Friend Finder - Node and Express Servers

Overview

In this activity, you'll build a compatibility-based "FriendFinder" application -- basically a dating app. This full-stack site will take in results from your users' surveys, then compare their answers with those from other users. The app will then display the name and picture of the user with the best overall match.

You will use Express to handle routing. Make sure you deploy your app to Heroku so other users can fill it out.

Before You Begin

Check out this demo version of the site. Use this as a model for how we expect your assignment look and operate.

Create a folder called FriendFinder. Inside the folder, organize your directories so it matches the following:

FriendFinder

- .gitignore

- app

- data

- friends.js

- public

- home.html

- survey.html

- routing

- apiRoutes.js

- htmlRoutes.js

- node\_modules

- package.json

- server.js

Instructions

Your survey should have 10 questions of your choosing. Each answer should be on a scale of 1 to 5 based on how much the user agrees or disagrees with a question.

* Your server.js file should require the basic npm packages we've used in class: express and path.
* Your htmlRoutes.js file should include two routes:

d

* + A GET Route to /survey which should display the survey page.
  + A default, catch-all route that leads to home.html which displays the home page.
* Your apiRoutes.js file should contain two routes:
  + A GET route with the url /api/friends. This will be used to display a JSON of all possible friends.
  + A POST routes /api/friends. This will be used to handle incoming survey results. This route will also be used to handle the compatibility logic.
* You should save your application's data inside of app/data/friends.js as an array of objects. Each of these objects should roughly follow the format below.

{

"name":"Ahmed",

"photo":"https://media.licdn.com/mpr/mpr/shrinknp\_400\_400/p/6/005/064/1bd/3435aa3.jpg",

"scores":[

5,

1,

4,

4,

5,

1,

2,

5,

4,

1

]

}

Determine the user's most compatible friend using the following as a guide:

* Convert each user's results into a simple array of numbers (ex: [5, 1, 4, 4, 5, 1, 2, 5, 4, 1]).

With that done, compare the difference between current user's scores against those from other users, question by question. Add up the differences to calculate the total Difference.

Example:

User 1: [5, 1, 4, 4, 5, 1, 2, 5, 4, 1]

User 2: [3, 2, 6, 4, 5, 1, 2, 5, 4, 1]

Total Difference: 2 + 1 + 2 = 5

Remember to use the absolute value of the differences. Put another way: no negative solutions! Your app should calculate both 5-3 and 3-5 as 2, and so on.

The closest match will be the user with the least amount of difference.

Once you've found the current user's most compatible friend, display the result as a modal pop-up.

The modal should display both the name and picture of the closest match.

Reminder: Submission on BCS

Please submit both the deployed Heroku link to your homework AND the link to the Github Repository!

Minimum Requirements

Attempt to complete homework assignment as described in instructions. If unable to complete certain portions, please pseudocode these portions to describe what remains to be completed. Adding a README.md as well as adding this homework to your portfolio are required as well and more information can be found below. This assignment must be deployed.

Hosting on Heroku

Now that we have a backend to our applications, we use Heroku for hosting. Please note that while Heroku is free, it will request credit card information if you have more than 5 applications at a time or are adding a database.

Please see Heroku’s Account Verification Information for more details.

See the Supplemental Heroku Deployment Guide for in-detail deployment instructions.

Create a README.md

Add a README.md to your repository describing the project. Here are some resources for creating your README.md. Here are some resources to help you along the way:

About READMEs

Mastering Markdown

Add To Your Portfolio

After completing the homework please add the piece to your portfolio. Make sure to add a link to your updated portfolio in the comments section of your homework so the TAs can easily ensure you completed this step when they are grading the assignment. To receive an 'A' on any assignment, you must link to it from your portfolio.

One More Thing

If you have any questions about this project or the material we have covered, please post them in the community channels in slack so that your fellow developers can help you! If you're still having trouble, you can come to office hours for assistance from your instructor and TAs.

Good Luck!