

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
Расчётно-графическое задание

Выполнил: ст. гр. ПВ-42

Борисенко Валерий Александрович

Проверил: Картамышев С.В.

Цель работы: познакомиться с основами разработки на языке JavaScript. Познакомиться с основами разработки frontend приложения, в частности с использованием клиентских фреймворков (Vue.js). Научиться переносить вёрстку в отдельные компоненты.

Задание к лабораторной работе:

1. Изучить основы разработки на языке JavaScript.
2. Изучить основы разработки frontend-приложения.
3. Развернуть базовое приложение на фреймворке Vue.js.
4. Добавить необходимые компоненты и перенести в них вёрстку, сделанную в прошлой лабораторной работе.
5. Продемонстрировать работу web-приложения.

Выполнение:

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

The screenshot shows a web application interface for a clinic. At the top, there is a header bar with a light blue background. On the left, it contains a phone icon and the text "Единая справочная +7 (4722) 50-48-48". In the center, there are links for "Вход" (Login) and "Регистрация" (Registration) next to a LinkedIn icon. On the right, it displays the address: "Адрес: улица Некрасова, дом 8/9, Белгород, 308007". Below the header, there is a main section with a light blue background. On the left, there is a blue circle with a white cross icon. To its right, the text reads "ОГБУЗ «Белгородская областная клиническая больница Святителя Иоасафа»". On the right side of this section, there are three links: "Главная" (Home), "О больнице" (About the hospital), and "Контакты" (Contacts). In the center of the page, there is a section titled "Авторизация" (Authorization) with a computer monitor icon. Below the title, there is a light blue rounded rectangle containing two input fields: "Логин" (Login) and "Пароль" (Password). Below the password field is a button labeled "Войти" (Login). At the bottom of the page, there is a light blue footer bar with a button labeled "Наверх" (Up) on the right side.



Единая справочная
+7 (4722) 50-48-48

[Вход](#)

[Регистрация](#)



Адрес:
улица Некрасова, дом 8/9, Белгород, 308007



ОГБУЗ «Белгородская областная
клиническая больница Святителя Иоасафа»

[Главная](#)

[О больнице](#)

[Контакты](#)

Регистрация

Логин

Пароль

Повторите пароль

[Зарегистрироваться](#)

[Наверх](#)



Единая справочная
+7 (4722) 50-48-48

[Вход](#)

[Регистрация](#)



Адрес:
улица Некрасова, дом 8/9, Белгород, 308007



ОГБУЗ «Белгородская областная
клиническая больница Святителя Иоасафа»

[Главная](#)

[О больнице](#)

[Контакты](#)

Записаться на приём

Ваши фамилия, имя, отчество

Ваш E-mail или номер телефона

Ваше сообщение

[Отправить](#)

[Наверх](#)

[illegible]

Успышны

[illegible]

Режим работы

[illegible]

Платные услуги

[illegible]

Covid-19



Контактная информация

[illegible]

О больнице

[illegible]

Цель работы: познакомиться с основами backend разработки web-приложений. Научиться разворачивать проект, производить его настройку. Научится работать с API в приложении Postman.

Задание к лабораторной работе:

1. Развернуть базовое приложение ASP.NET.
2. Настроить конфигурацию работы приложения с docker.
3. Добавить модуль для работы с API.
4. Добавить несколько контроллеров со статическими данными.
5. Продемонстрировать работу API в Postman.

Выполнение:

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

Код контроллера со статическими данными:

```
using Microsoft.AspNetCore.Mvc;

namespace BackendApp.Controllers;

[ApiController]
[Route("[controller]")]
public class WeatherForecastController : ControllerBase
{
    private static readonly string[] Summaries = new[]
    {
        "Freezing", "Bracing", "Chilly", "Cool", "Mild", "Warm", "Balmy",
        "Hot", "Sweltering", "Scorching"
    };

    private readonly ILogger<WeatherForecastController> _logger;

    public WeatherForecastController(ILogger<WeatherForecastController>
logger)
    {
```

```

        _logger = logger;
    }

    [HttpGet(Name = "GetWeatherForecast")]
    public IEnumerable<WeatherForecast> Get()
    {
        return Enumerable.Range(1, 5).Select(index => new WeatherForecast
        {
            Date = DateTime.Now.AddDays(index),
            TemperatureC = Random.Shared.Next(-20, 55),
            Summary = Summaries[Random.Shared.Next(Summaries.Length)]
        })
        .ToArray();
    }
}

```

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 ["BackendApp.csproj", "."]
RUN dotnet restore "BackendApp.csproj"
COPY . .
WORKDIR "/src/"
RUN dotnet build "BackendApp.csproj" -c Release -o /app/build

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

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

```

Файл docker-compose.yml:

```

version: "3.9"
services:
  database:
    container_name: database

```

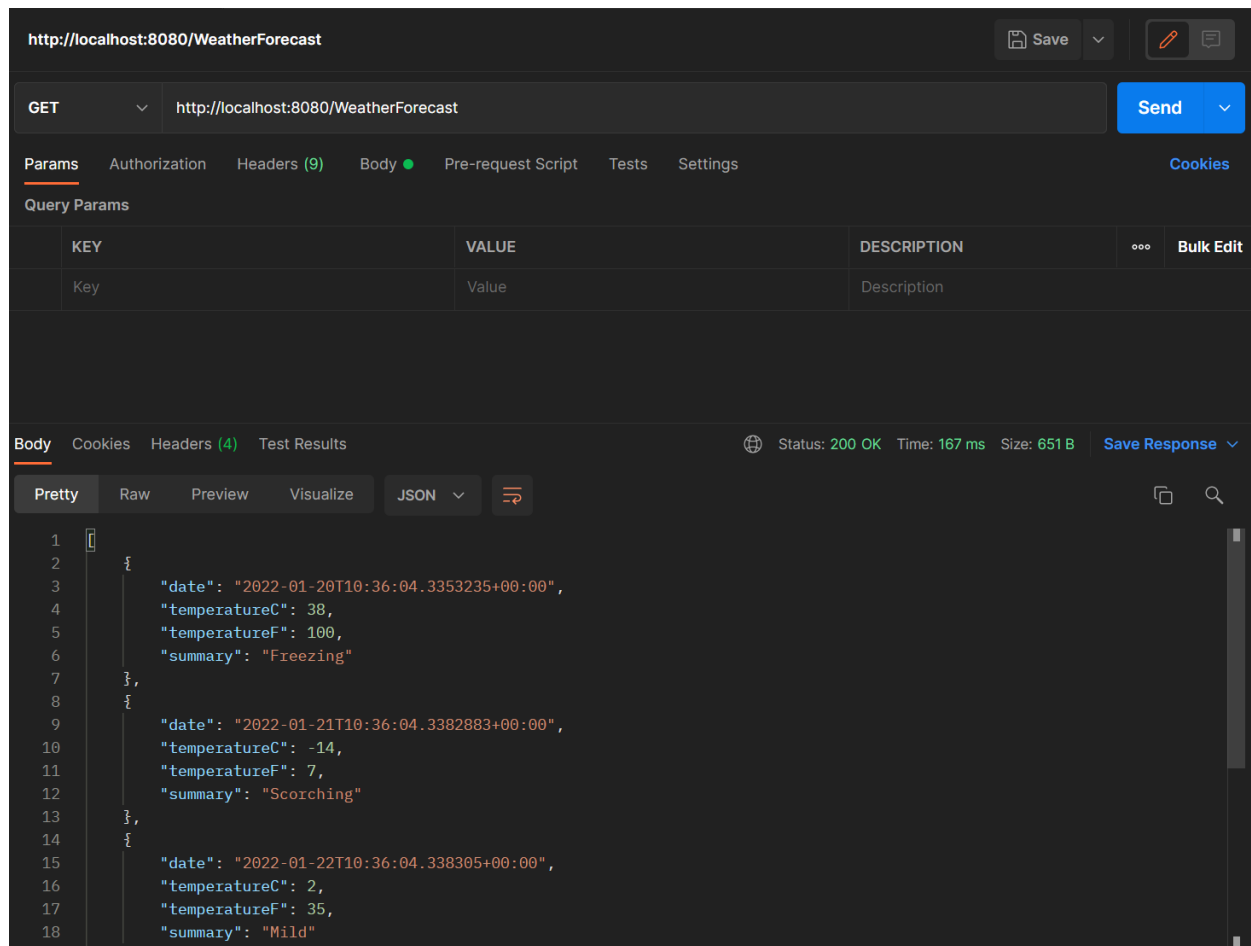
image: postgres:14
restart: unless-stopped
environment:
 POSTGRES_PASSWORD: postgrespassword
 POSTGRES_USER: boosterjz
 POSTGRES_DB: database
ports:
 - "5432:5432"
networks:
 - web-network
volumes:
 - dbdata:/var/lib/postgresql/data

backend:
 container_name: backend
 build: ./BackendApp
 ports:
 - "8080:80"
 restart: unless-stopped
 depends_on:
 - db
 networks:
 - web-network

frontend:
 container_name: frontend
 build: .
 ports:
 - "80:80"
 restart: unless-stopped
 depends_on:
 - backend
 volumes:
 - ./nginx:/etc/nginx/conf.d
 networks:
 - web-network

networks:
 web-network:
 driver: bridge

volumes:
 dbdata:
driver: local



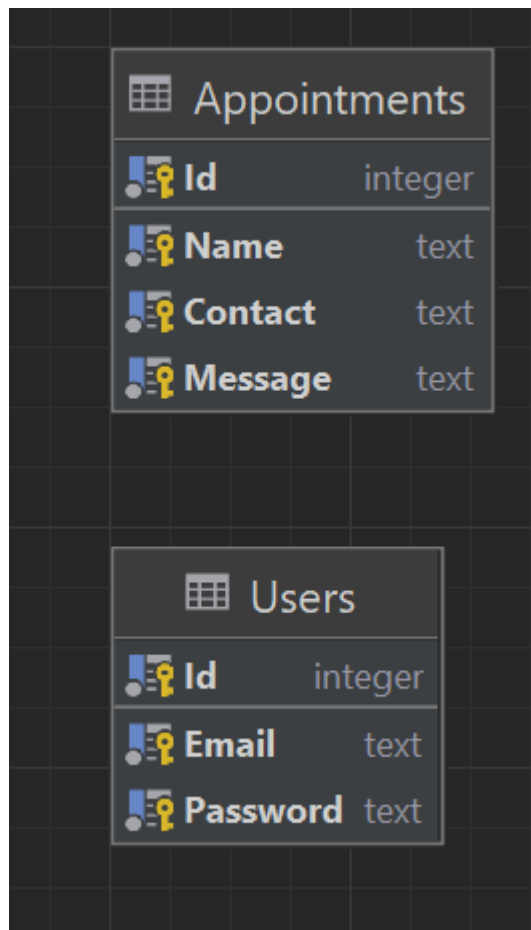
Цель работы: изучить основы взаимодействия web-приложения с базой данных. Спроектировать базу данных для хранения информации приложения (страницы, пользователи и т.п.).

Задание к лабораторной работе:

1. Выбрать подходящую СУБД.
2. Изучить методы взаимодействия web-приложения с базой данных (Eloquent).
 - а. Разработать структуру базы данных.
 - б. Разработать соответствующие модели в приложении.
 - в. В отчёт приложить схему базы данных, а также код одной из моделей.

Выполнение:

В ходе выполнения работы была создана следующая БД:



```
using Microsoft.EntityFrameworkCore;
```

```
namespace BackendApp.Models;
```

```
public class Context : DbContext
{
    public DbSet<Appointment> Appointments { get; set; }
    public DbSet<User> Users { get; set; }

    public Context(DbContextOptions options) : base(options)
    {
        Database.EnsureCreated();
    }
}
```

```
namespace BackendApp.Models;
```

```
public class Appointment
{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Contact { get; set; }
```

```
    public string Message { get; set; }  
}
```

```
namespace BackendApp.Models;
```

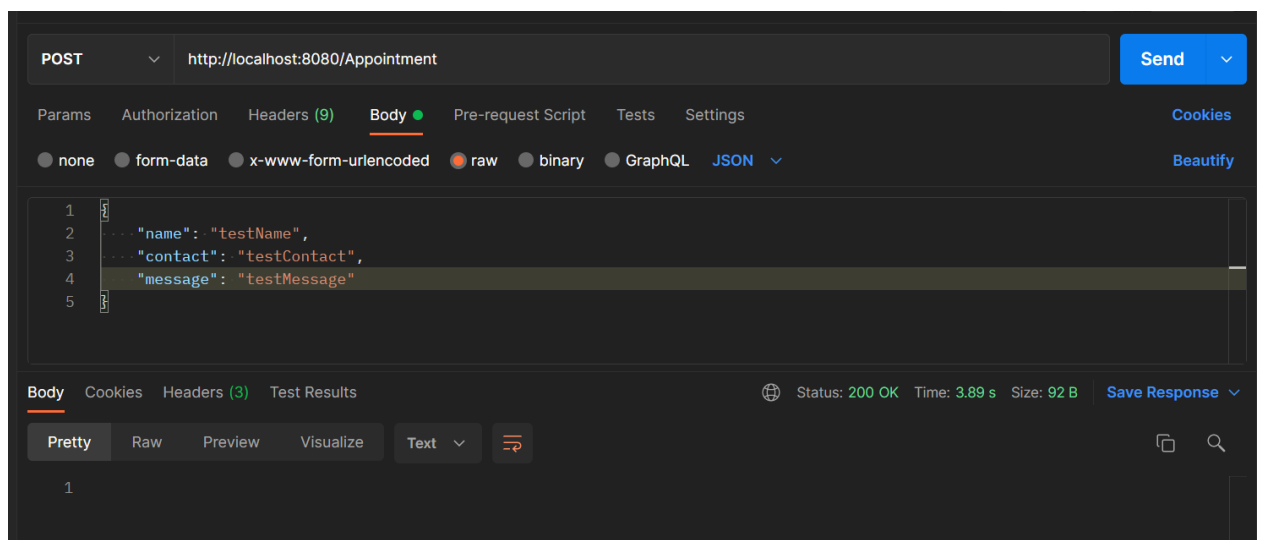
```
public class User  
{  
    public int Id { get; set; }  
    public string Email { get; set; }  
    public string Password { get; set; }  
}
```

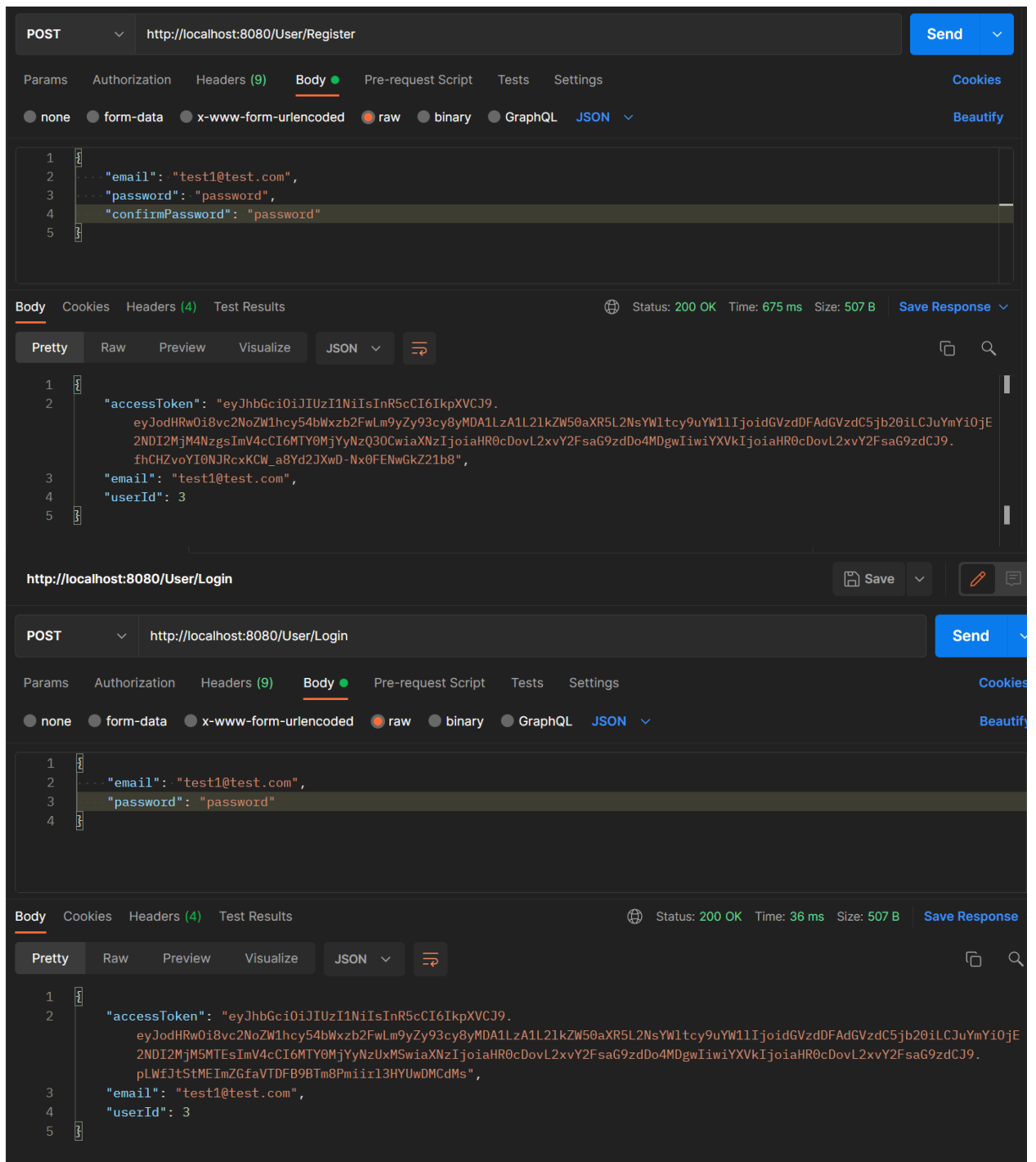
Цель работы: изучить основы разработки API для web-приложений.
Разработать REST API для своего проекта

Задание к лабораторной работе:

1. Изучить структуру формата представления данных JSON.
2. Изучить типы запросов к API: HEAD, GET, POST, PUT, DELETE.
3. Спроектировать и реализовать собственное REST API (Получение, создание, изменение и удаление каких-либо объектов).
4. В отчёт необходимо предоставить документацию к использованию методов. (Либо словесным описанием, либо через Swagger)

Выполнение:





Цель работы: изучить принципы работы отправки Ajax запросов. Изучить работу отправки синхронных и асинхронных запросов в Vue JS. Получить навыки работы с промисами.

Задание к лабораторной работе:

1. Изучить возможности Vue js для отправки http запросов.
2. Выбрать подходящую библиотеку для работы с запросами.
3. Реализовать взаимодействие фронтенда с REST API, спроектированном в прошлой лабораторной работе.

4. Продемонстрировать работу взаимодействия фронтенд приложения с REST API.

Выполнение:



Единая справочная
+7 (4722) 50-48-48

Вход



Регистрация
Адрес:
улица Некрасова, дом 8/9, Белгород, 308007



ОГБУЗ «Белгородская областная
клиническая больница Святителя Иоасафа»

[Главная](#) [О больнице](#) [Контакты](#)

Записаться на приём



Ваши фамилия, имя, отчество

Ваш E-mail или номер телефона

Ваше сообщение

Отправить

Наверх



Единая справочная
+7 (4722) 50-48-48

Вход



Регистрация
Адрес:
улица Некрасова, дом 8/9, Белгород, 308007



ОГБУЗ «Белгородская областная
клиническая больница Святителя Иоасафа»

[Главная](#) [О больнице](#) [Контакты](#)

Регистрация



Логин

Пароль

Повторите пароль

Зарегистрироваться

Наверх



Единая справочная
+7 (4722) 50-48-48

ВходРегистрация



Адрес:
улица Некрасова, дом 8/9, Белгород, 308007



ОГБУЗ «Белгородская областная
клиническая больница Святителя Иоасафа»

[Главная](#) [О больнице](#) [Контакты](#)

Авторизация

Логин

Пароль

Войти

[Наверх](#)