**תרגיל בית 3 חלק א**

**מגישים:**

*בר שטיינר 318401957*

*עומרי שלו 315838540*

*רן פולק 318265808*

*עמרי גאוי 313394850*

*עומר זומרשטיין 316439876*

*נועם אופיר 318295888*

**קישורים:**

* קישור לGITHUB של תרגיל בית 3 חלק א:

<https://github.com/Ranash12/cloudCourse/tree/main/HW3a>

* קישור לקוד בGoogle Colab:

<https://colab.research.google.com/github/OmriGawi/Cloud-Battleship-Trivia-Game/blob/main/CloudBattleshipTriviaGame.ipynb>

* סרטון : מצורף קובץ MP4

***חלק ראשון: הגדרת SDP – software development plan – איטרציה 2***

נגדיר את התפקידים בצוות:

scrum master – מרכז את העבודה- **רן פולק**

frontend developer – פיתוח החלק האחראי על הצגה ללקוח- **עמרי גאוי**

backend developer – פיתוח מסד הנתונים והעבודה מולו- **עומרי שלו**

product manager – ייצוג הלקוח בצוות (בהתאם לחשיבה העיצובית שבוצעה)- **נועם אופיר**

UI – עיצוב הממשק- **עומר סומרשטיין**

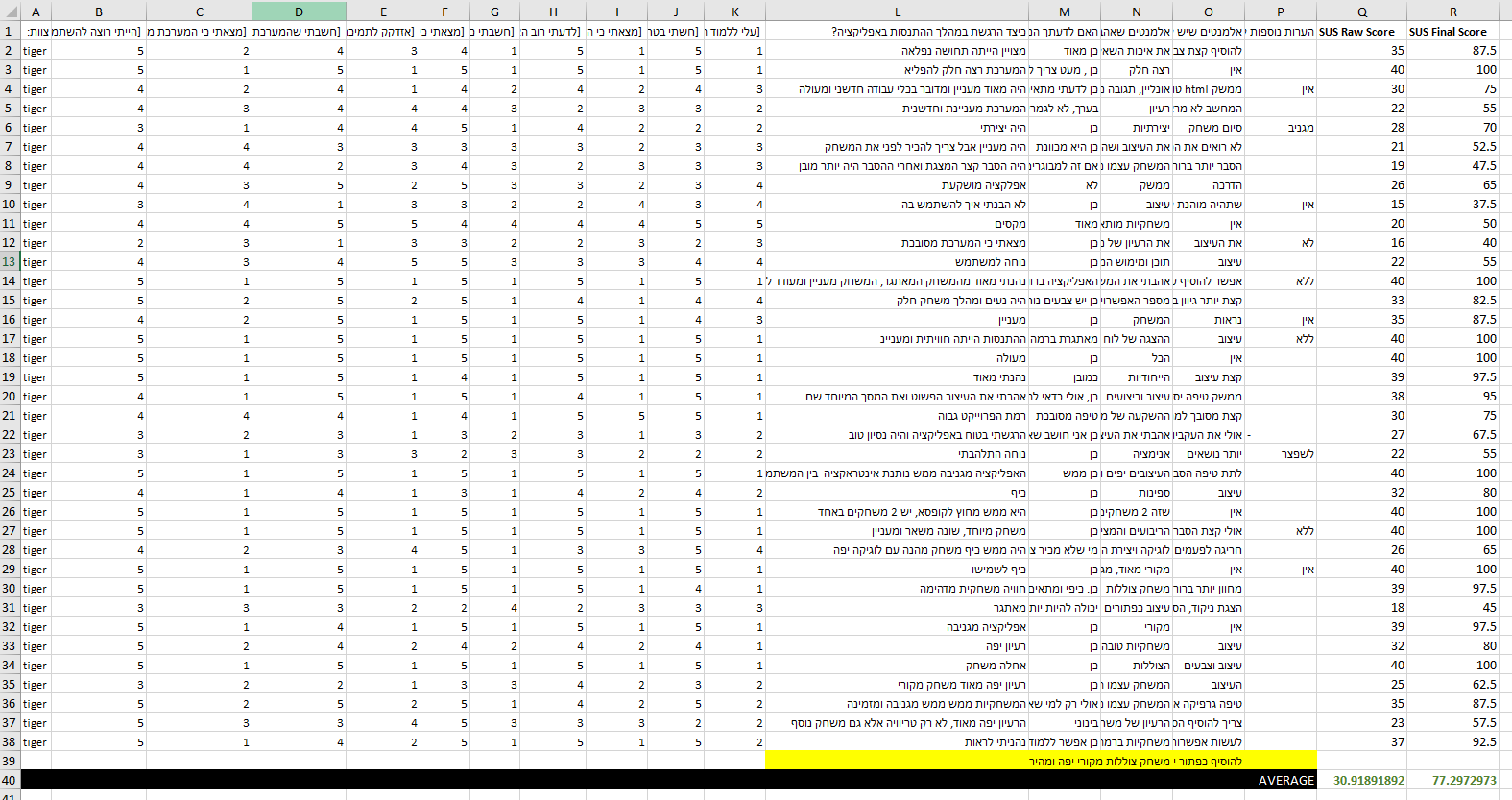
QA – בדיקות התוכנה- **בר שטיינר**

א. טבלת משימות בהתאם לאיטרציה 2:

| **Task** | **Requirement** | **Executor** |
| --- | --- | --- |
| **Task 1** | **Preparing the login database** | backend developer |
|  | Create a login table in the FireBase | backend developer |
| **Task 2** | **Preparing the database to save users session information** | backend developer |
|  | Create a users table in the FireBase | backend developer |
| **Task 3** | **Add leaderboard table** | **Frontend Developer Backend developer  Product Manager  UI  QA** |
|  | Top 3 users will be in different colors | Frontend developer Backend developerUI |
|  | Acceptance Test - The table shows the top10 users where the top 3 are shown with different colors. | QA Product Manager |
| **Task 4** | **Add new screen of the user’s board during the game** | **Scrum Manager Frontend developer Backend developer Product Manager UI** |
|  | Create new button for opening the board | Frontend developer  Scrum Master |
|  | Show the user’s board | Frontend developer  Backend developer UI |
|  | Acceptance Test - check if the user’s ships are the same as in the user bord | QA Product Manager |
| **Task 5** | **Add new Admin option in order to add new question** | **Frontend developer  Backend developer UI  QA** |
|  | Add the new button | UI Frontend |
|  | Send the new question to the DB | Frontend developer  Backend developer |
|  | Acceptance Test - The table shows all the questions from the database, and when the admin clicks on a specific question all the info of the question will be displayed. | QA |
| **Task 6** | **Admin can edit questions from database** | **UI  Backend developer Frontend developer scrum master product manager QA** |
|  | Add new button of edit questions | Frontend developer Backend developer product manager scrum master |
|  | Show the questions from the DB | Frontend developer Backend developer  UI |
|  | Let the user edit the questions | Backend developer  Frontend developer |
|  | Acceptance Test - Check if the information has been updated in the database accordingly to the input | QA |
| **Task 7** | **Add Login and Logout feature** | **UI Backend developer Frontend developer scrum master product manager QA** |
|  | Create login page | UI  Frontend developer  Product Manager  Scrum Master |
|  | Transfer data to and from the DB | Frontend developer Backend developer |
|  | Acceptance Test - Check if the information is existing in the DB | QA |
|  | Create Logout Button on the main screen | Frontend developer  UI Product Manager |
|  | Acceptance Test - Check if the information is existing in the DB | QA |
| **Task 8** | **Add registration option** | **Backend developer Frontend developer Product Manager QA** |
|  | Add register button | Frontend developer Product Manager |
|  | Save the user input into the database | Backend developer |
|  | Acceptance test - Check if the new user added to the database according to the input. | QA |
| **Task 9** | **Improve user interface from the first iteration** | **UI  Frontend developer scrum master Product Manager Backend developer** |
|  | Designing the Screens | UI  Frontend developer scrum master  Product Manager  Backend developer |
| **Task 10** | **Add rules page that describes how to play** | **UI Frontend developer Product Manager  scrum master  Backend developer** |
|  | Rewrite the rules of the game | Product Manager scrum master  Backend developer |
|  | Show the rules to the user | Frontend developer UI |
|  | Design the rules page | UI  Frontend developer |
| **Task 11** | **User can see his stats during the game by graph** | **UI  Backend developer  Product Manager  Frontend developer  scrum master  QA** |
|  | Create a “show me stats button” | Frontend developer |
|  | Take the stats from the DB | Backend developer |
|  | Show a graph by the stats | Frontend developer  Backend developer UI |
|  | Let the user close the stats and back to the game | Frontend developer scrum master |
|  | Acceptance test - Check the information in DB | QA |
| **Task 12** | **User can finish the game whenever he wants** | **Frontend developer Backend developer Product Manager scrum master** |
|  | Add exit button | Frontend developer |
|  | Show the finish game screen and update the DB | **Frontend developer** Backend developer **Product Manager scrum master** |

**חלק שני: סגירת פרויקט**

**SUS Score:**

****

**SUS Raw Score: 30.918**

**SUS Final Score: 77.297**

חישוב SUS התבצע באמצעות:

1. **שאלות בעלות קונטקסט חיובי:**

תפיסות חיוביות של המשתמש מהשימוש באב טיפוס.

לוקחים את התוצאה ומחסירים ממנה 1.

לדוגמה: אם x הייתה התשובה של העונה, אז מבצעים (result = x - 1)

1. **שאלות בעלות קונטקסט שלילי:**

תפיסות שליליות שהמשתמש חש לאחר אינטרקציה עם האב טיפוס.

לוקחים את התוצאה המקסימלית האפשרית, ומחסירים ממנה את התוצאה.

לדוגמה: אם התוצאה הכי גבוהה הייתה 7, ו- x היה התשובה של העונה, אז מבצעים (result = 7 - x)

לעומת זאת אם התוצאה הכי גבוהה הייתה 5, ו- x היה התשובה של העונה, אז מבצעים (result = 5 - x)

הרעיון: אם השאלה היא שלילית והמשתמש ענה 1, זה אומר שהוא דווקא כן היה מרוצה.

לאחר מכן, סוכמים את כל השורה לפי אופן החישוב לעיל וקיבלנו תוצאה שנקראת:

SUS Raw Score

כעת, מחשבים את SUS Final Score באמצעות הנוסחה:

(Sus Raw Score) \* 2.5

לבסוף, נבצע ממוצע ל- SUS Raw Score ול- SUS Final Score בנפרד עבור כלל המשתמשים ונקבל את התוצאה הרצויה.

**Cloud Battleship Trivia Game - Programming Doc**

**Login and Register functions:**

* display\_login\_menu(): Displaying the login page.
* check\_login(username,password): Checking if the username and password are legal.
* user\_exists(username): Check if the user exists in DB.
* register\_user(username,password): Add new user to DB.

We are using this part of code to create some of the login-page with HTML design:

****

**Main menu page:**

* display\_main\_menu(): Displaying the main menu page of the game.
* display\_rules(): Displaying the rules.

In this page we used bootstrap to design the rules using “collapse” to hide the rules and by clicking on the rule it will appear.

* display\_admin\_login(): This function will display the screen where the user can enter password to login into the admin page.
* admin\_options(): This function will be called when the user entered the correct password when login to the admin page. it will also display the options available to the admin.
* admin\_input(): Let the admin add a new question to the DB.
* add\_question(): We are using this function in admin input to write the new question to the DB.
* edit\_questions(): This function will let the admin to edit new questions from the database.
* list\_questions(): Function to get all the questions from the DB.
* edit\_question(question\_key, question\_data): function to send the database the info of the updated question.

**Leaderboard**

* display\_leaderboard(): This function will display the leaderboard table with the top10 users
* get\_leaderboard(): This function will get the top10 users from the database.

**Endgame**

* display\_end\_game(): This function will show the charts at the end of the game.
* save\_stats\_to\_db(winner): This function used to save the stats of the user to the DB after the game is over.
* get\_current\_user(): This function used to get the information about the current user that logged in.

**Start Game**

* create\_grid\_buttons(computer\_board, user\_board): This function creates the computer board with the buttons.
* create\_grid\_buttons\_start(): This function creates the user board with buttons, where the user after puts his ships.
* create\_ship\_buttons(ship\_list): This function creates the buttons for the ships of the user to choose.
* display\_ship\_placement(): This function allows the user to place the ships on his board.
* generate\_computer\_ships(ship\_list): This function puts the computer’s ships on its own board.
* display\_trivia\_question(question\_data, grid\_buttons, user\_board): This function displays a question to the user with the answers options (as buttons).
* on\_answer\_click(button, question\_data, grid\_buttons, answer\_buttons, user\_board): This function implements the user answers on the questions and the following results from the answer.
* on\_attack\_click(computer\_board, button, grid\_buttons, user\_board): This function implements the user attacks on the computer’s board.
* computer\_attack(user\_board): This function lets the computer attack the user’s board.
* is\_game\_over(board): This function checks if the game is over.
* create\_menu\_buttons(user\_board): This function creates the menu buttons in the middle of the game. (Exit , Show Statistics, Show my board).
* display\_user\_grid(user\_board): This function displays the user board in the middle of the game when clicking on the “Show my board” button.
* show\_statistics(): This function displays the users statistics in the current game.
* start\_game(user\_board, computer\_board): This is the main function to run the whole game page.

**Cloud Battleship Trivia Game - User Doc**

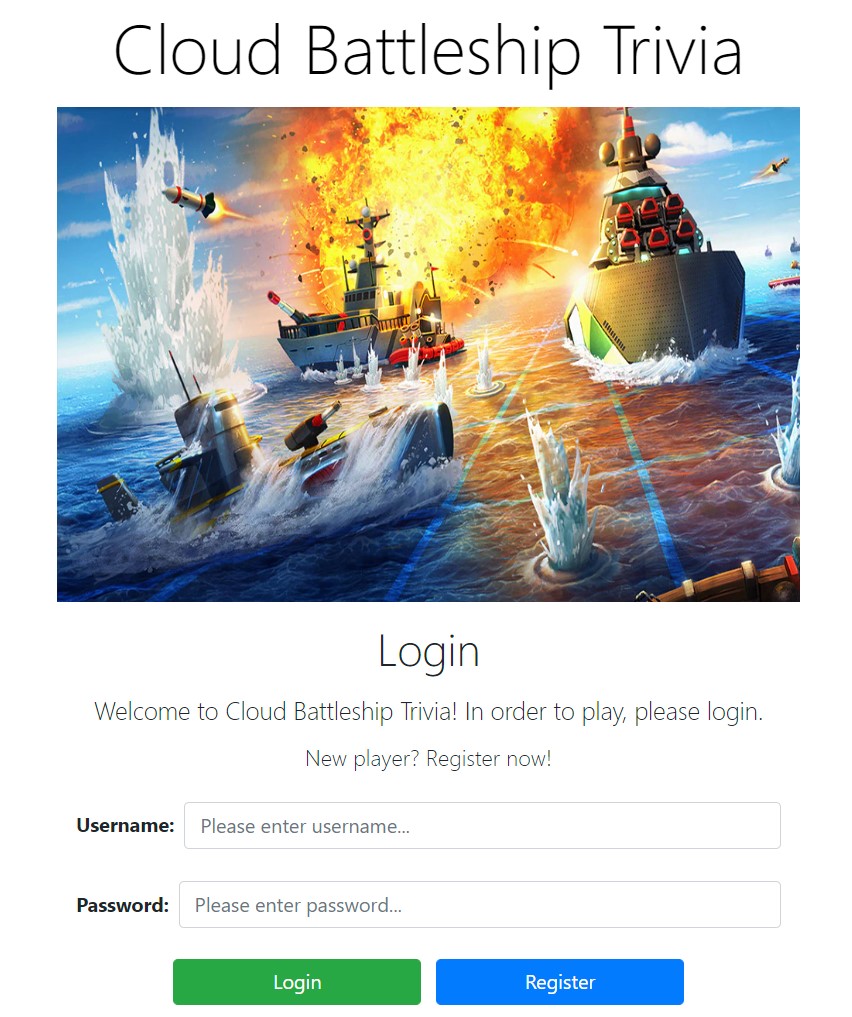
**Introduction**

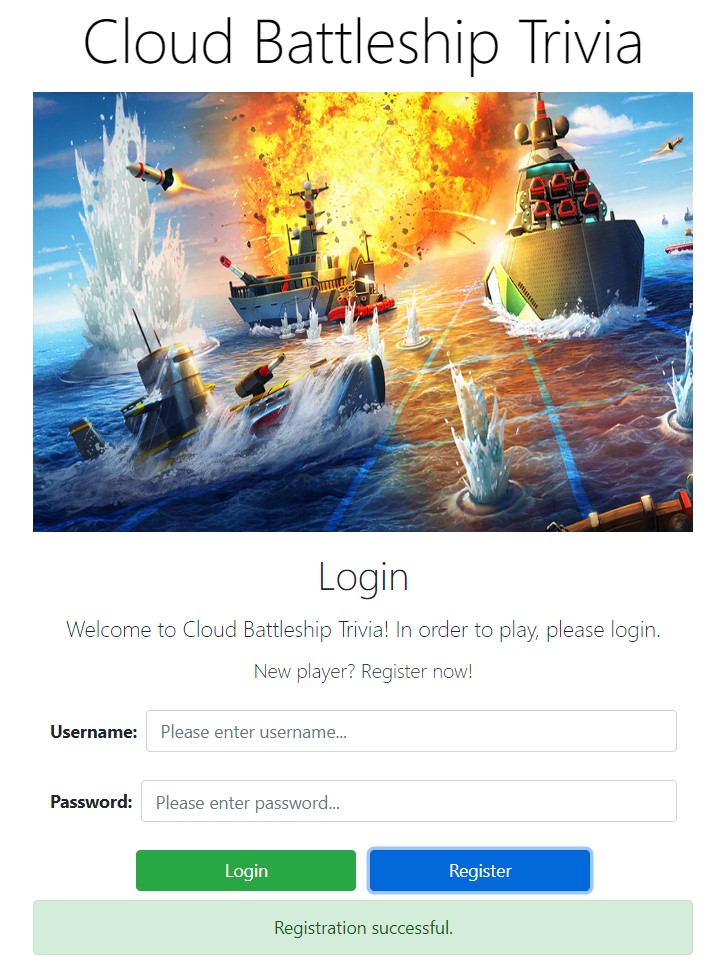
Welcome to the Cloud Battleship Trivia Game! This game combines the classic strategy of the Battleship game with trivia questions. The aim is to test your general knowledge while sinking enemy ships in a virtual sea battle. This document is intended to provide players with instructions on how to play the game, understand the scoring, and navigate through various options available.

**Registration and Login**

1. To begin, open the game.
2. Register for an account with a username and password.
3. Log in with your credentials.

when the user clicks on “Login” without enter anything in the fields, an error message is shown.

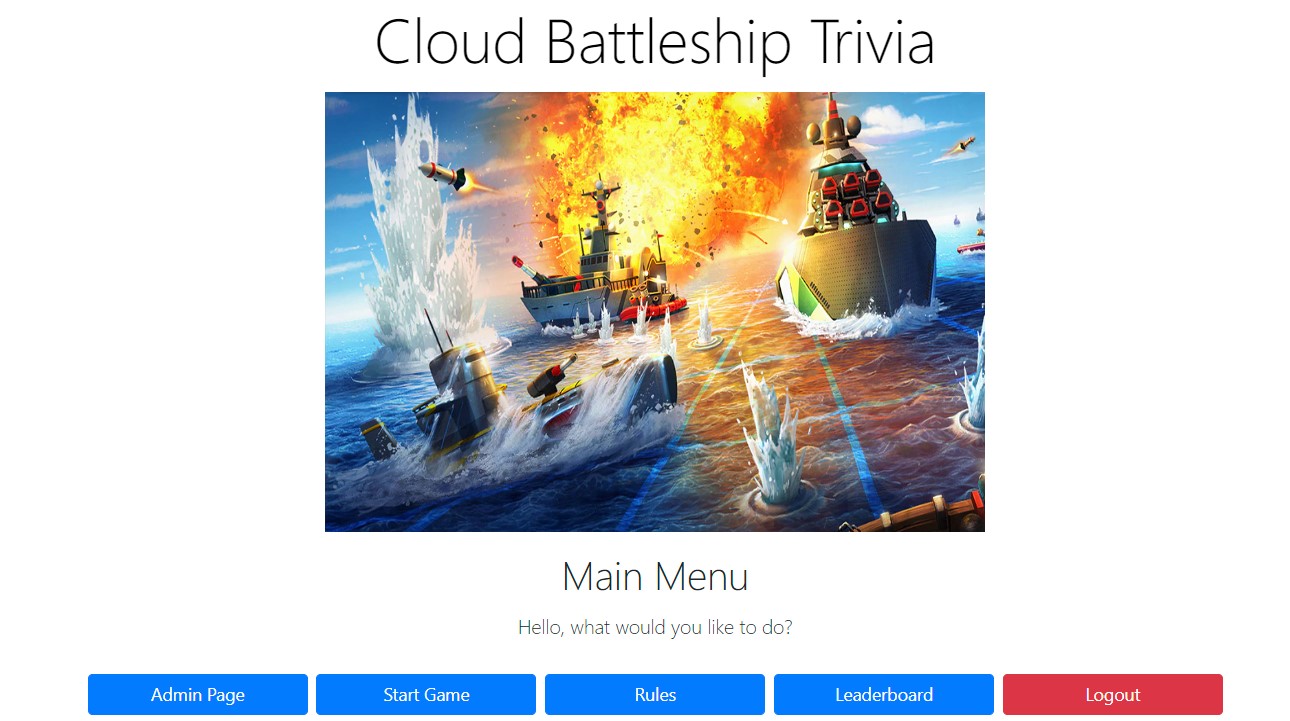
****

also when the user registered successfully, a successful message is shown.  


**Main Screen**

The main screen of the game:  
Few buttons are available to the user.

* Admin Page
* Start Game
* Rules
* Leaderboard

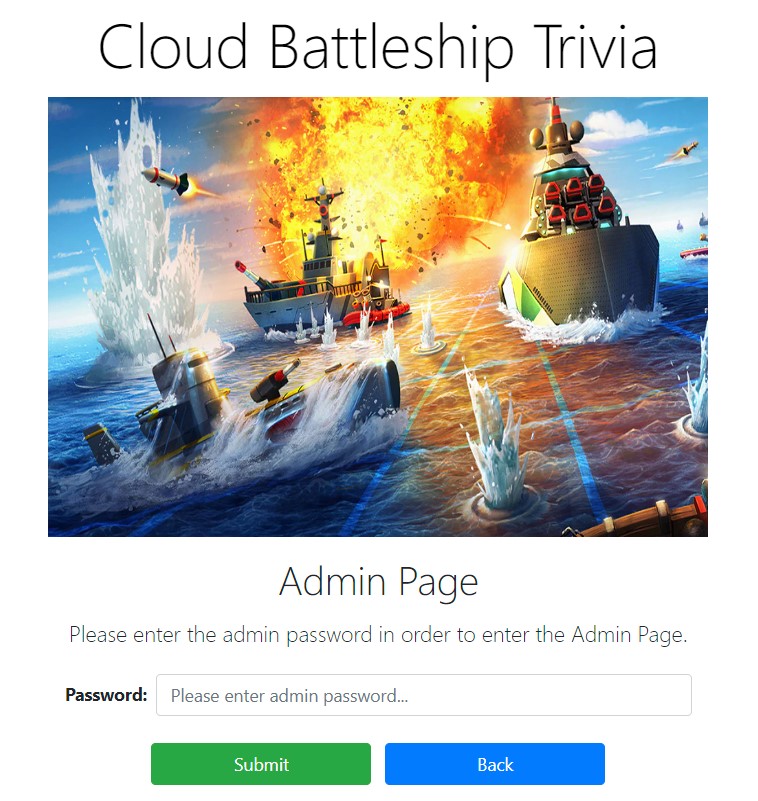


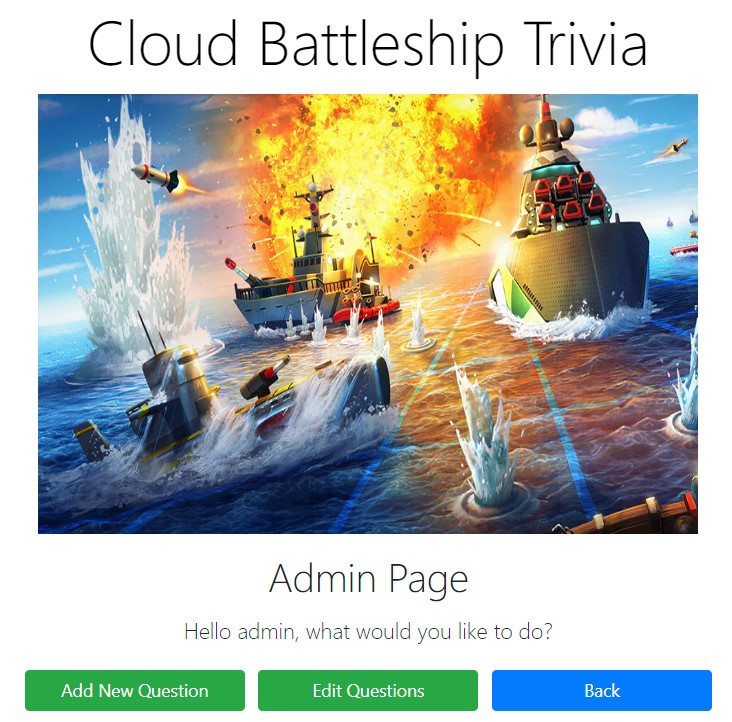
* Admin Page:

this button is for the admin purpose like to add new question to the database, edit questions

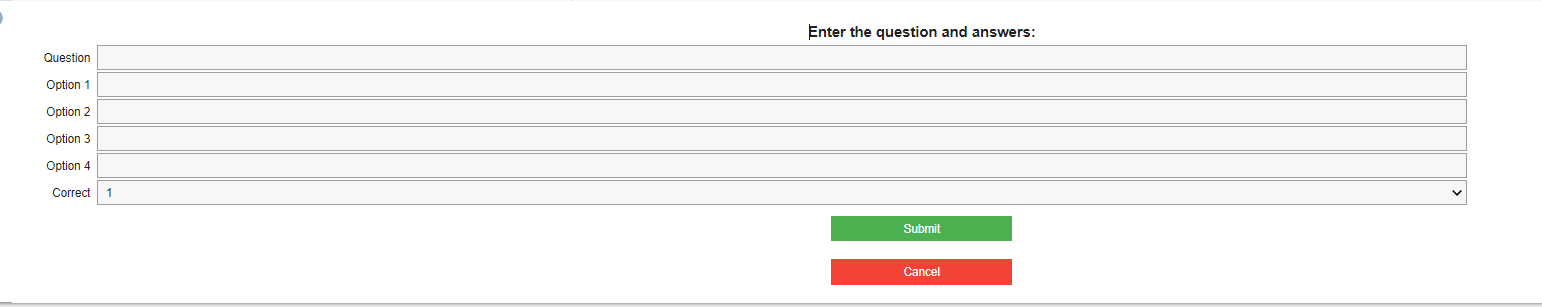
when the user click on the button, a login process will be shown.

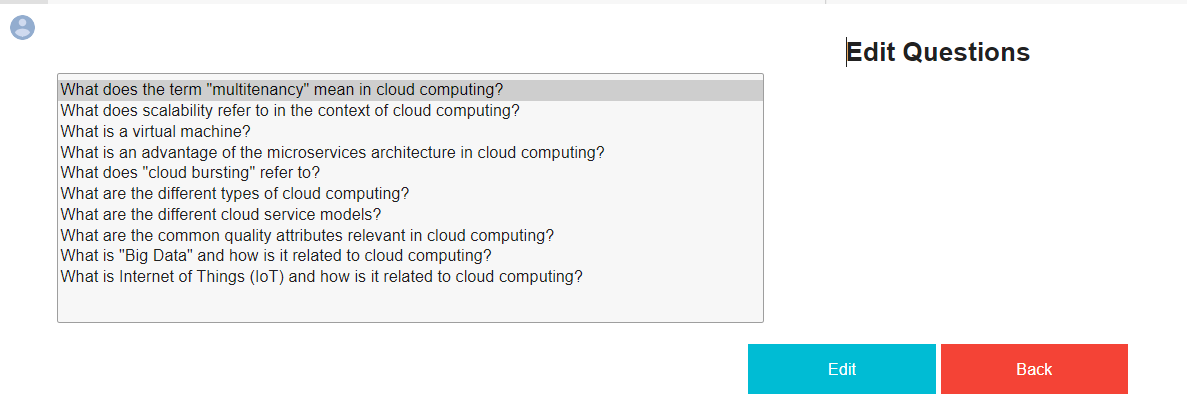
the user has to enter the admin password that saved in the database to move to the admin options.



when the user entered the correct password he will move to:  


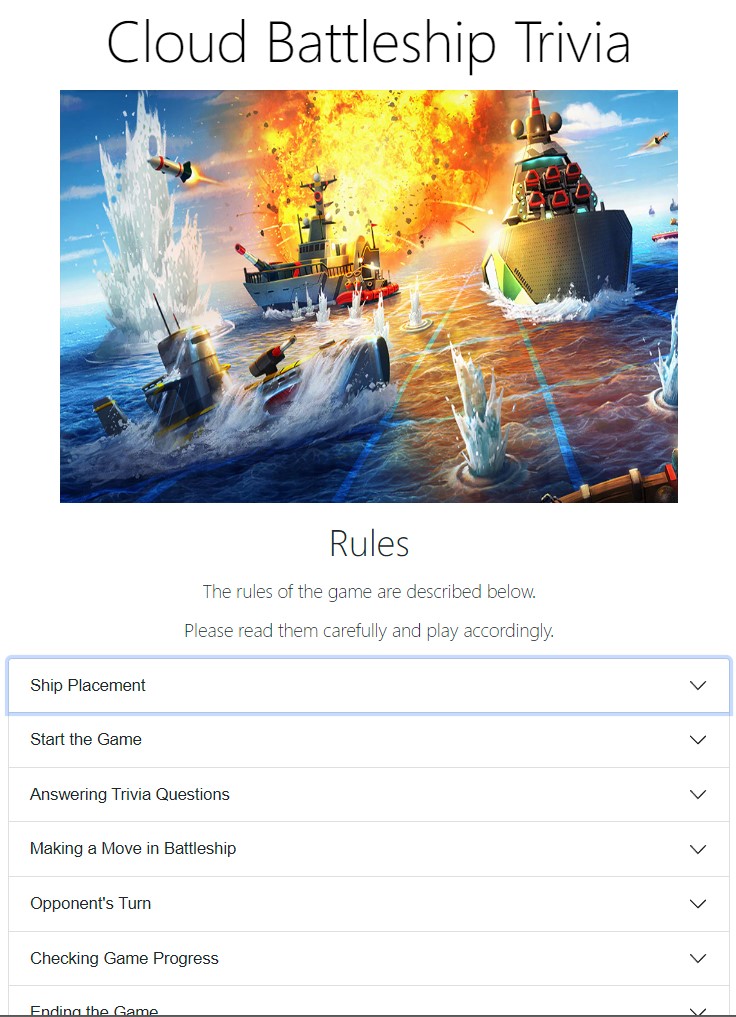
For selecting “Add New Question” he will move to the page where he add question.



For selecting “Edit Question” he will move to the page where he can view all the questions from the database and choose any question to edit:  


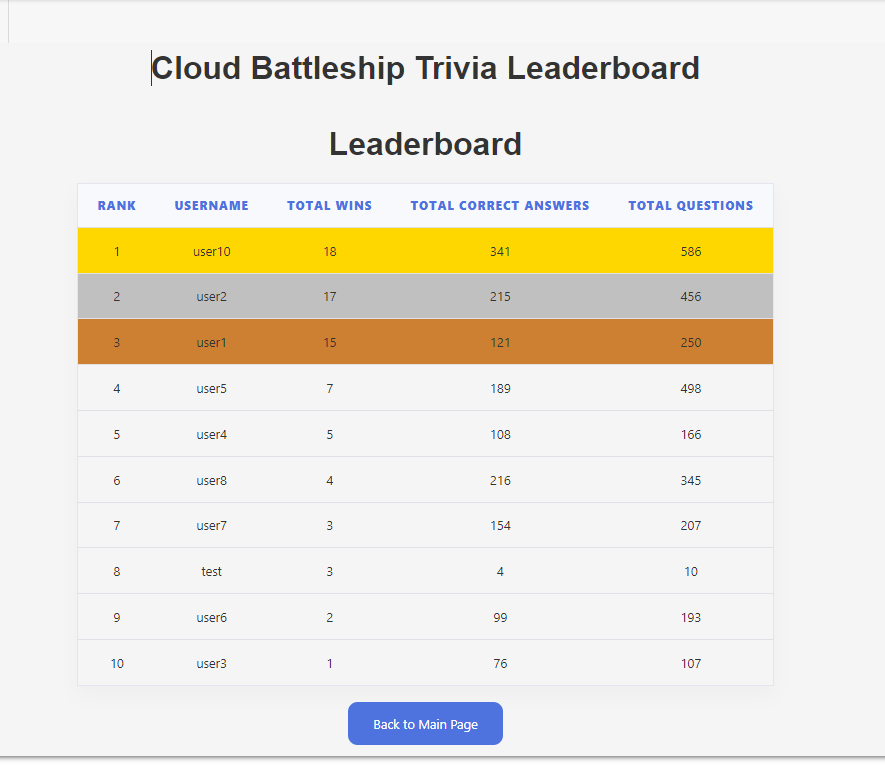
* Rules

This button will move the user to the rules section, where he can understand the whole functionality of the game.

The user can click on any part separately to see what to do in the selected section.  


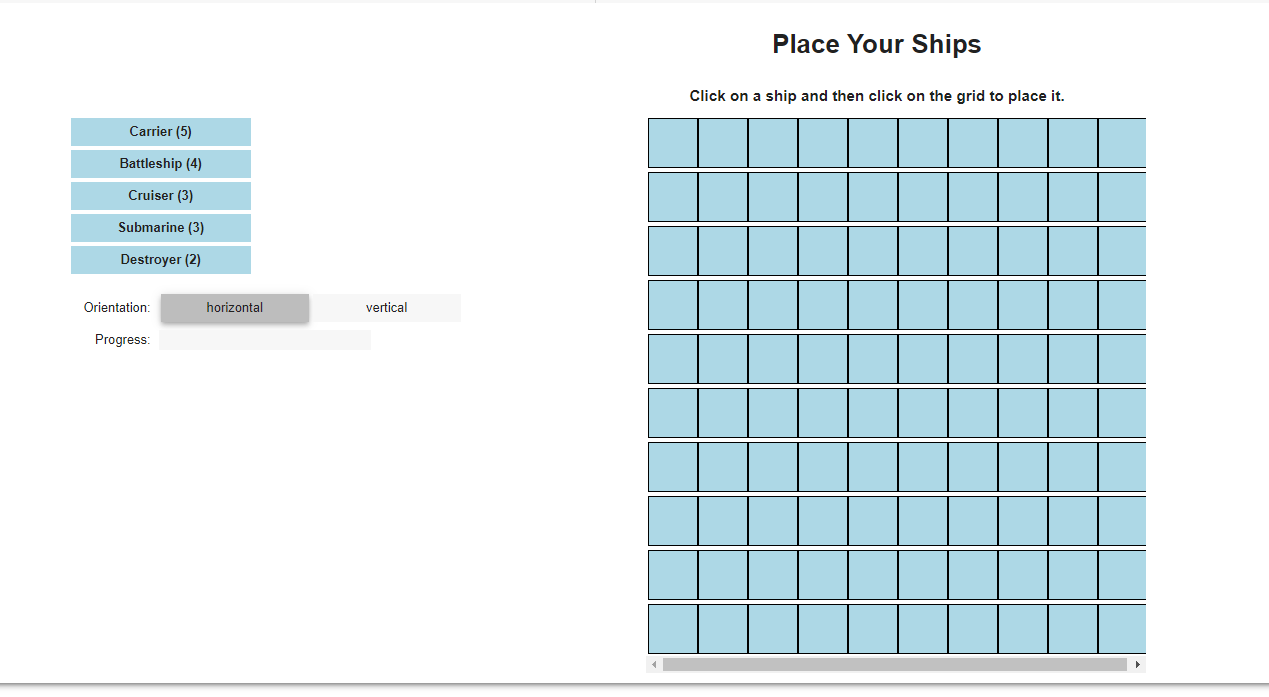
* Leaderboard

This button will move the user to the top 10 players in the game, where the top 3 players are printed in different colors.

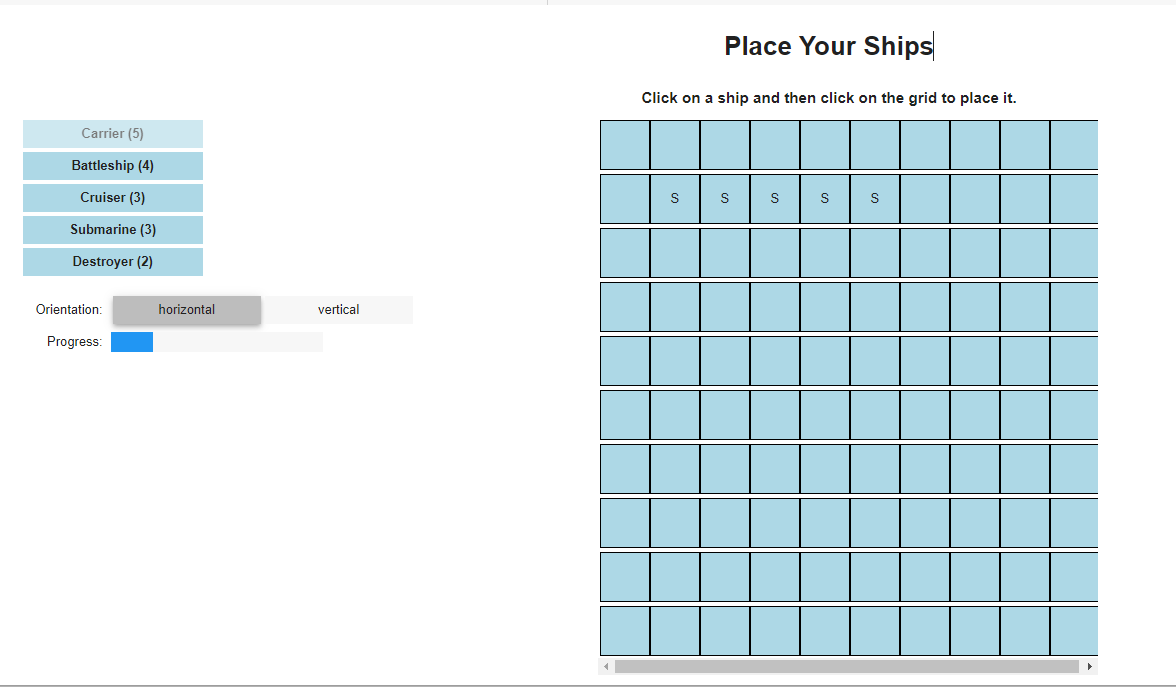


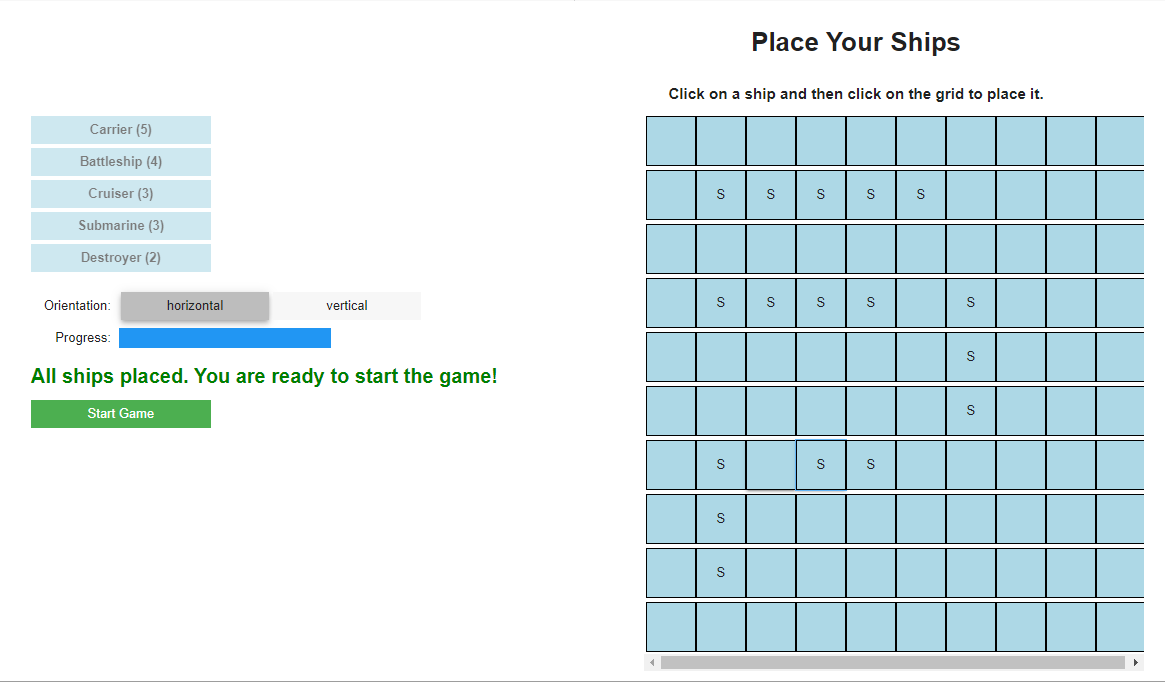
* Start Game

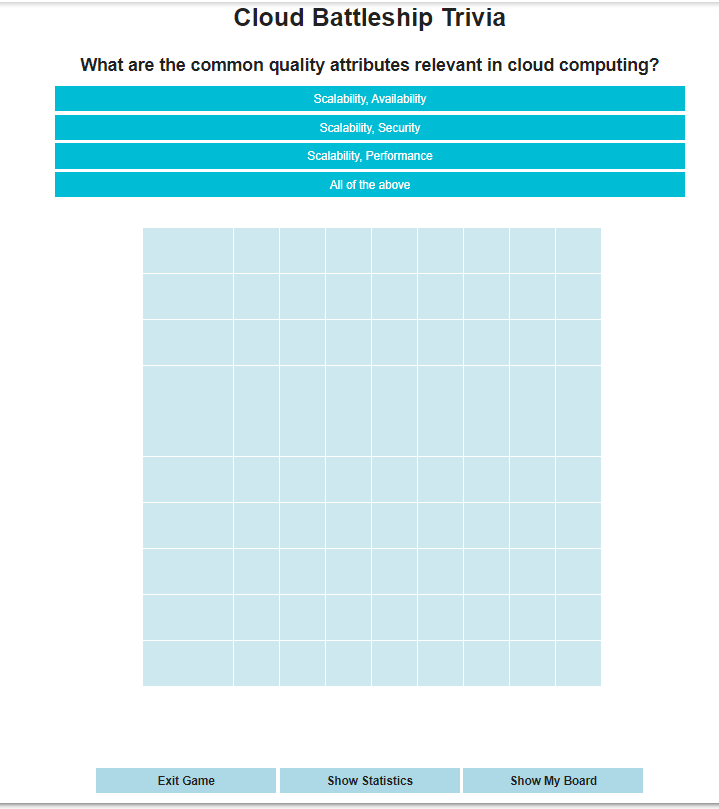
This is the main button to start the game process.  
At the start of the game the user has to select in the grid the places he want to place his ships.  
The main grid is shown in the middle and the ships available to the user are at the top left of the screen.



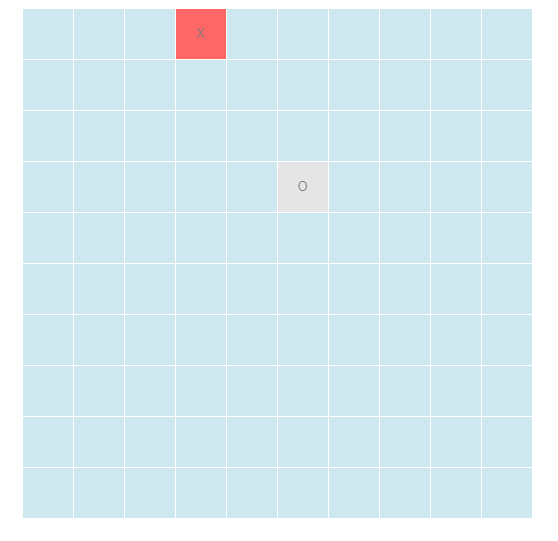
After selecting a ship and click on the grid it will look like this, where the progress bar show the placement process:



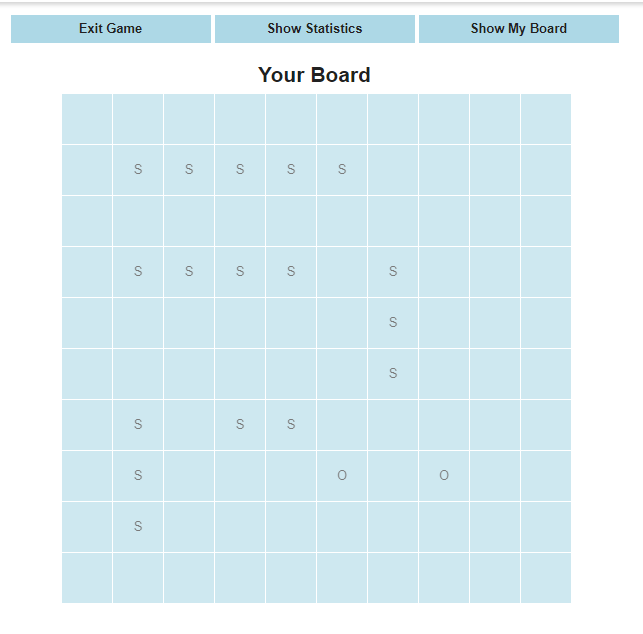
When the user selected all the ships to place, he can start the game.  


After clicking on the “Start Game” the game will start.  
A trivia question will be shown to the user and let him choose one of the 4 answers.  
if the answer is correct, it will let him attack the computer’s board, if the answer is incorrect, the computer will attack the user’s board.  
  
  


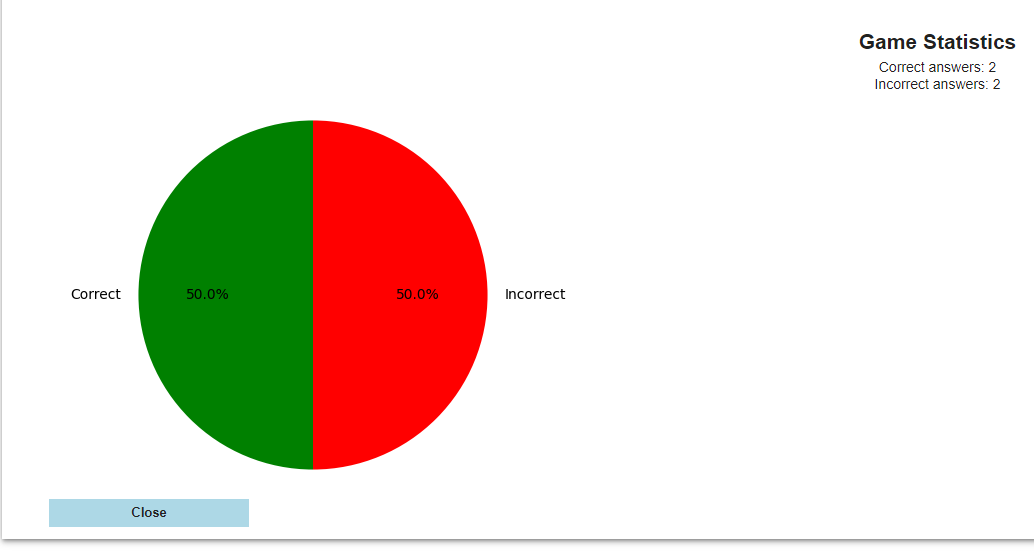
X - Hit  
O - Miss



At each time of the game, the user can see his board by clicking on the “Show my board” button.  
The “S” are representing the user ships selection and “O” the computer hits on his board.



Also the user can view the statistic on the game by clicking on the “Show Statistic” button:



When the game is over, either user won or computer won a statistic will be shown.  
