

Multiplayer trivia game

JavaScript and SocketIO part 2

Nathaniel Budijono

UMN ADC

October 19, 2021



Officer openings!

- Workshop instructors
- Marketing director

DM us on the discord!

<https://z.umn.edu/ADCdiscord>

Guest Speaker Event: Postman API 101!

November 2nd, 5-6pm

Tate Hall B20

Follow the guide!

<https://z.umn.edu/adc-mtg>

The screenshot shows the GitHub repository page for `ADC-UMN/multiplayer-trivia-game`. The repository is public and has 1 watch, 0 stars, and 1 fork. The main branch is `master` with 1 branch and 0 tags. The repository contains several files: `src`, `README.md`, `diagram.png`, `part-0.md`, `part-1.md`, `part-2.md`, and `part-3.md`. The `README.md` file is selected, showing the title "Create your own multiplayer trivia game with Flask and SocketIO". The README content includes a note that it may be helpful to look at Part 2 and Part 3 concurrently, and a list of links to Part 0: Overview, Environment, and Directory Structure; Part 1: Jinja Templating; and Part 2: Client-side Code. The right sidebar contains sections for About, Releases, Packages, and Languages. The About section includes a link to `nathanielbd-whodaman.net.app` and a Readme link. The Releases section shows no releases published. The Packages section shows no packages published. The Languages section shows a bar chart with the following data: JavaScript (41.1%), HTML (25.2%), CSS (18.9%), and Python (13.4%).

ADC-UMN / multiplayer-trivia-game

master 1 branch 0 tags

Go to file Add file Code

nathanielbd update to match bug fixes on whodaman repo 628w13 on Sep 17, 2020 4 commits

File	Description	Updated
src	update to match bug fixes on whodaman repo	13 months ago
README.md	link to original project and more challenges	13 months ago
diagram.png	first iteration of tutorial	13 months ago
part-0.md	update to match bug fixes on whodaman repo	13 months ago
part-1.md	update to match bug fixes on whodaman repo	13 months ago
part-2.md	update to match bug fixes on whodaman repo	13 months ago
part-3.md	update to match bug fixes on whodaman repo	13 months ago

README.md

Create your own multiplayer trivia game with Flask and SocketIO

It may be helpful to look at Part 2 and Part 3 concurrently.

- Part 0: Overview, Environment, and Directory Structure
- Part 1: Jinja Templating
- Part 2: Client-side Code

About

Let's learn how to use Flask and SocketIO to create a multiplayer trivia game!

nathanielbd-whodaman.net.app

Readme

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

JavaScript 41.1% HTML 25.2% CSS 18.9% Python 13.4%

The buzzing and scoring workflow

play.js

```
27
28 ✓ $buzzButton.on('click', function(event) {
29   event.preventDefault()
30   socket.emit('buzz', data)
31   $buzzButton.hide()
32   $state.show()
33 })
34
```

admin.js

```
132
133 socket.on('buzz', function(data) {
134   $buzzes.append(
135     ? '' : '<span><span class="judge" onclick="correct('$data.name')">✓</span> <span
136     class="judge" onclick="incorrect('$data.name')">✗</span></span>'</li>
137   )
138 }
```

The buzzing and scoring workflow

```
109
110 function correct(name) {
111     leaderboard[name] += stakes
112     $leaderboard.html(`
113     ${Object.entries(leaderboard).sort((a,b) => b[1]-a[1]).map(([key, value]) => `<li
114         class="panel_header">${key}<span>${value}</span></li>`).join('')}
115     `)
116     $buzzes.html('')
117     data.leaderboard = leaderboard
118     socket.emit('score', data)
119     $beginButton.click()
120
121 function incorrect(name) {
122     leaderboard[name] -= stakes
123     $leaderboard.html(`
124     ${Object.entries(leaderboard).sort((a,b) => b[1]-a[1]).map(([key, value]) => `<li
125         class="panel_header">${key}<span>${value}</span></li>`).join('')}
126     `)
127     $buzzes.find(':first-child').remove()
128     const next_name = $buzzes.find(':first-child').contents()[0].data.slice(0, -1)
129     $buzzes.find(':first-child').append(`<span><span class="judge" onclick="correct('${
130         {next_name}'>✓</span> <span class="judge" onclick="incorrect('${next_name}')">✗</
131         span></span>`)
132     `)
133     data.leaderboard = leaderboard
134     socket.emit('score', data)
135 }
```

Stuff to think about for features

Think about global variables needed in `app.py` or `<client>.js`

Stuff to think about for features

Think about global variables needed in `app.py` or `<client>.js`

- 1 `<client>.js` emits an event, perhaps with data

Stuff to think about for features

Think about global variables needed in `app.py` or `<client>.js`

- 1 `<client>.js` emits an event, perhaps with data
- 2 `app.py` performs security checks and broadcasts the event

Stuff to think about for features

Think about global variables needed in `app.py` or `<client>.js`

- 1 `<client>.js` emits an event, perhaps with data
- 2 `app.py` performs security checks and broadcasts the event
- 3 `<client>.js` listens for event, updates global variables, and updates the UI

You can look at the guides for details on the other uses of SocketIO!

<https://github.com/ADC-UMN/multiplayer-trivia-game/blob/master/part-2.md>

<https://github.com/ADC-UMN/multiplayer-trivia-game/blob/master/part-3.md>

Debugging tips

Try refreshing without hitting the cache or restarting the server if you don't see any updates.

Debugging tips

Try refreshing without hitting the cache or restarting the server if you don't see any updates.

Typing `Ctrl-Shift-i` will bring up developer tools. You can go to the console to execute arbitrary client-side JavaScript, see client-side error messages, or see outputs of `console.log()`.

Debugging tips

Try refreshing without hitting the cache or restarting the server if you don't see any updates.

Typing `Ctrl-Shift-i` will bring up developer tools. You can go to the console to execute arbitrary client-side JavaScript, see client-side error messages, or see outputs of `console.log()`.

Keep an eye on the terminal window running `python app.py`. This will have any print statements, log statements, and errors with your python code.

A challenge

Try implementing new features!

A challenge

Try implementing new features!

Fork <https://github.com/ADC-UMN/whodaman-ce> to get started!

A challenge

Try implementing new features!

Fork <https://github.com/ADC-UMN/whodaman-ce> to get started!

Some ideas:

- Try implementing 'Daily Doubles'!

A challenge

Try implementing new features!

Fork <https://github.com/ADC-UMN/whodaman-ce> to get started!

Some ideas:

- Try implementing 'Daily Doubles'!
- Try implementing real-time chat for the players!

A challenge

Try implementing new features!

Fork <https://github.com/ADC-UMN/whodaman-ce> to get started!

Some ideas:

- Try implementing 'Daily Doubles'!
- Try implementing real-time chat for the players!
- Add a feature to the landing page to display rooms in progress

A challenge

Try implementing new features!

Fork <https://github.com/ADC-UMN/whodaman-ce> to get started!

Some ideas:

- Try implementing 'Daily Doubles'!
- Try implementing real-time chat for the players!
- Add a feature to the landing page to display rooms in progress
- Add a menu to the admin page to configure things like what types of questions are used, a timer/max number of questions system, or other settings!