2kool4skool Escape Room Quiz

06.03.2020

Nelson Chan

Thomas Lem

NianJun (Christine) Shi

Spring 2020 CS 467 Online Capstone Project

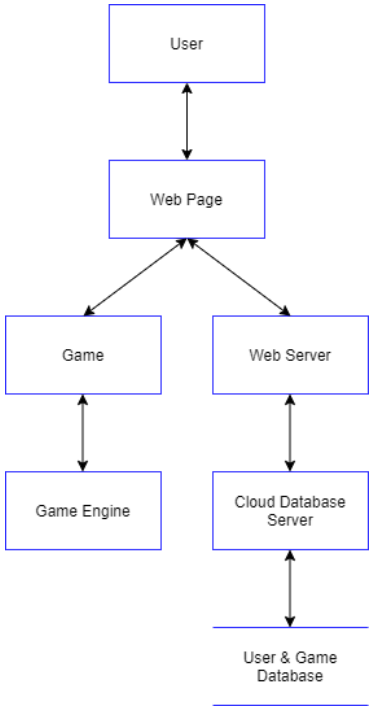
Oregon State University

Introduction

In the beginning of the quarter, our members gathered and selected this project due to our collective interest in escape room style games. In addition, we wanted a project that truly encompasses the concepts we have learned throughout this program, yet remains challenging and allows us to utilize new tools and platforms to develop an end product. Together as a team, we spent over 100+ total hours gathering and setting up resources, brainstorming ideas on how we wanted our product to behave, implementing new features, and testing the finished product. Also, we bonded over the challenges of learning and using new systems and the collective efforts of fixing bugs. With that said, the end product is a website that allows teachers to create quizzes that can be attached to an embedded escape room game for students to play. Moreover, students and teachers can view and print quiz results. In the next section, we will outline the platforms used to make the web application.

Design

Our overall design can be depicted with the following layered architecture data flow diagram.



System Interactions

MySQL Database ↔ Web Application:

The (front-end) web-based user interface provides Create, Read, Update, and Delete (CRUD) functionalities for a small database—handled by the (back-end) server.

MySQL Database ↔ Unity Game:

The web application retrieves the questions from the database and sends those to the game by calling a Unity script function through JavaScript. After the user answers the ten questions, the game sends the user's responses to the backend server using Unity's UnityWebRequest API. Then, the server sends an insert query to the database.

Website

We developed the web application using HTML, CSS Bootstrap, and JavaScript.

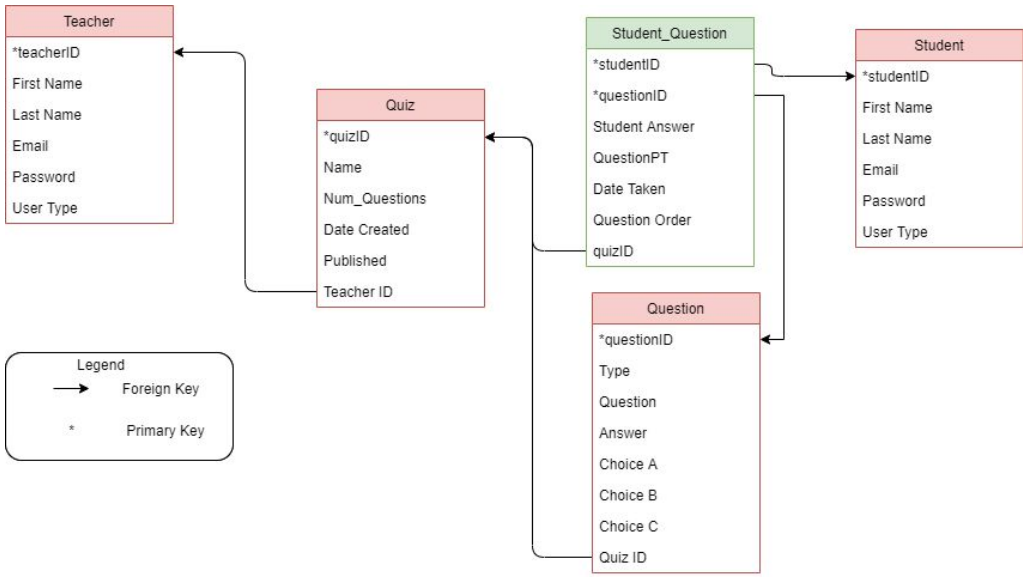
Server

We used Heroku to host the website and Node.js to send information between the website and the database.

Database

We used Google Cloud SQL (in MySQL) to host our database.

Database Schema



Game

We developed the escape room game using Unity in C# language. We exported the game to a WebGL API and embedded it on the website.

Administrative

We used GitHub for version control and code collaboration. We used GIMP for picture editing. We obtained our source images from mainly three websites: sketchfab.com, hiclipart.com and free3d.com. We used Visual Studio Code to write code. And, we consulted the Unity documentation to help us write the game.

Website Navigation

URL: https://osu-2kool4skool.herokuapp.com/

The website supports two types of users: teacher and student. Please use the following credentials to log in and view the different user-specific functionalities.

Home Page

Logging in:

We have created the following model accounts for the client to view and test website features.

1. Teacher:
    a. Username: sjackson@school.com
    b. Password: sjackson
2. Student:
    a. Username: ewilson@study.com
    b. Password: ewilson12345

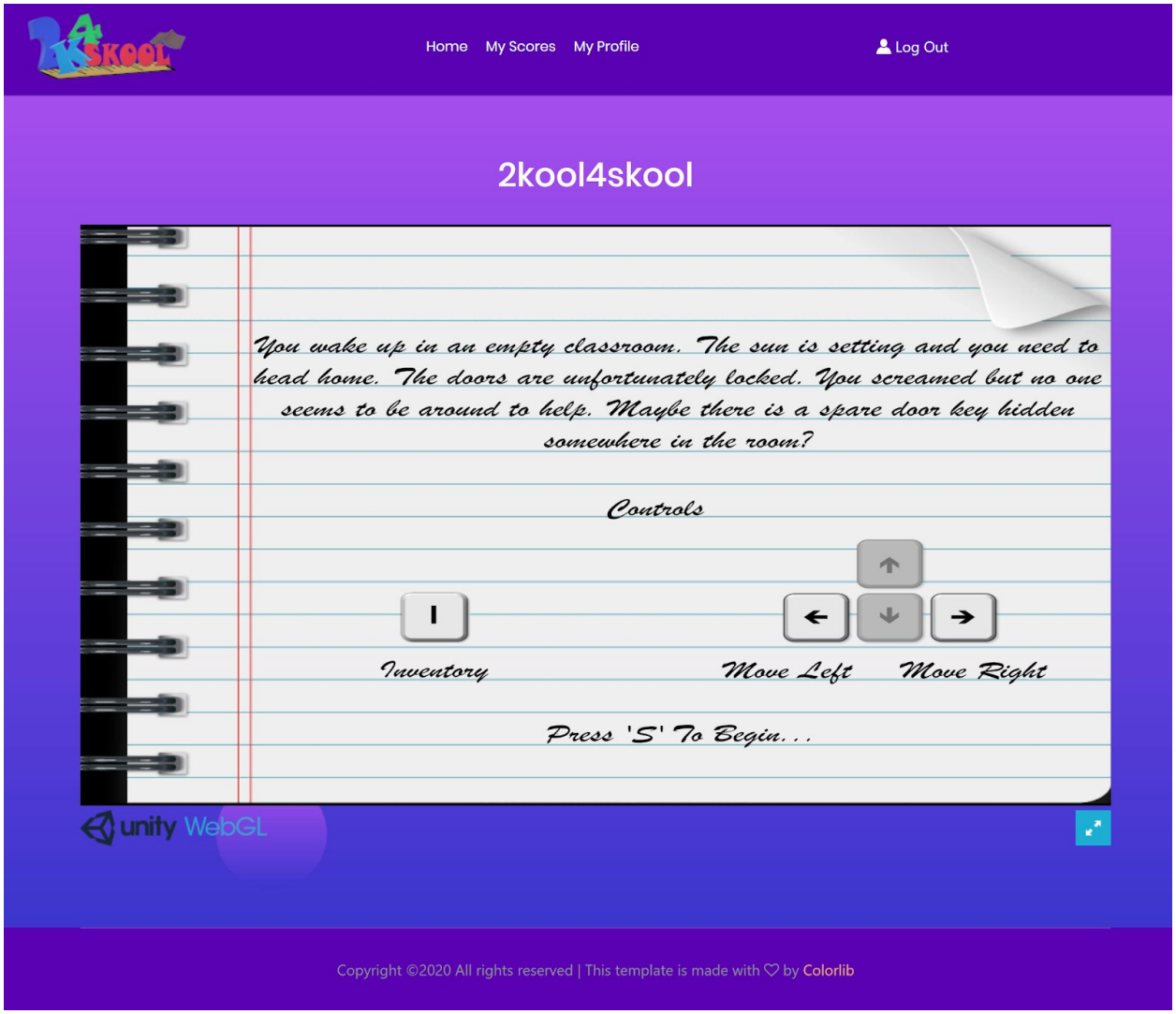
Creating an Account:

On the home page, select the “Create An Account” button. The user will then be prompted to input the required information. After the user submits their information, the server will validate the information. If the input is valid, the server will create the account and redirect the user to the student or teacher home page—depending on the selected user type.

NOTE: Please feel free to create your own student and/or teacher account.

Student Home Page:

The user enters a teacher-provided code and clicks on the “Go” button. The server validates the code. If the code is valid, then the server redirects the user to the game page where the game is loaded.



The user can click on the “Arrows” button to play the game in full screen.

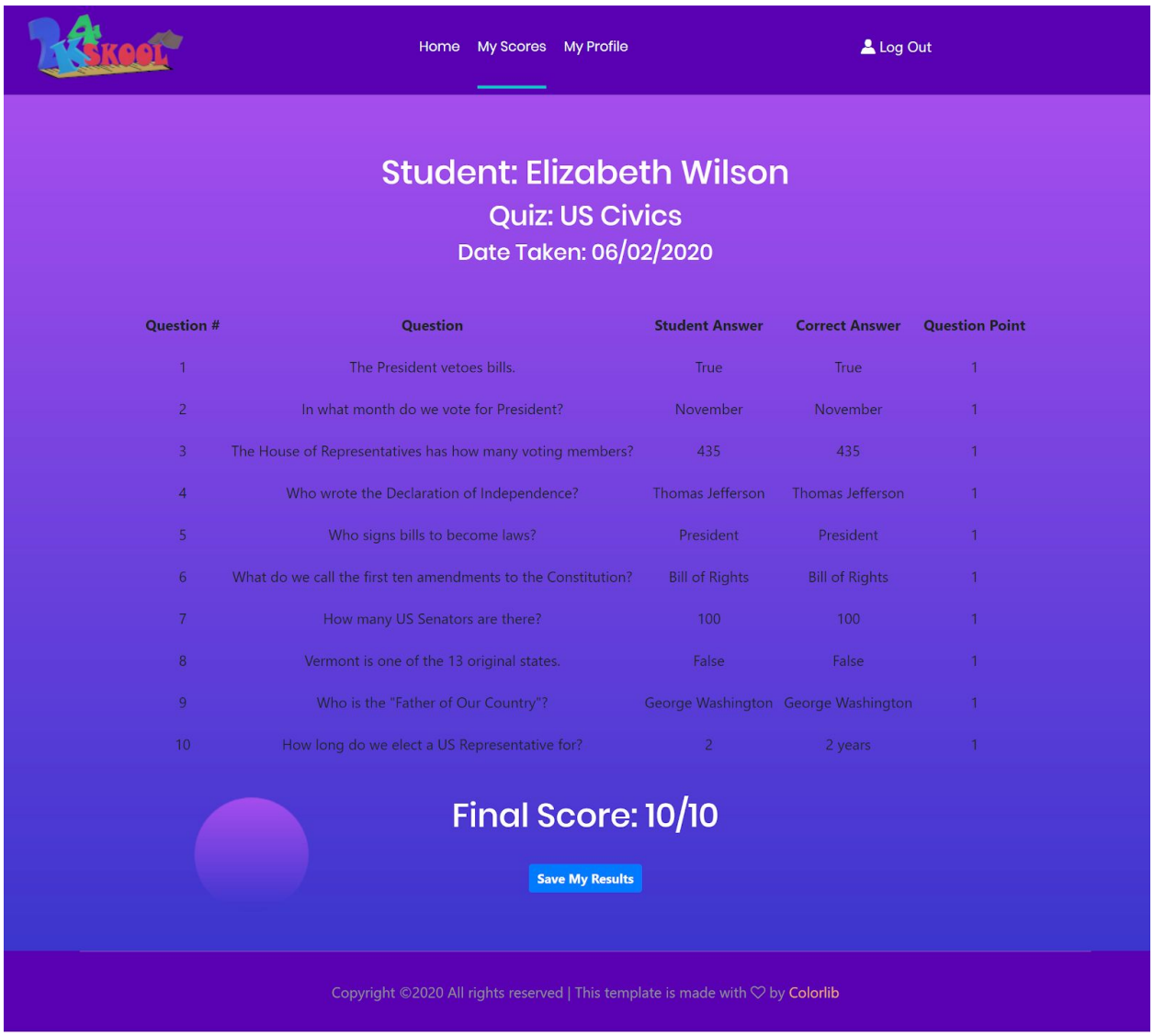
We have provided quiz code **635** (Ancient Egypt) or **2546** (Geography of South America) to be used to play the game while logged in as student Elizabeth Wilson (or any newly created student account). Each quiz code allows a student to play the game once. If the student wishes to replay the game with the same quiz code, a teacher must remove the student’s score from the “Teacher Scoreboard” page.

Student Scoreboard Page:

A table displays a score for quizzes the student has completed. The table can be sorted by quiz name or score using a dropdown filter. Next to each score is a “Results” button that displays the student’s results in detail.

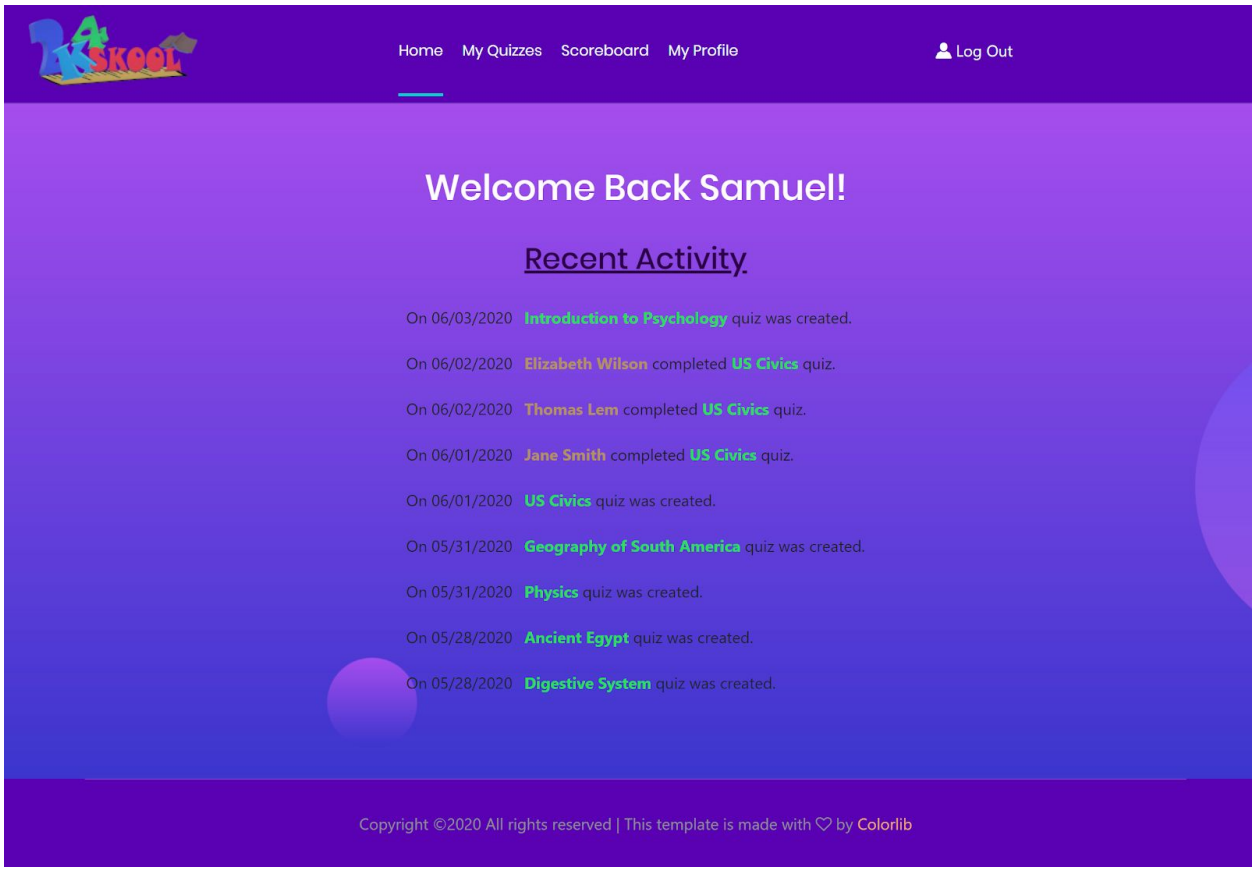
Student Results Page:

The user is presented with details of the student’s quiz.



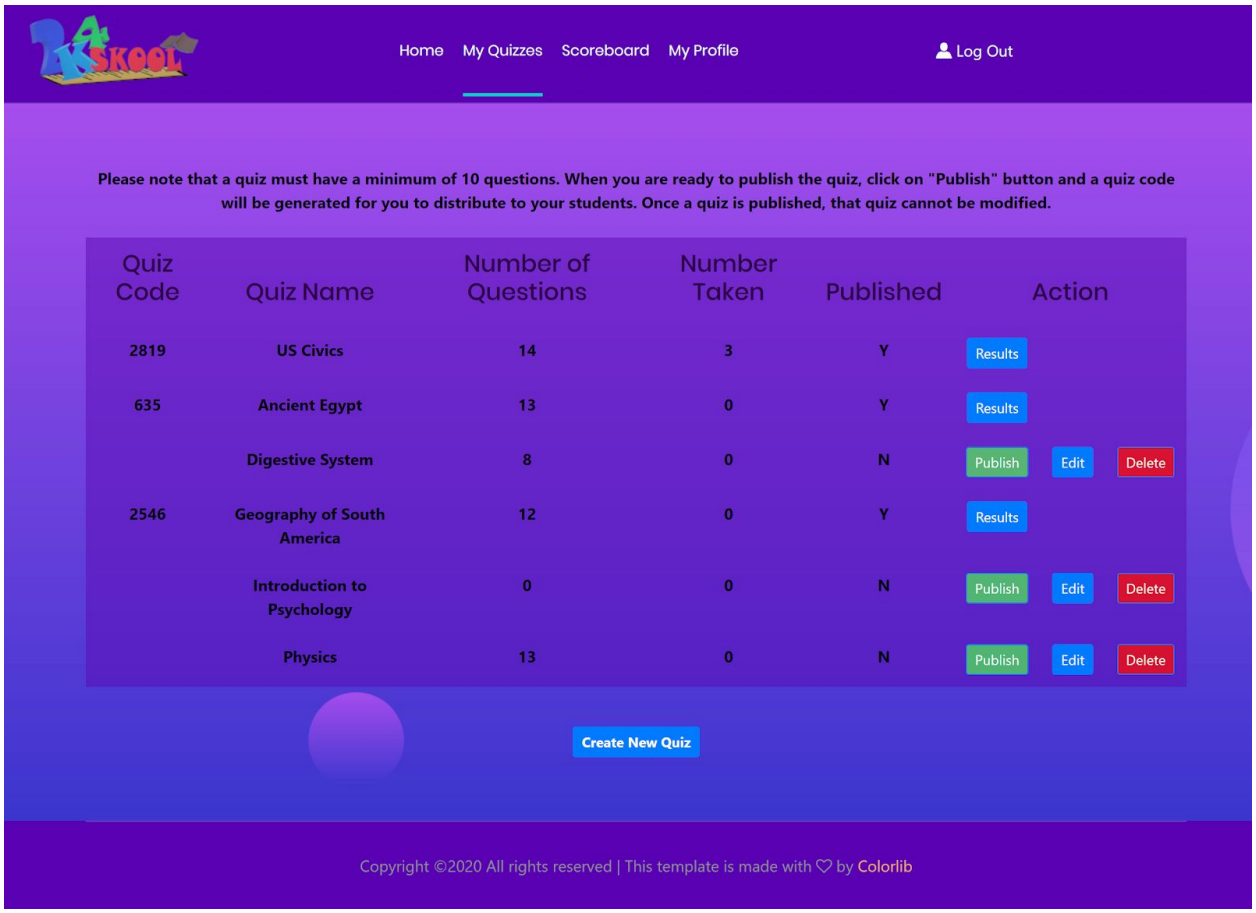
Teacher Home Page:

The web page displays a list of recent activities that are associated with the teacher or the teacher's quizzes.



Teacher Quizzes Page:

The user is presented with a list of quizzes they have created.



After the user publishes a quiz, the user will lose the ability to edit or delete the quiz but will have a button to redirect them to the “Scoreboard” page. On the contrary, the user will be able to publish, edit or delete unpublished quizzes. Only quizzes with a minimum of ten questions can be published. Once the quiz is published, a quiz code will be generated and displayed.

When the user clicks on the “Edit” button on “My Quizzes” page, the user is directed to the “Edit Quiz” page. If the quiz has existing questions, the questions will be displayed here, and the user can either update the quiz name, the current question prompt, or delete the selected question.

After the edit is submitted, the page will reload to display the most updated list of questions in the quiz. The user will also have the option to add a question to the quiz. When adding a new question, the user will type out the question, the question type, the answer and any wrong choice(s). Once the user submits the new question, the page

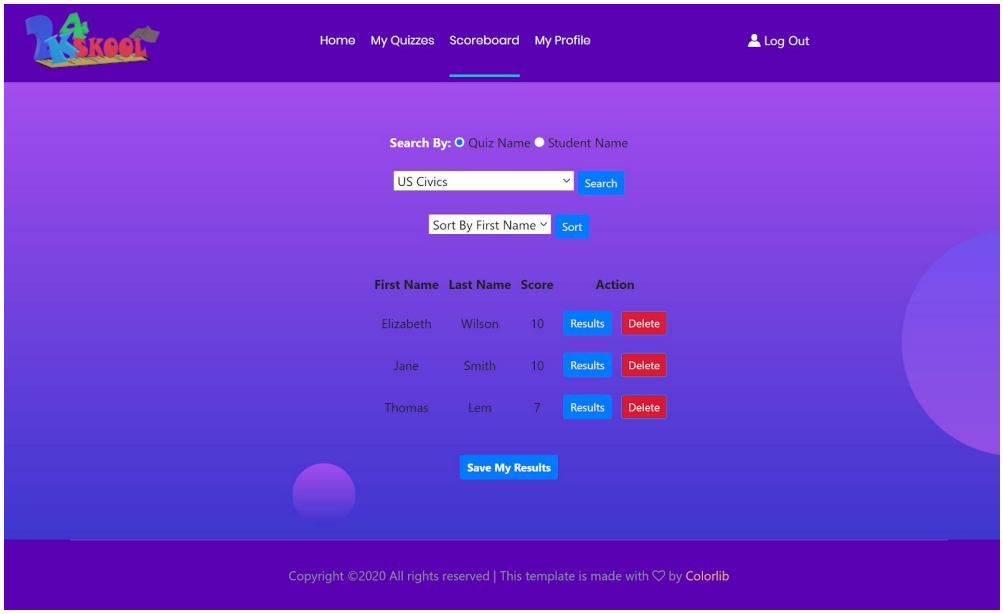
refreshes and the newly added question is now part of the quiz. The user can exit out of the “Edit Quiz” page by clicking the “Done” button, but any unsaved progress will be lost.

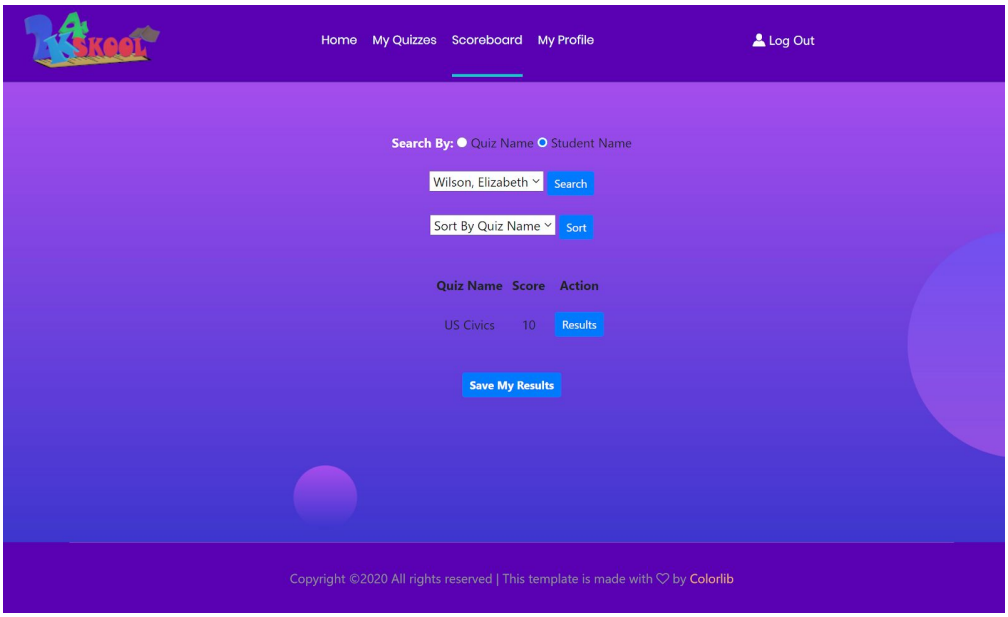
If the user clicks on the “Create New Quiz” button, the user will be directed to a page where they are prompted to enter the new quiz name. After clicking on “Next”, the user will be directed to the “Edit Quiz” page where they can add new questions.

NOTE: Please feel free to publish an existing quiz listed on the table, edit an existing quiz (update quiz name, add/delete/update questions), and/or create a new quiz!

Teacher Scoreboard Page:

The user can select how they want the quiz scores to be displayed.





Next to each score is a “Results” button that displays the student’s results in detail. Also, if the user views scores by quiz name, they can remove a student’s quiz record by clicking on the “Delete” button next to that student.

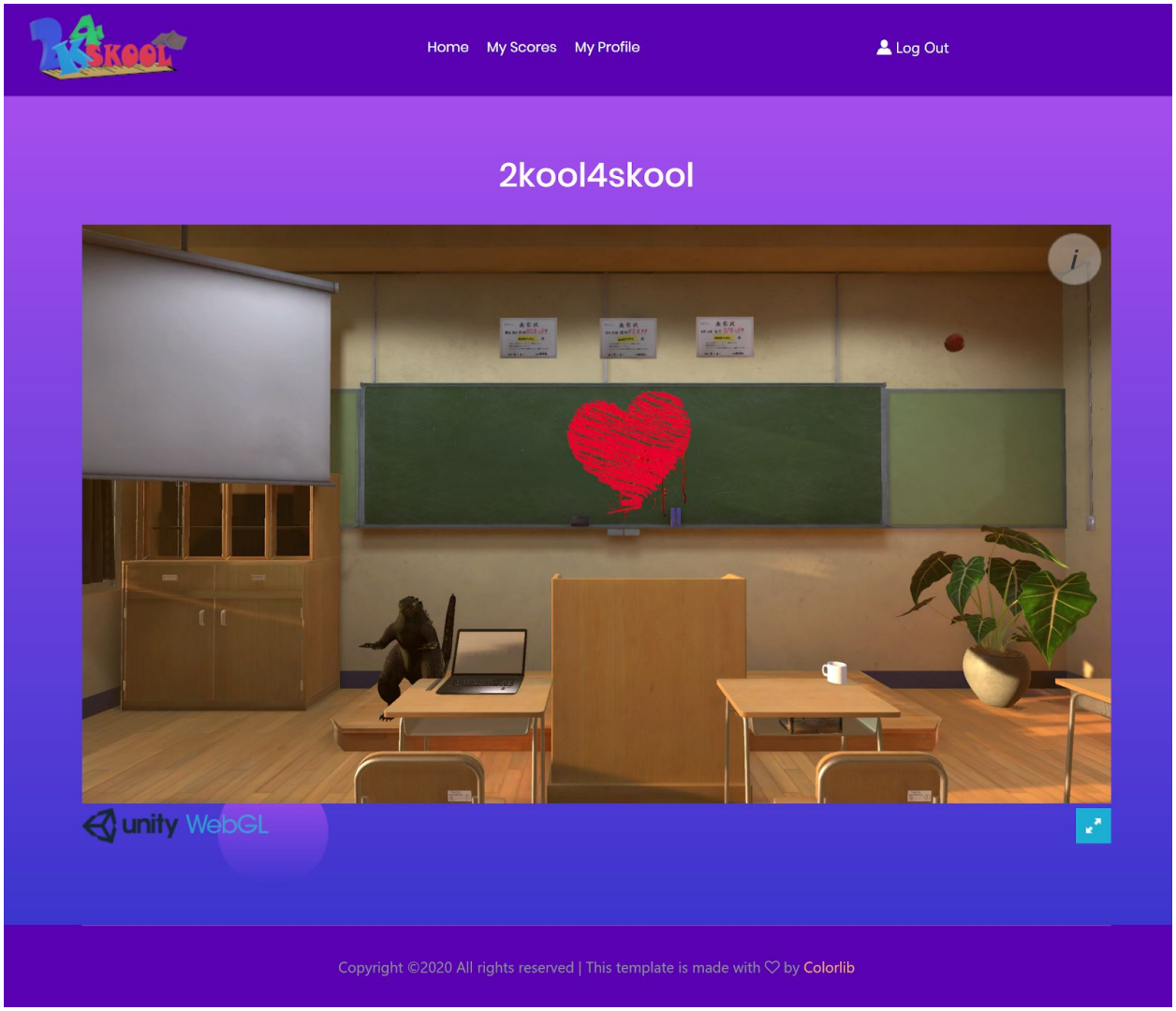
Profile Pages:

For both the student and teacher profile pages, there is a button to change the user’s first and last name, as well as a button to change the user’s password. Upon clicking on either button, a form field is presented that allows the user to update the information in the database.

NOTE: Please feel free to update user information on the student/teacher profile page.

Game Design

Setting:



The player plays as a student that fell asleep during a class lecture. The student wakes up and finds that the teacher and students are not in the classroom. The student discovers that school has ended, they are locked in. Essentially, the player needs to escape the classroom and is able to do that by answering a random set of ten teacher-generated questions in order to interact with objects. Specifically, the game randomizes the order of questions from the quiz and then selects the first ten questions to assign to the ten question objects. These objects are used to help the player escape the room.

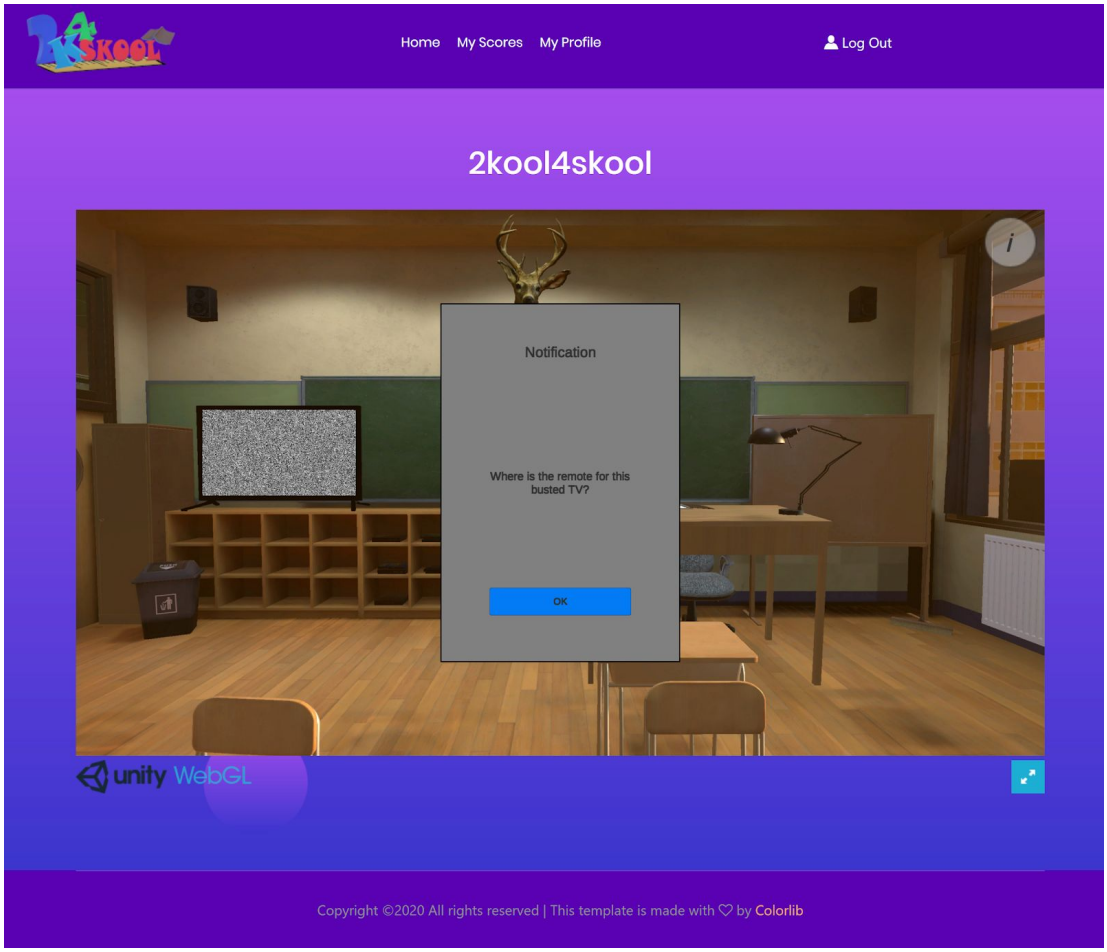
As the player progresses through the game and interacts with the many different objects, for each object they interact with, they will be prompted to answer a question (True/False,

Multiple Choice, or Short Answer). The player has to answer and can only answer the question once. After the question is answered, the player is able to obtain an inventory item or activate an event. The player can access their inventory where they can view acquired (inventory) items, questions remaining and elapsed time.

No questions are repeated, but all questions must be answered to finish the game. The game is not timed. The objective of the game is to have the student answer as many questions correct and escape the classroom.

Clickable Objects:

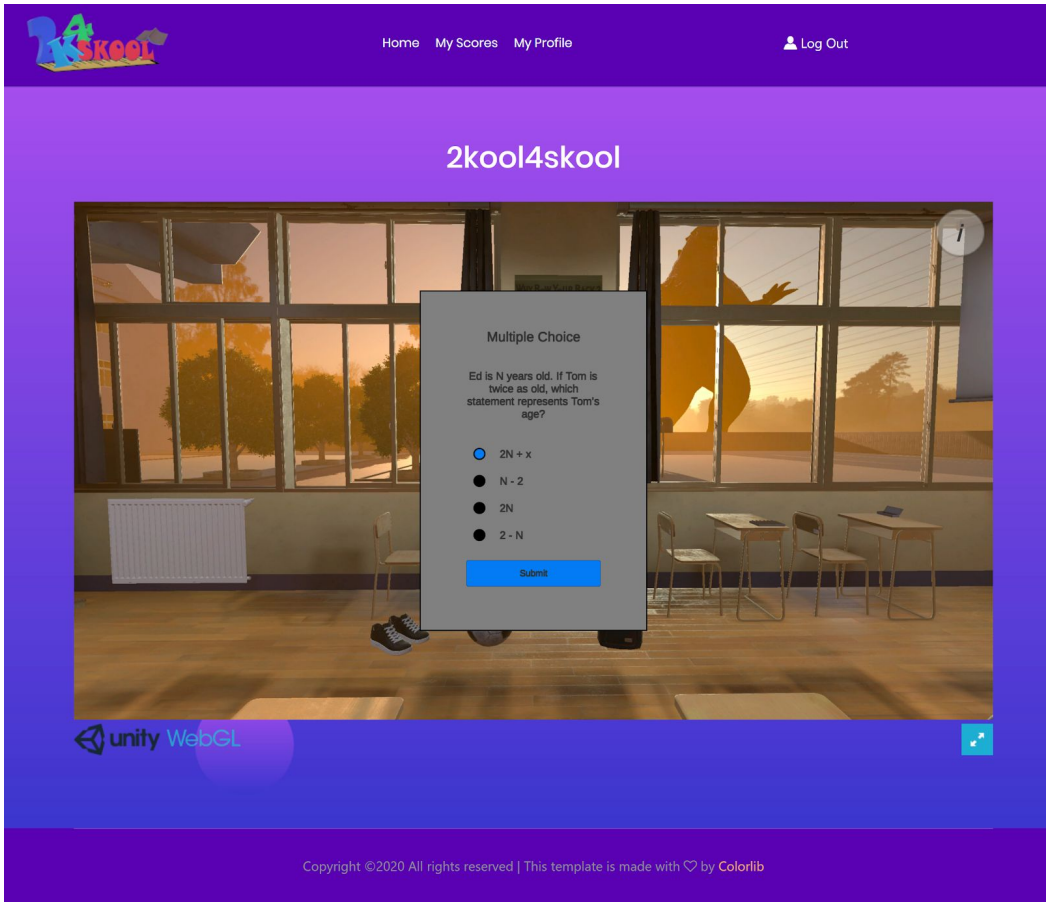
Static Objects:

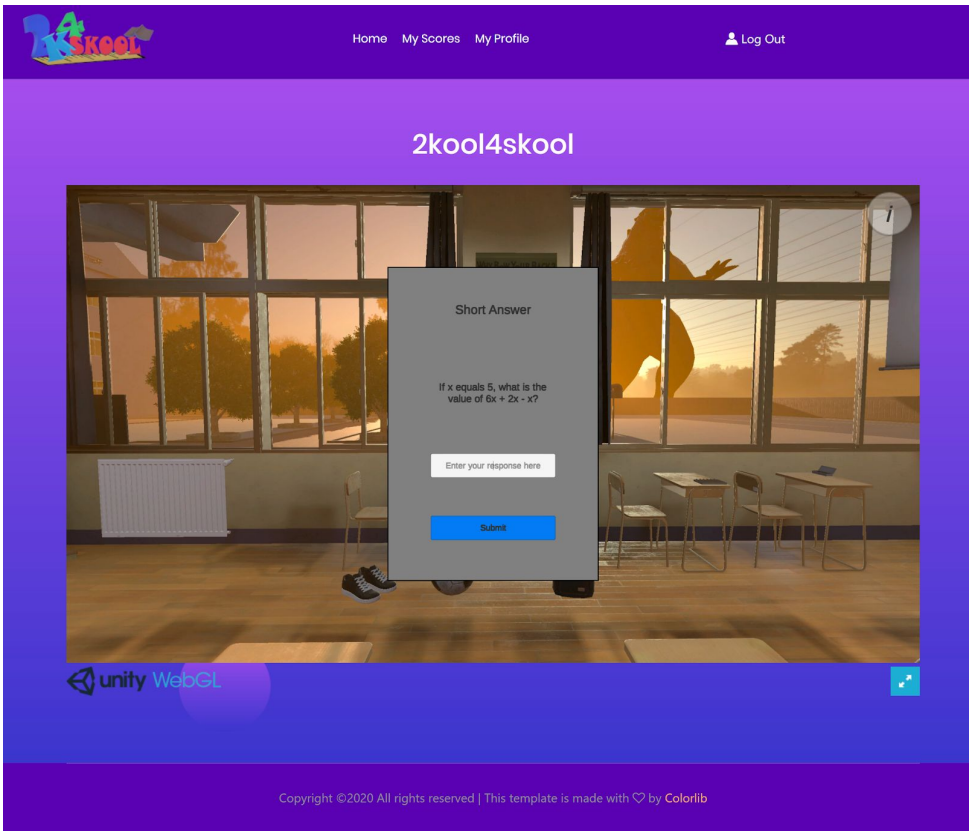


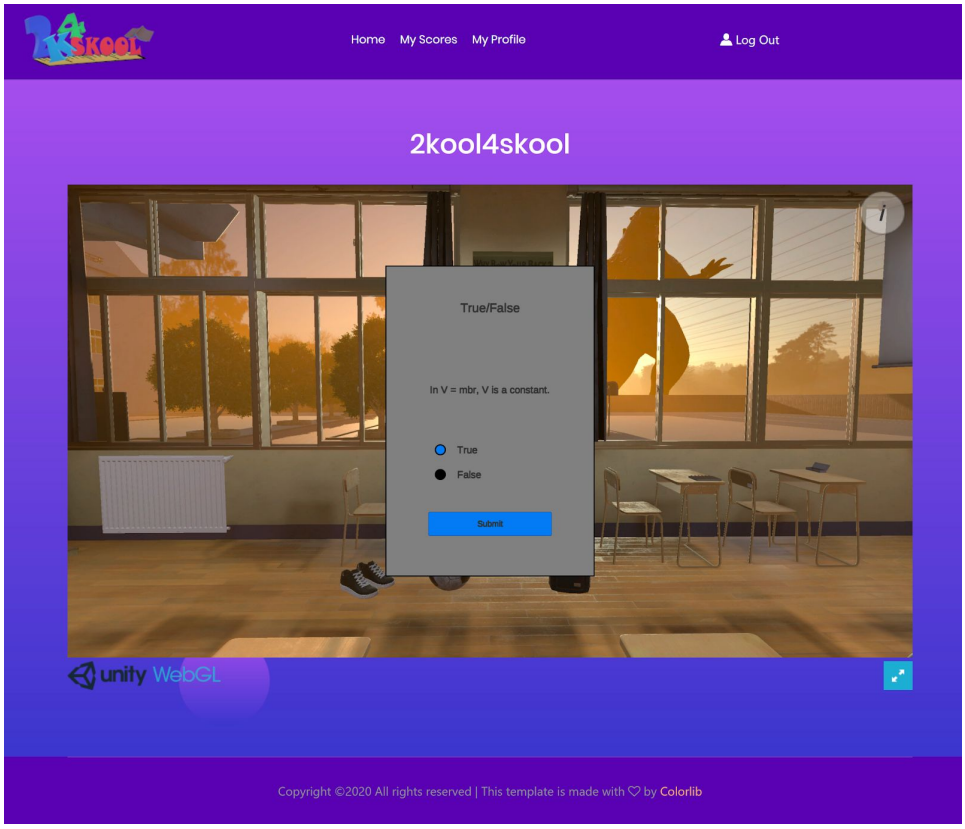
A static object in the game is defined as an object that does not have a question attached to it. Static objects are solely created for game enhancement. The classroom has the following static objects:

1. Godzilla
2. Blackboard
3. Fire Extinguisher
4. Fire Alarm
5. Floor Plant
6. Laptop
7. Lockbox
8. Stack of Paper
9. Book
10. Umbrella
11. Deer Head
12. Table Lamp
13. Wall Speaker
14. Wall Poster
15. Water Bottle
16. Mug
17. Certificate
18. Blue Bag
19. Eraser
20. Television
21. Pencil Case
22. Soccer Ball
23. Half Window 1
24. Half Window 2

Question Objects:



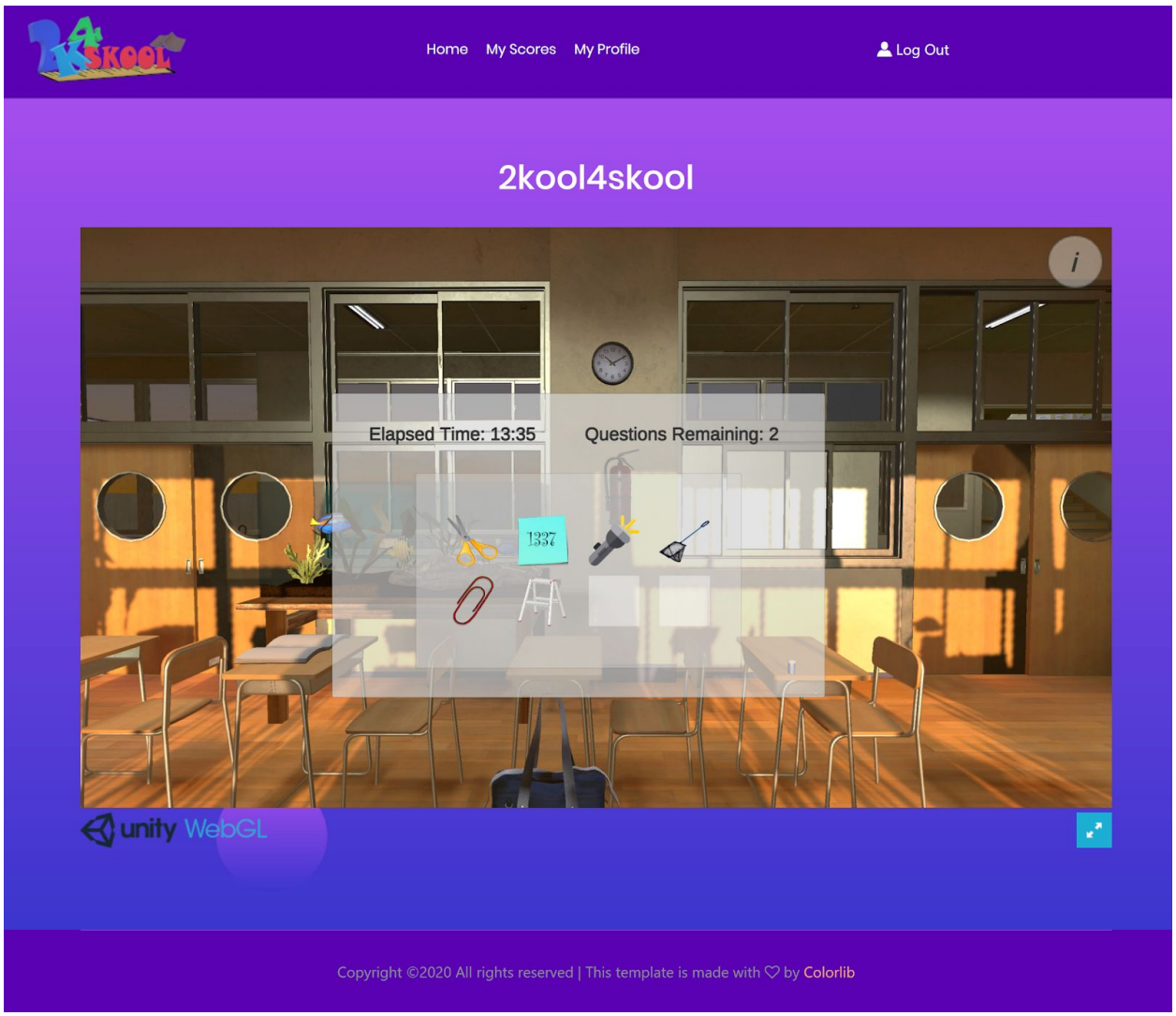

Home
My Scores
My Profile
Log Out
2kool4skool
Short Answer
If x equals 5, what is the value of 6x + 2x - x?
Enter your response here
Submit
unity WebGL
Copyright ©2020 All rights reserved | This template is made with ♡ by Colorlib


Home
My Scores
My Profile
Log Out
2kool4skool
True/False
In V = mbr, V is a constant.
True
False
Submit
unity WebGL
Copyright ©2020 All rights reserved | This template is made with ♡ by Colorlib

A question object in the game is defined as an object that has a question attached to it. The classroom has the following question objects:

1. Heater
2. Aquarium
3. Wall Clock
4. Shoes
5. Backpack
6. Teacher's Desk
7. Janitor's Closet
8. Trash Can
9. Door
10. Bookshelf

Inventory Items:



An inventory item in the game is defined as an item that is obtained by the player as they interact with question objects. The classroom has the following inventory items:

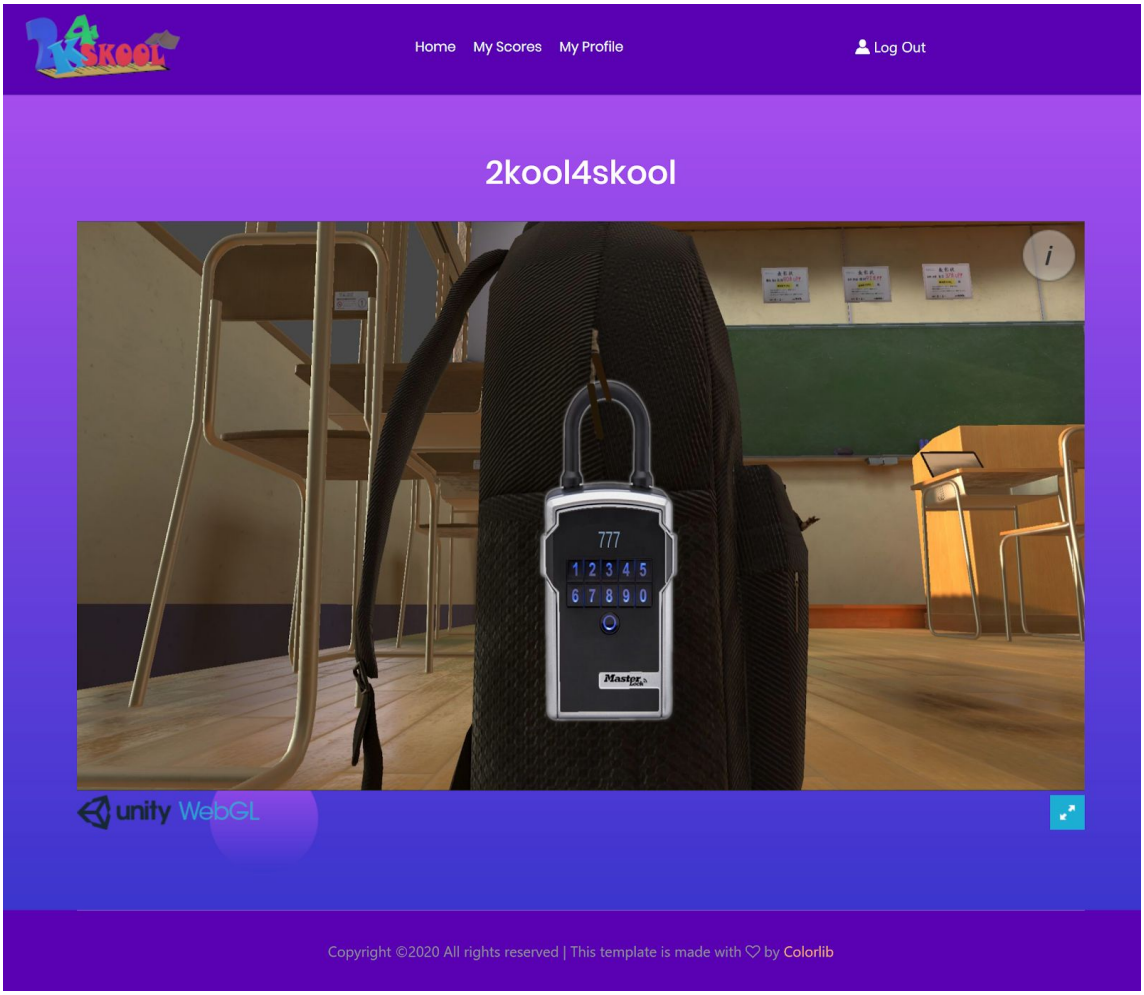
1. Scissors
2. Very Portable
3. Step Ladder
4. Paper Clip
5. Note with Pin Code
6. Flashlight without Batteries
7. AA Batteries
8. Key

Walkthrough:

Both the Flashlight Without Batteries and the AA Batteries have to be in player inventory (in no order) for the player to find the Key. Obtaining the flashlight and the batteries each consists of the following steps listed below.

Getting Flashlight Without Batteries:

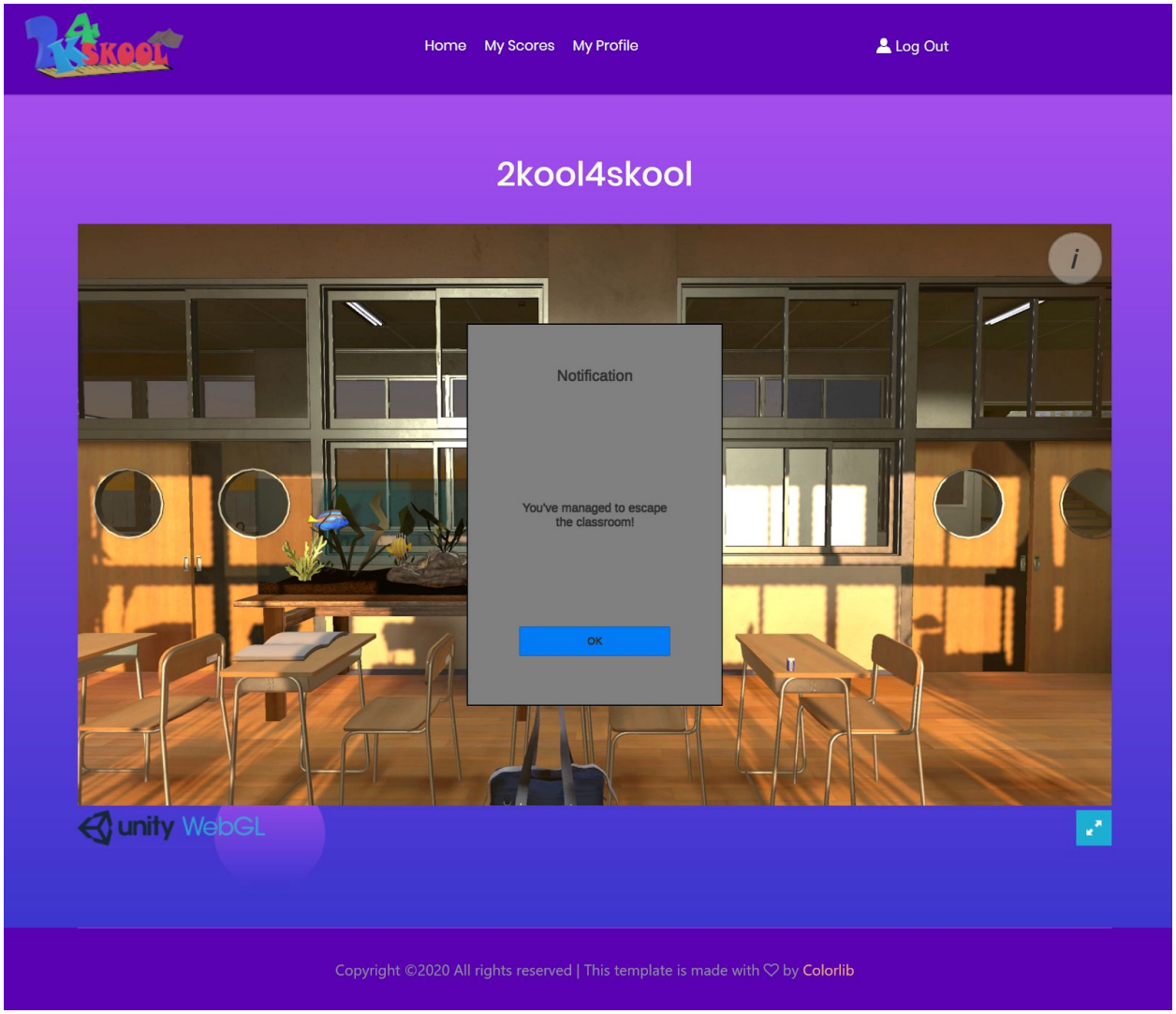
1. The player must interact with the Bookshelf. When this interaction occurs, the player is prompted with a quiz question and upon completion they will receive the item: Scissors.
2. While the player has the Scissors, interacting with the Shoes will result in the player being prompted a quiz question and upon completion the interaction will trigger an event and a new interaction with the Teacher's Desk will be possible. Prior interactions with the Shoes results in a hint suggesting the student return to the Shoe.
3. Once the player has triggered the Shoes event, the interaction with the Teacher's Desk will result in a prompt for a quiz question and upon completion, the player will receive the item: Note With Pin Code. Prior interactions with the Teacher's Desk results in a hint suggesting the player returns to the Teacher's Desk.
4. While the player has the Note With Pin Code, interacting with the Backpack will result in the player being prompted with a quiz question. After answering the question, a puzzle will appear that requires the player to enter a correct pin code. Upon completion, the player will receive the item: Flashlight Without Batteries. Prior interactions with the Backpack result in a hint suggesting the player return to the Backpack.



Getting the AA Batteries:

1. The player must interact with the Heater. When this interaction occurs, the player is prompted with a quiz question and upon completion the player will receive the item: Aquarium Net.
2. While the player has the Aquarium Net, interacting with the Aquarium will result in the player being prompted a quiz question and upon completion the player will receive the item: Paperclip. Prior interactions with the Aquarium result in a hint suggesting the player return to the Aquarium.
3. While the player has the Paperclip, interacting with the Janitor's Closet will result in the player being prompted a quiz question and upon completion the player will receive the item: Very Portable Step Ladder. Prior interactions with the Janitor's Closet result in a hint suggesting the player returns to the Janitor's Closet.
4. While the player has the Very Portable Step Ladder, interacting with the Wall Clock will result in the player being prompted a quiz question and upon completion the player will receive the item: AA Batteries. Prior interactions with the Wall Clock result in a hint suggesting the player return to the Wall Clock.

Getting Out:



1. While the player has both the Flashlight Without Batteries and the AA Batteries, interacting with Trash Bin will result in the player being prompted a quiz question and upon completion the player will receive the item: Key. Prior interactions with the Trash Bin result in a hint suggesting the player return to the Trash Bin.
2. While the player has the Key, interacting with the Door will result in a prompt for a quiz question and upon completion the player will successfully escape the room and the game will end. Prior interactions with the Door will result in a hint suggesting the player return to the Door.

After the game ends, the player is redirected to their quiz results page.

Team Member Contributions

Nelson Chan

For the website:

- Set up the front-end (i.e., supplied HTML with CSS templates)
- Set up the back-end (i.e., built and ran an Heroku application)
- Developed the navigation bar
- Created the home page (login and create account with input validation and error handling)
- Created the game page with input validation and error handling
- Created sessions (logging in, logging out, ensuring user is signed in to access their respective pages [e.g., student cannot access teacher's edit quiz web page and can only access student profile, home, and scoreboard pages])

For the game:

- Wrote method in game script to receive data from web application
- Wrote method in game script to send data from game to to server
- Wrote method that allows objects in the scene to be clicked
- Wrote method that recognizes mouse clicks and keyboard clicks on particular objects
- Created the notification and question panels
- Wrote method to redirect user to quiz results page after user finishes the game

Thomas Lem

For the website:

- Set up GitHub repository for team collaboration
- Developed the front-end and back-end for the teacher homepage, teacher scoreboard, and teacher profile pages
- Worked on localStorage to maintain values across web pages
- Developed the front-end and back-end for the student scoreboard, student results, and student profile pages
- Developed the question randomizer

For the game:

- Set up Unity cloud account for team collaboration
- Developed the user inventory
- Developed the game scenario and prompts for question objects and static objects
- Developed method to "connect the dots" (i.e., question object prompt displayed is dependent on items in player's inventory)
- Provided research and methods to communicate between Unity and the web server

Christine Shi

For the website:

- Developed front-end and back-end for the teacher quiz and edit quiz pages
- Created website logo and carousel images
- Set up the database on Google Cloud SQL

For the game:

- Developed the intro, chalkboard and TV animations
- Developed the backpack lock puzzle
- Created introduction, wall background and inventory item images

Team

- Participated in weekly meetings for a total of 12+ hours via Google Hangouts Meet
- Formatted website
- Created quizzes and user accounts
- Tested the website and game
- Completed the project plan, midpoint check, final report, poster in Google Docs

Closing Remarks

Completing this project has made us more well-rounded developers. We refreshed our web and database development skills, hardened our debugging and critical thinking skills, and picked up on game development along the way. Realistically, the journey was treacherous. There were times where things did not go as planned. For one, we underestimated the time we thought it would take to complete a task. On numerous occasions, we encountered issues and realized there is more than one way to solve a problem. For example, after implementing four different methods of playing on WebGL a MP4 white noise video on the TV, we came to a conclusion that due to constraint with

Unity's VideoPlayer component on WebGL, it was necessary to switch from video to animation to keep the feature.

We are proud of the product we put forth. We hope the end-user enjoys using our web application as much as we enjoyed developing it and sees the quality of the product reflect the amount of work we put in. Any questions can be directed to channe@oregonstate.edu, lemt@oregonstate.edu, and shini@oregonstate.edu. The source files can be accessed at https://github.com/lemtho/escape_room_quiz.