|  |  |
| --- | --- |
|  | Министерство образования и науки Российской Федерации  Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б.Н.Ельцина» (УрФУ)  ИРИТ-РТФ  Базовая кафедра «Аналитика больших данных и методы видео анализа» |

Оценка по проекту \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель проекта \_\_Медведев M.A.\_\_\_

Члены комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отчет по проекту на тему

**«Сайт школы информационной безопасности EduSec»**

Студент: Коротков Виталий Константинович \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент: Чуфаров Константин Николаевич \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ФИО) (Подпись)

Группа #: РИ-311055

Екатеринбург

2024

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc168262429)

[Анализ предметной области 6](#_Toc168262430)

[Проектирование приложения 8](#_Toc168262431)

[Клиентская часть (Frontend). 8](#_Toc168262432)

[Серверная часть (Backend). 9](#_Toc168262433)

[Выбор инструментальных средств разработки 11](#_Toc168262434)

[Разработка прототипов и дизайн-шаблонов 14](#_Toc168262435)

[Разработка клиентской части приложения 16](#_Toc168262436)

[index.html. 16](#_Toc168262437)

[1. Навигационное меню. 16](#_Toc168262438)

[2. Модальные окна. 18](#_Toc168262439)

[3. Блок оферты и формы записи. 29](#_Toc168262440)

[4. Блок преимуществ школы информационной безопасности. 30](#_Toc168262441)

[5. Слайдер с предоставляемыми услугами. 32](#_Toc168262442)

[6. Форма заказа услуги. 35](#_Toc168262443)

[7. Блок оферты. 37](#_Toc168262444)

[8. Форма получения консультации. 38](#_Toc168262445)

[9. Нижняя часть страницы - футер. 39](#_Toc168262446)

[10. Ссылки на собственные скрипты. 41](#_Toc168262447)

[advantages.html 41](#_Toc168262448)

[faq.html 43](#_Toc168262449)

[Разработка серверной части приложения 46](#_Toc168262450)

[ЗАКЛЮЧЕНИЕ 58](#_Toc168262451)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 60](#_Toc168262452)

# ВВЕДЕНИЕ

Актуальность темы. В современном мире информационная безопасность является одной из ключевых областей, влияющих на устойчивое развитие информационных систем и защиту данных. С увеличением числа кибератак и угроз кибербезопасности возрастает необходимость обучения и повышения осведомленности пользователей в данной сфере. Платформы, предоставляющие образовательные ресурсы и инструменты по информационной безопасности, становятся все более востребованными, что обуславливает актуальность создания и развития таких проектов.

Цель работы. Целью данного проекта является разработка веб-приложения «EduSec» – лендинга для школы информационной безопасности. Проект предназначен для пользователей, стремящихся повысить свою осведомленность и воспользоваться услугами в области информационной безопасности, а также для администраторов школы, управляющих содержанием платформы и предоставляющим свои услуги пользователям.

Для достижения поставленной цели необходимо было выполнить следующие задачи:

1. Разработать интерфейс платформы, главную страницу сайта обеспечивающую удобную навигацию и доступ к основным разделам сайта.
2. Реализовать основное наполнение главной страницы приложения.
3. Разработать функционал регистрации и авторизации пользователей с учетом требований безопасности.
4. Создать разделы, содержащие информацию о преимуществах платформы и ответы на часто задаваемые вопросы.
5. Организовать хранение данных в локальном хранилище (localStorage) для реализации псевдо-БД и обеспечения удобства управления данными.
6. Реализовать различные притягивающие внимание пользователя элементы, включая слайдер, анимации и тд.
7. Обеспечить защиту административного функционала от несанкционированного доступа при помощи авторизации.

Структура работы. Данная работа состоит из следующих разделов:

1. Введение – рассматриваются актуальность темы, цель работы и задачи, стоящие перед проектом.
2. Анализ предметной области – описание проекта, сферы деятельности в которых задействовано приложение.
3. Проектирование приложения – блок схема архитектуры проекта – посредством изображения и словесно.
4. Выбор инструментальных средств разработки – описание технологий, использованных в разработке приложения.
5. Разработка прототипов и дизайн-шаблонов – в данном разделе присутствуют прототипы, дизайн-шаблоны проекта.
6. Разработка клиентской части приложения – представлено поблочное описание разработки интерфейса, включая пояснения к его реализации.
7. Разработка серверной части приложения – в данном разделе представлены технологии, файлы скриптов, процессы взаимодействия с псевдо-серверной частью приложения, примеры кода.
8. Заключение.

# Анализ предметной области

Описание будущего проекта. Проект «EduSec» представляет собой веб-приложение, предназначенное для предоставления образовательных услуг в сфере информационной безопасности. В современных условиях стремительного развития информационных технологий и увеличения количества кибератак, образовательные платформы, такие как «EduSec», играют очень важную роль в повышении осведомленности и компетентности пользователей в области информационной безопасности.

Сферы деятельности, в которой будет использоваться данная система. Данная система предназначена для использования в следующих сферах деятельности:

* Образовательные учреждения – школы, колледжи, университеты и другие образовательные организации, которым необходимы программы по обучению информационной безопасности.
* Корпоративный сектор – компании, стремящиеся повысить осведомленность своих сотрудников о методах защиты информации и предотвращения кибератак.
* Частные пользователи – все, кто интересуется вопросами безопасности своих данных и хочет получить знания в области защиты информации.

Определение предметов, объектов, процессов и взаимодействий.

1. Предметы и объекты:
   * Пользователи – люди, использующие платформу для получения определенного вида услуг (обычные пользователи) и администраторы, управляющие содержанием платформы.
   * Интерфейс – основное наполнение страницы, содержащее оферту образовательных курсов, материалов и тренингов, предоставляемых платформой.
   * Данные – некоторые данные, содержащие информацию о предоставляемых услугах, пользователях сайта, преимуществах школы и тд.
2. Процессы:
   * Регистрация и авторизация пользователей – процесс создания учетных записей и входа в систему.
   * Просмотр и заказ услуг – пользователи могут просматривать доступные услуги и заказывать их через специальную форму.
   * Добавление и редактирование контента – администраторы могут добавлять новые услуги, преимущества и вопросы в FAQ, а также редактировать или удалять существующие элементы.
   * Создание запросов на обратную связь – возможность для пользователя оставить свои контактные данные через сайт для предоставления ему обратной связи.
3. Взаимодействия:
   * Пользователи с системой – пользователи взаимодействуют с интерфейсом платформы, регистрируются, авторизуются, просматривают контент и заказывают услуги.
   * Администраторы с системой – администраторы выполняют управление контентом платформы через специальные функции, доступные только им.
   * Система с локальным хранилищем (localStorage) – данные о пользователях, услугах, преимуществах и FAQ хранятся в локальном хранилище и управляются через интерфейс приложения.

# Проектирование приложения

Блок-схема архитектуры проекта – рисунок 1.

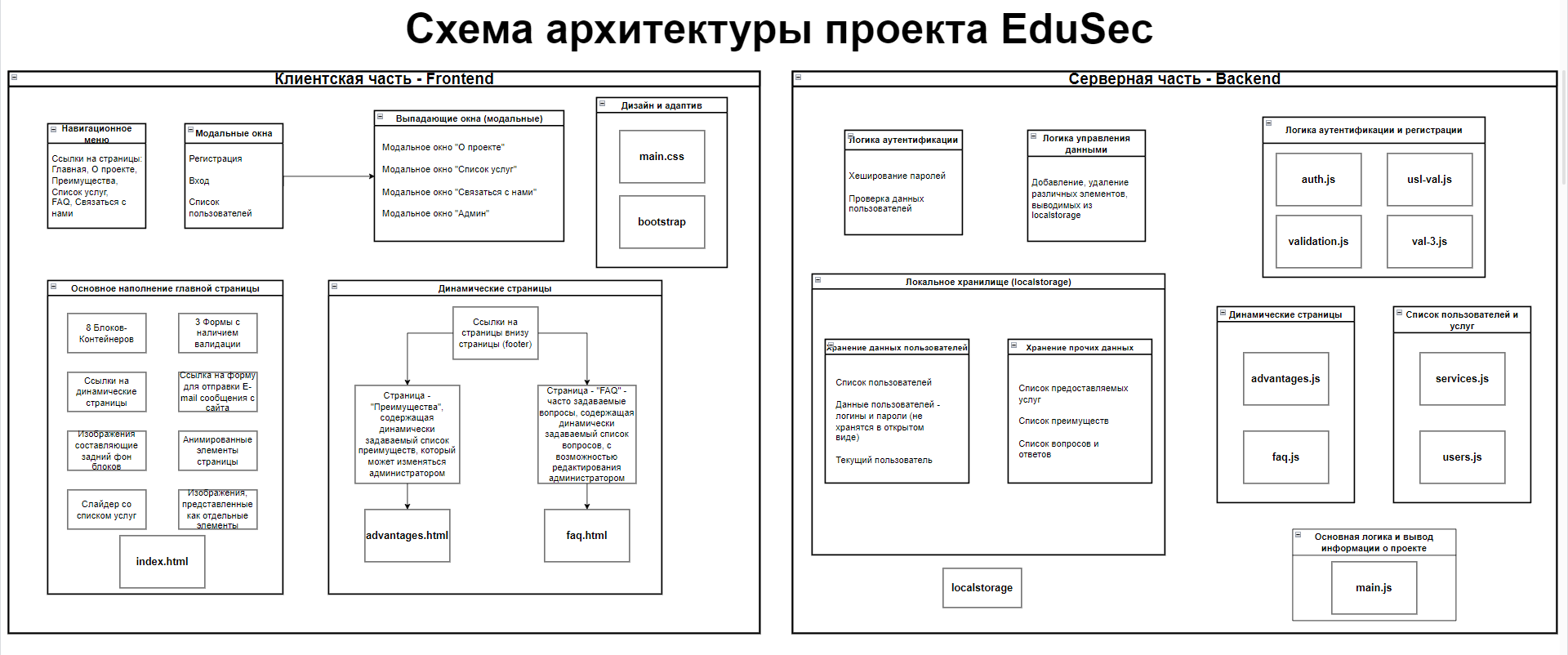


Рисунок 1 – Блок-схема архитектуры проекта

Словесное описание архитектуры.

## Клиентская часть (Frontend).

Клиентская часть отвечает за отображение интерфейса пользователя и обработку взаимодействий с ним. Основные компоненты включают навигационное меню для перемещения по различным разделам сайта, основное наполнение страницы, включая 8 блоков, 3 формы, ссылки на динамические страницы в футере, ссылка на форму отправки e-mail сообщений с сайта, изображения составляющие задний фон некоторых блоков, анимированные элементы страницы, включая преимущества проекта и слайдер предоставляемых услуг, также анимированные кнопки отправки форм, изображения представленные как отдельные элементы.

Дизайн сайта реализован также при помощи собственных наработок в main.css, импортируемые библиотек стилей, адаптив элементов сайта реализован при помощи bootstrap.

Навигационное меню предоставляет ссылки на ключевые страницы сайта, такие как главная страница, страница "О проекте", список преимуществ, список услуг, FAQ и контактная информация.

Модальные окна используются для регистрации новых пользователей, входа существующих пользователей и отображения списка пользователей (доступно только администратору).

Основные блоки сайта – представлены 8-ю контейнерами, содержащими основное наполнение сайта, включая формы, оферты, списки и тд.

Формы обратной связи, представленные в верхней и нижней части страницы, имеют как встроенную html валидацию, так и реализованную через javascript.

Слайдер услуг отображает доступные услуги с анимацией и позволяет пользователям ознакомиться с предлагаемыми возможностями.

Форма заказа услуги позволяет пользователям выбрать услугу из выпадающего списка, ввести личные данные и отправить заказ, валидация представлена как встроенными html методами, так и собственными скриптами javascript.

Динамические страницы отображают преимущества использования платформы и ответы на часто задаваемые вопросы (FAQ). Эти страницы могут быть обновлены администратором для добавления новых данных.

## Серверная часть (Backend).

Хотя в данном проекте серверная часть эмулируется с использованием логики на стороне клиента (JavaScript) и локального хранилища (LocalStorage), функциональность остается аналогичной классическому backend.

Логика аутентификации включает в себя хеширование паролей пользователей при регистрации и проверку введенных данных при входе. Это позволяет обеспечить безопасность данных пользователей.

Логика управления данными отвечает за добавление, удаление и редактирование услуг, преимуществ и FAQ, также удаление пользователей. Эти операции доступны только пользователю с правами администратора, то есть пользователю под аккаунтом admin.

Локальное хранилище (LocalStorage). Локальное хранилище используется для сохранения данных на стороне клиента. Включает данные о пользователях, текущем залогиненном пользователе, услугах, преимуществах и FAQ. LocalStorage позволяет хранить данные в виде ключ-значение, что упрощает доступ и управление ими через JavaScript.

* Хранение данных пользователей – включает список зарегистрированных пользователей и информацию о текущем залогиненном пользователе.
* Хранение данных услуг – содержит список доступных услуг, которые могут быть просмотрены пользователями, данный список может обновляться администратором.
* Хранение данных преимуществ и FAQ – включает списки преимуществ и часто задаваемых вопросов с ответами, которые могут быть обновлены администратором.

Также реализована логика отправки E-mail сообщений об обратной связи прямо с сайта посредством EmailJS.

# Выбор инструментальных средств разработки

Описание технологий, которые были использованы

В процессе разработки веб-приложения «EduSec» для школы информационной безопасности были использованы различные технологии и инструменты, обеспечивающие создание функционального, интерактивного и адаптивного пользовательского интерфейса. В данном разделе описаны основные технологии, которые использовались при разработке проекта.

**HTML (HyperText Markup Language).**

HTML является основным языком разметки, который используется для создания структуры веб-страниц. В проекте «EduSec» HTML используется для определения элементов пользовательского интерфейса, таких как заголовки, параграфы, формы, кнопки, списки, таблицы и другие элементы. С помощью HTML была создана структура всех страниц приложения, включая главную страницу, страницы «Преимущества», «FAQ».

**CSS (Cascading Style Sheets).**

CSS используется для стилизации и оформления HTML-элементов. В проекте «EduSec» CSS отвечает за внешний вид веб-страниц, включая цвета, шрифты, отступы, выравнивание, анимацию и адаптивность. С помощью CSS были созданы стили для навигационного меню, модальных окон, слайдера услуг, формы заказа и других компонентов интерфейса. Также CSS обеспечил адаптивность дизайна для различных устройств, включая настольные компьютеры, планшеты и смартфоны, реализована данная адаптивность была в дополнение к bootstrap посредством медиа-запросов.

**JavaScript.**

JavaScript является основным языком программирования, который используется для создания интерактивных и динамических веб-страниц. В проекте «EduSec» JavaScript отвечает за обработку событий, манипулирование элементами DOM, валидацию форм, а также взаимодействие с локальным хранилищем (LocalStorage). JavaScript также используется для реализации логики аутентификации пользователей, управления данными услуг, преимуществ и FAQ.

**LocalStorage.**

LocalStorage является встроенной возможностью веб-браузеров, позволяющей хранить данные в виде пар ключ-значение на стороне клиента. В проекте «EduSec» LocalStorage используется для хранения данных пользователей, текущего залогиненного пользователя, списка услуг, преимуществ и FAQ. Это позволяет сохранять состояние приложения между сеансами и обеспечивает возможность работы с данными без необходимости взаимодействия с сервером, в виду того, что сайт хостится на github pages и не поддерживает полноценный backend.

**Bootstrap.**

Bootstrap – это популярный фреймворк для создания адаптивных и мобильных веб-сайтов. В проекте «EduSec» Bootstrap используется для упрощения верстки и стилизации элементов интерфейса. С помощью компонентов Bootstrap были созданы адаптивные сетки, кнопки, формы, модальные окна и другие элементы интерфейса. Использование Bootstrap значительно ускорило процесс разработки и обеспечило единообразный внешний вид приложения на различных устройствах.

**CryptoJS.**

CryptoJS – это библиотека для JavaScript, предоставляющая различные алгоритмы шифрования и хеширования. В проекте «EduSec» CryptoJS используется для хеширования паролей пользователей перед их сохранением в LocalStorage. Это обеспечивает дополнительный уровень безопасности, защищая пароли от несанкционированного доступа.

**Draw.io (для создания диаграмм).**

Draw.io – это онлайн-инструмент для создания диаграмм и блок-схем. В проекте «EduSec» Draw.io использовался для создания схемы архитектуры приложения. С помощью этого инструмента была визуализирована структура проекта и взаимодействие между его компонентами.

**Git и GitHub, Github-Pages.**

Git – это система контроля версий, которая позволяет отслеживать изменения в коде и управлять различными версиями проекта. GitHub – это платформа для хостинга проектов, использующих Git. В проекте «EduSec» Git использовался для управления версионностью кода, а GitHub для хранения и совместной работы над проектом. Также сам проект был поставлен на хостинг через Github-Pages, для обеспечения его доступности в интернете.

**VSCode (Visual Studio Code).**

VSCode – это популярный текстовый редактор с поддержкой множества расширений и интеграций. В проекте «EduSec» VSCode использовался как основной инструмент для написания и редактирования кода. Благодаря поддержке различных расширений, таких как Prettier для форматирования кода и Live Server для локального хостинга, VSCode значительно упростил процесс разработки.

# Разработка прототипов и дизайн-шаблонов

Для разработки прототипа основной страницы сайта был использован электронный ресурс Craftum [1]. При помощи него был создан предварительный дизайн сайта с примерным расположением блоков, форм, первичный дизайн был переработан и адаптирован для проекта.

Предварительный дизайн проекта – рисунки 2, 3.

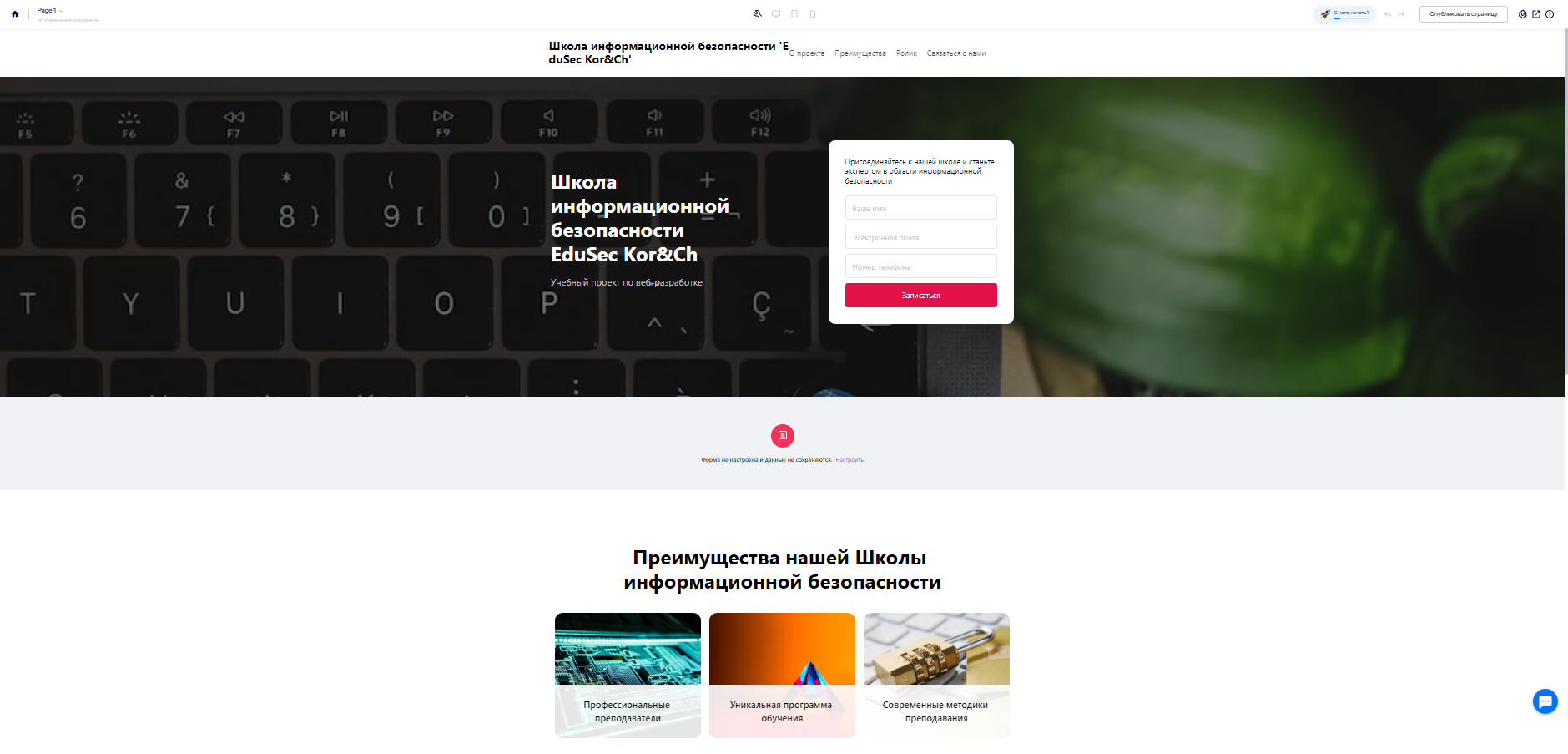


Рисунок 2 – Предварительный дизайн и расположение блоков проекта (1)

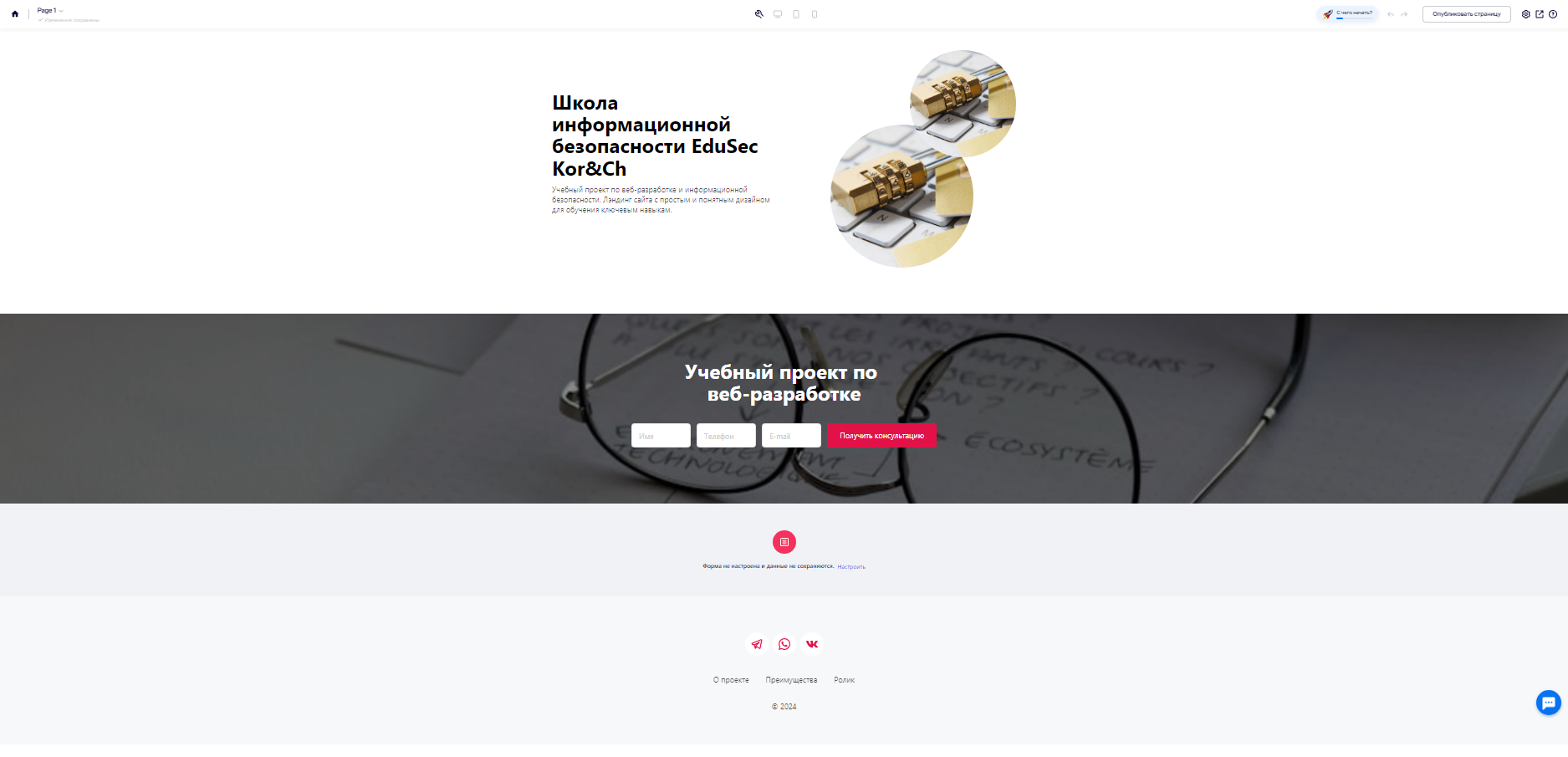


Рисунок 3 – Предварительный дизайн и расположение блоков проекта (2)

Дополнительно требуется отметить, что средства, использованные для создания предварительного дизайна проекта, не предполагали уже готовые страницы и стили, все страницы и стили писались вручную разработчиками проекта. Также предварительный дизайн был расширен по ходу разработки.

Дополнительные страницы – Преимущества и FAQ получились достаточно простыми, поэтому их дизайн был придуман по ходу разработки и без использования дополнительных средств.

# Разработка клиентской части приложения

**Публичные HTML файлы.**

В процессе разработки клиентской части приложения мы создали несколько HTML страниц, каждая из которых выполняет свою уникальную роль в структуре проекта. В данном разделе описывается назначение каждого блока интерфейса, его код и реализация адаптивности.

## index.html.

Файл index.html является главной страницей приложения и содержит следующие основные блоки:

* Навигационное меню
* Модальные окна для различных форм
* Секция с информацией о проекте
* Слайдер с услугами
* Форма заказа услуги
* Футер с контактной информацией и ссылками на другие страницы проекта.

## 1. Навигационное меню.

Блок кода навигационного меню представлен – рисунок 4.

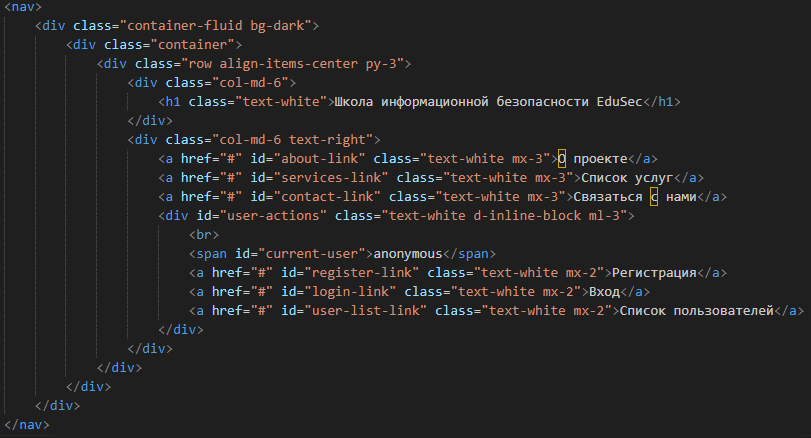


Рисунок 4 – Блок кода навигационного меню (верхней части сайта)

Адаптивность навигационного меню обеспечивается использованием Bootstrap классов, таких как container-fluid, container, row, col-md-6 и text-right. Это позволяет меню корректно отображаться на различных устройствах.

Выглядит оно так – рисунок 5.



Рисунок 5 – Внешний вид верхней части сайта (шапки-навигационного меню)

В данном случае реализовывать дополнительно стили в файле main.css не понадобилось, так как bootstrap предоставил подходящий дизайн этой части сайта.

Ссылки: «О проекте», «Список услуг», «Связаться с нами», «Регистрация», «Вход», «Список пользователей» при нажатии открывают модельные окна, которые заданы далее в коде index.html, а их функционал реализован через javascript файлы, стили через main.css.

Дополнительно адаптив для данного блока был реализован при помощи медиа-запросов – рисунок 6.



Рисунок 6 – Медиа-запросы для адаптива

## 2. Модальные окна.

Модальные окна используются для отображения различных форм и информации, таких как регистрация, вход, контактная форма и т.д. Большая часть ссылок открывающих модальные окна расположена в верхней части страницы, но таковые присутствуют и внизу страницы (footer).

*2.1 Модальное окно «О проекте»*

Блок кода – рисунок 7.

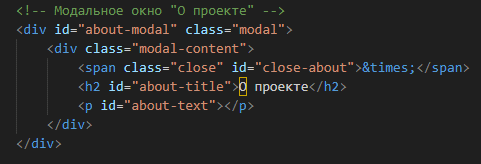


Рисунок 7 – Модальное окно «О проекте» (1)

Выглядит оно так – рисунок 8.



Рисунок 8 – Модальное окно «О проекте» (2)

По умолчанию текст для данного модального окна задан в файле main.js, но он также может быть задан через ссылку «Админ», реализованную в футере, которая из себя тоже представляет модальное окно.

Стили для всех модальных окон реализованы через bootstrap и main.css – рисунок 9.

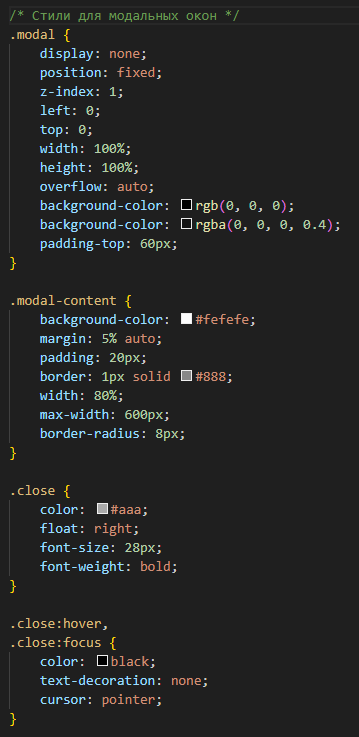


Рисунок 9 – Стили для модальных окон

*2.2 Модальное окно «Админ».*

Блок кода – рисунок 10.

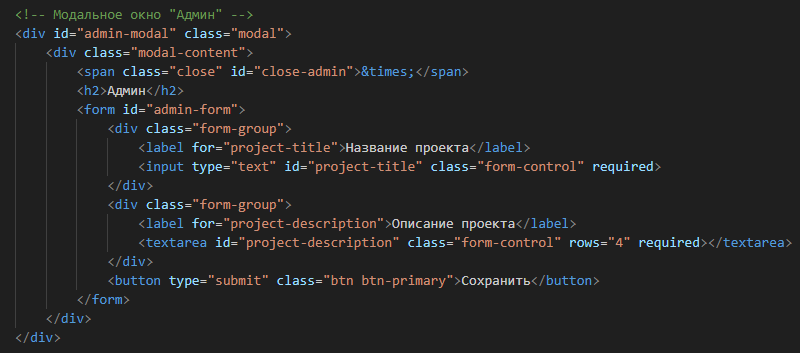


Рисунок 10 – Код модального окна «Админ»

Выглядит так – рисунок 11.

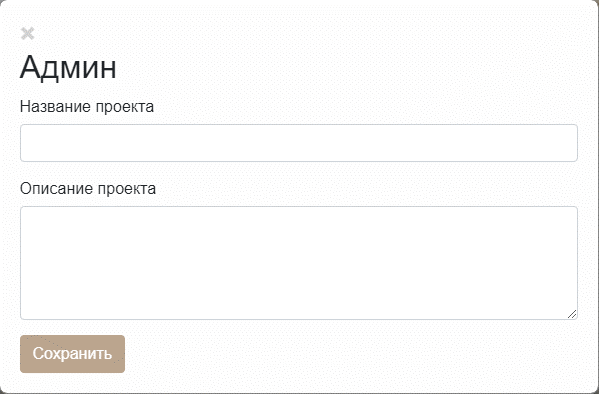


Рисунок 11 – Внешний вид модального окна «Админ»

Посредством данном модального окна задается содержимое модального окна «О проекте», содержимое сохраняется в localstorage, также оно по умолчанию задано в коде main.js.

Данное модальное окно не откроется, если текущий пользователь не admin.

*2.3 Модальное окно «Связаться с нами».*

Блок кода модального окна – рисунок 12.



Рисунок 12 – Код модального окна «Связаться с нами»

Посредством данного модального окна реализован пункт из технических требований к проекту – отправка e-mail сообщений с сайта. Реализован данный функционал через MailJS. Задан в файле main.js.

Также импорт скрипта для данного функционала – рисунок 13.

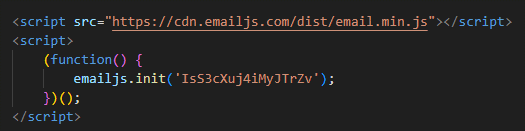


Рисунок 13 – Импорт функционала

Внешний вид модального окна – рисунок 14.

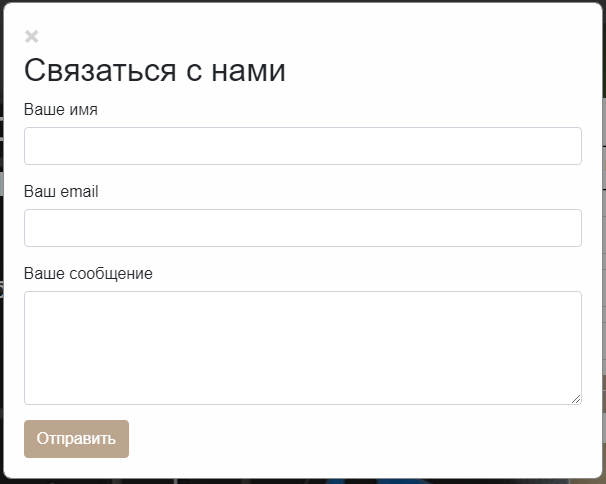


Рисунок 14 – Внешний вид «Связаться с нами»

Вот пример такого сообщения после заполнения этой формы – рисунок 15.

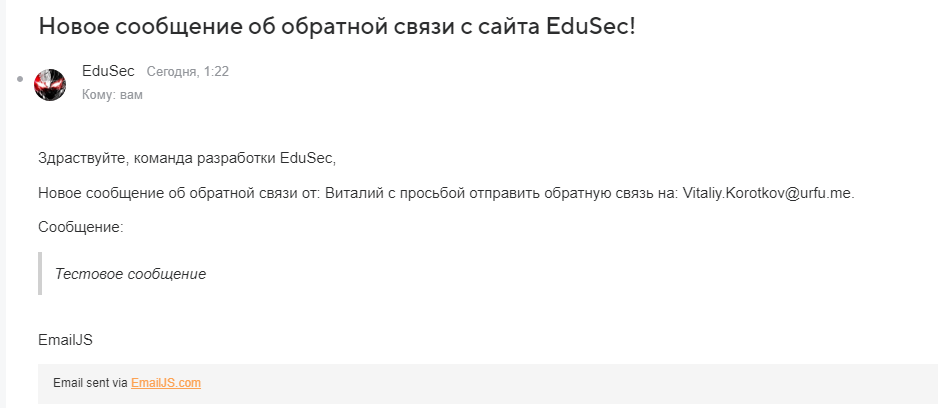


Рисунок 15 – Сообщение об обратной связи

*2.4 Модальное окно «Регистрация».*

Блок кода – рисунок 16.

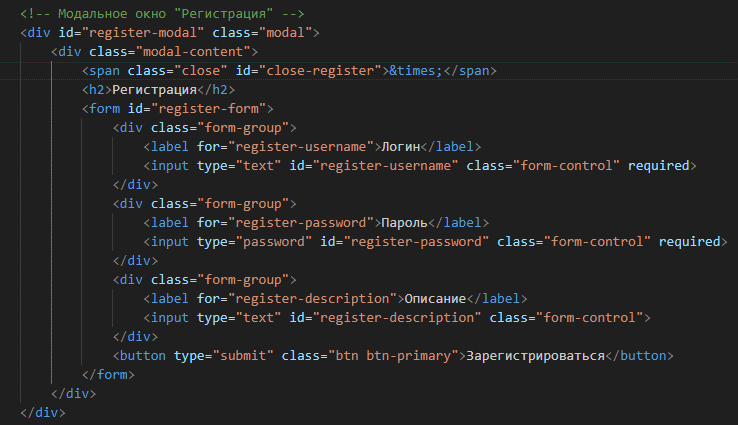


Рисунок 16 – Блок кода модального окна «Регистрация»

При помощи данного модального окна реализован функционал регистрации пользователей, при регистрации пользователя проверяется существует ли пользователь с таким логином, пароль пользователя не хранится в открытом виде, а хешируется при помощи CryptJS. Функционал регистрации реализован посредством auth.js разработчиками проекта.

Внешний вид данного окна – рисунок 17.

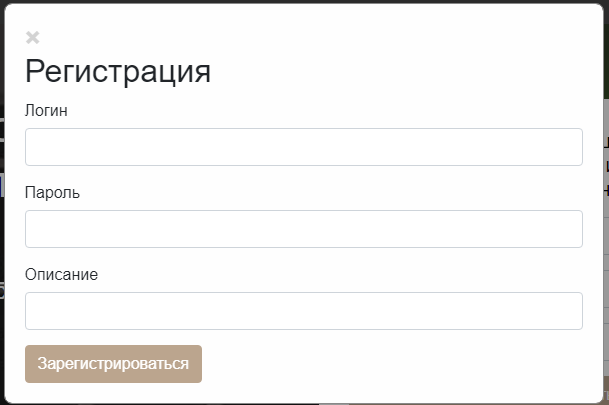


Рисунок 17 – Внешний вид окна «Регистрация»

*2.5 Модальное окно «Вход».*

Блок кода модального окна – рисунок 18.

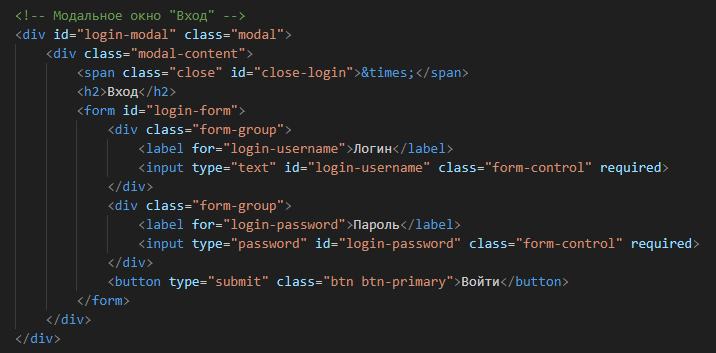


Рисунок 18 – Код модального окна «Вход»

Внешний вид данного модального окна – рисунок 19.

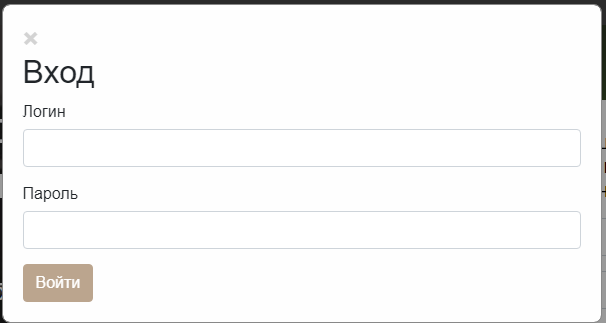


Рисунок 19 – Внешний вид модального окна «Вход»

Функционал входа реализован также посредством javascript а именно в файле auth.js.

Текущий пользователь отображается в верхней части страницы, если пользователь не авторизован, то отображается anonymous – рисунок 20.



Рисунок 20 – Отображение текущего пользователя

*2.5 Модальное окно «Список пользователей».*

Блок кода данного модального окна – рисунок 21.

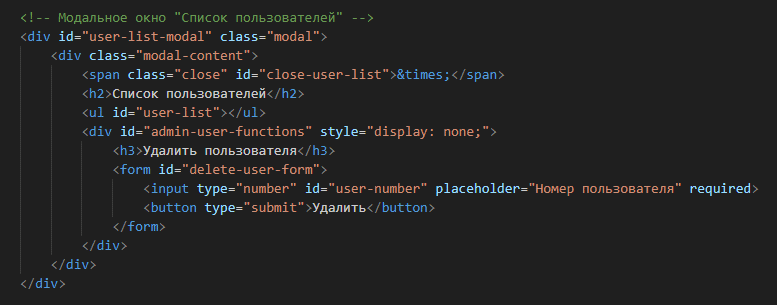


Рисунок 21 – Блок кода модального окна «Список пользователей»

Внешний вид данного окна – рисунок 22.

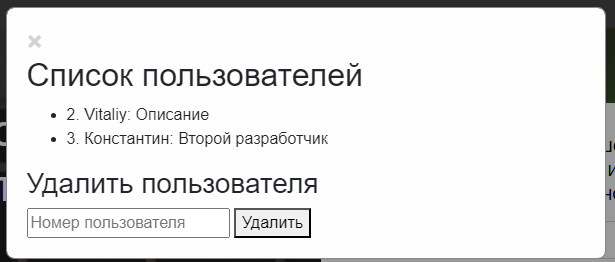


Рисунок 22 – Внешний вид модального окна «Список пользователей»

Если текущий пользователь не админ, то функционал удаления пользователя не показывается – рисунок 23.

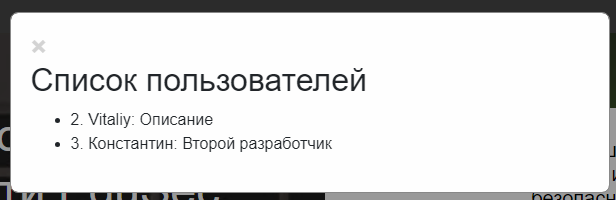


Рисунок 23 – Вид модального окна без авторизации под админом

*2.5 Модальное окно «Список услуг».*

Блок кода – рисунок 24.



Рисунок 24 – Блок кода модального окна «Список услуг»

Внешний вид у данного окна – рисунок 25.



Рисунок 25 – Внешний вид модального окна «Список услуг»

Без наличия авторизации под администратором функционал добавления услуг и удаления услуг показываться не будет – рисунок 26.

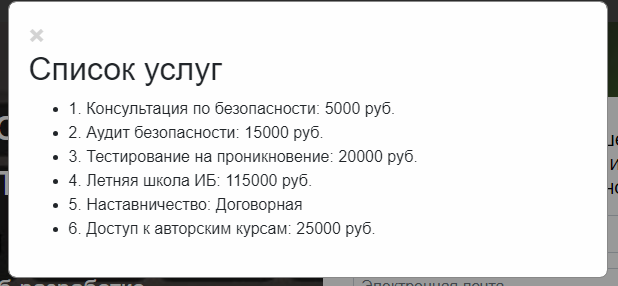


Рисунок 26 – Внешний вид без администратора

По умолчанию содержимое данного списка задано в файле services.js, там же реализован функционал удаления/добавления, данный блок симулирует динамический вывод из БД.

## 3. Блок оферты и формы записи.

Данный блок представлен ниже блока верха страницы с навигационной панелью, внешний вид его такой – рисунок 27.

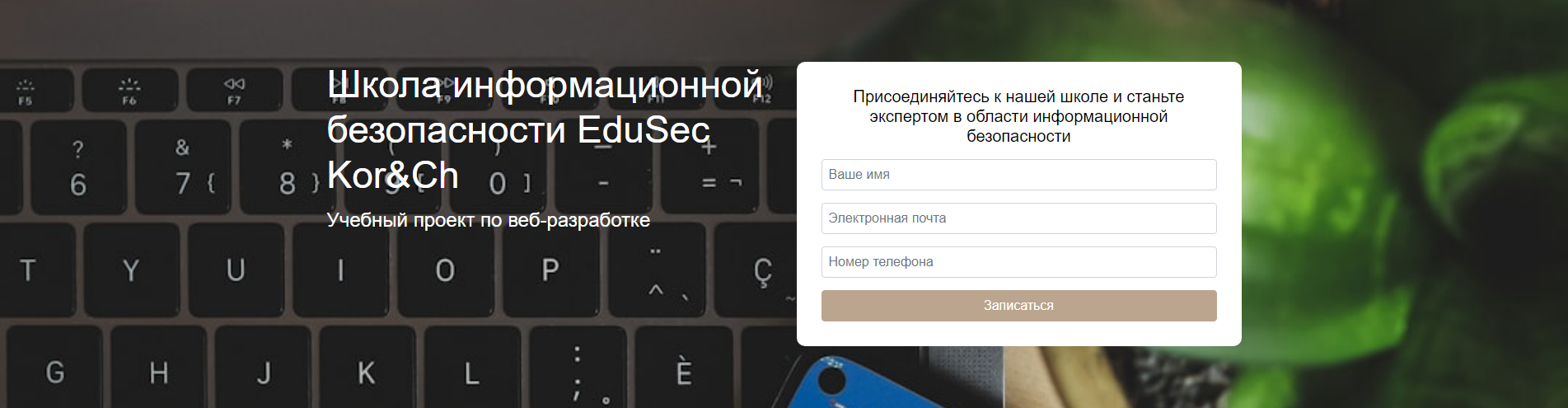


Рисунок 27 – Внешний вид блока

Блок кода в файле index.html – рисунок 28.



Рисунок 28 – Блок кода

Валидация в форме, представленной в данном блоке реализована при помощи файла val-3.js и встроенной валидации html. Отправка данных в этой форме не предусмотрена.

## 4. Блок преимуществ школы информационной безопасности.

Внешний вид данного блока – рисунок 29.

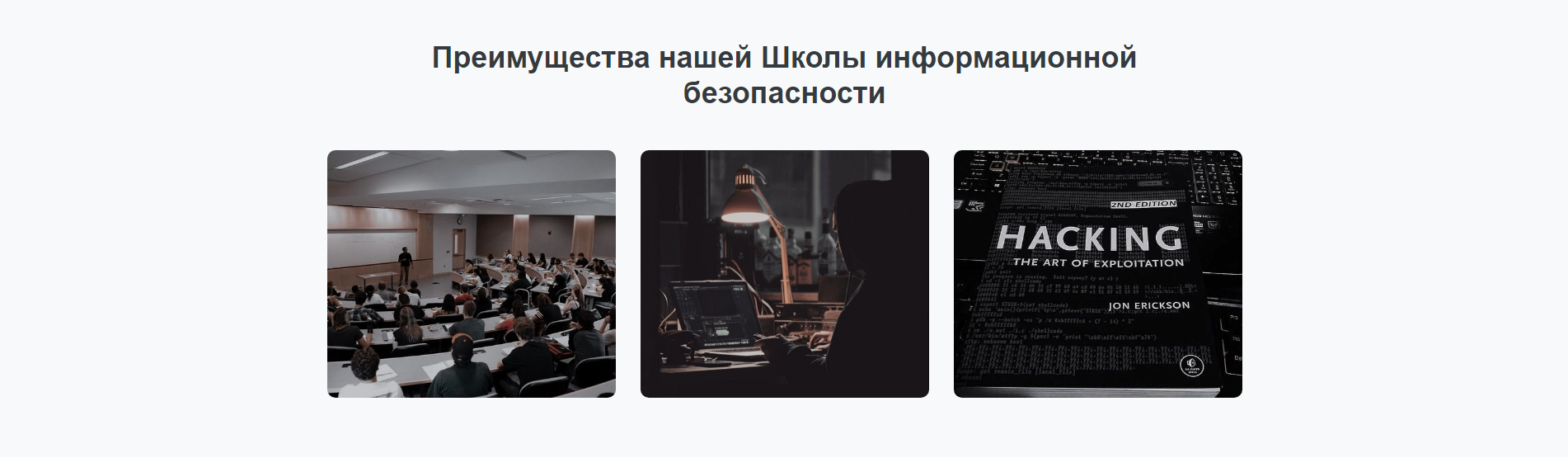


Рисунок 29 – Внешний вид блока

В данном блоке реализована анимация при наведении курсора на каждый мини-блок. Картинки импортируются из подкаталога проекта img.

Анимации реализованы в файле main.css, выглядят так – рисунок 30.

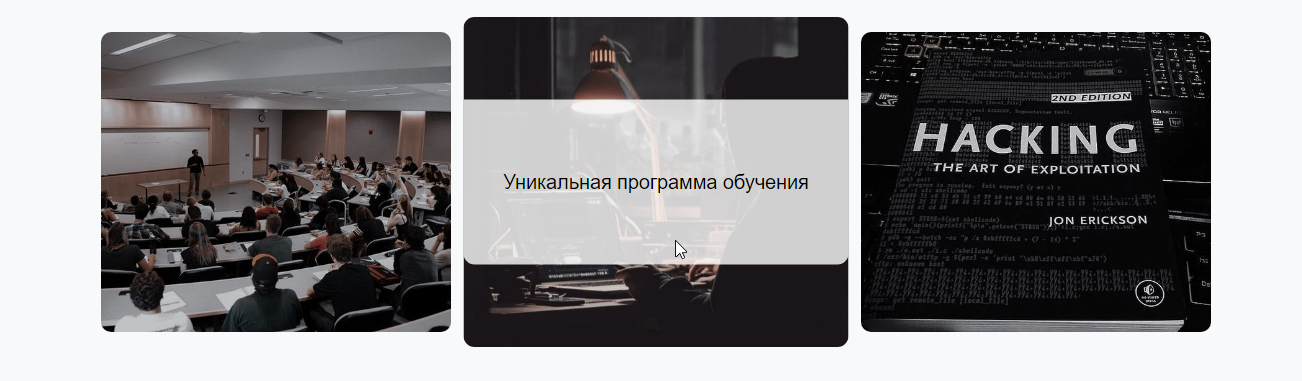


Рисунок 30 – Анимация

Код в файле index.html – рисунок 31.

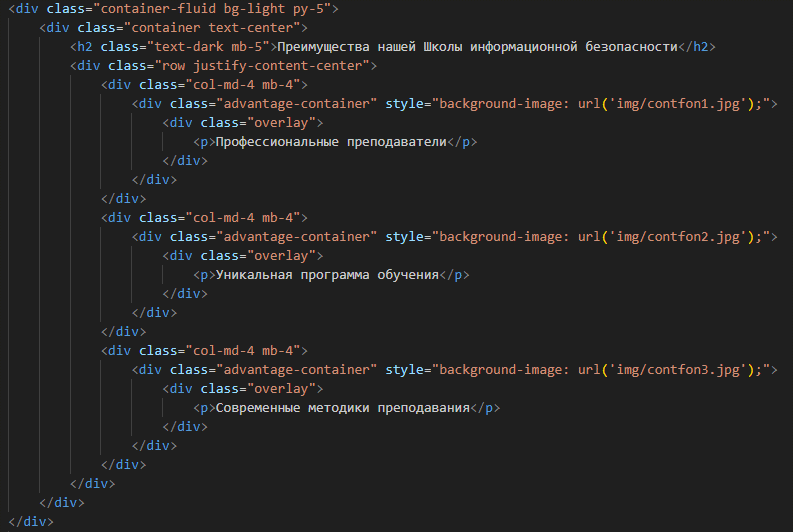


Рисунок 31 – Блок кода

Анимации в файле main.css представлены так – рисунок 32.

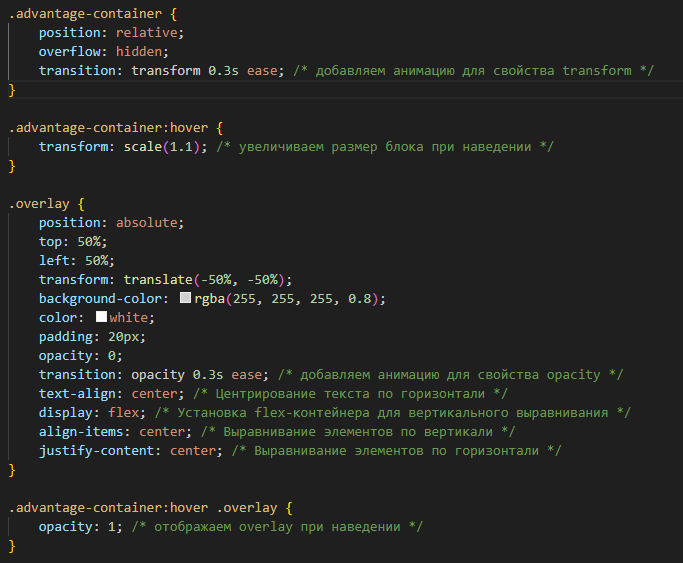


Рисунок 32 – Код в main.css

## 5. Слайдер с предоставляемыми услугами.

Внешний вид данного блока – рисунок 33.

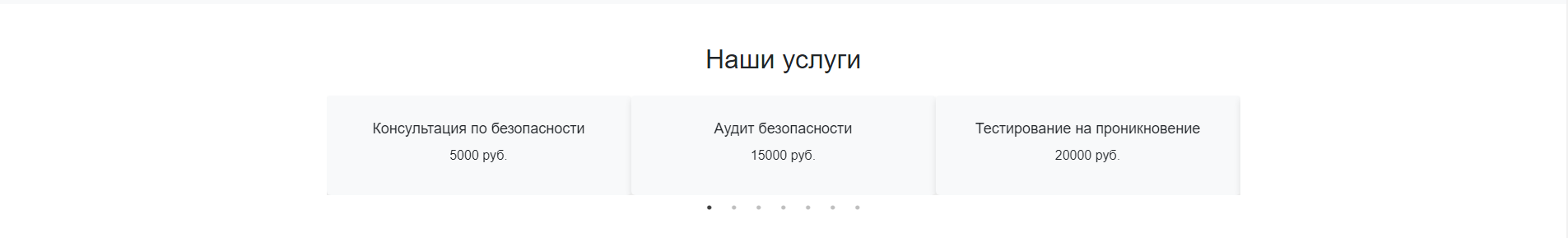


Рисунок 33 – Внешний вид блока

Код в index.html – рисунок 34.

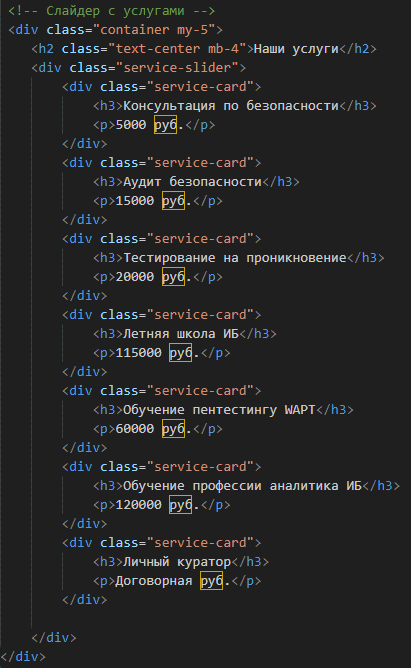


Рисунок 34 – Код в файле index.html

Стили для слайдера реализованы в main.css – рисунок 35.



Рисунок 35 – Стили в main.css

Также в index.html есть еще один блок кода, относящийся к данному слайдеру – рисунок 36.



Рисунок 36 – Блок кода

В данном случае импортируется сторонний скрипт для реализации данного слайдера, адаптив реализован.

## 6. Форма заказа услуги.

Внешний вид данного блока – рисунок 37.

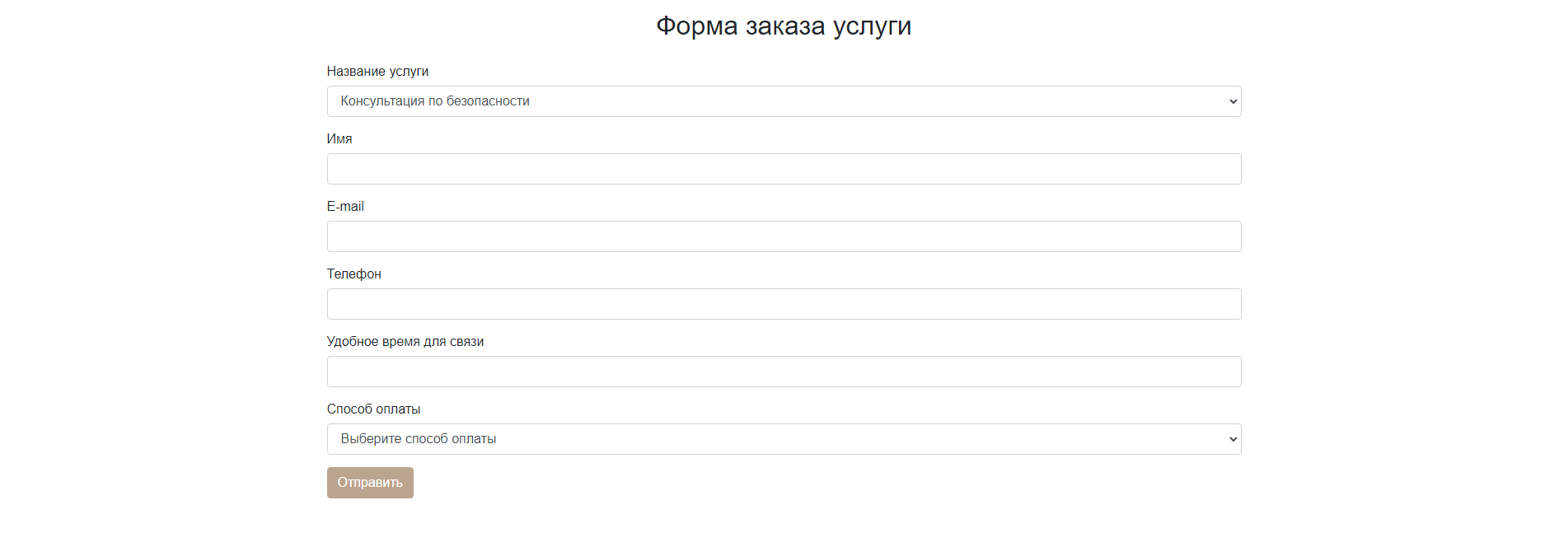


Рисунок 37 – Внешний вид блока

Код в файле index.html для блока – рисунок 38.

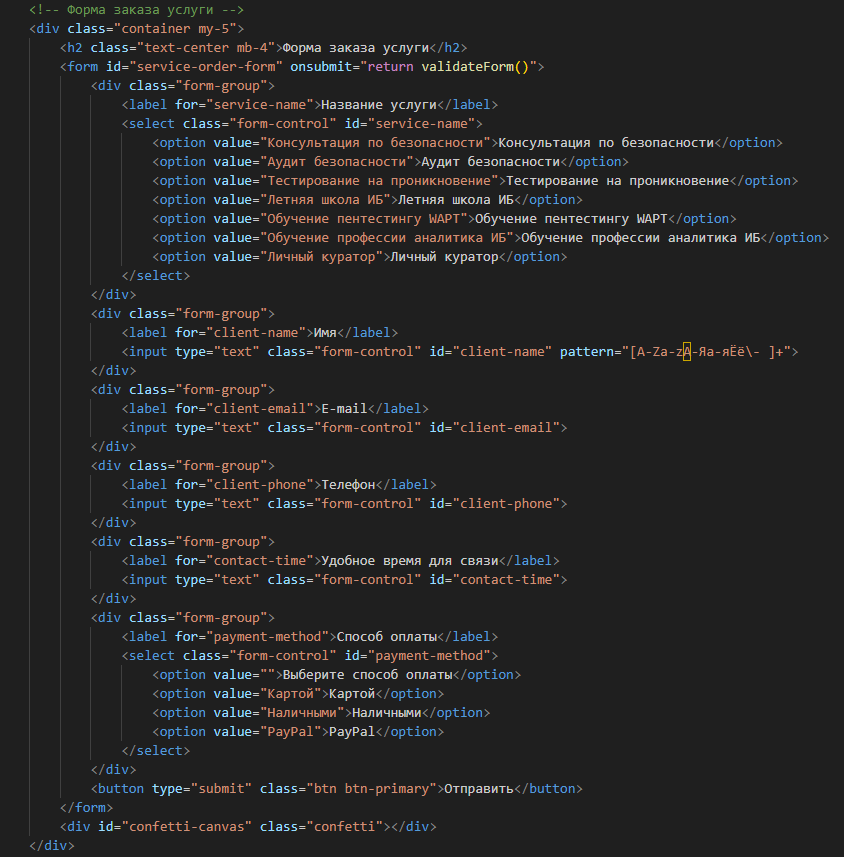


Рисунок 38 – Код блока

Валидация для данной формы реализована посредством javascript в файле – usl-val.js.

При отправке данной формы проигрывается анимация конфетти, данные в форме сбрасываются.

Анимация реализована сторонней библиотекой - <script src="https://cdn.jsdelivr.net/npm/canvas-confetti@1.5.1"></script>.

Также настроена в файле main.css – рисунок 39.

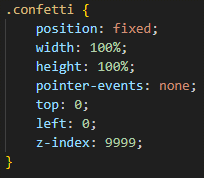


Рисунок 39 – Блок кода main.css

Адаптив реализован посредством bootstrap.

## 7. Блок оферты.

Внешний вид данного блока – рисунок 40.

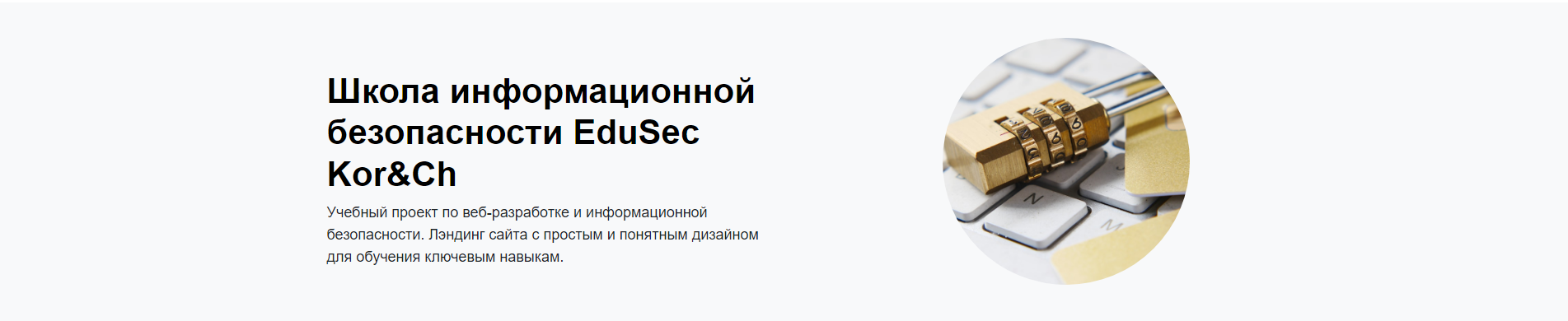


Рисунок 40 – Внешний вид блока

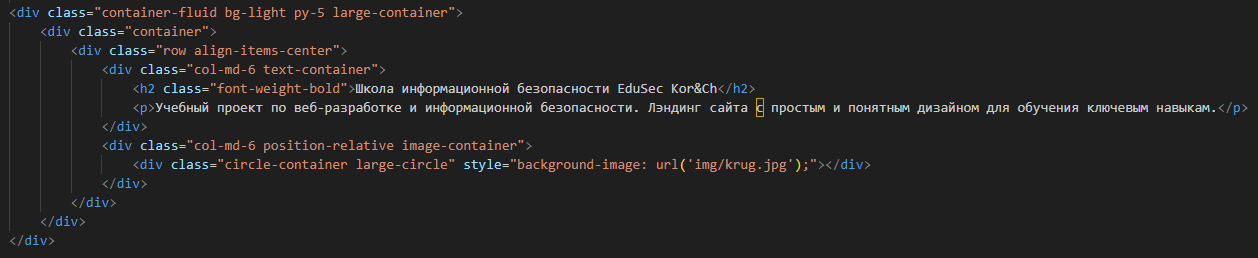


Рисунок 41 – Код блока

Также для данного блока заданы стили main.css и адаптив при помощи медиа-запросов и bootstrap – рисунок 42.

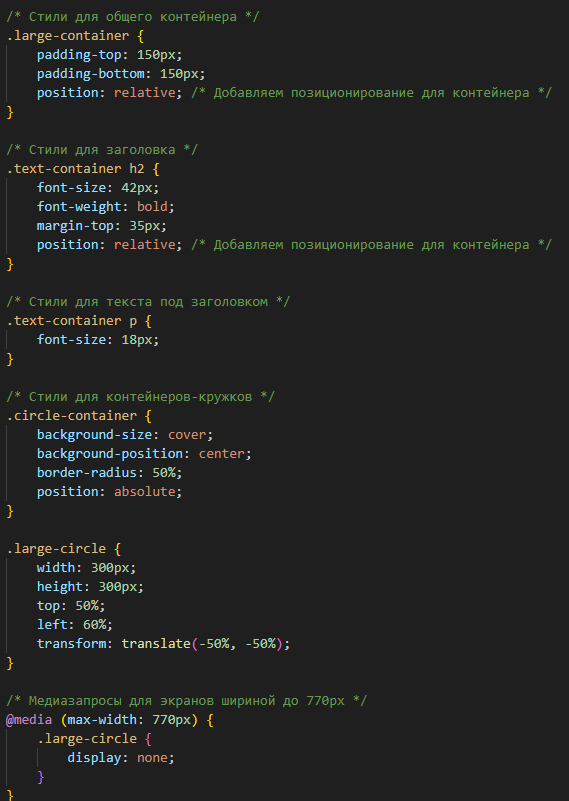


Рисунок 42 – main.css

## 8. Форма получения консультации.

Внешний вид данного блока – рисунок 43.

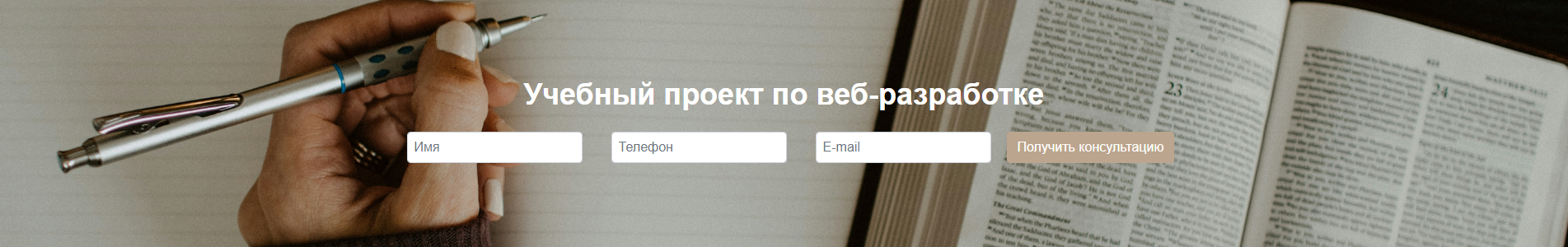


Рисунок 43 – Внешний вид блока

Код в файле index.html – рисунок 44.

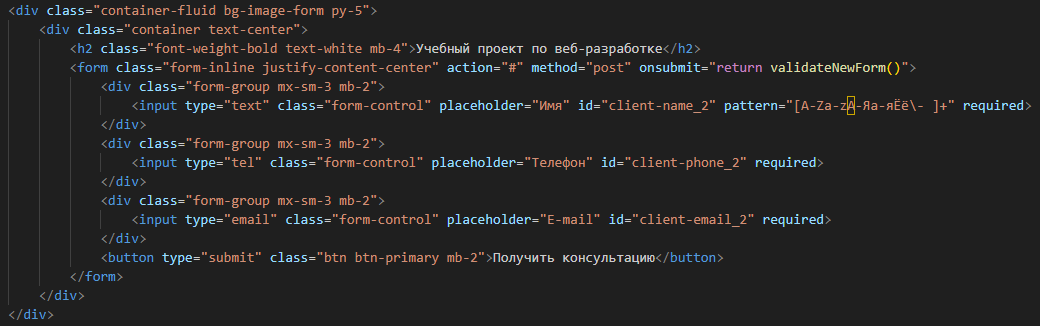


Рисунок 44 – Код в index.html

Валидация для данной формы реализована посредством javascript в файле validation.js.

Отправка куда-либо данных не предусмотрена, адаптив реализован при помощи bootstrap.

## 9. Нижняя часть страницы - футер.

Внешний вид данного блока – рисунок 45.

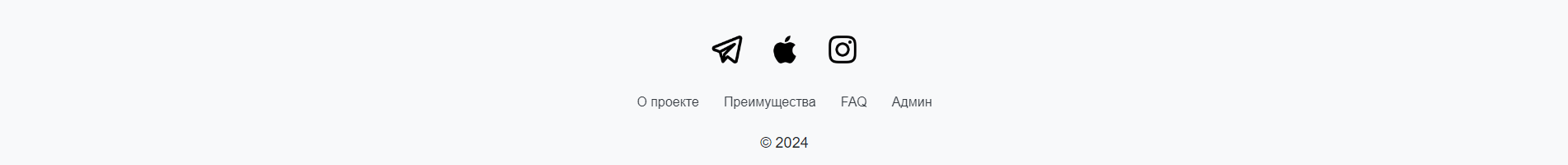


Рисунок 45 – Внешний вид футера

Блок кода в index.html – рисунок 46.

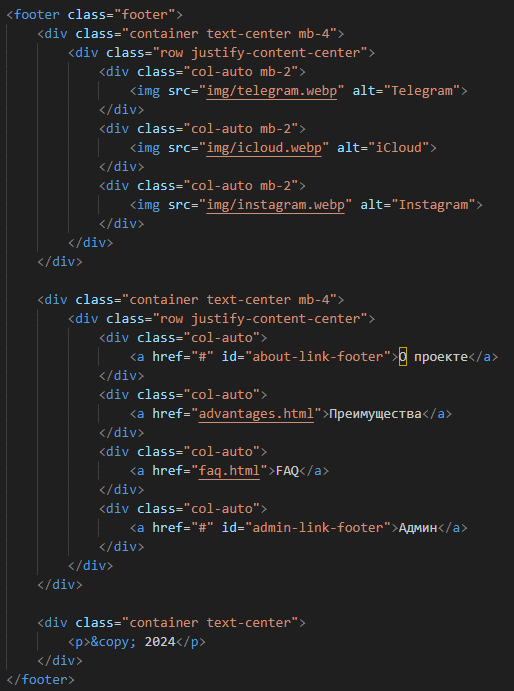


Рисунок 46 – Блок кода

Для картинок реализована анимация в main.css – рисунок 47.

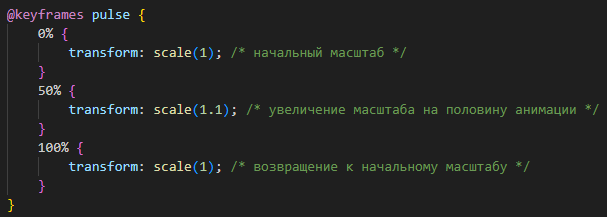


Рисунок 47 – Анимация keyframes

Легкая пульсация. В футере предусмотрены ссылки на модальные окна «О проекте», «Админ» и ссылки на дополнительные страницы – «Преимущества», и «FAQ».

## 10. Ссылки на собственные скрипты.

Ссылки на собственные скрипты расположены в нижней части файла index.html – рисунок 48.

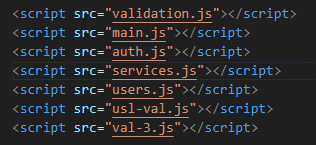


Рисунок 48 – Импорт скриптов

## advantages.html

Внешний вид данной дополнительной динамической страницы такой – рисунок 49.

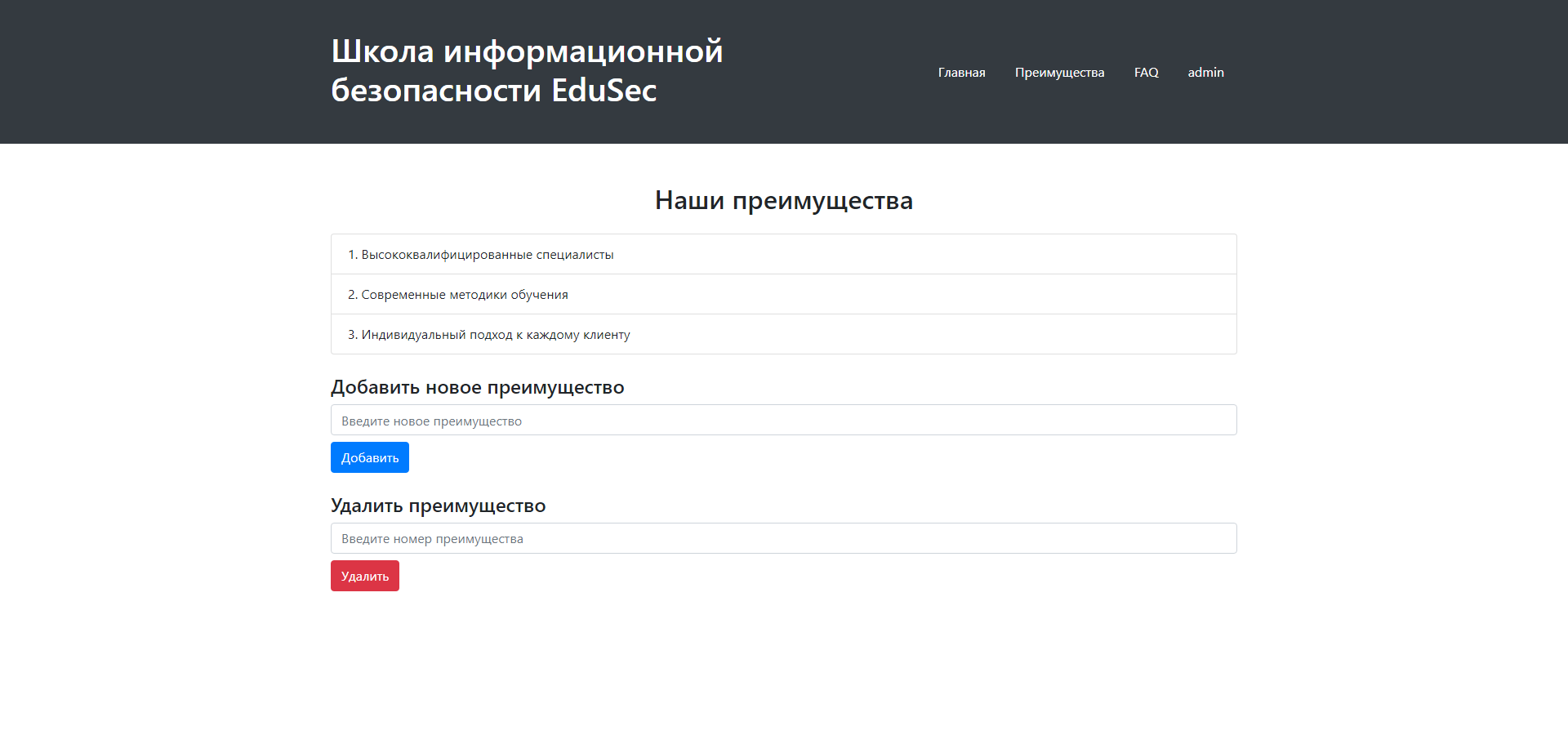


Рисунок 49 – Внешний вид

Здесь реализован список преимуществ выводящийся из localstorage. Функционал данной страницы реализован в файле advantages.js.

На странице отображается текущий пользователь (если пользователь admin, то отображается функционал удаления и добавления данных).

Также есть панель навигации.

Вот так выглядит страница без логина администратора – рисунок 50.

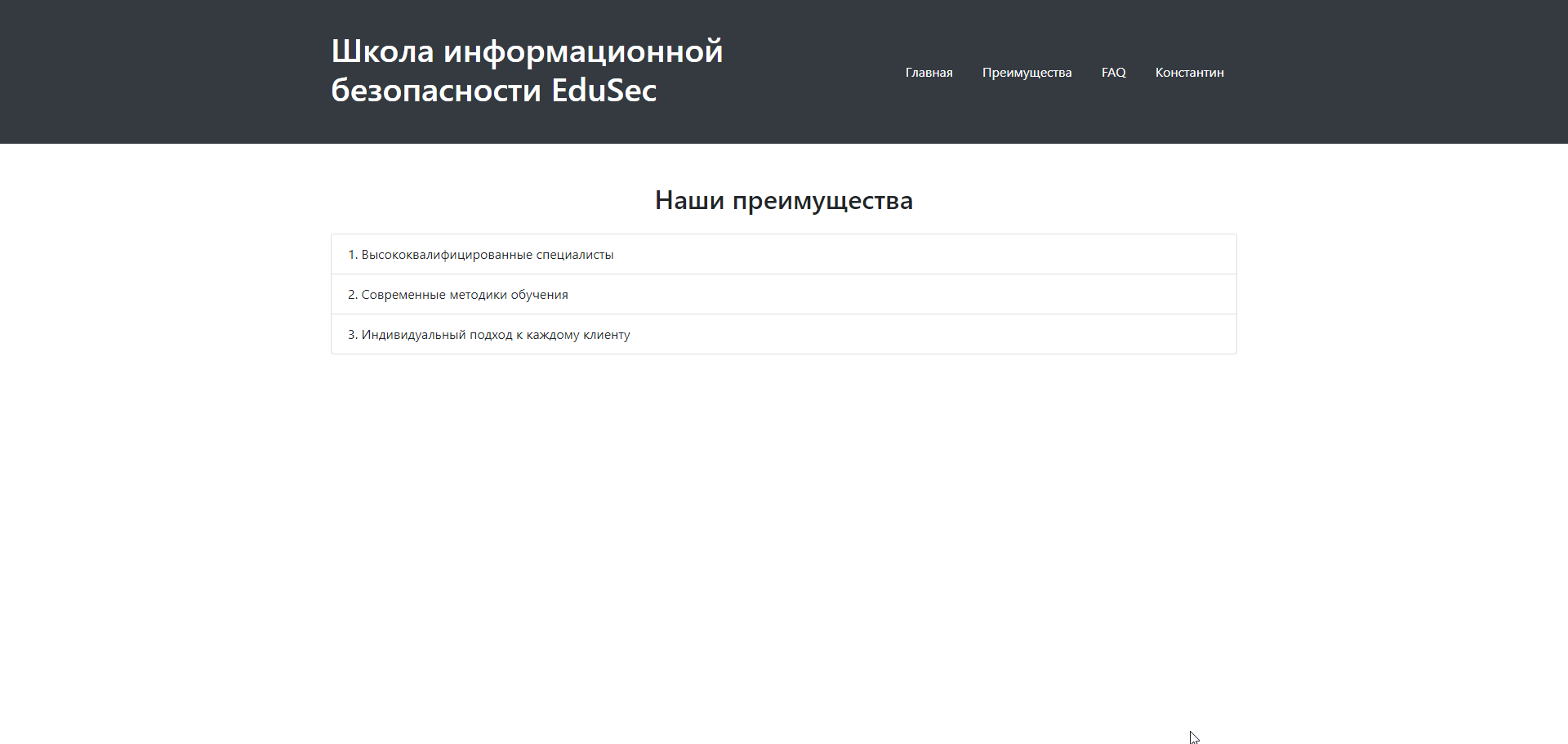


Рисунок 50 – Обычный пользователь

Код данной страницы – рисунок 51.

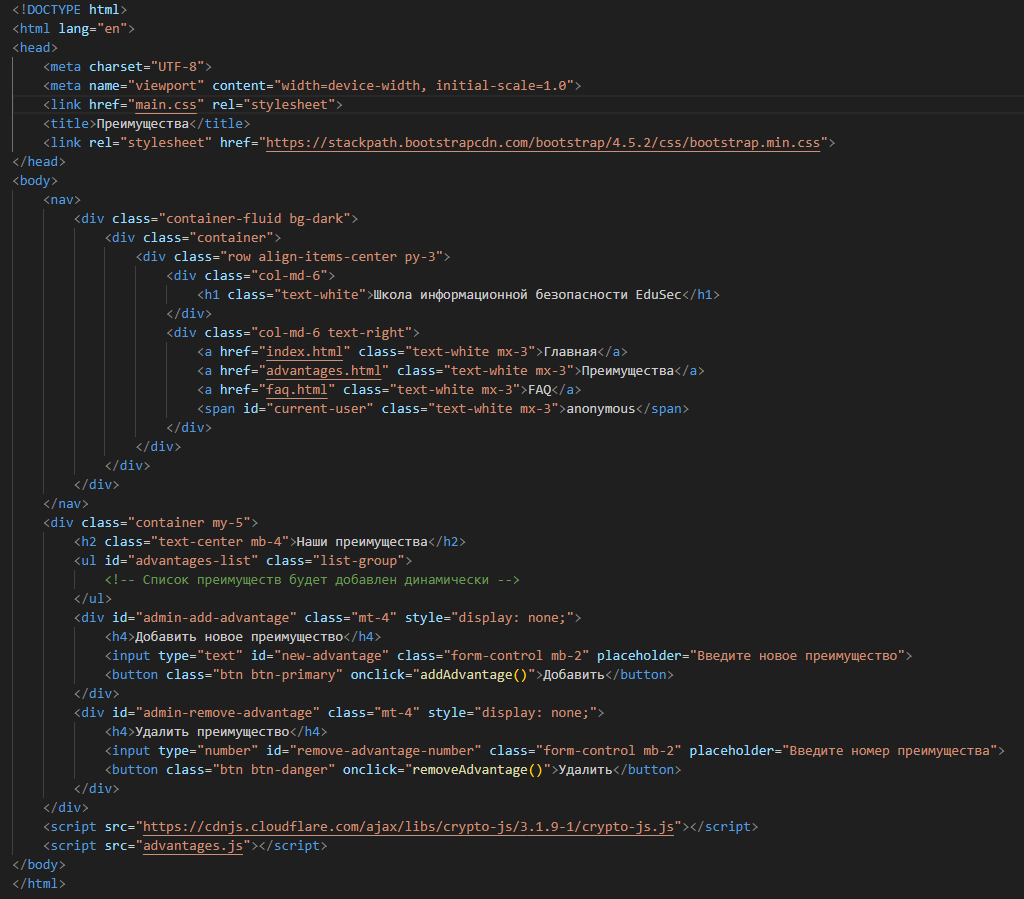


Рисунок 51 – Код страницы преимуществ

Адаптив также реализован при помощи bootstrap.

## faq.html

Внешний вид данной страницы под администратором – рисунок 52.

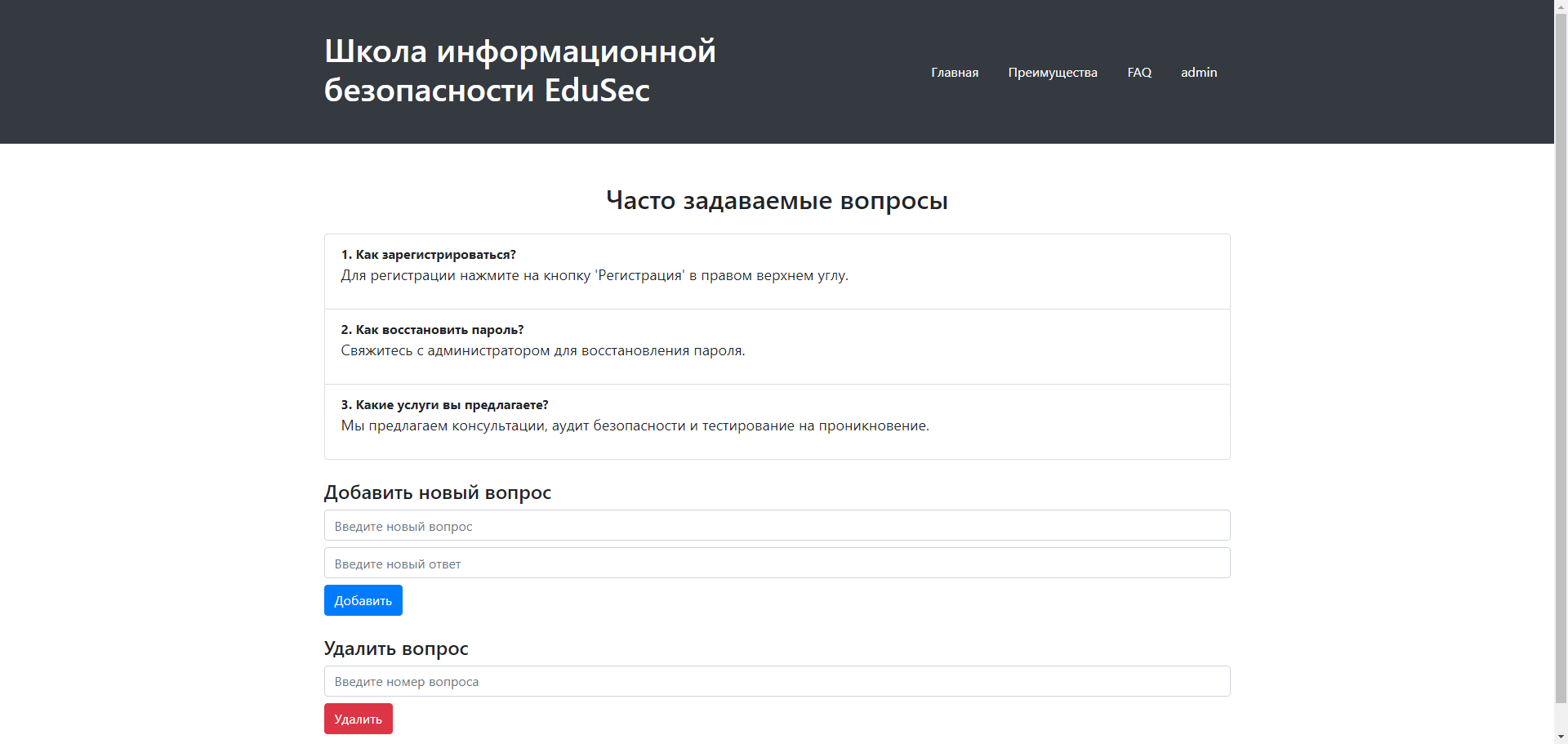


Рисунок 52 – Внешний страницы FAQ

На данной странице реализована навигационная панель, функционал данной страницы обеспечивается при помощи faq.js.

Реализован динамический вывод из псевдо-БД (localstorage), реализован функционал добавления и удаления данных под администратором.

Внешний вид без логина администратора – рисунок 53.

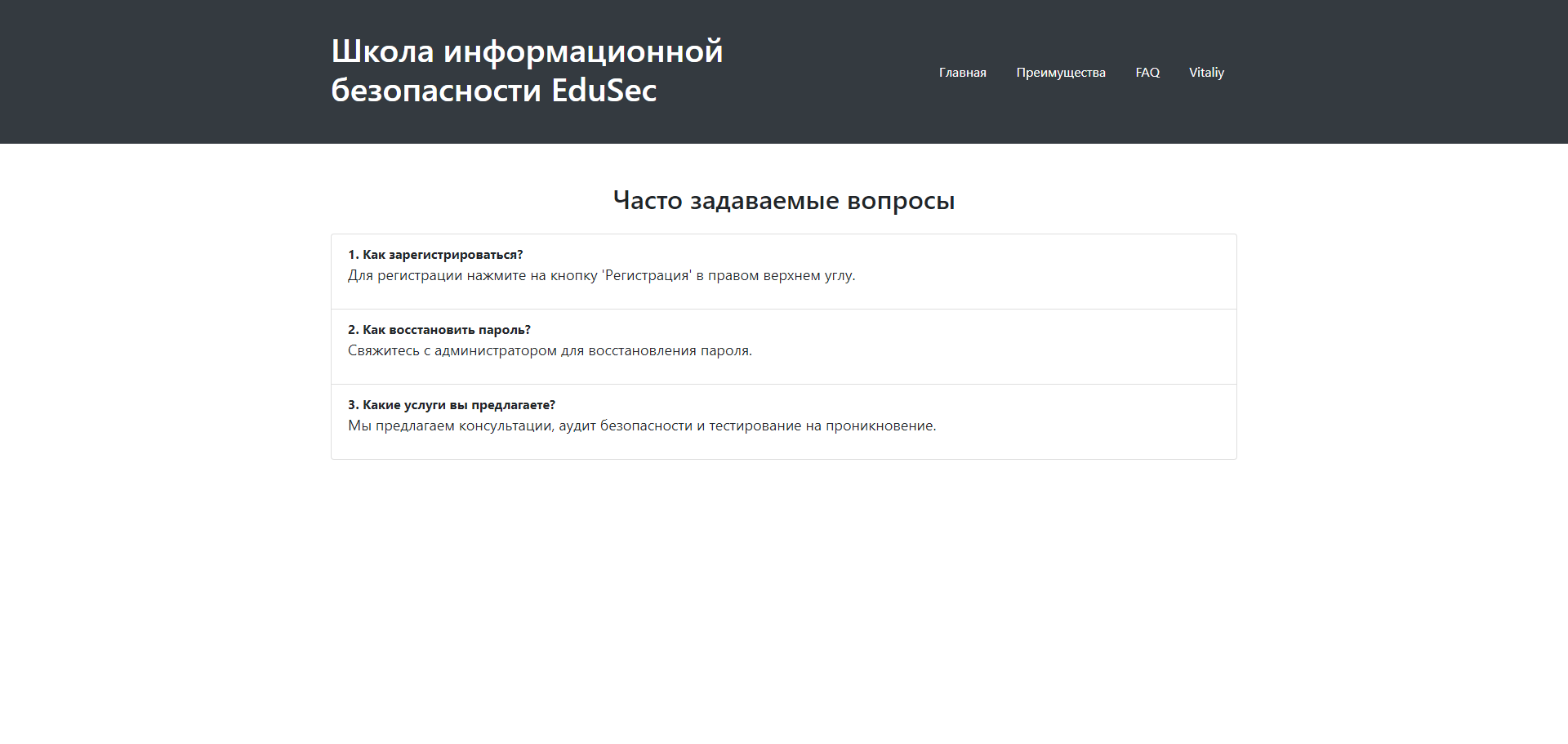


Рисунок 53 – Внешний вид страницы FAQ без администратора

Код страницы – рисунок 54.

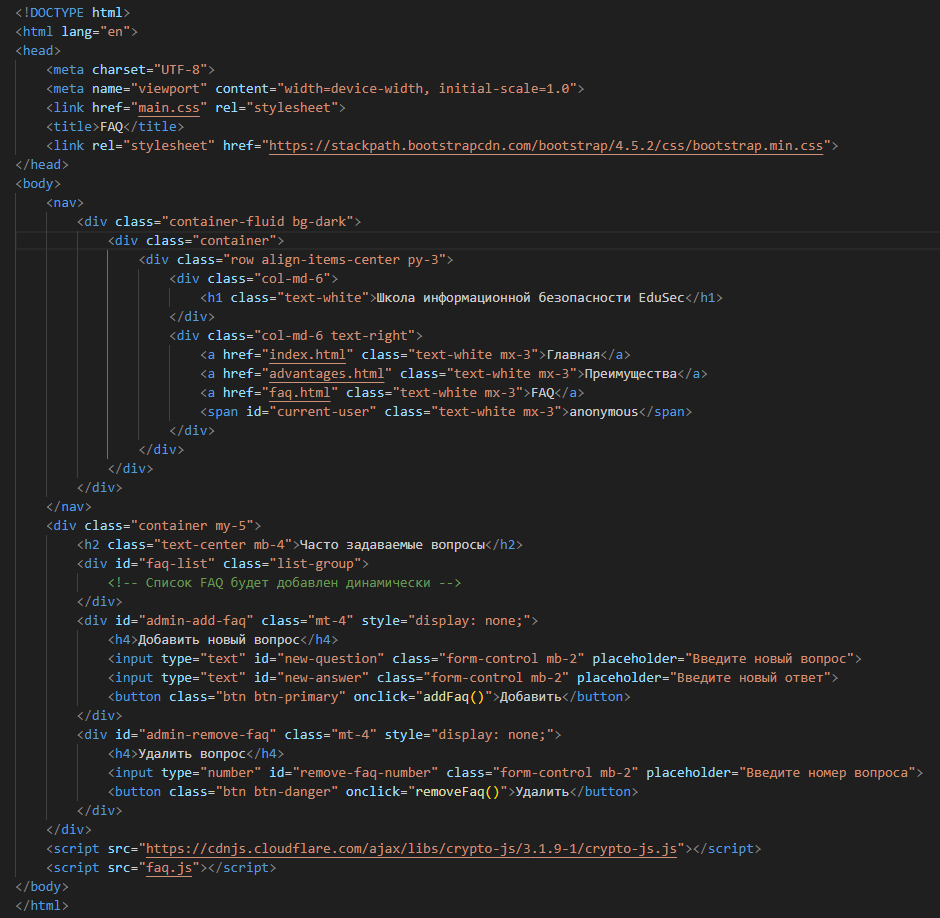


Рисунок 54 – Код страницы

Адаптив реализован при помощи bootstrap.

# Разработка серверной части приложения

В нашем проекте используется клиентская модель хранения данных с использованием LocalStorage для имитации работы с базой данных. Это решение подходит для небольших веб-приложений, где не требуется сложная серверная инфраструктура и постоянное подключение к интернету. Рассмотрим подробнее архитектуру и взаимодействие клиентской и «серверной» частей нашего приложения.

**Архитектура.**

Наша архитектура состоит из трех основных компонентов:

* Клиентская часть: Интерфейс пользователя, написанный на HTML, CSS и JavaScript.
* LocalStorage: Встроенное хранилище в браузере, используемое для сохранения данных.
* Логика взаимодействия: JavaScript-код, который отвечает за взаимодействие с LocalStorage.

**Схема данных.**

Вместо традиционной базы данных, мы используем LocalStorage для хранения информации о пользователях и заказах. В нашем проекте определены следующие основные сущности:

1. Пользователи (user-list):

register-username: Имя пользователя.

register-password: Пароль пользователя.

register-description: Описание пользователя.

2. Список услуг (services-list):

service-name: Название услуги.

service-price: Цена услуги.

3. Преимущества (advantages-list):

new-advantage: Текст преимущества.

4. Часто-задаваемые вопросы (FAQ, faq-list):

new-question: Текст вопроса.

new-answer: Текст ответа к вопросу.

5. Описание и название проекта (about-modal):

about-title: Название проекта.

about-text: Описание проекта.

**Процессы взаимодействия.**

1. Когда пользователь заполняет форму регистрации, данные сохраняются в LocalStorage.

2. При попытке входа, приложение проверяет данные в LocalStorage.

3. Список услуг выводится из localstorage.

4. Услуги добавляются администратором в localstorage.

5. Описание и название проекта выводится из localstorage.

6. Администратор обновляет содержимое описания и названия проекта в localstorage.

7. Администратор имеет возможность удаления пользователей из localstorage.

8. Список преимуществ проекта выводится из localstorage.

9. Администратор может обновлять список преимуществ проекта в localstorage.

10. Список часто-задаваемых вопросов выводится из localstorage.

11. Администратор может обновлять список часто-задаваемых вопрос в localstorage.

**Примеры кода.**

1. Код файла users.js – данный файл регулирует процесс удаления пользователей, вывода списка пользователей – рисунок 55.



Рисунок 55 – users.js

2. Код файла auth.js – данный файл регулирует задание пользователя по умолчанию (admin), подгрузку пользователей из локального хранилища, отображение текущего пользователя, взаимодействие: модальные окна 🡪 локальное хранилище, также здесь реализована регистрация и вход пользователей. Пример кода функций для регистрации и входа пользователей – рисунок 56.

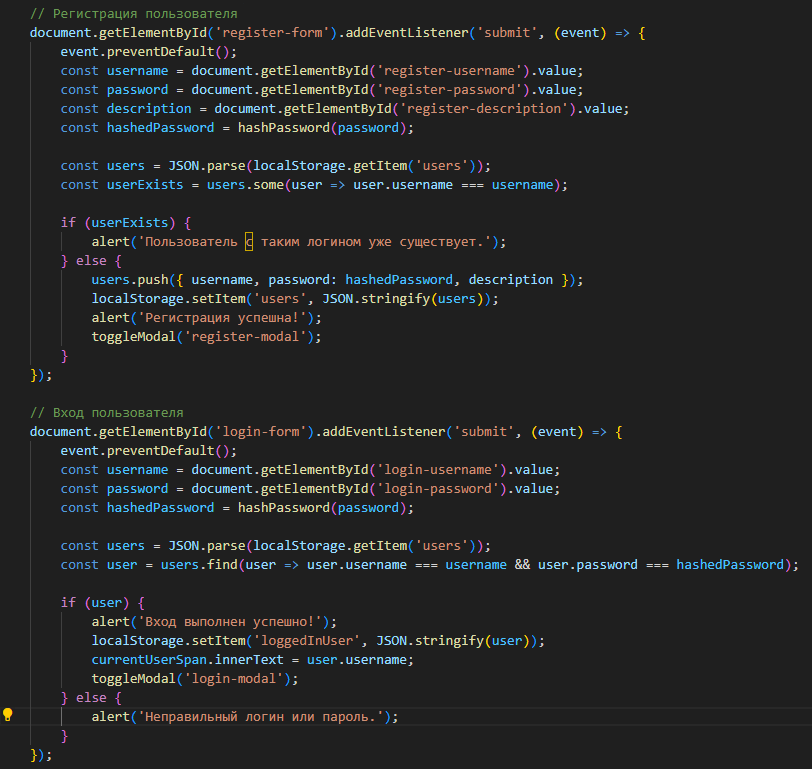


Рисунок 56 – auth.js

3. Код файла main.js – данный файл регулирует обработчики для модальных окон, данные о проекте по умолчанию, также регулирует, например, отправку e-mail сообщений с сайта – рисунок 57.



Рисунок 57 – Функции файла main.js

4. Код файла services.js – данный файл регулирует список предоставляемых услуг, добавляет функционал удаления, добавления услуг – рисунок 58.

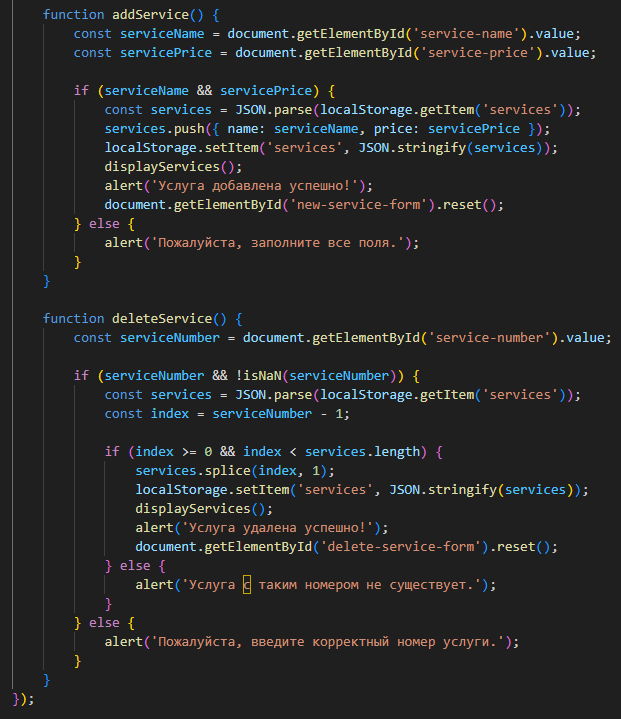


Рисунок 58 – Пример кода из файла services.js

5. Код файла advantages.js – данный файл отвечает за функционал дополнительной страницы «Преимущества» в нем реализованы функции отображения списка преимуществ из localstorage, добавления преимуществ, удаления преимуществ – рисунок 59.

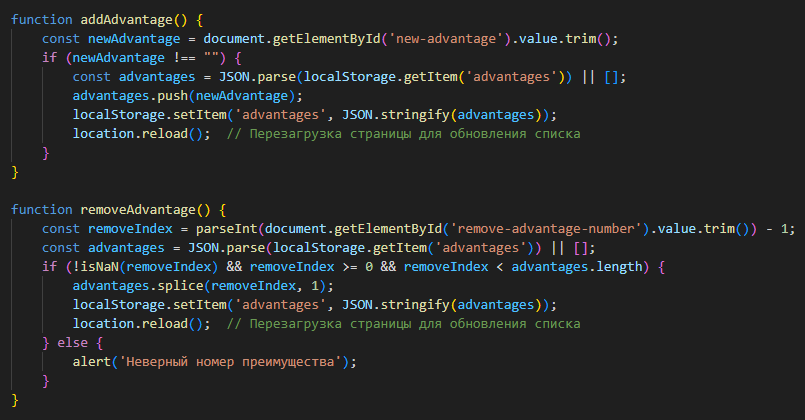


Рисунок 59 – Пример функций из advantages.js

6. Код файла faq.js – данный файл отвечает за функционал дополнительной страницы FAQ, в нем реализованы функции отображения списка часто-задаваемых вопросов из localstorage, удаление, добавление вопросов и ответов – рисунок 60.



Рисунок 60 – Пример функций из faq.js

7. Код файла validation.js – данный файл отвечает за валидацию некоторых форм на сайте, в нем заданы шаблоны по котором должен проверяться ввод и какие сообщения в случаях несоответствия должны выводиться – рисунок 61.



Рисунок 61 – Файл validation.js

8. Также есть два дополнительных файла скриптов валидации, которые были созданы для других форм по шаблону validation.js – это usl-val.js и val-3.js.

Файл usl-val.js предназначен для валидации ввода в форму заказа услуги, вся валидации этой формы целиком и полностью основана на этом файле, вот код файла – рисунок 62.

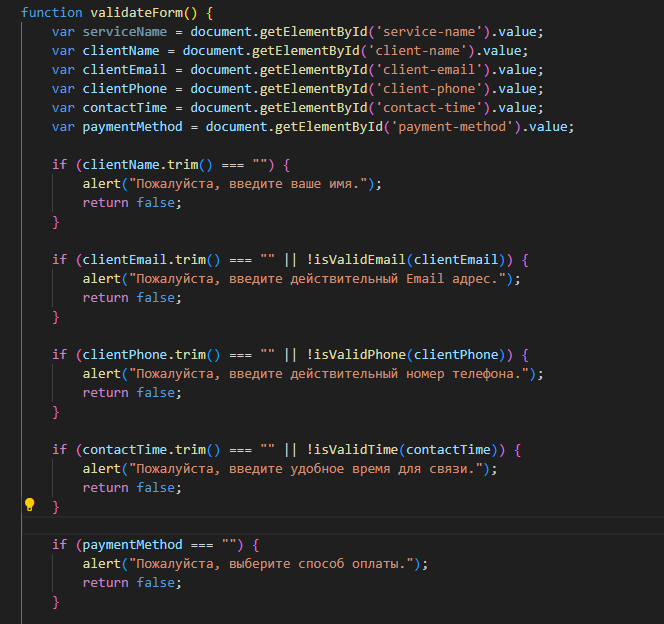


Рисунок 62 – Файл usl-val.js

Также в этом файле задана анимация конфетти для этой формы при отправке данных и шаблончики – рисунок 63.



Рисунок 63 – Файл usl-val.js

Файл val-3.js сделан для валидации формы записи под самым верхним блоком сайта – рисунок 64.



Рисунок 64 – Код файла val-3.js

# ЗАКЛЮЧЕНИЕ

В процессе разработки веб-приложения «Лендинг Школы по информационной безопасности EduSec» нами была выполнена значительная работа, которая включала проектирование, выбор инструментальных средств, создание клиентской части, а также использование LocalStorage для имитации серверной части.

На этапе проектирования была создана четкая структура проекта, включая разработку блок-схемы архитектуры. Это помогло нам визуализировать и понять взаимосвязи между различными компонентами приложения. Мы определили основные модули и их взаимодействие, что стало фундаментом для дальнейшей разработки.

Для реализации проекта были выбраны современные и проверенные технологии, такие как HTML, CSS, JavaScript, а также фреймворк Bootstrap для обеспечения адаптивности интерфейса. Использование LocalStorage позволило нам хранить данные пользователей и заказы услуг без необходимости настройки серверной части, что значительно упростило процесс разработки и снизило затраты.

Разработка интерфейса пользователя была выполнена с учетом современных требований к дизайну и юзабилити. Были реализованы различные секции сайта, включая навигационное меню, формы регистрации и входа, модальные окна для информации о проекте, контактные формы и т.д.

Вместо полноценной серверной архитектуры и базы данных, мы использовали LocalStorage для хранения данных пользователей и заказов. Это позволило реализовать основные функции приложения, такие как регистрация, вход, создание и просмотр заказов и тд. LocalStorage предоставил возможность сохранять данные непосредственно в браузере пользователя, обеспечивая быстрый доступ к ним.

На выходе мы получили функциональный лендинг для школы информационной безопасности EduSec, который включает в себя все заявленные функции:

1. Информативный и удобный, единообразный интерфейс – пользователь может легко найти необходимую информацию о школе и доступных услугах.
2. Возможность регистрации и входа – пользователи могут создавать учетные записи и входить в систему для использования дополнительных функций.
3. Форма заказа услуг – пользователи могут заказывать услуги школы, заполняя простые формы.
4. Адаптивный дизайн – сайт корректно отображается на различных устройствах, обеспечивая удобство использования для всех пользователей.

Процесс разработки веб-приложения «Лендинг Школы по информационной безопасности EduSec» продемонстрировал, что использование современных инструментов и технологий позволяет создавать качественные и функциональные приложения в короткие сроки. Мы достигли всех поставленных задач, реализовав заявленные функции и обеспечив удобство использования для конечных пользователей.

Кроме того, использование LocalStorage показало свою эффективность для небольших проектов, не требующих сложной серверной архитектуры. Тем не менее, для более крупных и сложных проектов рекомендуется использование полноценной базы данных и серверной части, что обеспечит безопасность данных и масштабируемость приложения.

В целом, проект можно считать успешным, так как он не только соответствует заявленным требованиям, но и предоставляет возможность для дальнейшего развития и улучшения функциональности.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Сraftum – сайт для создания первичного дизайна: [Электронный ресурс.]. // craftum.com. URL: <https://craftum.com>. (Дата обращения: 02.06.2024).

2. Репозиторий проекта EduSec: [Электронный ресурс.]. // Github.com URL: <https://github.com/ShadowLance2/EduSec>. (Дата обращения 02.06.2024).

3. Развернутый сайт: [Электронный ресурс.]. // Github.com. URL: <https://shadowlance2.github.io/EduSec/>. (Дата обращения: 02.06.2024).