

СОДЕРЖАНИЕ

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	2
1 АНАЛИТИЧЕСКАЯ ЧАСТЬ	3
1.1 Анализ конкурентных решений	3
1.2 Постановка задачи	4
Выводы по разделу 1	5
2 ПЛАНИРОВАНИЕ	6
2.1 Выбор программного обеспечения	6
2.2 Архитектура приложения	7
2.3 Обоснование языка программирования	8
Вывод по разделу 2.....	9
3 РАЗРАБОТКА РЕШЕНИЯ.....	10
Вывод по разделу 3.....	26
ЗАКЛЮЧЕНИЕ	27
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	28

ВВЕДЕНИЕ

Данная расчетно-графическая работа была создана с целью разработки полноценного веб-сайта, используя архитектуру MVC. Проект состоит из двух частей – серверной и пользовательской.

В основной части работы будут рассмотрены архитектура Model-View-Controller (MVC), маршрутизация, работа с базой данных и другие важные элементы разработки веб-приложений в фреймворке Laravel со стороны бекенда и разработка пользовательского интерфейса с использованием HTML, CSS и JavaScript со стороны фронтенда.

В завершении работы будет проведен анализ итогового веб-приложения и рассмотрено дальнейшее его развитие.

Целью данного курсового проекта является приобретение практических навыков разработки веб-приложений используя архитектуру Model-View-Controller.

1 АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1 Анализ конкурентных решений

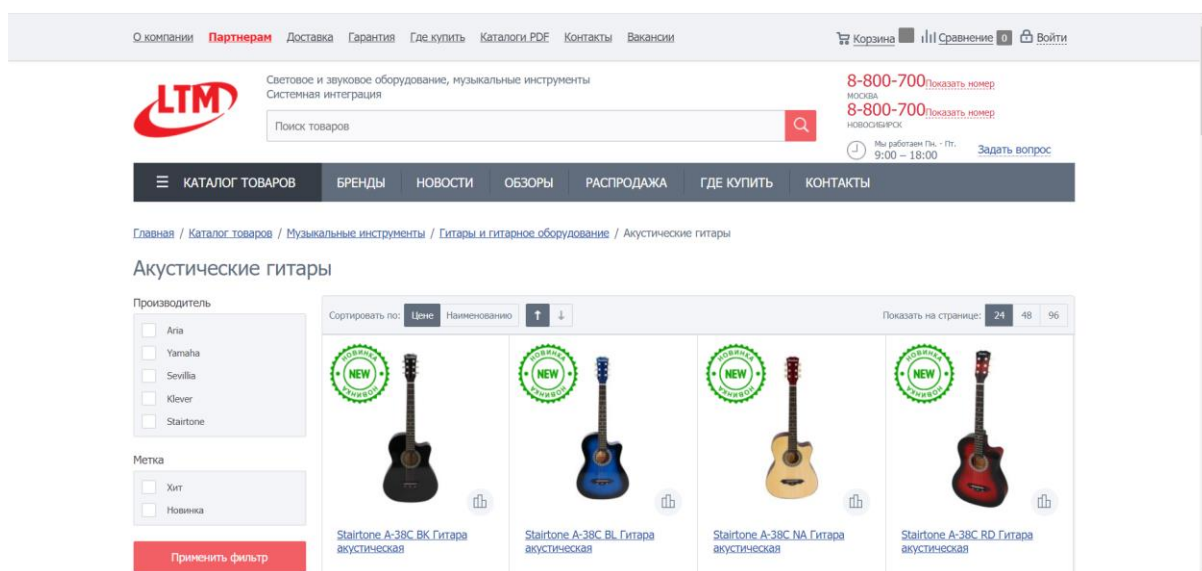


Рисунок 1.1 – Сайт LTM Music

LTM Music (<https://ltm-music.ru/>) – 20 лет компания является дистрибьютором звукового, светового и шоу-оборудования, музыкальных инструментов на российский рынок, также имеет собственное производство и сотрудничает с 80 компаниями-поставщиками музыкального оборудования, а также имеет сеть из 13 магазинов по стране. На их сайте можно найти и приобрести музыкальные инструменты и оборудования для концертов на любой вкус.

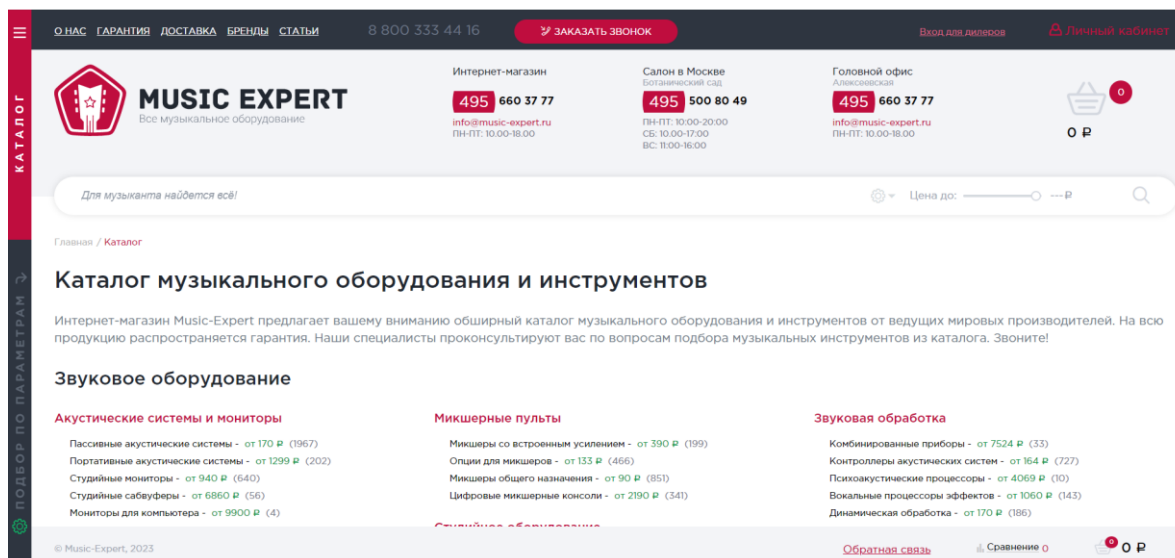


Рисунок 1.2 – Сайт Music Expert

Music Expert (<https://www.music-expert.ru/>) – работает на рынке музыкального оборудования уже 12 лет. Основным их преимуществом является наличие широкого ассортимента востребованного покупателями оборудования в наличии на складах, а также низкая стоимость.

1.2 Постановка задачи

Целью работы является разработка веб-приложения для продажи музыкальных инструментов под архитектуру MVC. Сайт должен предоставлять функциональность аутентификации администратора, после которой он сможет добавлять новые товары, акции, новости, а также просматривать заявки.

Реализовать функцию авторизации администраторов с помощью логина и пароль, создав страницу входа. Обеспечить защиту административной части сайта от несанкционированного доступа.

Разработать административную панель, позволяющую добавлять, новости о музыке и музыкальных инструментах, а также новые товары, которые пользователь может добавить в корзину и указать в заявке. Предусмотреть поля для заголовка, описания, картинки, а также даты создания новости. Добавить возможность форматирования содержания: изменение размера шрифта, толщины шрифта, добавление подчеркнутости, центрирования, добавление заголовков.

Создать страницу новостей, на которой будут отображаться все новости, а также создать страницу с товарами, в которой можно будет указать товар для добавления в корзину, также создать страницу корзины, в которой можно будет оформить заказ. Также создать страницу обратной связи и технической поддержки, а также встроить карту, на которой будет отображаться местоположение магазина.

Также предусмотреть возможность смены языка сайта, а также сохранение языка при переходе между страницами.

Выводы по разделу 1

В ходе анализа было выявлена задача создать веб-приложение, которое позволяло бы пользователям просматривать новости, товары и акции, а также возможность сделать заказ и оставить отзыв о приложении, а для администратора следует сделать возможность создавать новости, товары и акции, а также просматривать заказы и отзывы.

2 ПЛАНИРОВАНИЕ

2.1 Выбор программного обеспечения

В качестве редактора кода и по совместительству среды разработки было выбрано ПО Visual Studio Code, т.к. в нем присутствует подсветка синтаксиса, быстрый поиск по проекту а также статический анализатор для того чтобы увидеть программную ошибку до её запуска. Также в Visual Studio Code существует огромное количество расширений, которые позволяют ускорить разработку, а также поддержка работы с GitHub.

Для тестирования работы сайта были выбраны следующие три браузера – Opera, Yandex browser и Microsoft Edge.

В качестве системы контроля версий был использован веб-репозиторий GitHub.com и приложение GitHub Desktop для упрощения командной работы, а также для возможности откатить проект до предыдущей версии.

Для разворачивания локального сервера была использована программа Open Server Panel, созданная для того чтобы помочь веб-разработчикам отладке и тестировании сайтов непосредственно на компьютере.

В качестве системы управления базами данных для проверки корректности запросов к базе данных был использован PHPMyAdmin из-за следующих преимуществ: с помощью phpMyAdmin имеется возможность создавать, просматривать и удалять как базы данные, так и содержимое: таблицы, записи; все работы по созданию или удалению баз данных

выполняются интуитивно понятно даже для новичков; имеется простой и понятный поиск; Все работы подтверждаются сообщениями о выполнении.

Так как в разработке приложения был использован фреймворк Laravel был использован менеджер управления зависимостями приложений PHP – Composer.

Для связи в реальном времени между разработчиками и мгновенного согласования решений была использована кроссплатформенная система мгновенного обмена сообщениями с поддержкой VoIP – Discord.

2.2 Архитектура приложения

Для разработки приложения использовалась архитектура MVC. Она состоит из трех основных компонентов это модель, представление и контроллер.

При отправке HTTP запроса с помощью провайдера запрос переходит в роутер, который в свою очередь обращается к контроллеру – первому компоненту MVC который реагирует на HTTP запрос. Основная задача контроллера – выполнение бизнес логики обращаясь к моделям и возвращение представления с полученными из модели данными.

Модель представляет собой совокупность данных и ограничений/правил при работе с данными она также отвечает за работу с базой данных, при помощи паттерна ActiveRecord мы можем взаимодействовать с моделью так как будто это запись в таблице, изменяя её мы изменяем строку таблицы внутри базы данных. Также модель может

оставаться неизменной, но при этом позволять различные способы отображения данных что является ещё одним преимуществом архитектуры MVC.

Последний компонент - представление предоставляет различные способы отображения данных, полученных из модели.

2.3 Обоснование языка программирования

Так как задачей стоит разработка веб приложения то среди языков разработки обязательно присутствуют HTML, CSS и JS. JavaScript может: добавлять новый HTML-код на страницу, изменять существующее содержимое, модифицировать стили, реагировать на действия пользователя, щелчки мыши, перемещения указателя, нажатия клавиш, отправлять сетевые запросы на удалённые сервера, скачивать и загружать файлы (технологии AJAX и COMET), получать и устанавливать куки, задавать вопросы посетителю, показывать сообщения, запоминать данные на стороне клиента («local storage»). HTML использовался в чистом виде без препроцессоров, т.к. проект не подразумевает большого количества однотипной верстки.

Для языка CSS был выбран препроцессор SCSS так как он позволяет использовать переменные, вложенность и операторы, а также позволяет использовать модули, что упрощает организацию кода и его изменение в случае смены требований к дизайну. [8] Для доступа к Web API был использован JS в чистом виде, потому что проект не требует сложной архитектуры и взаимодействия множества компонентов системы. Для организации бэкенда был использован язык PHP, т.к. он с открытым

исходным кодом и это позволяет использовать его бесплатно и без лицензии.

Язык PHP активно поддерживается, что значительно упростит дальнейшее сопровождение программного продукта, также он легок в изучении, а в открытом доступе есть качественная документация по встроенному функционалу языка. Также данный язык идеально подходит под данную задачу так как PHP изначально был оптимизирован под быстрое создание веб-приложений.

Для запросов к базе данных был выбран язык MySQL, так как он обладает высокой скоростью и простым интерфейсом за счет отказа от некоторых стандартов языка.

Вывод по разделу 2

В ходе планирования были выявлены необходимые, а также упрощающие разработку программные продукты и языки программирования и разметки. Из-за специфики веб-разработки были использованы следующие три языка HTML, JS и CSS, а также PHP так как он изначально создан для создания веб-приложений и имеет огромное количество встроенных инструментов для работы с базами данных. Языком взаимодействия с базами данных был MySQL так как он имеет упрощенный синтаксис и высокую скорость. В качестве программных продуктов основным был редактор кода Visual Studio Code, а также Composer для сборки Laravel проекта.

3 РАЗРАБОТКА РЕШЕНИЯ

3.1 Разработка UI/UX

Для начала разработки приложения была собрана команда разработчиков и распределены роли:

Гальцов Денив. Фронтенд разработчик, дизайнер;

Кудин Виктор. Бэкенд разработчик, также ответственный за постановку задач;

Совместная разработка происходила с помощью гитхаб репозитория:
<https://github.com/AlfaR0meo/rgr-web>.

Шрифтом веб-приложения является Roboto Serif. На сайте используются следующие цвета:

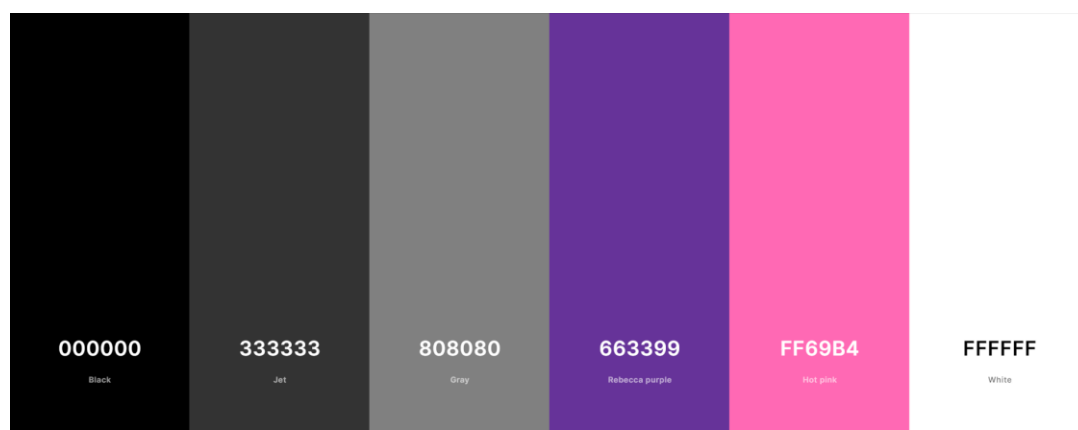


Рисунок 3.1 – Цвета, используемые в проекте

При переходе на сайт первое что видит пользователь это главная страница, она содержит в себе краткое описание предназначения сайта, а также историю компании

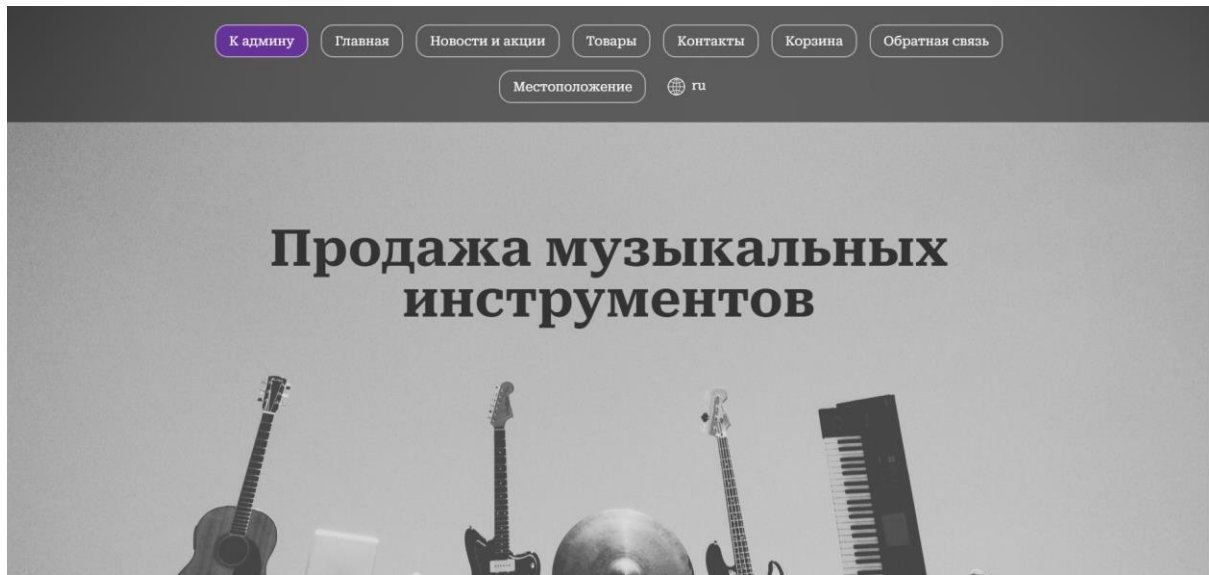


Рисунок 3.2 – Главная страница сайта

На странице новостей и акций отображаются все имеющиеся новости и акции.

Разбитую гитару Кобейна продали на американском аукционе почти за \$600 тыс.



Гитара, которую разбил создатель рок-группы Nirvana Курт Кобейн, ушла с молотка почти за \$600 тыс. Об этом в субботу, 20 мая, сообщ...

2023-04-12 00:12:30

Рисунок 3.3 – Блок с новостью

Горячая Распродажа, электрическая гитара высокого качества



Корпус из липы, кленовая шейка, под заказ, 6-струнная.

2023-06-17 00:00:00

Рисунок 3.4 – Блок с Акцией

На странице товаров отображаются все имеющиеся товары.

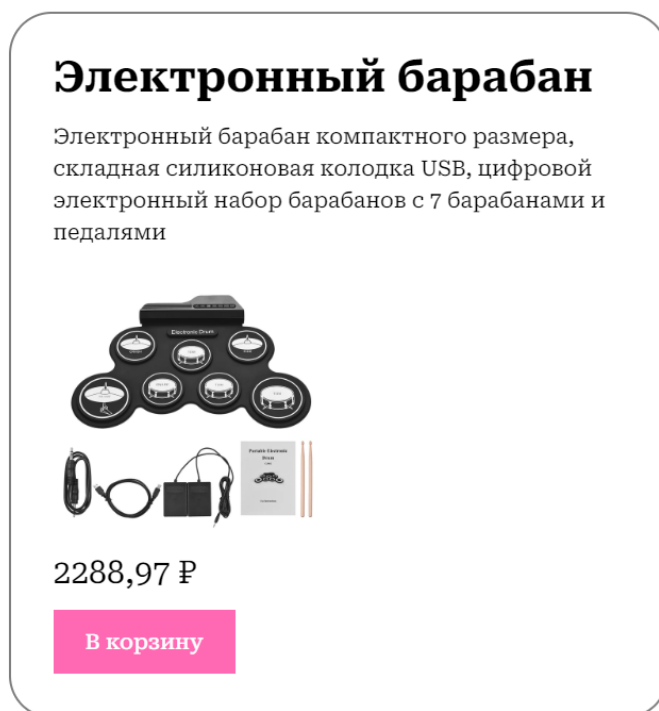


Рисунок 3.5 – Пример товара

На странице контакты будут отображены различные контактные данные для связи с администрацией.

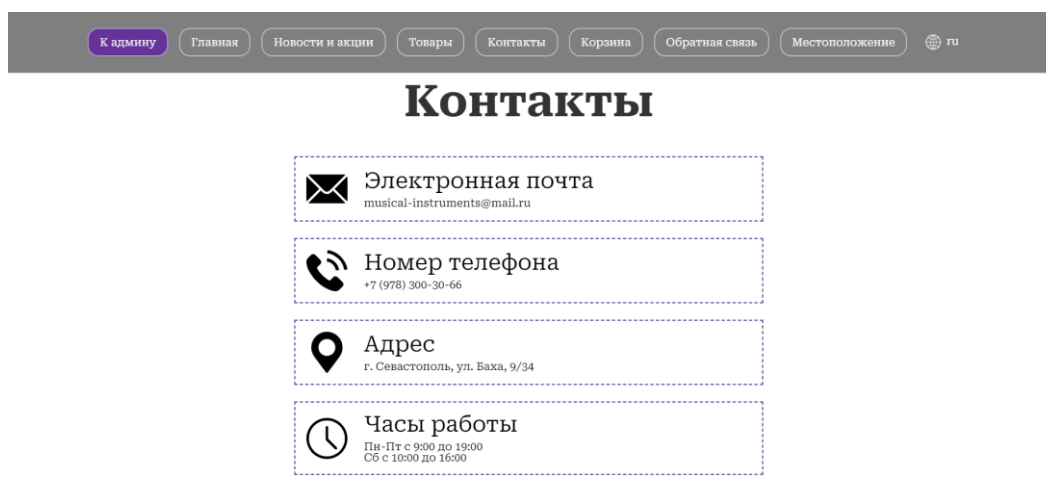


Рисунок 3.6 – Страница контактов

На странице корзины будут отображены выбранные товары а также форма для заказа, также там можно будет изменить количество товара и увидеть итоговую стоимость.



Рисунок 3.7 – Карточка товара на странице корзины

Delivery method

Choose delivery method ▾

Payment method

Choose payment method ▾

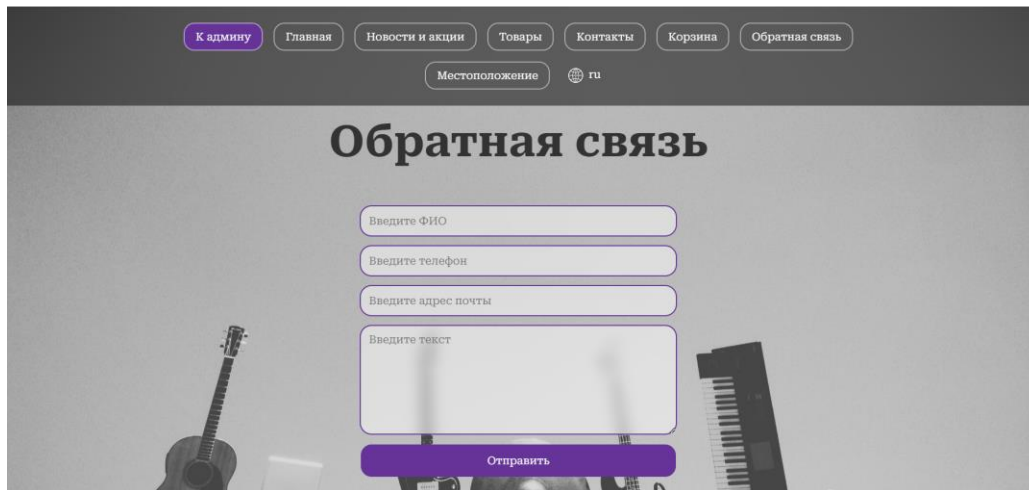
Fullname

Phone number

Make order

Рисунок 3.8 – Форма отправки заказа

На странице обратной связи находится форма для отправки отзыва.



The screenshot shows a web page titled 'Обратная связь' (Feedback). At the top, there is a navigation bar with links: 'К админу' (To admin), 'Главная' (Home), 'Новости и акции' (News and offers), 'Товары' (Products), 'Контакты' (Contacts), 'Корзина' (Cart), and 'Обратная связь' (Feedback). Below the navigation bar, there is a form with four input fields: 'Введите ФИО' (Enter FIO), 'Введите телефон' (Enter phone), 'Введите адрес почты' (Enter email address), and 'Введите текст' (Enter text). A purple button labeled 'Отправить' (Send) is located at the bottom of the form. The background of the page features a blurred image of musical instruments, including a guitar and a keyboard.

Рисунок 3.9 – Форма обратной связи

На странице местоположения находится карта где указано место магазина музыкальных инструментов.

Location

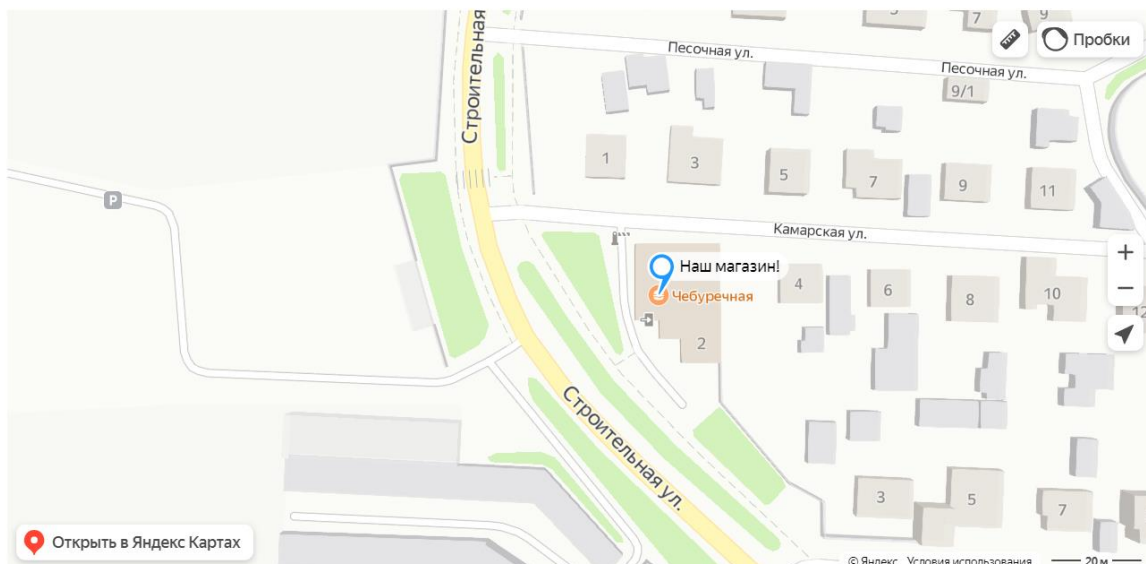


Рисунок 3.10 – Страница местоположения

На странице авторизации находится форма для заполнения логина и пароля заполнив которую можно авторизоваться и получить доступ к страницам администратора.

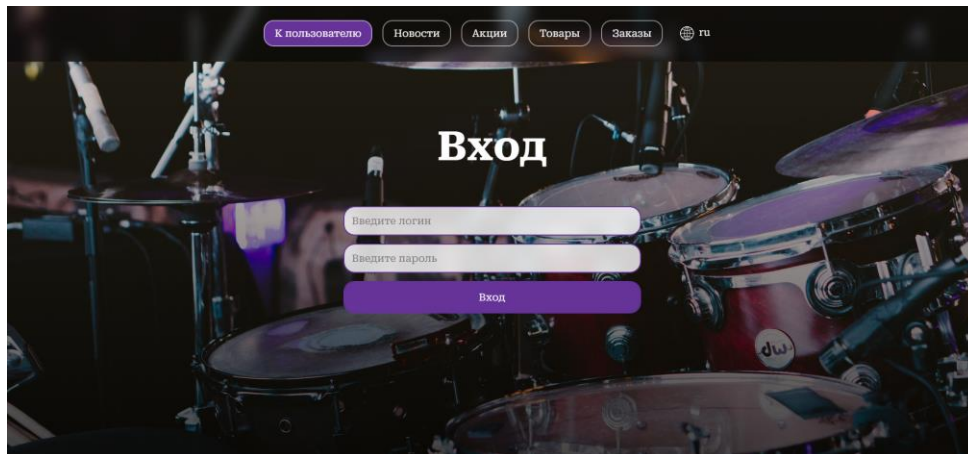


Рисунок 3.11 – Страница авторизации

На странице новостей в администраторской части можно будет создать новую новость и посмотреть уже существующие

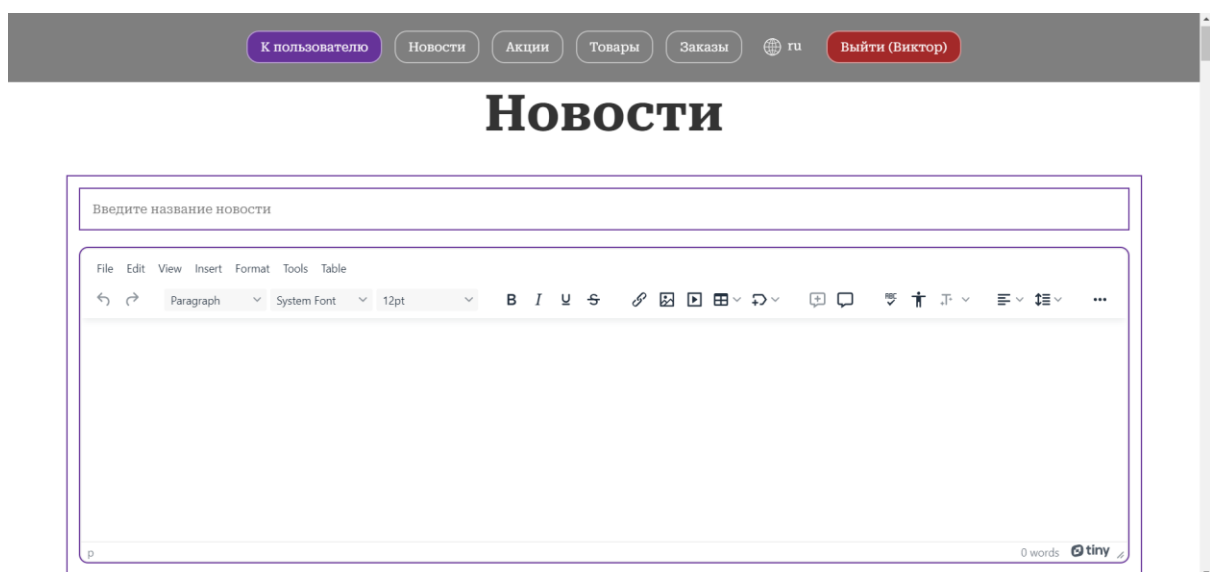
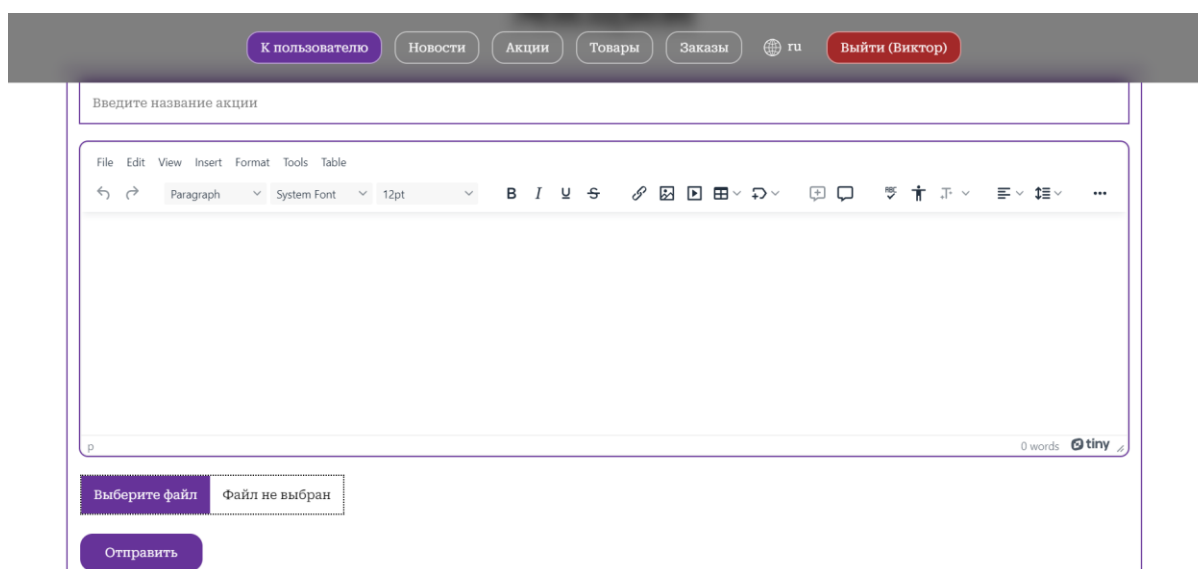


Рисунок 3.12 – Страница новостей в администраторской части

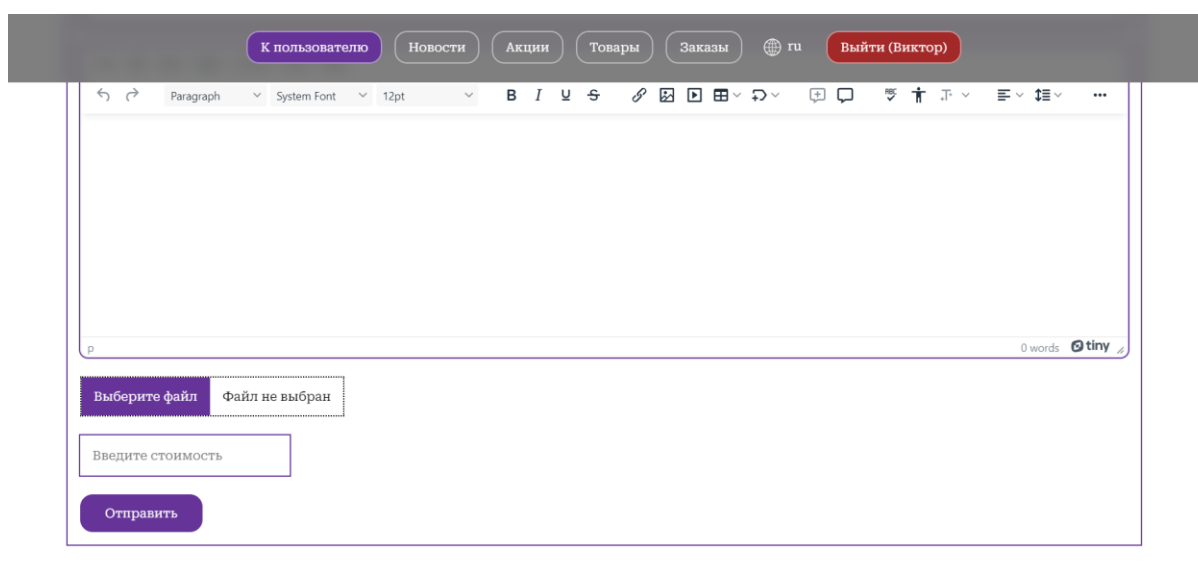
На странице акций в администраторской части можно будет создать новую акцию и посмотреть уже существующие



The screenshot shows the 'Actions' page in the administrator interface. At the top, there is a navigation bar with buttons: 'К пользователю', 'Новости', 'Акции', 'Товары', 'Заказы', a language selector 'ru', and a red 'Выйти (Виктор)' button. Below the navigation bar, there is a text input field with the placeholder 'Введите название акции'. Underneath this is a rich text editor with a menu (File, Edit, View, Insert, Format, Tools, Table) and a toolbar with various formatting icons. The editor's status bar at the bottom shows '0 words' and the 'tiny' logo. Below the editor, there is a file upload section with a button 'Выберите файл', a text box 'Файл не выбран', and an 'Отправить' button.

Рисунок 3.13 – Страница акций в администраторской части

На странице товаров можно будет создать новый товар и посмотреть уже существующие



The screenshot shows the 'Goods' page in the administrator interface. It has the same top navigation bar as the previous page. Below the navigation bar, there is a large, empty rich text editor with the same menu and toolbar as seen in the previous screenshot. The status bar at the bottom of the editor shows '0 words' and the 'tiny' logo. Below the editor, there is a file upload section with a button 'Выберите файл', a text box 'Файл не выбран', and an 'Отправить' button. Additionally, there is a text input field with the placeholder 'Введите стоимость' located below the file upload section.

Рисунок 3.14 – Страница товаров в администраторской части

Также на каждой странице в администраторской части была реализована кнопка выхода из аккаунта, которая не позволяла заходить без авторизации.

3.2 Проектирование базы данных

Для разработки проекта был выбран веб–интерфейс для администрирования СУБД phpMyAdmin. Перед разработкой базы данных были выделены основные сущности, определены их атрибуты и типы данных:

1) Таблица admins для хранения данных администраторов

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	More ▼
<input type="checkbox"/> 2	login	varchar(255)	utf8mb4_unicode_ci		No	None			More ▼
<input type="checkbox"/> 3	password	varchar(255)	utf8mb4_unicode_ci		No	None			More ▼
<input type="checkbox"/> 4	created_at	timestamp			Yes	NULL			More ▼
<input type="checkbox"/> 5	updated_at	timestamp			Yes	NULL			More ▼

Рисунок 3.15 – Структура таблицы администраторов

Таблица хранит в себе следующие поля: login – уникальное имя администратора; password - произвольный набор знаков, состоящий из букв, цифр и других символов, и предназначенный для подтверждения личности.

2) Таблица feedback для хранения данных об обратной связи, собираемой с пользователей

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	More ▾
<input type="checkbox"/> 2	fullName	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			More ▾
<input type="checkbox"/> 3	phone	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			More ▾
<input type="checkbox"/> 4	email	varchar(255)	utf8mb4_unicode_ci		No	None			More ▾
<input type="checkbox"/> 5	message	varchar(255)	utf8mb4_unicode_ci		No	None			More ▾
<input type="checkbox"/> 6	created_at	timestamp			Yes	NULL			More ▾
<input type="checkbox"/> 7	updated_at	timestamp			Yes	NULL			More ▾

Рисунок 3.16 – Структура таблицы обратной связи

Таблица хранит в себе следующие поля: **fullName** – имя пользователя; **phone** – номер телефона пользователя; **email** – почта пользователя; **message** – содержание обратной связи.

3) Таблица news для хранения новостей

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	More ▾
<input type="checkbox"/> 2	title	varchar(255)	utf8mb4_unicode_ci		No	None			More ▾
<input type="checkbox"/> 3	text	longtext	utf8mb4_unicode_ci		No	None			More ▾
<input type="checkbox"/> 4	image_path	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			More ▾
<input type="checkbox"/> 5	created_at	timestamp			Yes	NULL			More ▾
<input type="checkbox"/> 6	updated_at	timestamp			Yes	NULL			More ▾

Рисунок 3.17 – Структура таблицы новостей

Таблица хранит в себе следующие поля: **title** – название новости; **text** – содержание новости; **image_path** – путь к изображению новости, само изображение добавляется на диск.

4) Таблица orders для хранения заказов

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	More ▾
<input type="checkbox"/> 2	product_ids	longtext	utf8mb4_unicode_ci		No	None			More ▾
<input type="checkbox"/> 3	delivery_method	varchar(255)	utf8mb4_unicode_ci		No	None			More ▾
<input type="checkbox"/> 4	payment_method	varchar(255)	utf8mb4_unicode_ci		No	None			More ▾
<input type="checkbox"/> 5	fullname	varchar(255)	utf8mb4_unicode_ci		No	None			More ▾
<input type="checkbox"/> 6	phone	varchar(255)	utf8mb4_unicode_ci		No	None			More ▾
<input type="checkbox"/> 7	created_at	timestamp			Yes	NULL			More ▾
<input type="checkbox"/> 8	updated_at	timestamp			Yes	NULL			More ▾

Рисунок 3.18 – Структура таблицы заказов

Таблица хранит в себе следующие поля: `product_ids` – список `id` товаров, разделенный запятой, `delivery_method` – метод доставки, `payment_method` – способ оплаты; `fullname` – имя пользователя; `phone` – номер телефона.

5) Таблица products для хранения товаров

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	More ▾
<input type="checkbox"/> 2	name	varchar(255)	utf8mb4_unicode_ci		No	None			More ▾
<input type="checkbox"/> 3	text	longtext	utf8mb4_unicode_ci		No	None			More ▾
<input type="checkbox"/> 4	image_path	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			More ▾
<input type="checkbox"/> 5	cost	decimal(8,2)			No	None			More ▾
<input type="checkbox"/> 6	created_at	timestamp			Yes	NULL			More ▾
<input type="checkbox"/> 7	updated_at	timestamp			Yes	NULL			More ▾

Рисунок 3.19 – Структура таблицы товаров

Таблица хранит в себе следующие поля: name – наименование товара; text – описание товара; image_path – путь к изображению, само изображение сохраняется на диск; cost – стоимость товара.

6) Таблица special_offers для хранения акций

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	bigint		UNSIGNED	No	None		AUTO_INCREMENT	More ▼
<input type="checkbox"/> 2	title	varchar(255)	utf8mb4_unicode_ci		No	None			More ▼
<input type="checkbox"/> 3	text	longtext	utf8mb4_unicode_ci		No	None			More ▼
<input type="checkbox"/> 4	image_path	varchar(255)	utf8mb4_unicode_ci		Yes	NULL			More ▼
<input type="checkbox"/> 5	created_at	timestamp			Yes	NULL			More ▼
<input type="checkbox"/> 6	updated_at	timestamp			Yes	NULL			More ▼

Рисунок 3.20 – Структура таблицы акций

Таблица хранит в себе следующие поля: title – название акции; text – содержание акции; image_path – путь к изображению акции, само изображение добавляется на диск.

3.4 Проверка работоспособности веб-приложения

Была проведена проверка кроссбраузерности веб-приложения.

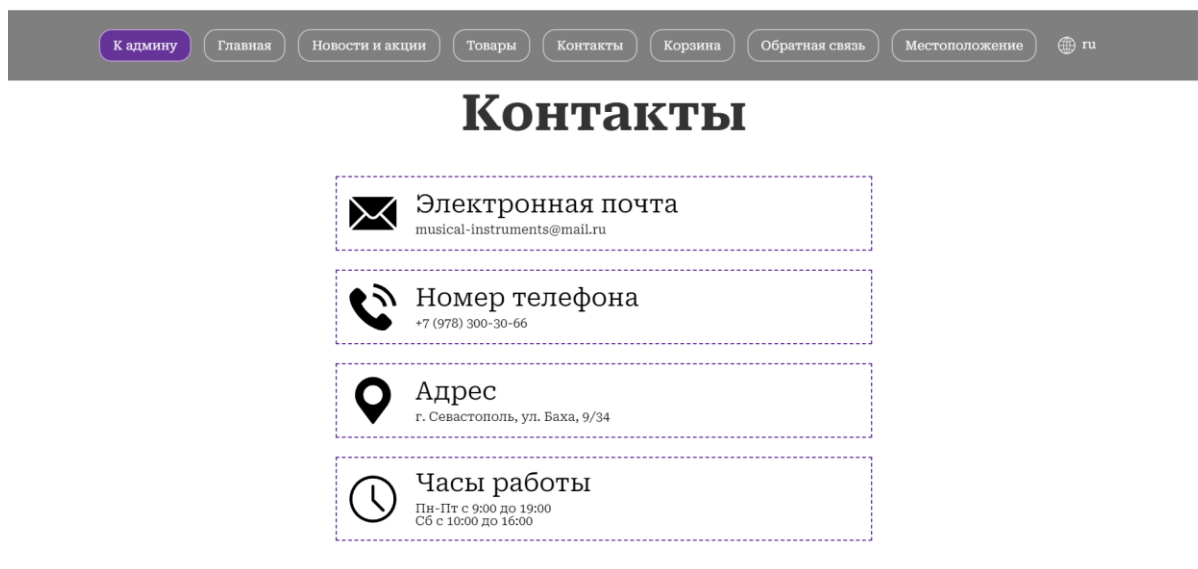


Рисунок 3.21 – Страница контактов в браузере Яндекс

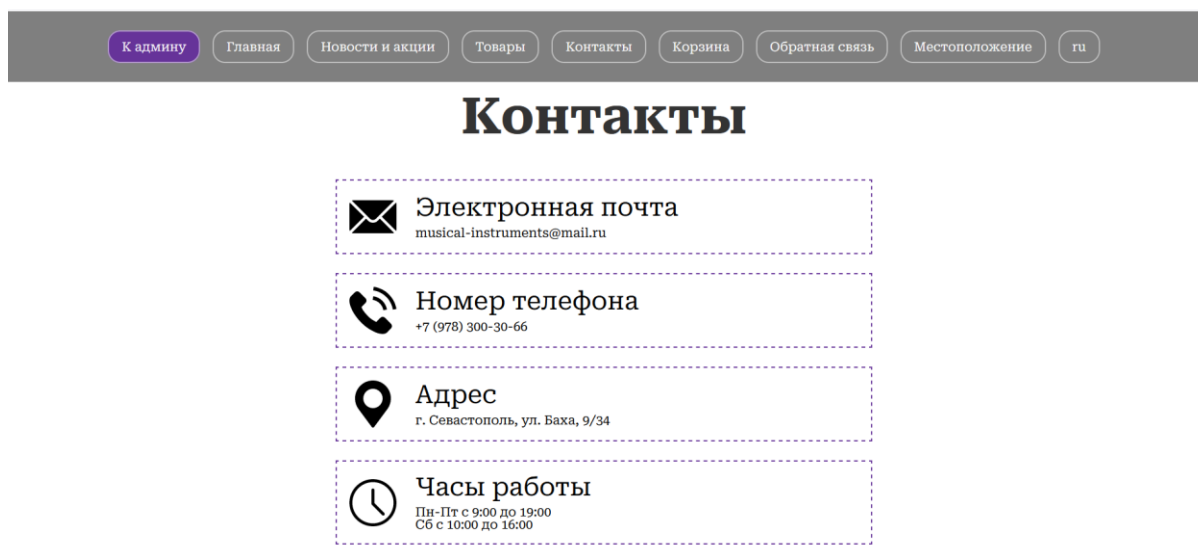


Рисунок 3.22 – Страница контактов в Firefox

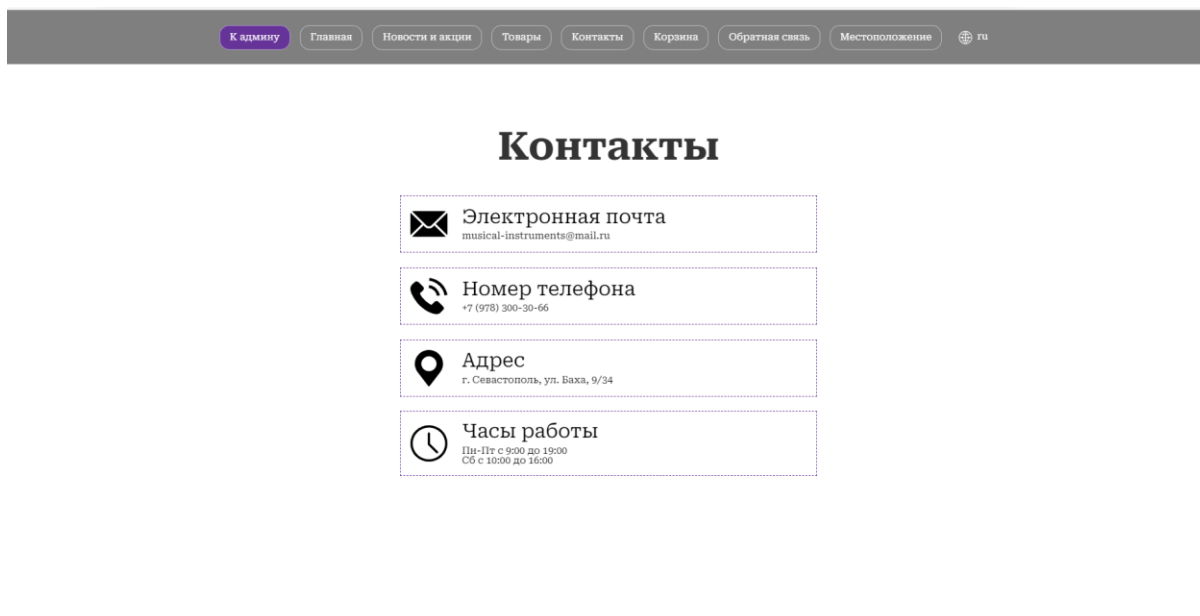


Рисунок 3.23 – Страница контактов в Microsoft Edge

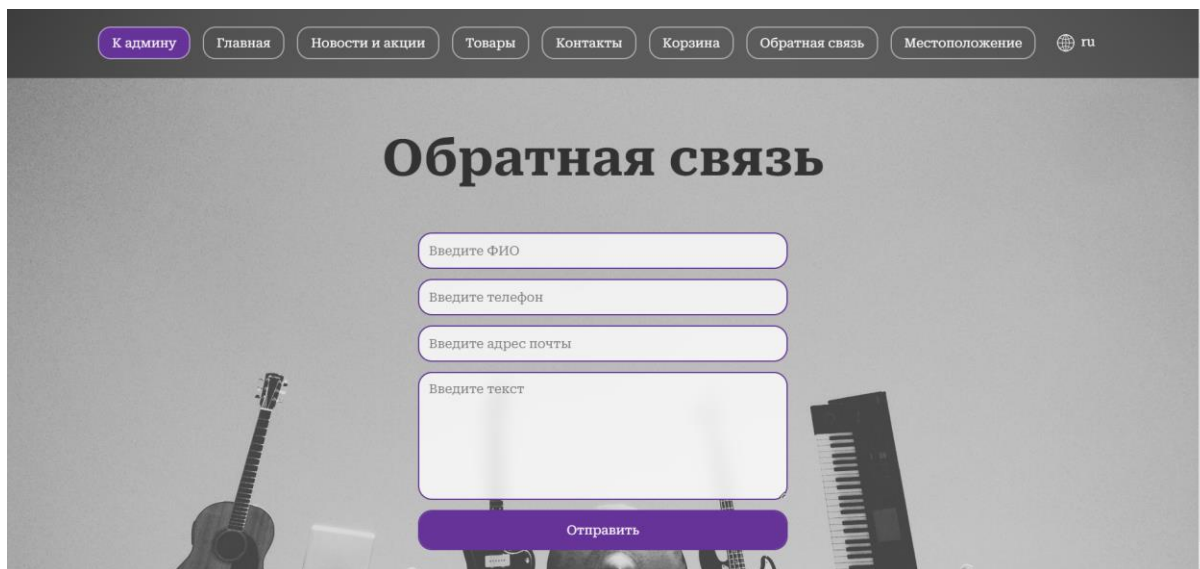


Рисунок 3.24 – Страница обратной связи в Яндекс браузере

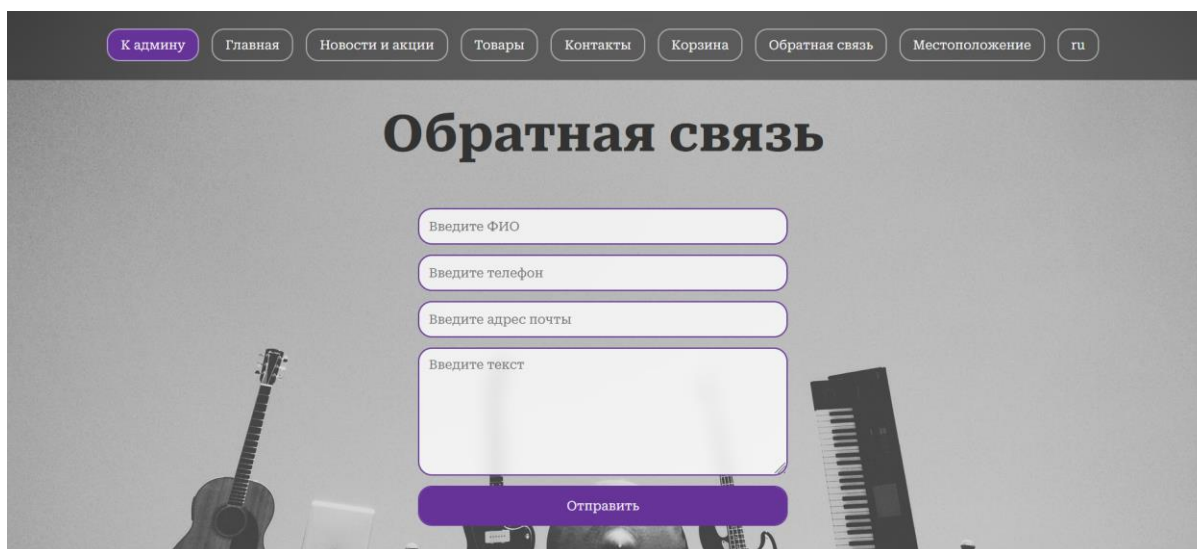


Рисунок 3.25 – Страница обратной связи в Firefox

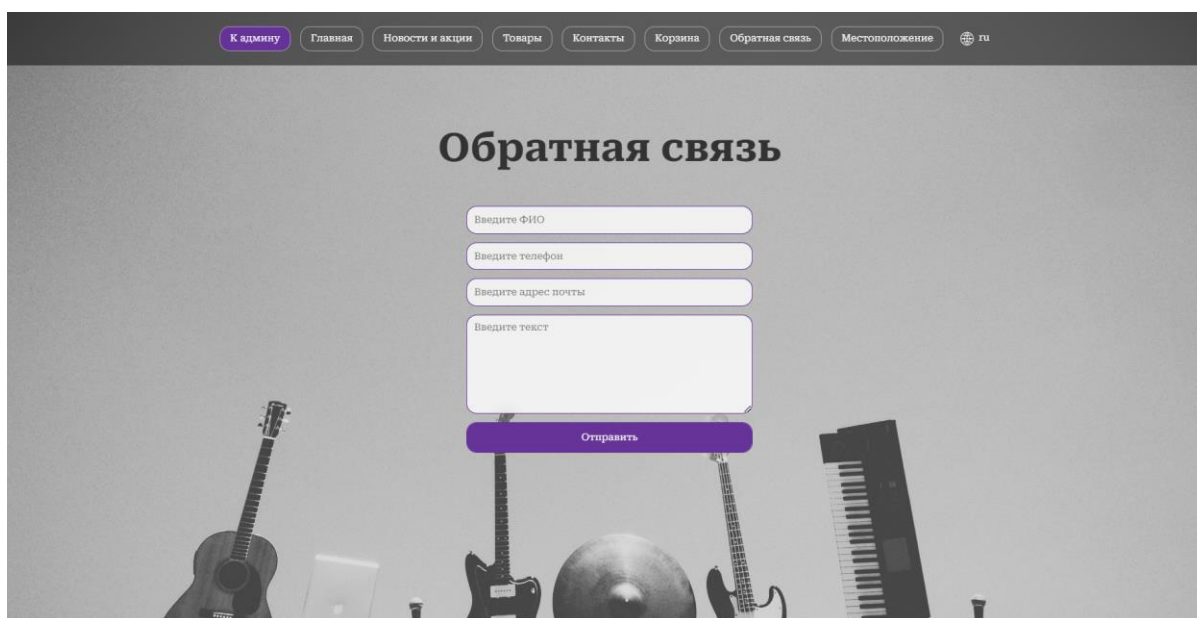


Рисунок 3.26 – Страница обратной связи в Microsoft Edge

Также была проведена проверка перевода страниц

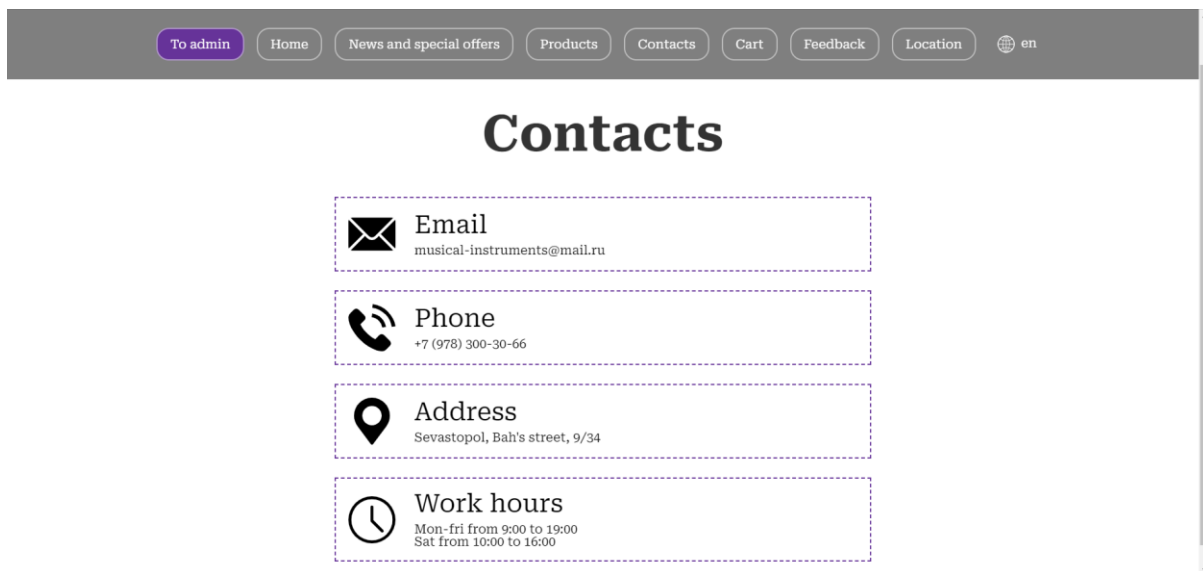


Рисунок 3.27 – Страница контактов на английском языке

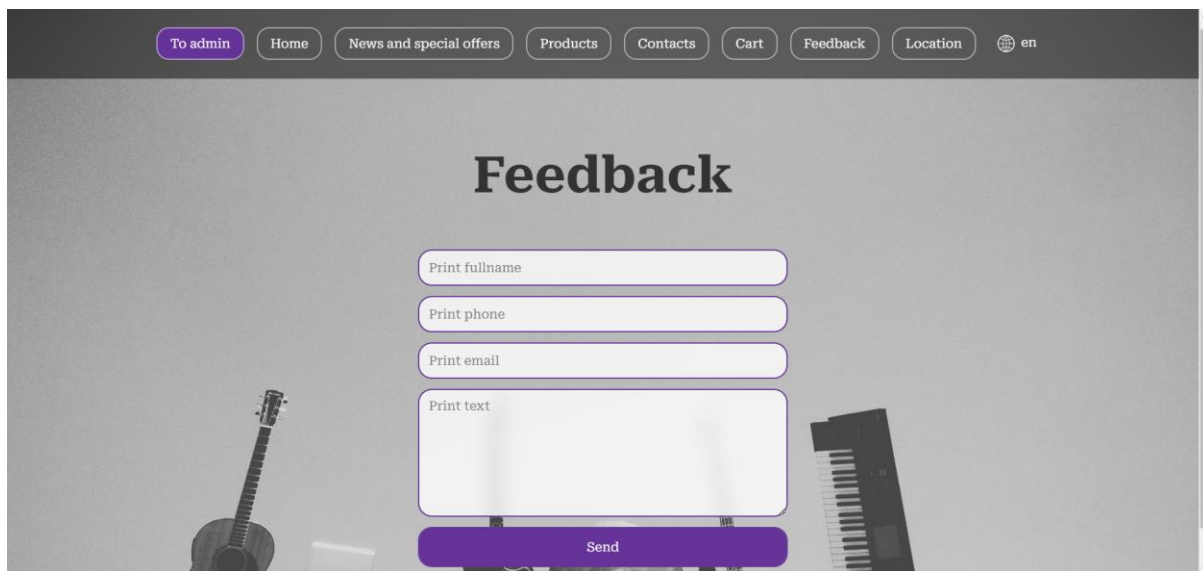


Рисунок 3.28 – Страница обратной связи на английском языке

Было проверена работа сайта в разных браузерах и на разных языках.

Вывод по разделу 3

В ходе разработки была спроектирована база данных, а также интерфейс приложения, после завершения разработки было проведено ручное тестирование функционала сайта а также его отображения в разных браузерах, также в ходе тестов не было выявлено ошибок при переводе страницы.

ЗАКЛЮЧЕНИЕ

В данной расчетно-графической работе было разработано приложение магазина музыкальных инструментов, в котором был реализован функционал добавления новостей и акций администратором, а также создания товаров. Со стороны пользователя был реализован функционал просмотра товаров, акций и новостей, а также корзина и страница обратной связи. Все данные записываются в специально спроектированную под задачу интернет магазина базу данных.

Приложение готово к эксплуатации, а также благодаря MVC архитектуре может быть легко расширено новым функционалом.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Laravel - The PHP Framework For Web Artisans // Laravel URL: <https://laravel.com/> (дата обращения: 10.05.2023).
2. VS Code Documentation // <https://code.visualstudio.com/docs/editor/vscode-web> (дата обращения: 25.04.2023).
3. phpMyAdmin Documentation // phpMyAdmin URL: <https://www.phpmyadmin.net> (дата обращения: 25.04.2023).
4. TinyMCE // <https://www.tiny.cloud/docs/> (дата обращения: 25.04.2023).
5. MDN Web Docs // MDN Web Docs URL: <https://developer.mozilla.org/ru/> (дата обращения: 25.04.2023).
6. Современный учебник JavaScript // JavaScript.ru URL: <https://learn.javascript.ru/> (дата обращения: 10.04.2023).
7. Справочник по HTML | htmlbook.ru // HTMLbook URL: <http://htmlbook.ru/html> (дата обращения: 25.04.2023).
8. Sass Basics // Sass URL: <https://sass-lang.com/> (дата обращения: 10.05.2023).
9. PHP: Documentation // PHP URL: <https://www.php.net/docs.php> (дата обращения: 10.05.2023).
10. MySQL :: MySQL Documentation // MySQL URL: <https://dev.mysql.com/doc/> (дата обращения: 25.04.2023).