**COS60010 - Technology Inquiry Project**

**Semester 1 - 2024**

**Deliverable 3**

**Project Delivery Documentation**

**Group Name:** GROUP 1

**Student members:** Arun Ragavendhar Arunachalam Palaniyappan 104837257

Christopher Opie 105380202

Layan Buddhimal Weerasingha Arachchige 104758921

Amirajsinh Pradhyumansinh Sonagara 104801333

Henil Mukeshbhai Pistolwala 105065800

**Workshop:** Thursday – 14:30-16:30 - Room EN207

**Facilitator:** Dr. Alfandi Yahya

Contents

[1. Introduction 4](#_Toc166408666)

[2.Project Specification Documentation for Business Usage 4](#_Toc166408667)

[2.1 Project Definition: 4](#_Toc166408668)

[2.2 Project Objectives: 5](#_Toc166408669)

[2.3 Intended Users 5](#_Toc166408670)

[2.4 Research Report 5](#_Toc166408671)

[3. User Stories, Implemented Functionalities and Priorities 5](#_Toc166408672)

[3.1 Main functionality of the quiz (displaying questions and collecting answer inputs) for chemical structure-based and reaction based chemistry questions 5](#_Toc166408673)

[3.2 Displaying a list of correct and incorrect answers at the end of the quiz 6](#_Toc166408674)

[3.3 Incorporating username/password-based login & authentication for students 6](#_Toc166408675)

[3.4 Recording and retrieving player statistics and scores via the database 6](#_Toc166408676)

[3.5 Project deployment to the web 6](#_Toc166408677)

[3.6 Admin access provided to teachers for database modifications 6](#_Toc166408678)

[3.7 Option to Choose Difficulty levels for questions 7](#_Toc166408679)

[3.8 Extension of the application with additional subjects and questions for the future 7](#_Toc166408680)

[3.9 Future scope and plans for Application Scalability 7](#_Toc166408681)

[4. Application Functionality and Usage after development 7](#_Toc166408682)

[4.1 Login Page: 7](#_Toc166408683)

[4.2 Welcome Page: 8](#_Toc166408684)

[4.3 Questions Page: 9](#_Toc166408685)

[4.4 Results Page: 10](#_Toc166408686)

[4.5 Admin Page: 11](#_Toc166408687)

[5.Software Requirements and Design Specifications for Technology Usage 12](#_Toc166408688)

[5.1 User-facing pages and their functionalities 12](#_Toc166408689)

[5.1.1 Login Page 12](#_Toc166408690)

[5.1.2 Welcome Page 13](#_Toc166408691)

[5.1.3 Questions Pages 13](#_Toc166408692)

[5.1.4 Results Page 14](#_Toc166408693)

[5.1.5 Admin Page 14](#_Toc166408694)

[5.2 Program Execution Flow Diagrams 15](#_Toc166408695)

[5.2.1 Program execution flow diagram: Student 15](#_Toc166408696)

[5.2.2 Program execution flow diagram: Admin 16](#_Toc166408697)

[5.3 Databases 17](#_Toc166408698)

[5.3.1 Users Table 17](#_Toc166408699)

[5.3.2 Questions Tables 17](#_Toc166408700)

[5.3.3 Scores Table 18](#_Toc166408701)

[6. Appendix 19](#_Toc166408702)

[6.1User Manual: Step by step guide to use the application: 19](#_Toc166408703)

[6.1.2 Student User: 19](#_Toc166408704)

[6.1.3 Admin User: 25](#_Toc166408705)

[6.2 Abbreviations 32](#_Toc166408706)

[6.2 List of figures and tables: 32](#_Toc166408707)

[7. References 32](#_Toc166408708)

# 1. Introduction

The main goal of this document is to discuss in detail about 3 main parts:

**Project Specification Documentation for Business Usage:**

This section provides an overview of the project definition, project objective, the intended target users of the project as well as the features implemented in the project from a business and usability point of view.

**Software Requirements and Design Specifications for Technology Usage:**

This section provides a detailed explanation of the features implemented from a programming point of view with the technical details, code base, design architecture and the database schema specifications.

**User Manual:**

This section provides a step by step guiding on how to navigate, use and benefit from the application.

The ultimate goal of the document is to discuss in depth about “what” has been built, “why” it has been built, “how” it has been built, from a business point of view as well as a technical programming point of view.

# 2.Project Specification Documentation for Business Usage

## 2.1 Project Definition:

Instatute Pvt.Ltd is a well-known tutoring company present at various locations throughout Australia. Currently, they offer one-on-one and group tutoring services to help high schools students prepare for university entrance exams for both Australian as well as international institutions.

Instatute is currently exploring gamification as a way to enhance and improve engagement and community in their teaching. But, as a result of their internal company policies and legal constraints, they can only use software tools that are specifically built for them.

Hence, after conducting a detailed technical business analysis and user requirements study with the client Instatute Learning Pty Ltd., this interactive Chem Quiz application has been designed and developed to support the preparation of their students for university entrance exams for both Australian and international institutions. The Chemistry quiz application is web based and has been developed to make the learning easier and more engaging for students struggling with challenging chemistry concepts.

## 2.2 Project Objectives:

The quiz has been implemented as a support application which is available within the main website of Instatute Learning Pty Ltd. The core idea is that every enrolled student at Instatute Learning Pty Ltd will have access to this application. A student user can login using his/her credentials and can play the game an unlimited number of times. They can modify the difficulty level and view their historical results, as well as the results of their peers through a leaderboard, to effectively evaluate where they stand.

The application also provides an admin login for a staff, allowing them to create and modify questions along with the access to modify student records.

## 2.3 Intended Users

The target audience of this application are high school students who are currently studying chemistry. The app at its core, is an interactive chemistry quiz where a student can view a set of chemistry related questions and answer it. Once finished, he/she can then instantly see their score, thereby offering instantaneous feedback. The questions are both structure based and reactions based, so the student would have to identify structures, solve chemical reaction equations as well as draw chemical structures to answer the questions. This helps to enhance the analytical and cognitive abilities of the student and helps in enhancing his/her active recollection and memory.

## 2.4 Research Report

Once the user requirements were obtained from Instatute, the group members set out to do extensive research in order to do the below activities:

* Convert the business requirements into user stories.
* Analyse the user stories and create a system design that can be implemented into a working project.
* Once the system design is locked in, start researching for technology stacks to program and develop the project.
* After a thorough research and reflection, the application has been designed and developed using HTML, CSS, JAVASCRIPT, PHP AND MYSQL. Further details of which has already been described in detail in the project concept report.
* The references used have been duly mentioned and credited in the document.

# 3. User Stories, Implemented Functionalities and Priorities

Based on the requirements purpose discussed in the Introduction, the following user stories have been designed and their respective priorities designated.

## 3.1 Main functionality of the quiz (displaying questions and collecting answer inputs) for chemical structure-based and reaction-based chemistry questions

Priority: 1 (Highest)

* Students are given a series of chemistry-based quiz questions displayed to them, and they can submit their answers to the application.
* The focus is on chemical structure-related and reactions related questions because it is fundamental to many branches of chemistry and is a challenging topic for students.
* This implementation helps to distinguish the program from existing chemistry-based quizzes.

## 3.2 Displaying a list of correct and incorrect answers at the end of the quiz

Priority: 2

* For an effective self-assessment, players are given feedback at the end of a game which displays the questions asked, their responses, and the correct answers.
* Students are then encouraged to revisit their textbooks or notes, to clear up any misunderstandings they may have.

## 3.3 Incorporating username/password-based login & authentication for students

Priority: 2

* Students can access the “Chem Quiz” lobby by entering their username and password which has been provided to them by the Instatute Learning Pty Ltd.

## 3.4 Recording and retrieving player statistics and scores via the database

Priority: 2

* Student scores are saved into the database after they attempt a quiz.
* Statistics of their previous attempts are provided to them after they log into the application, such as their highest score, scores of the last five attempts, as well as a Leader Board where they can see peer performances and evaluate where they stand.
* At the end of each attempt their score is immediately updated with the latest attempt, and the updated statistics displayed when they return back to the welcome page.

## 3.5 Project deployment to the web

Priority: 3

* The application has been deployed to the web as soon as Priority 1 and Priority 2 features have been implemented.
* Since the development environment is different to the production environment, an early deployment has provided ample time for testing and debugging.
* The project has been debugged and thoroughly tested and is running the latest stable version.

## 3.6 Admin access provided to teachers for database modifications

Priority: 4

* Teachers and staff have access to an administrative area of the application.
* They have access to perform tasks such as the addition, deletion or updating of quiz questions and student scores information.
* The addition and creation of questions feature specifically would allow teachers to provide a more customised experience for their students, resulting in questions being directly relevant to the students' current studies.

## 3.7 Option to Choose Difficulty levels for questions

Priority: 4

* Students can modify the difficulty level of the quiz questions, since this application is meant to be used by at least high school students from a range of year levels.

## 3.8 Extension of the application with additional subjects and questions for the future

Priority: 5 (Lowest)

* Apart from serving it current usage, the quiz application has been designed in such a way that the source code can be worked upon in the future to further add new questions types as well as more subjects apart from chemistry. The scope of this depends upon the client requirements and the response and feedback from the students, but, when the need arises, the application is ready to be upgraded.

## 3.9 Future scope and plans for Application Scalability

Priority: 5 (Lowest)

* The application is currently deployed on a remote server with a public IP address and is accessible to anyone who has access to the internet. Based on the student response and the client demands, the application can be containerized with its code, package dependencies and run time, and can be horizontally scaled and deployed on a set of distributed systems, if the need arises for a larger demand of the application.

# 4. Application Functionality and Usage after development

## 4.1 Login Page:

* Username/password-based login & authentication for Students and Admins
* Students /Admins can access the “Chem Quiz” application by entering their username and password which has been provided to them by the Instatute Learning Pty Ltd.
* Main entrance gate to the application.
* Routes the user to the welcome page, if he /she is a student.
* Routes the user to the admin page if he / she is an admin.

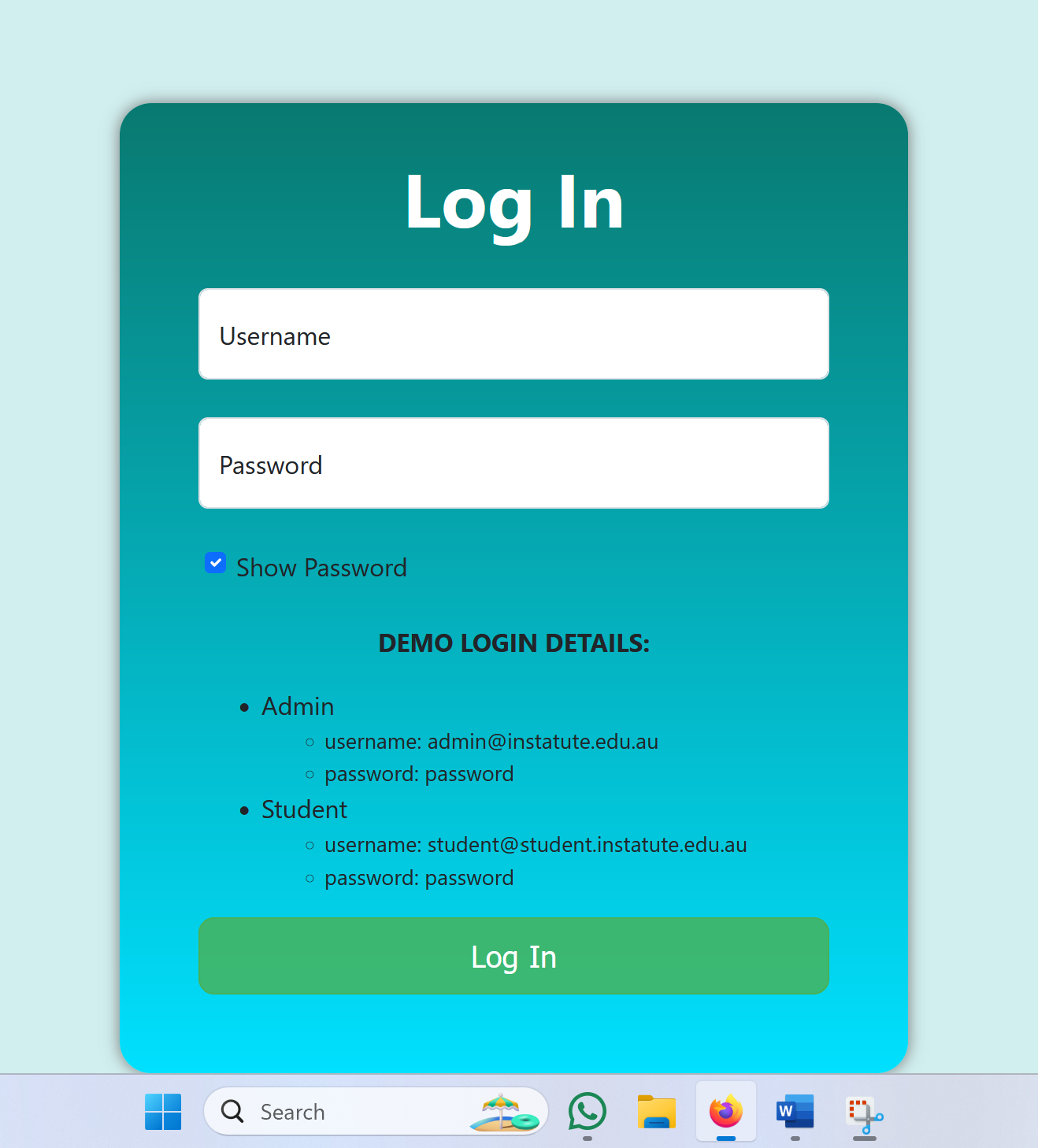


Fig. 1: User Interface of Login Page

## 4.2 Welcome Page:

* Option to start the Quiz, displays the Player statistics and a quiz Leaderboard.
* The logged in student can start the quiz, view his /her statistics and can view a leaderboard.
* The logged in student can view his /her game attempts count.
* Can view his/her highest score attained so far.
* Select a difficulty level for a single quiz attempt.
* It also has a LOGOUT button, with which the user can logout and return back to the login page.

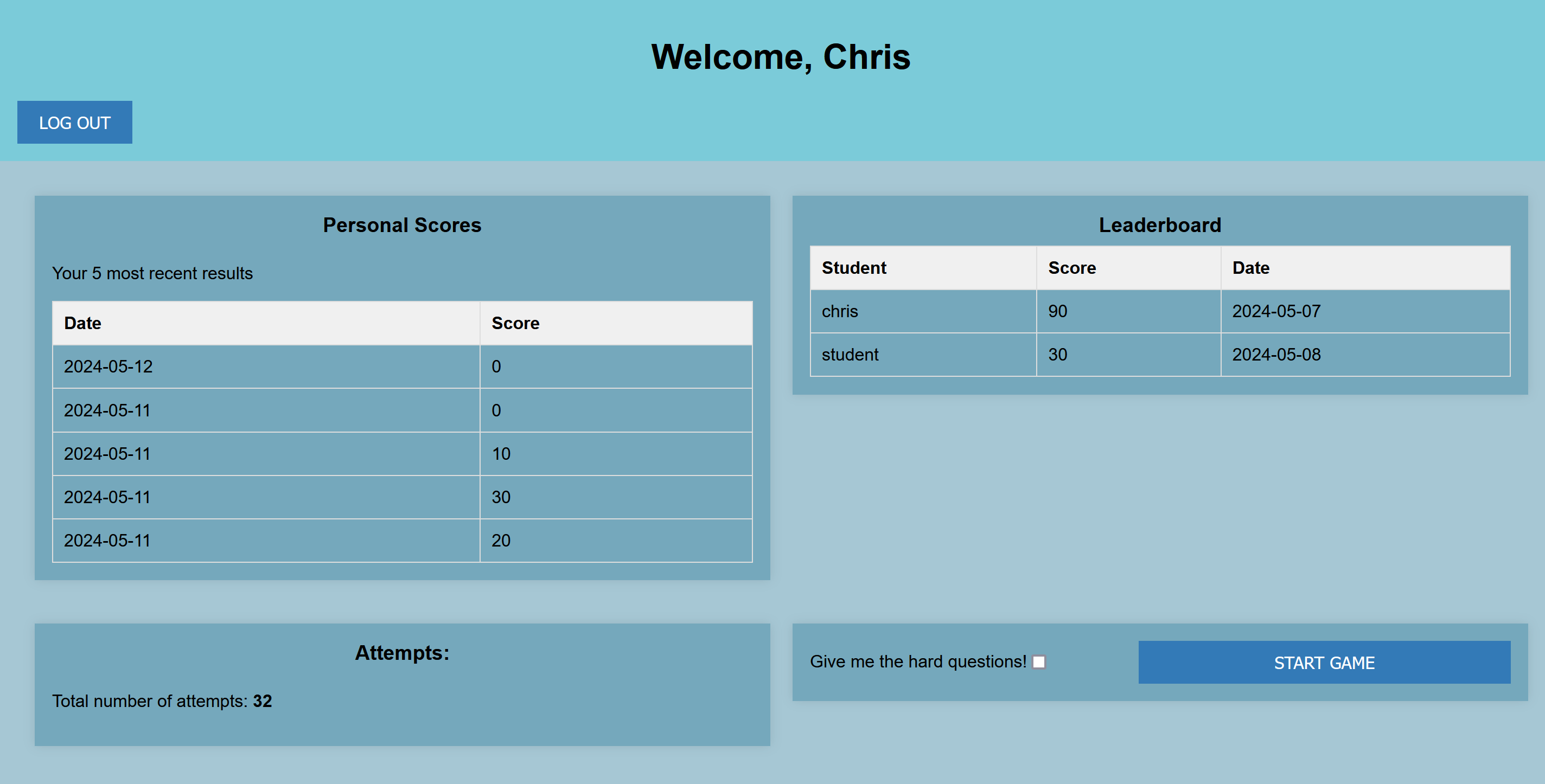


Fig. 2: User Interface of Welcome Page

## 4.3 Questions Page:

* Displays the questions to the user and collecting answer inputs for chemical structure-based and reaction-based chemistry questions.
* The page displays a set 10 questions one by one to the student.
* The question type is either a reaction based one or a chemical structure based one. Either, the student picks an answer out of the 4 options available or is asked to draw the structure to answer the question.
* After answering all the questions, the student can press the Submit button, and he/she is redirected to the results page where they get their results.

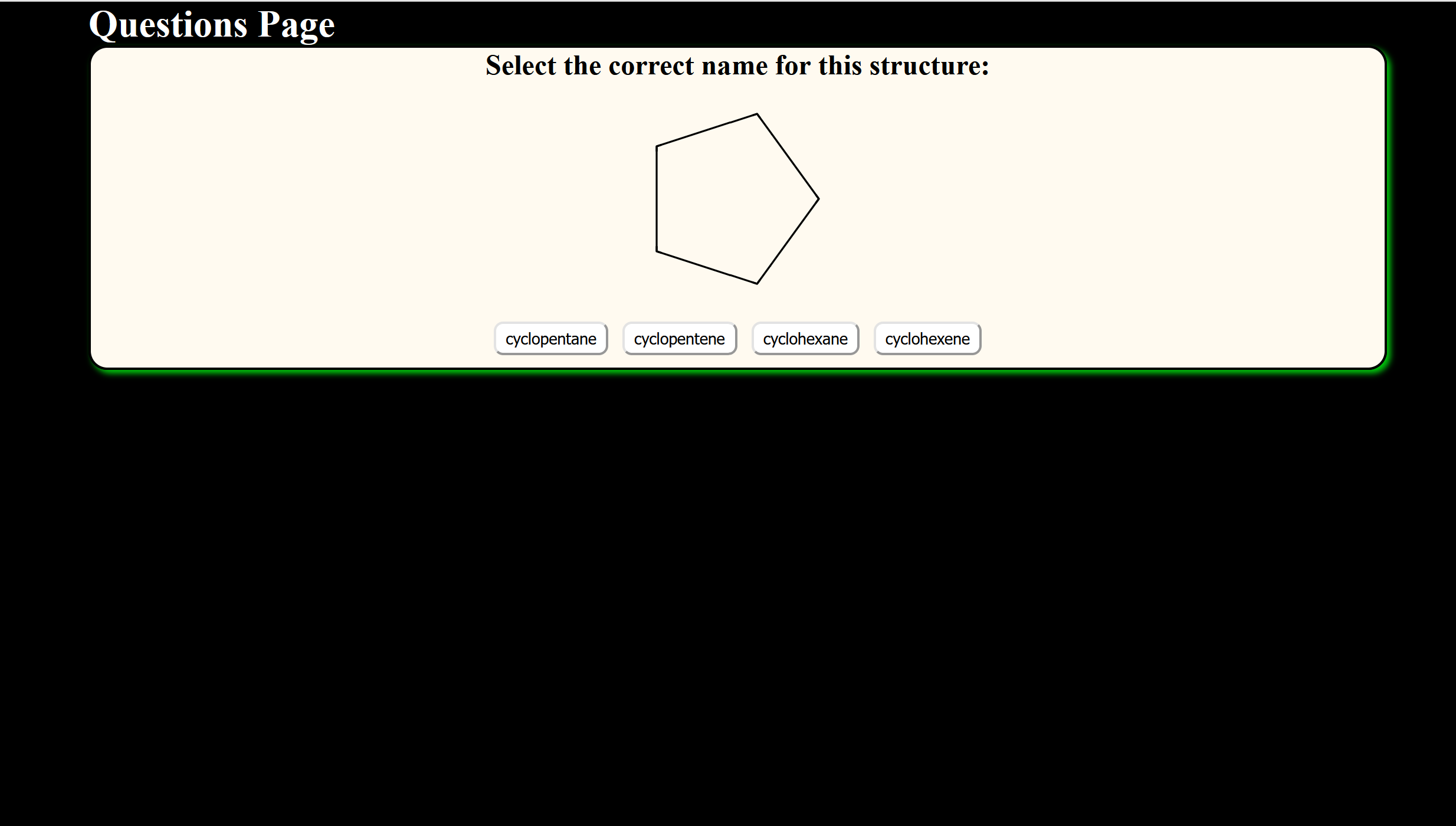


Fig. 3: User Interface of Questions page for a Structure Question

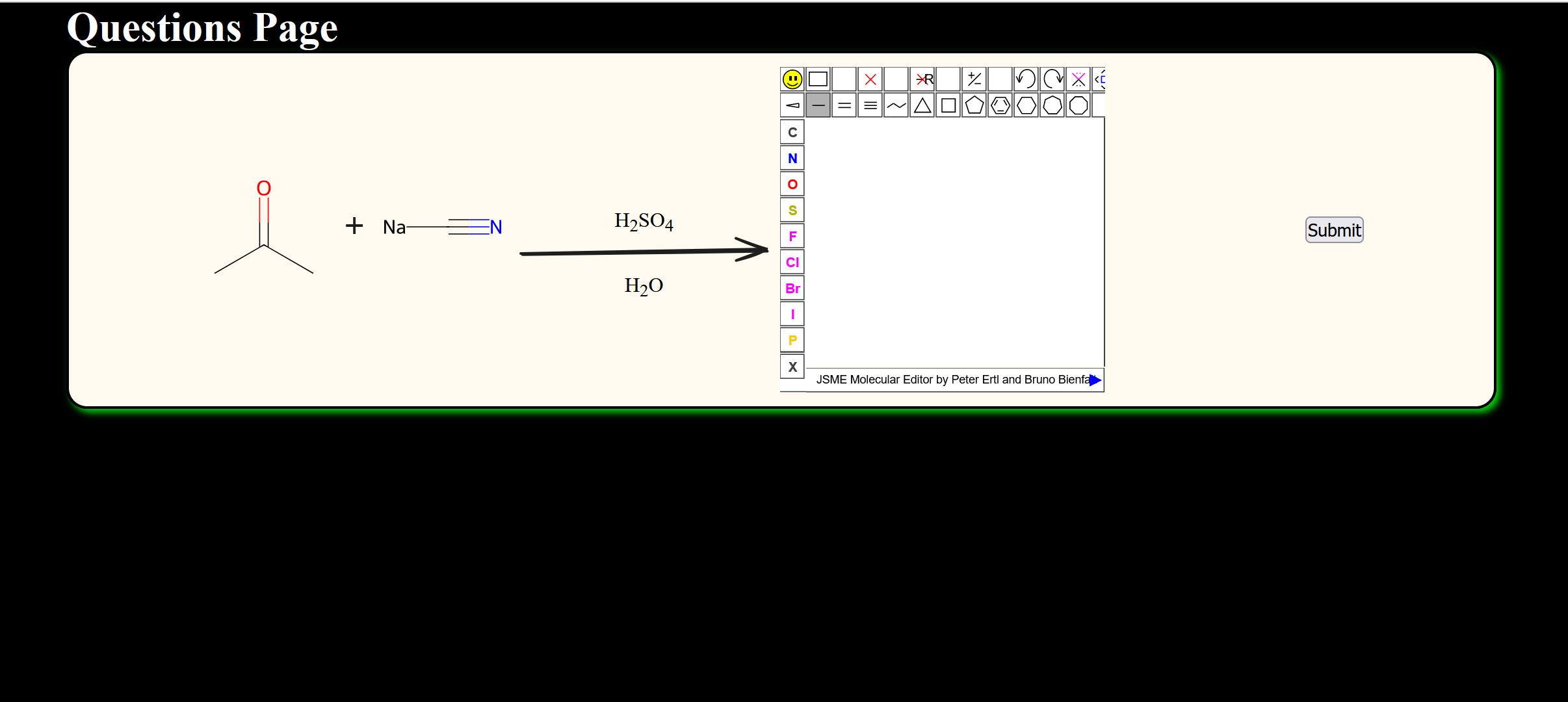


Fig. 4: User Interface of Questions page for a Structure Question

## 4.4 Results Page:

* The page displays the results of the user. The question, the user’s answer and the correct answer are displayed.
* It records, stores and retrieves player statistics and scores via the database, so that they can be displayed when the user returns back to welcome page.
* The user is also shown a custom feedback message based on the score he /she has secured in that attempt.
* The page also has a Return Button. When this button is clicked, the student is navigated back to the welcome page, from where they can either choose to play the quiz again or can logout of the application.

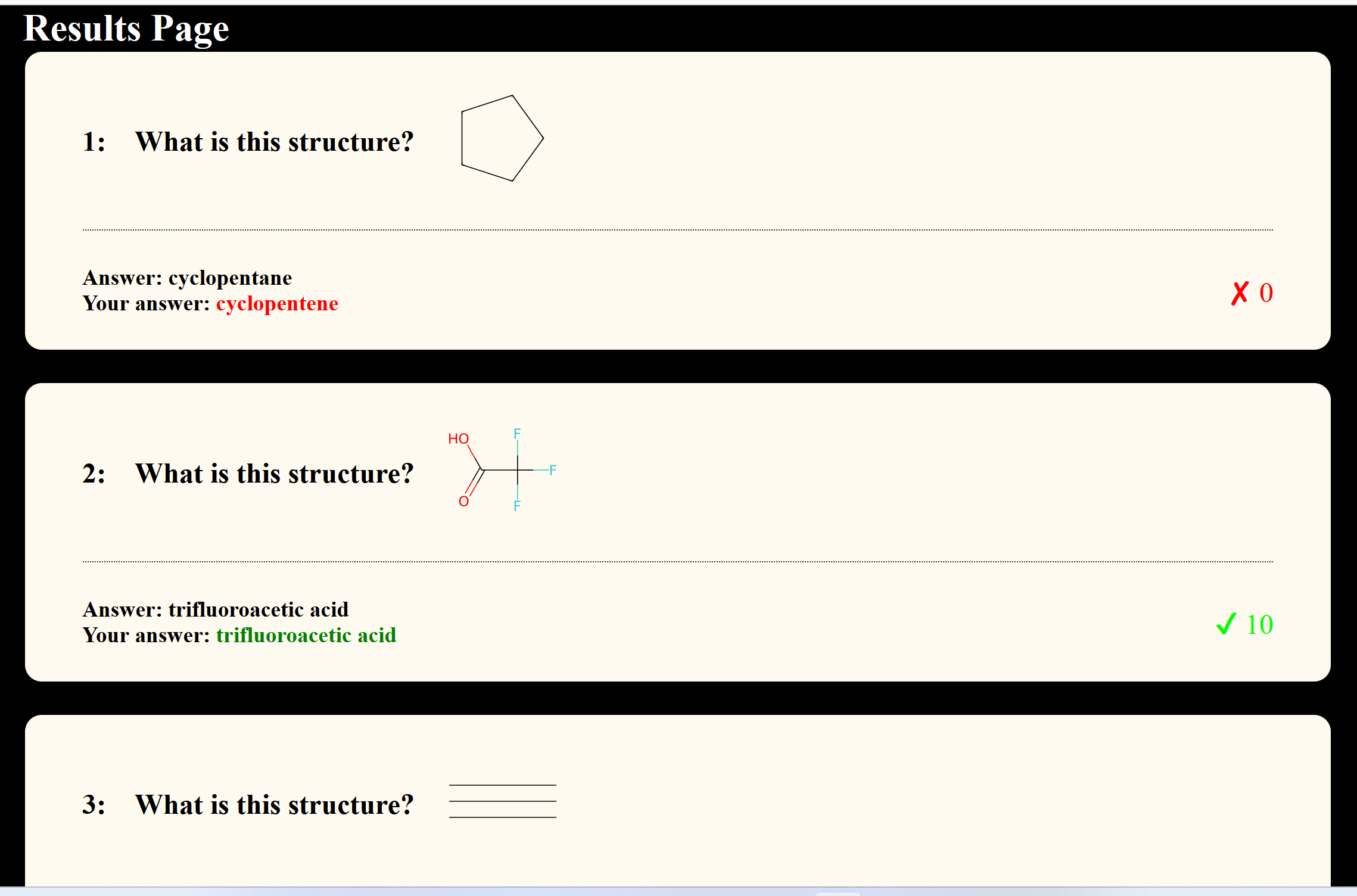


Fig. 5: User Interface of Results page

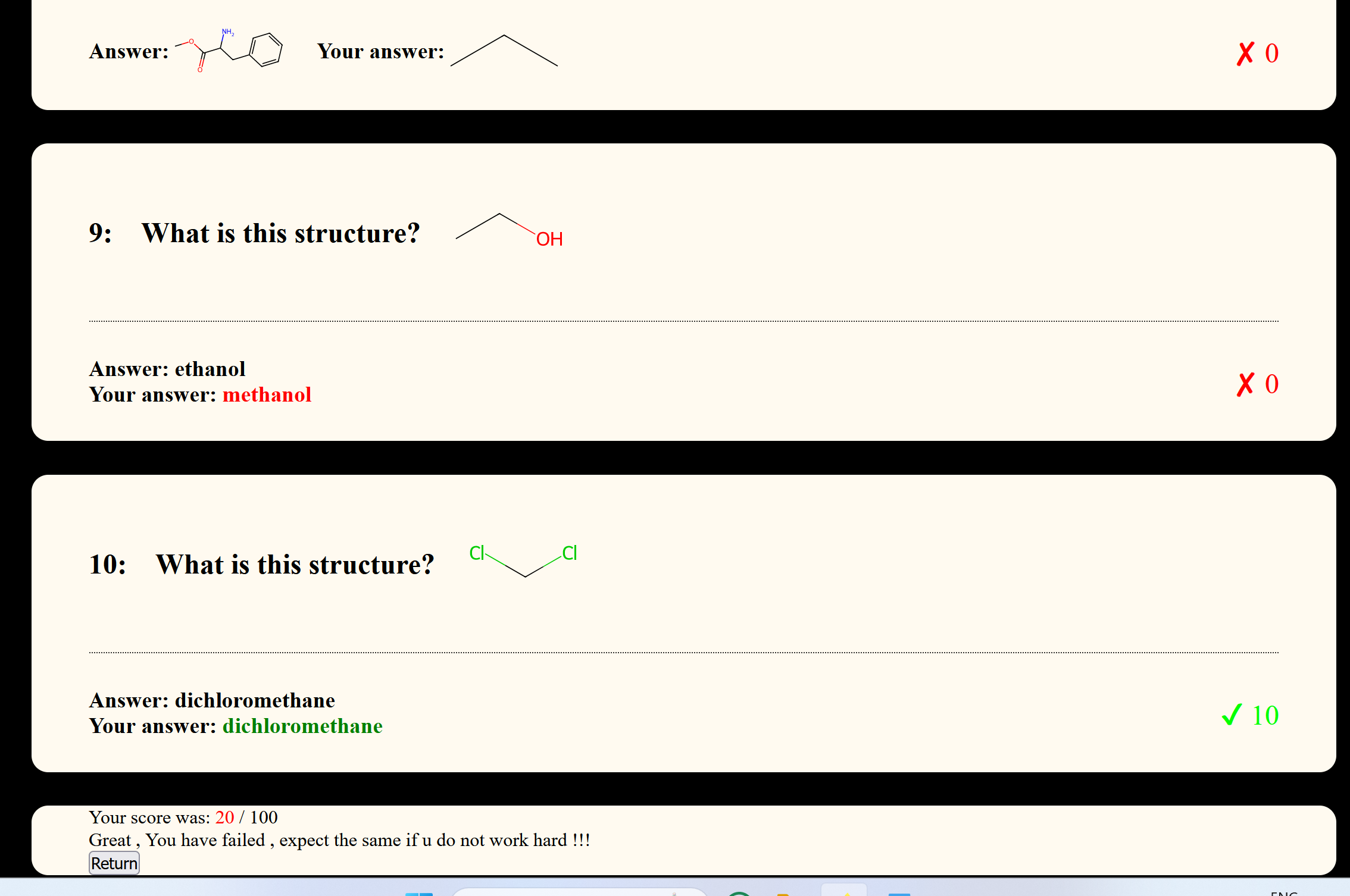


Fig. 6: User Interface of Results page with the scores and ‘Return’ button

## 4.5 Admin Page:

* This page enables an Admin user to create new questions and add them to the question bank.
* The admin can delete records from the database.
* The page has an option for the admin to change the passwords of other admin users.
* The admin can also modify and delete existing questions.
* An admin user is also allowed to create and add a new admin to the database. He/she can also delete as well as modify an admin record from the database.



Fig. 7: User Interface of Admin page with the scores and ‘Return’ button

# 5.Software Requirements and Design Specifications for Technology Usage

## 5.1 User-facing pages and their functionalities

### 5.1.1 Login Page

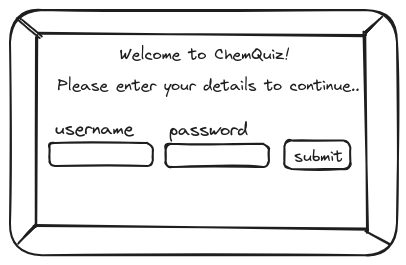
****

Fig. 8: Example user interface sketch for the login page

* Upon clicking the submit button, and after client-side validation, the entered username and password are sent to the server.
* The server sanitises the input, then queries the database using the username, and finally retrieves a hashed version of the user's password. The server hashes the entered password and compares it with the stored version.
* If the two don't match, or if the username was not located on the database, a failure response will be sent to the client, and the user will be prevented from proceeding. They will be prompted to retry. **(Prettyman, 2020).**
* If the credentials match, then the user has been validated. If the user is an admin, they will be directed to the administrative area. However, if the user is a student, the server will retrieve the score results of the student's 5 most recent quiz attempts, and a Leaderboard, via SQL queries. This data is sent to the client as JSON, stored in the session storage and then displayed to the user on the Welcome page.

## 5.1.2 Welcome Page

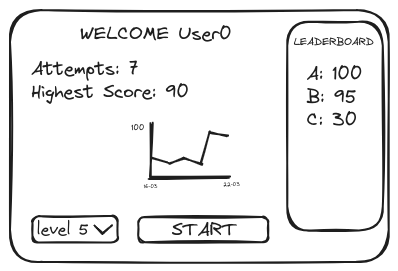
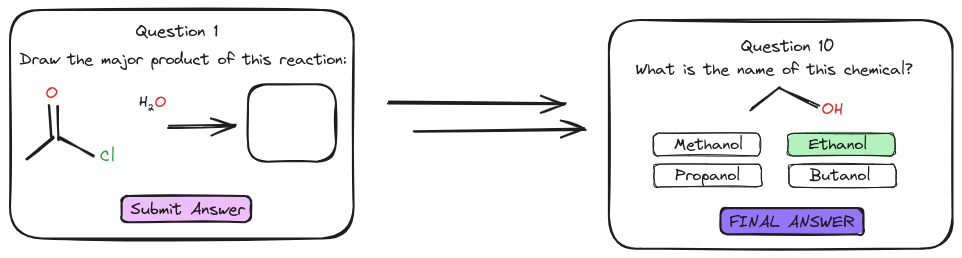
****

Fig. 9: Example user interface sketch for the welcome page

* After clicking the start button, the selected difficulty is sent to the server where it is incorporated in a SQL query to the database, retrieving 10 questions of appropriate difficulty.
* The questions are sent to the client in one batch as JSON for rendering the questions pages, preventing the need for network requests after each response. **(Smith, 2015).**

## 5.1.3 Questions Pages

 Fig. 10: Example user interface sketch for question pages 1 and 10

* Once a question has been answered, it gets stored as an array, and the next question is dynamically rendered to the screen using JavaScript. **(Simpson, 2023).**
* Once an answer to the final question is submitted, all responses in the array are sent to the server, alongside an identifier for the student.
* The server sends a SQL query to the database, retrieving the correct answers to all questions asked. It determines the student's score, and stores on the database using a SQL update query.
* The questions, answers and scores are sent to the client as JSON.

## 5.1.4 Results Page

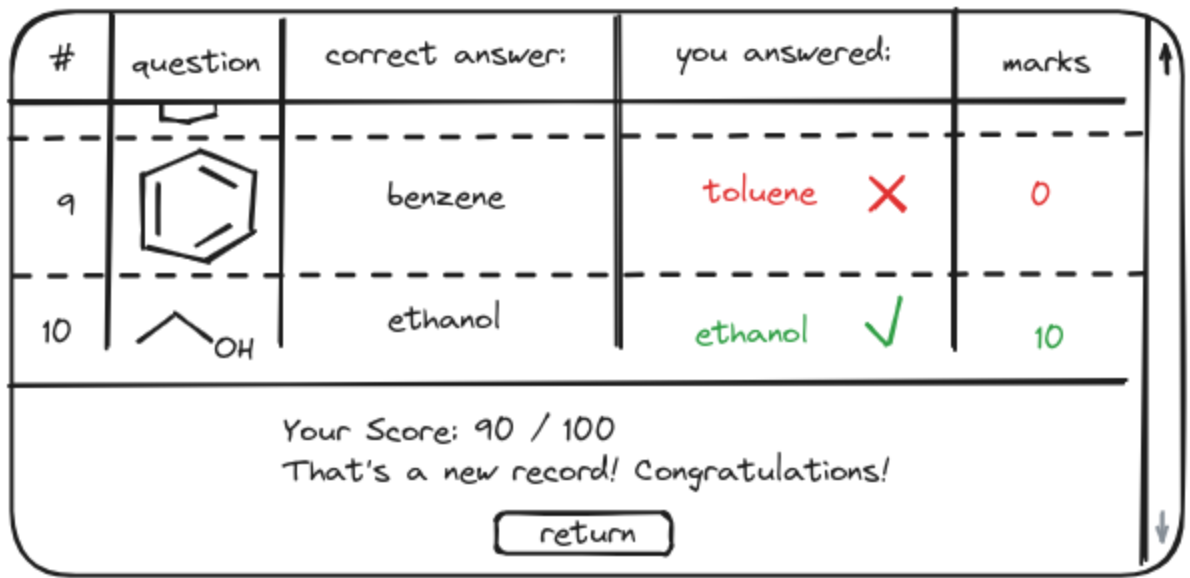
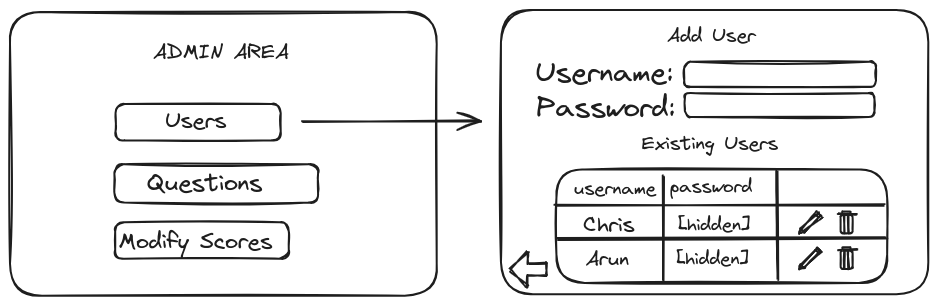


Fig. 11: Example user interface sketch for the results page

* Clicking the return button will cause a re-render of the screen to the welcome page, using the updated scores information that was retrieved earlier.

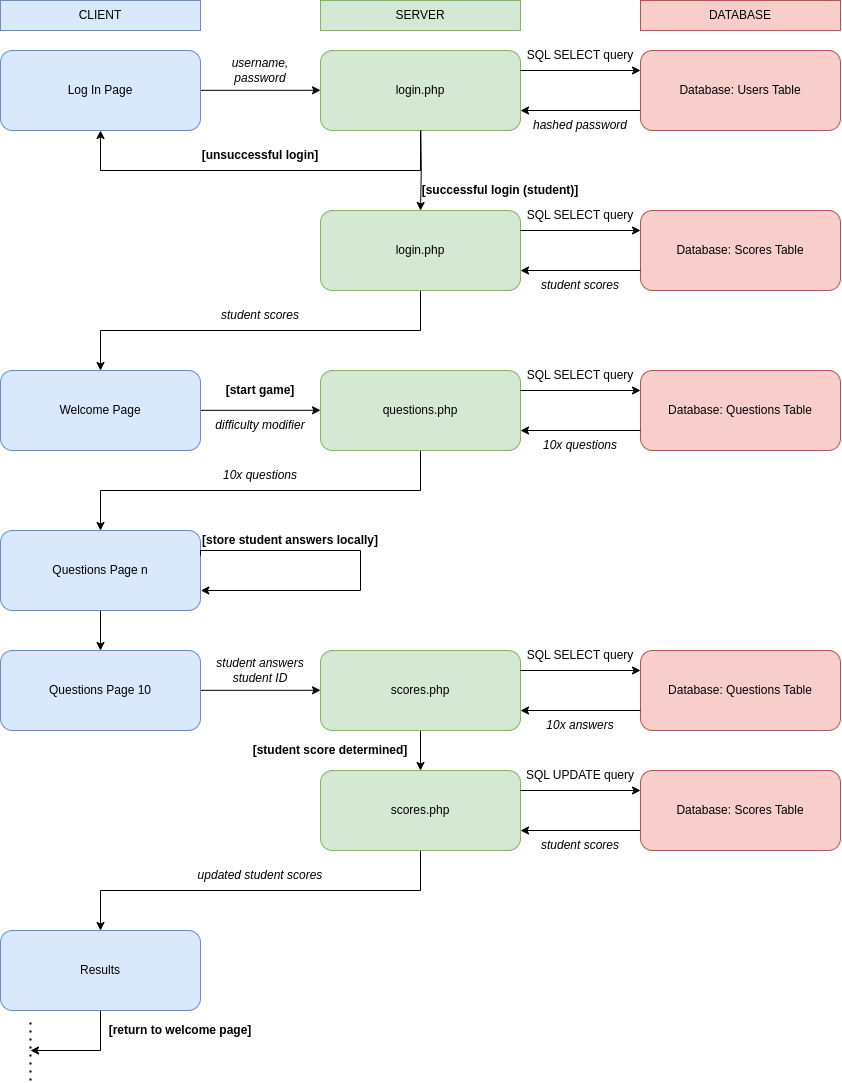
## 5.1.5 Admin Page

 Fig. 12: User interface sketch of results page, with the outcome of clicking the “Users” button

* Clicking a button corresponding to a table in the database triggers the server to retrieve all records of that particular table. These are sent to the client as JSON, the user interface is rendered to the screen consisting of an “add item” section and “existing items” section.
* Filling in the input fields in the “add item” section and attempting to submit the data will cause client-side validation, and if successful, the data will be passed to the server which sanitises it. A SQL update query is performed, and a success (or failure) message in the form of JSON is sent to the client based on the result of the procedure.
* Clicking a button to modify the item will provide an interface similar to the add item section, however the fields will be rendered containing the existing information. The user may make changes and submit the data, in which case a procedure resembling “add item” is performed. **(Gehani, 2011).**
* Clicking the button to delete a record on the database will send the unique identifier for the record to the server, and a SQL delete query will remove it from the appropriate table.

## 5.2 Program Execution Flow Diagrams

### 5.2.1 Program execution flow diagram: Student

Fig. 13: Step-by-step description of the flow of execution of the program for students

### 5.2.2 Program execution flow diagram: Admin

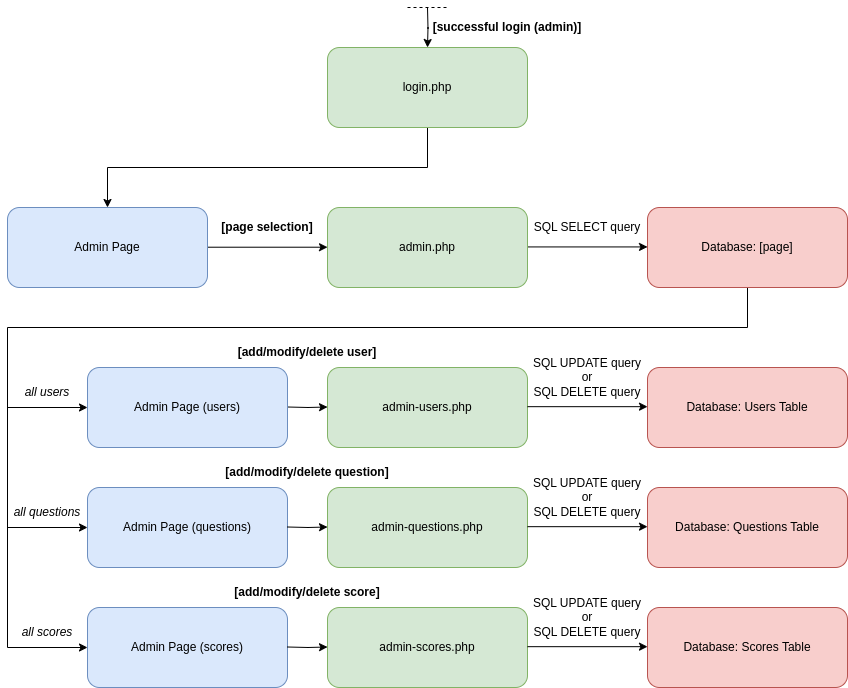


Fig. 14: Step-by-step description of the flow of execution of the program for administrators

## 5.3 Databases

A detailed database system with a list of associated tables is necessary to store, manage, and manipulate all essential data. To aid in this process, the following entity - relationship diagram was created. **(Delisle, 2006).**

A diagram of a product

Description automatically generated

Fig. 15: Entity-relationship diagram describing the data involved in the proposed database

### 5.3.1 Users Table

* Stores the username, hashed password and creation date of each user.
* Each record has a user ID which uniquely identifies that user.

A close-up of a password

Description automatically generated

Table 1: DB schema and an example record of USERS table. User Id is the primary key

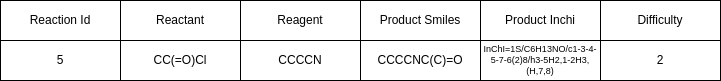
### 5.3.2 Questions Tables

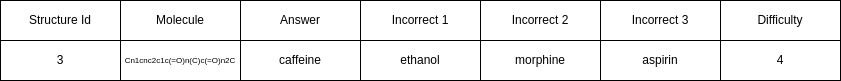
Each type of question will require its own table, containing information about the question to be asked, the correct answer, and a difficulty rating (see **Table 2** and **Table 3** below).

* Multiple-choice questions will include pre-chosen decoy answers.
* Each question will contain a unique ID to identify each question on each table.

Structural information about chemicals will be stored in two established formats:

* SMILES (simplified molecular-input line-entry system): a way to represent chemicals in string format.
  + For example, sodium chloride could be represented as "[Na+].[Cl-]".
  + SMILES can be transformed into SVGs (image format) on the client using a library called RDKit.js**. (Landrum, 2013).**
* InChI (International Chemical Identifier): a more recent method than SMILES to represent chemicals in string format. Unlike SMILES, chemicals in the InChI format are represented in a completely unambiguous manner.
* For example, sodium chloride is "InChI=1S/ClH.Na/h1H;/q;+1/p-1" in the InChI format, whereas both "[Na+].[Cl-]" and "[Cl-].[Na+]" are valid SMILES.
* To compare with the stored InChI string answer to a quiz question, a user's inputted chemical structure input needs to be converted the InChI format using RDKit.js.

 Table 2: DB schema and an example record of REACTIONS table. Reaction Id is the primary key

 Table 3: DB schema and an example record of STRUCTURES table. Structure Id is the primary key

### 5.3.3 Scores Table

* Contains details about the scores a specific user makes for each of their quiz attempts – the user Id corresponding to an entry in the Users table, the total score and the attempt date.
* Every record has a Game Id to uniquely identify each quiz attempt. **(Delisle, 2006).**

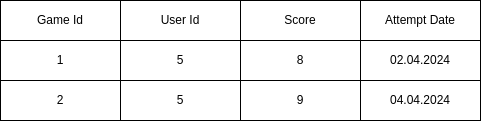


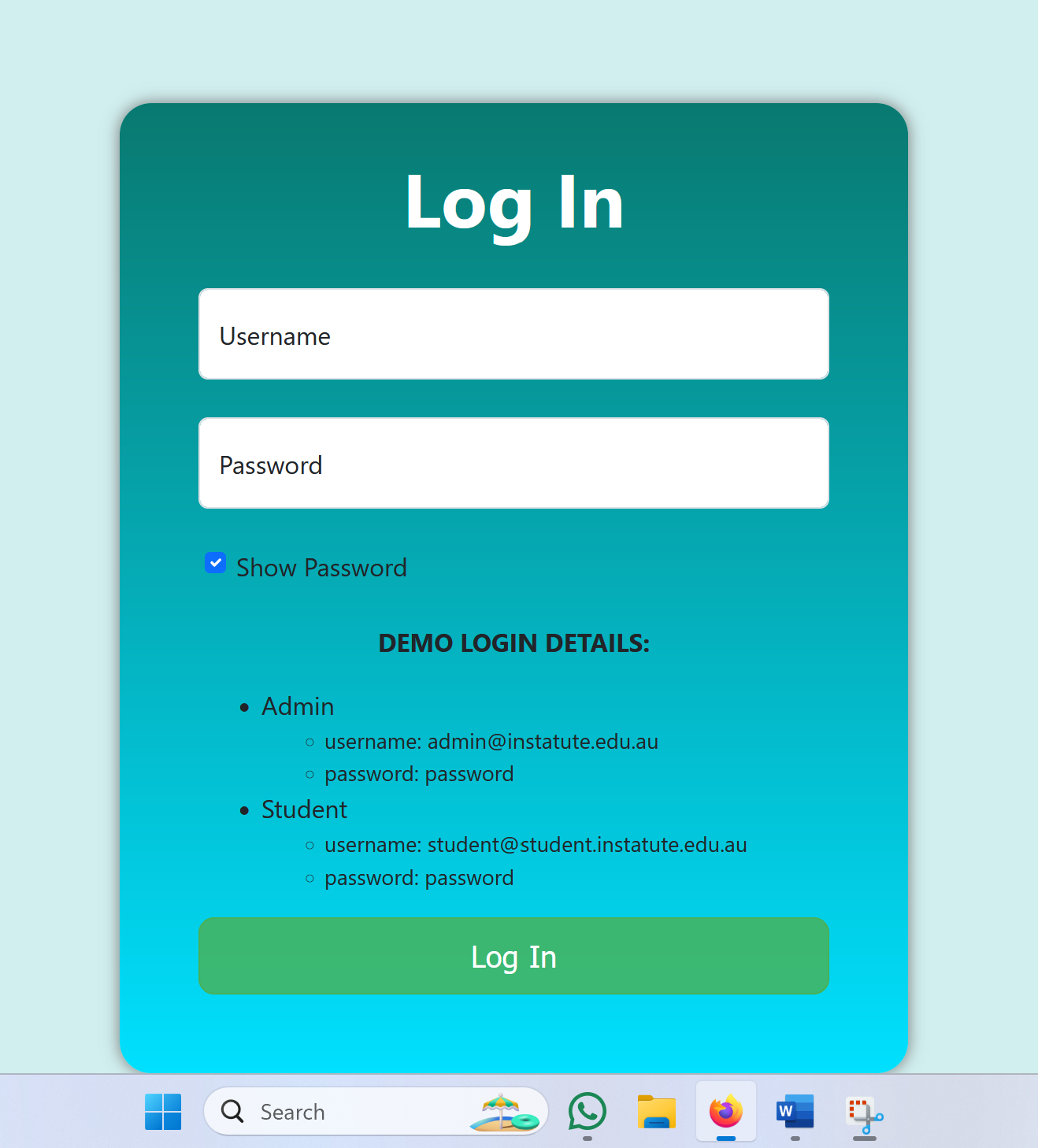
Table 4: DB schema and an example record of SCORES table. Game Id is a primary key, User Id is a foreign key referencing User Id in the USERS table

# 6. Appendix

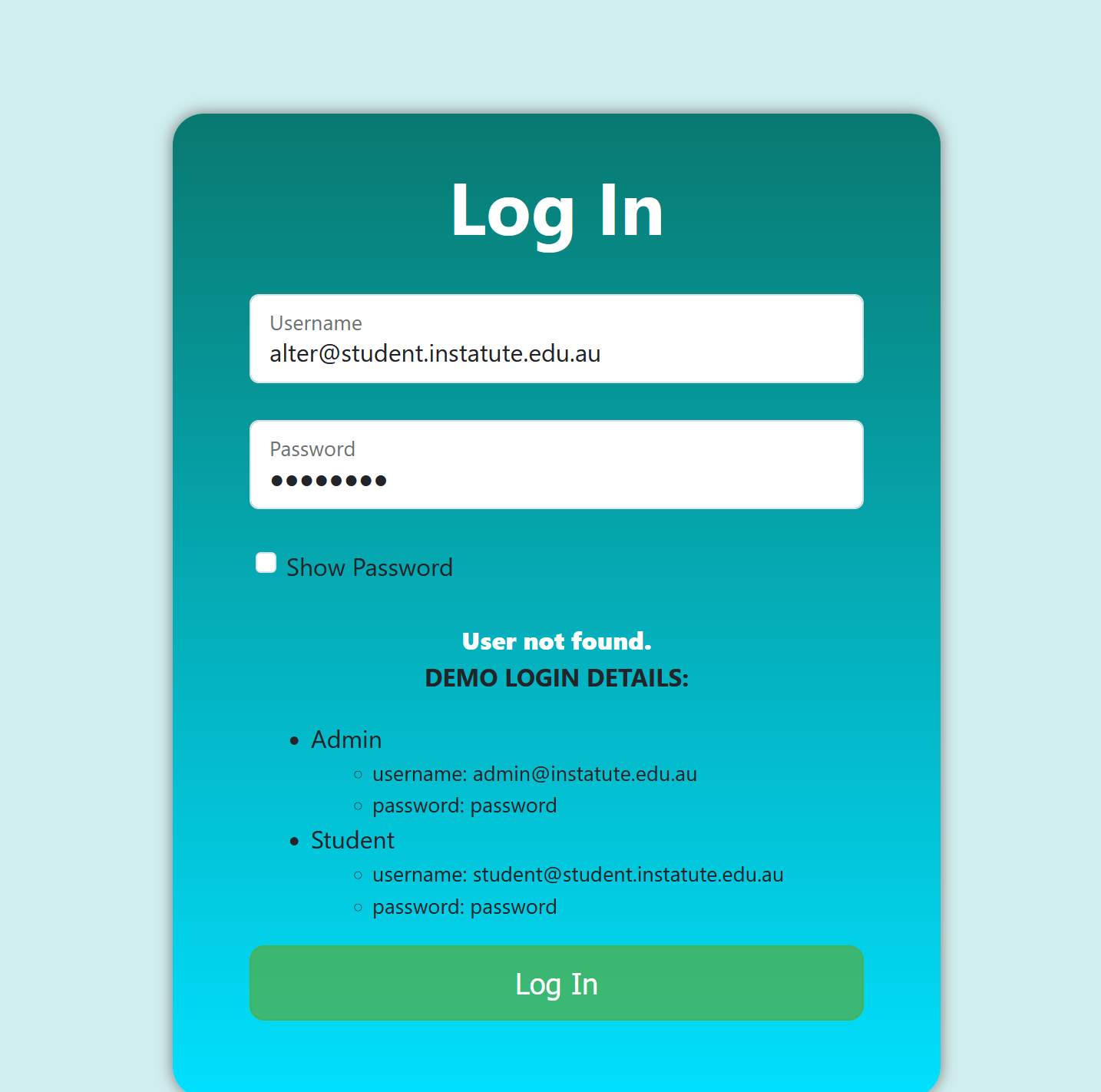
## 6.1User Manual: Step by step guide to use the application:

### 6.1.2 Student User:

1.Please enter the User Name and password.

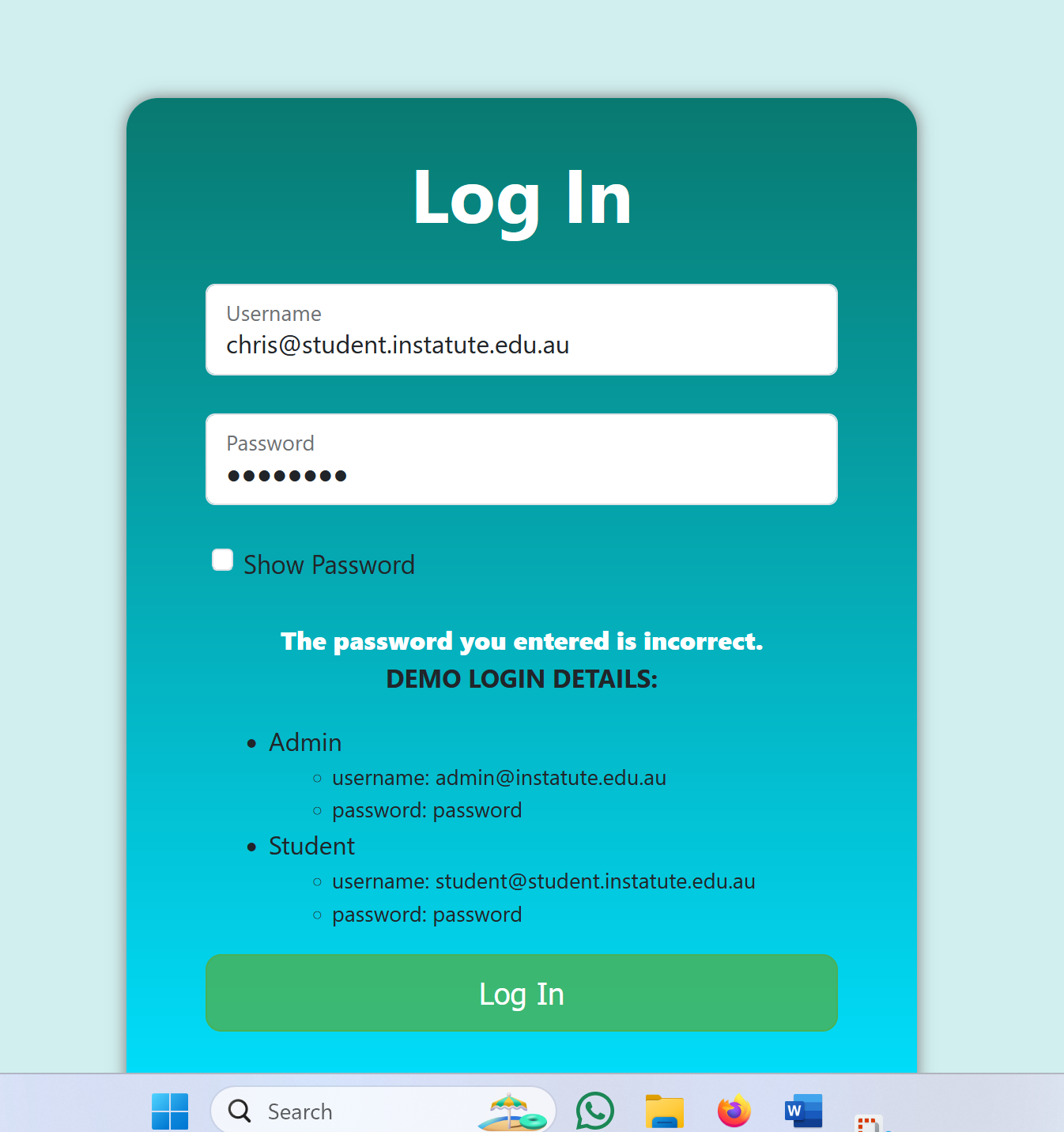


2. If user name is ‘incorrect’, user not found message will be displayed. Please retry.

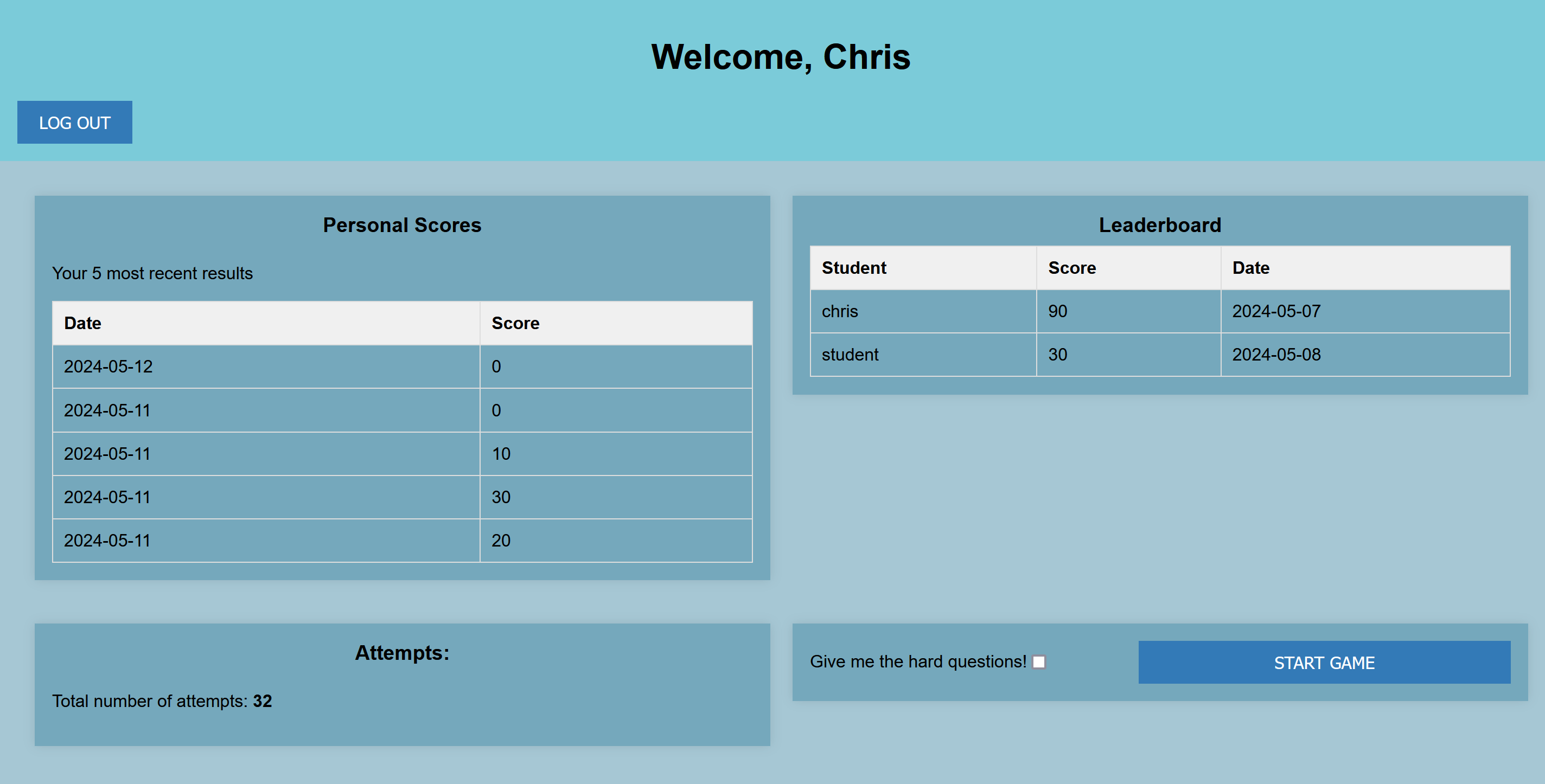


3.If the username is ‘correct’, but password is ‘incorrect’, password incorrect message is displayed.

Please retry.

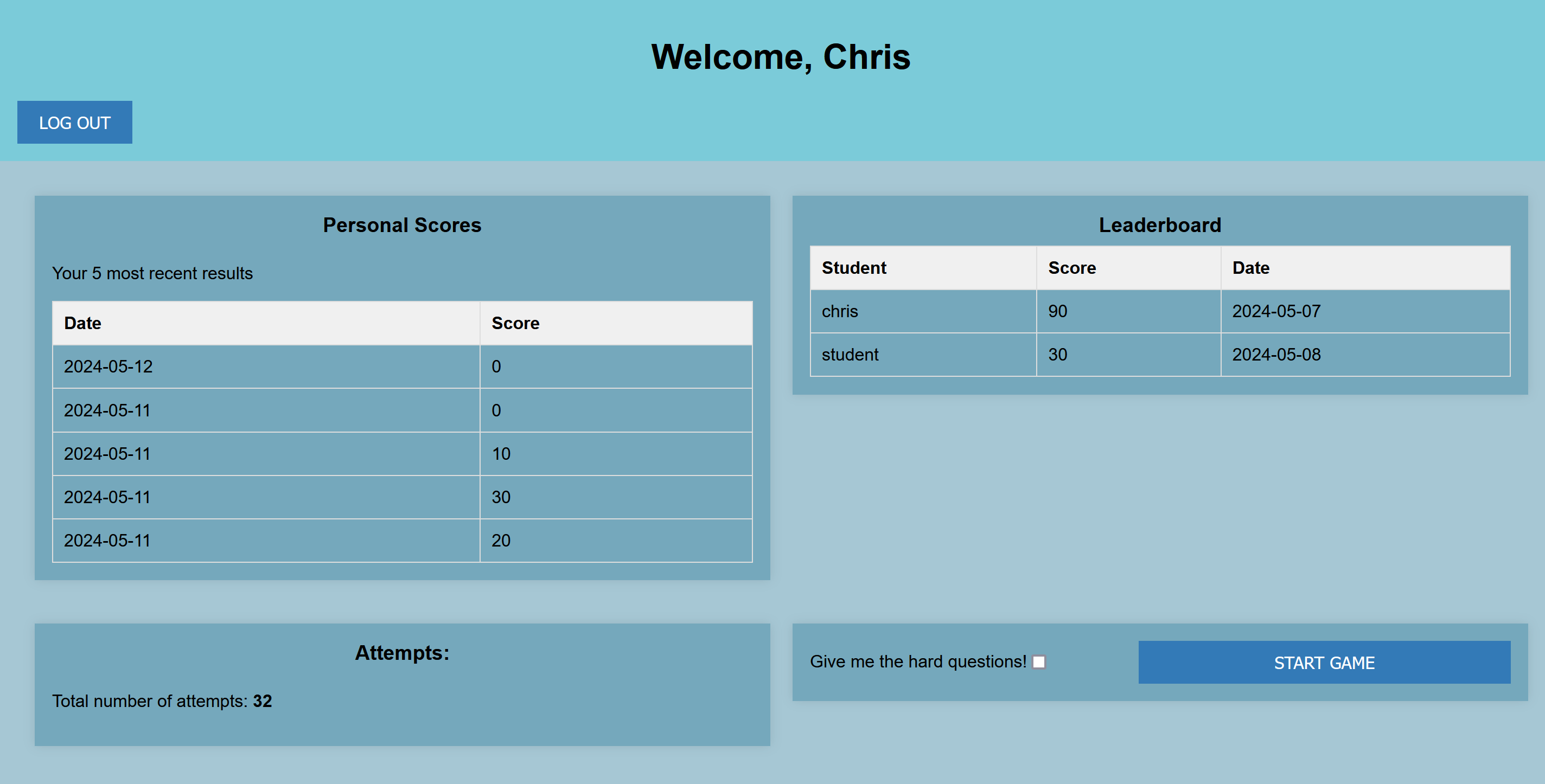


4.If Correct Username and password is entered, user is sent to the login page.



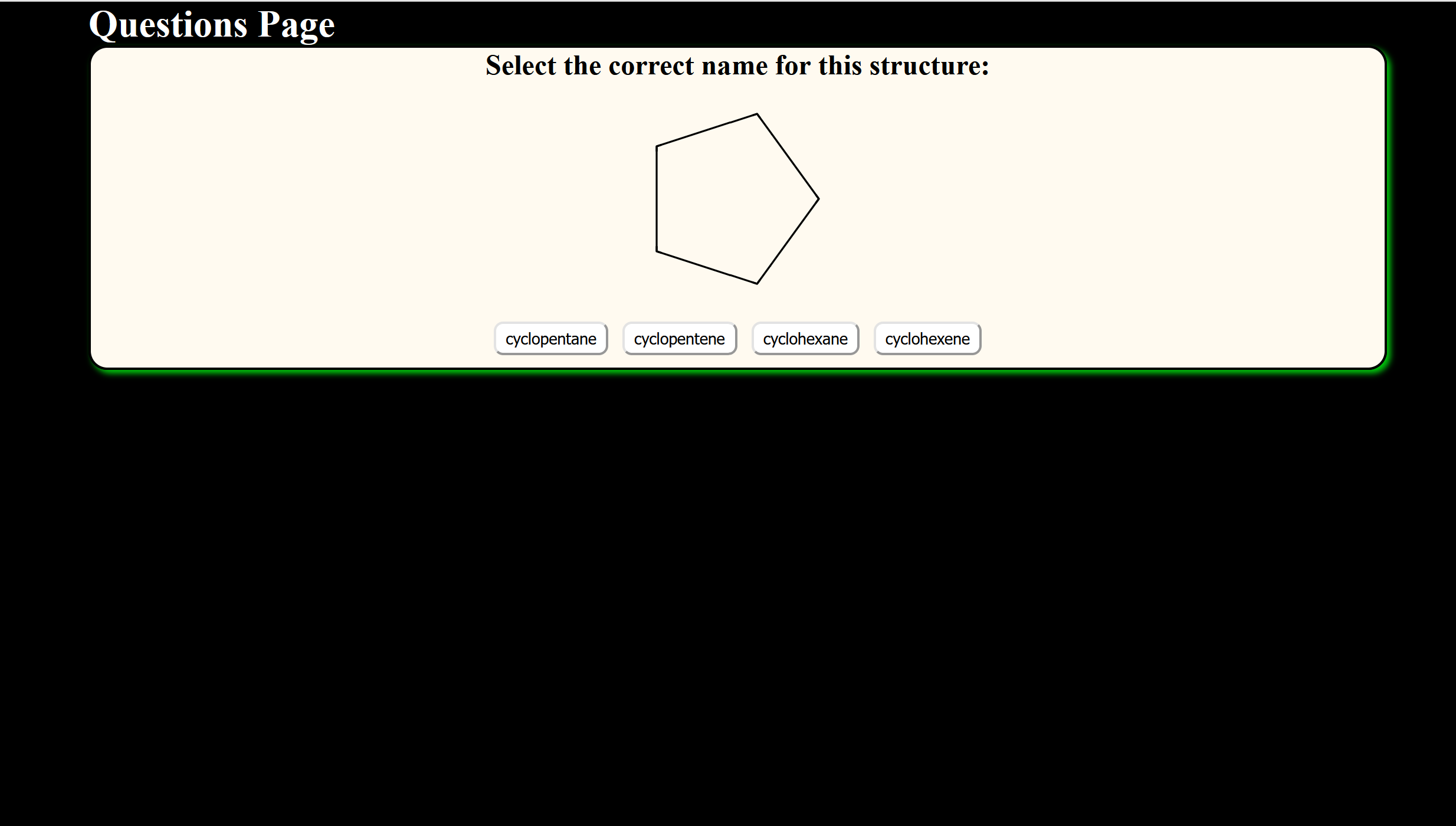
5. Please select the check box for difficult questions on welcome page. Leave it unselected for easy questions. User can see his/her 5 most recent results as well as the student leaderboard.

6.Press START GAME.



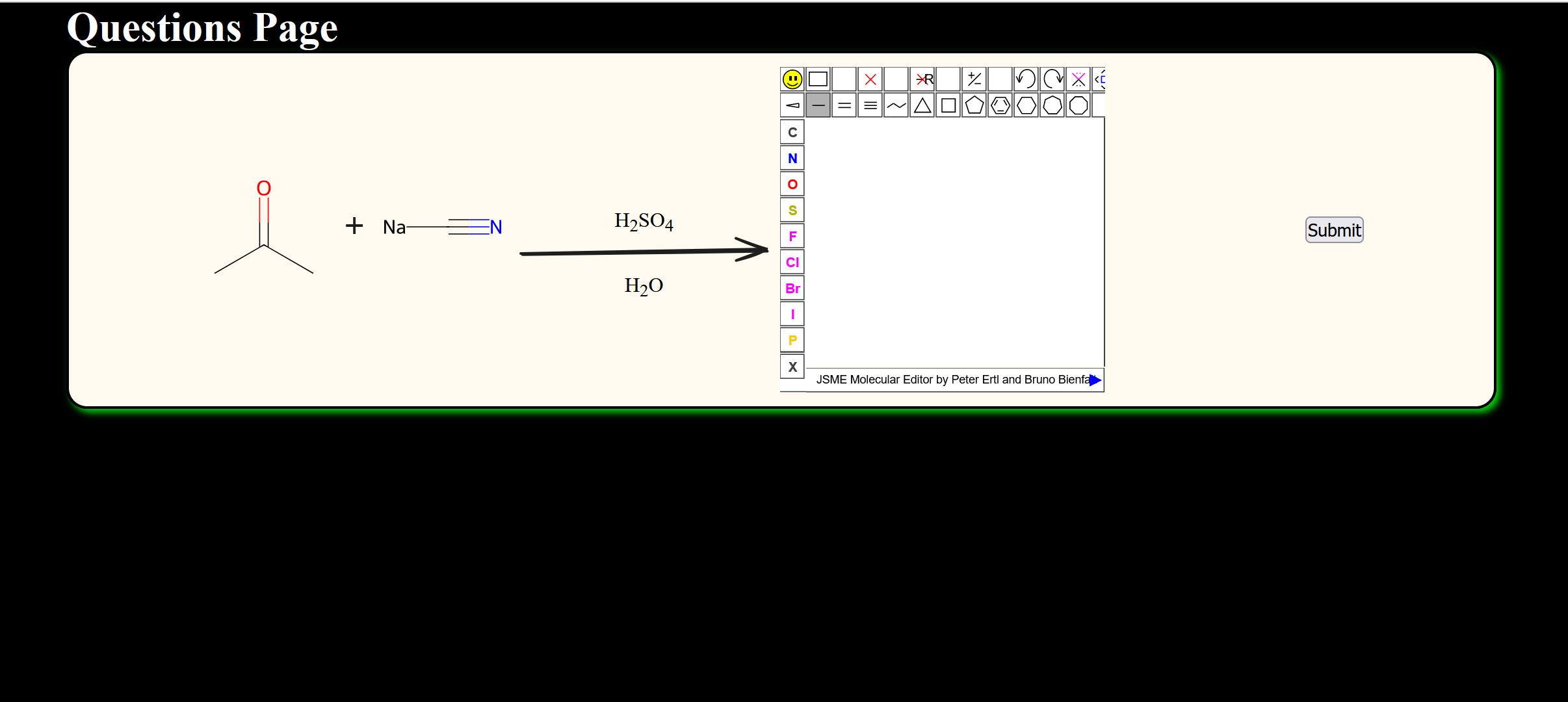
6.User is sent to Questions page.

7.Select an answer if it’s a multi choice question.



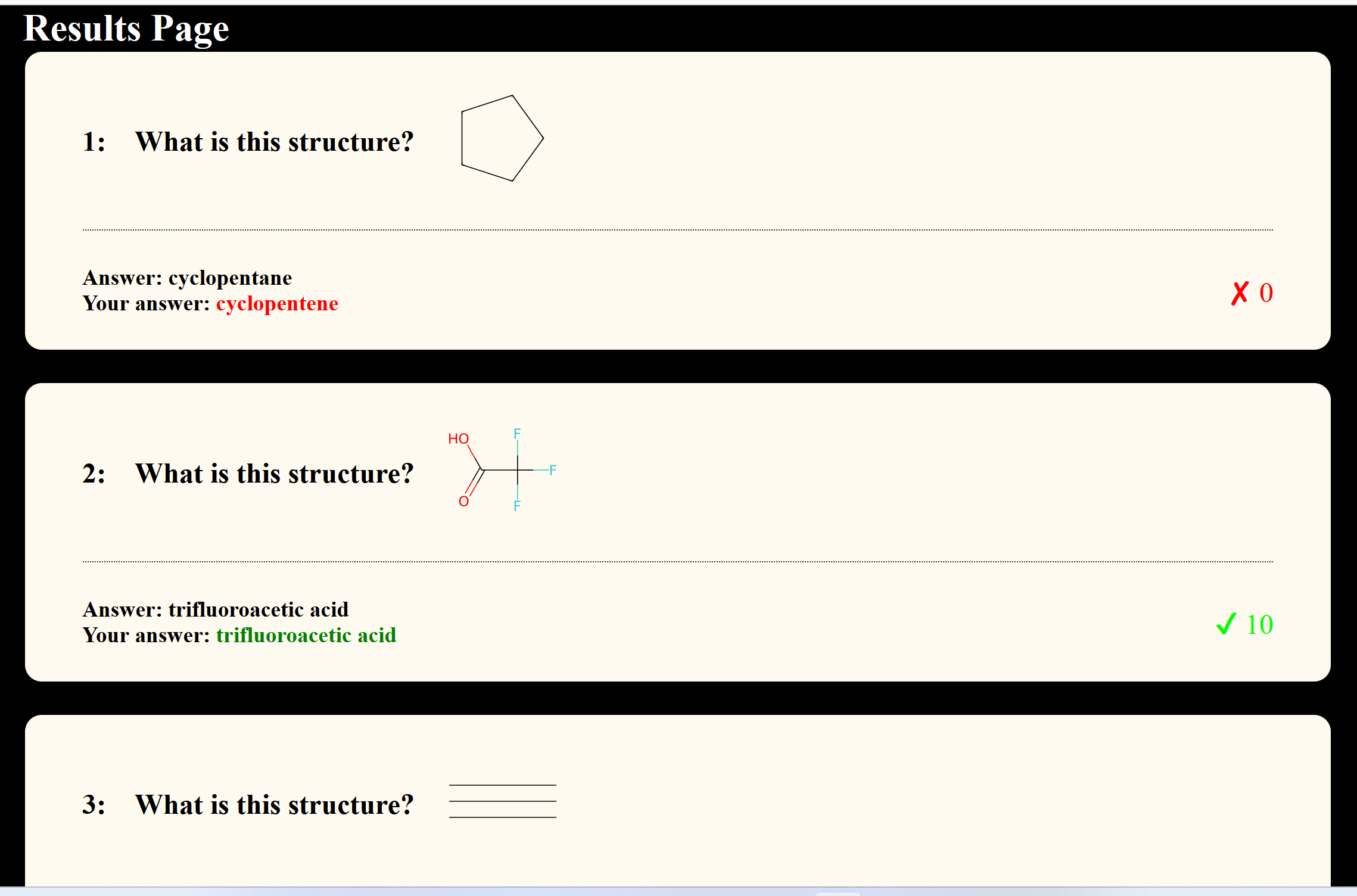
8. Draw the structure to answer the question, if it is a structure type of question.

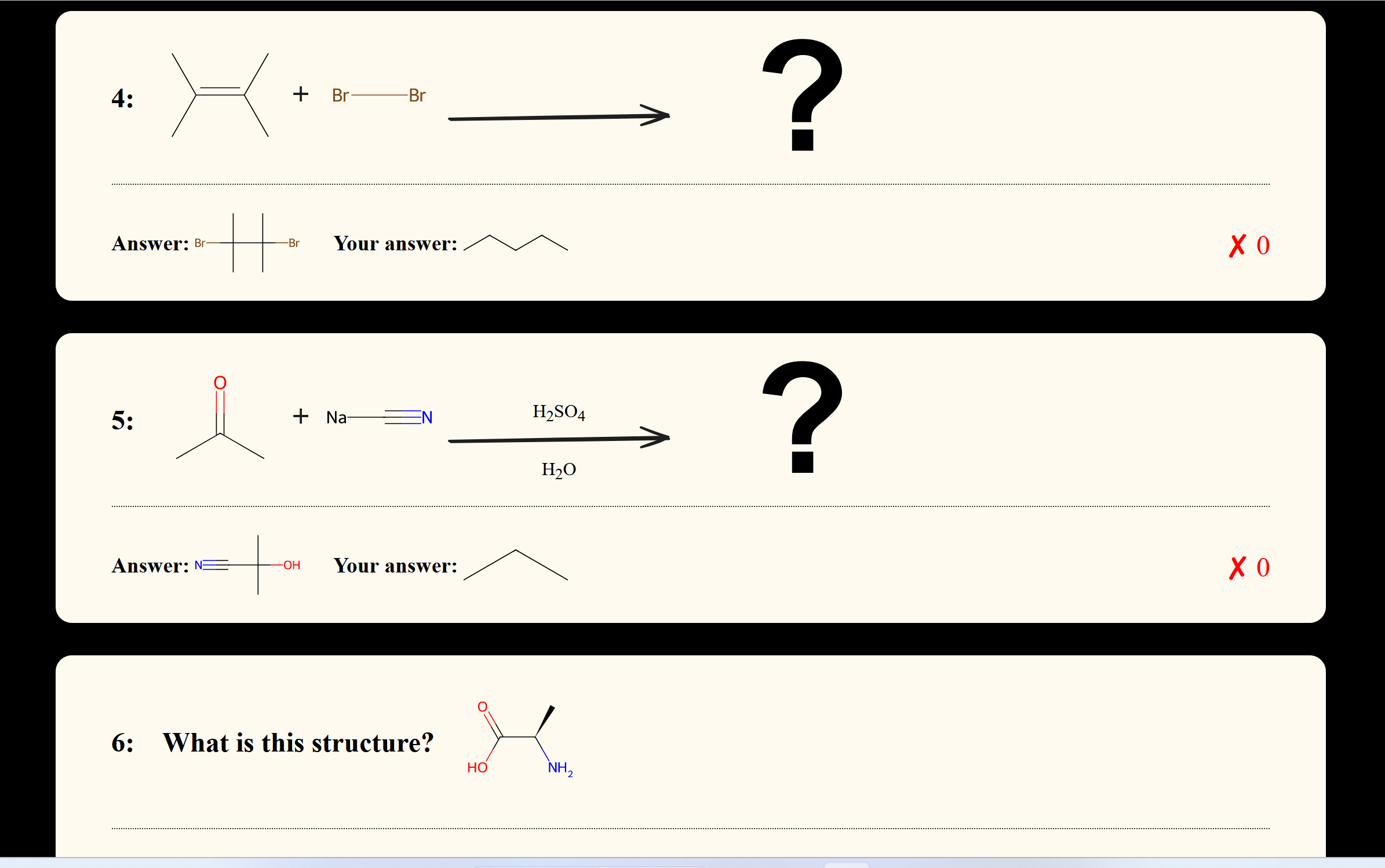
9.Click SUBMIT to submit questions.



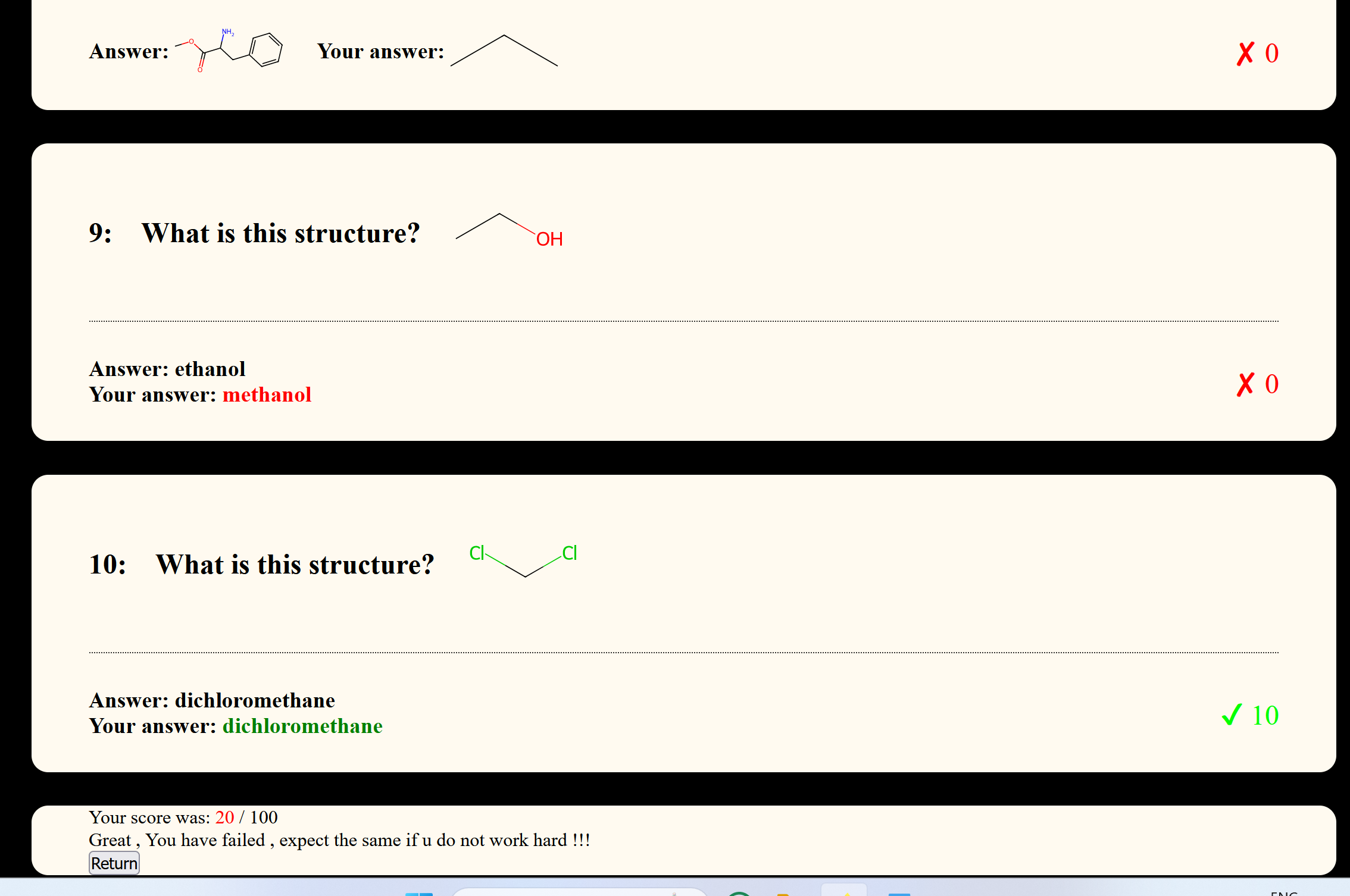
10. On clicking SUBMIT on the 10th and final question, the user is routed to Results page.

11. User is shown his/her set of questions which were given, his/her answer, and the actual correct answer.





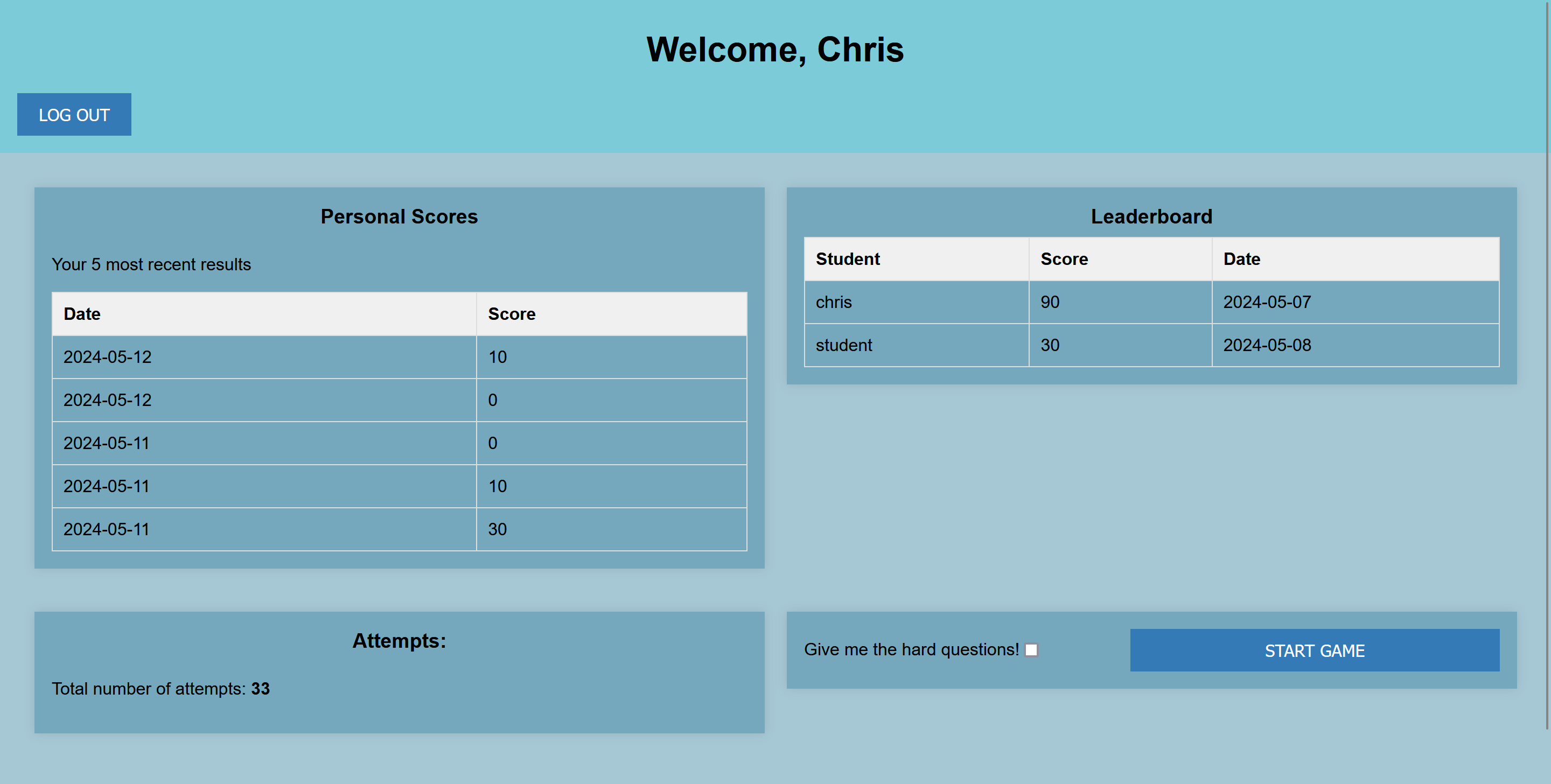
12. User is also shown their score, and a custom message based on their performance.



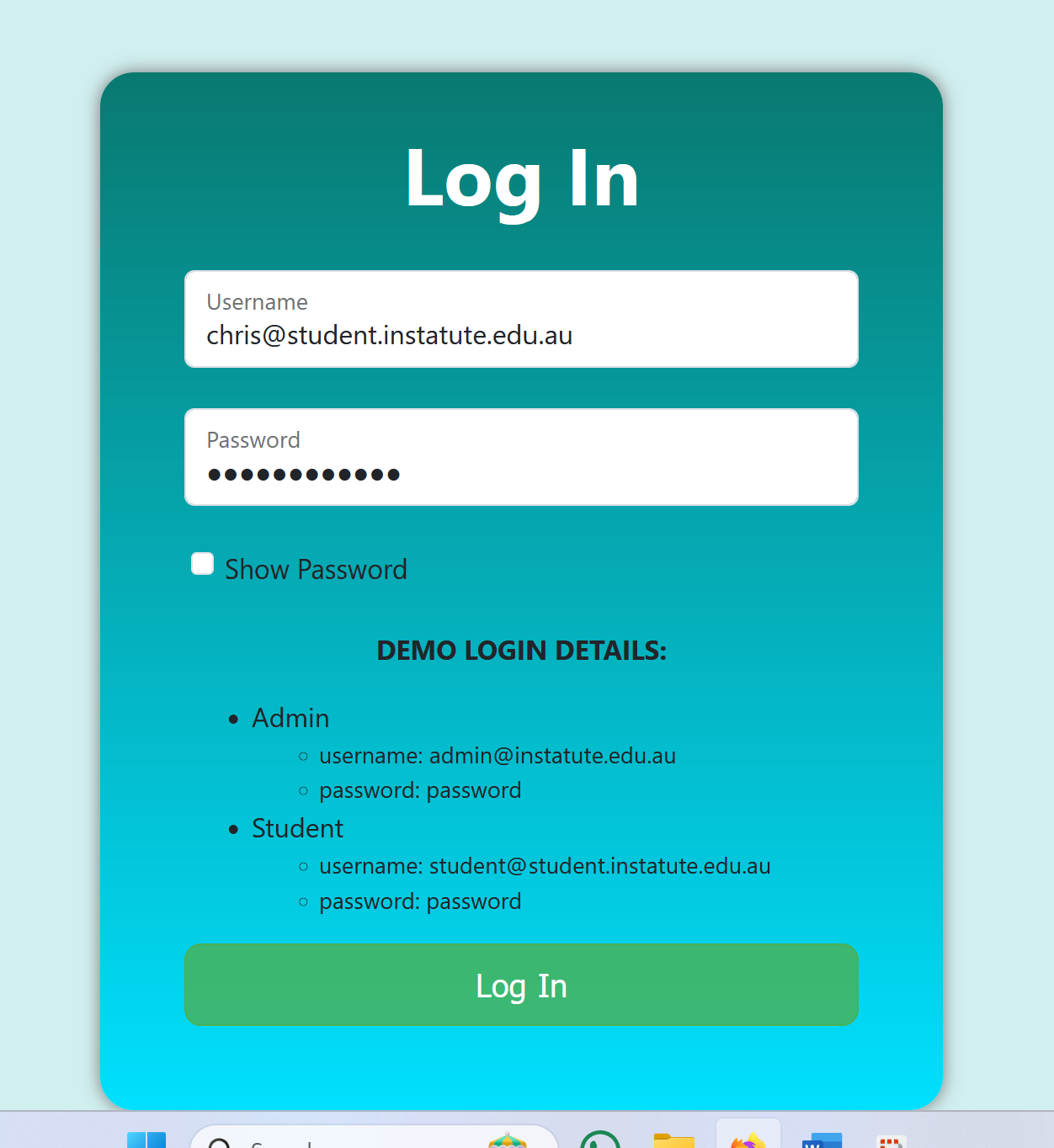
13.On clicking RETURN button, user is navigated back to the welcome page.

14. User can see his/her updated score and updated leaderboard.

15. User can press START GAME to either start playing the quiz again (or) can press LOGOUT to exit the session.



16. On pressing LOG OUT, the user is logged out of his/her session and end up on the Login page.



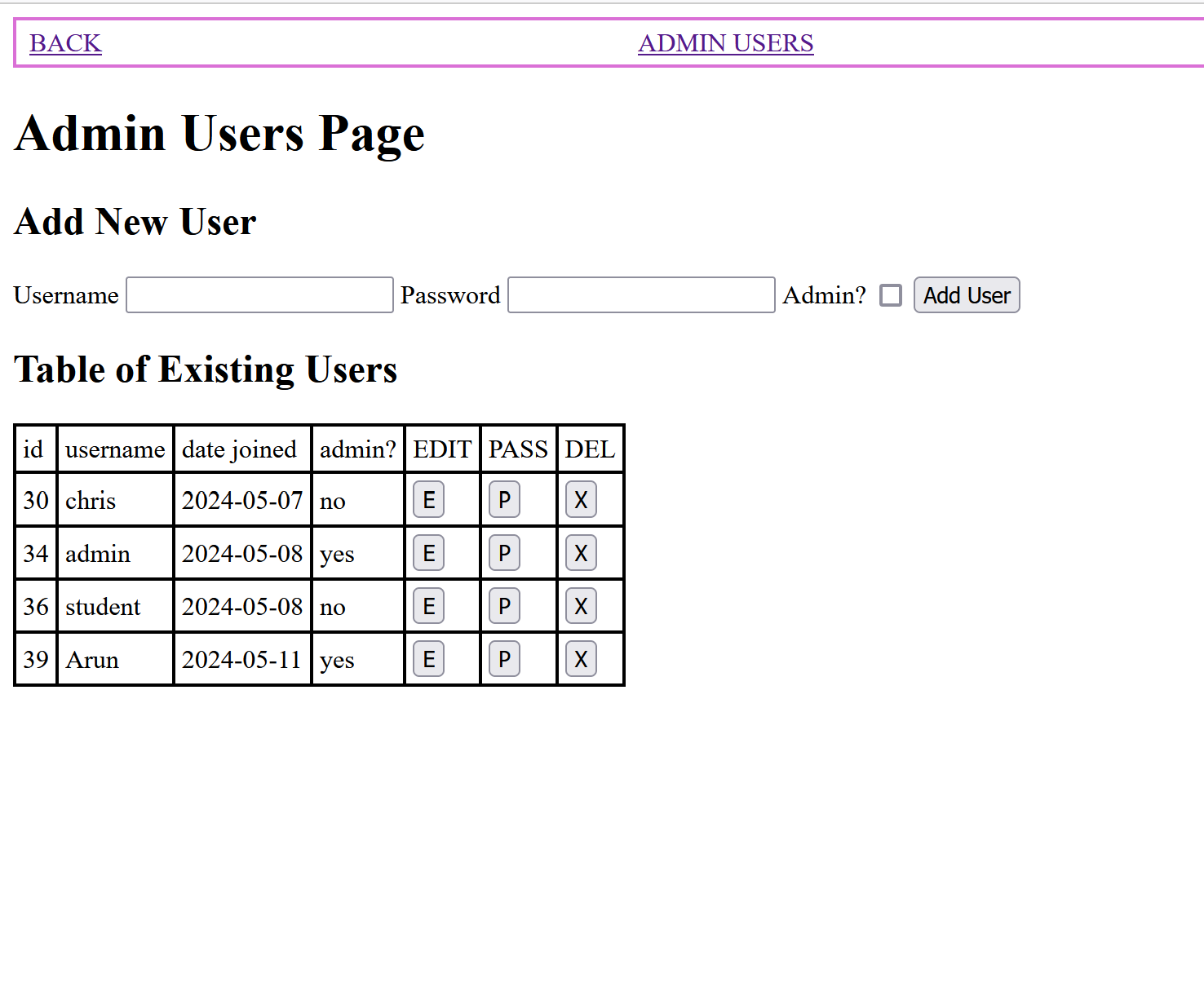
### 6.1.3 Admin User:

1.Upon entering Admin User name and password, an admin user is navigated to Admin page.

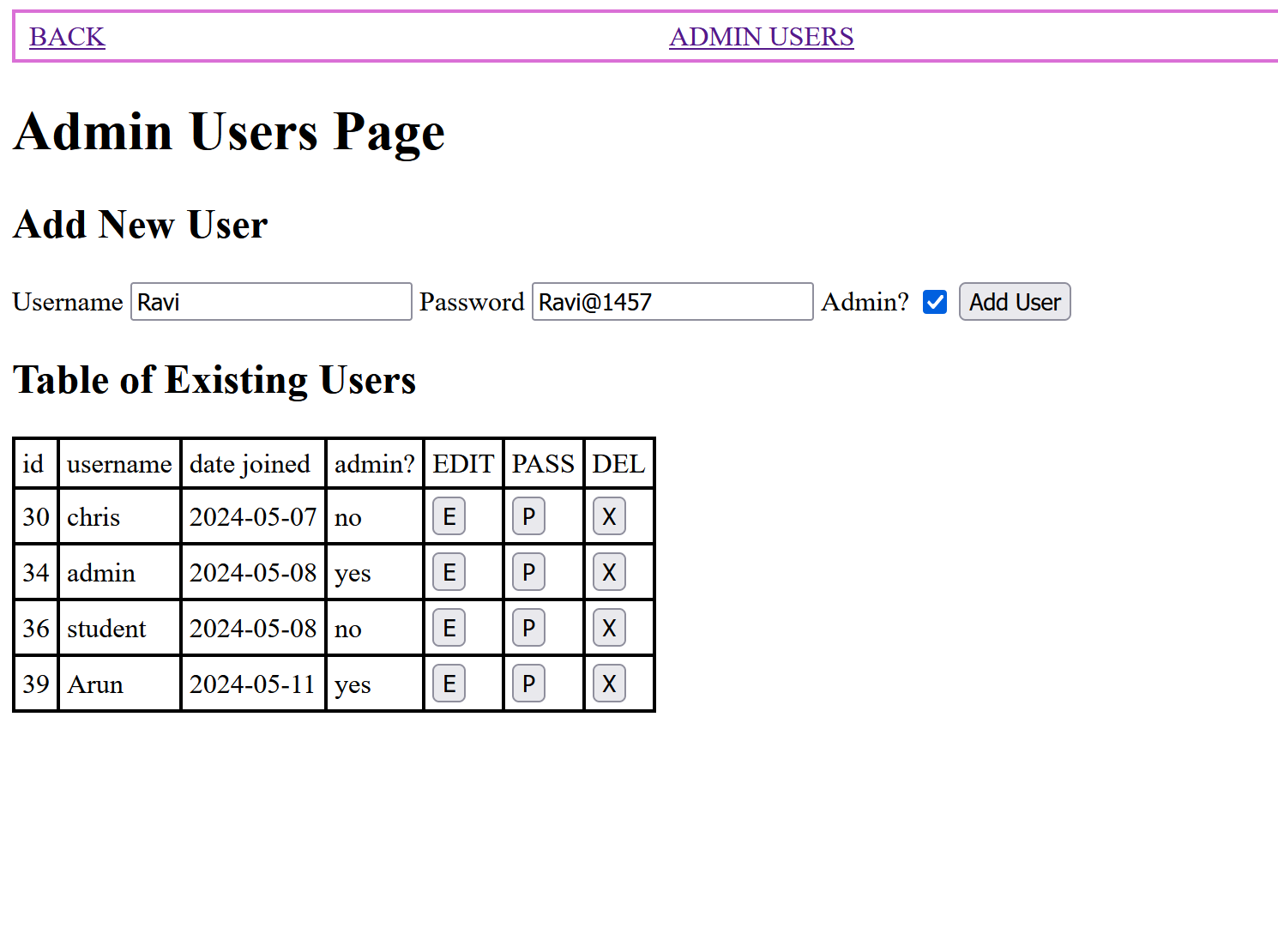


2. Admin user can choose ‘ADMIN USERS’ option to

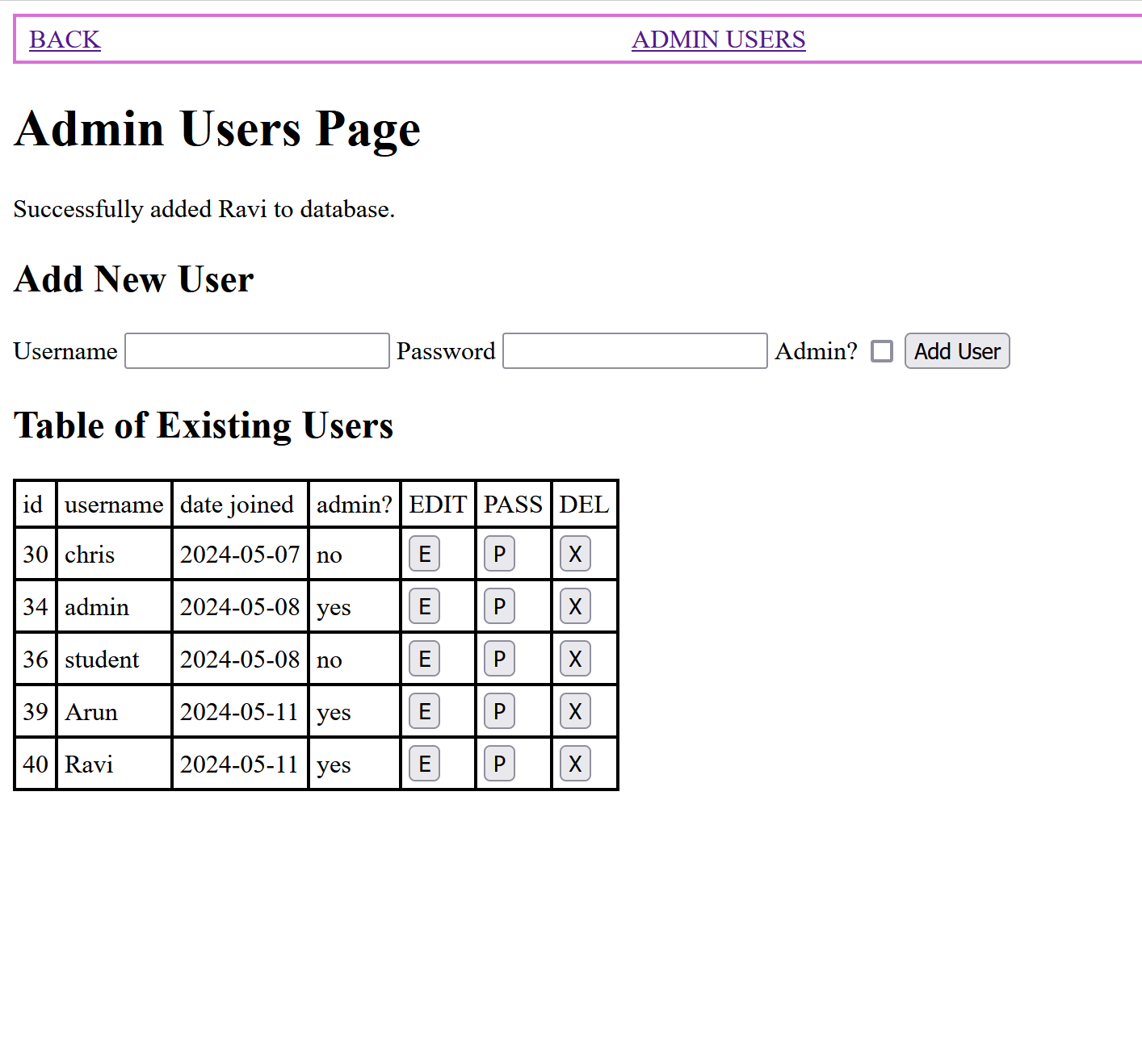
* ADD a new admin user
* DELETE or EDIT details of an existing admin user



3. Admin chooses to add a new user.

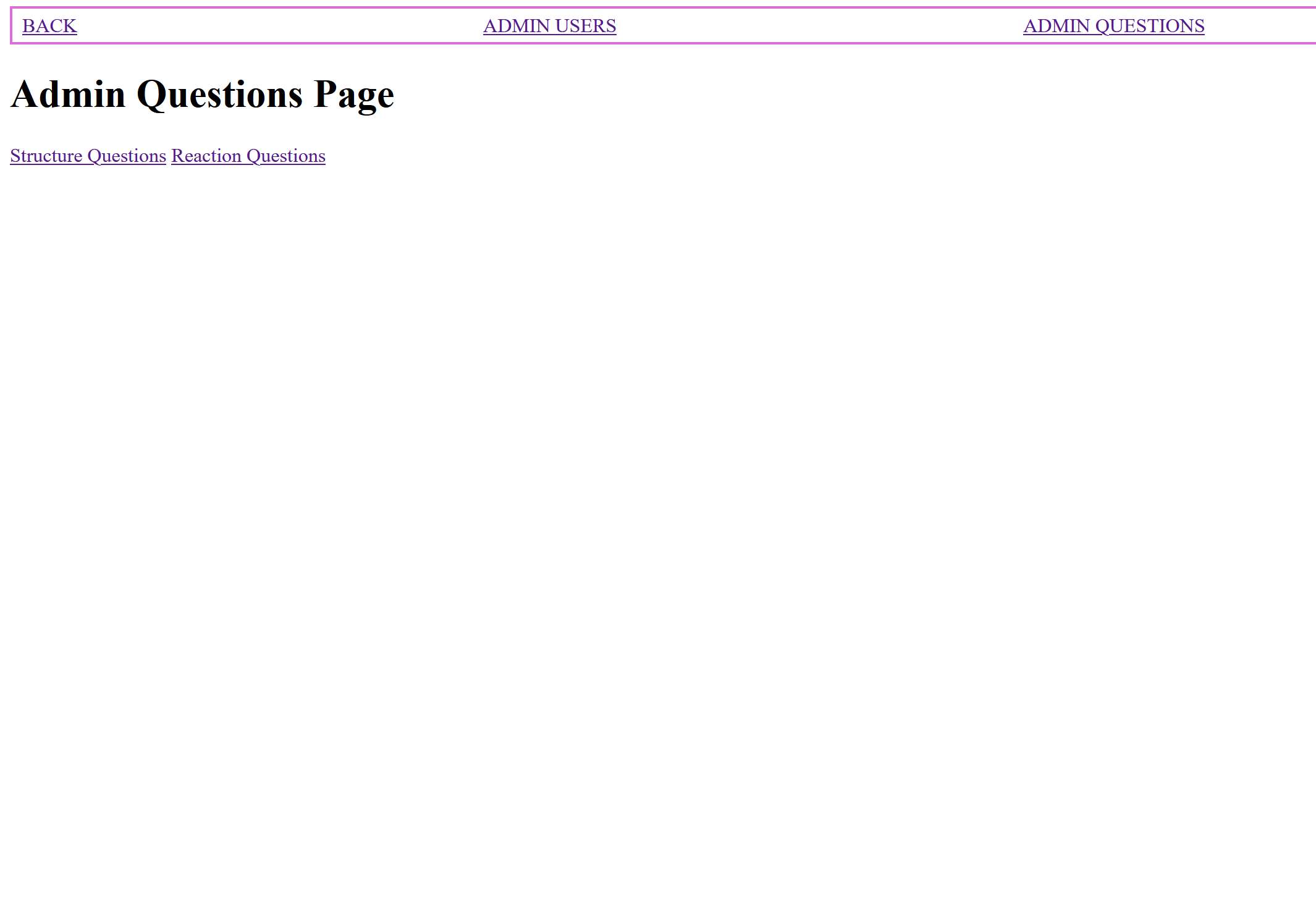


4.New user added and updated in the Users Database.

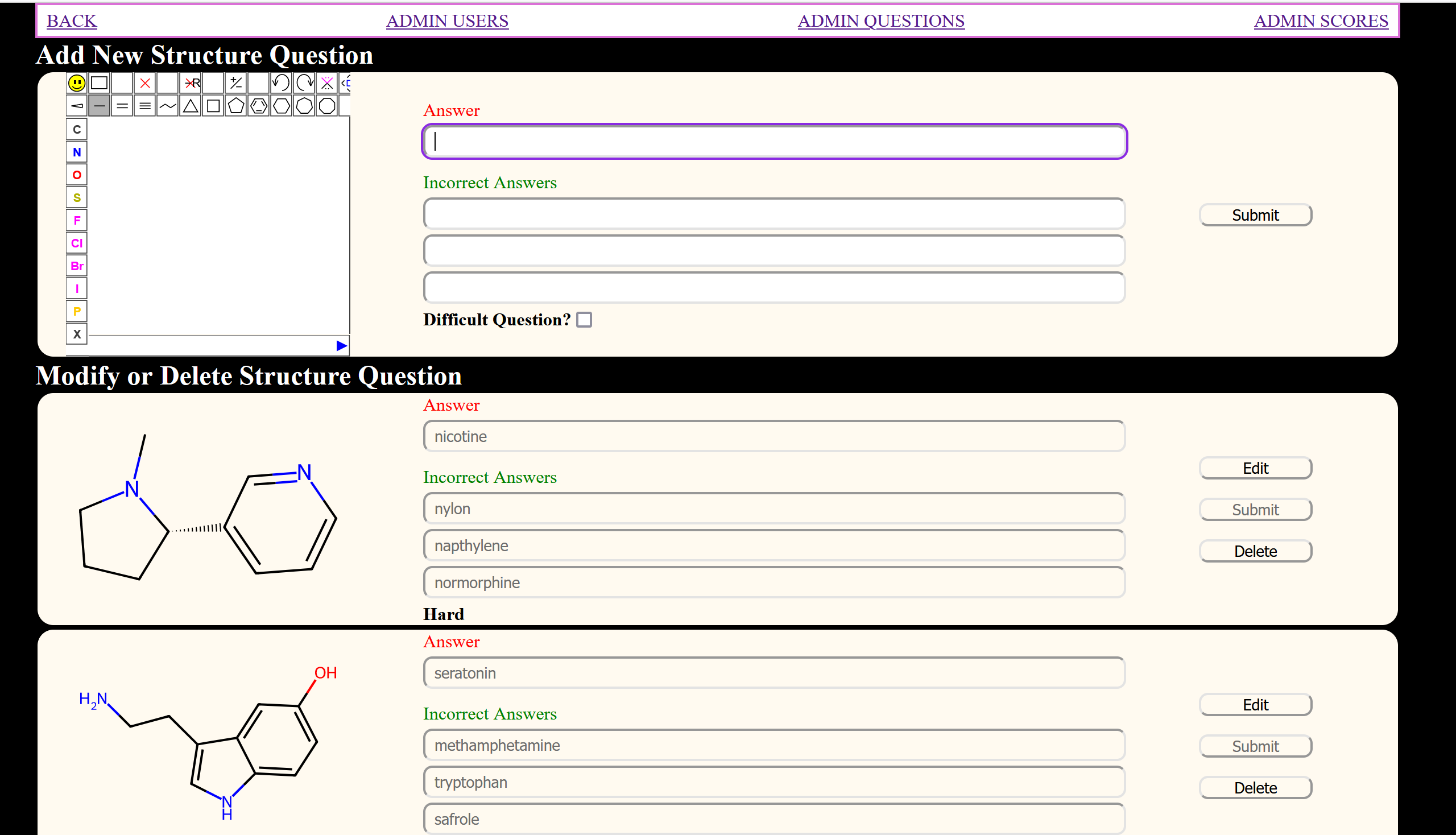


5. Admin can go to the ADMIN QUESTIONS option to

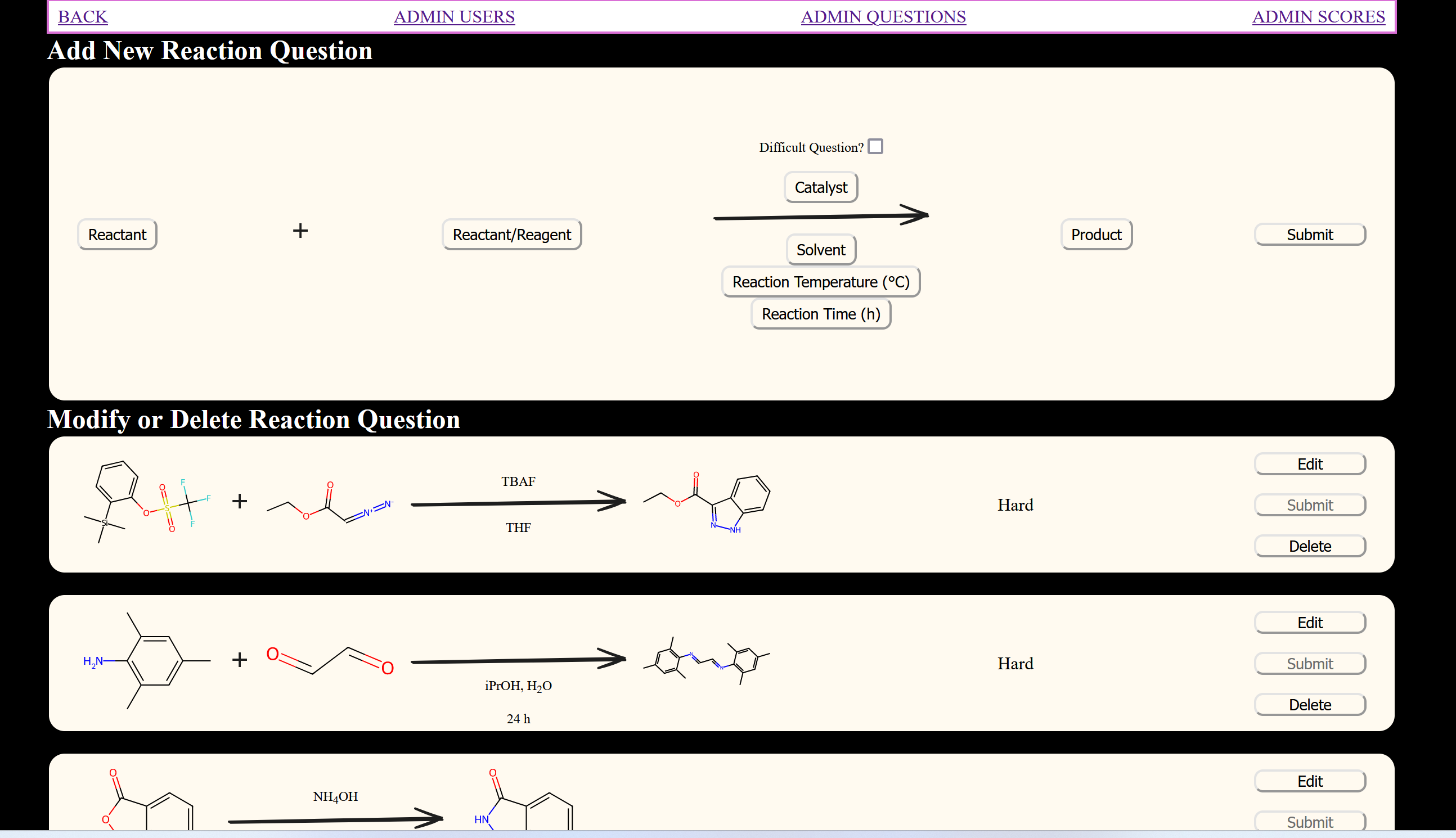
* CREATE and ADD, DELETE or MODIFY a STRUCTURE type question
* CREATE and ADD, DELETE or MODIFY a REACTION type question



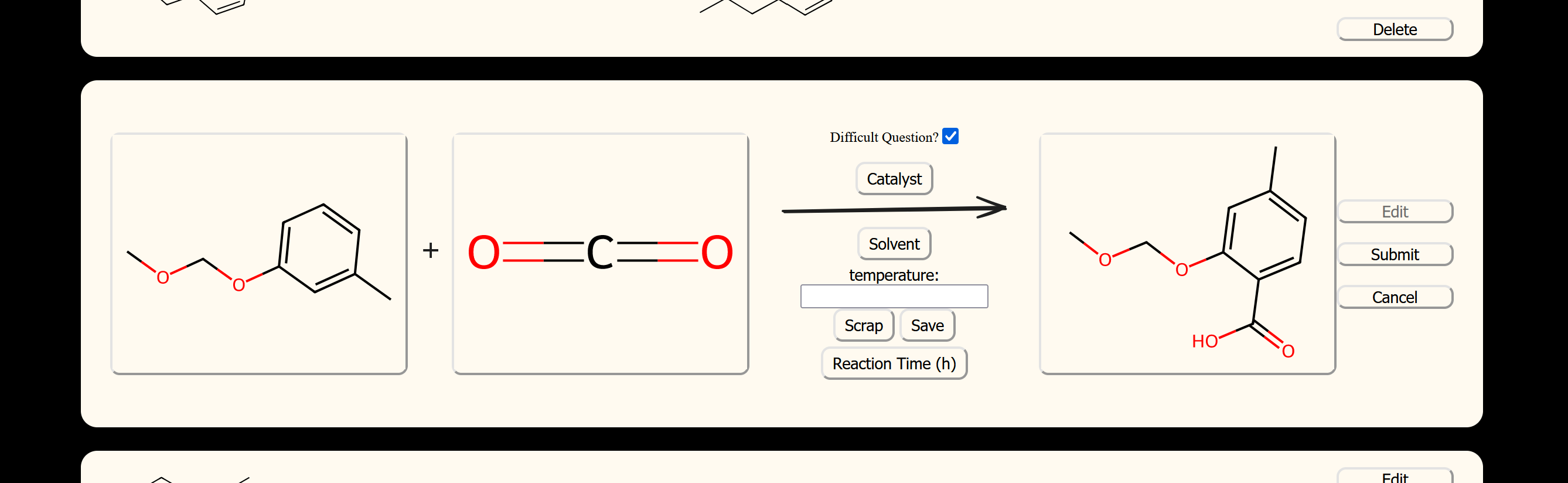
6. Admin chooses to create and add a new Structure type Questions



7. Admin chooses to modify an existing Multiple choice type Question.

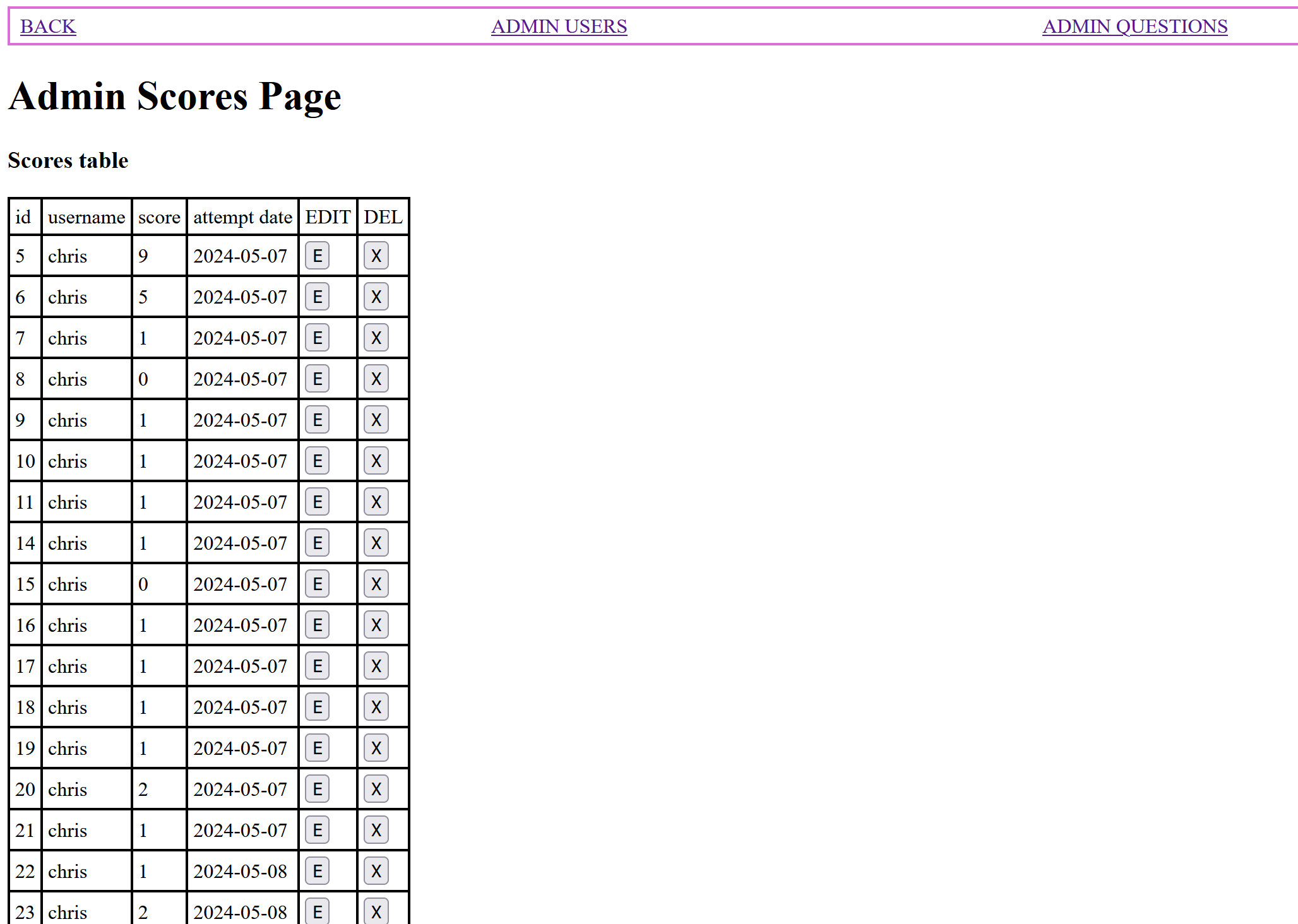


8. Admin chooses to modify an existing ‘Draw the Structure’ type Question and change its DIFFICULTY level.

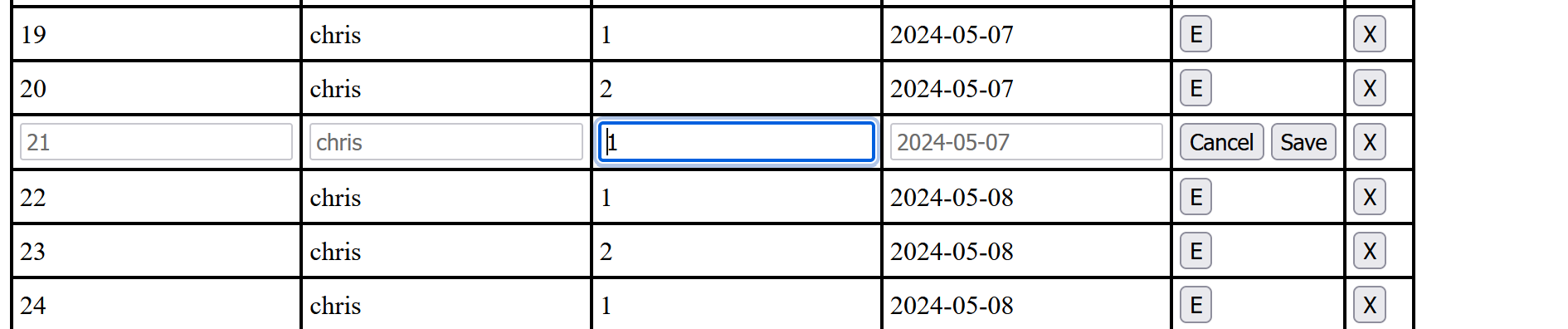


9. Admin can go to the ‘ADMIN Questions’ option and

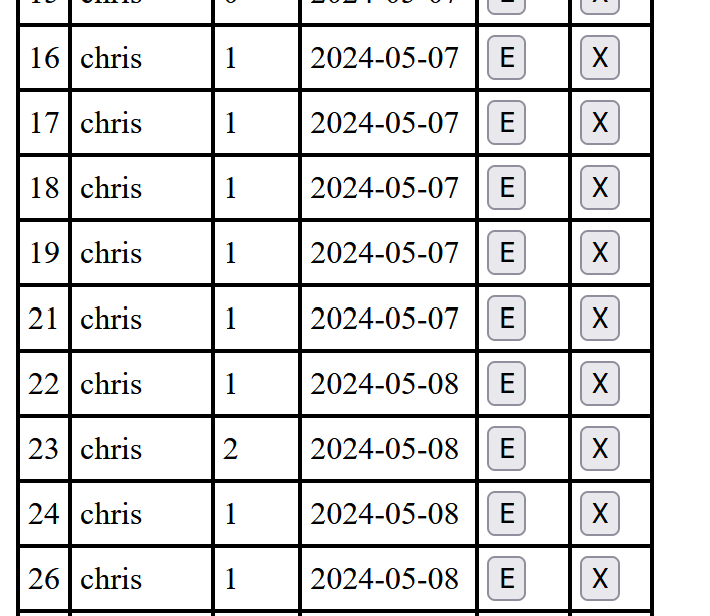
* MODIFY existing student scores
* DELETE existing student scores



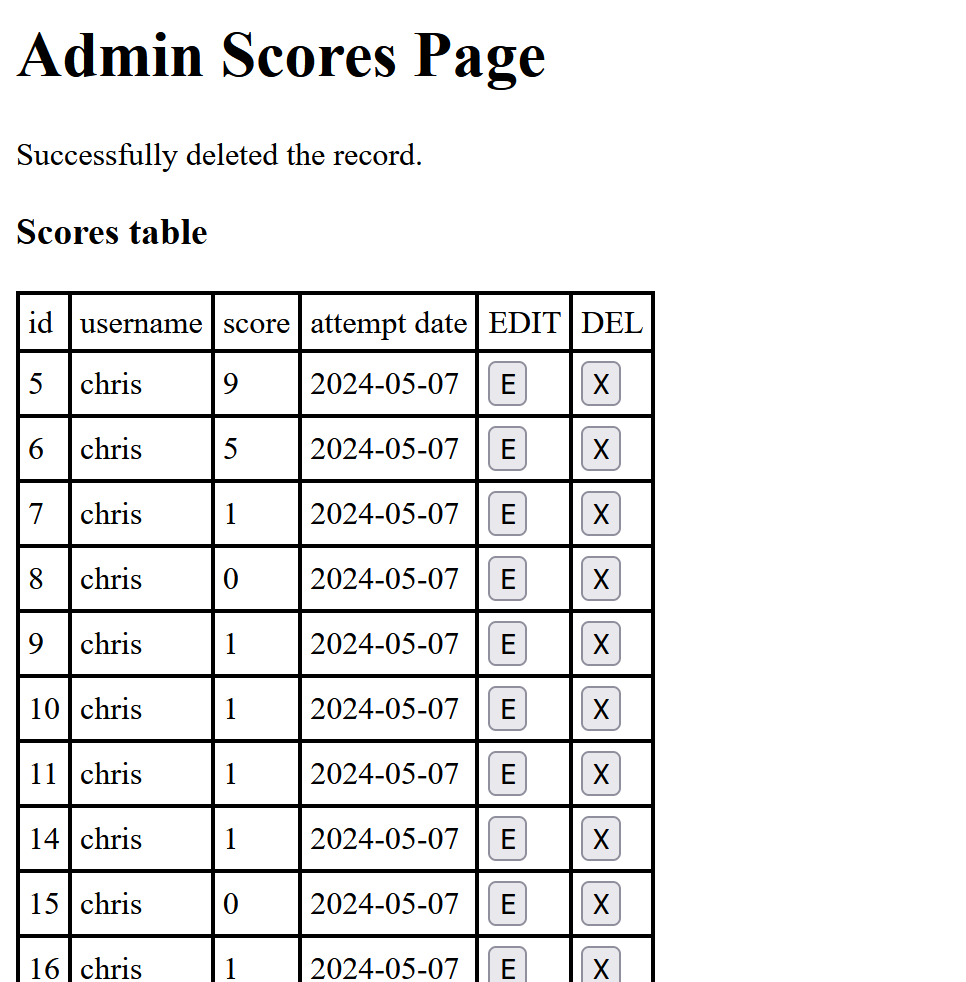
10. Admin choose to EDIT a student score from the Scores table.



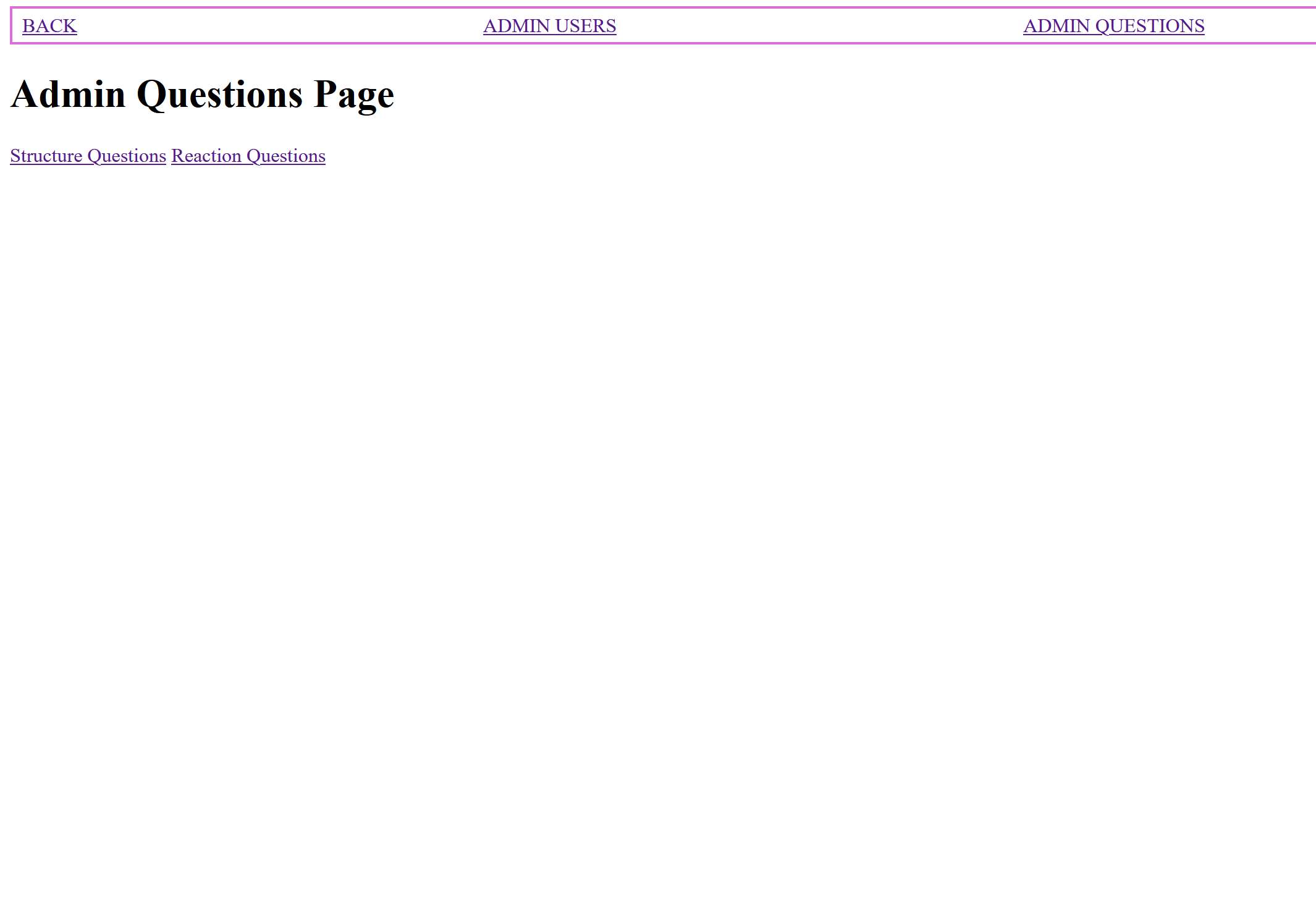
11. Admin DELETES a student record from the table.



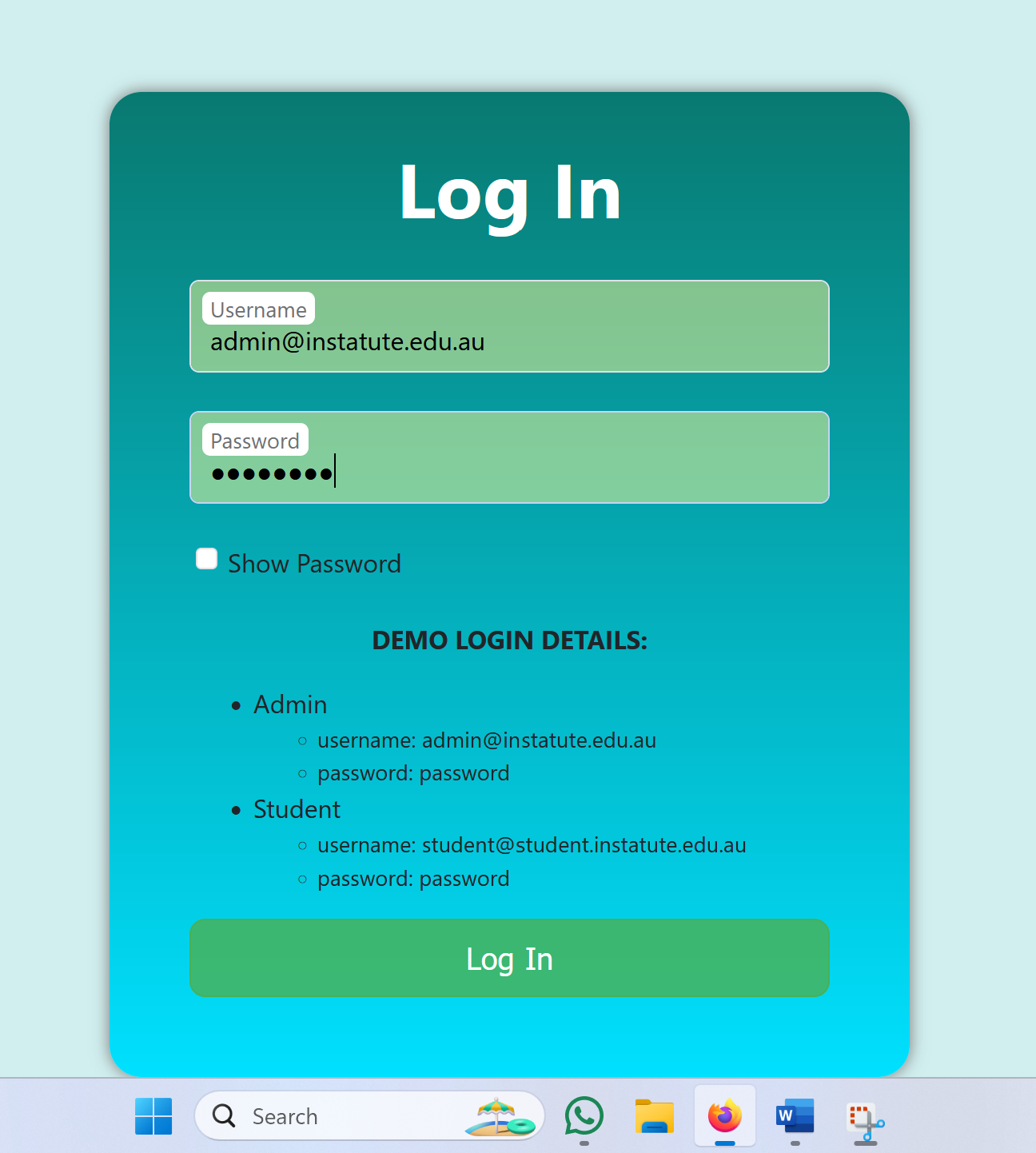
12. The record is deleted and the admin is notified of the same.



13. On pressing BACK option.



14. Admin user is logged out; the session is closed and the user is navigated to the Login page.



## 6.2 Abbreviations

**DB**: Database

**InChI**: International Chemical Identifier

**JSON**: JavaScript Object Notation

**PHP**: PHP Hypertext Preprocessor / Personal Home Page

**SMILES**: Simplified Molecular-Input Line-Entry System

**SQL**: Structured Query Language

**SVG**: Scalable Vector Graphics

## 6.2 List of figures and tables:

**Figure. 1**: User Interface of Login Page

**Figure. 2**: User Interface of Welcome Page

**Figure. 3**: User Interface of Questions page for a Structure Question

**Figure. 4**: User Interface of Questions page for a Structure Question

**Figure. 5**: User Interface of Results page

**Figure. 6**: User Interface of Results page with the scores and ‘Return’ button

**Figure. 7**: User Interface of Admin page with the scores and ‘Return’ button

**Figure 8**: Example user interface sketch for the login page

**Figure 9**: Example user interface sketch for the welcome page

**Figure 10**: Example user interface sketch for question pages 1 and 10

**Figure 11**: Example user interface sketch for the results page

**Figure 12**: User interface sketch for the results page, with the outcome of clicking the “Users” button

**Figure 13**: Step-by-step description of the flow of execution of the program for students

**Figure 14**: Step-by-step description of the flow of execution of the program for administrators

**Figure 15**: Entity-relationship diagram describing the tables involved in the proposed database

**Table 1**: DB schema and an example record of USERS table. User Id is the primary key

**Table 2**: DB schema and an example record of REACTIONS table. Reaction Id is the primary key

**Table 3**: DB schema and an example record of STRUCTURES table. Structure Id is the primary key

**Table 4**: DB schema and an example record of SCORES table. Game Id is a primary key, User Id is a foreign key referencing User Id in the USERS table.

# 7. References

Andrew, R., Ullman, C., & Waters, C. (**2003**). Fundamental Web Design and Development Skills. *Glasshouse*.

Bienfait, B., Ertl, P. J. Cheminformatics, L. (**2013**). JSME: a free molecule editor in JavaScript. *SpringerNature*.

Delisle, M. (**2006**). Creating Your MySQL Database. *Packt Publishing Ltd*.

Full, M. (**2020**). How the Internet Works and the Web Development Process. *Independently Published*.

Gehani, N. (**2011**). The Database Application Book Using the MySQL Database System. *Apress Media, Llc*.

Landrum, G. (**2013**). RDKit.js. https://www.rdkitjs.com/

Nixon, R. (**2021**). Learning PHP, MySQL, and JavaScript. *O’Reilly Media, Inc*.

Prettyman, S. (**2020**). Learn PHP 8: using MySQL, JavaScript, CSS3, and HTML5. *Apress Media, Llc.*

Simpson, J. (**2023**). How JavaScript Works. *Apress Media Llc*.

Smith, B. (**2015**). Beginning JSON: [learn the preferred data format of the web]. *Apress Media, Llc*.

Solomons, T. W. G., Fryhle, C. B., Snyder, S. A. (**2017**) Organic chemistry, 12th ed. *John Wiley & Sons, Inc*.