**Министерство образования Московской области**

**ГОУ ВО МО «Государственный гуманитарно-технологический университет» (ГГТУ)**

**Ликино-Дулевский политехнический колледж – филиал ГГТУ**

**ОТЧЁТ**

**ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

по ПМ.02 Разработка модулей программного обеспечения для компьютерных систем

Обучающегося \_\_\_\_\_\_\_\_Шабалкина Данилы Алексеевича \_\_\_\_\_

Ф.И.О.

Курс \_\_\_\_4\_\_\_\_\_ группа \_\_\_ИСП.18.1А\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность 09.02.07 Информационные системы и программирование \_\_\_

Место практики \_\_\_\_\_\_\_\_\_ООО «ЦА Максималист» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Период практики с с 02.03.2022 г. по 29.03.2022 г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководители практики

от колледжа \_Кузьмина Елена Евгеньевна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

  \_\_\_\_\_\_\_\_\_\_Пронина Алла Юрьевна     \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

  \_\_\_\_\_\_\_\_\_\_Селиверстова Ольга Михайловна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

   \_\_\_\_\_\_\_\_\_\_Сабитова Динара Равильевна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

от организации\_\_\_\_\_\_Гась Ян Янович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

МП

г. Ликино-Дулево

2022 г.

Содержание

[**1.** **Общие сведения о предприятии** 3](#_Toc100661083)

[1.1 Структура организации 4](#_Toc100661084)

[**2.** **Анализ материально-технической базы** 5](#_Toc100661085)

[2.1 Состав программного обеспечения 5](#_Toc100661086)

[2.2 Состав технических средств 6](#_Toc100661087)

[2.3 Структура локальной сети предприятия 11](#_Toc100661088)

[**3.** **Выполнение индивидуального задания** 13](#_Toc100661089)

[3.1 Разработка технического задания 13](#_Toc100661090)

[3.2. Проектирование предметной области 15](#_Toc100661091)

[3.3. Проектирование экранных форм 17](#_Toc100661092)

[3.4. Блок-схема приложения 20](#_Toc100661093)

[3.5. Проектирование и разработка баз данных 21](#_Toc100661094)

[3.6. Руководство программиста 21](#_Toc100661095)

[3.7. Руководство пользователя 27](#_Toc100661096)

[3.8. Отладка Сайта 30](#_Toc100661097)

[3.9. Тестирование Сайта 31](#_Toc100661098)

[3.10. Средства защиты сайта 32](#_Toc100661099)

[3.11. Текст программы 33](#_Toc100661100)

[Заключение 44](#_Toc100661101)

[Список литературы 45](#_Toc100661102)

1. **Общие сведения о предприятии**

Студия IS ART специализируется на создании сайтов по всей России и всеми сопровождающимися услугами. За это время компания успела собрать внушительное портфолио работ по созданию сайтов и большой список довольных клиентов. Компания предлагает выполнение работ по разработке сайтов в Москве под ключ, таким образом, клиент получает полностью готовый под его нужды проект с минимальным участием.

Для этих целей, грамотные специалисты, помогают преобразовать пожелания клиента по созданию сайта в грамотное техническое задание, с помощью которого будет создан именно тот сайт, который полностью бы охватывал все потребности.

Сферы деятельности:

* Создание и обслуживание сайтов;
* Разработка интернет-магазинов;
* SEO и трафик;
* Продвижение сайтов;
* Разработка ПО;
* Веб-программирование;
* Интернет ресурсы и сервисы;
* Интернет-агентство.

## 1.1 Структура организации

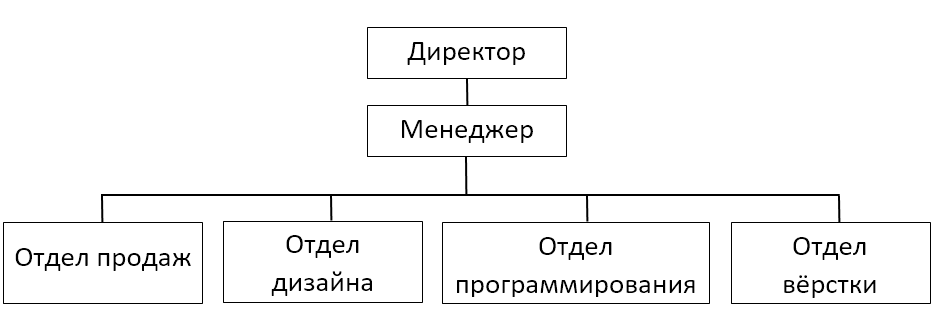


Рис. 1 «Структура организации»

1. **Анализ материально-технической базы**

## 2.1 Состав программного обеспечения

**Microsoft Office** - Офисный пакет приложений, созданных корпорацией Microsoft для операционных систем Microsoft Windows, Windows Phone, Android, macOS, iOS. В состав этого пакета входит программное обеспечение для работы с различными типами документов: текстами, электронными таблицами, базами данных и др. Microsoft Office является сервером OLE-объектов и его функции могут использоваться другими приложениями, а также самими приложениями Microsoft Office. Поддерживает скрипты и макросы, написанные на VBA.

 **Windows 10** — [операционная система](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) для [персональных компьютеров](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%81%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) и [рабочих станций](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B1%D0%BE%D1%87%D0%B0%D1%8F_%D1%81%D1%82%D0%B0%D0%BD%D1%86%D0%B8%D1%8F), разработанная корпорацией [Microsoft](https://ru.wikipedia.org/wiki/Microsoft) в рамках семейства [Windows NT](https://ru.wikipedia.org/wiki/Windows_NT). После [Windows 8.1](https://ru.wikipedia.org/wiki/Windows_8.1) система получила номер 10, минуя 9. Серверный аналог Windows 10 — [Windows Server 2016](https://ru.wikipedia.org/wiki/Windows_Server_2016).

**Kaspersky Endpoint Security** - Антивирусное программное обеспечение, разрабатываемое Лабораторией Касперского. Предоставляет пользователю защиту от вирусов, троянских программ, шпионских программ, руткитов, adware, а также от неизвестных компоненту «файловый антивирус» угроз с помощью проактивной защиты, включающей компонент HIPS. Первоначально, в начале 1990-х, именовался -V, затем - AntiViral Toolkit Pro.

**Microsoft Visual Studio** — линейка продуктов компании [Microsoft](https://ru.wikipedia.org/wiki/Microsoft), включающих [интегрированную среду разработки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как [консольные](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F) [приложения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), так и игры и приложения с [графическим интерфейсом](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F), в том числе с поддержкой технологии [Windows Forms](https://ru.wikipedia.org/wiki/Windows_Forms), а также [веб-сайты](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82), [веб-приложения](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5), [веб-службы](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%BB%D1%83%D0%B6%D0%B1%D0%B0) как в [родном](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4), так и в [управляемом](https://ru.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%8F%D0%B5%D0%BC%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) кодах для всех платформ, поддерживаемых [Windows](https://ru.wikipedia.org/wiki/Windows), [Windows Mobile](https://ru.wikipedia.org/wiki/Windows_Mobile), [Windows CE](https://ru.wikipedia.org/wiki/Windows_CE), [.NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework), [Xbox](https://ru.wikipedia.org/wiki/Xbox), [Windows Phone](https://ru.wikipedia.org/wiki/Windows_Phone) [.NET Compact Framework](https://ru.wikipedia.org/wiki/.NET_Compact_Framework) и [Silverlight](https://ru.wikipedia.org/wiki/Silverlight).

 **Django** — [свободный](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) [фреймворк](https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA) для [веб-приложений](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) на языке [Python](https://ru.wikipedia.org/wiki/Python), использующий шаблон проектирования [MVC](https://ru.wikipedia.org/wiki/Model-View-Controller)[[6]](https://ru.wikipedia.org/wiki/Django#cite_note-6). Проект поддерживается организацией [Django Software Foundation](https://ru.wikipedia.org/wiki/Django_Software_Foundation).

 **Django Rest Framework** (DRF) — это библиотека, которая работает со стандартными моделями Django для создания гибкого и мощного API для проекта.

## 2.2 Состав технических средств

1. Принтер HP LaserJet 1018



Рис. 2 «Принтер»

Таблица 1 «Основные характеристики»

|  |  |
| --- | --- |
| **Основные характеристики** | |
| Размещение: | настольный |
| Устройство: | принтер |
| Тип печати: | черно-белый |
| Технология печати: | лазерный |
| **Печать** | |
| Максимальный формат: | A4 |
| Максимальное разрешение для черно белой печати: | 600x600 dpi |
| Скорость печати: | 12 стр/мин (ч/б А4) |
| Время выхода первого отпечатка: | 10 c (ч/б) |
| **Расходные материалы** | |
| Плотность бумаги: | 60-163 г/м2 |
| Печать на: | пленках, этикетках, глянцевой бумаге, конвертах, матовой бумаге |
| Ресурс ч/б картриджа/тонера: | 2000 страниц |
| Количество картриджей: | 1 |

2. Корпус Zalman T7



Рис. 3 «Корпус»

Таблица 2 «Общие параметры»

|  |  |
| --- | --- |
| **Общие параметры** | |
| Модель | ZALMAN T7 |
| Год релиза | 2019 |
| **Форм-фактор и габариты** | |
| Типоразмер корпуса | Mid-Tower |
| Ориентация материнской платы | вертикально |
| Длина | 384 мм |
| Ширина | 202 мм |
| Высота | 438 мм |
| Вес | 3.2 кг |
| **Совместимость** | |
| Форм-фактор совместимых плат | Micro-ATX, Standard-ATX, Mini-ITX |
| Размещение блока питания | нижнее |
| Форм-фактор совместимых блоков питания | ATX |
| Горизонтальные слоты расширения | 7 |
| Максимальная длина блока питания | 150 мм |
| Вертикальные слоты расширения | нет |
| Количество отсеков 2.5" накопителей | 6 шт |
| Максимальная длина устанавливаемой видеокарты | 305 мм |
| Число внутренних отсеков 3.5" | 2 шт |
| Число внешних отсеков 3.5" | нет |
| Число отсеков 5.25" | нет |
| Максимальная высота процессорного кулера | 160 мм |
| **Охлаждение** | |
| Вентиляторы в комплекте | 2 x 120 мм |
| Поддержка фронтальных вентиляторов | 3 x 120 мм |
| Поддержка тыловых вентиляторов | 1 x 120 мм |
| Поддержка верхних вентиляторов | 2 x 120 мм |
| Поддержка нижних вентиляторов | 2 x 120 мм |
| Поддержка боковых вентиляторов | нет |
| Возможность установки системы жидкостного охлаждения | нет |
| Верхний монтажный размер радиатора СВО | 240 мм |
| **Внешний вид** | |
| Основной цвет | черный |
| Материал корпуса | пластик , сталь |
| Толщина металла | 0.4 мм |
| Наличие окна на боковой стенке | слева |
| Материал окна | оргстекло |
| Материал фронтальной панели | пластик |
| Тип подсветки | нет |
| Цвет подсветки | нет |
| Источник подсветки | нет |
| Разъем подключения подсветки | нет |
| Способ управления подсветкой | нет |
| **Разъемы и интерфейсы лицевой панели** | |
| Расположение I/O панели | сверху |
| Типы USB-портов на панели | USB 3.2 Gen1 Type-A, USB 2.0 Type-A |
| Разъемы | 2 x 3.5 Jack, 1 x USB 3.2 Gen1 Type-A, 2 x USB 2.0 Type-A |
| Встроенный кард-ридер | нет |

3. Сервер Dell PowerEdge R740 (210-AKXJ-167)



Рис. 4 «Сервер»

Таблица 3 «Сервер»

|  |  |
| --- | --- |
| **Общие** | |
| Тип корпуса | 2U |
| Код процессора | Silver 4116 |
| Количество ядер | 12 |
| Установленный объем ОЗУ | 384 (12 х 32) Гб |
| Установлено блоков питания | 2 шт |
| Тип процессора | Intel Xeon |
| Частота процессора | 2100 МГц |
| Установлено процессоров | 1 |
| Количество слотов ОЗУ | 24 шт |
| Мощность блока питания | 750 Вт |

4. Wi-Fi роутер TP-LINK Archer C6



Рис. 5 «Wifi роутер»

Таблица 4 «Роутер»

|  |  |
| --- | --- |
| **Параметры Wi-Fi** | |
| Wi-Fi | Есть |
| Стандарт Wi-Fi | 5 (802.11ac) , 4 (802.11n) |
| Класс Wi-Fi | AC1200 |
| Максимальная скорость по частоте 2.4 ГГц | 300 Мбит/с |
| Максимальная скорость по частоте 5 ГГц | 867 Мбит/с |
| Количество диапазонов 5 ГГц | 1 |
| Одновременная работа в двух диапазонах | Есть |
| Многопотоковая передача данных | MU-MIMO |
| Мощность передатчика | 20 dBm |
| Тип и количество антенн | внешняя несъемная x4, внутренняя x1 |
| Коэффициент усиления антенны | 5 dBi |
| Безопасность соединения | WPA-PSK/WPA2-PSK , WPA/WPA2-Enterprise , WEP |
| **Порты** | |
| Количество LAN портов | Количество LAN портов |
| 4 | 4 |
| Скорость передачи по проводному подключению | Скорость передачи по проводному подключению |
| 1000 Мбит/с | 1000 Мбит/с |
| Количество SFP портов | Количество SFP портов |

## 2.3 Структура локальной сети предприятия

На предприятии: «IS ART» данные передаются, в основном, с помощью интернета. Локальной сетью объединены только 2 компьютера.

Топология общая шина предполагает использование одного кабеля, к которому подключаются все компьютеры сети. Отправляемое какой-либо рабочей станцией сообщение распространяется на все компьютеры сети. Каждая машина проверяет кому адресовано сообщение, — если сообщение адресовано ей, то обрабатывает его.

Шина самой своей структурой допускает идентичность сетевого оборудования компьютеров, а также равноправие всех абонентов. При таком соединении компьютеры могут передавать информацию только по очереди, — последовательно — потому что линия связи единственная. В противном случае пакеты передаваемой информации будут искажаться в результате взаимного наложения (т. е. произойдет конфликт, коллизия). Таким образом, в шине реализуется режим полудуплексного (half duplex) обмена (в обоих направлениях, но по очереди, а не одновременно (т. е. последовательно а не параллельно)).

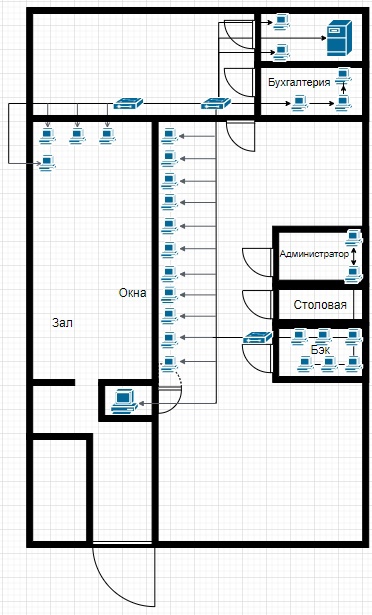


Рис. 6 «Локальная схема»

1. **Выполнение индивидуального задания**

## 3.1 Разработка технического задания

**3.1.1 Назначение разработки**

Форум Diploma позволяет программистам легко задавать вопросы и отвечать на вопросы других пользователей всем зарегистрированным пользователям.

**3.1.2** **Требования к функциональным характеристикам**

Сайт должен выполнять следующие задачи:

* + Позволять регистрировать аккаунты
  + Позволять входить в сайт уже зарегистрированным пользователям
  + Задавать вопросы
  + Отвечать на вопросы других пользователей

**Ввод данных:**

* + Логин
  + Электронная почта
  + Дата рождения
  + Данные о себе
  + Пароль
  + Название вопроса
  + Текст вопроса
  + Ссылка на фото для профиля

**Вывод данных:**

* + Название вопроса
  + Текст вопроса
  + Никнейм пользователя
  + Фото профиля пользователя

**3.1.3 Требования к надежности и безопасности**

Сайт обязательно должен иметь SSL-сертификат. Иметь надежный хостинг, который так же предоставляет SSL-сертификат, достаточно мощностей для большого потока посетителей.

**3.1.4 Требования к составу и параметрам технических средств**

Достаточно иметь любое современное устройство или ПК с минимальными характеристиками. Обязательное наличие сети.

**Конфигурация Компьютера**

Параметры - Wi-Fi

Процессор - Intel Pentium 4

Оперативная Память - 2 Гб

Жесткий Диск - 250 gb или меньше

**3.1.5** **Требования к информационной и программной совместимости**

Для доступа к сайту необходим ПК с операционной системой Windows, Linux или Mac OS либо гаджет с операционной системой Android или IOS. Так же необходим доступ к интернету и любой браузер.

**3.1.6 Требования к программной документации**

В ходе разработки программы должны быть подготовлены следующие программные документы: текст программы, описание программы, программа и методика испытаний, руководство пользователя, руководство программиста, технико-экономическое обоснование.

## 3.2. Проектирование предметной области

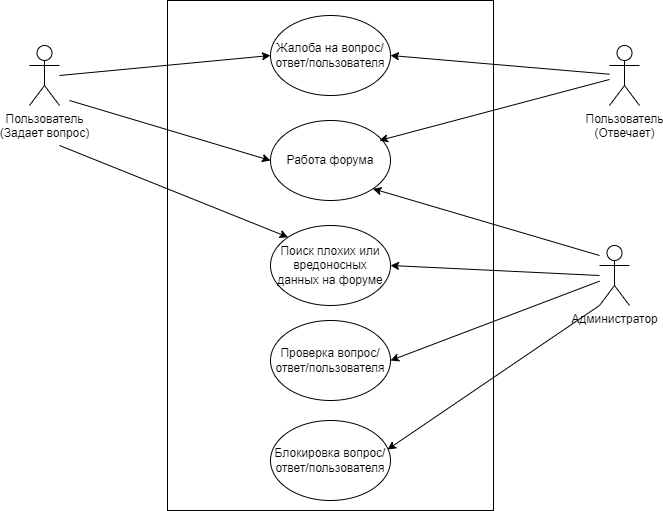
**3.2.1 Диаграмма прецедентов (таблица бизнес-процессов)**

Рис. 7 «Таблица бизнес процессов»

**3.2.2. Моделирование бизнес-процессов**

Таблица №6 «Определение бизнес-процессов»

|  |  |
| --- | --- |
| Номер бизнес-процесса | Название бизнес процесса |
| 1-Обр\_Жалоб | Отправка и обработка жалоб |
| 2-Раб\_Форума | Работа форума |
| 3-Поиск\_данных | Поиск плохих или вредоносных данных |
| 4-Пров\_ВО | Проверка вопросов и ответов пользователей |
| 5-Блок\_польз | Блокировка пользователя |

**3.2.3. Словесный алгоритм бизнес-процесса**

Все зарегистрированные пользователи форума могут посещать его с целью задания и ответа на вопросы.

Пользователь приходит, авторизуется на форуме и оставляет вопрос по своей проблеме

Затем этот пользователь ожидает ответов на его вопрос

Другие пользователи просматривают вопрос, и, если у них есть возможные решения, пишут их.

Если создатель вопроса видит решение своей проблемы, он может отметить ответ как решение, тем самым закрыв вопрос.

**3.2.4. Диаграмма действий бизнес-процесса**

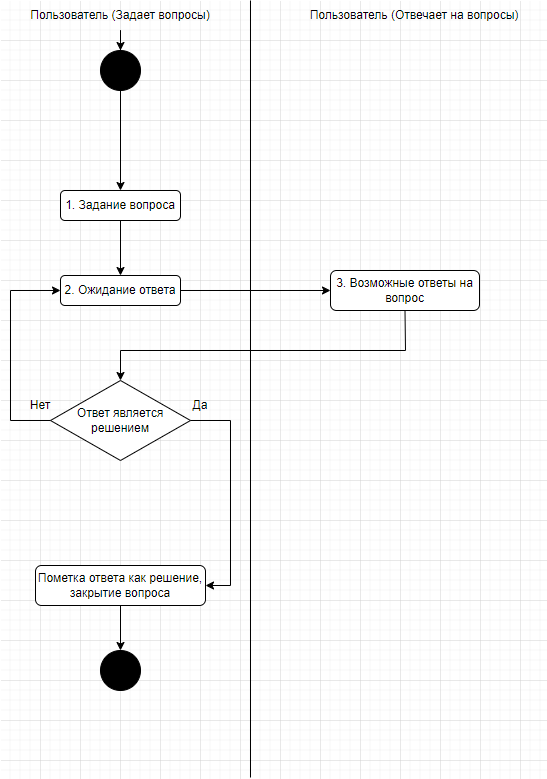


Рис. 8 «диаграмма бизнес-процессов»

**3.2.5. Таблица операций**

Таблица №7 «Таблица операций»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2-Раб\_форума | Операция | Исполнитель | Как часто | Входящие документы | Исходящий документ |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 2-Раб\_Форума1 | 1. Задание вопроса | Пользователь (задающий вопрос) | По мере необходимости | Текст вопроса | Нет |
| 2-Раб\_форума2 | 2. Ожидание ответа на вопрос | Пользователь (задающий вопрос) | По мере необходимости | Нет | Нет |
| 2-Раб\_Форума3 | 3. Ответ на вопрос | Пользователь (отвечающий на вопрос) | По мере необходимости | Текст вопроса | Текст ответа |
| 2-Раб\_Форума4 | 4. Пометка ответа как решения | Пользователь (задающий вопрос) | По мере необходимости | Текст ответа | Решение вопроса |

## 3.3. Проектирование экранных форм

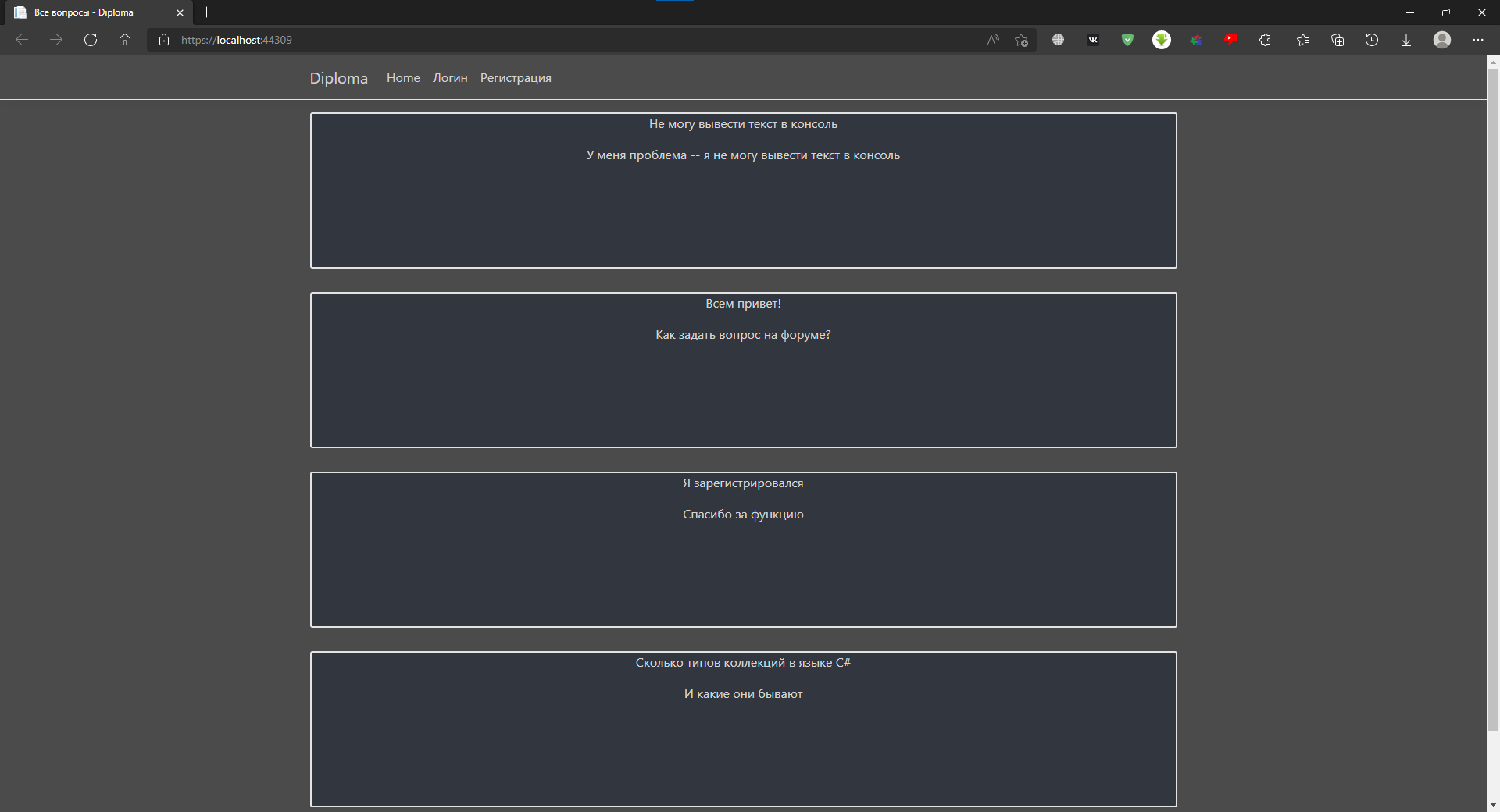


Рис. 8 «страница №1»

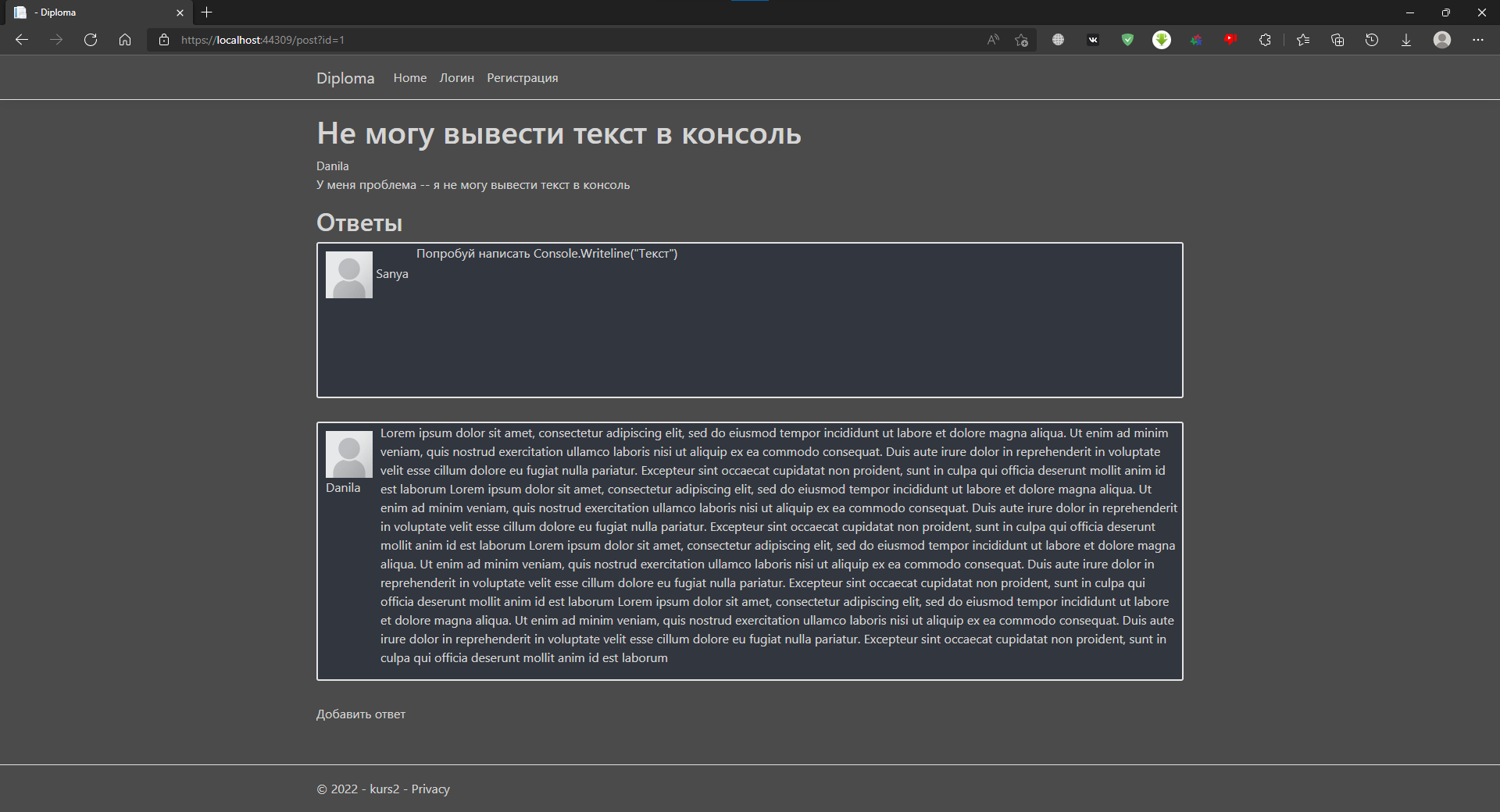


Рис. 9 «страница №2»

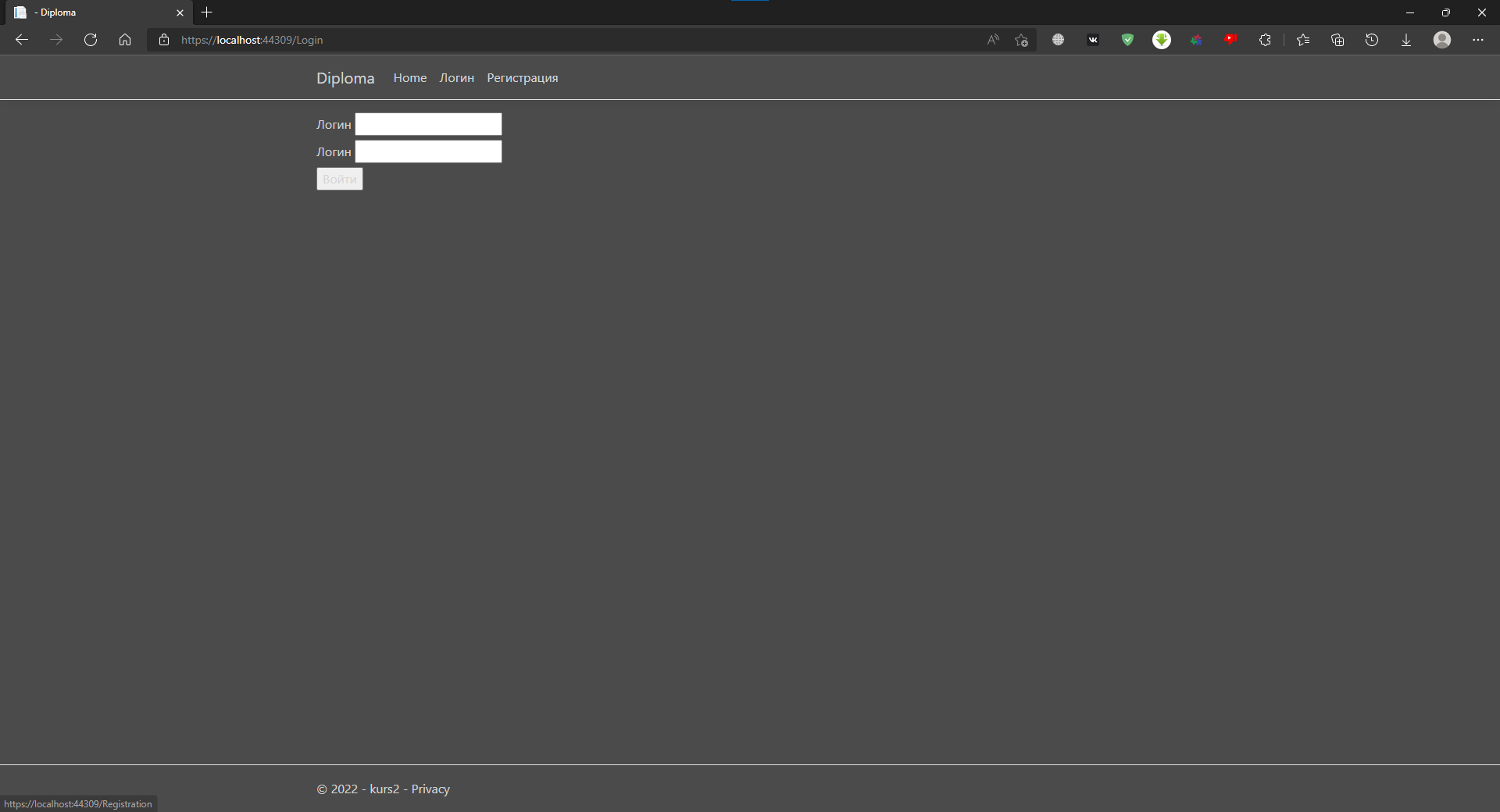


Рис. 10 «страница №3»

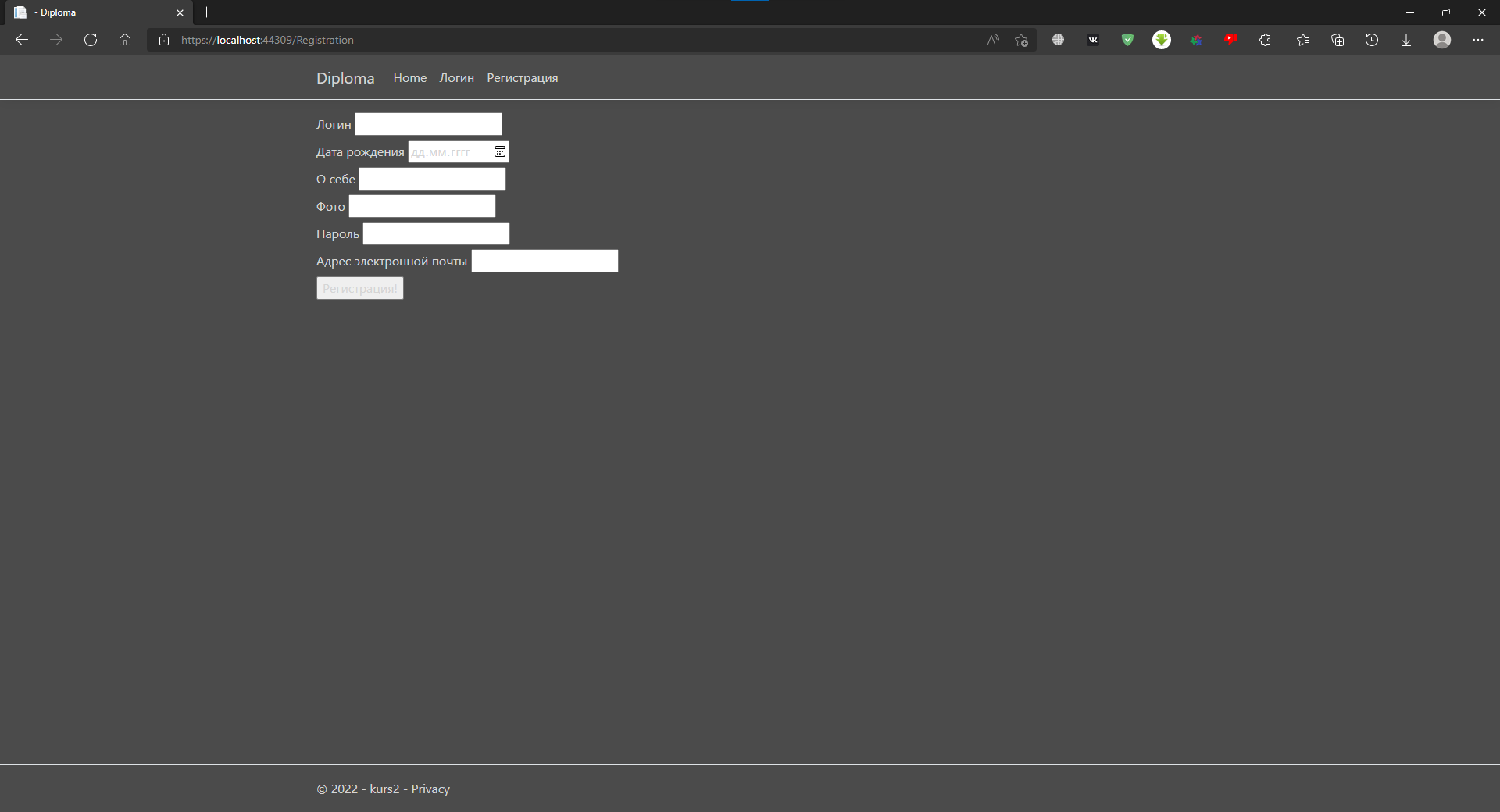


Рис. 11 «страница №4»

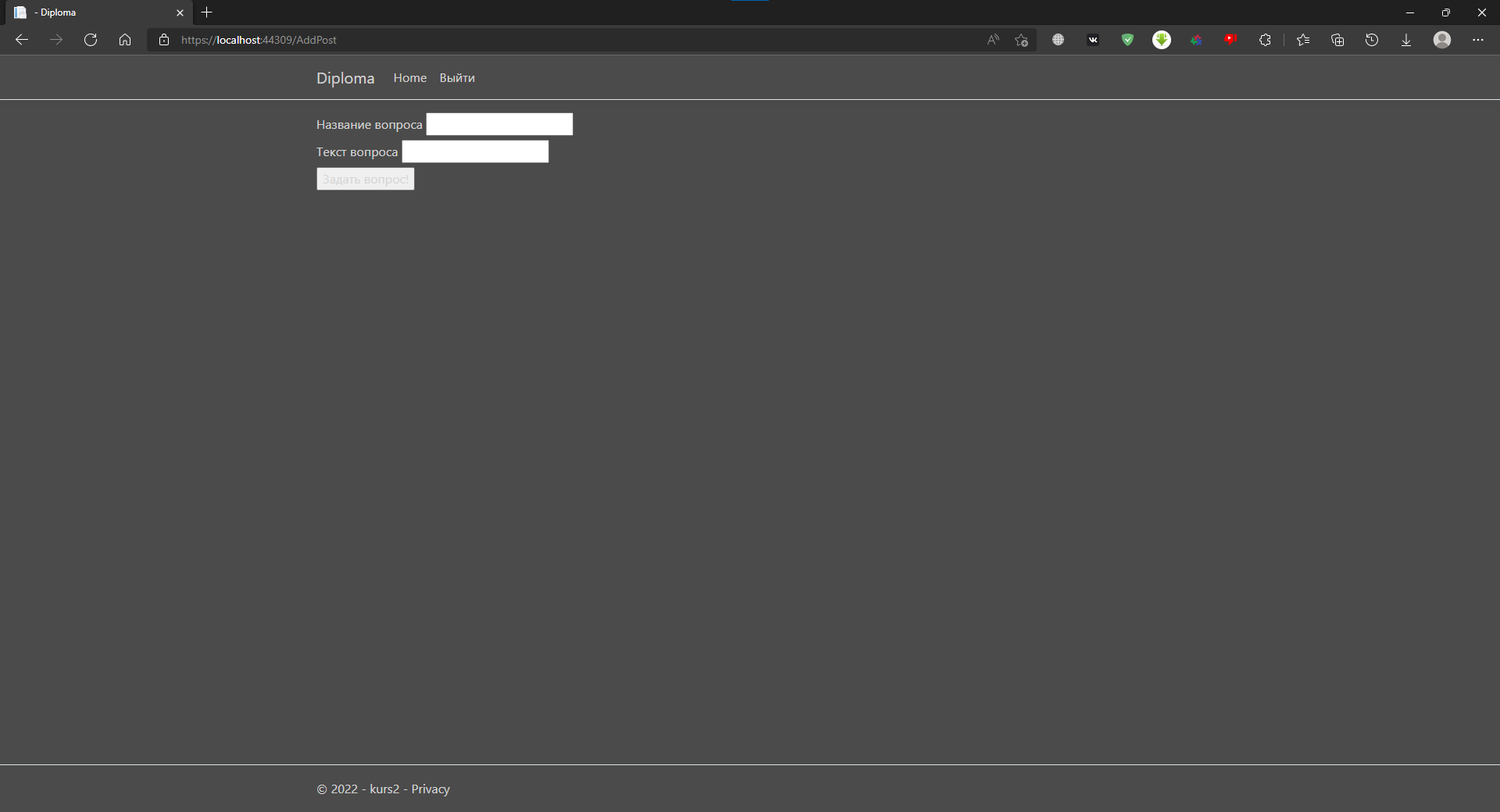


Рис. 12 «страница №5»

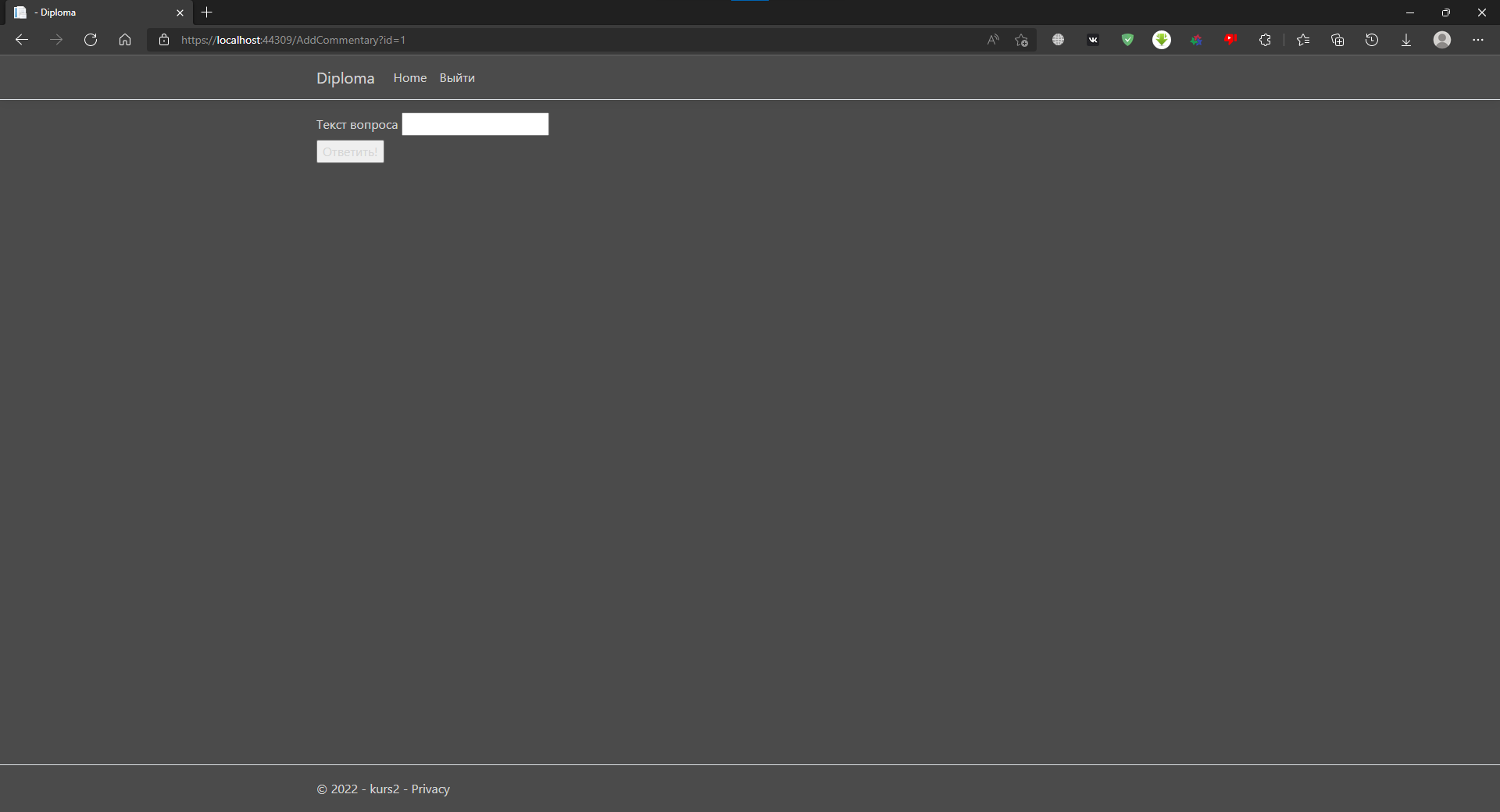


Рис. 13 «страница №6»

## 3.4. Блок-схема приложения

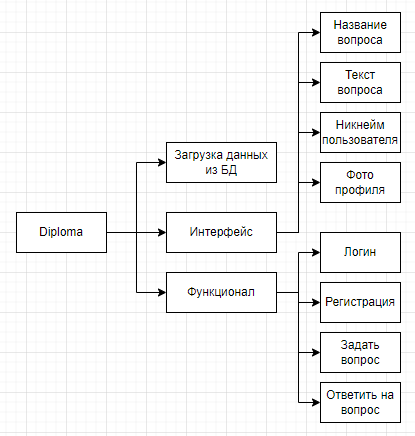
****

Рис. 15 «Блок схема»

## 3.5. Проектирование и разработка баз данных

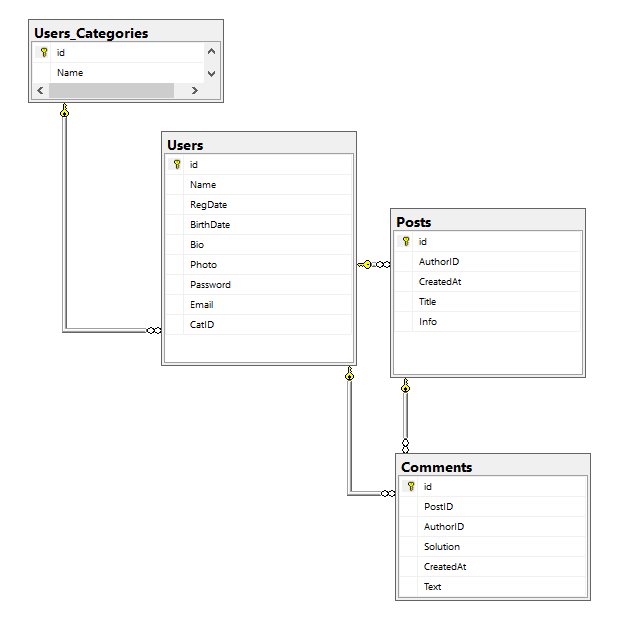


Рис. 17 «Структура БД»

## 3.6. Руководство программиста

**Требования к составу и параметрам технических средств:**

Оптимальные системные требования

Параметры - Wi-Fi

Процессор - Intel Pentium 4

Оперативная Память - 2 Гб

Жесткий Диск - 250 gb или меньше

**Требования к информационной и программной совместимости:**

Необходимы операционная система Windows, Linux или Mac OS. Любой текстовый редактор и браузер.

**Язык программирования, разметки и каскадных стилей: я C#, JavaScript, HTML, CSS.**

C# — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework и .NET Core.

JavaScript - мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript.

Html - стандартизированный язык разметки документов для просмотра веб-страниц в браузере. Веб-браузеры получают HTML документ от сервера по протоколам HTTP/HTTPS или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора.

CSS - формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Также может применяться к любым XML-документам, например, к SVG или XUL.

**Установка окружения для работы**

Нужно установить текстовый редактор для работы с кодом.

**Подключение к базе данных**

Для работы с данными необходимо настроить подключение к базе данных. Для этого была создана модель с таблицами базы данных, строка подключения и контекст, благодаря которому можно обратиться напрямую к базе в коде:

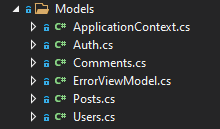


Рис. 4 «Модель данных»

Пример кода модели данных:

using System;

namespace kurs2.Models

{

public class Users

{

public int id { get; set; }

public string Name { get; set; }

public DateTime RegDate { get; set; }

public DateTime BirthDate { get; set; }

public string Bio { get; set; }

public string Photo { get; set; }

public string Password { get; set; }

public string Email { get; set; }

public int CatID { get; set; }

}

}

Создание контекста базы данных:

using Microsoft.EntityFrameworkCore;

using kurs2.Models;

namespace kurs2.Models

{

public class ApplicationContext : DbContext

{

public DbSet<Users> Users { get; set; }

public DbSet<Posts> Posts { get; set; }

public DbSet<Comments> Comments { get; set; }

public ApplicationContext(DbContextOptions<ApplicationContext> options)

: base(options)

{

Database.EnsureCreated();

}

}

}

Метод, инициализирующий контекст при старте сервера:

public void ConfigureServices(IServiceCollection services)

{

string connection = Configuration.GetConnectionString("DefaultConnection");

// добавляем контекст MobileContext в качестве сервиса в приложение

services.AddDbContext<ApplicationContext>(options =>

options.UseSqlServer(connection));

services.AddControllersWithViews();

}

Строка подключения в конфигурации приложения:

"ConnectionStrings": {

"DefaultConnection": "Server=.\\SQLEXPRESS;Database=KursForum;Trusted\_Connection=True;"

},

**Структура Базы Данных**

Диаграмма базы данных - включает в себя описания содержания, структуры и ограничений целостности, используемые для создания и поддержки базы данных. Для работы был использован Microsoft SSMS.

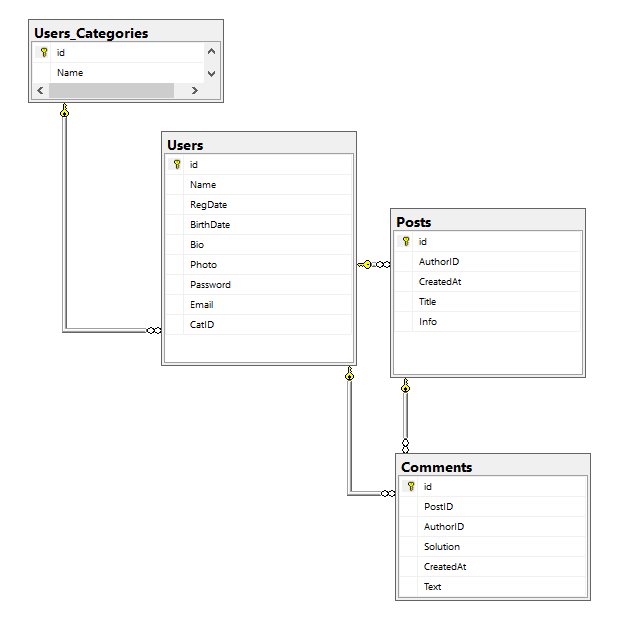


Рис. 20 «Схема БД»

**Шаги создания программы:**

**Создание проекта.**

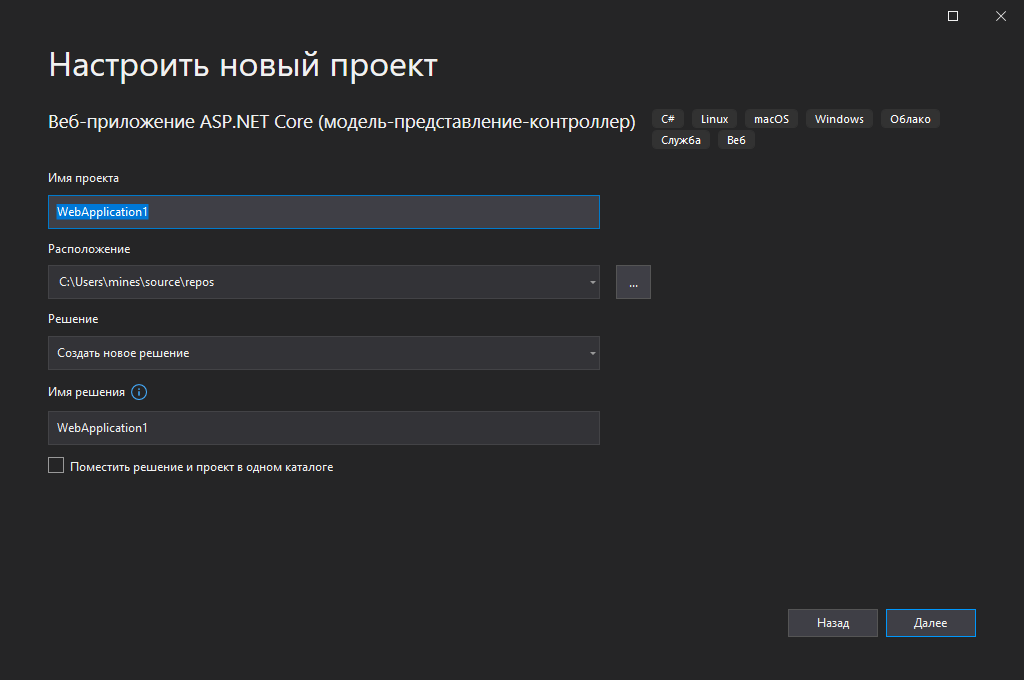


Рис. 21 «Создание проекта»

**Структура проекта**

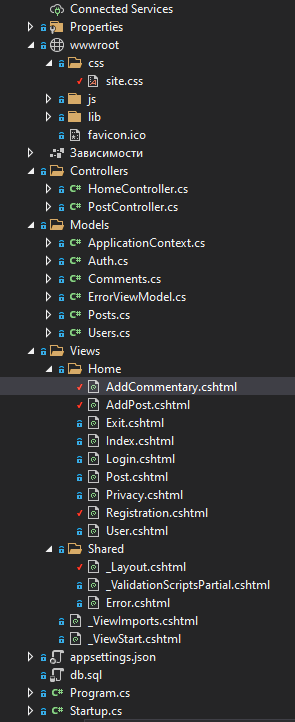


Рис. 22 «Структура проекта»

**Файл запуска сайта:**

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.HttpsPolicy;

using Microsoft.Extensions.Configuration;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Hosting;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.EntityFrameworkCore;

using kurs2.Models;

namespace kurs2

{

public class Startup

{

public Startup(IConfiguration configuration)

{

Configuration = configuration;

}

public IConfiguration Configuration { get; }

// This method gets called by the runtime. Use this method to add services to the container.

public void ConfigureServices(IServiceCollection services)

{

string connection = Configuration.GetConnectionString("DefaultConnection");

// добавляем контекст MobileContext в качестве сервиса в приложение

services.AddDbContext<ApplicationContext>(options =>

options.UseSqlServer(connection));

services.AddControllersWithViews();

}

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

else

{

app.UseExceptionHandler("/Home/Error");

// The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}");

endpoints.MapControllerRoute(

name: "post",

pattern: "{controller=Home}/{action=Post}/{id?}");

endpoints.MapControllerRoute(

name: "user",

pattern: "{controller=Home}/{action=User}/{id?}");

endpoints.MapControllerRoute(

name: "login",

pattern: "{controller=Home}/{action=Login}");

endpoints.MapControllerRoute(

name: "exit,",

pattern: "{controller=Home}/{action=Exit}");

endpoints.MapControllerRoute(

name: "addpost",

pattern: "{controller=Home}/{action=AddPost}");

endpoints.MapControllerRoute(

name: "registration",

pattern: "{controller=Home}/{action=Registration}");

endpoints.MapControllerRoute(

name: "addcommentary",

pattern: "{controller=Home}/{action=AddCommentary}/{id?}");

});

}

}

}

## 3.7. Руководство пользователя

При переходе на сайт открывается главная страница:

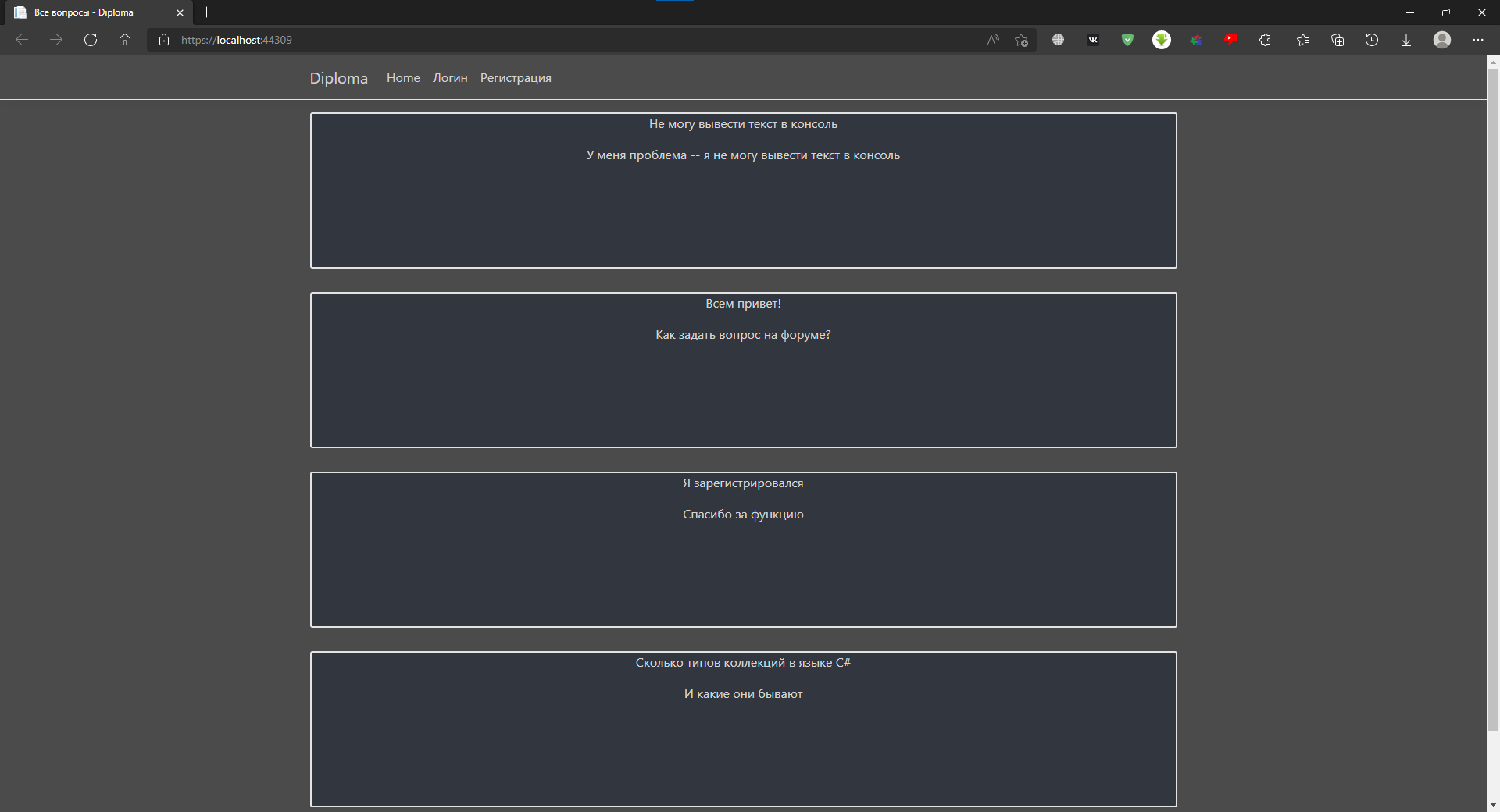


Рис 9 «Главная страница сайта»

Для просмотра имеющихся тем, достаточно нажать на одну из представленных на главной странице

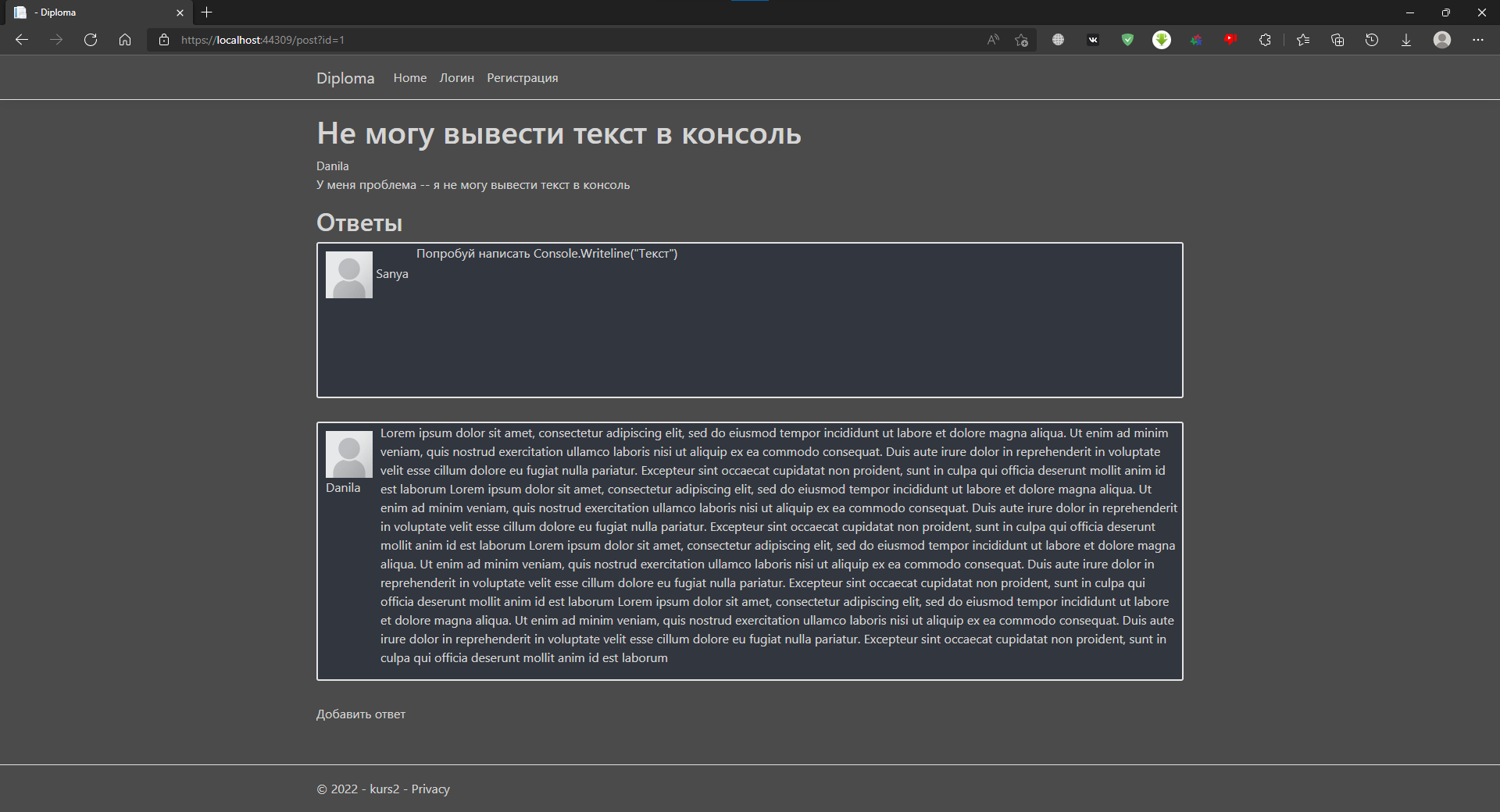


Рис 10 «Страница с вопросом»

Для того чтобы задать новый вопрос или ответить на текущий, необходимо зарегистрироваться или войти в существующий аккаунт

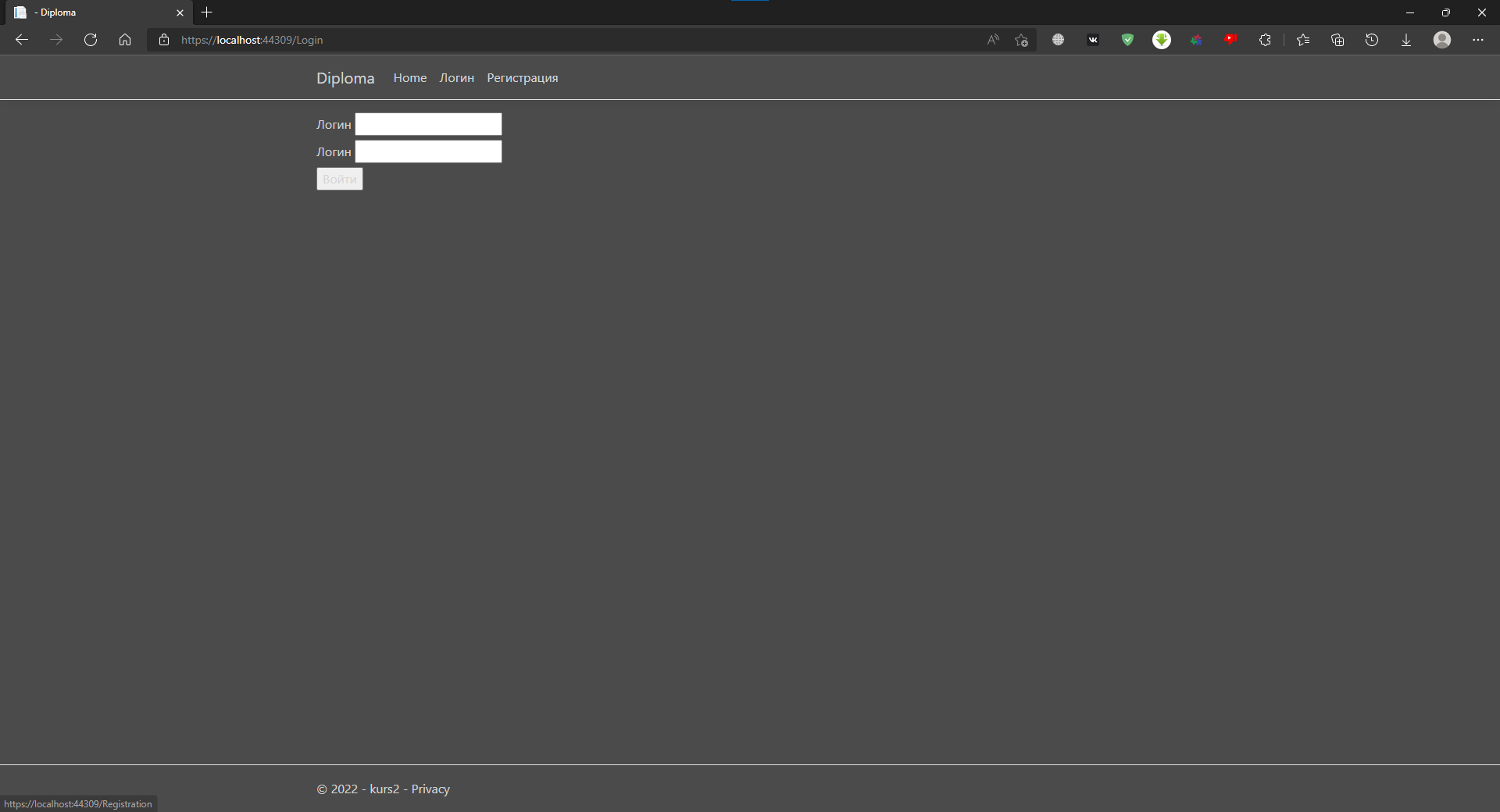


Рис 12 «Форма входа»

После ввода данных и нажатия кнопки «Войти», видно, что шапка сайта изменилась, и теперь вместо кнопок логина или регистрации, там расположена выхода из аккаунта

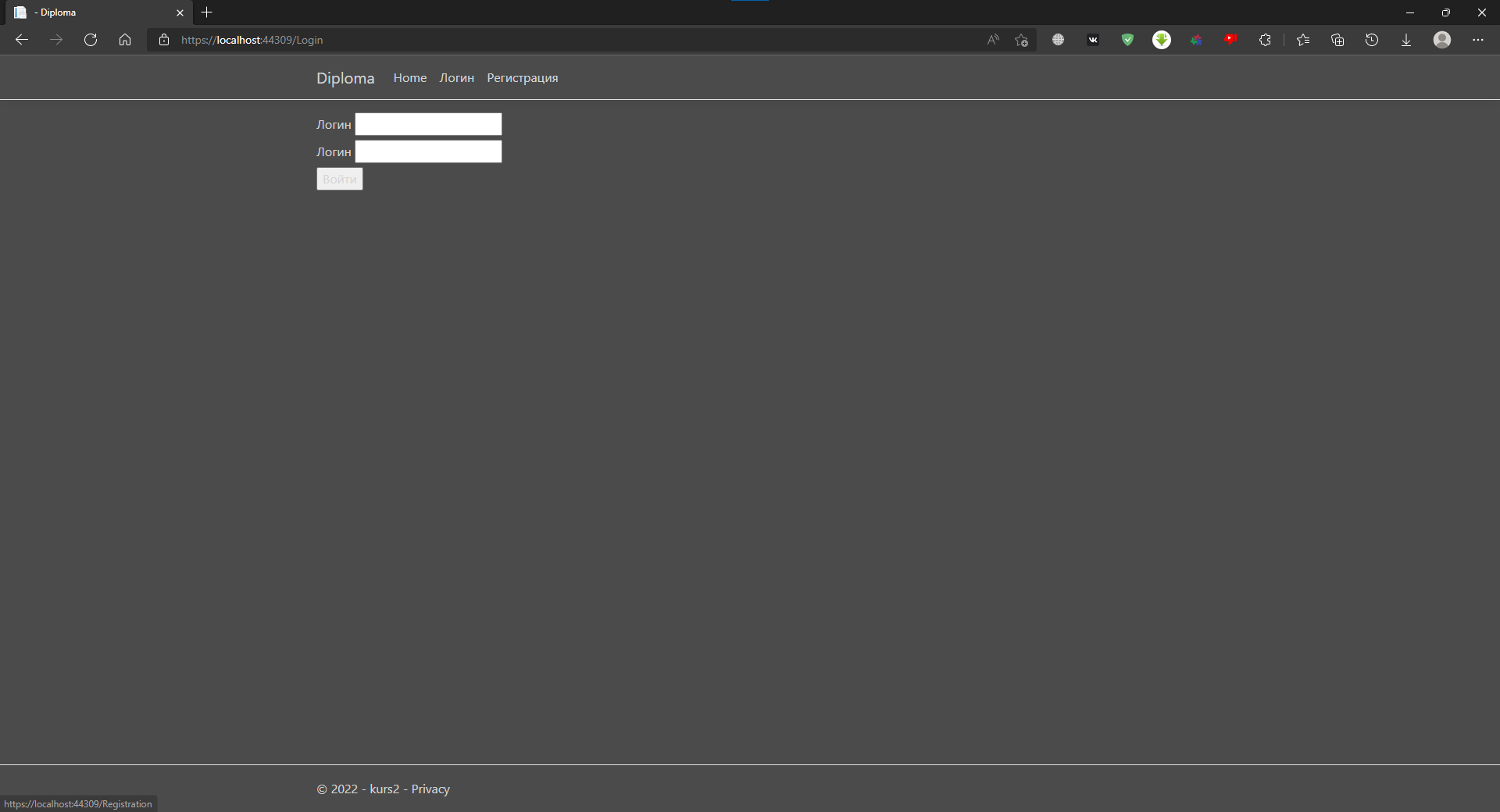


Рис 13 «Шапка сайта»

Для добавления вопроса, после входа в аккаунт, нужно нажать кнопку «Добавить вопрос» на главной

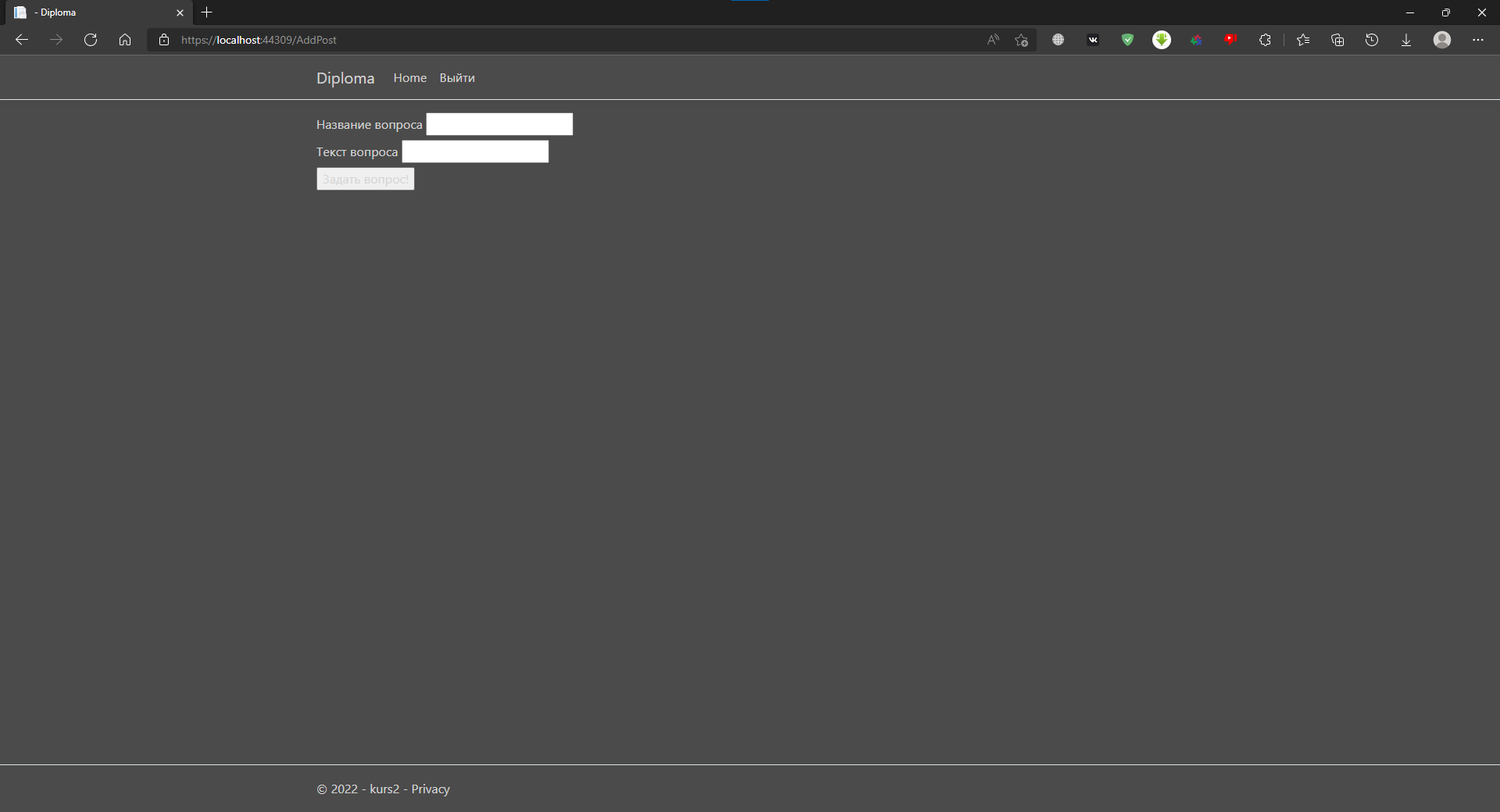


Рис 14 «Форма ввода темы с вопросом»

Откроется форма ввода, в которой необходимо ввести тему и содержание вопроса.

После ввода вопроса и нажатие на кнопку «Задать вопрос!», сайт перекинет пользователя на страницу с этим же вопросом

Чтобы добавить ответ, нужно нажать на кнопку «Добавить ответ», находящуюся снизу

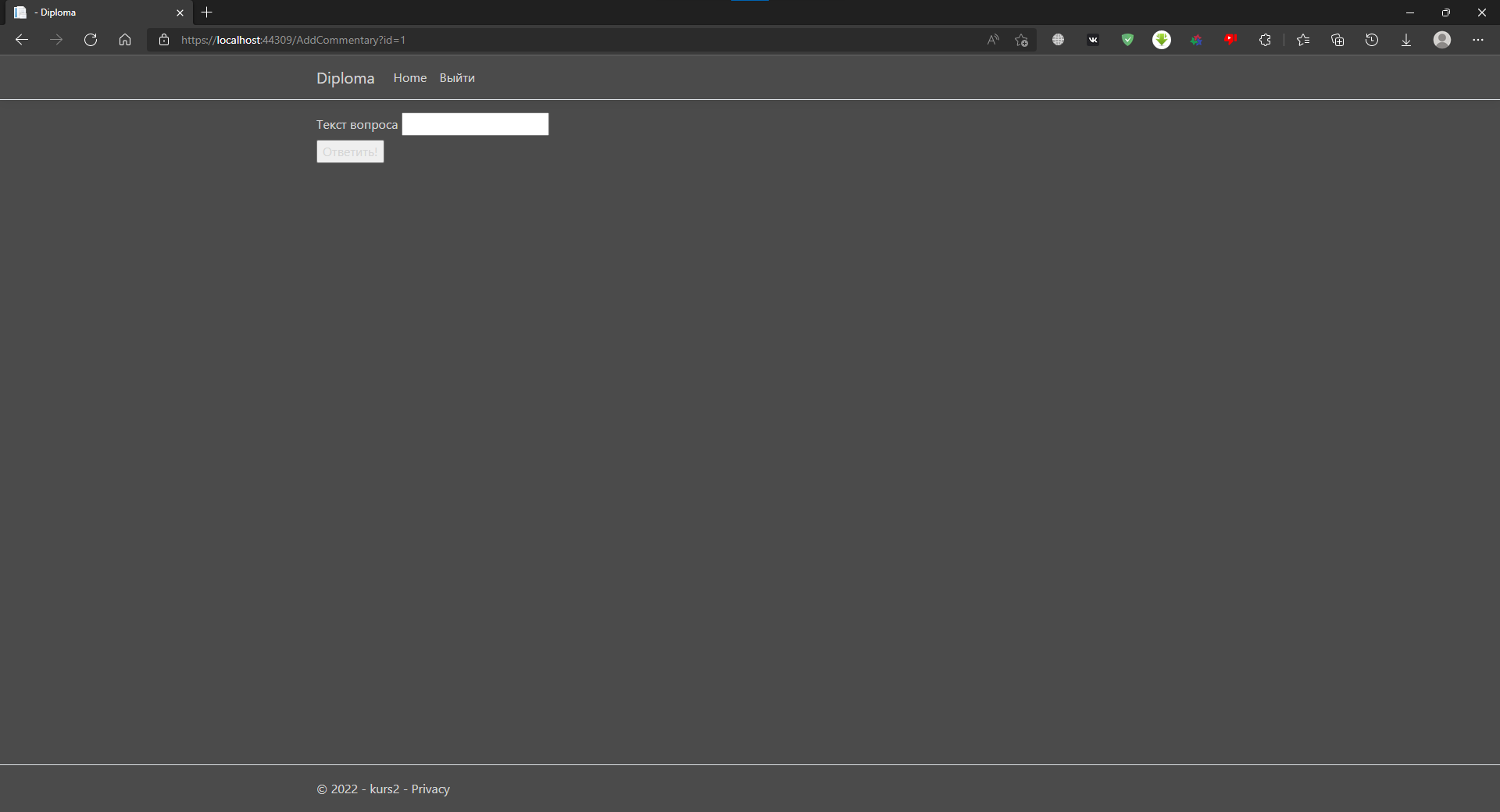


Рис 16 «Форма ввода ответа»

Откроется форма ввода с одним полем. В него необходимо ввести ответ, после чего нажать на кнопку для отправки ответа.

После этого, пользователя перекинет назад на страницу с вопросом, где уже выведен его ответ

## 3.8. Отладка Сайта

Таблица №8 «Отладка сайта»

|  |  |  |  |
| --- | --- | --- | --- |
| Название Ошибки | Скрин ошибки | Метод исправления | Результат |
| Не отображаются вопросы на главной |  | Добавить цикл для отображения вопросов |  |
| Черный текст на темном фоне |  | Исправить стиль для текста |  |
| При попытке добавить ответ, пользователь перенаправляется назад на страницу с вопросом |  | Исправить ошибку на странице с формой ответа |  |

## 3.9. Тестирование Сайта

Тестирование сайта – это последний и обязательный этап технической разработки сайта. Он играет важнейшую роль в процессе создания ресурса, так как именно от качества тестирования зависит дальнейшая жизнь ресурса. К сожалению, очень часто разработчики не уделяют должного внимания этому этапу, полагаясь на свой опыт. В результате существующие ошибки приводят к колоссальным затратам времени и денег.

Ведь ресурс, который имеет ошибки, вызывает негатив у посетителей и, как следствие, их потерю. В итоге владелец ресурса вынужден платить за доработку (а иногда за повторную разработку ресурса), а сотрудничество с бывшими разработчиками некачественного сайта оставляет только неприятный осадок.

**Цели и основные методы тестирования веб-сайта**

Целью тестирования является общая проверка реального функционирования веб-сайта на соответствие предъявленным требованиям. Весь этап представляет собой кропотливый труд специалистов, которые для выявления ошибок создают искусственные ситуации, которые могут возникнуть в период работы ресурса и анализируют «поведение» ресурса на предложенных условиях. После выявления багов (ошибок), тестировщик составляет отчет и передает его project-менеджеру, который распределяет работу по их устранению среди участников проекта. После доработки ресурс повторно тестируется. Такой цикл работ повторяется до тех пор, пока веб-сайт не достигнет поставленных результатов в проверке.

**Тестирование сайтов в разных браузерах**

Тестирование сайта на ошибки выполняется разными методами в разных браузерах. Чтобы добиться безупречного функционирования сайта, специалисты-тестировщики создают план действий, в который входит:

Тестирование функциональности;

Проверка usability (удобства пользования);

Тест производительности;

Проверка на безопасность;

UI Testing (тестинг интерфейса).

## 3.10. Средства защиты сайта

Можно выделить несколько основных способов защитить свой сайт:

* + обеспечить защиту от DDoS-атак;
  + подключить SSL-сертификат;
  + использовать надёжный хостинг;
  + использовать безопасные плагины/библиотеки/фреймворки/CMS (далее – «сторонние модули»);
  + применять существующие техники защиты от SQL-инъекций и XSS-атак;
  + обеспечить ведение журнала веб-сайта и мониторинг событий безопасности;
  + производить регулярное резервное копирование веб-сайта и всех важных данных;
  + использовать надёжные и сложные пароли, а также защиту от перебора паролей;
  + в случае наличия административной панели, с помощью которой происходит управление содержимым веб-сайта, необходимо изменить стандартный адрес входа и обеспечить контроль доступа.

**Основные правила защиты БД**

Существует два проверенных метода защиты:

**1. Парольная защита**. Самый простой, но в то же время самый ненадежный метод, особенно если использовать пароль без шифровки, так как вычислительная система не видит другого пользователя с идентичными паролями. Также недостатком является необходимость запоминания пароля либо фиксирования на отдельном носителе (листок бумаги или телефон), что не совсем удобно, так как простые для запоминания комбинации злоумышленники легко вычисляют, а записанные на шпаргалке могут быть преданы разглашению. Целесообразнее устанавливать для входа в систему несколько паролей, каждый из которых дает полномочия различной степени.

**2. Идентификация пользователя**. Довольно серьезный и надежный подход, при котором попытка получения разрешения сопровождается немедленным процессом аутентификации, то есть исключения подмены входящего.

Для идентификации используют:

* + запрос пароля;
  + контрольный вопрос, затрагивающий сугубо личные сведения пользователя;

2. Защита выбора данных при добавлении информации. Чтобы пользователь случайно не ошибся при вводе данных, есть выбор из готовых данных

3. Выбор даты. Чтобы пользователь случайно не ввел неправильный формат даты, осуществлен выбор даты.

## 3.11. Текст программы

Стили

/\* Please see documentation at https://docs.microsoft.com/aspnet/core/client-side/bundling-and-minification

for details on configuring this project to bundle and minify static web assets. \*/

a.navbar-brand {

white-space: normal;

text-align: center;

word-break: break-all;

}

/\* Provide sufficient contrast against white background \*/

a {

color: #d5d5d5 !important;

border: 0;

}

a:hover {

border: 0;

color: #d5d5d5 !important;

}

.btn-primary {

color: #d5d5d5 !important;

background-color: #1b6ec2;

border-color: #1861ac;

}

.nav-pills .nav-link.active, .nav-pills .show > .nav-link {

color: #d5d5d5 !important;

background-color: #1b6ec2;

border-color: #1861ac;

}

/\* Sticky footer styles

-------------------------------------------------- \*/

html {

font-size: 14px;

}

@media (min-width: 768px) {

html {

font-size: 16px;

}

}

.border-top {

border-top: 1px solid #e5e5e5;

}

.border-bottom {

border-bottom: 1px solid #e5e5e5;

}

.box-shadow {

box-shadow: 0 .25rem .75rem rgba(0, 0, 0, .05);

}

button.accept-policy {

font-size: 1rem;

line-height: inherit;

}

/\* Sticky footer styles

-------------------------------------------------- \*/

html {

position: relative;

min-height: 100%;

}

body {

/\* Margin bottom by footer height \*/

margin-bottom: 60px;

background-color: #4b4b4b;

}

.text-dark {

color: #d5d5d5 !important;

}

.footer {

position: absolute;

bottom: 0;

width: 100%;

white-space: nowrap;

line-height: 60px; /\* Vertically center the text there \*/

}

.post-pre {

border: 2px solid #e4e4e4;

border-radius: 3px;

min-height: 200px;

background-color: #32363e;

margin-bottom: 30px;

}

.post-info{

white-space: normal;

}

.\_comm{

display:flex;

}

.comment-usr-info{

margin: 10px;

}

.navbar {

background-color: #4b4b4b !important;

}

\* {

color: #d5d5d5 !important;

}

Html

@{

ViewData["Title"] = "Home Page";

@model dynamic

}

<div class="text-center">

@{

if (Context.Request.Cookies["auth"] != null)

{

<a **asp-action**="AddPost">Добавить вопрос</a>

}

}

@{

ViewBag.Title = "Все вопросы";

}

@foreach (var item in Model.Posts)

{

<a href="post?id=@item.id">

<div class="post-pre">

<div class="post-name">

<p class="pos-name-p">@item.Title</p>

<p></p>

</div>

<div>

<p class="post-info">@item.Info</p>

</div>

</div>

</a>

}

</div>

@model dynamic

@{

int Id = ViewBag.id;

}

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>foisdfj</title>

</head>

<body>

@{

var post = new Posts();

List<Comments> comms = new List<Comments>();

foreach(var a in Model.Comments)

{

comms.Add(a);

}

foreach (var a in Model.Posts)

{

if (a.id == Id)

{

post = a;

}

}

}

<h1>@post.Title</h1>

@{

var user = new Users();

foreach (var a in Model.Users)

{

if (a.id == post.AuthorID)

{

user = a;

}

}

}

<a href="user?id=@user.id" style="color: #808080; font-size: 15px">@user.Name</a>

<p>@post.Info</p>

<h2>Ответы</h2>

<div>

@\* @{

Dictionary<string, int> usrsDict = new Dictionary<string, int>();

foreach(var u in Model.Users)

{

usrsDict.Add(u.Name, u.id);

}

SelectList usrs = new SelectList(usrsDict);

}

<form method="get">

<label>Сортировать по пользователям</label>

<select asp-items="usrs"></select>

<input type="submit" />

</form>\*@

@{

foreach (var a in Model.Comments)

{

if (a.PostID == post.id)

{

<div class="post-pre \_comm">

<div class="comment-usr-info">

<img src="https://okeygeek.ru/wp-content/uploads/2020/03/no\_avatar.png" alt="" height="60" />

@foreach (var b in Model.Users)

{

if (b.id == a.AuthorID)

{

<a href="user?id=@a.AuthorID">@b.Name</a>

}

}

</div>

<p>@a.Text</p>

</div>

}

}

}

</div>

<a href="AddCommentary?id=@post.id">Добавить ответ</a>

</body>

</html>

@model kurs2.Models.Comments

<form **asp-action**="AddCommentary" **asp-controller**="home">

<input **type**="hidden" **value**="@Context.Request.Cookies["auth"]" **asp-for**="AuthorID" />

<input **type**="hidden" **value**="@ViewBag.id" **asp-for**="PostID" />

<input **type**="hidden" **value**="@false" **asp-for**="Solution" />

<input **type**="hidden" **value**="@DateTime.Now" **asp-for**="CreatedAt" />

<div>

<label **asp-for**="Text">Текст вопроса</label>

<input **type**="text" **asp-for**="Text" />

</div>

<input type="submit" value="Ответить!" />

</form>

@model kurs2.Models.Posts

<form **asp-action**="AddPost" **asp-controller**="Home">

<div>

<label **asp-for**="Title">Название вопроса</label>

<input **type**="text" **asp-for**="Title" />

</div>

<div>

<label **asp-for**="Info">Текст вопроса</label>

<input **type**="text" **asp-for**="Info" />

</div>

<input **type**="hidden" **asp-for**="AuthorID" **value**="@ViewBag.id" />

<input **type**="hidden" **asp-for**="CreatedAt" **value**="@DateTime.Now.Date" />

<input type="submit" value="Задать вопрос!" />

</form>

@{

ViewData["Title"] = "Home Page";

@model dynamic

}

<div class="text-center">

@{

if (Context.Request.Cookies["auth"] != null)

{

<a **asp-action**="AddPost">Добавить вопрос</a>

}

}

@{

ViewBag.Title = "Все вопросы";

}

@foreach (var item in Model.Posts)

{

<a href="post?id=@item.id">

<div class="post-pre">

<div class="post-name">

<p class="pos-name-p">@item.Title</p>

<p></p>

</div>

<div>

<p class="post-info">@item.Info</p>

</div>

</div>

</a>

}

</div>

@model kurs2.Models.Users

<form **asp-action**="login" **asp-controller**="home">

<div>

<label **asp-for**="Name">Логин</label>

<input **type**="text" **asp-for**="Name" />

</div>

<div>

<label **asp-for**="Password">Логин</label>

<input **type**="text" **asp-for**="Password" />

</div>

<input type="submit" value="Войти" />

</form>

@model kurs2.Models.Users

<form **asp-action**="Registration" **asp-controller**="Home">

<div>

<label **asp-for**="Name">Логин</label>

<input **type**="text" **asp-for**="Name" />

</div>

<div>

<label **asp-for**="BirthDate">Дата рождения</label>

<input **type**="date" **asp-for**="BirthDate" />

</div>

<div>

<label **asp-for**="Bio">О себе</label>

<input **type**="text" **asp-for**="Bio" />

</div>

<div>

<label **asp-for**="Photo">Фото</label>

<input **type**="text" **asp-for**="Photo" />

</div>

<div>

<label **asp-for**="Password">Пароль</label>

<input **type**="password" **asp-for**="Password" />

</div>

<div>

<label **asp-for**="Email">Адрес электронной почты</label>

<input **type**="email" **asp-for**="Email" />

</div>

<input **type**="hidden" **asp-for**="RegDate" **value**="@DateTime.Now.Date" />

<input **type**="hidden" **asp-for**="CatID" **value**="1" />

<input type="submit" value="Регистрация!" />

</form>

@model IEnumerable<Users>

@{

int ID = ViewBag.id;

var user = new Users();

foreach(var a in Model)

{

if (a.id == ID)

{

user = a;

}

}

}

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title></title>

</head>

<body>

<h1>@user.Name</h1>

<div class="user-info">

<img src="https://okeygeek.ru/wp-content/uploads/2020/03/no\_avatar.png" height="300"/>

</div>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - Diploma</title>

<link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />

<link rel="stylesheet" href="~/css/site.css" />

</head>

<body>

<header>

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">

<div class="container">

<a class="navbar-brand" **asp-area**="" **asp-controller**="Home" **asp-action**="Index">Diploma</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target=".navbar-collapse" aria-controls="navbarSupportedContent"

aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex flex-sm-row-reverse">

<ul class="navbar-nav flex-grow-1">

<li class="nav-item">

<a class="nav-link text-dark" **asp-area**="" **asp-controller**="Home" **asp-action**="Index">Home</a>

</li>

@{

if (Context.Request.Cookies["auth"] != null)

{

<li class="nav-item">

<a class="nav-link text-dark" **asp-action**="exit" **asp-controller**="home">Выйти</a>

</li>

}

else

{

<li class="nav-item">

<a class="nav-link text-dark" **asp-action**="login">Логин</a>

</li>

<li class="nav-item">

<a class="nav-link text-dark" **asp-action**="registration">Регистрация</a>

</li>

}

}

</ul>

</div>

</div>

</nav>

</header>

<div class="container">

<main role="main" class="pb-3">

@RenderBody()

</main>

</div>

<footer class="border-top footer text-muted">

<div class="container">

&copy; 2022 - kurs2 - <a **asp-area**="" **asp-controller**="Home" **asp-action**="Privacy">Privacy</a>

</div>

</footer>

<script src="~/lib/jquery/dist/jquery.min.js"></script>

<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>

<script **src**="~/js/site.js" **asp-append-version**="true"></script>

@RenderSection("Scripts", required: false)

</body>

</html>

<script src="~/lib/jquery-validation/dist/jquery.validate.min.js"></script>

<script src="~/lib/jquery-validation-unobtrusive/jquery.validate.unobtrusive.min.js"></script>

@model ErrorViewModel

@{

ViewData["Title"] = "Error";

}

<h1 class="text-danger">Error.</h1>

<h2 class="text-danger">An error occurred while processing your request.</h2>

@if (Model.ShowRequestId)

{

<p>

<strong>Request ID:</strong> <code>@Model.RequestId</code>

</p>

}

<h3>Development Mode</h3>

<p>

Swapping to <strong>Development</strong> environment will display more detailed information about the error that occurred.

</p>

<p>

<strong>The Development environment shouldn't be enabled for deployed applications.</strong>

It can result in displaying sensitive information from exceptions to end users.

For local debugging, enable the <strong>Development</strong> environment by setting the <strong>ASPNETCORE\_ENVIRONMENT</strong> environment variable to <strong>Development</strong>

and restarting the app.

</p>

Контроллер

using kurs2.Models;

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

using Microsoft.Extensions.Logging;

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Dynamic;

using System.Linq;

using System.Threading.Tasks;

namespace kurs2.Controllers

{

public class HomeController : Controller

{

private readonly ILogger<HomeController> \_logger;

public static ApplicationContext db;

public HomeController(ApplicationContext context, ILogger<HomeController> logger)

{

\_logger = logger;

db = context;

}

public IActionResult Index()

{

dynamic mymodel = new ExpandoObject();

mymodel.Users = db.Users.ToList();

mymodel.Posts = db.Posts.ToList();

return View(mymodel);

}

public IActionResult Privacy()

{

return View();

}

[Route("Post")]

public ActionResult Post(int id, int? usrId)

{

ViewBag.ID = id;

ViewBag.Controller = "Post";

ViewBag.Action = "Index";

dynamic postsmodel = new ExpandoObject();

postsmodel.Users = db.Users.ToList();

postsmodel.Comments = db.Comments.ToList();

postsmodel.Posts = db.Posts.ToList();

return View(postsmodel);

}

[Route("User")]

public ActionResult User(int id)

{

ViewBag.ID = id;

return View(db.Users);

}

[Route("Login")]

public ActionResult Login(Users user)

{

var usr = db.Users.FirstOrDefault(x => x.Name == user.Name && x.Password == user.Password);

if (usr != null)

{

Response.Cookies.Append("auth", usr.id.ToString(), new Microsoft.AspNetCore.Http.CookieOptions { Expires = DateTimeOffset.Now.AddMonths(1) });

Response.Cookies.Append("pass", usr.Password.ToString(), new Microsoft.AspNetCore.Http.CookieOptions { Expires = DateTimeOffset.Now.AddMonths(1) });

Response.Redirect("/");

}

else

{

Response.Cookies.Append("Test", "Texxt");

}

return View();

}

[Route("Exit")]

public ActionResult Exit()

{

Response.Cookies.Delete("auth");

Response.Cookies.Delete("pass");

Response.Redirect("/");

return View();

}

[Route("AddPost")]

public ActionResult AddPost(Posts post, string strId)

{

strId = Request.Cookies["auth"];

int id = Convert.ToInt32(strId);

ViewBag.ID = id;

if (post.Title != "" && post.Info != null)

{

db.Posts.Add(post);

db.SaveChanges();

}

if (db.Posts.FirstOrDefault(x => x.Title == post.Title) != null)

{

Response.Redirect($"post?id={post.id}");

}

return View();

}

[Route("Registration")]

public ActionResult Registration(Users usr)

{

if(usr.Name != "" && usr.Bio != null && usr.Password != null)

{

db.Users.Add(usr);

db.SaveChanges();

Response.Redirect("/");

}

return View();

}

[Route("AddCommentary")]

public ActionResult AddCommentary(Comments cm, int id)

{

ViewBag.id = id;

if(cm.Text != null)

{

db.Comments.Add(cm);

db.SaveChanges();

Response.Redirect($"post?id={cm.PostID}");

}

return View();

}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]

public IActionResult Error()

{

return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

}

//

}

}

@using kurs2

@using kurs2.Models

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

## Заключение

Производственную практику проходил в ООО «ЦА «Максималист»» Проведён анализ материально-технической базы. Спроектирована локальная сеть организации в MS Visio на основе топологии звезда. Материально-техническая база соответствует поставленной задаче.

В соответствии с постановкой задачи был разработан форум со следующими функциональными характеристиками:

* + Регистрация аккаунтов
  + вход в сайт уже зарегистрированным пользователям
  + Возможность задавать вопросы
  + Возможность отвечать на вопросы других пользователей

В техническом задании отражены требования к составу технических средств, информационно-программной совместимости и требования, к надёжности, которые должны быть соблюдены во время разработки, внедрении и эксплуатации приложения.

В соответствии с техническим заданием разработан разработаны страницы сайта, подключена административная часть с помощью WordPress.

Для разработки сайта использовались следующие средства:

1. MS Visio Studio
2. MS SSMS

Разработана техническая документация: руководство пользователя, руководство программиста, текст программы.

Для защиты сайта используется SSL.

Приложение разработано в установленные сроки и удовлетворяет потребностям заказчика.

## Список литературы

1. Федорова Г.Н. Разработка модулей программного обеспечения для компьютерных систем: учебник для студентов СПО /Г.Н. Федорова. 2-е изд., стер. - М.: Издательский центр «Академия»,2018. - 384с.
2. Федорова Г.Н. Осуществление интеграции программных модулей: учебник для студентов СПО / Г.Н. Федорова. 2-е изд., стер. - М.: Издательский центр «Академия»,2018. - 285 с.
3. Черпаков, И. В.Основы программирования: учебник и практикум для среднего профессионального образования /И. В. Черпаков. — Москва : Издательство Юрайт, 2019. — 219 с. — (Профессиональное образование). — ISBN 978-5-9916-9984-6. — Текст: электронный //ЭБС Юрайт [сайт]. — URL: <https://biblio-online.ru/bcode/436557>
4. Хахаев, И.А. Практикум по алгоритмизации и программированию на Python: курс /И.А. Хахаев. - 2-е изд., исправ. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 179 с.: ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429256>
5. Царёв, Р.Ю. Алгоритмы и структуры данных (CDIO): учебник /Р.Ю. Царёв, А.В. Прокопенко; Министерство образования и науки Российской Федерации, Сибирский Федеральный университет. - Красноярск: СФУ, 2016. - 204 с. : ил. - Библиогр. в кн. - ISBN 978-5-7638-3388-1; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=497016>
6. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для среднего профессионального образования / Д. Ю. Федоров. — 2-е изд. — Москва: Издательство Юрайт, 2020. — 161 с.
7. Гниденко, И. Г. Технология разработки программного обеспечения: учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва: Издательство Юрайт, 2020. — 235 с.
8. Соколова, В. В. Разработка мобильных приложений: учебное пособие для среднего профессионального образования / В. В. Соколова. — Москва: Издательство Юрайт, 2020. — 175 с.
9. C# 7.0. Справочник. Полное описание языка.: Пер. с англ. – СпБ.: ООО “Альфакнига”, 2018. – 1024 с. : ил. – Парал. тит. англ.
10. Microsoft Visual C#. Подробное руководство. 8-е изд. – СПб.: Питер, 2017. – 848 с.: ил. – (Серия «Библиотека программиста»).
11. Программирование на C# для начинающих. Основные сведения / Алексей Васильев. – Москва: Эксмо, 2018. – 592 с. – (Российский компьютерный бестселлер).
12. Язык программирования C# 7 и платформы .NET и .NET Core, 8-е изд. : Пер. с англ. – СПб. : ООО “Диалектика”, 2018 – 1328 с. : ил. – Парал. тит. англ.
13. Оппель, Эндрю Дж. SQL. Полное руководство / Оппель Эндрю Дж.. - М.: Диалектика / Вильямс, 2016. - 902 c.

**Интернет-ресурсы**

ЭБС «Университетская библиотека online» - <http://biblioclub.ru/>

ЭБС «Юрайт» - <http://www.urait.ru/>