# CVWO writeup

## Standard features

1. store data between sessions and devices, using a user account to login on any device
2. add tasks
    1. provide option of adding metadata like description, due date, urgency level 1-5
3. delete tasks
4. edit tasks
5. check tasks off as completed
6. search, find and replace tool, perhaps add Regex functionality
7. present tasks in a nice minimalist format with custom dark/light mode
8. cache previous edits in the session to undo tasks
9. history to view tasks in previous weeks
10. tagging + highlighting (different colours)
11. archive on-hold tasks
12. create method to organise tasks into different groups set in different tabs where other groups' tasks are not visible to reduce clutter (similar to archive)

## Additional features

1. drag and drop tasks to change priority of tasks dynamically
2. PWA with react native
3. reminders
4. spaced repetition system for reminders to revise certain chapters/tutorials
5. habit tracker
6. auto-repeat tasks of set frequency
7. auto-backup of database using CRON
8. API for other developers to perform CRUD operations from command line without accessing my application directly
9. Contribution.md
10. browser extension to add tasks more conveniently
    1. can add time tracker to view time spent on each task
11. Create project task type, add workflow description to each project task, only show the most current task in the project and show the next task after current task is done to reduce mental clutter (akin to GTD system)
    1. can also add notes section to store resources to read with respect to each task/task type/group akin to pocket
12. separate section to store follow-up tasks waiting on 3rd parties to respond to (GTD)
13. statistics to see number of tasks done in each task type
14. add images/screenshots
15. add notes section
    1. add code syntax highlighting
16. sync with google calender
17. ability to sync todo list with others, perhaps like a kanban board section
18. possibly sync with nusmods/luminus to track classes and exams
    1. add functionality to document query + regex details such as readings from module details tab
19. create more user-friendly method to bulk-add tasks at the beginning of the semester such as json or CSV
20. functionality to find other users with similar tasks/classes to link up

execution plan

1. create wireframe ?
2. create prototype in ruby ?
    1. Done 2-5
3. Learn ruby and rails (using the Odin project)
4. relearn react
5. create basic react prototype to do features 2-5
6. link react app to ruby on rails with dummy database
7. learn sqlite3
8. link ruby on rails with sqlite3 with dummy data for searching only
9. adjust to do CRUD but single user
10. add user management
11. sequentially complete the rest of the features
12. use finland MOOC to learn CSS properly, style the page better

Difficulties

1. being overwhelmed by the large workload involved in learning to do fullstack all at once
2. information overload where there are too many resources to look into with no guidance on which is best
3. juggling the project with school studies for the next semester