**תרגיל בית 2 מחשוב ענן**

*מגישים:* **Salamandras 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***

***צירפנו את קישור לGitHub ל Notebook מצורפים בסוף הקובץ הזה.***

*חלוקת התפקידים בצוות:*

***scrum master -מרכז את העבודה -*** *שיר כהן.*

***,frontend developer – פיתוח החלק האחראי על הצגה ללקוח-*** *גל פרפל.*

***,backend developer – פיתוח מסד הנתונים והעבודה מולו-*** *אלדד שניאור.*

***product manager – ייצוג הלקוח בצוות (בהתאם לחשיבה העיצובית שבוצעה)-*** *יותם אהרון.*

***UI – עיצוב הממשק -****דויד זבונריוב.*

***QA – בדיקות התוכנה -****עמית מלכה.*

**חלק ג׳:**

***חלק ראשון: הגדרת SDP – software development plan***

***א.*** *טבלה המתארת את שתי האיטרציות בהן תעבדו בסמסטר זה (תרגיל בית 2, תרגיל בית 3). שימו לב למטרה של כל איטרציה:*

|  |  |
| --- | --- |
| **Iteration 1** | **Implementing the basic logic of the Project** |
| Task 1 | Scrum backlog. |
| Task 2 | Writing acceptance test. |
| Task 3 | Building a database. |
| Task 4 | Designing flask, collab and html architecture. |
| Task 5 | Home screen. |
| Task 6 | Manager screen. |
| Task 7 | Login screen. |
| Task 8 | Register screen. |
| Task 9 | Integration and refactoring. |
| Task 10 | My profile screen. |
| Task 11 | Edit question screen. |
| Task 12 | Leaderboard screen. |
| Task 13 | Play trivia screen. |
| Task 14 | Statistics screen for each game. |
| Task 15 | Integration and refactoring. |
| **Iteration 2** | **Building a more full-featured version** |
| Task 1 | Play memory game screen. |
| Task 2 | Error 404 page. |
| Task 3 | Extensions in database. |
| Task 4 | Statistic screen for user. |
| Task 5 | Get hint screen. |
| Task 6 | Update design. |
| Task 7 | Improve Statistics screen for each game. |
| Task 8 | Integration and refactoring. |
| Task 9 | Database extensions. |
| Task 10 | Test the game- Acceptance and functional testing. |

**ב. טבלה מפורטת עבור איטרציה 1:**

|  |  |  |  |
| --- | --- | --- | --- |
| Executor | Task | Requirement | Iteration |
|  |  | Implementing the basic logic of the Project | Iteration 1 |
| Product manager | Writing user story. | Task Assignment - Scrum Backlog |  |
| Whole team | Creating main Use Case for the first prototype |  |  |
| Whole team | Modelling the Project according to the user story. |  |  |
| Backend developer, product manager, scrum master, Frontend developer. | Creating Requirements for each model in the project |  |  |
| Whole team | Selecting the software tools to work with |  |  |
| Product manager and scrum master | Create a list of questions and answers |  |  |
| Frontend developer, UI designer, Product manager. | Create the first draft of the screens in the application. |  |  |
|  |  | Building a database |  |
| Scrum master ,  Backend developer. | Creating an initial list of questions in a JSON file for the trivia game. |  |  |
| Scrum master ,  Backend developer. | Creating an initial list of users in a JSON file for the system use. |  |  |
|  |  | Writing an acceptance test.  (will be detailed in section b). |  |
| Scrum master,  Product manager, QA. | Checks where I can get an unsuccessful situation in the system. |  |  |
| Scrum master,  Product manager, QA. | Writing acceptance tests for all of the pages in the system the successful and unsuccessful situation. |  |  |
|  |  | Design flask ,colab and html architecture |  |
| Backend developer. | Creating routes for the backend and frontend for create, delete, update and read data. |  |  |
| Backend developer | implementing session cookies that will save the user data. |  |  |
| Backend developer | Creating a proxy server that will display our flask application. Using ngrok. |  |  |
| Backend developer, Frontend developer | Adding new routes according to the html pages received by the frontend developer |  |  |
|  |  | Home Screen |  |
| UI | Creating logos of clouds and font settings for the homepage |  |  |
| UI , Frontend developer | implementing buttons with styling that will match the theme of the website |  |  |
| UI, Frontend developer | implementing waves styling over the page |  |  |
| UI, Frontend developer, Backend developer. | Implementing game cards that will redirect to the game page |  |  |
| UI, Frontend developer. | Adding header for the homepage screen that will display navigation to all different routes of the pages |  |  |
|  |  | Login screen |  |
| Front developer | Create username and password text fields and labels for log in |  |  |
| Front developer | Add login button |  |  |
| Backhand developer | Checking the username and password in the database. |  |  |
| Front developer, backend developer | Move to home screen with user information |  |  |
| UI  Front developer | Design login page with main navigator |  |  |
|  |  | Edit question |  |
| UI, Front developer, Backend developer. | Get the all question from the database, and shows them as buttons. |  |  |
| Front developer, Backend developer. | Create text field for the filter question option. And make filter option. |  |  |
| UI , Front developer. | Create text fields for the questions description and his answers, and question level. |  |  |
| Front developer, Backend developer. | Implement the option to update questions details. |  |  |
| Front developer, Backend developer. | Implement the option to delete questions details. |  |  |
| UI, Frontend developer. | Adding header for the homepage screen that will display navigation to all different routes of the pages. |  |  |
|  |  | My profile screen |  |
| UI ,Front developer, backend developer. | Create a form that contains the following inputs: username, password ,nickname, my highscore. |  |  |
| Front developer | Connect the nav bar functionality to the my profile screen |  |  |
| Front developer, backend developer | Set the inputs to the current user’s data from the DB |  |  |
| UI, Front developer | Working on the page css |  |  |
| Front developer, backend developer | Writing the logics to allow the user to change his information except of the high-score which will be disabled |  |  |
|  |  | Statistics screen |  |
| UI ,Front developer, backend developer. | Create a graph for the number of correct answers and incorrect answers. |  |  |
| UI ,Front developer, backend developer. | Create a linear graph which shows the score and the question for each question in the game. |  |  |
| UI, Front developer, Backend developer. | Checking the result with the database high score. If the user breaks his high-score will be confetti on the screen and update his high-score in the database. |  |  |
| UI, Frontend developer. | Adding header for the homepage screen that will display navigation to all different routes of the pages. |  |  |
|  |  | Register screen |  |
| Frontend developer. | Designing the page with a gradient related to colour blue |  |  |
| Frontend developer | Creating username,password,nickname,id text fields and labels for registration |  |  |
| Frontend developer,Backend developer | Implementing register button that will send the data to backend |  |  |
| Frontend developer,backend developer | Redirect the client after registration back to home page and save the user as logged |  |  |
| Frontend developer,  Backend developer. | Implementing a redirect from the register page to the login page. |  |  |
|  |  | Integration and refactoring |  |
| Whole team | Code merge |  |  |
| QA | Check for bugs |  |  |
| Whole team | Fix bugs |  |  |
| Whole team | Refactor the code |  |  |
| QA | recheck |  |  |
| Whole team | Final overview |  |  |
| Frontend developer | Display progress bar. | Play trivia screen. |  |
| Frontend developer | Updating the progress bar after each question that is answered. |  |  |
| Backend developer. | Gets all the question list from the database. |  |  |
| Frontend developer, Backend developer | Display question and answer’s buttons with only one correct answer. |  |  |
| Frontend developer, Backend developer | Display player’s username, his score and the number of correct answers(Initial values: 0). |  |  |
| Frontend developer | Display an error message for each question that the user answered incorrectly. |  |  |
| Frontend developer | Display message for each question that the user answered correctly. |  |  |
| Frontend developer, UI | Display to the player when clicking on an answer's button an answer colour if the selected option is wrong the colour’s button turns to red. if the selected option is correct the colour’s button turns to green. |  |  |
| Backend developer | Checking the correct answer in the database. |  |  |
| Frontend developer | Updating the value of score and number of the correct answers for each correct answer. |  |  |
| Frontend developer | Display a button with the correct answer after the user selects an incorrect answer. |  |  |
| Frontend developer | Show the correct answer after clicking on the “Show correct answer” button. |  |  |
| Frontend developer | Display the next question button to display the next question. |  |  |
| Frontend developer | Updating the text in the answers questions and the text question after clicking on the “next question” button. |  | |  | | --- | |  | |
| Frontend developer | Display the exit button to exit the game. |  |  |
| UI, Frontend developer. | Adding header for the homepage screen that will display navigation to all different routes of the pages. |  |  |
|  |  | Integration and refactoring. |  |
| All team | Group meet up to talk about the final adjustments for the current iteration,  Where we discuss what changes we want to make, what bugs we need to fix, and then dividing the work based on the roles that need to participate on the final refactoring phase |  |  |
| All team | Splitting into teams to make the final adjustments |  |  |
| Scrum master | After the groups finished the changes the scrum master gathers everyone and asks questions about the work, and figuring out the last changes that needs to happen |  |  |
| All team | Integrating all of the adjustments |  |  |
| All team | Refactoring the code for optimization and semantics |  |  |
| All team | Final overview of the first iteration and celebration |  |  |
| Frontend developer | Creating “add question button” and “Show all questions button” | Manager Page |  |
| Backend developer | Linking the button to their appropriate routes. “Add a question” routes to addQuestion page.  “Show all questions” routes to the “showQuestions” page. |  |  |
| UI and Frontend developer. | Designing the page according to the main design of the application. |  |  |
| Frontend developer | Display a form with the required fields of the question: description, 4 options, difficulty, correct answer. | addQuestion page |  |
| Backend Developer | Handling the submit button action of the form by executing POST to the firebase. |  |  |
| Backend developer | Executing GET operation for the questions collection from the DB. | showAllQuestions |  |
| Frontend developer | Displaying all the existing questions from the DB as a list of buttons. |  |  |
| UI and Frontend developer | Designing the page according to the main design of the application. |  |  |
| Backend developer | Linking the button to its appropriate route. Each question button routes to a route with the question number. |  |  |
| UI and Frontend developer | Designing the page according to the main design of the application. | Leaderboard page |  |
| Backend developer | Executing GET operation for the users collection from the DB. |  |  |
| Frontend developer | Displaying the users data in a table according to their high-score (points). |  |  |

**Acceptance Test:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Section | Test ID | Description | Expected Result | Precondition | Requirement |
| Links | Link to trivia game valid. | The user scrolls down to the game section and clicks on “Trivia game” button. | The user is redirected to the game trivia screen | The user must be logged in to the system. | Implementing game cards that will redirect to the game page.  Connect the nav bar functionality to the my profile screen. |
|  | Link to “About us” valid. | The user clicks on “About us” button. | The user is redirected to About us screen |  | Adding header for the homepage screen that will display navigation to all different routes of the pages.  Connect the nav bar functionality to the my profile screen. |
|  | Link to “Our games” valid. | The user clicks on “Our games” button. | The user is redirected to “Our games” screen. |  | Adding header for the homepage screen that will display navigation to all different routes of the pages.  Connect the nav bar functionality to the my profile screen. |
|  | Link to “Leaderboard” valid. | The user clicks on “Leaderboard” button. | The user is redirected “Leaderboard” screen. | The user must be logged in to the system. | Adding header for the homepage screen that will display navigation to all different routes of the pages.  Connect the nav bar functionality to the my profile screen. |
|  | Link to “Home” valid. | The user clicks on “Home” button. | The user is redirected “Home” screen. |  | Adding header for the homepage screen that will display navigation to all different routes of the pages.  Connect the nav bar functionality to the my profile screen. |
|  | Link to “Login” valid. | The user clicks on “Login” button. | The user is redirected “Login” screen. |  | Adding header for the homepage screen that will display navigation to all different routes of the pages.  Connect the nav bar functionality to the my profile screen. |
|  | Link to “My profile” valid. | The user clicks on top right combox of the nav bar🡪 The user clicks on “My profile” button. | The user is redirected “My profile” screen. | The user must be logged in to the system. | Adding header for the homepage screen that will display navigation to all different routes of the pages.  Connect the nav bar functionality to the my profile screen. |
|  | Link to “Logout” valid. | The user clicks on top right combox of the nav bar🡪 The user clicks on “Logout” button. | The user is redirected “Home” screen. | The user must be logged in to the system. | Adding header for the homepage screen that will display navigation to all different routes of the pages.  Connect the nav bar functionality to the my profile screen. |
| Login test | Login successful. | The user is enter his username: "Rocket" and password:"123" and click log in. | System shows the home screen with user nickname at the to right corner. | The user is registered to the system and the user exist in the DB. | Checking the username and password in the database. |
| Login test | Login unsuccessful1  wrong username, correct password. | The user is enter his username: "Rackety" and password:"1234" and click log in. | message text " username or password are incorrect". |  | Checking the username and password in the database. |
| Login test | Login unsuccessfull2  correct username, wrong password. | The user is enter his username: "Rocket” and password:"1" and click log in. | message text " username or password are incorrect". |  | Checking the username and password in the database. |
| Login test | Login unsuccessfull3  wrong username ,wrong password. | The user is enter his username: "Rackety" and password:"1" and click log in. | message text " username or password are incorrect". |  | Checking the username and password in the database. |
| Logout test | Logout successfully. | The user clicks on top right combox of the nav bar🡪 The user clicks on “Logout” button. | The user is redirected “Home” screen. | The user logged in to the system. |  |
| Login Home page test | Display user’s nickname successfully. | The user is enter his username: "Rocket" and password:"123" and click log in. | User’s nickname is shown on top right combobox. | The user is not logged in to the system |  |
| Question’s page test. | update questions successful test. | Updating question from the DB successfully. | The user enters to the manager screen 🡪 the user clicks on the "show all questions" button🡪 "add questions" 🡪 the user fill the following data:  Question description:  "Which of the following is not a cloud service model?"  Option1:  "Infrastructure as a Service (IaaS)"  Option2:  "Platform as a Service (PaaS)"  Option3:  "Software as a Service (SaaS)"  Option4:  "Device as a Service (DaaS)"  Correct answer:  "Device as a Service (DaaS)"  Difficulty:  "Hard"  The user click on "Update question" button 🡪 the user clicks "show all question". | The question updated in the DB and shown in the all questions page. | all question from the database, and shows them as buttons.  Implement the option to update questions details |
| Question’s page test. | Filter questions test successfully. | The user types:” What is cloud” . | displayed questions change accordingly. | The user logged in to the system as admin.  The user is on “All questions” page. | Create text field for the filter question option. And make filter option. |
| Question’s page test. | Deleting question from the DB successfully. | The user enters to the manager screen 🡪 the user clicks on the "show all questions" button🡪 "select question" 🡪click on the "delete question" button. | The question deleted from the DB and removed from the all questions page. | The user is logged in as an administrator. | Implement the option to delete questions details. |
| Show my profile screen test. | Show My profile screen successfully. | The user is enter to the combo box on top right and clicks on the "My profile" button. | The user is taken to the my profile page where he can see all the user's data and the user can update is username, nickname or password. | The user logged in to the system. | Set the inputs to the current user’s data from the DB |
| Add question’s page test. | Adding question successfully. | The user enters to the manager screen 🡪 the user clicks on the "show all questions" button🡪 "add questions" 🡪 the user fill the following data:  Question description:  "Which of the following is not a cloud service model?"  Option1:  "Infrastructure as a Service (IaaS)"  Option2:  "Platform as a Service (PaaS)"  Option3:  "Software as a Service (SaaS)"  Option4:  "Device as a Service (DaaS)"  Correct answer:  "Device as a Service (DaaS)"  Difficulty:  "Easy"  The user click on "Submit" button 🡪 the user clicks "show all question". | The question added to the DB and shown in the all questions page. | The user is logged in as an administrator. | Handling the action submit button of the form by executing POST to the firebase |
| Add question’s page test. | Adding question unsuccessfully. | The user enters to the manager screen 🡪 the user clicks on the "show all questions" button🡪 "add questions" 🡪 the user fill the following data(but he doesn't fill one of the cells):  Question description:  "Which of the following is not a cloud service model?"  Option1:  "Infrastructure as a Service (IaaS)"  Option2:  "Platform as a Service (PaaS)"  Option3:  "Software as a Service (SaaS)"  Option4:  "Device as a Service (DaaS)"  Correct answer:  "Device as a Service (DaaS)"  Difficulty:  "Easy"  The user click on "Submit" button. | Message: "Please fill out this field". | The user is logged in as an administrator. | Handling the action submit button of the form by executing POST to the firebase |
| Show Manager screen test. | Show Manager screen successfully. | The user is enter to the combo box on top right and clicks on the "Manager dashboard" button. | The user is taken to the manger page where he can see all the questions and add new questions. | The user is logged as administer in to the system. |  |
| Leaderboard test | Leaderboard successful test. | home screen🡪click "Leaderboard" | table with all time best points. | user "click" leaderboard button. | Displaying the users data in a table according to their high-score (points).  Executing GET operation for the users collection from the DB |
| game test | Answer question correct successfully. | The question is displayed to the user and the 4 possible options:  "Which of the following is not a cloud service model?"  Option1:  "Infrastructure as a Service (IaaS)"  Option2:  "Platform as a Service (PaaS)"  Option3:  "Software as a Service (SaaS)"  Option4:  "Device as a Service (DaaS)"  🡪 the user chose the correct option from the 4 possible answers : Option4:  "Device as a Service (DaaS)" | button will become green , 2 points will added to user's score and message of "Good job! +2" will appear. | user logged in and choose to take a quiz. | Display message for each question that the user answered correctly.  Updating the value of score and number of the correct answers for each correct answer.  Display to the player when clicking on an answer's button an answer colour if the selected option is wrong the colour’s button turns to red. if the selected option is correct the colour’s button turns to green.  Display the next question button to display the next question. |
| Answer question wrong successfully. | The question is displayed to the user and the 4 possible options:  "Which of the following is not a cloud service model?"  Option1:  "Infrastructure as a Service (IaaS)"  Option2:  "Platform as a Service (PaaS)"  Option3:  "Software as a Service (SaaS)"  Option4:  "Device as a Service (DaaS)"  🡪 the user chose wrong option from the 4 possible answers : Option3:  "Software as a Service (SaaS)" | Button will become red and message of "wrong answer" will appear and new button with show correct answer will appear. | user logged in and choose to take a quiz. | Display an error message for each question that the user answered incorrectly.  Display to the player when clicking on an answer's button an answer colour if the selected option is wrong the colour’s button turns to red. if the selected option is correct the colour’s button turns to green.  Display the next question button to display the next question. |
| moving to next question successful. | user finish to answer a question and click "next question" button before the last question. | another question will appear on the screen and the game bar will "move" one question forward. | user start a game and answer a question. | Creating the trivia game screen with full **basic** logic.  Updating the progress bar after each question that is answered.  Display the next question button to display the next question. |
| Finish the game successfully. | user finish to answer a question and click "next question" button after the last question . | moving to player statistics page 🡪 display graphs ( Progress graph, number of questions answered correctly graph).  If the current score is bigger than the user's score🡪 update the user's highest score in DB. | User start a game and answer 10 questions. | Creating the trivia game screen with full **basic** logic.  Add current game score to DB.  Display info (Progress graph, Leader board etc.). |
| game test | Show correct answer successfully. | The question is displayed to the user and the 4 possible options:  "Which of the following is not a cloud service model?"  Option1:  "Infrastructure as a Service (IaaS)"  Option2:  "Platform as a Service (PaaS)"  Option3:  "Software as a Service (SaaS)"  Option4:  "Device as a Service (DaaS)"  🡪 the user chose wrong option from the 4 possible answers : Option3:  "Software as a Service (SaaS)"  🡪The user clicks on the “Show the correct answer button” | The correct answer displayed to the user. | The user answer wrong answer. | Show the correct answer after clicking on the “Show correct answer” button.  Display the next question button to display the next question. |
| game test | Exit game successfully. | The user clicks on the “Exit” button. | The user redirect to the home page. | The user is playing the game. | Display the exit button to exit the game. |
| Register test | Register successful test. | Home screen🡪click "register" 🡪enter username :"amit" password:"123" nickname: "aaamit" id:123456789🡪click "Register button" | The user move to home screen as logged in user. | The user is not registered to the system. | Implementing register button that will send the data to backend ).  Redirect the client after registration back to home page and save the user as logged |
| Username already taken successful test. | Home screen🡪click "register" 🡪enter username: "Rocket" password:"123" nickname:"aaamit1" id:123456789🡪click "Register button" . | message :"nickname or username already exist". | The username exists in DB. | Implementing register button that will send the data to backend |
| Nickname already taken successful test. | home screen🡪click "register" 🡪enter username:"Rocket1" password:"123" nickname: "aaamit" id:123456789🡪click "Register button" | message :"nickname or username already exist" | The nickname exists in DB. | Implementing register button that will send the data to backend |
| Register test | Registered page link to login page successfully. | The user clicks on the “Login” button | The user is redirect to the login page. | The user is on the registeration page. | Implementing a link from the register page to the login page. |
| Statistics Test | Statistics-showing successfully. | the user being showed statistics about his previous game | Presents 2 graphs according to the game the user played. | the user clicked on finish game. and the user just played a match of trivia. | Create a graph for the number of correct answers and incorrect answers.  Create a linear graph which shows the score and the question for each question in the game. |
| Statistics Test | Statistics With a new highscore record successful test. | the user gets a score which is higher than his current high score. | the user is being showed the statistics page and a message of “New high score”  with the message a confetti will be shown to the user | user high score is lower than the current high score he got. | Checking the result with the database high score.  If the user breaks his high-score will be confetti on the screen and update his high-score in the database. |
| showAllQuestions page test | display all questions on the screen - GET operation successful test. | the user clicks on the “manager dashboard” → the user clicks on show all questions. | multiple buttons with questions are displayed on the page. | The user is logged in as admin. | Executing GET operation for the questions collection from the DB.  Displaying all the existing questions from the DB as a list of buttons. |
| showAllQuestions page test | Link button to its page successful test. | the user clicks on the “manager dashboard” → the user clicks on show all questions →  the user clicks on a question button. | The user is redirected to a specific question page with a form which contains all the attributes of a question object filled with the chosen question data. | The user is logged in as admin. | Linking the button to its appropriate route. Each question button routes to a route with the question number. |
| Manager | Button\_  add\_a\_question | The user clicks on the button “Add a Question” and being redirect to a different page. | The user being transferred from the current page to the “add a question page. | The user must be logged in to the system.  And the user must be an admin | Linking the button to their appropriate routes. “Add a question” routes to addQuestion page.  “Show all questions” routes to the “showQuestions” page. |
|  | Button\_show\_all\_questions | The user clicks on the button “Show all Questions” and being redirect to a different page. | The user being transferred from the current page to the ”Show all Questions” page. | The user must be logged in to the system.  And the user must be an admin. | Linking the button to their appropriate routes. “Add a question” routes to addQuestion page.  “Show all questions” routes to the “showQuestions” page. |
| Manager  ShowAllQuestions page. | Display\_all\_questions. | The user entered the page Show All Questions.  And after entering the page the user will see all Questions. | All available questions being showed to the user | The user must be logged in to the system.  And the user must be an admin.  Moreover the user must be in the show all question page. | Displaying all the existing questions from the DB as a list of buttons. |

**חלק ב׳:**

**טבלת המשוב:**

|  |  |  |
| --- | --- | --- |
| **הערת משוב** | **האם התבצע שינוי ב אפליקציה** | **נימוק** |
| חווית משתמש | לא | היות ולא ניתן פירוט מעמיק יותר על חווית המשתמש לא היה דרך לדעת מה יש לתקן. הערה כללית מידי ולכן לא בצענו תיקון משמעותי עבורה |
| לא ראיתי אלמנטים שיש לשפר | לא |  |
| קצת ui | כן | בצענו שינויים נרחבים לאורך הפרויקט הן בכפתורים והן בגדלים שונים של הדפים |
| אפשר להוסיף עוד פונקציונאליות | לא | עבור הדרישות שהיו נתונות במטלה התבצעו ואף יותר מכך ( הוספנו באופן פרטני יותר מן הנדרש ) הערה אינה מספקת מספיק מידע בכדי לדעת איזה פונקציונאליות חסרה. |
| עיקרון המשחק לא כיפי | לא | החלטנו כי נבצע שינויים מרכזיים במשחק עצמו לקראת החלק השלישי של המשחק ולכן ההערה נרשמה לחלק הבא. |
| פרופורציה של הטקסט קשה לראות את כל המסך | כן | תוקן בחלקים עיקריים מן האתר. בחלק של השאלות נעשה שינוי כך שהשאלות יתאימו לגודל הדף |
| הכל סבבה | לא |  |
| אין לי הערות | לא |  |
| אין, ממה שראיתי המערכת מאוד מושקעת וחשבו על כל פרט קטן. | לא |  |
| עיצוב קצת , יש כפתורים שלא מוצאים ישר וצריך לחפש במסך | לא | בעקבות |
| קשה קצת למצוא כפתורים כדי ללחוץ. | כן | תיקנו כמה כפתורים בעייתיים שאכן היה קשה ללחוץ ולמצוא . |
| אני מעדיפה שהממשק יהיה יותר פשוט היה לי קצת קשה להבין מה עלי לעשות בכל שלב ואיפה צריך ללחוץ | לא | לא מצאנו כיצד לשפר את חווית המשתמש כך שתהיה פשוטה יותר אך לקחנו את הערה לצומת ליבנו וננסה לתת לה יותר חשיבות לקראת מטלה 3. |
| קצת יותר צבעים | לא | החלטנו כי האתר יהיה לפי מודל של 60 30 10. כאשר הצבע המרכזי הוא כחול והצבע הנגזר ממנו הוא מעין תכלת. לפי דעת הקבוצה הוספת צבעים נוספים תגרום לבלבול וחוסר אחידות בפרויקט. |
| לא מצאתי בקלות איך לחזור אחורה | לא | יש אפשרות לחזור אחורה בעבור כל דף. ניתן פשוט ללחוץ על כפתור המוח בדף הראשי  בנוסף כמו בכל אתר רגיל יש אופציות של כפתור חזור. |
| המסך תצוגה גדול מדי | לא | מסך התצוגה אכן גדול מידי. ולכן ננסה לבצע שינויים או לדון עליהם באופן מעמיק יותר לקראת מטלה 3. |
| המסך של השאלות קצת גדול מדי וקשה לשימוש כי צריך לכלול למטה אחרי כל שאלה | כן | כפי שציינו ב7. ביצענו שינויים בכדי שחלק של השאלות יהיה נוח יותר למשתמש.  בנוסף העברנו את מיקום הכפתורים כך שאין צורך יותר לגלול למטה. |
| המסך ענקי ולא נוח לשימוש | לא | באופן דומה ל16. |
| כשאני רוצה לדעת את התשובה הנכונה עדיף שזה יהיה באופן אוטומטי | לא | על פי עקרון המשחקיות שהלכנו דרכו.  החלטנו כי למשתמש ניתנת האופציה האם הוא רוצה לראות את התשובה או לא  הצגת התשובה באופן אוטומטי יכולה לפעמים להרוס למשתמש את החוויה  שכן הוא יכול בכל זאת לנסות לפתור את השאלה גם אם טעה. |
| המסך גדול ולא כל כך נוח | לא | באופן דומה ל16, 18. |
| בעריכת שאלות אולי לשנות את המבנה של איך בוחרים את השאלות לעדכון | לא | את מבנה בחירת עריכת השאלות החלטנו להשאיר כמו שהוא אך כן החלטנו לאפשר שינוי עיצובי לקראת המטלה ה3. |
| העיצוב אצל השחקן, השאלות גדולות מידי | כן | תוקן. באופן דומה ל17 וגם ל 7. |
| הכל מעולה | לא |  |
| אין לי משהו להגיד | לא |  |
| היה מדהים לא רואה שצריך לשפר | לא |  |
| דף הבית | לא | לא היה נימוק אז לא ידענו כיצד להתייחס להערה. |
| לתקן כפתור כניסה למשחק בתצוגה למטה | כן | תוקן אכן הייתה בעיה שחלק מן הכפתור היה "מת". והבעיה תוקנה כעת הכפתור לחיץ בעבור כל נקודה |

**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>

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

<https://github.com/cohensh96/CloudComputing.git>