Trivia App

Team

James Giang  
Danial Mohammad  
Elmer Sevillano  
Curtis Turner  
Vishwas Vasudev

**Overview of the Project**

1. Project Scope
2. Installation and Configuration

* System Requirements
* Software Requirements
* Installing Ember
* Virtualbox Install and Import Appliance

1. Website Overview
2. Ember

* Ember Model
* Ember Model Object Attributes

1. Emberfire

* Installation

1. API

* Trivia API

1. User Documentation

* Website
* Signup
* Login
* Game

1. Technology We Used
2. Reference Links

**Project Scope**

## **Project scope & functionality**

The project is a Trivia-based game allowing user to answering trivia in a in a multiple choice format. The user will have get 2 attempts by click on the right answer. Points will be given to the user if the user answer the question correct. The score will be recorded in the backend with Ember. There will be a leaderboard showing the top 10 players in the game. User will be able to play as many times as they want.

## **Features**

* Create Account
* Login
* Trivia Application

**Installation and Configuration**

**System requirements**

* Intel or AMD processor (VT-x/AMD-V extension support required)
* At least 8 GB of RAM
* Stable Internet connection (20 Mbps or higher is recommended)
* A 64-bit host operating system
  + Apple's OS X 10.7 or greater
  + Microsoft's Windows XP 64 bit or newer

**Software requirements**

* VirtualBox version 5 or greater (not version 4)
  + https://www.virtualbox.org/wiki/Downloads
* Ubuntu MATE 64-bit VM
  + https://www.dropbox.com/s/9frc9x5blloo7a4/CSUF\_Ubuntu\_Mate\_16\_04\_01\_amd64\_20170122.ova?dl=0
* Latest Chrome Browser
  + https://www.google.com/chrome/browser/desktop/index.html
* EmberFire
  + https://github.com/firebase/emberfire

**Installing Ember**

In order to install Ember, you need to have Node.js and npm installed first. The installer for Node.js can be found on <https://nodejs.org/en/download/> and npm comes with the installation of Node.js.

The command to install Ember is:

$npm install -g ember-cli@2.17

If you want to make sure that the installation was successful, run:

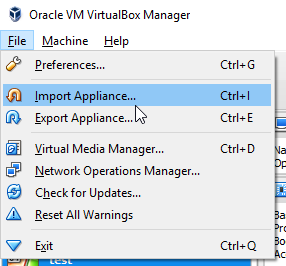
ember -v

To improve the performance of file watching on Ember, follow the installation guide for Watchman: <https://facebook.github.io/watchman/docs/install.html>

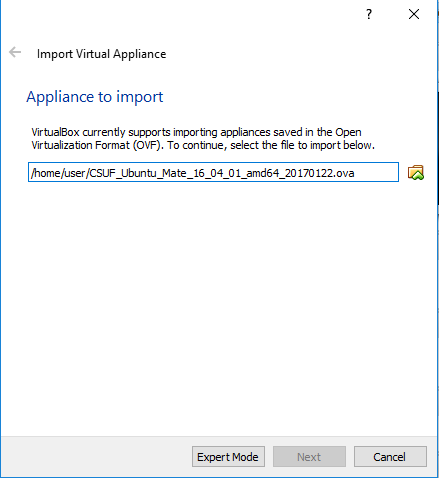
**Virtualbox Install and Import Appliance**

1. Download VirtualBox version 5: https://www.virtualbox.org/wiki/Downloads
2. Download Ubuntu MATE 64-bit: https://www.dropbox.com/s/9frc9x5blloo7a4/CSUF\_Ubuntu\_Mate\_16\_04\_01\_amd64\_20170122.ova?dl=0
3. Install VirtualBox version 5

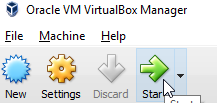
Click on File>import Appliance



Point to the CSUF\_Ubuntu\_Mate\_16\_04\_01\_amd64\_20170122.ova



Click on start



Use the following information username and password

* username: 'me'
* password: 'me123'

## **Downloading the Source Code**

* Please follow the Configuration and Installation Steps, making sure all the software requirements are met.
* Please go to <https://github.com/curtis-turner/triviaapp>
* Clone or download the source code from the Github
* Open a terminal, navigate to the root folder of the cloned / downloaded source code

**Website Overview**

## **Index page**

* Welcome to The Site

## **Sign In Page**

* Enter email
* Enter password
* Auth is completed
* User is redirected to Game
* If user needs an account they can click link to register

## **Sign Up page**

* Create username and password

## **Game page**

* User will use the mouse to click on the right answer
* User will get 2 attempts on selecting the right answer
* The game will ask 10 question before the game ends
* The score will be recorded in the backend.

## **Sign Out Page**

* If the user is signed in the can click the link to sign out
* If the user is not signed in the link to signout is inactive

# 

**Ember Installation**

**Ember Model**

|  |  |  |
| --- | --- | --- |
| Route | Route Path | Route Data |
| index | /index | None |
| sign-in | /sign-in | username, password |
| sign-up | /sign-up | Email, password, preffered name |
| sign-out | /sign-out | User session |
| game | /game | Answers Correct, User Answer, Attempts, Question, Answers |

Commands as follow

$ember generate route index

$ember generate route register

$ember generate route game

**Ember Model Object Attributes**

|  |  |
| --- | --- |
| User Model | Attribute Type |
| Name | string |
| emailAddress | string |
| password | string |
| Score | number |

Commands as follow

$ember generate model user

$ember generate model register

$ember generate model game

**Custom Controller**

Index page

* User login Auth
  + The username and password will check if the user input is in the database return false if it does not match

Register page

* The username and password will check if the user input is not the database return true if it does not match
* Connection controller port and session managements

Game page

* score count
  + Keep track on what question the user got correct
* submit count
  + keeps track of the number of tries
* Questions
  + Request 10 questions from the API
* Answers
  + Generate values for buttons from the answers that correlate to the question in the api json response.

Commands as follow

$ember generate controller index

$ember generate controller register

$ember generate controller game

# 

# **Emberfire**

# **Installation**

The README file on the GitHub repository has the commands and documentation necessary in order to incorporate EmberFire in the application.

The link to the repository is: <https://github.com/firebase/emberfire/blob/master/README.md>

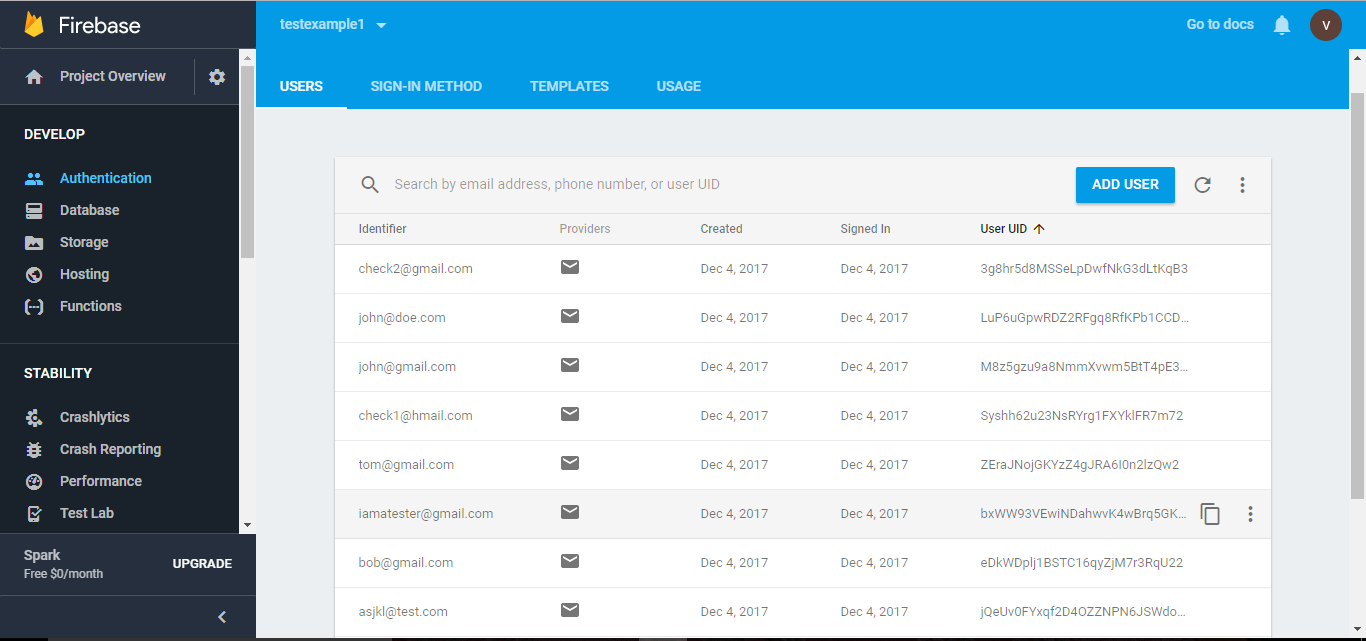
# **Integrating Firebase**

Firebase includes a step-by-step guide on including it within a web application. The link to the guide is <https://firebase.google.com/docs/web/setup>.

**Create user object**

const auth = this.get('firebaseApp').auth();

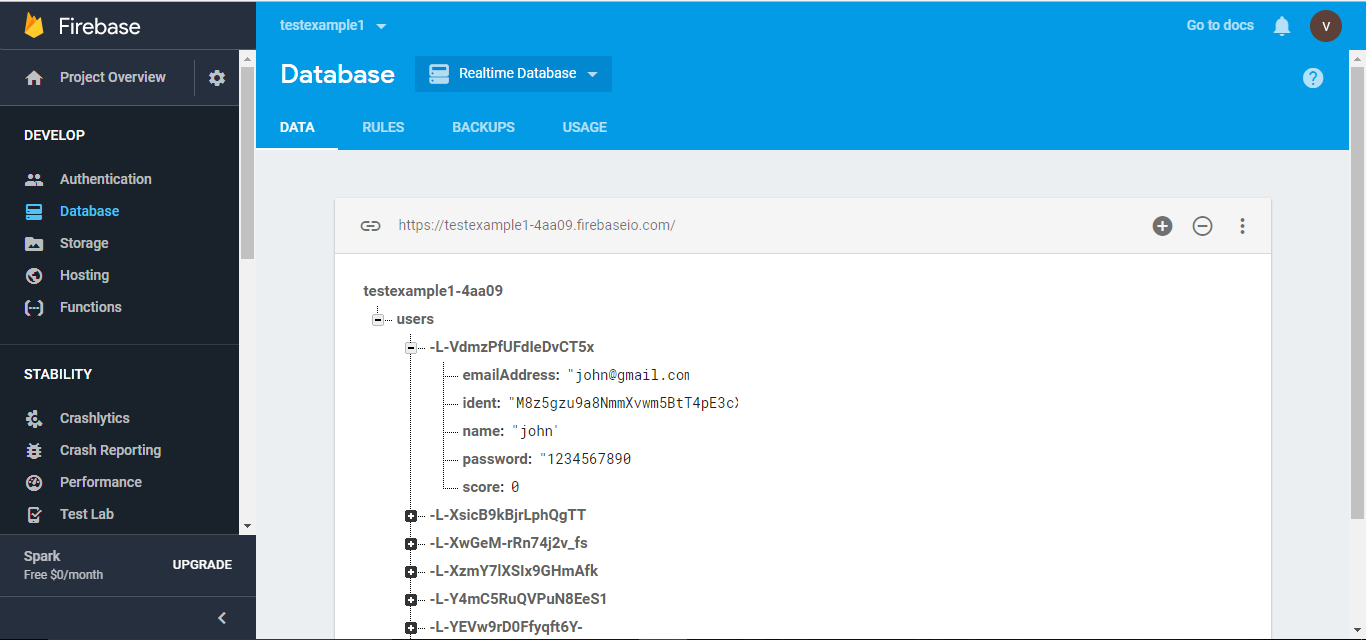
auth.createUserWithEmailAndPassword(email, password) // *This creates a records for Authentication*



context.store.createRecord(‘user’, { // This creates objects called users each time a new user is created

emailAddress: email,  
 name: name,  
 score:score

})

****

**Password authentication**

Password based authentication was done using with torii adapters.

Torii provides clean and clear abstractions for User Authentication in Ember Framework. Torii is built with the following features that enables easy and efficient User Auth.

1. Providers (Provides Authentication against different platforms like Google, Facebook, Twitter etc)
2. Session Management for user.
3. Adapters to hold session management ([*https://github.com/Vestorly/torii*](https://github.com/Vestorly/torii))

Torii Adapters use the API as follows,

1. **open** to create a new authori or to authenti
2. **fetch**, by which we mean validating an existing authorization
3. **close**, where an authorization is destroyed.([*https://github.com/Vestorly/torii*](https://github.com/Vestorly/torii))

this.get('session').open('firebase', { //*user authentication for sign in*

provider: provider,

email: email,

password: pwd,

})

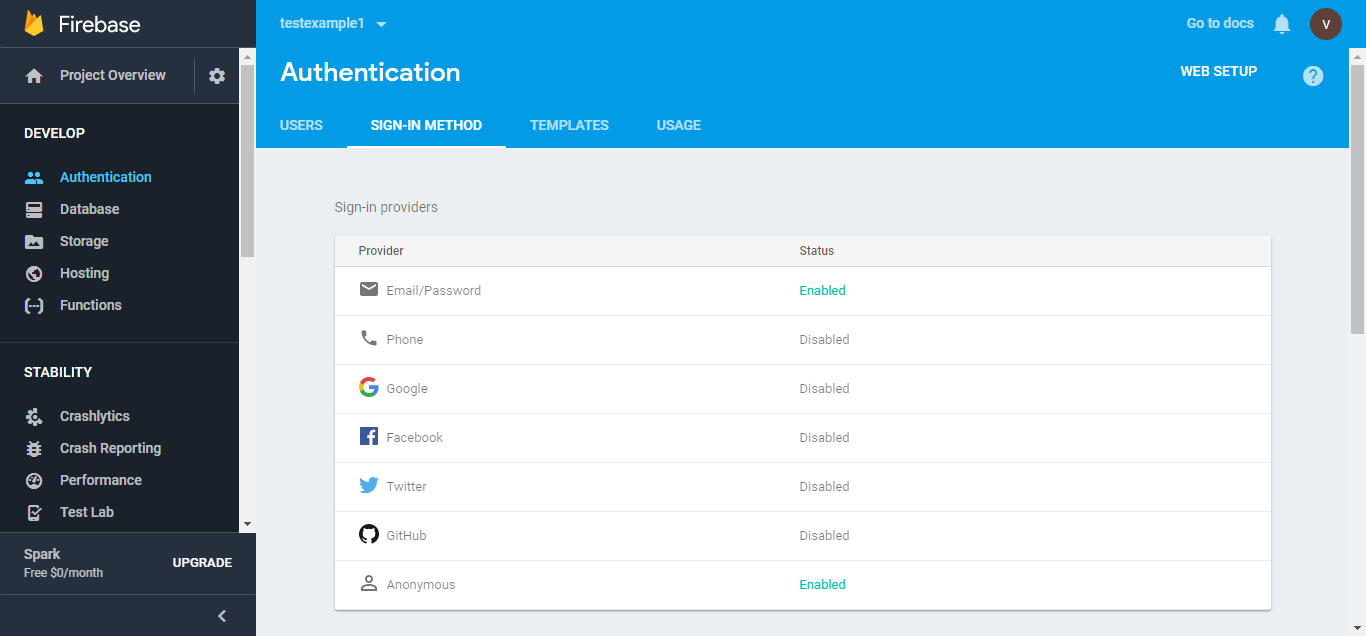
this.get('session').close() // *close the session*

Torii Session Management:

Torii performs session Management using a session that is injected to every controller that uses the session. One needs to specify sessionServiceName in the environment.js file and Torii takes care of the rest. ([*https://github.com/Vestorly/torii*](https://github.com/Vestorly/torii))

<https://github.com/firebase/emberfire/blob/master/docs/guide/authentication.md>

Link to API: <https://github.com/Vestorly/torii>



# 

# 

# 

# 

# 

# 

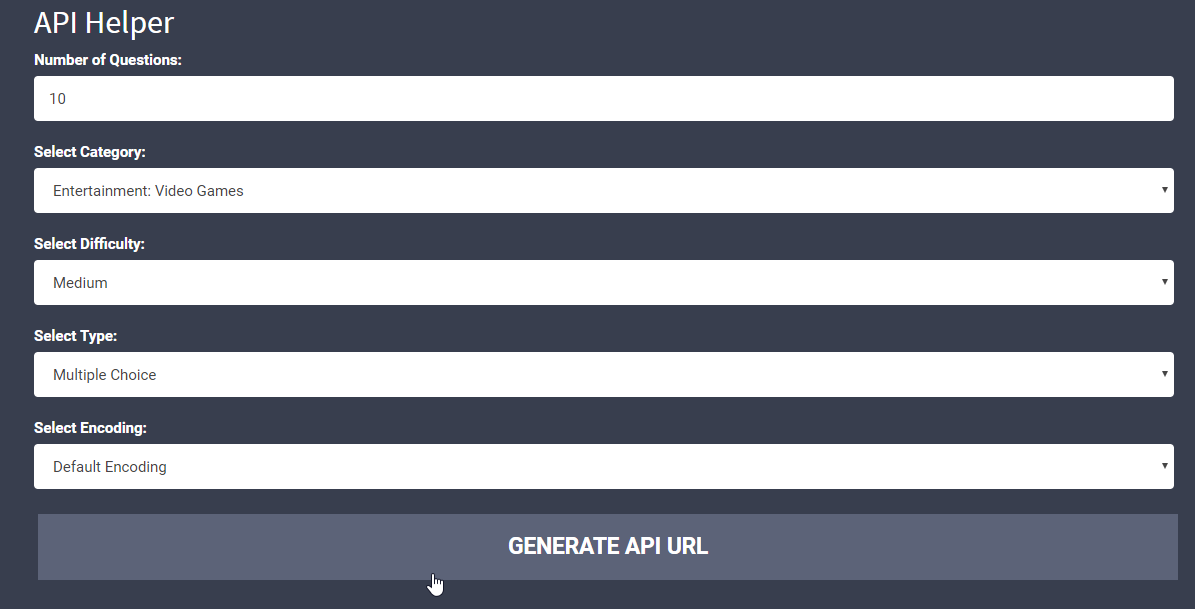
# 

# 

# **API Overview**

The API used in Trivia App is Open Trivia Database. To check out the documentation of the API, go to <https://opentdb.com>. The site includes an API helper where we can select

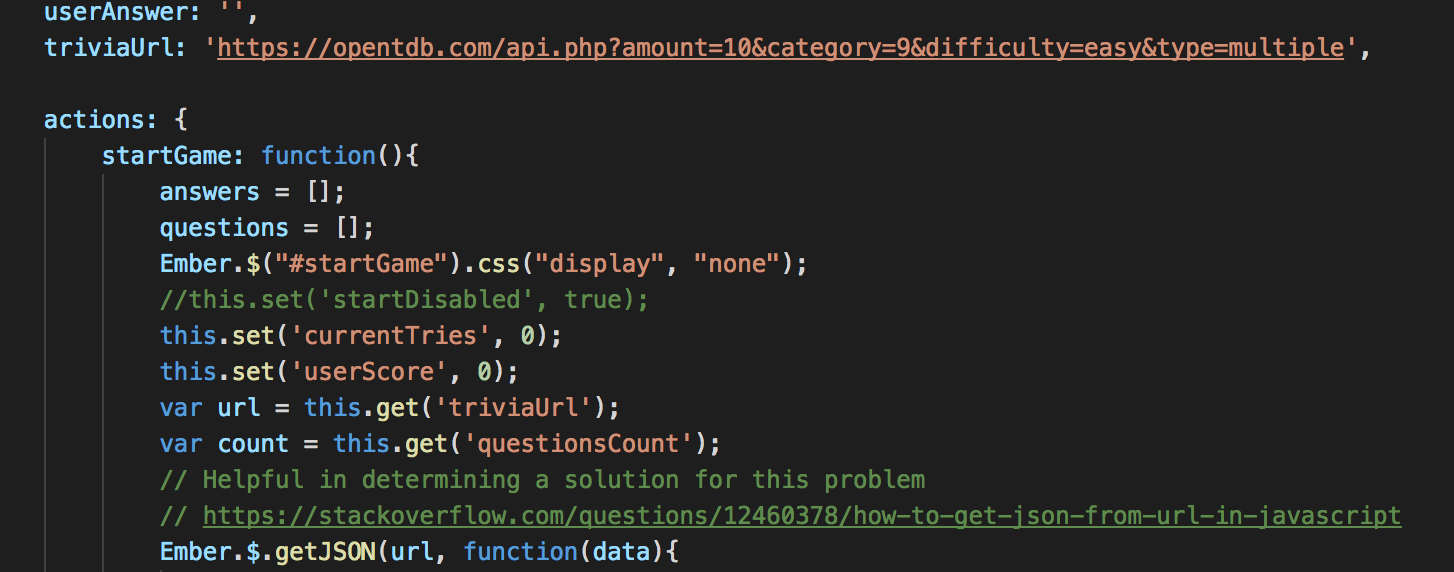
For our application, we used the API helper on the site and selected:



We will be using the API URL generated from the API helper

<https://opentdb.com/api.php?amount=10&category=15&difficulty=medium&type=multiple>

Input URL into the game route questions controller



# 

# 

# 

# 

# 

# 

# 

# 

# 

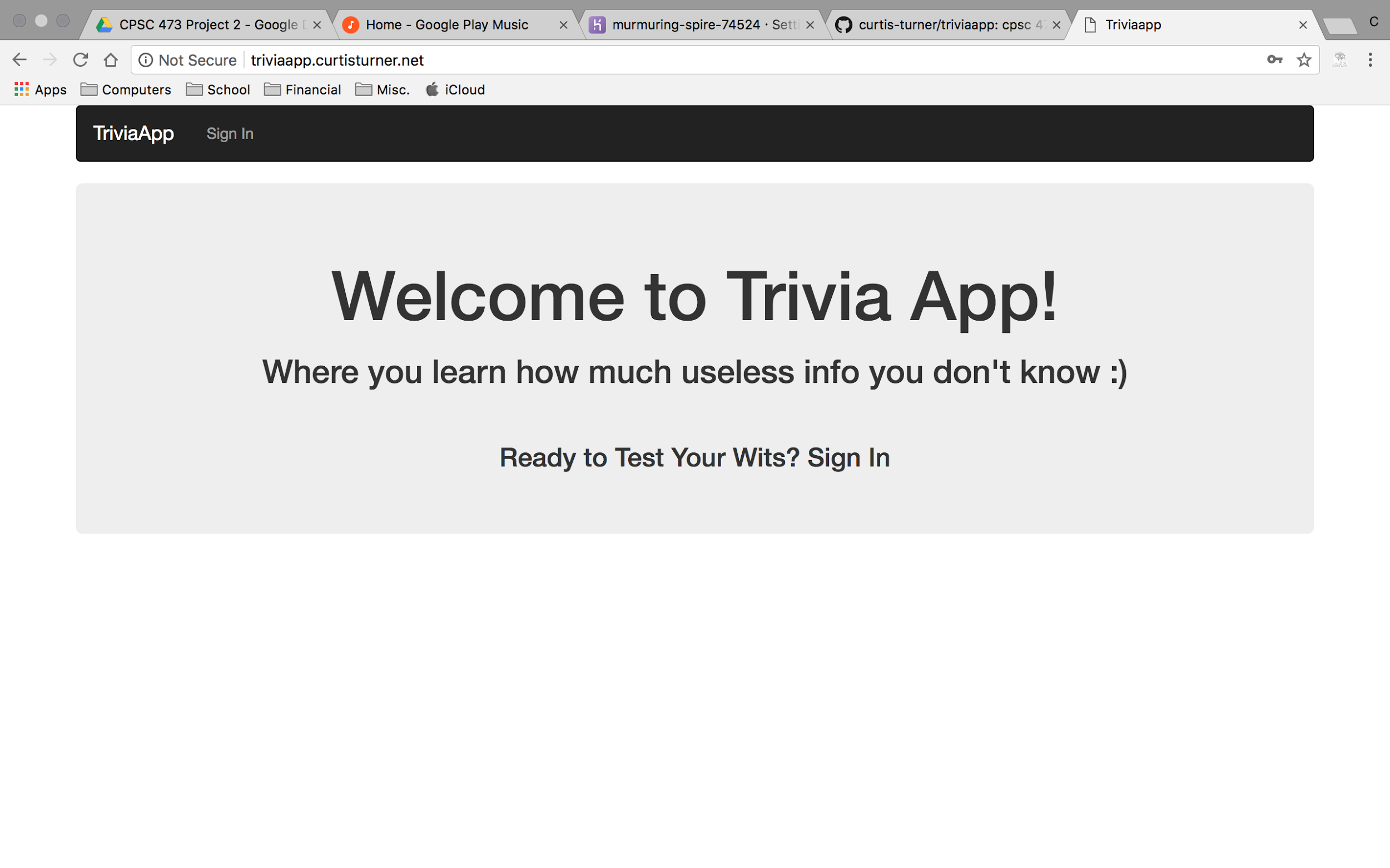
# 

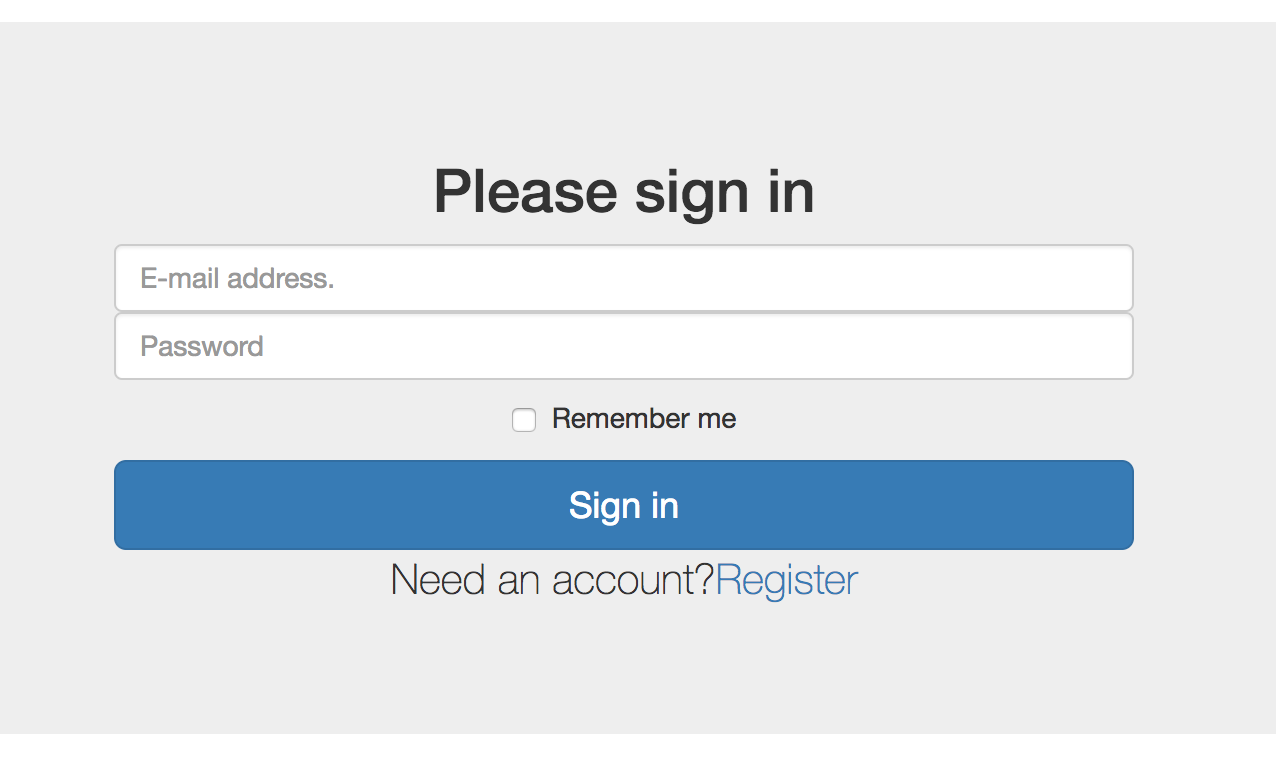
# 

# **User Documentation**

User will go to

<http://triviaapp.curtisturner.net/>

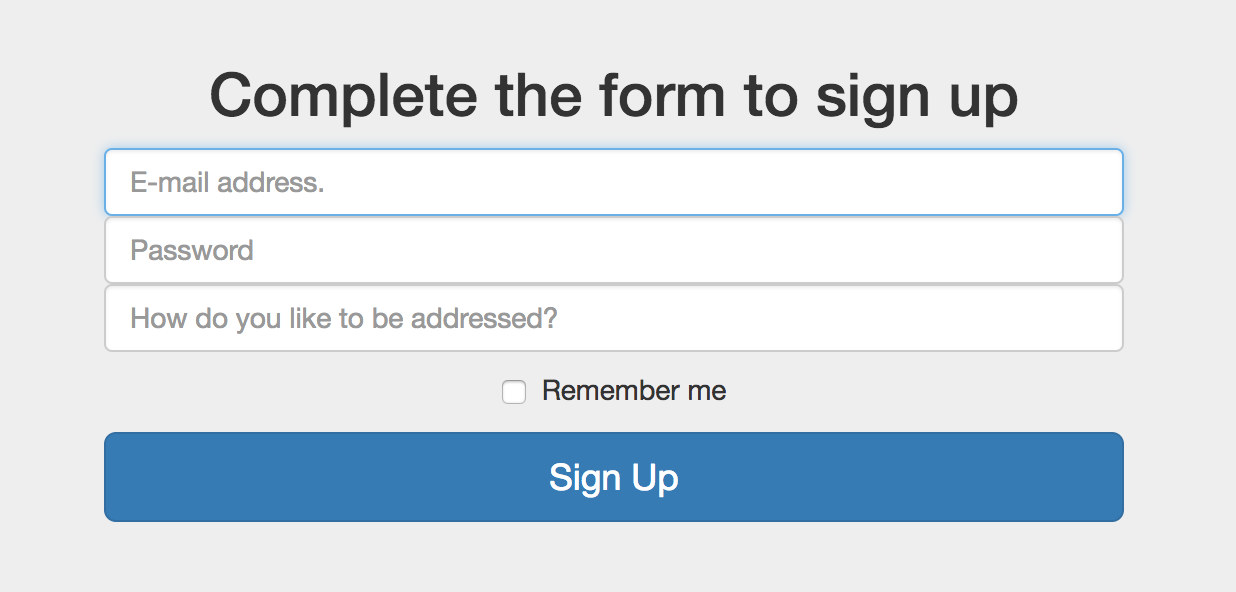


User will click on sign in

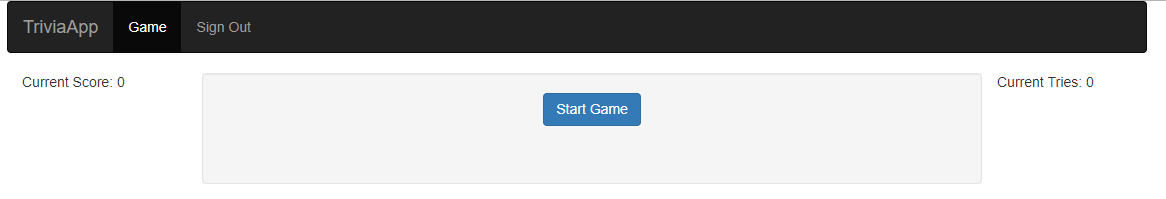
User will input email and password

User will login with authenticated account.

If user needs to create and account they can click the register link to sign-up

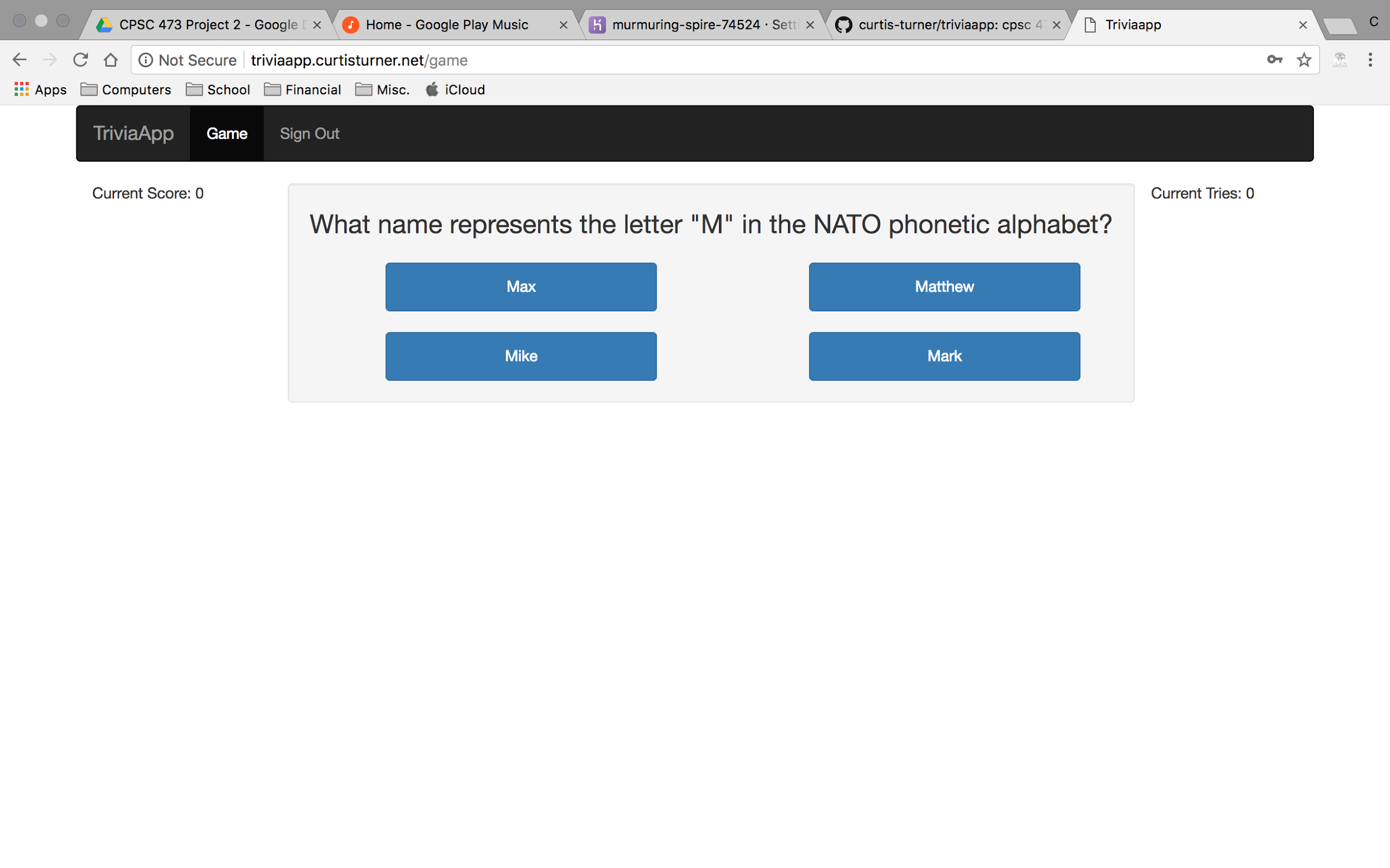


Click start



Game will ask 10 questions

User attempt to click on the correct answer



User will be able to see their correct score on the left side of the question.

User will be able to see their current tries on the right side of the question.

After two attempts the question will be skip and a new question will appear.

# 

# 

# 

# 

# 

# 

# 

# **Technology We Used**

## **Back-End**

* + Firebase
  + Ember

## **Development Tools**

* + Atom

## 

## **Languages**

* + HTML
  + JavaScript
  + CSS3

## 

## **Front-End**

* + Bootstrap 3.3.7
  + Google Fonts APIs

**Reference Links**

* <https://www.emberjs.com/>
* <https://opentdb.com>
* <http://www.programwitherik.com/emberjs_2-0_example_app_with_firebase/>
* <https://github.com/firebase/emberfire>