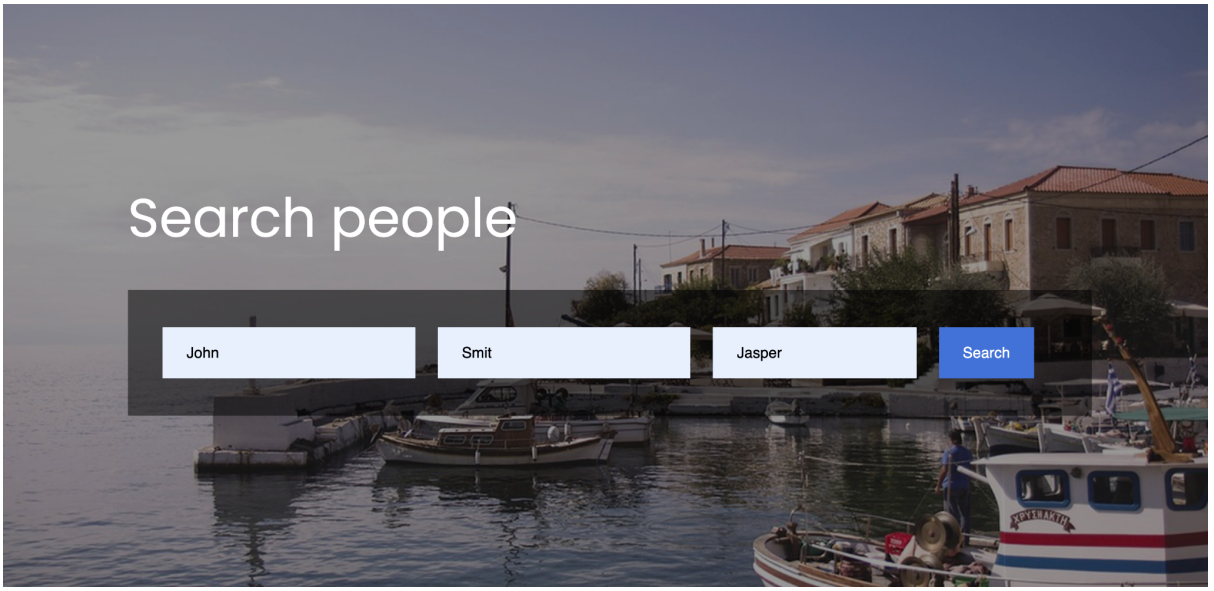
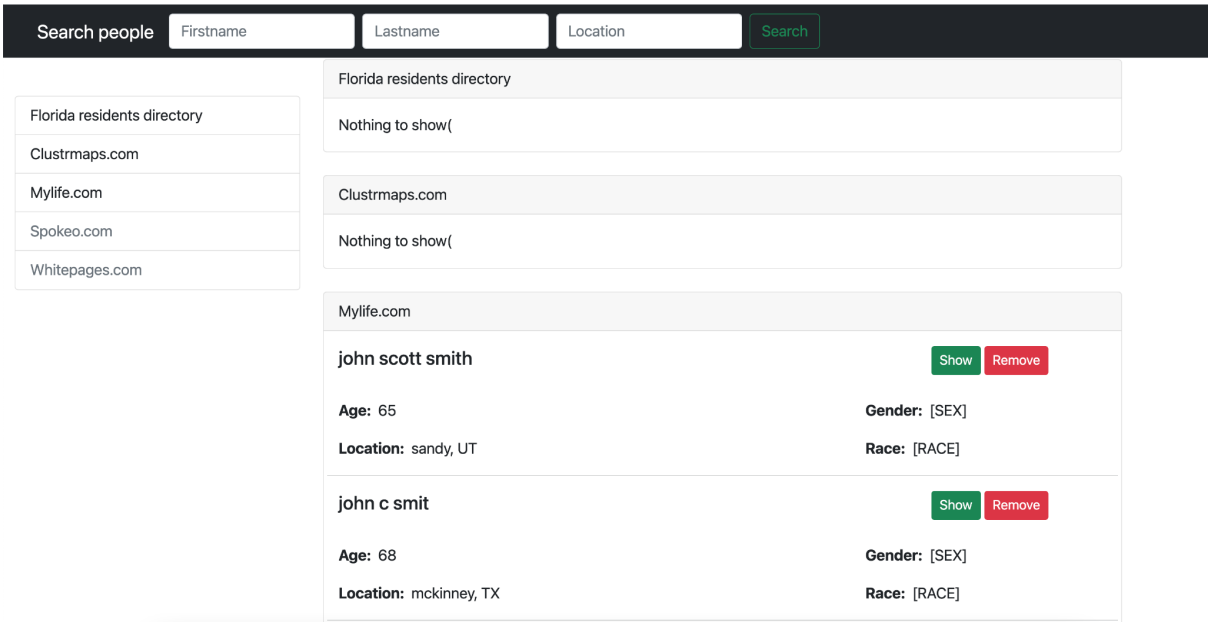
ТЗ

Описание
Есть информационные сайты, которые хранят информацию о людях проживающих на территории страны. Они предоставляют доступ к подробной информации о них. Согласно законодательству такие сайты должны также предоставлять возможность удалять информацию (opt-out страница). Цель - написать парсер (ы), которые парсят эти сайты, находят информацию и удаляют ее.

Ниже скрины mvp приложения, которое уже существует php + JS (node) . В самих парсерах или приложении нет никакой сложной логики. Существующий код очень простой и понятный, но при желании можно делать с нуля.





Задачи:

1. Приложение должно быть на Python или PHP
2. Система аутентификации на сервере. Регистрация, Логин, Логаут.
3. На этих сайтах стоит защита. Просто так открыть их не удастся. Нужно использовать residental proxy (будет предоставлен), чтобы скрипт мог открыть сайт. Так же необходимо

симулировать поведение пользователя, чтобы сайт не распознал, что это бот. Для этого используется headless browser (например puppeteer) и всякие уловки. Все это практически реализовано и тут нужно только доработать это.

4. Написать такие парсеры для примерно 10 сайтов. Все они однотипные и по большей части это будет копи паст.
5. Самая сложная часть - opt-out страницы. Страница на которой отправляется запрос на удаление данных о человеке. На этой странице почти всегда стоит капчта и верификация email адреса и/или верификация по телефону. Нужно, чтобы скрипт умел решать капчу (такие сервисы как https://www.deathbycaptcha.com/) и верификаю емейла (поднять свой мейл сервер или может есть какие то сервисы или использовать API каких нибудь почтовых серверов) и верификацию телефона. Верификация телефона есть не везде + существуют сервисы позволяющие это делать.
6. Так как все парсеры должны одновременно запуститься и начать искать на всех сайтах , то придумать архитектуру так, чтобы одновременно запущенные парсеры не положили сервак.
7. Веб интерфейс должен быть похож как на скринах. Верстка на bootstrap 4,5
8. Если делать с нуля, то прикрутить docker, чтобы сделать docker-compose up и все начало работать.
9. Код должен быть простой и понятный

Сайты, которые надо парсить:
FastPeopleSearch
Search People Free
MyLife.Com
Whitepages.com
Advancedbackgroundchecks.com
nuwber
Spokeo
Clustrmaps.com
Beenverified.com (and neighbor who)
Florida Residents Directory

Remove pages :
https://www.fastpeoplesearch.com/removal/search
https://www.searchpeoplefree.com/opt-out
https://www.mylife.com/ccpa/index.pubview
https://www.whitepages.com/suppression-requests
https://www.advancedbackgroundchecks.com/removal
https://nuwber.com/removal/link
spokeo.com/optout
https://clustrmaps.com/bl/opt-out
https://www.beenverified.com/f/optout/search
https://www.floridaresidentsdirectory.com/opt-out

Далее подробно о том как работает существующее mvp приложение.
Приложение написано на php (symfony). Пользователь заходит на сайт, вводит имя фамилию ,локацию и нажимает search. Открывается страница результатов (второй скрин) и одновременно отправляются ajax запросы на сервер. По одному ajax на каждый сайт (и его парсер). Это сделано для того чтобы не приходилось ждать пока сервер найдет все данные, а показывал результаты по мере того как будут спаршены разные сайты.
Когда пришел запрос сервер запускает nodejs скрипт ответственный за парсинг отдельного сайта и

выдает результат работы скрипта. Скрипты запускают headless browser chrome через puppeteer. И через прокси открывают сайт. Симулируют поведение человека и парсят данные.