

- Feedback from the TA
 - Folder naming conventions
 - More detailed commit messages
 - New architecture diagram
- Post table + Post API
 - Post table
 - User id
 - Post ID is generated using guild
 - Post title
 - Post body
 - Time created
 - Upvotes
 - Downvotes
 - Diary entry
 - If true: anonymous
 - To make an API endpoint
 - Make PostController.cs in Controller folder
 - Make Post.cs model in Model folder
 - Make Posts.postman_collection.json in Postman folder
 - For pre-formatted JSON for request responses
 - API models will be generated from the back-end
- Go over dev tasks
 - Go over everything for our features for tomorrow.
 - If we have more dev tasks to add, just let Cynthia know and she'll add it into GitHub.
- Finalize time estimates
 - Let's do it online and then calculate our total time. If something seems off, bring it up.
- Go over the rubric for sprint 1 if there's time
 - Sequence diagrams to show the interaction behaviour of a system. Each core feature will have one.
 - Track features and bugs using the issue-tracking system.
 - Frequent meetings with meeting minutes available.
 - Project work - plenty of work done, and features released are complete or well documented as "functionality only" with the Project Owner's sign-off.
 - CI/CD pipeline is set up: auto build, code review, regression test executed when a commit is pushed to a major branch
 - Completed Worksheet. Well done, clear and concise (doesn't ramble on), included images were helpful.

- Install Postman to use the UI for managing the API gateways
 - Post the localhost URL for testing locally
 - The initial value should be for the test environment.
 - Import the environment
 - Then move to collections and import
 - The backend should interact with the collection of requests we're making.
 - Cognito will be connected with the API for Postman and return null if that user does not exist.
 - Make sure the route contains API when making APIs in the backend.
- App.Settings under Models are being used for which environment it uses.
 - The UserPoolId is very important for Jenkins
 - Useful for
- Common folder will have our objects where we can build tests from it afterwards.
- The controller can access the common folder and won't always need to make API requests, so make sure we change the format of the test to not include the API call.
- Let's prioritize the website and getting everything completed for that, and then getting the app crammed in as a last resort.