PROJECT NAME
TEAM NAME: Chewing the Crud

Timeline [DAILY CHECK-INS: Straight after Standup, Straight after Lunch & Before EOP]

- Thursday 4th May 2PM (Step 4 Deadline // All Day)
- Friday 5th May (All Day)
- Saturday (Online from 12:00PM to 4:00PM but Flexible)
- Sunday (Flexible, on Slack)
- Monday (10AM -> 7 or 8PM with longer break in between or whatever works for each person)
- Tuesday (All Day)
- Wednesday 10th May 5PM (Recommended MVP Deadline // All Day)
- Thursday (Mock Presentation & Bugs, Changes etc // All Day)
- Friday 12th May 9AM (Deadline & Presentation)

Team

Hari Ramanathan(Git Maintainer), Franziska Kissling, Benjamin Khoury, Katie Loesch

Personal Strengths

Hari - Presenting
Fran - Presenting(Broad structure, not necessarily technical detail). Familiar with coding in the workplace., Happy to do any coding backend / front end.
Ben - Back-end and React
Katie - API queries, react-router

Personal Challenges (Could Use More Exposure)

Hari - Authentication, Front-end
Fran - Not sure about design skills.
Ben - CSS/ UI design
Katie - Authentication, presenting, responsive design

GROUP PLANNING

Team Goals & Values

- Functional Code
- Interactive, Attractive UI
- Thorough Communication Between Teammates

Team Communication Preferences

- Slack at any time; DND is on during sleep hours
- Respond within 1 hour.

CODING PRACTICES

Conventions:

Branch Naming

- main, dev;
- sub-branches use “initial-dev” convention, ie: *bk-feature-forms*
- *hotfix*, *feature*, *refactor*

Practices

1. **Never** code on, merge, or push to Main.
2. Git Maintainer will create a Dev branch and set the upstream so it's available to all group members.
3. Group members should run a *git checkout -b dev* and a *git pull origin dev* to link their remote branch to the origin branch.
4. Group members can then run *git checkout -b feature-branch-name* from their local Dev branch.
5. Make code additions.
6. Once a group member's code is complete and ready to merge with Dev,, that group member should run a "safety pull" from Dev to check for any updates made while you were coding.
7. If there are no conflicts, you can initiate a pull request on GitHub. Confirm the PR is pulling from your branch and comparing to Dev; title the PR "Brief Description/Keywords BRANCH NAME to DEV" This PR and merge will require the git maintainer's approval to finalize the merge.
8. Once that merge is finalised, all group members should add and commit their code to their local branch.
9. Then, they should run a *checkout* to their local Dev branch, followed by a *git pull origin dev*;
10. Then they should run a checkout to the feature branch they were working on and also run a *git pull origin dev*.
11. Continue coding on your feature branch. Return to step 6 as needed.
12. Once your Devt branch is ready to merge with Master, the Git Maintainer can make a PR from Development to Master. This PR requires at least 1 other group members' approval to finalise the merge.
13. Once that merge is finalised, all members should *pull origin master* to all local main and sub branches, if possible.