

Documentation

Front-end

NOTES:

- The sorting of the topics/replies occurs after a refresh - this was decided to increase efficiency, while being reasonable

HTML:

The initial HTML in index.html contains the basic template for the website, the instructions, and the basic template for every post and comment. Creating a post/comment clones the template before filling in the relevant information, then inserting it in the appropriate div.

CSS:

We are using bootstrap.css, from twitter, to style our website.

We used a modal styled with bootstrap to display instructions.

Site specific styles are placed in style.css

Scripts:

The scripts that interact with the backend are as follows:

- *Posting a topic*
 - This is triggered by the "Post!" button; it dynamically creates a topic on the page, while sending the topic information to the backend to be retrieved at a later time
- *Posting a comment*
 - This is triggered by the "Submit" button; it dynamically creates a comment on the page, while sending information to the backend to be retrieved a later time
- *Posting a comment of a comment*
 - This is triggered by the "Submit" button; it dynamically creates a comment on the page, while sending information to the backend to be retrieved a later time. This is different from simply posting a comment as the information send to the backend must contain the topicID as well as the parent commentID
- *Putting a vote*
 - This is triggered by the arrow picture; it dynamically increases the vote, under the comment and the topic, while sending the information to be backend to be retrieved at a later time
- *Getting the topics and comments*
 - This is called once when the page loads (after a refresh); it retrieves all the posts/comments from the backend, sorts them by their votes, and outputs them on to the screen

The scripts that modify the HTML

- *Toggling the comments*

- This is called whenever the user clicks the topic title (or the “submit” button; this feature is to avoid issues with the toggle)
- *Toggle the reply box*
 - This is called whenever the user clicks “reply”; it either hides or shows the reply box based on its current state

Back-end

Models:

Topic- The topic model holds information for a topic object, it has the following attributes:

- title (given in post data from user input)
- link (given in post data from user input)
- comment (an array of its immediate comments)
- upvote (initialized as 0)
- timestamp (generated when created)
- numComments (initialized as 0)

These attributes has associated functions that are used to manipulate and alter them. This is stored in topic.js.

Comment- The comment model holds information for a comment object, it holds the following attributes:

- body (the body of the comment, given in post data)
- comment (an array containing its immediate comments)
- timestamp (generated when created)
- upvote (initialized as 0)

These attributes has associated functions that are used to manipulate and alter them. This is stored in comment.js.

Controller:

The server.js file routes each request by using regex; This mechanism was written in the function `serveRest`. Once the different url are distinguished, we then process it's method type, and further route each request into its corresponding handler. If the method type is post, it is sent to a `postHandlerMethod` to convert the querystring, and then sent to its corresponding handler.

The docstring above each handler explain what data is required, and what data are sent back to the client.