**Cloud Computing –   
Final Project**

**Cloudy**

**מגישים:**

* אופק מלכה – 311381206
* ניר בטש- 313594418
* עדן בר- 205907256
* אלמוג חייקין- 318159019
* טל גאון- 319005484
* יובל גבאי- 207405788

**System name: Cloudy**

The main goal of the system is to make the topic of cloud computing accessible to the audience of software engineers in Israel and around the world. The system will contain a variety of closed questions simulating a trivia game on a variety of topics in cloud computing. The user will earn points for each correct answer he answers so he can track his progress.

1. **Functional Requierments:**

* The system will allow questions to be presented.
* The system will allow the accumulation of points.
* The system will allow the display of an explanation to solve each question.
* The system will allow displaying graphs.
* The system will allow saving data.

1. **Non-Functional Requirements:**

* The questions will be on different topics in cloud computing (types of cloud, cloud work models and more. (Effectiveness)
* Questions and user data will be retrieved from the database.
* The dataset will be hosted in Firebase. (Dependency on other parties)
* Question structure on DB will be uniform. (Reusability)
* The graphs that will be displayed will be according to user's progress in a specific game and in relation to all the games he has played in the past. (Effectiveness)
* The data that the system will save after every game is the score of the user on current game round. (Data retention)

***SDP – software development plan***

|  |  |
| --- | --- |
| Iteration 1 | Implementing the basic logic of the Project |
| Task 1 | Create a function that handle user's answer. |
| Task 2 | Create the option to move through screens. |
| Task 3 | Create a function to insert questions into the DB. |
| Task 4 | Create a function that edit or delete exists questions. |
| Task 5 | Create a function the fetch question from DB and display it. |
| Task 6 | Create a game walkthrough screen. |
| Task 7 | Create a function that display a graph of user's performance. |
| Task 8 | Create acceptance tests for all the tasks |
| Task 9 | UX\UI design. |
| Task 10 | Initialize DB. |
| Task 11 | Create question for the game. |

|  |  |
| --- | --- |
| Iteration 2 | Implementing the basic logic of the Project |
| Task 1 | Separate the code into files. |
| Task 2 | Build user's DB. |
| Task 3 | Create a function that handle each user's session. |
| Task 4 | Redesign graphs at the end of each game. |
| Task 5 | Create an admin's interface for questions control. |
| Task 6 | Create a register form. |
| Task 7 | Create validation for user login. |
| Task 8 | Complete and improve tasks from previous iteration according to students response. |
| Task 9 | Create acceptance test for all iteration 2 tasks |

***UseCase***

תמונה שמכילה תרשים

התיאור נוצר באופן אוטומטי

* **Note:** this Usecase display our first vision for the system.

|  |  |
| --- | --- |
| מספר שאלה | ניקוד |
| 1 | 3.357143 |
| 2 | 3.285714 |
| 3 | 3.428571 |
| 4 | 3.535714 |
| 5 | 3.5 |
| 6 | 3.392857 |
| 7 | 3.678571 |
| 8 | 3.75 |
| 9 | 3.5 |
| 10 | 3.357143 |
| סכום | 34.78571 |
| ניקוד כולל | 86.96429 |

***SUS***

**Developer manual:**

**General Information:**

Cloudy is a game structured similarly to a trivia quiz and its goal is to increase developers’ knowledge level about cloud computing.

The game is written in python, HTML, CSS and JavaScript. In addition, the data about users and game questions is stored in Firebase, which is a Google platform that helps with app development. We used the database services that it provides.

**This project consists of the following main functions:**

**Admin page:**

1. **open\_admin\_screen():**

**Purpose:** Display admin screen for adding or editing questions. It retrieves the questions from the Firebase database and displays them in a table.

**Trigger:** Click on "Add/ Edit question" button.

**Input:** None

**Output:** None

1. **delete\_question(id):**

**Purpose:** Delete an existing question from database.

**Trigger:** Click on "Delete question" button on admin screen.

**Input:** Question ID.

**Output:** None

1. **display\_question(id):**

**Purpose:** Display selected question for editing.

**Trigger:** Select question from questions table and click on "Edit question" button on admin screen.

**Input:** Question ID.

**Output:** None

1. **submit\_question(q, a1, a2, a3, a4, c):**

**Purpose:** Update question values or save a new question in DB.

**Trigger:** Click on "Submit" button on admin screen after all the relevant fields are filled.

**Input:** Question, 4 answers and the number of the correct answer.

**Output:** None

**End of game:**

1. **create\_line\_chart(correct\_answers):**

**Purpose:** After finishing a game the user will see 2 graphs to keep track of his progress. The graph will show the user the number of correct answers they answered in each of the games they played.

**Trigger:** User finished a game round (after answering 10 questions).

**Input:** The number of correct answers that the user achieved in his last game.

**Output:** None

1. **create\_pie\_chart(answers):**

**Purpose:** After finishing a game the user will see 2 graphs to keep track of his progress. The graph will show the user the distribution of the correct answers he answered versus the errors in the current game he finished.

**Trigger:** User finished a game round (after answering 10 questions).

**Input:** User answers in current game.

**Output:** None.

1. **get\_answer\_history(user):**

**Purpose:** Get users answers history for building graphs at the end of games.

**Trigger:** User starts a new game.

**Input:** User id.

**Output:** List of user's scores from all the games that they played.

1. **save\_answers(user, answers):**

**Purpose:** Save user's score from the last game in DB.

**Trigger:** User ends a game.

**Input:** User ID and game score.

**Output:** None.

1. **create\_end\_screen(answers, correct):**

**Purpose:** Display end game screen- calculate user score and activate functions that create progress graphs.

**Trigger:**

**Input:** List of user answers and the number of correct answers.

**Output:** End game screen.

**Game screen:**

1. **get\_questions():**

**Purpose:** Pull all questions from DB.

**Trigger:** Start new game.

**Input:** None

**Output:** List of questions and their answers.

1. **display\_question(question):**

**Purpose:** Display a question and its answers on screen at a game round.

**Trigger:** User is on a game session.

**Input:** Question and its answers.

**Output:** None.

1. **display\_victory():**

**Purpose:** Display end game screen with user's progress charts.

**Trigger:** User ends their game.

**Input:** None.

**Output:** None.

1. **next\_question():**

**Purpose:** Display the next question from the question series.

**Trigger:** User chose answer at current question.

**Input:** None.

**Output:** None.

1. **select\_answer(answer):**

**Purpose:** Display to user if they chose the right answer or wrong one by coloring the correct answer in green and the wrong answer in red.

**Trigger:** User chooses one answer from answer options.

**Input:** Answer (string).

**Output:** None.

1. **start\_game(header\_update\_cb, reset\_cb):**

**Purpose:** Start a game by initializing various variables, shuffling and selecting a series of questions, updating the header, displaying an answer form, and showing the first question to the user.

**Trigger:** User click on "Start Game" button from the main screen.

**Input:** Callback functions- one for updating the header and one for resetting the game.

**Output:** None.

1. **update\_header(score, currQuestion):**

**Purpose:** Initialize the header according to user score and the number of questions in current game.

**Trigger:** User clicks on "Start Game" button or answered a question correctly.

**Input:** User score and the number of the current question.

**Output:** None.

**Login Screen:**

1. **show\_login\_form():**

**Purpose:** Display "Login" page.

**Trigger:** Start running the program or click on "Already registered" from "Register" page.

**Input:** None.

**Output:** None.

1. **login(username, password):**

**Purpose:** Check if user exists in DB and if their password is correct. If it is, activate "open\_main\_menu" function for displaying main screen, otherwise error message will be displayed.

**Trigger:** User clicked on "Login" button.

**Input:** Username and password.

**Output:** None.

1. **open\_main\_menu():**

**Purpose:** Display main screen for user according to his permissions.

**Trigger:** User entered correct username and password.

**Input:** None.

**Output:** None.

1. **show\_register\_form():**

**Purpose:** Display register page for unregistered users.

**Trigger:** User clicked on "Not registered?" button.

**Input:** None.

**Output:** None.

**Main Screen:**

1. **start\_game\_mm():**

**Purpose:** Display game screen, display the relevant header (start new counting of questions) and activate "start\_game" function (mentioned before under "Game screen").

**Trigger:** User clicked on "Start Game" button on main screen.

**Input:** None.

**Output:** None.

1. **how\_to\_play():**

**Purpose:** Display game instructions page for the user.

**Trigger:** User clicked on "How to Play" button on main screen.

**Input:** None.

**Output:** None.

1. **edit\_questions():**

**Purpose:** Display edit question screen for admin.

**Trigger:** Admin clicked on "Edit question" button on admin main screen.

**Input:** None.

**Output:** None.

**Register Screen:**

1. **register(username, password):**

**Purpose:** Register user by inserting his username and password to DB (on Firebase). It checks if the username already exists and if it does, shows an error message, otherwise the registration succeeds, and the main screen will be displayed for the user.

**Trigger:** User clicked on "Register" button.

**Input:** Username and password.

**Output:** None.

***User Manual:***

1. תמונה שמכילה טקסט, צילום מסך, עיצוב גרפי, אומנות

   התיאור נוצר באופן אוטומטי**Login Screen:**

- Upon starting the application, you will be presented with a user login screen.

- If you already have an account, enter your username and password to log into the app.

- If you don't have an account, click on the "Not registered" button to create a new account.

- If you enter an incorrect username or password while logging in, you will receive an error message stating "Username or password is incorrect! Please try again

1. **תמונה שמכילה טקסט, צילום מסך, עיצוב, אומנות

   התיאור נוצר באופן אוטומטיRegister Screen:**

- To create a new account, click on the "Register" button on the login screen.

- The register screen will prompt you to enter the required information to create your account.

- Fill in the following fields:

- Username: Enter a username consisting of only numbers or digits, with a length between 6 and 14 characters.

- Password: Enter a password consisting of only numbers or digits, with a length between 6 and 14 characters.

- Re-enter Password: Re-enter the same password for confirmation.

- Once you have filled in the required fields, click on the "Register" button to create your account.

- If you already have an account, click on the "Already Registered" button to create a new account.

1. **Main Menu:**

תמונה שמכילה טקסט, צילום מסך, עיצוב, אומנות

התיאור נוצר באופן אוטומטי

- After successfully logging in or registering, you will be directed to the main menu screen.

- The main menu offers two options: "Start a new game" and "How to play."

1. **Start a New Game:**

תמונה שמכילה טקסט, צילום מסך, עיצוב

התיאור נוצר באופן אוטומטי

- Selecting "Start a new game" will begin a trivia game focused on cloud computing.

- The game consists of a series of questions related to cloud computing.

- After answering 10 questions, your performance will be displayed in a history graph.

- Your total score will also be displayed on the screen.

- You can view your performance on a line chart and a pie chart.

- To return to the main menu, use the "Return to main menu" button.

**The history graph shows your progress, indicating the number of correct answers you have provided.**

**Line chart**

**Y – axis = Correct answers counter**

**X – axis = Number of games**

**Pai chart**

**Wrong percentage answers – Orange**

**Correct answers – Blue**תמונה שמכילה טקסט, צילום מסך, תרשים, אסטרונומיה

התיאור נוצר באופן אוטומטי

* Return to main menu button will send you back to main screen.

1. תמונה שמכילה טקסט, צילום מסך

   התיאור נוצר באופן אוטומטי**How to Play:**

- Selecting "How to play" will provide instructions and guidelines on how to play the game.

- You will receive a welcome message with an overview of the game.

- The instructions will guide you through the gameplay process.

- To start the game, click the "Let's play" button.

1. **Header and Navigation:**



- Throughout the application, there is a header that displays a home button.

- The home button allows you to navigate back to the main menu from any screen.

- Logout button to log out from the user

**- In-Game Information:**



- During the game, you will see a progress indicator showing the current question number (e.g., 6/10).

- If you answer a question correctly, the correct answer count will be displayed (e.g., Correct answers: 6).

Note: This user manual provides an overview of the main features and functionalities of the Cloudy Trivia application. For more detailed instructions or troubleshooting, please refer to the in-app help section or contact our support team.

Enjoy playing Cloudy Trivia and have fun exploring cloud computing knowledge!

***Links:***

***Github:*** [***https://github.com/almogkh/cloud-computing***](https://github.com/almogkh/cloud-computing)

***Google Colab:*** [***https://colab.research.google.com/drive/1NSIp8A95S4Zbg1dsGcq6-DAN97GxA1Rw?usp=sharing***](https://colab.research.google.com/drive/1NSIp8A95S4Zbg1dsGcq6-DAN97GxA1Rw?usp=sharing)

***Firebase:*** [***https://lab6-6229a-default-rtdb.firebaseio.com/***](https://lab6-6229a-default-rtdb.firebaseio.com/)

***Challenges:***

* Design the game so it will be **attractive** on one hand, and on the other hand make it **understandable and restful** for the user.
* **System speed:** We encountered difficulty in improving the efficiency of the code to make the system run faster.
* Working on **unknown environment** (Google colab): we found difficulties in integrate JavaScript with Python.