Bilkent University

Department of Computer Engineering

CS 353 Term Project

CSCareer: A Quiz Based Hiring System

Final Report

Group 22: Hamza PEHLİVAN, Meryem EFE, Fırat YÖNAK, Selen UYSAL

Instructor: Özgür ULUSOY

Teaching Assistant: Arif USTA

Contents

Final Report

CSCareer: A Quiz Based Hiring System

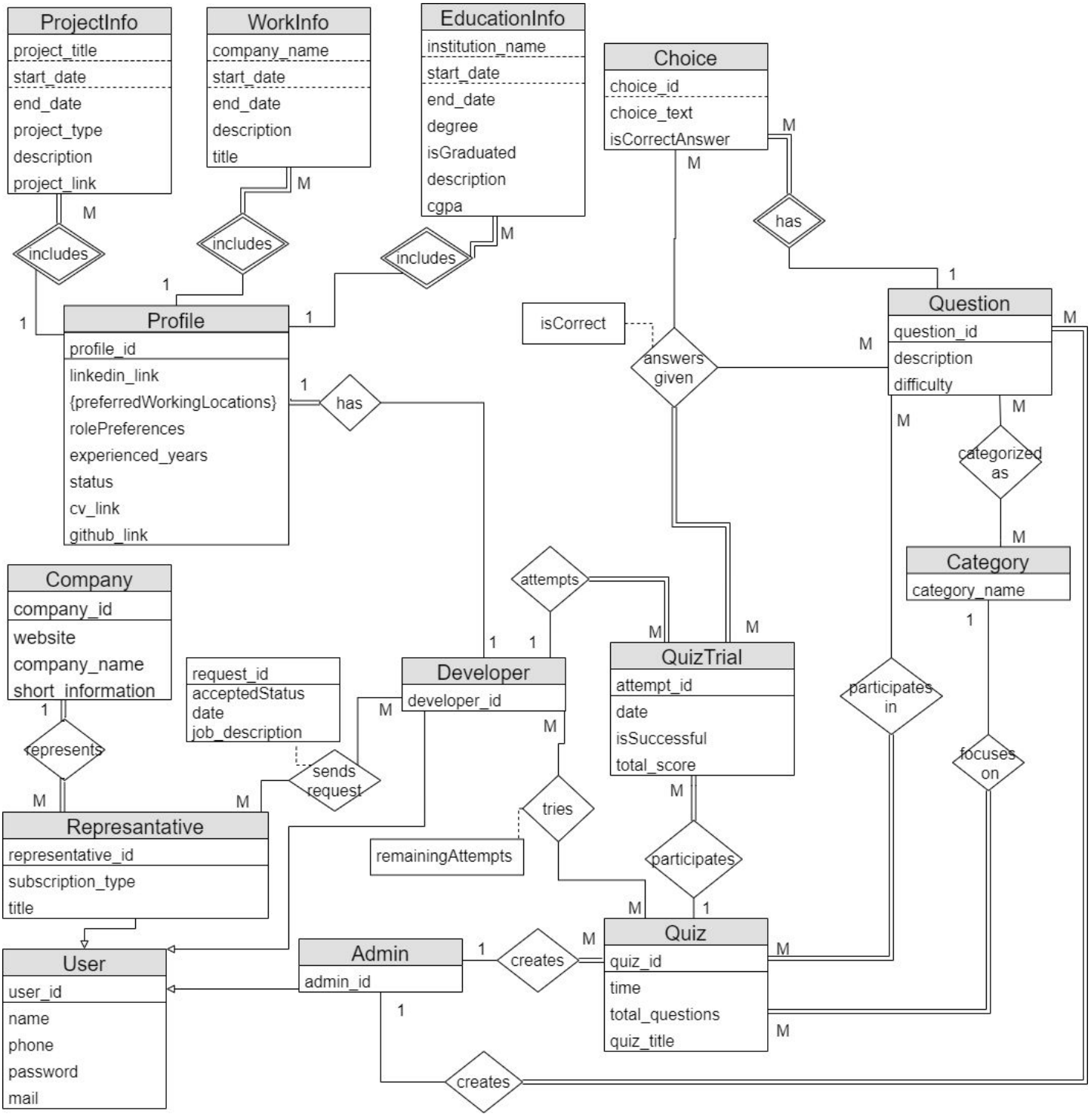
1. Project Description

CSCareer is a quiz based hiring system which basically includes quizzes related to computer science and provides technology companies with quizzes' results to hire qualified employees.

There are two parts of this system which are client side and admin side. In the admin side, the system provides admins with a web-based interface to create quizzes in different categories which include computer science related multiple choice questions.

On the client side of the system, there are mainly two functionalities. Firstly, developers who look for a job or want to prove themselves should sign up to the system. Then, they can take quizzes to demonstrate themselves. It is also important to note that the developers have 3 attempts for each quiz. Developers can also add education, work, and project information to their profiles. Representatives who want to hire a developer look at the quiz results and can send an interview request to one they chose on behalf of their companies. This request may be accepted or declined by the developers.

2. E/R Model


ProjectInfo
project_title
start_date
end_date
project_type
description
project_link
WorkInfo
company_name
start_date
end_date
description
title
EducationInfo
institution_name
start_date
end_date
degree
isGraduated
description
cgpa
Choice
choice_id
choice_text
isCorrectAnswer
includes
includes
includes
has
Profile
profile_id
linkedin_link
{preferredWorkingLocations}
rolePreferences
experienced_years
status
cv_link
github_link
has
isCorrect
answers given
Question
question_id
description
difficulty
categorized as
Category
category_name
Company
company_id
website
company_name
short_information
represents
request_id
acceptedStatus
date
job_description
sends request
attempts
Developer
developer_id
QuizTrial
attempt_id
date
isSuccessful
total_score
participates in
focuses on
tries
remainingAttempts
participates
Represantative
representative_id
subscription_type
title
User
user_id
name
phone
password
mail
Admin
admin_id
creates
Quiz
quiz_id
time
total_questions
quiz_title
creates

3. Relation Schemas

3.1. User

Relational Model:

user (user_id, email, password, fullname, phone)

3.2. Developer

Relational Model:

developer (developer_id, profile_id)

FK: developer_id to user(user_id)

FK: profile_id to profile(profile_id)

3.3. Representative

Relational Model:

representative (representative_id, subscription_type, title, company_id)

FK: representative_id to user(user_id)

FK: company_id to company(company_id)

3.4. Company

Relational Model:

company (company_id, website, company_name, short_information)

3.5. Admin

Relational Model:

admin (admin_id)

FK: admin_id to user(user_id)

3.6. Profile

Relational Model:

profile (profile_id, linkedin_link, experienced_years, status, cv_link, github_link, role_preferences)

3.7. PreferredWorkingLocations

Relational Model:

preferredworkinglocations (profile_id, city)

FK: profile_id to profile(profile_id)

3.8. EducationInfo

Relational Model:

educationinfo (profile_id, instution_name, start_date, end_date, degree, cgpa, isGraduated, description)

FK: profile_id to profile(profile_id)

3.9. WorkInfo

Relational Model:

workinfo (profile_id ,company_name, start_date, end_date, title, description)

FK: profile_id to profile(profile_id)

3.10. ProjectInfo

Relational Model:

projectinfo (profile_id, project_title, start_date, end_date, project_type, description, project_link)

FK: profile_id to profile(profile_id)

3.11. Question

Relational Model:

question (question_id, admin_id, description, difficulty)

FK: admin_id to admin(admin_id)

3.12. Choice

Relational Model:

choice (question_id, choice_id, choice_text, isCorrectAnswer)

FK: question_is to question(question_id)

3.13. Category

Relational Model:

category (category_name)

3.14. CategorizedAs

Relational Model:

categorized_as (category_name, question_id)

FK: category_name to category(category_name)

FK: question_id to question(question_id)

3.15. Quiz

Relational Model:

quiz (quiz_id, admin_id, category_name, total_questions, quiz_title, time)

FK: admin_id to admin(admin_id)

FK: category_name to category(category_name)

3.16. Quiz_questions

Relational Model:

quiz_questions (quiz_id, question_id)

FK: quiz_id to quiz(quiz_id)

FK: question_id to question(question_id)

3.17. QuizTrial

Relational Model:

quiztrial (attempt_id, developer_id, quiz_id, date, isSuccessful, total_score)

FK: developer_id to developer(developer_id)

FK: quiz_id to quiz(quiz_id)

3.18. Answer

Relational Model:

answer (attempt_id, question_id, choice_id, isCorrect)

FK: attempt_id to quiztrial(attempt_id)

FK: question_id to question(question_id)

FK: choice_id to choice(choice_id)

3.19. Tries

Relational Model:

tries (developer_id, quiz_id, remaining_attempts)

FK: developer_id to developer(developer_id)

FK: quiz_id to quiz(quiz_id)

3.20. Request

Relational Model:

request (request_id, acceptedStatus, date, job_description, developer_id, representative_id)

FK: developer_id to developer(developer_id)

FK: representative_id to developer(representative_id)

4. Implementation Details

In this project, there are mainly two parts which are functionality and user interface. In the functionality part, MySQL is used for the database while PHP is used for the server side.

For designing the UI, we used HTML to create the general structure of the website. To add and edit styles to the website, we used CSS. Specifically, we preferred Bootstrap 4.5.0 so that we can provide better design. Lastly, we used JavaScript in order to manage the behaviour of the website such as alerts. We also used JQuery library to implement Ajax methods.

We shared the work as offered in the Project Functionality Document. According to the plan, each team member is responsible from one part as follows:

- Meryem Efe → Common functionalities & Additional requirements
- Hamza Pehlivan → Developer takes quiz.
- Selen Uysal → Admin prepares questions.
- Fırat Yönak → Representative sends interview request.

During implementation, we faced some problems. These problems are generally because we are not familiar with web development and using different languages together.

Firstly, in some pages, we had to embed HTML into PHP code. For instance, in the developer's profile page, education / work / project info should be refreshed and seen on the screen when a new one is added. To do that, we take information from the server and show them on the website. Therefore, we used HTML and PHP as embedded.

Secondly, even though we used GitHub actively, we faced some problems while finalizing the website by combining our parts. There were some syntax errors in SQL statements. This is because we used localhost. Therefore, we could not update the

database when we made some typing mistakes. Each group member had to do the same changes locally in case an error occurs.

5. Advanced Database Features

Reports:

- This report is used when a representative sends an interview request to a developer. It ensures that the new request's id is different from other ids of requests.

SELECT MAX(request_id) AS max_id FROM request

- This report is used to display results of the developers.

SELECT category_name, SUM(CASE WHEN isCorrect = 1 THEN 1 ELSE -1 END) AS score FROM answer NATURAL JOIN categorized_as WHERE attempt_id = current_attempt GROUP BY category_name

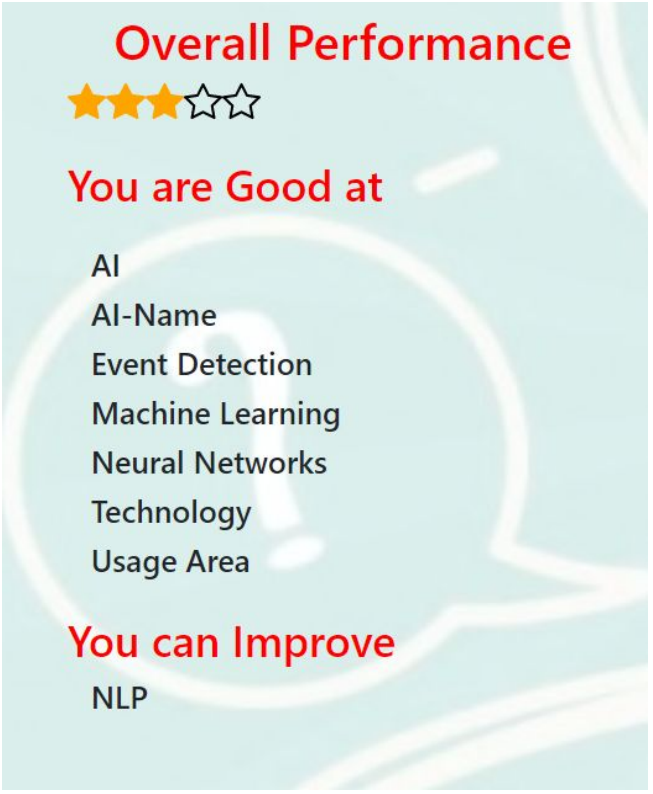


Figure 1: Report to Display Results

Trigger: When a new developer signs up to the system, s/he should be assigned quizzes with each category. This procedure is done using a trigger after insertion in the Developer table.

Figure 2: Trigger Result Screen

6. User Manual

6.1. Developer User Manual

Sign In / Sign Up

Firstly, if a developer does not have an account, s/he should sign up to the system from the register page. After the developer signs up, CSCareer directs her/him to the developer homepage.

If the developer has already had an account, s/he can sign in to the system from the login page.

Profile Page

After the developer signs in, s/he can see and edit profile information from the profile page. In this page, the developer can perform the following functions:

- Change Password

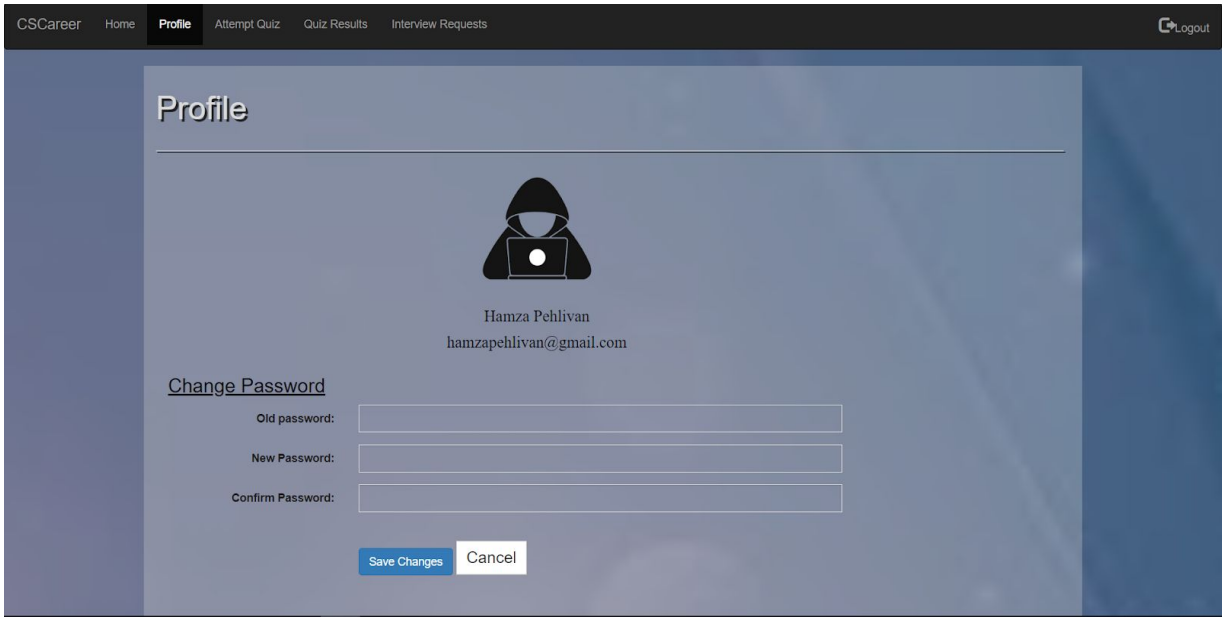


Figure 3: Change Password of Developer

- Edit General Information

In general information part, the developer can add or change phone number, linkedin link, github link, cv link. Also, s/he can indicate the number of years of experience, role preferences and whether s/he is open to new positions.

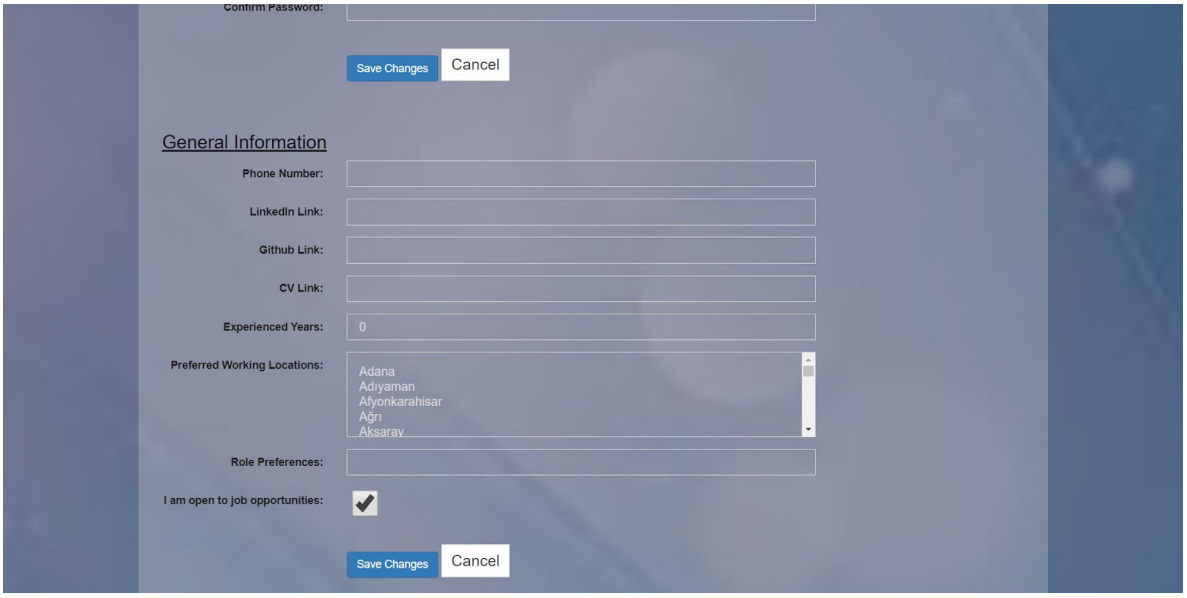


Figure 4: Edit General Information Part

- Add / Edit Education Information

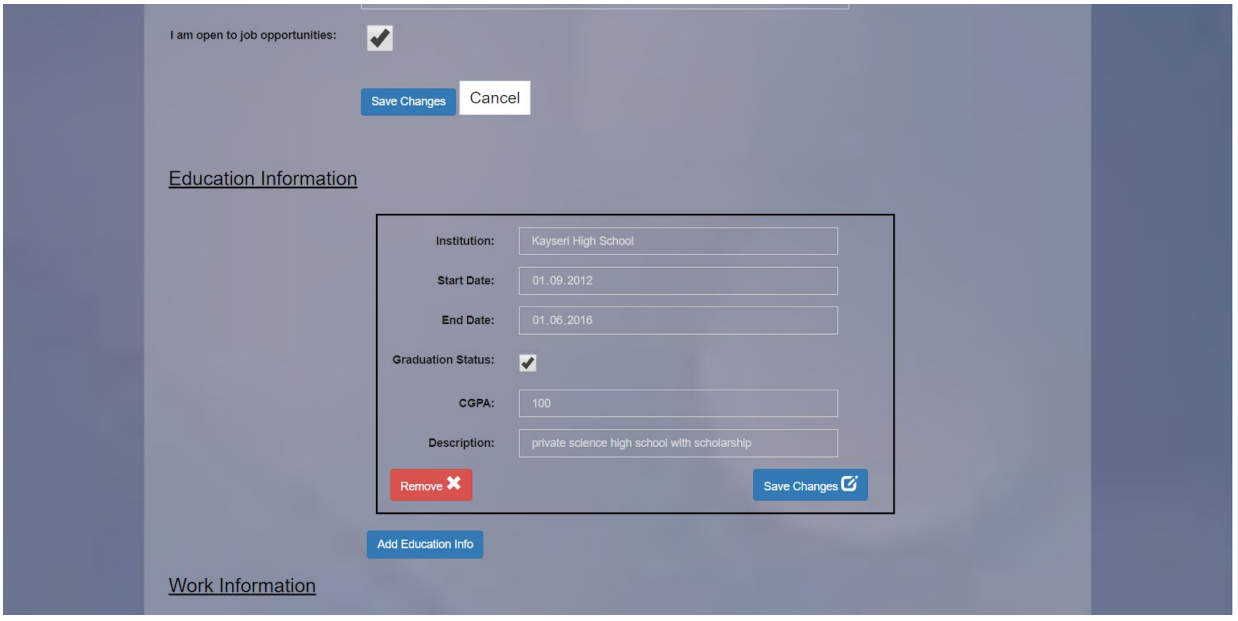


Figure 5: Add / Edit Education Info Part

- Add / Edit Work Information

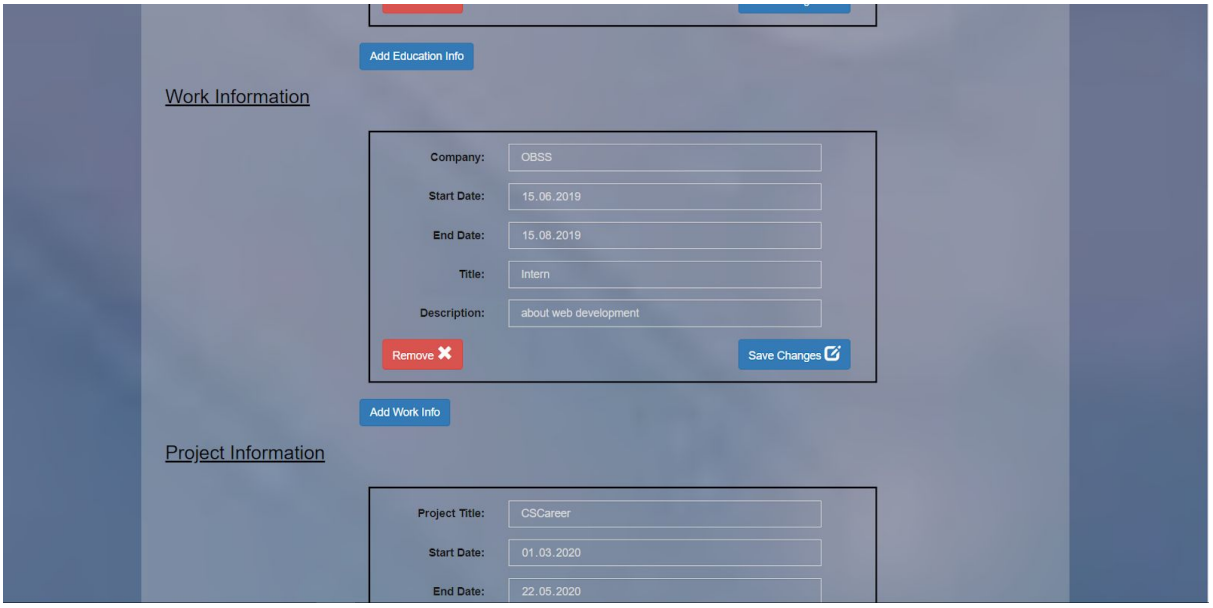


Figure 6: Add / Edit Work Info Part

- Add / Edit Project Information

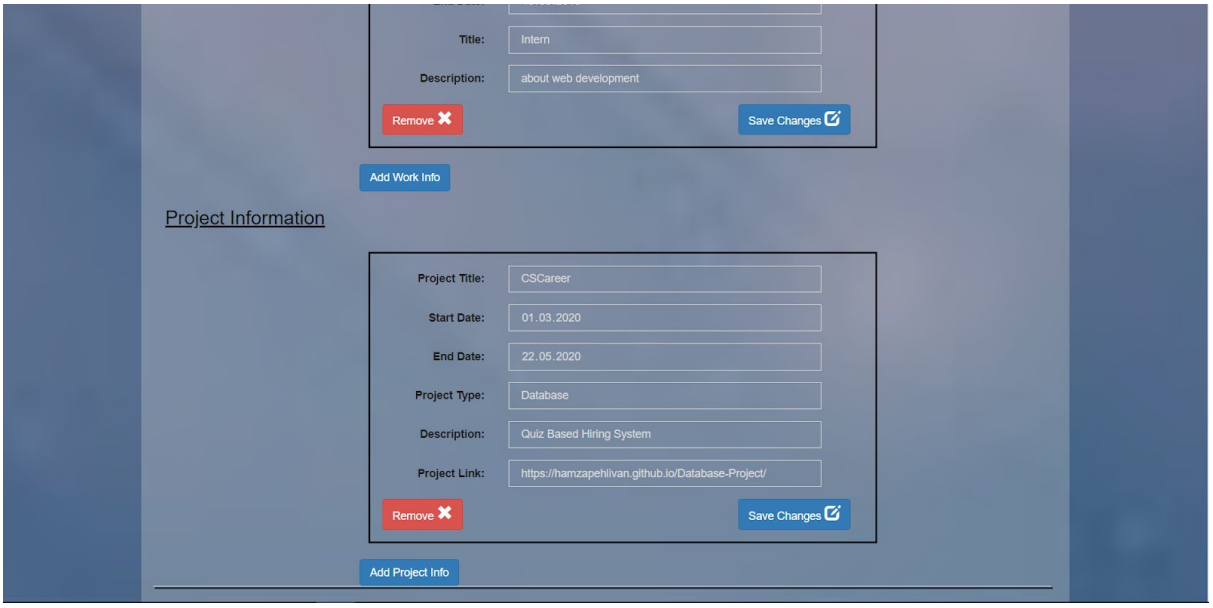


Figure 7: Add / Edit Project Info Part

Take Quiz

The developer can take a quiz from the Attempt Quiz page. In this page, s/he will see the categories. Then, the developer can start a quiz by clicking the category which s/he wants to take.

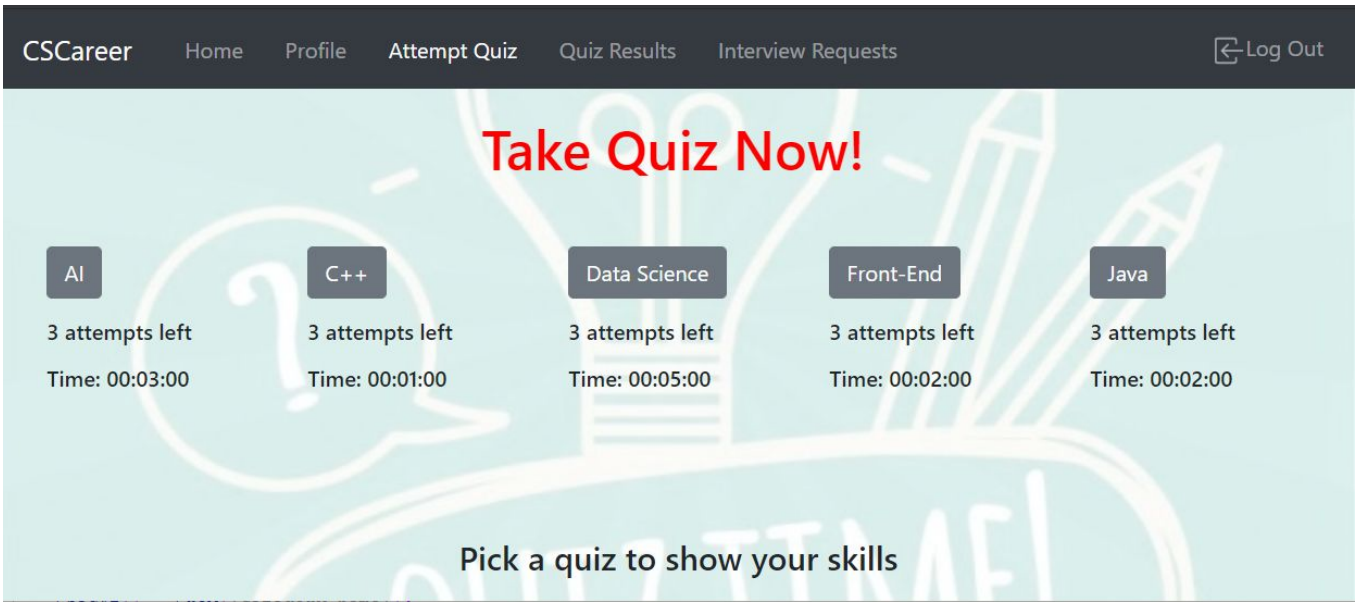


Figure 8: Attempt Quiz Page

Solve Questions

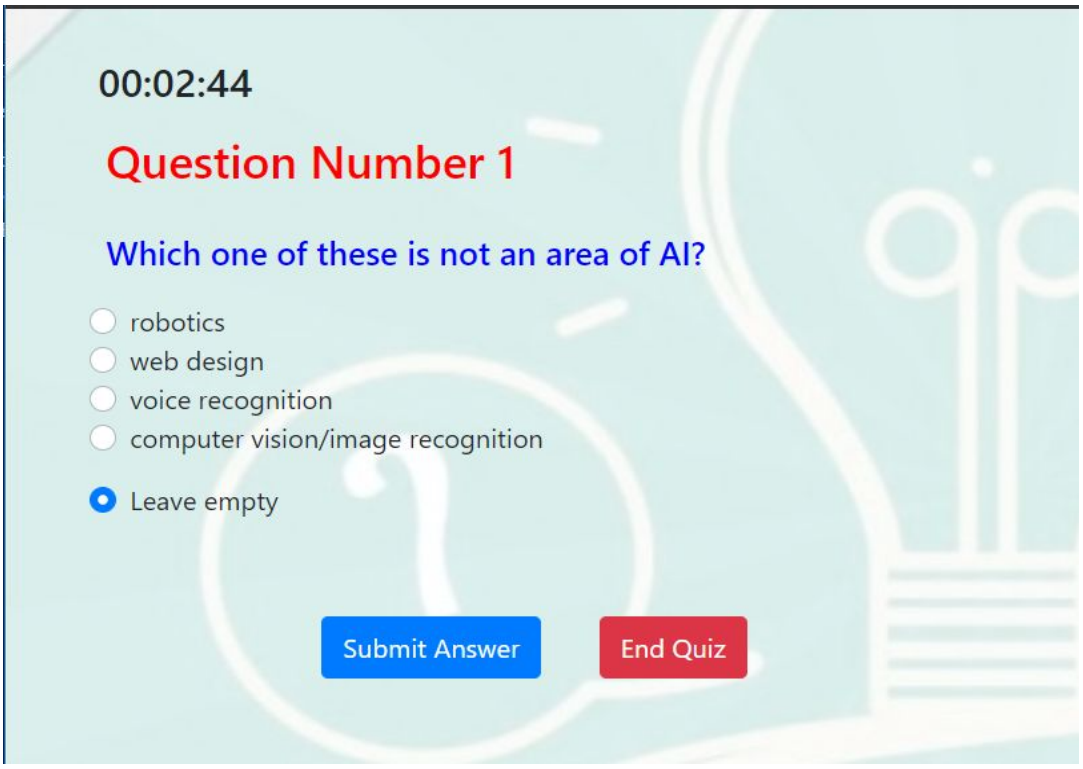


Figure 9: Solve Questions

Developer Views Results

Developers can view detailed results of attempted quizzes. They can also view which categories they are good at.

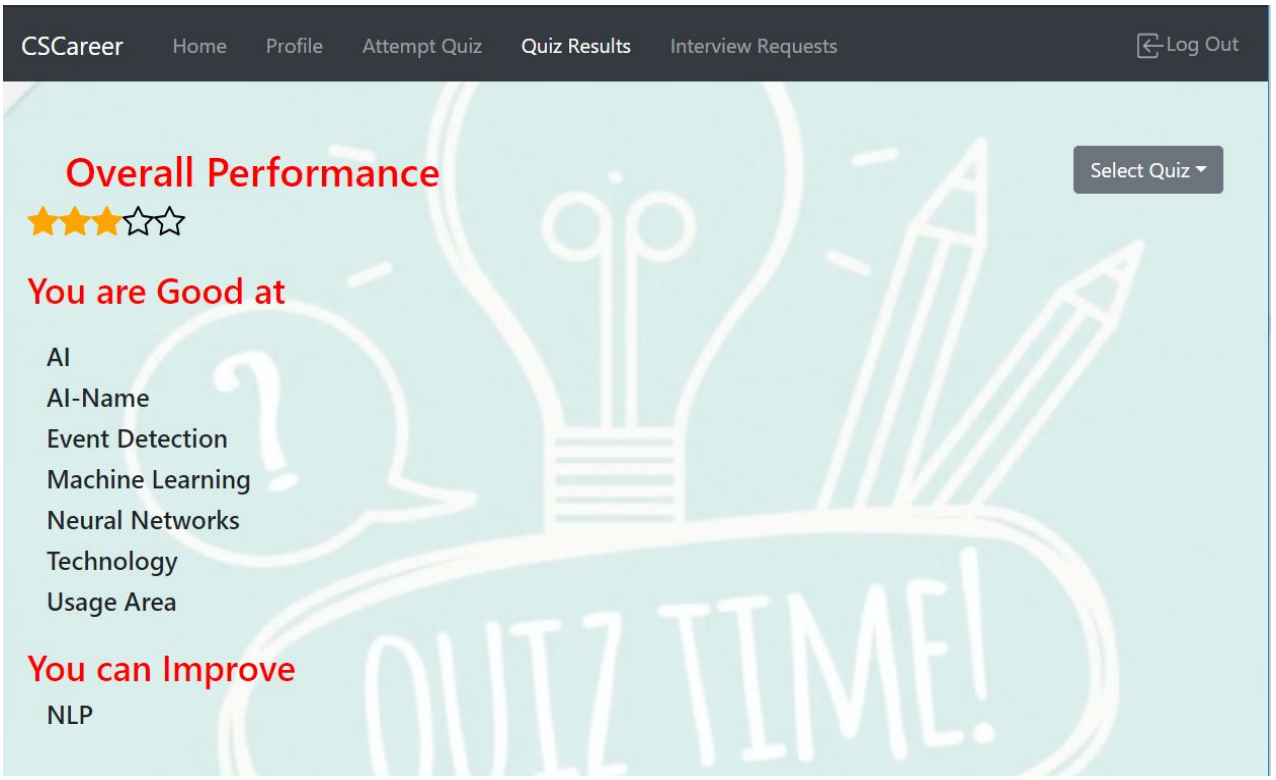


Figure 10: View Results

See / Accept / Decline Interview Requests

Developers can see interview requests which are sent by any representative in the system until the due date of the request. They can accept or decline the requests.

Figure 11: Developers' Interview Requests

6.2. Representative User Manual

Sign In / Sign Up

Firstly, if a representative does not have an account, s/he should sign up to the system from the register page. After the representative signs up, CSCareer directs her/him to the representative homepage.

If the representative has already had an account, s/he can sign in to the system from the login page.

Profile Page

After the representative signs in, s/he can see and edit profile information from the profile page. In this page, the representative can perform the following functions:

- Change Password
- Edit General Information

View Quizzes & Send Interview Requests

When a representative clicks view quizzes from the navigation bar, s/he will see this page which includes all quizzes in the website by their categories.

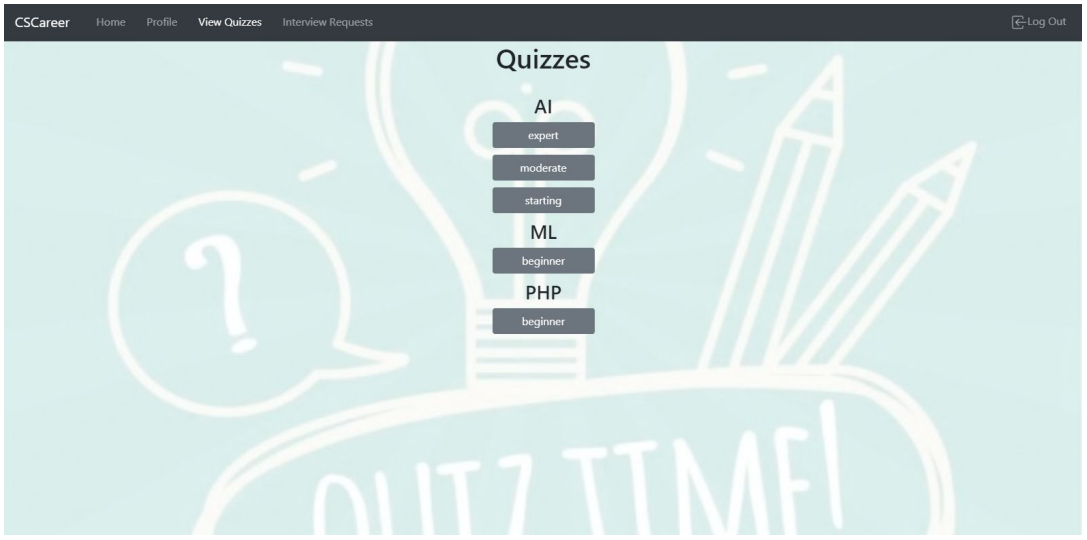


Figure 12: View Quizzes

After a representative selects a quiz from the first page, s/he will be able to see the scores of developers. They are ordered by their score. The representative can send an interview request to any developer who is available on the result table from this page by clicking send request by next to the results.

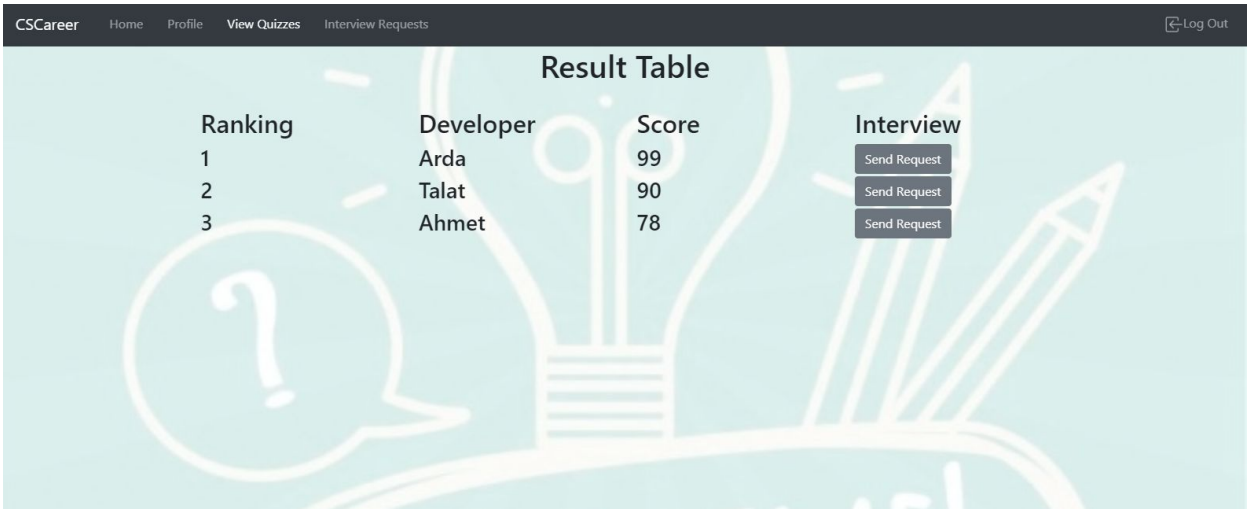


Figure 13: Rank of Developers

On this page the representative needs to fill the form to send an interview request to the developer. After clicking the submit button, the request is sent.

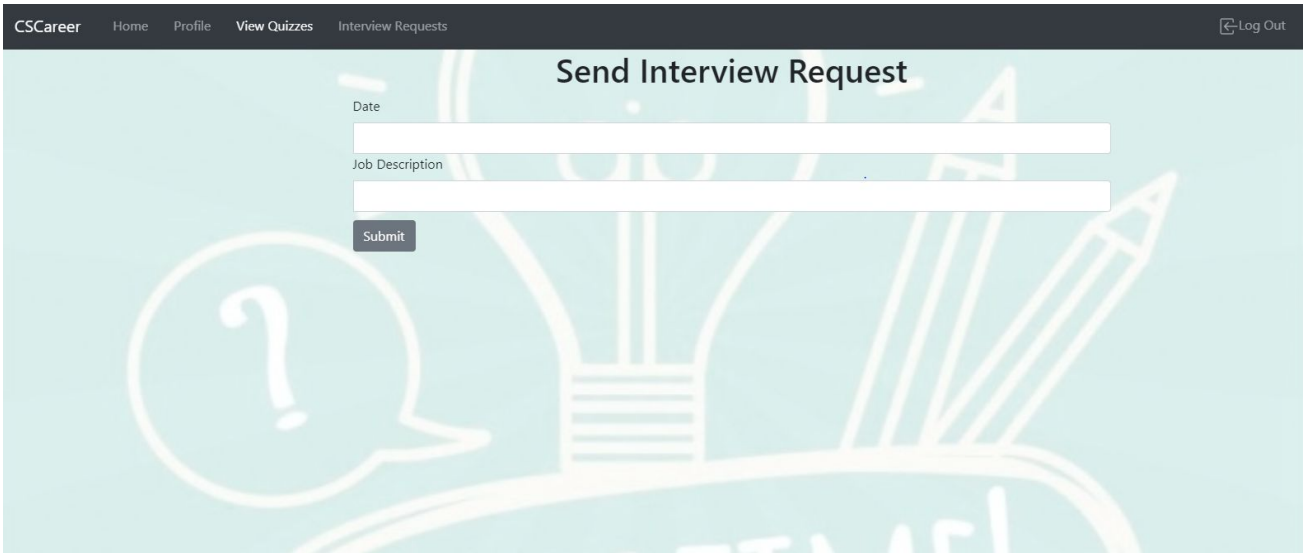


Figure 14: Send Interview Request

View Sent Interview Requests

When a representative selects sent interview requests from navigation bar, s/he can see the requests which his or her sent to developers and the status of the request.

Figure 15: View Sent Requests

6.3. Admin User Manual

Sign In

Admin can directly sign in to the system from the login page. (Only founders of CSCareer can be admin; therefore, there is no *sign up* opportunity for admins.)

Profile Page

After the admin signs in, s/he can see and edit profile information from the profile page. In this page, the admin can perform the following functions:

- Change Password
- Change Phone Number

Create / Edit Quizzes Page

In this page, admin can edit or delete a selected quiz or create a new one.

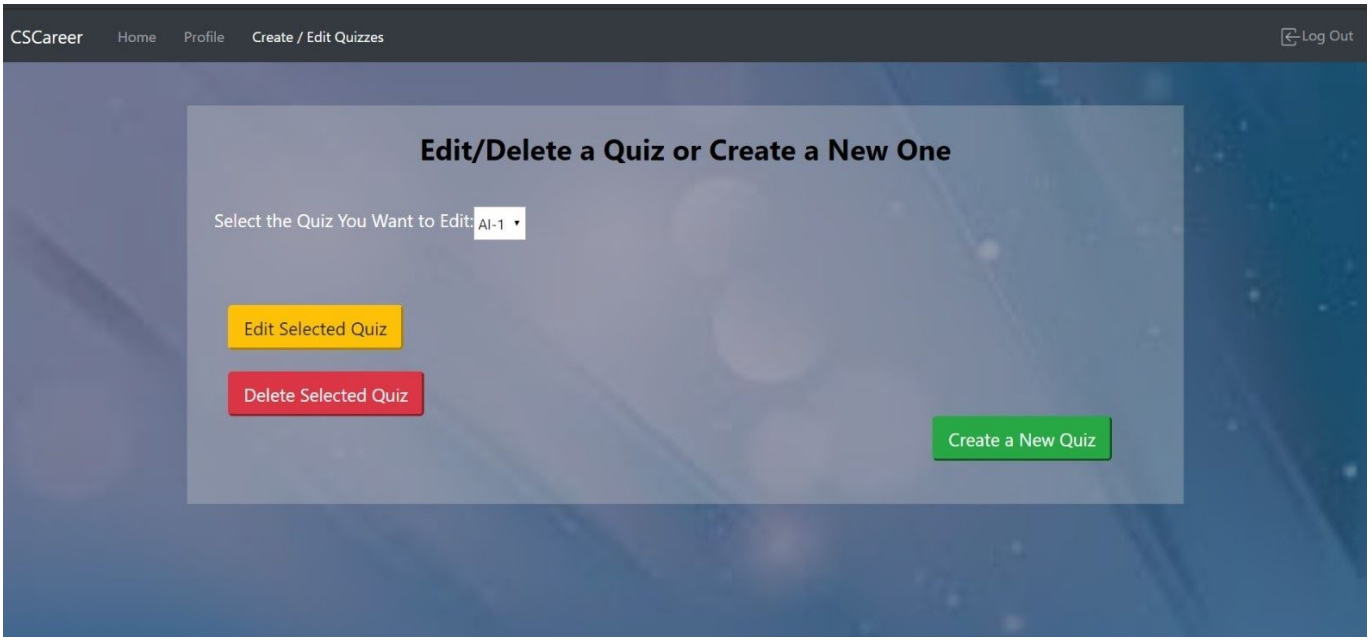


Figure 16: Create / Edit Quizzes Page

a) Edit a Selected Quiz

The admin can select a quiz from the dropdown menu and click the “Edit the Selected Quiz” button. Then, the admin will go to a page where all the quiz’s properties and questions can be seen and changed by clicking the relevant buttons.

a) In this page, the admin will see the selected quiz’s properties such as its title, category name, maximum time that the developers can solve the quiz and its number of questions. She/he can change these properties by clicking the “Edit Quiz Properties” button. This button will lead to a page in which the admin can see all the current properties of the quiz. She/he can change these properties and by clicking the “Save Changes” button, the admin will be able to save the new quiz properties.

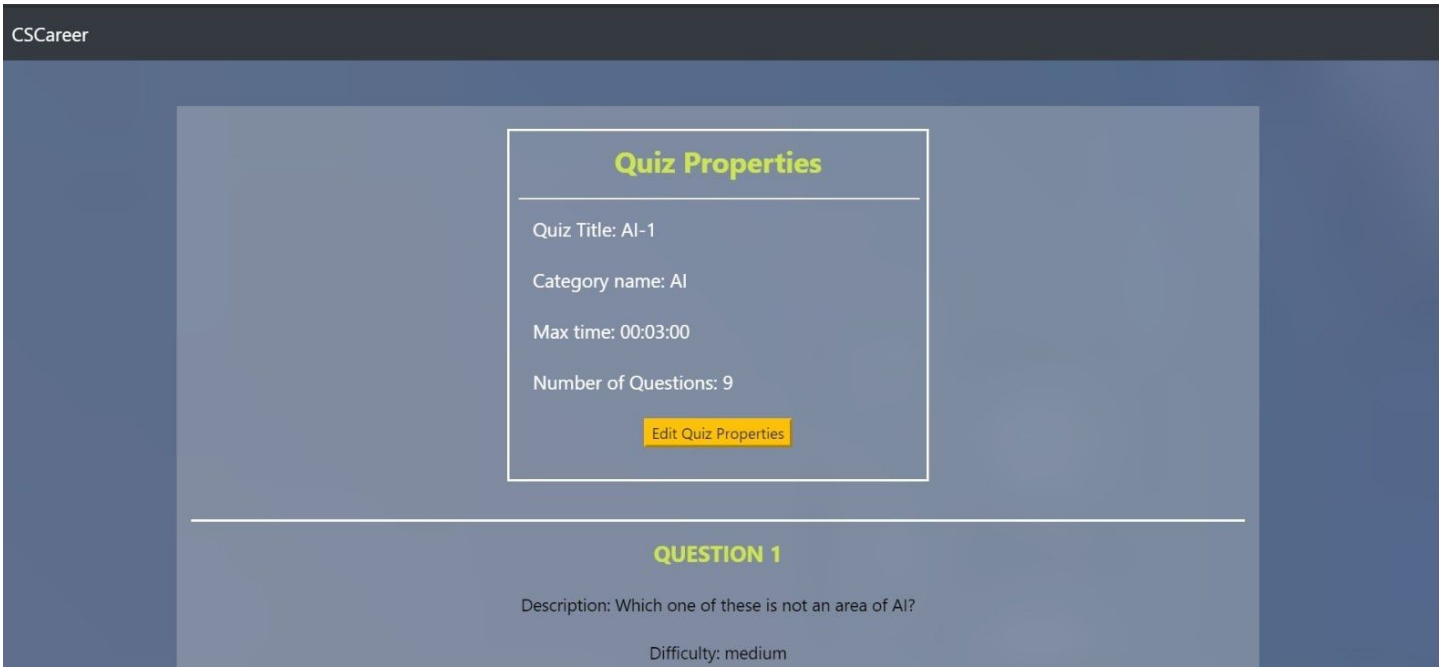


Figure 17: Quiz Properties are Displayed

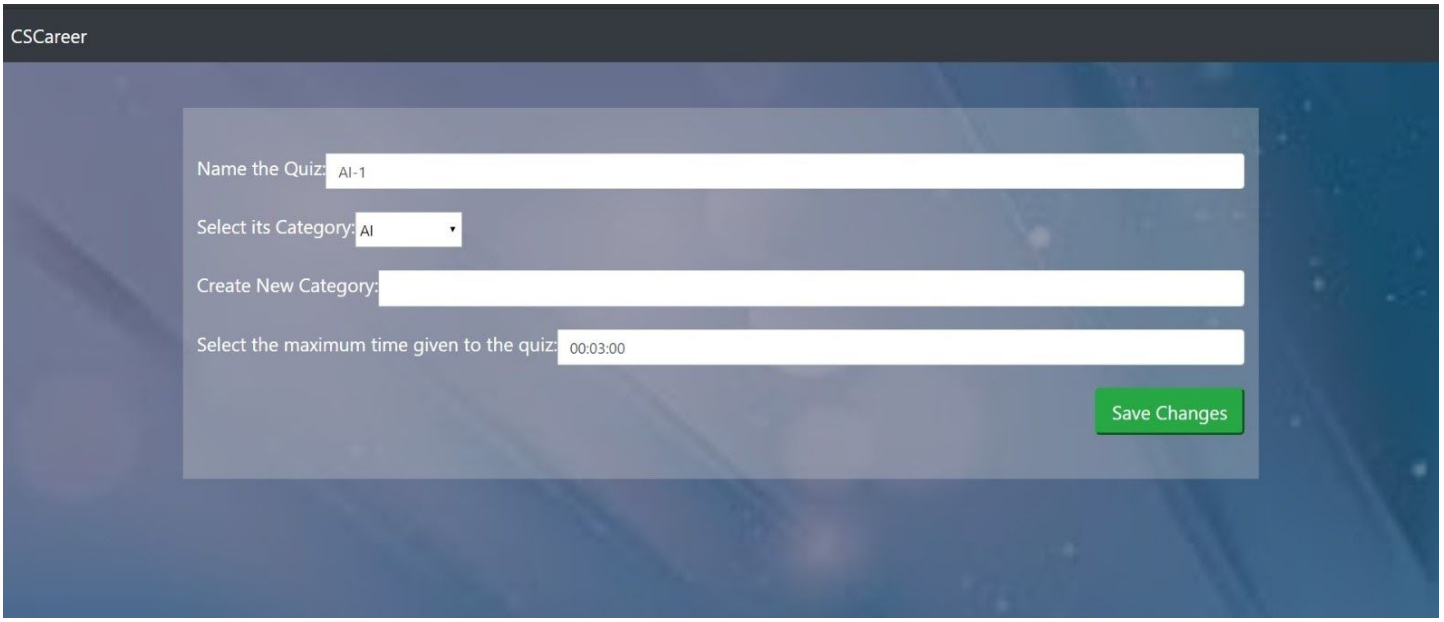


Figure 18: Editing the Quiz Properties

b) The admin can also edit a selected quiz's question by clicking the relevant question's "Edit Question" button. If she/he clicks this, she/he will lead to a page where all its current properties such as its description, choices, correct answer, difficulty and subjects are selected. If she/he wants to change some of them or all, the changes she/he made will be saved by clicking the "Save Changes" button.

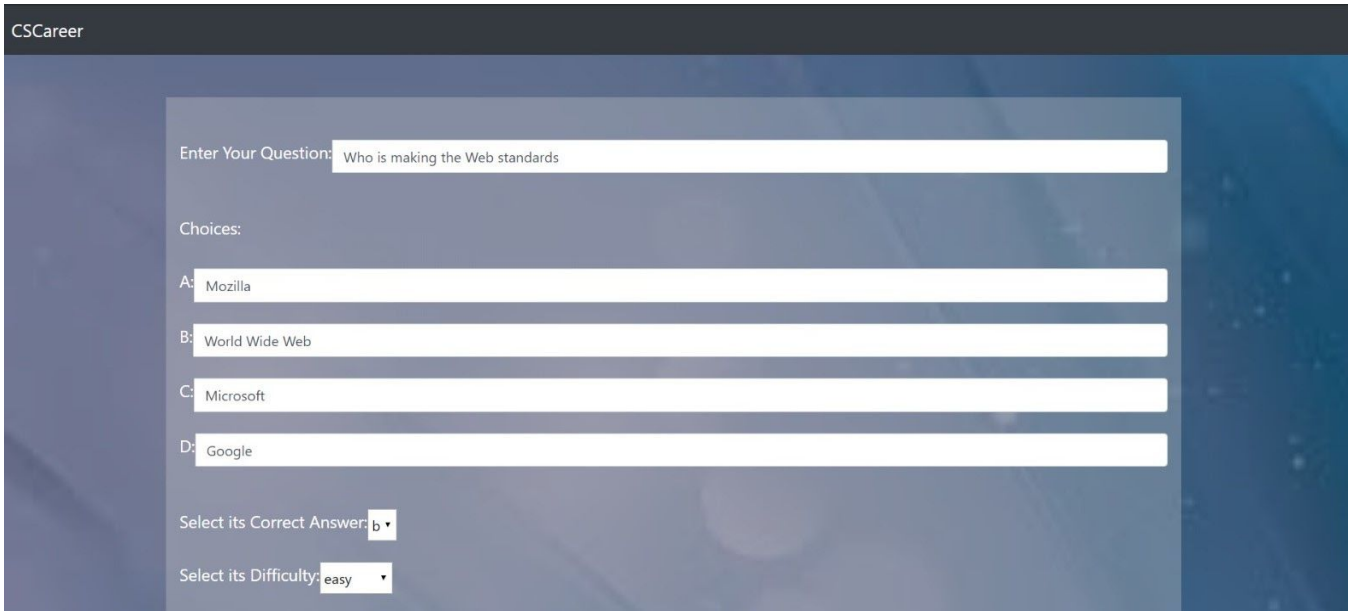


Figure 19: Edit Question

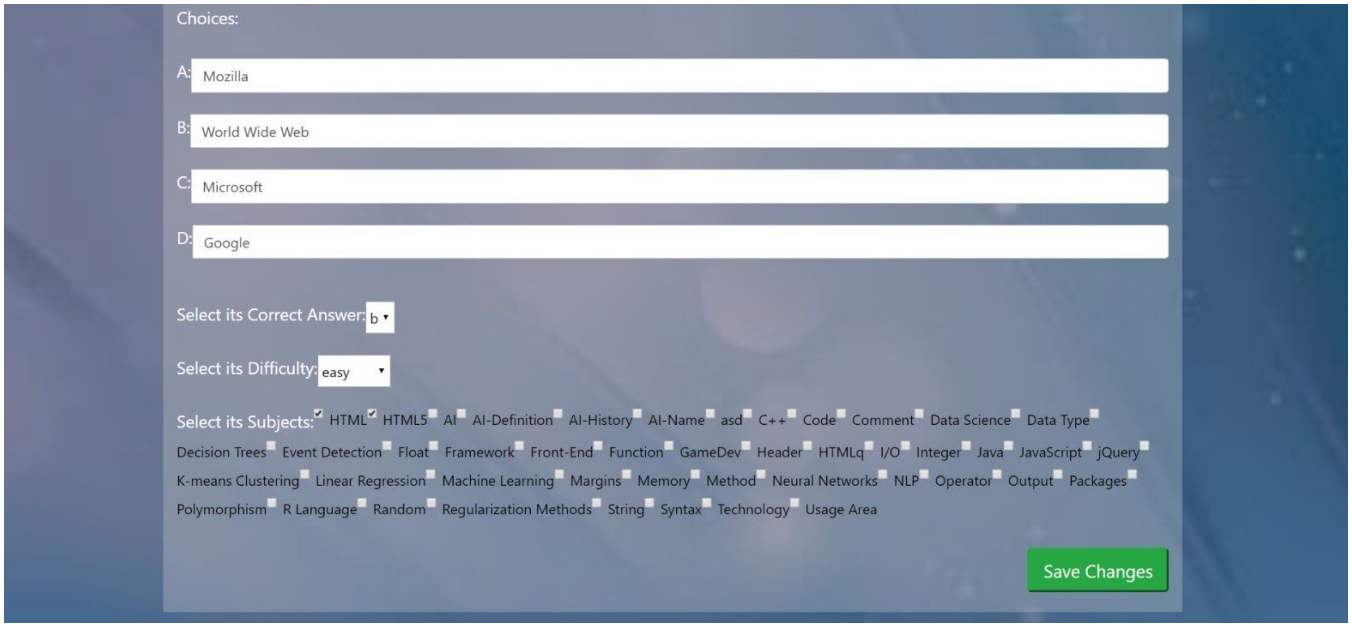


Figure 20: "Save Changes" Button

c) The admin can remove a selected quiz's questions by clicking the "Remove Question" of the relevant question. Then, the page will refresh and the admin will no longer see this question in this page. All the questions' numbers that come after the deleted question will be decreased by one. For example, if she/he deletes question 5, question 6 will now be seen as question 5, question 7 will be seen as question 6 and so on.

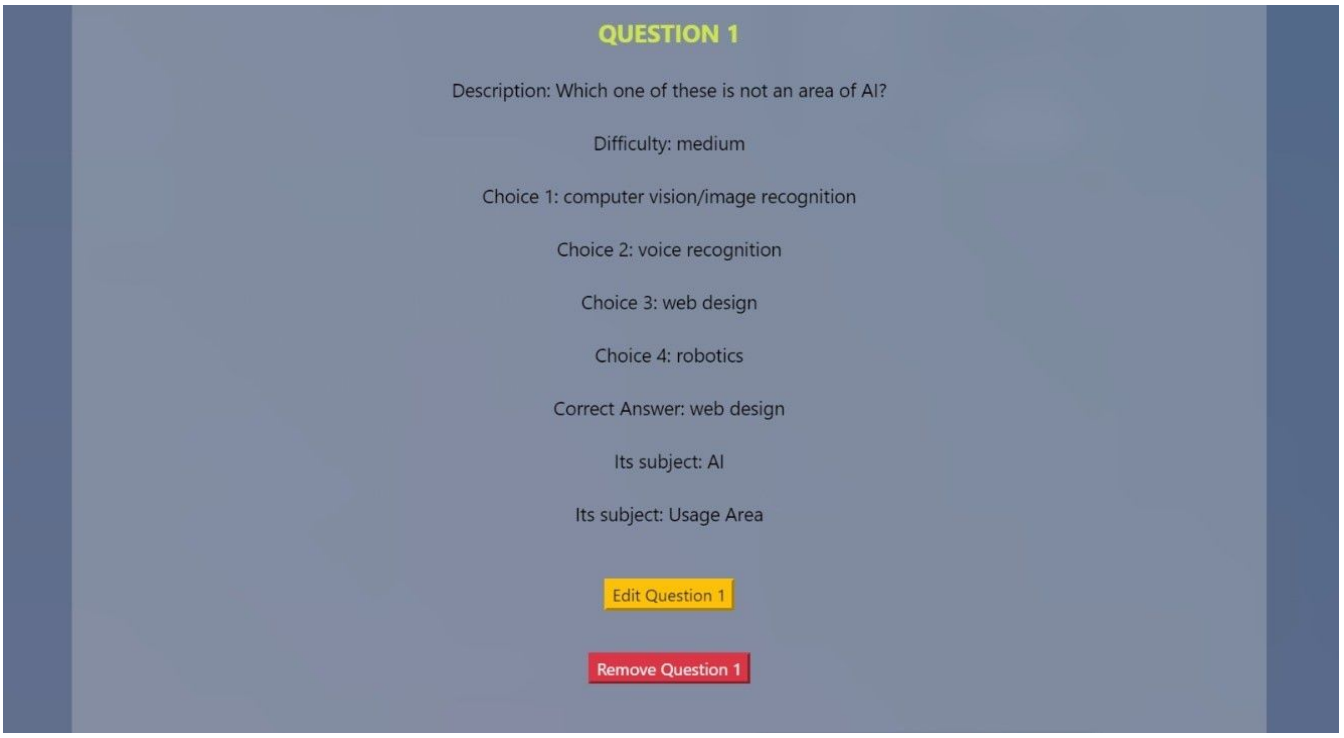


Figure 21: Remove Question Button

d) The admin can also add a new question to this selected quiz by clicking the "Add New Question" at the end of the page. Then, she/he will go to a page where she/he can specify all its properties such as its description, choices, correct answer and subjects. If she/he clicks the "Add More Questions" button, she/he will again add a new question. She/he can click the "Done" button whenever she/he does not want to add more (Figure 26).

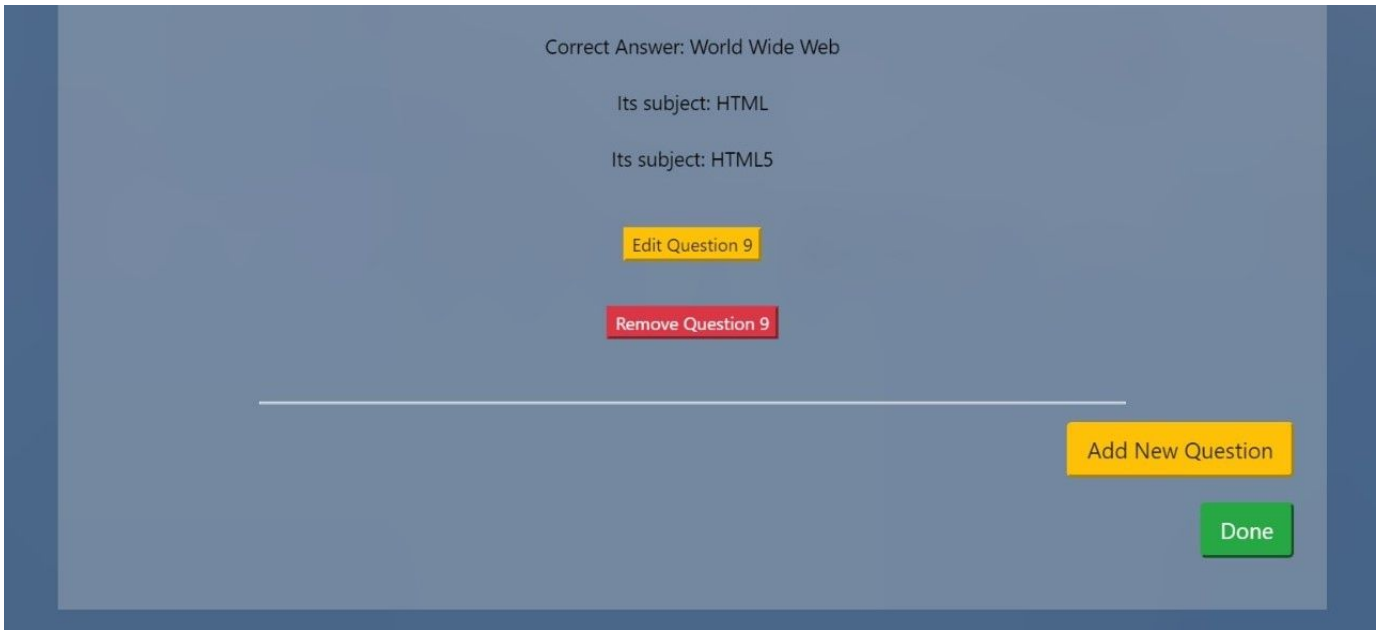


Figure 22: Add New Question Button

b) Delete a Selected Quiz

The admin can select a quiz from the dropdown menu where all the quiz titles are listed. Then, by clicking the “Delete Selected Quiz” button, she/he can delete it from the database. Then, there will be a message indicating that “The Selected Quiz is Deleted!”. If the admin clicks “OK”, she/he will see that the quiz is deleted and not listed in the dropdown menu.

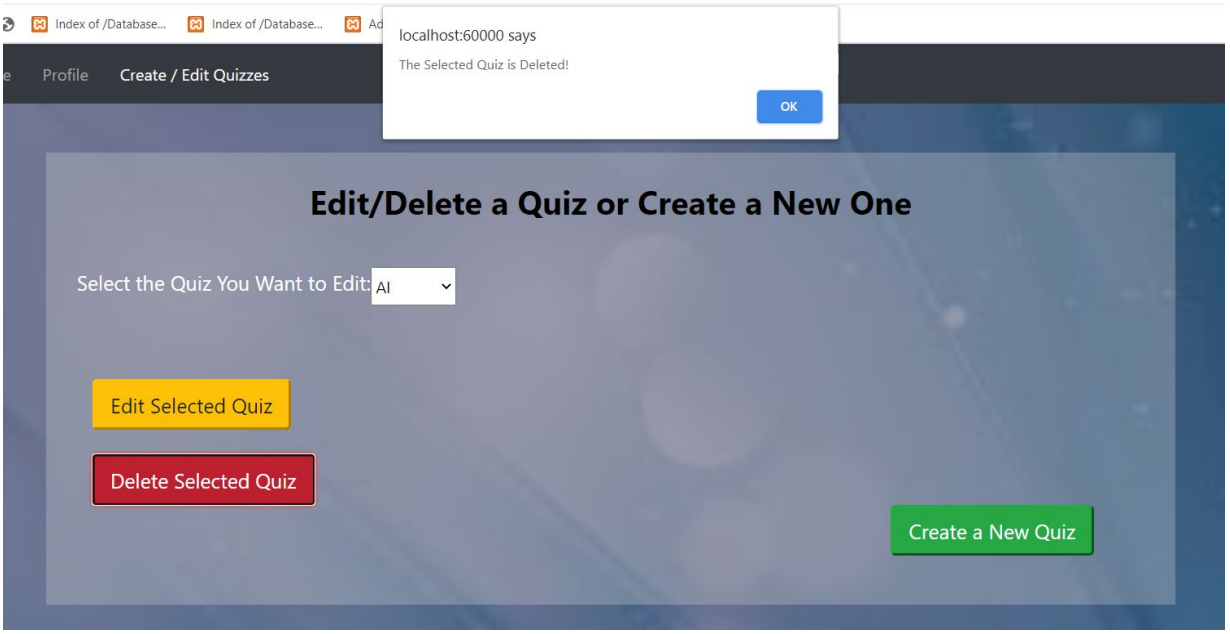


Figure 23: “The Selected Quiz is Deleted!” Message

c) Create a New Quiz

The admin can create a quiz by clicking “Create a New Quiz” button. This will lead the admin to a page where she/he can specify the quiz’s properties such as its title, category and the maximum time given to the quiz. The admin can select from the categories that are in the database or create a new category. When she/he clicks the “Next” button, she/he will go to a page where she can add a question. After specifying the question’s description, choices, difficulty level and its subjects, she/he can continue to add more questions by clicking “Add More Questions” or finish preparing the quiz by clicking the “Done” button. After clicking the “Done” button, the admin will go to a page where again she/he will see all the quiz’s properties and its questions. She/he can change the quiz properties by clicking the “Edit Quiz Properties” button. She/he can change the question's properties by clicking the relevant question’s “Edit Question” button. or remove a question by clicking its “Remove Question” button. To add more questions to this quiz, she/he can click the “Add New Question” button, or click “Done” to finish preparing.

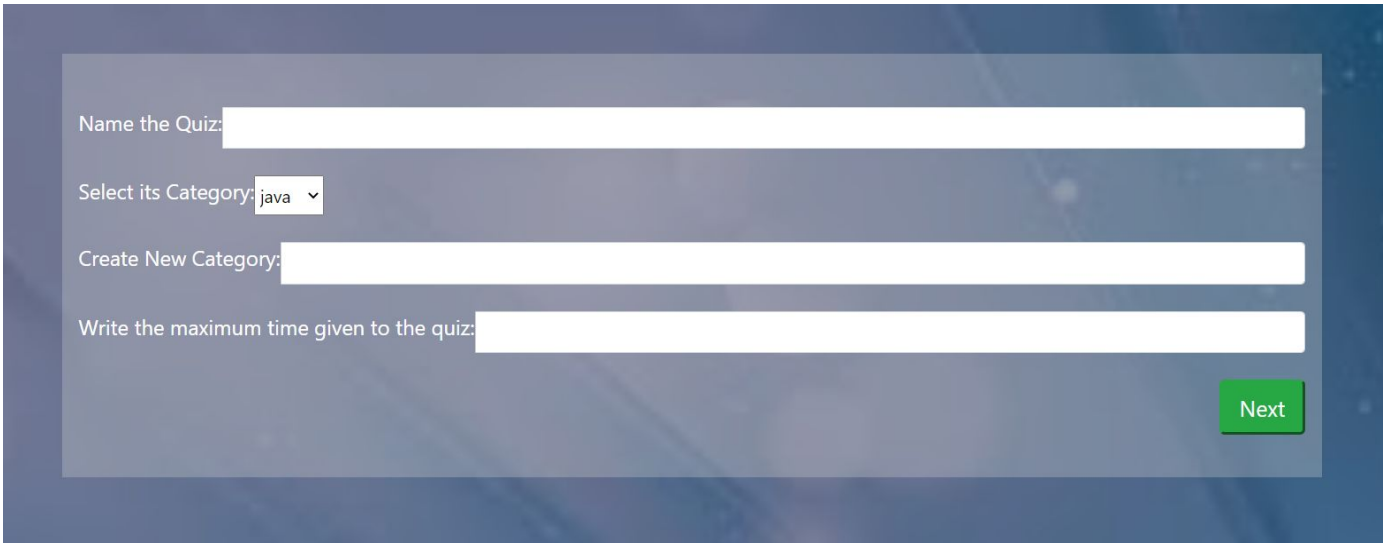


Figure 24: The Page after the Admin clicks “Create New Quiz” Button

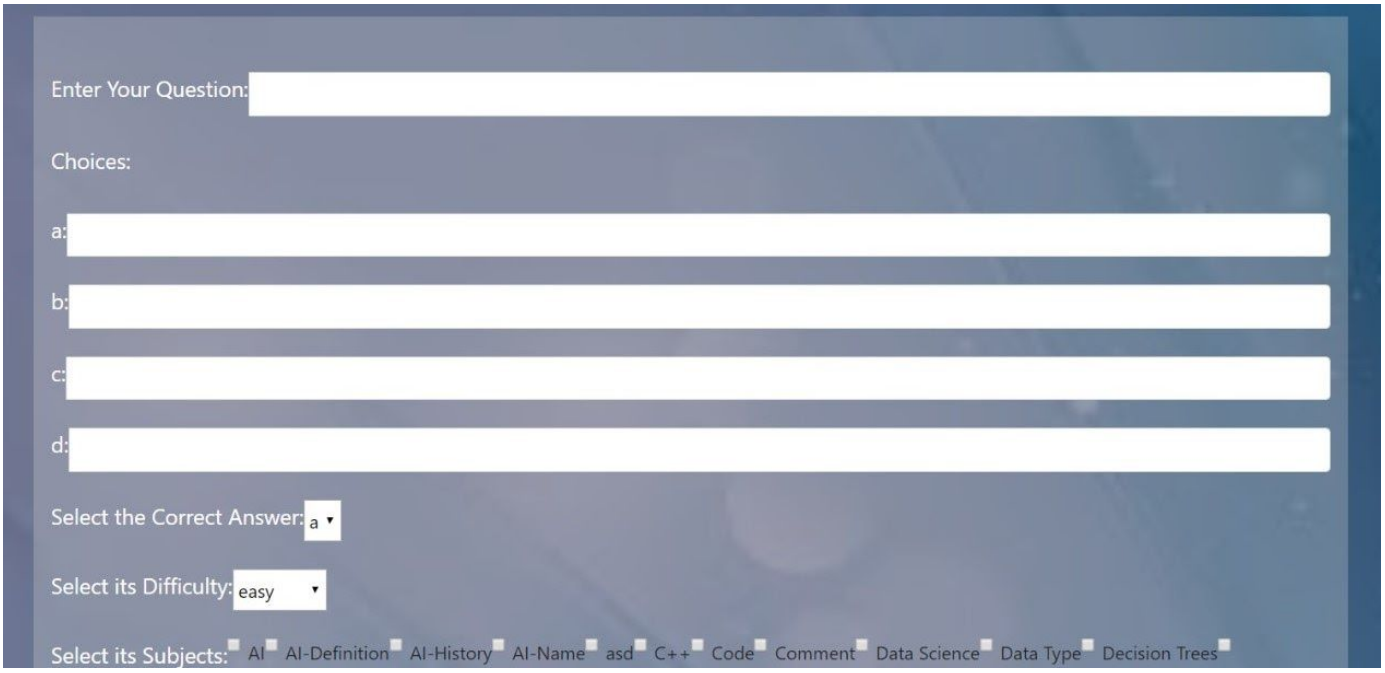


Figure 25: Adding Question to the New Quiz

Figure 26: “Add More Question” and “Done” Buttons

7. Website

Our website for further project information:

https://hamzapehlivan.github.io/Database-Project/

Source code is available in here

https://github.com/hamzapehlivan/Database-Project