**Developer Manual: Cloudy Trivia - Cloud Computing Quiz Application**

**אופק מלכה – 311381206**

**ניר בטש – 313594418**

**עדן בר – 205907256**

**אלמוג חייקין – 318159019**

**טל גאון – 319005484**

**יובל גבאי – 207405788**

**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.