. ניתן להוסיף מודול של פייתון בתא נפרד המכיל את כל פעולות שמבוצעות עם מסד הנתונים, הנקראות Crud Operations, פעולות אלה כוללות הוספה, עדכון, מחיקה וקריאה של נתונים מה-DB. במודול המוצע להלן, ניתן היה להוסיף מחלקה כזו של תקשורת עם ה-DB שמאותחלת בתחילת הקבצים שבהן יש תקשורת עם מסד הנתונים ובצורה כזו לבצע את כל פעולות הקריאה, עדכון, מחיקה והוספה למסד הנתונים בצורה גנרית וכללית יותר. הצעה זו מהווה דוגמה ל-Encapsulation של החיבור למסד הנתונים יחד עם הפעולות המבוצעות אל מול מסד הנתונים. |
| באגים וטעויות | האם ישנם מקרים בהם הקוד לא מתנהג כצפוי? אילו?  אין מקרים בהם הקוד לא מתנהג כצפוי.  התוכנה נבדקה בקפדנות ופועלת כמצופה. |
| טיפול בשגיאות | האם ישנה התראה על שגיאות משתמש?האם ישנו שימוש במנגנון תפיסת שגיאות?האם הבדיקה ודיבוג הקוד נוחים?האם הודעות המשתמש ברורות וידידותיות למשתמש?  ראשית, ההודעות המוצגות למשתמש ברורות מאוד. שנית, ישנו שימוש במנגנון תפיסת שגיאות הן בביצוע התחברות שגויה והן בהתחברות עם שם משתמש קיים.  כמו כן במהלך המשחק המשתמש מקבל חיווי טבעי ופשוט על נכונות התשובה שלו, כאשר תשובה לא נכונה מסומנת באדום ואילו תשובה נכונה מסומנת בירוק. |
| בדיקות | האם יש בדיקות שניתן להוסיף?  ניתן להוסיף בדיקות התנתקות מחלונות שונים באפליקציה, ניתן להוסיף בדיקה בתהליך ההתחברות עבור הזנת שם משתמש נכון וסיסמה קיימת במערכת אך לא סיסמה מתאימה לאותו שם המשתמש. כמו כן, ניתן להוסיף בדיקות עבור חלון המנהל. אחת הבדיקות שניתן להוסיף הינה כתיבת שאלה חדשה – מילוי כל הפרטיםשלה בטופס ולחיצה על כפתור עדכון השאלה. כמו כן, ניתן להוסיף בדיקה של עדכון שאלה על ידי שינוי פרטי השאלה שמעדכנים לפרטי שאלה שכבר קיימת ב-DB. |
| שימושיות | האם הממשק שימושי? (usability)? האם המשחק מעורר רצון להמשיך לשחק?  הממשק עורר בי רצון להמשיך לשחק ממספר סיבות. הסיבה המרכזית הינה הנוחות והפשטות של האפליקציה הן מבחינת ההבנה והן מבחינת הפעולות שהמשתמש עושה .הסיבה השנייה הינה העיצוב המושקע והקליל לעין של כל חלונות האתר ביחד עם הנפשות הכפתורים והמעברים החלקים בין העמודים – אשר יוצרים חווית משתמש נעימה. יתר על כן, ישנו חלון הסבר אשר מסביר על אופן המשחק והצגת התוצאות בגרפים שעוזר מאוד למשתמש להבין אילו אפשרויות יש לו באפליקציה וכיצד לשחק בה. כמו כן, הצגת תוצאות המשתמשים האחרים מוסיפה לרצון לשחק במשחק, לשבור שיאים ולעלות במיקום בטבלת המובילים.  לעומת זאת, אני חושב שהיה אפשר להוסיף כמות גדולה של שאלות בכדי לשפר את חווית הלמידה של המשתמש והמוטיבציה של המשתמש לשחק בכמה שיותר משחקים ולהיתקל בכל פעם בשאלות חדשות. |
| תיעוד | האם הממשק מתועד?  כן, הממשק מתועד בצורה מעולה. כל קטע קוד עיקרי מתועד במידה המתאימה, עיקרי המטרות והרעיונות מאחורי קטעי הקוד מוצגים בתיעוד בצורה הולמת וברורה. |
| אתיקה | האם יש שימוש בנתוני משתמש אשר מפרים פרטיותו?האם יש ניצול של התנהגות או חולשה אנושית?האם ישנו algorithmic bias כנגדקבוצה או פרטים מסוימים?  לא. |
| אבטחה | האם ישנו מידע אבטחה (שמות משתמש, ססמאות) גלויים?  ישנה אבטחת מידע, הסיסמאות אינן גלויות. כמובן ששם השמשתמש גלוי כאשר המשתמש עצמו מתחבר למערכת ואין בכך פגיעה באבטחת המידע. |
| ביצועים | האם לדעתך שינוי עתידי בקוד יכול לגרום לפגיעה בביצועים?  כמובן שהתשובה לשאלה זו תלויה בשינוי העתידי. אין תלויות הדוקות בין קטעי קוד, כלומר אין צורך שקטע קוד מסויים יחכה לקטע קוד אחר שיסיים לרוץ למשך זמן ארוך, ולכן אני מניח ששינוי עתידי תקין בקוד לא עלול לגרום לפגיעה בביצועים. |
| ביצועים | האם ניתן לשפר לדעתך את הביצועים?  לדעתי לא ניתן ואף לא צריך לשפר את הביצועים כיוון שהאפליקציה עובדת בצורה מהירה ונוחה. אין זמני המתנה ארוכים עבור האפליקציה להצגת פידבק, הצגת תשובה נכונה, הצגת שאלות, התחברות או הרשמה. |
| קריאות - readability | האם הקוד מובן בקלות?אילו חלקים בקוד היו לא ברורים עבורך?כיצד ניתן לשפר את קריאות הקוד?  הקוד מובן בקלות בזכות התיעוד המצויין המסביר את הרעיונות הכלליים של כל קטע קוד. כמו כן, הלוגיקה של הקוד ברורה וכתובה באופן פשוט וישנם שמות משמעותיים עבור המשתנים בקוד. בזכות תכונות אלו, הקוד קל לחיפוש (על אף היותו מרוכז כולו בתא יחיד של הקולאב). |
| **נגישות - Accessibility** | האם בעלי מוגבלויות יכולים להשתמש באפליקציה?  לפי דעתי, האפליקציה עבור המשתמש הפשוט (שאינו מנהל) מונגשת באופן טבעי עבור בעלי מוגבלויות.  כל הכפתורים גדולים יחסית, הכתב גדול וברור והחלון של האפליקציה עצמה לא גדול מידי, כלומר אין צורך לגלול או לחפש אלמנטים כלשהם על המסך. כל הכפתורים נגישים מאוד והאתר פשוט ונוח מאוד לתפעול. יתר על כן, ישנו חלון הסבר אשר מסביר על אופן המשחק והצגת התוצאות בגרפים שעוזר מאוד למשתמש להבין אילו אפשרויות יש לו באפליקציה וכיצד לשחק בה.  לעומת זאת, בחלון המנהל, הכתב קטן והשורה הראשונה של ה-Combo box לא מוצגת במלואה (כלומר המשפטים אינן מוצגים במלואם). כמו כן, בהינתן כמות גדולה של שאלות, צורה זו אינה נוחה לחיפוש שאלה ספציפית ועדכון שלה שכן כל השאלות צמודות מאוד האחת לשנייה בתוך התפריט של ה-Combo box. לכן, לפי דעתי החלון של המנהל אינו מספיק נגיש עבור בעלי מוגבלויות. |
| **Separation of Concerns – הפרדת דאגות** | כפי שניתן לראות, קודם כל קיימת הפרדה בין הקבצים הקשורים לחלון המנהל והקבצים הקשורים לחלון המשתמש הפשוט (שאינו מנהל) המשחק במשחק. בנוסף, ניתן לראות כי ישנה הפרדה של הקוד הקשור למשחק עצמו, ל-3 קבצים: CSS, HTML, JS ו-3 הקבצים הללו ביחד מרכיבים את החלון עבור ההתחברות, ההרשמה והמשחק עצמו. לפי דעתי, ניתן היה להפריד גם בין ההתחברות וההרשמה לבין המשחק עצמו על מנת לשפר את קריטריון זה למדידת איכות הקוד. כלומר, ליצור 3 קבצים כפי שצויין למעלה באופן נפרד עבור המסכים הנוגעים להתחברות וההרשמה, ויצירת עוד 3 קבצים עבור החלונות של המשחק. |

שאלות פתוחות:

1. **כיצד הרגשת במהלך החוויה של מעבר על הקוד?**  
   במהלך המעבר על הקוד הרגשתי כי הקבוצה שכתבה את הקוד השקיעה המון מאמצים בפיתוח האפליקציה. ראיתי שימוש נרחבים בכלים מתקדמים של עיצוב, הנפשות ואפקטים שמטרתו הייתה לשפר את חווית המשתמש.  
   בנוסף, הרגשתי שאני מבין את כל מה שכתוב בקוד ללא קשיים, כיוון שהקוד היה קריא, הגיוני ומסודר.  
   היה מעניין עבורי לראות פרוייקט של קבוצה אחרת, שכן הייתי חשוף עד היום רק לתכנים של הפרוייקט שלנו. יתר על כן, היה מעניין לראות את צורת החשיבה של הקבוצה ואופן המימוש שהם בחרו בכדי לענות על הדרישות ולבנות את האתר.
2. ***האם לדעתך ניתן לשפר תרגיל זה?***

לדעתי התרגיל מספיק טוב ומלמד, כלומר אין צורך לשפר אותו.

1. **הערות נוספות:** אין לי הערות נוספות.

**שם הסטודנט: אלדד שניאור**

|  |  |
| --- | --- |
| קריטריון | הסבר |
| מימוש | האם הקוד מבצע את הנדרש?  כן הקוד מבצע את הנדרש. בעבור המשתמש קיים מסך המשחק שכולל בתוכו מענה נכון על שאלות, הצגת תשובה לא נכונה, תצוגה של סטטיסטיקה, רשימה של משתמשים מובילים. ובעבור משתמש מנהל קיים אופציות של הוספת שאלה מחיקת שאלה ועדכון שאלה. נוסף על כך קיימים אלמנטים חדשים כמו איך לשחק אשר מאוד תרם להבנת המשחק. |
| יעילות | האם ניתן לשפר את יעילות הקוד?היות והקוד כולו כתוב בPython & HTML ,CSS, JAVASCRIPT אין יותר מידי מקום לשיפור יעילות. ישנה חלוקה ברורה לפונקציות ובכך בעצם מונעים כתיבת קוד חוזר. |
| פשטות | האם ניתן לפשט את הקוד?  הקוד מפושט בצורה טובה למדי לפי קטגוריות עיקריות שהן: HTML, CSS, Python, Javascript. ניתן היה לפשט את קטגורית הjavascript לתתי קטגוריות אשר כל אחת קשורה לנושא מסויים. למשל בעבור פונקציות הקשורות לגרף לשים תחת קובץ נפרד. ובכך לשמור על פשוט מקסימלית.  נוסף על כך, קיים בקוד מיפוי של הצבעים של האתר לפי שמות רלוונטים. ובכך במידה ונרצה לשנות ערך בCSS. נוכל לדעת לפי שם של אותו ערך. אני אהבתי את התוספת הזאת ואפילו נוסיף אותה לפרוייקט שלנו ! |
| מודולריות | האם הקוד מודולרי מספיק? (חלוקה תקינה לקבצים, פונקציות ומחלקות)  כן קיימת חלוקה תקינה לקבצים לפי הgoogle colab. כאשר הcell הראשון מתייחס לדף הHTML הראשי. הcell השני מתייחס לקובץ הCSS שלהם. הcell השלישי מתייחס לקובץ הjavascript והcell האחרון מתייחס לקובץ הpython.  ע"י חלוקה זאת השימוש בקוד באופן כולל ברור יותר וקל יותר. נוסף על כך הקוד כתוב בצורה קריאה ביותר. פונקציות אשר קשורות אחד לשנייה רשומות באופן צמוד וכתוצאה מכך ההבנה של הקוד הרבה יותר קלה. |
| תבניות עיצוב | האם ישנו מימוש ברור ונכון של תבניות עיצוב?  לא אין מימוש כלל של תבנית עיצוב. אך כן יש חלוקה ברורה ומסודת של קבצים. ניתן להוסיף תבניות עיצוב בכדי לשפר את השימוש כמו לדוגמא ע"י שימוש בSingleton אפשר לגרום ליצירת אובייקט יחיד אשר ניגש למסד הנתונים. |
| OOP | האם ישנו שימוש נכון בעקרונות OOP? האם ניתן לשפר?  אין שימוש בכלל בקוד בעקרונות הOOP. או בכלל באובייקטים מסוג כלשהו. ניתן ליצור מחלקות שונות ע"י לייעל את כתיבת הקוד ולהפוך אותו לנוחה. כך לדוגמא בעבור הפונקציות showModal,closeModal אפשר לאחד אותם ולהוסיף מחלקה חדשה Modal אשר שומרת בה את הtitle,information,modalElement. ומבצעת 2 פונקציות שונות שהן showModal, closeModal. שימוש זה יגדיל את ההפשטה של הפרוייקט. |
| באגים וטעויות | האם ישנם מקרים בהם הקוד לא מתנהג כצפוי? אילו?  לא מצאנו מקרים בהם הקוד התנהג לא כצפוי. הקוד עבד בכל חלקיו השונים. |
| טיפול בשגיאות | האם ישנה התראה על שגיאות משתמש?האם ישנו שימוש במנגנון תפיסת שגיאות?האם הבדיקה ודיבוג הקוד נוחים?האם הודעות המשתמש ברורות וידידותיות למשתמש?  בעבור הUI של המשתמש קיים מנגנון תפיסת שגיאות כמו למשל בעת הרשמה והתחברות של מתשמש קיים אימות של נתונים במידה והמשתמש קיים, או במידה והסיסמה אינה נכונה או משתמש שאינו קיים. ההודעות ברורות ומגיעות באופן ישיר אל המשתמש בעת שגיאה. |
| בדיקות | האם יש בדיקות שניתן להוסיף?  הקוד כולל הרבה בדיקות רבות אך חלקן יכולות להיות בעיתיות. המתכנתים בקוד משווים ערכים באופן לא ביטחותי ע"י שימוש ב’==’ במקום ב’===’ ובכך פתוחים ליותר סיכונים לאורך חיי הפרוייקט. |
| שימושיות | האם הממשק שימושי? (usability)? האם המשחק מעורר רצון להמשיך לשחק?  הממשק מאוד שימושי הבנת הממשק היא ברורה וקלילה. ובעבור כל משתמש מתחיל המשחק ברור מאוד לאורך כל חלקיו וקיים הסבר פשוט לכל חלק.  המשחק כולל בתוכו UI נעים וגם סטטסטיקה אל מול שחקנים אחרים שמהווה מתן חווי אל מול אנשים אחרים |
| תיעוד | האם הממשק מתועד?  כן הממשק תועד לאורך חלקיו השונים |
| אתיקה | האם יש שימוש בנתוני משתמש אשר מפרים פרטיותו?האם יש ניצול של התנהגות או חולשה אנושית?האם ישנו algorithmic bias כנגד קבוצה או פרטים מסוימים?  אין כלל שימוש בנתוני משתמש אשר מפרים את פרטיותו. יתרה על כך אין שמירה של נתונים שונים ע"י המשתמש אלא רק שם משתמש וסיסמה ובכך אין אופציה לכך שפרטיותו של משתמש תפגע בצורה כלשהי |
| אבטחה | האם ישנו מידע אבטחה (שמות משתמש, ססמאות) גלויים?  הקוד משתמש בשם המשתמש של השחקן לאורך חיי המשחק. אך לאורך כל השימוש במשחק הסיסמא אינה גלוייה לעיני המשתתפים. נוסף על כך בעת התחברות הסיסמה מוחבאת. |
| ביצועים | האם לדעתך שינוי עתידי בקוד יכול לגרום לפגיעה בביצועים?  לא. שינוי עתידי לא יפגע בביצועי הקוד במידה והמתכנתים ישמור על הסדר האחיד שקיים בקוד כרגע. בטווח הארוך במידה והמשחק יתפתח אפילו יותר השימוש בgoogle colab אולי יהווה בעייה והייתי מעלה את זה על סדר היום. |
| ביצועים | האם ניתן לשפר לדעתך את הביצועים?  ניתן לשפר את הביצועים היות ובקוד המתכנתים משנים את הדפים השונים ע"י שינוי של קובץ הHTML. הייתי שומר את קבצי הHTML השונים שמתשנים לאורך חיי הפרוייקט ובכך לחסוך בזמן הריצה של שינוי הערכים ורק בעצם לשנות את הדף. בנוסף לטווח הארוך הייתי מיישם OOP בכדי לשמור על עקורונות תכנות טובים יותר. |
| קריאות - readability | האם הקוד מובן בקלות?אילו חלקים בקוד היו לא ברורים עבורך?כיצד ניתן לשפר את קריאות הקוד?  הקוד מובן לחלוטין. בחלק מן הקטעים בקוד היו נקודות בהם שילבו HTML בתוך javascript בחלקים אלו יותר קשה להבין מה מבצע הקוד ולמה מתבצע השינוי של הHTML. הייתי משנה באותה צורה שאמרתי בעבור הביצועים למעלה.  המתכנתים בקוד דאגו לתת לכל צבע באתר ערך עם שם ובכך לגרום לקריאות של הקוד להיות מאוד מובנת במידה ונרצה לעבור על העיצוב. בנוסף הקוד משתמש בשמות משתנים והגדרתם באופן ברור ואחיד |
| סעיף פתוח  **ארכיטקטורה** | האם ארכיטקטורת בניית המשחק בטוחה לטווח הארוך ?  לפי דעתי כן. למרות שהצעתי להם לעבור לOOP במידה והמשחק יתרחב. המשחק עצמו מסוגל לבצע שינויים באופן קליל ומסתגל ללא שינויים עקריים מידי.  האפלקציה עצמה מחולקת לHTML CSS ו javascript שמירה על חלוקה טובה תוסיף לארכיטקטורה. |
| סעיף פתוח  **accessibility** | האם המשחק מונגש ?  כן המשחק מונגש לכלל המשתמשים השונים. הפונטים הם בגודל מעל הממוצע. נוסף על כך בעבור כל התמונות קיימת אופציה alt אשר בה קיים תיאור מילולי לתמונה. |

**שאלות פתוחות:**

1. ***כיצד הרגשת במהלך החוויה של מעבר על הקוד?***

*אני באופן אישי נהנתי לעבור על הקוד. המשחק עצמו נראה מדהים והקוד כתוב בצורה ברורה. בנוסף המעבר על הקוד נתן לי פרספקטיבה שונה על הישום והשימוש של דברים שונים בקוד. שנוכל ליישם במידת הצורך בקוד שלנו.*

1. ***האם לדעתך ניתן לשפר תרגיל זה?***

*כן התרגיל עצמו היה מאוד מהנה כתיבת הפרוייקט והבדיקה שלו. אך הוא היה צפוף מידי מבחינת לוחות זמנים. באופן אישי הרגשתי שהגשה של הפרוייקט עצמו + השאלות הפתוחות לאותו תאריך היה מעט מוגזם. חוץ מזה מאוד נהנתי לעבוד על האתר. לפי דעתי הייתי באופן אישי מוותר על בניית אתר ומנסה משהו אחר היות והתחושה האישית שלי ההתנסות עם firebase הייתה מינמלית ורוב הפרוייקט שלנו עסק בjavascript , html , CSS*

1. ***הערות נוספות***

אני הייתי מעדיף שהיינו נוגעים בAWS ובAZURE מאשר בfirebase של גוגל היות והוא מאוד מוכר ובעל המון מדריכים שונים באינטרנט. לפי דעתי אם היינו מבצעים פעילויות שונות תחת AWS וAZURE זה היה מוסף רבות להבנה.

**קישור ל-GIT הקבצים נמצאים בברנץ' TCCv2:**

<https://github.com/cohensh96/CloudComputing/tree/TCCv2>

**פרטי חשבון גוגל:**

העלנו את הקבצים לדרייב ומשם אנחנו טוענים אותם לקולאב לכן לפני הכניסה למחברת בקולאב יש להתחבר לחשבון הגוגל הבא:

**שם משתמש:** ThinkCloudComputing@gmail.com

**סיסמא:** TccInc123

**הקישור למחברת בקולאב:**

<https://colab.research.google.com/drive/13bYT4Qujn9O0EutgtdAiZ4nl6-GyhH68?usp=sharing>