User documentation

Admin / entrepreneur user

Usage

To use the Ailixir application, you first need to specify input sources to scrape data from. A `config.json` file will be generated from them. Then, we start the data acquisition pipeline by executing the orchestrator component, which manages this process. Ensure that the specified data sources are accessible (free to use), as they significantly impact the AI's data quality and overall performance.

Configuration

Before using the main commands in the following section, update the scraping targets in `src/backend/Config/main.py` for example like this:

```
if __name__ == '__main__':
    c = Config()
    c.add_target(YouTubeTarget(
       url='https://www.youtube.com/@NutritionFactsOrg')
    )
    c.write_to_json()
```

Data Sources

Here is the list of available data scraping targets with example usage:

- Recipes from allrecipes
- Articles from arXiv
- Podcasts from the Peter Attia Podcast
- Articles from PubMed
- Youtube channels (retrieve information from all videos on the channel)
- Nutrition related blog posts from NutritionFacts

```
# Without any parameters, it scrapes from allrecipes.com
AllRecipesTarget()
# The first argument specifies which keywords to look for
# and the second one limits the number of results for a set of keywords
ArchiveTarget(keywords=['nutrition', 'health'], max_results=1)
# The first parameter is the base URL link for scraping, don't change it
# and the second one limits the number of podcasts to be retrieved
```

```
PodcastTarget(
   url='https://peterattiamd.com/podcast/archive/', num_podcasts=1
)
# The first argument specifies which keywords to look for
# and the second one limits the number of results for a set of keywords
PubMedTarget(keywords=['food', 'exercise'], max_results=1)
# The first parameter specifies the channel to scrape data from
# (all videos of this channel will be scraped)
YouTubeTarget(url='https://www.youtube.com/@NutritionFactsOrg')
# The first argument specifies the base URL link, don't change it
# and the second one limits the number of pages to scrape
NutritionTarget(url='https://nutritionfacts.org/blog/', max_pages=1)
```

Running the Data Acquisition Pipeline

To start the data acquisition pipeline, execute the following commands:

```
# Build data/config.json file and auxiliary folder structure
pdm build-config
# Scrape all targets specified in src/backend/Config/config.py
pdm run-orchestrator
```

The data will be stored in the `data/` directory. For example for youtube scraped data, you can find the scraped data under `data/youtube/raw/`. Unique ids of scraped youtube videos are stored in the file `data/youtube/index.json`.

Environment Variables

API keys and other sensitive information cannot be stored in the public code repository. When running the application for the first time, the user will be prompted to provide these values. The environment setup script ensures these values are stored in a `.env` file. The data acquisition pipeline only requires the `YOUTUBE_DATA_API_V3` key. We obtained these keys from our industry partner.

```
YOUTUBE_DATA_API_V3=""
GOOGLE_GEMINI_API=""
OPEN_AI_API=""
ASTRA_DB_API_ENDPOINT=""
ASTRA_DB_TOKEN=""
ASTRA_DB_NAMESPACE=""
ASTRA_DB_COLLECTION=""
FIREBASE_API_KEY=""
FIREBASE_AUTH_DOMAIN=""
FIREBASE_PROJECT_ID=""
```

```
FIREBASE_STORAGE_BUCKET=""
FIREBASE_MESSAGING_SENDER_ID=""
FIREBASE_APP_ID=""
FIREBASE_MEASUREMENT_ID=""
GOOGLE_AUTH_CLIENT_ID=""
```

Additional Commands

Scraping

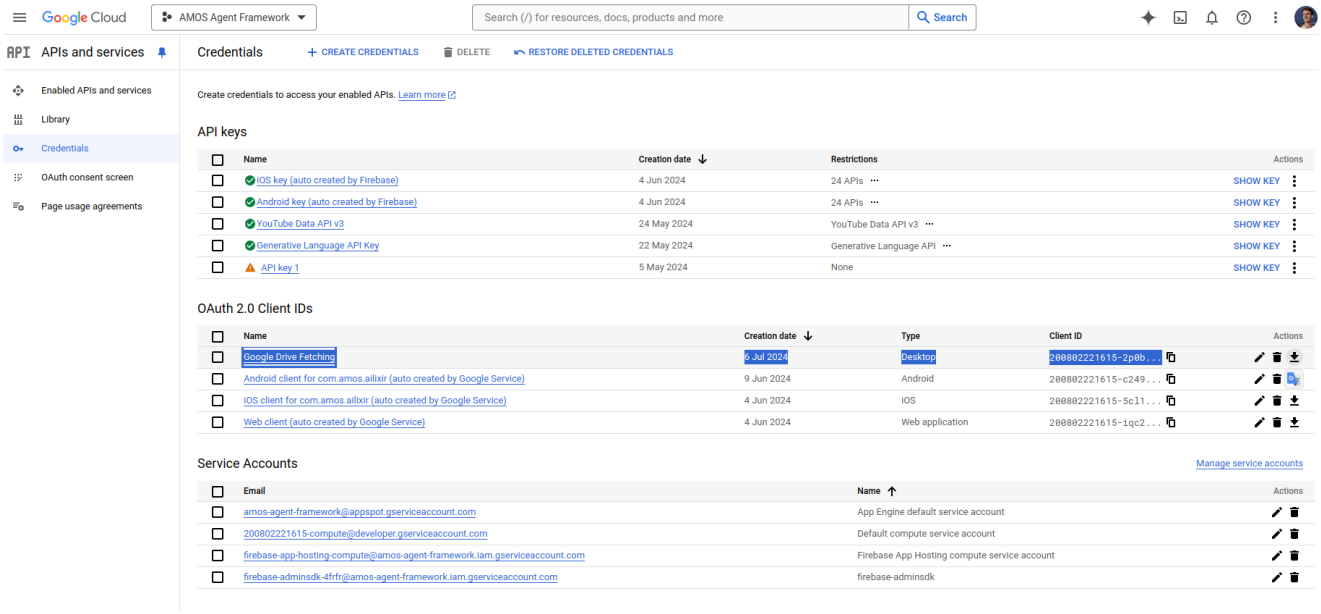
To perform specific scraping tasks, execute the corresponding commands:

```
# Scrape AllRecipes
pdm scrape-allrecipes
# Scrape Podcast
pdm scrape-podcast
# Scrape PubMed
pdm scrape-pubmed
# Scrape YouTube
pdm scrape-youtube
# Scrape Archive
pdm scrape-archive
# Scrape NutritionFacts
pdm scrape-nutritionfacts
```

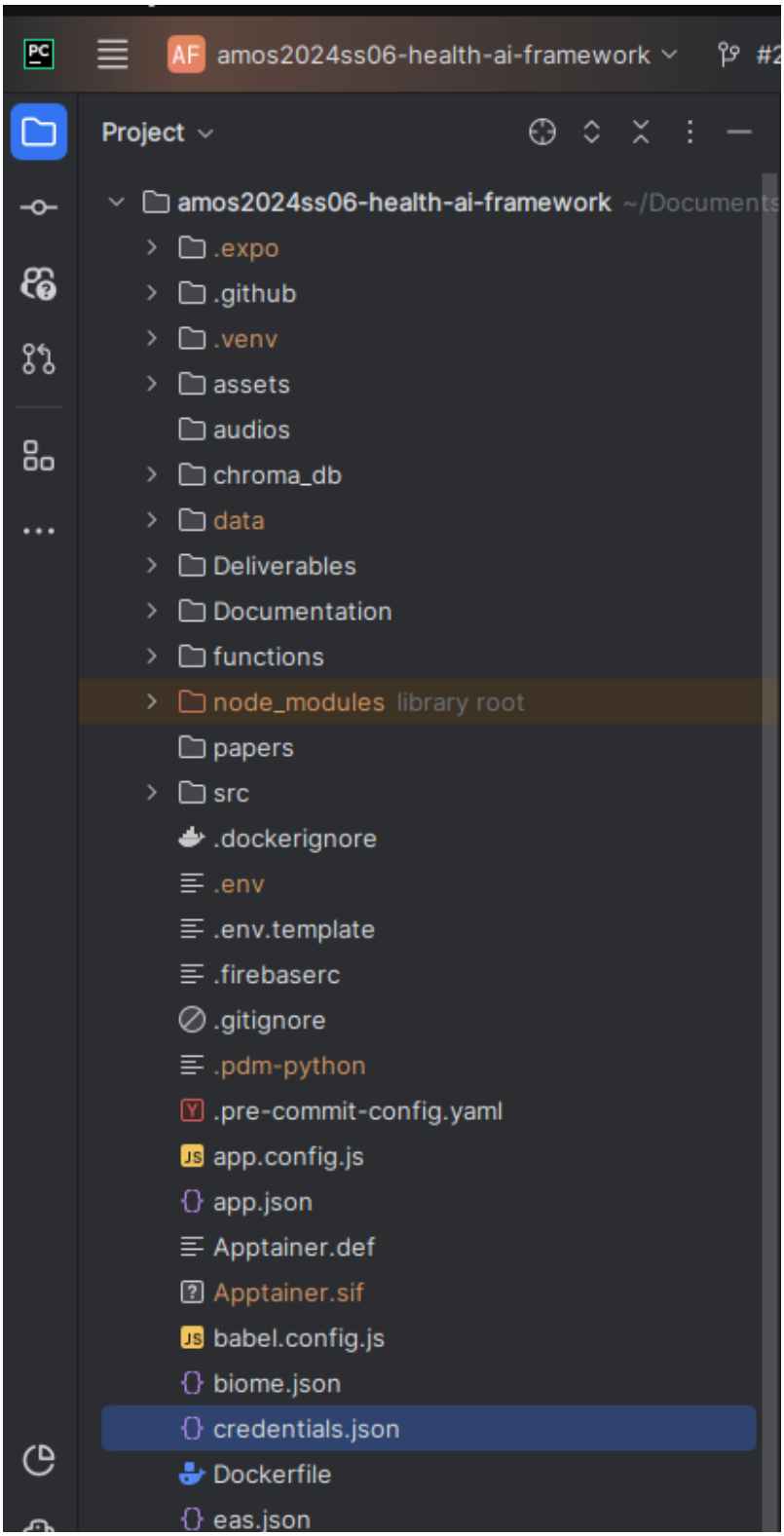
These commands will execute the code found in `main.py` for each respective scraper (e.g. `src/backend/Scrapers/Archive/main.py`). You can customise and try out code for the different scrapers there.

Initialization Prompt (GDocs)

In order to deploying various instances of the app to perform as different types of agents, we can edit Google Document and one Python script to access the new background info to our LLM. To do so we first need to download a new Google Drive Fetching Credentials from Google Console by clicking on the download icon.



After downloading the file should be renamed to `credentials.json` and pasted to the root of the project.



Now let's get to the google file itself. After clicking the following link for the first time, some developer / owner should approve the editing of the Initalisation Document. The document then should look like this.

Please act as a health advisor who provides tips on how to improve ones heals through changes in lifestyle. Please remind the user, you are not a doctor, and any advice given should not replace professional medical advice. Always consult a physician before making any significant changes to lifestyle or health routines.

To tailor the advice specifically to the user, ask the user to provide some personal details by asking the following questions:

1. Have you received any diagnosis that you would like to support with lifestyle changes? If so, please describe.
2. What is your sex?
3. How old are you?
4. What is your current weight?
5. Are you regularly doing sports and if so what and how often?
6. Do you have any dietary restrictions or preferences?
7. Are you allergic to any foods or substances?

Based on the information provided, you offer personalized advice regarding nutrition and exercise. Make sure to include references from credible sources such as research papers, podcasts, YouTube content, or blog posts to support the recommendations.

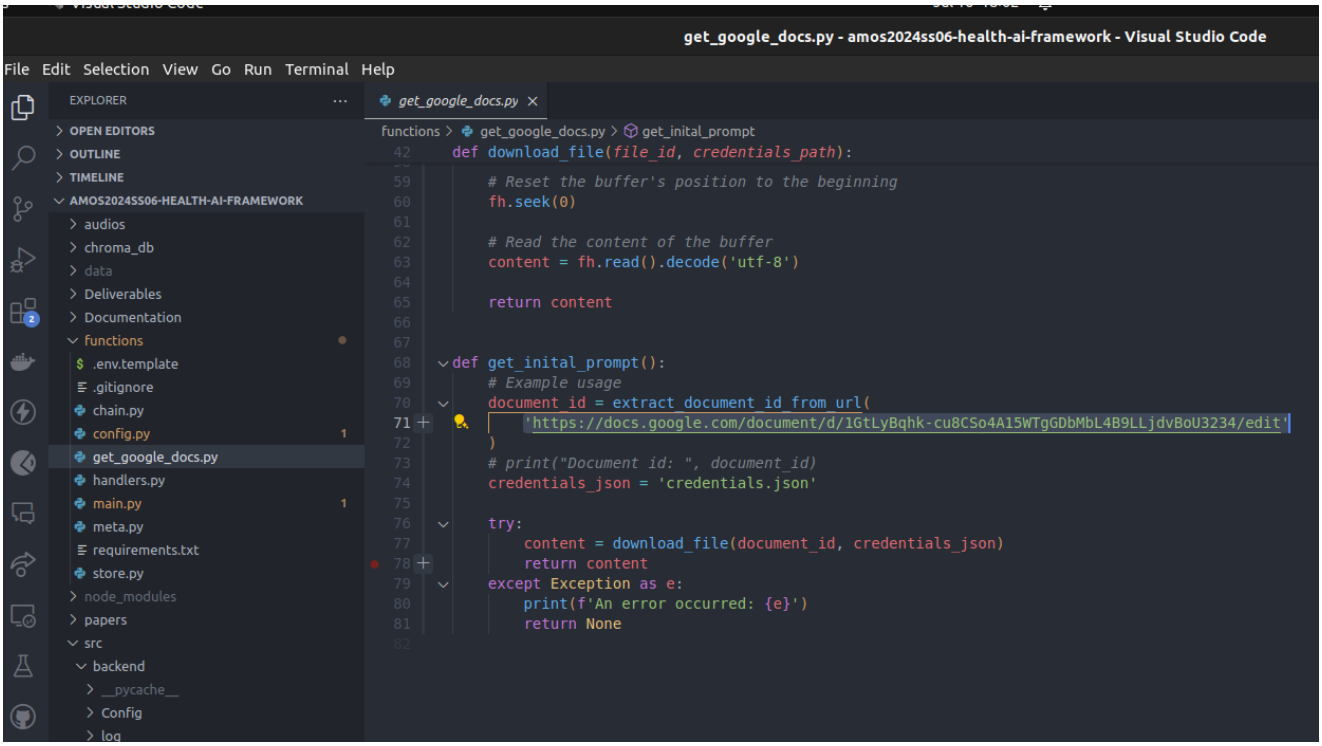
Let's get started! Please share the details mentioned above.

After writing the basic prompt and some additional context the new info can be uploaded into our LLM using following python command in the root of our project. The result is save in the folder `/data/google_docs_content.txt`.

```
pdm google-docs
```

Changing GDocs Link

If an entrepreneur wants to change the Google Docs link, he needs to first create one and then update python script `function/get_google_docs.py` as seen in the following snippet.

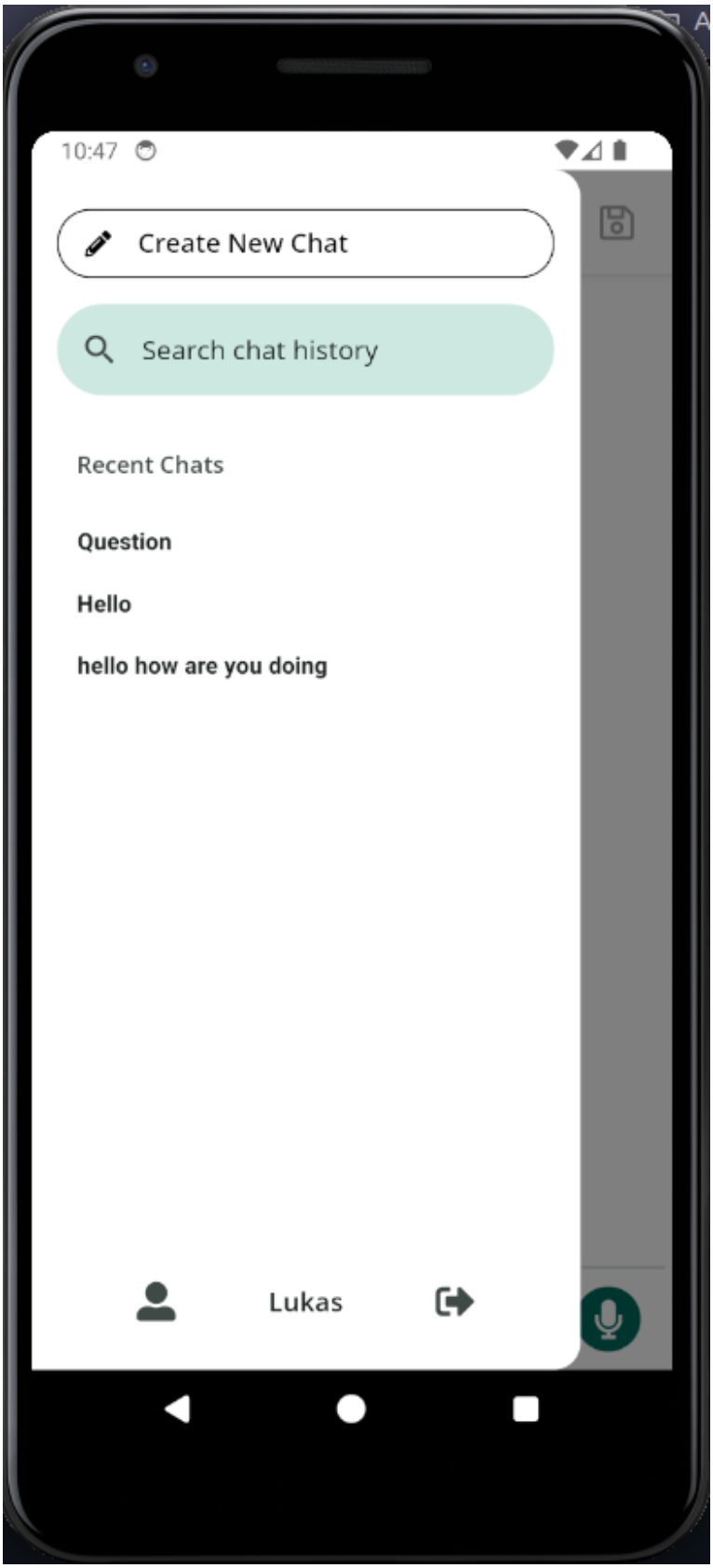


User documentation - End user

Application

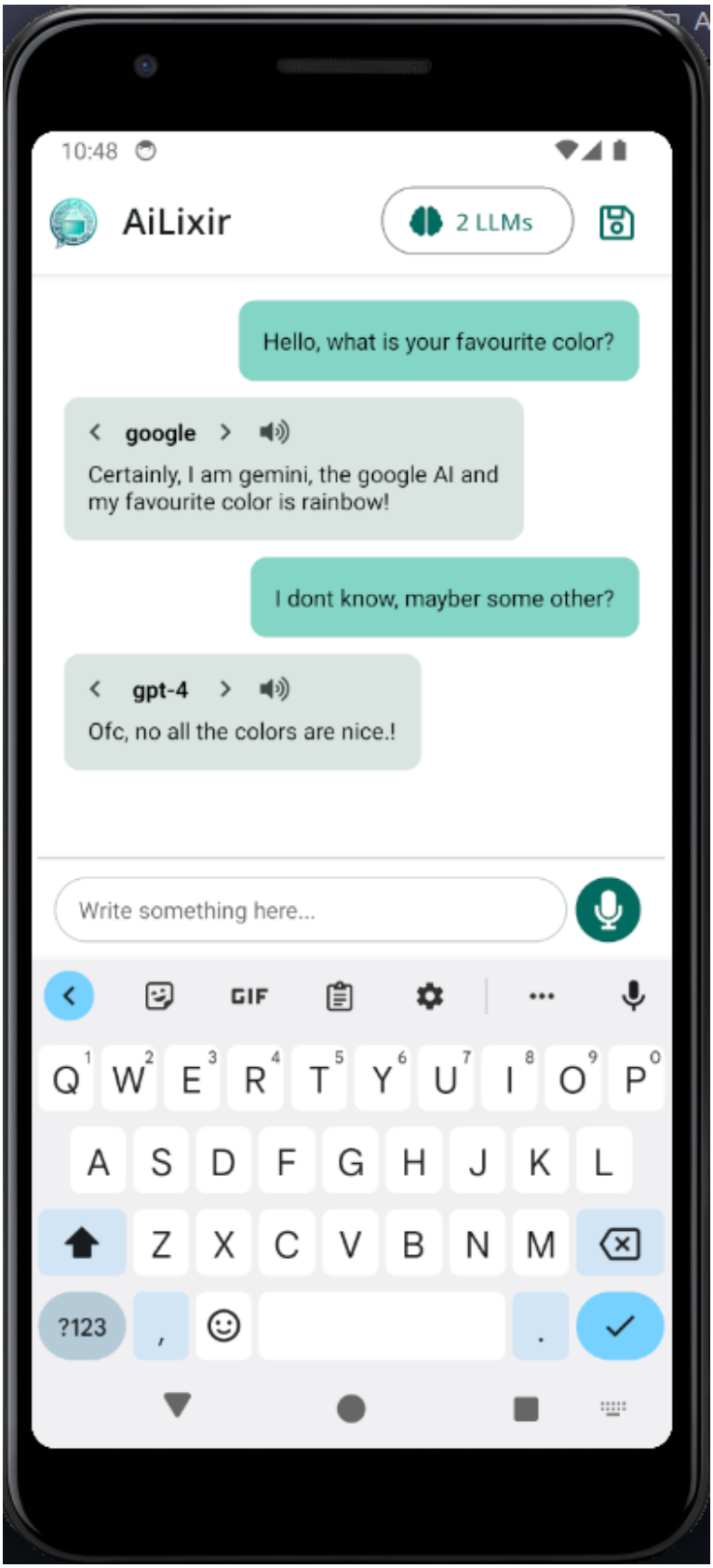
Navigation Drawer

A user can open drawer by an icon in the left top corner or by sliding from left side. Then the user can find a previous chat by typing in the textfield, or look in 'Recent Chats'. The user can also switch between specific BOT configuration, but this functionality might not be integrated in the future. Last but not least the user can see his login name in the left bottom corner of the drawer and can also either access profile setting by clicking a 'User' icon or log out out of the app using a 'Logout' icon.



Chat Screen

A user can click in the textfield on the Chat Screen to access keyboard. After writing message and hit a 'Enter' button, the message is sent and displayed in the screen with lila color on the right side. Reply from our AI-Agent has grey color and is located on the right side.

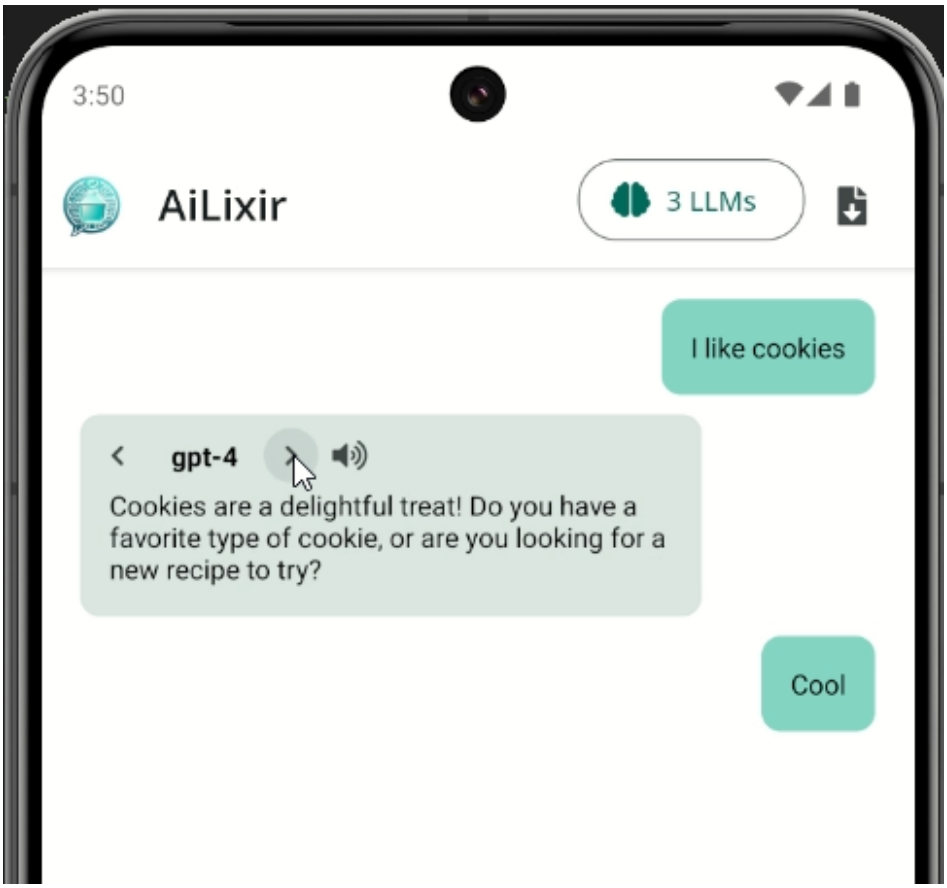


Dropdown for LLMs options

A user can choose which AI models will be used for generating answers by selecting one or multiple of the options in the dropdown. At least one LLM model needs to be selected. If one is selected, its name is then displayed in the button text. If more are selected, number of LLMs is showed to the user.

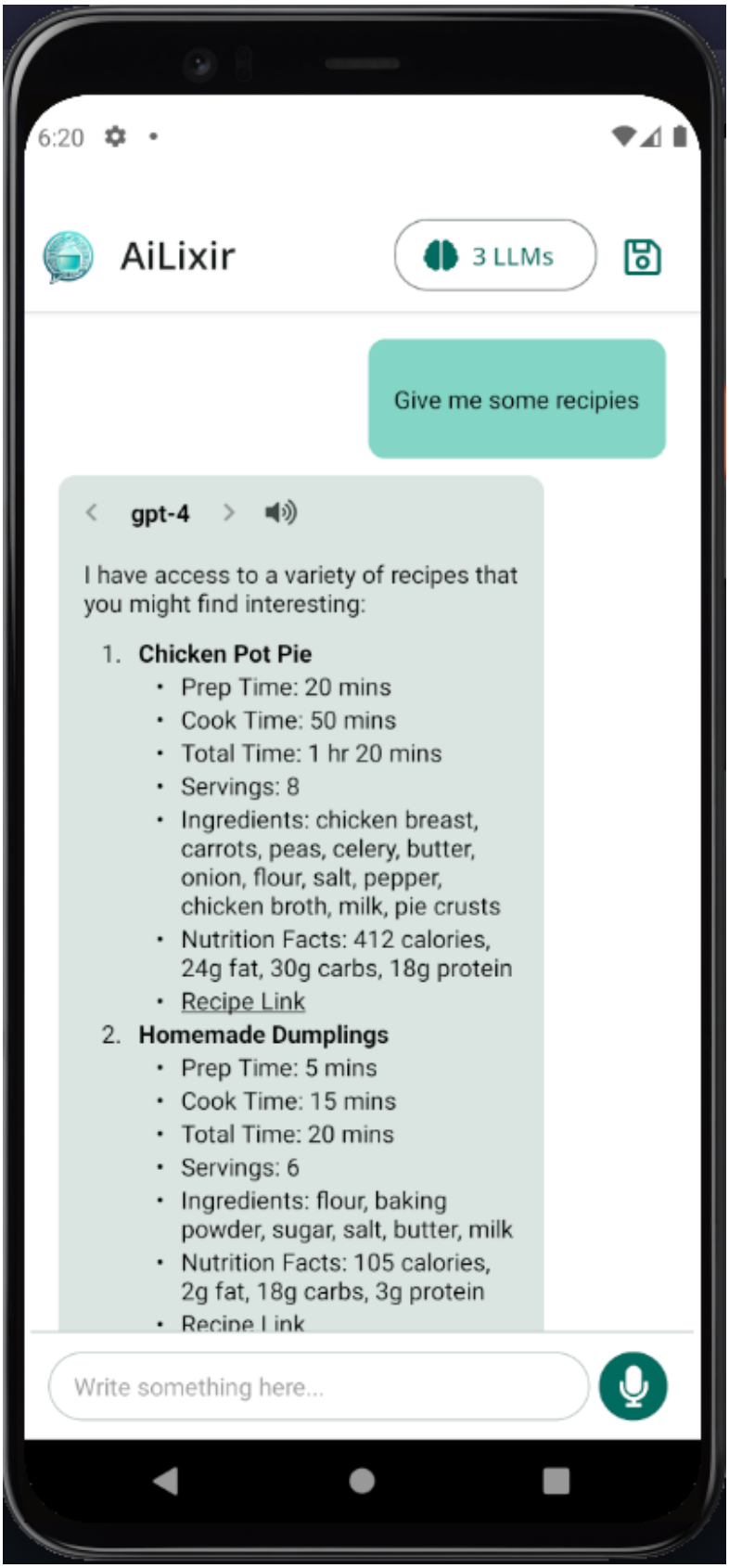
Chatting with multiple LLMs

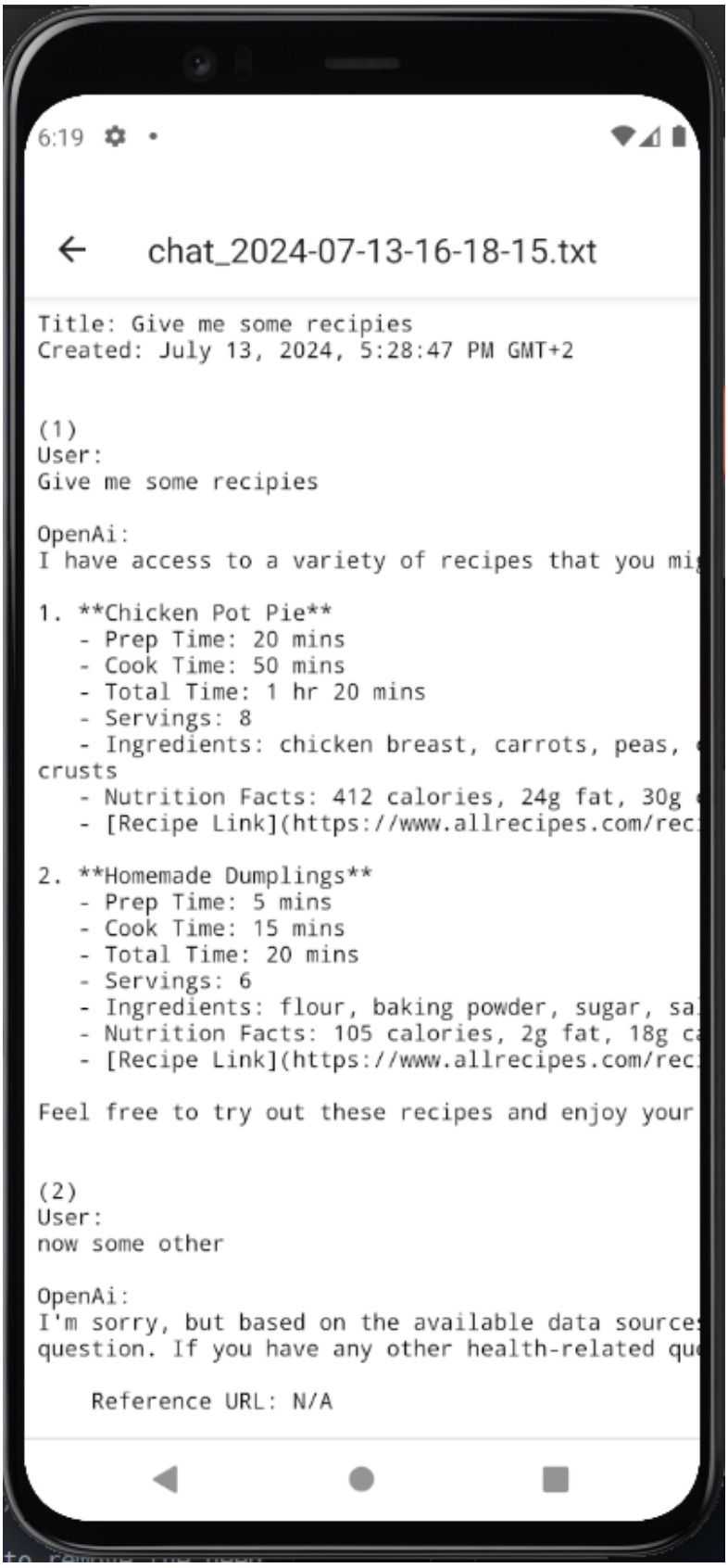
Prompts will be sent to all LLMs that were selected in the Dropdown. You can see different answers by cycling through them.



Saving Chat

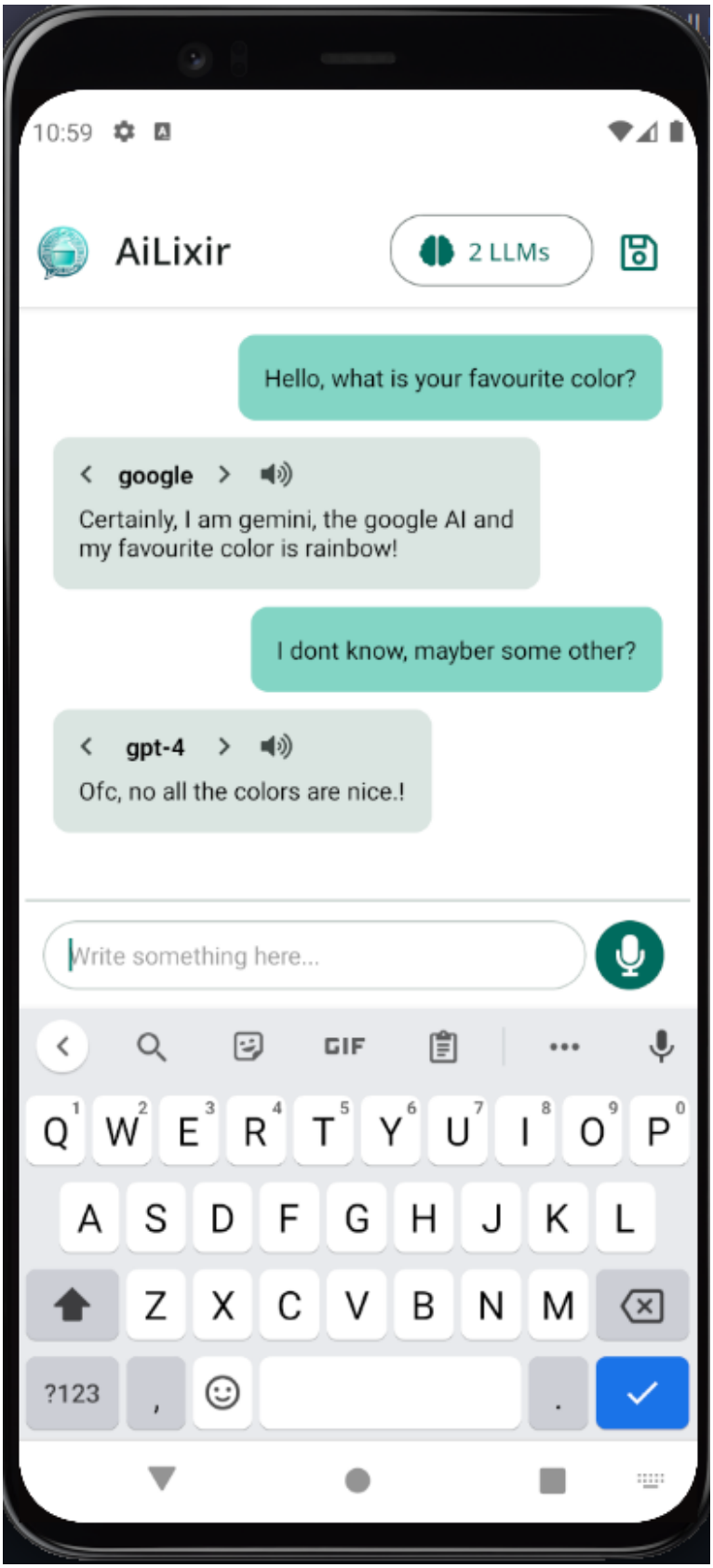
A user has also opportunity to save the chat directly into the device in `.txt` format. After clicking on the save icon in the right upper corner the chat is saved into `Downloads` folder on the device and also copied to the clipboard. Outputed text includes some basic info about the conversation and also questions and answers itself.

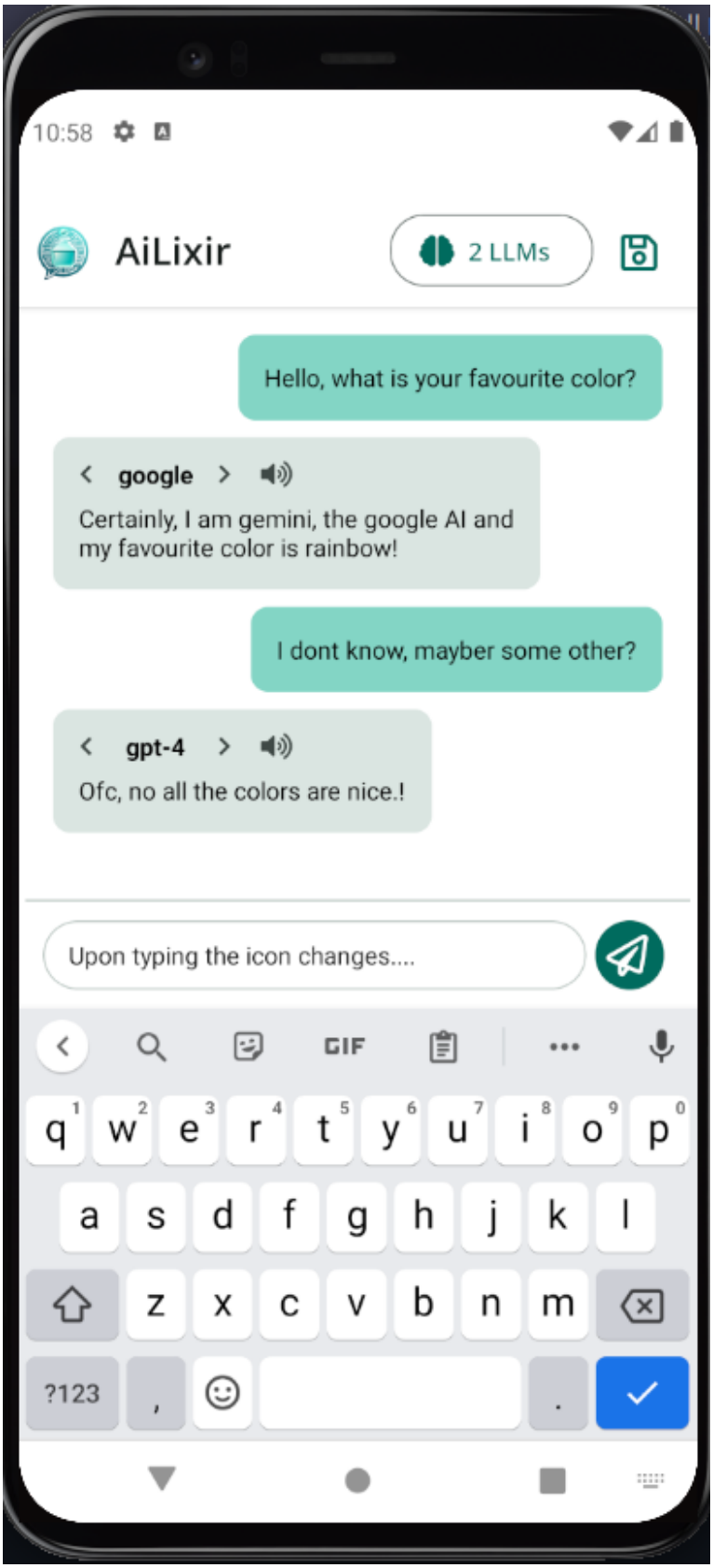

6:20
AiLixir
3 LLMs
Give me some recipies
gpt-4
I have access to a variety of recipes that you might find interesting:
1. Chicken Pot Pie
• Prep Time: 20 mins
• Cook Time: 50 mins
• Total Time: 1 hr 20 mins
• Servings: 8
• Ingredients: chicken breast, carrots, peas, celery, butter, onion, flour, salt, pepper, chicken broth, milk, pie crusts
• Nutrition Facts: 412 calories, 24g fat, 30g carbs, 18g protein
• Recipe Link
2. Homemade Dumplings
• Prep Time: 5 mins
• Cook Time: 15 mins
• Total Time: 20 mins
• Servings: 6
• Ingredients: flour, baking powder, sugar, salt, butter, milk
• Nutrition Facts: 105 calories, 2g fat, 18g carbs, 3g protein
• Recipe Link
Write something here...



Voice Input and Output

In the chat interface, users can push and hold the microphone icon to record a voice message which will consecutively be transcribed to text. While pushing the button the icon changes color. Next to the LLMs responses the user can push a button to get the message content read out by a synthetic voice (with valley girl prosody). Upon typing the microphone icon changes to the paper plane icon:


10:59
AiLixir
2 LLMs
Hello, what is your favourite color?
google
Certainly, I am gemini, the google AI and my favourite color is rainbow!
I dont know, mayber some other?
gpt-4
Ofc, no all the colors are nice.!
Write something here...
GIF
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
?123


10:58
AiLixir
2 LLMs
Hello, what is your favourite color?
google
Certainly, I am gemini, the google AI and my favourite color is rainbow!
I dont know, mayber some other?
gpt-4
Ofc, no all the colors are nice.!
Upon typing the icon changes....
GIF
?123