תרגיל בית 3- חלק ב יבש – עבודה בקבוצות וביחידים

קבוצה BEE  
  
רז קסל 206017725  
נבו גוטליב 206234353  
איתי בנג'מין 316514207  
דניאל ממן 318155884   
רום הראל 318474582  
בן אבנר מריג'ן 209240274

בדיקות ו - *code review* לקבוצת pacaton

חלק ראשון: עבודה קבוצתית

הציגו טבלה עם הבדיקות ( acceptance tests) שהגדרתם באיטרציות הקודמות (תרגיל 2 ותרגיל 3א):

בצעו את הטסטים שלכם לקבוצה אחרת שקיבלתם את שמה במסגרת התרגיל.  
 רשמו טבלה עם תוצאות הטסטים שהרצתם. במידה והוא לא עבר, יש לציין מה היתה הבעיה.

**(הפתרון מופיע בטבלאות למטה - בהן יש אינדקציה לכל טסט וכאשר הטסט לא עבר הסברנו את הבעיה)**

מה ניתן להסיק מההרצה? (האם הקוד עמד בכל היעדים , האם יש מקום לשיפור ומהו)

לאחר הרצת הקוד, והרצת הבדיקות שעשינו עבור הפרויקט שלנו ראינו כי רוב הבדיקות עברו, פרט לכמה בדיקות בודדות.  
נסיק מכך שרוב הפרויקט עובד בצורה תקינה, אך ישנם כמה דברים קטנים שיש לשפר. הדבר מעיד על עבודה נכונה וטובה של הקבוצה הנבדקת. ביקורת עצמית הינה דבר חשוב, לדעת לקבל הערות ולשפר אותן הינו תהליך בריא ותקין במחזור חיים של הפרויקט.  
לאחר ההרצה הסקנו כי המוצר הסופי מספק חווית משתמש ברמה גבוהה.

בכל שלב מתוארת בצורה ברורה איזו פעולה יש לעשות ואיך ניתן להתקדם. במהלך ההתנסות במשחק אנחנו התרשמנו שהוא ממלא את מטרתו, ניתן לומר שהקוד היה קריא וברור, ממש את הפונקציונליות במשחק ועונה על דרישות הפרויקט.

ניתן לשפר את הקוד בעניין הסדר/הארגון, כלומר חלוקה נכונה ויעילה לקבצים. מרכיב הכרחי נוסף הוא התיעוד, אנו חושבים כי הוספת הסבר עבור כל פונקציה תהפוך את הקוד למובן וקריא יותר. בנוסף, מבחינת חווית המשתמש אשר מועברת - לדעתנו יש צורך להוסיף פידבקים עבור פעולות מרכזיות שהוא מבצע במהלך השימוש באפליקציה.

לסיכום, מההרצה ראינו כי הפרויקט עונה על הדרישות אשר מספק חווית משתמש ברמה גבוהה ואופצית למידה של מושגים בנושא מחשוב ענן.

***Acceptance Test - Iteration 1***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Description** | **Expected Result** | **Precondition** | **Result** |
| **1** | **Login with valid username and password** | **User is successfully logged in** | **User account exists in the database** | **Pass** |
| **2** | **Login with invalid username and password** | **User is not logged in** | **User account does not exist in the database or incorrect credentials** | **Pass** |
| **3** | **Sign up with a new username and password** | **User account is created successfully** | **User account does not already exist in the database** | **Pass**  **But there is no massage** |
| **4** | **Sign up with an existing username** | **User account is not created** | **User account with the same username already exists in the database** | **Pass**  **But there is no massage** |
| **5** | **Start the game** | **Game starts and displays the first question** | **User is logged in and there are available questions** | **Pass** |
| **6** | **Select an answer for a question** | **Answer is recorded and next question is displayed** | **User is in the middle of the game and there are remaining questions** | **Pass** |
| **7** | **Select an answer for the last question** | **Answer is recorded and the end of the game is reached** | **User is on the last question of the game** | **Pass** |
| **8** | **View the result page** | **Result page is displayed with score and answers** | **User has completed the game** | **Pass** |
| **9** | **View answer for a question on the result page** | **Correct answer is displayed for the selected question** | **User has completed the game and is viewing the result page** | **Failed**  **There is no option to see the questions we did. During the game you see the correct answer.** |
| **10** | **Create a new user profile** | **User profile is successfully created** | **User is not already registered** | **Pass But there is no massage** |
| **11** | **User successfully logs into the application.** | **User should be able to log in successfully.** | **The trivia game application is open.** | **Pass** |
| **12** | **User plays three trivia games.** | **The application should track the user's gameplay and record the necessary statistics.** | **User is logged into the application.** | **Pass** |
| **13** | **User chooses to view statistics and plot after three games.** | **The application should display the user's statistics and plot.** | **User has played three trivia games.** | **Pass** |
| **14** | **User logs out of the application.** | **User should be able to log out successfully.** | **User is logged into the application.** | **Pass** |
| **15** | **User plays two trivia games and then exits the application.** | **The application should save the progress of the user's gameplay.** | **User is logged into the application and has played two games.** | **Failed**  **The graph shows data on the last game only** |
| **16** | **User reaches a high score in a trivia game.** | **The application should record the high score achieved by the user. (last 15)** | **User is logged into the application and has played a game.** | **Pass** |
| **17** | **User attempts to view statistics and plot without playing any games.** | **The application should display a message indicating that there is no data available as the user has not played any games.** | **User is logged into the application.** | **Pass** |
| **18** | **User logs out without playing any games.** | **User should be able to log out successfully without any issues.** | **User is logged into the application.** | **Pass** |
| **19** | **User attempts to add a question with missing fields.** | **The application should display an error message indicating that all required fields must be filled in to add a question.** | **User is logged into the application.** | **Pass** |
| **20** | **User attempts to remove a non-existent question.** | **The application should display an error message indicating that the selected question does not exist.** | **User is logged into the application and there are existing questions.** | **Pass** |
| **21** | **User successfully adds a question** | **After adding a question, the application should display the added question in the trivia game.** | **User is logged into the application.** | **Pass**  **But there is no massage** |
| **22** | **User attempts to remove the last remaining question from the trivia game.** | **The application should prevent the user from removing the last question and display a message indicating that it cannot be removed.** | **User is logged into the application and there is only one question remaining.** | **Failed**  **Pressed on delete question and nothing happened** |
| **23** | **Successfully retrieve a question from the database.** | **The question should be fetched from the database without errors.** | **The trivia game application is connected to the database and there is an existing question.** | **Pass** |
| **24** | **Successfully update a question in the database.** | **The question should be updated in the database without errors.** | **The trivia game application is connected to the database and there is an existing question.** | **Pass**  **But there is no massage** |
| **25** | **Successfully delete a question from the database.** | **The question should be deleted from the database without errors.** | **The trivia game application is connected to the database and there is an existing question.** | **Failed**  **Pressed on delete question and nothing happened** |

***Acceptance Test - Iteration 2***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Test ID*** | ***Description*** | ***Expected Result*** | ***Precondition*** | ***Result*** |
| *1* | *Display graph data after 3 games* | *The graph should display data for at least 3 games* | *User played 3 games* | **Pass**  **But the graph display data only for the last game** |
| *2* | *Verify that the graph data is not displayed before 3 games.* | *This should now display a graph* | *User played at least 2 games* | ***NOT  RELEVANT***  **We suggest improving the graph to show progress over several games and not just one** |
| *3* | *Verify the ability to update a question in the database.* | *The question updated successfully* | *The trivia game application is connected to the database and the admin entered valid data.* | **Pass**  **But there is no massage** |
| *4* | *Update question without required data* | *The question can not be updated* | *The trivia game application is connected to the database and there is required data that is missing.* | **Pass** |
| *5* | *Verify leaderboard displays players in descending score order* | *The leaderboard arranges players based on their scores in descending order, ith the highest score first.* | *The game has been completed* | **Pass** |
| *6* | *Verify leaderboard shows correct player information* | *The leaderboard correctly displays each player's name, score, and ranking.* | *The game has been completed* | **Pass** |
| *7* | *Verify leaderboard handles tie scores correctly* | *If multiple players have the same score, they should be ranked accordingly and displayed in the order they finished the game.* | *The game has been completed* | **Pass** |
| *8* | *Verify "back" option functionality on the tutorial page* | *When the user clicks on the "back" option on the tutorial page, they should be redirected back to the previous page (e.g., main menu).* | *User is on the tutorial page* | **Pass** |
| *9* | *Verify leaderboard handles large number of players* | *The leaderboard should be able to handle and display scores for a large number of players without errors.* | *The game has been played/completed* | **Pass** |

**חלק שני: עבודה אישית**

שם הסטודנט: **רום הראל**

|  |  |
| --- | --- |
| קריטריון | הסבר |
| מימוש | כן , הקוד מבצע את הנדרש, התוכנה שמימשו עונה על הדרישות מהפרויקט. נראה שהשתמשו בחבילות הרלוונטיות עבור פיתוח WEB , הקוד מחולק בצורה חכמה , לכן המימוש טוב לדעתי , כמובן מתקמפל ללא שגיאות . |
| יעילות | החיבור עצמו לDB יכול להיות חיבור יחיד סינגלטון ובכך לקצר את אופן הצגת השאלות , והתלות בחיבור עצמו . כך יהיה ניתן גם למנוע שכפול קוד . עדיין נראה שהקוד דיי יעיל כמובן שתמיד אפשר עוד לשפר . |
| פשטות | הקוד נראה דיי מפושט קל להבנה , הוא אמנם ארוך מאוד בגלל שכפולי קוד אבל עדיין נראה מאוד פשוט וברור . |
| מודולריות | הקוד מדולרי , יש חלוקה טובה למודלים לפי תחומי האחריות ופונקציות, כך יהיה גם קל בעתיד לגשת ולשנות חלקים ספציפיים בקוד לפי מה שהם מבצעים . הייתי אולי מוציא את קבצי ה CSS לקובץ אחד , אני רואה שהם חוזרים באופן קבוע במהלך כל אחד מהמודלים . |
| תבניות עיצוב | אני מזהה סוג של מימוש לתבנית עיצוב של שכבות , כן ניתן לראות שכל אחד מקטעי הקוד במחברת בפני עצמו מחלוק ל מעין MVC . |
| OOP | לא הצלחתי לזהות שימוש ב OOP , נראה ששימוש כזה היה מקל קצת על קריאת הקוד והיעילות שלו . |
| באגים וטעויות | לאחר ביצוע ACCEPTANCE TEST זיהיתי שגיאה בפאנל מנהל עבור מחיקת שאלה . נראה שזו שגיאה מאוד קטנה ומינורית , בסך הכל הקוד מבצע את הנדרש ללא באגים ושגיאות בעיקר בחלק המשמעותי של משחק המשתמש . |
| טיפול בשגיאות | קיימות הודעות למשתמש שהן ברורות מאוד , לדוגמא בהרשמה על שדות חסרים , או בהתחברות ובפאנל מנהל . נראה שברוב המקרים יש תפיסת שגיאות באמצעות ולידציה של טפסים . דיבוג הקוד והבדיקה בו נוחים . |
| בדיקות | הייתי מבצע בדיקות נוספות על ביצועים אולי , אבל מעט קשה לבצע דברים כאלה בעזרת הCOLLAB , ניתן היה לבדוק אולי מה קורה כאשר אין חיבור כלל ל DB . |
| שימושיות | האם הממשק שימושי? (usability)? האם המשחק מעורר רצון להמשיך לשחק? הממשק שימושי מעורר רצון להמשיך לשחק , העיצוב יפה ואלמנטי המשחוק הרבים באים לידי ביטוי במהלך המשחק והשימוש בו . ה BADGES הם רעיון מצוין שבדיעבד הייתי שמח שיהיה גם בפרויקט שלנו . |
| תיעוד | הממשק מתועד באופן חלקי , החלקים המרכזיים בהחלט מתועדים . ניתן תמיד לשפר את זה ולהוסיף תיעוד על כל פונקציה ופונקציה ולא רק על המרכזיות ביותר . |
| אתיקה | קיים פרוטוקול בסיסי של אתיקה , לא ניתן לגשת לאתר לפני התחברות והרשמה מסודרת , משתמש רגיל אינו יכול לגשת לאדמין מנהל . וגם המנהל בעצמו לא חשוף לפני המשתמשים . |
| אבטחה | לא זיהיתי פרצות אבטחה משמעותיות , המנהל לא חשוף לפרטי המשתמשים , והמשתמש עצמו גם לא יכול לגשת לפרטי הסיסמה שלו ובכך אולי היו נחשפים לגורמים בלתי רצוניים . גם לא הייתה דרישה לאבטחה ברמה מסוימת ולכן אני חושב שזה עומד בסטנדרטים . |
| ביצועים | בעזרת מניעת שכפול קוד ובעזרת חיבור יחיד לDB היה אפשר לשפר ביצועים , אך נראה שזה עומד בדרישות הפרויקט ולכן הביצועים טובים יותר . אצלנו הביצועים השתפרו משמעותית באמצעות חלוקה לקבצים הייתי מציע זאת . |
| קריאות - readability | הקוד נראה דיי ברור ף שמות משתנים משמעותיים , הייתי מוסיף קצת יותר תיעוד פונקציות מרכזיות .  קצת קשה להתחבר לקוד כאשר הוא לא מחולק עבור קבצי HTML פונקציות JS ופייתון בנפרד , אבל הקושי ברור ומובן בגלל הפלטפורמה. אני חושב שהוא עומד בדרישות וברור בהסתכלות כללית . |
| עמידה בדרישות | פורמט שמירת השאלה בDB הוא לא בדיוק לפי מה שביקשו . |
| עיצוב | האתר מעוצב ברמה גבוהה מאוד , אני ממש התחברתי לרקע הגרדיאנטי . |
| אלמנטי משחוק | כמו שציינתי , קיימים אלמנטי משחוק מעולים של BADGES וטבלאות LEADERBOARDS שמוסיפים את הרצון להתמיד ולשחק . |

שאלות פתוחות:

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

עבורי זוהי פעם ראשונה שאני מבצע רפלקציה על קוד של מישהו אחר שלא הייתי שותף בכתיבה שלו , כיף לראות איך הקבוצה וחבריה ניגשו לתרגיל בצורה שונה משלנו , האמת גם למדתי עוד דרך לגשת למשימה מהסוג הזה . החוויה מבחינתי מלמדת ומשמעותית .

*2.* האם לדעתך ניתן לשפר תרגיל זה?

ניתן לשפר תרגיל זה על ידי ציוות אחד מאנשי הקבוצה הבודקת והנבדקת עבור מענה על שאלות . היה קצת קשה ליצור קשר ולמצוא עם מי אפשר לדבר על חלקים כאלה ואחרים בקוד . אני חושב שאם הייתה לי האשפרות לדבר עם מישהו שכתב את הקוד שיוכל לתת לי תשובות וכיוון והסבר כללי התהליך היה קצת יותר מהיר ומשמעותי עבורי ועבורו .

*3.* הערות נוספות

אני חושב שהייתי שמח לראות דוגמה בכיתה כיצד מבצעים רפלקציה על מעבר קוד בצורה מפורטת , אני חושב שבסך הכל הקורס והחוויה כולה מסתכמת כאן ואני למדתי המון על מחשוב ענן ועל פיתוח באמצעות פלטפורמת COLLB ה PAASית .

**שם הסטודנט: איתי בנג'מין**

|  |  |
| --- | --- |
| קריטריון | הסבר |
| מימוש | המימוש מבוצע בצורה טובה. השימוש ב- Python, JavaScript, HTML ו-CSS מבוצע היטב. |
| יעילות | נראה שהקוד יעיל בתפעול שלו, עם זאת בחירת שפה מרכזית ולא לערבב בין השפות יכול להביא לקוד הרבה יותר יעיל וגנרי. |
| פשטות | הקוד באופן כללי מובנה ומאורגן היטב, אך ערבוב השפות (במיוחד CSS ו-HTML) יכול להקשות על מעקב אחר חלקים מסוימים. הפרדת ה-CSS למקטע או קובץ אחר יכולה לעזור בכך. |
| מודולריות | הקוד מאורגן בפונקציות, מה שמשפר את המודולריות. |
| תבניות עיצוב | מימוש הקוד נעשה ב procedural programming |
| OOP | אין שימוש בOOP, התכנות מתבצע בצורה פרוצדורלית |
| באגים וטעויות | לא זיהיתי באגים וטעויות |
| טיפול בשגיאות | נראה שאין הרבה טיפול בשגיאות. זה יכול להיות תחום לשיפור. |
| בדיקות | אין בדיקות קוד, לא התבקש. |
| שימושיות | ממשק המשתמש נראה מעוצב היטב, עם פריסה ברורה ופעולות מוגדרות היטב, שימוש בטבלת המשתמשים עם הכי הרבה נקודות ויכולת קבלת badge במספר נקודות מסויים מגדיל את השימושיות ונותן רצון להמשיך לשחק. |
| תיעוד | החלקים המרכזיים אכן מתועדים דבר שמקל על קריאת הקוד, הייתי מוסיף תיעוד גם בתוך קטעי הקוד של javacript . |
| אתיקה | נראה שהמשחק הוא מערכת ניקוד פשוטה עם לוח הישגים, שאינו מעלה חששות אתיים ברורים או אלגוריתמים פולשים להשגת מידע ואיפיון המשתמש. |
| אבטחה | הקוד כולל חיבור ל-Firebase, אך נראה שלא נתנו אמצעי אבטחה כגון הצפנה של סימאות משתמש, וכל ססמאות המשתמשים מוחזקות בלקוח בעת משחק (לא מאובטח כלל). |
| ביצועים | שינוי הקוד בעתיד כאשר יתבצע ברמה טובה אין שום סיבה שיפגע בביצועים של הקוד |
| ביצועים | כדי לשפר את הביצועים אני הייתי נמנע מלחבר את הפייתון לHTML בצורה כזאת, כאשר כל פעם שמשתנה המידע הקוד בפייתון צריך לטעון מחדש את קוד הHTML ולעשות רענון.  כאשר נעבור לשימוש בHTML JAVASCRIPT לכל עיבד הנתונים והצגתם הקוד ירוץ במהירות רבה יותר |
| קריאות - readability | הקוד לרוב קריא עם שמות פונקציות ברורות ושמות משתנים הגיונים. עם זאת היעדר הערות בJavaScript וערבוב השפות של Pyhton עם הHTML עלול להקשות על מעקב אחר חלקים מסויימים.  לדוגמא פונקציית startGame פונקצייה ארוכה שקשה לקריאה עקב חיבור וערבוב בין השפות. |
| תחזוקה  כמה קל ניתן להבין, לתקן ולעדכן את הקוד על ידי מפתחים אחרים | לקוד שסופק יש שמות פונקציות ברורים ושמות משתנים, התורמים באופן חיובי לתחזוקה.  עם זאת, היעדר הערות וערבוב השפות בחלק מהקטעים יכולים להפחית את יכולת התחזוקה. |
| מדרגיות  Scalability | השימוש ב-Firebase לאחסון נתונים מצביע על כך שהפרויקט עשוי להתמודד עם כמות סבירה של נתוני משתמש, שכן Firebase הוא מסד נתונים ענן NoSQL ניתן להרחבה. |

שאלות פתוחות:

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

החוויה הייתה מאירת עיניים. זה הביא לי הזדמנות לנתח באופן ביקורתי פיסת קוד מנקודות מבט שונות. התהליך אפשר לי להבין את החשיבות של שיטות תכנות שונות, וכיצד הן תורמות לאיכות הכללית של הקוד. זה גם הציג הזדמנות לחשוב כיצד ניתן לשפר את הקוד, תוך טיפוח חשיבה של פתרון בעיות.

*2.* האם לדעתך ניתן לשפר תרגיל זה?

הכללת שלב שבו המבקר מציע שינויים או שיפורים ספציפיים לקוד עשויה לקדם חווית למידה מעשית יותר. יתר על כן, התרגיל יועיל יותר עם יכלול דיאלוג בין המבקר לבין המתכנת המקורי. זה יאפשר הבנה מעמיקה יותר של תהליכי החשיבה שמאחורי החלטות תכנות מסוימות והאילוצים שהמתכנת עשוי לעבוד בהם.

*3.* הערות נוספות

בסך הכל, תרגיל זה היה הזדמנות חשובה להתנסות בסקירת ובדיקת קוד. זה היה חשוב במיוחד לסקור את הקוד במונחים של פרמטרים כמו אבטחה, יעילות ותחזוקה. זה חיזק את הרעיון שתכנות הוא לא רק לגרום לתוכנית לעבוד, אלא גם לשקול כיצד הקוד משתלב במערכות גדולות יותר וכיצד נשתשמש ונתחזק אותו בעתיד. בהמשך, אהיה מעוניין לחקור שיטות סקירת קוד מתקדמות יותר, כגון אלו המשמשות בצוותי פיתוח מקצועיים.

**שם הסטודנט: נבו גוטליב**

|  |  |
| --- | --- |
| קריטריון | הסבר |
| מימוש | הקוד מבצע את הנדרש אך יש לציין כי הוא מסורבל וקשה לקריאה מאחר ולא קיים תיעוד, חלוקה לקבצים וסדר כללי בין הפונקציות השונות. |
| יעילות | באופן כללי הקוד יעיל ומבצע את הנדרש אך יש אפשרות להפוך אותי ליעיל יותר לדעתי.  לאחר מעבר על הקוד ניתן לראות כי אפשר לצמצם את כמות שכפול הקוד הקיימת. למשל בפונקציות rankShop(), process\_data1() בשתי הפונקציות נמצא בלוקים זהים ללקיחת מידע מהמסד. עדיף היה לבנות פונקציה אחת ללקיחת מידע מהמסד ולזמן אותה בעת הצורך. |
| פשטות | כן, ניתן היה למשל להשתמש בתבנית html על מנת להפריד בין קוד ה-html ל-Python מה שיאפשר תחזוקה נוחה וקלה יותר. |
| מודולריות | הקוד לא מודולרי מספיק. לא קיימת חלוקה לקבצים, ופונקציות הקשורות לאותן מסכים או למסד הנתונים. |
| תבניות עיצוב | לא מצאתי שימוש ומימוש בתבנית עיצוב מסוימת בקוד. |
| OOP | לא מצאתי שימוש כלשהו בעקרונות oop. ניתן היה לשפר את הקוד ע''י שימוש בעקרונות oop כגון הורשה וחלוקה למחלקות על מנת לאפשר שימוש חוזר ונוח יותר. |
| באגים וטעויות | כן, תוך שימוש במסך המנהל שמתי לב למצב בו כאשר מכניסים שאלה חדשה אין מכוון על כך שהשאלה נוספה למאגר. בנוסף מחיקת שאלה בלתי אפשרית. |
| טיפול בשגיאות | במסך המנהל יש התראות על שגיאה כאשר מנסים להכניס שאלה עם שדה כלשהו ריק. ההתראה מובנת וברורה. כאשר מבצעים הרשמה לא נקבל התראה על שדה ריק.  לאחר רישום מוצלח למערכת לא נקבל התראה על הצלחה אך ההרשמה מתקבלת ועובדת באופן כללי. |
| בדיקות | הייתי מוסיף בדיקות בנוגע לכל הקשור להכנסה ומחיקת שאלות ממסד הנתונים. האם השאלה נוספה בהצלחה או נמחקה בהצלחה. |
| שימושיות | לחלוטין. היה כיף ממש לשחק. הכל ברור ונגיש. האלמנטים המשחוקיים נותנים מוטיבציה להמשיך לשחק ולצבור עוד תגים. |
| תיעוד | הקוד אינו מתועד כלל, אך ממשק המשתמש ברור לחלוטין ואין צורך להתאמץ להבין מה כל דבר אמור לעשות. |
| אתיקה | לא נתקלתי בדבר כזה במהלך הבדיקה. |
| אבטחה | לא. על מנת להכנס למשחק הייתי מחויב להרשם מאחר ושום נתוני משתמש לא היו חשופים עבורי. |
| ביצועים | ייתכן, מאחר והקוד דיי מסורבל וקשה לקריאה הוספת קוד או הסרתו מהקוד הקיים עלולה לפגוע בתקינות המערכת. |
| ביצועים | ניתן לשפר את הביצועים ע''י refactoring לכלל הקוד, לצמצם שכפול קוד ולהשתמש בחלוקה לקבצים. בנוסף הייתי מציע לבצע תיעוד מעמיק לכל פונקציות המערכת. |
| קריאות - readability | האם הקוד מובן בקלות?אילו חלקים בקוד היו לא ברורים עבורך?כיצד ניתן לשפר את קריאות הקוד? הקוד היה קשה מאוד לקריאה. הקוד ארוך מאוד ולא מחולק לקבצים. נוצר "בלאגן בעיניים", בנוסף הקוד לא מתועד וכל פונקציה כתובה בסדר שונה. הייתי מציע לערוך את הקוד ולקצר אותו ע''י ביצוע refactoring. |
| סעיף פתוח- תלויות | לא מצאתי תלויות מיותרות בקוד כגון ספריות שלא בשימוש או ייבוא שלא לצורך של framework כלשהו. |
| סעיף פתוח-יכולת בדיקה  testability | הקוד עלול להיות בעייתי לבדיקות Unitests מאחר ואין חלוקה מוגדרת לקבצים והתלויות של פונקציה. |

שאלות פתוחות:

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

האמת היה מאוד קשה לעבור על הקוד מאחר והוא היה עצום בגודלו ודיי מסורבל. לאחר שהצלחתי להבין את סדר הדברים בקוד, נהנתי לראות גישות שונות ודרכים שונות לפיתוח משחק שדומה מאוד למשחק שאנחנו יצרנו.

*2.* האם לדעתך ניתן לשפר תרגיל זה?

אולי להגביר את שיתוף הפעולה בין הקבוצה הבודקת לנבדקת. לבצע את התרגיל עם נציג הקבוצה הנבדקת שיוכל להסביר ולהכווין במהלך קריאת הקוד ולספק סדר יותר ברור של הדברים.

*3.* הערות נוספות

היה מהנה מאוד לשלב בתרגיל עבודה יחד עם קבוצות אחרות, גם הקבוצה הבודקת שלנו וגם הקבוצה הנבדקת. קיבלנו פרספקטיבה חדשה ועוד דוגמאות באמצעות הפרויקט של הקבוצה הנבדקת והפידבקים של הקבוצה הבודקת.

**שם הסטודנט: רז קסל**

|  |  |
| --- | --- |
| קריטריון | הסבר |
| מימוש | כן, בסופו של דבר הקוד מבצע את הנדרש, המשחק עולה בצורה חלקה ועובד היטב. |
| יעילות | כן, אפשר לצמצם את מספר הפעמים בהם הקוד מקיים אינטרקציה עם שירותים חיצוניים כמו חיבור לדאטה בייס לדוגמא. צמצום מקרים אלה ישפר את יעילות הקוד וזמן התגובה. |
| פשטות | כן, אפשר לחלק את הקוד לקבצים: html, js, css. בכך הפרוייקט יהיה מחולק לשכבות לפי ארכיטקטורה והקוד יהיה הרבה יותר קריא. |
| מודולריות | הקוד מודולרי, כנראה בגלל יכולת ההכלה של כל cell בקולאב, אבל אין חלוקה לקבצים, בדומה למה שרשמתי לעיל, הייתי מחלק לhtml,js,css. |
| תבניות עיצוב | לא הצלחתי לזהות תבנית עיצוב כלשהי… |
| OOP | לא הצלחתי לזהות עקרונות של תכנות מונחה עצמים, אולי הייתי מציע לעבוד עם מחלקות ברורות יותר של עצמים והורשה בינהם במקרה ויש צורך |
| באגים וטעויות | כן, בפנאל מנהל ניכר שבעת מחיקת שאלה לא קורה כלום. השאלה עדיין נמצאת במאגר. |
| טיפול בשגיאות | כן, לדוגמא שמנסים להיכנס עם שם משתמש או סיסמא לא נכונים מופיע חיווי למשתמש כי המשתמש לא נמצא. מבחינת דיבוג הקוד היה קצת קשה כי הקוד לא כתוב בצורה הכי מסודרת וקשה לעקוב אחר הדברים בפועל. |
| בדיקות | בדיקה אם leadership board מתעדכן |
| שימושיות | כן, הממשק ידידותי למשתמש, צבעוני ומושך את העין, יש עניין לחקור את התוכנית ולראות את כל האפשרויות שהיא מציעה. |
| תיעוד | ברור באילו מסכים אני נמצא בכל רגע ואיזה פעולות צפויות לקרות בעת לחיצה על כפתורים אם זאת הכוונה.  אם הכוונה היא לקוד, אין תיעוד נרחב. |
| אתיקה | לא. |
| אבטחה | לא. |
| ביצועים | תלוי בשינוי ומי עושה אותו, אני מניח ששינוי ע"י מתכנת שמכיר את הקוד לא יפגע בו, אבל מישהו שלא מכיר את הקוד עלול לפגוע בו ולהרוס אותו. |
| ביצועים | כן, חלוקה לקבצים תשפר את ביצועי המערכת, בנוסף צמצום אינטרקציות עם שירותי הדאטה בייס יועילו. |
| קריאות - readability | הקוד באופן יחסי מובן הודות לשמות המשתנים ושמות המטודות. אך הקוד מחולק לחלקים רצופים ולא מחולק לקבצים מה שמקשה על הקריאה. על מנת לשפר הייתי מחלק לקבצים. |
| סעיף פתוח - מימוש | Is a framework, API, library, or service used that should not be used?  בהתבסס על הקוד, לא נראה שיש שימוש שלא לצורך בפריימוורקס, API או ספריות. |
| סעיף פתוח- שימושיות | Is the API/UI intuitive to use?  כן הממשק אינטואיטיבי לשימוש, ברור באיזה מסך אני נמצא ואילו פעולות צפויות להתבצע בלחיצה על כפתור מסויים. |

שאלות פתוחות:

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

לעבור על קוד שלא אני כתבתי זו משימה לא פשוטה. אבל במהלך המעבר על הקוד היה נחמד לגלות דרכים וגישות שונות משלנו ליצירת משחק.

*2.* האם לדעתך ניתן לשפר תרגיל זה?

הייתי מציע שלתרגיל זה יפגשו שתי הקבוצות יחד על מנת ליעל את המעבר על הקוד, ולהציע שיפורים בפורום מצומצם. בכך המעבר על הקוד יהיה יותר ברור ויותר מהיר. הייתי מוסיף עבור כל קבוצה אנשי קשר על מנת שיהיה יותר פשוט ליצור איתם קשר.

*3.* הערות נוספות

בסיכום היה מלמד ומהנה לעבור וללמוד את הקוד של הקבוצה השניה. באופן כללי היה קורס מאוד מעניין ומספק מבחינה לימודית. נהנתי ללמוד על טכנולוגיות ענן, ולהשתמש בכלים הקשורים בענן ליישום משחק מחשב.

שם הסטודנט: **דניאל ממן**

|  |  |
| --- | --- |
| קריטריון | הסבר |
| מימוש | הקוד מבצע את הנדרש הוא מבצע את הפעולות בזמן חישוב יעיל וטוב. הוא מאפשר גישה למסד נתונים ושולף מידע כנדרש. בנוסף הקוד מודלרי וקריא. |
| יעילות | הקוד מודולרי וקריא אך כמוהבן שניתן לשפר אותו, להוסיף תיעוד רחב ואלוי לפצל את הפונקציות המרכזיות קצת יותר. אך בסה"כ הקוד נראה טוב, מובן ויעיל |
| פשטות | ניתן לפשט קצת את החלק של ה process data בכך להשתמש ב case ולא עם הרבה if |
| מודולריות | הקוד מודלורי, הוא מחולק לפונקציות מרכזיות ובנוסף הקולאב מחולק לפי נושאים מרכזיים |
| תבניות עיצוב | הפרדת העיצוב הלוגיה והמסכים - שכבות |
| OOP | לא נתקלתי בשימוש מסוים ב OOP של מחלקות עם מידע, פרט לשימוש בפונקציות אשר ממשמות כל חלק בפרוייקט |
| באגים וטעויות | לא ניתן למחוק שאלה, עדיין בDB |
| טיפול בשגיאות | כן, לא מקבלים חיווי על מילוי שדה ריק ואין הודעה ברורה עבור כניסה עם שם משתמש וסיסמא לא נכונים |
| בדיקות | היתי בודק האם שאלה נוספה / נמחקה מהמסד |
| שימושיות | הממשק מאוד שימושי, עוזר למשתמש ונוח להבנה. משאיר חשק להמשיך לשחק ולחזור בהמשך |
| תיעוד | ישנו תיעוד מימנלי, הייתי מוסיף עוד קצת עבור פרטי מידע חשובים או מסורבלים אשר שווים הסבר קטן |
| אתיקה | לא, הכללים הנשמרים. |
| אבטחה | לא, המידע שמור הצורה טובה ולא גלוי לעיניי. |
| ביצועים | תלוי מה מטרת השינוי, יש קודים שרגישים לשינוי יותר ויש פה פחות לכן תלוי מה משתנה ואיך, אך סהכ הקוד נראה אפשרי לשינוי ולא רגיש מידי |
| ביצועים | כן חלוקה לקבצים וביצוע refactor תוכל לשפר את הקוד. כמובן שביצוע אופטימזציה אך הדבר לא נדרש ברמת הקורס הנוכחית. |
| קריאות - readability | הקוד ארוך ולא מחולק בצורה הכי יעילה לקבצים, לא היה הכי קל לעקוב אך הדבר אפשרי |
| scalability | הייתי מנסה לבצע עומס על הקוד והמערכת של כניסת מתשמשים ומידע על מנת לבדוק כיצד היא מתפקדת בקצה והאם ישנם שגיאות |
| בדיקת חומרה | הייתי מנסה להריץ את הפרוייקט על מחשבים שונים עם חומרה משתנה - חזקה חלשה וכדו. בכך לראות האם יש תלות במערכת בעת שיחוק במשחק |

**שאלות פתוחות:**

*1.* כיצד הרגשת במהלך החוויה של מעבר על הקוד?  
2*.* האם לדעתך ניתן לשפר תרגיל זה?  
3*.* הערות נוספות

**תשובות:**  
1) החוויה במעבר על קוד אחר הייתה מעניינת אומנם הדבר לא חדש עבורי עקב מצבים לאורך התואר בהם ניסיי לעזור לחבר למצוא שגיאה אך הדבר היה מאתגר יותר שכן אף אחד לא הנחה אותי מה מתבצע בכל שורה / קובץ ולכן הייתי צריך לבד להבין הכל. היה מאתגר ולא פשוט, ובעיקר במצבים כאלה אתה מבין את חשיבותו של קוד קריא ומתועד.

2) תרגיל זה ניתן לשפר בכך שאם היינו מקבלים את התיק מתכנת שלהם היה עבורנו הרבה יותר ברור, נוח וקל להבין מה מתרחש בפרוייקט. בנוסף אולי מפגש עם הקבוצה או אפילו לתאם פגישה מעמיקה בשיעור על בסיס החלוקה של הקבוצות והסטודנטים.

3) כאשר היינו צריכים לחפש את האנשים אשר משתייכים לקבוצה (pacaton) היה קשה למצוא אותם מבלי לדעת מי חברי הקבוצה, ניסנו לשלוח הודעות בווטסאפ ובמייל אך ללא הצלחה לקח זמן רב. היה מומלץ לתת איש קשר עבור כל קבוצה על מנת שנוכל ליצור שיח ראשוני (כדי לקבל את הגוגל קולאב והרשאות לקריאת הקוד)

שם הסטודנט: **בן אבנר מריג'ן**

|  |  |
| --- | --- |
| קריטריון | הסבר |
| **מימוש** | ניכר כי הקוד מבצע את הנדרש ושהיתה מחשבה והשקעה של חברי הצוות. המשחק ממלא כמעט את כלל הדרישות שהתבקשנו. |
| **יעילות** | כן, נשים לב שיש נקודות בהן ניתן לשפר את הקוד. ישנם מספר גישות למסד נתונים, ניתן לייעל על שליפת הנתונים פעם אחת ואחסון במשתנה, css מוחל כל פעם שהפונקציה rankshop נקראת. |
| **פשטות** | ניתן להשתמש בשמות משתנים משמעותיים יותר על מנת שיהיו מובנים יותר, אפשר לאחד במחלקה משותפת לסגנונות ואלמנטים בCSS.  ישנם פונקציות נפרדות שניתן לשלב לכדי פונקציה אחת ובכך לבטל את הצורך בהעברת הנתונים ביניהן ולהפשיט את הקוד. |
| **מודולריות** | יש חלוקה תקינה לקבצים, הייתי יותר קובץ CSS חיצוני, ישנם פונקציות ארוכות מאוד לכן ניתן להוסיף פונקציות עזר ובכך הקוד יהיה גם קריא יותר. |
| **תבניות עיצוב** | לא ראיתי יישום ברור ומפורש של דפוס עיצוב כלשהו. |
| **OOP** | אין שימוש ב-OOP. |
| **באגים וטעויות** | כאשר המשתמש מחליט לצאת במהלך המשחק, מוצגת הודעת שגיאה בהצגת גרף ההתקדמות וריבוע לבן שאמור להציג את מספר השאלות שנענו נכון ריק. בנוסף, כאשר לוחצים על כפתור מחיקת שאלה בפאנל מנהל אין תגובה מהמערכת (אין הודעה מתאימה והשאלה לא נמחקת). |
| **טיפול בשגיאות** | שמים לב כי היתה חשיבה על מקרים בהם יכלו להתקיים שגיאות משתמש בפאנל מנהל, לא ניתן להוסיף שאלה כאשר יש שדות ריקים וגם מופיעה הודעה מתאימה.  לעומת זאת בהרשמה, כאשר נלחץ הרשמה עם שדות ריקים אין הודעה שיש צורך למלא את כלל השדות. בנוסף, אין הודעה שהצלחנו להירשם בהצלחה.  במקומות בהם כן יש הודעות משתמש, לדעתי הן ברורות ומובנות.  נוח היה לבדוק את הקוד. |
| **בדיקות** | יש צורך לבדוק מה קורה אם המשתמש מחליט לצאת במהלך המשחק ובדיקה כאשר רוצים למחוק שאלה ממסד נתונים. |
| **שימושיות** | כן הממשק שימושי ומעורר רצון להמשיך לשחק, בזכות זה שבמהלך המשחק אתה מקבל אינדיקציה למספר השאלות שענית נכון ולא נכון ובכך אתה שואף להוציא יותר בכל משחק.  בנוסף, הטבלה אשר מציגה את 5 המשתמשים המובילים באתר יוצרת תחרותיות אשר גורמת להתמיד, לשחק ולהשתפר על מנת להיות במקומות הראשונים. |
| **תיעוד** | אין תיעוד. |
| **אתיקה** | על סמך הקוד ניכר כי לא כללו כל עיבוד נתונים, אינטראקציה עם משתמשים או מניפולציה של נתונים שעלולים להעלות חששות הקשורים להפרות פרטיות, ניצול אנושי או הטיות אלגוריתמיות. |
| **אבטחה** | הממשק מאובטח. אין שמות משתמש או סיסמאות גלויים שעלולים להוות סיכון אבטחתי. |
| **ביצועים** | שינוי עתידי לא אמור לגרום לביצועים של הקוד. |
| **ביצועים** | הייתי משנה את הצורך לגשת כל פעם אל הHTML. |
| **קריאות - readability** | הקוד מובן ומסודר דבר שהקל בניתוח שלו. בכל זאת, הוספת תיעוד לפונקציות שמסביר את מטרתן או בשורות קוד קריטיות היה משפר את הקריאות של הקוד. |
| **אלמנטים משחקיים** | "האם ניתן לשפר אלמנטים משחקיים ?"  הייתי משפר את הגרף המוצג בסיוג המסמך לגרף שמציג השתפרות לאורך כלל השימוש במערכת ולאו דווקא סטטיסטיקה עבור משחק אחד |
| **שילוביות AI** | "האם בעתיד יהיה אפשר לעשות שימוש ב-AI ?"  לדעתי במשחק ניתן יהיה להוסיף סיווג רמות קושי לשאלות וככה במהלך המשחק נאפיין את רמת המשתמש ואפשר יהיה להתאים את רמת השאלות לרמתו העדכנית. |

שאלות פתוחות:

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

במהלך החוויה של מעבר על הקוד הרגשתי תהליך למידה, הקוד היה מסודר וקריא לכן קל היה לנסות לעקוב ולהבין אותו. דבר זה גרם לי להבין את היתרונות והחסרונות בגישה שלנו. מעצם החקירה של קוד שלא הכרתי אני לומד להכיר שיטות חדשות ויצירתיות לאותה בעיה.

*2.* האם לדעתך ניתן לשפר תרגיל זה?

כן, אם היתה דרישה לשלוח בין הקבוצות הבודקות והנבדקות מלבד קישור ל-COLAB גם את תיקי המשתמש והמתכנת של כל קבוצה הדבר היה מקל ומזרז את תהליך ניתוח הקוד. בנוסף, הדבר היה מעניק בסיס טוב יותר והבנה של הדרך שעברה הקבוצה הנבדקת בדרך לכתיבת המשחק ועל מה בחרו לשים דגש.

*3.* הערות נוספות

התרגיל היה חוויתי מאוד ומפתח. בהמשך למתואר בשאלה 2, בעתיד אני חושב כדאי יהיה לקיים שיעור בדומה למה שעשינו בשיעור סטודיו בו יהיו צריכים להגיע הקבוצות הנבדקות והבודקות ביחד - דבר זה יקל על חיפוש אנשי הקבוצה בווצאפ (לא תמיד כולם זמינים ולכן לפעמים לקבוצות בודקות אין את אותו הזמן לבדוק קוד של קבוצה אחרת). לסיכום, נהנתי לבצע תרגיל זה בו הגענו למוצר סופי ועבדנו טוב בצוות, וניתחנו קוד של קבוצת pacathon דבר שגרם לנו להעשיר את הידע.