

## Генерация изображений через запрос с использованием DALL-E

*Стрельцова Марина Николаевна*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

### Аннотация

Искусственный интеллект и нейронные сети вплотную подошли к повседневной жизни множества людей. Уже мало кого можно удивить голосовым помощником, который, основываясь на твоих музыкальных композициях, может помочь подобрать песню «по вкусу» или настроению. Умные алгоритмы также присутствуют и в социальных сетях: помогают находить новых друзей, предлагают новостные записи или выводят видеоролики, опираясь на лайки пользователя. Искусственный интеллект стремительными темпами интегрируется в жизнь людей, и уже даже на сегодняшний день тяжело представить обыденные вещи без него. В данной статье рассмотрена нейронная сеть DALL-E от компании OpenAI для генерации изображений по запросу пользователя посредством API.

**Ключевые слова:** DALL-E, API, OpenAI, php, Laravel, машинное обучение.

### Image generation via request using the DALL-E

*Streltsova Marina Nikolaevna*

*Sholom-Aleichem Priamursky State University*

*Student*

### Abstract

Artificial intelligence and neural networks have come close to the daily lives of many people. Few people can be surprised by a voice assistant that, based on your musical compositions, can help you choose a song “to your taste” or mood. Smart algorithms are also present in social networks: they help you find new friends, offer news posts or display videos based on user likes. Artificial intelligence is rapidly integrating into people's lives, and even today, it is hard to imagine everyday things without it. This article discusses the DALL-E neural network from OpenAI for generating images at the request of the user through the API.

**Keywords:** DALL-E, API, OpenAI, php, Laravel, machine learning.

## 1. Введение

### 1.1 Актуальность исследования

Искусственный интеллект и нейронные сети вплотную подошли к повседневной жизни множества людей. Уже мало кого можно удивить голосовым помощником, который, основываясь на твоих музыкальных композициях, может помочь подобрать песню «по вкусу» или настроению.

Умные алгоритмы также присутствуют и в социальных сетях: помогают находить новых друзей, предлагают новостные записи или выводят видеоролики, опираясь на лайки пользователя. Искусственный интеллект стремительными темпами интегрируется в жизнь людей, и уже даже на сегодняшний день тяжело представить обыденные вещи без него. В данной статье рассмотрена нейронная сеть DALL-E от компании OpenAI для генерации изображений по запросу пользователя посредством API. Реализованы функции повторной генерации и выбора размера изображения, а также прописана валидация полей для ввода.

## 1.2 Обзор исследований

В исследовании Д. А. Алферьева проведен анализ инструментов моделирования интеллектуальных систем в основе которых лежат искусственные нейронные сети, машинное обучение и компьютерное зрение [1]. Опыт разработки и реализацию нейронной сети для распознавания лиц с использованием линейного классификатора SVM описывают в научной работе Н. А. Попова, М. А. Назаров и М. В. Власов [2]. В своей статье Н. Л. Боброва и И. С. Чайкин рассматривают наиболее используемые алгоритмы распознавания изображений, в том числе и алгоритмы получения признаков распознавания [3]. Новую сеть генерации голограмм HGN (нейронная сеть с прямой связью, синтезированная из различных функциональных блоков) для быстрой генерации CGH-шаблонов 3D-объектов предлагают в научном исследовании С. Ма и соавторы [4]. В статье Е. И. Маркина, А. И. Мартышкина, В.В. Зупарова описана архитектура генеративной состязательной сети (GAN), которая позволяет генерировать изображения, музыку, речь или текст, максимально схожие с тем, что создаёт человек [5].

## 1.3 Цель исследования

Целью данного исследования является написание сервиса для генерации изображений через запрос с использованием нейронной сети DALL-E [6].

## 2. Методы исследования

Для реализации сервиса были выбраны следующие программные средства, языки и технологии:

- PHP - скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов [7].
- PHPStorm - интегрированная среда разработки на PHP с интеллектуальным редактором, которая глубоко понимает код, поддерживает PHP 5.3-8.2 для современных и классических

проектов, обеспечивает лучшее в индустрии автодополнение кода, рефакторинг, предотвращение ошибок налету и поддерживает смешивание языков [8].

- API - описание способов (набор классов, процедур, функций структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой. Часто реализуется отдельной программной библиотекой или сервисом операционной системы. Используется программистами при написании всевозможных приложений [9].

- Laravel - бесплатный веб-фреймворк с открытым кодом, предназначенный для разработки с использованием архитектурной модели MVC на языке программирования PHP [10].

С помощью вышеописанных инструментов будет реализован сервис, который, взаимодействуя с нейронной сетью DALL-E посредством API компании OpenAI, будет генерировать изображения по запросу пользователя в указанном размере.

Перед началом работы также необходимо создать проект на фреймворке Laravel.

### 3. Результаты исследования

Для начала зайдём в созданный заранее проект Laravel через PHPStorm и подключим библиотеку openai-php/laravel для работы с API DALL-E (Рис. 1).

```
^C
D:\projects\OpenAIImage\OpenAIImage>composer require openai-php/laravel
```

Рисунок 1 – Подключение библиотеки

Далее следует опубликовать файл конфигурации и прописать персональный OPENAI\_API\_KEY в файле .env, который можно получить на официальном сайте компании OpenAI [11] после регистрации в настройках профиля (Рис. 2-4).

```
D:\projects\OpenAIImage\OpenAIImage>
D:\projects\OpenAIImage\OpenAIImage>php artisan vendor:publish --provider="OpenAI\Laravel\ServiceProvider"
```

Рисунок 2 – Публикация файла конфигурации


```
58 VITE_PUSHER_APP_CLUSTER= ${PUSHER_APP_CLUSTER}
59
60 OPENAI_API_KEY="sk-1[REDACTED]a7fmPMk"
61
```

Рисунок 3 – персональный OPENAI\_API\_KEY

## API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically rotate any API key that we've found has leaked publicly.

SECRET KEY	CREATED	LAST USED	
sk-...mPMk	10 янв. 2023 г.	Never	

[+ Create new secret key](#)

## Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

Personal 

Note: You can also specify which organization to use for each API request. See [Authentication](#) to learn more.

## Рисунок 4 – API Keys OpenAI

Следующим шагом создадим домашнюю страницу, где будет вводиться запрос и выбираться размер будущего изображения (Рис. 5).

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
5 integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">
6 <meta charset="UTF-8">
7 <meta name="viewport"
8 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
9 <meta http-equiv="X-UA-Compatible" content="ie=edge">
10 <title>Document</title>
11 </head>
12 <body class="bg-success">
13 <div class="container mt-5">
14 <div class="row justify-content-center">
15 <div class="col col-md-6">
16 <div class="card">
17 <div class="card-body">
18 <div class="card-title text-center border-bottom">
19 <h1>AI-генератор изображений</h1>
20 </div>
21 <form method="POST" action="{{route('images')}}" novalidate>
22 @csrf
23 <div class="mb-3">
24 <label for="description" class="form-label">Введите описание изображения</label>
25 <input type="text" class="form-control @error('description') is-invalid @enderror" value="{{old('description')}}"
26 name="description" id="description" placeholder="Пример: Природа Россия Зима" autofocus required maxlength="100">
27 @error('description')
28 <div class="invalid-feedback">
29 {{ $message }}
30 @enderror
31 </div>
32 </div>
33 <div class="mb-3">
34 <label for="size" class="form-label">Выберите размер изображения</label>
35 <select class="form-select @error('size') is-invalid @enderror" id="size" name="size">
36 <option value="sm">Маленькое</option>
37 <option value="md">Среднее</option>
38 <option value="lg">Большое</option>
39 </select>
40 @error('size')
41 <div class="invalid-feedback">
42 {{ $message }}
43 @enderror
44 </div>>
45 </div>
46 <button class="btn btn-primary mt-5 mb-3 w-100">Сгенерировать</button>
47 </form>
48 </div>
49 </div>
50 </div>
51 </div>
52 </div>
53 </div>
54 </body>
55 </html>
```

Рисунок 5 – Код домашней страницы

Подключаем Bootstrap для определения стилей страницы и размещаем необходимые для запроса поля в div-блоки. Также создаем форму для отправки значений контроллеру, который будет взаимодействовать непосредственно с API DALL-E. Стоит учесть, что DALL-E понимает 3 размера изображений: 256x256, 512x512 и 1024x1024 что в создаваемом сервисе соответствует значениям маленькое, среднее и большое. Внешний вид страницы представлен на скриншоте 6.

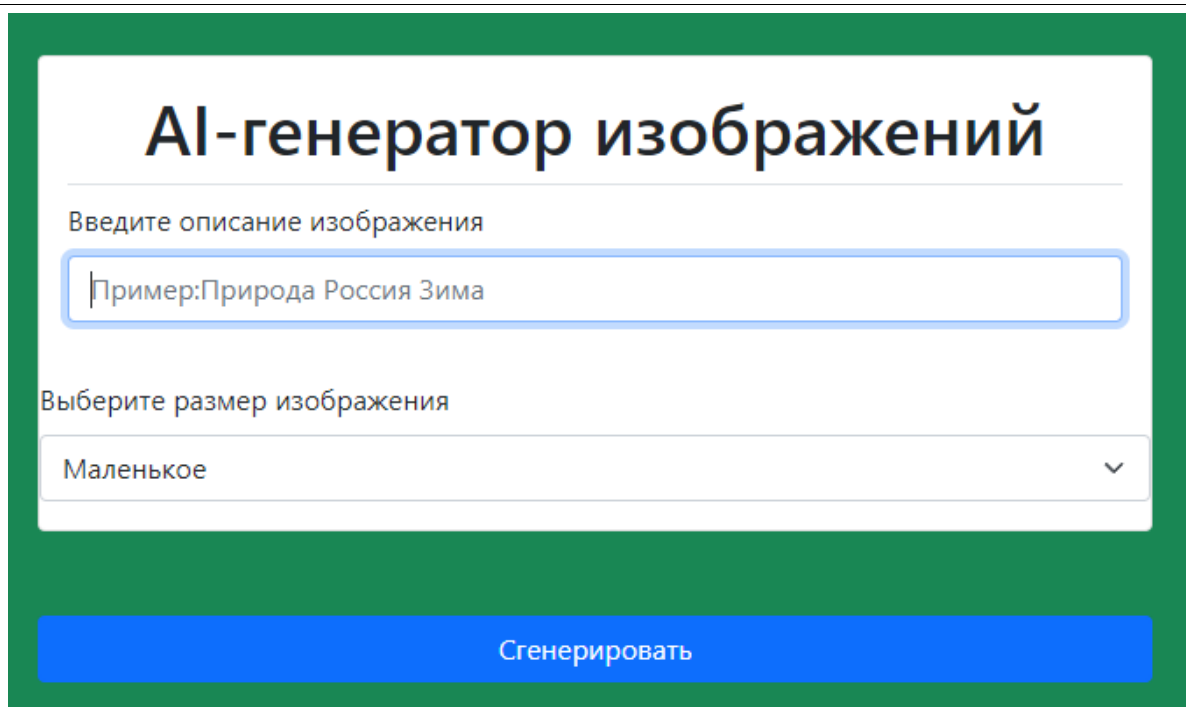
The image shows a web interface for an AI image generator. At the top, the title "AI-генератор изображений" is displayed in a large, bold, black font. Below the title, there is a text input field with the placeholder text "Введите описание изображения". The input field contains the example text "Пример: Природа Россия Зима". Below the input field, there is a dropdown menu with the label "Выберите размер изображения". The dropdown menu is currently set to "Маленькое". At the bottom of the form, there is a large blue button with the text "Сгенерировать". The entire form is enclosed in a green border.

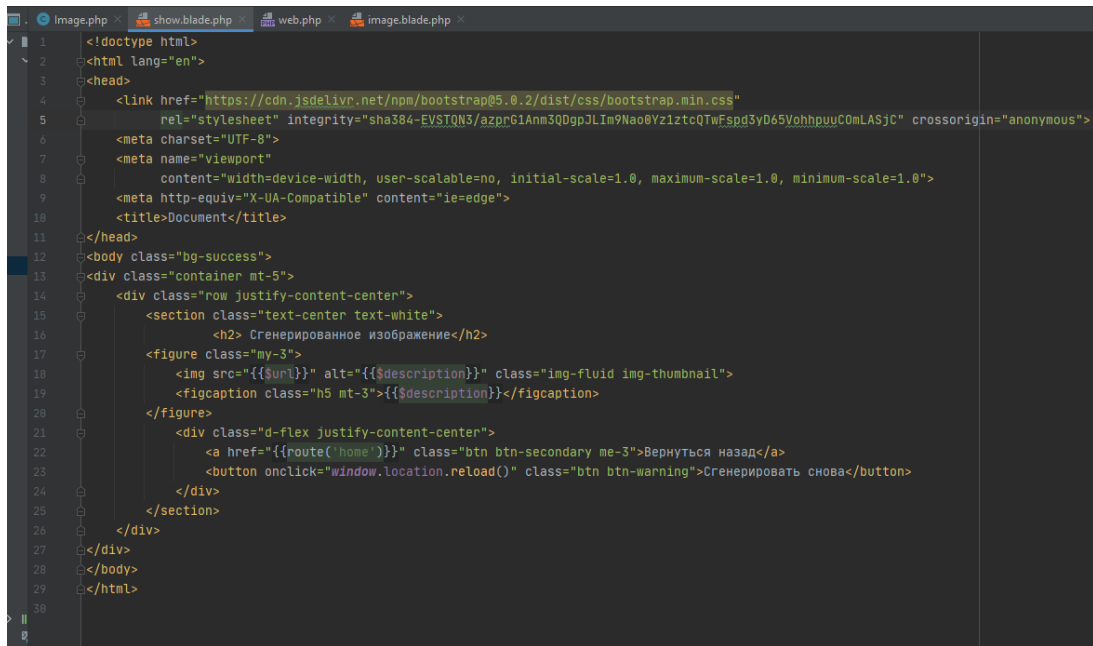
Рисунок 6 – Внешний вид домашней страницы

Теперь необходимо создать контроллер, отвечающий за создание запроса для передачи его DALL-E. С прошлой страница вытаскиваем значения из переменной `$request`. Пишем валидацию данных значений для того, чтобы избежать ошибок при работе кода. Далее формируем запрос и передаем его с необходимыми параметрами (персональный ключ, количество изображений, их размер и описание, а также формат ответа) нейронной сети DALL-E. Полученный ответ преобразуем в массив и отправляем результат на страницу, которая будет отвечать за отображение сгенерированного изображения (Рис. 7).

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Validation\Rule;
7 use OpenAI;
8
9 class Image extends Controller
10 {
11     public function ImageCreate(Request $request){
12         $request->validate([
13             'description' => 'required|string|max:100',
14             'size' =>Rule::in(['sm', 'md', 'lg'])
15         ]);
16         $description=$request->description;
17         switch ($request->size){
18             case 'lg':
19                 $size = '1024x1024';
20                 break;
21             case 'md':
22                 $size = '512x512';
23                 break;
24             default:
25                 $size = '256x256';
26         }
27         $client = OpenAI::client(env( key: 'OPENAI_API_KEY'));
28         $response = $client->images()->create([
29             'prompt' => $description,
30             'n' => 1,
31             'size' => $size,
32             'response_format' => 'url',
33         ]);
34         $url=$response->toArray()['data'][0]['url'];
35         return view ( view: 'show', compact( varname: 'url', _: 'description'));
36     }
37 }
38
```

Рисунок 7 – Код контроллера

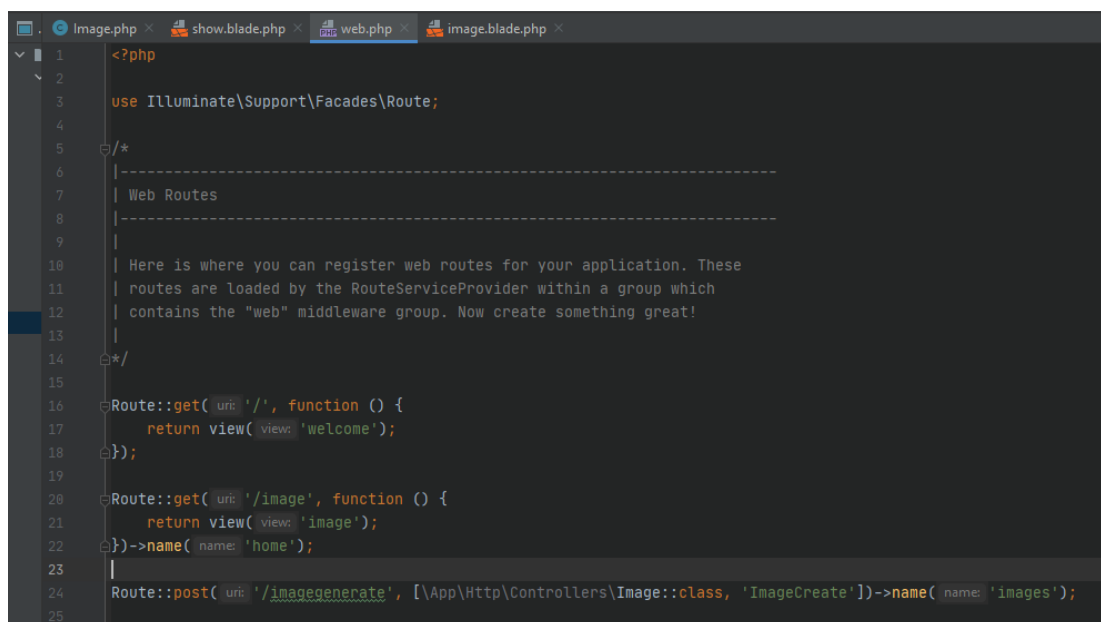
Далее создаем div-блоки с нужными классами и элемент `<img>`, которому передаем ссылку на сгенерированное изображение, полученную из контроллера. Также добавим две кнопки: «Сгенерировать снова» и «Вернуться назад». Они будут отвечать за повторную генерацию изображения по тому же запросу и возвращение на домашнюю страницу соответственно (Рис. 8).



```
1 <!doctype html>
2 <html lang="en">
3 <head>
4 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
5     rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1tcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">
6 <meta charset="UTF-8">
7 <meta name="viewport"
8     content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
9 <meta http-equiv="X-UA-Compatible" content="ie=edge">
10 <title>Document</title>
11 </head>
12 <body class="bg-success">
13 <div class="container mt-5">
14 <div class="row justify-content-center">
15 <section class="text-center text-white">
16 <h2> Сгенерированное изображение</h2>
17 <figure class="my-3">
18 
19 <figcaption class="h5 mt-3">{{ $description }}</figcaption>
20 </figure>
21 <div class="d-flex justify-content-center">
22 <a href="{{ route('home') }}" class="btn btn-secondary me-3">Вернуться назад</a>
23 <button onclick="window.location.reload()" class="btn btn-warning">Сгенерировать снова</button>
24 </div>
25 </section>
26 </div>
27 </div>
28 </body>
29 </html>
```

Рисунок 8 – Код страницы для отображения сгенерированного изображения

Последним шагом пропишем нужные роуты для корректных переходов между страницами и запустим готовый сервис командой «php artisan serve» через встроенный терминал PHPStorm (Рис. 9).



```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4
5 /*
6 |-----|
7 | Web Routes
8 |-----|
9 |
10 | Here is where you can register web routes for your application. These
11 | routes are loaded by the RouteServiceProvider within a group which
12 | contains the "web" middleware group. Now create something great!
13 |
14 |*/
15
16 Route::get( uri: '/', function () {
17     return view( view: 'welcome' );
18 });
19
20 Route::get( uri: '/image', function () {
21     return view( view: 'image' );
22 })->name( name: 'home' );
23
24 Route::post( uri: '/imagegenerate', [ \App\Http\Controllers\Image::class, 'ImageCreate' ]->name( name: 'images' );
25
```

Рисунок 9 – Роуты сервиса

Проверим работу сервиса, отправив пару отличающихся запросов с разными размерами (Рис. 10-11).



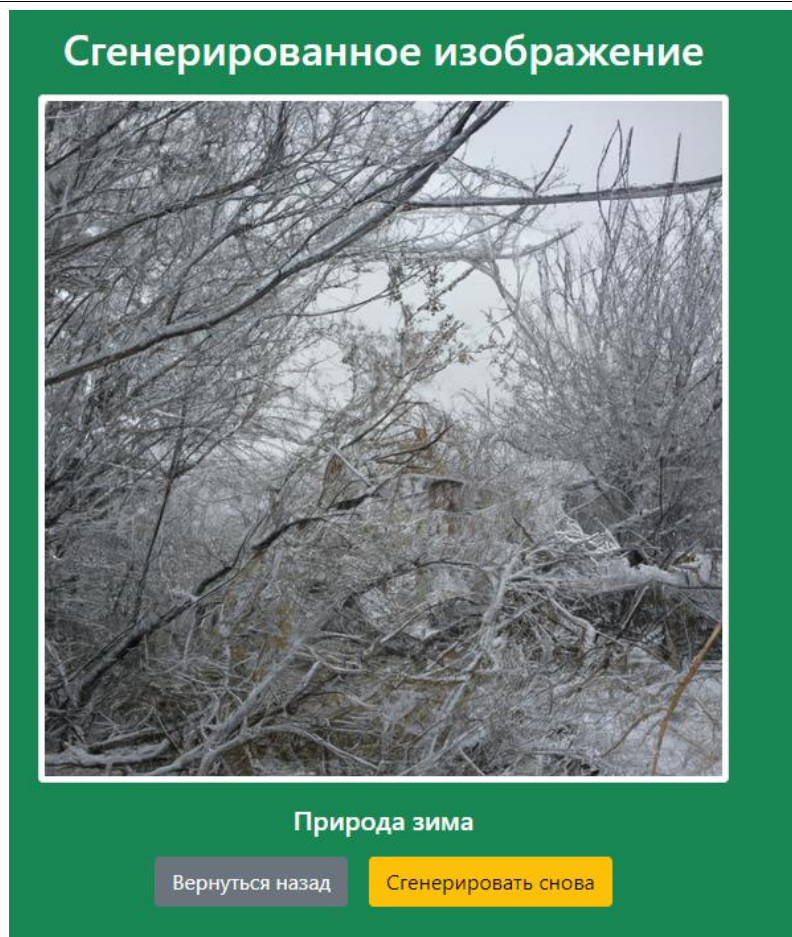


Рисунок 10 – Пример сгенерированного изображения по запросу «Природа зима» маленького размера



Рисунок 11 – Пример сгенерированного изображения по запросу «Медведь белый» большого размера

Как видно из скриншотов 10-11 реализованный сервис генерирует изображения по запросу пользователя нужного размера.

#### **4. Выводы**

Нейронные сети на сегодняшний момент оказывают существенное влияние на все сферы жизнедеятельности людей и в будущем их влияние будет только увеличиваться. Существуют как сторонники, так и противники всеобъемлющего внедрения искусственного интеллекта в жизнь людей, но пока одни спорят, другие живут и пользуются преимуществами данной технологии. В рамках данной статьи был написан и протестирован сервис для генерации изображений через запрос с использованием нейронной сети DALL-E. Также были реализованы функции повторной генерации и выбора размера изображения, а также прописана валидация полей для ввода.

#### **Библиографический список**

1. Алферьев Д. А. Программные средства моделирования и разработки систем искусственного интеллекта //Крымский научный вестник. 2020. №. 1 (26). С. 14-18.
2. Попова Н. А., Назаров М. А., Власов М. В. Решение задачи распознавания лиц с использованием алгоритмов машинного обучения //Моделирование, оптимизация и информационные технологии. 2018. Т. 6. №. 1. С. 408-415.
3. Боброва Н. Л., Чайкин И. С. Обзор алгоритмов распознавания изображений. 2022.
4. Ma C. et al. A novel feed-forward neural network-based method for fast hologram generation //Optics Communications. 2023. Т. 530. С. 129162.
5. Маркин Е. И., Мартышкин А. И., Зупарова В. В. Анализ возможностей нейронных сетей для генерации фотореалистичных изображений // Современные информационные технологии. 2021. №. 33 (33). С. 30.
6. DALL-E URL: <https://openai.com/dall-e-2/> (дата обращения: 10.01.2023).
7. PHP URL: <https://www.php.net/> (дата обращения: 10.01.2023).
8. PHPStorm URL: <https://www.jetbrains.com/ru-ru/phpstorm/> (дата обращения: 10.01.2023).
9. API URL: <https://ru.wikipedia.org/wiki/API> (дата обращения: 10.01.2023).
- 10.Laravel URL: <https://laravel.com/> (дата обращения: 10.01.2023).
- 11.OpenAI URL: <https://beta.openai.com/> (дата обращения: 10.01.2023).