**הגשת פרויקט סוף:**

**מגישים:**

נוי ביטון - 312238678

דולב סרי - 313428005

דבי דובינקר - 318179033

אייל קובי - 209291418

נופר בן-סימון - 308468826

אילנית חנוכו - 208657320

**שאלה 1:**

**שם המערכת:** Head in the clouds

**הסבר על המערכת:** קהל היעד של האתר הוא אנשים שרוצים להרחיב את הידע שלהם בתחום מחשוב ענן. האתר מכיל מידע בתחומים שונים בתחום הענן. האתר בנוי בתור משחק אינטראקטיבי. המשתמש יתחבר למערכת, ויבחר נושא ללמידה. בכל נושא המשתמש יכול לתרגל שאלות אמריקאיות. בנוסף, בכל שלב המשתמש יכול לראות את רמת התקדמות הלמידה שלו. ככל שהמשתמש יצבור יותר נקודות הוא יקבל מדלייה יותר מתקדמת (ברונזה / כסף / זהב).

**שאלה 2 – דרישות פונקציונאליות ולא פונקציונאליות:**

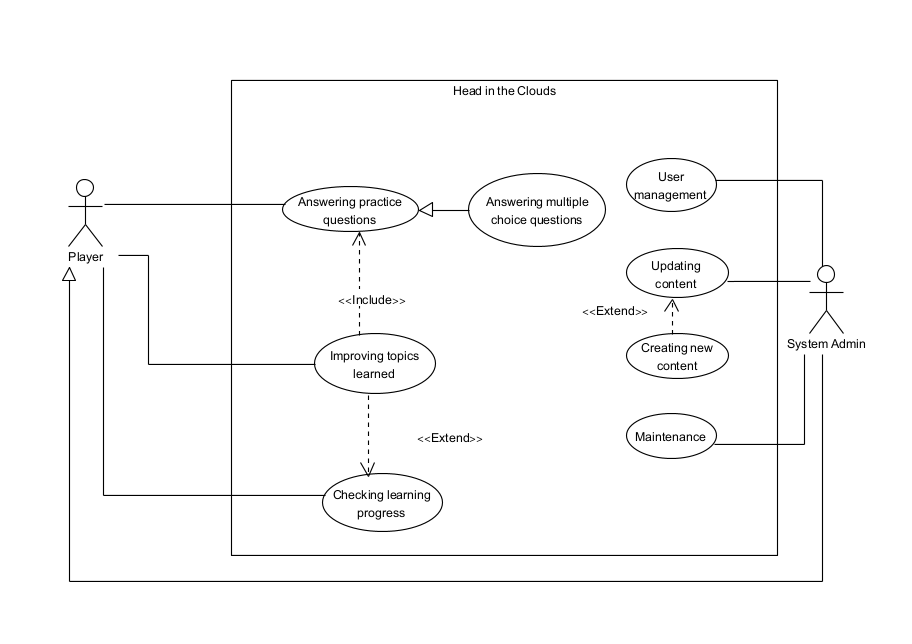
**דרישות פונקציונאליות:**

1. המערכת מאפשר צפייה בנושאי הקורס
2. המערכת תאפשר מענה על שאלות
3. המערכת תאפשר קבלת משוב באמצעות ציון
4. המערכת תאפשר צפייה ברמת ההתקדמות בקורס
5. המערכת תאפשר זכייה בתגי זכייה בקטגוריות שונות

**דרישות לא פונקציונאליות:**

1. לכל תרגיל יהיה ניקוד משלו (usability)
2. השאלות יהיו אמריקאיות (usability)
3. המשתמש יכול לזכות בתגי זכייה לפי הנקודות (usability)
4. המשתמש יוכל לבחור איזה נושא ללמוד (usability)
5. ניקוד המשתמש יאוחסן בדאטה בייס ויוצג כל פעם (backup)

**שאלה 3 – תרשים Use Case:**



**שאלה 4 – איטרציות עבודה:**

**טבלת SDP:**

|  |  |
| --- | --- |
| **Iteration 1** | **Implementing the basic logic of the Project** |
| **Task 1** | Implement opening screen with game instruction |
| **Task 2** | Build database with all the questions |
| **Task 3** | Implement question screen |
| **Task 4** | Implement manager control screen |
| **Task 5** | Implement end screen with results |
| **Iteration 2** | **Building a more full-featured version** |
| **Task 1** | implement extended feedback to each question |
| **Task 2** | implement login screen |
| **Task 3** | implement badges and achievements in user’s profile |
| **Task 4** | upgrade end screen to include graphs and stats |
| **Task 5** | testing and integration |

**איטרציה ראשונה:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Executor** | **Task** | **Requirement** | **Iteration** |
|  |  |  | **Iteration 1** |
| UI developer  אילנית | Show Profile screen | Start the application |  |
| Backend developer  **נופר** | Add current game info to DB |  |  |
| Frontend developer  **דבי** | Display info |  |  |
| QA **נוי** | Acceptance Test |  |  |
| **נוי** QA | Login to the system |  |  |
| UI developer  **נופר** | Show subjects to play | Start game |  |
| Backend developer  **נופר** | Display tutorials from DB | Watching tutorials |  |
| Backend developer  **אייל** | Display questions from DB | Answering questions |  |
| Backend developer  **אייל** | Compare chosen answer of the player to the correct answer from the DB |  |  |
| Frontend developer  **דבי** | Display feedback to player answer |  |  |
| Frontend developer  **דבי** | Display player’s progress | Shows player’s profile |  |
| UI developer  אילנית | Display badges and achievements |  |  |
| Backend developer  **אייל** | Adds questions to DB | Updating available topics and questions in the DB (manager) |  |
| Backend developer  **אייל** | Remove questions from DB |  |  |
| Backend developer  **דולב** | Adds topics to DB |  |  |
| Backend developer  **דולב** | Remove topics from DB |  |  |
| Backend developer  **דולב** | Update existing questions from the DB |  |  |

**איטרציה שנייה:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Executor** | **Task** | **Requirement** | **Iteration** |
|  |  |  | **Iteration 2** |
| Frontend developer  נופר | Show login screen | Start the application |  |
| דולב    Backend developer | Identify user type |  |  |
| UI developer  דבי | Design login screen |  |  |
| QA  אילנית | Acceptance Test |  |  |
| QA  אילנית | Login to the system |  |  |
| UI developer  דבי | Show subjects to play | Start game |  |
| Backend developer  דולב | Display questions from DB | Answering questions |  |
| Backend developer אייל | Compare chosen answer of the player to the correct answer from the DB |  |  |
| Frontend developer נופר | Display feedback to player answer |  |  |
| Frontend developer נופר | Display player’s progress | Shows player’s profile |  |
| UI developer דבי | Display badges and achievements |  |  |
| Backend developer דולב/אייל | Add more questions to DB | Updating available topics and questions in the DB (manager) |  |
| Backend developer אייל | Create users |  |  |
| Backend developer אייל | divide questions into topics |  |  |
| Scrum master נוי | Creating “Maintenance Guide” document for the developer | Reading “Maintenance  Guide” |  |
| Scrum master נוי | Creating “User Guide” document for the user | Watching “User Guide” tutorials |  |
| Scrum master נוי | Creating “User Guide” video for the user |  |  |
| QA אילנית/נוי | Final testing |  |  |

**שאלה 5:**

**סעיף א' – ציון SUS:**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **RAW SUS Score** | **SUS Mediate Score** |
| 5 | 1 | 5 | 2 | 5 | 1 | 5 | 1 | 5 | 1 | 39 | 97.5 |
| 4 | 1 | 4 | 1 | 4 | 1 | 4 | 1 | 5 | 5 | 32 | 80 |
| 5 | 1 | 4 | 1 | 5 | 1 | 4 | 2 | 4 | 1 | 36 | 90 |
| 3 | 2 | 4 | 1 | 4 | 2 | 4 | 2 | 4 | 2 | 30 | 75 |
| 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 |  | 3 | 33 | 82.5 |
| 5 | 1 | 4 | 1 | 4 | 1 | 5 | 1 | 5 | 1 | 38 | 95 |
| 3 | 1 | 3 | 1 | 3 | 1 | 3 | 1 | 4 | 1 | 31 | 77.5 |
| 5 | 5 | 5 | 1 | 2 | 1 | 5 | 1 | 4 | 1 | 32 | 80 |
| 3 | 2 | 5 | 2 | 1 | 1 | 5 | 1 | 5 | 5 | 28 | 70 |
| 3 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 4 | 1 | 37 | 92.5 |
| 5 | 1 | 5 | 5 | 4 | 1 | 5 | 1 | 5 | 5 | 31 | 77.5 |
| 1 | 2 | 5 | 1 | 4 | 1 | 5 | 1 | 5 | 1 | 34 | 85 |
| 4 | 1 | 4 | 1 | 4 | 1 | 5 | 1 | 5 | 2 | 36 | 90 |
| 4 | 1 | 4 | 1 | 4 | 1 | 4 | 1 | 4 | 1 | 35 | 87.5 |
| 4 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 39 | 97.5 |
| 3 | 3 | 3 | 3 | 3 | 4 | 3 | 2 | 3 | 2 | 21 | 52.5 |
| 3 | 3 | 4 | 3 | 3 | 3 | 4 | 2 | 3 | 3 | 23 | 57.5 |
| 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 40 | 100 |
| 3 | 1 | 4 | 3 | 4 | 2 | 2 | 2 | 3 | 2 | 26 | 65 |
|  | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 5 | 31 | 77.5 |
| 4 | 2 | 4 | 5 | 5 | 1 | 5 | 2 | 3 | 1 | 30 | 75 |
| 3 | 4 | 4 | 1 | 4 | 2 | 5 | 1 | 4 | 1 | 31 | 77.5 |
| 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 40 | 100 |
| 4 | 4 | 4 | 1 | 4 | 1 | 4 | 1 | 4 | 1 | 32 | 80 |
| 5 | 1 | 4 | 1 | 5 | 1 | 4 | 1 | 4 | 1 | 37 | 92.5 |
| 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 40 | 100 |
| 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 40 | 100 |
| 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 40 | 100 |
| 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 5 | 1 | 40 | 100 |
| 2 | 2 | 2 | 3 | 3 | 2 | 2 | 3 | 2 | 3 | 18 | 45 |
| 4 | 1 | 3 | 1 | 5 | 1 | 5 | 1 | 4 | 2 | 35 | 87.5 |

**Final SUS Score = 83.47**

**סעיף ב' – תיק למתכנת:**

**Maintenance Guide**

Head in the clouds Game is a Python application designed for students to test their knowledge in the field of Cloud Computing. It's written using Jupyter notebooks on Google Colab, with a combination of IPython widgets for the user interface.

**Database environment**

The system links with Google Firebase via the firebase module in the Main file. For maintenance and updates to the Firebase database, the user requires Firebase Admin SDK access.

**Special Requirements**

1. **Google Account**

a Google account is needed to log in to Google Colab. Once logged in, you can load the code into a new Python notebook.

1. **Firebase Connection**

The code connects to a Firebase database. Google Colab does not store environmental variables between sessions. The code requires secure access to Firebase, uploading these credentials is needed each time a new Colab session is started.

1. **Internet Connection**

Because Google Colab runs on the cloud, an active internet connection is required to run the cells in the notebook.

1. **Browser**

Google Colab works best on the latest versions of Chrome, Firefox, and Safari. Ensure your browser is up to date for the best performance.

**Main files and methods**

1. Welcome Screen

* **createWelcomeScreen()**: This function creates the initial welcome screen for the game.

1. Manager Screen

* **createManagerScreen():** This function creates the Manager screen, providing options to Add, Remove, or Update quizzes.
* **handleAddQuizButton(b):** This function handles the event when the "Add Quiz" button is clicked.
* **handleUpdateQuizButton(b):** This function handles the event when the "Update Quiz" button is clicked.
* **handleRemoveQuizButton(b):** This function handles the event when the "Remove Quiz" button is clicked.

1. Player Screen

* **createPlayerScreen():** This function creates a Player screen where players can view their level and start playing the game.
* **handle\_start\_play\_button\_click(b):** This function handles the event when the "Start Play" button is clicked.

1. Topic Selection Screen

* **createSelectionScreen():** This function creates a screen for players to select a topic for the quiz.
* **handle\_topic\_selection(button):** This function handles the event when a topic is selected.

1. Exit Buttons

* **exit\_selection\_handler(b):** This function handles the event when the exit button is clicked on the selection screen.
* **exit\_player\_handler(b):** This function handles the event when the exit button is clicked on the player screen.

**Design Patterns**

1. **Modular Design:**

The application is built using a modular approach, where each screen is a separate function. This allows easy maintenance, as changes to one screen do not affect others.

1. **Event-Driven Programming**

The game leverages an event-driven programming model. Actions in the game (like button clicks) trigger events which are then handled by specific functions.

1. **Functional Programming**

The application extensively uses Python's functional programming features. Functions are stateless, and output depends only on the input.

**Maintenance Instructions**

1. **Adding more quizzes**

To add more quizzes, add them through the UI using the manager account or add quizzes through the database on Firebase.

1. **Changing the game layout**

To change the game layout or styling, modify the HTML/CSS code in the functions.

1. **Updating the logic**

To update game logic, identify the function responsible for that part of the game and make changes there.

**Additional Notes**

1. **Refactoring**

The application has been built with functions for modularity. As the complexity grows, consider refactoring to classes for better encapsulation and readability.

**סעיף ג' – תיק למשתמש:**

**User Guide**

The game is an interactive quiz on Cloud Computing concepts. It is designed to challenge players to answer a series of questions on various topics relating to cloud computing. As players progress, they level up, earning various achievements along the way.

**Game Screens**

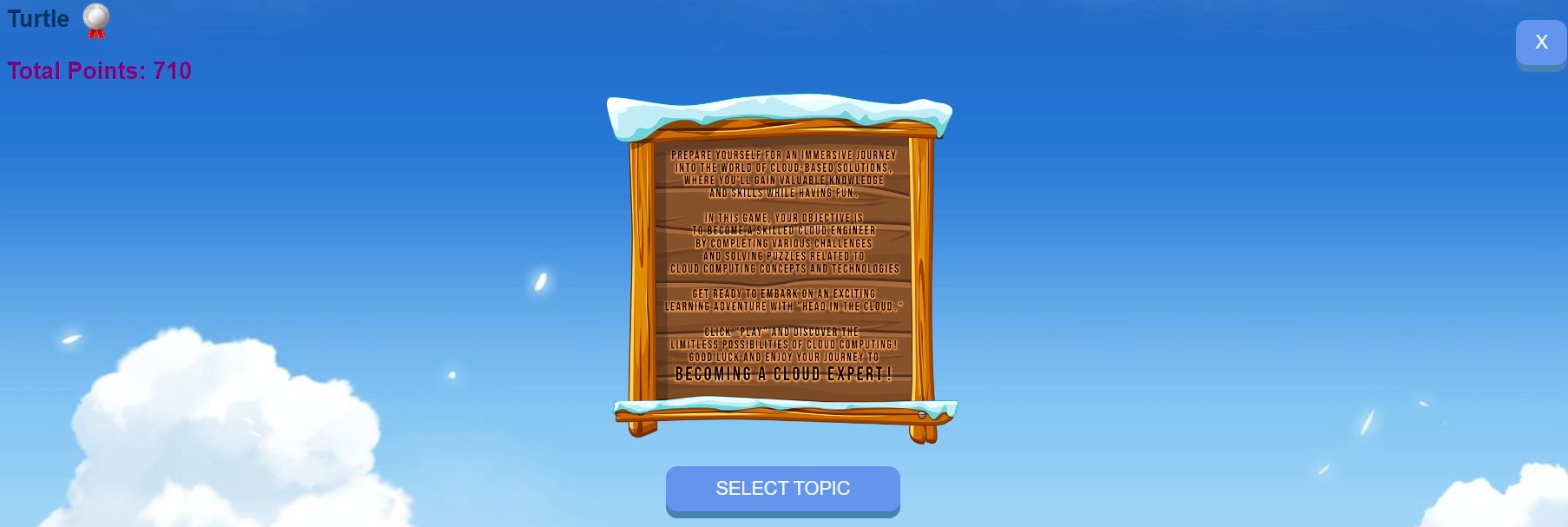
The opening Screen is the Welcome Screen:

1. Enter Username in the first field.
2. Enter password in the second field.
3. Click on 'Login'.
4. If wrong details are entered, a message: 'Invalid username or password' is displayed. Please try again with correct details.



Upon successful login with player logging in, The Player screen opens:

1. User's nickname and level are presented on the left-most top corner with a bronze, silver or golden medals for each level. Also, the player's point level is presented.
2. Click on "Select topic" for topic selection.
3. If you wish to return to the Welcome Screen, click the exit button "X" on the right-most top corner.

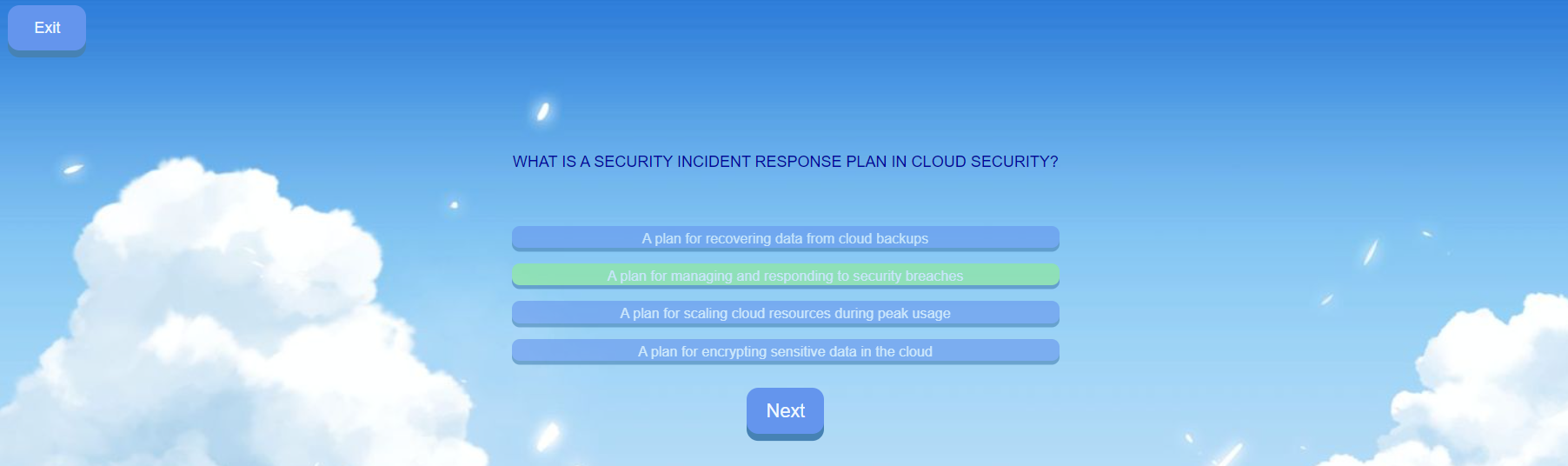
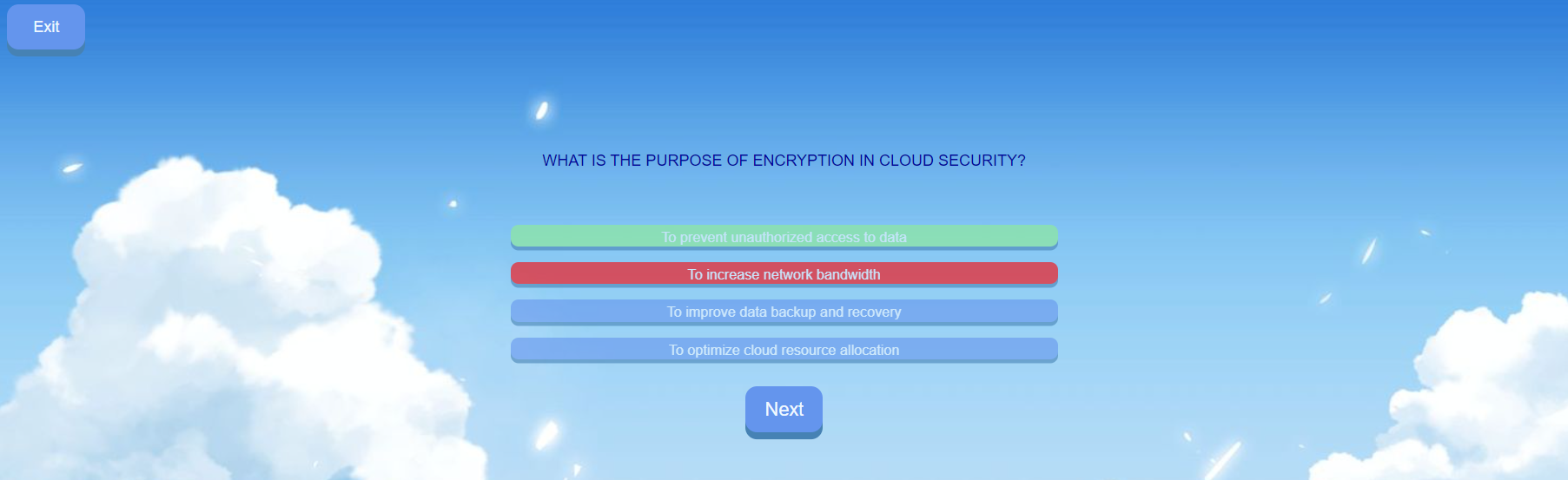


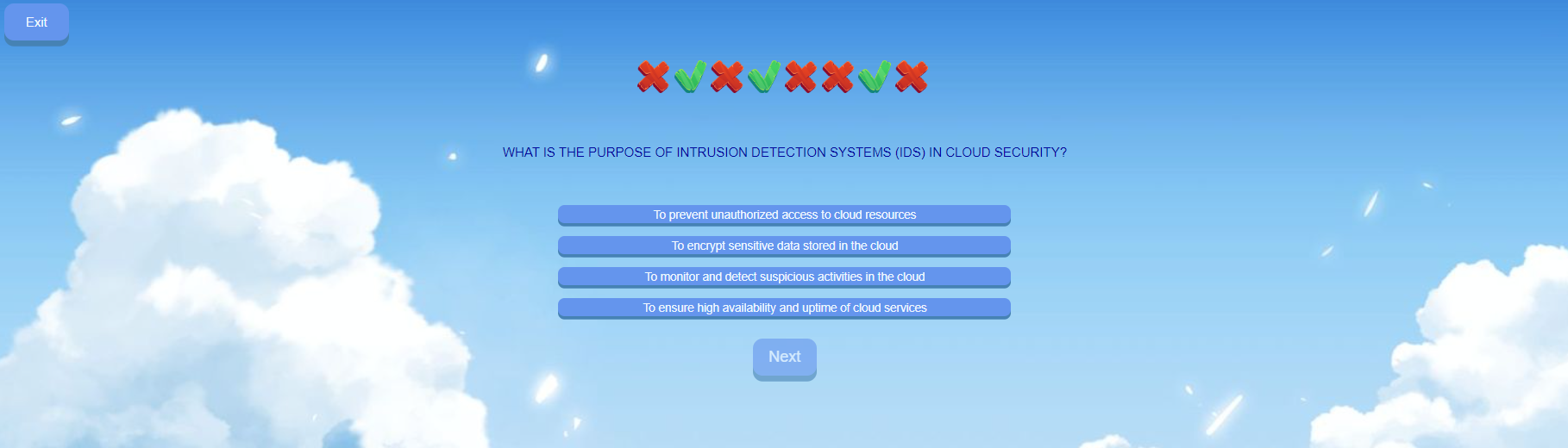
The Topic Selection screen is displayed, and players can select a topic for the quiz.

1. Click on the desired topic from the topic list.
2. If you wish to return to the Player Screen, click the exit button "X" on the right-most top corner.



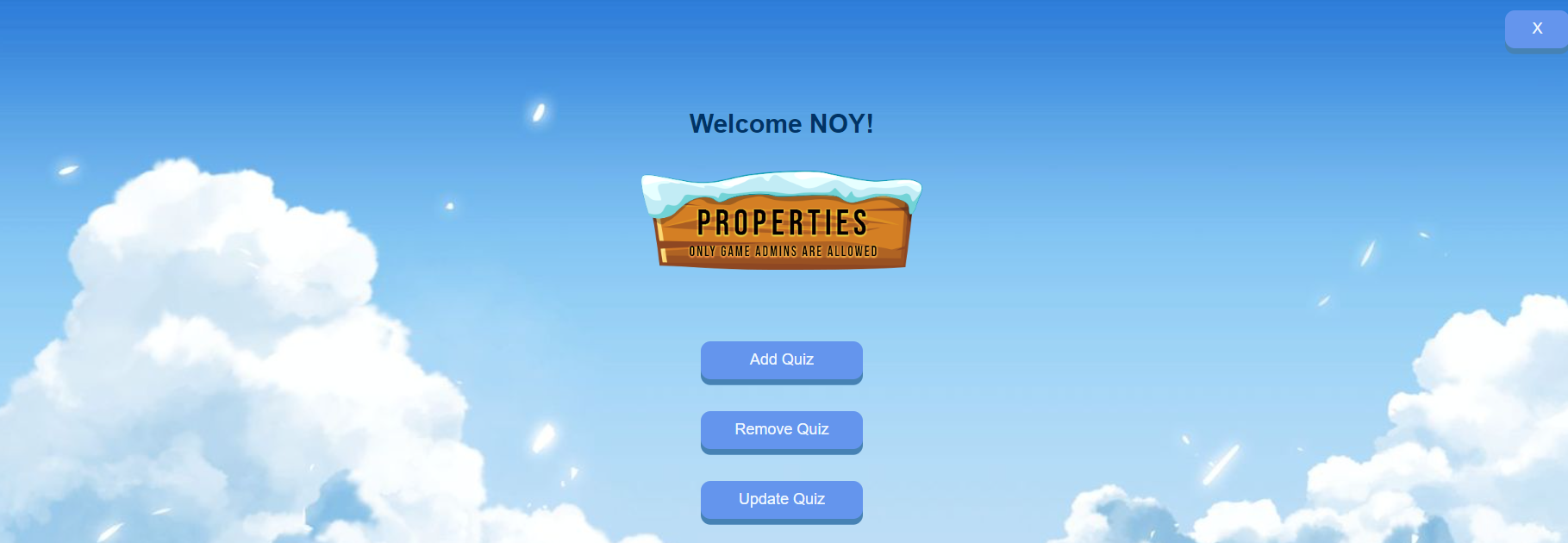
The Quiz Screen is presented:

1. Click on the answer you consider as right.
2. If player is correct, answer will be highlighted in green, and the player score is increased.  
   
3. If player is mistaken, the wrong answer is highlighted in red, and the right answer is highlighted in green.
4. Click "Next" to display the next question and view the progress bar grow.



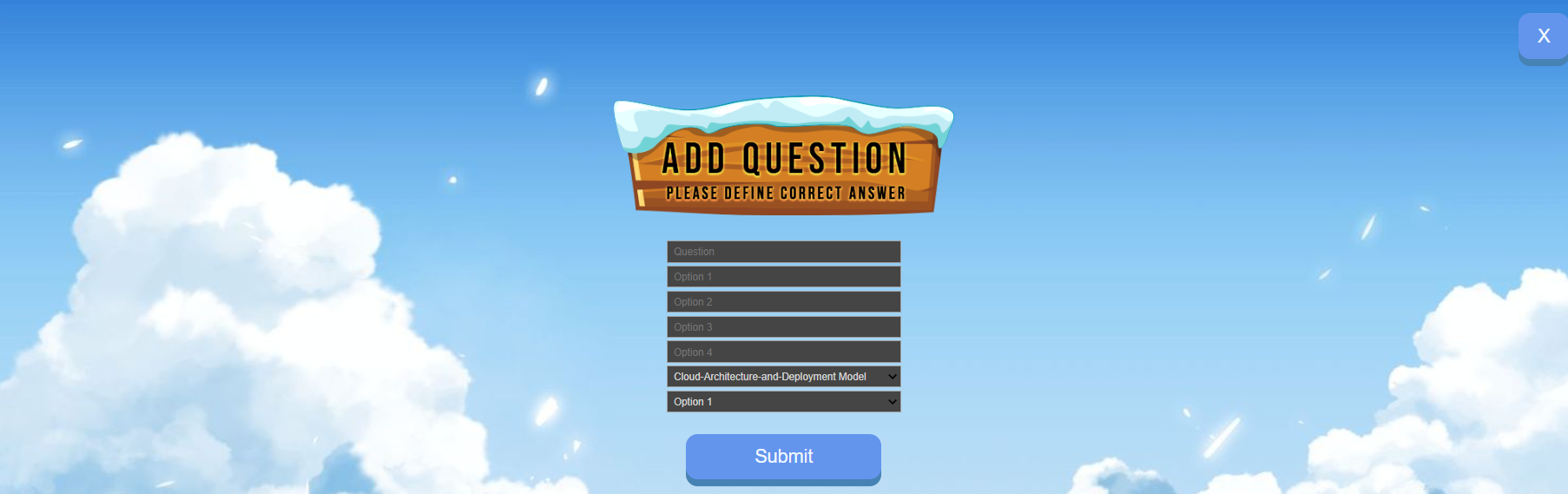
1. Keep answering until 10 questions are answered. The player is automatically transitioned to the game statistics screen.
2. Click "Back" to view the player screen.
3. Click "Try Again" to view the topic selection screen.



Upon successful login with manager logging in, The manager screen opens:  
  


Add Quiz:

1. Click on "Add Quiz" button on the manager screen.
2. Enter a question in the Question field.
3. Enter 4 possible answers. Make sure only one is correct.
4. Choose a topic the question relates to
5. Define which answer is correct.
6. Click "Submit"
7. If you wish to return to the Manager Screen, click the exit button "X" on the right-most top corner.



Remove Quiz:

1. Click on "Remove Quiz" on the manager screen.
2. Choose the topic of the question you wish to remove using the dropdown.
3. Click the "Remove" button next to the desired question.
4. If you wish to return to the Manager Screen, click the exit button "X" on the right-most top corner.



Update Question:

1. Click on "Update Quiz" on the manager screen.
2. Choose the topic that the question you wish to update relates to
3. Choose the question you wish to update from the dropdown.
4. The 4 possible answers will be visible, make the relevant corrections.
5. Define the correct answer in the last text field.
6. Click "update".
7. If you wish to return to the Manager Screen, click the exit button "X" on the right-most top corner.



**Transitions between Screens**

Transitions between screens are mainly handled by button clicks.  
Clicking the "Play" button on the Welcome Screen leads to the Player Screen.  
The "Select Topic" button on the Player Screen transitions to the topic-Selection Screen.  
Each topic button on the Selection Screen starts a quiz on the respective topic.  
After the quiz, the player is taken to the End Game Screen. In addition, there are "exit" buttons on each screen (excluding the Welcome Screen), which return to the previous screen.

**Explanation of Possible Mistakes**

1. **Username/Nickname Missing**  
   Players must enter a username and a nickname before starting the game. An error message will be displayed if one or both fields are left empty.
2. **Unsaved Progress**  
   Google Colab does not save variable values between sessions. If a player closes the game midway and reopens it later, their progress will be lost.
3. **Firebase Connection**  
   If the Firebase connection fails, the player's scores and progress will not be saved. Ensure you have a stable internet connection.
4. **Google Colab Session**  
   Google Colab disconnects after a period of inactivity. In case of a disconnection, you may need to rerun some cells to reinitialize the game.

**סעיף ד' – קישור ל-GIT:**

https://github.com/EyalKoubi/CloudProject

**שאלה 6 – אתגרים:**

1. סביבת עבודה חדשה, Google Colab - סביבת העבודה לא הייתה מותאמת מספיק לבניית אתרי אינטרנט (בניגוד לסביבות כמו visual studio code לדוגמא).
2. קושי בשילובים בין רכיבים של html ו-css לפונקציות ב-Python.
3. קושי כללי בניהול הארכיטקטורה של הקוד כולו ב-Colab - קושי בקישוריות בין הפונקציות והמעבר בין המסכים.
4. בחלוקה של כתיבת הקוד, אי תיאום בין סגנון הקישוריות בין המסכים, דבר שהקשה על מיזוג הקוד לקוד אחיד.