מחשוב ענן – תרגיל בית 3 חלק ב

שם קבוצה: Zebra

קישור לגיט: <https://github.com/audiblemaple/collab_test>

חברי הקבוצה:

|  |  |
| --- | --- |
| גלעד סגל | 314865841 - QA |
| כפיר חמו | 318262151 - Scrum Master |
| ניצן ממן | 205798515 - UI |
| רון שחר | 206755712 - Backend Developer |
| דניאל ורדימון | 209289685 - Product Manager |
| ליאור ז׳יגלו | 316109115 - Frontend Developer |

**חלק ראשון: מענה קבוצתי – ביצוע בדיקות קבלה עבור קבוצת chameleon:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Actual Result** | **Expected Result** | **Description** | **Test ID** |
| As Expected Result - PASS | The user moves to his home screen (player or manager). | User enters his username and password, then presses "OK". | User logs in to the game - 1 |
| As Expected Result - PASS | The user gets a message that his username/password are incorrect | User enters his username and password, then presses "OK". | User logs in to the game - 2 |
| a. As Expected Result – PASS  b. There is a message to specify #question\_number out of #total\_questions\_number in each question. No progress bar. PASS  Note: consider adding progress bar.  c. As Expected Result – PASS  d. As Expected Result – PASS | 1. The user chose single answer for each question before moving to the next question. 2. Each time the player moved to another question the progress bar of his correct/incorrect answers changed accordingly. **Irrelevant** 3. In the end of the quiz, a summary chart is presented with statistics about his performance. 4. The player returned to his home screen after summary screen. | Player presses the "Play" button.  Then he chooses topic of python and amount of 20 questions. He answers all the questions and then moves to a summary screen of his performance in the recent game. | Player's home screen – 1 |
| As Expected Result – PASS | The player now sees his best score in the game and his recent score in the last game. | Player presses the "My Progress" button. | Player's home screen – 2 |
| There is "Help" button PASS | The player moves to an info screen about the lore of the game. | Player presses the "?" button. | Player's home screen – 3 |
| There is "Logout" button PASS | The player exists the game entirely. | Player presses the "X" button. | Player's home screen – 4 |
|  | The Player is presented with a message from the local "Boss" and moves to Python questions screen. **Irrelevant** | Player presses topic of "Python" in quizzes menu screen. | Quiz menu - 1 |
|  | The Player is presented with a message from the local "Boss" and moves to DevOps questions screen. **Irrelevant** | Player presses topic of "DevOps" in quizzes menu screen. | Quiz menu – 2 |
|  | The Player is presented with a message from the local "Boss" and moves to JSON questions screen. **Irrelevant** | Player presses topic of "JSON" in quizzes menu screen. | Quiz menu – 3 |
|  | The Player is presented with a message from the local "Boss" and moves to XML questions screen. **Irrelevant** | Player presses topic of "XML" in quizzes menu screen. | Quiz menu – 4 |
|  | The player moves to an info screen about the lore of the game. **Irrelevant** | Player presses the "?" island in center of the map image of quizzes menu screen. | Quiz menu – 5 |
| PASS | The player exists the game entirely. | Player presses the "X" button. | Info screen - 1 |
| a. appears "Correct, Well Done :)" PASS  b. no progress bar, but there is #question\_number out of #total\_questions\_number in each question.  PASS  Note: consider adding progress bar.  c. No total score update FAIL  d. As Expected Result – PASS | 1. A message of "You noob LOL " appears in the screen. **Irrelevant** 2. In the progress bar, the question's cell is updated with color green. **Irrelevant** 3. The total score is incremented. 4. The Player continues to answer questions. | Player presses wrong answer. | Player run quiz - 1 |
| a. appears " Worng, Try Again Next Time :( " PASS  b. no progress bar, but there is #question\_number out of #total\_questions\_number in each question.  PASS  c. No total score update FAIL  d. As Expected Result – PASS | 1. A message of "good job" appears in the screen. 2. In the progress bar, the question's cell is updated with color red. 3. The total score stays the same. 4. The Player continues to answer questions. | Player presses correct answer | Player run quiz - 2 |
| "End Game" button in every question. PASS | The player returns to Quiz Menu screen. | Player presses "🡨" button | Player run quiz - 3 |
| a. Not an ADMIN tool, any user can register. PASS  b. press "Create Account" and new user has been added or press "Back" to return.  PASS | 1. The manager moves to a new screen to add new user's info: Username, Password and choose manager status (Yes or No). 2. The manager presses "Add User" to add new user or presses "Back" to return to his home screen. | The Manager presses "New User" button in manager home screen. | Manager tools – 1 |
| Note:  Consider adding an option for user to update his own info in the system (or make it an admin tool). | The manager moves to a new screen to write the username and the user's info: new username, new password and choose manager status (Yes or No). **Irrelevant**  The manager presses "Update User" to update user's info or presses "Back" to return to his home screen. **Irrelevant** | The Manager presses "Update User" button in manager home screen. | Manager tools – 2 |
| Note:  Consider adding an option for user to delete his own info in the system (or make it an admin tool). | The manager moves to a new screen to write the user's info: username and password for deletion. **Irrelevant**  The manager presses "Delete User" to delete user or presses "Back" to return to his home screen. **Irrelevant** | The Manager presses "Delete User" button in manager home screen. | Manager tools – 3 |
| No 4 types of quizzes but still can add new question or go back. PASS | The manager moves to a new screen to add new question's info: question header, subject (Python, XML, JSON or devops), 4 answer options, question image, correct answer, question level and type of question.  The manager presses "Add Question" to add new question or presses "Back" to return to his home screen. | The Manager presses "New Question" button in manager home screen. | Manager tools – 4 |
| The "update question" option is very good and based on combo box rather than inserting question header. After choosing question, you move to question info screen that enable edit or delete.  PASS | The manager moves to a new screen to update question's info.  The manager inserts question's header as primary key and then fill out all the question's info: updated question header, updated subject (Python, XML, JSON or devops), 4 answer options, question image, correct answer, question level and type of question.  The manager presses "Update Question" to update question's info or presses "Back" to return to his home screen. | The Manager presses "Update Question" button in manager home screen. | Manager tools – 5 |
| The "delete question" option is very good and based on combo box rather than inserting question header. After choosing question. you move to question info screen that enable edit or delete.  PASS | The manager moves to a new screen to write the question's header and subject for deletion.  The manager presses "Delete Question" to delete question or presses "Back" to return to his home screen. | The Manager presses "Delete Question " button in manager home screen. | Manager tools – 6 |
| The admin returns to login screen.  PASS | The manager moves to Player's home screen. | The Manager presses "LogOut" button in manager home screen. | Manager tools – 7 |
| Table instead of a graph  PASS | The user moves to a new screen with summery graph of all his games scores. | User choose to check his total scores from his entire game history. The player chooses this option in the main player's screen. | User chooses "My Scores" button |
| There are videos (links to YouTube) for learning cloud computing concepts. | The user is shown with a short clip of funny credits for the game developers. **Irrelevant** | In the quiz map screen, the user chooses the credit button next to the "X" | User chooses "CREDITS" button |

**חלק שני – עבודה אישית**

שם הסטודנט: גלעד סגל

|  |  |
| --- | --- |
| **קריטריון** | **הסבר** |
| מימוש | הקוד מיישם את הפונקציונליות הרצויה בצורה מדויקת, עומד בדרישות הקורס ומייצר את ההתנהגות הצפויה כפי שחברי הצוות שלו התכוונו שיקרה. |
| יעילות | הקוד מתבצע בזמן ריצה טוב, ללא חישובים מיותרים ובצריכת זיכרון הגיונית לדרישות הקורס. המפתחים גם יצרו פונקציות לא ארוכות אשר לא מכילות כפילויות מבחינת שורות קוד. המפתחים שמו לב לכתיבה יעילה למסד הנתונים (ללא קריאת כמות גדולה של נתונים שלא לצורך). |
| פשטות | הקוד כתוב בצורה תמציתית, ללא חישובים מיותרים או יצירת תנאים לוגים מסובכים. שמות המשתנים והתכונות בעיצוב הhtml כתובים בצורה ברורה אשר מאפשרת הבנה מהירה של הקוד והמימוש שלו. |
| מודולריות | הקוד מודולרי מהבחינה הזו שהמפתחים בקבוצה זו דאגו לסווג את הקוד שלהם לבלוקים שונים של colab לפי הייעוד שלהם. בתוך כך, הם ריכזו את כל עיצוב ה-front שלהם בבלוק אחד, קריאה לפונקציות שמבצעות קריאה וכתיבה ל-DB בבלוק אחר וכו'. הבלוק האחרון מהווה רק קריאה לכלל הפרויקט שלהם בצורה קצרה ותמציתית. |
| תבניות עיצוב | בסקירה של הקוד לא עלו תבניות עיצוב מסוימות. |
| OOP | מבחינת עקרונות של OOP, אין יותר מדי חלוקה למחלקות אלא בעיקר שימוש בפונקציות. מצד שני, עיצוב הממשק מוגדר כראוי ותוך ביצוע הפרדה ברורה בין מסכים שונים. בנוסף קיימת מודולריות והישענות על שימוש חוזר ברכיבים שונים. |
| באגים וטעויות | בריצת המערכת לא התגלו באגים או טעויות משמעותיות אשר מבטאות טעות לוגית חמורה בהקשר מימוש הפרויקט. |
| טיפול בשגיאות | המפתחים מדגימים הצגת מחוון למשתמשים במקרה של התחברות לא תקינה למערכת. מפתחי הקוד דאגו לכתיבת קוד מסודר ומחולק לפונקציות רבות, כך שיהיה נהיר במקרה של חיפוש באגים שעלולים להתגלע במהלך הפיתוח. |
| בדיקות | הפרויקט של הקבוצה הנתונה עבר את מרבית הבדיקות קבלה שיצרנו, למעט אחת. המפתחים לא אפשרו לשחקן לראות את סכום הנקודות שלו מתעדכן במהלך המשחק. |
| שימושיות | המשחק שימושי בכך שעונה על המטרה העיקרית שלו – לימוד מושגים של מחשוב ענן בדרך משחקית ועם עניין. |
| תיעוד | הקוד מתועד בצורה טובה. עבור כל פונקציה יש תיעוד קצר על מטרתה ועבור כל עיצוב מסך יש כותרת על איזה מסך מדובר. |
| אתיקה | המפתחים מדגימים הגנה על נתוני המשתמשים. המפתחים מאחסנים בבטחה את פרטי ההתחברות של המשתמשים שלהם במסד נתונים ללא אפשרות לגישה חיצונית של גורם זר לפיתוח המערכת. על ידי כך, הם מאפשרים לכל משתמש להיות הצופה היחיד על נתוני ההתקדמות שלו במשחק. |
| אבטחה | אני לא רואה סיבה לבצע פרצת אבטחה לקוד של קורס במחשוב ענן ובהתאם לכך, מסקירת הקוד של הקבוצה הנתונה אין נקיטת פעולות ברורות למנוע פרצות אבטחה. |
| ביצועים 1 | לפי דעתי שינוי בקוד לא בהכרח יגרום לשינוי בביצועים מפני שהקוד מאורגן בצורה טובה כך שאין משתנים או פונקציות שהשימוש בהם הוא חסר מחשבה. בהכנסת מאפיינים נוספים למערכת, המפתחים יכולים לעבו |
| ביצועים 2 | ניתן תמיד לייעל קוד קיים. בהקשר הפרויקט של הקבוצה הנבדקת ניתן היה להכניס קצת יותר עקרונות של OOP כמו שימוש נכון במחלקות, הורשה ופולימורפיזם. בנוסף, ניתן להשתמש במבני נתונים או אלגוריתמים שונים כדרך לייעל את היישום הקיים. כלומר, לחפש חלופות ליישום הקיים אבל בדרך של מבנה נתונים שיכסה על פונקציונליות מסוימת. |
| קריאות | הקוד קל לקריאה ולהבנה. המתכנתים דאגו לשמירה על מוסכמות עיצוב קוד כגון הזחה, אורך שורה ושימוש נכון ברווחים. הקוד מעוצב היטב ולכן משפר קריאתו ומקל על הבנת המבנה והזרימה של הקוד. קיימת בחירה נכונה של שמות משתנים, פונקציות וכד'. |
| שקיפות ואחריות | בסיס הקוד והמימוש שלו מתקיים דרך פלטפורמת פיתוח של סביבת ענן (Google Colab) כך שהמפתחים יכולים בקלות לחשוף את התנהגות הקוד שלהם לכל גורם אשר מעוניין לצפות במהלך עבודתם. בנוסף, קיימת אחריות בצורה של ליווי צמוד לתכני הקורס ולדרישותיו באופן שלא גורם ליצירה של פרויקט בעל חששות פוטנציאליים כגון בעיות אתיות וכד'. |
| עמידה בדרישות משפטיות ורגולטוריות | המפתחים יצרו את הקוד תחת מסגרת קורס במכללה אקדמית. באופן זה, קיימת חשיפה מרכזית לכלים ורכיבים אשר עומדים תחת דרישות רגולטוריות שונות ולא חורגים מזכויות יוצרים פרטיים, רישוי או כל תקנות אחרות. עמידה במסגרות חוקיות עוזרת להבטיח התנהגות אתית ושימוש אחראי בטכנולוגיה. |

1. כיצד הרגשת במהלך החוויה של מעבר הקוד?

אני בעיקר מרגיש שביעות רצון בכך שאני מסייע לתרום לשיפור קוד קיים ובכך להפוך אותו ליותר מתוחזק וברור. מציאה של הבדלים בין הפרויקט שלנו לבין הפרויקט הנבדק גם מסייעת לי באופן אישי בכך קודם כל חושפת אותי לדרכים שונות למימוש, הן בסגנון חלוקת הפונקציונליות והן בסגנון העיצוב הכללי. בהתאם לכך, אני לוקח את ההזדמנות הזו לבצע סקירת קוד כדרך לחקור גישות שונות שלא חשבתי עליהן בעצמי או במסגרת הצוות שאני שייך אליו.

1. האם לדעתך ניתן לשפר תרגיל זה?

אני חושב שהיה הרבה יותר מועיל לבצע ראיונות בנושאים שונים של סקירת קוד עם הקבוצה הנבדקת. במקום לבצע את סקירת הקוד בצורה עצמאית וללא תקשורת רציפה עם הקבוצה הנבדקת, היה אולי עדיף לדון ביישום שלו עם הקבוצה הנבדקת, לשאול שאלות בצורה ישירה ולקבל תשובה מיידית. בצורה זו, אולי המשימה תקבל גוון יותר פרקטי במובן של שיתוף פעולה בין צוותי פיתוח שונים.

1. הערות נוספות

אין.

שם הסטודנט: ליאור ז'יגלו

|  |  |
| --- | --- |
| **קריטריון** | **הסבר** |
| מימוש | כן, הקוד אכן מבצע את הנדרש ועובד. |
| יעילות | הקוד מחולק לבלוקים של קולאב, שהרצה של כולם לפי הסדר גורמת לריצה של הקוד כשמריצים את הבלוק האחרון. גישה זו לדעתי אינה אינטואיטיבית, "מבולגנת" ומקשה על תחזוקה של הקוד. |
| פשטות | תמיד ניתן לשפר את הקוד, אך לא אוכל להצביע על משהו ספציפי כי איני מכיר את הקוד לעומק. במעבר על הקוד, נראה שהקוד יחסית פשוט. |
| מודולריות | הקוד מחולק לבלוקים של קולאב ולפונקציות בעלות שימושים ספציפיים.  לכן נראה שהוא אכן מודולרי ובמקרים שיידרש שינוי, יהיה אפשר לעשות את זה יחסית בקלות. |
| תבניות עיצוב | לא נראה שיש שימוש בתבניות עיצוב.  הקוד הפונקציונלי מורכב מרצף של קריאות לפונקציות. |
| OOP | לא נראה שיש שימוש בקלאסים, אולי רק כאשר אוספים מידע בפורמט JSON ע"מ לשלוח ל FIREBASE. |
| באגים וטעויות | 1. כן, כאשר לוחצים על LOGOUT וישר על START אז המשחק עדיין מתחיל.  (זה הוא "באג" קטנוני אשר דורש תכנות אסינכרוני נכון מה שלא היה בדרישות)  2. כאשר ממלאים username או password לא נכונים אז מודפסות הודעות והם לא נעלמות אז כמות הפעמים שלוחצים על CONNECT זו כמות ההודעות. |
| טיפול בשגיאות | אין שימוש ב try \ except או משהו כזה, כמו כן, לא נראה שיש יותר מדי בדיקות לערכים לא תקינים חוץ מ username \ password.  הודעות למשתמשים ברורות, איפה שיש. |
| בדיקות | יש הרבה מקומות לביצוע בדיקות.  לדוגמה: בפונקציה CREATERANKUSERS, יש בדיקה אם השחקן הוא ADMIN.  אז אין בדיקה אם הערך שקיבלנו לא תקין. |
| שימושיות | הממשק שימושי, והצבעים תואמים לקטנספט המשחוק.  יש רצון להמשיך לשחק. |
| תיעוד | הממשק מתועד בכל פונקציה ומוסבר השימוש של כל אחת. |
| אתיקה | לא מצאתי. |
| אבטחה | כן, כשנכנסים למסך הראשי, מוצג שם המשתמש. |
| ביצועים 1 | קשה לדבר על ביצועים בפלטפורמת גוגל קולאב, אך כמובן יתכן שהוספת אלמנטים תפגע בביצועים. |
| ביצועים 2 | לא, גוגל קולאב ידוע בביצועים יחסית נמוכים בגרסא החינמית. כמו כן אם נריץ את הקוד בפלטפורמה אחרת אז הקוד ירוץ מהר מאוד. |
| קריאות | הקוד קריא ושמות משתנים ברורים. ניתן לשנות את ההזחות ל 4 רווחים במקום 2 ע"מ שיהיה יותר קל לקרוא את הקוד. בנוסף להוסיף רווחים בין שורות. |
| אחידות מסכים | ישנם מסכים גבוהים יותר וישנם מסכים נמוכים יותר, זה מרגיש קצת מוזר כאשר מנווטים במשחק כי יש הרגשה של חוסר אחידות קצת. |
| שימוש בכלים הנתונים | הקוד מציג הבנה בכלים הנתונים (google colab), ומראה שימוש ב kernel של הפלטפורמה ע"מ לתקשר עם בלוקים שונים. הבנה של היכולות של הפלפורמה אפשרה לקבוצה לממש את הרעיון שלהם באופן שבו הם ראו אותו. |

1. כיצד הרגשת במהלך החוויה של מעבר הקוד?

אף על פי שזה תמיד מעניין לעבור על קוד של מישהו אחר ולראות דרכי חשיבה של מתכנתים אחרים, לא נהנתי מהמשימה. היו לא מעט משימות במהלך הקורס ומשימה זו מרגישה מיותרת, במיוחד להיקף קורס של 3 נק"ז שגם ככה היה מאוד עמוס.

1. האם לדעתך ניתן לשפר תרגיל זה?

תלוי, אם לשנות את מבנה הקורס, אולי אפשר להשאיר את התרגיל כמו שהוא.

אך אם בסמסטרים הבאים מבנה הקורס יהיה זהה אז אולי לבטל את תרגיל זה.

1. הערות נוספות

אין.

שם הסטודנט: ניצן ממן

|  |  |
| --- | --- |
| **קריטריון** | **הסבר** |
| מימוש | כן, הקוד מבצע את הפונקציות והלוגיקה הנדרשת לכל אחת מהפונקציות המופיעות בקוד. הוא מאגד פונקציות שונות לפי דפי האפליקציה (דף הבית, דף הניהול, דף המשחק וכו') ומבצע פעולות כמו כניסה ויציאה מהמערכת, הרשמה של משתמש חדש, הוספת שאלה או סריקה של שאלה קיימת, עדכון נתונים במסד הנתונים, טעינת מידע לדפים השונים, וכו'. הוא מקושר למסד הנתונים Firebase כדי לקבל ולעדכן את המידע הרלוונטי למשתמשים ולשאלות. |
| יעילות | ניתן לשפר את היעילות לדוגמא: 1) שימוש בפונקציות/מבנים במקום משתנים גלובליים 2) במקום כל פעם לבצע clear\_output() ולטעון מחדש דף וכל האובייקטים עליו, ניתן לשנות\למחוק את האובייקטים הרלוונטים 3) השורה : resQuestion=FBconn.get('/questions/',None) נקראת סתם 6 פעמים במקום לשמור את השאלות שנשלפות מהDB בפעם אחת. |
| פשטות | ניתן לשפר את הפשטות לדוגמא: 1) שימוש בפונקציות/מבנים במקום משתנים גלובליים 2) במקום ליצור הרבה עמודי HTML ניתן ליצור עמוד בסיס ולבצע שינויים דינמים עליו. בנוסף לאגד css במקום לכתוב לאותם אובייקטים בכל page בנפרד. |
| מודולריות | הקוד מציג מודולריות במידה והוא מחולק לפונקציות ומשתנים נפרדים עבור כל חלק מהתוכנית. ישנם פונקציות המבצעות פעולות שונות כמו תצוגת עמודים, עדכון מידע במסד הנתונים, טעינת שאלות וכו'. הפונקציות מתחלקות לקטגוריות שונות על פי המטרה שלהן, כך שקל ונוח לנווט בקוד ולתחזקו בעתיד. |
| תבניות עיצוב | לא נמצא יישום תבנית עיצוב ספציפית. הקוד מורכב בעיקר מפונקציות ומשתנים גלובליים לניהול דפים ופעולות שונות. עם זאת, דפוסי עיצוב לא תמיד מיושמים באופן מפורש ויכולים להופיע באופן טבעי בהתבסס על יחסים וקשרי גומלין בין מחלקות או אובייקטים. |
| OOP | כמו שהסברתי בתבניות עיצוב הקוד מורכב בעיקר מפונקציות ומשתנים גלובליים לניהול דפים ופעולות שונות. גישה זו טובה ,אך, אינה מכניסה נתונים והתנהגות למחלקות ואובייקטים.  הצעות לשיפור: 1) יצירת מחלקות ואובייקטים שייצגו את המשחק ויכללו את הנתונים שלו ופונקציות כמו הרשמה, התחברות וכו.. 2) שימוש במשתני מופע במקום במשתנים גלובליים. 3) שימוש בירושה ופולימורפיזם כדי לייצג סוגי דפים שונים (למשל, דף בית, דף ניהול, דף קולנוע). כל עמוד יכול להיות תת מחלקה של מחלקת עמוד בסיס. |
| באגים וטעויות | ) בעת מספר נסיונות כניסה הודעות השגיאה נערמות:  2) למשתמשים חדשים לא מתעדכן הscore בleader table: |
| טיפול בשגיאות | האם ישנה התראה על שגיאות משתמש?  כן, בlogin ובsignup מוצגות הודעות שגיאה כאשר לא ממלאים את כל הinputs הנדרשים. האם ישנו שימוש במנגנון תפיסת שגיאות?  אין תפיסת Exception או בtry ו catch אך יש פשוט בדיקות if לערכים ואיך נפעל בהתאם. האם הבדיקה ודיבוג הקוד נוחים?  הבדיקה די אינטואיטיבית למרות שבקולאב לא הכי נוח לדאבג... האם הודעות המשתמש ברורות וידידותיות למשתמש? כן ההודעה מסבירה בדיוק את הבעיה ואיך עליו לפעול. |
| בדיקות | כנראה הוספת בדיקות אשר יתקנו את הבאגים שנמצאו כמו אם קיים כבר הודעת שגיאה בLOGIN לא להוסיף חדשה. |
| שימושיות | האם הממשק שימושי? (usability)? מאוד הוא מינימלי, מובן וניתן ללמוד אותו בקלות ולשחק חלק כבר בכניסה הראשונה.  האם המשחק מעורר רצון להמשיך לשחק? כן, המשחק מהיר קליל לא צורך הרבה זמן ומאמץ, בנוסף ישנו טריגר של תחרות המושך אותך לשחק בשביל להגיע גבוהה בleaderboard. |
| תיעוד | כן, בעת כניסה למשחק ישנו הסבר ראשוני שישר מכניס אותך לעניינים. בנוסף, ישנם 2 דפי help אחד בדף הבית לאחר כניסה המסביר במפורט את כל הממשק המשחקי (איך לשחק ואיך לצבור נקודות). השני בדף הcinema המסביר על אפשרות הלמידה. |
| אתיקה | האם יש שימוש בנתוני משתמש אשר מפרים פרטיותו? לא, החיבור למשחק מצריך רק משתמש וסיסמא ואין אפשרות להפרת פרטיות מסוג כל שהיא. האם יש ניצול של התנהגות או חולשה אנושית? לא, המשחק בא כדי לדרבן ולהיות כעזר ללמידה ולא למען ניצול ו\או רווח. האם ישנו algorithmic bias כנגד קבוצה או פרטים מסוימים? לא, אין הטיה אלגוריתמית מאף סוג. המשחק פונה לכולם בצורה שווה ומבוסס על הידע של המשתמש בבחירת תשובות אמריקאיות. |
| אבטחה | האם ישנו מידע אבטחה (שמות משתמש, ססמאות) גלויים? למרות כי אין שום קוד שמציג או חושף מידע רגיש כמו שמות משתמש או סיסמאות. כל המשתמשים מהDB מיובאים לתוך res (משתנה גלובלי שמחזיר את כל המשתמשים) אבל כולנו ביצענו כך לפי דרישת הקורס. |
| ביצועים 1 | ייתכן ששינוי עתידי בקוד עלול לגרום לפגיעה בביצועים כגון: 1) הוספת דפי HTML עמוסים וטעינתם (כי אין שינוי דינמי אלא טעינה חדשה). 2) הוספת שאלות\משתמשים\לינקים יאטו את השליפה מהDB שכן ישנן קריאות מרובות לשאלות. |
| ביצועים 2 | כן, בדומה לשיפור ביעילות הקוד ישתפרו גם ביצועי הקוד: 1) שימוש בפונקציות/מבנים במקום משתנים גלובליים 2) במקום כל פעם לבצע clear\_output() ולטעון מחדש דף וכל האובייקטים עליו, ניתן לשנות\למחוק את האובייקטים הרלוונטים 3) השורה : resQuestion=FBconn.get('/questions/',None) נקראת סתם 6 פעמים במקום לשמור את השאלות שנשלפות מהDB בפעם אחת. |
| קריאות | האם הקוד מובן בקלות? כן, הקוד די מובן בעזרת החלוקה לחלקים: 1) קריאה מDB  2) דפי HTML 3) פונקציות ולוגיקה – מתועדות קלות ובעל כרוניקה מובנת 4) קריאות לdisplay ו- callback אילו חלקים בקוד היו לא ברורים עבורך? הקוד די מובן, המשתנים הגלובלים קצת מקשים, התיעוד הפנים פונקציונלי מועט, שימוש דליל בOOP כיצד ניתן לשפר את קריאות הקוד? ניתן לשפר את הקריאות ואפשרות המעקב אחר קוד כך: 1) שימוש במבנים במקום משתנים גלובלים 2) תיעוד שורות לא טריוואליות בפנים הפונקציות לא רק מה הפונקציה עושה 3) שימוש במחלקות ואובייקטים. |
| עיצוב חוויה | כיצד הממשק מעוצב לחווית משחק? לדעתי המשחק מעוצב באופן גנרי המתאים לכלל המשתמשים ומהווה אווירה קלילה וטובה ללמידה. |
| נגישות | האם למשחק יש גרסת תאימות לבעלי מוגבלויות?  לא, אין תמיכה בנגישות כלל. |

1. כיצד הרגשת במהלך החוויה של מעבר הקוד?

האמת היא שתחילה הרגשתי קצת בילבול וחוסר בהירות מכיוון שהיה לי קצת קשה להיכנס לקוד אחר שלא מתועד ומסוגנן כפי שאני רגיל אליו (מהקודים שכתבתי בעצמי), אך מפאת הסדר ההגיוני של הקוד והסקרנות שלי הצלחתי להיכנס לראש של המתכנתים שכתבו אותו ולהבין אותו.

1. האם לדעתך ניתן לשפר תרגיל זה?

כן, מעבר תחילה על אסטרטגיות גישה לחשיפה לקוד בפעם הראשונה.

1. הערות נוספות

אין.

שם הסטודנט: רון שחר

|  |  |
| --- | --- |
| **קריטריון** | **הסבר** |
| מימוש | כן, הקוד ממש את דרישות הפרוייקט.  אפשר לראות שקיימים יוזרים, מסך התחברות, שאלונים, ניקוד (מוטיב משחוק), אחרי כל משחק יש מסך סיכום משחק הנותן לנו נתונים על זמן משחק וpi chart  בנוסף הם מימשו טבלאת high score שמוסיך לכל מוטיב המשחוק. |
| יעילות | הקוד יחסית יעיל, לדעתי אפשר לשפר את המצב שכל הזמן הם מייבאים מהfirebase במקום לייבא פעם אחת ולשמור את הנתונים בתוך משתנה זמני. |
| פשטות | אפשר לפשט את הקוד בכך שבמקום כל הפונקציות שקוראות לכל מסך אפשר לעשות פונקציה אחת גנרית שקוראת למסכים למשל.  בנוסף יש שימוש רב במשתנים גלובלים דבר היכול לסרבל את הקוד שלהם בעתיד אם הם רוצים להוסיף עוד פונקציונליות. |
| מודולריות | אכן הקוד מודולרי, לדעתי הם עשו עבודה טובה מאד בכך שהם חילקו את כל התוכנית לכמה קבצים שונים, ולבסוף צריך להריץ רק בלוק קוד קטן שמחבר את הכל. אם מסתכלים קרוב מספיק אכן אפשר לעשות את הקוד יותר מודלרי בכך שאפשר לחבר כמה פונקציות דומות, ולהימנע משימוש יתר של משתנים גלובלים. |
| תבניות עיצוב | לא הצלחתי למצוא שימוש של תבניות עיצוב בקוד. |
| OOP | הקוד לא משתמש בתכנות מונחה עצמים |
| באגים וטעויות | העדכון של הנקודות למשתמש בטבלה של הנקודות לא מתבצע כראוי. |
| טיפול בשגיאות | כאשר אני לא רושם פריטים בהתחברות או רישום קופצת לי הודעת שגיאה אבל חוץ מזה אין התייחסות לשגיאות כל כך. |
| בדיקות | הם צריכים לדעתי להוסיף בדיקה הבודקת עדכון נכון של הניקוד בטבלת הניקוד הכללית. בנוסף בדיקה של רישום וכניסת משתמש, הכנסת שאלות וסרטונים למאגר המידע שלהם  אבל כמו שציינתי הכי חשוב לבדוק את loadLeaderTable |
| שימושיות | הממשק אכן נותן לי את הרצון להמשיך לשחק, זה נחמד לקבל את ההודעות שעניתי נכון על שאלה ואני מחכה לראות מהי השאלה הבאה. |
| תיעוד | התיעוד קיים רק מעל הפונקציות (הסבר על כל אחת) לא קיים תיעוד על הקוד עצמו, והממשק של האפליקציה דיי ברור למשתמש. (יש מסך עזרה) |
| אתיקה | הכל תקין בחלק זה. |
| אבטחה | שם המשתמש והסיסמא לא הכי מאובטחים כי כאשר אני מתחבר הם מושכים את כל המשתמשים מהדטא בייס לתוך הקלאינט. |
| ביצועים 1 | כן שינוי עתידי יכול לפגוע בביצועים, הקוד כבר יחסית גדול ומסורבל, לדעתי עבודה בגוגל קולאב לפרוייקטים גדולים הי דיי בעייתית. במיוחד שלא עושים reuse ולא משתמשים בפונקציות גנריות הקוד יכול לגדול וזה יפגע בביצועים. |
| ביצועים 2 | כן, לקרוא לדטאה בייס פעם אחד ולשמור את הנתונים הרצויים. צוואר הבקבוק פה הוא בעיקר התקשורת עם הנתונים בfirebase. |
| קריאות | הקוד לא היה לי מובן בקלות יחסית כי התיעוד של פונקציות מסובכות לא קיים.  למשל הפונקציה : checkEndGame()  וגם loadLeaderTable() לא היה לי ברור כל כך איך הן עובדות, כי אין תיעוד בכלל |
| נגישות | האם הקוד נגיש לאוכלוסייה בעלת מוגבלות (ראייה למשל)  לא, אין התייחסות לאוכלוסיות בעלי מוגבלות למשל עיוורים.  אין סאונד או הקראה של הכתוב על המסך |
| בדיקות | האם הקוד הזה ניתן לבדיקה?  לדעתי קשה לבדוק את התוכנית הזו בשל העובדה שקיימים פה אלמנטים של HTML , אישית אני לא יודע איך אפשר לבדוק HTML בצורה טובה, אך אפשר לומר שאת הקוד בפייתון (הבלוק השלישי) אכן הוא testable |

1. כיצד הרגשת במהלך החוויה של מעבר הקוד?

בגלל שלא היה תיעוד זה היה לא הכי קל לעבור על הכל ולהבין, למרות שהקוד לא כזה מסובך ישנם חלקים שאני לא הכרתי כי לא רשמתי את זה במיוחד שאין לי גישה לfirebase.

1. האם לדעתך ניתן לשפר תרגיל זה?

כן! יש לי רעיון מעניין : אולי לקחת כל אחד מחברי הקבוצה ולחבר אותו עם חבר מהקבוצה המקבילה, וששניהם ישבו על הקוד של אחד של השני ביחד ובכך לא יהיה חוסר הבנה של מה שהקוד עושה וגם הצד השני ישמע את הביקורת ומה יש לתקן.

1. הערות נוספות

אין.

שם הסטודנט: דניאל ורדימון

|  |  |
| --- | --- |
| **קריטריון** | **הסבר** |
| מימוש | כן. ניתן לראות בבירור מימוש לפונקציות מרכזיות של הפרויקט לפי הנדרש, למשל הוספת שאלות, הוספת סרטונים, בחירת שאלות, הצגה של גרפים וכדומה. |
| יעילות | ניתן לשפר במספר דרכים אפשריות, לדוגמא:  שימוש בפונקציות חיפוש ופילטור של רשימות במקום מעברים עם לולאות for  שימוש בפרמטרים לפונקציות במקום משתנים גלובליים  הפחתת שימוש ב clear\_output |
| פשטות | ישנם מספר דרכים בהן ניתן לפשט את הקוד- שימוש בDictionary במקום if ו else רבים. חלוקה יותר מסודרת לפונקציות, פונקציות קצרות יותר. |
| מודולריות | הקוד נראה מודולרי ברמה מספקת, ישנה חלוקה ברורה בין הקבצים (קובץ לצד מנהל, צד משתמש ו HTML), וישנה חלוקה לפונקציות אשר כל אחת מבצעת תפקיד ברור ויחודי. |
| תבניות עיצוב | לא הבחנתי בתבנית עיצוב מסוימת בקוד. |
| OOP | לא נעשה שום שימוש במחלקות, ירושה, פולימורפיזם וכדומה. ניתן לשפר על ידי יצירת מחלקות לכל ישות, שימוש באובייקטים של המחלקות במקום משתנים גלובלים, שימוש בירושה כדי למזער שימוש מחדש של קוד (מחלקות לשחקן ומנהל שתיהן יורשות ממשתמש) וכו'. |
| באגים וטעויות | לא נתקלתי בבאגים בזמן הרצת הקוד. |
| טיפול בשגיאות | לא נמצאות התראות על שגיאות משתמש. |
| בדיקות | ניתן להוסיף בדיקה ששאלה שהוספה אכן הוספה כראוי. |
| שימושיות | הממשק לדעתי שמיש, יש פידבק טוב לפקודות המשתמש (הודעות על תשובות נכונות/שגויות וכו׳). |
| תיעוד | לרוב אין תיעוד מלבד הסבר על הפונקציה מעל שורת החתימה. |
| 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 הבחנתי בהמצאות דברים אלה. |
| אבטחה | כאשר שולפים משתמש מהמסד נתונים נשלפים נתוני כל שאר המשתמשים. |
| ביצועים 1 | האם לדעתך שינוי עתידי בקוד יכול לגרום לפגיעה בביצועים?  כן, הקוד כבר מסובך למדי, הוספה עתידית של עוד קטעי קוד, דפים וכדומה עלולה לפגוע בביצועים. |
| ביצועים 2 | האם ניתן לשפר לדעתך את הביצועים?  שימוש בתבניות עיצוב, מבני נתונים ועקרונות OOP עשויים לשפר את הביצועים. |
| קריאות | הקוד מסורבל והתיעוד ברמה מזערית. הקטעים של ה HTML לא היו ברורים עבורי. היה אפשר לשפר את הקריאות על ידי הפשטה של הקוד והורדת שימושים בהרבה if else ותיעוד מפורט יותר. |
| עיצוב ויזואלי | הפרויקט מעוצב בצורה טובה, שימוש בתמונות רקע מעניינות, הנפשות, ואלמנטי ממשק משתמש אשר נראים טוב. |
| רמת קושי של המשחק | המשחק ברמת קושי מספקת, בלי להיות קשה מדי. |

1. כיצד הרגשת במהלך החוויה של מעבר הקוד?

היה מעניין לראות קוד של מתכנתים אחרים אך לא נהניתי יותר מדי.

1. האם לדעתך ניתן לשפר תרגיל זה?

לא.

1. הערות נוספות

אין.