**Содержание**

[1.ВВЕДЕНИЕ 2](#_Toc107931644)

[2. ТЕМА ПРАКТИЧЕСКОЙ РАБОТЫ 2](#_Toc107931645)

[3. ЦЕЛЬ РАБОТЫ 2](#_Toc107931646)

[3.1 АКТУАЛЬНОСТЬ 3](#_Toc107931647)

[3.2. ПОСТАНОВКА ЗАДАЧ 3](#_Toc107931648)

[4. ОСНОВНАЯ ЧАСТЬ 4](#_Toc107931649)

[4.1. ИЗУЧЕНИЕ ТЕОРИТИЧЕСКОГО МАТЕРИАЛА 4](#_Toc107931650)

[4.1.1. ОСНОВЫ САЙТА. ЕГО ВИДЫ. 4](#_Toc107931651)

[4.1.2. ЯЗЫК РАЗМЕТКИ ГИПЕРТЕКСТА HTML 6](#_Toc107931652)

[4.1.3. ЯЗЫК ТАБЛИЦ СТИЛЕЙ CSS 8](#_Toc107931653)

[4.1.4. ЯЗЫК ПРОГРАММИРОВАНИЯ JAVASCRIPT. 9](#_Toc107931654)

[4.2. ПРАКТИЧЕСКАЯ ЧАСТЬ 11](#_Toc107931655)

[4.2.1. СОСТАВЛЕНИЕ МАКЕТА САЙТА, РАЗРАБОТКА ЛОГОТИПА. 11](#_Toc107931656)

[4.2.2. ВЕРСТКА ОТДЕЛЬНЫХ СТРАНИЦ ВЕБ-САЙТА. 12](#_Toc107931657)

[4.2.3. ПРОРАБОТКА СИСТЕМЫ НАВИГАЦИИ. 12](#_Toc107931658)

[4.2.4. ДЕМОНСТРАЦИЯ РАБОТЫ. 12](#_Toc107931659)

[ЗАКЛЮЧЕНИЕ 17](#_Toc107931660)

[СПИСОК ИСТОЧНИКОВ 18](#_Toc107931661)

[ПРИЛОЖЕНИЯ 19](#_Toc107931662)

# 1.ВВЕДЕНИЕ

С каждым днём во всемирной сети появляются всё больше сайтов, компаниям без веб-страниц тяжело конкурировать с теми, у кого они есть. Сайты помогают хранить и распространять необходимую информацию на огромные массы людей.

# 2. ТЕМА ПРАКТИЧЕСКОЙ РАБОТЫ

Темой практической работы является разработка сайта для магазина. Я решил, что это будет сайт для рыболовного магазина.

# 3. ЦЕЛЬ РАБОТЫ

Цель работы – сформировать навыки создания шаблона web-страницы с помощью HTML, научиться красиво оформлять отдельные блоки на странице с помощью CSS и научиться с помощью JavaScript реализовывать обрабатывать события.

# 3.1 АКТУАЛЬНОСТЬ

Актуальность создания сайта любого вида – донесение информации до широкого круга пользователей интернета. Веб-ресурс позволяет представить информацию о компании и ее товарах или услугах сжато и одновременно полноценно. Также сайт может сообщать о новостях фирмы, об изменениях в прайсе или режиме работы, содержать отзывы благодарных клиентов.

# 3.2. ПОСТАНОВКА ЗАДАЧ

Перед разработкой программного обеспечения были поставлены следующие задачи:

* Изучение теоретического материала:

1. Изучение основ построения сайта,
2. Изучение язык гипертекстовой разметки HTML,
3. Изучение языка таблиц стилей CSS;
4. Изучение основ языка JavaScript

* Составление макета сайта, разработка логотипа;
* Верстка отдельных страниц веб-сайта;
* Проработка системы навигации;
* Демонстрация работы.

# 4. ОСНОВНАЯ ЧАСТЬ

# 4.1. ИЗУЧЕНИЕ ТЕОРИТИЧЕСКОГО МАТЕРИАЛА

# 4.1.1. ОСНОВЫ САЙТА. ЕГО ВИДЫ.

Веб-разработка  — процесс создания [веб-сайта](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B0%D0%B9%D1%82) или [веб-приложения](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5). Основными этапами процесса являются [веб-дизайн](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B4%D0%B8%D0%B7%D0%B0%D0%B9%D0%BD), [вёрстка страниц](https://ru.wikipedia.org/wiki/%D0%92%D1%91%D1%80%D1%81%D1%82%D0%BA%D0%B0_%D0%B2%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86), [программирование](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) на стороне клиента и сервера, а также [конфигурирование](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BA%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [веб-сервера](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80).

Сайт — это совокупность веб-страниц, объединённых по смыслу и расположенных на одном сервере. Содержание и объем сайта зависит от того, какую задачу он должен выполнять.

Сайты можно разделить на следующие виды:

Сайт-визитка: Самый простой вид сайта. Как правило, содержит 1—5 страниц и ограничен функциональностью. Может быть использован для публикации краткой информации о чем-либо.

В данный момент менее популярен, поскольку за счет малой информативности вытесняется социальными сетями, которые в своем роде также являются визитками, но содержат гораздо больше полезных функций. Сайт визитка может быть полезен при интеграции с социальными сетями.

Пример использования сайта-визитки:

Создание уникального дизайна, публикация краткой информации + ссылки на социальные сети.

Одним из вариантов сайта-визитки является Лэндинговая страница. Такая разновидность сайта используется в электронной коммерции для рекламы и продвижения определенного продукта. Лэндинговая страница, как правило, представлена единственной страницей, где информация о товаре или услуге является ключевой и подана наиболее наглядным образом. Разработка Лэндинговой страницы ведется с участием маркетолога.

Сайт представительство: Сайт обладает большим, чем сайт-визитка, количеством страниц и функций: более подробное описание услуг, портфолио, отзывы, форма обратной связи, карта проезда и т. д. Набор страниц и функций утверждается, как правило, заранее и описывается в техническом задании на сайт.

Корпоративный сайт: Содержит наиболее полную информацию о компании ее услугах и предоставляемой продукции. Большое внимание уделяется публикации материалов из жизни компании: новости, заметки в СМИ, награды, сотрудники. Зачастую более мощный функциональные инструменты для работы с публикуемым содержимым (полнотекстовый поиск по сайту, карта сайта, календарь событий, корпоративные блоги, галереи). Корпоративный сайт может быть интегрирован с внутренними информационными системами компании (CRM, бухгалтерия). Может содержать закрытые разделы для разных групп пользователей — сотрудников, клиентов, дилеров, контрагентов и пр.

Каталог продукции: Каталог продукции — как правило, такой сайт не может существовать в независимости от сайта представительства или корпоративного сайта. Каталог предназначен для публикации более подробного описания товаров или услуг, технических характеристик, фотографий, сертификатов. Может содержать отзывы, полученные от клиентов. Такой сайт создается, если у заказчика есть потребность в более наглядной демонстрации существующего прайса.

Интернет-магазин: Интернет-магазин — сайт с каталогом продукции, с помощью которого клиент может заказать нужные ему товары. Интернет-магазины поддерживают интеграцию с распространенными бухгалтерскими программами (1C и др.), CRM, а также различные системы оплаты: пластиковые карты, электронные деньги, оплата через банк. Дополнительными функциями магазинов могут быть онлайн обработка заказов, управление закупками и доставкой, складской учет, работа с клиентской базой. Параллельно с продажей товара выстраивается система рекомендаций, отзывы посетителей о товарах и качестве обслуживания.

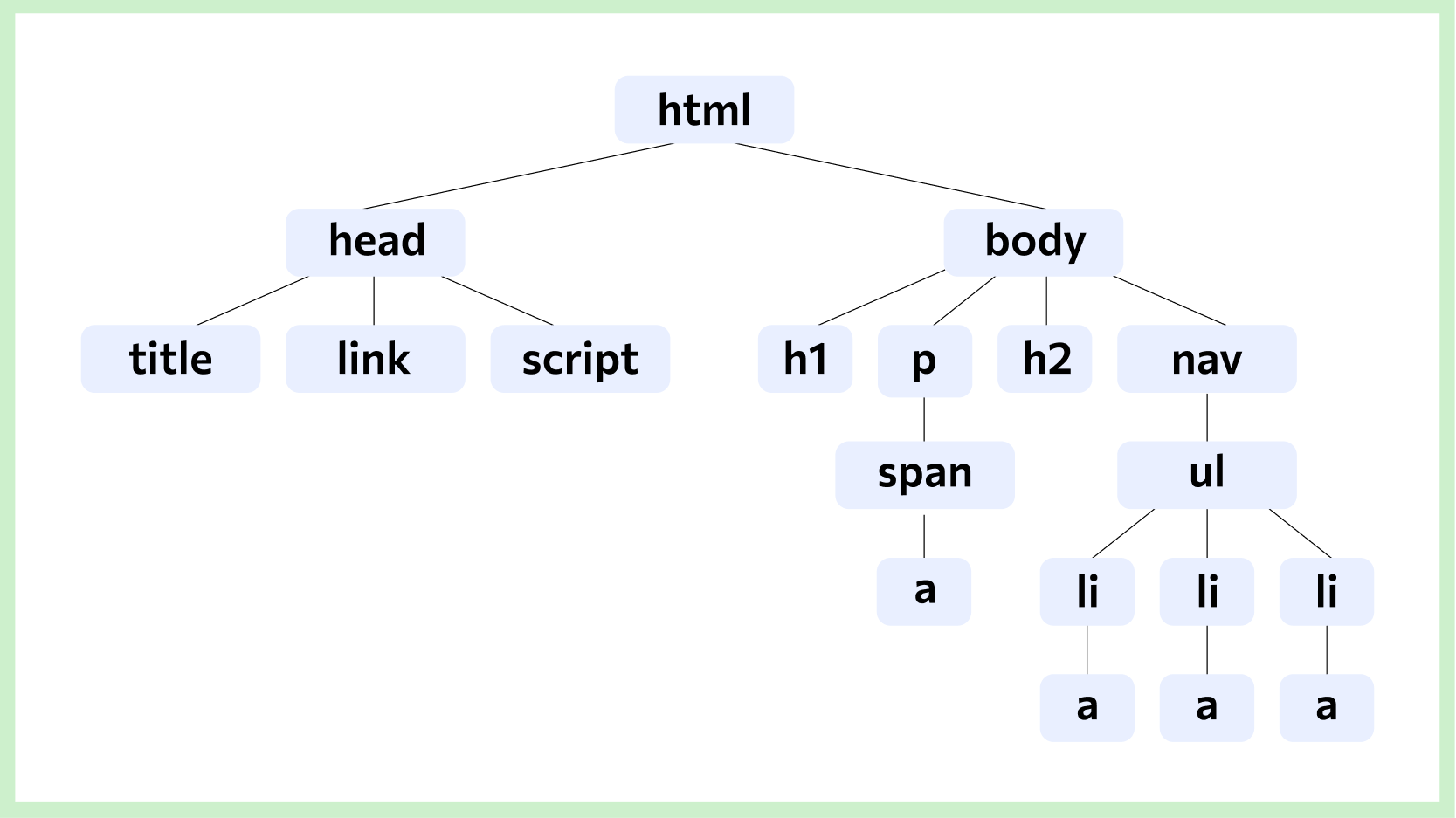
Информационный портал: Сайт, содержащий большой объем информации в виде текстовых статей, аудио- и видеоматериалов. Такие сайты, как правило, содержат умные фильтры для поиска содержимого, метки для записей, большое количество тематических разделов. Для обсуждения статей добавляются возможность комментирования. Регистрация и авторизация посетителей возможно с помощью аккаунта в социальных сетях.

# 4.1.2. ЯЗЫК РАЗМЕТКИ ГИПЕРТЕКСТА HTML

HTML — это язык разметки гипертекстовых документов. Он нужен, чтобы отображать в браузере специальным образом отформатированный документ с множеством вложенных элементов: заголовками, абзацами, списками, гиперссылками, медиаисточниками, расположением изображений, видео и аудио.

HTML-документ это текстовый файл с расширением .html или .htm. В браузере он преобразуется в веб-страницу и состоит из набора тегов. Они как раз и помогают представлять текст на экране: благодаря им браузер понимает, что он читает не просто текст, а структурированную информацию, разбитую на блоки.

Тег выглядит как набор символов, заключенный в угловые скобки. Символы в скобках обозначают имя тега, которое описывает его функции. Вот несколько примеров:

1. <h1> </h1> — заголовок;
2. <p> </p> — абзац;
3. <i> </i> — курсив.

Тег – это составной элемент, определяющий разметку структурных блоков. Он открывается, и этим начинает свое действие; и закрывается, обозначая завершение команды. Закрытые и открытые теги различаются только слешем перед именем тега. Эти теги создают оболочку, в которую помещается текст.

HTML не обрабатывает данные, а только их отображает. То есть с помощью него нельзя выполнить сложение или умножение, можно только показать текст, в котором будет содержаться нужная формула с ответом. Он отвечает за разметку – ограниченный набор действий, который помогает браузеру отображать страницы.

# 4.1.3. ЯЗЫК ТАБЛИЦ СТИЛЕЙ CSS

CSS — это язык описания внешнего вида документа, то есть он отвечает за то, как выглядят веб-страницы: цвет фона и декоративных элементов, размер и стиль шрифтов. Термин расшифровывается как Cascading Style Sheets (каскадные таблицы стилей). CSS взаимодействует с другим языком разметки — HTML, который отвечает за размещение элементов на странице.

Использование тегов для форматирования текста в HTML загромождает исходный код, усложняет его, а значит, вероятность допустить в нем ошибку становится выше. Для того чтобы этого избежать, создали отдельный язык для стилевой разметки — CSS. Кроме размера и цвета шрифтов, этот язык разметки регулирует разделение заголовков, подзаголовков и основного текста, размер полей и отступов, отдельные цветовые фреймы для выделения текста, цвет основного фона, шапки и подвала.

CSS, как и любой язык, имеет свой синтаксис. В нем есть правила — значения, которые определяют внешний вид элементов. CSS-правило состоит из селектора, CSS-свойств и их значений:

1. Селекторы — это метки, которые помогают браузеру понять, к какой части HTML-кода нужно применить заданные параметры.
2. CSS-свойства — это определенные параметры оформления, например цвет элемента или текста (color) или цвет фона (background).
3. Значение — это просто значение, оно выражается текстом или числом, например черный (black).

CSS-правила в коде заключаются в фигурные скобки {…}. Перед открытием скобки обязательно нужно указать селектор, к которому относится это правило.

# 4.1.4. ЯЗЫК ПРОГРАММИРОВАНИЯ JAVASCRIPT.

JavaScript — это язык программирования, который используют для написания frontend - и backend-частей сайтов, а также мобильных приложений. Часто в текстах и обучающих материалах название языка сокращают до JS. Это язык программирования высокого уровня, то есть код на нем понятный и хорошо читается. JS поддерживают все популярные браузеры.

*Для чего нужен JavaScript*

1. Взаимодействие с пользователем. Всплывающие окна с полем для ввода почты или других данных, уведомления об использовании cookie.
2. Обработка данных HTML. JS помогает заполнить необходимые поля в формах, проверять буквенные и числовые значения, а также управлять HTML-элементами на странице. Например, в конструкторе лендингов Tilda можно перетаскивать блоки и по клику менять шрифты, цвета текста, фонов и другие значения.
3. Анимация. Это не только движущиеся объекты на странице и кнопки, но и карусель в галерее или плавная прокрутка страницы. Например, Pinterest с помощью JavaScript создал эффект бесконечной ленты.
4. Математические вычисления. С помощью JS-скриптов на сайтах реализованы калькуляторы, например подсчет суммы заказа в корзине интернет-магазина.

*Как работает JavaScript в браузере:*

Действия пользователя на странице вызывают события: клик на кнопке запускает анимацию, захват мышкой и перемещение курсора двигают объект по странице. Каждое из этих действий выполняется с помощью скриптов, написанных на JavaScript. У них есть определенный алгоритм работы:

1. Пользователь совершает действие — нажимает на кнопку «Зарегистрироваться».
2. Браузер фиксирует выполнение определенных условий — клик мыши в конкретной области на странице. За реакцию на действие отвечает команда onclick, которая запускает прописанный JS-код.
3. Затем запускается JS-код, который прописан для этих условий, — анимация нажатия кнопки и открытие формы регистрации.
4. Страница полностью или частично обновляется — в зависимости от настроек форма может открыться в этой же или в соседней вкладке.

# 4.2. ПРАКТИЧЕСКАЯ ЧАСТЬ

# 4.2.1. СОСТАВЛЕНИЕ МАКЕТА САЙТА, РАЗРАБОТКА ЛОГОТИПА.

При составлении макета я начал изучать сайты с похожей тематикой, после определился с основными цветами для сайта и разделами.  
Определившись с тематикой сайта, решил придумать название и логотип. Первый варианты логотипа (Рис.1) и конечный вариант (Рис.2).

|  |
| --- |
| Рис. 1  C:\Users\akbul\AppData\Local\Microsoft\Windows\INetCache\Content.Word\logo2.png |
| Рис. 2 |

# 4.2.2. ВЕРСТКА ОТДЕЛЬНЫХ СТРАНИЦ ВЕБ-САЙТА.

Ранее я использовал среду разработки PyCharm она показалась мне удобной. Поэтому при написании сайта я выбрал её.

Помимо главной страницы были реализованы страница товаров, страница информации о компании, страница обратной связи и страницы авторизации и регистрации. Каждая страница включает разные функции.

Во время всего этапа разработки использовалась эта среда разработки.

# 4.2.3. ПРОРАБОТКА СИСТЕМЫ НАВИГАЦИИ.

Переход между страницами был использован через кнопки на шапке сайта. Переход страницу товаров реализован нажатием на карточку из блока категорий или нажатием на поиск. Кнопки авторизации и регистрации переносят нас на нужный блок, а верный ввод данных возвращает на главную страницу. Нажатие на логотип сайта переносит нас на главную страницу.

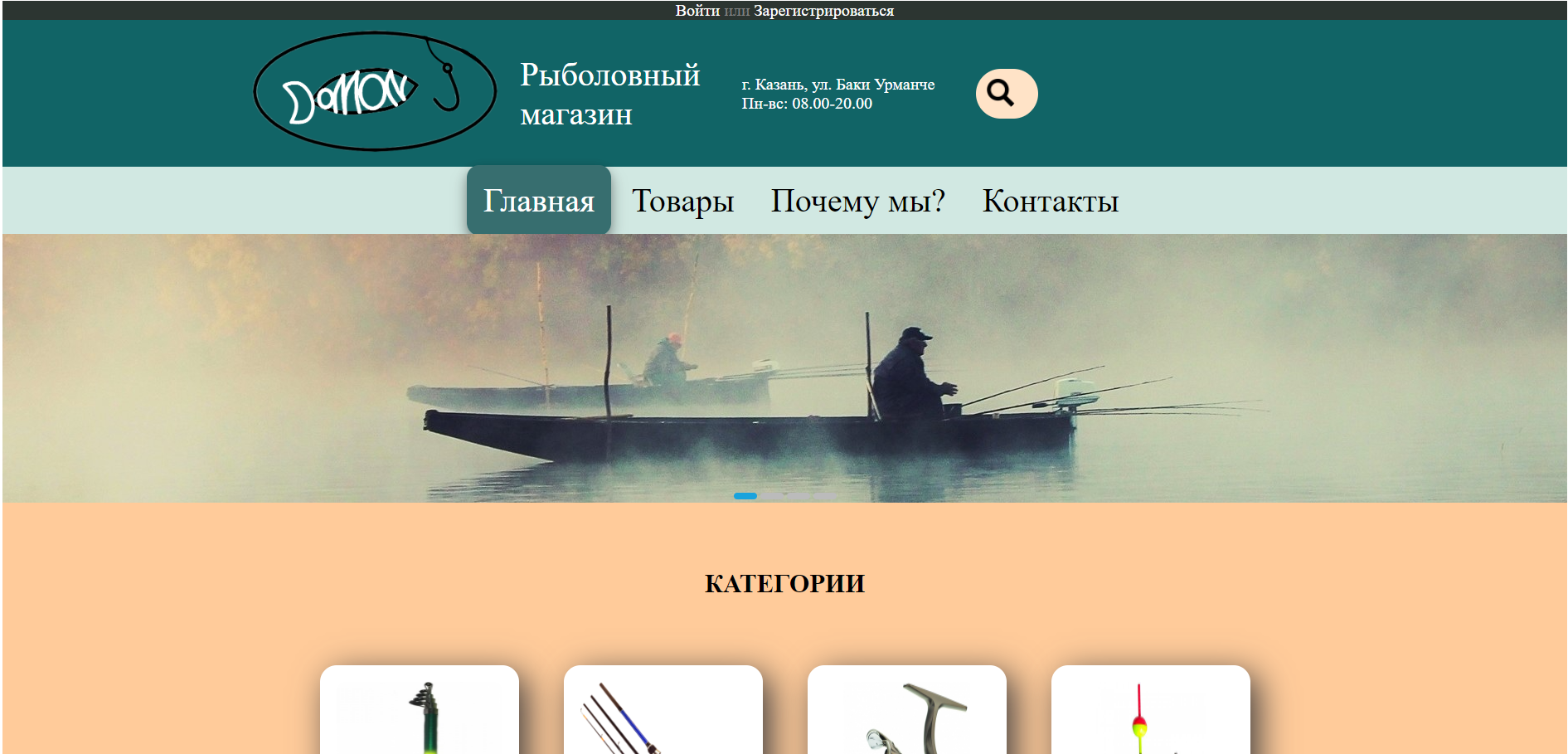
# 4.2.4. ДЕМОНСТРАЦИЯ РАБОТЫ.

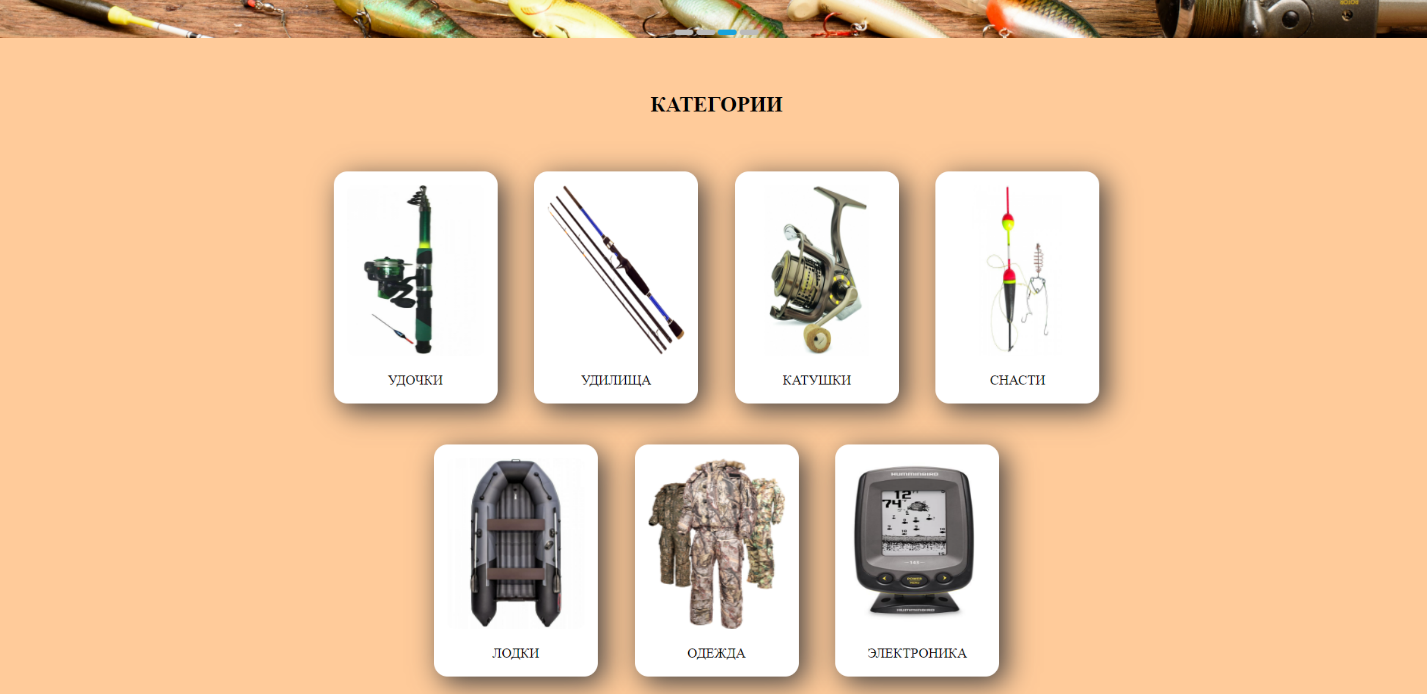
Сайт реализует функции:

* Авторизация/Регистрация
* Форма связи с администрацией сайта
* Слайдер с использованием чистого CSS (JavaScript реализует таймер)
* Переход к странице товаров через карточку или блок поиска
* Переключение между всеми вкладками через шапку сайта
* Интерактивные кнопки, карточки и формы

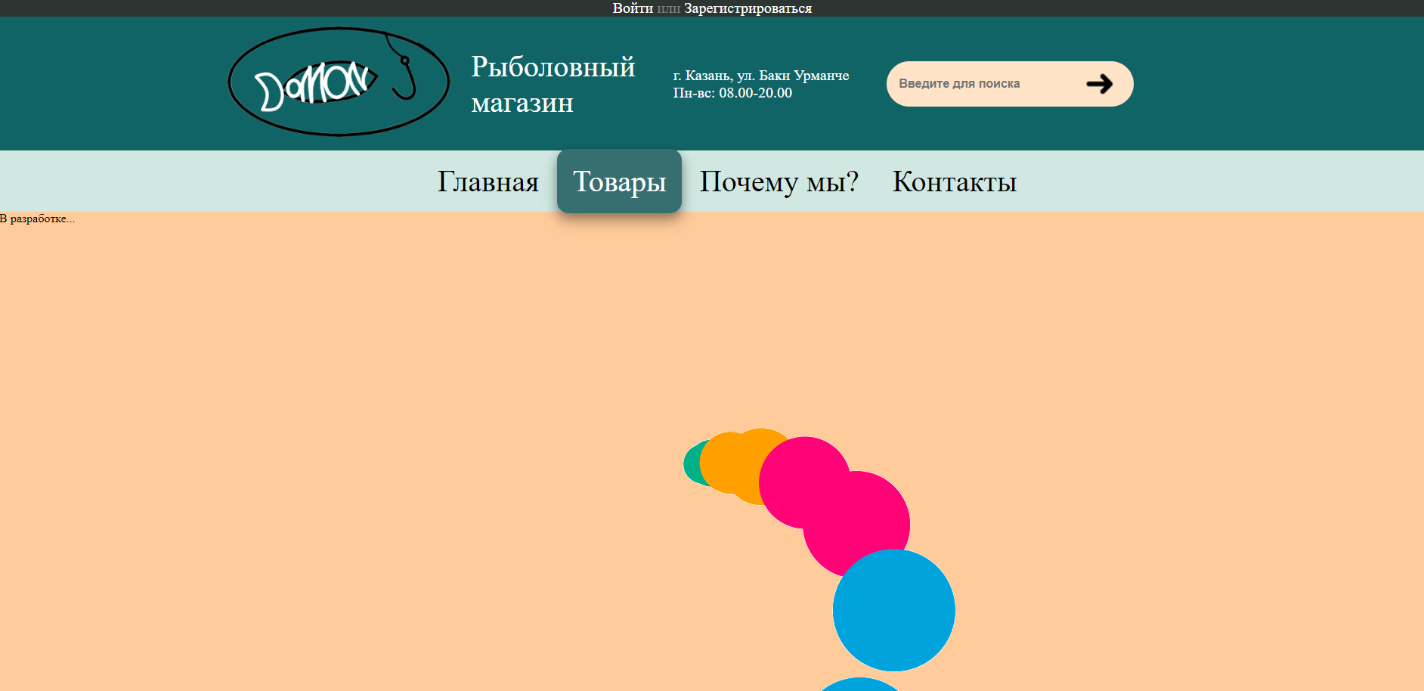
На данном сайте представлены:

- Основная страница. При нажатии на карточки категорий происходит переход на вкладку товаров. Слайдер автоматически меняет изображения каждые 5 секунд, также возможно переключение по кнопкам.



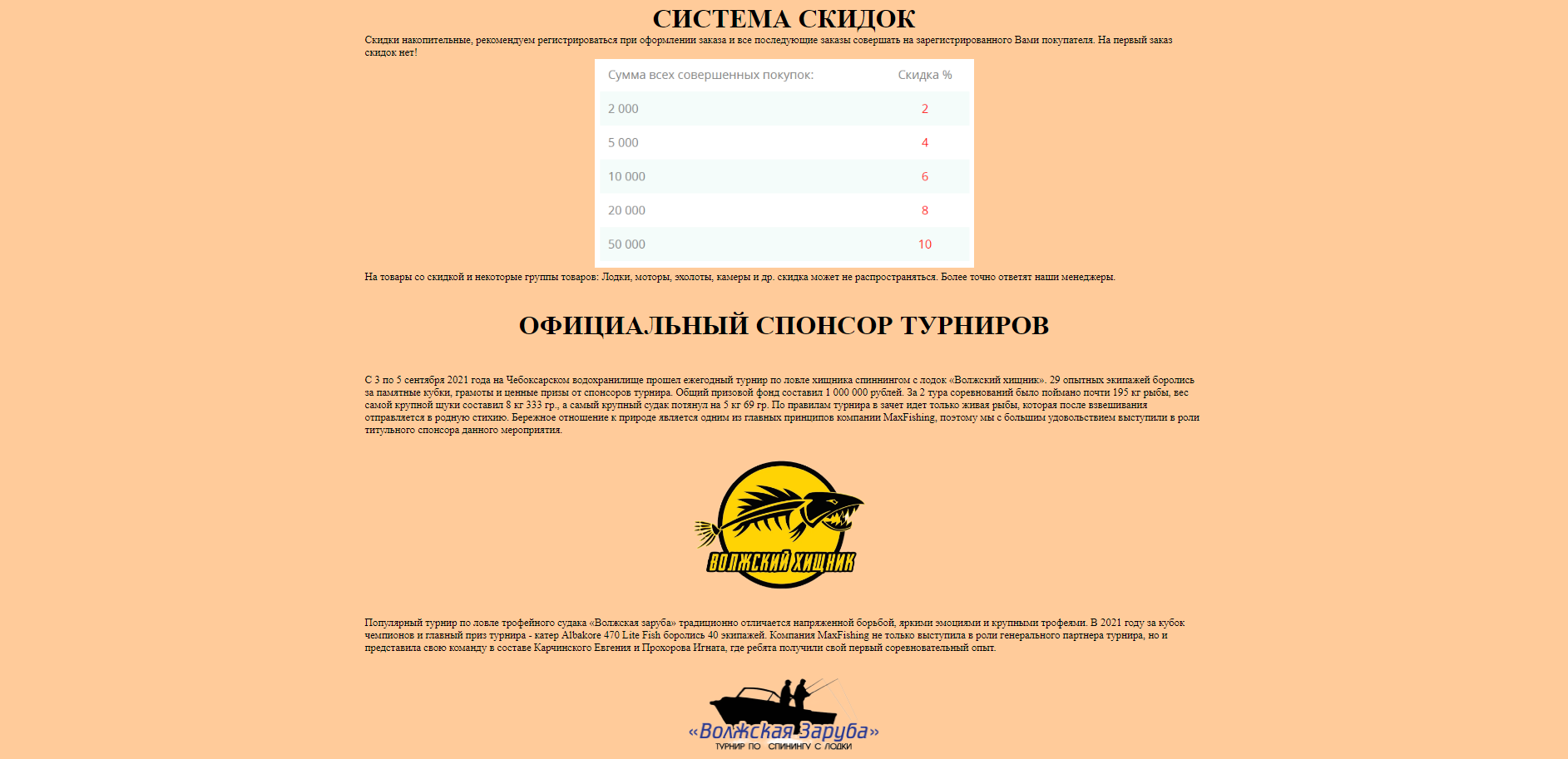


- Страница товаров.

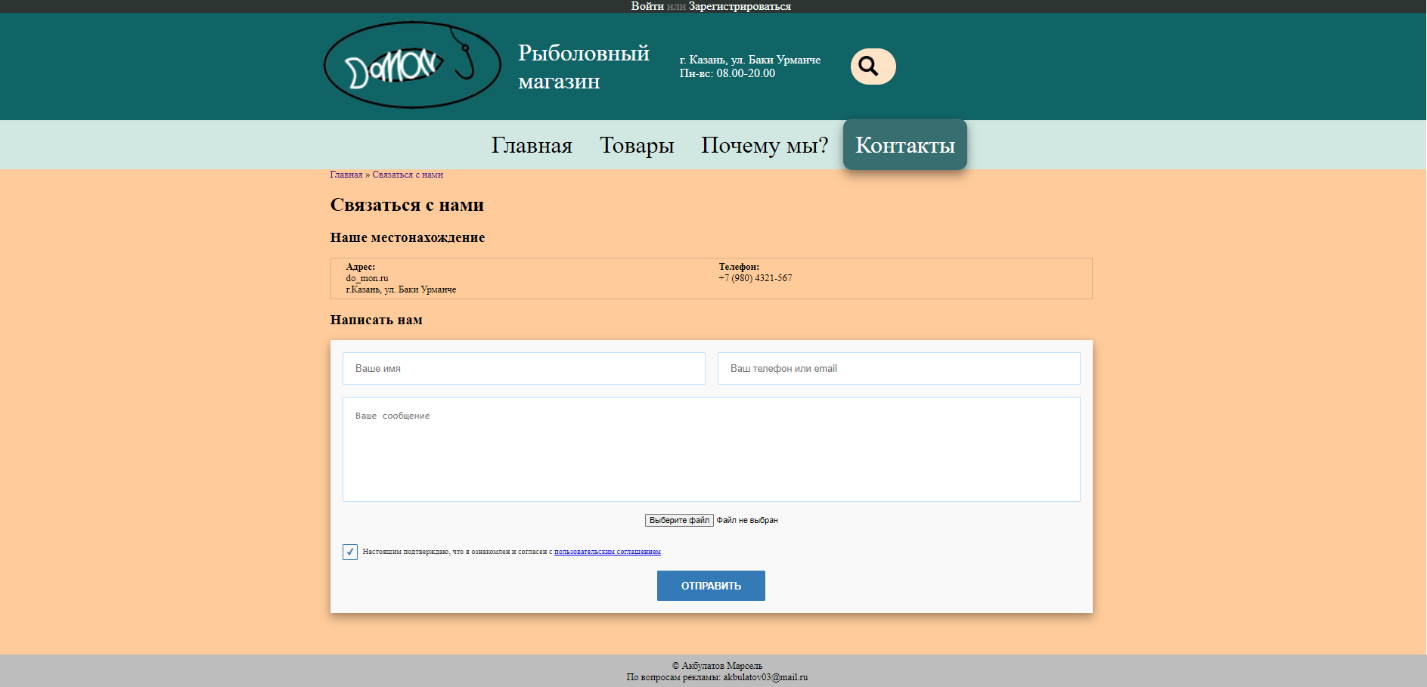


- Страница информации о компании.

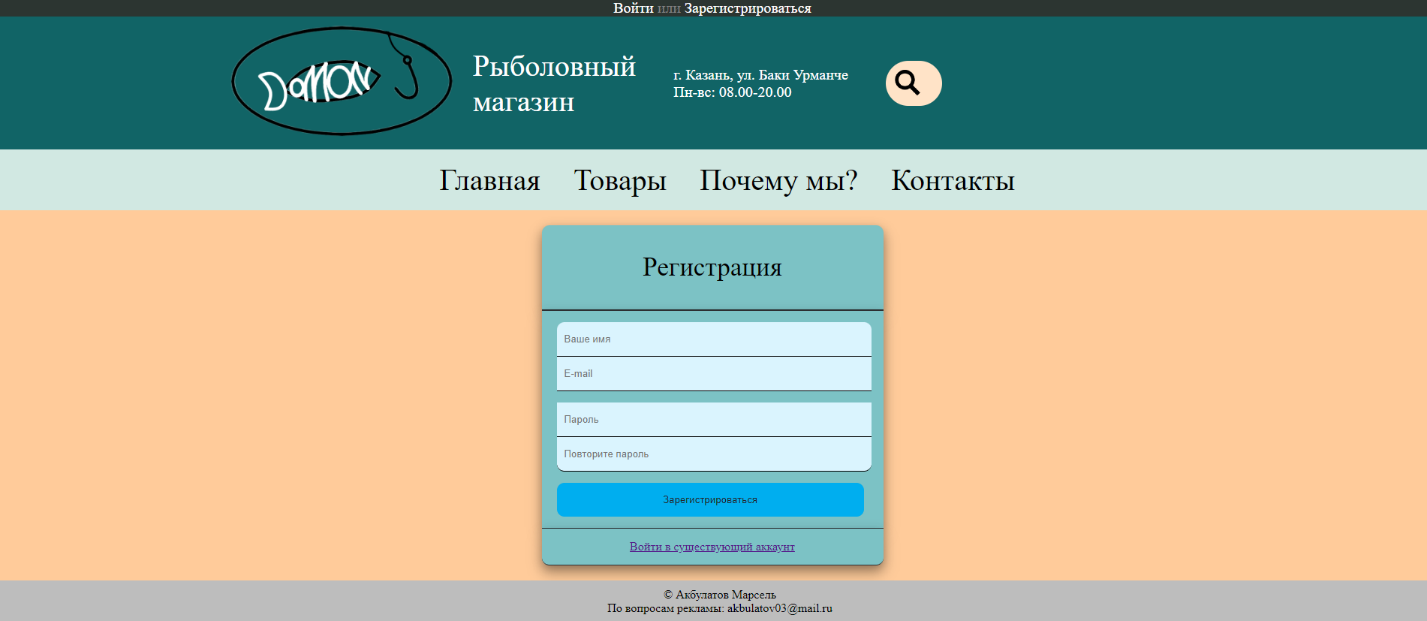




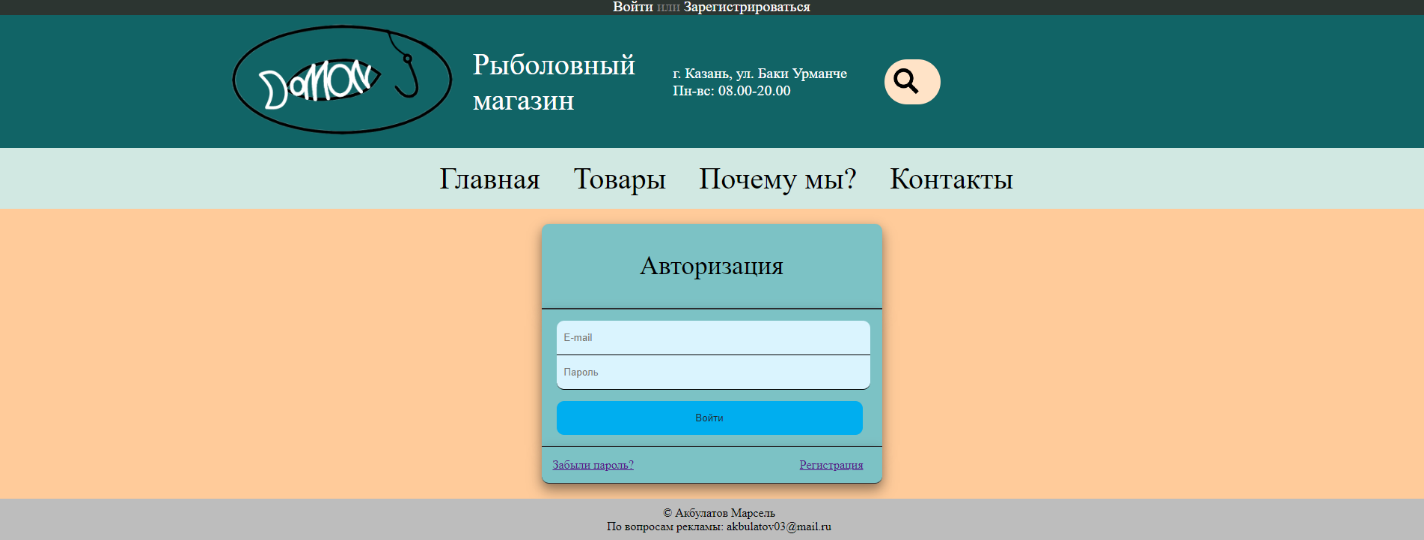
- Страница обратной связи. В случае неполного заполнения формы выводит сообщения с конкретной ошибкой.



- Страница регистрации. В случае неполного заполнения формы выводит сообщения с конкретной ошибкой. При правильном заполнении переходит на главную страницу.



- Страница авторизации. В случае неверного заполнения формы выводит сообщения с конкретной ошибкой. При правильном заполнении переходит на главную страницу.



# ЗАКЛЮЧЕНИЕ

Основная цель - создание сайта для магазина успешно выполнена. В процессе верстки и программирования сайта была реализованы 5 полноценных страниц и 1 одна страница с товарами с залогом на будущее. Задачи, поставленные перед выполнением учебной практики, стали достигнуты.

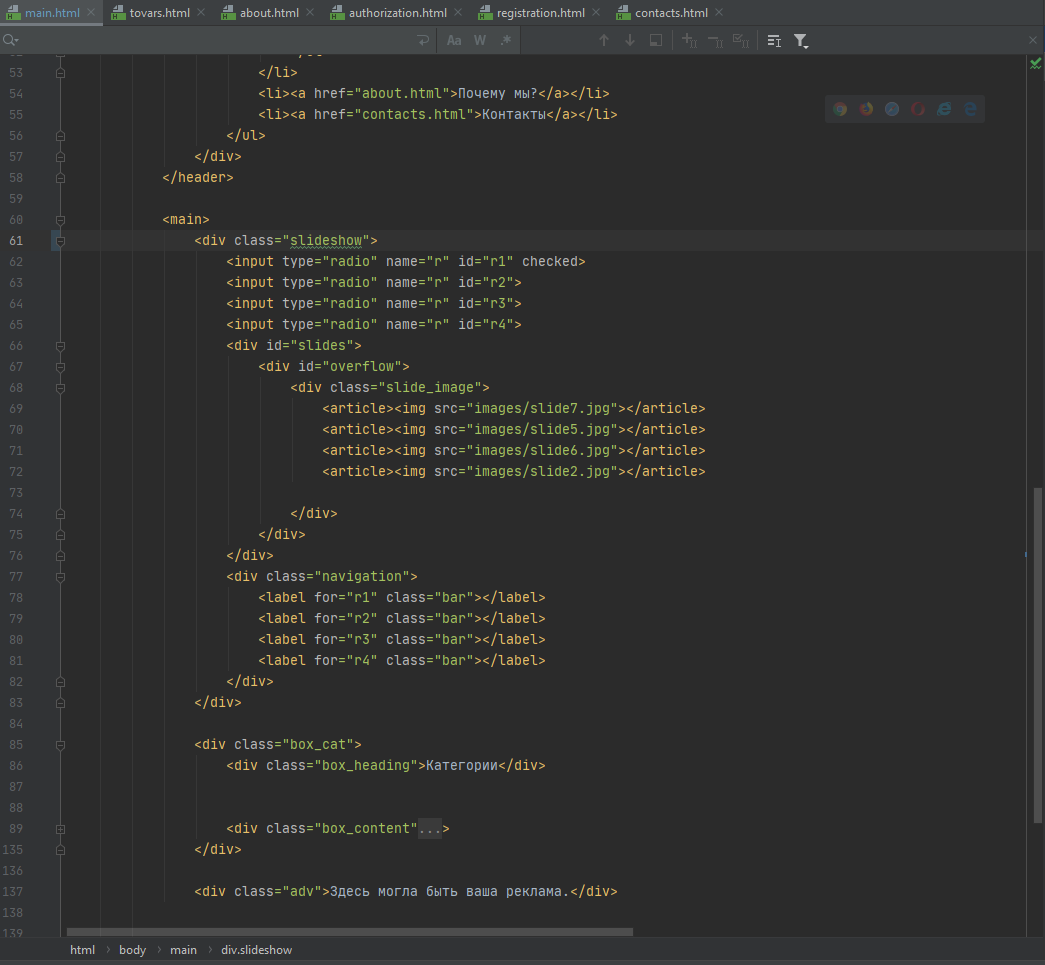
В ходе выполнения данного проекта было получена множество практических навыков. Я научился пользоваться языком гипертекстовой разметки, научился применять CSS к веб-проектам. И изучил основы языка программирования JavaScript.

# СПИСОК ИСТОЧНИКОВ

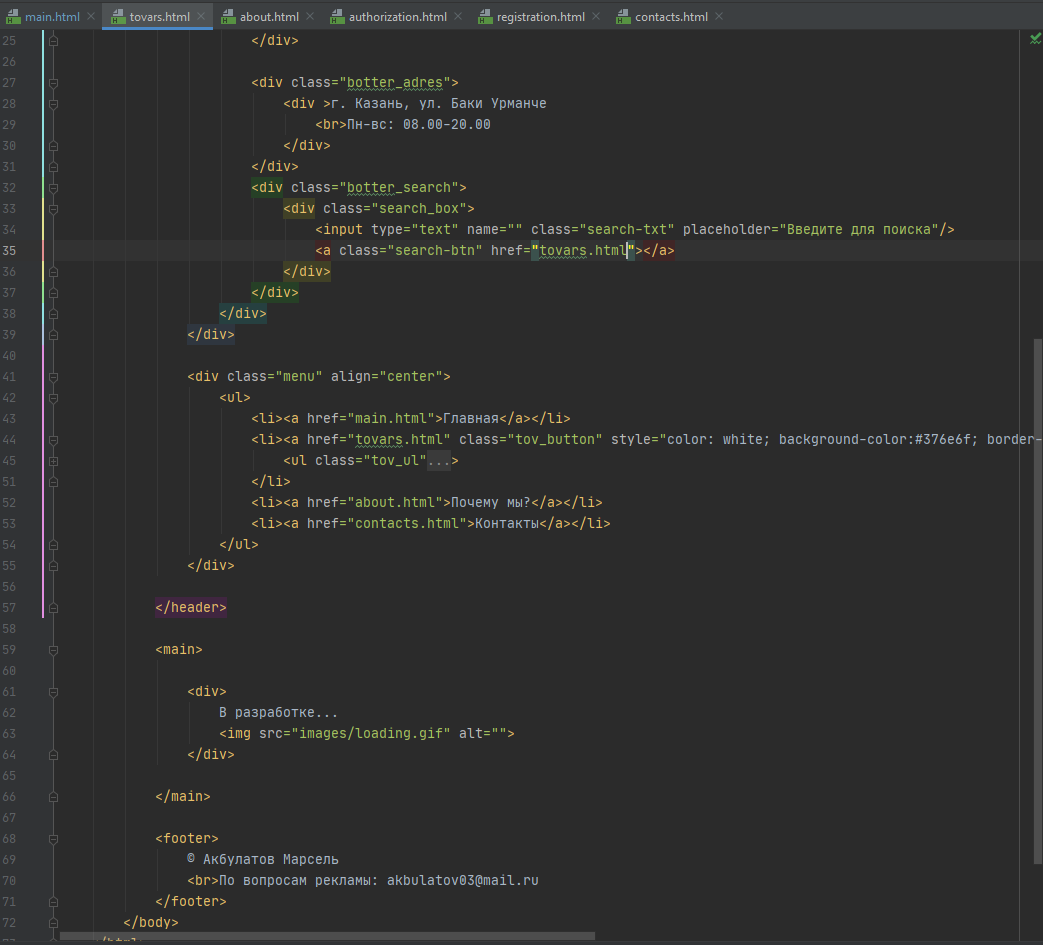
1. CSS Документация и справочники [https://developer.mozilla.org/ru/docs/Web/CSS/]
2. Как сделать слайдер для сайта на HTML+CSS [https://www.youtube.com/watch?v=gVxyTOPPEPc]
3. JavaScript Решает. 11 Решаем задачи про логин и пароль [https://www.youtube.com/watch?v=PtXYtNyZEJU&t=1s]
4. Подборка цветов [https://rusability.ru/pfanshtil/tsveta-dlya-saita-50-idei-ot-luchshih-dizainerov/5fd296642dda593c3483ebd2]
5. Барабан цветов [https://colorscheme.ru/]
6. Теория по JavaScript [https://blog.skillfactory.ru/glossary/javascript/]
7. Теория по CSS [https://blog.skillfactory.ru/glossary/CSS/]
8. Теория по HTML [https://blog.skillfactory.ru/glossary/HTML/]
9. О видах сайтов [http://design.drevolife.ru/blog/obshhie-ponyatiya-o-sayte-vidyi-saytov-domen-i-hosting-etapyi-razrabotki-sayta-posleduyushhee-obsluzhivanie.html]

# ПРИЛОЖЕНИЯ

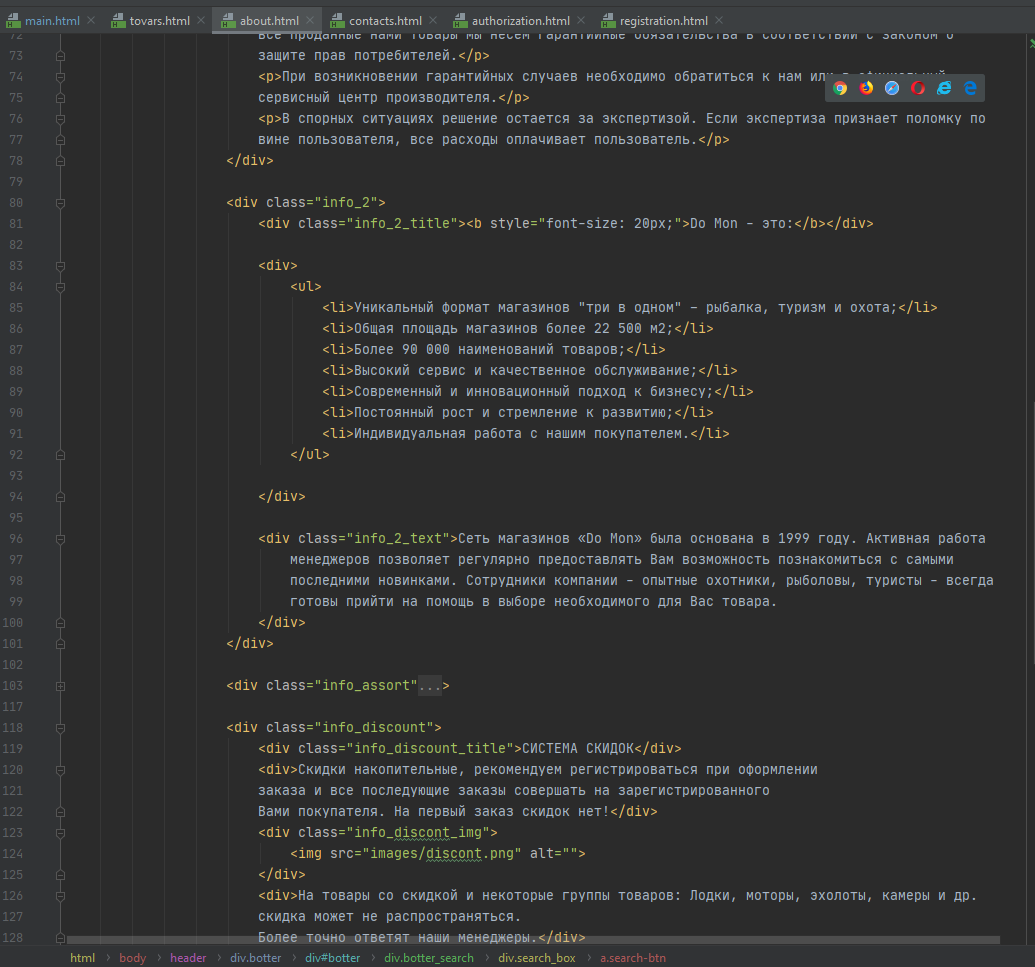
Код 1 (Компонент – Главная страница)



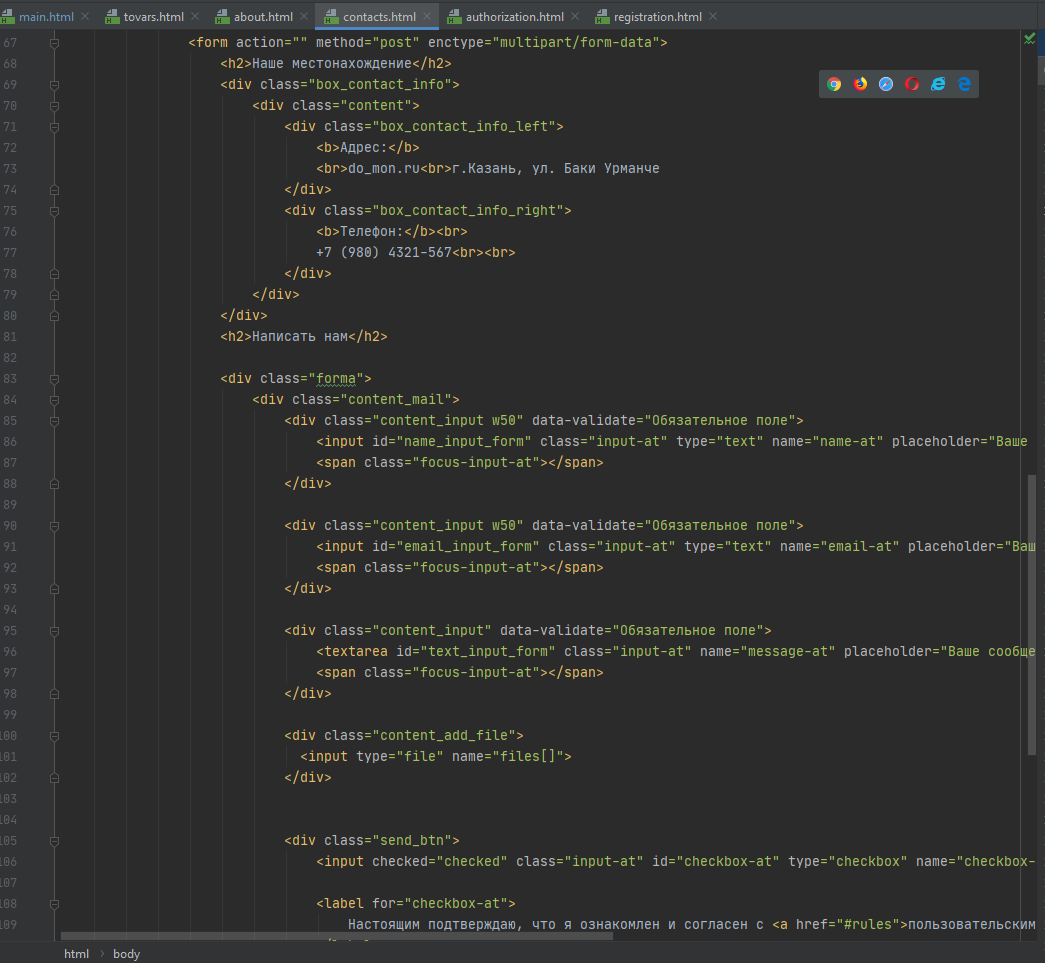
Код 2 (Компонент - Товары)



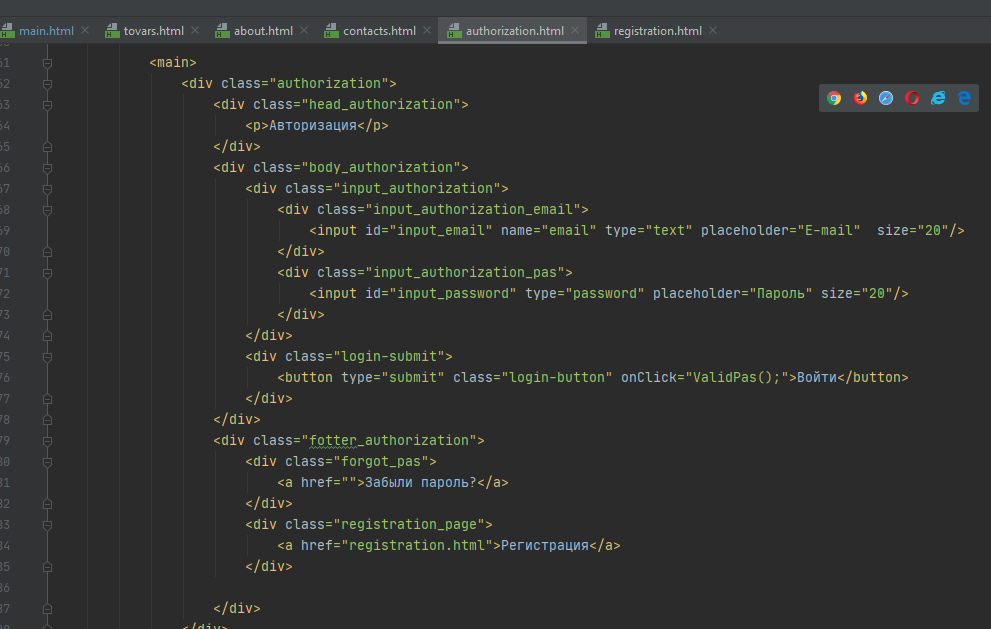
Код 3 (Компонент – Почему мы)



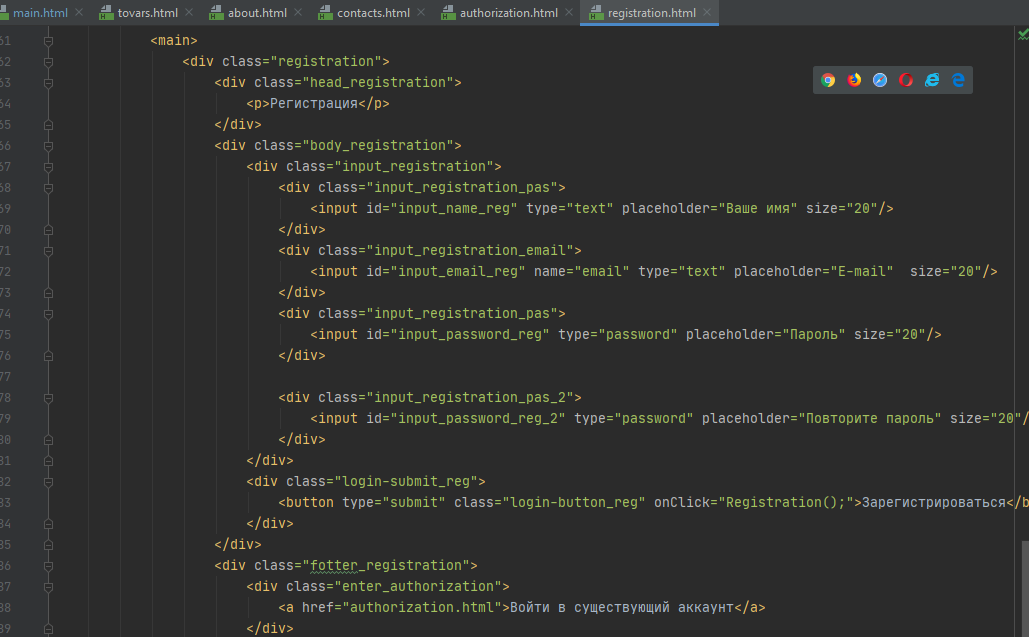
Код 4 (Компонент – Формы обратной связи)



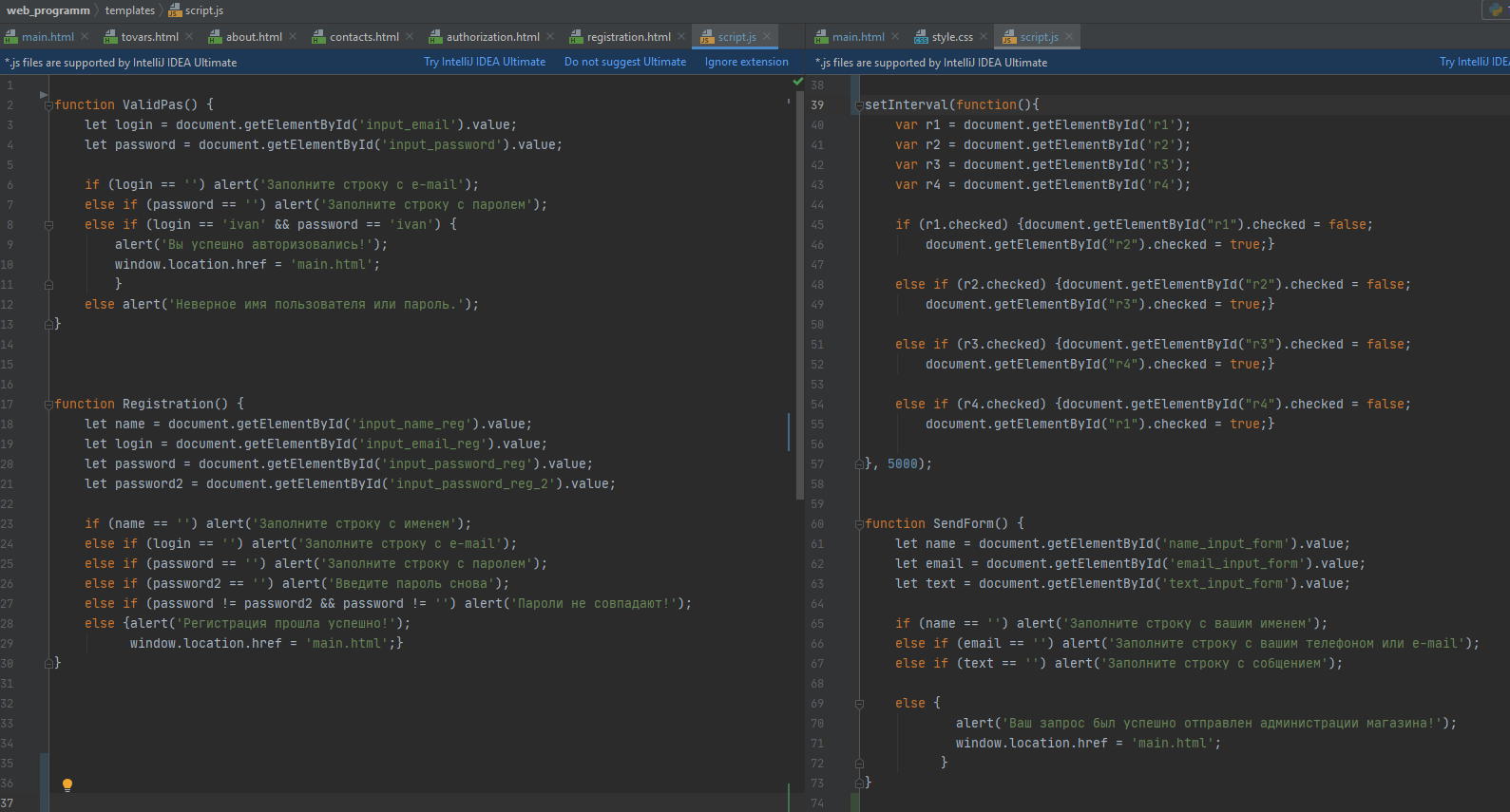
Код 5 (Компонент – Авторизация)



Код 6 (Компонент – Регистрации)



Код 7 (Функции JavaScript)



Код 8 (Комонент CSS)

