**Project Journal**

Meeting 1 - 05/10/23

* Initial project meeting
* Discussing initial project ideas and rough plan going forward

In between Meetings

* Researched possible projects to pursue – looked at how manageable and viable each idea would be as well as the difficulties that it would present. Also looked at where research areas would be

Meeting 2 - 11/10/2023

* Second meeting
* Further discussed project topics to refine ideas
* Talked about what was needed to be complete in order to be ready for the first submission during week 4

In Between Meetings:

12/10/2023

* Decided on the web browser option for the project
* Researched web crawling APIs/ ways in which I could make my own one that would give me the desired functionality for the web browser
* Played around in python with uses of a crawler and how it could be implemented successfully
* Started implementation of a basic search engine with simple UI and web crawler – user enters their search which is given to the web crawler that returns a list of all websites that it has crawled. Will need a lot more refining – ie only show user the relevant links as some don’t relate to the search.

15/10/2023:

* Created Gantt chart
* Planned ways in which web crawler will actually be used – will need to make a txt file that stores the URL and key words of a webpage. File will be searched when user enters a search rather than using web crawlers to find websites
* Will need to be able to efficiently search through the txt file to return results to user quickly – Maybe order websites by name and use a binary search?

17/20/2023:

* Put all current research into a word document called “research notes”

22/10/2023:

* Changed search approach – makes a lot more sense to run the web crawler independently so that it can make a ‘crawl list’ of ranked websites with their title and a list of keywords relating to them
* Then when the user enters a search via the search engine, the browser looks at entries in the crawl list that match the keywords
* Started implementation of this – was fairly simple to create the search functionality in the crawl list using dummy data ti simulate what would be in the file after the crawler had been run.
* Started off only having only single word searches but this isn’t realistic so changed search module to cope with looking at searches with multiple words – main premise is to split the search into a list of the individual words and search each of these words for matches in the saved websites
* Could also look at the string as a whole or a partial part of it and see if there’s a match there – this could be a way to judge if the website is a stronger match to the user’s search than the other websites in the list.
* Need to think of way to make the search more efficient – currently its fine with a small data set but most likely It will be slow when it has to deal with and search through a long list of websites that the crawler has made. Should be some good options but getting a working search engine that does everything I want it to do is more important at this stage.

Meeting 3 – 26/10/2023:

28/10/2023:

* Researched ways to summarise text/webpages
* Found the best method was through the use of PyTorch – a very good NLP model that I use to create a summary for the text that I extract from a webpage
* This summary is stored in the crawl list and is shown to the user along with the title when the list of websites are shown after a search.

29/10/2023

* Worked on ways to store the relevant information in the crawl list such that it is easily readable later on for the search engine – best way so far is a txt file that is ordered by title (ensure that any webpage with the same url isn’t included)
* Later down the line it might make sense to have different crawl lists relating to different topics so that the search engine doesn’t have to go through thousands (and potentially millions) of websites to return the list of relevant ones to the user – especially when most wont match the search and hence wont need to be returned
* Would need to make sure that I have a working way to choose the correct crawl list to search through otherwise lots of relevant websites wont be included in the list returned to the user – mostly likely use keywords to help make the decision.

30/10/2023:

* Completed functionality to read the data in the crawl list to display it to the user on the webpage – Title is a hyperlink and a summary is underneath that to give the user an overview of the webpage before they click on it
* Added to UI to accommodate these changes
* Thought about if I will need to include functionality that only loads the first x amount of webpages at a time to improve speed – makes sense as results will be returned faster because the search will stop sooner. But maybe not necessary for mine as the crawl list wont be huge and no server traffic as there will probably only be one use at a time. If I do implement this then I will need to try and figure out how I can rank all websites without looking at all of them – or is it a case of going through whole crawl list but not showing the user all of the results unless they want to see them all.

03/11/2023:

* Research – PageRank algorithm (Yu, Li and Zeng, 2021)

04/11/2023:

* Switched from using a txt file to store the website data to a db
* Using sqlite – mostly for convenience as its what I had downloaded already and have used before
* Have columns for:
* Title
* URL
* Keyword
* Summary
* Relevance score
* Page rank
* A composite primary key, consisting of url and keyword, is used to identify each individual entity in the database
* Wrote code to properly process the data once it has been read from the database – needed to create the original keyword list from the individual keywords taken
* Have to do the opposite of this when it comes to writing new data to the database in python

05/11/2023:

* Did some report stuff
* Literature review – web crawling, needs more
* Implementation – basic overview of how the web crawler and search engine will work but cant write too much because the method will mostly likely change
* But can talk about process – ie how I started out wanting to implement compared to how I actually implemented and the reasons why

13/11/2023:

* Researched cookies and their uses in webpages
* Implemented a basic webpage that stores and uses cookies to remember simple information about the user – currently only a template and doesn’t track data yet
* Integrated this functionality in main program
* Added a login option as well that also makes use of cookies – made a new database to store users of the system, containing: first name, surname, username, password, date of birth

15/11/2023:

* Added register new user option – user enters name, username, password and dob and this is stored in a database called ‘users’
* Added settings popup – currently only have the option to delete your account

22/11/2023:

* Worked on project report
* Literature review almost finished – needs a little more detail and potentially more sections to add for the AI implementations
* Design sections completed up to the current state/planning of project – search engine, crawler, databases (that have currently been made and put into use) are websites, users and search history
* Implementation sections started – will add/change parts as I progress but np way to know the state of final implementation yet

27/11/2023:

* Added more functionality to search engine
* User can now view their search history and delete any entries that they want – adding this required that search history is tracked, so when the user makes a search, the results page is told if the user is logged in and will save any website visits to the search history database if possible
* Added extra functionality with cookies – user can accept or decline the use of cookies when they first visit the website. Theres always the use of the essential cookie “visited before” that makes note of if the user has visited the website before and so will not ask for cookie permissions again. Also have a cookie that specifies if the user allows the use of non-essential cookies in order to track their search history and other useful user data.
* Privacy settings option added to the settings menu that allows the user to change personal info and their cookie preferences. (Front end only)

01/12/2023:

* Implemented functionality for a user to change their personal info by querying database to change an entry in column ‘columnName’ linked to their username

Need to ask at next meeting:

Can only track the initial website that a user goes to from my webpage – after that theres no way because ive not coded the other websites – how big a deal is it for the project

Maybe use a webview??? Show a webpage within my own so then I can keep track of user activity?

28/12/2023:

* Found that the Hugging Face Library used to summarise text from webpages changed the way in which it required me to specify my input – now need to give a specific pipeline in order to get the most out of the summariser. Download new json files as well
* Had to research and fix this problem – aim is to make code more robust so that it can cope with a change from the API if it happens again.

Christmas Holiday:

Completed the January Deliverable:

* Literature review could be directly taken from report as all research is relevant
* Introduction is simplified but covered mostly the same points
* Description of progress made is a brief overview of work completed – more a description of what the programs do rather than how they do it, which is what is described in the final report
* Remaining work is highlighted and will form the basis for my plan going forward from here but will not be included in the final report because the work will be completed and written about in the future