Project requirements

Goal: SnickerSync provides a user-friendly experience for syncing and comparing files, working well with GiggleGit to make the "snicker" feature intuitive.

Non-Goal: Syncing functionality and the user experience of "snickering" during syncs.

Non-functional requirement 1: Admins only has permission

Functional requirements:

- **Admin User Management**: Implement a role-based access control system that distinguishes between admin users and regular users within GiggleGit, allowing only admins to view and modify snickering settings.

- **Snicker Concept Editing**: Provide an interface for admins to create, edit, or delete snickering concepts, ensuring that non-admin users cannot access these functionalities.

Non-functional requirement 2: Random user assign to ensure unbiased data collection

Functional requirements:

- **User Study Group Assignment**: Develop a system that randomly assigns users to either the control group or the variant group (SnickerSync), logging the assignment for future analysis.

- **Study Data**: Create a function to export user study data including which group the user was assigned to, for post-study analysis.

Agile

Theme: Get GiggleGit demo into a stable enough alpha to start onboarding some adventurous clients

Epic: Onboarding experience

User story 1: As a vanilla git power-user who has never seen GiggleGit before, I want to understand how to perform the basic Git workflows (commit, branch, merge, etc.) within GiggleGit, so I can confidently adopt the tool without a steep learning curve.

User story 2: As a team lead onboarding an experienced GiggleGit user, I want to access a clear process for setting up a team workspace and manage permissions for my developers, so I can easily onboard my team and start using GiggleGit collaboratively.

User Story 3: As a developer who likes experimenting with new tools, I want to explore the unique features of GiggleGit, especially the "merges managed by memes" concept, so I can evaluate its usefulness and have fun while learning a new version control system.

Task 1: Create a tutorial for the meme-based merge feature

Ticket 1:

Title: Design the meme-based merge interface

Details: Define the layout and user interactions for selecting and applying memes during merges. Ensure it’s intuitive for users familiar with basic git merge conflicts and provides enough customization options to make the process enjoyable.

Ticket 2:

Title: Build the meme repository

Details: Develop a repository of memes that users can access when resolving merge conflicts. Include basic search functionality and categories for memes, allowing users to browse and select appropriate content during a merge.

DEPENDENCIES