Progress Report 2

https://cosc4p02.tpgc.me/

Members:

Maulik Mann	6994214	mm20as@brocku.ca
Adrian Binu	6970677	ab19xe@brocku.ca
Thanushan Pirapakaran	6890206	tp19uy@brocku.ca
Vishant Khamar	7062920	vk20if@brocku.ca
Devaan Zastrow	6932842	dz19xr@brocku.ca
Michael Boulos	6973523	mb19ep@brocku.ca
Jason Hunter	6744247	jh19pp@brocku.ca
Brett Terpstra	6920201	bt19ex@brocku.ca

PROJECT NAME	Web Summarizer and URL Shortener
PROJECT MANAGER	Maulik
SCRUM MASTER	Vishant

START DATE	END DATE	OVERALL PROGRESS
01-29	04-28	≈70%

TASK NAME	FEATURE TYPE	RESPONSIBLE	START	FINISH	DURATION in days	STATUS	COMMENTS
SPRINT 1			01-29	02-14	17		
Deploy Website		Brett and Michael	01-29	02-14	17	Complete	
Create URL Shortener		Maulik	01-29	02-14	17	Complete	URL shortener created, adding to website pending
Use LLM to summarize		Brett, Michael, Deevan, Vishant	01-29	02-14	17	Complete	LLM is currently being trained and adjusted, accuracy of the summarization provided by the LLM is a challenge
Create UI		Jason, Adrian, Thanushan	01-29	02-14	17	Complete	
SPRINT 2			02-14	02-27	14		
Log in with various methods		Maulik	02-14	02-27	14	Complete	
Pro User Dashboard		Jason, Adrian, Thanushan	02-14	02-27	14	In Progress	
Incorporate Web Summarizer into site		Brett, Michael, Deevan, Vishant	02-14	02-27	14	In Progress	Difficulty with implementing the web summarizer into the site
Incorporate URL shortener into site		Maulik	02-14	02-27	14	Complete	Difficulty with implementing the url redirection on site
SPRINT 3			02-27	03-12	15		
Pro Features: Summarization/Url Shortener history, etc.		Jason, Adrian, Thanushan, Maulik	02-27	03-12	15	In Progress	
Pro feature: Summarization Adjustment		Brett, Michael, Deevan, Vishant	02-27	03-12	15	Not Started	
SPRINT 4			03-12	03-26	15		
Dark Theme for site		Jason, Adrian, Thanushan, Maulik	03-12	03-26	15	Complete	
Text-to-speech for summarization		Brett, Michael, Deevan, Vishant	03-12	03-26	15	Complete	

Web Team Log:

Login: Since the last progress report the Logging in and authentication has been improved, there are 3 more ways to log in now alongside Google: Twitter/X, Github and Yahoo. Firebase was used to implement the logging in functionality and to keep track if the user is logged in, with this we can save Summarization and URL shortened history based on the user and manipulate content on the site based on if the user is logged in or not. The user can now also sign out.

URL Shortener: Completely finished now, it is implemented into the site. The url shortener saves all shortened links into the MariaDB in our server alongside the FireBase userId (if the user is logged in), the amount of times the shortened url is clicked and the date the url was created is also kept. The URL shortener is now being run as a service on the server.

Dashboard: The UI of the dashboard is still in progress but the content for the URL shortener is there. If a user is logged in they can view the dashboard to see the last 5 urls they have shortened along with other data. The Dashboard is not viewable unless you are signed in. The web summarizer data in the dashboard is yet to be implemented.

LLM Team Log:

Brett / System Administrator's Note:

Access to the underlying host which manages the shared VM and executes the LLM will not be granted to anyone under any circumstances. Access to the cosc4p02 VM can be granted to anyone involved in the project, including stakeholders (the TA), but I am unwilling to compromise the security of my hardware.

Summarization: Our LLM is still an operational service on the backend; the primary component produces acceptably consistent summaries given a Url to a webpage or hosted video.

We use Python's requests library with BeautifulSoup4 to parse through webpages and extract all text.

We added a new component to preprocess web pages using statistical methods that reduces the fetched text into usable prompt data. By shortening webpage content into the statistically most significant portion and using that as input we get better responses from the LLM. We do this by taking measurements of the word count per line in the site. Websites with low variance in line word count length are generally left alone, as these sites usually only contain relevant information the user wants summarized. The example test which sparked this consideration was the input of Linux man pages, university professor websites, and blog posts written in an incoherent style. Specifically, the cutoff for low variance sites is -1σ, meaning lines with length less then mean word count - 1σ are not considered by the LLM. For sites with high variances, we set cutoff at +1σ. This has shown to be remarkably successful for modern blog/news style sites where there are lots of short extraneous bits of irrelevant information such as links to other stories or site headers.

Result: LLM output is more consistent and takes less time because of the reduced prompt size and more syntactically pure input.

Saving Summaries: Summaries generated by the LLM for specific webpages are now saved on the backend MySQL Database, to remove redundancy of having to regenerate summaries for the same links and to provide a way to allow users to retrieve previous summarization entries.

Summarization is fully functional for the following

- **Text-based websites**
- **Video-based websites**

Mixed sites are not supported, if the site has a video that is downloaded, it will not attempt to summarize any text from the site.

- We use yt-dlp to extra videos from sites, the large extensive list of supported sites is available here raw video files are supported as well.
- OpenAI's small whisper model is used to extra the transcript of videos. This allows for support of more than just YouTube videos.

Summarizer Access: Both summarizer for webpages/videos and previous summary retrieval (through the project database) are accessible using separate server endpoints; backend is complete. A frontend template for summarization was developed by the LLM team to make it easier for the front-end team to develop a styled web-frontend that can interact with summarizer endpoints.

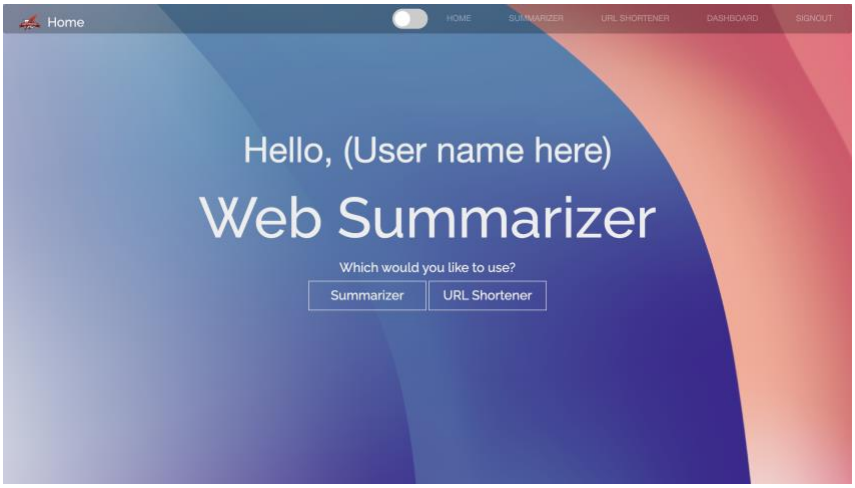
Challenges

- LLM takes forever to run so we needed a non-blocking way of handling client / server interactions.
- Running the LLM is a blocking action, but the webserver needs to not block
    - Python's concurrency model is interesting...
- Frontend integration with backend; assigning team members to work on just the integration of the frontend to the backend would fix this.
- LLM lacks configuration features currently – these can be added through prompt tweaking -- as there needs to be more concrete discussion regarding the required final pro features.

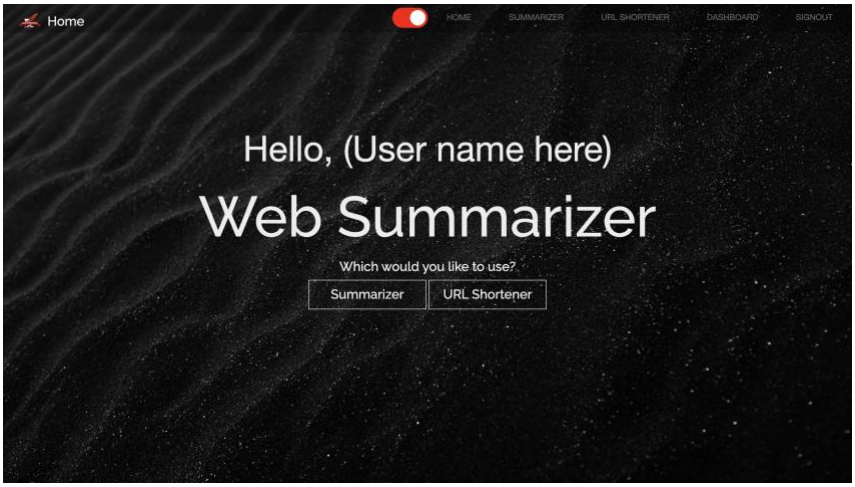
All Backend changes were pushed to the GitHub by Brett following each weekly meeting; project remote repository should reflect up-to-date server scripts.

Screenshots:

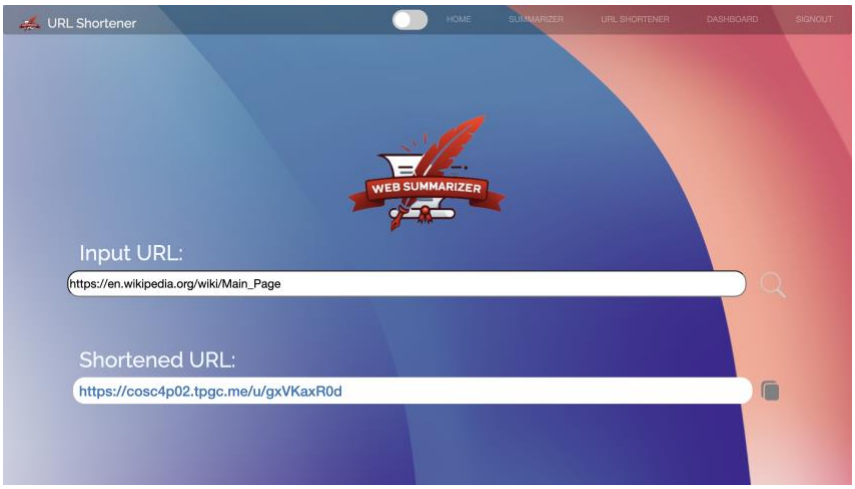
Home Page:



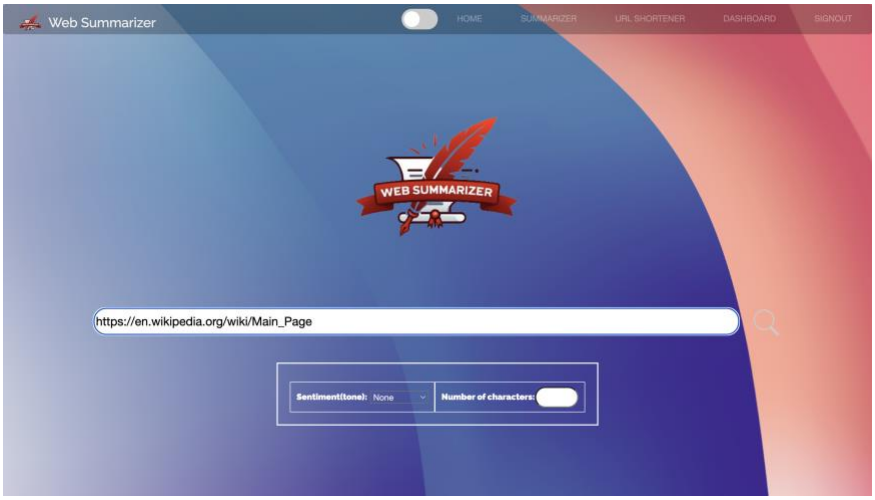
Home Page (Dark Mode):



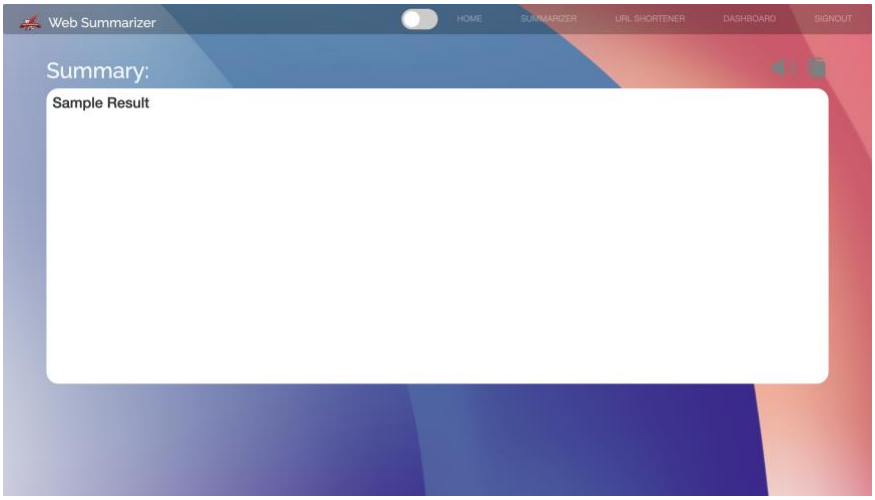
URL Shortener Page:



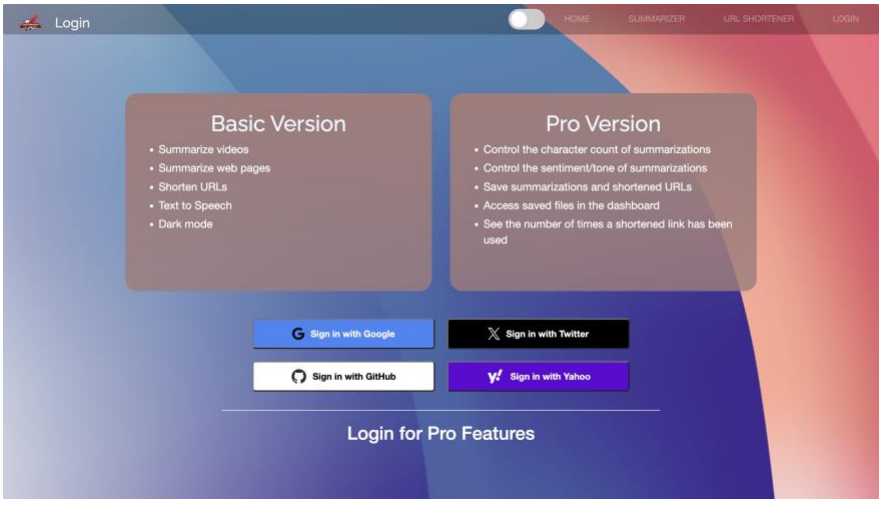
Web Summarizer Page:



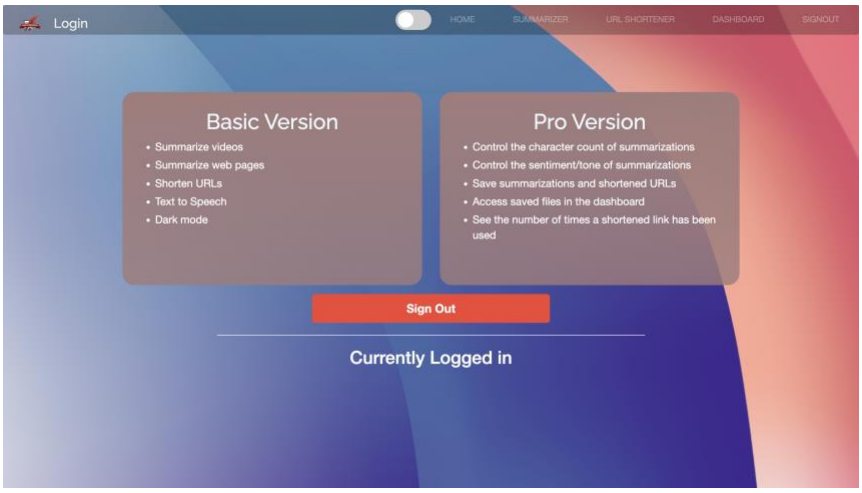
Web Summarizer (Result):



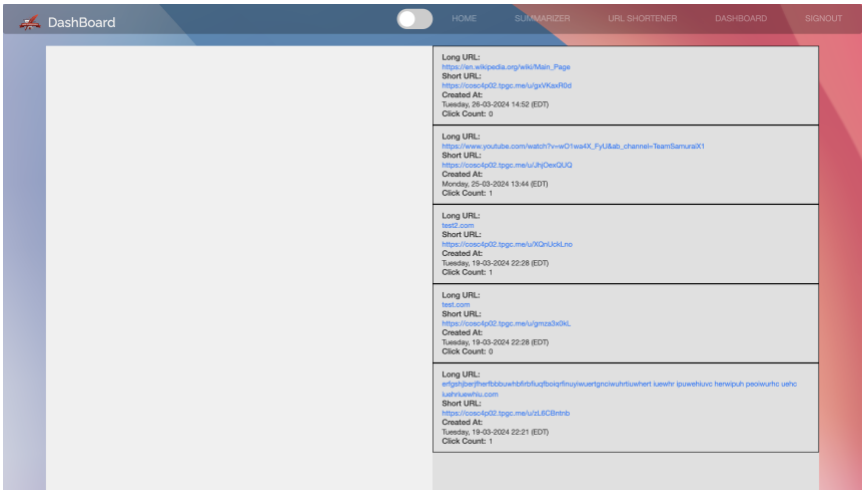
Login Page (if user is not logged in):



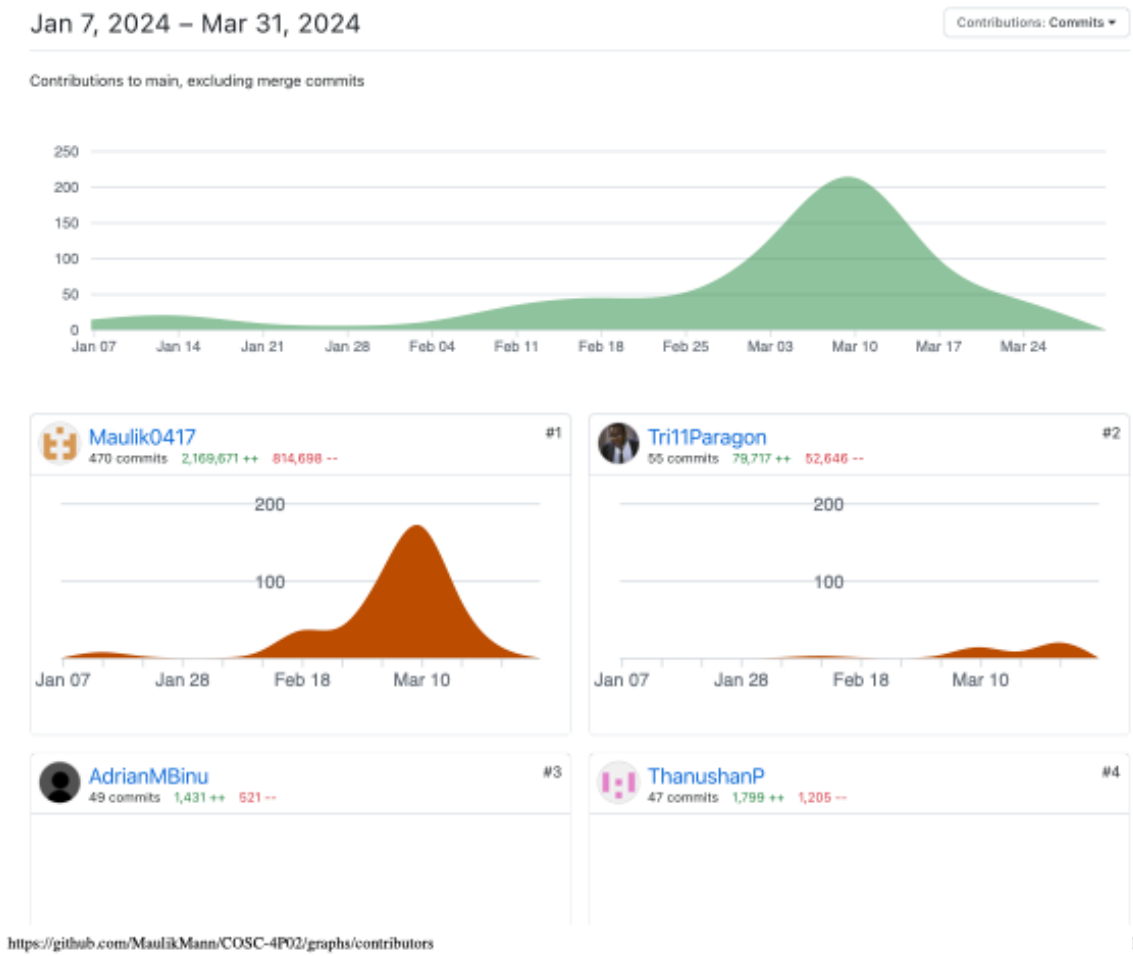
Login Page (if user is logged in):

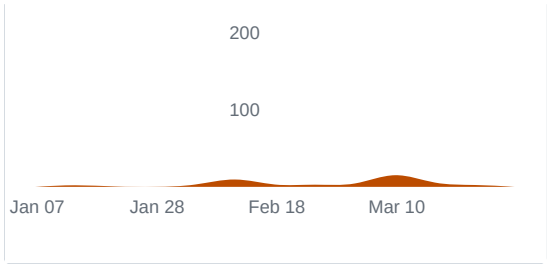


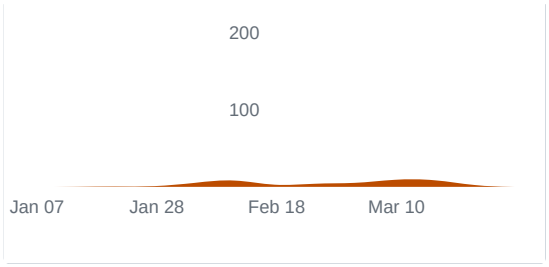
Dashboard Page:



GitHub contribution log:





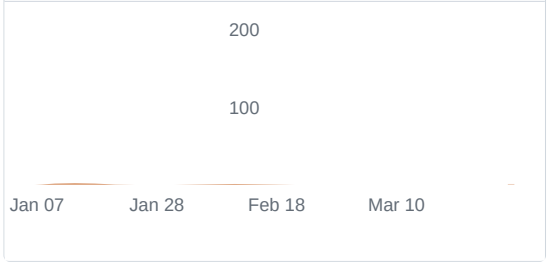


MaulikMann #5
22 commits 59,716 ++ 69 --

JasonHTD #6
21 commits 191,092 ++ 60,618 --

vnkhamar10 #7
12 commits 121 ++ 0 --

MikeBlu #8
3 commits 8 ++ 0 --



DZastrow #9
1 commit 1 ++ 0 --

Code frequency over the history of **MaulikMann/COSC-4P02**