Моя интерпретация технического задания:

Требуется разработать систему состоящую из нескольких частей:

1. Пользовательского интерфейса для сайта бизнеса доступ к которой будет у любого посетителя сайта, который хочет устроиться в компанию на сайте которой он находится. С помощью этого интерфейса в виде чата, соискатель генерирует для себя сопроводительное письмо. Для этого пользователю необходимо отправить в чат свое CV(резюме) и описание желаемой вакансии, после чего система сгенерирует сопроводительное письмо на основе предоставленных данных. При этом, резюме кандидата сохраняется в базе данных компании(CRM Bitrix).

2. Пользовательского интерфейса для HR менеджеров компании, которые работают непосредственно с уже существующим интерфейсом Битрикс. Этот интерфейс необходимо дополнить кастомным виджетом, который также будет представлен в виде окна мессенджера. HR менеджер отправляет в чат этого мессенджера описание вакансии, на которую необходимо нанять человека. В ответ чат-бот ищет подходящее резюме из списка тех, которые загрузили соискатели в пункте 1. Если подходящего резюме нет, система генерирует демонстрационное резюме по указанному шаблону.

3. Со стороны бекенда функционалом пункта 1. занимается скрипт на Python, который, используя REST API ChatGPT, генерирует сопроводительные письма на основе данных полученных с помощью REST API Bitrix.

4. Со стороны бекенда функционалом пункта 2. занимается скрипт на PHP, который, используя REST API ChatGPT, генерирует демонстрационное резюме а также находит подходящее под вакансию резюме используя REST API Bitrix.

Предлагаемый выбор технологий:

1. Фронтенд

Для создания пользовательского интерфейся предлагается использовать HTML, CSS, JS и React фреймворк. React фреймворк позоволит обеспечить модульный, читаемый код, который легко подерживать и переиспользовать.

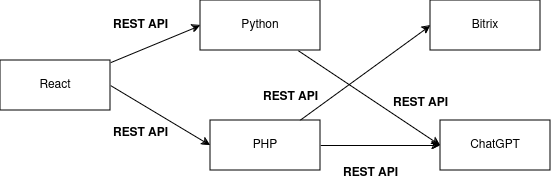
2. Связь между фронтендом и бекендом предлагается сделать в форме REST API. Это значит, что как для PHP, так и для Python скрипта будет необходимо реализовать свое REST API. Это позволит обеспечить независимость бекенда от пользовательского интерфейса. Также это обеспечит достаточную масшабируемость в будущем, для расширения функционала приложения. В случае Python предлагается использовать Fast API. В случае PHP предлагается использовать Slim. Оба фреймворка позиционируются как легковесные фреймворки для создания REST API.

Архитектура приложения

- Пользовательский интерфейс на React

- Python REST API

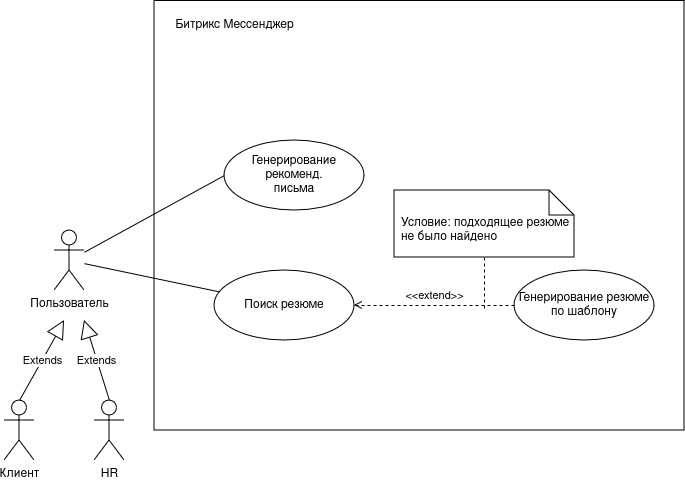
- PHP REST API



План реализации:

1. Разработать документацию REST API.
2. Разработать REAST API на Python.
3. Разработать REST API на PHP.
4. Разработать React интерфейс.

По моему мнению разработка сначала документации вместе с фронтенд коммандой является необходимостью, что позволит фронтенд команде утвердить формат и желаемый способ взаимодействия с серверной частью приложения и работать параллельно с бекенд командой, разрабатывая утвержденную документацию.

Сценарии использования:

Тестирование php:

<?php

require\_once 'src/Resume.php';

require\_once 'src/BitrixAPI.php';

use App\Models\Resume;

use App\Services\BitrixAPI;

function getCV(string $vacancyDescription): Resume {

try {

$cvList = BitrixAPI::searchCV($vacancyDescription);

} catch (Exception $e) {

error\_log("API Error: " . $e->getMessage());

return generateDemoCV();

}

if (empty($cvList)) {

return generateDemoCV();

} else {

return new Resume($cvList[0]->name, $cvList[0]->skills, $cvList[0]->experience);

}

}

function generateDemoCV(): Resume {

return new Resume("Демо Имя", "Навыки: PHP, MySQL", "Опыт: 5 лет");

}

$resume = getCV("Описание вакансии");

echo $resume->name;

?>

Исправления:

1. Добавлена обработка исключений метода searchCV();

2. Изменена проверка на null, лучше использовать empty();

3. Указаны явные типы аргументов и возвращаемых значений;

4. Создан класс Resume для улучшенной типизации и структуры кода.

Анализ кода

import requests

def prepare\_questions(vacancy\_description):

    response = requests.post('https://api.openai.com/v1/chat/completions', json={

        'model': 'gpt-4',

        'messages': [{'role': 'system', 'content': 'Prepare a list of questions for the client based on the vacancy description.'},

                     {'role': 'user', 'content': vacancy\_description}]

    }, headers={'Authorization': 'Bearer YOUR\_API\_KEY'})

    return response.json()['choices'][0]['message']['content']

Python код:

1. Формирует POST запрос при помощи библиотеки request.
2. Запрос содержит prompt с указанием чату GPT создать список вопросов для кандидата на соответствующую вакансию.
3. Прикрепляется описание вакансии
4. В заголовках вопроса указывается api token приложения зарегистрированного в openai.
5. Из вариантов ответов выбирается первый.

Советы по развертыванию моего приложения(Linux):

git clone <https://github.com/MykytaFedorin/bitrix_widget>

cd bitrix\_widget

1. Фронтенд часть

cd frontend

npm install

Фронт сделан на React.

В папке /frontend необходимо создать файл .env с указанием

REACT\_APP\_PYTHON\_API\_URL=http://localhost:8000

REACT\_APP\_PHP\_API\_URL=http://localhost:8080

адреса хоста где лежат php и python скрипты. Предполагается запуск на одной машине, поэтому aдреса указаны как localhost. Это не очень чувствительные данные, но впоследствии все дополнительные api ключи следует располагать там.

Соответственно на сервер, где будет запускаться

Приложение необходимо установить node.js и npm.

С помощью npm run build необходимо собрать приложение. С помощью npm install -s serve устанавливается простенький prod сервер. Запускается он с помощью serve -s build.

2. Бекенд часть

**Python**

cd bitrix\_widget/backend/python

python3 -m venv env

. ./env/bin/activate

pip install -r requirements.txt

В папке bitrix\_widget/backend/python тоже необходимо разместить файл .env

И указать там OPENAI\_API\_KEY=yourapikey для корректной работы с ChatGPT.

Запустить FastAPI сервер поможет комманда

fastapi dev app.py

**PHP**

cd bitrix\_widget/backend/python

composer install

Точно так же в /php/src нужно создать .env файл который будет содержать

OPENAI\_API\_KEY=yourapikey

BITRIX\_BASE\_URL=url

BITRIX\_BASE\_URL это url входящего вебхука созданного на платформе битрикс который имеет вид

<https://b24-yourdomen.bitrix24.ru/rest/1/yourwebhookkey/>

Запустить сервер поможет команда

php -S localhost:8080 -t public

Таким образом если запускать весь проект на одном сервере, то запустить необходимо три программы сервера: react, python и php.