Adding Group Name field for todo entries

Implementer: Andrew Woll

API name: Express (in index.js)

Problems and Solutions

Problems

* Originally when entries are submitted they lack orginization and can only be retrieved collectively or individually.

Solutions

* Adding a group name field to the data sent to the server provides an additional ability of presenting entries which have been determined as related to each other.

Requirements (User Stories)

* Requirement statement: User entries must have the ability to be grouped together so that related entries are easier to keep track of.
* As a user, I want the ability to group entries together so that I have the ability to track related entries more efficiently.

Design

* Direction: Explain how information (data) flows in your feature.
* Input: Group tag
* Output: A “group”: field in the request data sent to the server
* Process: User inputs a group tag within a form input field within an HTML page. That tag is added as a data field in the data sent to the server to be stored.
* Diagram:



* Data Flow: The group tag will be created within the entry page as form data. That data is sent to the server as data and can be retrieved by users based on the group tags they’ve created. A separate HTML page will have the ability to sort shown entries by group tag.

Usage Example

* <http://localhost:5500/list/d> (This is an example, modify it)
* *Direction: Explain about this feature*





User Manuals

On "/write" users input entries with a group name. Upon submitting this data, this information has its own file within a MongoDB database with a { groupname: ":groupname" } field. "/write" additionally displays unique group names in a list as hyperlinks which are accessed from the database. Selecting one of these hyperlinks will redirect the user to "/list/:groupname" which displays entries of the selected group name.

Implementation Details

**/views/write.ejs**

Lines 38-52

Description: Creates a bootstrap card with a title "Created Groups:" and runs an ejs for loop to implement a list element for each item in the "groupnames" array sent by the "/write" API. Each list element containes a hyperlink with the name of the group which links to the "/list/:groupname" API to create a list.ejs page based on the selected group name.

**index.js:**

Lines 179-184

Description: API call for '/write' that searches 'post' database for unique 'groupname' values. It then renders 'write.ejs' with the result data.

Description: API call for '/add' with the only change being the addition of a "groupname" field within the stored database file based on the value of "req.body.groupname".

Line 203

Description: API call for '/list/:groupname' that finds entries with a "groupname" field that matches the value of "req.params.groupname". It then stores the database files in an array. This array of "posts" is used to render 'list.ejs' with the specified group name entries.

Lines 233-239

**views/list.ejs:**

Lines 36

Description: Inserts the value of res.params.groupname sent by the '/list/:groupname' API to list.ejs into a paragraph element in each instance of "posts".

Test

In order to test my API functionality, I had to utilize the Supertest Node.js library instead of jest so that I could export my API calls and test if the data flow I've created works correctly. supertestgn.js, testgnapp.js, and testgn.js act as my testing framework (located in the /tests/ folder in GitHub). supertestgn.js is where API calls are being checked for validity. testgnapp.js acts as the middleware for the test with testgn.js sending test data to be checked within supertestgn.js.

Deployment

* GitHub URL: [*https://github.com/jmlpamonag/ASE-285-final-project*](https://github.com/jmlpamonag/ASE-285-final-project)