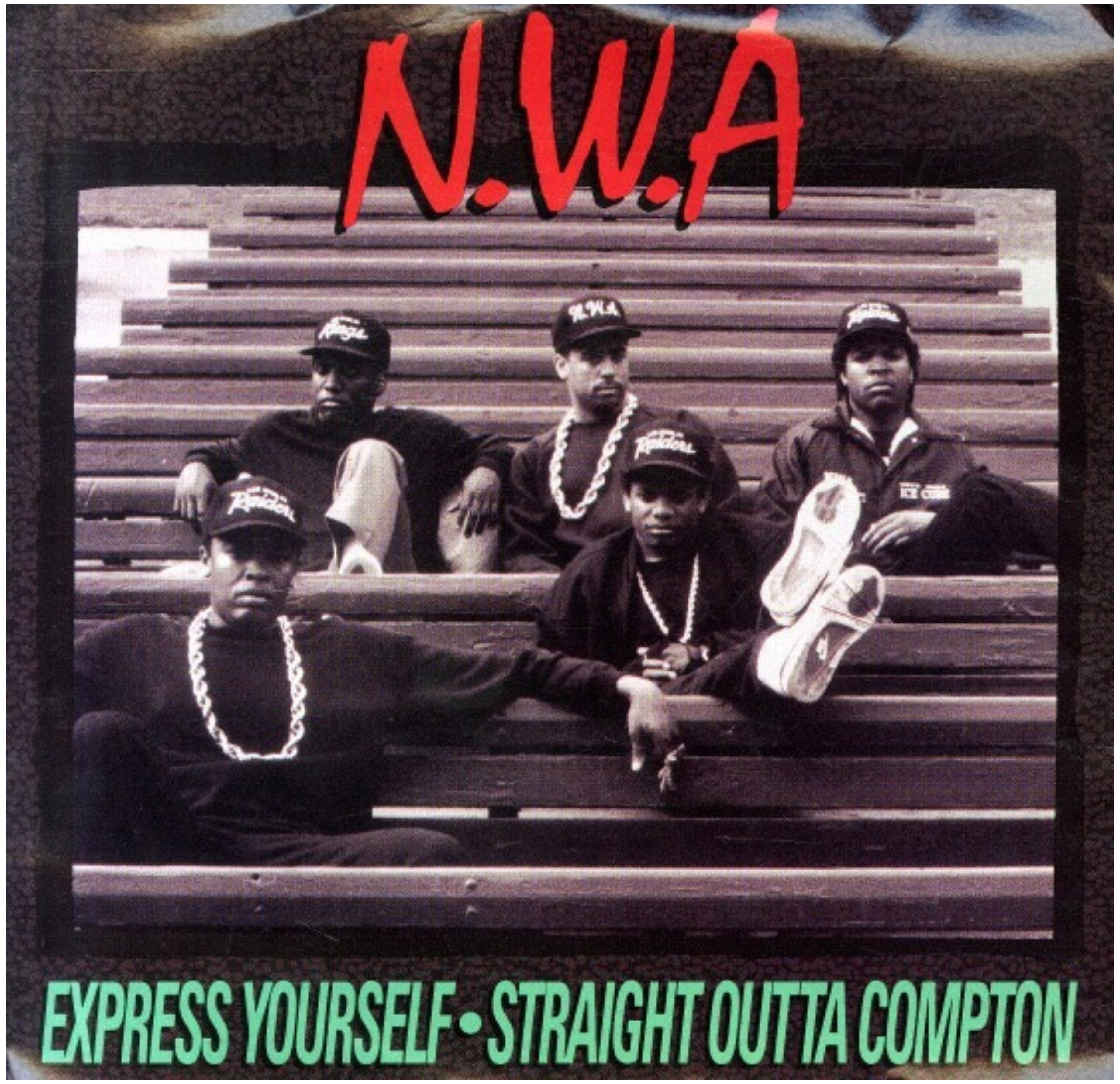


RUTGERS | CODING BOOTCAMP

Session 13.2 - Express Yourself

Express Yourself

Comment #2 - Concern

“I feel like I’m just copying and pasting”

Comment #2 - Response

- **Backend code libraries mean YOU have to code less.**
- Often you are just copying “best-practices” over and over again.

Objectives

- Know how to create a generic Express Server (copy and paste is fine).
- Know how to create a basic Express GET route
- Know how to create an Express POST route
- Know what POST Man is for
- Understand “conceptually” how to use AJAX to GET and POST data to an Express server

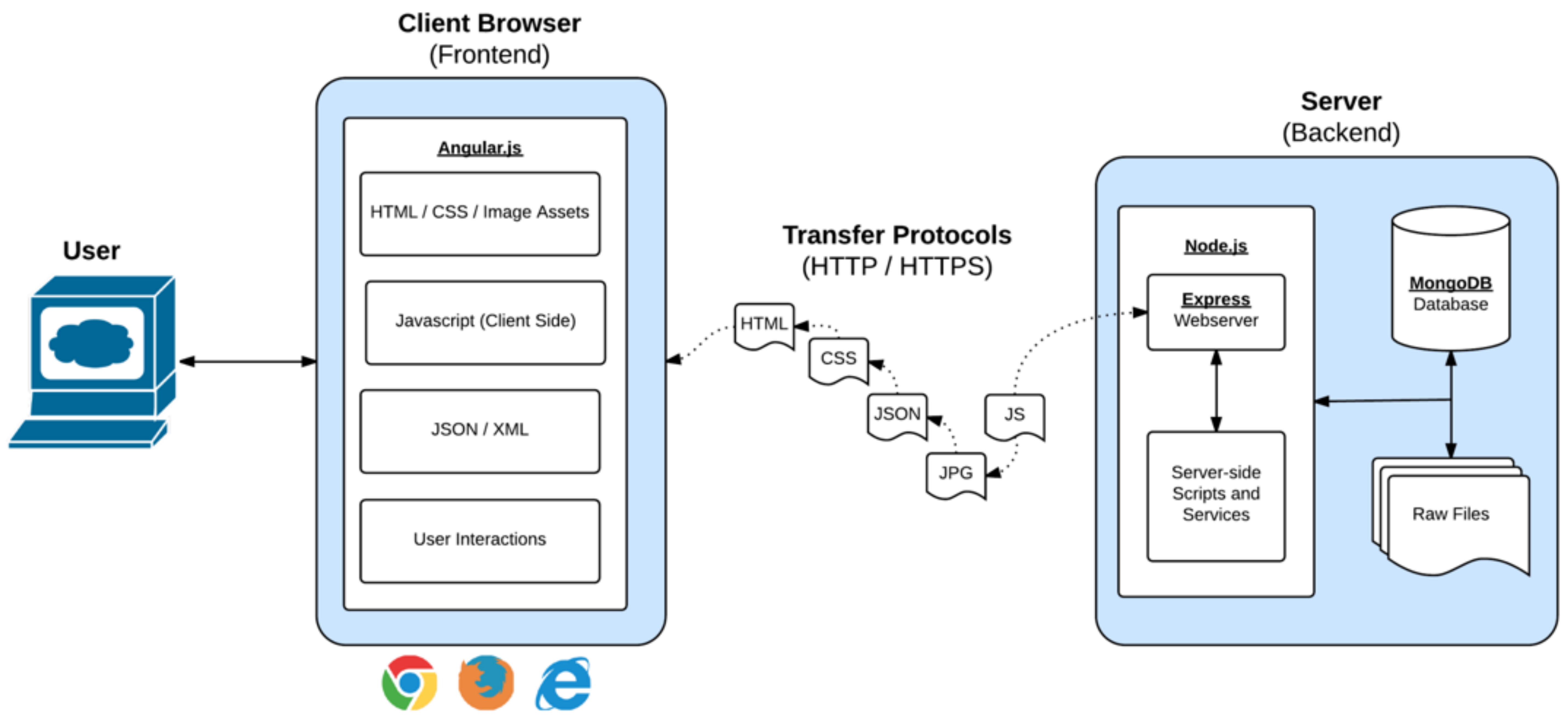
Node + Express Servers and Routing are two of the **MOST IMPORTANT CONCEPTS** in the ENTIRE program.

Relax. That doesn’t mean its hard.

A Moment to Refresh

FULL STACK
DEVELOPER
memegenerator.net

Full-Stack Development

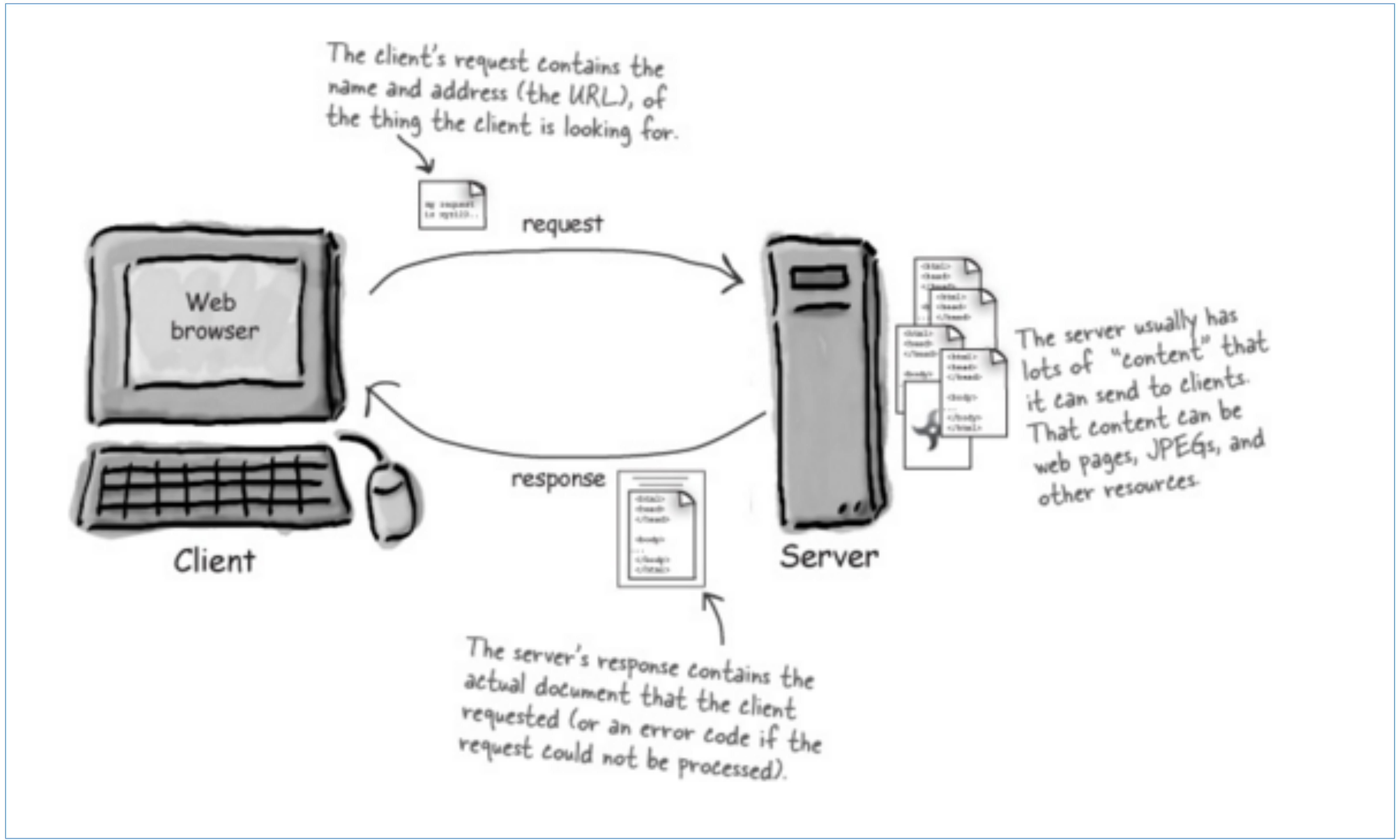


In modern **web applications** there is a constant back-and-forth communication between the visuals displayed on the user's browser (**frontend)** and the data and logic stored on the server (**backend).**

RUTGERS | CODING BOOTCAMP

What is a server?

What is a server?


The client's request contains the name and address (the URL), of the thing the client is looking for.
request
Web browser
Client
response
Server
The server usually has lots of "content" that it can send to clients. That content can be web pages, JPEGs, and other resources.
The server's response contains the actual document that the client requested (or an error code if the request could not be processed).

Server: Code that handles requests and respond to them.

What are examples of server-side functions?

What is a server?

API that parse URL parameters to provide selective JSONs

Firebase methods that provide a timestamp back to users

Clicking an invoice that provides a PDF report

Image processing software that takes an image applies a filter, then saves the new version

Google providing “results” relevant to your searches on other sites.

Where does the server live?

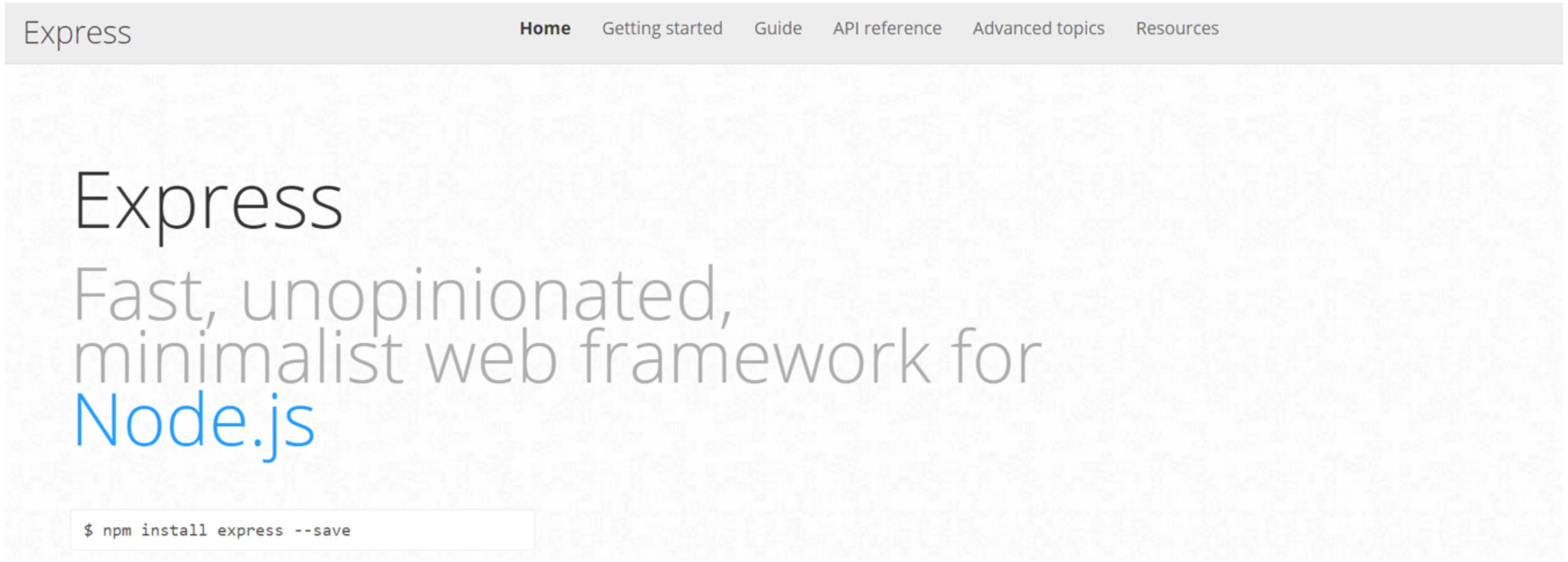
Hosting Environments

Servers live in dedicated hardware intended to handle ALL the requests and responses of many clients.

Servers can live most often live on cloud platforms like AWS, Heroku, Google Cloud, etc.

Express

Express.JS



Express

Fast, unopinionated, minimalist web framework for Node.js

```
$ npm install express --save
```

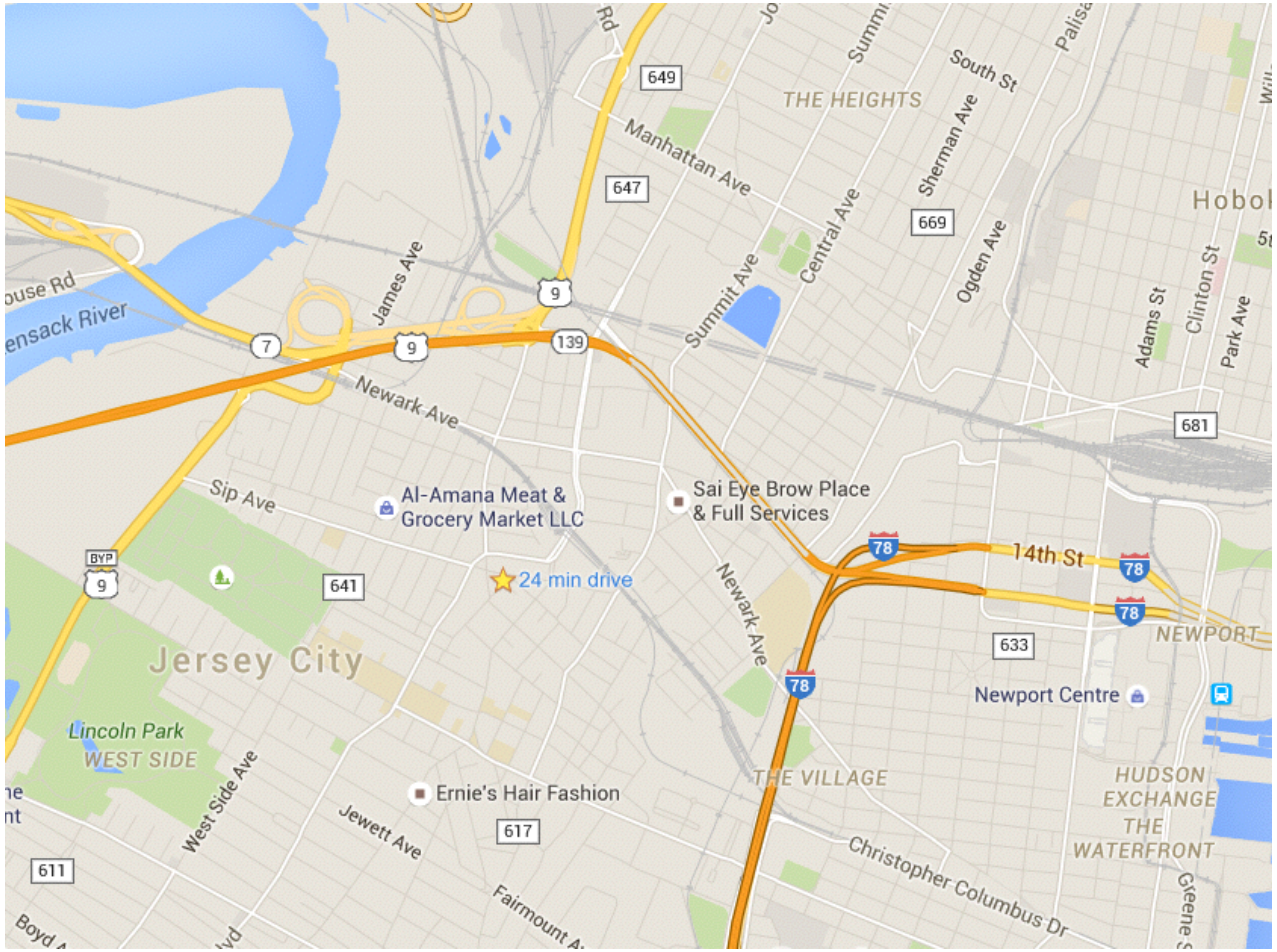
Web framework for node to make creating code for **server** much simpler

Remind me again…

What is a **route?**

Routes = Maps

/andyneuenschwander/times-archer-taught-you-how-to-avoid-people

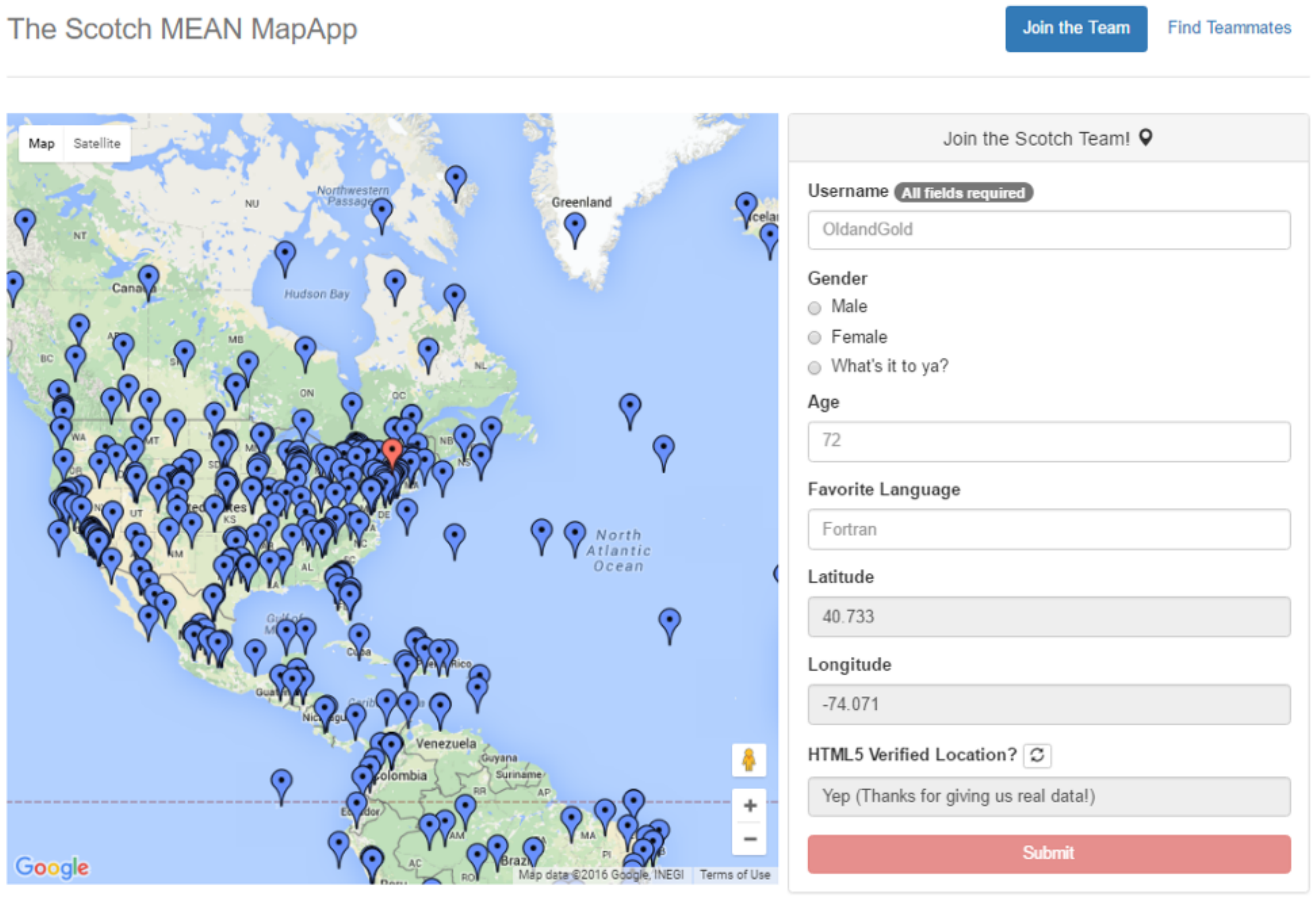


Routes = Maps

User visits a link…

/join	User is shown the “Join” Page
/add	User is shown the “Add” Page
/search/:name	User is shown the “search” page with just the results relevant to the name.

Quick Example


The Scotch MEAN MapApp

Join the Team | Find Teammates

Join the Scotch Team!

Username **All fields required**

OldandGold

Gender

- Male
- Female
- What's it to ya?

Age

72

Favorite Language

Fortran

Latitude

40.733

Longitude

-74.071

HTML5 Verified Location?

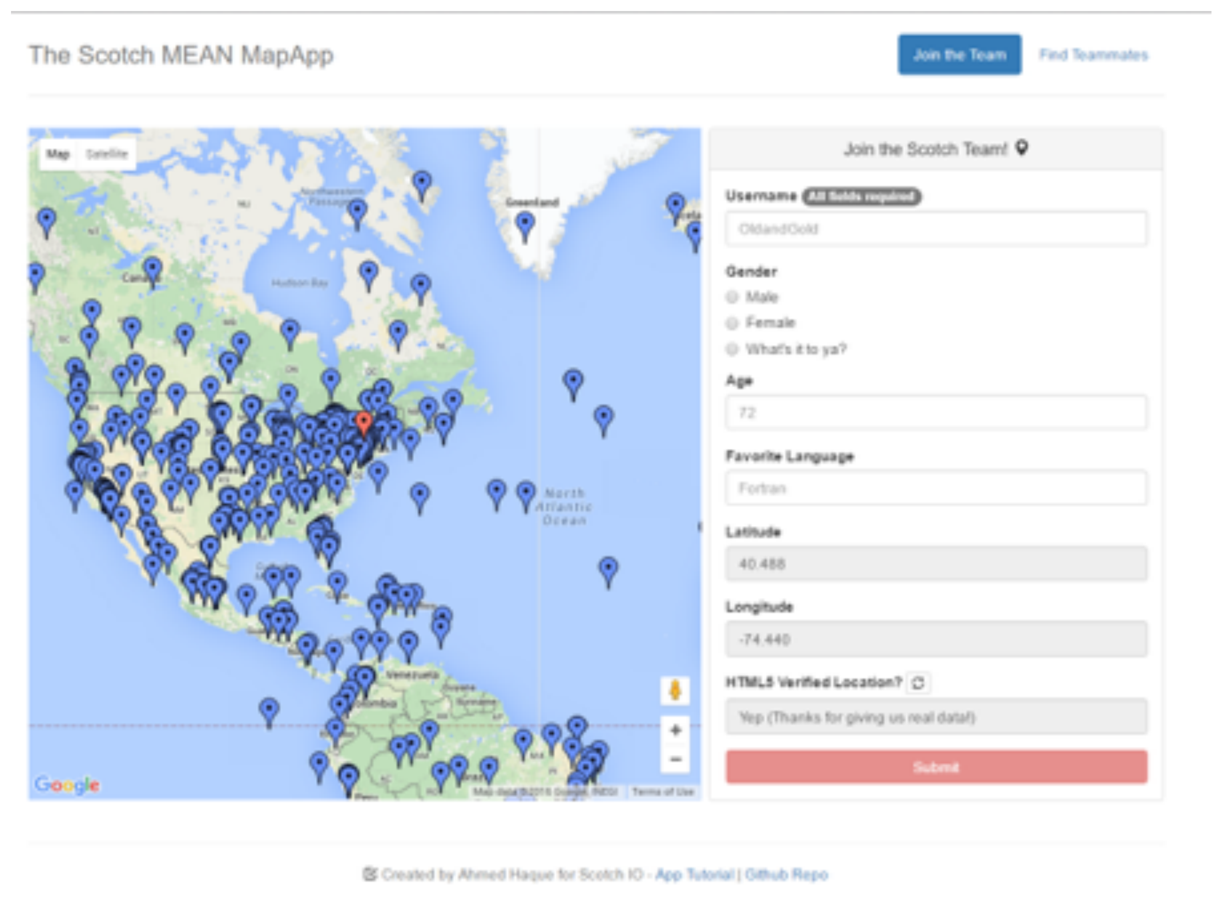
Yep (Thanks for giving us real data!)

Submit

https://mean-google-maps.herokuapp.com/

Client-Server Communication (GET)

Client Browser



The Scotch MEAN MapApp

Join the Scotch Team!

Username All fields required

OldandGold

Gender

- Male
- Female
- What's it to ya?

Age

72

Favorite Language

Fortran

Latitude

40.488

Longitude

-74.440

HTML5 Verified Location?

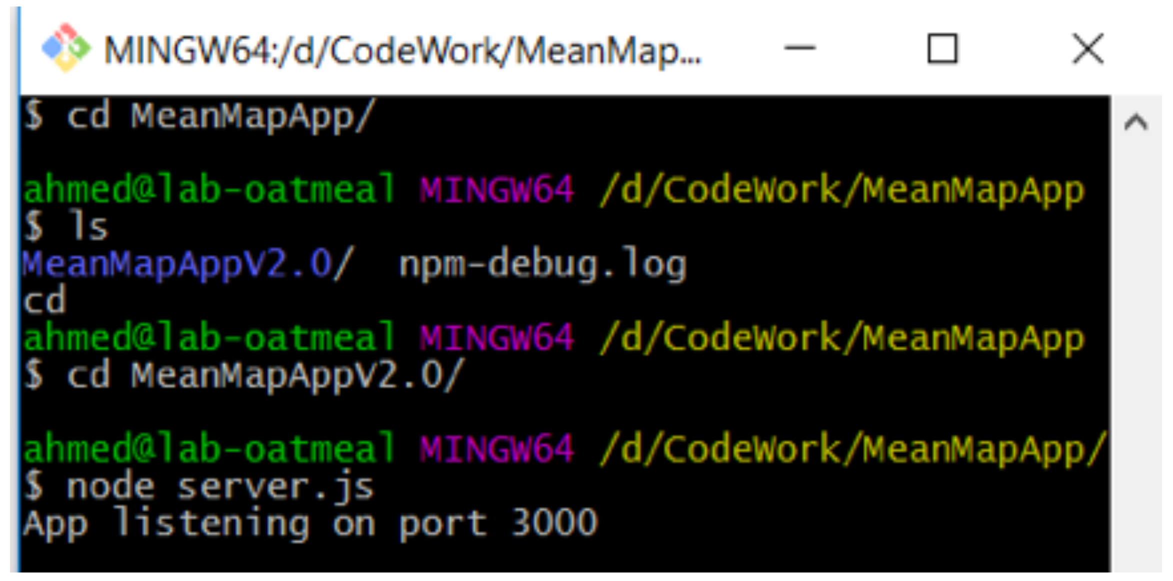
Yep (Thanks for giving us real data!)

Created by Ahmed Haque for Scotch IO - App Tutorial | Github Repo

1) User visits /join

(GET Request)

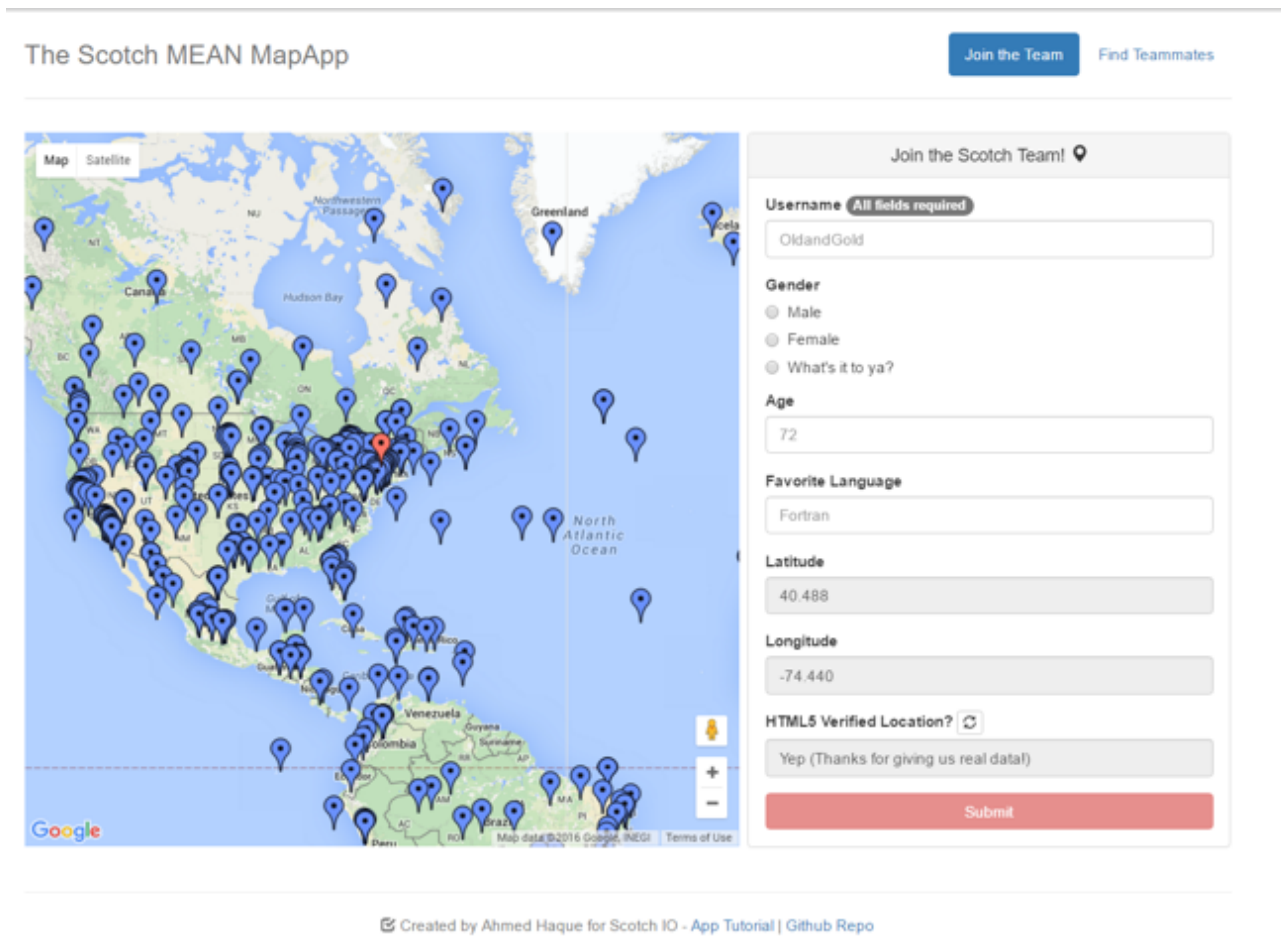
Node Server



```
$ cd MeanMapApp/

ahmed@lab-oatmeal MINGW64 /d/CodeWork/MeanMapApp
$ ls
MeanMapAppV2.0/  npm-debug.log
cd
ahmed@lab-oatmeal MINGW64 /d/CodeWork/MeanMapApp
$ cd MeanMapAppV2.0/

ahmed@lab-oatmeal MINGW64 /d/CodeWork/MeanMapApp/
$ node server.js
App listening on port 3000
```

Client-Server Communication (GET)

Client Browser



The Scotch MEAN MapApp

Join the Team | Find Teammates

Join the Scotch Team!

Username All fields required

OldandGold

Gender

- Male
- Female
- What's it to ya?

Age

72

Favorite Language

Fortran

Latitude

40.488

Longitude

-74.440

HTML5 Verified Location?

Yep (Thanks for giving us real data!)

Submit

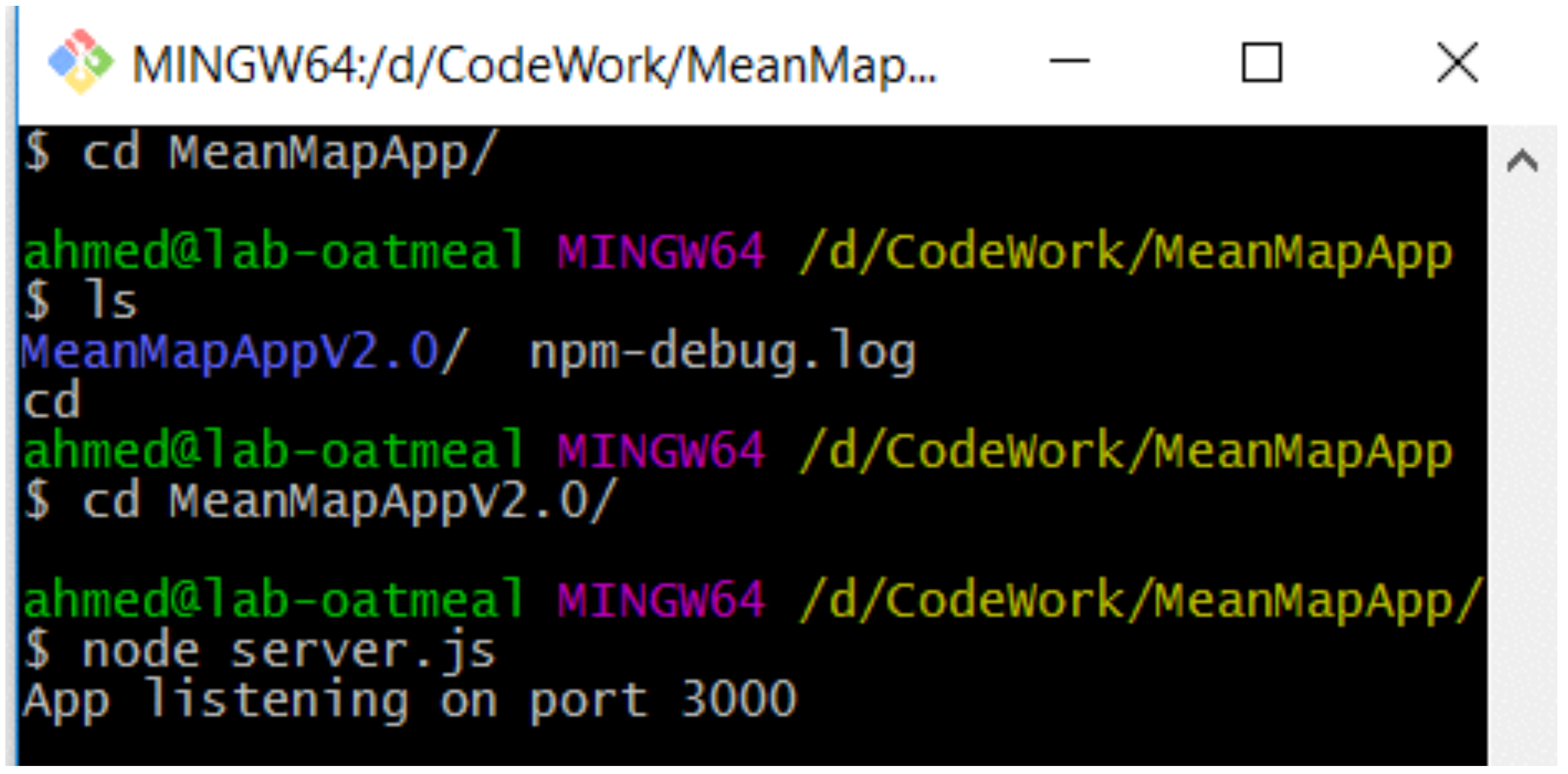
Created by Ahmed Haque for Scotch IO - App Tutorial | Github Repo

1) User visits /join

(GET Request)

3) Server responds by providing HTML with web form data

Node Server



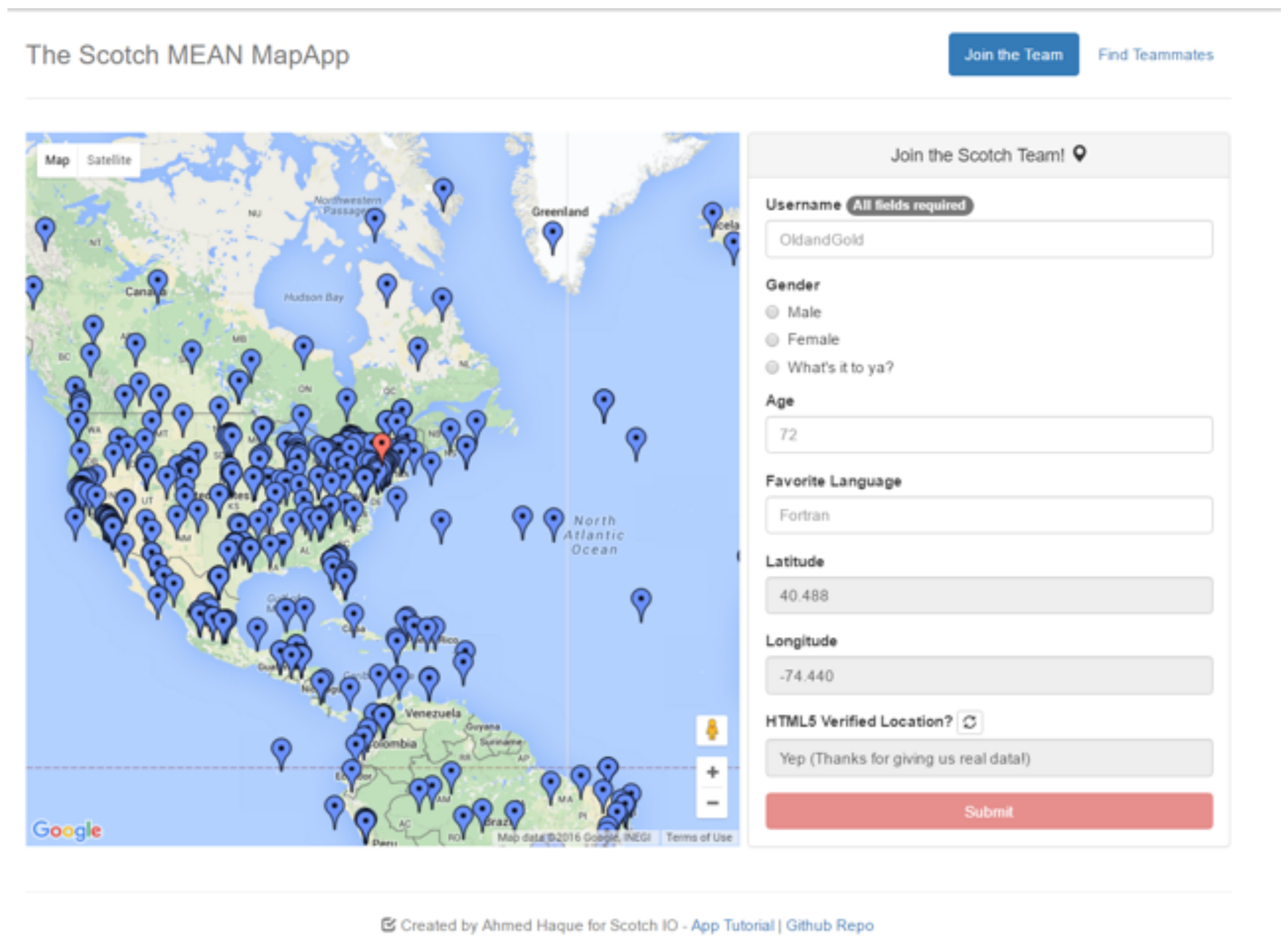
MINGW64:/d/CodeWork/MeanMap...

```
$ cd MeanMapApp/

ahmed@lab-oatmeal MINGW64 /d/CodeWork/MeanMapApp
$ ls
MeanMapAppV2.0/  npm-debug.log
cd
ahmed@lab-oatmeal MINGW64 /d/CodeWork/MeanMapApp
$ cd MeanMapAppV2.0/

ahmed@lab-oatmeal MINGW64 /d/CodeWork/MeanMapApp/
$ node server.js
App listening on port 3000
```

2) Request triggers the code in the Server route. The server then finds the relevant HTML content and data

Client-Server Communication (POST)

Client Browser



The Scotch MEAN MapApp

Join the Scotch Team!

Username All fields required

OldandGold

Gender

- Male
- Female
- What's it to ya?

Age

72

Favorite Language

Fortran

Latitude

40.488

Longitude

-74.440

HTML5 Verified Location?

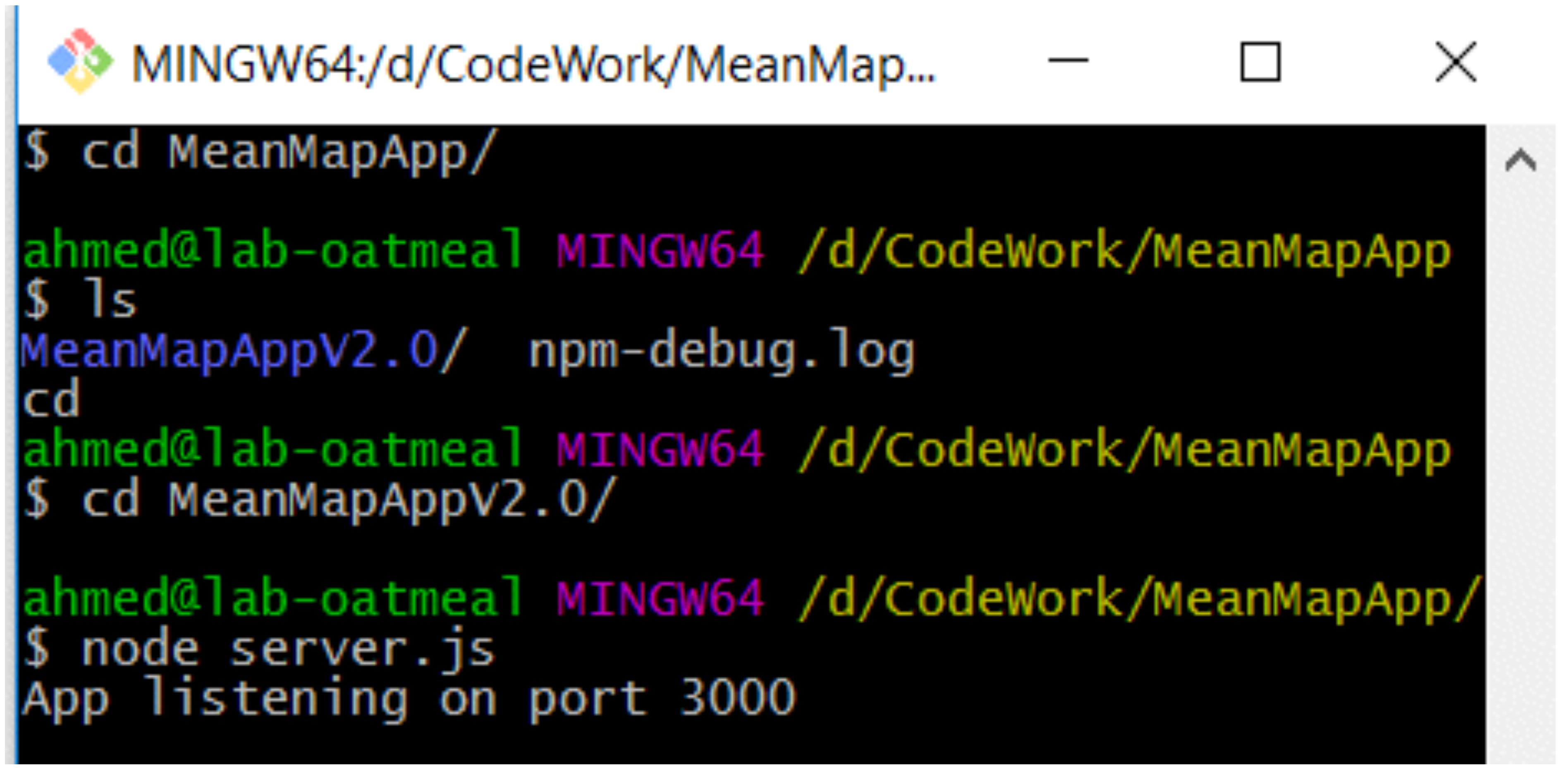
Yep (Thanks for giving us real data!)

Submit

Created by Ahmed Haque for Scotch IO - App Tutorial | Github Repo

1) User submits form entry
(POST Request)

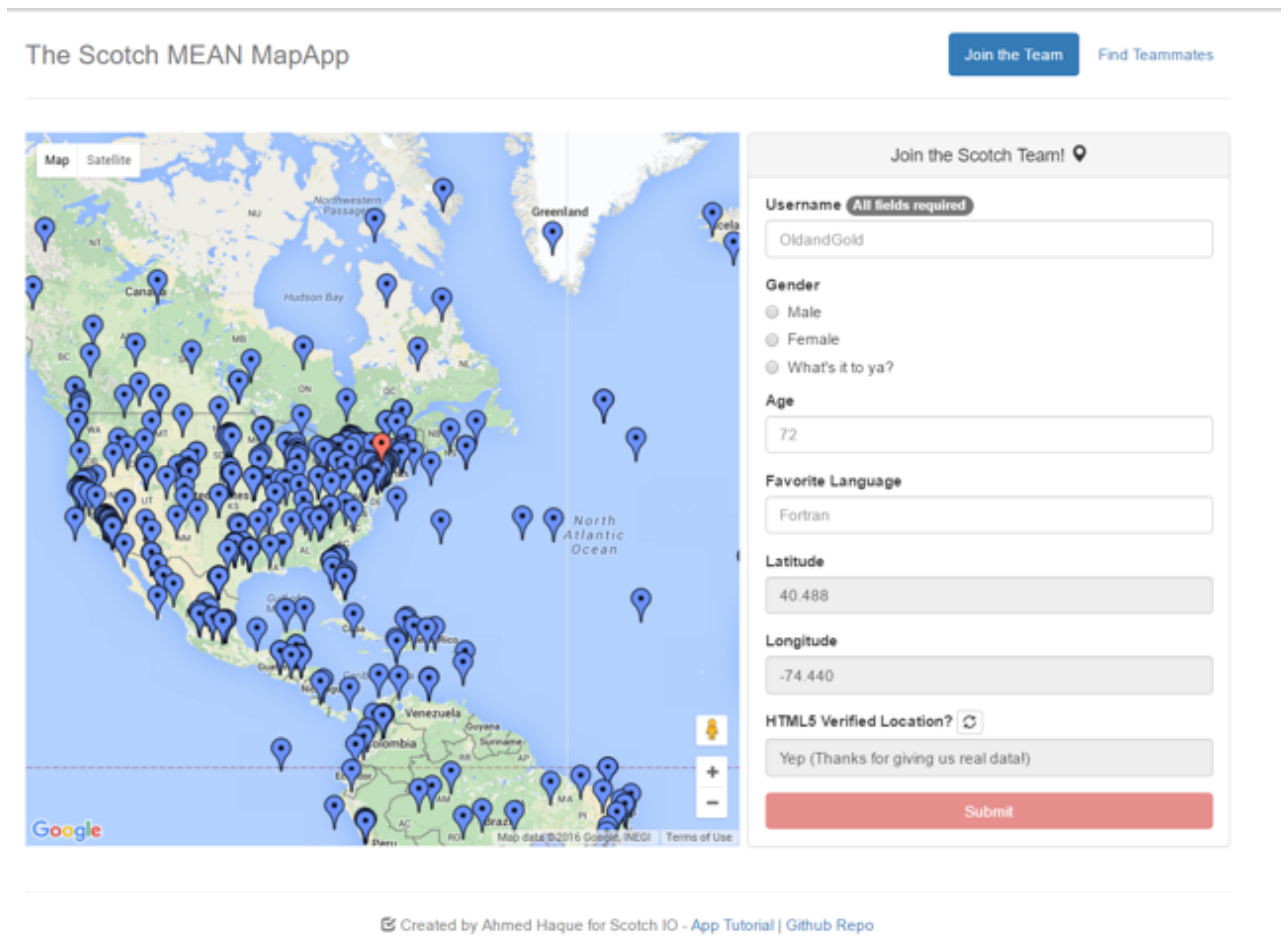
Node Server



```
MINGW64:/d/CodeWork/MeanMap...
$ cd MeanMapApp/

ahmed@lab-oatmeal MINGW64 /d/CodeWork/MeanMapApp
$ ls
MeanMapAppV2.0/  npm-debug.log
cd
ahmed@lab-oatmeal MINGW64 /d/CodeWork/MeanMapApp
$ cd MeanMapAppV2.0/

ahmed@lab-oatmeal MINGW64 /d/CodeWork/MeanMapApp/
$ node server.js
App listening on port 3000
```

Client-Server Communication (POST)

Client Browser



Join the Scotch Team!

Username All fields required

OldandGold

Gender

- Male
- Female
- What's it to ya?

Age

72

Favorite Language

Fortran

Latitude

40.488

Longitude

-74.440

HTML5 Verified Location?

Yep (Thanks for giving us real data!)

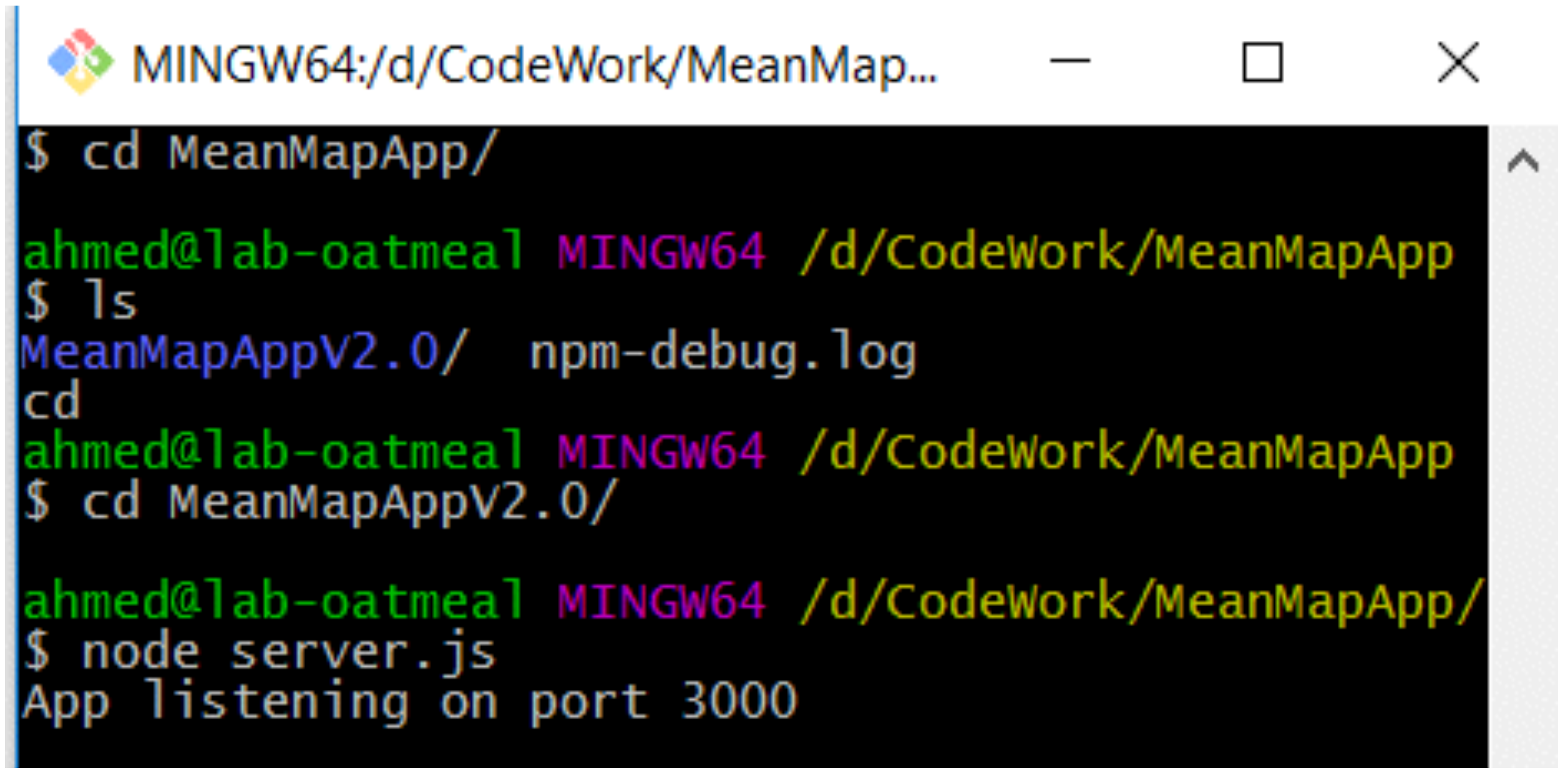
Submit

Created by Ahmed Haque for Scotch IO - App Tutorial | Github Repo

1) User submits form entry **(POST Request)**

3) Server responds by providing HTML with web form data

Node Server



```
$ cd MeanMapApp/

ahmed@lab-oatmeal MINGW64 /d/CodeWork/MeanMapApp
$ ls
MeanMapAppV2.0/  npm-debug.log
cd
ahmed@lab-oatmeal MINGW64 /d/CodeWork/MeanMapApp
$ cd MeanMapAppV2.0/

ahmed@lab-oatmeal MINGW64 /d/CodeWork/MeanMapApp/
$ node server.js
App listening on port 3000
```

2) Request triggers the code in the Server route. The server then adds the user to the map and database.

Activity Time!

Create a Web Server

```
//Lets require/import the HTTP module
var http = require('http');

//Lets define a port we want to listen to
var PORT=8080;

//We need a function which handles requests and send response
function handleRequest(request, response){
    response.end('It Works!! Path Hit: ' + request.url);
}

//Create a server
var server = http.createServer(handleRequest);

//Lets start our server
server.listen(PORT, function(){
    //Callback triggered when server is successfully listening. Hurray!
    console.log("Server listening on: http://localhost:%s", PORT);
});
```

Podcast of the Week

Shop Talk Show

An internet radio show about the internet starring Dave Rupert and Chris Coyier.

http://shoptalkshow.com/

Coding Tips

- Create a Real Programming Environment
- Make Programs From Scratch
- Start Small
- Write Lots of Code
- Ask for Help
- Ask for Help the Right Way

http://www.programmingforbeginnersbook.com/blog/when_you_know_the_basics_but_you_still_cant_code/