**User Information App - Web Server**

Create a Node.js application that is the beginning of a user management system. Your users are all saved in a "users.json" file, and you can currently do the following:  
- search for users

Read the user.json file with fs. require (‘express’)  
If-statement (for loop)

Appending the file.

- add new users to your users file.  
- get your starter file here: [users.json](https://nycda.instructure.com/courses/32/files/3833/download?verifier=D9lxWjba5KZSx6F5YTe0mkbTW11aIuEnOSrT8Jpj&wrap=1" \o "users.json)[View in a new window](https://nycda.instructure.com/courses/32/files/3833/download?verifier=D9lxWjba5KZSx6F5YTe0mkbTW11aIuEnOSrT8Jpj&wrap=1)

**fs.readfile json**

**fs.writefile -> stringify**

**Part 0**Create one route:- route 1: renders a page that displays all your users.

Read the JSON file and renders the users on the route users.

**Part 1**Create two more routes:  
- route 2: renders a page that displays a form which is your search bar.  
- route 3: takes in the post request from your form, then displays matching users on a new page. Users should be matched based on whether either their first or last name contains the input string.

Render get. Form

Render post.Form

fs.appendFile('message.txt', 'data to append', function (err) {

if (err) throw err;

console.log('Saved!');

});

**Part 2**Create two more routes:  
- route 4: renders a page with a form with three inputs on it (first name, last name, and email) that allows you to add new users to the users.json file.  
- route 5: takes in the post request from the 'create user' form, then adds the user to the users.json file. Once that is complete, redirects to the route that displays all your users (from part 0).