Paul Stanton

CS-04605 – Advanced Web Programming

Final Project Part 2

07/16/23

**Backend Design**

This part will focus on the NodeJS design and implementation. For this part my goal is to accomplish the following items:

1. Come up with what endpoints I will want to create for my project.
2. Extend the NodeJS and express template to include swagger and TypeScript.
3. Stub out the endpoints needed without implementation. They will have implementation detail in part 3 and function once Mongo is connected.

**Endpoint Plan**

For my app I will need the ability to setup retrospective templates, retrospectives (Implementation of templates), and some level of user management. I will keep the endpoints relatively simple for now and expand on them if I have time.

1. Templates: A retro template is what retrospectives will be created from; Templates will need the following data:
   1. Name – Identifying template name (EX: Sailboat retrospective)
   2. Columns – These are where people will place their retro results, this will be an array with objects containing the following data:
      1. Name – Name of the column
      2. Position – Indicates which order the column should appear in
   3. ID – Unique ID for the template
   4. Created By – User who created the template.
2. Retrospectives: These will be specific implementations of the templates with the following data:
   1. Name – Independent of the template name (EX: CS04605 Retro 07/16/23)
   2. Template ID – Link back to the template
   3. Columns – Implementation of the template column with the following data
      1. Name – Name of the Column
      2. Position – Indicates which order the column should appear in
      3. Responses – Array with the following data
         1. Value – Text that was input into the column.
         2. User – Name of the user who entered the response.
         3. Position – Indicates which order the responses should appear in.
         4. Input Date – Date and time response was submitted.
      4. Comments – Comments that take place outside of the retro. Perhaps could be used as a chat. Array with the following data
         1. Comment – Value entered in as a comment.
         2. User – Name of the user who placed the comment.
         3. Input Date – Date Comment was submitted.
3. Users – For users I’m not going to store users locally, and instead utilize an external authentication provider called Auth0. User’s identities will be stored there, and so will token creation and retrieval. On my API I will just need to read from the token passed in for any user data.

**Project Init**

I want to setup the project with express and Typescript. Here is a general outline for the project:

* I’m not a fan of horizontal slice architecture, so it is more of a vertical slice. The organization structure is more feature based than technical based for the most part. I do not have a “controllers” folder or a “repository” folder, instead they are broken up into domain folders. For example, the folder “retrospectives” will house everything dealing with retrospectives, both db calls and controllers. I feel like it is much easier to support that way, otherwise you’ll have an endpoint spread across 5 different folders potentially.
* A screen shot of a computer

  Description automatically generated

**Endpoint Setup**

I’ll setup endpoints around Retrospectives and Retrospective Templates. Part 3 will contain the full implementation with MongoDB.

A screenshot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated