



## Instructions for starting and configuring our quiz app RAADIM

### Starting

#### 1. Prerequisites

- If not already installed, install the following JavaScript runtime environment, Node.js and npm
- Make sure all files are stored in the same folder.
- Before starting, make sure that the client and server are in the same local or network environment.

#### 2. Project Structure

Create a folder (here: RAADIM) and place the individual code files in this folder according to the following scheme:

```
/RAADIM
|
|-- client.js           //Client-side JavaScript for interactivity
|-- index.html         //HTML file (Client Interface)
|-- questions.json     //JSON file (questions)
|-- server.js          //WebSocket server script
|-- style.css          //CSS file for styling
|-- RAADIM.jpeg        //RAADIM logo
|-- background.jpg     //Background image
```

#### 3. Installing

Install the required packages in the project directory:

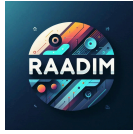
```
cd /RAADIM
npm install ws          //Package for WebSockets implementation
```

#### 4. Start the WebSocket-Server

- Change the port variable if needed
- Start the server with the command: `node server.js`
- If the settings are correct, the server should be running, and the IP address should be displayed.

#### 5. Open the Client

- Change the WebSocket connection IP Address and Port in `client.js` to the corresponding `server.js` websocket details.
- If you want to use the app in a local environment, open the `index.html` file with a Live Server Extension, otherwise create an external public http-server
- All users that try to connect to the game must enter the IP Address of the server in combination with the port of the created http server.



## 6. Using the quiz

- a. The first user to log in will be the Admin. This person can start the quiz and post the questions.
- b. All other users are players and can take part in the quiz by entering answers via the user interface.
- c. Have fun! 😊

## 7. Important Notice

If there is trouble with the files, please use our github repository, which is available with following link: <https://github.com/shr1mpsy/Quiz-Projekt.git>

## Additional Configuration Tips

### 1. How to implement new questions into the JSON-File

It is important to keep the same format as the given .json sample.

- **category**: Keep the following templates -> "yes-no", "sorm-choice", "guess"
- **question**: Your free choice, but keep it understandable for the users and specify your requirements as good as possible
- **answers**: Only for the category types "yes-no" and "sorm-choice"!
- **correctAnswer**: If you create a question for the type "yes-no" and "sorm-choice", please use a list datatype to store the correct answers. Insert integers corresponding to the indices of the answers that are correct. Otherwise, just use an integer or float datatype for the "guess" category.
- **tolerance**: Only for the "guess" category, the given integer will create a tolerance range in both directions for the clients.

### 2. Individual Time Adjustments

To adjust the timer of the given questions, it is possible to change the variable "time" of `sendTimeToClient()` in `server.js`. We chose a time of 10 seconds.

### 3. Optional removal of IP Address and Port Display

It is possible to remove the dynamic IP address and port output in `server.js`, if needed. This won't harm any process of the server script.