Project Name: PollingSurveyAppClient

Cohort #

Project Name Date: _____

Table of Contents

Project Name Date: _____

Mission Statement

This is where you write a short description on what your project is, the target audience, and the reasoning behind your choice of your project.

You work for a consulting company that just hired your team to build an internal polling tool that they can utilize to get information and opinions from the company's employees. Think things like:

User Stories

This is where you can work on developing a collection of user stories. These user stories should be planned through the perspective of your expected audience and each of the potential roles your end user may have.
Here's an example:

Project Name Date: _____

User Stories

-As a user, I want to be able to create an account, so I can use the application

- As a user, I want to see a list of the poll questions available, so I can vote on one

- As a user, I want to select a single item from the list of choices, so I can cast my vote

- As a user, I want to see the vote totals after I submit mine, so I can see the current results

- As an admin, I want to be able to create an account with admin permissions, so I can use the admin functionality of the application

- As an admin, I want to create a single response poll, so that users can only choose one option

- As an admin, I want to create a multi-response poll, so that users can choose more than one option

- As an admin, I want to be able to change poll questions from “draft” to “published” mode, so that I can control which polls are available for users to vote on

- As an admin, I want to be able to edit a poll, so I can make changes

- As an admin, I want to be able to delete a poll, so I can keep the polls organized and remove unneeded ones

- As an admin, I want to be able to see all the poll questions, so that I can know what is available for users to vote on

- As an admin, I want to be able to see the results of the polls, so I can know what our team members think

Project Name Date: _____

Bonus feature ideas

- Anything else you can think of that you want to add
- Allow admins to customize the color scheme of the poll questions
- Set up email confirmation for all authenticated users
- Add time-outs for polls
- Notify admins via email when someone votes on a poll
- Notify admins when a poll times-out

Database

This is where you write out what your database will look like. List out each table, the columns (include the dataTypes), and the database associations your project will have on the server-side.

Tables

Here's an example:

Users

ID	*FirstName*	*LastName*	*Email*	*Password*
num	*string*	*string*	*string*	*string*

Table 1

Table 2

Project Name Date: _____

Table 3

Additional Tables

Comments

- Id : string
- username: string
- Content:string

Endpoints

Here's an example of a collection of endpoints for a single table:

```
### Endpoints

* Account
   * GET api/Account/userInfo - Allows a registered user to login using a remote provider (google, iOs). The GET will require login to Provider before access.
   * POST api/Account/Logout - Allow user to Logout of account.
   * GET api/Account/ManageInfo?returnUrl={returnUrl}&generateState={generateState} - The Get is in relationship with GET api/Account/UserInfo
   * POST api/Account/ChangePassword - Allows account user password to be changed
   * POST api/Account/SetPassword - Setup for new user password
   * POST api/Account/AddExternalLogin - Generates external Access Token
   * POST api/Account/RemovelLogin - Allows for the removal of external LoginProvider and ProviderKey
   * GET api/Account/ExternalLogin?provider={provider}&error={error} - Will give an error if Provider keyin is wrong.
   * GET api/Account/ExternalLogins?returnUrl={returnUrl}&generateState={generateState} - Authernication of external token for login
   * POST api/Account/Register - Register account for new users
   * POST api/Account/RegisterExternal - using user's Email to register with external provider. generating user ID and Login.

* Department
   * GET api/Department - Internal use only. DO NOT USE

* Survey
   * POST api/Survey - Allows Admin to tag survey/polling to Data Table.
   * GET api/Survey - Will generate a table/list of all survey/polling input from POST
```

```
        * GET api/Survey?surveyId={surveyId} - Will produce survey/polling
register by ID generated from POST
        * PUT api/Survey - Allow's for Survey/Polling generated from GET ID to
edit and save changes
        * DELETE api/Survey?surveyId={surveyId} - Allow's for Survey/Polling
generated from GET ID to be deleted and removed.

    * Category
        * GET api/Category - Internal use only. DO NOT USE

    * Admin
        * POST api/Admin -  Create a new account user
        * GET api/Admin - Allows for table/list of all user accounts
        * GET api/Admin/{id} -  Will produce user account by POST generated ID
        * PUT api/Admin - Allow to edit user account by POST ID
        * Delete api/Admin/{id} - CURRENTLY NOT AVAILABLE
```

Features

Features are instances or examples of different pieces of functionality. This is where you list out the features you are planning on implementing. Consider the different steps and logic those features required to do the expected job. This could include fetching data from a 3rd party API or simply looping over data from your server. Differentiate between your version 1.0 or MVP (minimal viable product) and version 2.0 or stretch goals.

Version 1.0 / MVP	Version 2.0 / Stretch Goals
• You can register/login • Polls can be created • Users can respond to a poll • Users can view results of current poll after submission • Create navbar • Create footer • Styling exists • Create sign out screen	• Edit polls • Admin login • Admin can view past poll results • Sign In with google • Adapt for different screen sizes • Create thumb nav footer • forgot/change password

Project Name

Date: _____

Project Name Date: _____

Data Flow

When planning a project, we also need to consider what data needs to be available in certain components. Where do you need to declare your variable and which components feed into each other so that the value of the variable is accessible when you click here or navigate to this part of your project? Building out the data flow of your project will also be helpful when building out your folder structure. Take the time to draw out the structure of your project, where you will declare certain variables and build out pieces of logic, and trace the flow of where they will be utilized.

Project Name Date: _____

Wireframe

This is where you draw out a blueprint or a skeleton of the visual layout of the project. This is also known as the UI (User Interface) of a project. This can be drawn out with paper and pencil, whiteboards, and/or tools such as sketch, xd adobe, or balsamiq. Consider how the components will fit into each and what data needs to be passed between components. This is also a great place to think about adding a style guide and words or images that you are using for inspiration.

Project Name Date: _____

Schedule

This project is the equivalent of one sprint in the agile methodology. In this section, write out a schedule spanning over the next couple of weeks. This should include deployment, time set aside to tackle especially challenging features, testing, etc. The table below is a guide. It is not necessary to specify each day's work/logic. Feel free to estimate your time. Consider whether you will be working over the weekend(s).

Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
		Day 10	Day 11	Day 12	Day 13	Day 14
		Day 17	Day 18			

Project Name Date: _____

Final Notes

Great job with planning! You are now set to start coding. Planning a project is incredibly beneficial to your success and the success of your project. Here are some resources to help you with your planning.

- How to plan a web application
- Step By Step: Planning a web application

Friends list is many to many connection