

## Meeting 11/14

**Attendance:** Colin, Jamie, Josh

Verified that we accomplished our independent research on previous targets.

Missing: Hamzeh

**Game plan:** By Sunday

Create rough UI(Colin), access the database chrome.storage.local \*save to/set\* (Josh), autofill data fields structure \*more robust\* (Jamie), \*undo/clear button (Josh)\*

\*Extra feature: "Undo" button that undoes the autofill, provides more user control\*

- You can use Git without Github, but you can't use Github without Git

Git

- Free, version control, open-source tool

Github

- Cloud-based platform that uses Git, security features

**ls** - view files in the folder you are in

**ls -a** - view hidden files too

**Git status** - tells which branch you are in and if there are untracked files or files to be committed

**Git add** - adds file(s) to staging area (preparing to be committed)

**Git add .** - adds all files to staging area, careful, check with **status**

**Git commit -m "COMMIT MESSAGE"** - commits the files in the staging area to the local rep

**Git push -u origin master** - pushes the local master branch to remote rep master branch

**Git pull** - fetches and merges new commits from remote rep to local branch (usually master)

**Git clone <url or something else>** - clones remote rep to directory, creating new local rep

**Clear** - clears the terminal

**Git branch** - lists all the branches

**Git branch <branch name>** - creates new branch

**Git checkout <branch name>** - switches to another branch

**Git log** - will show the log of commits, most recent on top

**Git fetch** - downloads new commits/files/refs from remote rep to local rep (doesn't merge)

**Git config --global user.name** - "NAME"

**Git config --global user.email** - "EMAIL"

^^USED TO TRACK COMMITS^^

Order of updating the main repository or a branch

Status (to check) -> add -> commit -> merge(if off main branch) -> push (approval step)

## Github

### **Do's**

- Consider using a .gitignore to avoid tracking unnecessary files in the rep
- CONFIRM WITH ALL MEMBERS BEFORE MAIN BRANCH CHANGES
- Descriptive commits with name+email
- Download something to help with Git, such as Git Bash and/or Github Desktop
- Synchronize regularly (git fetch - checks for new commits/files/etc)
- Fetch instead of pull when syncing

### **Don'ts**

- Right-click save as a file in order to save its "raw" file data
- Don't git pull, just git fetch when trying to sync, review changes first
- Change main branch without full consideration and consensus
- Any sensitive data (API keys/credentials) should NOT be committed
- Do NOT push to main branch directly, push to a different branch