



College of Science, Technology & Applied Arts of Trinidad & Tobago

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

INDIVIDUAL ASSIGNMENT COVER PAGE

ASSIGNMENT TITLE: TRINI TING

ASSIGNMENT SUBTITLE: FINAL PROJECT

COURSE CODE: JTEC 245

COURSE TITLE: INTRODUCTION TO SCRIPTING LANGUAGES

CRN: 19087 **SEMESTER:** 202110

STUDENT NAME: VIKASH DOOKIE

STUDENT ID: 00062486

PROGRAMME: BSc – IT NETWORKING

DATE OF SUBMISSION: 08 – 01 – 2021

LECTURER'S NAME: ALICIA DENNIS-NAGEE

Project Summary

Purpose

The Ministry of Culture, Community and Youth in Trinidad and Tobago has hired me to build a web application to promote the culture of the country in the format of a trivia game. The site is to be called TriniTing. Players are to guess answers to questions about the Trinbago culture, a hint must be available to help them figure out the answer. A player must also be able to leave the game at any time they wish to do so.

Audience

The target audience for this site is both tourists and locals of the country. The game can test players on how much they know about the country and discover new things about it.

Functionality

All functionality requested for the game is fully working.

Personal Challenges

Some of the challenges I personally faced was achieving the perfect layout while making the site responsive and mobile friendly, getting questions to put on the site, and time management.

There was a leaderboard feature I wanted to implement by using indexedDB as the database, however I ran out of time and couldn't do it.

Recommendations for future development

I recommend using a database to store questions, and player's high scores, and make the layout more mobile friendly.

Tools Used

The tools used to develop this site were:

- **Sublime** – Text editor for coding the site.
- **Figma** – An interface design tool used to create SVG images for the logo and hangman.
- **RX 7 Audio Editor** – Used to cut snippets off of songs.

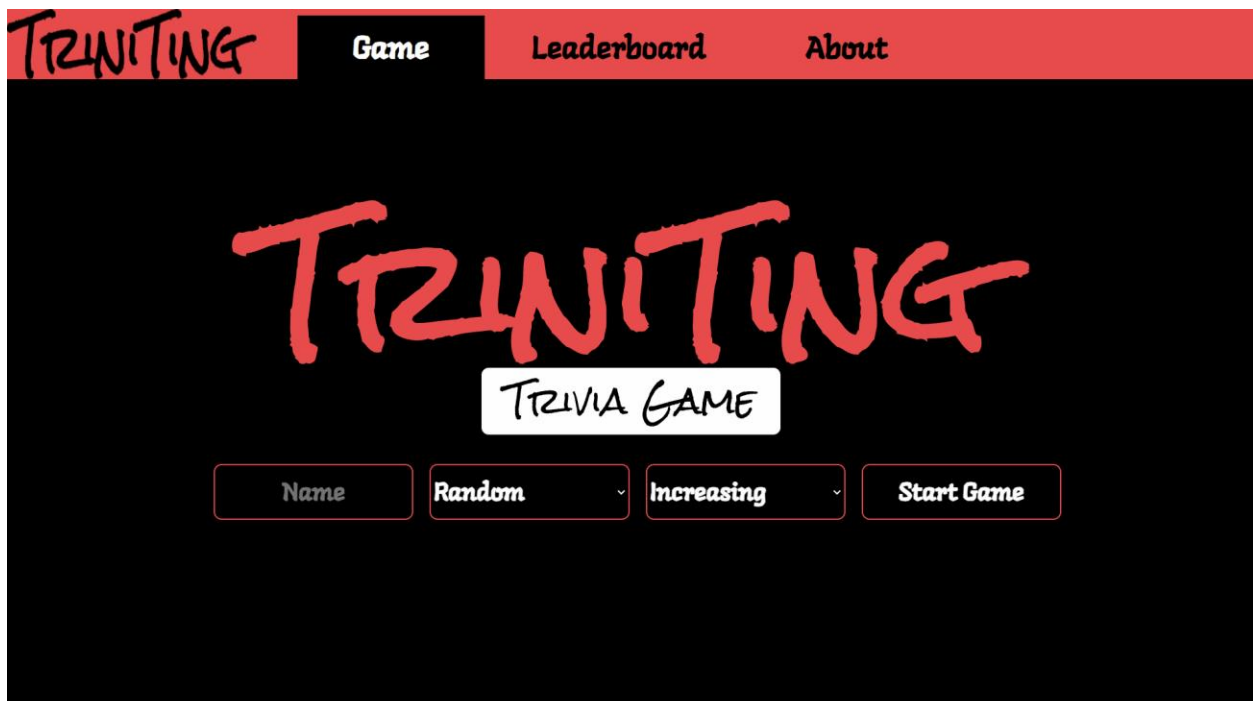
A Statement of Functionality

Level 1: Select Category and Start Game

Full functionality was achieved.

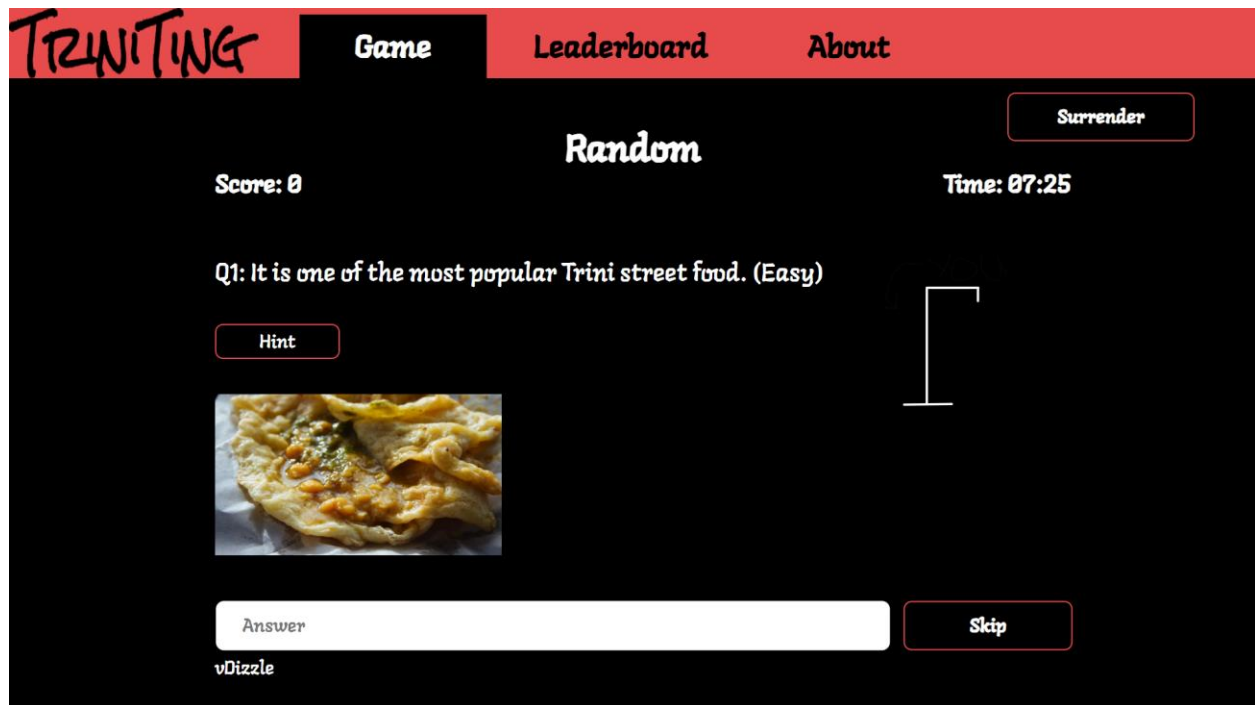
Upon opening the site an animation is played then the user is provided with a form to enter their name, select category, select difficulty and a submit button which starts the game. Once all fields are complete, the user can start the game.

```
//What to do when a user enters name and click start game
startForm.onsubmit = function() {
  player = new Player(startForm.elements[0].value);
  startForm.elements[0].value = "";
  category = startForm.elements[1].value;
  difficulty = startForm.elements[2].value;
  labelSwitcher();
  nameSpan.innerHTML = player.name;
  categorySpan.innerHTML = categoryLabel;
  startPanel.style.display = "none";
  gamePanel.style.display = "initial";
  startGame();
}
```



The screenshot shows the TRINITING Trivia Game interface. At the top, there is a red navigation bar with the word "TRINITING" in a stylized font on the left, and three menu items: "Game", "Leaderboard", and "About". Below the navigation bar, the word "TRINITING" is displayed in a large, red, hand-drawn font. Underneath it, the words "TRIVIA GAME" are written in a smaller, black, hand-drawn font. At the bottom, there is a form with four input fields: "Name", "Random", "Increasing", and "Start Game". The "Random" and "Increasing" fields have dropdown arrows, indicating they are select menus. The "Start Game" field is a button.

Once the game is loaded, the user sees a heading of the selected category, their score, the timer, a surrender button, the question, a button that displays a hint at the cost of 3 points, an attachment if there is one (audio / image / video), a hangman (shows how many chances are left), the answer input field, a skip / submit button (depends on if the answer field is empty or not) and their name.



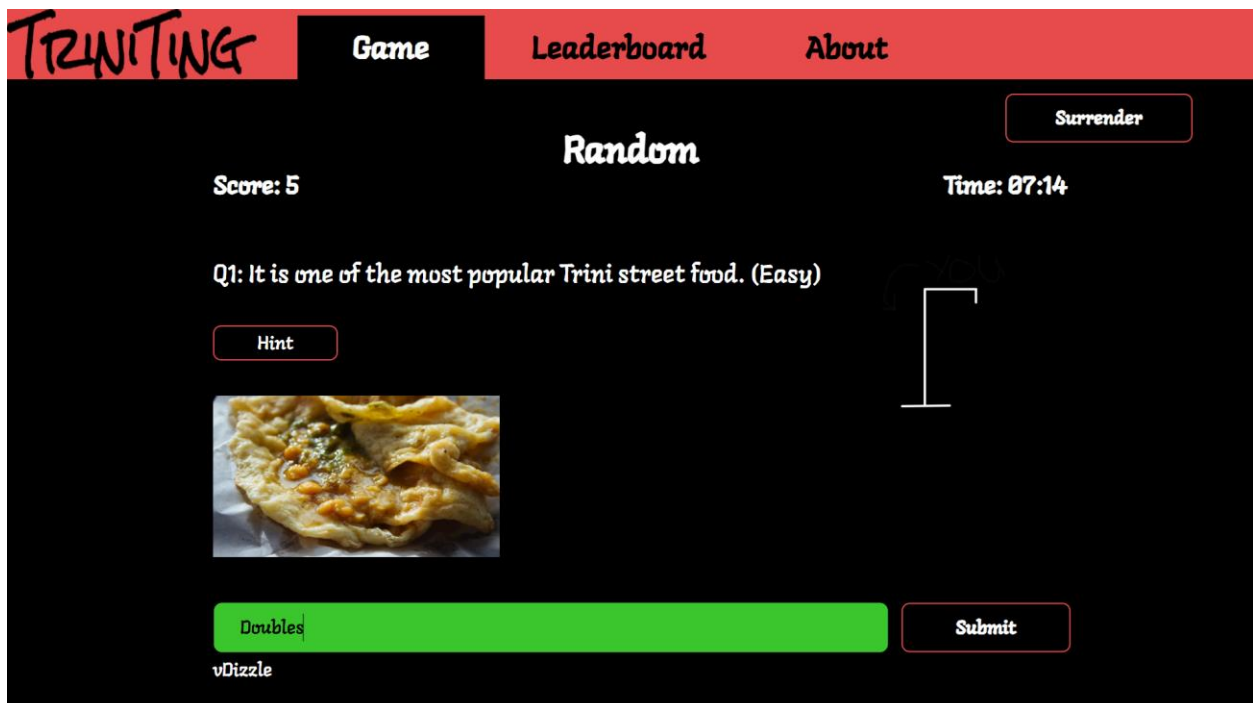
Level 2: Process the Submission

Full functionality was achieved.

Once the user submits their guess the app checks to see if the answer is right or wrong. If it is right the question points is added to the user's score, a sound is played and the answer input field blinks green.

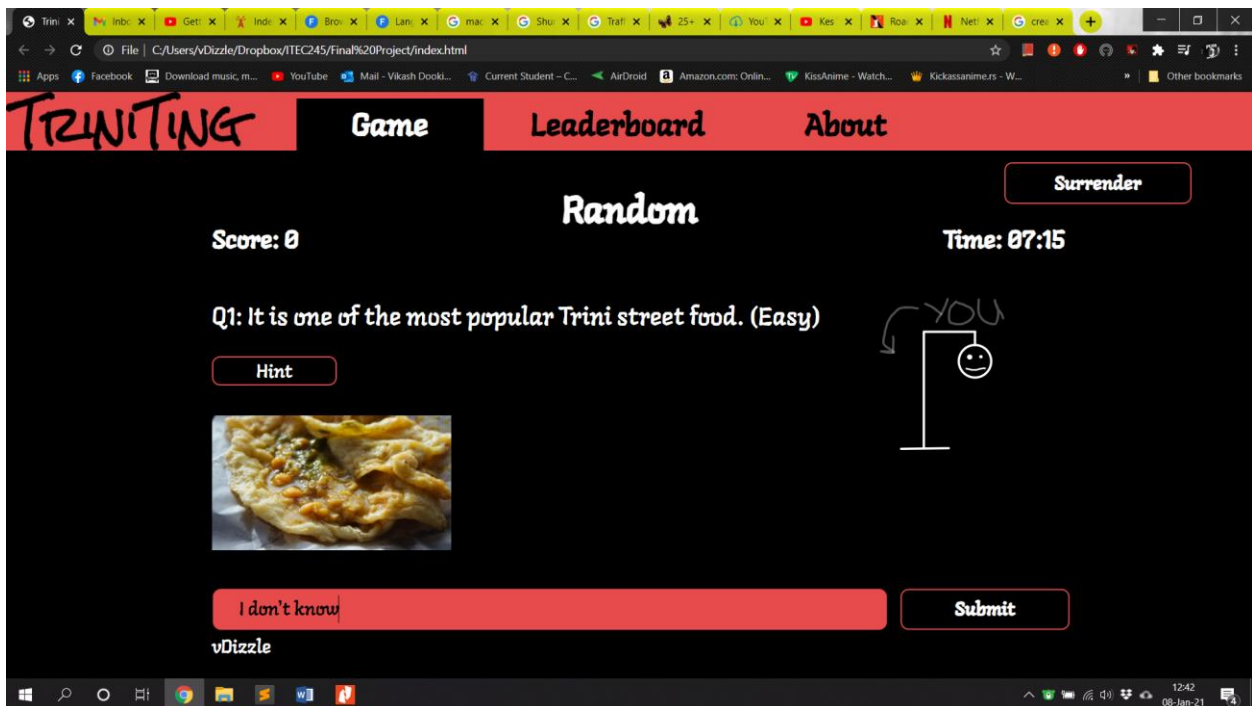
```
//when answer is submitted check if it is right or wrong and add score if right
answerForm.onSubmit = function() {
  let answer = answerInput.value.toLowerCase();
  let correct = false;
  for(let i = 0; i < questions[currentQuestion].a.length; i++) {
    if(answer == questions[currentQuestion].a[i].toLowerCase()) {
      answerInput.style.backgroundColor = "#3BC62E";
      player.score += questions[currentQuestion].points;
      scoreSpan.innerHTML = "Score: " + player.score;
      if(lives < 3) {
        lives = 5;
        livesFunc();
      }
      answerTrue.play();
      setTimeout(nextQuestion, 500);

      break;
    }
  }
  else if((i + 1) == questions[currentQuestion].a.length) {
    answerFalse.play();
    answerInput.style.backgroundColor = "rgba(231, 76, 76, 100)";
    livesFunc();
  }
}
}
```

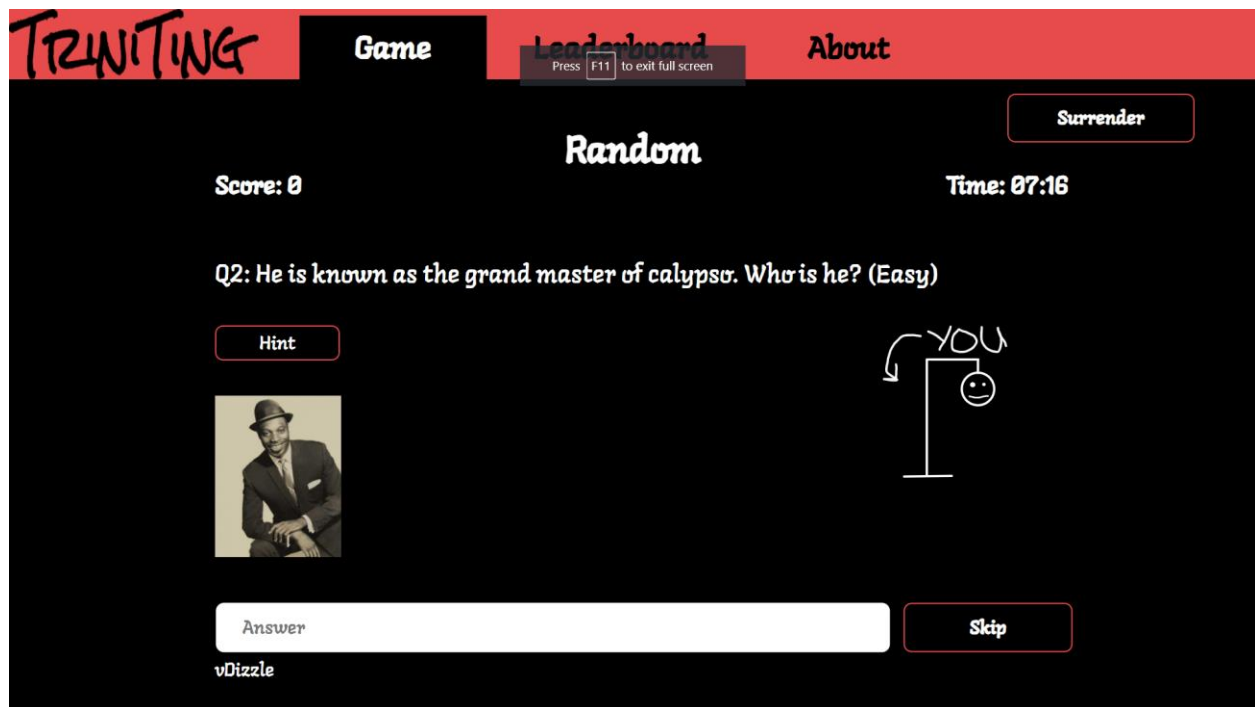


If the answer is wrong the head is drawn on the hangman, a sound is played, and the answer input field blinks red. The same thing happens if the question is skipped.

```
//Used to add or remove hangman body parts
function livesFunc() {
  --lives;
  let aliveHead = document.querySelectorAll('.aliveHead');
  let torso = document.querySelectorAll('.torso');
  let deadHead = document.querySelectorAll('.deadHead');
  let legs = document.querySelectorAll('.legs');
  switch(lives) {
    case 4:
      for(let i = 0; i < aliveHead.length; i++) {
        aliveHead[i].style.display = "none";
      }
      for(let i = 0; i < deadHead.length; i++) {
        deadHead[i].style.display = "none";
      }
      for(let i = 0; i < torso.length; i++) {
        torso[i].style.display = "none";
      }
      for(let i = 0; i < legs.length; i++) {
        legs[i].style.display = "none";
      }
      lives = 3;
      break;
    case 2:
      for(let i = 0; i < aliveHead.length; i++) {
        aliveHead[i].style.display = "initial";
      }
      setTimeout(nextQuestion, 500);
      break;
    case 1:
      for(let i = 0; i < aliveHead.length; i++) {
        aliveHead[i].style.display = "none";
      }
      for(let i = 0; i < deadHead.length; i++) {
        deadHead[i].style.display = "initial";
      }
      for(let i = 0; i < torso.length; i++) {
        torso[i].style.display = "initial";
      }
      setTimeout(nextQuestion, 500);
      break;
    case 0:
      for(let i = 0; i < legs.length; i++) {
        legs[i].style.display = "initial";
      }
      clearInterval(timerInterval);
      clearInterval(inputInterval);
      gameLose.play();
      setTimeout(function() {myAlert(("You Lost " + player.name + "!"), ("You Score: " + player.score + "<p>Time Left: " + timeSpan.innerHTML.split(" ")[1]));}, 500);
      break;
  }
}
```

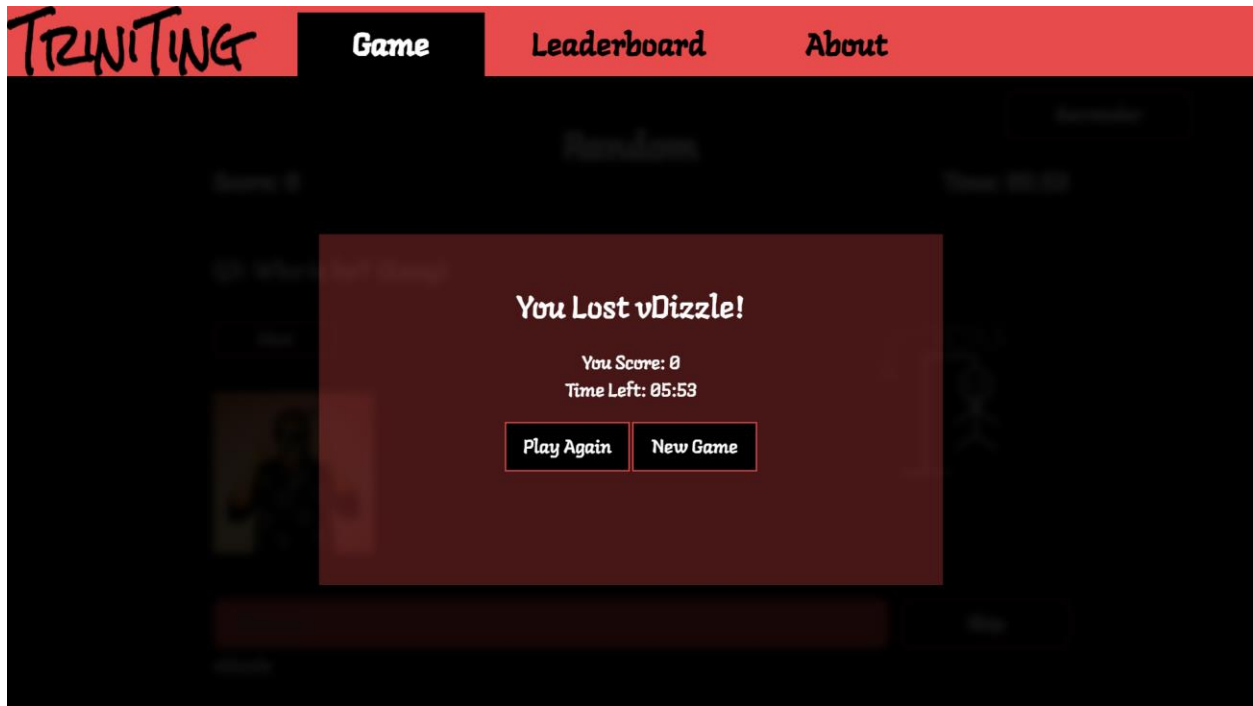


If there is another question it will be displayed next.

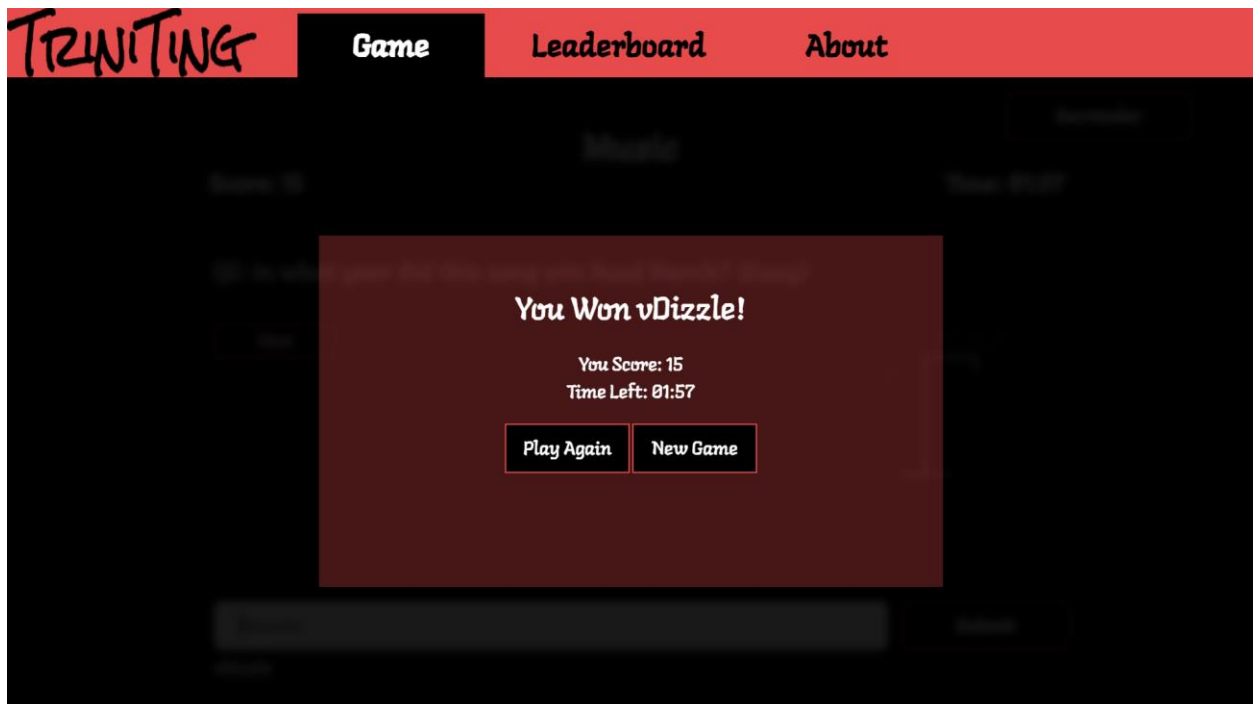


If the user guesses wrong again the torso is drawn, and a third time in a row the legs are drawn and the player loses. When the player loses a message is displayed to the screen.

```
//Used to display a custom alert
function myAlert(head, msg, button1, button2) {
  modalHead.innerHTML = head;
  modalMsg.innerHTML = msg;
  modalDiv.style.display = "initial";
}
```

If the player doesn't lose and there are no more questions, the player wins. A message is displayed to the screen showing the score and time remaining.



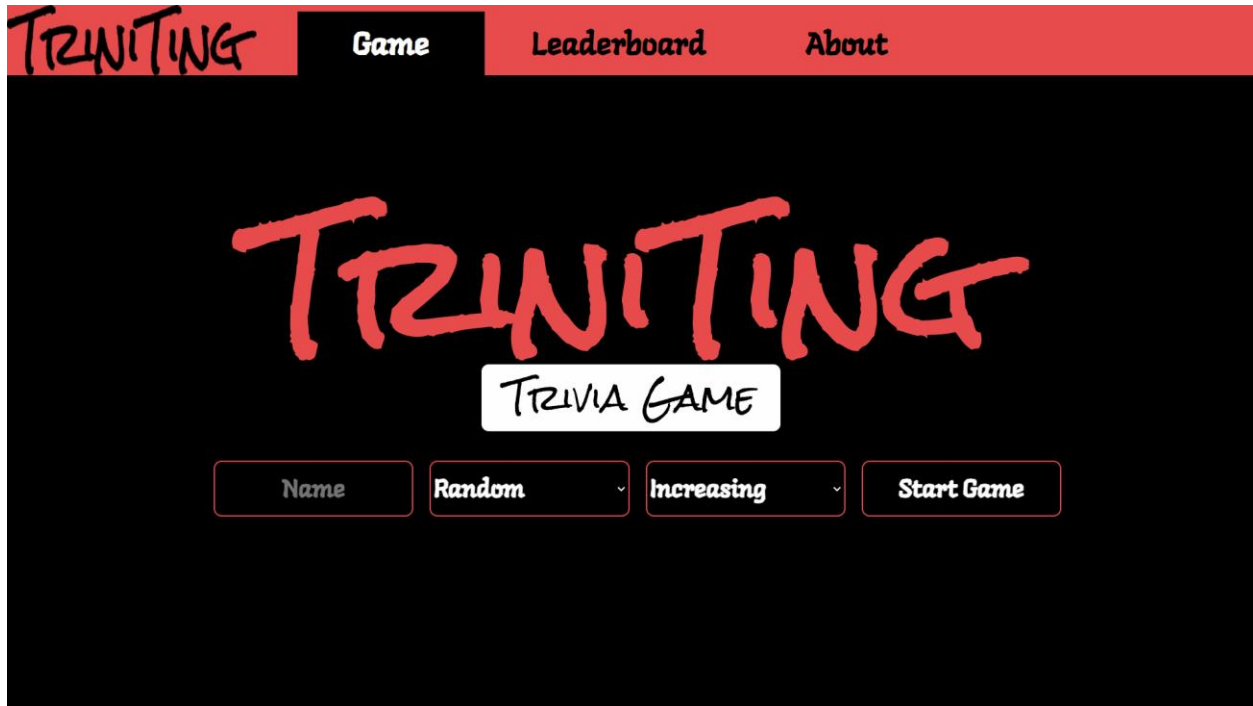
Level 3: Make it Pretty

Full functionality was achieved.

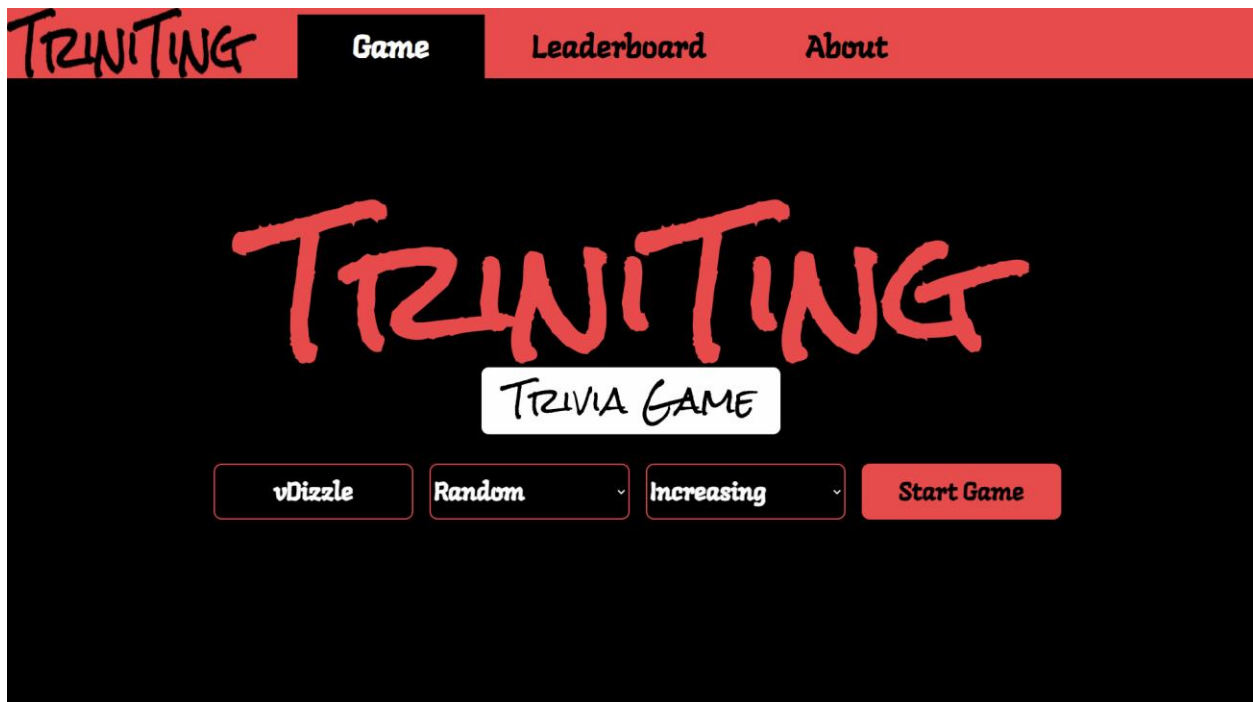
External CSS was used to style the web application. The style would also be manipulated using JavaScript. Since this is a site for Trinidad and Tobago the national colours were used to style it. An animation was used on the landing page (home page) by animating a SVG element, and transitions were used on the nav buttons as well as an empty section element. The empty section element uses the transition to swipe across the screen when another page on the app is selected.

Game in Operation

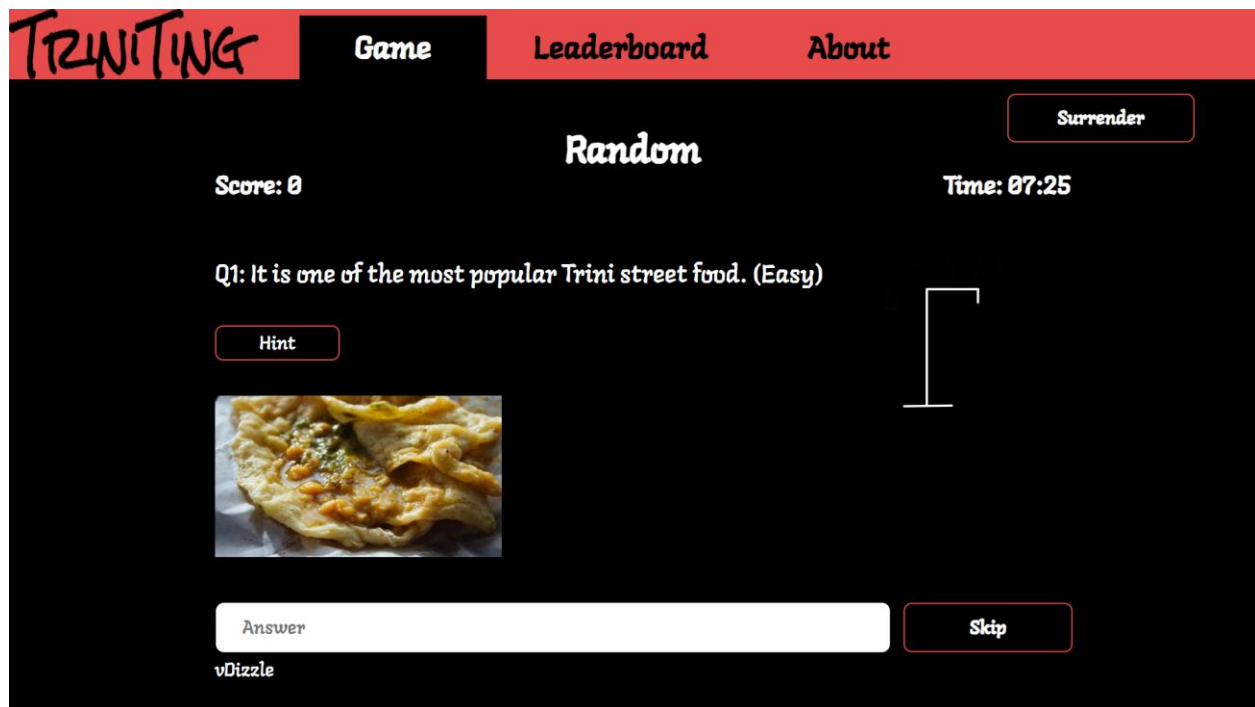
The landing page is loaded on the screen.



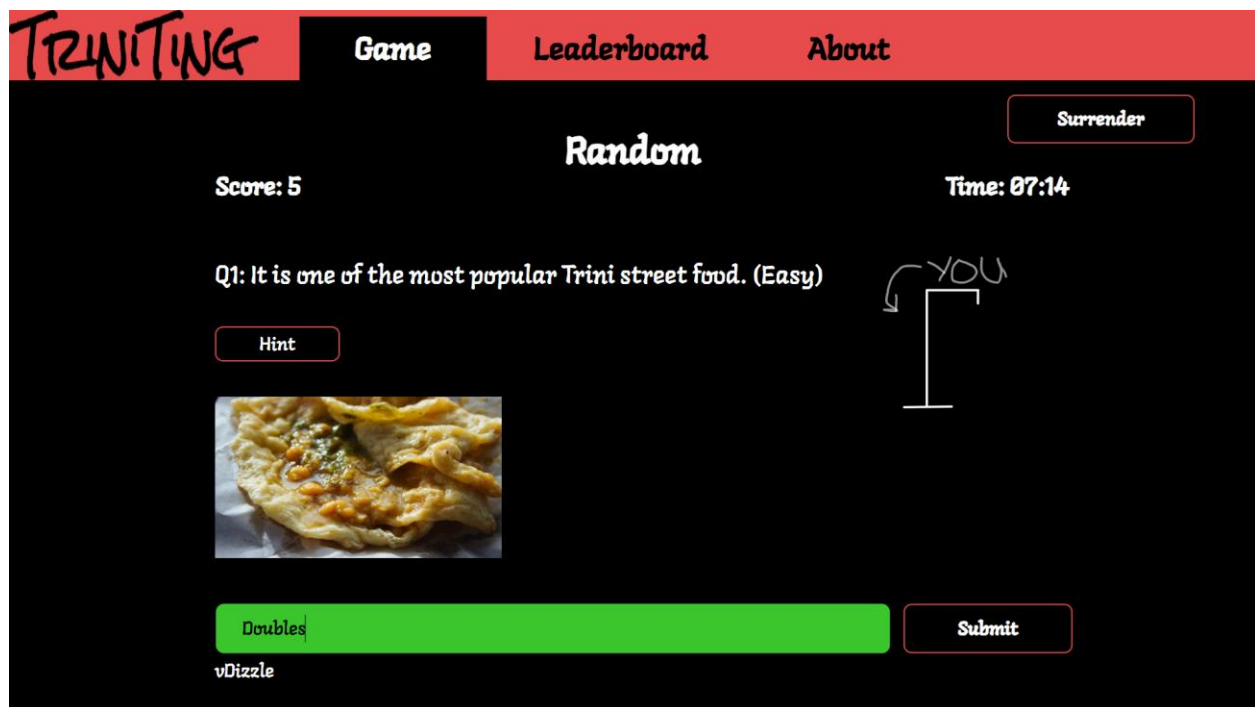
Once name is entered the game can start.



The game page is loaded with a question.



If the question is answered correct the question points is added to the user's score, a sound is played and the answer input field blinks green.



If the answer is wrong a body part is drawn on the hangman, a sound is played, and the answer input field blinks red. The same thing happens if the question is skipped.

TRINITY**Game****Leaderboard****About**

Score: 5


Random

Time: 06:51

Surrender

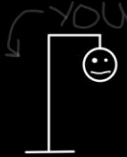
Q2: This is a celebration of the Hindu new year. (Easy)

Hint



I don't know

Submit



vDizzle

If the hint button is used 3 points will be deducted from your score and the hint will be displayed.

TRINITY**Game****Leaderboard****About**

Score: -3


Random

Time: 07:13

Surrender


Q1: In what part of Trinidad is this located? (Easy)

Hint: This is Bamboo Cathedral.



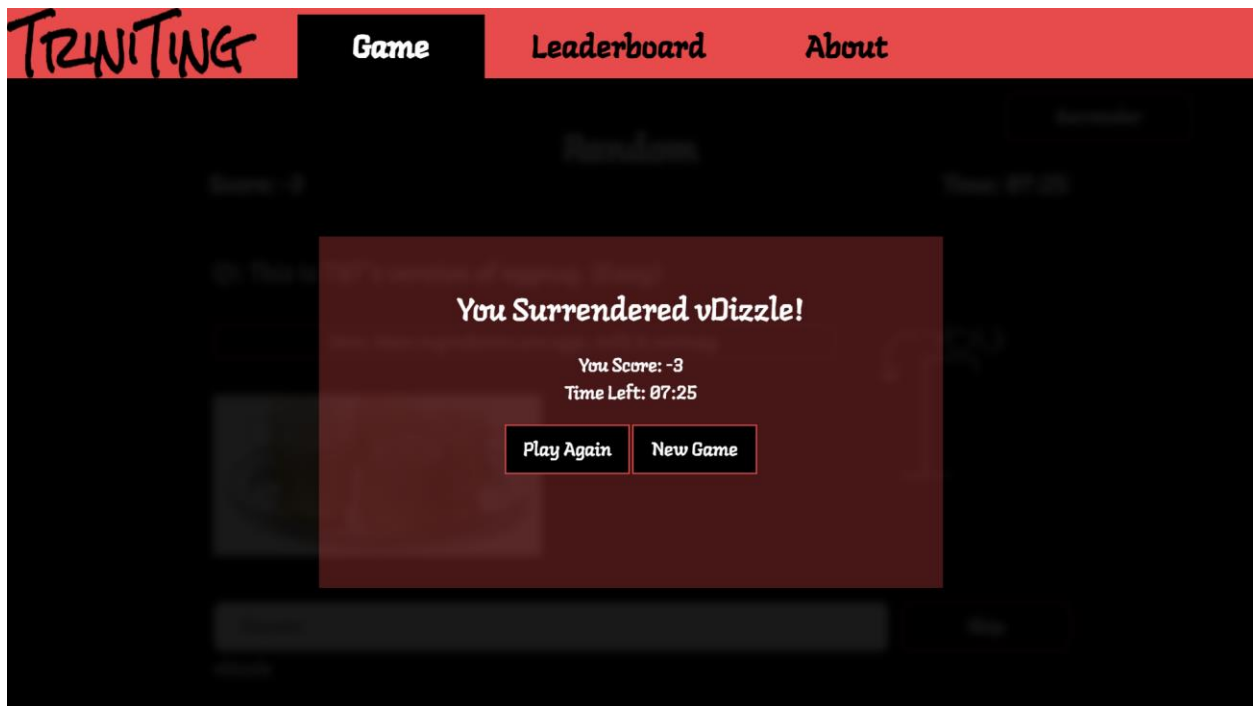
Answer

Skip

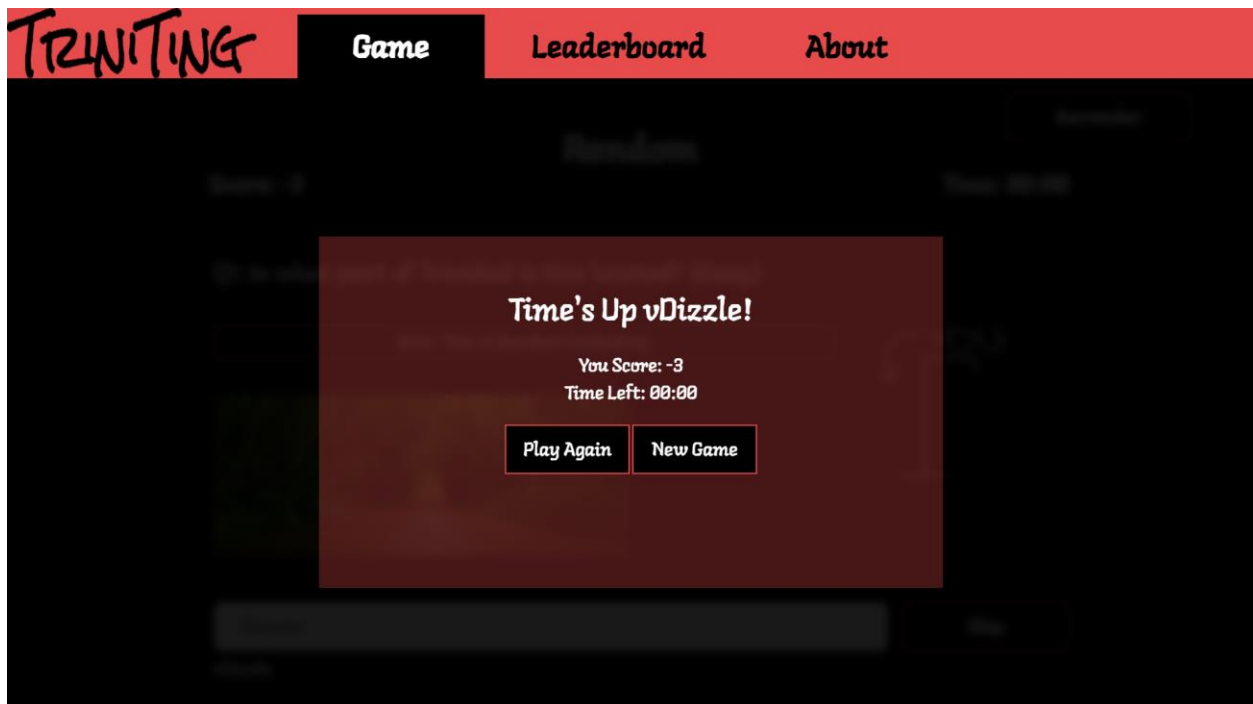


vDizzle

The surrender button displays a message and gives you the option to play again or start a new game which carries you back to the home screen.



If time runs out you lose, a message is displayed and you are given the option to play again or start a new game.



Borrowed material

This sound object from w3schools was modified and used for audio. ("Game Tutorial", 2021)

https://www.w3schools.com/graphics/game_sound.asp

```
function sound(src) {  
  this.sound = document.createElement("audio");  
  this.sound.src = src;  
  this.sound.setAttribute("preload", "auto");  
  this.sound.setAttribute("controls", "none");  
  this.sound.style.display = "none";  
  document.body.appendChild(this.sound);  
  this.play = function(){  
    this.sound.play();  
  }  
  this.stop = function(){  
    this.sound.pause();  
  }  
}
```

References

Ed, D. (2021). Retrieved 8 January 2021, from

https://www.youtube.com/watch?v=vJNVramny9k&ab_channel=DevEd

Game Tutorial. (2021). Retrieved 8 January 2021, from

https://www.w3schools.com/graphics/game_sound.asp