*Документация*

*Кратко описание:*  
Приложението е изработено на PHP с Laravel Framework.  
Използваните допълнителни технологии са:  
- Docker  
- MailTrap  
- MySQL Workbench  
- Bootstrap  
- Javascript  
- Blade Templates  
- Eloquent ORM  
- Composer  
- Sail

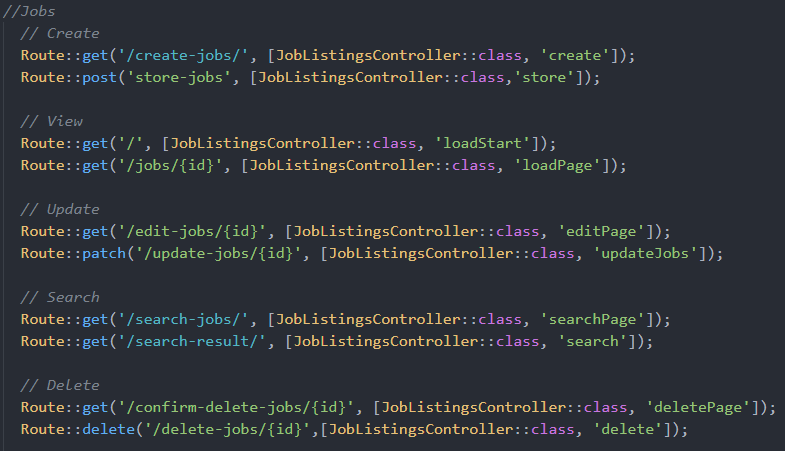
*Функционалности:*  
- Регистрация като кандидат или като работодател  
- Преглед като гост  
- Създаване, редактиране, посещаване и кандидатстване по обяви  
- Категориите и таговете са „динамични“ – т.е. имат двойна функционалност, но в зависимост от посещението имат различна функционалност  
- Динамично търсене, което проверява за съвпадение по категория, потребител, заглавие на обява, ключова дума, част от ключова дума, част от заглавие  
- Изпращане на имейл до работодател след кандидатура  
- Съхраняване на кандидатурите и възможност за преглед на позиции, за които кандидат се е записал

*Особености:*

~ За включването на контейнерите, трябва настроен *WSL* и инсталирано *Ubuntu* дистро. Включването на контейнерите става чрез посещение на директорията jobs и пускане на командата **“bash vendor/bin/sail up”.** Препоръчително е да се направи alias на командата в WSL терминала чрез командата **”nano ~/.bashrc”** и добавяне на **”alias sail=bash vendor/bin/sail”**, за да се създаде асоциация.  
~ *.env* файл с желаните променливи за потребителя като потребителско име за базата, парола и доставчик за имейла.  
~ *dockercompose.yml* или *dockerrun* файл за пущането на контейнерите, изтеглянето на „образите“ и създаване на обособено хранилище за всеки контейнер.  
~ Приложението може да бъде стартирано:  

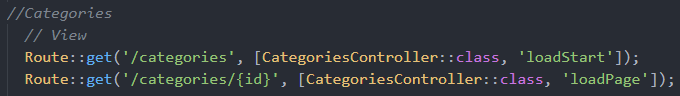

*Реализация:*

Самите HTML образци се възползват от Blade Template-изацията и са разделени в отделни файлове, като има един общ файл mysections. Данните за генерирането на файловете са подавани от контролерите, като пък контролерите използват съответните Eloquent модели.

**~ JobListingsController:**  
В **web.php** са създадени следните routes/endpoints:  
  
Реализирани са динамични (и статични) крайни точки за контролера със съответни функции и съответната HTTP заявка:

-Например функцията “create” само връща съответния blade template view файл, а запазването на работата става в store функцията. Има реализирани проверки за потребителите.



**~ CategoriesController** има няколко вградени функции за валидиране на категориите и няколко публични функции за браузване по категории със следните крайни точки:  


**~ JobCategoryController:**

- няма крайни точки – действа като посредник между категориите и обявите и има собствен модел, като целта е да се достигне трета нормална форма. Грижи се за динамичното изтриване, пресъздаване и обновяване на категориите.

**~ CandidacyController:**

**-** грижи се за запазване историята на кандидатурите и като хранилище на даден потребител.

**~ Mail/CandidateMail:**

-генериращ се имейл, който се изпраща при създаване на нова кандидатура, реално се управлява от JobListingsController-a.