File organization:

- app.py contains all the backend development.
- docker-compose, Dockerfile, Makefile, requirements,files contains all the settings necessary to run the service
- README contains the instructions.

Server behavior:

- the server will receive a long_url and create a short version;
- Each time a new request to short a long url is made, a entry on the database is either created or a count related to that long url is incremented
- Each time a request to access the short url is made, a log is saved on the database.

Database Structure:

We have two main tables on the service:

- urls:
  - id: id entry
  - long_url: original long url
  - short_url: shortened url
  - creation_attempts: number of times the request to short the long_url was attempted
  - access_count: number of times the short_url was accessed using our server
  - created_at: initial time the entry was created
  - updated_at: last updated on the entry
- url_analytics:
  - id: id entry
  - url_id: id reference to the original url entry on the table urls
  - ip_address: ip address of the access
  - browser: browser used for the access
  - created_at: time of the access

Development Timeline (around 4 hours):

- initially a simple service was made to receive a url and create a short version of it.
- a database structure was created to store the long url and short url.
- docker and makefile were set up for deployment.
- url_analytics table were included.
- code was revised, adding sqlAlchemy and other improvements.
- readme and documentation were created.
- some fixes in the behavior were done (only update analytics when a access request is made and other ones)