Sprint 2 Plan, SlimeLine,
Completion Date: Feb 14, 2024
Revision 1: Jan 31, 2024

Goal:

Allow our user interface components to query the database and dynamically serve the website.

User Story 1

As a developer, I should be able to make filler posts for testing purposes

1. (TASK 1) Create a post button and have a form pop up. Filling out the form results in post appearing:
    a. Handle edge cases (i.e. ill formatted input)
    b. Fields for post:
        i. Title
        ii. Description
        iii. Image (optional)
        iv. email/account name
2. (TASK 2) Backend for posts:
    a. Store the post in the database (Club Name, Posts, Info)
    b. Load post from database, render it to the / path
3. Post fields:
    a.

User Story 2

As a user, I want to see what's new.

Note: Implementing the root path of the website

1. Read posts from database, dynamically serve it to the user on the root path

User Story 3

As a user, I want to see what I follow

1. Implement /follow route path
2. Hard code a post that shows something unique to that user account
    a. “HELLO your_username!”
3. Create a “My Profile” button that appears on all pages
    a. Will show these tabs when clicked:
        i. ~~Settings~~ - not really essential to product function
        ii. Following
        iii. User Info
        iv. Logout

User Story 4

As a user, I want to discover new clubs (Jack)

1. Redirect the user to the discover page by a button
2. Query the list of clubs from the database
3. Implement search feature on club list

Infrastructure

1. Firestore database setup
    a. Videos to learn about it, sample database setup
    b.

At the end of your team's sprint planning meeting, the team needs to turn in a sprint plan. This document needs to be typewritten (or the team needs to use a web-based agile planning tool and provide the TA/tutor access to the tool to view the project) and have the following elements:

- **Heading:** Document name ("Sprint {number} Plan"), product name, team name, sprint completion date, revision number & revision date.

- **Goal:** Allow our user interface components to dynamically serve the website.

- **Task listing, organized by user story:** This section lists the user stories, in priority order from

most important (top) to least important (bottom). Within each user story, there needs to be a list of tasks required to implement the user story, along with the time estimate for each tasks (preferably less than or equal to 6 ideal hours). This should look like:

User story 1 ("As a {user role}, I want {goal} [so that {reason}]")
Task 1 description (time estimate for task 1)
Task 2 description (time estimate for task 2)
...
Task N description (time estimate for task N of user story 1)
Total for user story 1: XX hours

User story 2
Task 1 description (time estimate for task 1)
Task 2 description (time estimate for task 2)
...
Task N description (time estimate for task N of user story 2)
Total for user story 2: YY hours

...

User story M
Task 1 description (time estimate for task 1)
Task 2 description (time estimate for task 2)
...
Task N description (time estimate for task N of user story M)
Total for user story M: ZZ hours

- **Team roles:** Give a listing of all team members. Next to the team member, list their role(s) for this sprint. Assign each person to at least one role (for example, this role might be "Developer"). This looks like:

  Team member 1: role 1 {, role 2, role 3}
  Team member 2: role 1 {, role 2, role 3}
  ...
  Team member N: role 1 {, role 2, role 3}
- **Initial task assignment:** A listing of each team member, with their first user story and task assignment. This should look like:

  Team member 1: user story, initial task
  Team member 2: user story, initial task
  ...
  Team member N: user story, initial task

- **Initial burnup chart:** A graph giving the initial burnup chart for this sprint and is labeled as such with sprint number and project name and is located in the lab.

- **Initial scrum board:** Also known as a task board, the scrum board is a physical board and labeled as such with sprint number and project name and located in the lab. This board has four columns, titled user stories, tasks not started, tasks in progress, and tasks completed. Index cards or post-it notes representing the user stories and the tasks for this sprint should be placed in the user stories, tasks not started, and tasks in progress columns. Tasks associated with a user story should be placed in the same row as the user story.
- **Scrum times:** List at least the three days and times during the week when your team will meet and conduct Scrum meetings. Also, indicate which of these meetings will have the TA/tutor visit as arranged with the TA/tutor. It is expected the TA/tutor will visit during the Scrum meeting during your lab time.

Note that if the team ended up modifying its release plan during sprint planning, submit an updated release plan document also with the sprint plan.

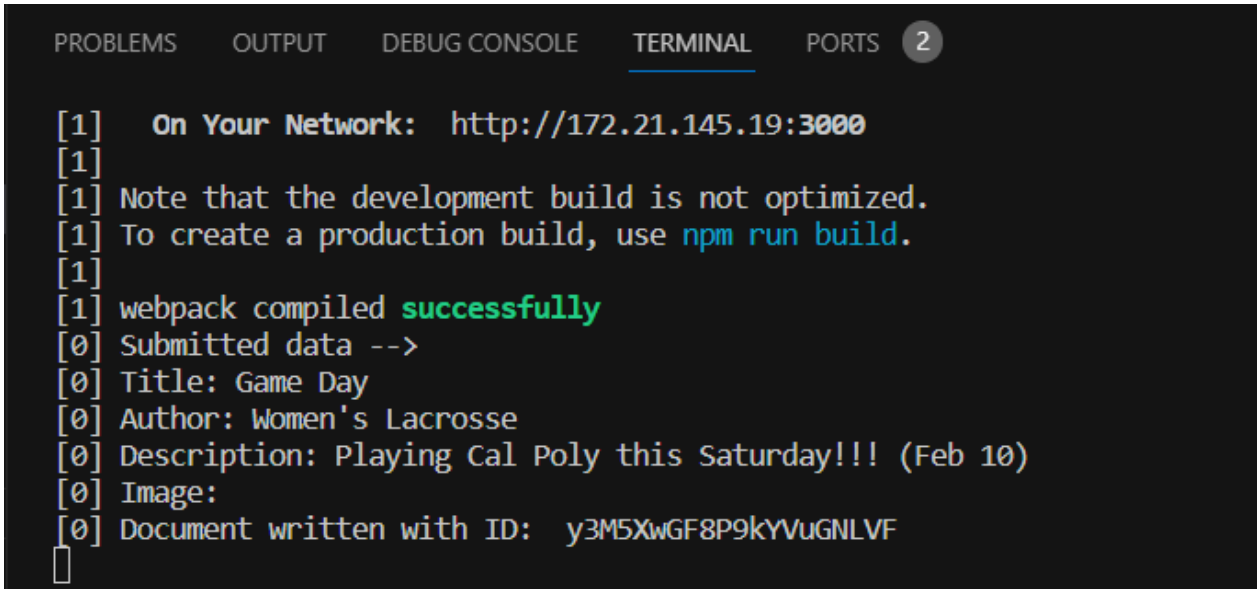
Last modified: 10/15/13 adapted from materials for cmps171

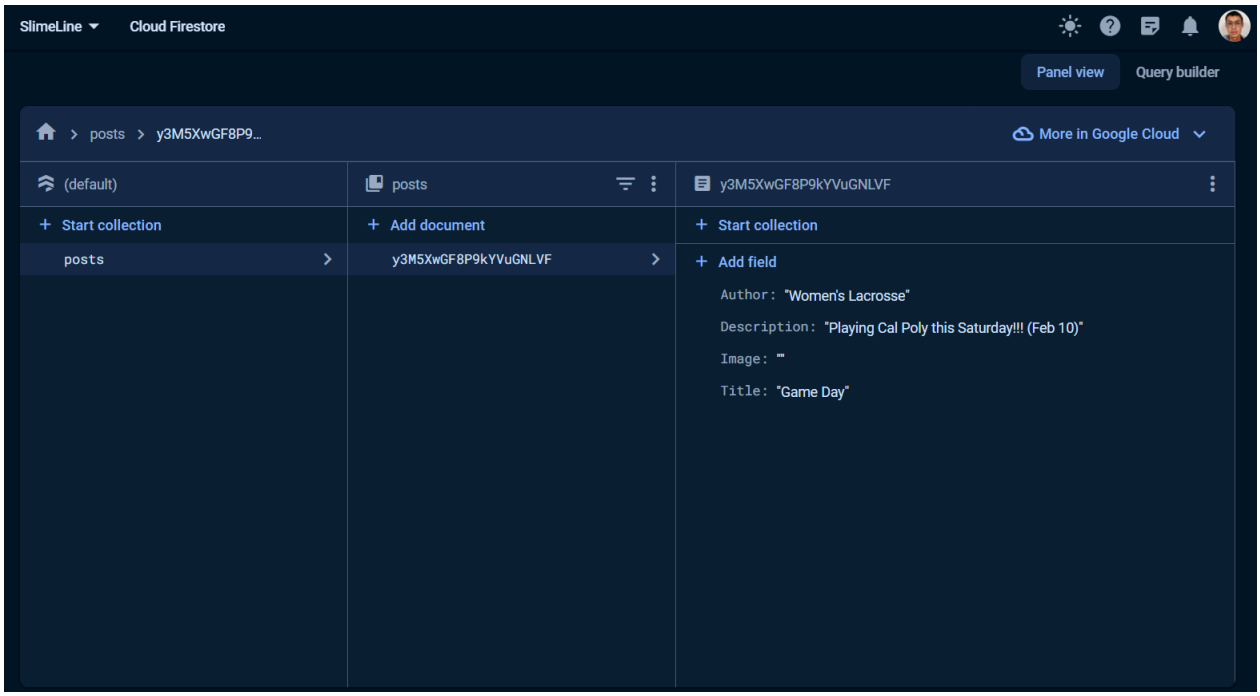
1, 2, 3, 5, 8, 13, 21

FireBase Database Setup

Post fields:

1. Title
2. Description
3. Author
4. Image




SlimeLine
Cloud Firestore
Panel view
Query builder
posts
y3M5XwGF8P9...
More in Google Cloud
(default)
posts
y3M5XwGF8P9kYVuGNLVF
Start collection
Add document
Start collection
posts
y3M5XwGF8P9kYVuGNLVF
Add field
Author: "Women's Lacrosse"
Description: "Playing Cal Poly this Saturday!!! (Feb 10)"
Image: ""
Title: "Game Day"