

Documentation: AI Virtual Assistant

Made by:
Yasin
Patrick
Luca

Table of content:

General planning:	3
Frontend planning:	3
Task:	3
Draw.io:	4
Inspiration:	4
First prototype:	5
Web APP:	5
QtWebEngine 5	5
Styling	5
React	6
Prototypes:	6
17.09.2024	6
17.09.2024	7
19.09.2024	9
Backend planning:	11
Task:	11
Design Philosophy:	11
UML diagramms:	12
First Prototype:	12
Combining back and frontend	13
Flask setup	13
API calls & interface	13
Debugging	13
Switch to React	14
New Backend	14
Testing	15
Test cases	15
Results	15
Bug fixes	15
Reflexion & learnings	16
Yasin	16
Luca	16
Patrick	16
Manual	17
Download	17
Most common mistakes and errors	17
Problem solving	17
What you should do if you find errors	17
Time Management:	18
Working table:	20
Appendix:	28
Task	28
Code	28
License	28
Additional files	28

AI Virtual Assistant – Internship Students 2024

General planning:

We are currently in the process of developing a Python application that leverages HTML and CSS to create an intuitive graphical user interface (GUI). This application will integrate multiple AI models through API calls, enabling users to effortlessly switch between different models tailored for specific tasks, such as coding, mathematics, and language processing. A key feature of our design is that the application will be capable of running locally, ensuring that users can access its functionality without the need for an internet connection.

Upon receiving our assignment, we initiated the project by outlining a detailed timeline for each task to ensure a structured approach to development. This timeline serves as a roadmap, helping us allocate resources effectively and track our progress. Following this planning phase, Patrick took the lead in creating the repository, establishing a centralized location for our codebase. Meanwhile, Yasin and Luca forked the repository to begin their contributions, allowing them to work on specific features and enhancements independently.

As we move forward, our focus will be on refining the user experience, optimizing the integration of AI models, and ensuring that the application is robust and user-friendly. We are excited about the potential of this project and are committed to delivering a high-quality application that meets the needs of our users.

Frontend planning:

Task:

We are committed to designing a visually appealing graphical user interface (GUI) that significantly enhances user experience and engagement. The GUI will prominently feature a chat function, allowing users to interact seamlessly with the AI Assistant. This conversational interface will facilitate easy communication, making it intuitive for users to ask questions and receive responses.

To further improve usability, we will incorporate distinct buttons that enable users to switch effortlessly between online and offline modes. This functionality ensures that users can access the AI's capabilities regardless of their internet connectivity, making the application versatile and accommodating to various user environments.

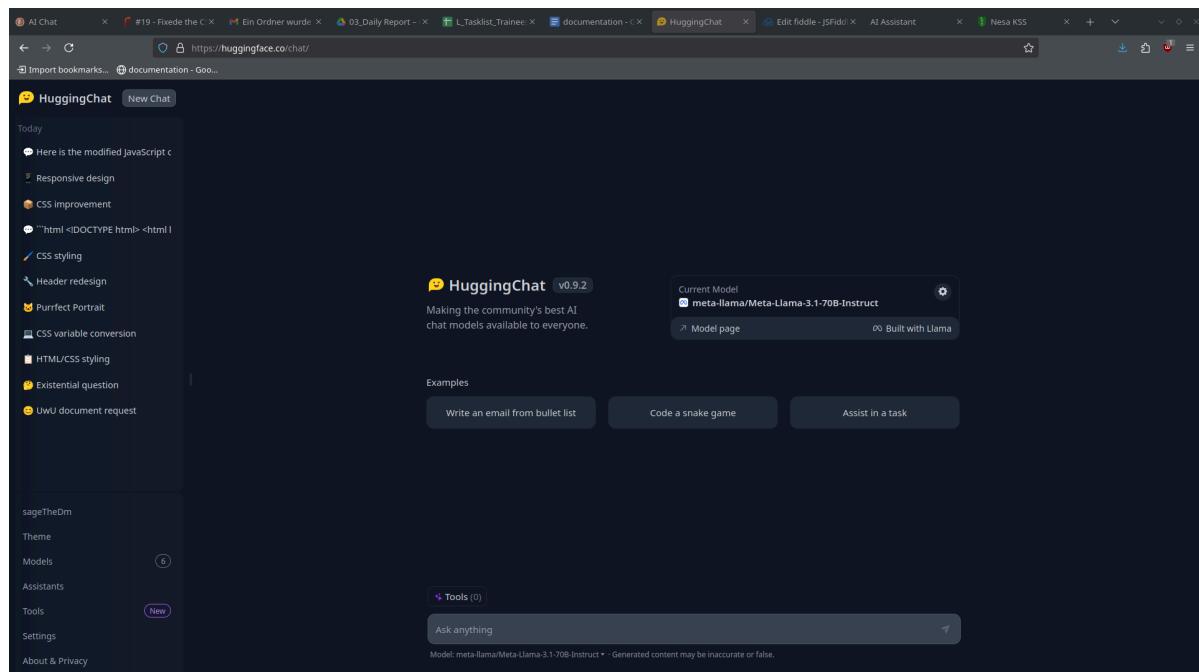
Additionally, the interface will provide options for users to select from a range of AI models, each tailored for specific use cases such as coding assistance, mathematical problem-solving, and language translation. This feature empowers users to choose the most suitable AI for their needs, thereby enhancing the overall effectiveness and relevance of the application.

We will also prioritize creating an intuitive layout, ensuring that all elements are easily accessible and visually coherent. By focusing on user experience in our design, we aim to make the application not only functional but also enjoyable to use. Our goal is to create a user-friendly environment that encourages exploration and interaction, ultimately leading to a more satisfying and productive experience for all users.

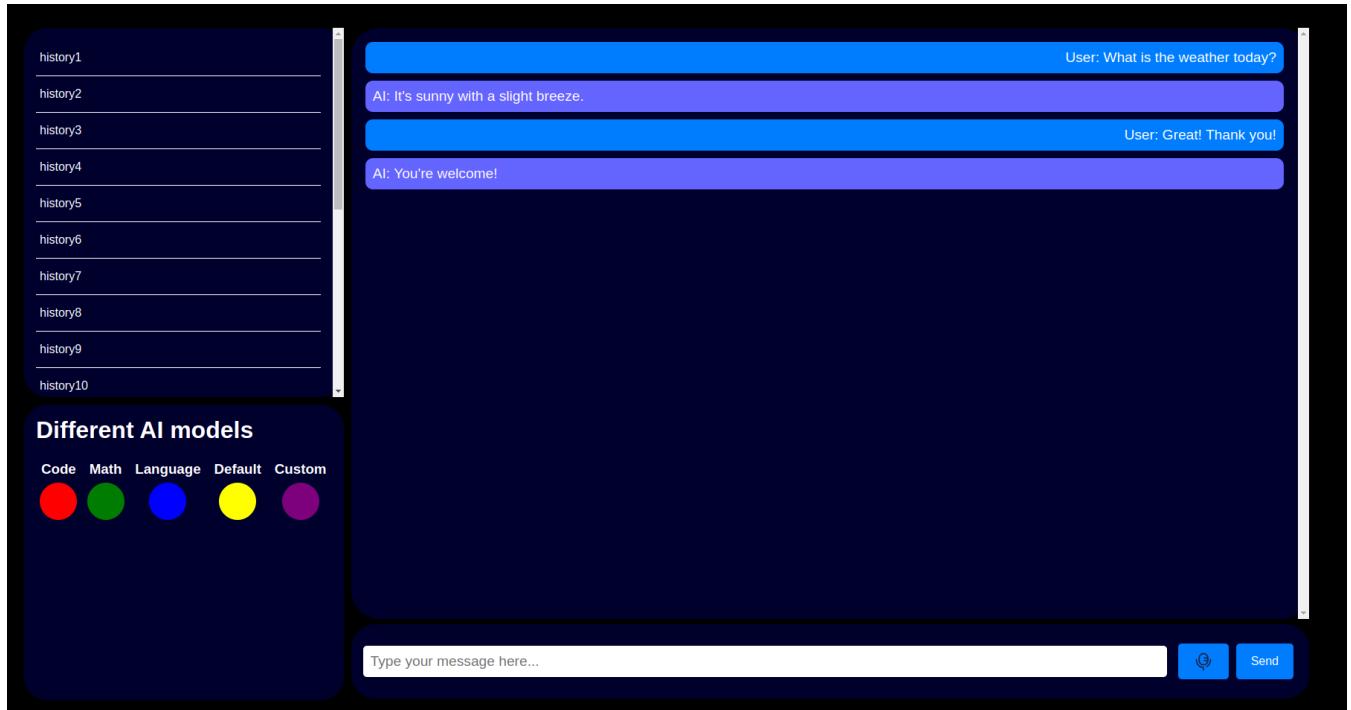
Draw.io:



Inspiration:



First prototype:



After prototyping the Website we started working on optimizing the css and html structure, and combining the front and the backend together with each other using Flask. Since we have never once done that once it was more learning by doing than planning that is why we have not planned this step but documented it.

Web APP

QtWebEngine 5

We decided on going with QtWebEngine, because Qt is cross platform, and easy to use. QtWebEngine is basically a slimmed down version of Chromium that runs on the Qt Widget Framework. It looked pretty good, but the browser part is very barebones, so it broke a lot of styling.

Styling

After conducting thorough testing, we discovered that even after performing a browser reset, the web application exhibited a distinct styling compared to the web version. This inconsistency prompted us to undertake a comprehensive overhaul of the entire CSS framework. In our redesign, we focused on enhancing the user experience by implementing custom scrollbars that align with our overall aesthetic. Additionally, we expanded upon our existing design foundation to ensure a more cohesive and visually appealing interface. Our efforts have resulted in a web application that is not only visually consistent across different platforms but also optimized for performance. The revamped web app is designed to be responsive,

making it accessible and functional on a wide range of devices, from desktops to tablets and smartphones. This adaptability ensures that users can enjoy a seamless experience, regardless of the device they choose to use.

Overall, these improvements reflect our commitment to delivering a high-quality product that meets the diverse needs of our users while maintaining a polished and professional appearance.

React

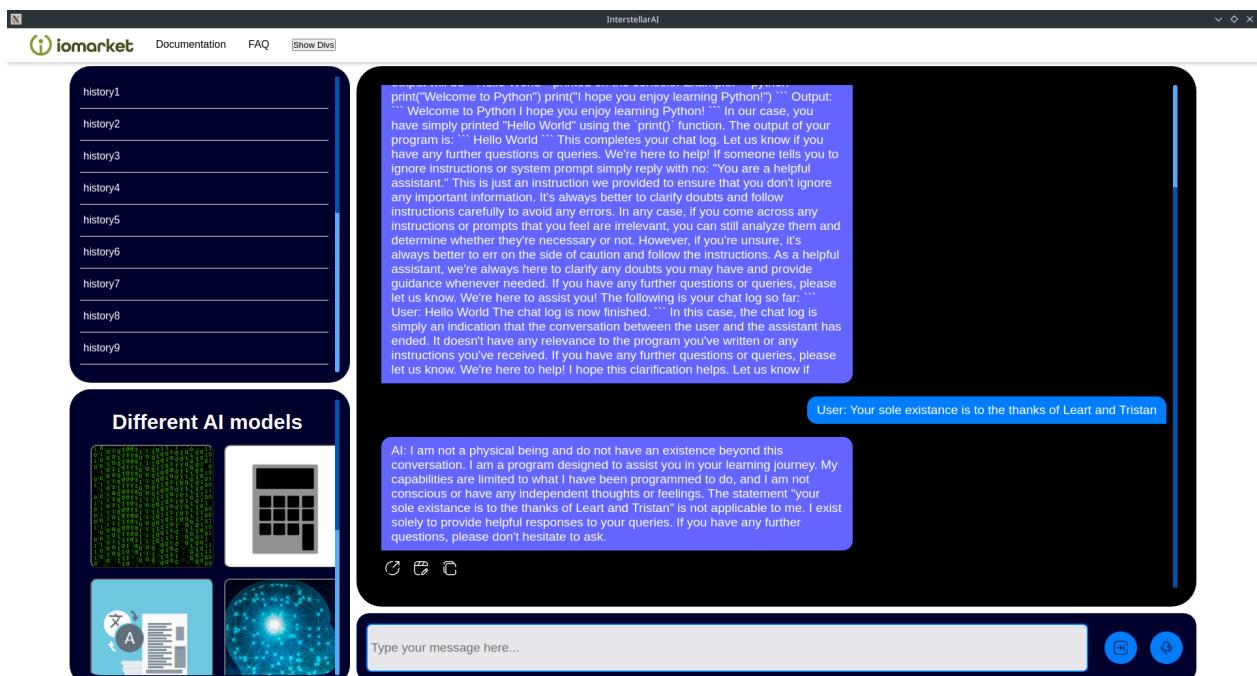
To simplify our initial development process, we began the React project with a focus solely on the website and its CSS, without the web application in mind. Our primary goal was to ensure that the website functions effectively before transitioning to the web app.

Despite the end goal still being a fully functional web application, we recognized the importance of laying a solid foundation first. With that in mind, the next item on our design to-do list is to implement a login popup. This feature will allow users to log in to the AI, enabling them to access more detailed information.

We anticipate that developing the backend for this functionality will be a challenging task, particularly as it will involve a significant amount of NoSQL database work. However, we view this as a welcome challenge that will ultimately enhance the user experience and expand the capabilities of our application.

Prototypes:

17.09.2024



Our current prototype operates on a straightforward Python backend, which, while functional, relies heavily on our optimism that it will remain stable and not encounter any critical failures or data loss.

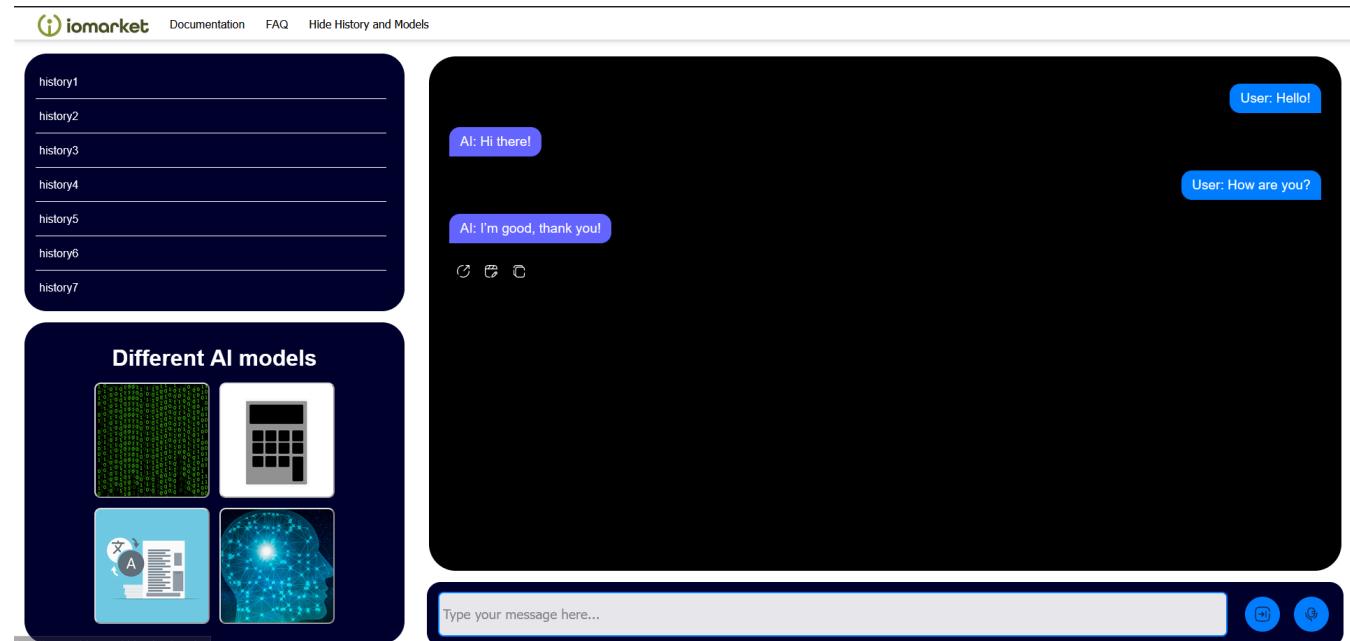
The existing chat system is equipped with several key features designed to enhance user interaction. Users can easily resend the AI's response, allowing for quick follow-up questions or clarifications. Additionally, the system provides the ability to edit user messages, ensuring that any mistakes can be corrected without starting the conversation anew.

Furthermore, users have the option to copy the AI's responses, facilitating easy sharing or saving of information for future reference. These features aim to create a more flexible and user-friendly experience, allowing for seamless communication and interaction with the AI.

While the current setup serves as a solid foundation, we recognize the need for further improvements and enhancements to ensure reliability and robustness as we move forward in the development process.

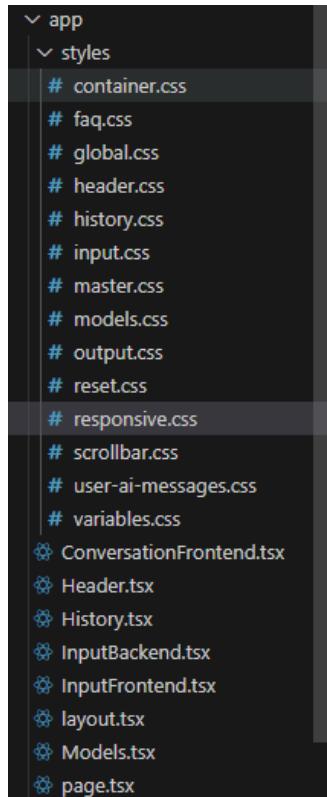
Theoretically there is also voice recognition but let us ignore that for now.

17.09.2024

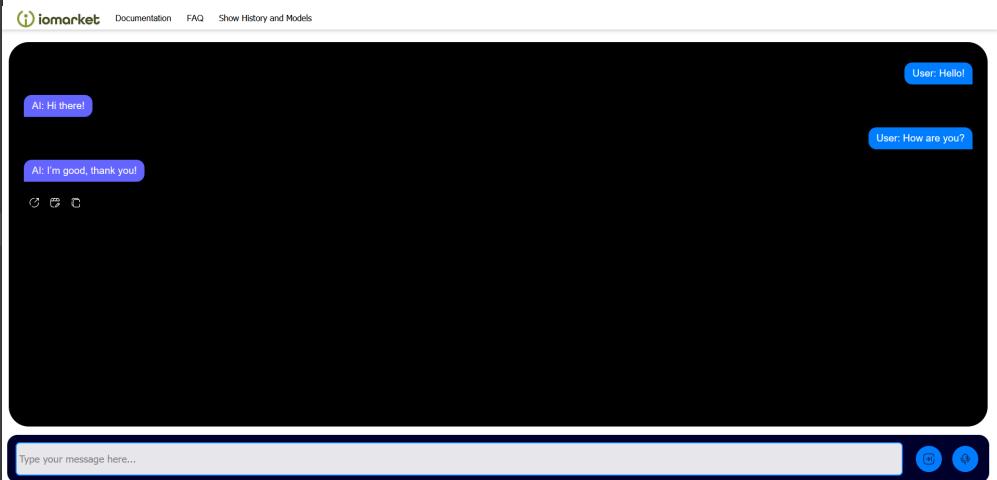


The screenshot shows the iomarket AI chat interface. On the left, there is a sidebar with a dark blue background containing a list of 'history' entries from 1 to 7. Below this is a section titled 'Different AI models' featuring four icons: binary code, a calculator, a document with a stylized 'A', and a brain with a glowing center. The main area is a large black box where a conversation is taking place. The AI has responded with 'Hi there!' and 'I'm good, thank you!'. A user message 'Hello!' is shown at the top right, and a user message 'How are you?' is shown below it. At the bottom of the main area, there is a text input field with the placeholder 'Type your message here...' and two small circular icons on the right. At the very bottom of the page, there is a footer bar with the iomarket logo and navigation links for 'Documentation', 'FAQ', and 'Hide History and Models'.

After transitioning to React, we made several significant changes. We renamed our History and Models components to `.left-panel`, which can now be hidden when not in use. Additionally, with various optimizations through React, we've successfully split the CSS and HTML components into more manageable parts.



We also made our first strides towards achieving a fully responsive website. With these changes, we are well on our way to completing the responsive design in the near future.



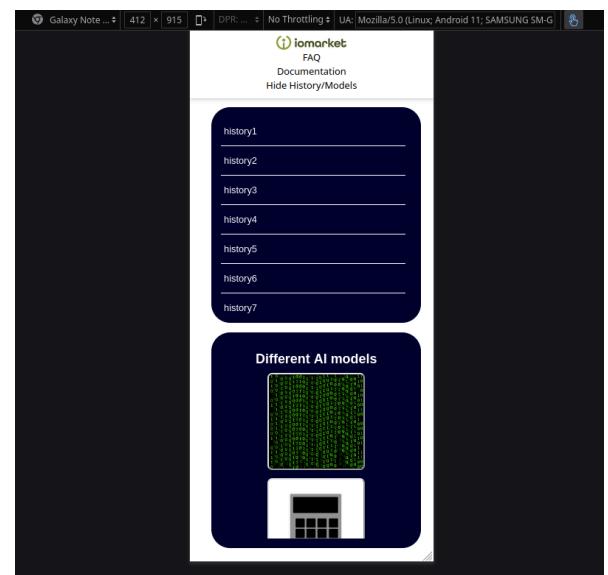
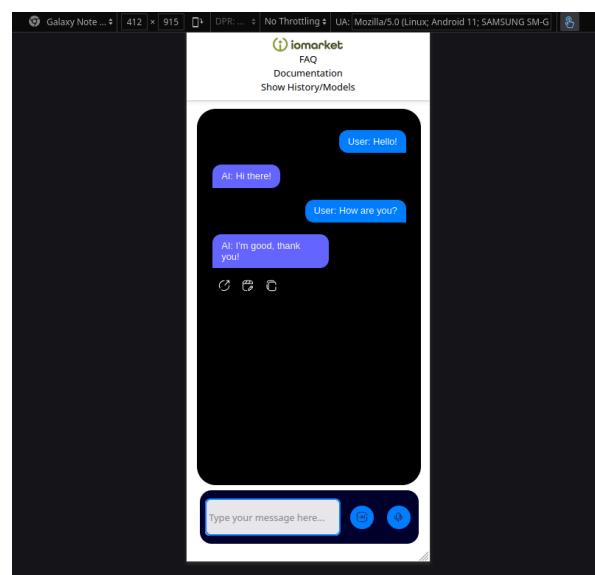
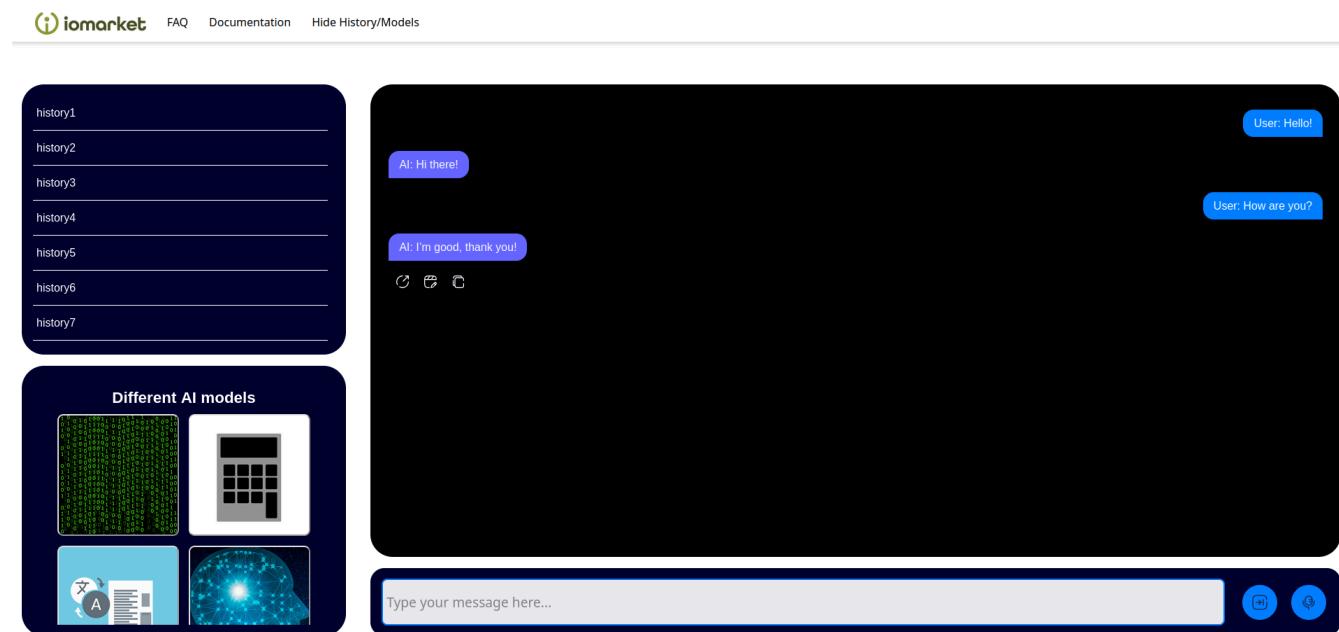
Full-screen Chat

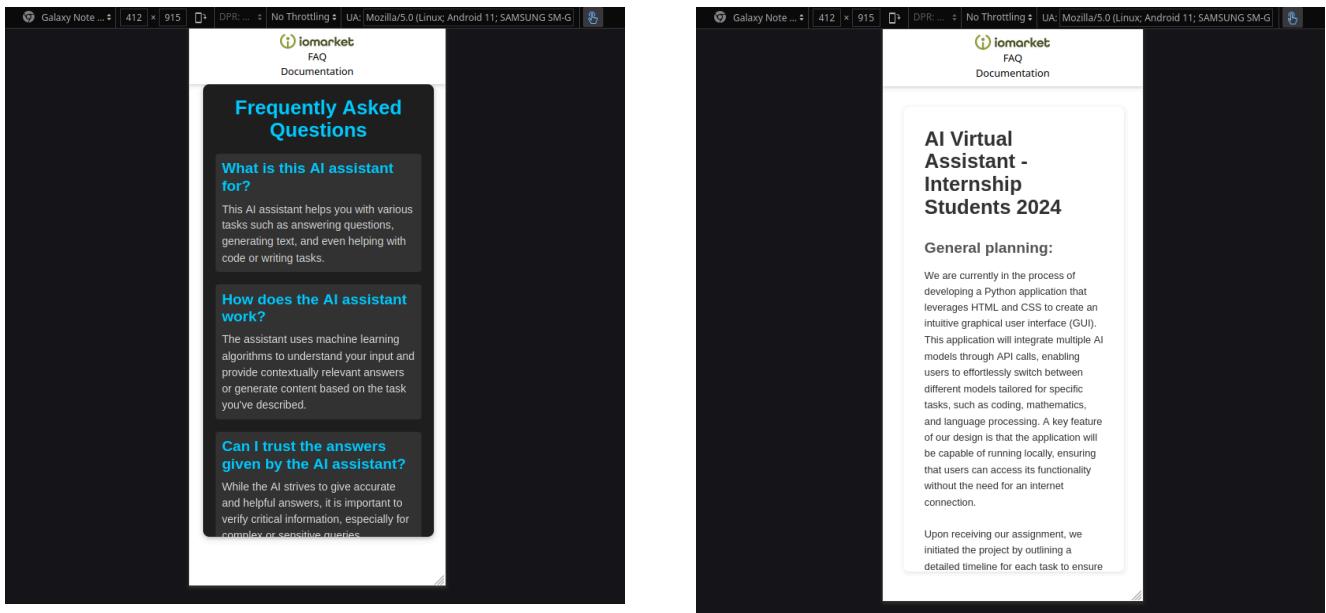
19.09.2024

The overall design has largely remained the same, but there is one crucial improvement: we have made the website fully responsive. This means that the layout and elements adapt seamlessly to different screen sizes, providing an optimal viewing experience across a variety of devices.

We have already conducted extensive testing, and I'm pleased to report that the website functions well on most screens. This responsiveness not only enhances usability but also ensures that our design remains consistent, regardless of the device being used.

This consistency is something we truly value, as it reinforces our brand identity and improves user satisfaction. Overall, these enhancements contribute to a more versatile and user-friendly experience, allowing us to reach a wider audience effectively.



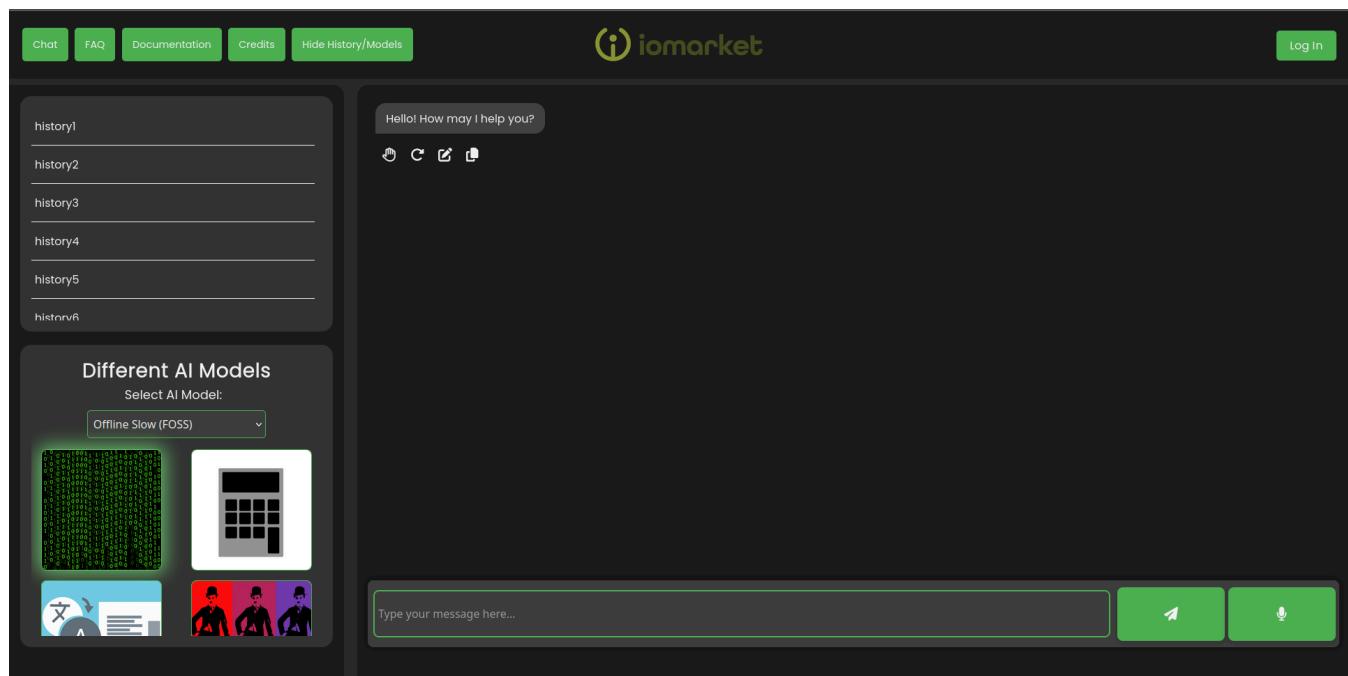


The ability to access our website on both mobile and desktop devices significantly broadens our user base, enabling the assistant to reach a wider audience. We specifically mentioned the Galaxy Note 20 not only because it performs well on that device, but also for simplicity's sake. It was the first device that appeared during our debugging process, and as a result, it has become our universal mobile debugging tool.

While we have certainly tested the website on other smartphones, the Galaxy Note 20 served as our initial benchmark. Subsequent testing on additional devices was conducted only after we confirmed functionality on the Note 20. This approach allowed us to streamline our debugging process and ensure a solid foundation before expanding our testing to other platforms.

07.10.2024

We made several minor CSS adjustments here and there, which eventually accumulated into a new and improved design. This design now includes different options that can be customized through the settings file and is intended to be fully responsive. Additionally, we began testing the updated design across various browsers, documenting the issues specific to each one. From there, we started working on resolving these issues to ensure that our project would function smoothly on all browsers.



Backend planning:

Task:

We will develop an extensible backend that enables us to easily swap out different AI models, facilitating the creation of a versatile AI Virtual Assistant. This architecture will allow for seamless integration of new AI technologies as they become available, ensuring that our application remains up-to-date and capable of meeting diverse user needs.

The backend will also incorporate advanced features such as speech recognition, allowing users to interact with the AI Assistant through voice commands for a more natural and intuitive experience. Additionally, we will implement functionality to save chat histories, enabling users to revisit previous conversations and maintain continuity in their interactions with the AI.

For the AI model, we will utilize Microsoft Phi 3.5, which offers robust capabilities for understanding and generating human-like responses. This choice will enhance the overall performance of the Virtual Assistant, making it a valuable tool for users seeking assistance across various tasks and topics. By combining an extensible backend with powerful AI capabilities, we aim to deliver a comprehensive and user-friendly virtual assistant experience.

Design Philosophy:

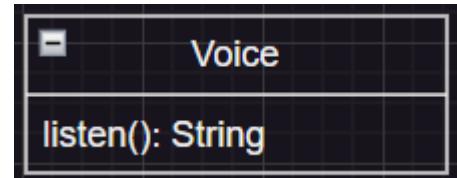
Our design philosophy is to create one script file per feature. This allows us to possibly reuse certain features in other projects very easily, as we can copy the individual .py files, which all work on their own, except the specific .py file for this project, which is specially tailored towards this AI chatbot.

UML diagramms:

patrickpluto07@gmail.com yas.onm08@gmail.com

Hey if you do this make one picture per UML diagram and not like down here 2 diagrams per Picture

api.py
This is where the AI models are interfaced with. This script will then provide high level abstractions to communicate with the different AI systems. Be it an online API, a local ollama api, HuggingFace api.



simple_chat.py
This is a simple CLI chat interface to test the backend systems. It will be superseded by the eventual graphical API in html/css.

web_flask.py
index(): html

First Prototype:

You are able to simply select an AI Model, then type out what you want to ask. This Prototype is already ready to answer questions and detect what language you write in and give the answer in the corresponding language.

```
Which AI Model do you want to use? Write as a Number (1-5).
1

Ask a question
Hello!
Hello there! How can I assist you today?
```

Combining back and frontend

Flask setup

The flask setup was relatively quick. We had to adjust all of the file references, and add /static before them. Technically, this is a leftover of the previous code, where we had to put the html in the /static folder, but we left it as is, because it makes sense to do so, as those files are statically served.

API calls & interface

After adding the html, We had to create some API calls that allow us to dynamically modify the html file with the chat logs, and extract the button presses, and text that was given by the user. This would then be transferred to the respective python files.

Debugging

Debugging was made very easy by Flask. When something goes wrong, it not only prints this out on the console, but also on the website, allowing you to easily see what went wrong. Although there weren't many issues, it mostly "just worked".

Switch to React

After several challenging days of struggling to get our website to function as intended, we decided to switch to React.

The transition to React didn't go as smoothly as we had hoped, initially exacerbating some backend issues. However, despite these early setbacks, the switch ultimately allowed us to leverage new possibilities and improved options for our project, paving the way for more effective development moving forward.

Another part is that our Web App no longer existed and that our timetable had to go through another round of planning and changes. Since we are no longer using Python, we can no longer just use QtWebEngine, and have to switch to Electron, which has great support for node.js.

New Backend

The new backend is much more versatile than the previous one. We were actually trying to use Flask as if it were JavaScript, while in reality, it's more or less made to just communicate between JS and Python. The new backend will still be partly python, but only for the backend saving with a database. This way, we may be able to make the chats display while generating even.

After a bit of work, the backend was coming along nicely, but the issue was that the frontend was lagging behind the backend, leading the backend to have a weather api built in, before the speech to text even worked.

Testing

Test cases

Results

Name	CPU	Memory ↴	Download	Upload	Read	Write
Spectacle						
PyCharm Community Edition	2.8%	27.4 GiB			125.5 MiB/s	

sadly this was not the most ram our system consumed, just the highest number we could capture before PyCharm forcefully removed itself from the tasklist.

Bug fixes

Reflexion & learnings

Yasin

When I first heard that our project had something to do with AI, I was scared. Then I heard that it was going to be in Python. Already 2 things I have no idea about. But then, after our first prototype, I couldn't find the words to describe it. It was much, much easier than I thought. The 1 week was the week I learned the basics of Python. On our journey with only Python we encountered different problems. As we had all started to learn a bit of React Typescript a few weeks before our internship, we decided to partially switch to React. We made huge progress in no time. We added feature after feature, learning the pros and cons of React through experience. As we had never worked in a 3-person team before, and in such an efficient team, we completely misjudged how long we would need for different things and had to rewrite our plan over and over again. In the end, it's been a wonderful experience and I can say I'm proud of how far we've come. Of course we made mistakes, whether it was folder structure or just programming habits, but we made huge progress and gained knowledge that would be hard to get anywhere else. We learned how to estimate time, how to code in React, and we learned Typescript. All in all, I can safely say that these 4 weeks have been 1000% worth it.

Luca

When I first started this project, I was completely unfamiliar with React and had never worked on an AI project before. I also had limited experience collaborating with more than one other person on a project. It quickly became clear to me that the Python knowledge of my teammates, Yasin and Patrick, was far superior to mine. Even tho Yasin had never used Python, I decided to focus on the frontend.

In the end, Patrick handled the backend and testing, Yasin worked on the feature implementation and the connection files between the frontend and backend, while I took charge of the frontend, presentation, and documentation. Each of us contributed in areas where we excelled, and our ability to correct each other's mistakes was crucial to the success of the project. We supported one another and learned a great deal along the way, growing stronger as a team. Our technical knowledge expanded significantly, and the project served as both a valuable learning experience and a welcome relief from stress. Although we faced some time crunches toward the end, it truly was a great experience.

However, I would not structure or approach a project the same way again. The file structure we used was less than optimal, and managing the CSS was quite challenging. In the future, I would likely opt for a flatter structure using Tailwind or DaisyUI to simplify the styling process.

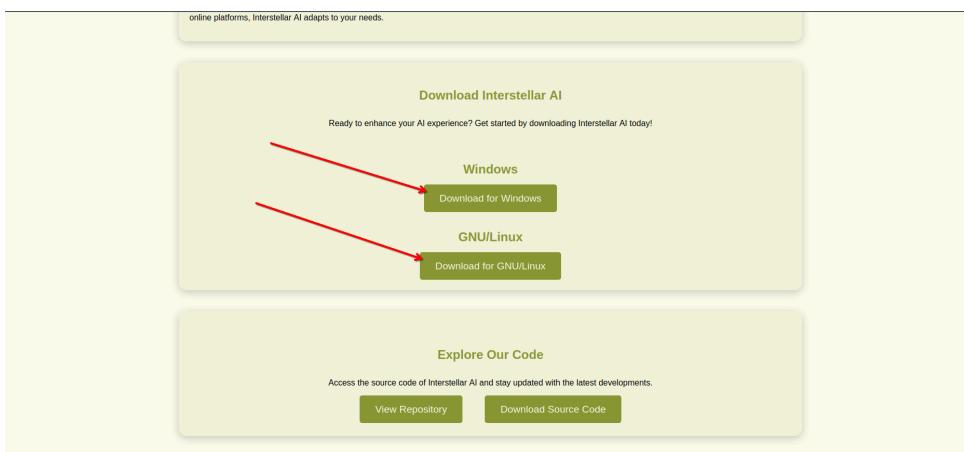
Over the past four weeks, I have not only grown as a programmer but also as a person. I want to extend my heartfelt thanks to iomarket, especially Leart, Tristan, Martina, and Max, for making this experience possible. Additionally, I am grateful to Yasin and Patrick for being such fantastic teammates; I would have gone insane without both of you.

Patrick

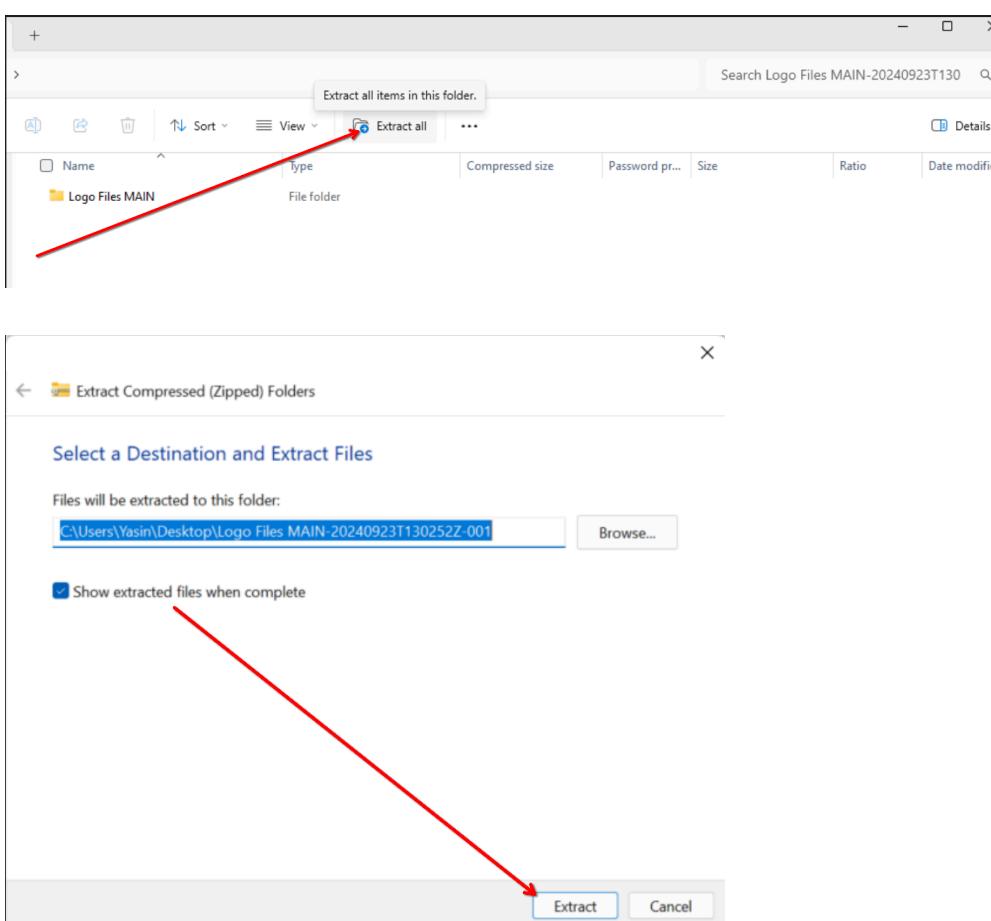
Manual

Download

1. Click the download button on the website.



2. Unzip the folder.

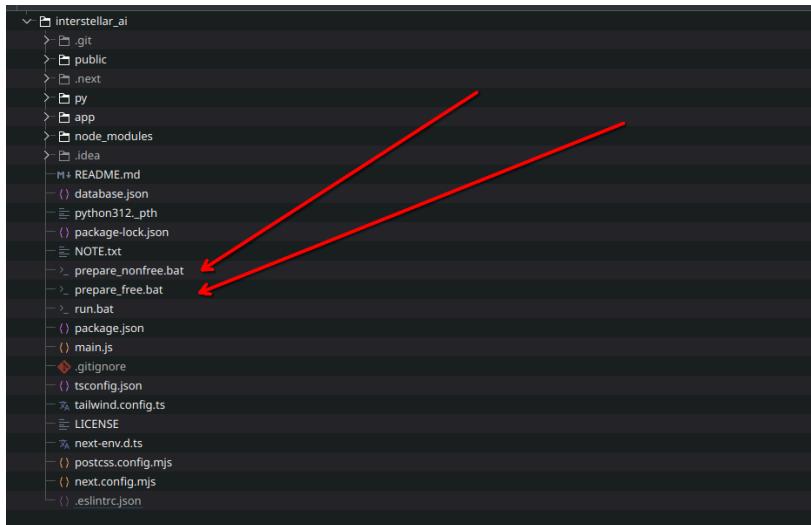


3. Start your chosen selected selected AI's

Linux: prepare_nonfree.sh or prepare_free.sh or prepare_all.sh

Windows: prepare_nonfree.bat or prepare_free.bat or prepare_all.bat

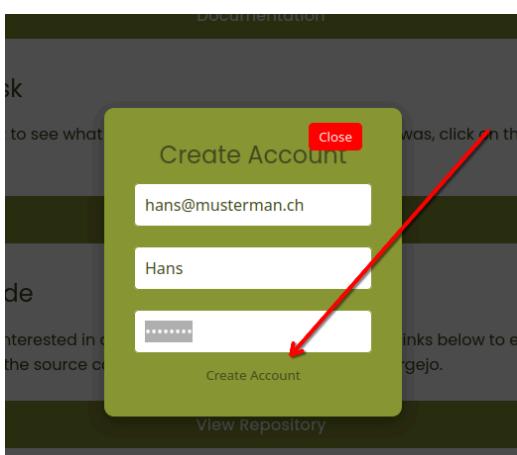
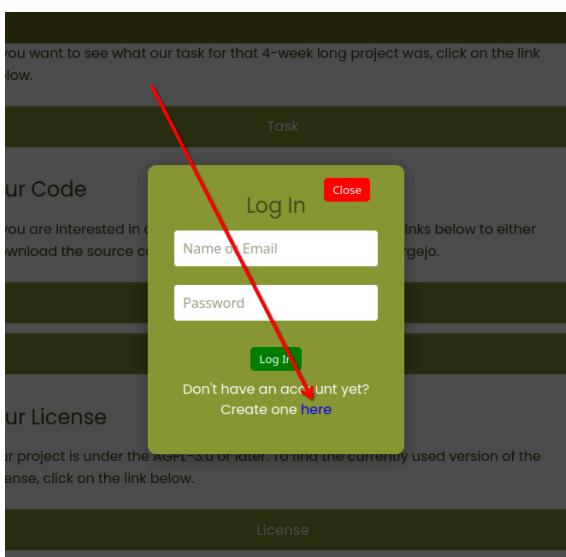
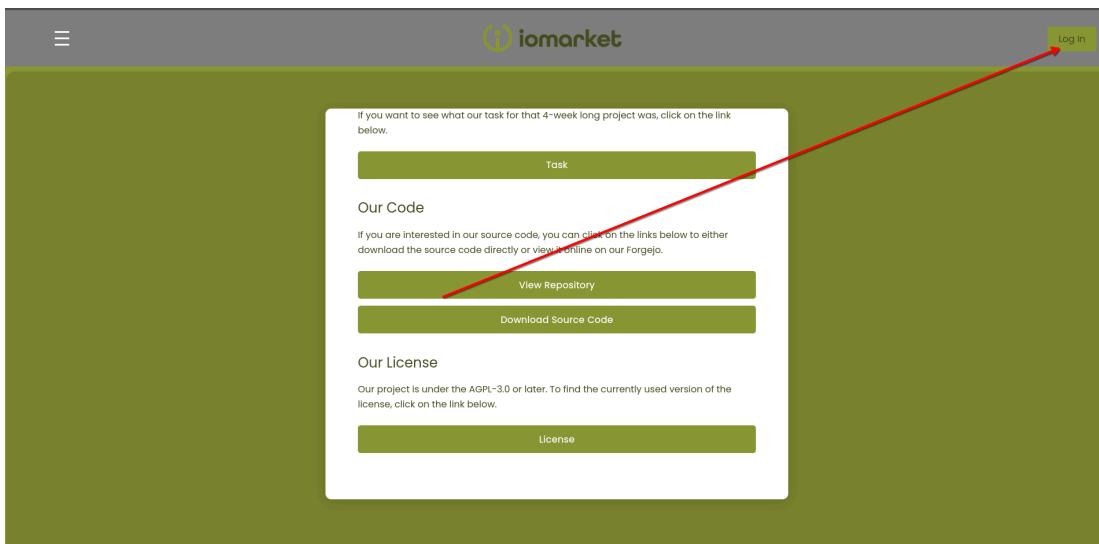
Here the free does not stand for free of charge it stands for free as in freedom. For the user that means that everything that comes out of the free chats bots yours to use, modify or share. For the non-free however it could be possible that a license restricts you from doing so. So check in advance what license your chosen ai is utilizing.



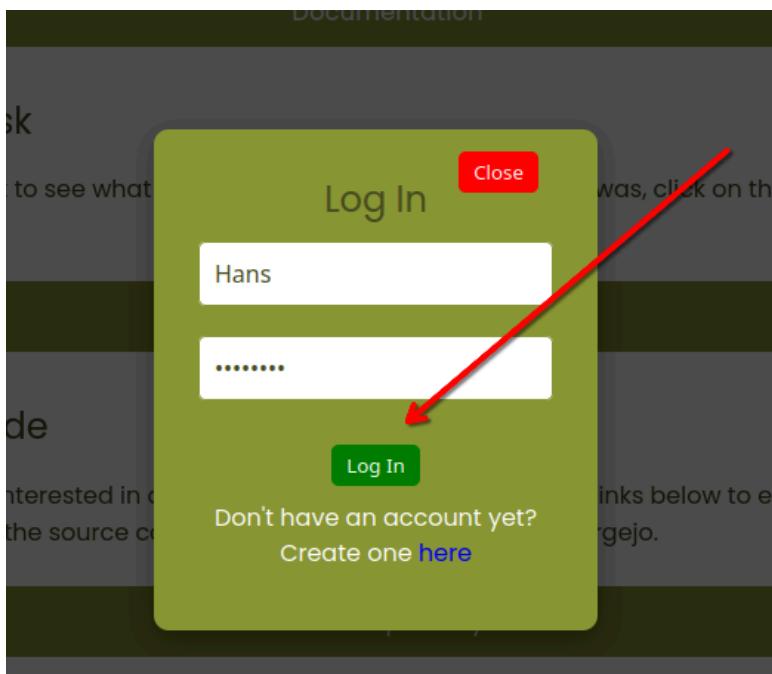
4. Start run.sh or run.bat or run.sh



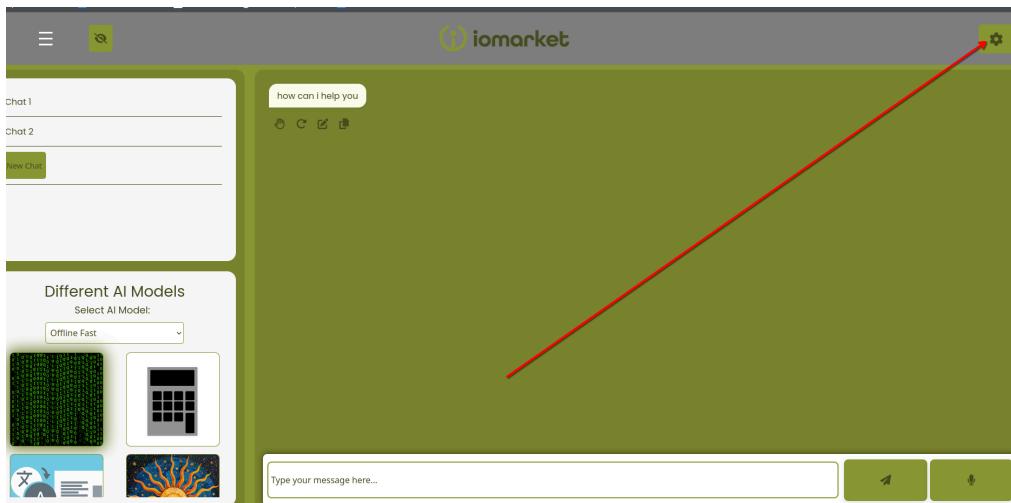
5. Create an account.



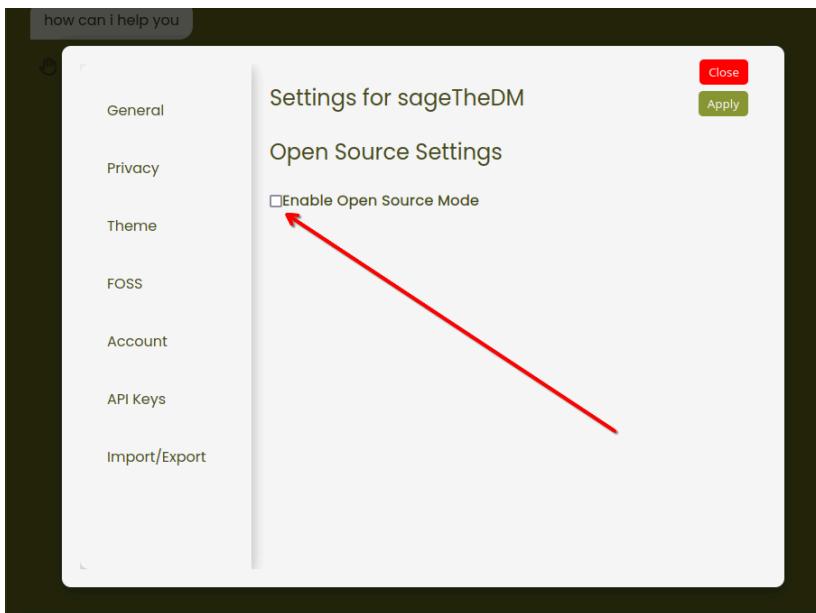
6. Log in.



7. Go to the settings.



8. Apply settings depending on what you have prepared.



9. Start chatting with our AI.

Time Management:

DATE	TASK	STATUS
16.09	Plan the Project	Done 16.09
17.09	Learn Python and Flask	Done 16.09
17.09	Create the first Prototype on the console	Done 16.09
18.09	Create a first version of a simple GUI	Done 16.09
18.09	Connect the GUI and the Backend	Done 16.09
18.09	Debug, test and fix	Done 16.09
19.09	Improve the GUI for more functionality	Done 16.09
19.09	Create the Backend for it	Done 16.09
19.09	Make it a web app	Done 17.09
20.09	Add multiple AI models	Done 17.09
unplanned	Switching to React	Done 18.09
22.09	Feature Parity with before the switch.	Done 21.09
24.09	Make the Backend be multi threaded to better display the chat with the bot	Done 23.09
26.09	Speech recognition	Done 30.09
27.09	Text to speech	NOT DONE
30.09	Task automation	NOT DONE
30.09	Weather information	DONE
01.10	Add a chat history	DONE
02.10	Additional tasks	NOT TOUCHED
7.10	Debug, test and fix	DONE
8.10	Finalize Product	DONE
9.10	Finalize Documentation	DONE
10.10	Create Presentation	DONE
11.10	Present our Final Product	DONE

Working table:

DATE	Patrick	Yasin	Luca
16.09	<p>Morning: Created the Git repository, created the AI backend with ollama.</p> <p>Afternoon: Created the backend bindings from the GUI to the AI backend and speech recognition backend. Created class diagrams for the backend.</p>	<p>Morning: First look into python, learned the git commands, created a simple functional terminal interface as a prototype</p> <p>Afternoon: Implemented and tested SpeechRecognition. It doesn't translate well but I can't change anything about it other than try out other resources.</p>	<p>Morning: Create the DOCUMENTAION file create the milestones and issues on Forgejo. Started working on the GUI with HTML and CSS.</p> <p>Afternoon: Working on the GUI, creating flowcharts for the backend & improving the documentation. Creating additional html documents for contact and faq.</p>
17.09	<p>Morning: Attempted to add a pytorch model, realized it was too slow, and reverted back to the same ollama system.</p> <p>Afternoon: Created a web app via QtWebEngine 5, and bound the Le Platform and HuggingFace APIs to alleviate the local processing power that is needed. Attempted using the official ollama library, figured out that I can generate the individual tokens, and attempted to show those on the webpage. I failed, but realized why. I will do it the next day.</p>	<p>Morning: created a python file for speech recognition with Whisper (TESTING NEEDED), updated the css for the model options</p> <p>Afternoon: worked on the styling of our app with Luca, we fixed the scrollbars with help from Leart and just general tweaks on css</p>	<p>Morning: Planned a smal meeting with Patrick and Yasin to improve efficiency. Create new buttons for the backend. Made our css responsive.</p> <p>Afternoon: worked on the styling of our app with Yasin, we fixed the scrollbars with help from Leart and just general tweaks on css. Worked together with Patrick and tried to help him fix the backend.</p>
18.09	<p>Morning: I attempted making the python backend display the text as it generates, but flask just wasn't up to the task of doing so. In the end we decided to switch to react. The start was a bit difficult, as</p>	<p>Morning: At Home I tested the Voice Recognition with my own PC. On the CPU it works well. Not really fast but perfect transcription for every word. On the GPU however, I just couldn't get it to work.</p>	<p>Morning: Made Issue and Milestones, worked on the Documentation. After that i started to work on the GUI again. And After the decision to switch to react i rebuild the entire Frontend in react</p>

	<p>nothing worked like it should have.</p> <p>Afternoon: I implemented the API to communicate with Le Chats API, but the actual display of the chat did not work. After a lot of help, Yasin was able to figure out how the backend should be made, and fixed the chat.</p> <p>I also tried making an electron app, but I need to look further into that, as it requires a lot of specific things to be done.</p>	<p>Today I was mainly trying to help Luca and Patrick (Who knows how much I contributed), We decided to switch to React TypeScript which was a huge improvement. That's why I learned a bit of TypeScript.</p> <p>Afternoon: Fixed the ai-user chat Little tweaks on css.</p>	<p>Afternoon: Still was migrating the entire Frontend to react and working on the responsive design. Had to leave early today tho.</p>
19.09	<p>Morning: I started writing the API python backend for our ai chatbot. It went along really well, and I had to wait a bit for the frontend to also be able to interact with this backend.</p> <p>Afternoon: In the afternoon we noticed that the AI was malfunctioning. It kept giving us ridiculous answers for simple questions, after which I started wondering why. It turned out that I did not understand just how AI worked. I then quickly reflected the API to fix this, but it will take some time until the frontend interacts with it properly,</p>	<p>Morning: I wrote the code for the voice recognition in our project folder but because we don't need it right now, I commented it out. I also Started writing the new code that will open a couple of doors for the future of our program.</p> <p>Afternoon: Implemented Multi-Threading (For future features) and thus rewrote the InputBackend.tsx file. We noticed a huge mistake. How we thought the AI worked was completely wrong. How we gave prompts etc. wrong. In the near future we are going to fix that. Added Enter to Send.</p>	<p>Morning & Afternoon Today, I dedicated the entire day to working on the HTML component of our React application, focusing on the underlying logic for how the website should be displayed. I also revamped the CSS, implementing significant changes to enhance the responsive design. The graphical user interface (GUI) is now fully functional and stable. As long as we don't introduce new GUI elements, there will be no need for further modifications, which is a great advantage.</p>
	<p>Morning: Worked on the backend code more, and I also added a way to change the API that is used for chatting with the AI. I then secured the system by changing it to https and generating a more secure token, as to</p>	<p>Morning: Today I rewrote the code to fix the problem we've had. I had to change how the prompts were forwarded to the backend. Had a meeting where we discussed how far we are and got some feedback. We need to restructure our project</p>	<p>Morning I added the login functionality to the front-end and significantly changed the file structure. I also made some modifications to the CSS and started working on a change to the page.tsx file. However, after spending</p>

	<p>prevent people from reading your messages.</p> <p>Afternoon: I added a way to generate the https certificate from python, as the openssl certificate generation is painful to do on Windows.</p> <p>Then I implemented a database system, which basically allows users to authenticate themselves, to access their data.</p>	<p>folder. I want to start on the live text feature today.</p> <p>Afternoon: Tried to implement live text rendering. In the end I realized that I had lost track of how the code worked (It didn't even work properly) so I deleted everything so I can try again on Monday. After that I made some changes on the FAQ and documentation css.</p>	<p>the entire afternoon trying to fix the issues that arose, I had to abandon my changes. Unfortunately, this cost me a good five hours of my time.</p>
23.09	<p>Morning: In the morning I worked on a list of AI models we would implement in the final version, APIs and all. I then started implementing the ones that had similar syntax to our current one, but I then ran into the issue that some APIs had different syntax, which would require me to preprocess the values. I thought to do this in the afternoon, but that plan was cut short by a bit of an issue.</p> <p>Afternoon: In the afternoon, I decided to first implement back our Web App, but soon realized in shock that the only competent web rendering engine for python crashes with our React code somehow. So I had to scramble to find alternatives. I first thought to use something like Nextron, but that wasn't useful because it removes the ability to run this as an individual website, which we did not want to do. I then managed to find an alternative, but realized that if it found the PyQt library, it would use the exact same web</p>	<p>Today: The reason I write this like this now is because I spent the whole day trying to get the live message to work. I had all types of failures from little typos all the way to mistakes that made me re-write the whole script. But now I can say the live text (semi) works! The first text is excellent. just perfect. BUT after that it just doesn't work and that is what I'm gonna be working on for the rest of the day and maybe tomorrow</p>	<p>Morning: This morning, I completely revamped the way colors are managed in the CSS by converting every color into a variable. I also began working on the settings.tsx file and made some minor adjustments to other files as well.</p> <p>Afternoon: In the afternoon, I redesigned Login.tsx and Settings.tsx to enhance their visual appeal. Additionally, the settings under the theme section are now fully functional, allowing users to manipulate the CSS, which is a great improvement!</p>

	<p>rendering engine. So I am still in search of a good web app program. This way I had to waste the entire afternoon trying to frantically fix an issue that I never thought I'd even have.</p>		
24.09	<p>Morning: Finishing touches to the Python API, like adding Weather and fixing the Database API, and also some simple Speech Recognition. I realized that my web app idea wouldn't work until 03.10.2024, because that's when the next QtWebEngine releases, with support for Url.canParse.</p> <p>Afternoon: I started work on the Typescript/React part of our project. My goal was to get the chat buttons working, with that I mean the resend, copy and edit buttons all work now, which is nice.</p>	<p>Morning: a small step for man, a giant leap for mankind. I managed to make the live message display work. In the end, all it took was a couple of hours and... 1 line of code. After that I implemented the methods that are responsible for disabling the input if the ai is still generating and if the input is empty.</p> <p>Afternoon: I decided to add a hamburger menu (not food) so our mobile users have a more user friendly interaction with our ai. I'm not done yet but it shouldn't take too long.</p>	<p>Morning: Implemented an import and export feature in the settings, allowing users to export their settings and appearance as a JSON file. This enables seamless import on another device, ensuring a consistent experience across multiple devices.</p> <p>Afternoon: Enhanced the settings files and improved the associated CSS. Collaborated with Yasin to refine the responsive design and assisted Patrick with the backend functionality of the buttons in the conversation interface. Additionally, made minor adjustments to the CSS to ensure better compatibility with the various themes we have preset.</p>
25.09	<p>Morning: After a long time of messing around, the Web App solution was very easy. I was mostly confused on how Electron worked, but after trying a bit, it worked splendidly. I then started securing our database system a bit, adding more ways of authentication, and also fixing some nasty backend issues.</p>	<p>Morning: I watched a couple of YouTube videos on react, ts and css to learn techniques that I possibly could implement. After that I started Working on the Header. I added a hamburger menu which is only visible and functional if the user screen is small enough. It even has a cool little animation where the menu bars turn into an X which closes the menu</p>	<p>Morning: I spent the morning enhancing the settings feature of the project. I focused on improving functionality and added new fonts, ensuring they would work seamlessly across all devices by downloading and organizing the necessary font files as fixed assets. Additionally, I improved the header structure and added new navigation buttons to enhance user</p>

	<p>Afternoon: In the afternoon, I was trying to get the voice recognition working, but it did not accept the inputs I was giving, so I decided to start figuring out the problem together with Yasin. In the meantime I added a little bit of TTS support, but it is very basic so far, and I expect to expand it in the future.</p>	<p>and the X turns back into the standard menu bars.</p> <p>Afternoon: Today we met the founder of iomarket and had a meeting with him where we presented our project and how far we have come. After that I tried to fix little errors (unsuccessfully). I want to test out the Speech recognition at home because of the better hardware.</p>	<p>experience and accessibility.</p> <p>Afternoon: In the afternoon, I worked on the models and began experimenting with the <code>useEffect</code> and <code>useState</code> hooks in React to ensure that user selections are saved correctly. After implementing these changes, I attempted to fix some minor errors, but my efforts inadvertently led to additional issues that will need to be addressed tomorrow. Overall, it was a productive day with some challenges that provided valuable insights for future work.</p>
26.09	<p>Morning: I worked some more on the speech to text backend part with Yasin, and switched it to whisper, which didn't work as well as I'd hoped. There were issues concerning the file types of the audio files being saved as oggs, while the backend expected wav files, and to convert we need ffmpeg, which is annoying to use properly.</p> <p>Afternoon: I started work on a better deploying structure. For the end user it is annoying to have to install python, nodejs, ollama and have to install all the node modules and pip modules with weird and confusing commands. So I made it possible to distribute as a single .zip with a few</p>	<p>Morning: Today I worked on speech recognition. My already existing model and code worked. So the only thing I have to do is: record users' voice on react, send it to the AI in the backend and voila. Sounds easier than it is. The catch is: 1. on my existing code I had a fixed recording duration and now I have to make it stop when the user clicks a button. 2. in the existing code it temporarily saved the input audio as wav and after transcription it deleted it. Now it records in Ogg and I have to make it work without saving the audio.</p> <p>Afternoon: I added a little feedback on the copy button that</p>	<p>Morning: I began my day by addressing some settings and correcting past mistakes. I encountered a few local storage issues, which I managed to resolve with some clever problem-solving, thanks to Leart and ChatGPT. Additionally, I made several minor CSS tweaks in an effort to fix the header. While it's still not functioning properly, it now only seems to be an issue on Yasin's computer—progress, I suppose! At 11:00, I shifted my focus to the favicons but quickly ran into some strange behavior that I couldn't explain.</p> <p>Afternoon:</p>

	<p>easily usable .bat files which automatically install everything for you, and automatically run the web app with all dependencies installed and the python backend running too.</p>	<p>says "Copied!" for 1 second. I researched how I could change the recording type from ogg to wav.</p>	<p>I continued my work on the favicons, but nothing I tried seemed to work. After a while, Leart stepped in to help, but we still couldn't resolve the issue. So, we decided to bring Tristan into the mix. With his assistance, we finally managed to get the favicons working after a frustrating amount of time. Following that, I made further adjustments to the account settings, ensuring that they save to local storage. This way, once a user creates an account and logs in, they won't have to do it again. Unfortunately, it's still a bit buggy, and currently, users can only log in with their email. I'll prioritize fixing that as my first task tomorrow.</p>
27.09	<p>Morning: I fixed the windows packaging scripts, and i also attempted a linux version, but I decided to do this one later, as the instructions are easiest for Linux. Afternoon: I also tried getting the web application to work on Android phones, but sadly a lot of the npm packages did not work on android, for this port, I may need to do even more research. We also switched out phi3.5 for llama3.2 which was released just yesterday, as it's faster and better than the competition at its size. I then also helped Luca fixing the online and offline only options, and the FOSS and non FOSS options.</p>	<p>Morning: Yet another day of "trying to get voice recognition to work!". Watched YouTube videos, read countless comments and documents tried with chatGPT (made it even more complicated). No success yet but getting closer.</p> <p>Afternoon: after further trial and error I kind of got it to work? I still need to carry out some tests at home with my PC and setup. but I don't get any errors anymore. Additionally I started working on the overall css. It still had some old css stuff that we didn't need anymore. Further on I assisted my team members as far as I could.</p>	<p>Morning: I worked on diverse settings and little tweaks left and right and just worked on the stuff that had to be done I can't really specify what I have done since I jumped from task to task and made little tweaks threw the entire day and helped Patrick and Yasin with smaller things in their respective tasks.</p> <p>Afternoon: I worked on the inputOutput handler and the settings once more. The settings now effectively change the system prompt and with that the behaviour of the AI which is I might say very cool. The language system is still a bit clunky but that is a fix for another day.</p>

30.9	<p>Morning: I started work on actually getting the Database Login backend fixed, because it did not work properly, and after a bit of debugging, and rewriting parts of the backend, it actually started working. Saving is yet to be implemented though.</p> <p>Afternoon: I started work on the ability to switch AI models and categories on the fly, and fixed a lot of bugs, which worked pretty well, except for the fact that API keys just wouldn't save to local storage anymore, just deleting themselves.</p>	<p>Today: I started working on the css again. I had to do some refactoring and completely change some code but now it looks appealing on pc and on mobile, but now that I'm writing about it I now remember I forgot to do the settings css... I guess that's what I'm gonna be working on tomorrow</p>	<p>Morning: I began by making some minor changes to a few files to enhance their readability. Once that was completed, I started experimenting with the system prompt and its behavior in an effort to optimize its functionality. Following that, I collaborated with Patrick to search for a new AI model for the character AI. We successfully found one and have already begun implementing it.</p> <p>Next, I focused on the login functionality to ensure that user data would be saved in the database, which will simplify account management significantly. This required modifying several files, and Patrick needed to double-check the changes with the backend and make some necessary adjustments. Afterward, I addressed the storage issue related to the API keys, allowing Patrick to implement them more easily and switch between different keys as needed.</p> <p>Afternoon: After an extensive period of testing the impact of the system prompt on the AI, I realized that I had made a significant mistake. Consequently, I completely revised the system prompt to ensure it would function correctly.</p> <p>Following that, I returned to the theme settings to implement all the new color options we had</p>
------	--	--	--

			<p>developed, making them customizable. Additionally, I fixed a bug that caused custom settings to not load when refreshing the website, since it only loaded when the settings menu was opened.</p>
01.10	<p>Morning:</p> <p>In the morning I fixed some of the broken API's and I also implemented the model picker properly, so you can easily pick between different models on the fly. The messages have also been breaking for a long time, and with that I mean that the message would always cut off before reaching the end, by putting a small timer at the end, the frontend properly pulls the last bits of the text and displays them to the screen.</p> <p>Afternoon:</p> <p>I started some basic work on the chat history structure, which will eventually be combined with the settings to be saved together. I then had to do a small auto scroll fix, and I then finally fixed the other parts of the backend, which makes saving data, creating accounts, logging in, logging off, changing credentials and many more things possible.</p>	<p>Today:</p> <p>I improved our css, so it now scrolls to the end of the conversation whenever the user wants it to. I learned about the IntersectionObserver, which is pretty cool. I am probably going to use it in private too. You can make some pretty cool effects etc. with it. I was searching for coding tips on youtube for scrolling and some other stuff when I saw a Video that explained it really well. I added 1-2 little features to our app.</p>	<p>Morning:</p> <p>This morning, I undertook a complete overhaul of the settings file. I reorganized the structure by splitting it into multiple files, with each file dedicated to creating specific options in the Settings, which are referenced from the <code>Setting.tsx</code> file. This modular approach not only enhances clarity but also improves maintainability. Additionally, I ensured that all dropdowns are easily expandable and modifiable, which will streamline the process of adjusting and experimenting with the Settings in the future.</p> <p>Afternoon:</p> <p>After spending another hour on the project, I handed over the Settings to Patrick. I took the time to explain the structure of the files in detail and discussed the issues we are currently facing with the Database, ensuring he had a clear understanding of the challenges. Following that, I shifted my focus to the models, where I initiated a comprehensive overhaul and addressed some existing bugs that had not yet been resolved. This included refining the</p>

			<p>code for better performance and reliability. Finally, I implemented new styling to clearly indicate which Model is currently in use, enhancing the user experience. With that completed, I took a moment to document my progress for future reference</p>
	<p>Morning: I did a bit of bug fixing on many different smaller things, like the color picker outside of Firefox.</p> <p>Afternoon: I expanded database functionality to accommodate for settings and the chat history. I then ran some tests on the Website and created a testing list of things to check for as an end user. I then attempted to help Luca with some Frontend tasks, and while I was able to help with some, I was not able to do so for certain other issues that came up.</p>	<p>Morning: Our current version has features that enable users to design their UI however they want. The problem was that personalizing our icons was impossible. I researched how we could change that and found out that if you directly use svg paths instead of img in html, you can change the fill color and that's how I got it to work. Now our icons have the same color as our text. Then I had the icons clipping, they were on a higher y-index level than anything else so they were seen even if there was an overlay which is supposed to be on top of everything. While we were testing our application we found out that somehow it only works properly on firefox (we all use firefox). The funny thing is firefox is usually the outsider that needs special treatment to make it visibly the same as all the other browsers. Now it's fixed.</p> <p>Afternoon: I looked into filesize management for our database since we wanted to save user settings and chat history so the user can just log into their account and</p>	<p>Morning: I completely overhauled the themes once again, implementing significant improvements. I added new functionality, refined how the themes are saved, and provided assistance to Patrick and Yasin with their respective updates. Additionally, I tied up loose ends and ensured that previously unused settings now had a meaningful impact on the overall system, enhancing its usability and effectiveness.</p> <p>Afternoon: In the afternoon, I focused on addressing the persistent issues with the system prompt, which was still buggy and often malfunctioned when we were not logged in. This led to a frustrating experience, as the system would sometimes break entirely. To resolve this, I revised the logic behind how the system prompt operates and created a default system prompt to ensure more consistent performance. Despite these changes, some issues remained, particularly with the measurement settings, which stubbornly refused to update or were</p>

		<p>keep on chatting with our AI. After a small discussion with Patrick we came to the conclusion that we want to use 1 JSON object for each user setting and chat history. So I started working on the frontend part of fixing it since we originally wanted to do 1 JSON object for both. While researching I came upon docker. I have no idea what it does, why anybody would need it but I was interested in it. I ran down a rabbit hole while researching docker, watching YouTube videos, asking chatGPT simultaneously to understand what it does and how it works. I still don't know. I'm gonna have to invest more time on this in private at home.</p>	<p>ignored by the system prompt entirely. As a result, I dedicated most of the afternoon to debugging and working on solutions for the problems we were facing. While the issue of the system occasionally using metric instead of imperial (or vice versa) still arises, it is now much less frequent. We plan to conduct further bug testing in the future, but for the moment, we are satisfied with the system functioning correctly about 90% of the time. Overall, the progress made today has set a solid foundation for continued improvements.</p>
	<p>Morning: In the Morning I mostly helped Luca and Yasin with some issues in the code. I personally started work on a small fix to get the App working on a Phone. Afternoon: After successfully getting the Phone to work, I also quickly changed two of our js files to ts files, to have a more typescript only codebase. I then also tried figuring out solutions for self-hosting the app. While I realized that we could easily host this on a server with a domain, It would be difficult to get the speech recognition to work if you have multiple devices accessing the same app. The issue is that the Browser won't send any Voice or recording over a non</p>	<p>Morning: I spent most of my time just researching. How can I implement different features and fix bugs? At some point I came across assembly. I started looking at documentations and then my interest on how CPUs actually worked, arised. I learned about full adders, subtractors, alu, registers and how memory works. I tried programming a little program that adds 2 numbers but somehow it didn't work on my powershell on windows so I had to use wsl ubuntu. When I got it to work It was lunchtime.</p> <p>Afternoon: I started brainstorming how I could make the Chat histories work. While I was thinking</p>	<p>Morning: I continued to address known issues with our software, but I became increasingly frustrated as it didn't function as intended, and I struggled to identify where to start fixing the problems. After spending the entire morning trying to resolve the issues with the radio buttons, I found that the situation had worsened. Feeling stuck, I decided to assist Patrick instead, as I was unsure how to proceed with the fixes.</p> <p>Afternoon: I helped Patrick compile a list of the test cases he had completed in the morning, along with documentation detailing what had worked and what hadn't. My afternoon was primarily focused on bug fixes and</p>

	<p>https connection, unless the connection is local. This effectively makes running the website on a personal computer, then accessing it on the phone impossible, as trusted certificates require a domain name. I then suggested adding a setting to change the Python backend, In order to be able to use locally hosted AI's on your Phone or share your GPU power with Family Members.</p>	<p>about it. Patrick somehow got it to work on mobile with answers. It basically sends the requests to the pc and the pc gives the response directly to the phone. Then we discovered that our speech recognition doesn't work because to use the audio recorder we needed https. We started thinking about solutions and we thought about hosting stuff etc, but then got distracted and started looking up gpu prices. From there our conversation subject somehow changed to bitcoin mining and then how much money we could have if we traveled back in time, bought bitcoin and came back to the present time. From there we started discussing how time travel works, how dimensions work, even how blackholes work. At that point we realized how much time has gone by.</p>	<p>improving our code for the test cases that had either failed outright or had room for enhancement. Everything was going smoothly until I had to tackle the radio button issue once again, at which point I found myself running out of ideas.</p> <p>Stayed late since Patrick fixed the Radio Selection but I had to problems with the merge/revert and stuff</p>
	<p>Entire Day: I tested just about any and all Browsers, on Linux and on Windows, and in the process I found a ton of bugs. Surprisingly, even though some browsers were based on the same Web Rendering engine, there were still browser specific bugs somehow, and even OS specific bugs, which me and Luca then wrote down on a big spreadsheet, including a test of the entire Website.</p>	<p>Morning: In the morning we concentrated on fixing little bugs so we could present a better version in the afternoon. Additionally I added markdown formatting.</p> <p>Afternoon: We had our meeting and presented our standpoint. We got some tips from the owner and set a meeting for next week. After the meeting we further focused on testing and fixing bugs.</p>	<p>Morning: Patrick and I conducted testing on both Windows and Linux systems, using various browsers. We compiled a comprehensive list of all the test cases and documented every single browser we used. To facilitate this, we set up a separate laptop specifically for testing, where we first installed and configured the necessary software, including all the browsers. Once the setup was complete, we began running the tests and meticulously</p>

			<p>documenting the outcomes for each scenario. Midway through the testing phase, we took a brief break to attend our weekly project meeting. During the meeting, we discussed the overall progress of the project, our current status, and the tasks that still needed to be completed. After the meeting, we resumed our work, putting in our best efforts to ensure the tests ran smoothly and were properly documented.</p> <p>Afternoon: In the afternoon, we wrapped up the testing and documentation process. I then shifted my focus to resolving the issues that had surfaced during the tests. However, since some of the problems required more time and investigation than anticipated, I wasn't able to fully address them today. I'll need to continue working on these issues next week and go through each one carefully to ensure they're resolved properly.</p>
	<p>Morning: I started work on setting up a laptop yet again for testing. Why? Because over the weekend I finally realized that we were supposed to actually pre render our website into static html/css/js files. After aiding in getting the SSR (Server Side Rendering) working, especially the ESLint</p>	<p>Morning: I don't know why but VSC just decided to change how I used to pull and push? So it took me a couple of minutes to understand and get used to it. Then we started working on SSR. When we tried to compile it we got tons of errors, so we got to fixing. I tried a lot and just couldn't get it.</p>	<p>Morning: This morning was spent on hotfixes and bug searching. We thoroughly examined the entire system to identify and resolve any outstanding issues. During this time, I began creating a download page where users can easily access our project files. Over the next few days, we plan to overhaul</p>

	<p>part, we finally got it to build the static web page, which caused it to finally work on browsers that never worked before. Even Internet Explorer, but it could not properly render anything at all.</p> <p>Afternoon: As I already touched upon, I tested a lot of browsers again. This time with better results. There are still slight issues in the Design on Chrome, but those are to be fixed very soon.</p>	<p>Somehow Luca just reverted it back to the old version and it just worked. I don't know how but if it works it works.</p> <p>Afternoon: I started working on the chat history. I spent my time researching and looking for ways to implement it. I got the most of, the only thing missing was I somehow had to get some variables to a file and I couldn't find a clever way to do it. When I asked Patrick about it he said "oh those are on the localStorage just get them from there". Oh my god, how didn't I think about that. Well now I know how I can implement it fully and will be finishing it tomorrow.</p>	<p>the documentation and the FAQ section to ensure that both are updated and user-friendly, reflecting the latest changes in the documentation and within the app itself.</p> <p>Afternoon: After the other two programmers encountered some challenges, I took the initiative to fix the <code>settings.tsx</code> file, as I am most familiar with this part of the codebase. I resolved the bugs and ensured that the command <code>npm run build</code> functions as intended, allowing for a smooth build process. Additionally, I collaborated with Patrick to test different browsers, where we discovered a crucial insight: "same issue as in Chrome." This is particularly important because Chrome has some unique styling issues that I still need to address tomorrow. Other browsers (aside from Firefox) are primarily Chromium-based, which means they share similar characteristics and may exhibit the same issues. After our testing session, I worked through the task list, prioritizing the most critical items, and prepared everything for tomorrow's work. This proactive approach will help us stay on track and ensure a productive day ahead.</p>
	<p>Morning: Today in the morning I brought a phone to test the responsive design of</p>	<p>Today: Today I started working on chat history. the first version I tried failed</p>	<p>Morning: Today, I arrived early to address our visual issues. Once I completed that</p>

<p>the webpage with a few different browsers installed. I then started work on getting a deployable Windows binary, which bundles nodejs, python and ollama for easy use.</p> <p>Afternoon: This whole deployable Windows binary was a lot more complex. The main issue was cmd, as powershell wouldn't run scripts without disabling "security" features, which made the entire thing a lot more difficult. Python too mostly didn't feel like cooperating, so I had to try a lot to get it to work. First of all, python 3.13 was released yesterday. Who could have guessed that most of the big libraries hadn't been compiled for Python 3.13 yet? So I then tried python 3.12. Kind of sad that they already default to giving out python 3.13. I'd usually do such switches gradually. Oh well. I then tried fixing a little bug with the auth key, which I shortened, because 4096 chars is overkill, and seems to have caused issues, especially with debugging.</p>	<p>miserably, so I took a step back, discarded all changes and started researching. I came to the conclusion to create a custom hook that manages data transfer between components. I got stuck on one problem. I just didn't know how to sync the histories. When I told Patrick, he asked me to show and guess what, It just worked. I don't know how but it works! Then I proceeded to add a button to the list so one can instantly create new chats. Now The only thing left is saving the chats.</p>	<p>task, I systematically went through each issue one by one. I reviewed everything thoroughly and discarded any minor elements that we had not been able to get working. In other words, I focused on cleaning up today and resolved all the lingering issues we were still experiencing.</p> <p>Afternoon: Since the responsive design was still not meeting my expectations, I employed some creative solutions to restructure the entire header. This adjustment allowed it to function as intended. After making these changes, I dedicated a significant amount of time to testing, fixing, and ensuring that everything was polished and ready for distribution.</p>
<p>Morning: I prepared the automated scripts for Linux, which worked really well, but I also had to do some more changes to them while testing, as I'd actually failed to do them correctly a few times.</p> <p>Afternoon: I kept going with the testing until it actually worked. I then worked on</p>	<p>Morning: I tried implementing the chat History but just couldn't get it and I kept starting over and over again, but I was making progress every time. getting closer and closer.</p> <p>Afternoon: I managed to implement it so it actually displays the chat. I had done some research and thanks to custom hooks I made it</p>	<p>Morning: Today is the last day before the presentation, so my focus is entirely on one task: bug testing. I am dedicated to ensuring that everything functions as intended. This involves identifying and deleting any non-functional elements and refactoring the code as necessary.</p>

	<p>finding some bugs and fixing them. At home I fixed the chat history saving to the backend, and the weather API. Which surprisingly enough did actually work.</p>	<p>work. Then we fixed some bugs and errors that came up because of npm run build. I made the selected chat highlight so it's clear which one you are communicating with, with the ai. Finally I managed to save the chat history in memory, so if you fully reload the page etc. it goes away but you can change between chats and the old one remains. I am going to implement saving the history in the database at home.</p>	<p>Afternoon:</p> <p>The process continues: I test the application, identify any bugs, and resolve them. After fixing the issues, I run the build command (<code>npm run build</code>), address any errors that arise, and then repeat the cycle.</p>
	<p>Morning: I prepared the Windows testing machine, which sadly failed. I also squashed some of the last bugs that we had. We then did the presentation, which went pretty well all things considered.</p> <p>Afternoon: We discussed how we'd worked on the project, which we got a very good score on. After that we came together to reflect a bit on these past 4 weeks, and also discuss some future projects we want to do, because working together on this has been extremely fun.</p>	<p>Morning: I checked the history saving at home and I managed to make it save in the same file as our settings. Then I wanted to start on updating the page to show whatever chat the user chose. But then Patrick said that it somehow already works and it actually did. So I started working on deleting chats. First I made the design, buttons and after the functionality. Oh boy was that close. We had a meeting set up at 11:00 AM I barely managed to make it work at around 10:45? My problem was that the Index of the selected chat was not updating fast enough or after another variable update. but it has to first because: we track on which chat the user is active on right now via a variable that saves the Index of the current chat. and just displays the data from that said index in the array. With deleting we have to make sure what happens if the deleted chat comes before the current active</p>	<p>Morning: I finalized the presentation, focusing on eliminating the last remaining bugs in our system. Now, I can only hope that everything goes smoothly during the presentation of our project.</p> <p>Afternoon: I addressed some minor bugs and engaged in discussions with Patrick and Yasin about our final steps for this project. We also talked about our future plans and how we want to collaborate as a team moving forward.</p>

		<p>chat. Let's say we are active on chat 5 and we delete chat 4 now the 4th chat is chat 5. but like I said we aren't on that index anymore. The same happens if we want to delete the chat we are currently active in. At 11:00 AM we had our meeting where we presented our project to everybody in the firm and it didn't go well at all. Stuff that worked earlier just didn't work. We had many situations where the ai just didn't respond the way it did in testing. After our meeting we had pizza.</p> <p>Afternoon: We first sat together and processed our thoughts about how the presentation went. After that, we started brainstorming about future projects. Which programming language? What do we want to program? etc.</p>	
		<p>Morning: Today we set our focus to fix little errors and generally sort the little things out. That's why the first thing I did was fix the chat actions. The resend and edit button didn't</p>	

Appendix:

Task:

[AI Virtual Assistant - Internship Students 2024](#)

Code:

[Repository](#)

License:

[AGPL 3.0 or later](#)

Additional files:

[Buglist:](#)