

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

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

Дисциплина: Технологии web-программирования

Расчетно-графическое задание

«Создание Web-приложения»

Выполнил: ст. гр. ПВ-42

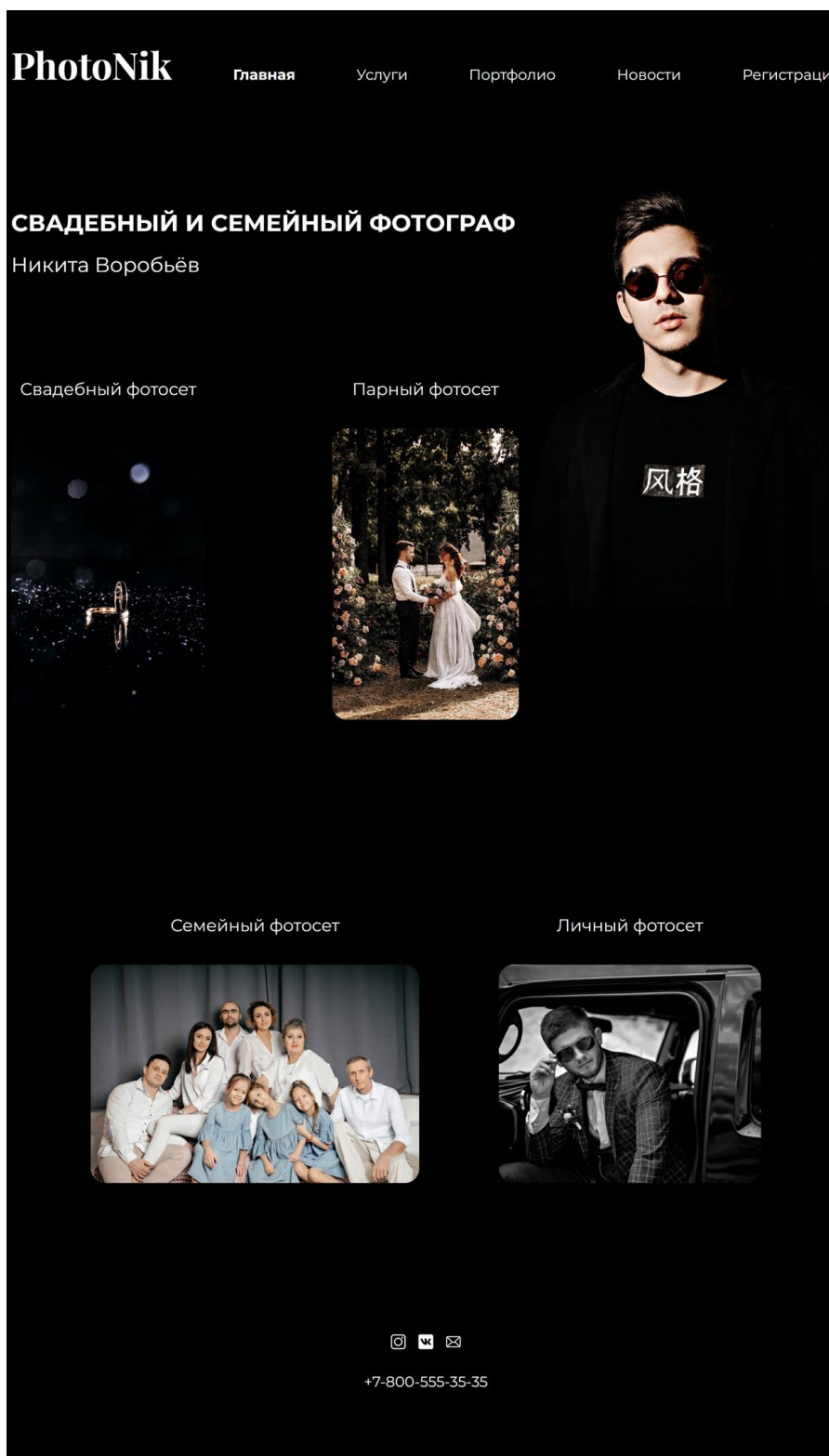
Ковынев В.В.

Проверил: Картамышев С.В.

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

В ходе выполнения работы были созданы следующие страницы:

1) Главная



2) Услуги

PhotoNik

Главная

Услуги

Портфолио

Новости

Регистрация

Личная фотосессия

30 минут

1 час

3 часа

6 часов

1000

2000

4500

7500

Свадебная фотосессия

30 минут

1 час

3 часа

6 часов

1000

2000

4500

7500

Семейная фотосессия

30 минут

1 час

3 часа




6 часов

1000

2000

4500

7500



+7-800-555-35-35

3)Портфолио

PhotoNik

[Главная](#)

[Услуги](#)

[Портфолио](#)

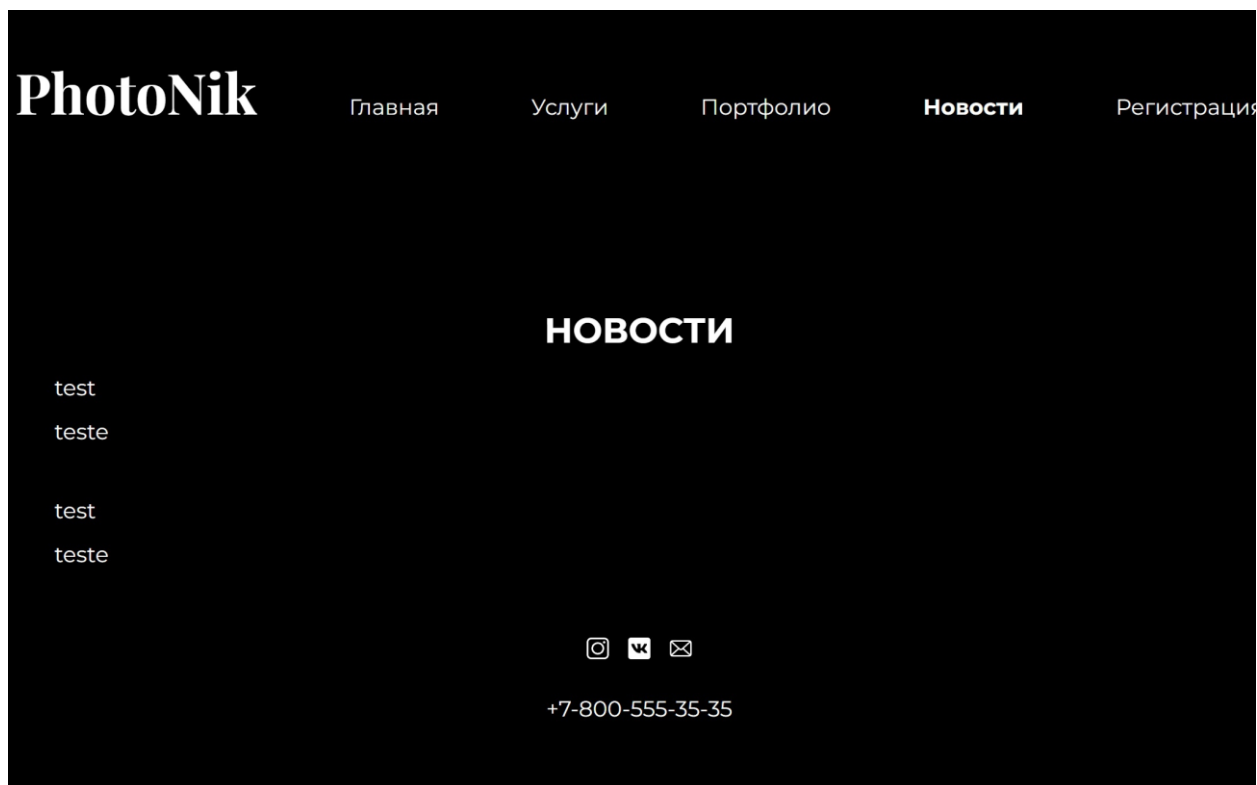
[Новости](#)

[Регистрация](#)

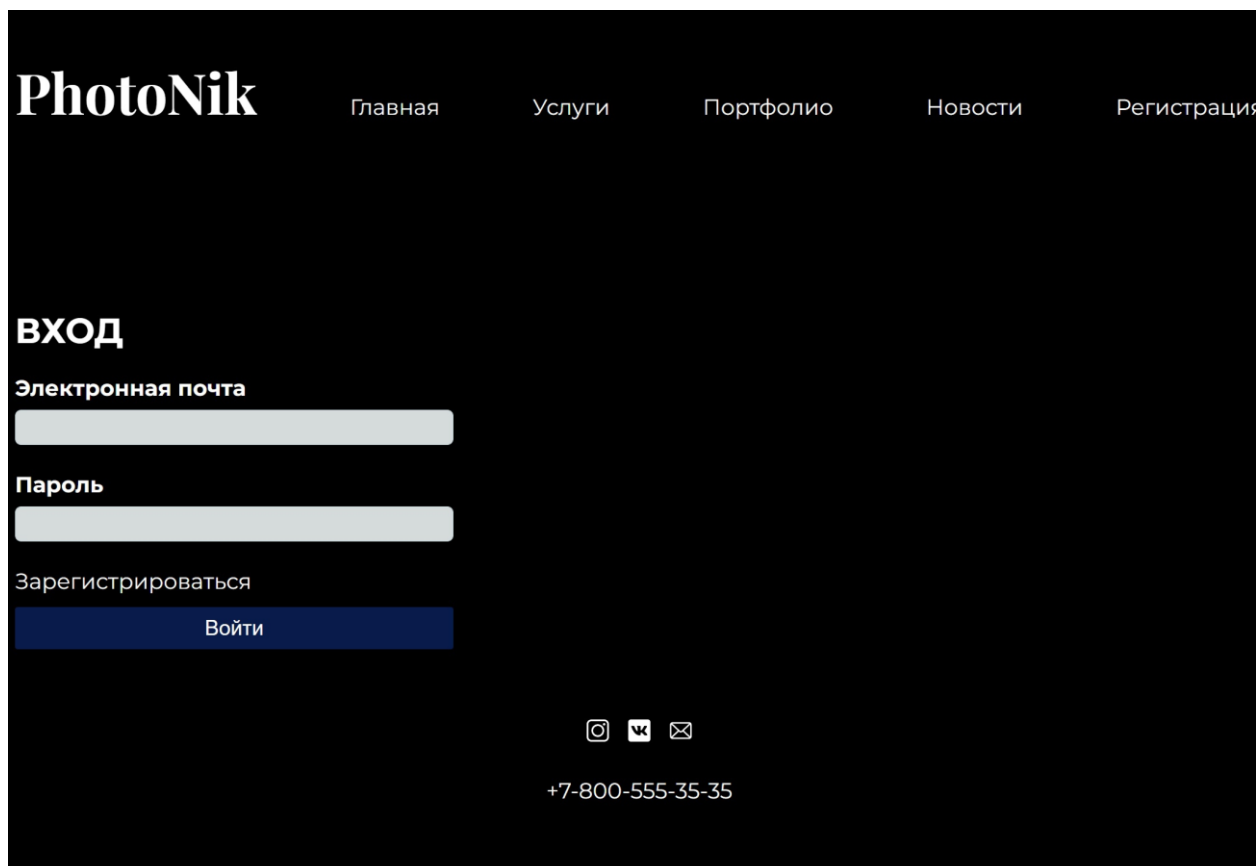


+7-800-555-35-35

4)Новости



5)Авторизация



6)Регистрация

PhotoNik[Главная](#)[Услуги](#)[Портфолио](#)[Новости](#)**Регистрация**

РЕГИСТРАЦИЯ




Электронная почта

Пароль

Подтверждение пароля

Войти, если уже зарегистрированы

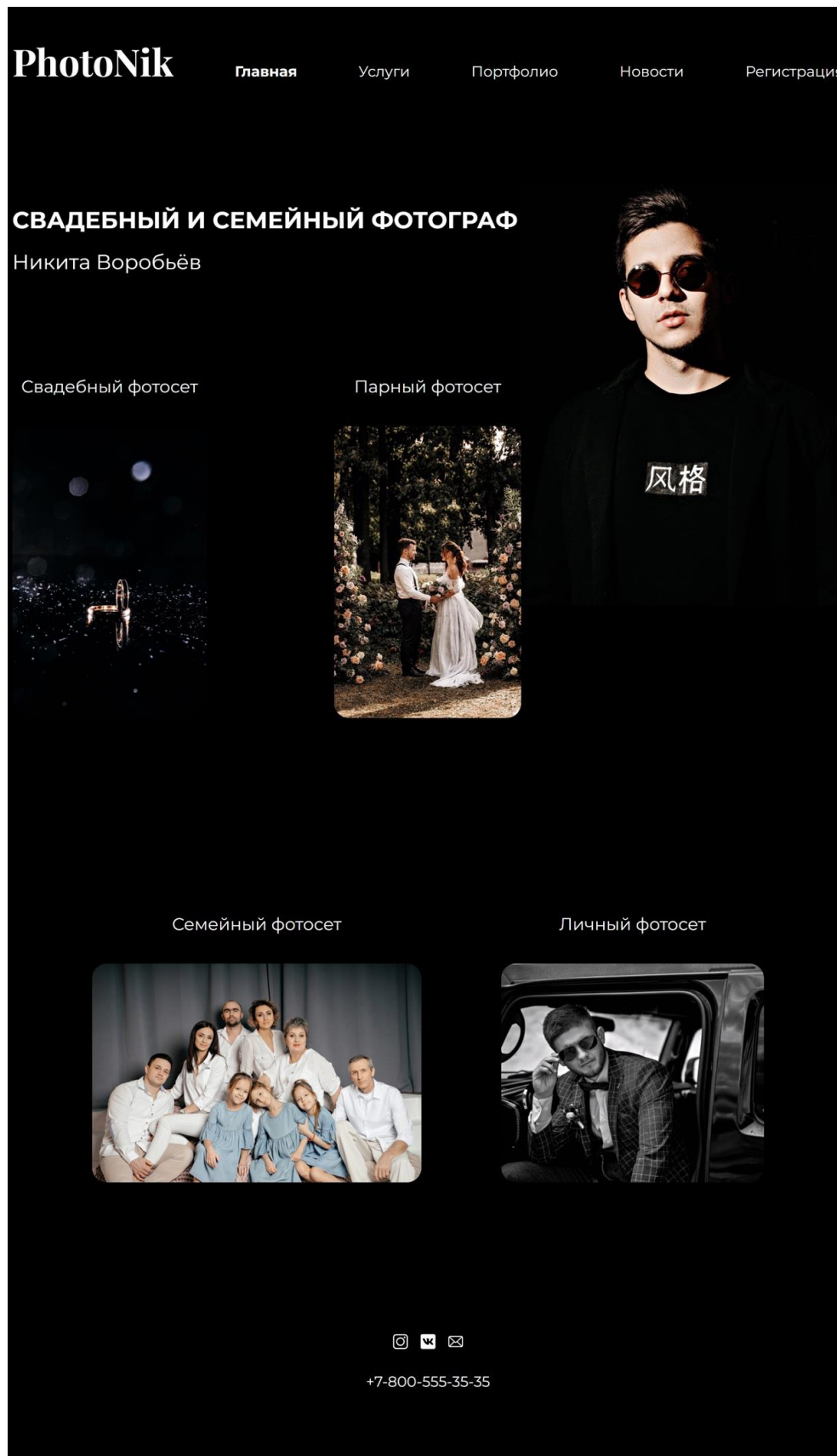
[Зарегистрироваться](#)



+7-800-555-35-35

Клиентское программирование. Vue.js

В ходе работы был изучен фреймворк Vue.js, верстка прошлой работы перенесена в компоненты. Внешний вид получившегося сайта:



Личная фотосессия

[30 минут](#)[1 час](#)[3 часа](#)[6 часов](#)[1000](#)[2000](#)[4500](#)[7500](#)

Свадебная фотосессия

[30 минут](#)[1 час](#)[3 часа](#)[6 часов](#)[1000](#)[2000](#)[4500](#)[7500](#)

Семейная фотосессия

[30 минут](#)[1 час](#)[3 часа](#)[6 часов](#)[1000](#)[2000](#)[4500](#)[7500](#)

+7-800-555-35-35



+7-800-555-35-35

НОВОСТИ

test

teste

test

teste



+7-800-555-35-35

РЕГИСТРАЦИЯ

Электронная почта

Пароль

Подтверждение пароля

Войти, если уже зарегистрированы

[Зарегистрироваться](#)



+7-800-555-35-35

ВХОД

Электронная почта

Пароль

Зарегистрироваться

Войти



+7-800-555-35-35

Серверное программирование. Laravel.

Для работы был выбран фреймворк ASP.NET, язык программирования C#. В ходе работы был создан контроллер пользователей.

Код контроллера со статическими данными:

```
using Microsoft.AspNetCore.Mvc;

// For more information on enabling Web API for empty projects, visit
https://go.microsoft.com/fwlink/?LinkID=397860

namespace WebBackend.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class StaticDataController : ControllerBase
    {
        private List<string> values = new List<string> { "value1", "value2" };
        // GET: api/<StaticDataController>
        [HttpGet]
        public IEnumerable<string> Get()
        {
            return values;
        }

        // GET api/<StaticDataController>/5
        [HttpGet("{id}")]
        public async Task<ActionResult<string>> Get(int id)
        {
            return Ok(values[id]);
        }

        // POST api/<StaticDataController>
        [HttpPost]
        public async Task<ActionResult> Post([FromBody] string value)
        {
            values.Add(value);
            return Ok();
        }

        // PUT api/<StaticDataController>/5
        [HttpPut("{id}")]
        public async Task<ActionResult> Put(int id, [FromBody] string value)
        {
            if(id >= values.Count)
            {
                return BadRequest();
            }

            values[id] = value;
            return Ok();
        }

        // DELETE api/<StaticDataController>/5
        [HttpDelete("{id}")]
        public async Task<ActionResult> Delete(int id)
        {
            if (id >= values.Count)
            {
                return BadRequest();
            }
        }
    }
}
```

```

        values.RemoveAt(id);
        return Ok();
    }
}
}

```

Dockerfile для приложения WebBackend:

```

FROM mcr.microsoft.com/dotnet/aspnet:6.0 AS base
WORKDIR /app
EXPOSE 80
EXPOSE 443

FROM mcr.microsoft.com/dotnet/sdk:6.0 AS build
WORKDIR /src
COPY ["WebBackend.csproj", "."]
RUN dotnet restore "./WebBackend.csproj"
COPY . .
WORKDIR "/src/."
RUN dotnet build "WebBackend.csproj" -c Release -o /app/build

FROM build AS publish
RUN dotnet publish "WebBackend.csproj" -c Release -o /app/publish

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "WebBackend.dll"]

```

Файл docker-compose.yml:

```

version: "3.9"

services:
  backend:
    build: ./WebBackend
    restart: unless-stopped
    depends_on:
      - mysql
    ports:
      - "80:80"
    networks:
      - web-app-network

networks:
  web-app-network:
    driver: bridge

```

Протестируем работу бэкенда с помощью Postman:

GET

localhost/api/StaticData

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body

Cookies

Headers (4)

Test Results

Status: 200 OK

Time: 32 ms

Size: 167 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

[

2

"value1",

3

"value2"

4

]

POST

localhost/api/StaticData

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

1

"value"

Body

Cookies

Headers (3)

Test Results

Status: 200 OK

Time: 13 ms

Size: 92 B

Save Response

Pretty

Raw

Preview

Visualize

Text

1

GET

localhost/api/StaticData

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

1

Body

Cookies

Headers (4)

Test Results

Status: 200 OK

Time: 48 ms

Size: 176 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

[

2

"value1",

3

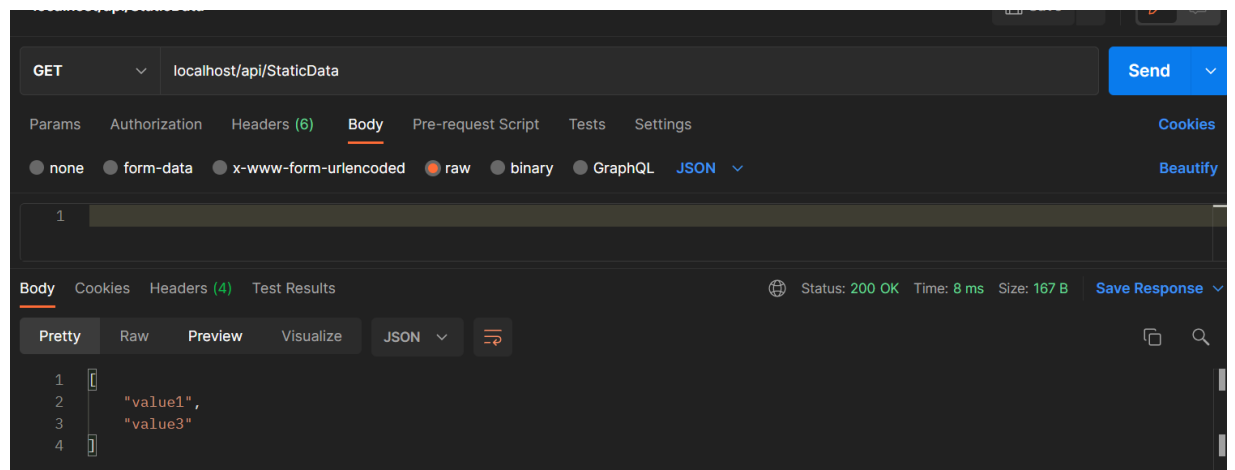
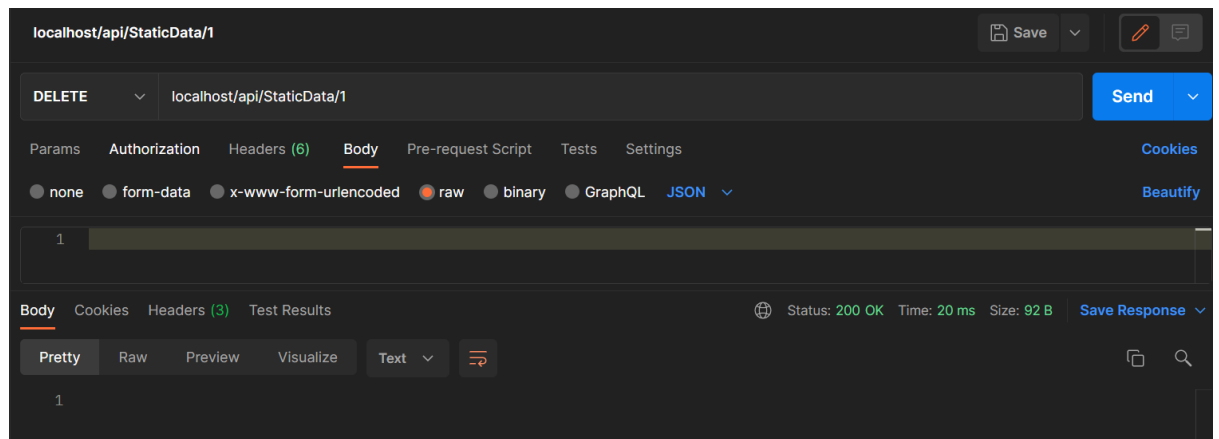
"value2",

4

"value3"

5

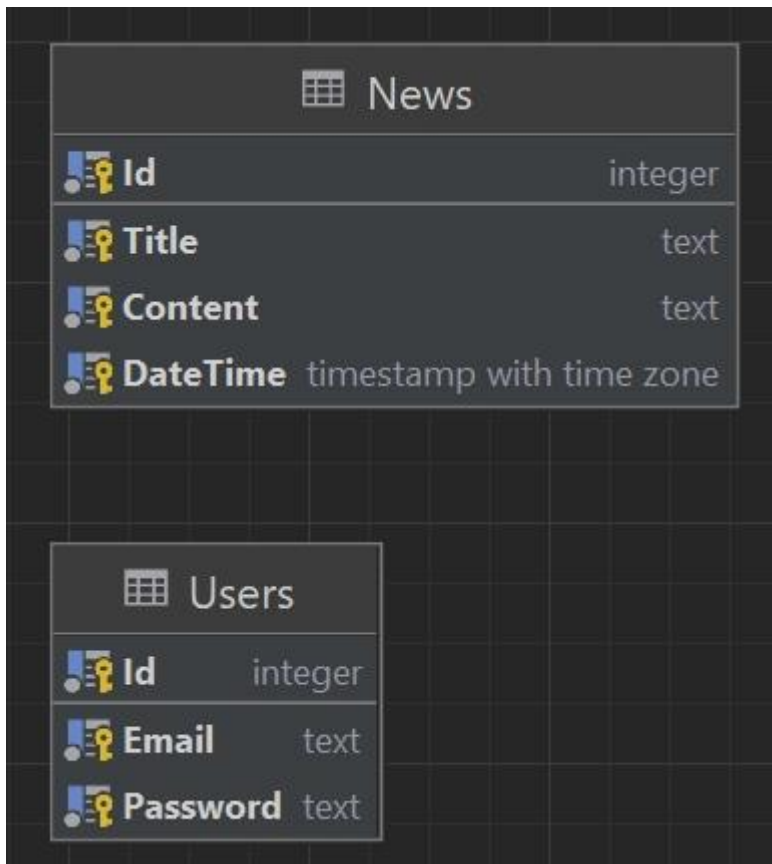
]



Проектирование базы данных приложения.

Выполнение:

В ходе выполнения работы была создана следующая структура БД:



Код модели User:

```
using System.Text.Json.Serialization;

namespace WebBackend.Models;

public class User
{
    public int Id { get; set; }
    public string Email { get; set; }
    [JsonIgnore]
    public string Password { get; set; }
}
```


Код модели News:

```
namespace WebBackend.Models;

public class News
{
    public int Id { get; set; }
    public string Title { get; set; }
    public string Content { get; set; }
    public DateTime DateTime { get; set; }
}
```

Код контекста базы данных:

```
using Microsoft.EntityFrameworkCore;

namespace WebBackend.Models;

public sealed class WebDbContext :
    DbContext
{
    public DbSet<User> Users { get; set; }
    public DbSet<News> News { get; set; }

    public WebDbContext(DbContextOptions
contextOptions) : base(contextOptions)
    {
        Database.EnsureCreated();
    }
}
```

REST API

Описание REST API в Swagger:

News		^
GET	/api/News	▼
POST	/api/News	▼
GET	/api/News/{id}	▼
User		^
POST	/api/User/Login	▼
POST	/api/User/Register	▼

Проверим работу API в Postman:

The screenshot shows the Postman interface. At the top, a GET request is configured for the URL `http://localhost:9000/api/News`. The 'Body' tab is selected, showing a raw JSON payload:

```
{  "email": "test@test.com",  "password": "test12133password"}
```

. Below the request, the 'Body' tab of the response is shown, displaying a JSON array of two news objects:

```
[  {    "id": 2,    "title": "test",    "content": "teste",    "dateTime": "2022-01-18T14:04:12.629167Z"  },  {    "id": 1,    "title": "test",    "content": "teste",    "dateTime": "2022-01-18T14:04:12.629176Z"  }]
```

. The status bar at the bottom indicates a successful response with status `200 OK`, a time of `68 ms`, and a size of `315 B`.

POST ▼ http://localhost:9000/api/User/Login Send ▼

Params Authorization Headers (8) **Body** ● Pre-request Script Tests Settings Cookies

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▼ Beautify

```
1 {
2   ... "email": "test@test.com",
3   ... "password": "test12133password"
4 }
```

Body Cookies Headers (4) Test Results 🌐 Status: 200 OK Time: 113 ms Size: 506 B Save Response ▼

Pretty Raw Preview Visualize **JSON** ▼ 🔍

```
1 {
2   "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzbnZlLnR5cy8yMDA1LzA1L2lkZW50aXR5L2NsYW1tcy9uYW11IjoidGVzdEB0ZXN0LmNvbSI6Im5iZiI6MTY0MjUzNTM5NTc5LCJpc3MiOiJodHRwOi8vbG9jYWxob3N0OjkwMDAiLCJhdWQiOiJodHRwOi8vbG9jYWxob3N0In0.hWcbIxmBoPNSB0G4R-K1yAWM04pMDqGYXk31fns9EEM",
3   "email": "test@test.com",
4   "userId": 11
5 }
```

POST ▼ http://localhost:9000/api/User/Register Send ▼

Params Authorization Headers (8) **Body** ● Pre-request Script Tests Settings Cookies

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▼ Beautify

```
1 {
2   ... "email": "test@test.com",
3   ... "password": "test12133password",
4   ... "confirmPassword": "test12133password"
5 }
```



Body Cookies Headers (4) Test Results 🌐 Status: 200 OK Time: 401 ms Size: 506 B Save Response ▼

Pretty Raw Preview Visualize **JSON** ▼ 🔍

```
1 {
2   "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzbnZlLnR5cy8yMDA1LzA1L2lkZW50aXR5L2NsYW1tcy9uYW11IjoidGVzdEB0ZXN0LmNvbSI6Im5iZiI6MTY0MjUzNTM5NTc5LCJpc3MiOiJodHRwOi8vbG9jYWxob3N0OjkwMDAiLCJhdWQiOiJodHRwOi8vbG9jYWxob3N0In0.CdySVd3WPE7jZB7y16tNbaf0qpKunTgPyo3F2F-gVck",
3   "email": "test@test.com",
4   "userId": 11
5 }
```

http://localhost:9000/api/News

Save



POST

http://localhost:9000/api/News

Send

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Cookies

Headers

8 hidden

	KEY	VALUE	DESCRIPTION		Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...				
	Key	Value	Description			

Body

Cookies

Headers (5)

Test Results

Status: 201 Created

Time: 800 ms

Size: 279 B

Save Response


Pretty



Raw

Preview

Visualize

JSON





1

2

3

4

5

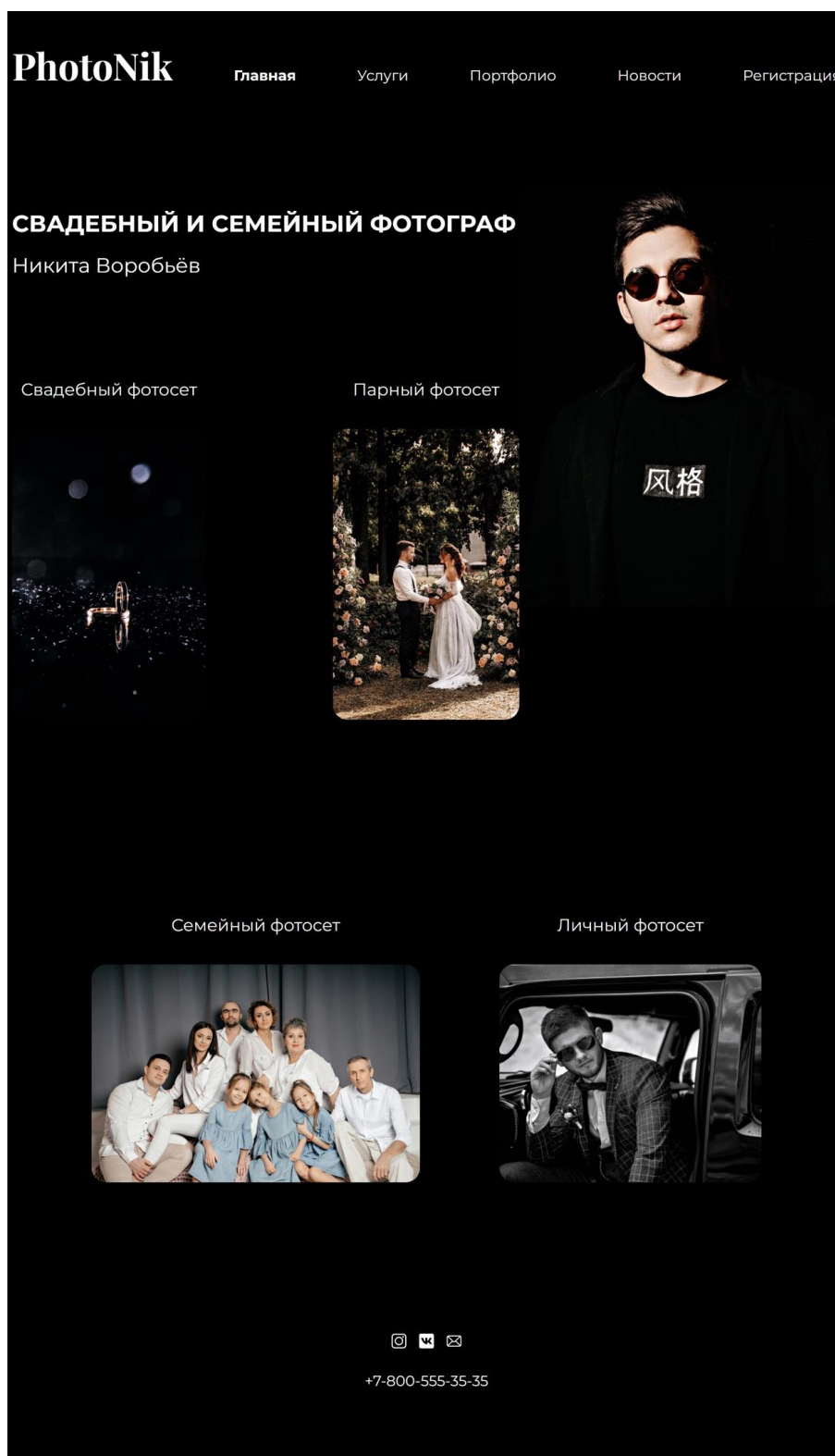
6

```
"id": 3,
"title": "test",
"content": "test",
"dateTime": "2022-01-18T20:46:45.2459498Z"
```

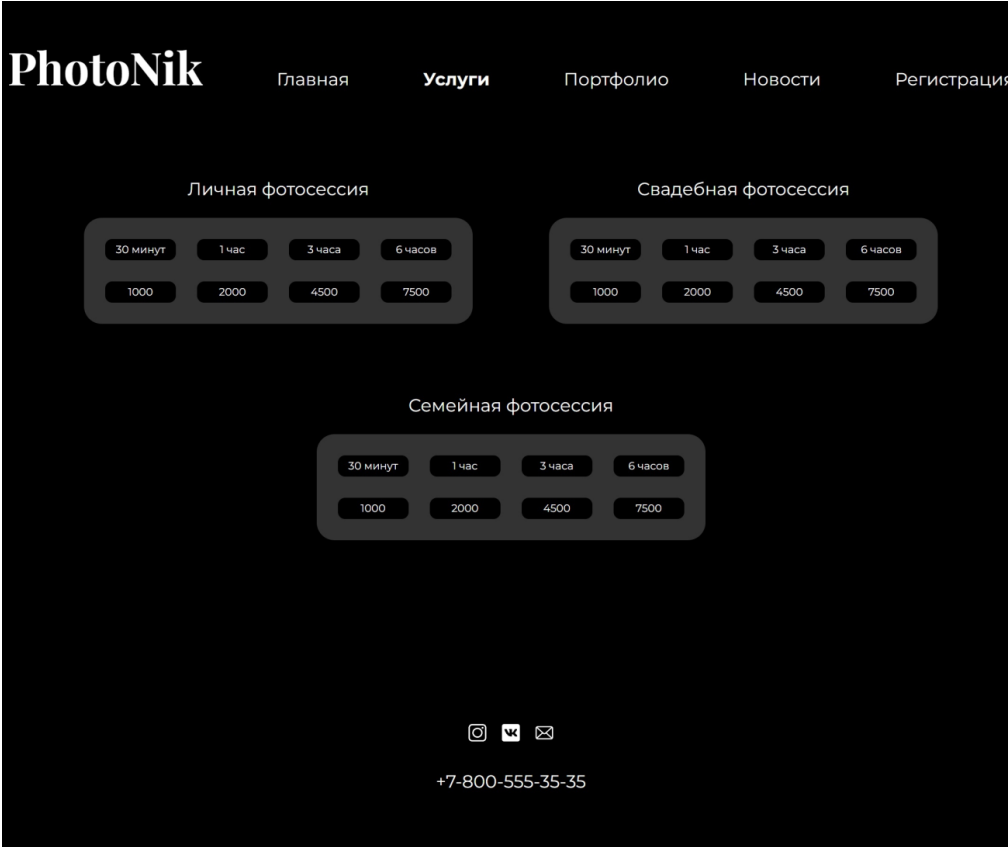
Работа с HTTP-Запросами

Выполнение:

1) Главная



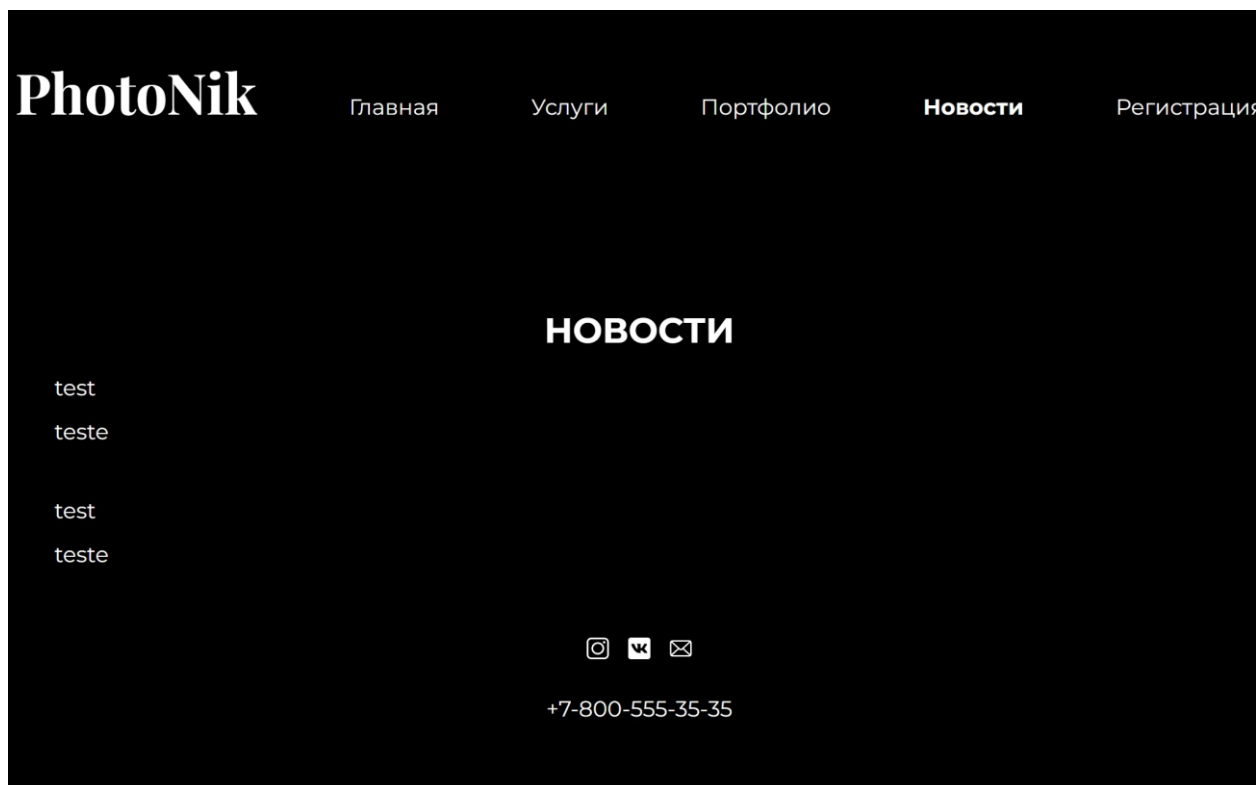
2)Услуги



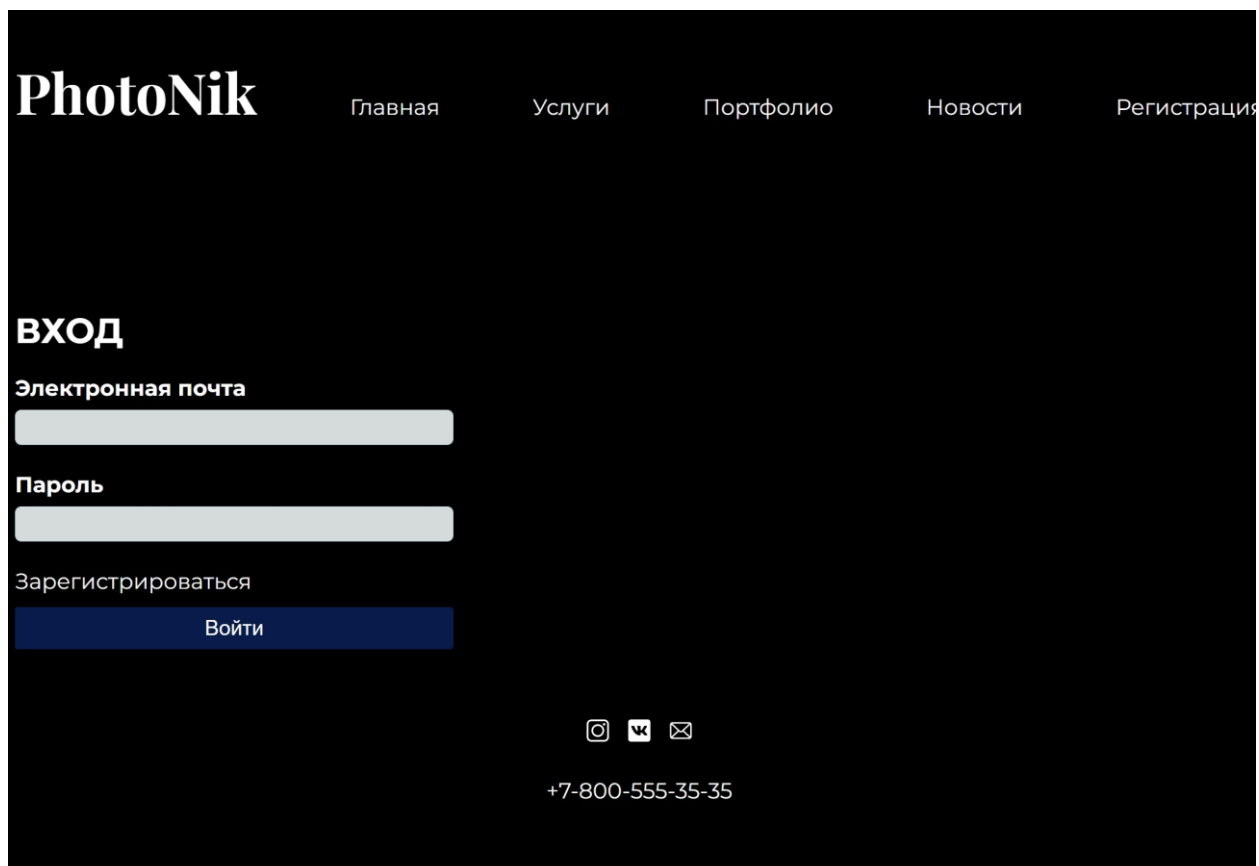
3)Портфолио



4)Новости



5)Авторизация



6)Регистрация

PhotoNik[Главная](#)[Услуги](#)[Портфолио](#)[Новости](#)**Регистрация**

РЕГИСТРАЦИЯ




Электронная почта

Пароль

Подтверждение пароля

Войти, если уже зарегистрированы

[Зарегистрироваться](#)



+7-800-555-35-35