

					<i>КП.ПО-11.1-40-01-01</i>	<i>Лист</i>
						1
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

					<i>КП.ПО-11.1-40-01-01</i>	<i>Лист</i>
						2
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>		

## СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ .....</b>	<b>4</b>
<b>1.АНАЛИЗ ЗАДАЧИ .....</b>	<b>5</b>
1.1 РАЗРАБОТКА МОДУЛЬНОЙ СТРУКТУРЫ .....	5
ПРОГРАММЫ.....	6
1.2 ВЫБОР СПОСОБОВ ОПИСАНИЯ И ОБЪЕДИНЕНИЯ ВХОДНЫХ ДАННЫХ .....	6
1.3 РАЗРАБОТКА ПЕРЕЧНЯ ПОЛЬЗОВАТЕЛЬСКИХ ФУНКЦИЙ ПРОГРАММЫ.....	6
<b>2. РАЗРАБОТКА АЛГОРИТМ ВЫБОР СПОСОБОВ ОПИСАНИЯ И ОБЪЕДИНЕНИЯ ВХОДНЫХ ДАННЫХ .....</b>	<b>9</b>
<b>3. РАЗРАБОТКА И ОПИСАНИЕ РАБОТЫ ПРОГРАММЫ.....</b>	<b>12</b>
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>21</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....</b>	<b>22</b>

## ВВЕДЕНИЕ

В современном информационном обществе веб-приложения и базы данных играют важную роль в различных сферах деятельности, предоставляя пользователю удобный доступ к информации и обеспечивая эффективное хранение и управление данными. В данной курсовой работе я сосредоточился на разработке веб-приложения с использованием базы данных, с целью создания функционального и надежного инструмента для работы с данными.

Основной целью данной работы было создание веб-приложения, способного эффективно обрабатывать и хранить данные, а также предоставлять пользователю удобный интерфейс для взаимодействия с этими данными. Для достижения этой цели были использованы современные технологии разработки веб-приложений и баз данных.

В ходе работы были определены основные требования к функциональности приложения и структуре базы данных. Был выбран подходящий стек технологий, включающий язык программирования, фреймворк для веб-разработки и систему управления базами данных.

В результате разработки было создано функциональное веб-приложение, способное взаимодействовать с базой данных, выполнять операции добавления, изменения и удаления данных, а также предоставлять пользователю удобный интерфейс для просмотра и поиска информации. Кроме того, были реализованы механизмы обработки ошибок и обеспечения безопасности данных.

Данная курсовая работа представляет собой описание процесса разработки веб-приложения с использованием базы данных, включая анализ требований, проектирование структуры базы данных, выбор технологий и реализацию функциональности. Результаты работы могут быть полезными для разработчиков, интересующихся созданием веб-приложений с использованием баз данных.

## 1.АНАЛИЗ ЗАДАЧИ

Разработка программы планирования бюджета проекта.

В компании по разработке программного обеспечения для планирования работ по проекту необходимо обработать информацию следующего вида: наименование проекта, вид работ (работа над требованиями, разработка архитектуры, реализация, тестирование), Ф.И.О. сотрудника, предполагаемое количество часов, стоимость одного часа.

Индивидуальное задание: для каждого проекта вывести его итоговую стоимость, перечень видов работ с указанием общего количества задействованных специалистов и стоимости данного этапа.

Для разработки программы будут использоваться следующая технология: объектно-ориентированное программирование (ООП).

Средствами разработки программы будут: язык программирования C#, язык программирования JavaScript, HTML – язык гипертекстовой разметки, CSS – каскадные таблицы стилей.

Среда разработки – Visual Studio

Фреймворк для работы с веб – ASP.NET Core

СУБД – Microsoft SQL Server

### 1.1 РАЗРАБОТКА МОДУЛЬНОЙ СТРУКТУРЫ ПРОГРАММЫ

Первым этапом работы программы является авторизация. Для этого необходимо создать таблицу, содержащую сведения об учётных записях пользователей следующего вида: login, password, role. В качестве password хранится хеш-значение, а не исходное. Role равняется 1 для администратора и 0 для обычного пользователя. Изначально необходимо вручную записать в таблицу учётную запись администратора.

Вторым этапом работы программы является работа с таблицей данных, которая становится возможной после прохождения авторизации.

Модуль администратора включает следующий подмодули со следующими функциональными возможностями:

1. Управление учётными записями пользователей: просмотреть все учётные записи; добавить учётную запись; отредактировать учётную запись; удалить учётную запись.

					КП.ПО-11.1-40-01-01	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

2. Работа с базой данных: создать таблицу; открыть таблицу; удалить таблицу.

3. Работа с данными: просмотреть все данные; добавить запись; удалить запись; редактировать запись; выполнить поиск данных; выполнить сортировку по различным полям; для каждого проекта вывести его итоговую стоимость, перечень видов работ с указанием общего количества задействованных специалистов и стоимости данного этапа.

Модуль пользователя включает подмодуль работы с данными со следующими функциональными возможностями: просмотреть все данные; выполнить поиск данных; выполнить сортировку по различным полям; для каждого проекта вывести его итоговую стоимость, перечень видов работ с указанием общего количества задействованных специалистов и стоимости данного этапа.

## 1.2 ВЫБОР СПОСОБОВ ОПИСАНИЯ И ОБЪЕДИНЕНИЯ ВХОДНЫХ ДАННЫХ

class **DbContext**: определяет контекст данных, используемый для взаимодействия с базой данных.

class **ApplicationDbContext**: класс, производный от DbContext, определяет контекст данных посредством которого происходит взаимодействие с текущей базой данных.

DbSet<Project> **Projects** – представляет набор проектов, которые хранятся в базе данных.

DbSet<User> **Users** – представляет набор учётных записей, которые хранятся в базе данных.

class **DbContextOptionsBuilder**: устанавливает параметры подключения.

## 1.3 РАЗРАБОТКА ПЕРЕЧНЯ ПОЛЬЗОВАТЕЛЬСКИХ ФУНКЦИЙ ПРОГРАММЫ

interface **IBaseRepository<T>**: определяет базовые методы работы с данными, которые хранятся в базе данных.

Task **Create**(T entity) – записывает данные в базу данных.

Task<T> **Get**(int id) – получаем данные по идентификатору из базы данных.

IEnumerable<T> **Select**() – получаем все данные из базы данных.

					КП.ПО-11.1-40-01-01	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

Task **Delete**(T entity) – удаляет определённые данные из базы данных.

Task<T> **Update**(T entity) – обновляет данные в базе данных.

interface **IProjectRepository**: интерфейс, производный от интерфейса **IBaseRepository<Project>**, определяет методы для работы с данными, связанные с проектами, которые хранятся в базе данных.

Task<Project> **GetByName**(string name) – получаем данные по имени проекта из базы данных.

interface **IUserRepository**: интерфейс, производный от интерфейса **IBaseRepository<User>**, определяет методы для работы с данными, связанные с учётными записями, которые хранятся в базе данных.

Task<User> **GetByLogin**(string login) – получаем данные по логину из базы данных.

class **ProjectRepository**: класс, реализующий интерфейс **IProjectRepository**, содержит методы для работы с данными, связанные с проектами, которые хранятся в базе данных.

class **UserRepository**: класс, реализующий интерфейс **IUserRepository**, содержит методы для работы с данными, связанные с учётными записями, которые хранятся в базе данных.

class **BaseResponse<T>**: класс, реализующий интерфейс **IBaseResponse<T>**, представляющий собой "ответ", полученный из базы данных, благодаря которому можем осуществить взаимодействие между базой данных, методами работы с базой данных, полученными данными и самим приложением

interface **IProjectService**: определяет методы для сервиса для работы с проектами.

**IBaseResponse<IEnumerable<Project>> GetProjects()** – получаем все проекты из базы данных.

**Task<IBaseResponse<ProjectViewModel>> GetProject(int id)** – получаем проект по идентификатору из базы данных.

**Task<IBaseResponse<ProjectViewModel>> GetProjectByName(string name)** – получаем проект по имени из базы данных.

**Task<IBaseResponse<Project>> CreateProject(ProjectViewModel projectViewModel)** – создаём проект и записываем в базу данных.

**Task<IBaseResponse<bool>> DeleteProject(int id)** – удаляем проект по идентификатору из базы данных.

**Task<IBaseResponse<Project>> Edit(int id, ProjectViewModel projectViewModel)** – редактируем проект по идентификатору и сохраняем в базе данных.

interface **IUserService**: определяет методы для сервиса для работы с учётными записями.

					<i>КП.ПО-11.1-40-01-01</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		7

**IBaseResponse<IEnumerable<User>> GetUsers()** – получаем все учётные записи из базы данных.

**Task<IBaseResponse<UserViewModel>> GetUserByLogin(string login)** – получаем учётную запись по логину из базы данных.

**Task<IBaseResponse<User>> CreateUser(UserViewModel userViewModel)** – создаем учётную запись и записываем в базу данных.

**Task<IBaseResponse<bool>> DeleteUser(int id)** – удаляем учётную запись по идентификатору из базы данных.

**Task<IBaseResponse<User>> Edit(int id, UserViewModel userViewModel)** – редактируем учётную запись по идентификатору и сохраняем в базе данных.

**interface IAccountService:** определяет методы для сервиса для регистрации.

**Task<IBaseResponse<ClaimsIdentity>> Register(RegisterViewModel registerViewModel)** – регистрирует пользователя (добавляет учётную запись).

**Task<IBaseResponse<ClaimsIdentity>> Login(LoginViewModel loginViewModel)** – вход пользователя.

**class ProjectService:** класс, реализующий интерфейс **IProjectService**, своего рода слой между базой данных и контроллерами для повышения безопасности.

**class UserService:** класс, реализующий интерфейс **IUserService**, своего рода слой между базой данных и контроллерами для повышения безопасности.

**class AccountService:** класс, реализующий интерфейс **IAccountService**, своего рода слой между базой данных и контроллерами для повышения безопасности.

**class ProjectController:** класс представляющий из себя контроллер для проектов, методы которого являются действия, в зависимости от которых будет возвращено определённое представление.

**class UserController:** класс представляющий из себя контроллер для учётных записей, методы которого являются действия, в зависимости от которых будет возвращено определённое представление.

**class AccountController:** класс представляющий из себя контроллер для пользователей, методы которого являются действия, в зависимости от которых будет возвращено определённое представление.



## 2. РАЗРАБОТКА АЛГОРИТМ ВЫБОР СПОСОБОВ ОПИСАНИЯ И ОБЪЕДИНЕНИЯ ВХОДНЫХ ДАННЫХ

Получение проекта по имени(Task<IBaseResponse<ProjectViewModel>> GetProjectByName(string name)):

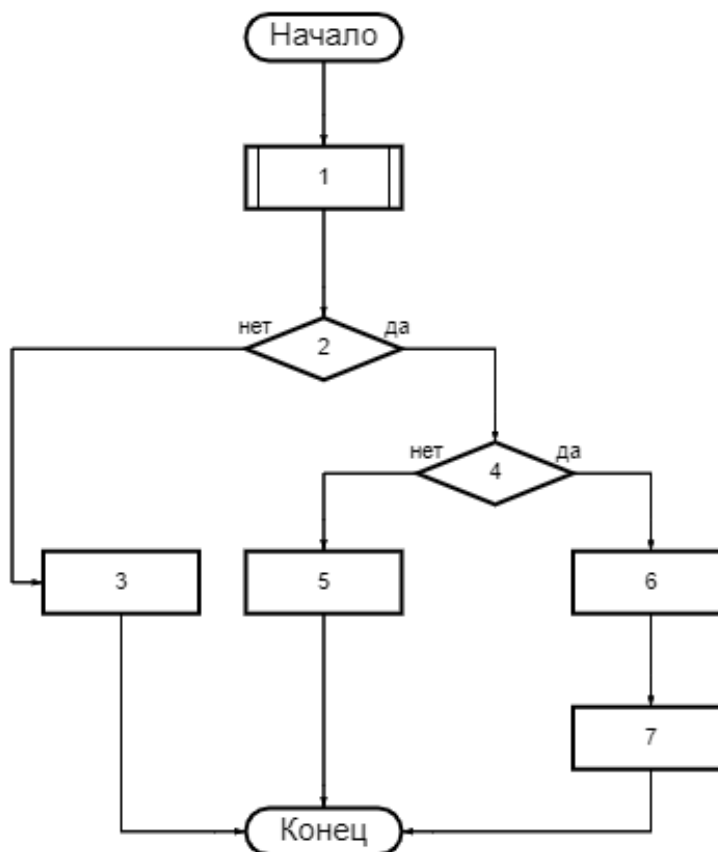
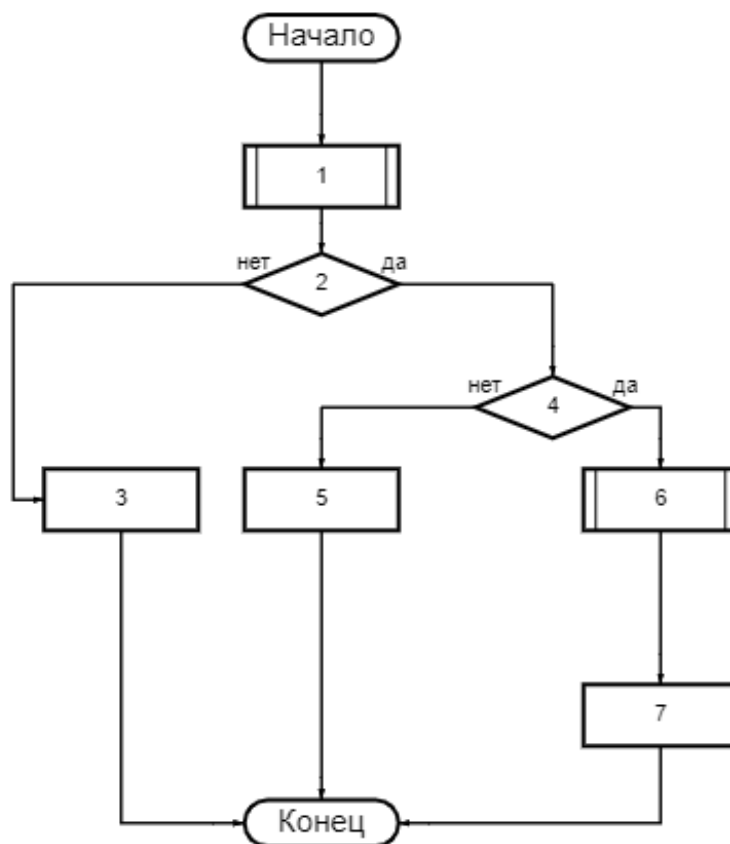


Рисунок 1 – Получение проекта по имени

1. Task<Project> **GetByName**(string name).
2. Не возникло ли ошибок при обращении к базе данных?
3. Вернуть ответ с сообщением о возникшей ошибке.
4. Существует ли проект?
5. Вернуть ответ с сообщением об отсутствии проекта.
6. Присвоение значений проекту.
7. Вернуть ответ с сообщением об успешном получении проекта.

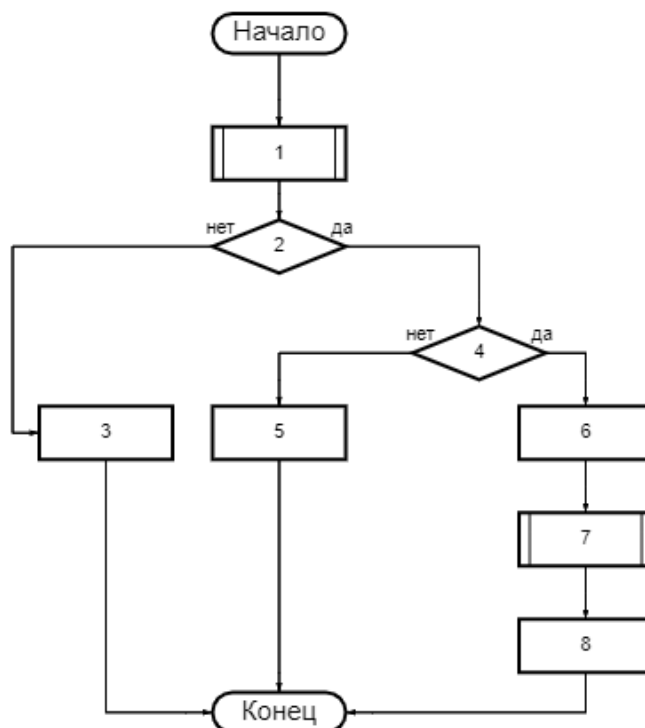
**Получение всех проектов(Task<IBaseResponse<bool>>  
DeleteProject(int id)):**



*Рисунок 2 – Удаление проекта по идентификатору*

1. Task<Project> **Get**(int id).
2. Не возникло ли ошибок при обращении к базе данных?
3. Вернуть ответ с сообщением о возникшей ошибке.
4. Существует ли проект?
5. Вернуть ответ с сообщением об отсутствии проекта.
6. Task **Delete**(Project entity).
7. Вернуть ответ с сообщением об успешном удалении проекта.

**Получение всех проектов**(Task<IBaseResponse<User>>  
**Edit**(int id, UserViewModel userViewModel):



*Рисунок 3 – Редактирование проекта*

1. Task<Project> **Get**(int id).
2. Не возникло ли ошибок при обращении к базе данных?
3. Вернуть ответ с сообщением о возникшей ошибке.
4. Существует ли проект?
5. Вернуть ответ с сообщением об отсутствии проекта.
6. Присвоение значений проекту.
7. Task<Project> **Update**(Project entity).
8. Вернуть ответ с сообщением об успешном редактировании проекта.

### 3. РАЗРАБОТКА И ОПИСАНИЕ РАБОТЫ ПРОГРАММЫ

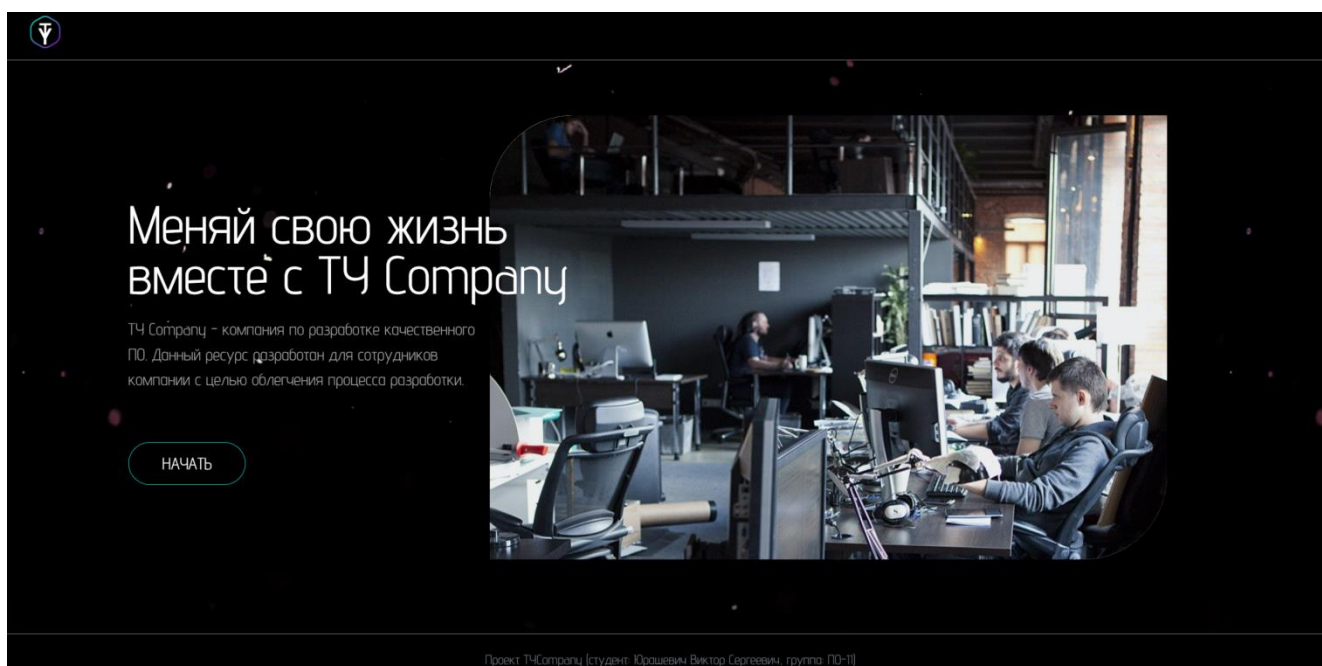


Рисунок 4 – Вкладка “Домашняя страница”

При запуске программы пользователя встречает вкладка “Домашняя страница” (рис. 4). В этой вкладке есть кнопка “НАЧАТЬ”, при нажатии на которую нас перенесёт на вкладку “Аутентификация” (рис. 5).

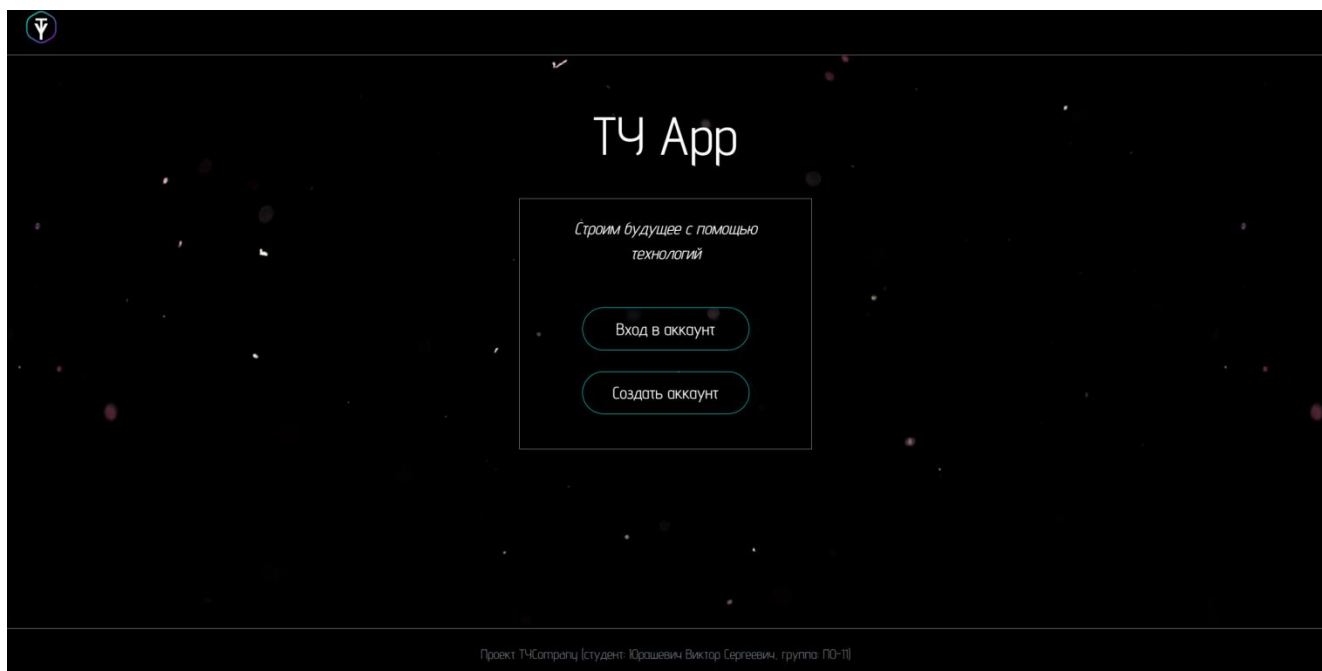
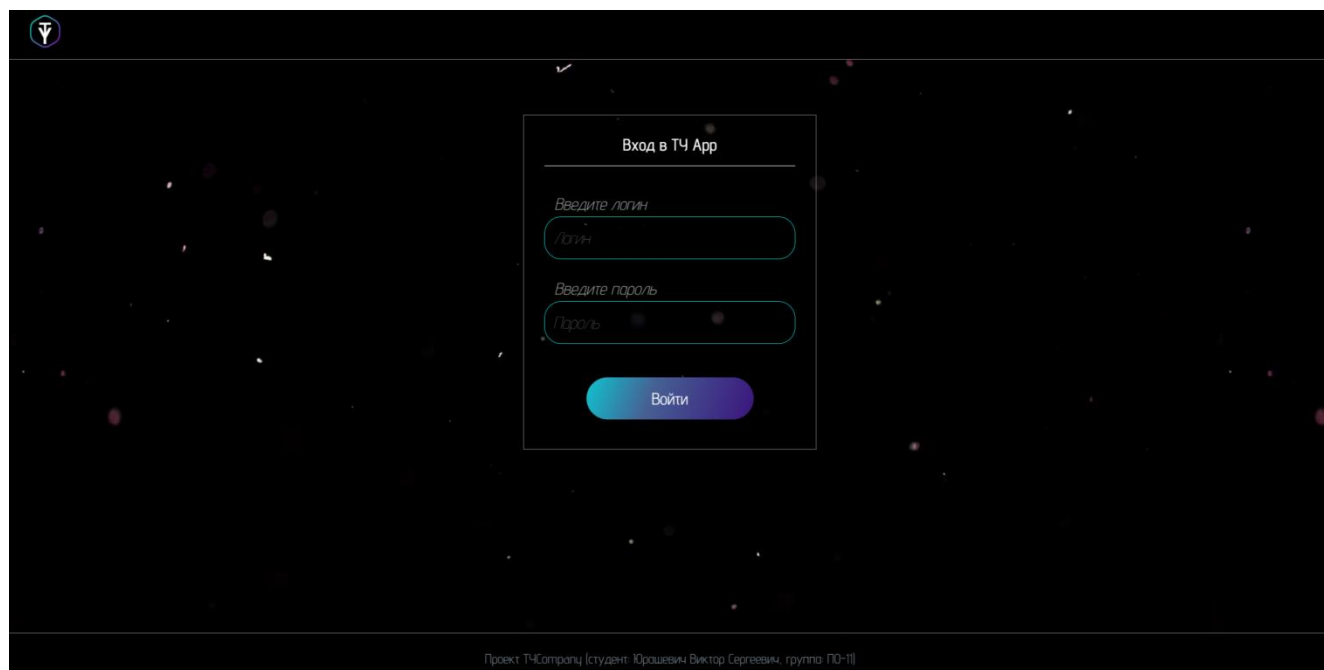


Рисунок 5 – Вкладка “Аутентификация”

					КП.ПО-11.1-40-01-01	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

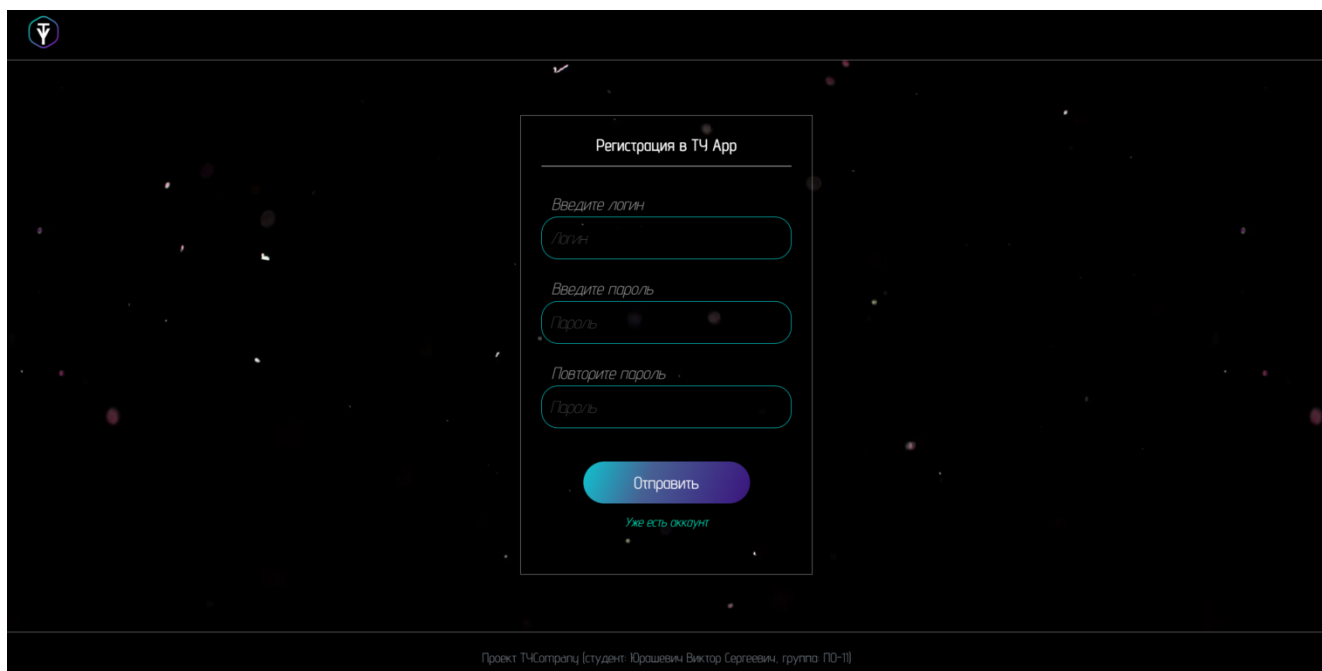
Мне показалось реализация аутентификации является более подходящим решением для моего задания нежели реализация авторизации с дальнейшим добавлением пользователей админом, поэтому, чтобы не возникало проблем с вводом логина, я добавил шаблон, ограничивающий ввод, благодаря которому пользователи могут вводить только логины типа @tycomp.by. Наличие такого шаблона ограничивает ввод логина пользователям, то есть зарегистрироваться могут только пользователи у которых есть почта @tycomp.by, то есть сотрудники компании.



*Рисунок 6 – Вкладка “Авторизация”*

При нажатии кнопки “Вход в аккаунт” на вкладке “Аутентификация” нас переносит на вкладку “Авторизация” (рис. 6). Во вкладке “Авторизация” вводится логин и пароль, если в таблице есть запись с соответствующим логином и паролем, то определяем тип пользователя по третьему полю (Role). Если в таблице нет записи с соответствующим логином, то выводится, что пользователь не найден. Если пользователь существует, но мы ввели неверный пароль, то выводится, что пароль неверный.

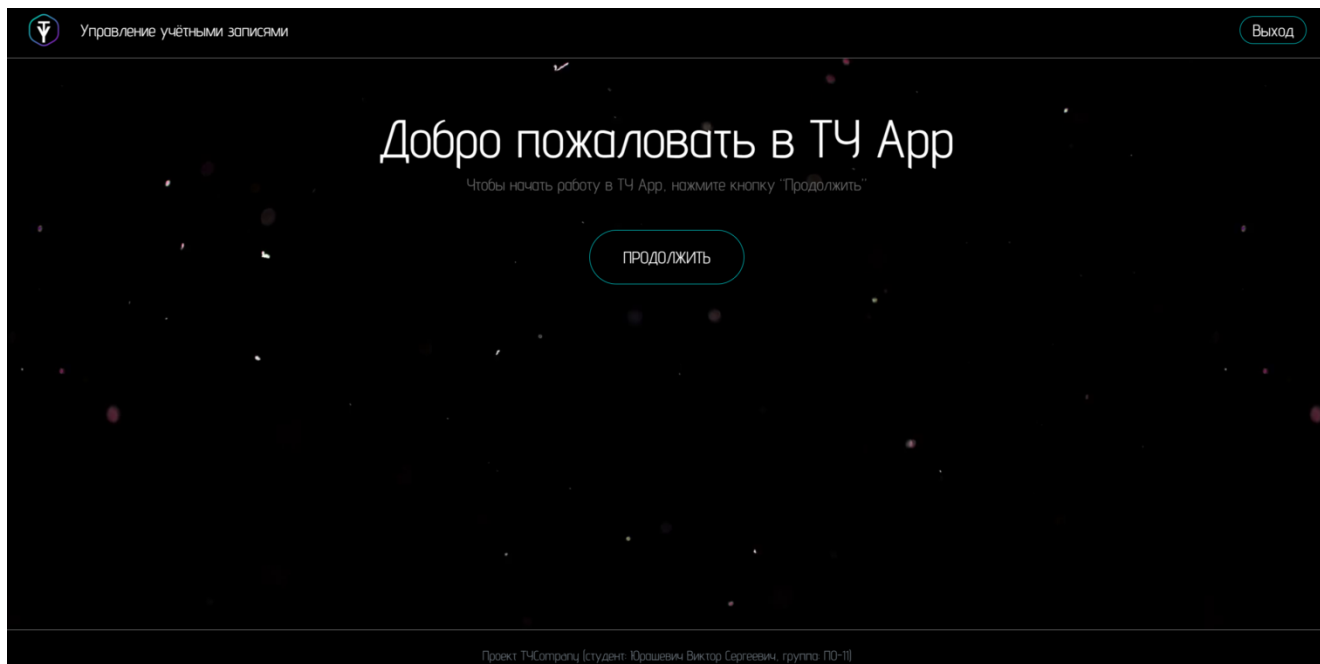
					КП.ПО-11.1-40-01-01	Лист
Изм.	Лист	№ докум.	Подпись	Дата		13



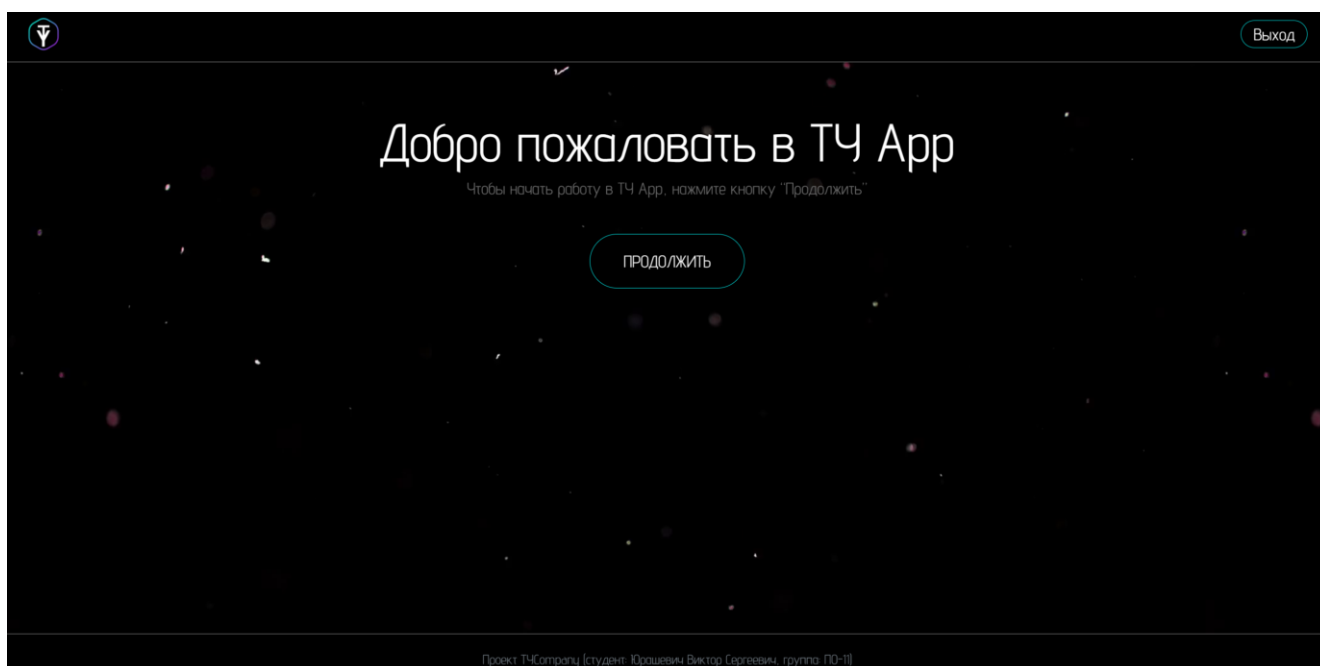
*Рисунок 7 – Вкладка “Регистрация”*

При нажатии кнопки “Создать аккаунт” на вкладке “Аутентификация” нас переносит на вкладку “Регистрация” (рис. 7). Во вкладке “Регистрация” вводится логин, пароль и затем повторяем пароль. Если введенные данные соответствуют установленным стандартам, то пользователь создается, и по умолчанию третье поле (Role) установит значение “Пользователь”. Если логин и пароль не соответствует установленным стандартам, то учетная запись не создается, а именно: логин должен иметь тип @tycomp.by и количество символов должно быть меньше 20, пароль должен быть больше 6 символов, но меньше 16 и введенные пароли должны совпадать.

					КП.ПО-11.1-40-01-01	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14



*Рисунок 8(а) – Вкладка “Добро пожаловать”(администратор)*



*Рисунок 8(а) – Вкладка “Добро пожаловать”(пользователь)*

Если пользователь – администратор, то ему открывается вкладка администратора (рис. 8(а)). Кнопка «Управление учетными записями» откроет вкладку “Таблица учётных записей” (рис. 9), в которой доступна возможность добавления, удаления и редактирования таблицы пользователей. Кнопка «ПРОДОЛЖИТЬ» откроет вкладку “Работа с проектами” (рис. 12), в которой отображается описание работы с проектами и кнопка, при нажатии которой мы приступаем непосредственно к работе с проектами. Помимо этого есть кнопка “Выход”, которая позволяет выйти из учётной записи и вернуться на вкладку

					<i>КП.ПО-11.1-40-01-01</i>	Лист
						15
Изм.	Лист	№ докум.	Подпись	Дата		

“Домашняя страница” (рис. 4).

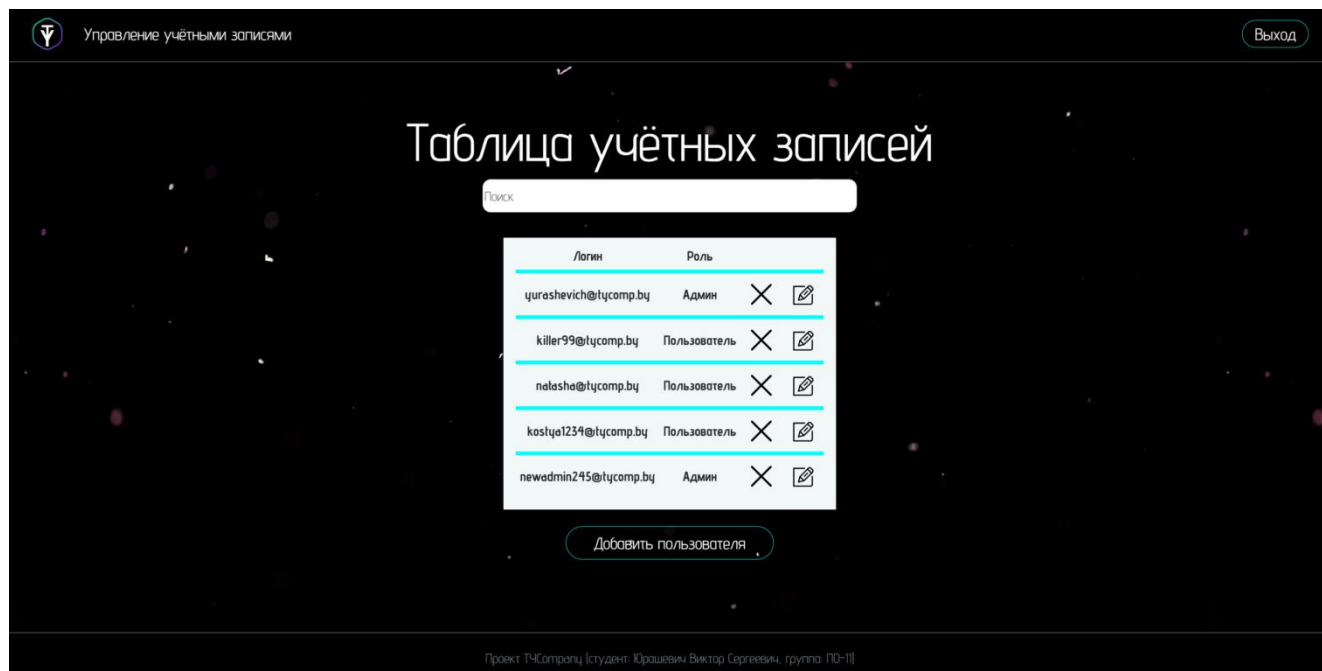


Рисунок 9 – Вкладка “Таблица учётных записей”

В базу данных изначально записывается учётная запись админа, поэтому на рис. 9 мы видим учётную запись с логином yurashevich@tycomp.by с ролью админа. В двух последних колонках мы видим два значка “Крест” и “Блокнот” – это операции удаления и редактирования соответственно. Также мы видим поисковую строку, её работа изображена на рис. 10, в таблице поиск осуществляется по имени. Помимо всего мы можем также отсортировать таблицу по полю, на заголовок которого мы кликнем, в порядке возрастания или убывания в зависимости от количества кликов по полю. Есть кнопка “Добавить пользователя”, при нажатии на которую нас переносит на вкладку “Добавить пользователя” (рис. 11).



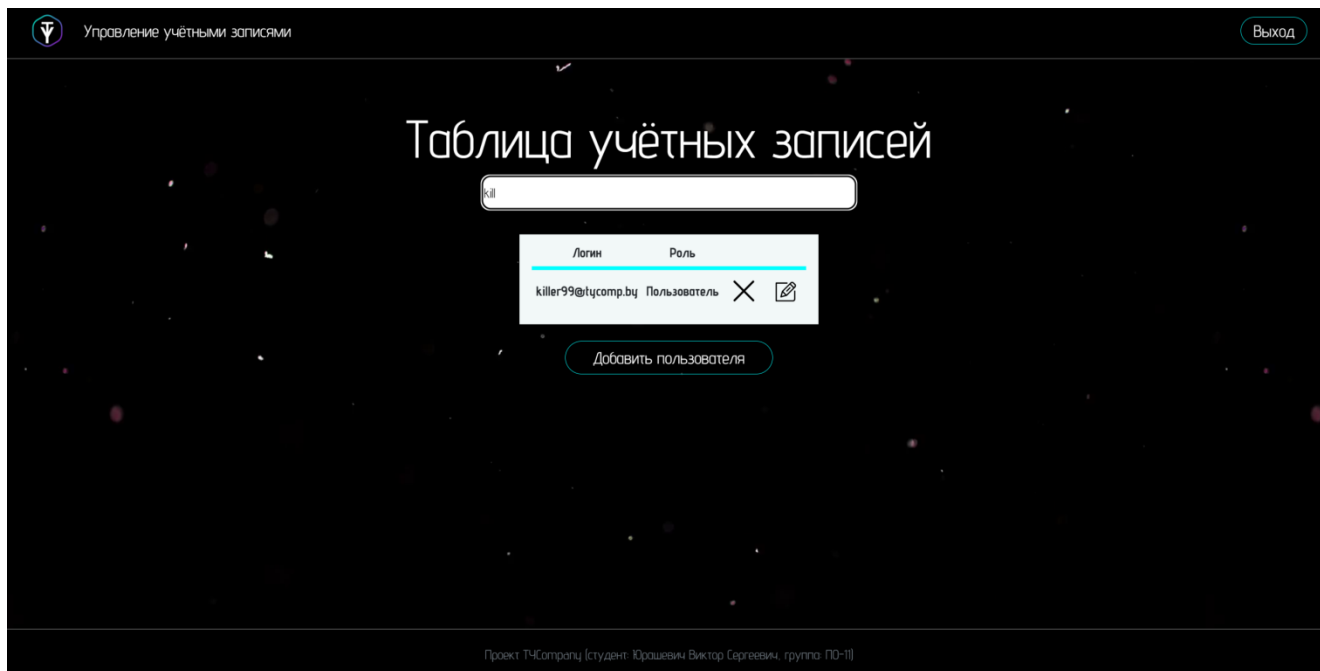


Рисунок 10 - Работа поисковой строки в таблице

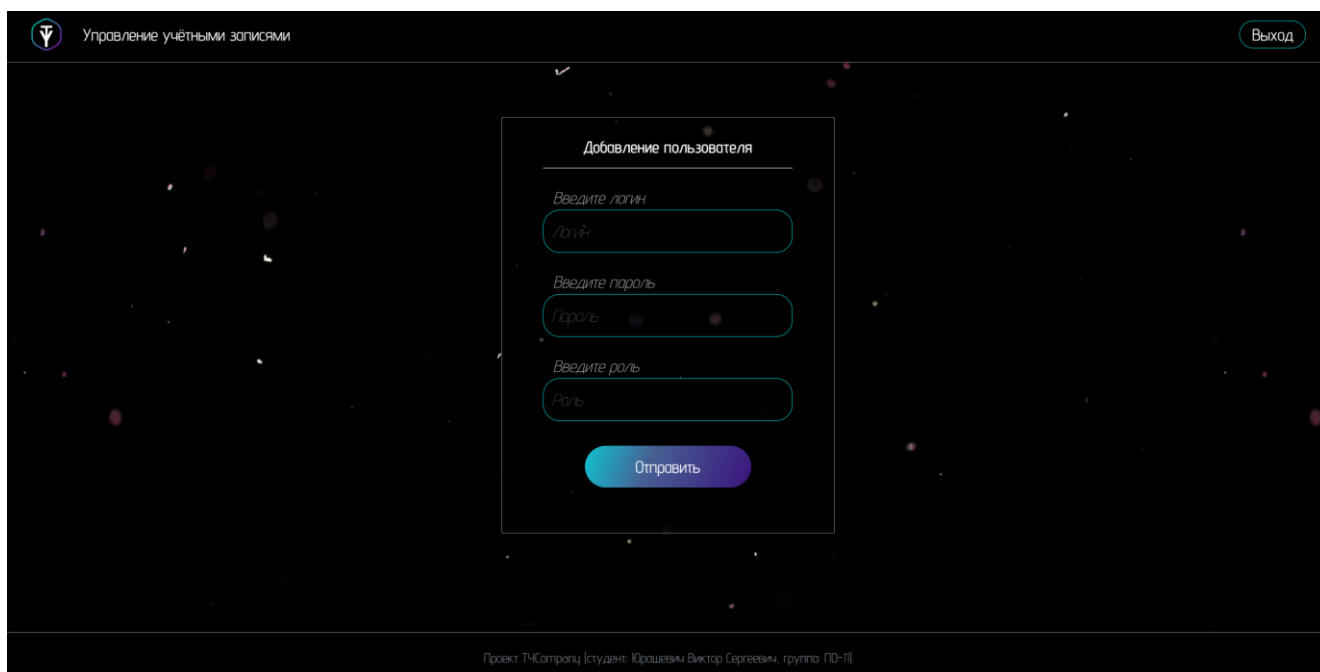


Рисунок 11 – Вкладка “Добавление пользователя”

При нажатии на кнопку “Добавить пользователя” нас переносит на вкладку “Добавление пользователя”, где мы вводим логин, пароль и роль. Если все введенные данные соответствуют установленным стандартам, указанные выше, то пользователь создается. Если нет, то выполняются действия, указанные выше.

					КП.ПО-11.1-40-01-01	Лист
Изм.	Лист	№ докум.	Подпись	Дата		17

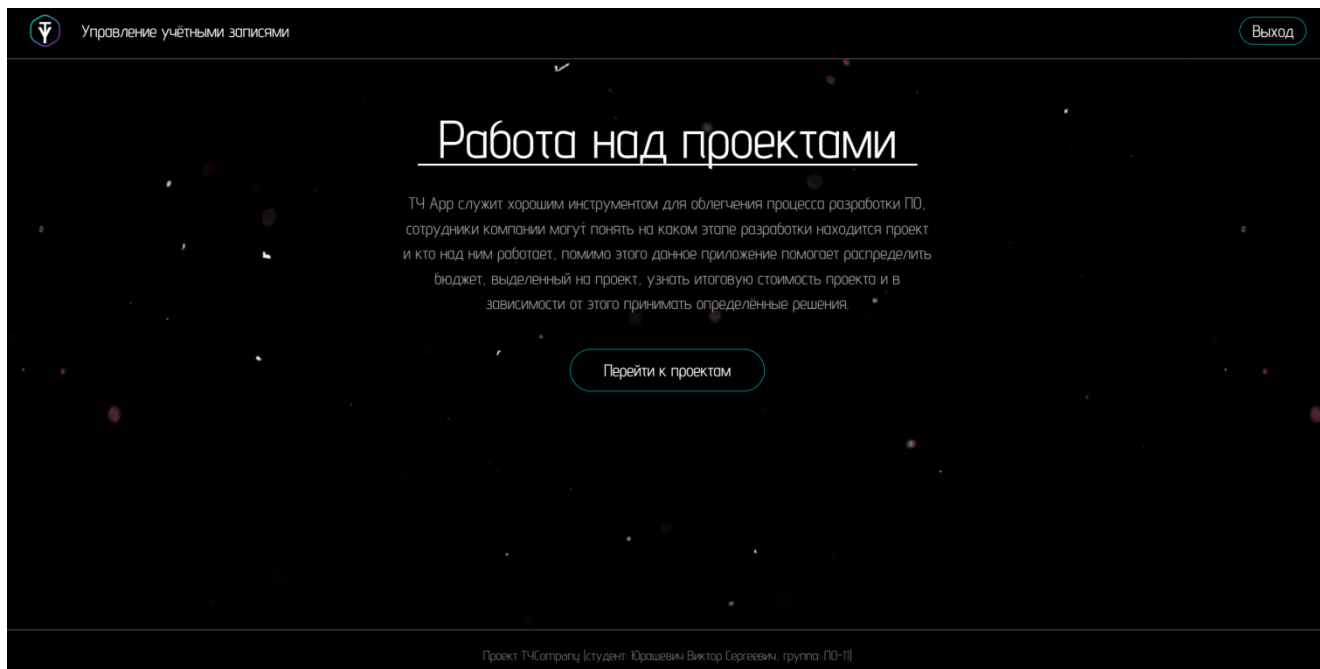


Рисунок 12 – Вкладка “Работа с проектами”

При нажатии на кнопку “ПРОДОЛЖИТЬ” переходим на вкладку “Работа с проектами” (рис. 12). На этой вкладке описывается работа с проектами, её возможности, что облегчает работу с проектами. Есть кнопка “Перейти к проектам”, при нажатии на которую мы переходим непосредственно к работе над проектами на вкладку “Таблица проектов” (рис. 13).

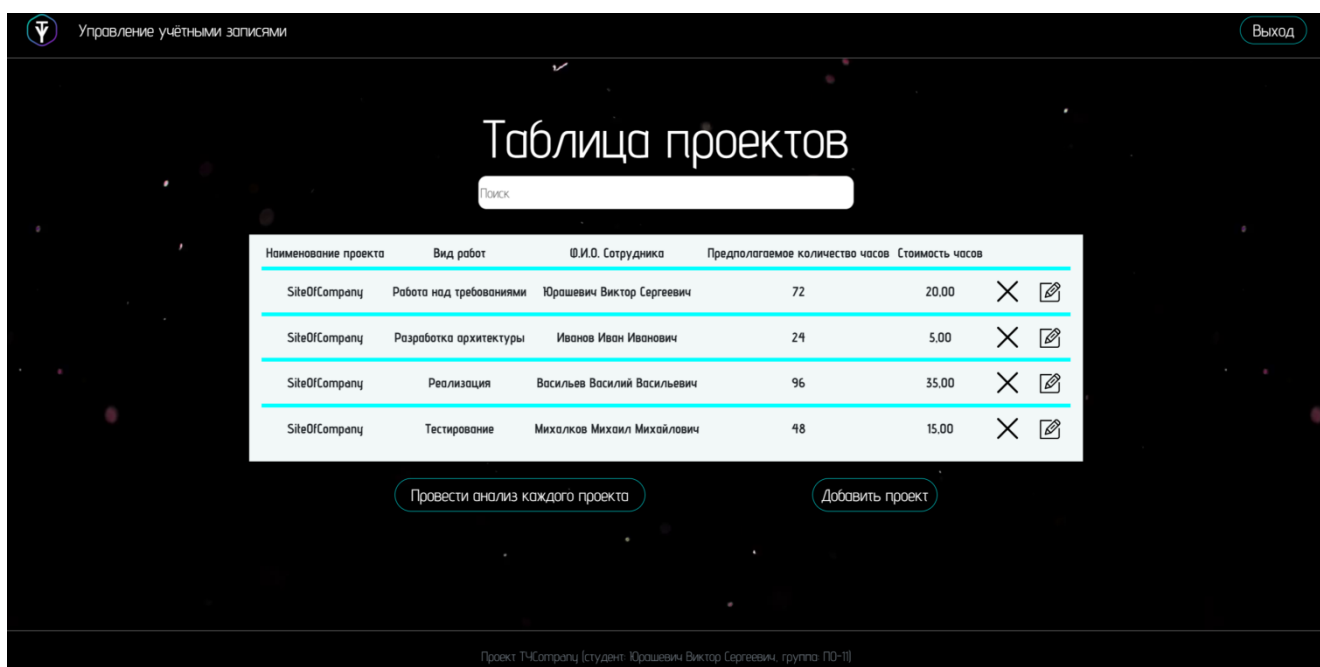
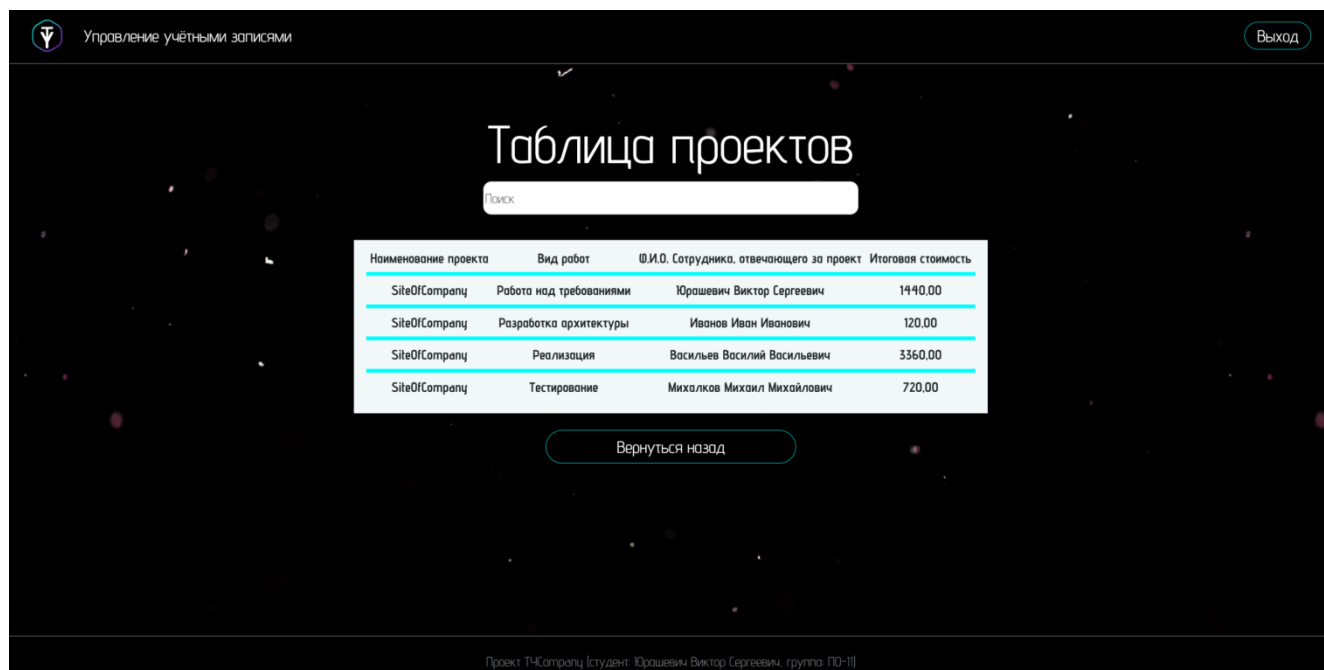


Рисунок 13 – Вкладка “Таблица проектов”

При нажатии на кнопку “Перейти к проектам” мы переходим на вкладку “Таблица проектов” (рис. 13). В двух последних колонках мы видим два значка


					КП.ПО-11.1-40-01-01	Лист
Изм.	Лист	№ докум.	Подпись	Дата		18

“Крест” и “Блокнот” – это операции удаления и редактирования соответственно. Также мы видим поисковую строку, её работа аналогична работе, изображённой на рис. 10, в таблице поиск осуществляется по имени. Помимо всего мы можем также отсортировать таблицу по полю, на заголовок которого мы кликнем, в порядке возрастания или убывания в зависимости от количества кликов по полю. Также есть две кнопки “Провести анализ каждого проекта” и “Добавить проект”, при нажатии одной из этих кнопок ведёт к вкладкам “Анализ проектов” (рис. 14) и “Добавление проекта” (рис. 15) соответственно.



*Рисунок 14 – Вкладка “Анализ проектов”*

При нажатии на кнопку “Провести анализ каждого проекта” мы переходим на вкладку “Анализ проектов” (рис. 14), где можем увидеть таблицу с проанализированным этапом каждого проекта. Кнопка “Вернуться назад” позволяет нам вернуться на вкладку “Таблица проектов”.


Управление учётными записями

Выход

Добавление проекта

Введите наименование проекта  
Наименование

Введите вид работ  
Вид работ

Введите сотрудника  
Сотрудник

Введите количество часов  
Количество часов

Введите стоимость часа  
Стоимость часа

Отправить

*Рисунок 15 – Вкладка “Добавление проекта”*

При нажатии на кнопку “Добавить проект” мы переходим на вкладку “Добавление проекта” (рис. 15). Если все введённые данные соответствуют установленным стандартам, то проект добавится.

					<i>КП.ПО-11.1-40-01-01</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		20

## ЗАКЛЮЧЕНИЕ

Разработка веб-приложения с базой данных является сложным, но весьма интересным процессом. Она требует глубокого понимания основных принципов веб-разработки и навыков работы с базами данных.

Использование ASP.NET Core для разработки веб-приложения позволяет создать мощное, масштабируемое и безопасное приложение. Фреймворк предлагает широкий выбор инструментов и возможностей для реализации различных функциональностей.

Использование базы данных позволяет хранить и управлять данными, необходимыми для работы веб-приложения. Взаимодействие с базой данных может быть реализовано с помощью ORM-фреймворков, таких как Entity Framework Core, что упрощает работу с данными и обеспечивает высокую степень абстракции от конкретной базы данных.

Разработка веб-приложения с базой данных на ASP.NET Core требует систематического подхода и хорошей организации работы. Важно планировать и структурировать процесс разработки, а также следовать передовым практикам и принципам разработки.

В целом, выполнение курсовой работы по разработке веб-приложения с базой данных позволяет углубить знания и навыки веб-разработки, а также получить практический опыт работы с ASP.NET Core и базами данных. Этот опыт может быть полезен для дальнейшего развития в сфере веб-разработки и создания более сложных и функциональных приложений.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. METANIT.COM. Сайт о программировании: [Электронный ресурс] URL: <https://metanit.com/>
2. Microsoft Learn. Техническая документация: [Электронный ресурс] URL: <https://learn.microsoft.com/ru-ru/docs/>
3. Герберт Шилдт, С# 4.0: полное руководство. : Пер. с англ. – М. : ООО “И.Д. Вильямс”, 2015. – 1056с.
4. WebReference. Главная: [Электронный ресурс] URL: <https://webref.ru/>
5. Bootstrap. CSS: [Электронный ресурс] URL: <https://getbootstrap.ru/docs/3.3.7/css/>
6. JQuery. Русская документация: [Электронный ресурс] URL: <https://jquery-docs.ru/>
7. Дэвид Флэнаган. JavaScript. Полное руководство, 7-е изд. : Пер. с англ. – СПб. : ООО “Диалектика”, 2021. – 720с.

					<i>КП.ПО-11.1-40-01-01</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		22