

Project #3: Fun with Node.js

Prof. John P. Baugh – CIS 435 – Fall 2021

Points: _____ / 100

Instructions

You may optionally work with a group of up to 3 people. (Not 4. Not 5. Not 17. 3. Only up to 3.)

For this project, you will write a Web application, including:

- A visually attractive front-end with appropriate HTML, CSS
- JavaScript on the front end to ensure valid entries
 - Make sure the user at least enters something for their username and their note content
- JavaScript on the front end to communicate with the back-end
- A Node.js application on the back-end, exposing a REST API so that the front end can communicate with it

JavaScript in the browser is not capable of writing to files, or reading from arbitrary files. However, Node.js does have this capability.

Adding Notes

Your application must allow the users to **add** notes. Each user just has to enter a username (no password) into a text field (textbox) on the front end, and a short note into another text field (textbox). Then, when the user submits the note with their username, the Node.js application in the back should use the username to find the appropriate note (which will be labeled *somename.txt*, where *somename* is the username). If the note doesn't exist, Node.js will create one and enter the note. If it already exists, Node.js will append it to the file (or re-write the file with the new note added, which may be easier.)

View, Edit, and Remove Notes

The front end should allow the user to view the notes associated with a particular username. This should list them in some UI component such as a list, where the user may select one or more elements. The user may simply view the notes by scrolling through (as necessary). They may also select one or more elements for deletion or single note (at a time) for editing.

Editing should allow a user to load the contents of a single specified note, and edit the contents of that note. The user may cancel and leave the notes as is, or confirm the edit. The confirmed edit will be modified appropriate in any in-memory copies of the note data, as well as overwrite the contents in the note in the file system.

Deletion should ensure that any in-memory copies of the notes are updated for that user, and also should change the text files for that particular user.

Testing / Analysis / Design Requirements

1. You must include a test plan with **at least 3 days worth** of testing recorded
2. Screenshots of your application working (or not working), inserted into a PDF at the top level of your application

Deliverables

Please zip up the entire folder containing the appropriate HTML, CSS, JavaScript etc., also containing subfolders for things like scripts, styles, and images. Include your test plan and analysis document as well. Upload the zip file containing all these items to the appropriate project directory on Canvas.