OUR FIRST WEB PROGRAM

A simple game

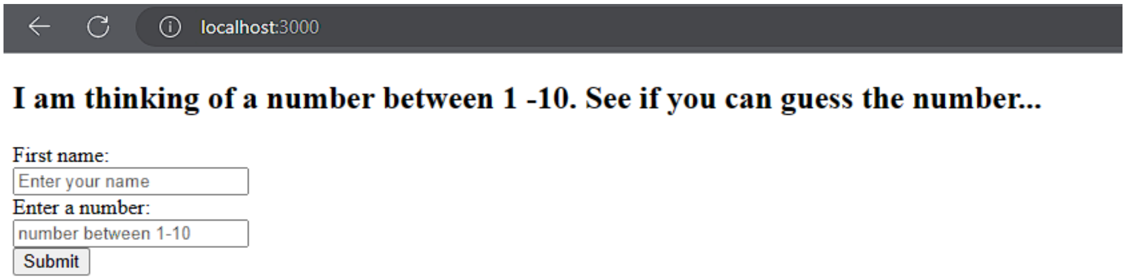
A Simple Web application

Directory Structure

Open Webstorm (IDE of your choice) and create a new project. Inside the root of your project you should add two files. The first is an HTML file called index.html and a javascript file called index.js. The index.html file will be where we have our home page, and the index.js will be where we have our server app.

THE GAME

The game is simple. When you navigate to url, “http://localhost:3000 your server will send the homepage. The homepage should will ask the user to give a name and a number in the range of 1 -10. When the user inputs these values and clicks the submit button, the page will make a request to the server. The URL should pass the name in the path and the number as a query parameter. (i.e. http://localhost:3000/Joe?num=7). When this request is sent to the server, there will be a route that generates a random number within the same range and compares the numbers. If the numbers are equal, you will print out a winner message. If they are not equal, you will print out a losing message. If the number given is invalid for any reason whatsoever, you should print out a message shaming the user for not following the instructions. All other requests to the server should redirect to home.



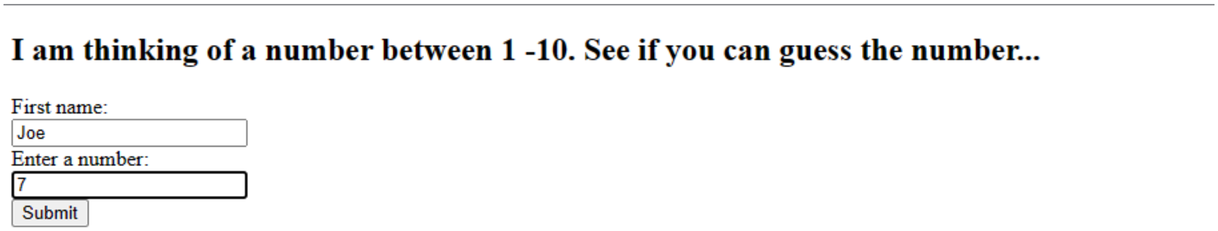
I am thinking of a number between 1 -10. See if you can guess the number...

First name:

Enter your name

Enter a number:

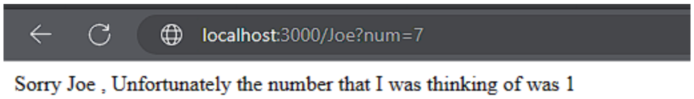
number between 1-10



I am thinking of a number between 1 -10. See if you can guess the number...

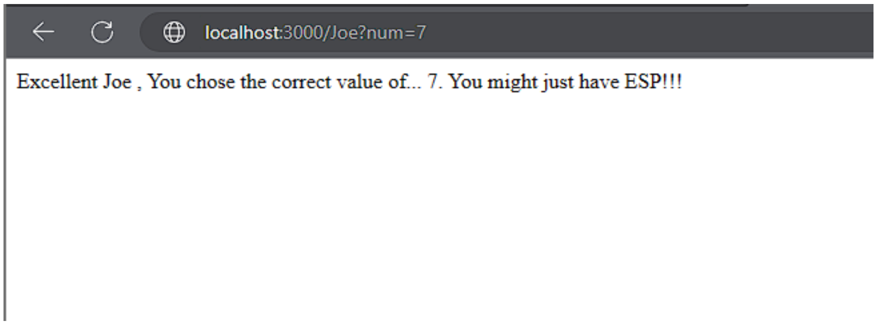
First name: Joe

Enter a number: 7



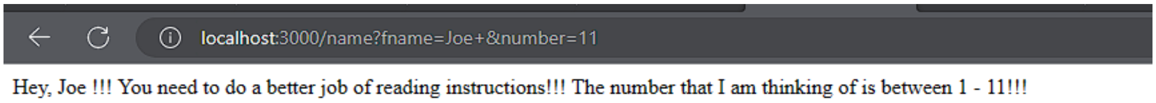
localhost:3000/Joe?num=7

Sorry Joe , Unfortunately the number that I was thinking of was 1



localhost:3000/Joe?num=7

Excellent Joe , You chose the correct value of... 7. You might just have ESP!!!



localhost:3000/name?fname=Joe+&number=11

Hey, Joe !!! You need to do a better job of reading instructions!!! The number that I am thinking of is between 1 - 11!!!

All other routes should redirect you back to home.

Tips:
Server:

- First consider what it is that you want to do and try to decide how many routes you will need on the server.
- If you want to know how to use some things, you can find essential help from the Express.js documentation.

Client:

- Use HTML forms for getting user data
- You should also use some client side js to prevent the default behavior of submitting a form, you will need to select and take the data from the form and use it to send a request in the appropriate format. http://localhost:3000/Joe?num=30 (for example)
- Making the request from the client side JS and be done with window.location.href