Case study Farie AG

Task:
In this case study we would like you to write a crawler with selenium[1] in a Jupyter Notebook that fetches all listings for Zurich from homegate.ch. For this you can send your crawler directly to the URL: https://www.homegate.ch/rent/real-estate/city-zurich/matching-list?ep=1. Crawl from here all listings on all pages for Zurich. Save the crawled data in the database, which you can find in the attachment. Run the crawler on two different days and make sure that no duplicates make it into the database.

Database:
The database only contains one table. The columns are as follows:

listing_id int PRIMARY KEY	->	The id of the listing, can be found in the listings link e.g. https://www.homegate.ch/rent/3002199679
price float	->	Rent price
size float	->	Size in m^2
rooms float	->	Number of rooms
address str	->	Address
extraction_date str	->	Date the listing got crawled

Hint:
One way to go through the pages would be to replace the number at the end of the link with the number of the page you want to crawl.
https://www.homegate.ch/rent/real-estate/city-zurich/matching-list?ep=1
https://www.homegate.ch/rent/real-estate/city-zurich/matching-list?ep=2

.

.

https://www.homegate.ch/rent/real-estate/city-zurich/matching-list?ep=n

n = max page

Presentation:
Guide us through your code.
Show us what thoughts you had.
Take us through your ETL process
What challenges have you encountered?
What was easy to implement?
Imagine crawling 70 data points per listing, how would you model that database?

[1] https://selenium-python.readthedocs.io/