SYSTEM REQUIREMENTS DOCUMENT

Functional Requirements

Feature Name: Website Downloader		Priority: High
Description: Uses HTTrack to visit a website and download data related to the webpage. This is a command line tool which the URL is passed into and the resulting output is the html and css along with other source files that make up the website. These files are stored in a specific directory which a program can move or zip the files so the next processes can clean and database the files associated with the URL		
Inputs: A URL of the website the user wants to save.	Outputs: Zip/source HTML	Processes: It would run in the background and capture the HTML and source code to display that webpage.
Dependencies: HTTrack and whether the website can be saved or not. The websites that we foresee not being able to download would be websites that contain a large amount of data, websites that need WiFi access to function (i.e. a WiFi-based game or homework solver), etc.		

Feature Name: User Interface to download website		Priority: High
Description: An HTML page with a text box for the user to enter a website to be saved. Clicking the Save button will download the website using the downloader tool. It will act as the front-end main page to the user.		
Inputs: URL from the user	Outputs: text string of the URL	Processes: The user chooses a website to download, types into the text box, and selects download. The submit button will send the URL string to the downloader to process and download the page
Dependencies: Javascript, HTML, and CSS		

Feature Name: Saved website database		Priority: High
Description: Store the directories of the downloaded websites onto MySQL to allow for easy retrieval when the user wants to access their saved web pages. The users will be able to title their saved websites and may categorize them as well.		
Inputs: Website name (as SQL key). This can be saved on a dedicated web page that	Outputs: Directories for the necessary webpage to be loaded for the user	Processes: Once the user requests a web page to be loaded, an SQL query will run

has a dynamic list of previously saved web pages. The selected page would act as the input key		with the required input, returning the directory of the saved page so it can be loaded into the browser for the user
Dependencies: MySQL, a dynamic list of previously saved web pages stored on the website, a script that adds website directories to the database as they are saved by the user		

Feature Name: Web page to access previously saved websites		Priority: High
Description: A web page or list that holds all websites saved by the user. The user should be able to select saved websites from here or delete websites. This list would be dynamic based on saved websites. This page should be accessible to the user from the main page of the website. Here, they may also delete or rename their website. In addition, users will be able to access the hyperlink to the actual version of the website.		
Inputs: Website downloader tool and directory of the saved website	Outputs: Dynamic list of saved website	Processes: When a website is saved, an SQL query runs to add the website to the user's database of saved websites. The database will store the website name and directory of the saved site for future access. A script would then run to add that website to the dynamic list, which can be accessed by the user later
Dependencies: Local Webserver, SQL database		

Feature Name: Current Download Status		Priority: High
Description: Once a user enters a URL and a title (if they choose to add one), they will be able to see whether their website has saved or not. There will be a queue if the user enters another website while the current website is still downloading. There will be no limit to this queue.		
Inputs: Website URL	Outputs: A textbox underneath the URL entering box that will display the titles that are currently downloading/queued.	Processes: Once HTTrack begins downloading the website, the UI will display that the website is saving. Once HTTrack is finished, the UI will display that the

		website is finished saving by moving it out of the queue and into the saved websites list.
Dependencies: Being able to sync the data of the website download process from HTTrack.		

Feature Name: Search Feature		Priority: Medium
Description: The search feature will allow users to enter words into a text box. Here, the user can search for a title and relevant titles of saved websites will appear.		
Inputs: A text box that the user clicks and enters words that they would like to search the titles for.	Outputs: A relevant list of websites that contain relevant saved website titles to the search query.	Processes: Using MySQL to retrieve titles that contain the string that was inputted. If the string inputted wasn't exactly like a title, it will bring up titles that contain the words that were searched. Ex: the website title given by the user is "Chocolate Chip Cookie Recipe" and the user searches "Chocolate Recipe", their chocolate chip cookie recipe will still appear, so they don't have to enter the exact title.
Dependencies: The user must have websites stored.		

Feature Name: Auto Save via Chrome Extension		Priority: Medium
Description: A Chrome extension that passes the URL of the current site into the system to be downloaded. In the Chrome extension, you will add a title labeling the saved website. This data will then be displayed in the UI.		
Inputs: Button clicks from the user in manual mode. The user will also enter a title.	Outputs: It stores files on the users system that represent the save state of each website.	Processes: It runs in the background and stores each webpage as its loaded in the browser.
Dependencies: It has to be installed in chrome or a browser based on chrome such as brave.		

Feature Name: Sorting Feature	Priority: Medium-Low

Description: The sorting feature that we add will allow the user to be able to sort their cached websites by features such as sorting alphabetically and sorting from most recently saved.		
Inputs: A drop down box that is clicked and displays the sorting options, from which a user could choose one and then receive a sorted list.	Outputs: A sorted list of cached websites based on the sorting method chosen.	Processes: Using MySQL “ORDER BY” feature to get the ordered list of data.
Dependencies: The user must have websites stored.		

Feature Name: Local communication between multiple users		Priority: Low
Description: Users on the same network will be able to see other users' local saved bookmarks listed under that user in the bookmarks plane, this way bookmarks can be locally and privately shared so they aren't publicly available to stay within legal boundaries. This feature would send out a request on the network to see what other users are using the software and whether they want to share their bookmarks or not. If they want to share then the data can be accessed on their computer from the user’s computer.		
Inputs: Responses from other users on the network	Outputs: Other users' bookmarks in other users' bookmark tab	Processes: Background process that scans for friends that use the same software
Dependencies: Special network packet and port to be open on users computer for communication		

Feature Name: AI search		Priority: Very low
Description: Somehow use AI to read all saved websites and search for targeted queries. Its basically a fancy search engine. This will display a list of the relevant websites saved and perhaps a snippet of the content if the words appear within the saved website.		
Inputs: A prompt or search from the user into a text box	Outputs: Relevant saved files from the users' hard drive.	Processes: AI would somehow run when called and search the given documents. This would probably work through an API call. If they want it to work offline, this gets a lot more complicated because now you have to use local models. That would require tensorflow or similar.
Dependencies: Tensorflow if local, an API call if not local. It would likely need a paid account		

with chatGPT or some similar service to perform the search. This would require the user to make a paid account and register an API key.

Feature Name: Download Queue		Priority: Low
Description: Adds a waiting line when downloading websites to download one at a time		
Inputs: URL links	Outputs: One URL at a time to HTTrack	Processes: Last in first out queue
Dependencies: URL chrome extension output, a queue structure		

Non-Functional Requirements

Quality Attributes		
Feature Name	Description	Priority
Reliability	Low rate of failure for download completion (<2%)	High
Usability	Low number of steps for users to complete to download a website: paste a URL, add a title, add category (optional), hit submit Since not every website will be downloadable, we are aiming to be able to download approximately 60% of the websites that have been entered.	Medium
Efficiency	Very efficient due to the low number of processes, will not fail to download, but may not produce usable data depending on the website	Medium
Portability	Want to have it run on every device with 100% success of installation of usage	Low
Security	Stored locally so it is 99% safe with exceptions of physically compromised or hacked devices	Low

System Performance		
Feature Name	Description	Priority
Response time	It doesn't have to be that responsive when downloading the websites	High

	taking up to 30 minutes per website, it should show the downloaded webpage within 10 seconds due to the data being stored locally Response time for displaying downloaded content and website URL once tapping on the title should be instantaneous.	
Throughput	This system will be limited to downloading one at a time to maximize efficiency, but there is a queue. The amount of data moved will be dependent on the website downloaded.	High
Resource utilization	Load only the page the user wants to see instead of loading up all websites downloaded by the user	Low

Constraints		
Feature Name	Description	Priority
Software dependencies	Java Script HTTrack MySQL Google Chrome (or related like Edge or Brave) if using the browser extension	High
Regulatory requirements	Downloaded data can not be public	High
Hardware limitations	How many pages you can hold will be dependent on the amount of storage on your device The download speed will be dependent on your device configuration	N/A

Related Requirements	
Requirements	Dependencies
Reliability, usability, and hardware limitations	All of these requirements depends on how fast HTTrack is able to download the website or even if it can download the website.
Response time and secruity	Users hardware, depending on if ther user has a hard drive or a solid state drive data retrieval could vary widely. If the user has an insecure machine the response time could also vary and make it more prone to data theft.