**МИНОБРНАУКИ РОССИИ**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Расчетно-графическое задание  
дисциплина: «Технологии web-программирования»

Выполнил: ст. группы ВТ-41  
Дешина Оксана Юрьевна

Белгород 2021 г.

Оглавление

[1. HTML. Разработка макетов и верстка шаблонов web-приложения с помощью языков HTML и CSS 3](#_Toc91338828)

[2. Клиентское программирование. Vue js. 9](#_Toc91338829)

[3. Серверное программирование. Laravel 15](#_Toc91338830)

[4. Разработка и проектирование базы данных 19](#_Toc91338831)

[5. REST API 20](#_Toc91338832)

[6. Работа с HTTP запросами 22](#_Toc91338833)

# 1. HTML. Разработка макетов и верстка шаблонов web-приложения с помощью языков HTML и CSS

В процессе выполнения были изучены основы работы протокола передачи гипертекста HTTP, основы разметки страниц с помощью языка HTML, основы вёрстки страниц, адаптированных под различные устройства и основы работы задания стилей для разметки страницы с помощью языка CSS. Разработаны макеты и «сверстаны» страницы для web-приложения.

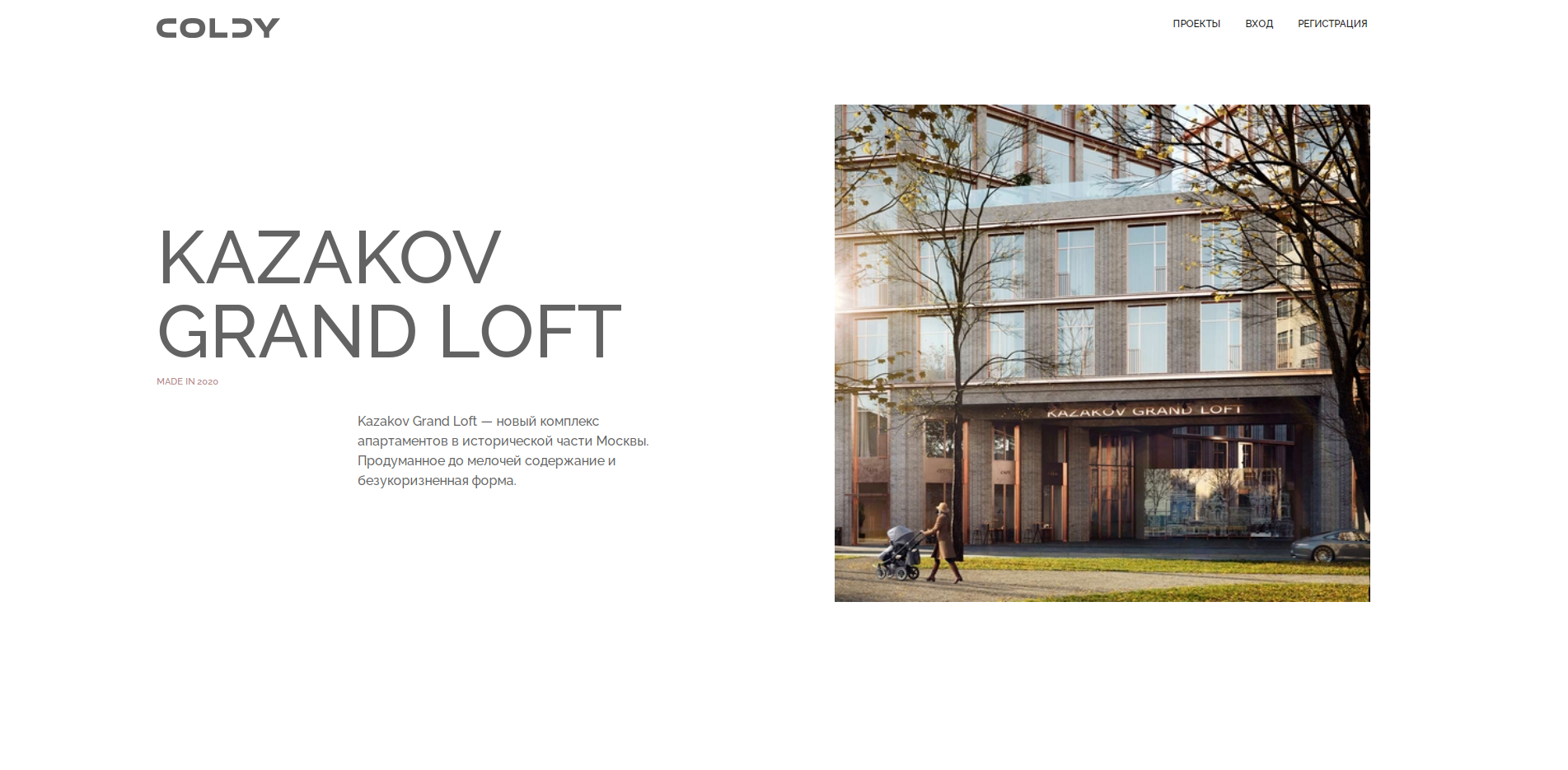
*Содержание index.html*

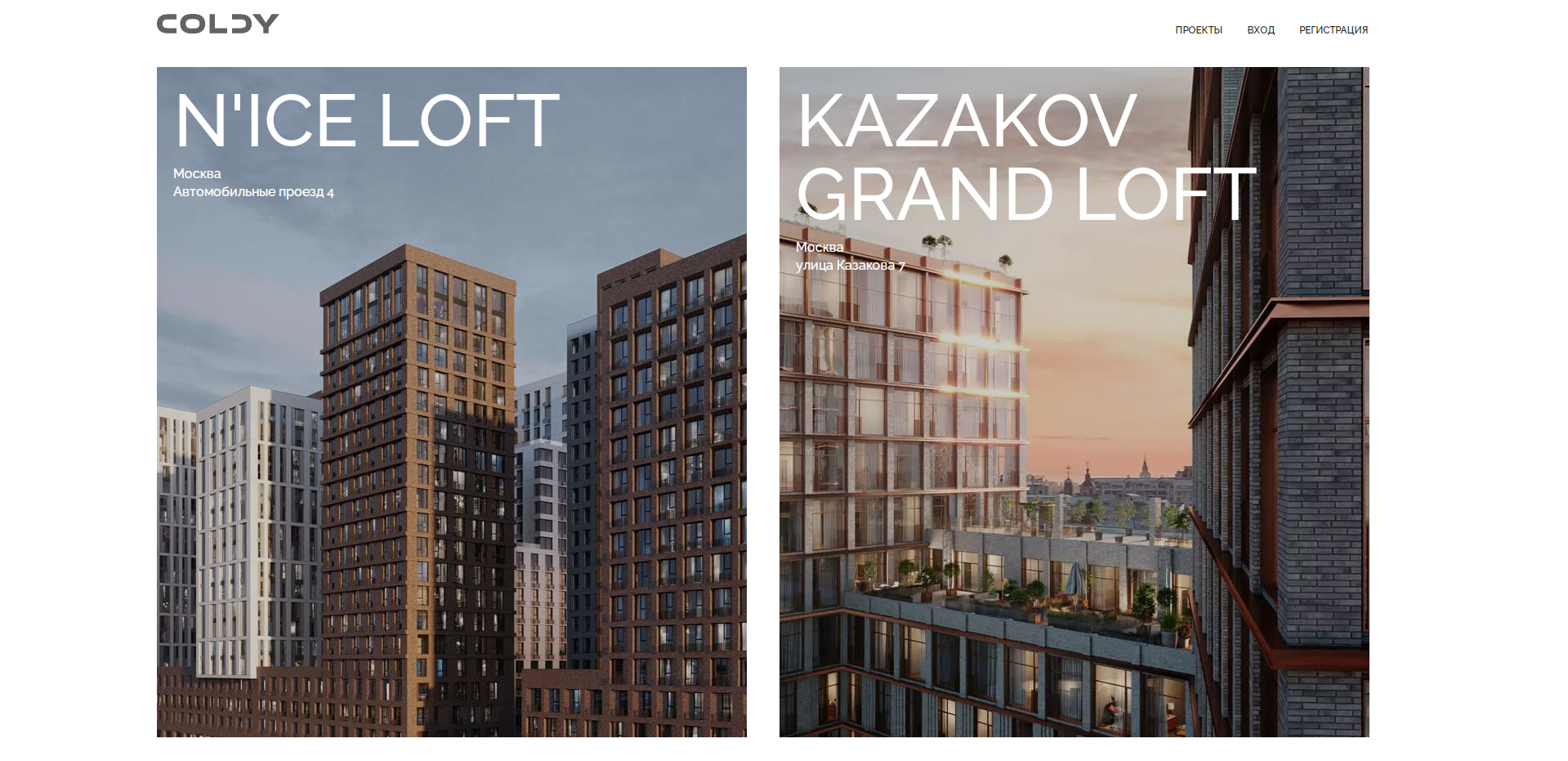
<!doctype html>  
<html class="no-js" lang="">  
  
<head>  
 <meta charset="utf-8">  
 <title>COLDY | Главная</title>  
 <meta name="description" content="">  
 <meta name="viewport" content="width=device-width, initial-scale=1">  
  
 <meta property="og:title" content="">  
 <meta property="og:type" content="">  
 <meta property="og:url" content="">  
 <meta property="og:image" content="">  
  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"  
 integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">  
 <link rel="manifest" href="site.webmanifest">  
 <link rel="apple-touch-icon" href="icon.png">  
 <!-- Place favicon.ico in the root directory -->  
  
 <link rel="stylesheet" href="css/normalize.css">  
 <link rel="stylesheet" href="css/main.css">  
  
 <meta name="theme-color" content="#fafafa">  
</head>  
  
<body>  
  
<!-- Add your site or application content here -->  
  
<nav>  
 <div class="div\_nav\_logo">  
 <a href="index.html"><img src="/../img/logo.svg"></a>  
 </div>  
 <div class="div\_nav\_menu">  
 <a href="projects.html"><p>ПРОЕКТЫ</p></a>  
 <a href="sign\_in.html"><p>ВХОД</p></a>  
 <a href="registration.html"><p>РЕГИСТРАЦИЯ</p></a>  
 </div>  
</nav>  
  
<main>  
  
 <div id="img\_hero" class="div\_hero\_img">  
 <img src="/../img/hero\_img.png">  
 </div>  
  
 <div class="div\_hero">  
 <h3>KAZAKOV GRAND LOFT</h3>  
 <h6>MADE IN 2020</h6>  
 <p>Kazakov Grand Loft — новый комплекс апартаментов в исторической части Москвы.  
 Продуманное до мелочей содержание и безукоризненная форма.</p>  
 </div>  
  
 <div class="div\_make">  
 <h4>WE MAKE NEW LIVING</h4>  
 <p>Мы создаем проекты, которые меняют повседневную жизнь, собирают вместе единомышленников  
 и новое поколение жителей мегаполиса.</p>  
 <p>Проекты COLDY предвосхищают ваши интересы и желания.</p>  
 </div>  
  
 <div id="img\_coldy">  
 <img src="/../img/block\_coldy.jpg">  
 </div>  
 <div class="div\_coldy">  
 <h4>COLDY</h4>  
 <p>Наша команда — сообщество людей, устремленных в будущее, которые бережно относятся  
 к архитектурным традициям и ценят индивидуальность: каждый сотрудник и проект компании уникальны.</p>  
 <p>Coldy — наш ежедневный драйв, который заряжает на новые свершения.</p>  
 <p>17 лет мы задаём новые стандарты качества, ежедневно открывая новые возможности и сохраняя  
 память города.</p>  
  
 </div>  
  
  
 <div class="div\_command">  
 <h4>КОМАНДА</h4>  
 <p>Хочешь сделать что-то идеальным — начни с себя. Так считаем мы, принимая в команду амбициозных людей,  
 для которых работа — это драйв и творчество, выход за рамки стереотипов.</p>  
 <p>Мы радуемся новым вызовам так же сильно, как и новым победам.</p>  
 <p>Мы гордимся своим уровнем, но всегда готовы расти дальше. И для этого у нас есть все условия.</p>  
 </div>  
 <div class="img\_command">  
 <img src="/../img/comand1.png">  
 <img src="/../img/comand2.png">  
 </div>  
  
 <div class="progr\_loyalty">  
 <h4>ПРОГРАММА ЛОЯЛЬНОСТИ</h4>  
 <p>Мы создаем проекты, которые меняют повседневную жизнь, собирают вместе единомышленников и новое  
 поколение жителей мегаполиса. Проекты MADE IN COLDY предвосхищают ваши интересы и желания.</p>  
 </div>  
 <div id="img\_loyalty">  
 <img src="/../img/loyalty.png">  
 </div>  
  
</main>  
  
  
<footer>  
  
 <div class="foot\_logo">  
 <img src="/../img/logo.svg">  
 </div>  
  
 <div class="foot\_menu">  
 <div class="foot\_inf">  
 <p>Центральный офис</p>  
 <h5>+7 (495) 644-42-44</h5>  
 </div>  
  
 <div class="foot\_inf">  
 <p>Общие вопросы</p>  
 <h5>info@coldy.ru</h5>  
 </div>  
  
 <div class="foot\_inf">  
 <p>Реклама и PR</p>  
 <h5>pr@coldy.ru</h5>  
 </div>  
 </div>  
  
</footer>  
  
  
<script src="js/vendor/modernizr-3.11.2.min.js"></script>  
<script src="js/plugins.js"></script>  
<script src="js/main.js"></script>  
  
<!-- Google Analytics: change UA-XXXXX-Y to be your site's ID. -->  
<script>  
 ***window***.ga = function () {  
 ga.q.push(arguments)  
 };  
 ga.q = [];  
 ga.l = +new ***Date***;  
 ga('create', 'UA-XXXXX-Y', 'auto');  
 ga('set', 'anonymizeIp', true);  
 ga('set', 'transport', 'beacon');  
 ga('send', 'pageview')  
</script>  
<script src="https://www.google-analytics.com/analytics.js" async></script>  
</body>  
  
</html>

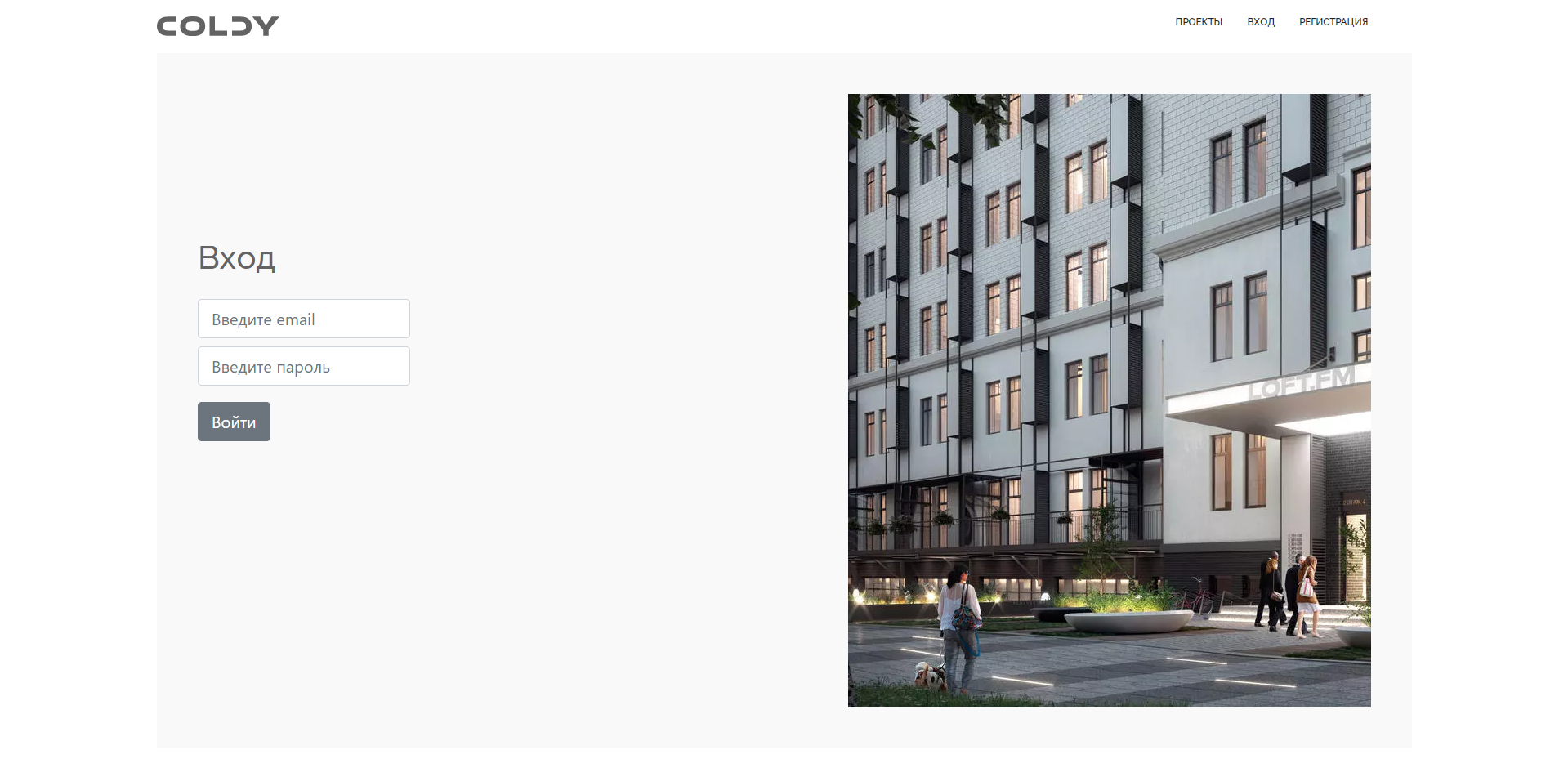
*Неполное содержание main.css*

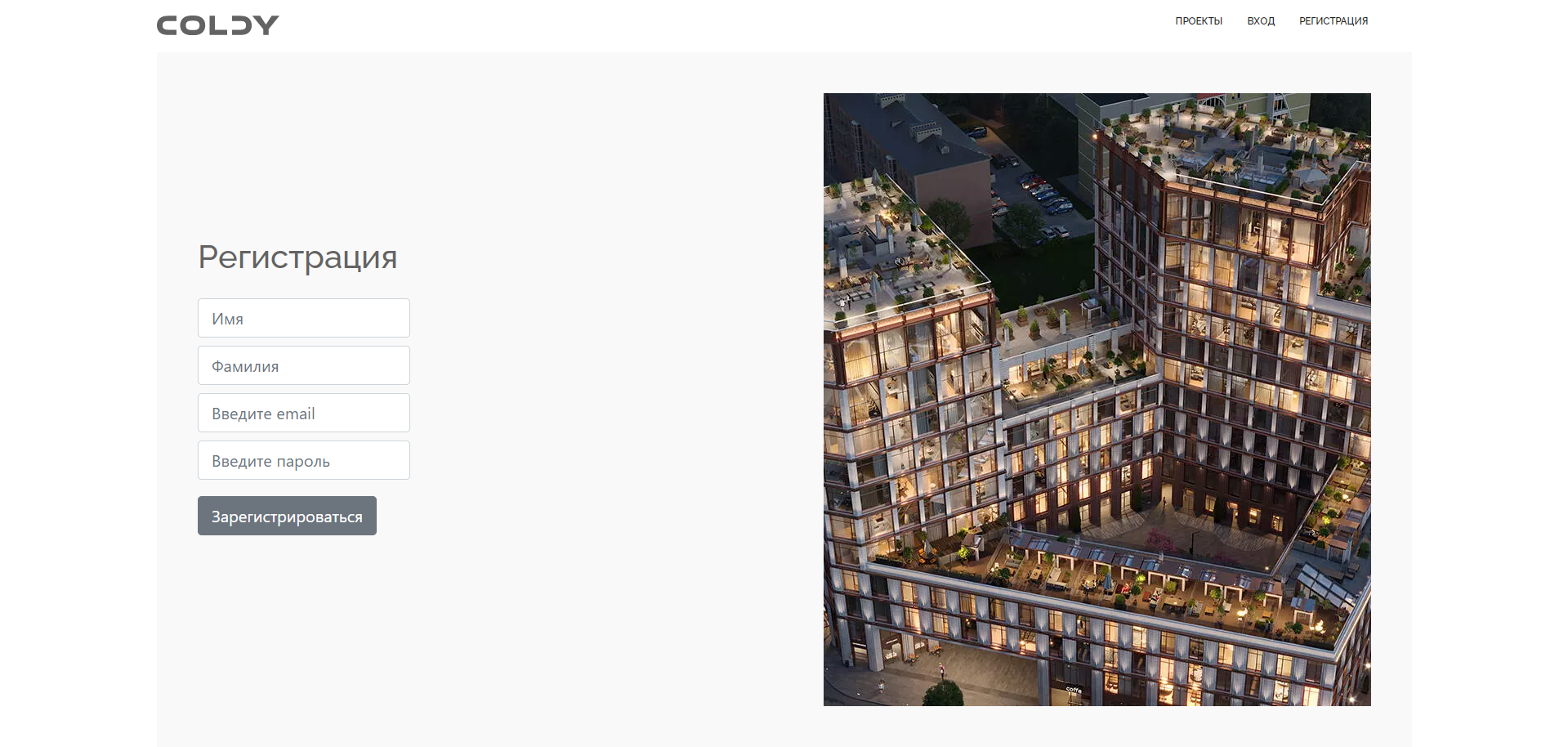
ul {  
 list-style-type: none;  
 margin: 0;  
 padding: 0;  
}  
  
li {  
 list-style-type: none;  
 margin: 0;  
 padding: 0;  
}  
  
a {  
 text-decoration: none;  
}  
  
nav {  
 max-width: 80%;  
 margin-left: auto;  
 margin-right: auto;  
 padding-bottom: 20px;  
 padding-top: 20px;  
 display: flex;  
}  
  
nav ul img{  
 width: 150px;  
}  
  
.div\_nav\_menu {  
 display: flex;  
 justify-content: space-between;  
 position: absolute;  
 right: 12%;  
}  
  
.div\_nav\_menu p {  
 font-family: 'Raleway', sans-serif;  
 margin-right: 15px;  
 margin-left: 15px;  
 font-size: 12px;  
 color: #222222;  
 font-weight: 600;  
}  
  
.div\_nav\_menu p:hover {  
 color: #b68178;  
}  
  
.div\_nav\_menu p:active {  
 color: #b68178;  
}  
  
main {  
 max-width: 80%;  
 margin-left: auto;  
 margin-right: auto;  
}  
  
.div\_hero {  
 max-width: 40%;  
 margin-bottom: 350px;  
 margin-top: 200px;  
}  
  
.div\_hero h3 {  
 font-family: 'Raleway', sans-serif;  
 font-size: 90px;  
 font-weight: 500;  
 line-height: 1;  
 color: #636363;  
 margin: 0;  
 padding: 0;  
}  
  
.div\_hero h6 {  
 font-family: 'Raleway', sans-serif;  
 font-weight: 500;  
 color: #b68178;  
 font-size: 11px;  
 margin-top: 10px;  
 margin-bottom: 30px;  
 padding: 0;  
}  
  
.div\_hero p {  
 font-family: 'Raleway', sans-serif;  
 font-weight: 500;  
 color: #636363;  
 font-size: 16px;  
 max-width: 60%;  
 margin-left: auto;  
}  
  
#img\_hero {  
 float: right;  
 position: relative;  
 top: -140px;  
 left: -50px;  
}  
  
.div\_hero\_img img {  
 width: 650px;  
}

**Результат работы**









# 2. Клиентское программирование. Vue js.

В процессе выполнения познакомились с основами разработки на языке JavaScript, с основами разработки frontend приложения, в частности с использованием клиентских фреймворков (Vue js), научились переносить вёрстку в отдельные компоненты.

В процессе выполнения были созданы следующие образы:

1. Home (компонент Home\_page)
2. Projcets (компонент Project\_card)
3. Login (компонент Login\_page)
4. Registration (компонент Reg\_page)

В качестве демонстрации работы был выбран образ Projcets

*Содержание Projects.vue*

<template>  
 <section class="projects\_page">  
 <div v-for="item in project\_card" class="projects-item">  
 <Project\_card :project\_card="item"></Project\_card>  
 </div>  
 </section>  
</template>  
  
<script>  
import Project\_card from "../components/Project\_card";  
  
export default {  
 name: "Projects",  
 components: {Project\_card},  
 data() {  
 return {  
 project\_card: [  
 {  
 id: 1,  
 name: 'NICE LOFT',  
 city: 'Москва',  
 address: 'Автомобильные проезд 4'  
 },  
 {  
 id: 2,  
 name: 'KAZAKOV GRAND LOFT',  
 city: 'Москва',  
 address: 'улица Казакова 7'  
 }  
 ]  
 }  
 },  
}  
</script>  
  
<style scoped>  
.projects\_page {  
 display: flex;  
 justify-content: space-between;  
}  
  
</style>

*Содержание Project\_card.vue*

<template>  
 <a href="pr\_niceloft.html"><div class="projects">  
 <h3>N'ICE LOFT</h3>  
 <p>Москва</p>  
 <p>Автомобильные проезд 4</p>  
 </div></a>  
</template>  
  
<script>  
export default {  
 name: "Project\_card",  
 props: ['project\_card'],  
}  
</script>  
  
<style scoped>  
  
.projects {  
 height: 820px;  
 background: url(/frontend/src/assets/pr\_niceloft.png) rgba(0, 0, 0, 0.3);  
 margin-top: 20px;  
 padding: 0;  
}  
  
.projects h3 {  
 font-family: 'Raleway', sans-serif;  
 font-size: 90px;  
 font-weight: 500;  
 line-height: 1;  
 color: #ffffff;  
 margin: 0;  
 padding-left: 20px;  
 padding-top: 20px;  
 margin-bottom: 10px;  
}  
  
.projects p {  
 font-family: 'Raleway', sans-serif;  
 font-size: 16px;  
 font-weight: 500;  
 color: #ffffff;  
 padding-left: 20px;  
 margin: 0;  
}  
  
#pr\_kazakov {  
 height: 820px;  
 background: url(/frontend/src/assets/pr\_kazakov.png) rgba(13%, 13%, 13%, 0.3);  
 margin-top: 20px;  
 padding: 0;  
}  
  
</style>

*Содержание App.vue*

<template>  
 <div id="app">  
 <nav>  
 <div class="div\_nav\_logo">  
 <router-link :to="{name: 'Home'}"><img :src="logo"/></router-link>  
 </div>  
 <div class="div\_nav\_menu">  
 <router-link :to="{name: 'Projects'}"><p>ПРОЕКТЫ</p></router-link>  
 <router-link :to="{name: 'Login'}"><p>ВХОД</p></router-link>  
 <router-link :to="{name: 'Registration'}"><p>РЕГИСТРАЦИЯ</p></router-link>  
 </div>  
 </nav>  
  
 <main>  
 <router-view/>  
 </main>  
  
 <footer>  
 <div class="foot\_logo">  
 <img :src="logo"/>  
 </div>  
  
 <div class="foot\_menu">  
 <div class="foot\_inf">  
 <p>Центральный офис</p>  
 <h5>+7 (495) 644-42-44</h5>  
 </div>  
  
 <div class="foot\_inf">  
 <p>Общие вопросы</p>  
 <h5>info@coldy.ru</h5>  
 </div>  
  
 <div class="foot\_inf">  
 <p>Реклама и PR</p>  
 <h5>pr@coldy.ru</h5>  
 </div>  
 </div>  
 </footer>  
  
  
 </div>  
</template>  
<script>  
import logo from "./assets/logo.svg";  
  
export default {  
 computed: {  
 layout() {  
 return this.$route.meta.layout || "default-layout"  
 }  
 },  
 data: function () {  
 return {  
 logo: logo  
 }  
 }  
}  
</script>  
  
<style scoped>  
main {  
 max-width: 80%;  
 margin-left: auto;  
 margin-right: auto;  
}  
  
ul {  
 list-style-type: none;  
 margin: 0;  
 padding: 0;  
}  
  
li {  
 list-style-type: none;  
 margin: 0;  
 padding: 0;  
}  
  
a {  
 text-decoration: none;  
}  
  
  
nav {  
 max-width: 80%;  
 margin-left: auto;  
 margin-right: auto;  
 padding-bottom: 20px;  
 padding-top: 20px;  
 display: flex;  
}  
  
nav ul img{  
 width: 150px;  
}  
  
.div\_nav\_menu {  
 display: flex;  
 justify-content: space-between;  
 position: absolute;  
 right: 12%;  
}  
  
.div\_nav\_menu p {  
 font-family: 'Raleway', sans-serif;  
 margin-right: 15px;  
 margin-left: 15px;  
 font-size: 12px;  
 color: #222222;  
 font-weight: 600;  
}  
  
.div\_nav\_menu p:hover {  
 color: #b68178;  
}  
  
.div\_nav\_menu p:active {  
 color: #b68178;  
}  
  
  
footer {  
 max-width: 80%;  
 margin-left: auto;  
 margin-right: auto;  
 display: flex;  
 margin-top: 300px;  
 margin-bottom: 50px;  
 justify-content: space-between;  
}  
  
.foot\_inf {  
 margin-right: 30px;  
}  
  
.foot\_inf p {  
 font-family: 'Raleway', sans-serif;  
 font-weight: 500;  
 color: #636363;  
 font-size: 12px;  
 line-height: 13px;  
 margin-bottom: 20px;  
 padding: 0px;  
}  
  
.foot\_inf h5 {  
 font-family: 'Raleway', sans-serif;  
 font-weight: 500;  
 color: #222222;  
 font-size: 22px;  
 line-height: 21px;  
 padding: 0px;  
 margin: 0px;  
}  
  
.foot\_menu {  
 display: flex;  
}  
</style>

*Содержание router/index.js*

import Vue from 'vue'  
import VueRouter from 'vue-router'  
import Home from '../views/Home.vue'  
import Projects from '../views/Projects.vue'  
import Login from '../views/Login.vue'  
import Registration from '../views/Registration.vue'  
  
Vue.use(VueRouter)  
  
const routes = [  
 {  
 path: '/',  
 name: 'Home',  
 component: Home  
 },  
 {  
 path: '/projects',  
 name: 'Projects',  
 component: Projects  
 },  
 {  
 path: '/login',  
 name: 'Login',  
 component: Login  
 },  
 {  
 path: '/registration',  
 name: 'Registration',  
 component: Registration  
 }  
]  
  
const ***router*** = new VueRouter({  
 mode: 'history',  
 base: ***process***.env.BASE\_URL,  
 routes  
})  
  
export default ***router***

**Результат выполнения**



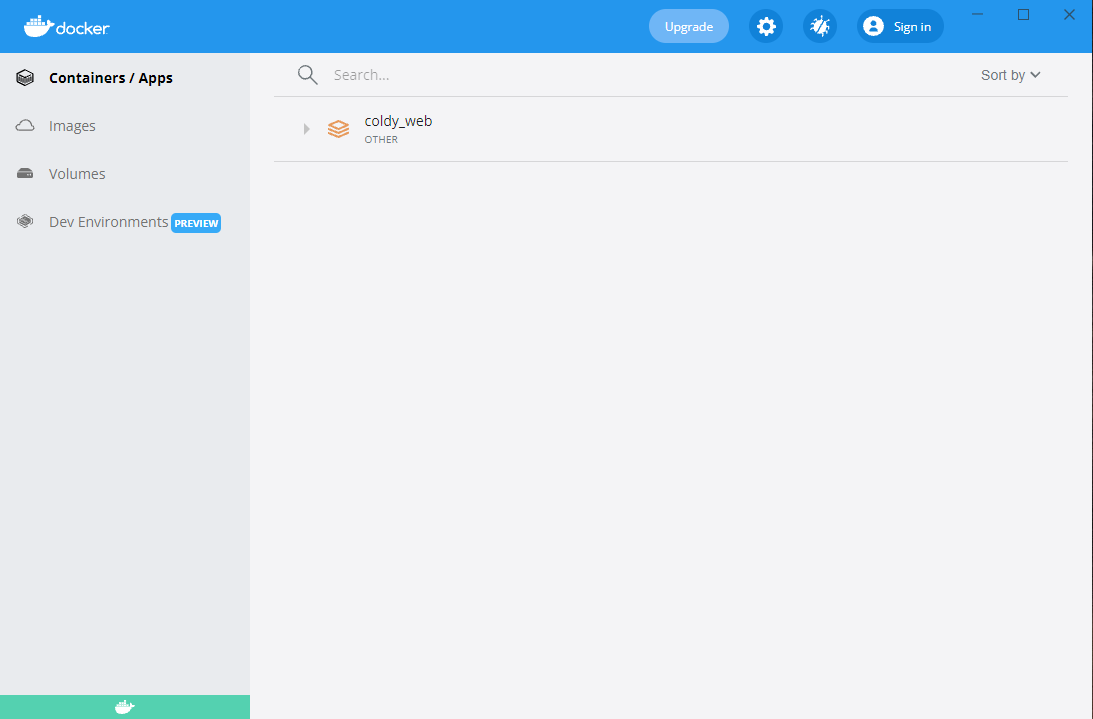
После подключения backend на фоне карточек появятся картинки.

# 3. Серверное программирование. Laravel

**Выполнение**

В ходе выполнения познакомились с основами backend разработки web-приложений. Научились писать скрипты. Познакомились с основами работы docker. Познакомились с фреймворком Laravel Lumen и научились разворачивать проект, производить его настройку. Научились работать с API в приложении Postman.

Установим сам docker с сайта docker.com



Затем добавим в проект следующие файлы из репозитория:

docker-compose.yml

version: '3'  
services:  
  
 app:  
 build:  
 context: ""  
 dockerfile: docker/Dockerfile  
 image: digitalocean.com/php  
 restart: unless-stopped  
 tty: true  
 environment:  
 SERVICE\_NAME: app  
 SERVICE\_TAGS: dev  
 working\_dir: /var/www  
 volumes:  
 - ./:/var/www  
 - ./php/local.ini:/usr/local/etc/php/conf.d/local.ini  
 networks:  
 - app-network  
  
 webserver:  
 image: nginx:alpine  
 restart: unless-stopped  
 tty: true  
 ports:  
 - "80:80"  
 - "443:443"  
 volumes:  
 - ./:/var/www  
 - ./nginx/conf.d/:/etc/nginx/conf.d/  
 networks:  
 - app-network  
  
 db:  
 image: mysql:5.7  
 restart: unless-stopped  
 tty: true  
 ports:  
 - "3306:3306"  
 environment:  
 MYSQL\_ROOT\_PASSWORD: 'Oksana123!'  
 MYSQL\_DATABASE: 'web\_coldy'  
 MYSQL\_USER: 'dioxid'  
 MYSQL\_PASSWORD: 'Oksana123'  
 networks:  
 - app-network  
  
networks:  
 app-network:  
 driver: bridge

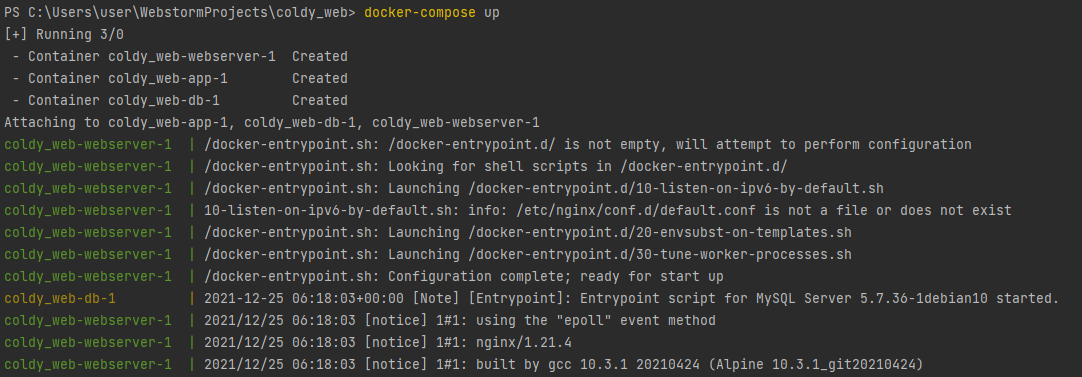
docker/Dockerfile

FROM php:7.4-fpm  
  
# Copy composer.lock and composer.json  
# COPY composer.json /var/www/  
  
# Set working directory  
WORKDIR /var/www  
  
# Install dependencies  
RUN apt-get update && apt-get install -y \  
 build-essential \  
 libpng-dev \  
 libjpeg62-turbo-dev \  
 libfreetype6-dev \  
 locales \  
 libzip-dev \  
 zip \  
 jpegoptim optipng pngquant gifsicle \  
 vim \  
 unzip \  
 git \  
 curl  
  
# Clear cache  
RUN apt-get clean && rm -rf /var/lib/apt/lists/\*  
  
# Install extensions  
RUN docker-php-ext-install pdo\_mysql zip exif pcntl  
RUN docker-php-ext-configure gd --with-freetype=/usr/include/ --with-jpeg=/usr/include/  
RUN docker-php-ext-install gd  
  
# Install composer  
RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer  
  
# Add user for laravel application  
RUN groupadd -g 1000 www  
RUN useradd -u 1000 -ms /bin/bash -g www www  
  
# Copy existing application directory contents  
COPY . /var/www  
  
# Copy existing application directory permissions  
COPY --chown=www:www . /var/www  
  
# Change current user to www  
USER www  
  
# Expose port 9000 and start php-fpm server  
EXPOSE 9000  
CMD ["php-fpm"]

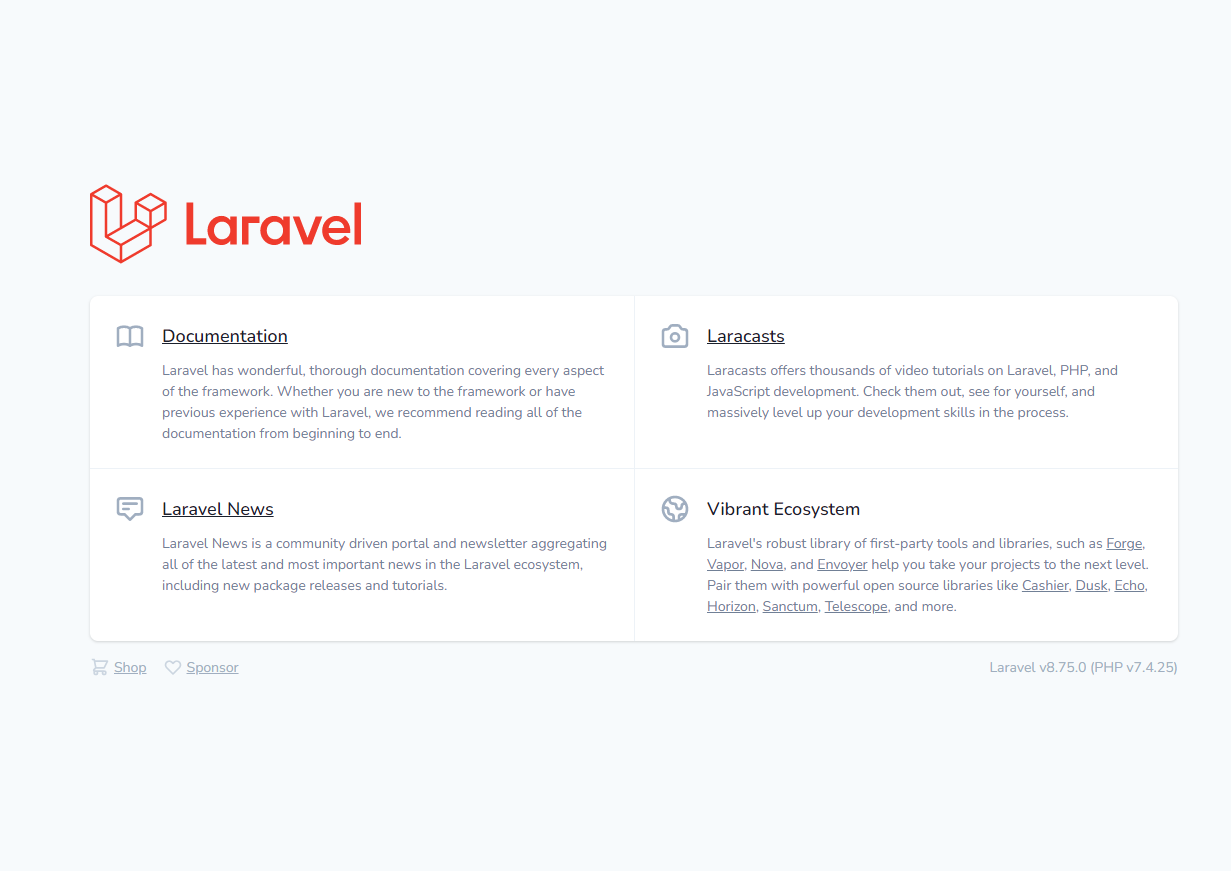
nginx/conf.d/app.conf

server {  
 listen 80;  
 index index.php index.html;  
 error\_log /var/log/nginx/error.log;  
 access\_log /var/log/nginx/access.log;  
 root /var/www/public;  
 location ~ \.php$ {  
 try\_files $uri =404;  
 fastcgi\_split\_path\_info ^(.+\.php)(/.+)$;  
 fastcgi\_pass app:9000;  
 fastcgi\_index index.php;  
 include fastcgi\_params;  
 fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;  
 fastcgi\_param PATH\_INFO $fastcgi\_path\_info;  
 }  
 location / {  
 try\_files $uri $uri/ /index.php?$query\_string;  
 gzip\_static on;  
 }  
}

Запустим docker при помощи команды docker-compose up



Далее установим Laravel. При переходе на <http://localhost/> получаем следующий результат:



# 4. Разработка и проектирование базы данных

В процессе выполнения были изучены основы взаимодействия web-приложения с базой данных и спроектирована база данных для хранения информации приложения (страницы и пользователи).

В качестве СУБД для разработки была использована PostgreSQL, в качестве ORM - асинхронная библиотека Tortoise.

В ходе выполнения было разработано 2 модели:

1. User (для пользователей)

2. Projects (для карточек проекта и их внутреннем содержании)

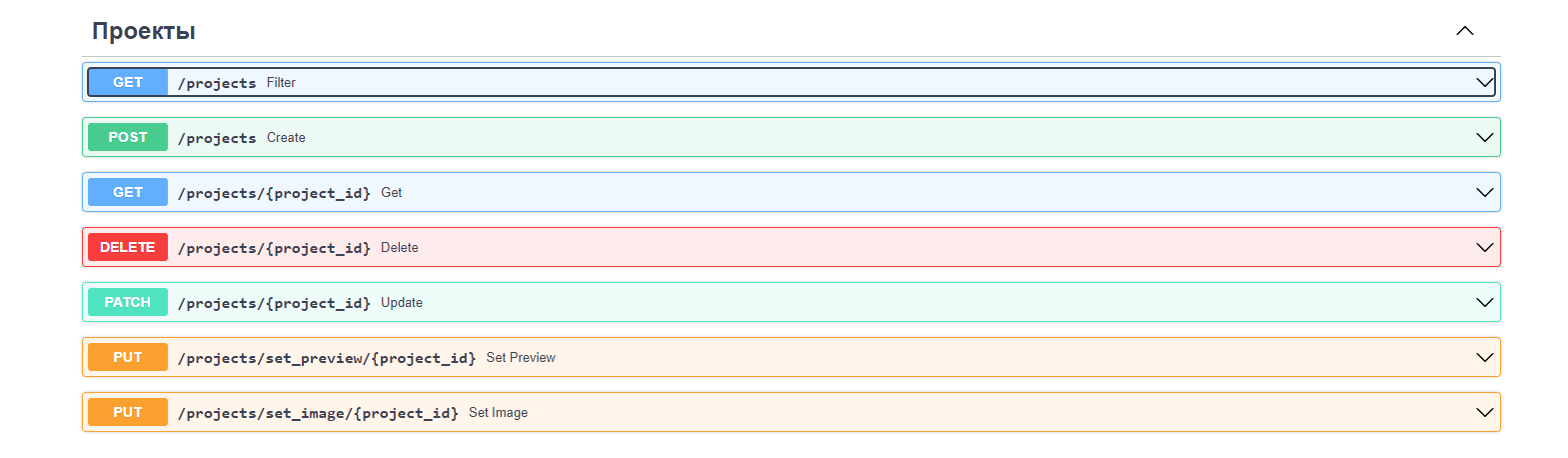
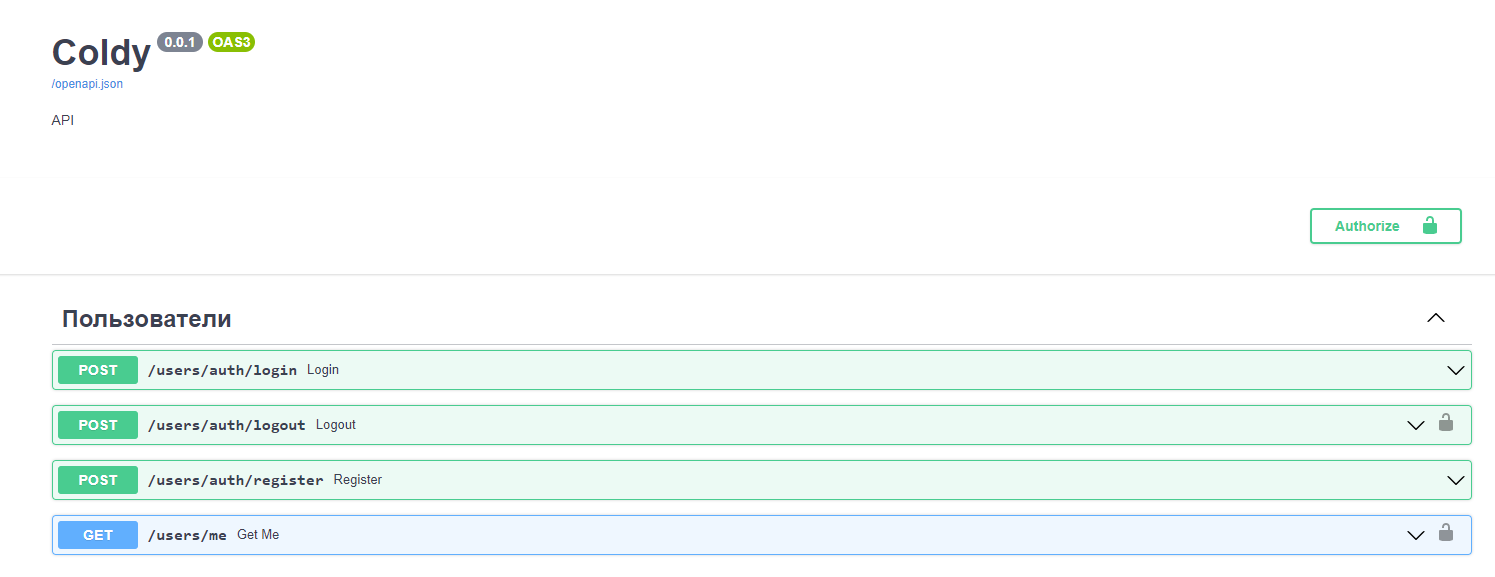
В качестве демонстрации работы была выбрана модель Projects:

from tortoise import Tortoise, fields, models  
  
from config import DBPaths  
  
Tortoise.init\_models(DBPaths.project, 'models')  
  
  
class Project(models.Model):  
 id = fields.BigIntField(pk=True)  
  
 name = fields.CharField(max\_length=150, default="", description="Название")  
 city = fields.CharField(max\_length=100, default="", description="Город")  
 address = fields.CharField(max\_length=100, default="", description="Адрес")  
 title = fields.CharField(max\_length=100, default="", description="Заголовок")  
 description = fields.CharField(max\_length=100, default="", description="Описание")  
  
 type = fields.CharField(max\_length=100, default="", description="Класс")  
 square = fields.FloatField(default=0, description="Площадь")  
 count\_levels = fields.IntField(default=0, description="Количество этажей")  
 year = fields.IntField(default=0, description="Год сдачи")  
  
 preview = fields.ForeignKeyField(  
 'models.File', related\_name='previewed', on\_delete=fields.SET\_NULL,  
 null=True, default=None, description='Превью'  
 )  
  
 image = fields.ForeignKeyField(  
 'models.File', related\_name='imaged', on\_delete=fields.SET\_NULL,  
 null=True, default=None, description='Большая картинка'  
 )

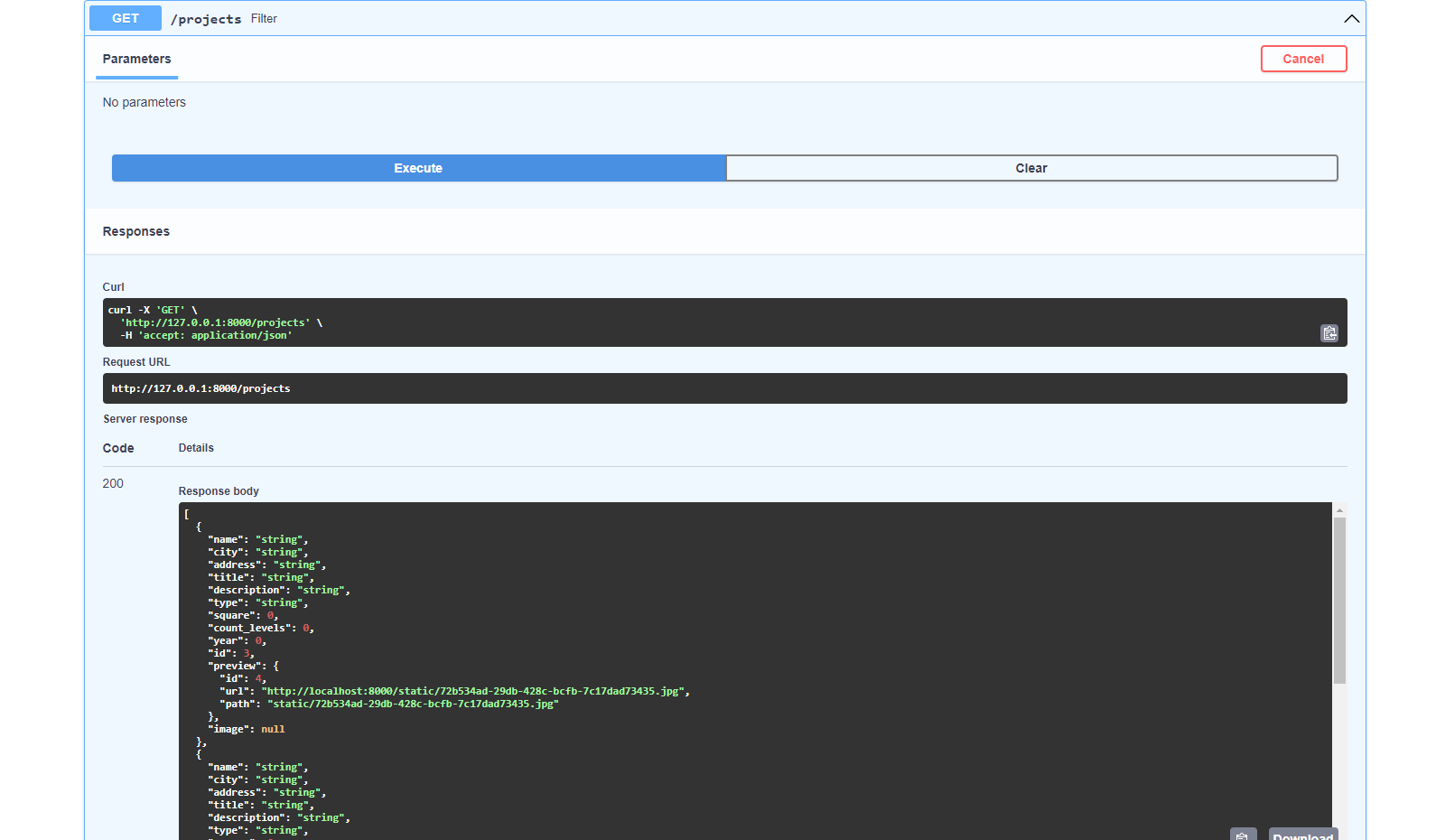
# 5. REST API

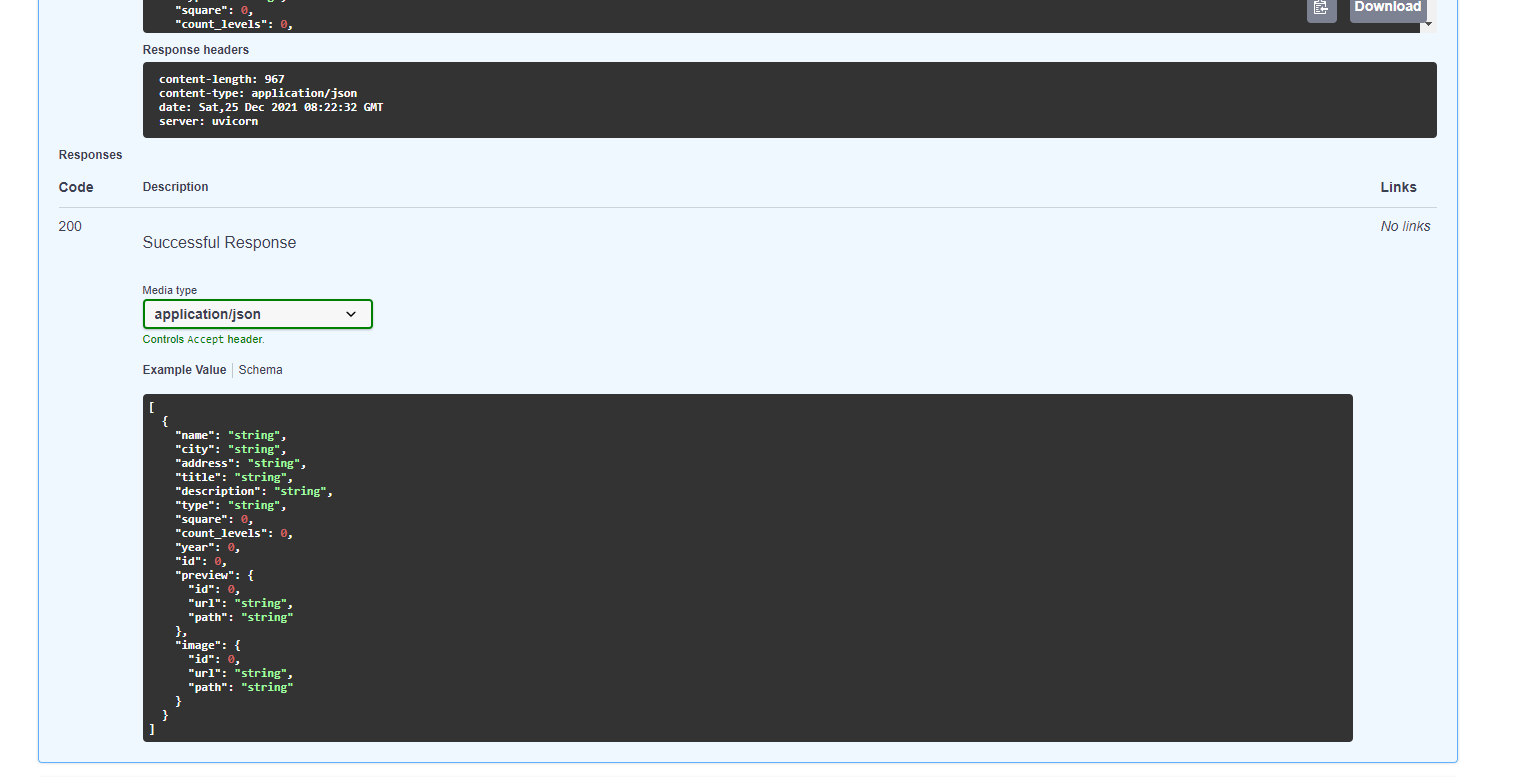
В процессе выполнения были изучены основы разработки API для web-приложений и разработан REST API для своего проекта.

Описание API:



Для примера работы приведено содержимое GET projects Filter





# 6. Работа с HTTP запросами

В процессе выполнения были изучены принципы работы отправки Ajax запросов. Изучена работа отправки синхронных и асинхронных запросов в Vue JS. Получены навыки работы с промисами.

Для работы с Ajax запросами была выбрана библиотека Axios.

Содержимое файла api/index.js

import ***Axios*** from "axios";  
  
const axiosConfig = {  
 baseURL: 'http://127.0.0.1:8000',  
 timeout: 30000,  
 withCredentials: true  
};  
const ***index*** = ***Axios***.create(axiosConfig);  
  
export default ***index***

Содержимое файла api/projects.js

import ***axios*** from "./index";  
  
const baseAuthURL = '/projects'  
  
const getData = response => response.data  
  
export function getProjects() {  
 return ***axios***.get(`${baseAuthURL}`)  
 .then(getData)  
}

Содержимое файла api/auth.js

import ***axios*** from "./index";  
  
const baseAuthURL = '/users'  
  
export function login({username, password}) {  
 const bodyFormData = new FormData();  
  
 bodyFormData.append('username', username)  
 bodyFormData.append('password', password)  
  
 return ***axios***.post(`${baseAuthURL}/auth/login`, bodyFormData)  
}  
  
export function getUserName() {  
 return ***axios***.get(`${baseAuthURL}/me`)  
 .then(response => response.data.first\_name)  
}  
  
export function register(userData) {  
 return ***axios***.post(`${baseAuthURL}/auth/register`, userData)  
}  
  
  
export function logout() {  
 return ***axios***.post(`${baseAuthURL}/auth/logout`)  
}

В результате получаем веб-приложение, благодаря которому пользователь может регистрироваться, входить в аккаунт, выходить из него и просматривать проекты.

