Documentation  
  
Yap Corner is a social media app that allows the users to create typed up posts (yaps). This app allows users to sign in and post yaps that are handled inside MongoDB

Work Completed:

* Schemes to keep track of posts and users
* Connected to MongoDB
* App uploaded on Heroku
* Basic controller, handlebar, and router implemented

To be completed:

* Add more handlebars
* Add CSS
* Error Handling
* Add premium functions
* Improve and add more functions (there are currently some issues and code fixes needed)

First I will focus on fixing the current code I have and will then start by creating different handlebars/pages. After creating those I will focus on GET and POST requests followed by styling. I will need to work on this throughout the week as I will need time to code, fix bugs, and especially stylize everything.

Endpoints:

URL: /signup  
Supported Methods: POST  
Middleware: None  
Body Params: username , password   
Description: Creates a new user in the MongoDB database  
Return Type: JSON message or error

URL: /log  
Supported Methods: GET  
Middleware: None  
Description: Logs the user out of their session  
Return Type: JSON message

URL: /login  
Supported Methods: POST  
Middleware: None  
Body Params: username , password  
Description: Verifies a user's login credentials  
Return Type: JSON with success or error message

URL: /createPost  
Supported Methods: POST  
Middleware: None  
Body Params: content , isPublic , userId  
Description: Creates a new yap post tied to the user  
Return Type: JSON message or error

URL: /getPosts  
Supported Methods: GET  
Middleware: None  
Query Params: userId   
Description: Returns all yaps from the given user  
Return Type: JSON array of posts

URL: /  
Supported Methods: GET  
Middleware: None  
Description: Renders the index page with forms for signup, login, and yap creation  
Return Type: HTML

URL: \*  
Supported Methods: GET  
Middleware: None  
Description: Renders the 404 page if no other route matches  
Return Type: HTML

React:

Currently I do have not established anything with react, but I plan to add components related to premium mechanics and displaying/containing/handling the posts.

Data:

* Posts (content,public,id,date)
* Users(name,password,premium)

Profit Model:

I have not implemented the profit model, but I will implement it by using the premium parameter in the User Schema that will allow a larger word limit when yapping.

Above and Beyond:

To go above and beyond my goal is to have really nice styling in order to create a satisfying user interface to make it feel like a real application people would use.

Borrowed code: I have been trying to have similar file, function, and variable naming conventions like the ones used in class. If this is supposed to be accounted for borrowed code then these naming conventions and structured functions are from class, primarily the simple models assignment.