Министерство образования и науки Республики Башкортостан

Государственное бюджетное профессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Заместитель директора по учебной работе  \_\_\_\_\_\_\_ З.З. Курмашева  «\_\_\_\_»\_\_\_\_\_\_\_\_ 2020 г. |

АВТОМАТИЗАЦИЯ МОНИТОРИНГА УЧАЩИХСЯ РБ ПО ОПРЕДЕЛЕНИЮ ПРОФЕССИОНАЛЬНЫХ НАМЕРЕНИЙ И ОЖИДАНИЙ БУДУЩИХ СПЕЦИАЛИСТОВ

Пояснительная записка к дипломному проекту

|  |  |
| --- | --- |
| Рецензент  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Е.П. Зорина  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. | Руководитель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_О.В. Фатхулова  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |
|  | Выпускник гр. 16П-2  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.А. Каюмов  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |

2020

Министерство образования и науки Республики Башкортостан

Государственное бюджетное профессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Заместитель директора  по учебной работе  З.З. Курмашева  « » 2020 год |

ЗАДАНИЕ

на дипломный проект студенту дневного отделения, группы 16П-2, специальности 09.02.03 Программирование в компьютерных системах.

Фамилия, имя, отчество: Каюмов Сажод Акрамчонович

Тема дипломного проекта: «Автоматизация мониторинга учащихся РБ по определению профессиональных намерений и ожиданий будущих специалистов».

Текст задания:

при выполнении дипломного проекта должны быть решены следующие задачи:

1. разработана структура программы;
2. реализованы функции создания, изменения, удаления опросов
3. регистрации и добавления пользователей системы;
4. возможность добавления ответов пользователя на опросы;
5. построена справочная система.

В результате выполнения дипломного проекта должны быть представлены:

1. пояснительная записка, состоящая из следующих разделов:

Введение

1 Постановка задачи

2 Экспериментальный раздел

3 Экономический раздел

Заключение

Приложения

Список сокращений

Список использованных источников

1. электронный носитель, содержащий разработанный программный продукт;
2. презентация дипломного проекта в электронном виде.

Список рекомендуемых источников:

1. ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения. [Текст]. – Введ. 1992–01–01 -М.: Изд-во стандартов, 1992.- 24 с.
2. [Агальцов В. П.](http://znanium.com/catalog.php?item=booksearch&code=%D0%B1%D0%B0%D0%B7%D1%8B+%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85&page=1#none) Базы данных. В 2-х кн. Кн. 2. Распределенные и удаленные базы данных: учеб. / В.П. Агальцов. - М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 272 с.: ил. - <http://znanium.com/bookread2.php?book=372740>
3. Википедия [Электронный ресурс] // Свободная энциклопедия. – Режим доступа: <http://ru.wikipedia.org/wiki/>, свободный
4. Эванс, Э. Предметно-ориентированное программирование. Структуризация сложных программных систем [Текст]: / Э. Эванс - М.: Вильямс, 2018. – 448 с.

Задание к выполнению получил «30» марта 2020г.

Студент Каюмов Сажод Акрамчонович

Срок окончания «08» июня 2020 г.

Руководитель дипломного проекта О.В. Фатхулова

Задание рассмотрено на заседании цикловой комиссии информатики   
«25» марта 2020 г. Протокол №5.

Председатель цикловой комиссии информатики О.В. Фатхулова

АННОТАЦИЯ

Пояснительная записка к дипломному проекту содержит постановку и программу решения задачи «Автоматизация мониторинга учащихся РБ по определению профессиональных намерений и ожиданий будущих специалистов».

Web-приложение PollDesigner написано на языках программирования C#, JavaScript в среде программирования Visual Studio 2019 с использованием технологии ASP.NET CORE 3.0 и сервера баз данных MSSQL SERVER и предназначено для работы в браузере на любой платформе, отлажено на данных контрольного примера.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

2

40.К-1948-2020 09.02.03 ДП-ПЗ

Разраб.

Каюмов С.А.

Провер.

Фатхулоа О.В.

Реценз.

Зорина Е.П.

Н. Контр.

Каримова Р.Ф.

Утверд.

Курмашева З.З.

Автоматизация мониторинга учащихся РБ по определению профессиональных намерений и ожиданий будущих специалистов

Лит.

Листов

124

16П-2 УКСИВТ

СОДЕРЖАНИЕ

|  |  |
| --- | --- |
|  | лист |
| Введение | 4 |
| 1 Постановка задачи | 6 |
| 1.1 Описание предметной области | 6 |
| 1.2 Диаграмма прецедентов предметной области | 8 |
| 1.3 Описание входной информации | 9 |
| 1.4 Описание выходной информации | 10 |
| 1.5 Общие требования к программному продукту | 12 |
| 1.6 Описание структуры базы данных | 13 |
| 1.7 Контрольный пример | 14 |
| 2 Экспериментальный раздел | 16 |
| 2.1 Описание программы | 16 |
| 2.2 Протокол тестирования программного продукта | 19 |
| 2.3 Руководство пользователя | 27 |
| 3 Экономический раздел | 39 |
| 3.1 Расчет затрат на создание программного продукта | 39 |
| 3.2 Расчет цены предложения | 43 |
| Заключение | 45 |
| Приложение А. Выходные данные контрольного примера | 46 |
| Приложение Б. Исходный код программы | 48 |
| Приложение В. Результаты работы приложения | 123 |
| Список использованных источников | 124 |

ВВЕДЕНИЕ

В наше время становится не актуальным проведение тестов и опросов с помощью бумажных носителей, так как результат должен храниться в виде отчётов что не является целесообразным. На замену таким тестам приходят программные средства, которые позволят формировать различные тематические тесты и хранить информацию по тестам и ответам на них на компьютере. Но создание отдельных программ для новой темы опроса — это не практично. Правильным решением данной проблемы будет написание программы-оболочки, которая позволит конструировать различные опросы по выбранной теме.

Такие опросы унифицированы и будут иметь определённые ограничения, но они помогут сэкономить большое количество времени на разработку отдельных программ.

Для того чтобы тестируемый не устанавливал различные программные средства наиболее удобно будет делать всё в браузере с использованием соответствующего web-приложения.

Актуальность темы заключается в том, что бумажные опросы по сей день используются в школах, колледжах и вузах. Каждый год ученики заканчивают школы и в учебные заведения приходят множество новых студентов. Для определения их дальнейших планов на жизнь создаются тесты и опросы, которые проходят большое количество человек. Автоматизировать данный процесс поможет создаваемое web-приложение.

Цель дипломного проекта состоит в разработке автоматизированной системы, которая упростит проведения опросов среди учащихся и обработку их ответов.

Для достижения поставленной цели необходимо решить следующие задачи:

* исследовать предметную область, а именно процесс создания и прохождения теста;
* выполнить проектирование базы данных, содержащую сведения о тестируемых;
* написать программный продукт, позволяющий формировать Json файл, содержащий тест;
* разработать и протестировать web-приложение, которое позволит проходить опросы и сформировать различные отчёты для анализа по всем опрашиваемым.

1 Постановка задачи

1.1 Описание предметной области

Требуется разработать web-приложение «Автоматизация мониторинга учащихся РБ по определению профессиональных намерений и ожиданий будущих специалистов» для учебных учреждений. Опрос должен иметь анонимный характер.

Каждый опрос должен быть унифицирован и содержать следующую информацию:

* логин пользователя (администратора), создающего опрос;
* тип опроса;
* номер (код) опроса;
* текст опроса.

Опрос содержит вопросы, информация о которых определяется следующими полями:

* номер (код) опроса;
* номер вопроса;
* текст вопроса;
* обязательный ли вопрос.

Вопрос может содержать ответы, о которых известна следующая информация:

* код вопроса;
* номер ответа;
* текст ответа.

Вопросы могут быть нескольких типов:

* код типа;
* наименование.

Опрашиваемый (аккаунт) должен иметь возможность пройти опрос по определённой теме. Тема опроса устанавливается администратором. Для администратора должна быть реализована возможность сгенерировать аккаунты для опрашиваемых. Данные аккаунты будут предоставлены пользователям, которые будут проходить данный опрос. Аккаунты должны быть привязаны к конкретному администратору и опросу.

Созданные аккаунты будут иметь следующие поля:

* код аккаунта;
* код пользователя;
* номер опроса;
* логин;
* пароль;
* учебное заведение;
* опрос пройден.

Приложение должно предусматривать возможность авторизации аккаунта.

Для администратора должна быть реализована возможность регистрации нового аккаунта. Каждого аккаунта содержится следующая информация:

* код пользователя;
* фамилия;
* имя;
* отчество;
* электронная почта;
* логин;
* пароль.

Для работы с web-приложением будут предусмотрены следующие ограничения:

* при регистрации нового аккаунта, логин и пароль могут содержать только английские буквы в любом регистре и цифры;
* нельзя зарегистрировать аккаунт с уже существующим логином;
* один пользователь может иметь максимум 10 опросов;
* каждый опрос может содержать максимум 150 вопросов;
* каждый вопрос может содержать максимум 20 ответов;
* для каждого опроса за раз можно будет сгенерировать не боле 500 аккаунтов.

С данным приложением должны работать следующие группы пользователей:

* администратор опроса;
* опрашиваемый;

Администратор должен иметь возможность решать следующие задачи:

* создавать новые опросы;
* генерировать аккаунты для опроса;
* формировать отчёты по результатам опроса.

Опрашиваемый должен иметь возможность прохождения опроса.

1.2 Диаграмма прецедентов предметной области

Диаграмма прецедентов предметной области представлена на рисунке 1.2.1.

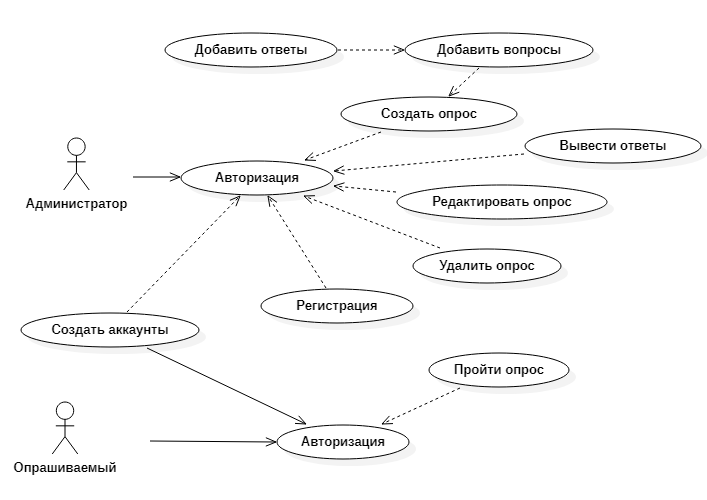


Рисунок 1.2.1 – Диаграмма прецедентов

1.3 Описание входной информации

Входными документами для задачи будут являться:

* список опрашиваемых с логинами и паролями;
* опросный лист с вопросами и предложенными ответами на выбор.

Входной информацией будут являться ответы опрашиваемых на вопросы опроса.

Описание входных документов представлено в таблице 1.3.1.

Таблица 1.3.1 – Описание входных документов

|  |  |  |
| --- | --- | --- |
| Наименование документа (шифр) | Периодичность поступления документа | Откуда поступает документ |
| Список опрашиваемых | Перед прохождения опроса | От администратора |
| Опросный лист | Перед прохождения опроса | От администратора |

Входным файлом является Json файл, где содержатся сведения об опросе. Описание Json файла представлено в таблицах 1.3.2, 1.3.3.

Таблица 1.3.2 – Описание входного файла

|  |  |  |  |
| --- | --- | --- | --- |
| Название файла | Шифр файла | Тип файла | Источник поступления |
| Опрос | poll.json | Текстовый | От администратора |

Таблица 1.3.3 – Описание реквизитов входного файла

|  |  |  |  |
| --- | --- | --- | --- |
| Шифр файла | Наименование реквизитов | Шифр реквизитов | Форма представления |
| 1 | 2 | 3 | 4 |
| poll.json | Логин администратора | CreatorLogin | Строка |
| Номер опроса | Number | Число |
| Текст опроса | PollText | Строка |
| Список вопросов | tQuestion | Строка |

Продолжение таблицы 1.3.3

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
|  | Номер вопроса | Number | Число |
| Текст вопроса | QuestionText | Строка |
|  | Тип | Type | Строка |
| Обязательный вопрос | IsRequired | true/false |
| Список ответов | tResponse | Строка |
| Номер ответа | Number | Число |
| Текст ответа | ResponseText | Строка |
| Использовать поле для ввода | UseTextBox | true/false |

1.4 Описание выходной информации.

Выходным документом будет являться отчёт по прохождению опроса всеми опрашиваемыми, который будет записываться в таблицу Excel. В нем выводятся ответы только по определённому опросу.

Описание выходного документа представлено в таблице 1.4.1.

Таблица 1.4.1 – Описание выходного документа

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование документа (шифр) | Периодичность выдачи документа | Кол-во экз. | Куда передаются | Поля сортировки | Поля группировки | Итоги |
| Отчёт на опрос | При выводе результатов | 1 | Администратору | Код опрашиваемого | - | - |

Шаблон выходного документа представлен на рисунке 1.4.1.

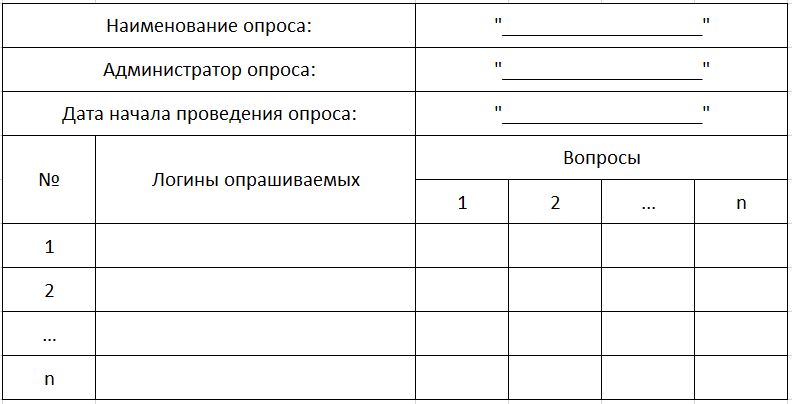


Рисунок 1.4.1 – Шаблон выходного документа

Выходным файлом является Json файл, содержащий сведения о прохождении опроса всеми пользователями. Описание выходного файла и его реквизитов представлено в таблицах 1.4.2 и 1.4.3.

Таблица 1.4.2 – Описание выходного файла

|  |  |  |  |
| --- | --- | --- | --- |
| Название файла | Шифр файла | Тип файла | Источник приема |
| Ответы на опрос | answer.json | Текстовый | От опрашиваемых |

Таблица 1.4.3 – Описание реквизитов выходного файла

|  |  |  |  |
| --- | --- | --- | --- |
| Шифр файла | Наименование реквизитов | Шифр реквизитов | Форма представления |
| answer.json | Логин администратора | CreatorLogin | Строка |
| Номер опроса | Number | Число |
| Дата проведения опроса | PollDate | Дата |
| Список вопросов | tQuestion | Строка |
| Номер вопроса | Number | Число |
| Список ответов | tResponse | Строка |
| Номер ответа | Number | Число |
| Ответ | Response | Строка |

1.5 Общие требования к программному продукту

Общее наименование продукта «PollDesigner». Для работы с ним администратор и опрашиваемый должны обладать базовыми навыками работы с ПК.

Надежность приложения должна быть обеспечена выполнением следующих пунктов:

* разграничение прав пользователей опроса;
* аутентификация и идентификация пользователей опроса;
* возможность создания и восстановления резервной копии опроса администратором;

Отказы программы вследствие некорректных действий пользователя при взаимодействии с приложением недопустимы.

Технические и программные средства необходимые для ввода продукта в эксплуатацию:

* стабильный хостинг;
* объём памяти на хостинге не менее 10 Гб;
* поддержка СУБД MSSQL на хостинге.

Минимальные системные требования для компьютера, на котором планируется использование приложения:

* операционная система не ниже Windows 7;

- объем ОЗУ не менее 512 Гб;

- процессор Intel Pentium(R) 2.16 GHz и выше;

- операционная система Windows 7 и выше;

- не менее 500 Мб свободного места на жестком диске.

Минимальные системные требования для мобильного устройства, на котором планируется использование приложения:

* операционная система не ниже Android 4.4 / IOS 10;

- объем ОЗУ не менее 512 Мб;

Необходимые виды периферийных устройств для работы с компьютера:

- манипулятор типа "мышь";

- манипулятор типа "клавиатура";

- монитор.

Необходимые виды периферийных устройств для работы с мобильных устройств:

- экран с диагональю не менее 4 дюймов;

- манипулятор типа "клавиатура".

1.6 Описание структуры базы данных

На основании входных документов создаются таблицы базы данных, описание которых приведено ниже в таблицах 1.6.1.

Для создания базы данных будет использованы СУБД MSSQL SERVER и Json файлы.

Таблица 1.6.1 – Описание структуры базы данных в MSSQL SERVER

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Creator (Список администраторов опроса) | | | | |
| Имя поля | Описание поля | Тип данных | Размер поля | Тип ключа |
| 1 | 2 | 3 | 4 | 5 |
| id | Код администратора | int | 4 | PK |
| login | Логин | varchar | 50 | UQ |
| password | Пароль | varchar | 50 |  |
| email | Электронная Почта | varchar | 50 |  |
| firstname | Фамилия | varchar | 50 |  |
| name | Имя | varchar | 50 |  |
| lastname | Отчество | varchar | 50 |  |
| User (Список опрашиваемых) | | | | |
| id | Код опрашиваемого | int | 4 | PK |

Продолжение таблицы 1.6.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| login | Логин | varchar | 50 | UQ |
| password | Пароль | varchar | 11 |  |
| pollNumber | Номер опроса | int | 4 |  |
| institution | Учебное заведение | varchar | 50 |  |
| isPassed | Опрос пройден? | varchar | 50 |  |
| creatorId | Код администратора | int | 4 | FK |

\*PK-первичный ключ, FK-внешний ключ, UQ – уникальный индекс

1.7 Контрольный пример

Основными методами устранения ошибок является тестирование и прогонка web-приложения по заранее подготовленным тестам.

Тестирование позволяет не только устранить ошибки, но и оценить готовность программного обеспечения к эксплуатации.

Контрольный пример представляет собой ручной вариант решения задачи с известными результатами и используется для проверки правильности.

В таблицах 1.7.1-1.7.4 представлены входные и выходные данные для контрольного примера.

Таблица 1.7.1 – Входные данные контрольного примера для регистрации администратора

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Фамилия | Имя | Отчество | Почта | Логин | Пароль | Подтвердите пароль |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Иванов | Иван | Иванович | ivan@mail.ru | ivan | 123 | 123 |
| Каюмов | Сажод | Акрамчонович | sazhod@yandex.ru | sazhod | 123 | 123 |
| Петров | Олег | Константинович | oleg@gmail.com | oleg | 123 | 123 |

Таблица 1.7.2 – Входные данные опроса для контрольного примера

|  |  |
| --- | --- |
| Логин администратора | Наименование опроса |
| ivan | Информация об учащихся |
| sazhod | Место жительства |
| oleg | Информация о родителе (опекуне) |

Таблица 1.7.3 – Входные данные контрольного примера для создания вопроса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Логин  администратора | Номер опроса | Номер вопроса | Вопрос | Предложенные ответы |
| 1 | 2 | 3 | 4 | 5 |
| ivan | 1 | 1 | Сколько вам лет? | Меньше 16 |
| От 16 до 18 |
| Больше 18 |
| 2 | Ваш пол | Мужской |
| Женский |
| 3 | Укажите ваше учебное заведение |  |
| sazhod | 2 | 1 | Ваше Ф.И.О |  |
| 2 | Ваш адрес проживания |  |
| 3 | Ваш адрес прописки |  |
| 4 | За сколько по времени вы доходите от дома до учебного заведения? | Меньше 10 минут |
| От 10 до 20 минут |
| Больше 20 минут |
| oleg | 3 | 1 | Ф.И.О вашего родителя |  |
| 2 | Пол родителя | Мужской |
| Женский |
| 3 | Дата рождения родителя |  |
| 4 | Номер телефона родителя |  |

Выходные данные контрольного примера для отчётов представлены в приложении А.

2 Экспериментальный раздел

2.1 Описание программы

Приложение было спроектировано в среде разработки Visual Studio 2019 с использованием технологии ASP.NET CORE 3.0 и паттерна проектирования MVC (Model-View-Controller). Структура паттерна проектирования представлена на рисунке 2.1.1.



Рисунок 2.1.1 – Структура паттерна проектирования

При открытии приложения первым делом срабатывает файл Program.cs который содержит в себе указания о том какой файл дальше запускать. Им является файл Startup.cs, который содержит в себе все серверные настройки приложения. Описание всех модулей и их процедур содержится в таблице 2.1.1.

Таблица 2.1.1 – Описание модулей и их процедур

|  |  |
| --- | --- |
| Модуль | Назначение |
| 1 | 2 |
| Модуль Startup – серверные настройки приложения | |
| ConfigureServices | Основные серверные настройки приложения |
| Configure | Подключение расширений и работа с маршрутизацией |
| Модуль AccountController –отвечает за работу с аккаунтами | |
| Login | Авторизация пользователя |
| Register | Регистрация пользователя |
| Authenticate | Аутентификация пользователя |
| Logout | Выход из аккаунта |
| VerifiedAccount | Подтверждение аккаунта по почте |
| VerifiedMessage | Отображение сообщения об отправке письма на почту |
| CheckEmail | Проверка почты для восстановления пароля |
| RestorePassword | Восстановление пароля |
| EditEmail | Изменение эл. почты |
| EditFullName | Изменение Ф.И.О. |
| EditPassword | Изменение пароля |
| Модуль MainPageController – отображение опросов | |
| Questioning | Отображение опросов для пользователя на странице просмотра |
| PassingPoll | Отображение опроса для опрашиваемого |
| CompletionPoll | Завершение опроса и отправка ответов на сервер |
| ProcessingAnswer | Обработка и сохранение ответов |
| ExportPoll | Экспорт опроса |
| ImportPoll | Импорт опроса |
| Модуль EditPollController –формирование Json файла опроса | |
| Add | Переход на страницу добавления нового опроса |
| SaveAdd | Сохранение созданного опроса в Json файл |
| Edit | Переход на страницу изменения выбранного опроса |
| SevaEdit | Сохранение изменённого опроса |
| Delete | Удаление опроса |
| Модуль UserController – изменение, добавление аккаунтов опрашиваемых | |
| ShowUser | Переход на страницу и отображение всех аккаунтов привязанных к выбранному опросу |

Продолжение таблицы 2.1.1

|  |  |
| --- | --- |
| 1 | 2 |
| AddUser | Добавление новых аккаунтов для опрашиваемых |
| Download | Загрузка pdf файла со списком аккаунтов для опрашиваемых |
| CreateUser | Генерация логинов и паролей для аккаунтов опрашиваемых |
| DeleteUser | Удаление аккаунта опрашиваемого |
| DeleteUsers | Удаление списка аккаунтов опрашиваемых |
| CreateReport | Формирование отчёта |
| Модуль AnswerRepository – связь с Json файлом ответов | |
| JsonFromAnswer | Формирование Json строки ответов |
| DeleteAnswer | Удаление ответов опрашиваемого |
| GetAnswers | Получение списка ответов опрашиваемых |
| SaveAnswer | Сохранение ответов опрашиваемого |
| Модуль CreatorRepository – связь с таблицей администратора | |
| AllCreators | Возвращает список всех пользователей |
| AddCreator | Добавляет нового пользователя в базу данных |
| GetCreatorFio | Получение Ф.И.О. администратора опроса |
| GetCreator | Получение всех данных об администраторе опроса |
| UpdateCreator | Обновление данных об администраторе опроса |
| Модуль PollRepository – связь с Json файлом опроса | |
| GetPolls | Получение списка всех опросов пользователя |
| GetPoll | Получение конкретного опроса пользователя |
| SaveAddPoll | Сохранение нового опроса в Json файл |
| SaveEditPoll | Сохранение изменённого опроса в Json файл |
| DeletePoll | Удаление выбранного опроса из Json файла |
| ExportPoll | Формирование Json строки опроса для экспорта |
| ImportPoll | Формирование опроса из строки для импорта |
| Модуль UserRepository - связь с таблицей опрашиваемых | |
| GetUsers | Возвращает список опрашиваемых по выбранному опросу |
| AddUser | Добавляет нового опрашиваемого в базу данных |
| GetUser | Возвращает пользователя |
| UpdateUser | Обновление данных об опрашиваемом |
| GetPassingDate | Получение даты прохождение опроса |

Продолжение таблицы 2.1.1

|  |  |
| --- | --- |
| 1 | 2 |
| DeleteUser | Удаление опрашиваемого |
| Модуль ScriptJs – функционал на страницах html | |
| AddFieldQuestion | Добавление конструктора вопроса на страницу |
| DeleteFieldQuestion | Удаление конструктора вопроса со странице |
| AddFieldResponse | Добавление конструктора ответа на страницу |
| DeleteFieldResponse | Удаление конструктора ответа со странице |
| ShowForm | Отображение формы подтверждения удаления опроса |
| CheckQuestion | Проверка ответов опрашиваемого |
| FillCheckPoll | Заполнение списка вопросов и их состояния |
| SubmitForm | Подтверждение отправки формы с ответами опрашиваемого |
| CheckUserDelete | Формирование списка для удаления опрашиваемых |
| SubmitUserForm | Отправка формы для удаления опрашиваемых |
| AllDeleteCheckBox | Формирование списка для удаления всех опрашиваемых |
| CheckImportForm | Проверка формы для импорта |
| HidePassword | Скрытие или отображение полей с паролем |

Модульная схема приложения представлена на рисунке 2.1.2.

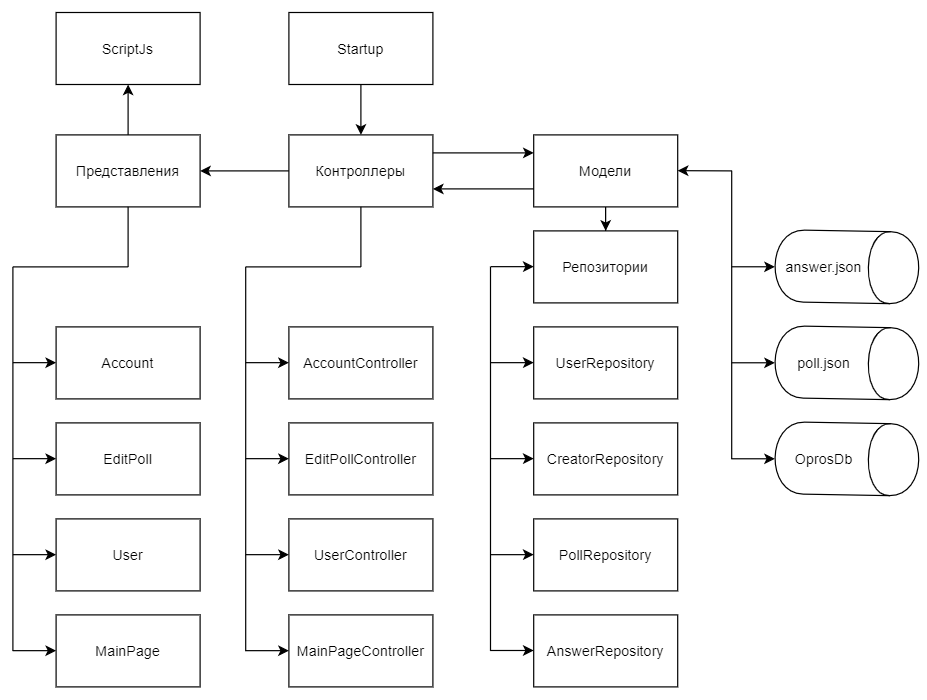


Рисунок 2.1.2 – Модульная схема приложения

2.2 Протокол тестирования программного продукта

В ходе тестирования программного продукта на корректных и некорректных данных не было обнаружено ошибок, которые влияли бы на работу самого программного продукта и всей системы.

Данный программный продукт удовлетворяет всем предъявленным требованиям, имеет комфортный интерфейс и интуитивно понятный функционал, исключает появления системных ошибок.

В таблицах 2.2.1 – 2.2.10 представлена общая информация о тестировании приложения.

Таблица 2.2.1 – Протокол тестирования авторизации на некорректных данных

|  |  |
| --- | --- |
| Общая информация о тестировании | |
| 1 | 2 |
| Название проекта | PollDesigner |
| Номер версии | 1.0 |
| Имя тестера | Каюмов Сажод Акрамчонович |
| Даты тестирования | 05.05.2020 |
| Описание информационных полей для тестирования | |
| Наименование | Авторизация |
| Наименование проекта | PollDesigner |
| Номер версии | 1.0 |
| Имя тестера | Каюмов Сажод Акрамчонович |
| Даты тестирования | 05.05.2020 |
| Test Case # | TC\_UI\_1 |
| Приоритет тестирования | Высокий |
| Название тестирования/ Имя | Проверка авторизации на не корректных данных аккаунта |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе не правильных данных |
| Шаги тестирования | Ввод не верного логина  Ввод не верного пароля  Нажатие кнопки «Войти» |
| Данные тестирования | Логин: tester  Пароль: 123 |

Продолжение таблицы 2.2.1

|  |  |
| --- | --- |
| 1 | 2 |
| Ожидаемый результат | Программа должна сообщить пользователю о том, что введённые данные не корректны |
| Фактический результат | На странице отобразилось сообщение о некорректных данных (см. рисунок 2.2.1) |
| Предпосылки | Перейти на страницу авторизации |
| Постусловия | Система не зависает, находится в состояние полной работоспособности. |
| Статус *(Pass/Fail)* | Pass |

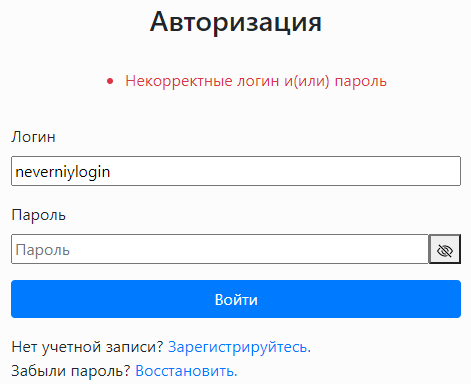


Рисунок 2.2.1 – Сообщение о некорректных данных аккаунта

Таблица 2.2.2 – Протокол тестирования авторизации на корректных данных

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| 1 | 2 |
| Наименование | Авторизация |
| Наименование проекта | PollDesigner |
| Номер версии | 1.0 |
| Имя тестера | Каюмов Сажод Акрамчонович |
| Даты тестирования | 05.05.2020 |
| Test Case # | TC\_UI\_2 |
| Приоритет тестирования | Высокий |

Продолжение таблицы 2.2.2

|  |  |
| --- | --- |
| 1 | 2 |
| Название тестирования/ Имя | Проверка авторизации на корректных данных аккаунта |
| Резюме испытания | Необходимо добиться корректного поведения программы при вводе правильных данных |
| Шаги тестирования | Ввод не верного логина  Ввод не верного пароля  Нажатие кнопки «Войти» |
| Данные тестирования | Логин: sazhod  Пароль: 123 |
| Ожидаемый результат | Пользователя должно перенаправить на страницу просмотра опроса |
| Фактический результат | После ввода корректных данных пользователя перенаправило на страницу просмотра опроса (см. рисунок 2.2.2) |
| Предпосылки | Перейти на страницу авторизации |
| Постусловия | Система не зависает, находится в состояние полной работоспособности |
| Статус (Pass/Fail) | Pass |

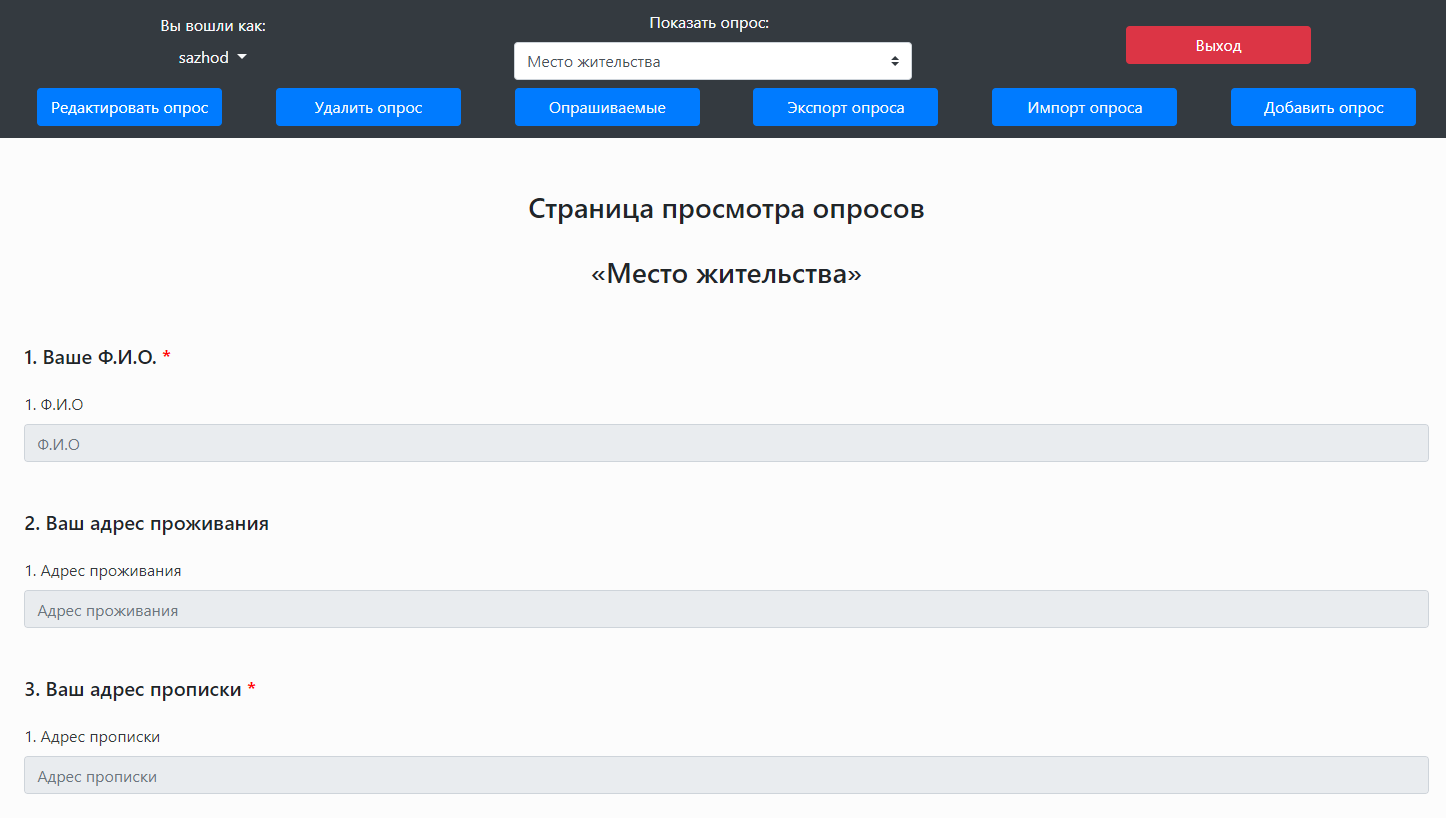


Рисунок 2.2.2 – Страница просмотра опроса

Таблица 2.2.3 – Протокол тестирования создания опроса на некорректных данных

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| Наименование | Опрос |
| Наименование проекта | PollDesigner |
| Номер версии | 1.0 |
| Имя тестера | Каюмов Сажод Акрамчонович |
| Даты тестирования | 05.05.2020 |
| Test Case # | TC\_UI\_3 |
| Приоритет тестирования | Высокий |
| Название тестирования/ Имя | Проверка сохранения опроса |
| Резюме испытания | Необходимо добиться корректного поведения программы при сохранение опроса с пустыми полями |
| Шаги тестирования | Ввод логина  Ввод пароля  Нажатие кнопки «Войти»  Нажатие кнопки «Добавить»  Нажатие кнопки «Добавить вопрос»  Нажатие кнопки «Сохранить» |
| Данные тестирования | Логин: sazhod  Пароль: 123 |
| Ожидаемый результат | Пользователя должно уведомить о том что необходимо заполнить поля |
| Фактический результат | После нажатие кнопки «Сохранить» пользователя уведомляет о том, что необходимо заполнить поля (см рисунок 2.2.3) |
| Предпосылки | Авторизоваться и перейти на страницу создания нового опроса |
| Постусловия | Система не зависает, находится в состояние полной работоспособности |
| Статус (Pass/Fail) | Pass |

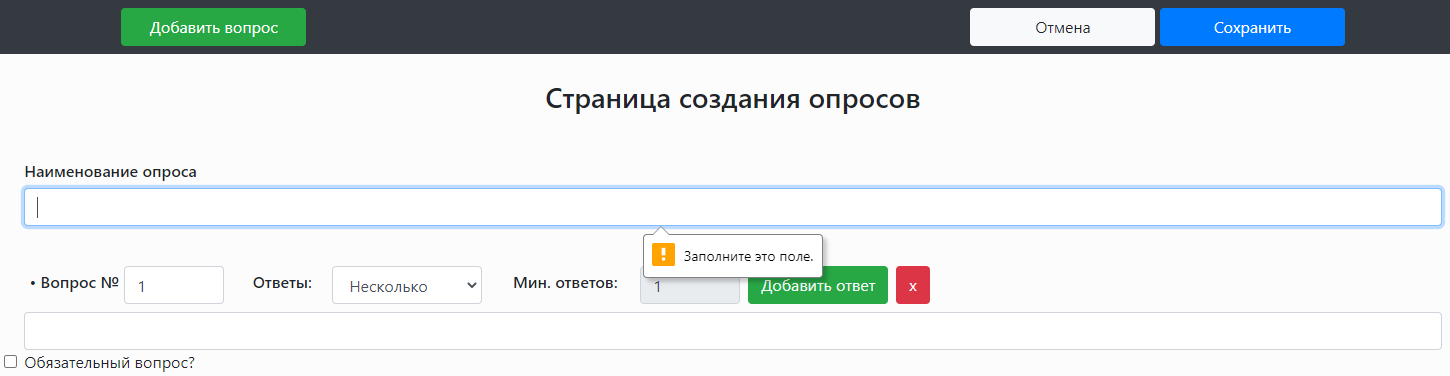


Рисунок 2.2.3 – Сообщение о не заполненном поле

Таблица 2.2.4 – Протокол тестирования создания опроса на корректных данных

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| 1 | 2 |
| Наименование | Опрос |
| Наименование проекта | PollDesigner |
| Номер версии | 1.0 |
| Имя тестера | Каюмов Сажод Акрамчонович |
| Даты тестирования | 05.05.2020 |
| Test Case # | TC\_UI\_4 |
| Приоритет тестирования | Высокий |
| Название тестирования/ Имя | Проверка сохранения опроса |
| Резюме испытания | Необходимо добиться корректного поведения программы при сохранение опроса с заполнеными полями |
| Шаги тестирования | Ввод логина  Ввод пароля  Нажатие кнопки «Войти»  Нажатие кнопки «Добавить»  Нажатие кнопки «Добавить вопрос»  Заполнить поля  Нажатие кнопки «Сохранить» |

Продолжение таблицы 2.2.4

|  |  |
| --- | --- |
| 1 | 2 |
| Данные тестирования | Логин: sazhod  Пароль: 123 |
| Ожидаемый результат | Опрос должен быть сохранён, а пользователя должно перенаправить на страницу просмотра опроса |
| Фактический результат | После нажатие кнопки «Сохранить» опрос сохраняется, а пользователя перенаправляет на страницу просмотра опроса (см рисунки 2.2.4 и 2.2.5) |
| Предпосылки | Авторизоваться, перейти на страницу создания нового опроса и заполнить все поля |
| Постусловия | Система не зависает, находится в состояние полной работоспособности |
| Статус (Pass/Fail) | Pass |

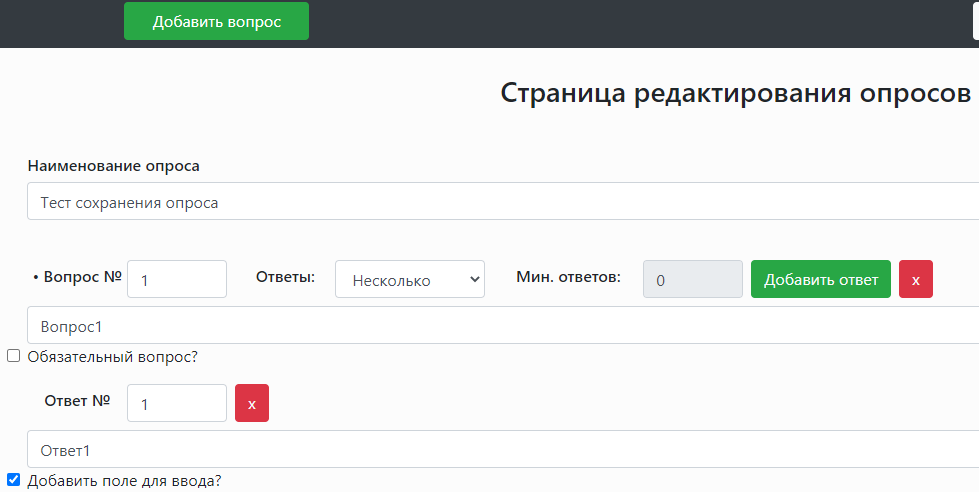


Рисунок 2.2.4 – Создание нового опроса

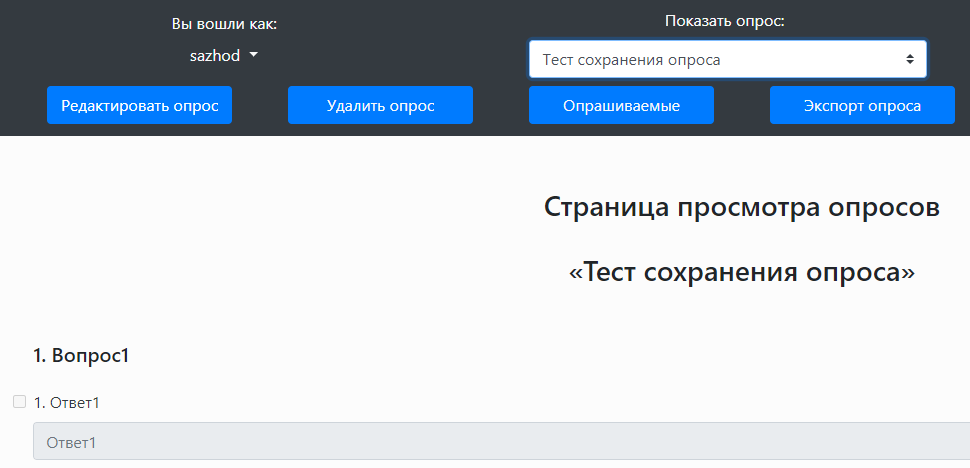


Рисунок 2.2.5 – Корректное сохранение опроса

Таблица 2.2.5 – Протокол тестирования прохождения опроса на некорректных данных

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| 1 | 2 |
| Наименование | Опрос |
| Наименование проекта | PollDesigner |
| Номер версии | 1.0 |
| Имя тестера | Каюмов Сажод Акрамчонович |
| Даты тестирования | 05.05.2020 |
| Test Case # | TC\_UI\_5 |
| Приоритет тестирования | Высокий |
| Название тестирования/ Имя | Прохождение опроса |
| Резюме испытания | Необходимо добиться корректного поведения программы при прохождение опроса с невыбранными ответами |
| Шаги тестирования | Ввод логина  Ввод пароля  Нажатие кнопки «Отправить» |
| Данные тестирования | Логин: OxrU5O2tx7Dt598  Пароль: iP76XGwS5gs4qyM |
| Ожидаемый результат | Опрашиваемого должно уведомить о том, что он не ответил на вопрос |
| Фактический результат | После нажатие кнопки «Отправить» опрашиваемого уведомляет о том что он не ответил на вопрос (см рисунок 2.2.6) |
| Предпосылки | Авторизоваться как опрашиваемый, отправить результат прохождения опроса |
| Постусловия | Система не зависает, находится в состояние полной работоспособности |
| Статус (Pass/Fail) | Pass |

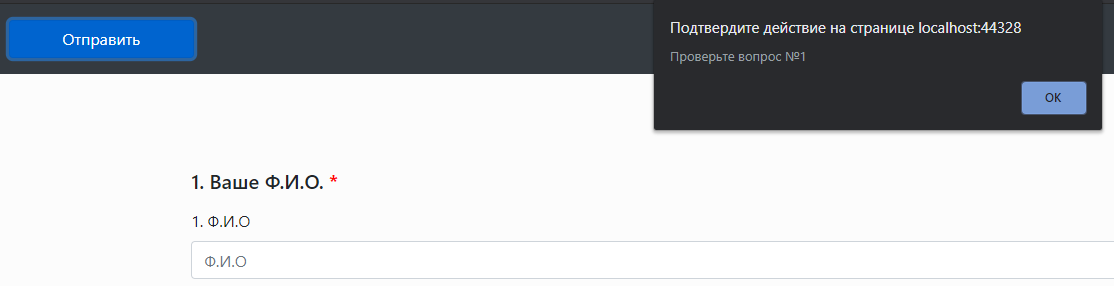


Рисунок 2.2.6 – Уведомление о некорректном ответе на вопрос

2.3 Руководство пользователя

Цель руководства заключается в расписании подробной инструкции и информации для пользователя, чтобы он смог самостоятельно пользоваться приложением и правильно ее эксплуатировал.

Пользователями приложения являются администратор опроса и опрашиваемый. В функционал администратора опроса входит создание, редактирование и удаление опроса, создание и удаление аккаунтов для опрашиваемых. В свою очередь опрашиваемому доступно только прохождение опроса.

Руководство пользователя для администратора опросов.

Для того чтобы пользоваться приложением, пользователю необходимо иметь доступ к интернету. Пользоваться приложением можно с любого устройства, будь то ПК, планшет или телефон. Далее необходимо перейти на сайт приложения. После этого пользователь окажется на странице авторизации. На ней пользователь увидит поля для ввода данных от аккаунта и кнопку входа, а также ссылку для перехода к регистрации нового пользователя. Страница авторизации представлена на рисунке 2.3.1.

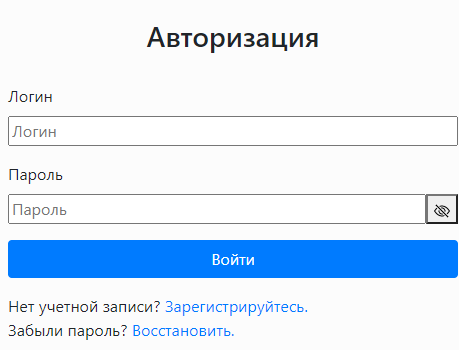


Рисунок 2.3.1 – Страница авторизации

Если пользователь уже имеет аккаунт в приложении, то для использования вам необходимо ввести логин и пароль от него. Далее необходимо нажать на кнопку «Войти». Если введённые данные корректны, то пользователя перенаправит на страницу просмотра опросов. Страница опроса представлена на рисунке 2.3.2.

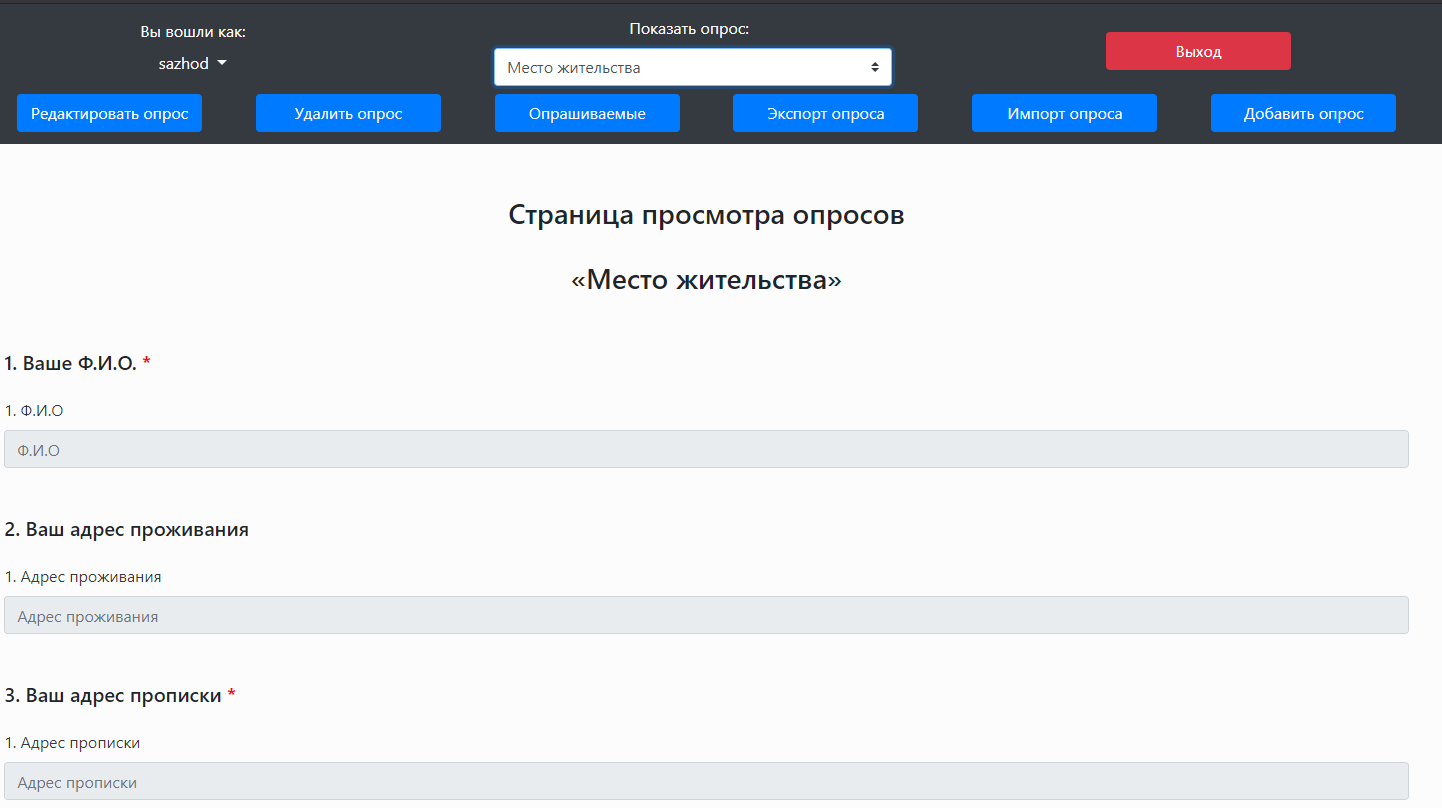


Рисунок 2.3.2 – Страница просмотра опросов

Если пользователь не имеет аккаунта, то ему необходимо перейти на страницу регистрации. Для этого нужно нажать на ссылку «Зарегистрируйтесь». После этого пользователя перенаправит на страницу регистрации. Страница регистрации представлена на рисунке 2.3.3.

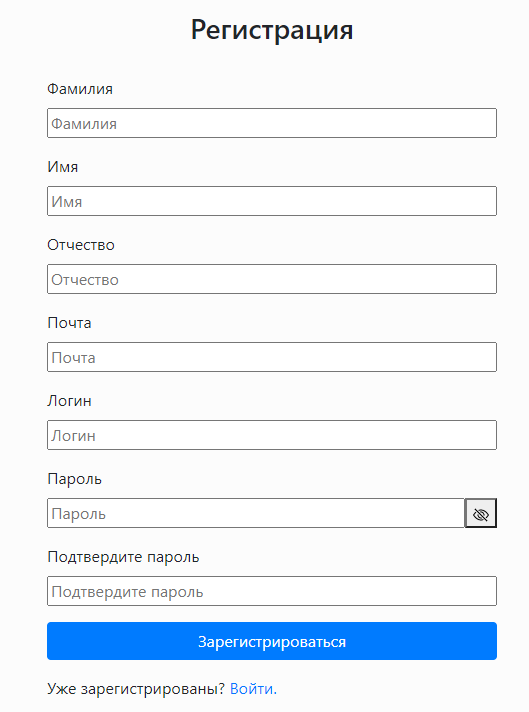


Рисунок 2.3.3 – Страница регистрации

После ввода всех данных, на указанную почту будет оправлено письмо с подтверждением аккаунта. После перехода по ссылке из письма пользователя перенаправит на страницу авторизации. После авторизации пользователя перенаправит на страницу просмотра опросов. Она показана на рисунке 2.3.4. Поскольку пользователь только создал аккаунт, то у него ещё нет доступных для просмотра опросов.

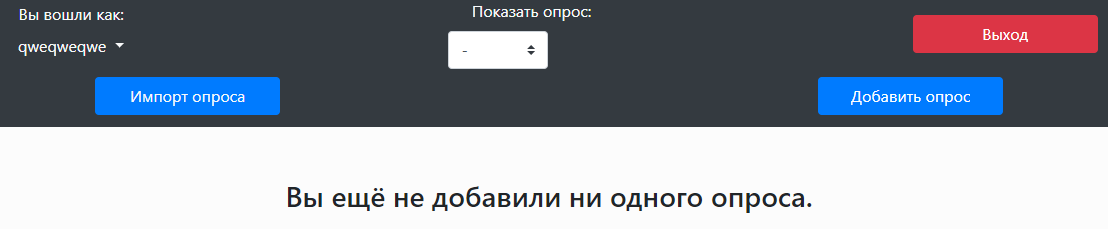


Рисунок 2.3.4 – Окно просмотра опросов нового пользователя

При нажатии на кнопку «Выход» пользователя перенаправит на страницу авторизации Страница авторизации представлена на рисунке 2.3.1.

Если на аккаунте пользователя нет опросов, то ему не доступно большинство функционала приложения. Первым делом пользователю необходимо создать новый опрос. Для этого нужно нажать на кнопку «Добавить опрос». Пользователя перенаправит на страницу конструирования опроса. Она представлена на рисунке 2.3.5. Первым делом на ней необходимо ввести наименование опроса.

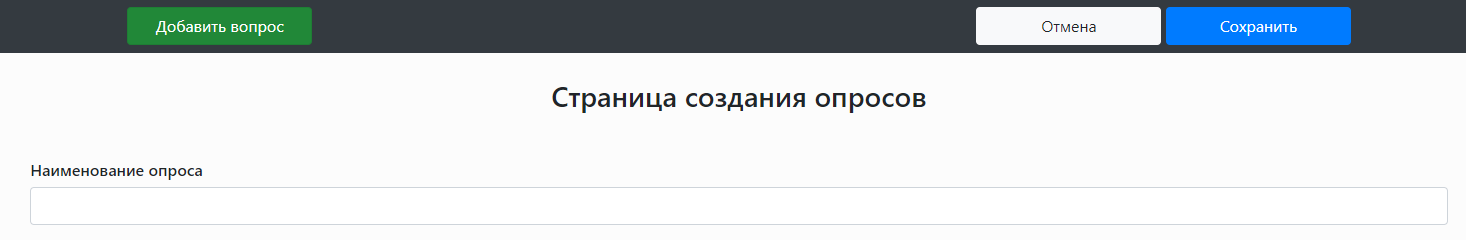


Рисунок 2.3.5 – Страница конструирования опроса

Для добавления нового вопроса необходимо нажать на кнопку «Добавить вопрос». После этого будет сгенерированы поля необходимые для создания вопроса. Они представлены на рисунке 2.3.6. В них можно выбрать номер вопроса, тип вопроса, текст вопроса и указать обязательный ли вопрос.

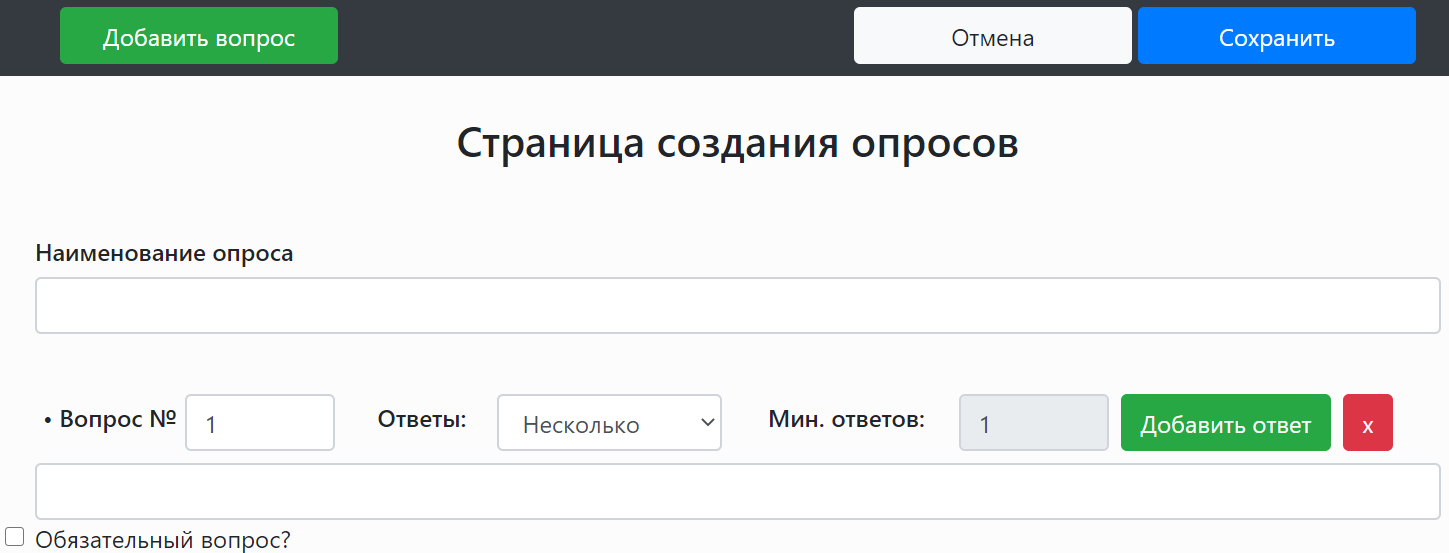


Рисунок 2.3.6 – Поля для создания вопроса

При нажатии на пиктограмму «Х» поля для этого вопроса будут удалены.

При нажатии на кнопку «Добавить ответ» для этого вопроса будут сгенерированы поля для ответа. Поля для ответа показаны на рисунке 2.3.7. В них можно выбрать номер ответа, текст ответа и указать будет ли добавлено поле для ввода для этого ответа.

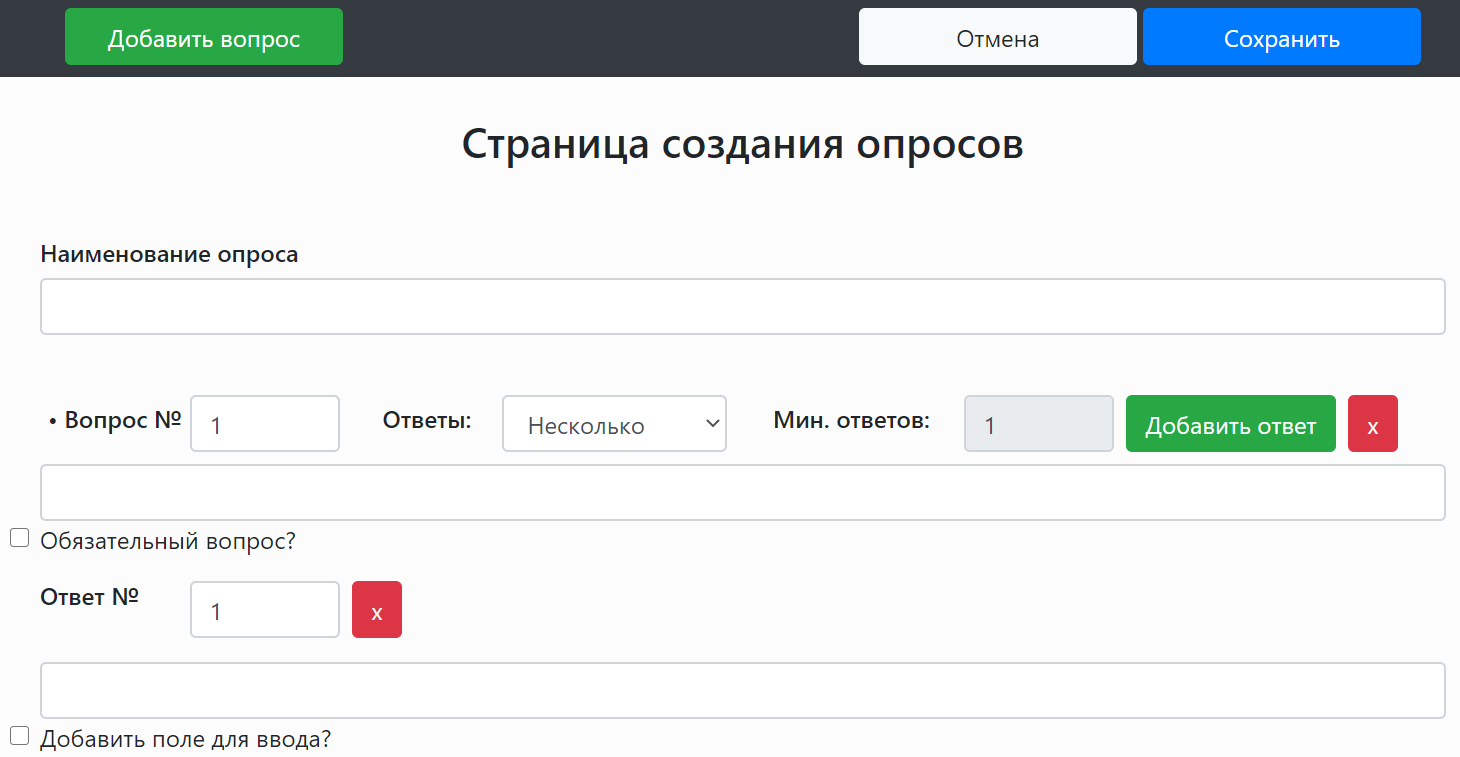


Рисунок 2.3.7 – Поля для создания ответа

При нажатии на пиктограмму «Х» поля для этого ответа будут удалены.

При нажатии на белую кнопку «Отмена» пользователя перенаправит на страницу просмотра опросов, а текущий опрос не будет сохранён.

При нажатии на синюю кнопку «Сохранить» пользователя перенаправит на страницу просмотра опросов, а текущий опрос будет сохранён.

Созданный опрос можно будет выбрать в списке опросов. Для этого в выпадающем списке «Показать опрос» необходимо выбрать номер созданного опроса.

После создания первого опроса, пользователю станет доступен весь функционал.

При нажатии на кнопку «Редактировать опрос» пользователя перенаправит на страницу редактирования опроса. Она представлена на рисунке 2.3.8.

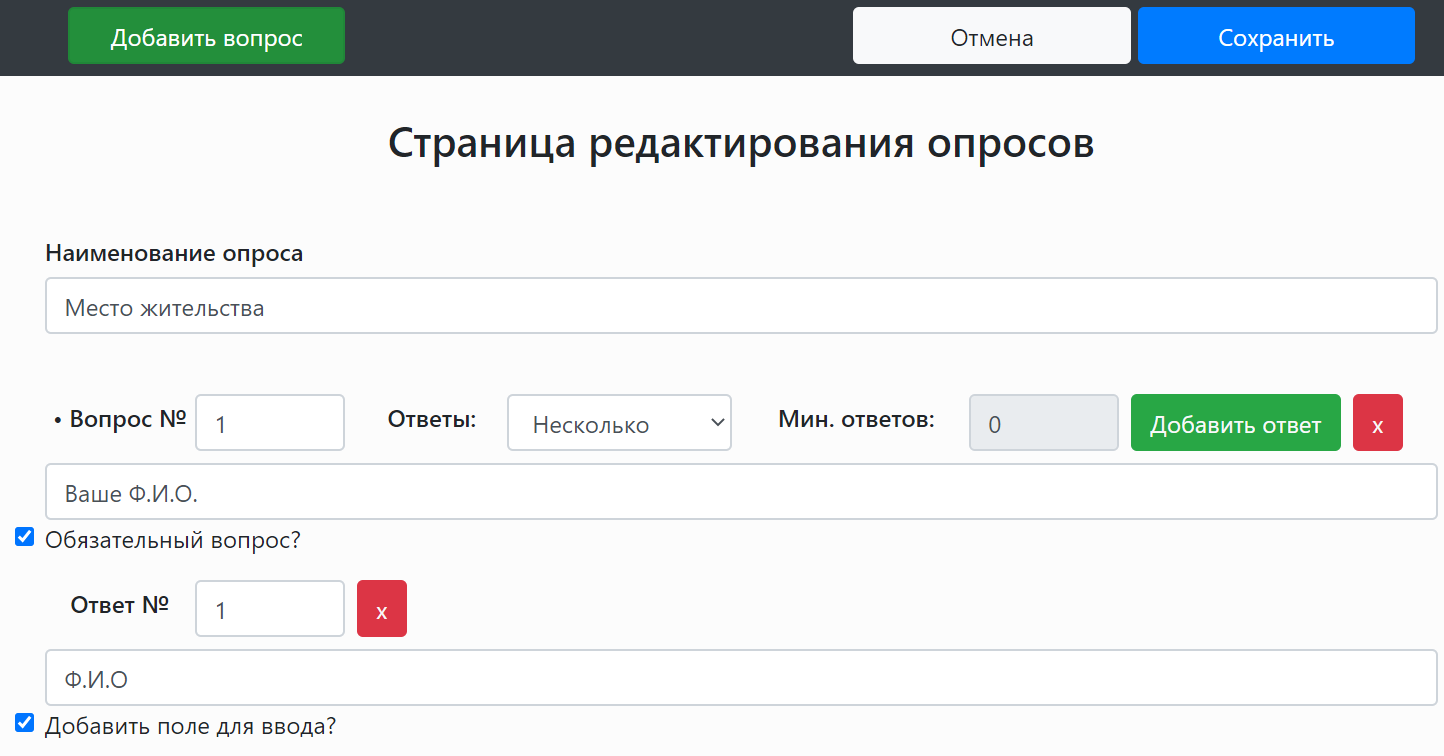


Рисунок 2.3.8 – Страница редактирования опроса

Весь функционал страницы редактирования повторяет функционал страницы добавления опроса.

При нажатии на кнопку «Удалить опрос» на странице просмотра опроса, пользователю будет предложено удалить выбранный опрос, либо же отменить действие. Окно удаления опроса представлено на рисунке 2.3.9.

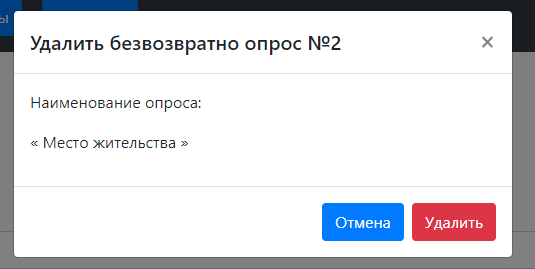


Рисунок 2.3.9 – Окно удаления опроса

При нажатии кнопки «Отмена» окно будет скрыто, а опрос не будет удалён.

При нажатии на кнопку «Удалить опрос» опрос будет удалён и пользователя перенаправит на страницу просмотра опросов.

При нажатии на кнопку «Опрашиваемые» на странице опроса, пользователя перенаправит на страницу работы с аккаунтами. Она представлена на рисунке 2.3.10. На ней пользователь может добавлять новые аккаунты и удалять уже созданные аккаунты.

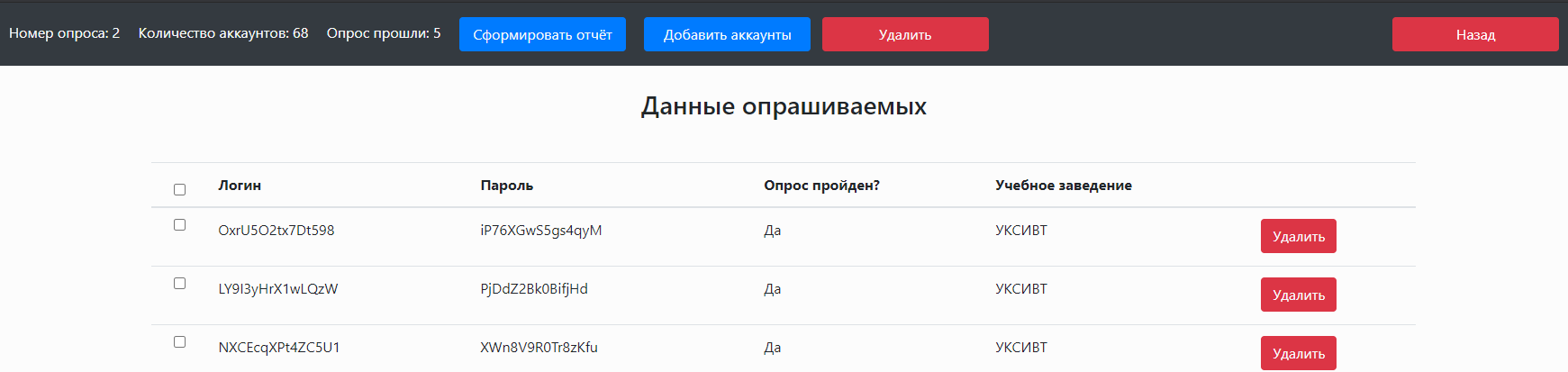


Рисунок 2.3.10 – Страница работы с аккаунтами

При нажатии на кнопку «Добавить аккаунты» пользователю будет предложено добавить аккаунты, либо отменить действие. Окно добавления аккаунтов представлено на рисунке 2.3.11. В этом окне необходимо указать количество аккаунтов, которое будет сгенерировано и учебное заведения для которого будут сгенерированы аккаунты.

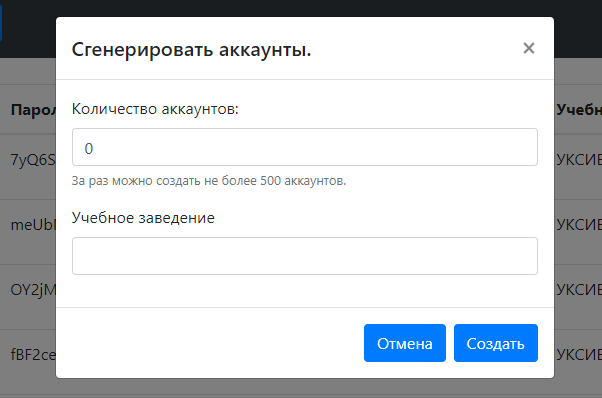


Рисунок 2.3.11 – Окно добавление аккаунтов

При нажатии на кнопку «Отмена» окно будет скрыто.

При нажатии на кнопку «Создать» будет сгенерирован pdf файл с логинами и аккаунтами для опрашиваемых. Он показан на рисунке 2.3.12.



Рисунок 2.3.12 – Сгенерированный pdf файл с аккаунтами опрашиваемых

После обновления страницы созданные аккаунты буду отображены в таблице.

Для того чтобы сформировать отчёт по результатам ответа опрашиваемых, администратору необходимо перейти на страницу просмотра опрашиваемых и нажать на кнопку «Сформировать отчёт». Она показана на рисунке 2.3.13. После этого автоматически будет сформирован и скачан отчёт по выбранному опросу. Отчёт представлен на рисунке 2.3.14.

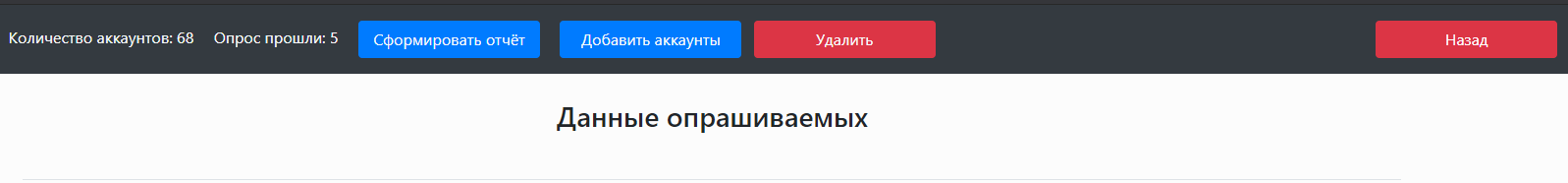


Рисунок 2.3.13 – Кнопка формирования отчёта

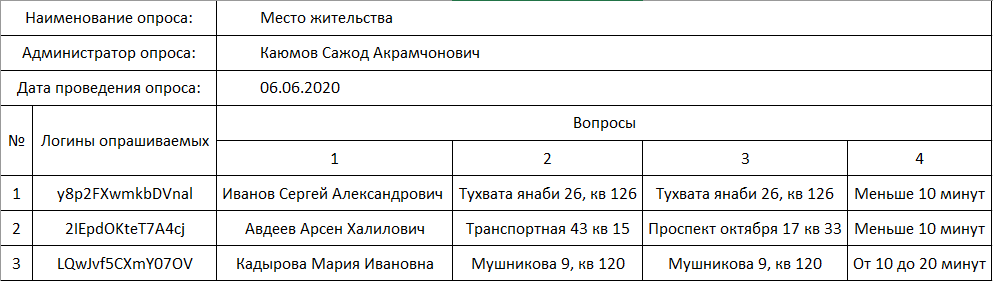


Рисунок 2.3.14 – Сформированный отчёт по опросу по контрольному примеру!

Для того чтобы передать опрос от одного администратора опроса другому или сделать резервную копию опроса необходимо перейти на страницу просмотра опросов и нажать кнопку «Экспорт опроса». Она представлена на рисунке 2.3.15.

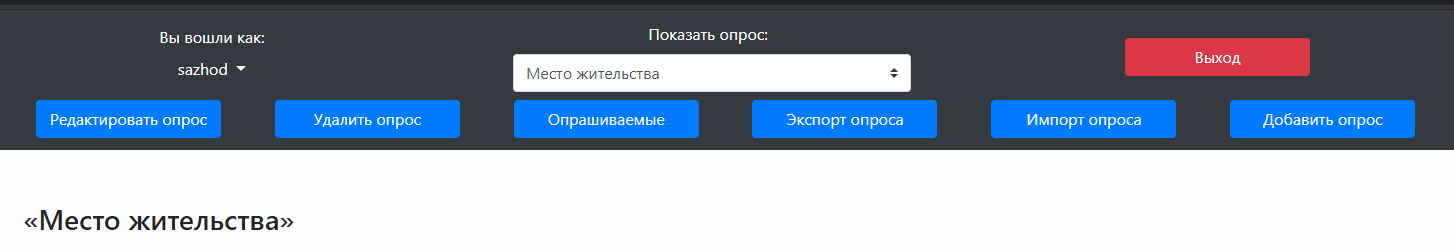


Рисунок 2.3.15 – Кнопки экспорта и импорта опросов

После этого автоматически будет скачан опрос в формате \*.json. Он показан на рисунке 2.3.16.

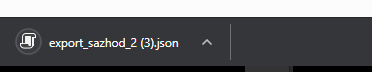


Рисунок 2.3.16 – Файл экспортированного опроса

Далее для того чтобы загрузить экспортированный опрос на сайт необходимо на странице просмотра опросов нажать на кнопку «Импортировать опрос». Эта кнопка показана на рисунке 2.3.15. После этого появиться окно для импортирования опроса, представленное на рисунке 2.3.17.

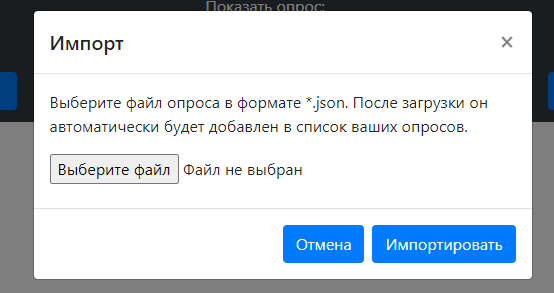


Рисунок 2.3.17 – Окно импорта опросов

Для того чтобы импортировать опрос в окне импорта необходимо выбрать файл опроса и нажать на кнопку «Импортировать». После этого опрос появиться в конце списка опросов. Это показано на рисунке 2.3.18.

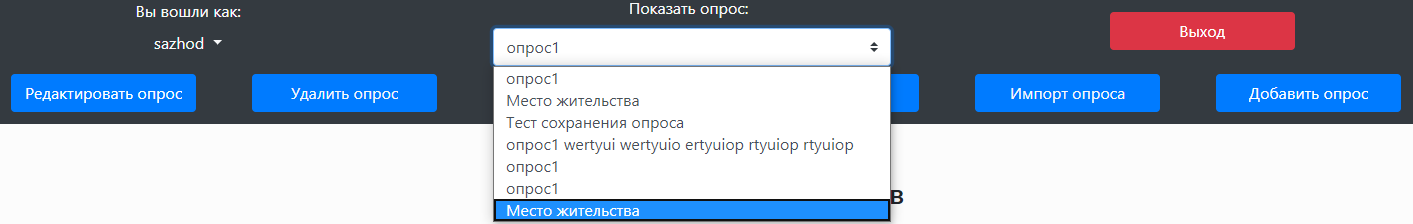


Рисунок 2.3.18 – Импортированный опрос

Руководство пользователей для опрашиваемых.

Перед началом прохождения опроса, опрашиваемому будут выданы данные от аккаунта.

Для того чтобы начать проходить опрос, у опрашиваемого должен быть доступ к любому устройству (ПК, планшет или телефон) с выходом в интернет. Далее необходимо перейти на страницу авторизации.

На странице авторизации опрашиваемому необходимо ввести данные от своего аккаунта и нажать на кнопку «Войти». Далее опрашиваемый будет перенаправлен на страницу прохождения опроса. Страница прохождения опроса представлена на рисунке 2.3.19.

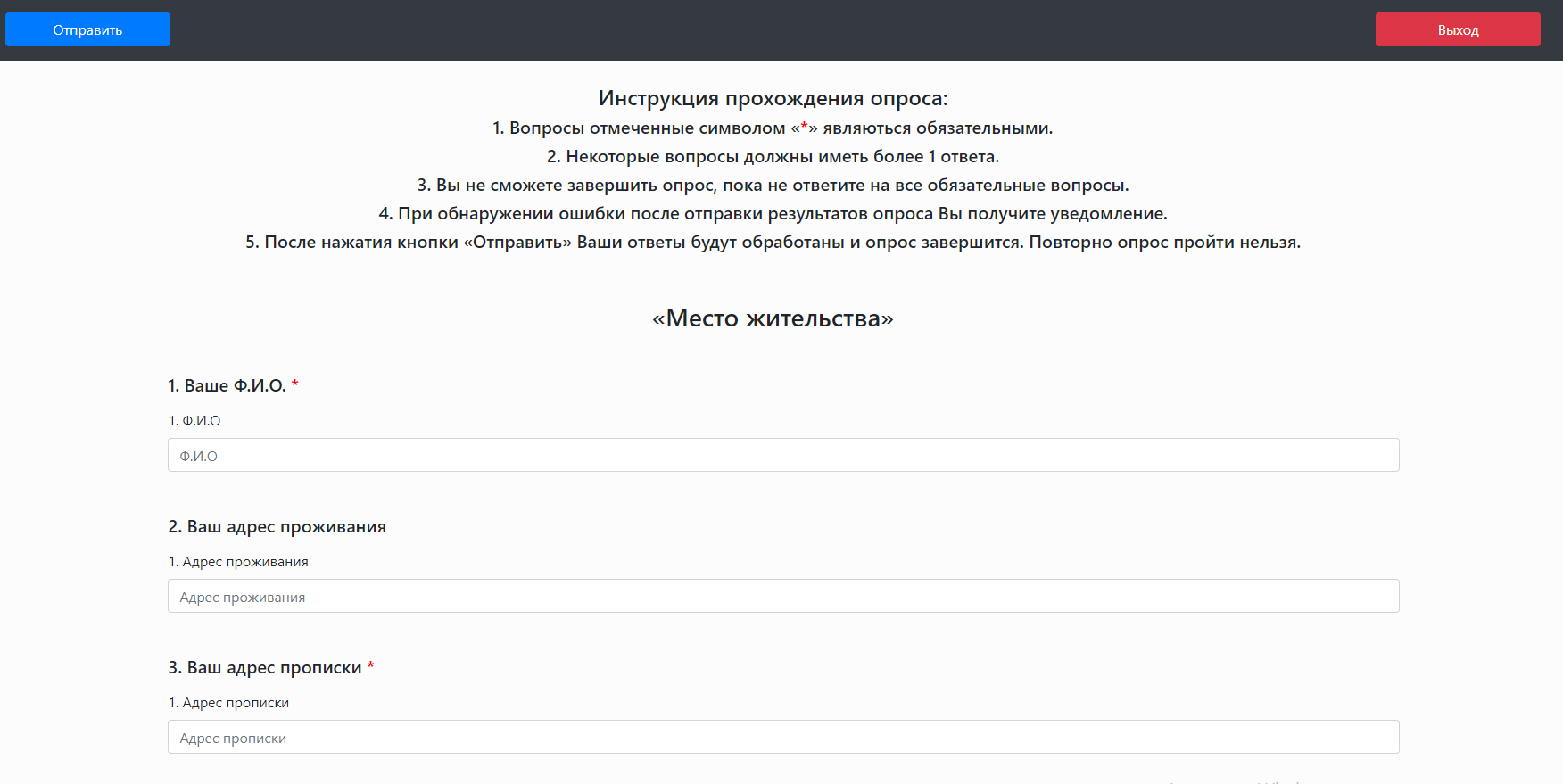


Рисунок 2.3.19 – Страница прохождения опроса

Перед тем как начать отвечать на вопросы, необходимо прочитать инструкцию по прохождению опроса. она представлена на рисунке 2.3.20

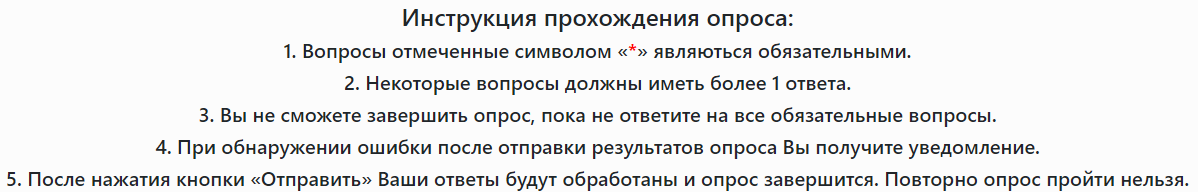


Рисунок 2.3.20 – Страница прохождения опроса

После заполнения всех обязательных вопросов опрашиваемый должен отправить результат опроса. Для этого необходимо нажать на кнопку «Отправить». Она показана на рисунке 2.3.21. Поле этого пользователя перенаправит на страницу с уведомлением о том, что опрос завершён. Она представлена на рисунке 2.3.22. После отправки результата, нельзя будет повторно пройти опрос.

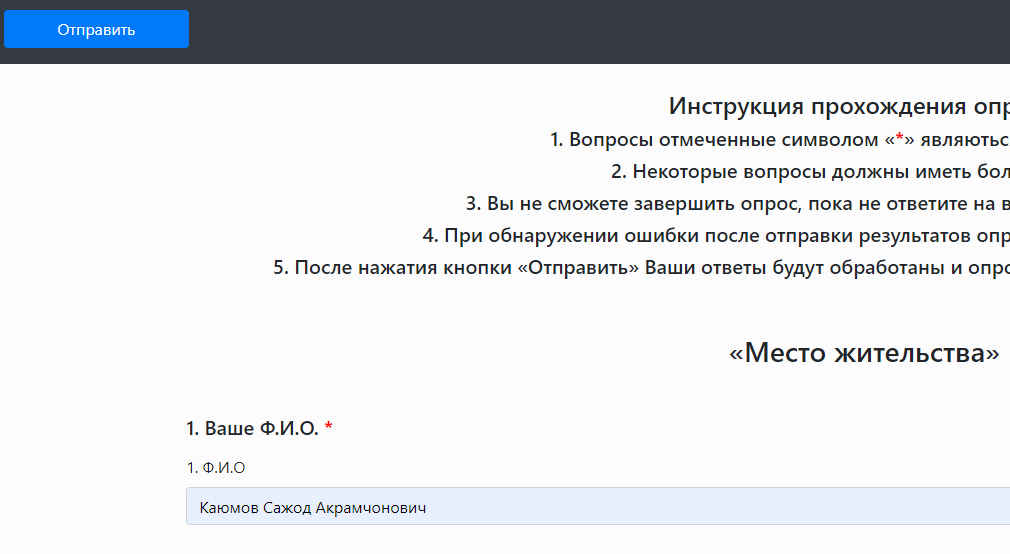


Рисунок 2.3.21 – Кнопка отправки ответов опрашиваемого

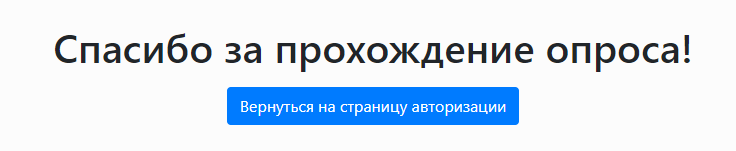


Рисунок 2.3.22 – Страница с сообщением о завершении опроса

3. Экономический раздел

3.1 Расчет затрат на создание программного продукта

Расчет себестоимости машинного часа эксплуатации вычислительной и оргтехники (ВиОТ):

, (3.1)

где См.ч. - себестоимость машинного часа;

Зi - годовые затраты, связанные с эксплуатацией и обслуживанием ВиОТ;

Fп - годовой полезный фонд времени работы единицы оборудования;

Кг = 0,95 - коэффициент готовности.

, (3.2)

где Fн = 1954 ч. – номинальный годовой фонд рабочего времени в РБ по 40-часовой рабочей недели в 2020 г.

αР = 0,125 - коэффициент, учитывающий время, затраченное на ремонт, настройки, обслуживание ВиОТ.

Fп=1954\* (1 - 0,125) = 1710 (ч.).

Расчет суммарных годовых затрат.

Для расчета годовых затрат, необходимо определить балансовую стоимость условного комплекта:

Таблица 3.1 - Состав условного комплекта

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование | Кол-во,  шт. | Цена,  руб. |
| 1 | Системный блок | 1 | 15000 |
| 2 | Монитор | 1 | 5000 |
| 3 | Клавиатура | 1 | 500 |
| 4 | Мышь | 1 | 300 |
|  | Итого: | 20800 | |

, (3.3)

где Цi - цена единицы условного комплекта;

Ki - количество единиц условного комплекта;

РД - дополнительные расходы на доставку, установку, первоначальную наладку.

 (3.4)

РД = 0,1\*(15000\*1+5000\*1+500\*1+300\*1) = 2080 (руб.).

СБАЛ = 15000\*1+5000\*1+500\*1+300\*1+500\*1+2080 = 22880 (руб.).

Затраты на материалы:

 (3.5)

ЗМ = 0,02\*22880 = 457,6 (руб.).

Основная и дополнительная заработная плата персонала, обслуживающего ВиОТ.

Предусмотрен 1 человек.

, (3.6)

где Оi - оклад i-го работника, обслуживающего ВиОТ;

Кд = 0,1 - коэффициент дополнительной заработной платы;

КР = 0,15 - районный коэффициент.

 = 12650 (руб.).

Начисления на заработную плату.

Таблица 3.2 - Состав единого страхового налога на 2019 год

|  |  |  |
| --- | --- | --- |
| № | Статья | % |
| 1 | 2 | 3 |
| 1 | Пенсионный фонд | 22 |
| 2 | Фонд социального страхования | 2,9 |

Продолжение таблицы 3.2

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 3 | Федеральный фонд обязательного медицинского страхования | 5,1 |
| 4 | Страховой налог от несчастного случая на производстве для служащих | 0,2 |
|  | Итого: | 30,2 |

 (3.7)

НЗ/П = 0,302\*12650 = 3820,3 (руб.).

Амортизационные отчисления ВиОТ.

, (3.8)

где  = 0,2 - норма амортизационных начислений.

АООБОР = 22880\*0,2 = 4576 (руб.).

Износ программных продуктов.

Условный комплект обладает необходимыми продуктами, представленными в таблице 3.3.

Таблица 3.3 - Используемые программные средства

|  |  |
| --- | --- |
| Наименование | Цена (руб.) |
| MS Windows 10 Pro | 8499 |
| MS Office 2016 | 4900 |
| MS Visual Studio Community 2019 | 0 |
| Google Chrome | 0 |
| Итого | 13399 |

Амортизационные отчисления программных продуктов - АОпп.

**,**  (3.9)

где  = 0,5 - норма амортизационных начислений;

 = 13399 руб. - суммарная стоимость программных продуктов.

**** = 13399\*0,5\*0,01 = 66,995 (руб.).

Расходы на содержание и эксплуатацию оборудования.

Расходы на содержание и эксплуатацию оборудования состоят из:

* затраты на ремонт и специальное обслуживание

; (3.10)

ЗРЕМ = 0,03\*22880= 686,4 (руб.).

* затраты на электрическую энергию

, (3.11)

где М = 1,055 кВт - суммарная мощность,

ЦкВт/ч = 3,06 руб. - стоимость кВт/ч,

КГ = 0,95 - коэффициент готовности.

ЗЭ = 1,055\*1710\*0,95\*3,06 = 5244,4 (руб.).

Расходы на содержание и эксплуатацию оборудования

 (3.12)

ЗРЭ = 686,4+5244,4 = 5930,8 (руб.).

Прочие расходы.

Так как на объекте используется один тип техники и количество обслуживающего персонала не превышает трех человек, то

 (3.13)

В нашем случае:



ЗПР = 0,5\*22880= 11440 (руб.).

Суммарные годовые затраты

 (3.14)

Зi = 457,6+12650+3820,3+4576+66,995+5930,8+11440= 39041,7 (руб.)

Себестоимость машинного часа, из формулы (3.1), составляет:

 = 39041,7/(1710\*0.95) = 24,03(руб.).

3.2 Расчет цены предложения

Фонд оплаты труда за время работы над программным продуктом.

, (3.15)

где О = 10000 руб. - оклад, работающего над продуктом;

ТОБ = 1 месяц - общее время работы над программным продуктом;

КР = 0,15 - районный коэффициент.

ФОТ = 10000\*1\*(1+0,15) = 11500 (руб.)

Начисления на ФОТ:

 (3.16)

НЗ/П = 11500 \* 0,302 = 3473 (руб.).

Затраты, связанные с эксплуатацией и обслуживанием ВиОТ:

, (3.17)

где ТМ = 1 месяц - машинное время работы над программным продуктом;

ЧРМ = 161 - число рабочих часов в месяце;

КИ = 0,9 - коэффициент использования оборудования;

N = 1 - количество условных комплектов.

ЗОВТ = 1\*161\*0,9\*1\*24,03= 3481,95 (руб.).

Затраты на специальные программные продукты (ЗСПП).

Специальные программные продукты не использовались.

ЗСПП = 0 (руб.).

Затраты на хозяйственные операции и нужды (ЗХН):

ЗХН = 0 (руб.).

Накладные расходы:

 (3.18)

РН = 0,45\*11500 = 5175 (руб.).

Полные затраты на разработку программного продукта:

 (3.19)

ЗПОЛ = 11500+3820,3+3481,95+0+0+5175 = 23997,25 (руб.).

Расчет установочной прибыли:

, (3.20)

где РУ = 20 % - установочная рентабельность.

ПРУ = 23997,25 \*0,2 = 4795,45 (руб.).

Расчет величины налога на добавленную стоимость (НДС):

 (3.21)

НДС = (23997,25 +4795,45)\*0,2 = 5758,54 (руб.).

Цена предложения разработанного программного продукта:

 (3.22)

ЦПР = 23997,25 +4795,45+5758,54 = 34551,24 (руб.).

ЗАКЛЮЧЕНИЕ

В процессе выполнения дипломного проекта была разработана структура и алгоритм работы web-приложения «PollDesigner». При этом были изучены особенности реализации компонентов информационной системы, а также решены следующие задачи:

* реализованы функции создания, изменения, удаления опросов;
* регистрация и добавление пользователей системы;
* возможность добавления ответов пользователя на опросы;
* построена справочная система.

Результатом работы стало создание web-приложения для автоматизации проведения опросов среди учебных учреждений «PollDesigner». Приложение написано на языках программирования C#, JavaScript, а также язык гипертекстовой разметки Html и каскадную таблицу стилей Css, с использованием среды разработки Visual Studio 2019, СУБД MSSQL, и технологий ASP.NET Core 3.0 EntityFrameworkCore, Linq и Bootstrap.

Приложение позволяет конструировать, редактировать и удалять опросы, а также создавать и удалять аккаунты для опрашиваемых. Также в приложении была реализована возможность формирования отчёта по результатам ответов опрашиваемых.

Для устранения ошибок в работе приложения были проведены тесты различных функций приложения, которое завершилось успешно.

С помощью приложения на основании данных контрольного примера были получены результаты, которые полностью совпадают с выходной информацией контрольного примера.

Результат работы программы PollDisigner, отчёт о прохождении опроса, представлен в приложении В. Текст программы - приложение Б.

Таблица А.1 - Выходные данные контрольного примера для отчётов по опросам

Приложение А

Выходные данные контрольного примера

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Отчёт по опросу | Администратор | Дата проведения опроса | Логин опрашиваемого | Вопрос | Ответ |
| 1 | 2 | 3 | 4 | 5 | 6 |
| Информация об учащихся | ivan | 13.04.2020 | ME8wiZ0JmMVQ6V6 | Сколько мне лет | Больше 18 |
| Пол | Мужской |
| Укажите своё учебное заведение | УКСИВТ |
| MSBOURIWPMs5Xes | Сколько мне лет | От 16 до 18 |
| Пол | Мужской |
| Укажите своё учебное заведение | УКСИВТ |
| uquKQOxJxik8qfB | Сколько мне лет | Больше 18 |
| Пол | Женский |
| Укажите своё учебное заведение | УКСИВТ |
| Место жительства | sazhod | 06.06.2020 | y8p2FXwmkbDVnal | Ваше Ф.И.О | Иванов Сергей Александрович |
| Ваш адрес проживания | Тухвата янаби 26, кв 126 |
| Ваш адрес прописки | Тухвата янаби 26, кв 126 |
| За сколько по времени вы доходите от дома до учебного заведения? | Меньше 10 минут |
| 2IEpdOKteT7A4cj | Ваше Ф.И.О | Авдеев Арсен Халилович |
| Ваш адрес проживания | Транспортная 43 кв 15 |
| Ваш адрес прописки | Проспект октября 17 кв 33 |
| За сколько по времени вы доходите от дома до учебного заведения? | Меньше 10 минут |
| LQwJvf5CXmY07OV | Ваше Ф.И.О | Кадырова Мария Ивановна |
| Ваш адрес проживания | Мушникова 9, кв 120 |
| Ваш адрес прописки | Мушникова 9, кв 120 |
| За сколько по времени вы доходите от дома до учебного заведения? | От 10 до 20 минут |
| Информация о родителе (опекуне) | oleg | 28.04.2020 | XbdegE8j2YKwFzc | Ф.И.О вашего родителя | Иванова Елена Николаевна |
| Пол родителя | Женский |
| Дата рождения родителя | 20.03.1975 |

Продолжение таблицы А.1

Продолжение приложения А

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |
|  |  |  |  | Номер телефона родителя | 89638573087 |
| wiH8Ub3gbltvc6T | Ф.И.О вашего родителя | Иванов Роберт Артурович |
| Пол родителя | Мужской |
| Дата рождения родителя | 11.09.1979 |
| Номер телефона родителя | 89647563589 |
| TAvmbNoAH3rXiFz | Ф.И.О вашего родителя | Ахметова Ирина Владимировна |
| Пол родителя | Женский |
| Дата рождения родителя | 06.11.1973 |
| Номер телефона родителя | 89683543546 |

Приложение Б

Исходный код программы

Модуль Startup

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.Http;

using Microsoft.Extensions.Configuration;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Hosting;

using Microsoft.EntityFrameworkCore;

using Microsoft.EntityFrameworkCore.SqlServer;

using Microsoft.Extensions.Logging;

using CourseProject.Data.Models;

using CourseProject.Data;

using CourseProject.Data.Interfaces;

using CourseProject.Data.Repository;

using Microsoft.AspNetCore.Authentication.Cookies;

using Microsoft.Extensions.FileProviders;

using System.IO;

namespace CourseProject

{

public class Startup

{

//Серверные настройки сайта

private IConfigurationRoot \_confstring;

public Startup(IHostEnvironment hostEnv)

{

\_confstring = new ConfigurationBuilder().SetBasePath(hostEnv.ContentRootPath).AddJsonFile("dbsettings.json").Build();

}

public void ConfigureServices(IServiceCollection services)

{

services.AddRazorPages().AddRazorRuntimeCompilation();

//Подключение БД

services.AddDbContext<AppDbContent>(options => options.UseSqlServer(\_confstring.GetConnectionString("DefaultConnection")));

services.AddSingleton<IHttpContextAccessor, HttpContextAccessor>();

services.AddSingleton<IFileProvider>(new PhysicalFileProvider(Path.Combine(Directory.GetCurrentDirectory(), "wwwroot")));

//services.AddScoped(sp => ShopCart.GetCart(sp));

//Подключение репозиториев

services.AddTransient<IPolls, PollRepository>();

services.AddTransient<IPollsCreator, CreatorRepository>();

services.AddTransient<IUsersCreator, UserRepository>();

services.AddTransient<IAnswer, AnswerRepository>();

Продолжение приложения Б

services.AddMvc(option => option.EnableEndpointRouting = false);

services.AddAuthentication(CookieAuthenticationDefaults.AuthenticationScheme)

.AddCookie(options => //CookieAuthenticationOptions

{

options.LoginPath = new PathString("/Account/Login");

});

services.AddMemoryCache();

services.AddSession();

}

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

app.UseDeveloperExceptionPage();

app.UseBrowserLink();

app.UseStatusCodePages();

app.UseStaticFiles();

app.UseAuthentication();

app.UseAuthorization();

//app.UseMvcWithDefaultRoute();

app.UseSession();

app.UseMvc(routes =>

{

routes.MapRoute(name: "default", template: "{controller=Account}/{action=Login}/{link?}");

});

using (var scope = app.ApplicationServices.CreateScope())

{

AppDbContent content = scope.ServiceProvider.GetRequiredService<AppDbContent>();

DbObjects.Initial(content);

}

}

}

}

Модуль AccountController

using CourseProject.Data.Interfaces;

using CourseProject.Data.Models;

using CourseProject.Services;

using CourseProject.ViewModels;

using Microsoft.AspNetCore.Authentication;

using Microsoft.AspNetCore.Authentication.Cookies;

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Security.Claims;

using System.Threading.Tasks;

namespace CourseProject.Controllers

Продолжение приложения Б

{

public class AccountController : Controller

{

private readonly IPollsCreator \_allCreators;

private readonly IUsersCreator \_allUser;

//Конструктор

public AccountController(/\*IAllPolls iAllPolls,\*/ IPollsCreator iAllCreators, IUsersCreator iAllUser)

{

//\_allPolls = iAllPolls;

\_allCreators = iAllCreators;

\_allUser = iAllUser;

}

//Авторизация пользователей

[HttpGet]

public IActionResult Login()

{

if (Request.Cookies.ContainsKey("Login"))

{

Response.Cookies.Delete("Login");

}

return View(new LoginViewModel());

}

[HttpPost]

[ValidateAntiForgeryToken]

public IActionResult Login(LoginViewModel model)

{

if (ModelState.IsValid)

{

Creator creator = \_allCreators.AllCreators.FirstOrDefault(u => u.login == model.Login && u.password == model.Password);

if (creator != null)

{

if(creator.isVerified == true)

{

Authenticate(model.Login); // аутентификация

return RedirectToAction("Questioning", "MainPage");

}

else

{

ModelState.AddModelError("", "Вы ещё не подтвердили свой аккаунт, проверьте свою почту.");

return View(model);

}

}

else

{

User user = \_allUser.GetUser(model.Login, model.Password);

if(user != null)

Продолжение приложения Б

{

if (user.isPassed == 0)

{

Authenticate(model.Login);

Response.Cookies.Append("creatorId", user.Creator.id.ToString());

return RedirectToAction("PassingPoll", "MainPage");

}

else

{

Authenticate(model.Login);

Response.Cookies.Append("Login", model.Login);

return RedirectToAction("CompletionPoll", "MainPage", new { isReEntry = true});

}

}

}

ModelState.AddModelError("", "Некорректные логин и(или) пароль");

}

return View(model);

}

//Регистрация администратора опросов

[HttpGet]

public IActionResult Register()

{

return View(new RegisterViewModel());

}

[HttpPost]

[ValidateAntiForgeryToken]

public IActionResult Register(RegisterViewModel model)

{

if (ModelState.IsValid)

{

Creator creator = \_allCreators.AllCreators.FirstOrDefault(u => u.login == model.Login);

if (creator == null)

{

if(\_allCreators.AllCreators.FirstOrDefault(u => u.email == model.Email) == null)

{

EmailService emailService = new EmailService();

emailService.SendEmail(model.Email, "Подтверждение аккаунта",

$"Для подтверждения аккаунта вам необходимо перейти по данной ссылке:\nhttps://localhost:44328/Account/VerifiedAccount?creatorLogin={model.Login}. Если вы не регистрировались на сайте, то проигнорируйте это письмо."

);

// добавляем пользователя в бд

\_allCreators.AddCreator(new Creator { login = model.Login, password = model.Password, email = model.Email, firstname = model.FirstName, name = model.Name, lastname = model.LastName, isVerified = false });

return RedirectToAction("VerifiedMessage", "Account");

}

else

{

Продолжение приложения Б

ModelState.AddModelError("", "Введённая Почта уже используется!");

model.Email = "";

}

}

else

{

ModelState.AddModelError("", "Данный логин уже используется!");

model.Login = "";

}

}

else

{

if (model.Password.Equals(model.ConfirmPassword))

{

ModelState.AddModelError("", "Введёна некорректная Почта!");

model.Email = "";

}

else

{

ModelState.AddModelError("", "Пароли не совпадают!");

model.ConfirmPassword = "";

}

}

return View(model);

}

//Аутентификация

private void Authenticate(string userName)

{

// создаем один claim

var claims = new List<Claim>

{

new Claim(ClaimsIdentity.DefaultNameClaimType, userName)

};

// создаем объект ClaimsIdentity

ClaimsIdentity id = new ClaimsIdentity(claims, "ApplicationCookie", ClaimsIdentity.DefaultNameClaimType, ClaimsIdentity.DefaultRoleClaimType);

// установка аутентификационных куки

HttpContext.SignInAsync(CookieAuthenticationDefaults.AuthenticationScheme, new ClaimsPrincipal(id));

}

//Выход пользователя

public IActionResult Logout()

{

HttpContext.SignOutAsync(CookieAuthenticationDefaults.AuthenticationScheme);

return RedirectToAction("Login", "Account");

}

//Подтверждение аккаунта администратора опросов

public IActionResult VerifiedAccount(string creatorLogin)

{

Creator creator = \_allCreators.AllCreators.First(c => c.login == creatorLogin);

creator.isVerified = true;

Продолжение приложения Б

\_allCreators.UpdateCreator(creator);

//Authenticate(creatorLogin);

return RedirectToAction("Login", "Account");

}

//Отображение страницы подтверждения Email

public IActionResult VerifiedMessage(string creatorLogin)

{

return View();

}

//Подтверждение аккаунта

[HttpGet]

public IActionResult CheckEmail()

{

return View(new CheckEmailViewModel());

}

//Проверка Email для восстановления пароля

[HttpPost]

public IActionResult CheckEmail(CheckEmailViewModel model)

{

if (ModelState.IsValid)

{

Creator creator = \_allCreators.AllCreators.FirstOrDefault(c => c.email == model.email);

if (creator != null)

{

EmailService emailService = new EmailService();

emailService.SendEmail(model.email, "Восстановление пароля",

$"Для восстановления пароля вам необходимо перейти по данной ссылке:\nhttps://localhost:44328/Account/RestorePassword?email={model.email}. Если вы не восстанавливаете пароль на сайте, то проигнорируйте это письмо."

);

model.isCorrectEmail = true;

return View(model);

}

else

{

ModelState.AddModelError("", "К указанной почте не привязан аккаунт!");

}

}

else

{

ModelState.AddModelError("", "Не верный формат Эл. почты!");

}

return View(model);

}

//Восстановление пароля

[HttpGet]

public IActionResult RestorePassword(string email)

Продолжение приложения Б

{

Response.Cookies.Append("email", email);

return View(new RestorePasswordViewModel());

}

[HttpPost]

public IActionResult RestorePassword(RestorePasswordViewModel model)

{

if (ModelState.IsValid)

{

Creator creator = \_allCreators.AllCreators.FirstOrDefault(c => c.email == Request.Cookies["email"]);

creator.password = model.Password;

\_allCreators.UpdateCreator(creator);

model.isPasswordСhanged = true;

return View(model);

}

else

{

ModelState.AddModelError("", "Пароли не совпадают!");

}

return View(model);

}

//Изменение Данных администратора опросов

[Authorize]

[HttpGet]

public IActionResult EditEmail()

{

Creator creator = \_allCreators.AllCreators.FirstOrDefault(c => c.login == User.Identity.Name);

return View(new EditEmailViewModel() { email = creator.email});

}

[Authorize]

[HttpPost]

public IActionResult EditEmail(EditEmailViewModel model)

{

if (ModelState.IsValid)

{

Creator creator = \_allCreators.AllCreators.FirstOrDefault(c => c.login == User.Identity.Name);

if (model.password == creator.password)

{

if(model.email != creator.email)

{

if (\_allCreators.AllCreators.FirstOrDefault(c => c.email == model.email) == null)

{

creator.email = model.email;

\_allCreators.UpdateCreator(creator);

Response.Cookies.Append("action", "email");

return RedirectToAction("Questioning", "MainPage");

}

else

{

ModelState.AddModelError("", "Введённая почта уже используется!");

}

}

Продолжение приложения Б

else

{

ModelState.AddModelError("", "Введена текущая почта!");

}

}

else

{

ModelState.AddModelError("", "Введён неправильный пароль!");

}

}

else

{

ModelState.AddModelError("", "Введёна некорректная Почта!");

model.email = "";

}

return View(model);

}

[Authorize]

[HttpGet]

public IActionResult EditFullName()

{

Creator creator = \_allCreators.AllCreators.FirstOrDefault(c => c.login == User.Identity.Name);

return View(new EditFullNameViewModel() { firstName = creator.firstname, name = creator.name, lastName = creator.lastname});

}

[Authorize]

[HttpPost]

public IActionResult EditFullName(EditFullNameViewModel model)

{

Creator creator = \_allCreators.AllCreators.FirstOrDefault(c => c.login == User.Identity.Name);

creator.firstname = model.firstName;

creator.name = model.name;

creator.lastname = model.lastName;

\_allCreators.UpdateCreator(creator);

Response.Cookies.Append("action", "fullname");

return RedirectToAction("Questioning", "MainPage");

}

[Authorize]

[HttpGet]

public IActionResult EditPassword()

{

return View(new EditPasswordViewModel());

}

[Authorize]

[HttpPost]

public IActionResult EditPassword(EditPasswordViewModel model)

{

if (ModelState.IsValid)

{

Creator creator = \_allCreators.AllCreators.FirstOrDefault(c => c.login == User.Identity.Name);

if (creator.password == model.oldPassword)

Продолжение приложения Б

{

if (creator.password != model.newPassword)

{

creator.password = model.newPassword;

\_allCreators.UpdateCreator(creator);

Response.Cookies.Append("action", "password");

return RedirectToAction("Questioning", "MainPage");

}

else

{

ModelState.AddModelError("", "Старый пароль и новый совпадают!");

}

}

else

{

ModelState.AddModelError("", "Старый пароль введён не правильно!");

}

}

else

{

ModelState.AddModelError("", "Пароли не совпадают!");

}

return View(model);

}

}

}

Модуль MainPageController

using CourseProject.Data.Interfaces;

using CourseProject.Data.Models;

using CourseProject.ViewModels;

using Microsoft.AspNetCore.Authentication;

using Microsoft.AspNetCore.Authentication.Cookies;

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Configuration;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.Serialization.Json;

using System.Text;

using System.Text.Encodings.Web;

using System.Text.Json;

using System.Text.Unicode;

using System.Threading.Tasks;

namespace CourseProject.Controllers

{

public class MainPageController : Controller

{

private readonly IPolls \_poll;

private readonly IAnswer \_answer;

Продолжение приложения Б

private readonly IUsersCreator \_allUser;

private readonly IPollsCreator \_allCreator;

//Конструктор класса

public MainPageController(IPolls iPoll, IUsersCreator iUserCreator, IPollsCreator iPollsCreator, IAnswer iAnswer)

{

\_poll = iPoll;

\_answer = iAnswer;

\_allUser = iUserCreator;

\_allCreator = iPollsCreator;

}

//Отображение страницы просмотра опросов для администратора опросов

[Authorize]

public ViewResult Questioning()

{

ViewData["action"] = "";

if (Request.Cookies.ContainsKey("action"))

{

ViewData["action"] = Request.Cookies["action"];

Response.Cookies.Delete("action");

}

return View(\_poll.GetPolls(User.Identity.Name));

}

//Прохождение опроса опрашиваемыми

[Authorize]

public ViewResult PassingPoll()

{

string userLogin = User.Identity.Name;

User user = \_allUser.GetUser(userLogin);

Creator creator = new Creator();

if (user.Creator != null)

{

creator = \_allCreator.GetCreator(user.Creator.id);

}

else

{

creator = \_allCreator.GetCreator(Convert.ToInt32(Request.Cookies["creatorId"]));

}

tPoll poll = \_poll.GetPoll(creator.login, user.pollNumber);

var pollObject = new PassingPollViewModel

{

user = user,

creator = creator,

poll = poll

};

return View(pollObject);

}

Продолжение приложения Б

//Заверншение опроса

[Authorize]

public ViewResult CompletionPoll(PassingPollViewModel model, bool isReEntry = false)

{

string userLogin = User.Identity.Name;

if (!isReEntry)

{

if (ProcessingAnswer(model.userAnswer))

{

\_allUser.UpdateUser(userLogin);

ViewBag.Title = "Спасибо за прохождение опроса!";

}

else

{

ViewBag.Title = "При обработке данных произошла ошибка!";

}

}

else

{

ViewBag.Title = "Повторное прохождение опроса запрещено!";

}

HttpContext.SignOutAsync(CookieAuthenticationDefaults.AuthenticationScheme);

ViewBag.Message = $"Опрашиваемый {userLogin} прошёл опрос {\_allUser.GetPassingDate(userLogin)}";

return View();

}

//Обработка ответов опрашиваемого

[Authorize]

public bool ProcessingAnswer(tAnswer userAnswer)

{

tAnswer answer = new tAnswer();

answer = userAnswer;

for (int i = 0; i < answer.QuestionsLists.Count; i++)

{

for (int j = 0; j < answer.QuestionsLists[i].ResponsesLists.Count; j++)

{

if (answer.QuestionsLists[i].ResponsesLists[j].isChecked == "false")

{

answer.QuestionsLists[i].ResponsesLists.RemoveAt(j);

j--;

}

}

}

return \_answer.SaveAnswer(answer);

}

//Экспорт опроса

[Authorize]

public IActionResult ExportPoll(string creatorLogin, int pollNumber)

Продолжение приложения Б

{

string fileName = $"export\_{creatorLogin}\_{pollNumber}.json";

string exportJson = \_poll.ExportPoll(\_poll.GetPoll(creatorLogin, pollNumber));

System.IO.File.WriteAllText($@"Data\{fileName}", exportJson);

var path = Path.Combine(Directory.GetCurrentDirectory(), "Data", fileName);

var memory = new MemoryStream();

using (var stream = new FileStream(path, FileMode.Open))

{

stream.CopyTo(memory);

}

memory.Position = 0;

System.IO.File.Delete($@"Data\{fileName}");

return File(memory, "APPLICATION/octet-stream", Path.GetFileName(path));

}

//Импорт опроса

[Authorize]

public async Task<IActionResult> ImportPoll(string creatorLogin, IFormFile uploadedFile)

{

if (uploadedFile != null)

{

string path = $@"Data\{uploadedFile.FileName}";

using (var fileStream = new FileStream(path, FileMode.Create))

{

await uploadedFile.CopyToAsync(fileStream);

}

tPoll importPoll = \_poll.ImportPoll(path);

System.IO.File.Delete(path);

if (\_poll.SaveAddPoll(importPoll))

{

}

}

return RedirectToAction("Questioning", new { creatorLogin });

}

[Authorize]

public ActionResult Details(string creatorLogin, int numberPoll)

{

tPoll poll = \_poll.GetPolls(creatorLogin).Find(p => p.Number == numberPoll);

return PartialView(poll);

}

}

}

Модуль EditPollController

using CourseProject.Data.Interfaces;

using CourseProject.Data.Models;

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Rendering;

using System;

Продолжение приложения Б

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace CourseProject.Controllers

{

public class EditPollController : Controller

{

private readonly IPolls \_poll;

private readonly IUsersCreator \_user;

private readonly IAnswer \_answer;

//Конструктор класса

public EditPollController(IPolls iPoll, IUsersCreator iUsersCreator, IAnswer iAnswer)

{

\_poll = iPoll;

\_user = iUsersCreator;

\_answer = iAnswer;

}

//Добавление нового опроса

[Authorize]

public ViewResult Add()

{

string addPollLogin = User.Identity.Name;

int number = \_poll.GetPolls(addPollLogin)[\_poll.GetPolls(addPollLogin).Count - 1].Number + 1;

tPoll newPoll = new tPoll() { CreatorLogin = addPollLogin, Number = number };

return View(newPoll);

}

//Сохранение нового опроса

[Authorize]

public IActionResult SaveAdd(tPoll addedPoll)

{

if (\_poll.SaveAddPoll(addedPoll))

{

return RedirectToAction("Questioning", "MainPage");

}

return View();

}

///Редактирование опроса

[Authorize]

public ViewResult Edit(int editPollNumber)

{

string editPollLogin = User.Identity.Name;

tPoll editPoll = \_poll.GetPolls(editPollLogin).Find(p => p.Number == editPollNumber);

if(editPoll != null)

{

return View(editPoll);

}

else

{

return View(new tPoll());

}

}

Продолжение приложения Б

//Сохранение изменения опроса

[Authorize]

[HttpPost]

public IActionResult SaveEdit(tPoll changedPoll)

{

if (changedPoll.tQuestion != null)

{

for (int i = 0; i < changedPoll.tQuestion.Count; i++)

{

if (String.IsNullOrEmpty(changedPoll.tQuestion[i].IsRequired))

{

changedPoll.tQuestion[i].IsRequired = "false";

}

if (changedPoll.tQuestion[i].tResponse != null)

{

for (int j = 0; j < changedPoll.tQuestion[i].tResponse.Count; j++)

{

if (String.IsNullOrEmpty(changedPoll.tQuestion[i].tResponse[j].UseTextBox))

{

changedPoll.tQuestion[i].tResponse[j].UseTextBox = "false";

}

}

}

}

if (\_poll.SaveEditPoll(changedPoll))

{

return RedirectToAction("Questioning", "MainPage");

}

}

return View();

}

//Удаление опроса

[Authorize]

public IActionResult Delete(string deletePollLogin, int deletePollNumber)

{

List<User> users = \_user.GetUsers(deletePollLogin, deletePollNumber).ToList();

for(int i = 0; i < users.Count; i++)

{

\_answer.DeleteAnswer(users[i].login);

\_user.DeleteUser(users[i].login);

}

tPoll deletedPoll = \_poll.GetPolls(deletePollLogin).Find(p => p.Number == deletePollNumber);

\_poll.DeletePoll(deletedPoll);

return RedirectToAction("Questioning","MainPage");

}

}

}

Продолжение приложения Б

Модуль UserController

using CourseProject.Data.Interfaces;

using CourseProject.Data.Models;

using CourseProject.ViewModels;

using iTextSharp.text;

using iTextSharp.text.pdf;

using Microsoft.AspNetCore.Mvc;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using ClosedXML;

using DocumentFormat.OpenXml;

using DocumentFormat.OpenXml.Packaging;

using DocumentFormat.OpenXml.Spreadsheet;

using System.Threading.Tasks;

using ClosedXML.Excel;

namespace CourseProject.Controllers

{

public class UserController : Controller

{

private readonly IUsersCreator \_usersCreator;

private readonly IPollsCreator \_allCreators;

private readonly IAnswer \_answer;

private readonly IPolls \_poll;

//Конструктор класса

public UserController(IUsersCreator iUsersCreator, IPollsCreator iAllCreators, IAnswer iAnswer, IPolls iPolls)

{

\_usersCreator = iUsersCreator;

\_allCreators = iAllCreators;

\_answer = iAnswer;

\_poll = iPolls;

}

//Отображение опрашиваемых

public IActionResult ShowUser(int pollNumber)

{

IEnumerable<User> user = \_usersCreator.GetUsers(User.Identity.Name, pollNumber);

var userObject = new UserViewModel

{

User = user,

creatorLogin = User.Identity.Name,

pollNumber = pollNumber

};

return View(userObject);

}

//Добавление опрашиваемых

public IActionResult AddUsers(UserViewModel userViewModel)

{

Продолжение приложения Б

List<User> users = new List<User>();

string filename = $@"users {DateTime.Now.ToString("dd\_MM\_yyyy hh\_mm")}.pdf";

var document = new Document(PageSize.A4, 20, 20, 30, 20);

using (var writer = PdfWriter.GetInstance(document, new FileStream($@"wwwroot\{filename}", FileMode.Create)))

{

document.Open();

int n = 0;

int x = userViewModel.countUser / 20;

int residue = userViewModel.countUser % 20;

if (residue == 0)

x--;

var helvetica = new iTextSharp.text.Font(iTextSharp.text.Font.FontFamily.HELVETICA, 12);

var helveticaBase = helvetica.GetCalculatedBaseFont(false);

for (int j = 0; j <= x; j++)

{

document.NewPage();

writer.DirectContent.SetFontAndSize(helveticaBase, 12f);

document.Add(new Paragraph($"Page\_{j + 1}", helvetica));

writer.DirectContent.ShowTextAligned(iTextSharp.text.Element.ALIGN\_LEFT, "Login", 100, 766, 0);

writer.DirectContent.ShowTextAligned(iTextSharp.text.Element.ALIGN\_LEFT, "Password", 300, 766, 0);

int countIteration = 20;

if(j == x && residue != 0)

{

countIteration = residue;

}

for (int i = 0; i < countIteration; i++)

{

writer.DirectContent.BeginText();

User \_user = createUser(userViewModel.institution, userViewModel.creatorLogin, userViewModel.pollNumber);

users.Add(\_user);

writer.DirectContent.ShowTextAligned(iTextSharp.text.Element.ALIGN\_LEFT, $"{n + 1})", 50, 736 - (i \* 30), 0);

writer.DirectContent.ShowTextAligned(iTextSharp.text.Element.ALIGN\_LEFT, \_user.login, 100, 736 - (i \* 30), 0);

writer.DirectContent.ShowTextAligned(iTextSharp.text.Element.ALIGN\_LEFT, \_user.password, 300, 736 - (i \* 30), 0);

Продолжение приложения Б

writer.DirectContent.EndText();

n++;

}

}

document.Close();

writer.Close();

}

\_usersCreator.AddUser(users);

return RedirectToAction("Download", new { filename = filename, creatorLogin = userViewModel.creatorLogin, pollNumber = userViewModel.pollNumber });

}

//Скачивание файла с сервера

public IActionResult Download(string filename, string creatorLogin, int pollNumber)

{

if (filename == null)

return Content("filename not present");

var path = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot", filename);

var memory = new MemoryStream();

using (var stream = new FileStream(path, FileMode.Open))

{

stream.CopyTo(memory);

}

memory.Position = 0;

System.IO.File.Delete($@"wwwroot\{filename}");

return File(memory, "APPLICATION/octet-stream", Path.GetFileName(path));

}

//Генерация аккаунтов опрашиваемых

public User createUser(string institution, string creatorLogin, int pollNumber)

{

string randomLogin = "";

char charLogin = '0';

string randomPassword = "";

char charPassword = '0';

Random random = new Random();

int i = 0;

while (true)

{

i = 0;

while (i < 15)

{

charLogin = (char)random.Next(33, 125);

if (Char.IsLetterOrDigit(charLogin))

{

randomLogin += charLogin;

i++;

Продолжение приложения Б

}

}

if (\_usersCreator.GetUser(randomLogin) != null)

{

randomLogin = "";

}

else

{

break;

}

}

i = 0;

while (i < 15)

{

charPassword = (char)random.Next(33, 125);

if (Char.IsLetterOrDigit(charPassword))

{

randomPassword += charPassword;

i++;

}

}

return new User { login = randomLogin, password = randomPassword, institution = institution, isPassed = 0, Creator = \_allCreators.AllCreators.First(c => c.login == creatorLogin), pollNumber = pollNumber };

}

//Удаление опрашиваемых

public IActionResult DeleteUser(string creatorLogin, int pollNumber, string userLogin)

{

\_answer.DeleteAnswer(userLogin);

\_usersCreator.DeleteUser(userLogin);

return RedirectToAction("ShowUser", new { pollNumber });

}

//Удаление списка опрашиваемых

public IActionResult DeleteUsers(string creatorLogin, int pollNumber, string deleteUserList)

{

string[] userLogin = deleteUserList.Split(',');

for(int i = 0; i < userLogin.Length; i++)

{

\_answer.DeleteAnswer(userLogin[i]);

\_usersCreator.DeleteUser(userLogin[i]);

}

return RedirectToAction("ShowUser", new { pollNumber });

}

//Формирование опроса

public IActionResult CreateReport(UserViewModel userViewModel)

{

string fileName = $@"report {DateTime.Now.ToString("dd\_MM\_yyyy hh\_mm")}.xlsx";

string filePath = $@"wwwroot\{fileName}";

Продолжение приложения Б

tPoll poll = \_poll.GetPoll(userViewModel.creatorLogin, userViewModel.pollNumber);

List<tAnswer> answer = \_answer.GetAnswers(userViewModel.creatorLogin, userViewModel.pollNumber);

using (var workbook = new XLWorkbook())

{

var worksheet = workbook.Worksheets.Add("Отчёт");

worksheet.Rows().Style.Alignment.Horizontal = XLAlignmentHorizontalValues.Center;

worksheet.Rows().Style.Alignment.Vertical = XLAlignmentVerticalValues.Center;

workbook.CalculateMode = XLCalculateMode.Auto;

worksheet.Cell("A4").Value = "№";

worksheet.Cell("B4").Value = "Логины опрашиваемых";

//Заполнение из json

int row = 6;

int answerNumber = 1;

int rowNumberQuestion = 5;

int column = 3;

for (int i = 0; i < answer.Count; i++)

{

column = 3;

worksheet.Cell($"A{row}").Value = answerNumber;

//worksheet.Cell($"B{row}").Value = \_usersCreator.GetPassingDate(answer[i].userLogin);

worksheet.Cell($"B{row}").Value = answer[i].userLogin;

for (int j = 0; j < answer[i].QuestionsLists.Count; j++)

{

List<string> response = new List<string>();

worksheet.Cell(rowNumberQuestion, column).Value = answer[i].QuestionsLists[j].numberQuestion;

for (int x = 0; x < answer[i].QuestionsLists[j].ResponsesLists.Count; x++)

{

if(answer[i].QuestionsLists[j].typeQuestion == "CheckBox" || answer[i].QuestionsLists[j].typeQuestion == "Radio")

{

if(answer[i].QuestionsLists[j].ResponsesLists[x].isUsedTextBox == "true")

{

response.Add(answer[i].QuestionsLists[j].ResponsesLists[x].textResponse + ": " + answer[i].QuestionsLists[j].ResponsesLists[x].inputTextResponse);

}

else

{

response.Add(answer[i].QuestionsLists[j].ResponsesLists[x].textResponse);

}

}

else

{

response.Add(answer[i].QuestionsLists[j].ResponsesLists[x].inputTextResponse);

}

Продолжение приложения Б

}

worksheet.Cell(row, column).Value = string.Join("; ", response);

column++;

}

row++;

answerNumber++;

}

//Конец заполнения

worksheet.Columns().AdjustToContents();

worksheet.Cell("A1").Value = "Наименование опроса:";

worksheet.Cell("A2").Value = "Администратор опроса:";

worksheet.Cell("A3").Value = "Дата проведения опроса:";

worksheet.Cell("C1").Value = $" {poll.PollText}";

worksheet.Cell("C2").Value = $" {\_allCreators.GetCreatorFio(poll.CreatorLogin)}";

var cellWithFormulaA1 = worksheet.Cell("C3");

cellWithFormulaA1.FormulaA1 = $"=\" \"&TEXT(\"{userViewModel.date}\", \"дд.ММ.гггг\")";

worksheet.Cell("C4").Value = $"Вопросы";

worksheet.Range("A1:B1").Merge();

worksheet.Range("A2:B2").Merge();

worksheet.Range("A3:B3").Merge();

worksheet.Range("A4:A5").Merge();

worksheet.Range("B4:B5").Merge();

//worksheet.Range("C4:C5").Merge();

worksheet.Range(worksheet.Cell("C1"), worksheet.Cell(1, column - 1)).Merge();

worksheet.Range(worksheet.Cell("C2"), worksheet.Cell(2, column - 1)).Merge();

worksheet.Range(worksheet.Cell("C3"), worksheet.Cell(3, column - 1)).Merge();

worksheet.Range(worksheet.Cell("C4"), worksheet.Cell(4, column - 1)).Merge();

worksheet.Range("C1:C3").Style.Alignment.Horizontal = XLAlignmentHorizontalValues.Left;

worksheet.Range(worksheet.Cell("A1"), worksheet.Cell(row-1, column-1)).Style.Border.OutsideBorder = XLBorderStyleValues.Thin;

worksheet.Range(worksheet.Cell("A1"), worksheet.Cell(row-1, column-1)).Style.Border.InsideBorder = XLBorderStyleValues.Thin;

worksheet.Rows().Height = 22.5;

worksheet.Unprotect();

workbook.SaveAs(filePath);

}

return RedirectToAction("Download", new { filename = fileName, creatorLogin = userViewModel.creatorLogin, pollNumber = userViewModel.pollNumber });

}

Продолжение приложения Б

}

}

Модуль AnswerRepository

ListAnswer.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace CourseProject.Data.Models

{

public class ListAnswer

{

public List<tAnswer> Answer { get; set; }

}

public class tAnswer

{

public string creatorLogin { get; set; }

public string userLogin { get; set; }

public int pollNumber { get; set; }

public List<tQuestionsList> QuestionsLists { get; set; }

}

public class tQuestionsList

{

public int numberQuestion { get; set; }

public string textQuestion { get; set; }

public string typeQuestion { get; set; }

public List<tResponsesList> ResponsesLists { get; set; }

}

public class tResponsesList

{

public int numberResponse { get; set; }

public string textResponse { get; set; }

public string inputTextResponse { get; set; }

public string isChecked { get; set; }

public string isUsedTextBox { get; set; }

}

}

IAnswer.cs

using CourseProject.Data.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace CourseProject.Data.Interfaces

{

public interface IAnswer

{

Продолжение приложения Б

string CreateJsonFromAnswer(tAnswer answer);

bool SaveAnswer(tAnswer answer);

void DeleteAnswer(string userLogin);

List<tAnswer> GetAnswers(string creatorLogin, int pollNumber);

}

}

using CourseProject.Data.Interfaces;

using CourseProject.Data.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.Json;

using System.Threading.Tasks;

namespace CourseProject.Data.Repository

{

public class AnswerRepository : IAnswer

{

string answerPath = @"Data\answer.json";

//Формирование Json строки с ответами опрашиваемого

public string CreateJsonFromAnswer(tAnswer answer)

{

return JsonSerializer.Serialize<tAnswer>(answer);

}

//Удаление ответов опрашиваемого

public void DeleteAnswer(string userLogin)

{

try

{

string jsonString = System.IO.File.ReadAllText(answerPath, Encoding.UTF8);

List<tAnswer> json;

if (string.IsNullOrEmpty(jsonString))

{

json = new List<tAnswer>();

}

else

{

json = JsonSerializer.Deserialize<List<tAnswer>>(jsonString);

}

for(int i = 0; i < json.Count; i++)

{

if(json[i].userLogin == userLogin)

{

json.RemoveAt(i);

break;

}

}

Продолжение приложения Б

string newJson = JsonSerializer.Serialize<List<tAnswer>>(json);

System.IO.File.WriteAllText(answerPath, newJson, Encoding.UTF8);

}

catch

{

}

}

//Получение ответов опрашиваемого

public List<tAnswer> GetAnswers(string creatorLogin, int pollNumber)

{

string jsonString = System.IO.File.ReadAllText(answerPath, Encoding.UTF8);

return JsonSerializer.Deserialize<List<tAnswer>>(jsonString).FindAll(i => i.creatorLogin == creatorLogin && i.pollNumber == pollNumber);

}

//Сохранение ответов опрашиваемого

public bool SaveAnswer(tAnswer answer)

{

bool result = false;

try

{

string jsonString = System.IO.File.ReadAllText(answerPath, Encoding.UTF8);

List<tAnswer> json;

if (string.IsNullOrEmpty(jsonString))

{

json = new List<tAnswer>();

}

else

{

json = JsonSerializer.Deserialize<List<tAnswer>>(jsonString);

}

json.Add(answer);

string newJson = JsonSerializer.Serialize<List<tAnswer>>(json);

System.IO.File.WriteAllText(answerPath, newJson, Encoding.UTF8);

result = true;

}

catch

{

}

return result;

}

}

}

Продолжение приложения Б

Модуль CreatorRepository

Creator.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace CourseProject.Data.Models

{

public class Creator

{

public int id { get; set; }

public string login { get; set; }

public string password { get; set; }

public string email { get; set; }

public string firstname { get; set; }

public string name { get; set; }

public string lastname { get; set; }

public bool isVerified { get; set; }

public List<User> Users { set; get; }

}

}

IPollsCreator.cs

using CourseProject.Data.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace CourseProject.Data.Interfaces

{

public interface IPollsCreator

{

IEnumerable<Creator> AllCreators { get; }

Creator GetCreator(int creatorId);

void AddCreator(Creator creator);

void UpdateCreator(Creator creator);

string GetCreatorFio(string creatorLogin);

}

}

using CourseProject.Data.Interfaces;

using CourseProject.Data.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

Продолжение приложения Б

namespace CourseProject.Data.Repository

{

public class CreatorRepository : IPollsCreator

{

private readonly AppDbContent appDbContent;

public CreatorRepository(AppDbContent appDbContent)

{

this.appDbContent = appDbContent;

}

//Получение всех администраторов опросов

public IEnumerable<Creator> AllCreators => appDbContent.Creator;

//Добавление нового администратора в базу

public void AddCreator(Creator creator)

{

appDbContent.Creator.Add(creator);

appDbContent.SaveChanges();

}

//Поллучение конкретного администратора опросов

public Creator GetCreator(int creatorId)

{

return appDbContent.Creator.First(c => c.id == creatorId);

}

//Получение ФИО конкретного администратора опросов

public string GetCreatorFio(string creatorLogin)

{

Creator creator = appDbContent.Creator.First(i => i.login == creatorLogin);

return $"{creator.firstname} {creator.name} {creator.lastname}";

}

//Обновлекние данных конкретного администратора опросов

public void UpdateCreator(Creator creator)

{

appDbContent.Creator.Update(creator);

appDbContent.SaveChanges();

}

}

}

Модуль PollRepository

ListPoll.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace CourseProject.Data.Models

{

public class ListPoll

{

Продолжение приложения Б

public List<tPoll> tPoll { get; set; }

}

public class tPoll

{

public string CreatorLogin { get; set; }

public int Number { get; set; }

public string PollText { get; set; }

public List<tQuestion> tQuestion { get; set; }

}

public class tQuestion

{

public int Number { get; set; }

public string QuestionText { get; set; }

public string Type { get; set; }

public string IsRequired { get; set; }

public int minCountResponse { get; set; }

public int maxCountResponse { get; set; }

public List<tResponse> tResponse { get; set; }

}

public class tResponse

{

public int Number { get; set; }

public string ResponseText { get; set; }

public string UseTextBox { get; set; }

public int JumpTo { get; set; }

}

}

IPolls.cs

using CourseProject.Data.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace CourseProject.Data.Interfaces

{

public interface IPolls

{

void AddPoll(ListPoll poll);

List<tPoll> GetPolls(string creatorLogin);

tPoll GetPoll(string creatorLogin, int pollNumber);

bool SaveEditPoll(tPoll tpoll);

bool SaveAddPoll(tPoll tpoll);

bool DeletePoll(tPoll tpoll);

string ExportPoll(tPoll tpoll);

tPoll ImportPoll(string pollJsonPath);

}

}

using CourseProject.Data.Interfaces;

Продолжение приложения Б

using CourseProject.Data.Models;

using Microsoft.EntityFrameworkCore;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.Json;

using System.Threading.Tasks;

namespace CourseProject.Data.Repository

{

public class PollRepository : IPolls

{

private readonly AppDbContent appDbContent;

public PollRepository(AppDbContent appDbContent)

{

this.appDbContent = appDbContent;

}

public void AddPoll(ListPoll poll)

{

}

//Получение списка опросов

public List<tPoll> GetPolls(string creatorLogin)

{

string jsonString = System.IO.File.ReadAllText(@"Data\json.json", Encoding.UTF8);

List<tPoll> json;

if (JsonSerializer.Deserialize<List<tPoll>>(jsonString).FindAll(c => c.CreatorLogin.Contains(creatorLogin)).Count != 0)

{

json = JsonSerializer.Deserialize<List<tPoll>>(jsonString).FindAll(c => c.CreatorLogin.Contains(creatorLogin));

}

else

{

json = new List<tPoll>() { new tPoll { CreatorLogin = creatorLogin, Number = 0 } };

}

return json;

}

//Получение опроса

public tPoll GetPoll(string creatorLogin, int pollNumber)

{

string jsonString = System.IO.File.ReadAllText(@"Data\json.json", Encoding.UTF8);

List<tPoll> json = JsonSerializer.Deserialize<List<tPoll>>(jsonString).FindAll(c => c.CreatorLogin.Contains(creatorLogin) && c.Number == pollNumber);

return json[0];

}

//Сохранение нового опроса

Продолжение приложения Б

public bool SaveAddPoll(tPoll tpoll)

{

tPoll t = tpoll;

bool result = false;

try

{

string jsonString = System.IO.File.ReadAllText(@"Data\json.json", Encoding.UTF8);

List<tPoll> json = JsonSerializer.Deserialize<List<tPoll>>(jsonString);

while (true)

{

if (json.Find(p => p.CreatorLogin == t.CreatorLogin && p.Number == t.Number) == null)

{

json.Add(t);

break;

}

else

{

t.Number++;

}

}

string newJson = JsonSerializer.Serialize<List<tPoll>>(json);

System.IO.File.WriteAllText(@"Data\json.json", newJson, Encoding.UTF8);

result = true;

return result;

}

catch

{

}

return result;

}

//Сохранение изменённого опроса

public bool SaveEditPoll(tPoll tpoll)

{

tPoll t = tpoll;

bool result = false;

try

{

string jsonString = System.IO.File.ReadAllText(@"Data\json.json", Encoding.UTF8);

List<tPoll> json = JsonSerializer.Deserialize<List<tPoll>>(jsonString);

for(int i = 0; i< json.Count; i++)

{

if(json[i].CreatorLogin == tpoll.CreatorLogin && json[i].Number == tpoll.Number)

{

json[i] = tpoll;

}

}

string newJson = JsonSerializer.Serialize<List<tPoll>>(json);

System.IO.File.WriteAllText(@"Data\json.json", newJson, Encoding.UTF8);

Продолжение приложения Б

result = true;

return result;

}

catch

{

}

return result;

}

//Удаление опроса

public bool DeletePoll(tPoll tpoll)

{

bool result = false;

try

{

string jsonString = System.IO.File.ReadAllText(@"Data\json.json", Encoding.UTF8);

List<tPoll> json = JsonSerializer.Deserialize<List<tPoll>>(jsonString);

for (int i = 0; i < json.Count; i++)

{

if (json[i].CreatorLogin == tpoll.CreatorLogin && json[i].Number == tpoll.Number)

{

json.RemoveAt(i);

}

}

string newJson = JsonSerializer.Serialize<List<tPoll>>(json);

System.IO.File.WriteAllText(@"Data\json.json", newJson, Encoding.UTF8);

result = true;

return result;

}

catch

{

}

return result;

}

//Экспорт опроса

public string ExportPoll(tPoll tpoll)

{

return JsonSerializer.Serialize<tPoll>(tpoll);

}

//Импорт опроса

public tPoll ImportPoll(string pollJsonPath)

{

string jsonString = System.IO.File.ReadAllText(pollJsonPath, Encoding.UTF8);

tPoll json = JsonSerializer.Deserialize<tPoll>(jsonString);

return json;

}

}

}

Продолжение приложения Б

Модуль UserRepository

User.cs

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Threading.Tasks;

namespace CourseProject.Data.Models

{

public class User

{

public int id { get; set; }

public string login { get; set; }

public string password { get; set; }

public string institution { get; set; }

public int isPassed { get; set; }

public int pollNumber { get; set; }

[DisplayFormat(DataFormatString = "{0:MM/dd/yyyy}")]

public DateTime datePAssingPoll { get; set; }

public virtual Creator Creator { set; get; }

}

}

IUsersCreator.cs

using CourseProject.Data.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace CourseProject.Data.Interfaces

{

public interface IUsersCreator

{

void AddUser(List<User> user);

IEnumerable<User> GetUsers(string creatorLogin, int pollNumber);

User GetUser(string userLogin, string userPassword);

User GetUser(string userLogin);

void UpdateUser(string userLogin);

string GetPassingDate(string userLogin);

void DeleteUser(string userLogin);

}

}

using CourseProject.Data.Interfaces;

using CourseProject.Data.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

Продолжение приложения Б

namespace CourseProject.Data.Repository

{

public class UserRepository : IUsersCreator

{

private readonly AppDbContent appDbContent;

public UserRepository(AppDbContent appDbContent)

{

this.appDbContent = appDbContent;

}

//Получение списка опрашиваемых

public IEnumerable<User> GetUsers(string creatorLogin, int pollNumber)

{

return appDbContent.User.Where(c => c.Creator.login == creatorLogin && c.pollNumber == pollNumber);

}

//Добавление опрашиваемых

public void AddUser(List<User> user)

{

for(int i = 0; i < user.Count; i++)

{

appDbContent.User.Add(user[i]);

}

appDbContent.SaveChanges();

}

//Получение опрашичаемого

public User GetUser(string userLogin)

{

return appDbContent.User.FirstOrDefault(u => u.login == userLogin);

}

//Получение опрашиваемого

public User GetUser(string userLogin, string userPassword)

{

return appDbContent.User.FirstOrDefault(u => u.login == userLogin && u.password == userPassword);

}

//Обновление данных опрашиваемого

public void UpdateUser(string userLogin)

{

appDbContent.User.FirstOrDefault(u => u.login == userLogin).isPassed = 1;

appDbContent.User.FirstOrDefault(u => u.login == userLogin).datePAssingPoll = DateTime.Now;

appDbContent.SaveChanges();

}

//Получение даты прохождения опроса опрашиваемым

public string GetPassingDate(string userLogin)

{

Продолжение приложения Б

return appDbContent.User.FirstOrDefault(u => u.login == userLogin).datePAssingPoll.ToString();

}

//Удаление опрашиваемого

public void DeleteUser(string userLogin)

{

appDbContent.User.Remove(appDbContent.User.FirstOrDefault(u => u.login == userLogin));

appDbContent.SaveChanges();

}

}

}

Модуль ScriptJs

let limitCountQuestions = 150;

let limitCountResponses = 20;

let IdAddQuestion = 0;

//Добавление полей вопросов

function addFieldQuestion() {

IdAddQuestion = 0;

let boxIdAddQuestion = IdAddQuestion;

while (document.getElementById("tQuestion\_" + boxIdAddQuestion + "\_\_Number") != null) {

boxIdAddQuestion++;

}

if (boxIdAddQuestion < limitCountQuestions) {

let container = document.getElementById("questions");

let div = document.createElement("div");

div.setAttribute("class", "w-75 h-100 d-flex flex-column mt-4 mb-3");

div.setAttribute("id", "question\_" + (boxIdAddQuestion + 1));

let h6 = document.createElement("h6");

h6.setAttribute("class", "w-100 h-100");

h6.setAttribute("style", "max-width: 100px");

h6.innerText = "• Вопрос №";

let input2 = document.createElement("input");

input2.setAttribute("class", "form-control mr-2");

input2.setAttribute("id", "tQuestion\_" + boxIdAddQuestion + "\_\_Number");

input2.setAttribute("name", "tQuestion[" + boxIdAddQuestion + "].Number");

input2.setAttribute("style", "max-width: 100px");

input2.setAttribute("type", "number");

input2.setAttribute("required", "");

input2.setAttribute("value", (boxIdAddQuestion + 1));

let h6\_2 = document.createElement("h6");

h6\_2.setAttribute("class", "w-100 h-100");

h6\_2.setAttribute("style", "max-width: 100px");

h6\_2.innerText = " Ответы: ";

Продолжение приложения Б

let option1 = document.createElement("option");

option1.setAttribute("value", "CheckBox");

option1.innerText = "Несколько";

let option2 = document.createElement("option");

option2.setAttribute("value", "Radio");

option2.innerText = "Один";

let option3 = document.createElement("option");

option3.setAttribute("value", "");

option3.innerText = "Без выбора";

let select = document.createElement("select");

select.setAttribute("class", "form-control mr-2");

select.setAttribute("id", "tQuestion\_" + boxIdAddQuestion + "\_\_Type");

select.setAttribute("name", "tQuestion[" + boxIdAddQuestion + "].Type");

select.setAttribute("style", "max-width: 150px");

select.setAttribute("onchange", "addFieldCountResponseInQuestion(" + (boxIdAddQuestion + 1) + ");");

select.appendChild(option1);

select.appendChild(option2);

select.appendChild(option3);

let h6\_3 = document.createElement("h6");

h6\_3.setAttribute("class", "w-100 h-100");

h6\_3.setAttribute("style", "max-width: 150px");

h6\_3.innerText = " Мин. ответов: ";

let input4 = document.createElement("input");

input4.setAttribute("class", "form-control mr-2");

input4.setAttribute("id", "tQuestion\_" + boxIdAddQuestion + "\_\_minCountResponse");

input4.setAttribute("name", "tQuestion[" + boxIdAddQuestion + "].minCountResponse");

input4.setAttribute("style", "max-width: 100px");

input4.setAttribute("type", "number");

input4.setAttribute("value", 1);

input4.setAttribute("disabled", "");

let a = document.createElement("a");

a.setAttribute("class", "btn btn-success text-light mr-2");

a.setAttribute("onclick", "addFieldResponse(" + (boxIdAddQuestion + 1) + ")");

a.innerText = "Добавить ответ";

let a2 = document.createElement("a");

a2.setAttribute("class", "btn btn-danger text-light");

a2.setAttribute("id", (boxIdAddQuestion + 1));

a2.setAttribute("onclick", "deleteFieldQuestion(this)");

a2.innerText = "x";

let div2 = document.createElement("div");

div2.setAttribute("class", "d-flex flex-row justify-content-start text-center align-items-center mb-2");

div2.appendChild(h6);

div2.appendChild(input2);

div2.appendChild(h6\_2);

div2.appendChild(select);

Продолжение приложения Б

div2.appendChild(h6\_3);

div2.appendChild(input4);

div2.appendChild(a);

div2.appendChild(a2);

div.appendChild(div2);

let input = document.createElement("input");

input.setAttribute("class", "form-control");

input.setAttribute("id", "tQuestion\_" + boxIdAddQuestion + "\_\_QuestionText");

input.setAttribute("name", "tQuestion[" + boxIdAddQuestion + "].QuestionText");

input.setAttribute("type", "text");

input.setAttribute("value", "");

input.setAttribute("required", "");

div.appendChild(input);

let input3 = document.createElement("input");

input3.setAttribute("class", "form-check-input");

input3.setAttribute("onchange", "valueChecked(this);addFieldCountResponseInQuestion(" + (boxIdAddQuestion + 1) + ");");

input3.setAttribute("id", "tQuestion\_" + boxIdAddQuestion + "\_\_IsRequired");

input3.setAttribute("name", "tQuestion[" + boxIdAddQuestion + "].IsRequired");

input3.setAttribute("type", "checkbox");

input3.setAttribute("value", "false");

let lable = document.createElement("lable");

lable.setAttribute("class", "form-check-lable");

lable.innerText = "Обязательный вопрос?";

let div3 = document.createElement("div");

div3.setAttribute("class", "mb-3")

div3.appendChild(input3);

div3.appendChild(lable);

div.appendChild(div3);

container.appendChild(div);

}

else {

alert("Опрос не может содержать больше " + limitCountQuestions + " вопросов.");

}

}

//Удаление полей вопросов

function deleteFieldQuestion(e) {

var deletedId = 0;

deletedId = Number(e.id);

var elem = document.getElementById("question\_" + e.id);

elem.parentNode.removeChild(elem);

while (document.getElementById("question\_" + (deletedId+1)) != null) {

let newIdField = document.getElementById("question\_" + (deletedId + 1));

newIdField.setAttribute("id", "question\_" + deletedId);

Продолжение приложения Б

let newIdFieldNumber = document.getElementById("tQuestion\_" + deletedId + "\_\_Number");

newIdFieldNumber.setAttribute("id", "tQuestion\_" + (deletedId - 1) + "\_\_Number");

newIdFieldNumber.setAttribute("name", "tQuestion[" + (deletedId - 1) + "].Number");

newIdFieldNumber.setAttribute("value", deletedId);

let newIdFieldType = document.getElementById("tQuestion\_" + deletedId + "\_\_Type");

newIdFieldType.setAttribute("id", "tQuestion\_" + (deletedId - 1) + "\_\_Type");

newIdFieldType.setAttribute("name", "tQuestion[" + (deletedId - 1) + "].Type");

let newIdFieldQuestionText = document.getElementById("tQuestion\_" + deletedId + "\_\_QuestionText");

newIdFieldQuestionText.setAttribute("id", "tQuestion\_" + (deletedId - 1) + "\_\_QuestionText");

newIdFieldQuestionText.setAttribute("name", "tQuestion[" + (deletedId - 1) + "].QuestionText");

let newIdFieldIsRequired = document.getElementById("tQuestion\_" + deletedId + "\_\_IsRequired");

newIdFieldIsRequired.setAttribute("id", "tQuestion\_" + (deletedId - 1) + "\_\_IsRequired");

newIdFieldIsRequired.setAttribute("name", "tQuestion[" + (deletedId - 1) + "].IsRequired");

let newIdButtonDelete = document.getElementById(deletedId+1);

newIdButtonDelete.setAttribute("id", deletedId);

deletedId++;

}

}

let IdAddResponse = {};

//Добавление полей ответов

function addFieldResponse(tquestion\_) {

console.log(tquestion\_);

IdAddResponse[tquestion\_] = 0;

let boxTquestion\_ = Number(tquestion\_);

boxTquestion\_--;

while (document.getElementById("tQuestion\_" + boxTquestion\_ + "\_\_tResponse\_" + Number(IdAddResponse[tquestion\_])+"\_\_Number") != null) {

IdAddResponse[tquestion\_]++;

}

if (IdAddResponse[tquestion\_] < limitCountResponses) {

let container = document.getElementById("question\_" + tquestion\_);

let div = document.createElement("div");

div.setAttribute("id", "response\_" + tquestion\_ + "\_" + (Number(IdAddResponse[tquestion\_]) + 1))

let h6 = document.createElement("h6");

h6.setAttribute("class", "w-100 h-100");

h6.setAttribute("style", "max-width: 100px");

h6.innerText = "Ответ №";

let input = document.createElement("input");

input.setAttribute("class", "form-control mr-2");

Продолжение приложения Б

input.setAttribute("id", "tQuestion\_" + boxTquestion\_ + "\_\_tResponse\_" + IdAddResponse[tquestion\_] + "\_\_Number");

input.setAttribute("name", "tQuestion[" + boxTquestion\_ + "].tResponse[" + IdAddResponse[tquestion\_] + "].Number");

input.setAttribute("style", "max-width: 100px");

input.setAttribute("type", "number");

input.setAttribute("value", (Number(IdAddResponse[tquestion\_]) + 1));

input.setAttribute("required", "");

let a = document.createElement("a");

a.setAttribute("class", "btn btn-danger text-light");

a.setAttribute("id", tquestion\_ + "\_" + (Number(IdAddResponse[tquestion\_]) + 1));

a.setAttribute("onclick", "deleteFieldResponse(this)");

a.innerText = "x";

let p = document.createElement("p");

p.appendChild(a);

let div2 = document.createElement("div");

div2.setAttribute("class", "d-flex flex-row justify-content-start")

div2.appendChild(h6);

div2.appendChild(input);

div2.appendChild(p);

div.appendChild(div2);

let input2 = document.createElement("input");

input2.setAttribute("class", "form-control");

input2.setAttribute("id", "tQuestion\_" + boxTquestion\_ + "\_\_tResponse\_" + IdAddResponse[tquestion\_] + "\_\_ResponseText");

input2.setAttribute("name", "tQuestion[" + boxTquestion\_ + "].tResponse[" + IdAddResponse[tquestion\_] + "].ResponseText");

input2.setAttribute("type", "text");

input2.setAttribute("value", "");

input2.setAttribute("required", "");

div.appendChild(input2);

let input3 = document.createElement("input");

input3.setAttribute("class", "form-check-input");

input3.setAttribute("id", "tQuestion\_" + boxTquestion\_ + "\_\_tResponse\_" + IdAddResponse[tquestion\_] + "\_\_UseTextBox");

input3.setAttribute("name", "tQuestion[" + boxTquestion\_ + "].tResponse[" + IdAddResponse[tquestion\_] + "].UseTextBox");

input3.setAttribute("onchange", "valueChecked(this)");

input3.setAttribute("type", "checkbox");

input3.setAttribute("value", "false");

let lable = document.createElement("lable");

lable.setAttribute("class", "form-check-lable");

lable.innerText = "Добавить поле для ввода?";

let div3 = document.createElement("div");

div3.setAttribute("class", "mb-3")

div3.appendChild(input3);

div3.appendChild(lable);

div.appendChild(div3);

Продолжение приложения Б

container.appendChild(div);

}

else {

alert("Вопрос не может содержать больше " + limitCountResponses + " ответов.");

}

}

//Удаление полей ответов

function deleteFieldResponse(e) {

let elem = document.getElementById("response\_" + e.id);

elem.parentNode.removeChild(elem);

let deletedId = String(e.id).split("\_");

deletedId[0] = (Number(deletedId[0]) - 1);

while (document.getElementById("response\_" + (Number(deletedId[0]) + 1) + "\_" + (Number(deletedId[1]) + 1)) != null) {

let newId = (Number(deletedId[1]) - 1);

let newIdField = document.getElementById("response\_" + (Number(deletedId[0]) + 1) + "\_" + (Number(deletedId[1]) + 1));

newIdField.setAttribute("id", "response\_" + (Number(deletedId[0]) + 1) + "\_" + (deletedId[1]));

let newIdFieldNumber = document.getElementById("tQuestion\_" + deletedId[0] + "\_\_tResponse\_" + deletedId[1] + "\_\_Number");

console.log("deleteFieldResponse2: " + "tQuestion\_" + deletedId[0] + "\_\_tResponse\_" + deletedId[1] + "\_\_Number");

newIdFieldNumber.setAttribute("id", "tQuestion\_" + deletedId[0] + "\_\_tResponse\_" + newId + "\_\_Number");

newIdFieldNumber.setAttribute("name", "tQuestion[" + deletedId[0] + "].tResponse[" + newId + "].Number");

newIdFieldNumber.setAttribute("value", deletedId[1]);

let newIdButtonDelete = document.getElementById((Number(deletedId[0]) + 1) + "\_" + (Number(deletedId[1]) + 1));

newIdButtonDelete.setAttribute("id", (Number(deletedId[0]) + 1) + "\_" + deletedId[1]);

let newIdFieldResponseText = document.getElementById("tQuestion\_" + deletedId[0] + "\_\_tResponse\_" + deletedId[1] + "\_\_ResponseText");

newIdFieldResponseText.setAttribute("id", "tQuestion\_" + deletedId[0] + "\_\_tResponse\_" + newId + "\_\_ResponseText");

newIdFieldResponseText.setAttribute("name", "tQuestion[" + deletedId[0] + "].tResponse[" + newId + "].ResponseText");

let newIdFieldUseTextBox = document.getElementById("tQuestion\_" + deletedId[0] + "\_\_tResponse\_" + deletedId[1] + "\_\_UseTextBox");

newIdFieldUseTextBox.setAttribute("id", "tQuestion\_" + deletedId[0] + "\_\_tResponse\_" + newId + "\_\_UseTextBox");

newIdFieldUseTextBox.setAttribute("name", "tQuestion[" + deletedId[0] + "].tResponse[" + newId + "].UseTextBox");

deletedId[1]++;

}

}

Продолжение приложения Б

function valueChecked(e) {

if (e.checked) {

e.value = "true";

}

else {

e.value = "false";

}

}

//Добавление полей количества ответов в вопросе

function addFieldCountResponseInQuestion(numberQuestion) {

let questionId = numberQuestion - 1;

let inputIsRequired = document.getElementById("tQuestion\_" + questionId + "\_\_IsRequired");

let selectType = document.getElementById("tQuestion\_" + questionId + "\_\_Type");

let inputMinCountResponse = document.getElementById("tQuestion\_" + questionId + "\_\_minCountResponse");

if (inputIsRequired.checked == true && selectType.value == "CheckBox") {

inputMinCountResponse.removeAttribute("disabled");

}

else {

inputMinCountResponse.setAttribute("disabled", "");

inputMinCountResponse.value = 1;

}

}

//Отображение формы

function showForm(e, creatorLogin) {

let options = e.options;

for (var i = 0; i < options.length; i++) {

let formId = document.getElementById("form\_" + e.options[i].value);

if (e.options[i].value == e.value) {

formId.setAttribute("style", "display: flex");

}

else {

formId.setAttribute("style", "display: none;");

}

}

let numberPoll = document.getElementById("numberPoll").value;

let buttonEdit = document.getElementById("buttonEdit");

buttonEdit.setAttribute("href", "/EditPoll/Edit?editPollLogin=" + creatorLogin + "&editPollNumber=" + numberPoll);

let buttonDelete = document.getElementById("buttonDelete");

buttonDelete.setAttribute("href", "/EditPoll/Delete?deletePollLogin=" + creatorLogin + "&deletePollNumber=" + numberPoll);

let buttonUser = document.getElementById("buttonUser");

buttonUser.setAttribute("href", "/User/ShowUser?creatorLogin=" + creatorLogin + "&pollNumber=" + numberPoll);

let buttonExport = document.getElementById("buttonExport");

Продолжение приложения Б

buttonExport.setAttribute("href", "/MainPage/ExportPoll?creatorLogin=" + creatorLogin + "&pollNumber=" + numberPoll);

let exampleModalLabel = document.getElementById("exampleModalLabel");

exampleModalLabel.innerText = "Удалить безвозвратно опрос №" + numberPoll;

let modalBodyPollText = document.getElementById("modalBodyPollText");

modalBodyPollText.innerText = document.getElementById("pollText\_" + numberPoll).innerText;

}

let checkPoll = {};

//Проверка вопросов

function checkQuestion(e) {

let classQuestion = e.getAttribute("data-checkQuestion");

let elementList = $("." + classQuestion);

let checkedElement = document.getElementById(classQuestion + "\_check");

checkPoll[classQuestion] = false;

let minCountResponse = e.getAttribute("data-minResponse");

let currentCountResponse = 0;

for (let i = 0; i < elementList.length; i++) {

console.log(elementList[i].type);

if (elementList[i].type == "checkbox") {

if (minCountResponse != 0 && minCountResponse != null) {

if (elementList[i].checked) {

currentCountResponse++;

}

if (currentCountResponse >= minCountResponse) {

checkPoll[classQuestion] = true;

break;

}

else {

checkPoll[classQuestion] = false;

}

}

}

else if (elementList[i].type == "radio") {

if (elementList[i].checked) {

checkPoll[classQuestion] = true;

break;

}

else {

checkPoll[classQuestion] = false;

}

}

Продолжение приложения Б

else if (elementList[i].type == "text") {

if (elementList[i].value.length > 0) {

checkPoll[classQuestion] = true;

break;

}

else {

checkPoll[classQuestion] = false;

}

}

/\*

if (elementList[i].checked) {

checkPoll[classQuestion] = true;

break;

}

else {

if (elementList[i].type == "text") {

if (elementList[i].value.length > 0) {

checkPoll[classQuestion] = true;

break;

}

else {

checkPoll[classQuestion] = false;

}

}

else {

checkPoll[classQuestion] = false;

}

}\*/

}

console.log(checkPoll);

}

//проверка инпутов вопросов

function checkInputQuestion(e) {

let input = document.getElementById(e.id + "\_input");

if (input) {

if (e.checked) {

input.removeAttribute("disabled");

}

else {

input.setAttribute("disabled", "");

input.value = "";

}

}

}

//Заполнение валидных опросов

function fillcheckPoll(questionId) {

checkPoll[questionId] = false;

console.log(checkPoll);

Продолжение приложения Б

}

//Отправка формы

function submitForm(formId) {

for (let key in checkPoll) {

if (!checkPoll[key]) {

alert("Проверьте вопрос №" + key.replace("tquestion", ""));

let focusQuestion = document.getElementById("div\_" + key).offsetTop - 100;

$('body, html').animate({ scrollTop: focusQuestion }, 500);

return false;

}

}

let inputList = document.getElementsByTagName("input");

for (let i = 0; i < inputList.length; i++) {

if (inputList[i].getAttribute("type") == "Radio" || inputList[i].getAttribute("type") == "CheckBox") {

let hiddenInput = document.getElementById(inputList[i].id + "\_h");

if (inputList[i].checked) {

hiddenInput.value = true;

}

else {

hiddenInput.value = false;

}

}

}

$("#" + formId).submit();

}

let checkUser = {};

//Заполнение списка удаляемых опрашиваемых

function checkUserDelete(e) {

if (e.checked) {

checkUser[e.id] = e.id;

}

else {

delete checkUser[e.id];

}

}

//Отправка формы опрашиваемых

function submitUserForm() {

let countKeys = Object.keys(checkUser).length;

if (countKeys > 0) {

let deletetUserList = document.getElementById("deleteUserList");

let userForm = document.getElementById("UserForm");

let userLoginString = "";

let i = 0;

for (key in checkUser) {

Продолжение приложения Б

if (i < countKeys - 1) {

userLoginString += checkUser[key] + ",";

i++;

}

else {

userLoginString += checkUser[key];

i++

}

}

deletetUserList.setAttribute("value", userLoginString);

userForm.submit();

}

else {

alert("Для удаления необходимо отметить хотябы 1 пользователя!")

}

}

//Удаление всех опрашиваемых

function allDeleteCheckBox(e) {

let deleteCheckBoxList = document.getElementsByClassName("deleteUserCheckbox");

if (e.checked) {

for (let i = 0; i < deleteCheckBoxList.length; i++) {

deleteCheckBoxList[i].checked = true;

checkUser[deleteCheckBoxList[i].id] = deleteCheckBoxList[i].id;

}

}

else {

for (let i = 0; i < deleteCheckBoxList.length; i++) {

deleteCheckBoxList[i].checked = false;

delete checkUser[deleteCheckBoxList[i].id];

}

}

console.log(checkUser);

}

//Проверка формы импорта

function checkImportForm() {

let inputFile = document.getElementById("inputFile");

let importForm = document.getElementById("ImportForm");

if (inputFile.value) {

importForm.submit();

}

else {

alert("Вы не выбрали файл!");

}

}

let passwordHidden = true;

//Скрытие полей пароля

function hidePassword() {

let eyeOff = document.getElementsByClassName("bi-eye-slash");

let eyeOn = document.getElementsByClassName("bi-eye");

let inputPass = document.getElementsByClassName("pass");

if (passwordHidden) {

eyeOn[0].removeAttribute("hidden");

Продолжение приложения Б

eyeOff[0].setAttribute("hidden", "");

for (let i = 0; i < inputPass.length; i++) {

inputPass[i].type = "text";

}

passwordHidden = false;

}

else {

eyeOff[0].removeAttribute("hidden");

eyeOn[0].setAttribute("hidden", "");

for (let i = 0; i < inputPass.length; i++) {

inputPass[i].type = "password";

}

passwordHidden = true;

}

}

//вывод форм страны

Account

CheckEmailViewModel.cs

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Threading.Tasks;

namespace CourseProject.ViewModels

{

public class CheckEmailViewModel

{

[RegularExpression(@"^\w+\@\w+\.\w+", ErrorMessage = "Некорректный адрес")]

public string email { get; set; }

public bool isCorrectEmail { get; set; }

}

}

CheckEmail.cshtml

@model CheckEmailViewModel

<div class="min-vh-100 min-vw-100 row align-items-center justify-content-center">

@{

if (Model.isCorrectEmail == true)

{

<div class="d-flex flex-column align-items-center w-100">

<h3 class="mb-3">На указанную почту была выслана инструкция с дальнейшими действиями для восстановления пароля.</h3>

</div>

}

else

{

Продолжение приложения Б

<form asp-action="CheckEmail" asp-controller="Account" asp-anti-forgery="true" class="d-flex flex-column align-items-center w-100">

<h3 class="mb-3">Восстановление пароля</h3>

<p class="mb-3">@Html.ValidationSummary(true, "", new { @class = "text-danger" })</p>

<div class="form-group w-100 col-sm-3">

<div class="mb-3 d-flex flex-column">

<label asp-for="email">Введите вашу Эл. почту к которой привязан аккаунт:</label>

<input type="text" name="email" value="" placeholder="Почта" required />

</div>

<div class="mb-3 d-flex flex-column">

<button type="submit">Далее</button>

</div>

<div class="mb-3 d-flex flex-column">

<p>Уже зарегистрированы? <a asp-action="Login" asp-controller="Account">Войти.</a></p>

</div>

</div>

</form>

}

}

</div>

CheckEmailViewModel.cs

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Threading.Tasks;

namespace CourseProject.ViewModels

{

public class EditEmailViewModel

{

public string password { get; set; }

[RegularExpression(@"^\w+\@\w+\.\w+", ErrorMessage = "Некорректный адрес")]

public string email { get; set; }

}

}

EditEmail.cshtml

@model EditEmailViewModel

<div class="min-vh-100 min-vw-100 row align-items-center justify-content-center">

<form asp-action="EditEmail" asp-controller="Account" asp-anti-forgery="true" class="d-flex flex-column align-items-center w-100">

<nav class="navbar fixed-top navbar-expand-lg navbar-dark bg-dark">

<div class="collapse navbar-collapse justify-content-around" id="navbarSupportedContent">

<div>

<a class="btn btn-light" style="min-width: 185px" asp-controller="MainPage" asp-action="Questioning">Отмена</a>

</div>

Продолжение приложения Б

<div>

<button type="submit" style="min-width: 185px" class="btn btn-primary">Сохранить</button>

</div>

</div>

</nav>

<h3 class="mb-3">Редактирование Эл. почты</h3>

<p class="mb-3">@Html.ValidationSummary(true, "", new { @class = "text-danger" })</p>

<div class="form-group w-100 col-sm-3">

<div class="mb-3 d-flex flex-column">

<label asp-for="email">Эл. почта:</label>

<input class="" type="text" name="email" value="@Model.email" placeholder="Эл. почта" required />

</div>

<div class="mb-3 d-flex flex-column">

<label asp-for="password">Введите пароль:</label>

<div class="d-flex flex-row w-100">

<input class="w-100 pass" type="password" name="password" value="" placeholder="Пароль" required />

<button calss="btn" type="button" onclick="hidePassword()">

<svg class="bi bi-eye-slash" width="1em" height="1em" viewBox="0 0 16 16" fill="currentColor" xmlns="http://www.w3.org/2000/svg">

<path d="M13.359 11.238C15.06 9.72 16 8 16 8s-3-5.5-8-5.5a7.028 7.028 0 0 0-2.79.588l.77.771A5.944 5.944 0 0 1 8 3.5c2.12 0 3.879 1.168 5.168 2.457A13.134 13.134 0 0 1 14.828 8c-.058.087-.122.183-.195.288-.335.48-.83 1.12-1.465 1.755-.165.165-.337.328-.517.486l.708.709z" />

<path d="M11.297 9.176a3.5 3.5 0 0 0-4.474-4.474l.823.823a2.5 2.5 0 0 1 2.829 2.829l.822.822zm-2.943 1.299l.822.822a3.5 3.5 0 0 1-4.474-4.474l.823.823a2.5 2.5 0 0 0 2.829 2.829z" />

<path d="M3.35 5.47c-.18.16-.353.322-.518.487A13.134 13.134 0 0 0 1.172 8l.195.288c.335.48.83 1.12 1.465 1.755C4.121 11.332 5.881 12.5 8 12.5c.716 0 1.39-.133 2.02-.36l.77.772A7.029 7.029 0 0 1 8 13.5C3 13.5 0 8 0 8s.939-1.721 2.641-3.238l.708.709z" />

<path fill-rule="evenodd" d="M13.646 14.354l-12-12 .708-.708 12 12-.708.708z" />

</svg>

<svg hidden class="bi bi-eye" width="1em" height="1em" viewBox="0 0 16 16" fill="currentColor" xmlns="http://www.w3.org/2000/svg">

<path fill-rule="evenodd" d="M16 8s-3-5.5-8-5.5S0 8 0 8s3 5.5 8 5.5S16 8 16 8zM1.173 8a13.134 13.134 0 0 0 1.66 2.043C4.12 11.332 5.88 12.5 8 12.5c2.12 0 3.879-1.168 5.168-2.457A13.134 13.134 0 0 0 14.828 8a13.133 13.133 0 0 0-1.66-2.043C11.879 4.668 10.119 3.5 8 3.5c-2.12 0-3.879 1.168-5.168 2.457A13.133 13.133 0 0 0 1.172 8z" />

<path fill-rule="evenodd" d="M8 5.5a2.5 2.5 0 1 0 0 5 2.5 2.5 0 0 0 0-5zM4.5 8a3.5 3.5 0 1 1 7 0 3.5 3.5 0 0 1-7 0z" />

</svg>

</button>

</div>

</div>

</div>

</form>

</div>

Продолжение приложения Б

EditFullNameViewModel.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace CourseProject.ViewModels

{

public class EditFullNameViewModel

{

public string firstName { get; set; }

public string name { get; set; }

public string lastName { get; set; }

}

}

EditFullName.cshtml

@model EditFullNameViewModel

<div class="min-vh-100 min-vw-100 row align-items-center justify-content-center">

<form asp-action="EditFullName" asp-controller="Account" asp-anti-forgery="true" class="d-flex flex-column align-items-center w-100">

<nav class="navbar fixed-top navbar-expand-lg navbar-dark bg-dark">

<div class="collapse navbar-collapse justify-content-around" id="navbarSupportedContent">

<div>

<a class="btn btn-light" style="min-width: 185px" asp-controller="MainPage" asp-action="Questioning">Отмена</a>

</div>

<div>

<button type="submit" style="min-width: 185px" class="btn btn-primary">Сохранить</button>

</div>

</div>

</nav>

<h3 class="mb-3">Редактирование Ф.И.О</h3>

<p class="mb-3">@Html.ValidationSummary(true, "", new { @class = "text-danger" })</p>

<div class="form-group w-100 col-sm-3">

<div class="mb-3 d-flex flex-column">

<label asp-for="firstName">Фамилия:</label>

<input class="pass" type="text" name="firstName" value="@Model.firstName" placeholder="Фамилия" required />

</div>

<div class="mb-3 d-flex flex-column">

<label asp-for="name">Имя:</label>

<input class="pass" type="text" name="name" value="@Model.name" placeholder="Имя" required />

</div>

<div class="mb-3 d-flex flex-column">

<label asp-for="lastName">Отчество:</label>

Продолжение приложения Б

<input class="pass" type="text" name="lastName" value="@Model.lastName" placeholder="Отчество" required />

</div>

</div>

</form>

</div>

EditPasswordViewModel.cs

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Threading.Tasks;

namespace CourseProject.ViewModels

{

public class EditPasswordViewModel

{

public string oldPassword { get; set; }

public string newPassword { get; set; }

[Compare("newPassword", ErrorMessage = "Пароль введен неверно")]

public string сonfirmNewPassword { get; set; }

public bool isPasswordСhanged { get; set; }

}

}

EditPassword.cshtml

@model EditPasswordViewModel

<div class="min-vh-100 min-vw-100 row align-items-center justify-content-center">

<form asp-action="EditPassword" asp-controller="Account" asp-anti-forgery="true" class="d-flex flex-column align-items-center w-100">

<nav class="navbar fixed-top navbar-expand-lg navbar-dark bg-dark">

<div class="collapse navbar-collapse justify-content-around" id="navbarSupportedContent">

<div>

<a class="btn btn-light" style="min-width: 185px" asp-controller="MainPage" asp-action="Questioning">Отмена</a>

</div>

<div>

<button type="submit" style="min-width: 185px" class="btn btn-primary">Сохранить</button>

</div>

</div>

</nav>

<h3 class="mb-3">Смена пароля</h3>

<p class="mb-3">@Html.ValidationSummary(true, "", new { @class = "text-danger" })</p>

<div class="form-group w-100 col-sm-3">

<div class="mb-3 d-flex flex-column">

<label asp-for="oldPassword">Введите старый пароль:</label>

Продолжение приложения Б

<div class="d-flex flex-row w-100">

<input class="w-100 pass" type="password" name="oldPassword" value="" placeholder="Старый пароль" required />

<button calss="btn" type="button" onclick="hidePassword()">

<svg class="bi bi-eye-slash" width="1em" height="1em" viewBox="0 0 16 16" fill="currentColor" xmlns="http://www.w3.org/2000/svg">

<path d="M13.359 11.238C15.06 9.72 16 8 16 8s-3-5.5-8-5.5a7.028 7.028 0 0 0-2.79.588l.77.771A5.944 5.944 0 0 1 8 3.5c2.12 0 3.879 1.168 5.168 2.457A13.134 13.134 0 0 1 14.828 8c-.058.087-.122.183-.195.288-.335.48-.83 1.12-1.465 1.755-.165.165-.337.328-.517.486l.708.709z" />

<path d="M11.297 9.176a3.5 3.5 0 0 0-4.474-4.474l.823.823a2.5 2.5 0 0 1 2.829 2.829l.822.822zm-2.943 1.299l.822.822a3.5 3.5 0 0 1-4.474-4.474l.823.823a2.5 2.5 0 0 0 2.829 2.829z" />

<path d="M3.35 5.47c-.18.16-.353.322-.518.487A13.134 13.134 0 0 0 1.172 8l.195.288c.335.48.83 1.12 1.465 1.755C4.121 11.332 5.881 12.5 8 12.5c.716 0 1.39-.133 2.02-.36l.77.772A7.029 7.029 0 0 1 8 13.5C3 13.5 0 8 0 8s.939-1.721 2.641-3.238l.708.709z" />

<path fill-rule="evenodd" d="M13.646 14.354l-12-12 .708-.708 12 12-.708.708z" />

</svg>

<svg hidden class="bi bi-eye" width="1em" height="1em" viewBox="0 0 16 16" fill="currentColor" xmlns="http://www.w3.org/2000/svg">

<path fill-rule="evenodd" d="M16 8s-3-5.5-8-5.5S0 8 0 8s3 5.5 8 5.5S16 8 16 8zM1.173 8a13.134 13.134 0 0 0 1.66 2.043C4.12 11.332 5.88 12.5 8 12.5c2.12 0 3.879-1.168 5.168-2.457A13.134 13.134 0 0 0 14.828 8a13.133 13.133 0 0 0-1.66-2.043C11.879 4.668 10.119 3.5 8 3.5c-2.12 0-3.879 1.168-5.168 2.457A13.133 13.133 0 0 0 1.172 8z" />

<path fill-rule="evenodd" d="M8 5.5a2.5 2.5 0 1 0 0 5 2.5 2.5 0 0 0 0-5zM4.5 8a3.5 3.5 0 1 1 7 0 3.5 3.5 0 0 1-7 0z" />

</svg>

</button>

</div>

</div>

<div class="mb-3 d-flex flex-column">

<label asp-for="newPassword">Введите новый пароль:</label>

<input class="pass" type="password" name="newPassword" value="" placeholder="Подтвердите пароль" required />

</div>

<div class="mb-3 d-flex flex-column">

<label asp-for="сonfirmNewPassword">Подтвердите новый пароль:</label>

<input class="pass" type="password" name="сonfirmNewPassword" value="" placeholder="Подтвердите новый пароль" required />

</div>

</div>

</form>

</div>

LoginViewModel.cs

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Threading.Tasks;

namespace CourseProject.ViewModels

{

public class LoginViewModel

Продолжение приложения Б

{

[Required(ErrorMessage = "Не указан логин")]

public string Login { get; set; }

[Required(ErrorMessage = "Не указан пароль")]

[DataType(DataType.Password)]

public string Password { get; set; }

}

}

Login.cshtml

@model LoginViewModel

<div class="min-vh-100 min-vw-100 row align-items-center justify-content-center">

<form asp-action="Login" asp-controller="Account" asp-anti-forgery="true" class="d-flex flex-column align-items-center w-100">

<h3 class="mb-3">Авторизация</h3>

<p class="mb-3">@Html.ValidationSummary(true, "", new { @class = "text-danger" })</p>

<div class="form-group w-100 col-sm-3">

@{

if (!string.IsNullOrEmpty(Model.Login))

{

<div class="mb-3 d-flex flex-column">

<label asp-for="Login">Логин</label>

<input type="text" name="login" value="@Model.Login" placeholder="Логин" required />

</div>

}

else

{

<div class="mb-3 d-flex flex-column">

<label asp-for="Login">Логин</label>

<input type="text" name="login" value="" placeholder="Логин" required />

</div>

}

<div class="mb-3 d-flex flex-column">

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

<div class="d-flex flex-row w-100">

<input class="w-100 pass" type="password" name="password" value="" placeholder="Пароль" required />

<button calss="btn" type="button" onclick="hidePassword()">

<svg class="bi bi-eye-slash" width="1em" height="1em" viewBox="0 0 16 16" fill="currentColor" xmlns="http://www.w3.org/2000/svg">

<path d="M13.359 11.238C15.06 9.72 16 8 16 8s-3-5.5-8-5.5a7.028 7.028 0 0 0-2.79.588l.77.771A5.944 5.944 0 0 1 8 3.5c2.12 0 3.879 1.168 5.168 2.457A13.134 13.134 0 0 1 14.828 8c-.058.087-.122.183-.195.288-.335.48-.83 1.12-1.465 1.755-.165.165-.337.328-.517.486l.708.709z" />

<path d="M11.297 9.176a3.5 3.5 0 0 0-4.474-4.474l.823.823a2.5 2.5 0 0 1 2.829 2.829l.822.822zm-2.943 1.299l.822.822a3.5 3.5 0 0 1-4.474-4.474l.823.823a2.5 2.5 0 0 0 2.829 2.829z" />

<path d="M3.35 5.47c-.18.16-.353.322-.518.487A13.134 13.134 0 0 0 1.172 8l.195.288c.335.48.83 1.12 1.465 1.755C4.121 11.332 5.881 12.5 8 12.5c.716 0 1.39-.133 2.02-.36l.77.772A7.029 7.029 0 0 1 8 13.5C3 13.5 0 8 0 8s.939-1.721 2.641-3.238l.708.709z" />

Продолжение приложения Б

<path fill-rule="evenodd" d="M13.646 14.354l-12-12 .708-.708 12 12-.708.708z" />

</svg>

<svg hidden class="bi bi-eye" width="1em" height="1em" viewBox="0 0 16 16" fill="currentColor" xmlns="http://www.w3.org/2000/svg">

<path fill-rule="evenodd" d="M16 8s-3-5.5-8-5.5S0 8 0 8s3 5.5 8 5.5S16 8 16 8zM1.173 8a13.134 13.134 0 0 0 1.66 2.043C4.12 11.332 5.88 12.5 8 12.5c2.12 0 3.879-1.168 5.168-2.457A13.134 13.134 0 0 0 14.828 8a13.133 13.133 0 0 0-1.66-2.043C11.879 4.668 10.119 3.5 8 3.5c-2.12 0-3.879 1.168-5.168 2.457A13.133 13.133 0 0 0 1.172 8z" />

<path fill-rule="evenodd" d="M8 5.5a2.5 2.5 0 1 0 0 5 2.5 2.5 0 0 0 0-5zM4.5 8a3.5 3.5 0 1 1 7 0 3.5 3.5 0 0 1-7 0z" />

</svg>

</button>

</div>

</div>

}

<div class="mb-3 d-flex flex-column">

<button class="btn btn-primary" type="submit">Войти</button>

</div>

<div class="d-flex flex-column">

<lable>Нет учетной записи? <a asp-action="Register" asp-controller="Account">Зарегистрируйтесь.</a></lable>

</div>

<div class="mb-3 d-flex flex-column">

<label>Забыли пароль? <a asp-action="CheckEmail" asp-controller="Account">Восстановить.</a></label>

</div>

</div>

</form>

</div>

RegisterViewModel.cs

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Threading.Tasks;

namespace CourseProject.ViewModels

{

public class RegisterViewModel

{

public string Login { get; set; }

public string Password { get; set; }

[Compare("Password", ErrorMessage = "Пароль введен неверно")]

public string ConfirmPassword { get; set; }

public string FirstName { get; set; }

public string Name { get; set; }

public string LastName { get; set; }

[RegularExpression(@"^\w+\@\w+\.\w+", ErrorMessage = "Некорректный адрес")]

public string Email { get; set; }

Продолжение приложения Б

}

}

Register.cshtml

@model RegisterViewModel

<div class="min-vh-100 min-vw-100 row align-items-center justify-content-center">

<form asp-action="Register" asp-controller="Account" asp-anti-forgery="true" class="d-flex flex-column align-items-center w-100">

<h3 class="mb-3">Регистрация</h3>

<p class="mb-3">@Html.ValidationSummary(true, "", new { @class = "text-danger" })</p>

@{

if (!string.IsNullOrEmpty(Model.FirstName) || !string.IsNullOrEmpty(Model.Name) || !string.IsNullOrEmpty(Model.LastName) ||

!string.IsNullOrEmpty(Model.Email) || !string.IsNullOrEmpty(Model.Login) || !string.IsNullOrEmpty(Model.Password))

{

<div class="form-group w-100 col-sm-3">

<div class="mb-3 d-flex flex-column">

<label asp-for="FirstName">Фамилия</label>

<input type="text" name="FirstName" value="@Model.FirstName" placeholder="Фамилия" required />

</div>

<div class="mb-3 d-flex flex-column">

<label asp-for="Name">Имя</label>

<input type="text" name="Name" value="@Model.Name" placeholder="Имя" required />

</div>

<div class="mb-3 d-flex flex-column">

<label asp-for="LastName">Отчество</label>

<input type="text" name="LastName" value="@Model.LastName" placeholder="Отчество" required />

</div>

<div class="mb-3 d-flex flex-column">

<label asp-for="Email">Почта</label>

<input type="text" name="Email" value="@Model.Email" placeholder="Почта" required />

</div>

<div class="mb-3 d-flex flex-column">

<label asp-for="Login">Логин</label>

<input type="text" name="Login" value="@Model.Login" placeholder="Логин" required />

</div>

<div class="mb-3 d-flex flex-column">

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

<input type="password" name="Password" value="@Model.Password" placeholder="Пароль" required />

</div>

<div class="mb-3 d-flex flex-column">

<label asp-for="ConfirmPassword">Подтвердите пароль</label>

<input type="password" name="ConfirmPassword" value="" placeholder="Подтвердите пароль" required />

</div>

<div class="mb-3 d-flex flex-column">

Продолжение приложения Б

<button class="btn btn-primary" type="submit">Зарегистрироваться</button>

</div>

<div class="mb-3 d-flex flex-column">

<p>Уже зарегистрированы? <a asp-action="Login" asp-controller="Account">Войти.</a></p>

</div>

</div>

}

else

{

<div class="form-group w-100 col-sm-3">

<div class="mb-3 d-flex flex-column">

<label asp-for="FirstName">Фамилия</label>

<input type="text" name="FirstName" value="" placeholder="Фамилия" required />

</div>

<div class="mb-3 d-flex flex-column">

<label asp-for="Name">Имя</label>

<input type="text" name="Name" value="" placeholder="Имя" required />

</div>

<div class="mb-3 d-flex flex-column">

<label asp-for="LastName">Отчество</label>

<input type="text" name="LastName" value="" placeholder="Отчество" required />

</div>

<div class="mb-3 d-flex flex-column">

<label asp-for="Email">Почта</label>

<input type="text" name="Email" value="" placeholder="Почта" required />

</div>

<div class="mb-3 d-flex flex-column">

<label asp-for="Login">Логин</label>

<input type="text" name="Login" value="" placeholder="Логин" required />

</div>

<div class="mb-3 d-flex flex-column">

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

<div class="d-flex flex-row w-100">

<input class="w-100 pass" type="password" name="Password" value="" placeholder="Пароль" required />

<button calss="btn" type="button" onclick="hidePassword()">

<svg class="bi bi-eye-slash" width="1em" height="1em" viewBox="0 0 16 16" fill="currentColor" xmlns="http://www.w3.org/2000/svg">

<path d="M13.359 11.238C15.06 9.72 16 8 16 8s-3-5.5-8-5.5a7.028 7.028 0 0 0-2.79.588l.77.771A5.944 5.944 0 0 1 8 3.5c2.12 0 3.879 1.168 5.168 2.457A13.134 13.134 0 0 1 14.828 8c-.058.087-.122.183-.195.288-.335.48-.83 1.12-1.465 1.755-.165.165-.337.328-.517.486l.708.709z" />

<path d="M11.297 9.176a3.5 3.5 0 0 0-4.474-4.474l.823.823a2.5 2.5 0 0 1 2.829 2.829l.822.822zm-2.943 1.299l.822.822a3.5 3.5 0 0 1-4.474-4.474l.823.823a2.5 2.5 0 0 0 2.829 2.829z" />

<path d="M3.35 5.47c-.18.16-.353.322-.518.487A13.134 13.134 0 0 0 1.172 8l.195.288c.335.48.83 1.12 1.465 1.755C4.121 11.332 5.881 12.5 8 12.5c.716 0 1.39-.133 2.02-.36l.77.772A7.029 7.029 0 0 1 8 13.5C3 13.5 0 8 0 8s.939-1.721 2.641-3.238l.708.709z" />

<path fill-rule="evenodd" d="M13.646 14.354l-12-12 .708-.708 12 12-.708.708z" />

</svg>

<svg hidden class="bi bi-eye" width="1em" height="1em" viewBox="0 0 16 16" fill="currentColor" xmlns="http://www.w3.org/2000/svg">

Продолжение приложения Б

<path fill-rule="evenodd" d="M16 8s-3-5.5-8-5.5S0 8 0 8s3 5.5 8 5.5S16 8 16 8zM1.173 8a13.134 13.134 0 0 0 1.66 2.043C4.12 11.332 5.88 12.5 8 12.5c2.12 0 3.879-1.168 5.168-2.457A13.134 13.134 0 0 0 14.828 8a13.133 13.133 0 0 0-1.66-2.043C11.879 4.668 10.119 3.5 8 3.5c-2.12 0-3.879 1.168-5.168 2.457A13.133 13.133 0 0 0 1.172 8z" />

<path fill-rule="evenodd" d="M8 5.5a2.5 2.5 0 1 0 0 5 2.5 2.5 0 0 0 0-5zM4.5 8a3.5 3.5 0 1 1 7 0 3.5 3.5 0 0 1-7 0z" />

</svg>

</button>

</div>

</div>

<div class="mb-3 d-flex flex-column">

<label asp-for="ConfirmPassword">Подтвердите пароль</label>

<input class="pass" type="password" name="ConfirmPassword" value="" placeholder="Подтвердите пароль" required />

</div>

<div class="mb-3 d-flex flex-column">

<button class="btn btn-primary" type="submit">Зарегистрироваться</button>

</div>

<div class="mb-3 d-flex flex-column">

<p>Уже зарегистрированы? <a asp-action="Login" asp-controller="Account">Войти.</a></p>

</div>

</div>

}

}

</form>

</div>

RestorePasswordViewModel.cs

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Threading.Tasks;

namespace CourseProject.ViewModels

{

public class RestorePasswordViewModel

{

public string email { get; set; }

public string Password { get; set; }

[Compare("Password", ErrorMessage = "Пароль введен неверно")]

public string ConfirmPassword { get; set; }

public bool isPasswordСhanged { get; set; }

}

}

RestorePassword.cshtml

@model RestorePasswordViewModel

<div class="min-vh-100 min-vw-100 row align-items-center justify-content-center">

Продолжение приложения Б

@{

if (Model.isPasswordСhanged == true)

{

<div class="d-flex flex-column align-items-center w-100">

<h3 class="mb-3">Пароль был успешно изменён.</h3>

<h4 class="mb-3">Для того чтобы продолжить работу с сайтом авторизуйтесь.</h4>

<a asp-action="Login" asp-controller="Account">Перейти на страницу авторизации</a>

</div>

}

else

{

<form asp-action="RestorePassword" asp-controller="Account" asp-anti-forgery="true" class="d-flex flex-column align-items-center w-100">

<h3 class="mb-3">Восстановление паролья</h3>

<p class="mb-3">@Html.ValidationSummary(true, "", new { @class = "text-danger" })</p>

<div class="form-group w-100 col-sm-3">

<div class="mb-3 d-flex flex-column">

<label asp-for="Password">Введите новый пароль:</label>

<div class="d-flex flex-row w-100">

<input class="w-100 pass" type="password" name="Password" value="" placeholder="Пароль" required />

<button calss="btn" type="button" onclick="hidePassword()">

<svg class="bi bi-eye-slash" width="1em" height="1em" viewBox="0 0 16 16" fill="currentColor" xmlns="http://www.w3.org/2000/svg">

<path d="M13.359 11.238C15.06 9.72 16 8 16 8s-3-5.5-8-5.5a7.028 7.028 0 0 0-2.79.588l.77.771A5.944 5.944 0 0 1 8 3.5c2.12 0 3.879 1.168 5.168 2.457A13.134 13.134 0 0 1 14.828 8c-.058.087-.122.183-.195.288-.335.48-.83 1.12-1.465 1.755-.165.165-.337.328-.517.486l.708.709z" />

<path d="M11.297 9.176a3.5 3.5 0 0 0-4.474-4.474l.823.823a2.5 2.5 0 0 1 2.829 2.829l.822.822zm-2.943 1.299l.822.822a3.5 3.5 0 0 1-4.474-4.474l.823.823a2.5 2.5 0 0 0 2.829 2.829z" />

<path d="M3.35 5.47c-.18.16-.353.322-.518.487A13.134 13.134 0 0 0 1.172 8l.195.288c.335.48.83 1.12 1.465 1.755C4.121 11.332 5.881 12.5 8 12.5c.716 0 1.39-.133 2.02-.36l.77.772A7.029 7.029 0 0 1 8 13.5C3 13.5 0 8 0 8s.939-1.721 2.641-3.238l.708.709z" />

<path fill-rule="evenodd" d="M13.646 14.354l-12-12 .708-.708 12 12-.708.708z" />

</svg>

<svg hidden class="bi bi-eye" width="1em" height="1em" viewBox="0 0 16 16" fill="currentColor" xmlns="http://www.w3.org/2000/svg">

<path fill-rule="evenodd" d="M16 8s-3-5.5-8-5.5S0 8 0 8s3 5.5 8 5.5S16 8 16 8zM1.173 8a13.134 13.134 0 0 0 1.66 2.043C4.12 11.332 5.88 12.5 8 12.5c2.12 0 3.879-1.168 5.168-2.457A13.134 13.134 0 0 0 14.828 8a13.133 13.133 0 0 0-1.66-2.043C11.879 4.668 10.119 3.5 8 3.5c-2.12 0-3.879 1.168-5.168 2.457A13.133 13.133 0 0 0 1.172 8z" />

<path fill-rule="evenodd" d="M8 5.5a2.5 2.5 0 1 0 0 5 2.5 2.5 0 0 0 0-5zM4.5 8a3.5 3.5 0 1 1 7 0 3.5 3.5 0 0 1-7 0z" />

</svg>

</button>

</div>

</div>

<div class="mb-3 d-flex flex-column">

<label asp-for="ConfirmPassword">Подтвердите новый пароль:</label>

<input class="pass" type="password" name="ConfirmPassword" value="" placeholder="Подтвердите пароль" required />

</div>

<div class="mb-3 d-flex flex-column">

Продолжение приложения Б

<button type="submit">Сохранить</button>

</div>

<div class="mb-3 d-flex flex-column">

<p>Уже зарегистрированы? <a asp-action="Login" asp-controller="Account">Войти.</a></p>

</div>

</div>

</form>

}

}

</div>

VerifiedMessage.cshtml

<div class="d-flex flex-column align-items-center w-100 h-100">

<div class="d-flex flex-column align-items-center w-75 h-100 text-center mt-5">

<h4>Спасибо за регистрацию.</h4>

<h5>Для того, чтобы продолжить работу Вам необходимо подтвердить свой аккаунт. Письмо для подтверждения было выслано на указанную вами почту. Перейдите по ссылке в письме и авториуйтесь чтобы начать создавать опросы.</h5>

</div>

</div>

EditPoll

Add.cshtml

@model tPoll

@{

string creatorLogin = Model.CreatorLogin;

}

<div class="d-flex flex-column align-items-center w-100 h-100 mt-5">

@{

<form id="questions" class="d-flex flex-column align-items-center w-100 h-100 mt-2 mb-3 " asp-action="SaveAdd" method="post">

<input hidden asp-for="CreatorLogin" value="@Model.CreatorLogin" name="CreatorLogin" />

<input hidden asp-for="Number" value="@Model.Number" name="Number" />

<nav class="navbar fixed-top navbar-expand-lg navbar-dark bg-dark">

<div class="collapse navbar-collapse justify-content-around" id="navbarSupportedContent">

<div>

<button style="min-width: 185px" type="button" class="btn btn-success" onclick="addFieldQuestion()">Добавить вопрос</button>

</div>

<div>

<a style="min-width: 185px" class="btn btn-light" asp-controller="MainPage" asp-action="Questioning" >Отмена</a>

<button style="min-width: 185px" type="submit" class="btn btn-primary">Сохранить</button>

</div>

</div>

</nav>

<div class="w-75 h-100 mt-4 mb-3 text-center">

Продолжение приложения Б

<h3> Страница создания опросов </h3>

</div>

<div class=" w-75 h-100 d-flex flex-column mt-4 mb-3 ">

<h6 class="w-100 h-100">Наименование опроса</h6>

<input asp-for="PollText" class="form-control" type="text" value="@Model.PollText" required />

</div>

</form>

}

</div>

Edit.cshtml

@model tPoll

@{

string creatorLogin = "";

if (Model.CreatorLogin != null)

{

creatorLogin = Model.CreatorLogin;

}

if (Model.tQuestion != null)

{

int idAddQuestion = Model.tQuestion.Count;

}

}

<div class="d-flex flex-column align-items-center w-100 h-100 mt-5">

@{

<form id="questions" class="d-flex flex-column align-items-center w-100 h-100 mt-2 mb-3 " asp-action="SaveEdit" method="post">

<input hidden asp-for="CreatorLogin" value="@Model.CreatorLogin" name="CreatorLogin" />

<input hidden asp-for="Number" value="@Model.Number" name="Number" />

<nav class="navbar fixed-top navbar-expand-lg navbar-dark bg-dark">

<div class="collapse navbar-collapse justify-content-around" id="navbarSupportedContent">

<div>

<button style="min-width: 185px" type="button" class="btn btn-success" onclick="addFieldQuestion()">Добавить вопрос</button>

</div>

<div>

<a style="min-width: 185px" class="btn btn-light" asp-controller="MainPage" asp-action="Questioning">Отмена</a>

<button style="min-width: 185px" type="submit" class="btn btn-primary">Сохранить</button>

</div>

</div>

</nav>

<div class="w-75 h-100 mt-4 mb-3 text-center">

<h3> Страница редактирования опросов </h3>

</div>

<div class=" w-75 h-100 d-flex flex-column mt-4 mb-3 ">

<h6 class="w-100 h-100">Наименование опроса</h6>

Продолжение приложения Б

<input asp-for="PollText" class="form-control" type="text" value="@Model.PollText" required />

</div>

@{

int i = 0;

if (Model.tQuestion != null)

{

@foreach (tQuestion tquestion in Model.tQuestion)

{

int idAddResponse = tquestion.tResponse.Count;

<div id="question\_@tquestion.Number" class=" w-75 h-100 d-flex flex-column mt-4 mb-3 ">

<div class="d-flex flex-row justify-content-start text-center align-items-center mb-2">

<h6 class="w-100 h-100 quest" style="max-width: 100px">• Вопрос №</h6>

<input asp-for="tQuestion[i].Number" class="form-control mr-2" style="max-width: 100px" type="number" value="@tquestion.Number" required />

<h6 class="w-100 h-100" style="max-width: 100px"> Ответы: </h6>

<select asp-for="tQuestion[i].Type" class="form-control mr-2" style="max-width: 150px" onchange="addFieldCountResponseInQuestion(@(tquestion.Number))">

@{

if (tquestion.Type == "CheckBox")

{

<option selected value="CheckBox">Несколько</option>

}

else

{

<option value="CheckBox">Несколько</option>

}

if (tquestion.Type == "Radio")

{

<option selected value="Radio">Один</option>

}

else

{

<option value="Radio">Один</option>

}

if (tquestion.Type == "")

{

<option selected value="">Без выбора</option>

}

else

{

<option value="">Без выбора</option>

}

}

</select>

<h6 class="w-100 h-100" style="max-width: 150px"> Мин. ответов: </h6>

@{

if (tquestion.IsRequired == "true" && tquestion.Type == "CheckBox")

{

Продолжение приложения Б

<input asp-for="tQuestion[i].minCountResponse" class="form-control mr-2" style="max-width: 100px" type="number" value="@tquestion.minCountResponse" />

}

else

{

<input asp-for="tQuestion[i].minCountResponse" class="form-control mr-2" style="max-width: 100px" type="number" value="@tquestion.minCountResponse" disabled/>

}

}

<a class="btn btn-success text-light mr-2" onclick="addFieldResponse(@tquestion.Number)">Добавить ответ</a>

<a id="@tquestion.Number" class="btn btn-danger text-light" onclick="deleteFieldQuestion(this)">x</a>

</div>

<input asp-for="tQuestion[i].QuestionText" class="form-control" type="text" value="@tquestion.QuestionText" required />

<div class="mb-3">

@{

if (@tquestion.IsRequired == "true")

{

<input asp-for="tQuestion[i].IsRequired" onchange="valueChecked(this); addFieldCountResponseInQuestion(@(tquestion.Number));" class="form-check-input" type="checkbox" value="true" checked />

}

else

{

<input asp-for="tQuestion[i].IsRequired" onchange="valueChecked(this); addFieldCountResponseInQuestion(@(tquestion.Number));" class="form-check-input" type="checkbox" value="false" />

}

<lable class="form-check-label mb-2">Обязательный вопрос?</lable>

int j = 0;

}

</div>

@if (tquestion.tResponse != null)

{

@foreach (tResponse tresponse in tquestion.tResponse)

{

<div id="response\_@(tquestion.Number)\_@tresponse.Number">

<div class="d-flex flex-row justify-content-start text-center align-items-center mb-2">

<h6 class="w-100 h-100" style="max-width: 100px">Ответ №</h6>

<input asp-for="tQuestion[i].tResponse[j].Number" class="form-control mr-2" style="max-width: 100px" type="number" value="@tresponse.Number" required />

<a id="@(tquestion.Number)\_@tresponse.Number" class="btn btn-danger text-light" onclick="deleteFieldResponse(this)">x</a>

</div>

<input asp-for="tQuestion[i].tResponse[j].ResponseText" class="form-control" type="Text" value="@tresponse.ResponseText" required />

<div class="mb-3">

@{

if (@tresponse.UseTextBox == "true")

Продолжение приложения Б

{

<input asp-for="tQuestion[i].tResponse[j].UseTextBox" onchange="valueChecked(this)" class="form-check-input" type="checkbox" value="true" checked />

}

else

{

<input asp-for="tQuestion[i].tResponse[j].UseTextBox" onchange="valueChecked(this)" class="form-check-input" type="checkbox" value="false" />

}

<lable class="form-check-label">Добавить поле для ввода?</lable>

j++;

}

</div>

</div>

}

}

@{

i++;

}

</div>

}

}

}

</form>

}

</div>

MainPage

CompletionPoll.cshtml

<div class="min-vh-100 min-vw-100 row align-items-center justify-content-center">

<div class="justify-content-center text-center align-items-center w-75 h-100 d-flex flex-column">

<h1>@ViewBag.Title</h1>

<!--<h3>@ViewBag.Message</h3>-->

<a id="buttonLogout" class="btn btn-primary m-2" style="max-width: 300px" asp-controller="Account" asp-action="Logout">Вернуться на страницу авторизации</a>

</div>

</div>

PassingPollViewModel.cs

using CourseProject.Data.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace CourseProject.ViewModels

{

public class PassingPollViewModel

{

public Creator creator { get; set; }

public User user { get; set; }

public tPoll poll { get; set; }

Продолжение приложения Б

public tAnswer userAnswer { get; set; }

}

}

PassingPoll.cshtml

@model PassingPollViewModel

@{

tPoll poll = Model.poll;

}

<div class="d-flex flex-column align-items-center w- w-100 h-100">

@{

if (!String.IsNullOrEmpty(Model.poll.PollText))

{

int i = 0;

<form id="form\_@(poll.CreatorLogin)\_@poll.Number" style="display: flex" class=" flex-column align-items-center w-100 h-100" asp-action="CompletionPoll" asp-controller="MainPage" method="post">

<nav class="navbar fixed-top navbar-expand-lg navbar-dark bg-dark mb-5">

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarTogglerDemo01" aria-controls="navbarTogglerDemo01" aria-expanded="false" aria-label="Toggle navigation">

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

</button>

<div class=" collapse navbar-collapse justify-content-around" id="navbarTogglerDemo01">

<div class="align-items-center w-75">

<button style="min-width: 185px" type="button" onclick="submitForm('form\_@(poll.CreatorLogin)\_@poll.Number')" class="btn btn-primary m-2">Отправить</button>

</div>

<div class="d-flex flex-row">

<a style="min-width: 185px" id="buttonLogout" class="btn btn-danger m-2" asp-controller="Account" asp-action="Logout">Выход</a>

</div>

</div>

</nav>

<div class="mt-5"></div>

<div class="text-center mt-5">

<h4>Инструкция прохождения опроса:</h4>

<h5>1. Вопросы отмеченные символом «<b style="color: red">\*</b>» являються обязательными.</h5>

<h5>2. Некоторые вопросы должны иметь более 1 ответа.</h5>

<h5>3. Вы не сможете завершить опрос, пока не ответите на все обязательные вопросы.</h5>

<h5>4. При обнаружении ошибки после отправки результатов опроса Вы получите уведомление.</h5>

<h5>5. После нажатия кнопки «Отправить» Ваши ответы будут обработаны и опрос завершится. Повторно опрос пройти нельзя.</h5>

Продолжение приложения Б

</div>

<h3 id="pollText\_@(poll.CreatorLogin)\_@poll.Number" class="w-75 h-100 mt-5 mb-4 text-center">«@poll.PollText»</h3>

@foreach (tQuestion tquestion in poll.tQuestion)

{

<input asp-for="userAnswer.QuestionsLists[i].typeQuestion" value="@tquestion.Type" hidden />

<div id="div\_tquestion@(i+1)" class="custom-control w-75 h-100 d-flex flex-column mt-4 mb-3 isRequired" data-isRequired="">

@{

int j = 0;

if (tquestion.IsRequired == "true")

{

<h5 class="mb-3">

<input asp-for="userAnswer.QuestionsLists[i].numberQuestion" value="@(tquestion.Number)" hidden /> @(tquestion.Number).

<input asp-for="userAnswer.QuestionsLists[i].textQuestion" value="@tquestion.QuestionText" hidden /> @tquestion.QuestionText <b style="color: red">\*</b>

</h5>

<script>

fillcheckPoll("tquestion@(tquestion.Number)");

</script>

@foreach (tResponse tresponse in tquestion.tResponse)

{

<div>

@{

if (tresponse.UseTextBox == "true")

{

<input asp-for="userAnswer.QuestionsLists[i].ResponsesLists[j].isUsedTextBox" value="true" hidden />

}

else

{

<input asp-for="userAnswer.QuestionsLists[i].ResponsesLists[j].isUsedTextBox" value="false" hidden />

}

if (!string.IsNullOrEmpty(tquestion.Type))

{

<input id="tquestion@(tquestion.Number)\_tresponse@(tresponse.Number)\_h" asp-for="userAnswer.QuestionsLists[i].ResponsesLists[j].isChecked" value="" hidden />

Продолжение приложения Б

<input id="tquestion@(tquestion.Number)\_tresponse@(tresponse.Number)" data-checkQuestion="tquestion@(tquestion.Number)" data-isRequired="true" data-minResponse="@(tquestion.minCountResponse)" data-maxResponse="@(tquestion.maxCountResponse)" onchange="checkQuestion(this); checkInputQuestion(this);" class="tquestion@(tquestion.Number) form-check-input mb-2" type="@tquestion.Type" name="tquestion@(tquestion.Number)" value="">

<label class="form-check-label mb-2" for="">

<input asp-for="userAnswer.QuestionsLists[i].ResponsesLists[j].numberResponse" value="@(tresponse.Number)" hidden /> @(tresponse.Number).

<input asp-for="userAnswer.QuestionsLists[i].ResponsesLists[j].textResponse" value="@tresponse.ResponseText" hidden /> @tresponse.ResponseText

</label>

if (tresponse.UseTextBox == "true")

{

<input id="tquestion@(tquestion.Number)\_tresponse@(tresponse.Number)\_input" class="form-control mb-2" type="text" placeholder="@tresponse.ResponseText" asp-for="userAnswer.QuestionsLists[i].ResponsesLists[j].inputTextResponse" disabled>

}

}

else

{

<label class="form-check-label mb-2" for="">

<input asp-for="userAnswer.QuestionsLists[i].ResponsesLists[j].numberResponse" value="@(tresponse.Number)" hidden /> @(tresponse.Number).

<input asp-for="userAnswer.QuestionsLists[i].ResponsesLists[j].textResponse" value="@tresponse.ResponseText" hidden /> @tresponse.ResponseText

</label>

if (tresponse.UseTextBox == "true")

{

<input id="tquestion@(tquestion.Number)\_tresponse@(tresponse.Number)\_input" data-checkQuestion="tquestion@(tquestion.Number)" data-isRequired="true" data-minResponse="@(tquestion.minCountResponse)" data-maxResponse="@(tquestion.maxCountResponse)" onchange="checkQuestion(this)" class="tquestion@(tquestion.Number) form-control mb-2" type="text" placeholder="@tresponse.ResponseText" asp-for="userAnswer.QuestionsLists[i].ResponsesLists[j].inputTextResponse">

}

}

j++;

}

</div>

}

}

else

{

Продолжение приложения Б

<h5 class="mb-3">

<input asp-for="userAnswer.QuestionsLists[i].numberQuestion" value="@(tquestion.Number)" hidden /> @(tquestion.Number).

<input asp-for="userAnswer.QuestionsLists[i].textQuestion" value="@tquestion.QuestionText" hidden /> @tquestion.QuestionText

</h5>

@foreach (tResponse tresponse in tquestion.tResponse)

{

<div>

@{

if (tresponse.UseTextBox == "true")

{

<input asp-for="userAnswer.QuestionsLists[i].ResponsesLists[j].isUsedTextBox" value="true" hidden />

}

else

{

<input asp-for="userAnswer.QuestionsLists[i].ResponsesLists[j].isUsedTextBox" value="false" hidden />

}

if (!string.IsNullOrEmpty(tquestion.Type))

{

<input id="tquestion@(tquestion.Number)\_tresponse@(tresponse.Number)\_h" asp-for="userAnswer.QuestionsLists[i].ResponsesLists[j].isChecked" value="" hidden />

<input id="tquestion@(tquestion.Number)\_tresponse@(tresponse.Number)" onchange="checkInputQuestion(this);" class="form-check-input mb-2" type="@tquestion.Type" name="@tquestion.Number" value="">

<label class="form-check-label mb-2" for="@(tquestion.Number)\_@tresponse.Number">

<input asp-for="userAnswer.QuestionsLists[i].ResponsesLists[j].numberResponse" value="@(tresponse.Number)" hidden /> @(tresponse.Number).

<input asp-for="userAnswer.QuestionsLists[i].ResponsesLists[j].textResponse" value="@tresponse.ResponseText" hidden /> @tresponse.ResponseText

</label>

if (tresponse.UseTextBox == "true")

{

<input id="tquestion@(tquestion.Number)\_tresponse@(tresponse.Number)\_input" class="form-control mb-2" type="text" placeholder="@tresponse.ResponseText" asp-for="userAnswer.QuestionsLists[i].ResponsesLists[j].inputTextResponse" disabled>

}

}

else

{

<label class="form-check-label mb-2" for="@(tquestion.Number)[\_@tresponse.Number](mailto:_@tresponse.Number)">

Продолжение приложения Б

<input asp-for="userAnswer.QuestionsLists[i].ResponsesLists[j].numberResponse" value="@(tresponse.Number)" hidden /> @(tresponse.Number).

<input asp-for="userAnswer.QuestionsLists[i].ResponsesLists[j].textResponse" value="@tresponse.ResponseText" hidden /> @tresponse.ResponseText

</label>

if (tresponse.UseTextBox == "true")

{

<input id="tquestion@(tquestion.Number)\_tresponse@(tresponse.Number)\_input" class="form-control mb-2" type="text" placeholder="@tresponse.ResponseText" asp-for="userAnswer.QuestionsLists[i].ResponsesLists[j].inputTextResponse">

}

}

j++;

}

</div>

}

}

i++;

}

</div>

}

<div hidden>

<input asp-for="userAnswer.creatorLogin" value="@Model.creator.login">

<input asp-for="userAnswer.pollNumber" value="@Model.poll.Number">

<input asp-for="userAnswer.userLogin" value="@Model.user.login">

</div>

</form>

}

else

{

<div class="flex-column align-items-center w-75 h-100 mt-5 mb-3">

<h3 class="flex-column text-center w-100 h-100 mt-5 mb-3">Опрос ещё не создан!</h3>

</div>

}

}

</div>

Questioning.cshtml

@model List<tPoll>

@{

string PollLogin = "";

int PollNumber = 0;

string editPollText = "";

int countPoll = 0;

int limitCountPoll = 10;

if (Model.Count > 0)

Продолжение приложения Б

{

PollLogin = Model[0].CreatorLogin;

PollNumber = Model[0].Number;

editPollText = Model[0].PollText;

}

}

<div class="d-flex flex-column align-items-center w- w-100 h-100">

<nav class="navbar fixed-top navbar-expand-lg navbar-dark bg-dark align-content-center">

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarTogglerDemo01" aria-controls="navbarTogglerDemo01" aria-expanded="false" aria-label="Toggle navigation">

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

</button>

<div class=" collapse navbar-collapse justify-content-center align-content-center align-items-center text-center" id="navbarTogglerDemo01">

<div class=" w-75 d-flex flex-column ">

@{

<div class="row justify-content-around align-items-center m-1">

<div class="d-flex flex-column ">

<label class="nav-item text-light m-0" style="min-width: 150px">Вы вошли как:</label>

<div class="nav-item dropdown">

<a class="nav-link dropdown-toggle text-light" href="#" id="navbarDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">

@PollLogin

</a>

<div class="dropdown-menu" aria-labelledby="navbarDropdown">

<a class="dropdown-item" asp-controller="Account" asp-action="EditPassword">Изменить пароль</a>

<div class="dropdown-divider"></div>

<a class="dropdown-item" asp-controller="Account" asp-action="EditEmail">Изменить эл. почту</a>

<div class="dropdown-divider"></div>

<a class="dropdown-item" asp-controller="Account" asp-action="EditFullName">Изменить Ф.И.О.</a>

</div>

</div>

<!-- <a class="nav-item text-light" style="min-width: 150px" asp-controller="MainPage" asp-action="CreatorProfile"><u>@PollLogin</u></a> -->

</div>

<div class="d-flex flex-column">

<label class="nav-item text-light ml-2" style="min-width: 150px">Показать опрос: </label>

@{

if (!String.IsNullOrEmpty(Model[0].PollText))

{

<select id="numberPoll" class="custom-select ml-2" style="max-width: 500px;" onchange="showForm(this,'@Model[0].CreatorLogin')">

@{

foreach (tPoll poll in Model)

{

if (poll.Number == PollNumber)

Продолжение приложения Б

{

<option data-numberPoll="@poll.Number" selected value="@poll.Number">@poll.PollText</option>

}

else

{

<option data-numberPoll="@poll.Number" value="@poll.Number">@poll.PollText</option>

}

countPoll++;

}

}

</select>

}

else

{

<select id="numberPoll" class="custom-select " style="max-width: 100px">

<option value="-">-</option>

</select>

}

}

</div>

<div class="d-flex flex-column ">

<a style="min-width: 185px" id="buttonLogout" class="btn btn-danger m-1" asp-controller="Account" asp-action="Logout">Выход</a>

</div>

</div>

<div class="row justify-content-around align-items-center">

@{

if (!String.IsNullOrEmpty(Model[0].PollText))

{

//align-content-center align-items-center text-center

<a style="min-width: 185px" id="buttonEdit" class="btn btn-primary m-1" asp-controller="EditPoll" asp-action="Edit" asp-route-editPollNumber="@PollNumber">Редактировать опрос</a>

<a style="min-width: 185px" class="btn btn-primary m-1 text-light" data-toggle="modal" data-target="#exampleModal">Удалить опрос</a>

<a style="min-width: 185px" id="buttonUser" class="btn btn-primary m-1 text-light" asp-controller="User" asp-action="ShowUser" asp-route-pollNumber="@PollNumber">Опрашиваемые</a>

<a style="min-width: 185px" id="buttonExport" class="btn btn-primary m-1 text-light" asp-controller="MainPage" asp-action="ExportPoll" asp-route-creatorLogin="@PollLogin" asp-route-pollNumber="@PollNumber">Экспорт опроса</a>

}

if (countPoll < limitCountPoll)

{

<a style="min-width: 185px" id="\_buttonImport" class="btn btn-primary m-1 text-light" data-toggle="modal" data-target="#exampleModal2">Импорт опроса</a>

<a style="min-width: 185px" id="buttonAdd" class="btn btn-primary m-1" asp-controller="EditPoll" asp-action="Add">Добавить опрос</a>

Продолжение приложения Б

}

else

{

<a style="min-width: 185px" id="\_buttonImport" class="btn btn-primary m-1 text-light" onclick="alert('На одном аккаунте можно иметь максимум 5 опросов.\nДля того чтобы импортировать опрос удалите один из существующих.')">Импорт опроса</a>

<a style="min-width: 185px" id="buttonAdd" class="btn btn-primary m-1 text-light" onclick="alert('На одном аккаунте можно иметь максимум 5 опросов.\nДля того чтобы создать новый опрос удалите один из существующих.')">Добавить опрос</a>

}

}

</div>

}

</div>

</div>

</nav>

<div>

<br />

<br />

<br />

<br />

</div>

@{

if (!String.IsNullOrEmpty(Model[0].PollText))

{

foreach (tPoll poll in Model)

{

if (poll.Number == PollNumber)

{

<form id="form\_@poll.Number" style="display: flex" class=" flex-column align-items-center w-100 h-100 mt-5 mb-3" action="/" method="post">

<div class="w-75 h-100 mt-5 mb-4 text-center">

<h3> Страница просмотра опросов </h3>

</div>

<div class="w-75 h-100 mb-4 text-center">

<h3>«@poll.PollText»</h3>

</div>

@foreach (tQuestion tquestion in poll.tQuestion)

{

<div class=" w-75 h-100 d-flex flex-column mt-4 mb-3">

<h5>

@tquestion.Number. @tquestion.QuestionText

@if (tquestion.IsRequired == "true")

{

<b style="color: red">\*</b>

}

</h5>

Продолжение приложения Б

<div class="mb-3">

@if (tquestion.minCountResponse > 1)

{

<small class="form-text text-muted">Для ответа необходимо выбрать минимум @tquestion.minCountResponse ответа</small>

}

</div>

@foreach (tResponse tresponse in tquestion.tResponse)

{

<div>

@{

if (!string.IsNullOrEmpty(tquestion.Type))

{

<input disabled id="" class="form-check-input mb-2" type="@tquestion.Type" name="" value="">

}

<label class="form-check-label mb-2 text-dark">

@tresponse.Number. @tresponse.ResponseText

</label>

if (tresponse.UseTextBox == "true")

{

<input disabled id="" class="form-control mb-2" type="text" placeholder="@tresponse.ResponseText">

}

}

</div>

}

</div>

}

</form>

}

else

{

<form id="form\_@poll.Number" style="display: none" class=" flex-column align-items-center w-100 h-100 mt-5 mb-3" action="/" method="post">

<div class="w-75 h-100 mt-5 mb-4 text-center">

<h3> Страница просмотра опросов </h3>

</div>

<div class="w-75 h-100 mb-4 text-center">

<h3>«@poll.PollText»</h3>

</div>

@{

if (poll.tQuestion != null)

{

@foreach (tQuestion tquestion in poll.tQuestion)

{

<div class=" w-75 h-100 d-flex flex-column mt-4 mb-3">

<h5>

@tquestion.Number. @tquestion.QuestionText

@if (tquestion.IsRequired == "true")

{

Продолжение приложения Б

<b style="color: red">\*</b>

}

</h5>

<div class="mb-3">

@if (tquestion.minCountResponse > 1)

{

<small class="form-text text-muted">Для ответа необходимо выбрать минимум @tquestion.minCountResponse ответа</small>

}

</div>

@{

if (tquestion.tResponse != null)

{

@foreach (tResponse tresponse in tquestion.tResponse)

{

<div>

@{

if (!string.IsNullOrEmpty(tquestion.Type))

{

<input disabled id="" class="form-check-input mb-2" type="@tquestion.Type" name="" value="">

}

<label class="form-check-label mb-2 text-dark">

@tresponse.Number. @tresponse.ResponseText

</label>

if (tresponse.UseTextBox == "true")

{

<input disabled id="" class="form-control mb-2" type="text" placeholder="@tresponse.ResponseText">

}

}

</div>

}

}

}

</div>

}

}

}

</form>

}

}

}

else

{

<div class="flex-column align-items-center w-75 h-100 mt-5 mb-3">

<h3 class="flex-column text-center w-100 h-100 mt-5 mb-3">Вы ещё не добавили ни одного опроса.</h3>

</div>

}

}

Продолжение приложения Б

</div>

<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">

<div class="modal-dialog" role="document">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalLabel">Удалить безвозвратно опрос №@(PollNumber)?</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<p>Наименование опроса:</p>

<p id="modalBodyPollText">@(editPollText)</p>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-primary" data-dismiss="modal">Отмена</button>

<a id="buttonDelete" class="btn btn-danger" asp-controller="Delete" asp-action="EditPoll" asp-route-deletePollLogin="@PollLogin" asp-route-deletePollNumber="@PollNumber">Удалить</a>

</div>

</div>

</div>

</div>

<div class="modal fade" id="exampleModal2" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">

<div class="modal-dialog" role="document">

<form id="ImportForm" class="modal-content" asp-action="ImportPoll" asp-controller="MainPage" method="post" enctype="multipart/form-data" asp-route-creatorLogin="@PollLogin">

<div class="modal-header">

<h5 class="modal-title" id="exampleModal2Label"> Импорт</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<p>Выберите файл опроса в формате \*.json. После загрузки он автоматически будет добавлен в список ваших опросов.</p>

<input id="inputFile" class="mb-2" type="file" name="uploadedFile" accept="application/json" />

<input class="mb-2" type="text" value="@PollLogin" hidden />

</div>

<div class="modal-footer">

<button type="button" class="btn btn-primary" data-dismiss="modal">Отмена</button>

<input id="buttonImport" type="button" class="btn btn-primary" value="Импортировать" onclick="checkImportForm()" />

</div>

</form>

</div>

</div>

@{

Продолжение приложения Б

if (ViewData["action"].ToString() == "password")

{

<script>

$(window).load(function () {

alert("Пароль был успешно изменён.");

});

</script>

}

if (ViewData["action"].ToString() == "fullname")

{

<script>

$(window).load(function () {

alert("Ф.И.О. были успешно изменёны.");

});

</script>

}

if (ViewData["action"].ToString() == "email")

{

<script>

$(window).load(function () {

alert("Эл. почта была успешно изменена.");

});

</script>

}

}

Shared

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>@ViewBag.Title</title>

<script src="~/js/jquery-3.4.1.min.js"></script>

<link href="~/css/bootstrap.css" rel="stylesheet" type="text/css" />

<link href="~/css/style.css" rel="stylesheet" type="text/css" />

<script src="~/js/script.js"></script>

<script src="~/js/bootstrap.js"></script>

<link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">

<script src="https://code.jquery.com/jquery-1.12.4.js"></script>

<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>

</head>

<body class="container-fluid w-100 h-100">

@RenderBody()

</body>

</html>

ShowUser

UserViewModel.cs

using CourseProject.Data.Models;

using System;

Продолжение приложения Б

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Threading.Tasks;

namespace CourseProject.ViewModels

{

public class UserViewModel

{

public IEnumerable<User> User { get; set; }

public string creatorLogin { get; set; }

public int pollNumber { get; set; }

public int countUser { get; set; }

public string institution { get; set; }

public string deleteUserList { get; set; }

public string date { get; set; }

}

}

ShowUser.cshtml

@model UserViewModel

@{

}

<script>

$(function () {

$("#datepicker").datepicker({

dateFormat: 'dd.mm.yy',

dayNamesMin: ["Пн", "Вт", "Ср", "Чт", "Пт", "Сб", "Вс"],

monthNames: ["Январь", "Февраль", "Март", "Апрель", "Май", "Июнь", "Июль", "Август", "Сентябрь", "Октябрь", "Ноябрь", "Декабрь"],

maxDate: "+0d"

});

});

</script>

<form id="UserForm" class="d-flex flex-column align-items-center w-100 h-100" asp-action="DeleteUsers" asp-controller="User">

<nav class="navbar fixed-top navbar-expand-lg navbar-dark bg-dark">

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarTogglerDemo01" aria-controls="navbarTogglerDemo01" aria-expanded="false" aria-label="Toggle navigation">

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

</button>

<div class=" collapse navbar-collapse justify-content-around" id="navbarTogglerDemo01">

<div class="align-items-center w-75">

<lable class="nav-item text-light m-2" style="min-width: 150px">Номер опроса: @Model.pollNumber </lable>

<lable class="nav-item text-light m-2" style="min-width: 150px">Количество аккаунтов: @Model.User.Count() </lable>

<lable class="nav-item text-light m-2" style="min-width: 150px">Опрос прошли: @Model.User.Where(i => i.isPassed == 1).Count() </lable>

@{

Продолжение приложения Б

if (@Model.User.Where(i => i.isPassed == 1).Count() == 0)

{

<a style="min-width: 185px" class="btn btn-primary m-2 text-light" onclick="alert('Для формирования отчёта опрос должен пройти хотябы 1 опрашиваемый!')">Сформировать отчёт</a>

}

else

{

<a style="min-width: 185px" id="" class="btn btn-primary m-2 text-light" data-toggle="modal" data-target="#exampleModal2">Сформировать отчёт</a>

}

}

<a style="min-width: 185px" id="buttonLogout" class="btn btn-primary m-2 text-light" data-toggle="modal" data-target="#exampleModal">Добавить аккаунты</a>

<button style="min-width: 185px" type="button" onclick="submitUserForm()" class="btn btn-danger text-light">Удалить</button>

</div>

<div class="d-flex flex-row">

<a style="min-width: 185px" id="buttonLogout" class="btn btn-danger m-2" asp-controller="MainPage" asp-action="Questioning">Назад</a>

</div>

</div>

</nav>

<div class="d-flex flex-column w-100 h-100 align-items-center mt-5">

<div class="w-75 h-100 mt-5 mb-3 text-center">

<h3> Данные опрашиваемых </h3>

</div>

<div class="d-flex flex-column w-75 h-100 align-items-center mt-4">

<table class="table">

<thead>

<tr>

<th scope="col">

<div class="d-flex justify-content-center">

<input id="checkedAllDeleteCheckBox" onchange="allDeleteCheckBox(this)" type="checkbox" />

</div>

</th>

<!-- <th scope="col">Код</th>-->

<th scope="col">Логин</th>

<th scope="col">Пароль</th>

<th scope="col">Опрос пройден?</th>

<th scope="col">Учебное заведение</th>

<th scope="col"></th>

</tr>

</thead>

<tbody>

@{

foreach (User user in Model.User)

{

<tr>

<th scope="row">

Продолжение приложения Б

<div class="d-flex justify-content-center">

<input id="@(user.login)" class="deleteUserCheckbox" onchange="checkUserDelete(this)" type="checkbox" />

</div>

</th>

<!-- <td>@(user.id)</td> -->

<td>@(user.login)</td>

<td>@(user.password)</td>

<td>

@{

if (user.isPassed == 0)

{

<p>Нет</p>

}

else

{

<p>Да</p>

}

}

</td>

<td>@(user.institution)</td>

<td><a id="buttonDeleteUser\_@(user.login)" class="btn btn-danger text-light" asp-controller="User" asp-action="DeleteUser" asp-route-creatorLogin="@Model.creatorLogin" asp-route-pollNumber="@Model.pollNumber" asp-route-userLogin="@user.login">Удалить</a></td>

</tr>

}

}

</tbody>

</table>

</div>

</div>

<input id="deleteUserList" asp-for="deleteUserList" hidden value="">

<input asp-for="creatorLogin" hidden value="@Model.creatorLogin">

<input asp-for="pollNumber" hidden value="@Model.pollNumber">

</form>

<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">

<div class="modal-dialog" role="document">

<form asp-action="AddUsers" asp-controller="User" class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalLabel"> Сгенерировать аккаунты.</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<div class="form-group" hidden>

<input asp-for="creatorLogin" type="text" value="@Model.creatorLogin" class="form-control">

<input asp-for="pollNumber" type="number" value="@Model.pollNumber" class="form-control">

Продолжение приложения Б

</div>

<div class="form-group">

<label for="countUser">Количество аккаунтов:</label>

<input asp-for="countUser" type="number" value="1" min="1" max="500" class="form-control" id="countUser" aria-describedby="countUserHelper">

<small id="countUserHelper" class="form-text text-muted">За раз можно создать не более 500 аккаунтов.</small>

</div>

<div class="form-group">

<label for="exampleInputInstitution">Учебное заведение:</label>

<input asp-for="institution" type="text" class="form-control" id="exampleInputInstitution">

</div>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-primary" data-dismiss="modal">Отмена</button>

<button type="submit" class="btn btn-primary">Создать</button>

</div>

</form>

</div>

</div>

<div class="modal fade" id="exampleModal2" tabindex="-1" role="dialog" aria-labelledby="exampleModal2Label" aria-hidden="true">

<div class="modal-dialog" role="document">

<form asp-action="CreateReport" asp-controller="User" class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModal2Label"> Сформировать отчёт.</h5>

<button type="button" class="close" data-dismiss="modal" aria-label="Close">

<span aria-hidden="true">&times;</span>

</button>

</div>

<div class="modal-body">

<div class="form-group" hidden>

<input asp-for="creatorLogin" type="text" value="@Model.creatorLogin" class="form-control">

<input asp-for="pollNumber" type="number" value="@Model.pollNumber" class="form-control">

</div>

<div class="form-group">

<label for="countUser">Дата проведения опроса:</label>

<input asp-for="date" type="text" id="datepicker" class="form-control" readonly>

</div>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-primary" data-dismiss="modal">Отмена</button>

<button type="submit" class="btn btn-primary" >Сформировать</button>

</div>

</form>

</div>

</div>

Приложение В

Результаты работы приложения 

Рисунок В.1 – Сгенерированный pdf файл с аккаунтами опрашиваемых

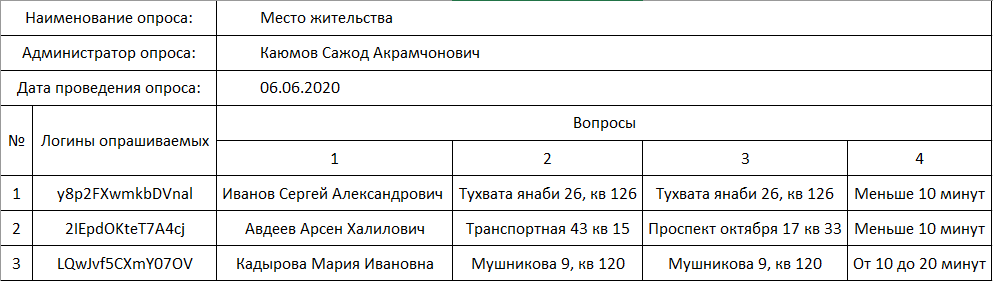


Рисунок В.2 – Сформированный отчёт по опросу

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 ГОСТ 19.201-78 ЕСКД. Техническое задание. Требования к содержанию и оформлению. [Текст]. – Введ. 1980–01–01 -М.: Изд-во стандартов, 1979.- 4 с.

2 ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения. [Текст]. – Введ. 1992–01–01 -М.: Изд-во стандартов, 1992.- 24 с.

3 ГОСТ 2.105-95 ЕСКД. Общие требования к текстовым документам. [Текст]. – Введ. 1996–01–07 -М.: Изд-во стандартов, 1996.- 36 с.

4 ГОСТ 7.82-2001 ЕСКД. Система стандартов по информации, библиотечному и издательскому делу [Текст]. – Введ. 2001–05–22 -М.: Изд-во стандартов, 2001.- 26 с.

5 Бен, А. C# 7.0. Справочник. Полное описание языка [Текст]: / А. Бен, А. Джозеф - М.: Вильямс, 2018. – 1024 с.

6 Бланшет Ж. HTML5: Верстка c использованием CSS. [Текст] / Ж. Бланшет, М. Саммерфилд.- 2-е изд., доп.— М.: «КУДИЦ-ПРЕСС», 2015. —736 с.

7 Бьюли, А. Изучаем SQL [Текст]: учеб. / А. Бьюли - 3-е изд. - СПб.: Символ-плюс, 2016. - 312 с.

8 Гуриков, С. Введение в программирование на языке Visual C#. Учебное пособие. [Текст]: учеб. / С. Гуриков - М.: Форум, 2018. - 448 с.

9 Дино, Э. Разработка современных веб-приложений. Анализ предметных областей и технологий [Текст]: / Э. Дино - М.: Вильямс, 2017. – 464 с.

10 Живицкая, Е. Н. Системный анализ и проектирование [Текст]: учеб. / Е. Н. Живицкая - Мн.: БГУИР, 2015. 241 с.

11 Лоусон Б., Шарп Р. Изучаем HTML5. Библиотека специалиста. [Текст]/ Б. Лоусон, Р. Шарп - 2-е издание. – СПб.: Питер, 2012 – 286 с.

12 Марк, О. Введение в Фреймворк Bootstrap 4.4 [Электронный ресурс]. -Режим доступа: URL: https://metanit.com/sharp/aspnet5/, свободный, (28.03.2020).

13 Мартин, Р. С. Принцип, паттерны и методика гибкой разработки на языке C# [Текст]: / Р. С. Мартин, М. Мартин. - М.: Символ-Плюс, 2017. – 768 с.

14 Мартин, Р. Чистый код: создание, анализ и рефакторинг. Библиотека программиста [Текст]: учеб. / Р. Мартин - СПб.: Питер, 2015. - 464 с.

15 Эванс, Э. Предметно-ориентированное программирование. Структуризация сложных программных систем [Текст]: / Э. Эванс - М.: Вильямс, 2018. – 448 с.

16 Эндрю, Т. Язык программирования C# 7 и платформы .NET и .NET Соre [Текст]: / Т. Эндрю, Д. Филипп - М.: Вильямс, 2018. – 1328 с.

17 Георгий, Д. Изучение ASP.NET Core MVC [Электронный ресурс]. -Режим доступа: URL: https://www.youtube.com/watch?v=U4nNZS1bnj8&list=PL0lO\_mIqDDFWltIe7D6aUS5f4k1y2-rgn, свободный, (25.03.2020).

18 Рик, А. Разработка приложений ASP.NET Core [Электронный ресурс]. -Режим доступа: URL: https://docs.microsoft.com/ru-ru/aspnet/core/?view=aspnetcore-3.1., свободный, (01.04.2020).

19 Метанит Руководство по ASP.NET Core 3 [Электронный ресурс]. -Режим доступа: URL: https://metanit.com/sharp/aspnet5/, свободный, (02.04.2020).

20 Хабр [Электронный ресурс] / Режим доступа https://habr.com/, свободный, (01.04.2020).