Project 2 Grading

Open student’s submission in VS code. Install third party packages and start the app in the terminal.

Rubric 1 (10pts). Check the file structure to make sure that it follows the MVC pattern

- app.js: other than the error handlers, the app.js should not define any routes. Instead, it should mount two modules: main route and connection route (5pts)
    - Deduct 1pt for each route defined in app.js
- models folder - contain file for data objects and interface functions (1pt)
- views folder files (HTML, EJS) (1pt)
- public folder - contains CSS, im- contains front endages, and other static files (1pt)
- routes folder - contains route modules (1pts)
- controllers folder - contains controller modules (1pts)

Rubric 2 (14pts). Connection data

- Check the model in the models folder.
- A connection object is defined with **at least 7properties**. For example: (1) id, (2) title, (3) category, (4) details, (5) date and time, (6) location, (7) image (7pts)
    - Deduct 1pt per 1 missing
- At least 3 connection objects per category and at least 2 categories are created in the array. (5pts)
    - If there is only 1 category, deduct 3pts
    - If there are two categories, but less than 3 connections per category, deduct 1pt
- All connection objects are stored in a **single** array (2pts)

Rubric 3 (4pts). General site navigation

- in the browser, type: http://localhost:8084/, renders the landing page (1pt)
- click Image in the nav bar: renders the landing page (1pt)
- click about in the footer: renders the about page (1pt)
- click contact in the footer: renders the contact page (1pt)

Rubric 4 (15pts). Show all connections

- Click ‘connections’ in the nav bar renders the connections view (1pt)
- The page renders all connections in the database (6pts)
    - There should be 6 connections in total. Deduct 1pt per missing connection
- The connections are sorted by categories/topics. (8pts)

Rubric 5 (12pts). Create new connection

- Click “start a new connection” in the nav bar, the app renders the new connection page (1pt)
- in the form, fill in information for a new connection with a new topic/category (different from the two provided by the student), and click create. The app redirects to connections view (1pt)

- In the connections view, the new connection is displayed (2pts), the new topic/category is added and all connections are sorted by topics (3pts)
- print out the new connection in the console to make sure all information are stored properly (5pts)
    - deduct 1pt if any information is not stored properly.

Rubric 6 (12pts). Show connection details

- Click any of the connections, the app renders the connection detail view (1pt)
- connection information is complete, dynamic and correct (9pts)
    - all fields other than the id must be present, deduct 1pt for any information missing
    - check several connections to make sure all information is dynamic, if any information is static, i.e., displays the same information for different connections, deduct 1pt
    - if any information is not consistent with what is stored in the database, deduct 1pt
    - Give 0pt if no information is rendered
- edit button and delete button are present (2pts)

Rubric 7(12pts). Update a connection

- Click the edit button, renders the edit form (1pt)
- the form is pre-filled with information of the connection (5pts)
    - deduct 1pt per any information that is not correctly filled in the form
- Change multiple fields, and click submit
    - the app redirects to the connection detail page (1pt)
    - Check the connection detail page to make sure that all of the changes took effect (5pts)
        - deduct 1pt if a change did not take effect, or if any un-changed field got changed.

Rubric 8 (6pts). Delete a connection

- Click the delete button of the new connection you created: the app renders the connections page (1pt)
- The connection is removed from the list (2pts), the topic/category is removed and all the connections are sorted by the category (3pts)

Rubric 9 (12pts). Error handling

- All errors are handled consistently (should use the same error.ejs, but with different error code). (2pts)
- In case of a request for an unmatched endpoint, display a 404 error (2pts)
    - get /contacts
- In case of unmatched id for querying, updating and deleting, display a 404 error (8pts, 2pts each)
    - get /connections/100 (test in browser)
    - get /connections/100/edit (test in browser)

  - put /connections/100 (use postman)
  - delete /connections/100 (use postman)

Rubric 10 (8pts). Views

- Check the source code for ejs templates to make sure that partials are used for common elements, including headers and footers. (3pts)
- All elements are easily visible on the pages and fonts/colors used are easy to read and look at. (2pts)
- Layout looks good and consistent across all pages (3pts)

Rubric 11 (5pts). Documentation and submission

- zip file (1pt)
- pdf file answers all questions(4pts)
  - screenshots
  - Status
  - Easy part
  - Challenges