Project Assignment
CSC 324: Web Programming

Your assignment is to build a web application with Node and Express.

Specifications

- Use Express to make the app.
- The application should consist of at least the following web pages:
  - Home
  - About
  - Contact
  - Login
  - Subscribe
  - 404.html
  - At least one page consisting of secure material, i.e., content that can be viewed only by authenticated users.
- Use EJS and partials to build the views for the pages, so that they have a fairly uniform appearance.
- At least one page should contain an element that contains randomized content (e.g., an image that changes randomly each time the user requests to see the page). The content should make sense, considering the purpose of the app. There should be at least three possible items, with each one have the same chance to be shown each time the page is visited.
- The Contact page should have a form that allows the user to send an email message to the owner of the web app, without knowing the owner's email address. (It will be sufficient for the email to go through Formspree.)
- There should be at least two user accounts:
  - A user with username "user" and password "pass". The role of this user should be "member".
  - A user with username "admin" and password "secret". The role of this user should be "administrator".
- The Login page should have a form with which a user can submit username and password. If the login succeeds, then the user can visit the page containing secure material. (If the user does not log in or if the login fails, then the user should be told to log in, in order to acquire access.)
- If the user is logged in as an administrator, then the secure material page should also show a button allowing the user to download the subscribers (first name, last name, and email) as a CSV file.
- On the Subscribe page the user (whether logged in or not) should be able to submit his or her first name, last name and email address. This information should be saved in a JSON file.
- If the user requests to see a page that does not exist on the site, then the 404.html page should be shown.

- CSS styling for the pages can be quite basic, but it should be sufficient for the user to be able to navigate the site and to understand all content.

Example App

An app that satisfies the foregoing specifications may be found at:

https://safe-retreat-54208.herokuapp.com/

Some remarks on the sample app:

- There are more pages than required.
- The styling is a good bit better than required, considering that CSC 324 does not deal with front-end web development.
- The secure material is found at Interact -> Members Only.
- The randomized content consists of the quotes next to the banner on the Home Page. (On smaller devices these will be hidden but can be displayed by clicking a "plus" icon near the banner.)

Requirements for Submission of the Project

- The source code for the project should be on a public GitHub repository, named "Project-CSC324".
- The project should be self-contained. This means that after someone downloads or clones the project, it he/she should be able to serve it on localhost:3000 with the standard commands:
    - npm install
    - npm start
- The project is due at the beginning of the final exam period. The final exam period is devoted to presentation of projects, so make sure to be present and prepared to show off your app to the class.

Extra Credit

For extra credit, deploy the project on Heroku, and provide the URL for the project in the README file in your repository.

Grading Rubrics

- *App Specs (80 points)*
    - Express used. (10)
    - Required pages present. (5)
    - Efficient and DRY templating with EJS. (10)
    - Randomized content is obvious to the user, makes sense for the site, and is based on at least three items. (10)

  - Contact page verifiably sends email. (10)
  - Subscribe page correctly adds to the subscriber JSON file. (10)
  - User accounts with required roles are present, and secure material is displayed according to member and administrator roles. App responds meaningfully to successful and failed logins. (5)
  - Access to distinct levels of secure content is correctly governed by the user’s role. (5)
  - Subscribers can be downloaded by the administrator as a CSV file, and the administrator’s session continues after the download. (5)
  - Site is organized and styled well enough to be navigated and understood. (10)
- *Submission Requirements (20 points)*
  - Properly named GitHub repository. (5)
  - App is self-contained. (15)
- *Extra Credit (5 points)*
  - App is deployed on Heroku, with URL specified in the README. (5)