|  |
| --- |
|  |

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**

**УП.08.02 Практика программирования**

**профессионального модуля ПМ.08 Разработка дизайна веб-приложения**

|  |  |  |
| --- | --- | --- |
| **Практикант** |  | **Рудаков Т.С.** |
| **Руководитель практики** |  | **Кривенко И.Д.** |

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc73010662)

[ОСНОВНАЯ ЧАСТЬ 4](#_Toc73010663)

[1. Описание целевой аудитории 4](#_Toc73010664)

[2. Структура сайта (концептуальная модель) 4](#_Toc73010665)

[3. Структура ролей пользователей (концептуальная модель) 5](#_Toc73010666)

[4. Структура CMS 6](#_Toc73010667)

[5. Описание страниц 6](#_Toc73010668)

[6. Описание плагинов и модулей 6](#_Toc73010669)

[7. Описание тестирования 6](#_Toc73010670)

[8. Руководство администратора 6](#_Toc73010671)

[ЗАКЛЮЧЕНИЕ 7](#_Toc73010672)

[ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ ИНФОРМАЦИИ 8](#_Toc73010673)

**ВВЕДЕНИЕ**

Практика УП.08.02 Практика программирования в период с 28.04.2025 г. по 10.05.2025 проводилась на базе мастерской «Разработка виртуальной и дополненной реальности» в БПОУ ОО «Омский авиационный колледж имени Н.Е. Жуковского».

Учебная практика УП.08.02 Разработка мультимедийных приложений направлена на комплексное освоение студентами вида профессиональной деятельности «Разработка дизайна веб-приложений» по специальности СПО 09.02.07 «Информационные системы и программирование», формирование общих и профессиональных компетенций, а также приобретение необходимых умений и опыта практической работы.

Практика способствует формированию, закреплению и развитию практических навыков и компетенций в процессе выполнения определенных видов работ, связанных с будущей профессиональной деятельностью.

Виды работ:

* Настройка виртуального Web-сервера;
* Моделирование и создание структуры сайта;
* Установка и настройка расширений компонентов, модулей, плагинов;
* Настройка интерфейсов пользователя;
* Тестирование и отладка проекта;
* Размещение сайта в сети Интернет;
* Презентация решений.

**ОСНОВНАЯ ЧАСТЬ**

1. **Описание целевой аудитории**

Целевая аудитория сайта интернет-котокафе, предлагающего заказ еды и бронирование столиков, включает жителей одного города, которые ценят уникальную концепцию котокафе, сочетающую еду и общение с кошками. Основной фокус — на людях от 18 до 35 лет, активно пользующихся онлайн-сервисами для заказа еды и бронирования, хотя общий возрастной диапазон варьируется от 16 до 45 лет.

Молодежь (16–22 года) — старшеклассники и студенты, привлеченные атмосферой котокафе и возможностью делать фото для социальных сетей (Instagram, TikTok). Они часто финансово зависимы от родителей, поэтому выбирают бюджетные блюда, акции или комбо-наборы. Предпочитают бронировать столики для встреч с друзьями, реже используют доставку из-за непредсказуемого времени. Им нужен простой интерфейс сайта, яркий дизайн и информация об акциях.

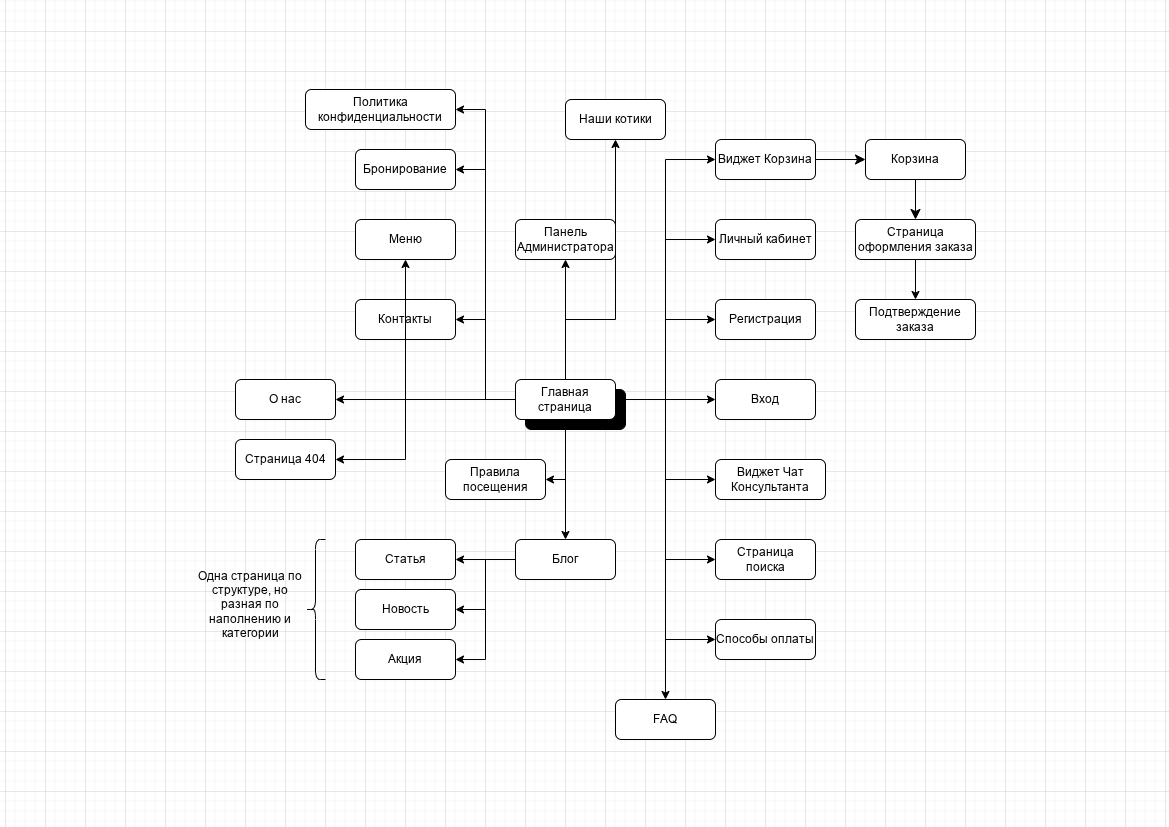
Молодые взрослые (23–30 лет) — специалисты и фрилансеры, ценящие удобство и уникальный опыт. Они бронируют столики для вечерних посиделок или заказывают еду домой/в офис, несмотря на возможные задержки доставки. Проверяют историю заказов в личном кабинете и реагируют на программы лояльности. Им важны адаптивный дизайн сайта, информация о составе блюд и удобная навигация.

Взрослые (31–45 лет) — работающие люди, часто семейные, выбирают котокафе для отдыха или семейных ужинов. Они активно бронируют столики, реже используют доставку, ценят качество еды и обслуживания. Им нужна надежная система бронирования и удобный личный кабинет для управления заказами.

Любители кошек (все возраста) посещают кафе или заказывают еду, чтобы поддержать заведение и насладиться кошачьей тематикой. Они изучают информацию о кошках на сайте, бронируют столики и интересуются благотворительными инициативами. Доставка для них менее приоритетна.

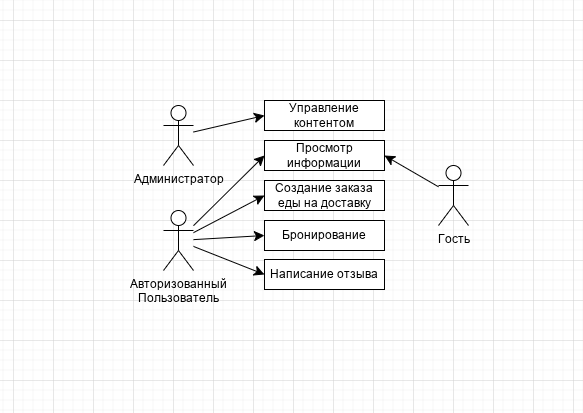
Аудитория ограничена городскими жителями, уверенно использующими смартфоны и компьютеры. Они ожидают быстрой загрузки сайта, адаптивного дизайна и функционального личного кабинета. Интересы включают кошачью тематику, экологию (например, экологичную упаковку) и локальные инициативы по защите животных. Молодежь заказывает или бронирует 1–2 раза в неделю, взрослые — реже, но чаще посещают кафе лично.

1. **Структура сайта (концептуальная модель)**

На основе анализа предметной области и технического задания была разработана концептуальная модель сайта интернет-котокафе в среде Microsoft Visio. Процесс включал выделение основных сущностей, таких как меню, бронирование, пользователи, заказы, корзина и информационные разделы (например, "О нас", "Наши котики", "Блог"), а также определение их атрибутов и взаимосвязей. Это позволило структурировать данные и процессы, связанные с заказом еды, бронированием столиков и взаимодействием с пользователями. Концептуальная модель отражает ключевые объекты и их связи, обеспечивая основу для проектирования базы данных и функционала сайта. Созданная модель визуализирована и представлена на рисунке 1, что облегчает понимание структуры и логики работы сайта интернет-котокафе.

**Рисунок 1 - Концептуальная модель сайта**

1. **Структура ролей пользователей (концептуальная модель)**

На Рисунке 2 представлена диаграмма вариантов использования, которая иллюстрирует взаимодействие различных категорий пользователей (актеров) с функциями разрабатываемой системы. Эта диаграмма помогает наглядно представить логику работы веб-приложения и доступные операции для каждого типа пользователя.

**Рисунок 2 - Концептуальная модель сайта**

Данный веб-ресурс разработан с учетом потребностей следующих основных категорий пользователей:

* **Гость.** Посетитель, который еще не прошел процедуру регистрации и авторизации на сайте. Гости имеют возможность свободно просматривать общедоступную информацию о кафе, изучать меню, знакомиться с услугами. Однако функционал, связанный с совершением заказов или бронированием, для них ограничен.
* **Авторизованный пользователь.** Это зарегистрированный пользователь системы. Помимо доступа ко всей информации, доступной гостям, авторизованные пользователи получают расширенные возможности. Они могут оформлять заказы на доставку еды, бронировать столики в кафе, а также оставлять свои отзывы и комментарии.
* **Администратор сайта.** Данная роль обладает максимальными правами доступа в системе. Администратор управляет содержимым веб-приложения через специализированный интерфейс — административную панель. Его задачи включают добавление и редактирование информации, работу с заказами и бронированием, управление учетными записями пользователей и другие действия, обеспечивающие функционирование сайта.

1. **Структура CMS**

#### Для реализации серверной части веб-приложения и системы управления контентом была выбрана October CMS. Это современная CMS с открытым исходным кодом, построенная на базе PHP-фреймворка Laravel.

В рамках данного проекта October CMS применялась как Headless CMS. Это значит, что CMS отвечала только за хранение, организацию и предоставление данных. Управление контентом происходит через удобную административную панель October CMS. Визуальная часть сайта и пользовательский интерфейс разработаны отдельно с использованием Node.js, Vue.js и Tailwind CSS.

Взаимодействие между фронтенд-приложением и October CMS реализовано через REST API. Для создания API-эндпоинтов использовалась функциональность тем October CMS, чтобы передавать данные в формате, понятном фронтенду.

Управление данными для сайта котокафе сделано через модели October CMS. Модели создавались с использованием плагина RainLab.Builder и кастомного плагина Teiwi.Cafe. Основные типы данных, которые используются:

* **Product** (меню котокафе)
* **Orders** (заказы пользователя)
* **Cart** (корзина пользователя)
* **Posts** (публикации из блога)
* **BlogCategories** (категории блога)
* **Categories** (для product)
* **Bookings** (бронирование мест в котокафе)
* **Testimonials** (отзывы на заказы)
* **UserAddress** (список адресов для юзера)
* **User** (расширение от RainLab.User плагина)
* **Cats** (список кошек нашего котокафе)

Плагин RainLab.Builder помог быстро создать структуру этих данных (модели, таблицы в базе данных) и сделать интерфейсы для управления ими в админке October CMS. Кастомный плагин Teiwi.Cafe объединяет специфичные для проекта данные и логику, созданные с помощью Builder. В этом плагине хранятся миграции, модели, контроллеры и пункты меню для админки. Обработка данных и основная логика на стороне бэкенда, а создание внешнего вида и взаимодействие с пользователем – на фронтенде.

October CMS, как система на базе Laravel, использует шаблон Model-View-Controller (MVC) для организации кода бэкенда.

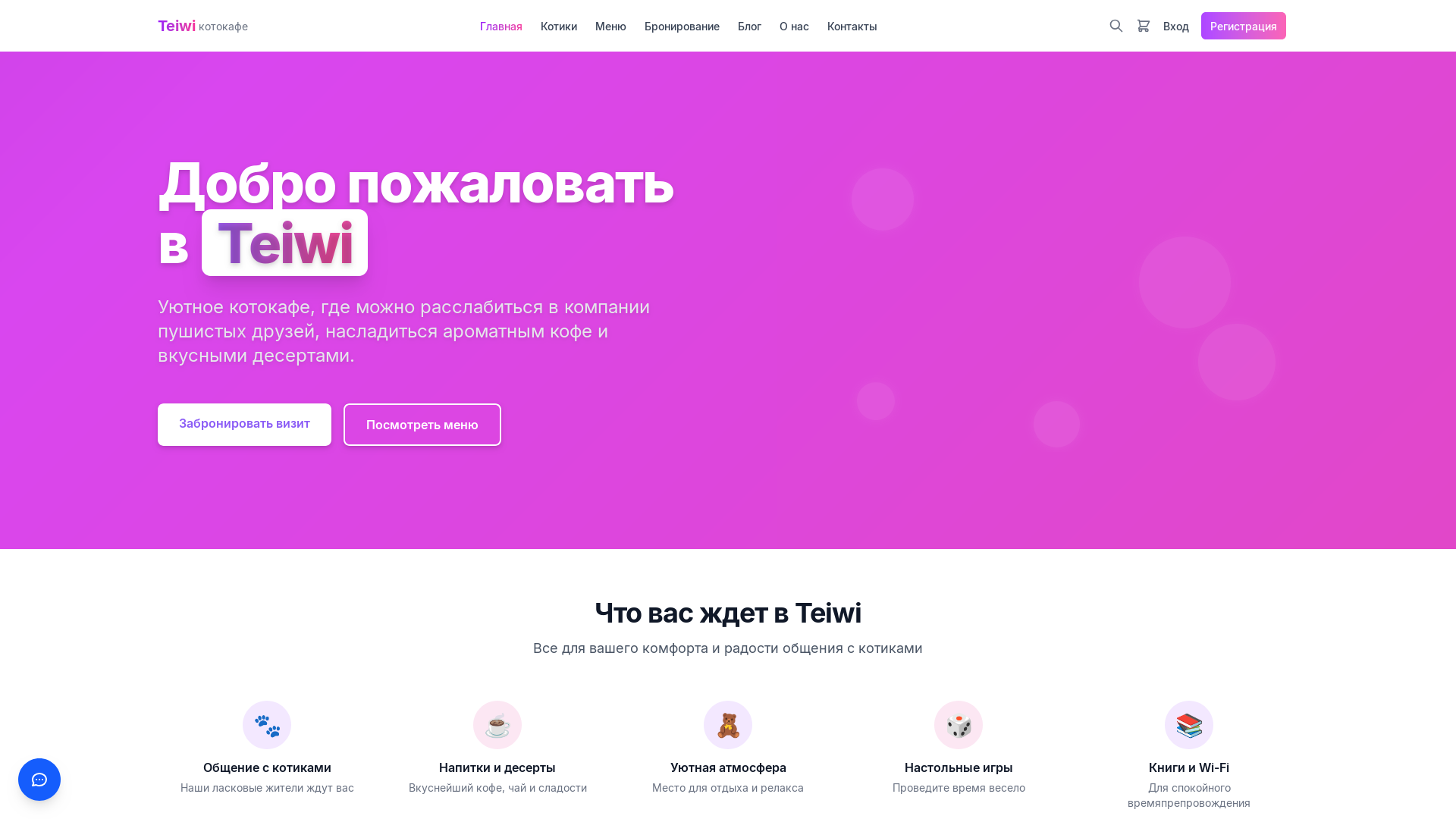
* Модель отвечает за данные и связь с базой данных (это наши сущности Product, Orders и другие).
* Контроллер обрабатывает запросы (от фронтенда через API) и работает с Моделями.
* View в Headless CMS по сути просто готовит данные для отправки фронтенду, а не создает HTML страницы.

1. **Описание страниц**

Визуальная часть веб-приложения котокафе "Teiwi" представляет собой набор ключевых страниц, разработанных с использованием фреймворка Vue.js для логики интерфейса и библиотеки стилей Tailwind CSS для оформления и адаптивной верстки. Каждая страница имеет свое специфическое назначение и структуру, взаимодействуя с серверной частью (October CMS) посредством REST API для получения и отправки динамических данных. Ниже представлено описание основных страниц приложения.

**Главная страница**

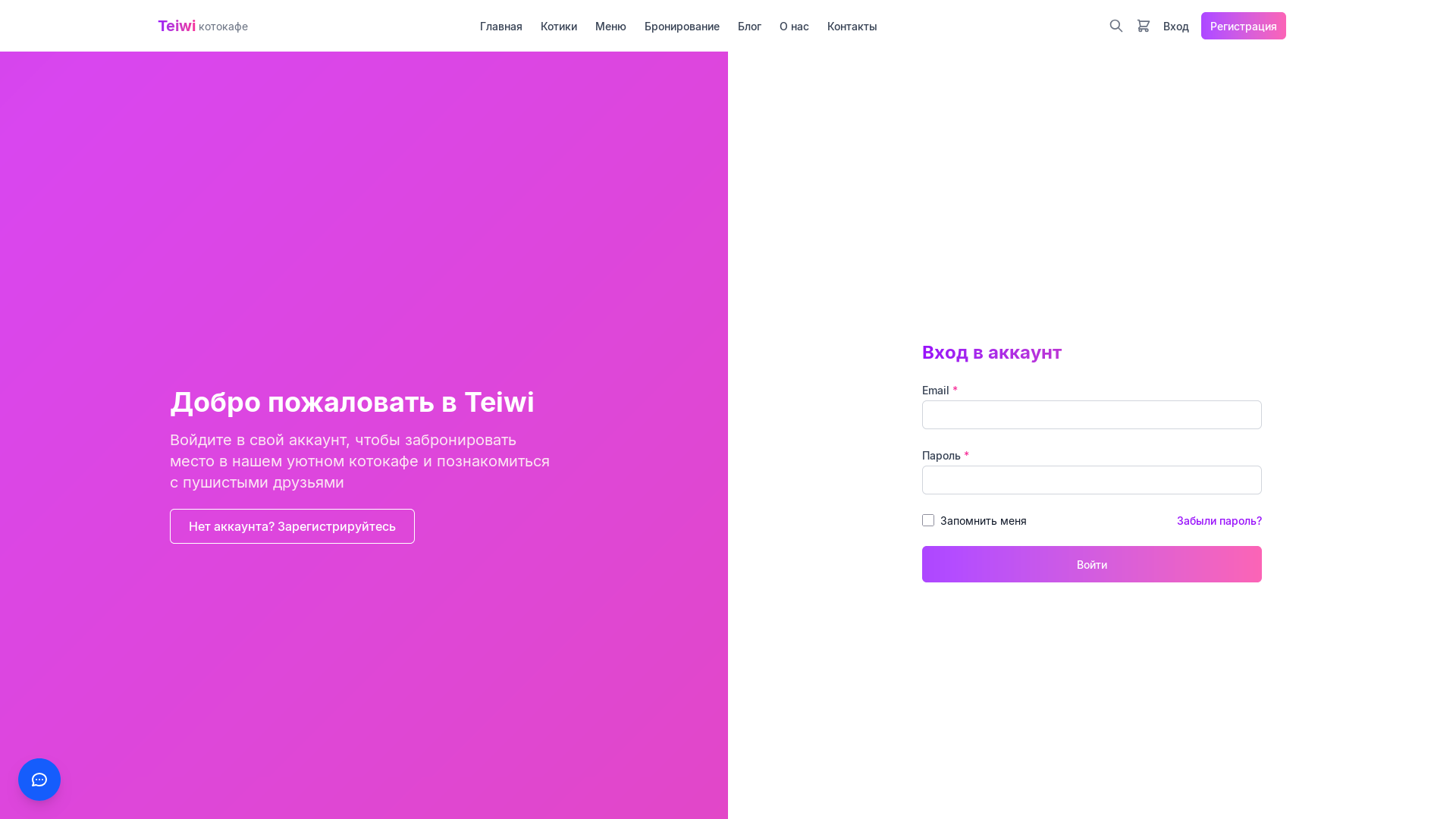
Является стартовой точкой для посетителей сайта. Ее основная задача — представить котокафе, его уникальную концепцию и услуги, а также мотивировать пользователя к дальнейшим действиям, таким как бронирование визита или ознакомление с меню. На странице расположен крупный геройский баннер с призывами к действию, секции с описанием услуг кафе, блок с избранными позициями из меню с возможностью добавления в корзину, а также блок с отзывами клиентов. Дизайн страницы адаптивен, использует возможности Tailwind CSS для создания сеток, карточек и оформления элементов. Данные для секций избранных блюд и отзывов (если они динамические) запрашиваются через соответствующие API-эндпоинты бэкенда, описанного в разделе 4.



**Рисунок 3 - Главная страница**

**Страница Входа**

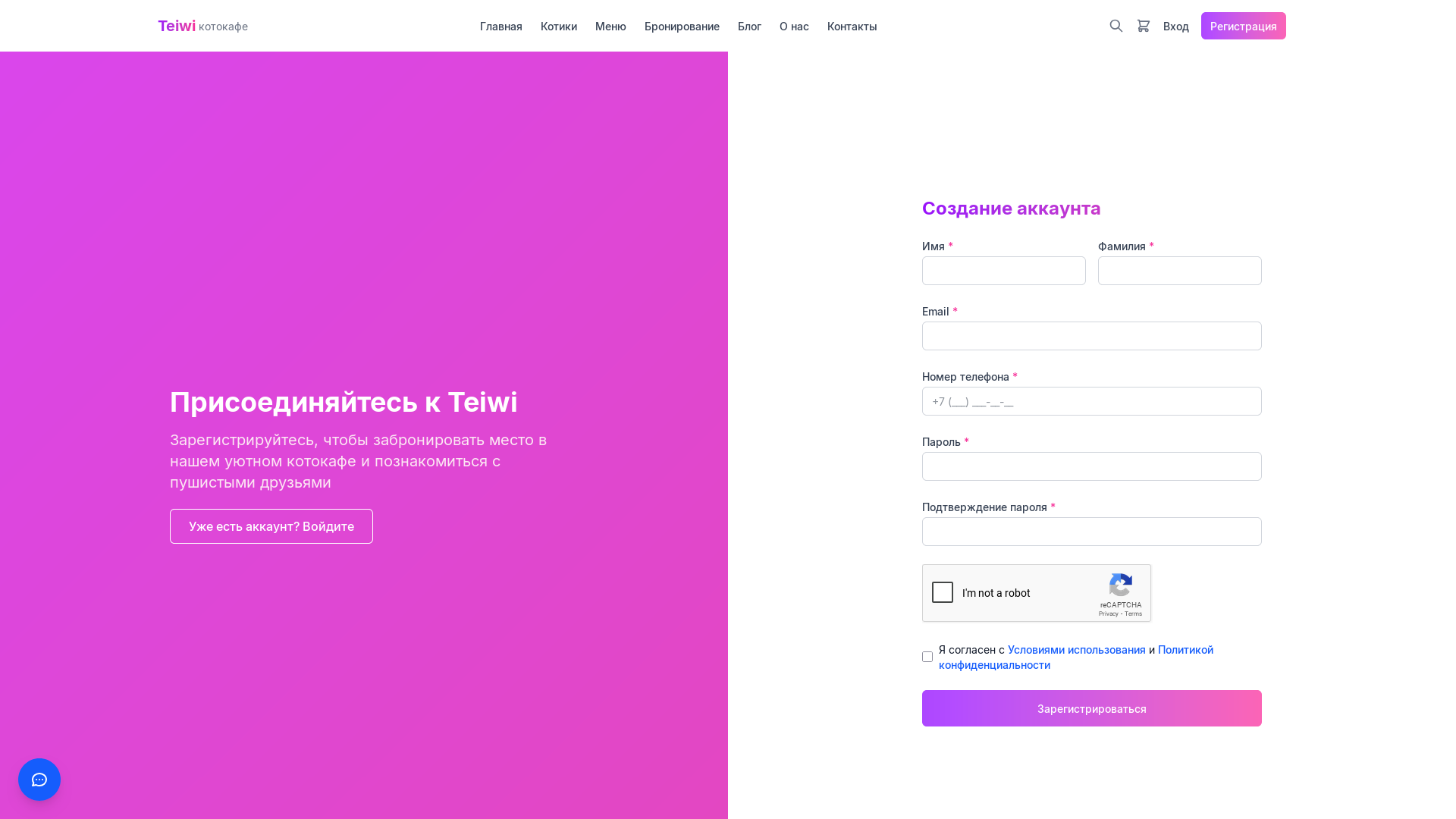
Предназначена для аутентификации существующих пользователей. Здесь расположена форма с полями для ввода email и пароля, опцией "Запомнить меня", а также ссылками для регистрации нового пользователя и восстановления пароля. Страница использует компонент AuthLayout для общего оформления, обеспечивающего адаптивное отображение на разных устройствах. После ввода данных форма отправляет их на API-эндпоинт авторизации в October CMS. При успешном входе данные пользователя и содержимое корзины синхронизируются с сервером через соответствующие хранилища состояния (Pinia Stores).



**Рисунок 4 - Страница входа**

**Страница Регистрации**

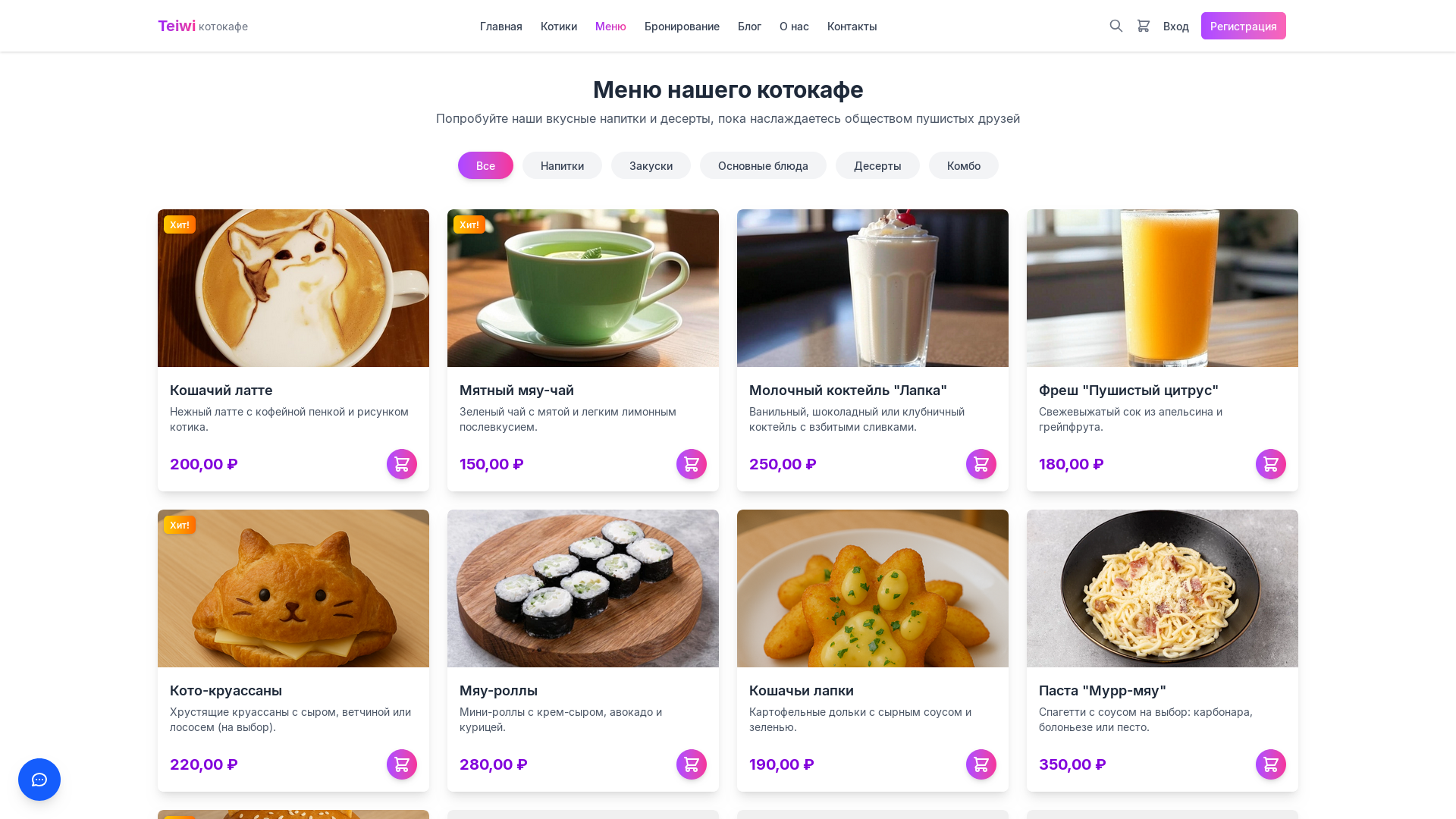
Позволяет новым пользователям создать учетную запись для доступа к полному функционалу сайта. На странице представлена форма с полями для ввода имени, фамилии, email, телефона, пароля и его подтверждения. Для защиты от автоматической регистрации используется reCAPTCHA v2. Визуально страница оформлена схоже со страницей входа, используя AuthLayout и адаптивный дизайн на Tailwind CSS. Введенные данные отправляются на API-эндпоинт регистрации в October CMS, после чего происходит синхронизация данных пользователя и корзины.



**Рисунок 5 - Страница Регистрации**

**Страница Меню**

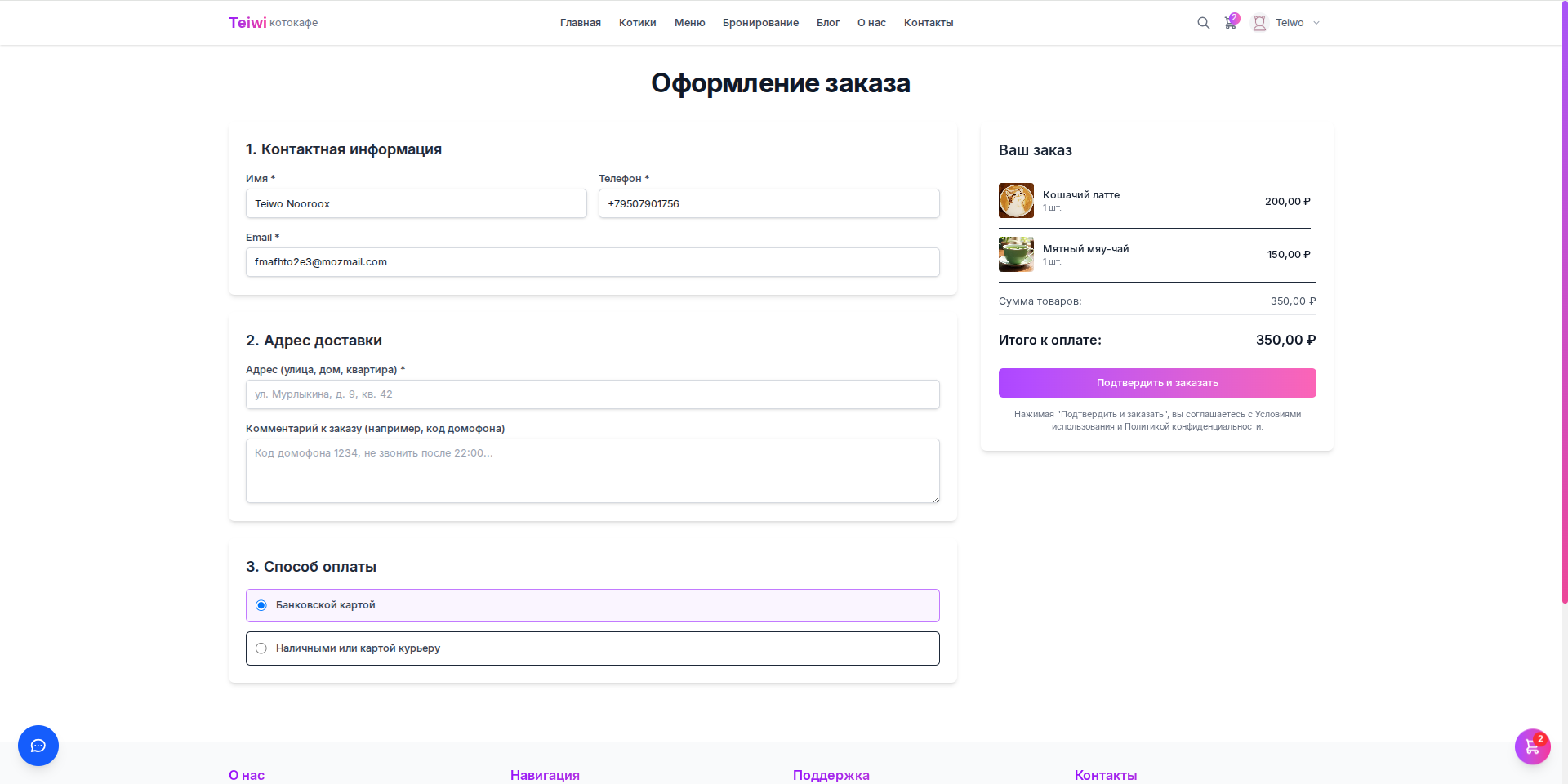
Представляет полный список блюд и напитков, доступных для заказа в котокафе. Пользователи могут просматривать позиции меню, применять фильтры по категориям для удобной навигации. Основную часть страницы занимает сетка карточек с изображениями, названиями, ценами и кнопками для добавления блюд в корзину. Реализованы состояния загрузки, ошибки и пустого меню. Данные о блюдах и категориях получаются из October CMS через соответствующие API-эндпоинты. Добавление товаров в корзину также синхронизируется с бэкендом через API.



**Рисунок 6 - Страница Меню**

**Страница Подтверждения заказа**

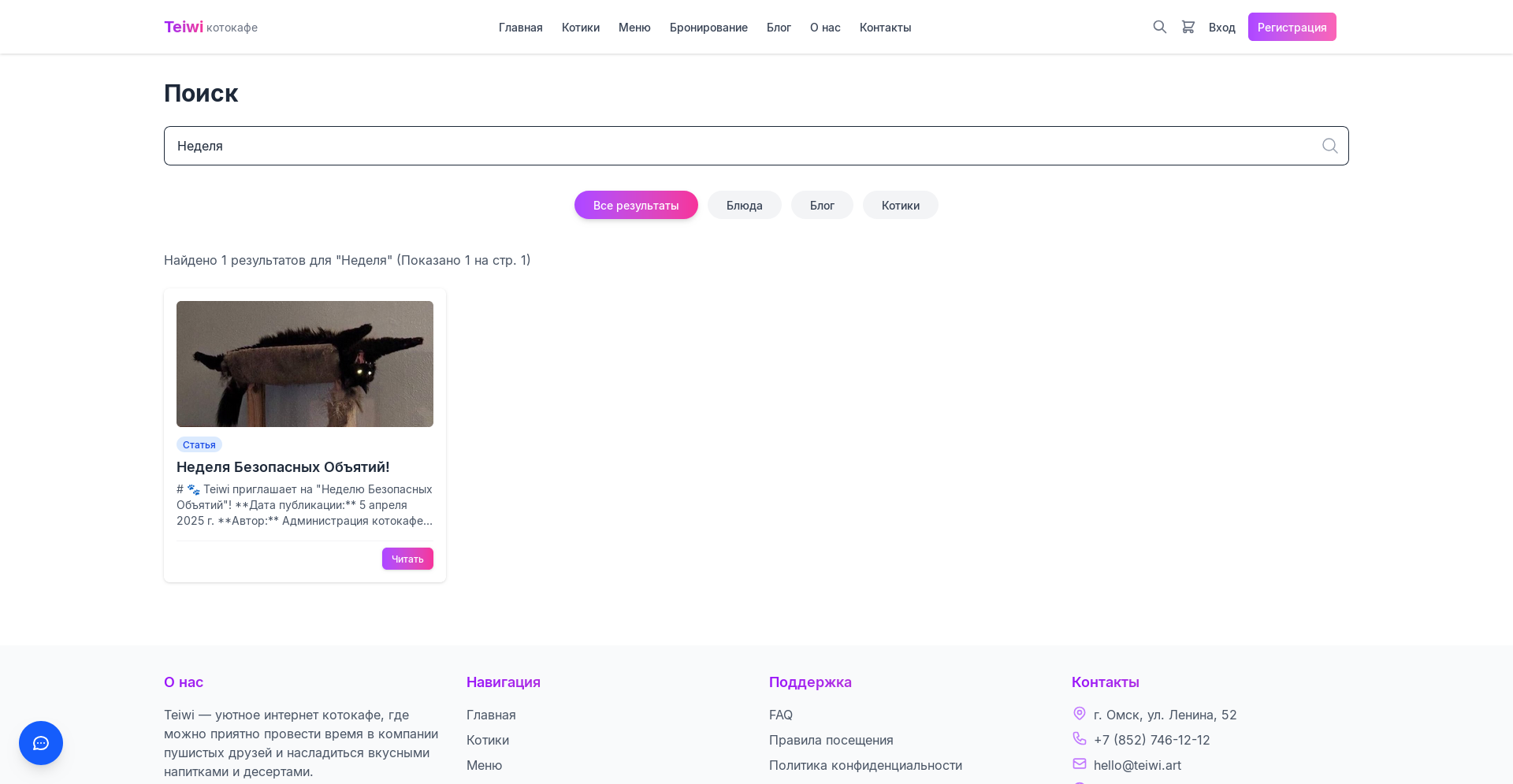
Отображается после успешного оформления заказа, предоставляя пользователю полную информацию о его деталях: номер заказа, дата, текущий статус, перечень заказанных товаров с количеством и ценой, общая сумма, а также указанный адрес доставки и способ оплаты. Страница запрашивает детали конкретного заказа с бэкенда по его идентификатору через API. Предусмотрена обработка случаев, когда данные заказа недоступны.



**Рисунок 7 - Подтверждения заказа**

**Страница Поиска**

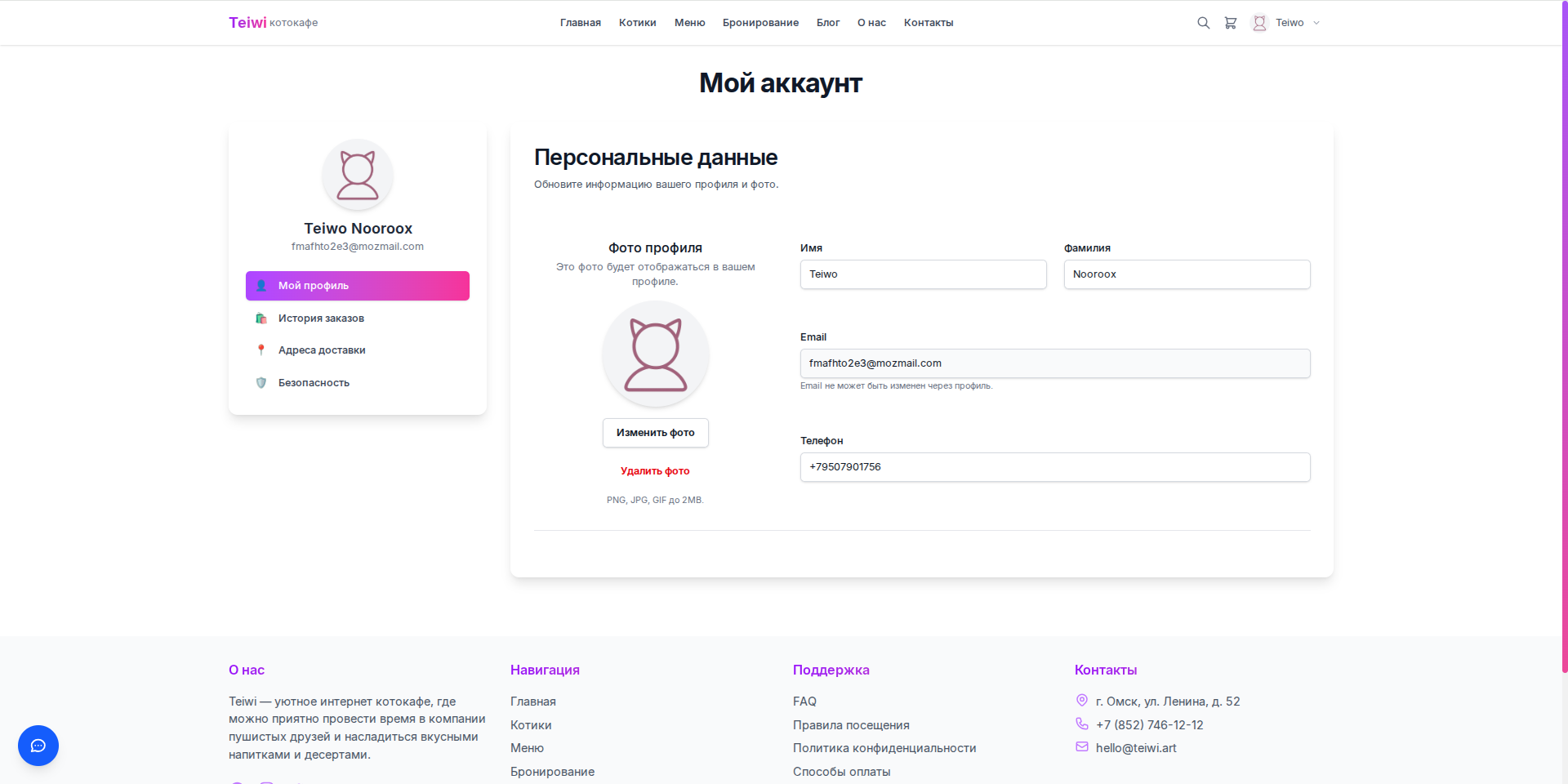
Реализует функциональность поиска по различным разделам сайта, включая меню, записи блога и список кошек. На странице расположено поле для ввода поискового запроса и вкладки для фильтрации результатов по типам. Результаты поиска отображаются в виде сетки карточек, каждая из которых содержит ключевую информацию о найденном элементе. Реализована пагинация для управления большим количеством результатов. Страница отправляет поисковые запросы на специализированный API-эндпоинт в October CMS, который выполняет поиск по заданным критериям и возвращает стандартизированные результаты.



**Рисунок 8 - Страница поиска**

**Личный кабинет**

Предоставляет авторизованным пользователям доступ к управлению своей учетной записью и просмотру истории активности. Раздел выполнен в виде разделенного макета с боковым меню, содержащим вкладки: "Профиль" (для обновления личных данных и аватара), "История заказов" (список предыдущих заказов), "Адреса доставки" (управление сохраненными адресами) и "Безопасность" (заглушка для смены пароля). Данные для каждой вкладки запрашиваются с бэкенда через соответствующие API-эндпоинты. Управление адресами (добавление, редактирование, удаление) также осуществляется через API.

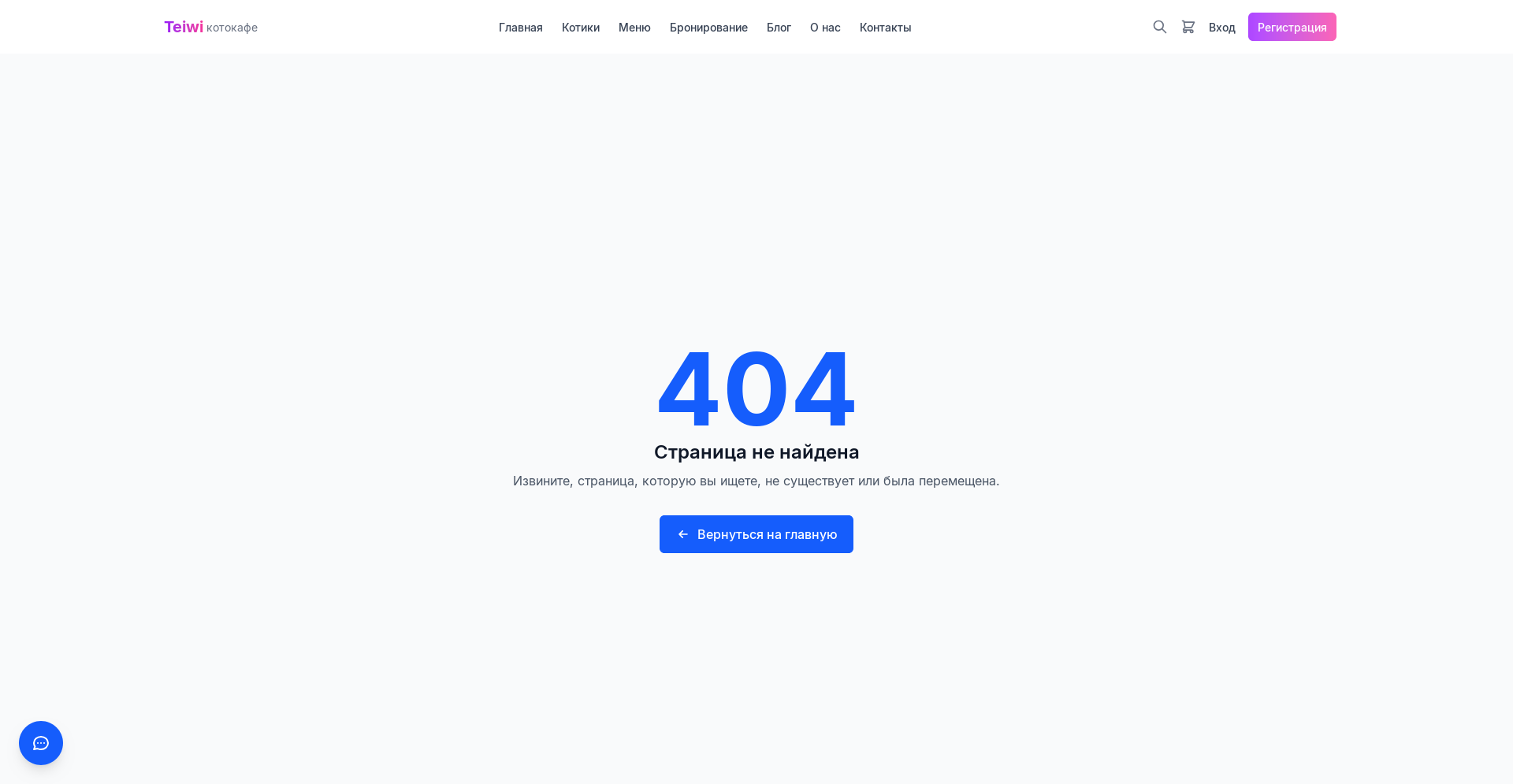


**Рисунок 9 - Страница личного кабинета**

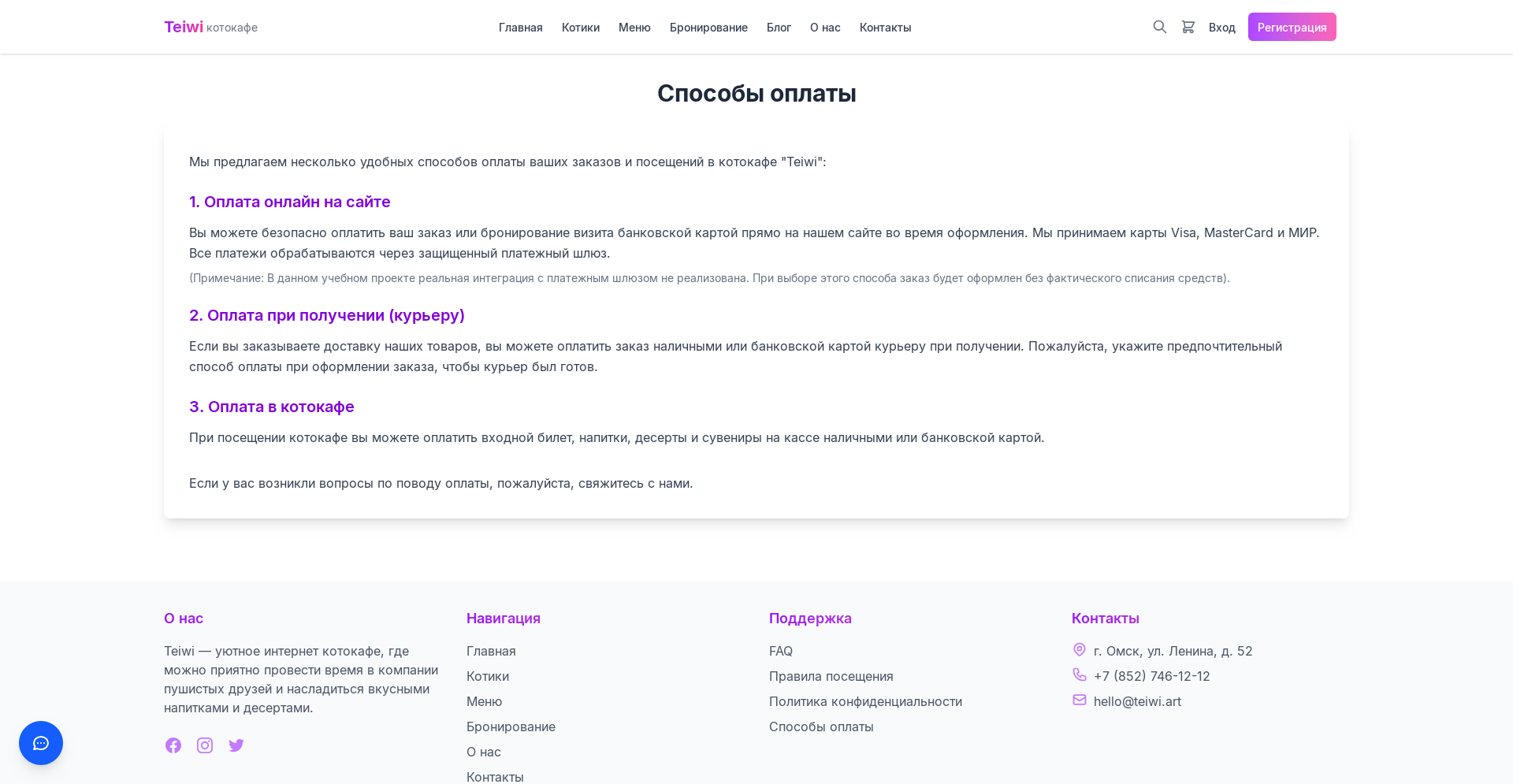
**Статические страницы:**

Помимо страниц с динамическим контентом, в приложении реализованы статические страницы, содержащие информацию, не требующую регулярного обновления из бэкенда. К ним относятся:

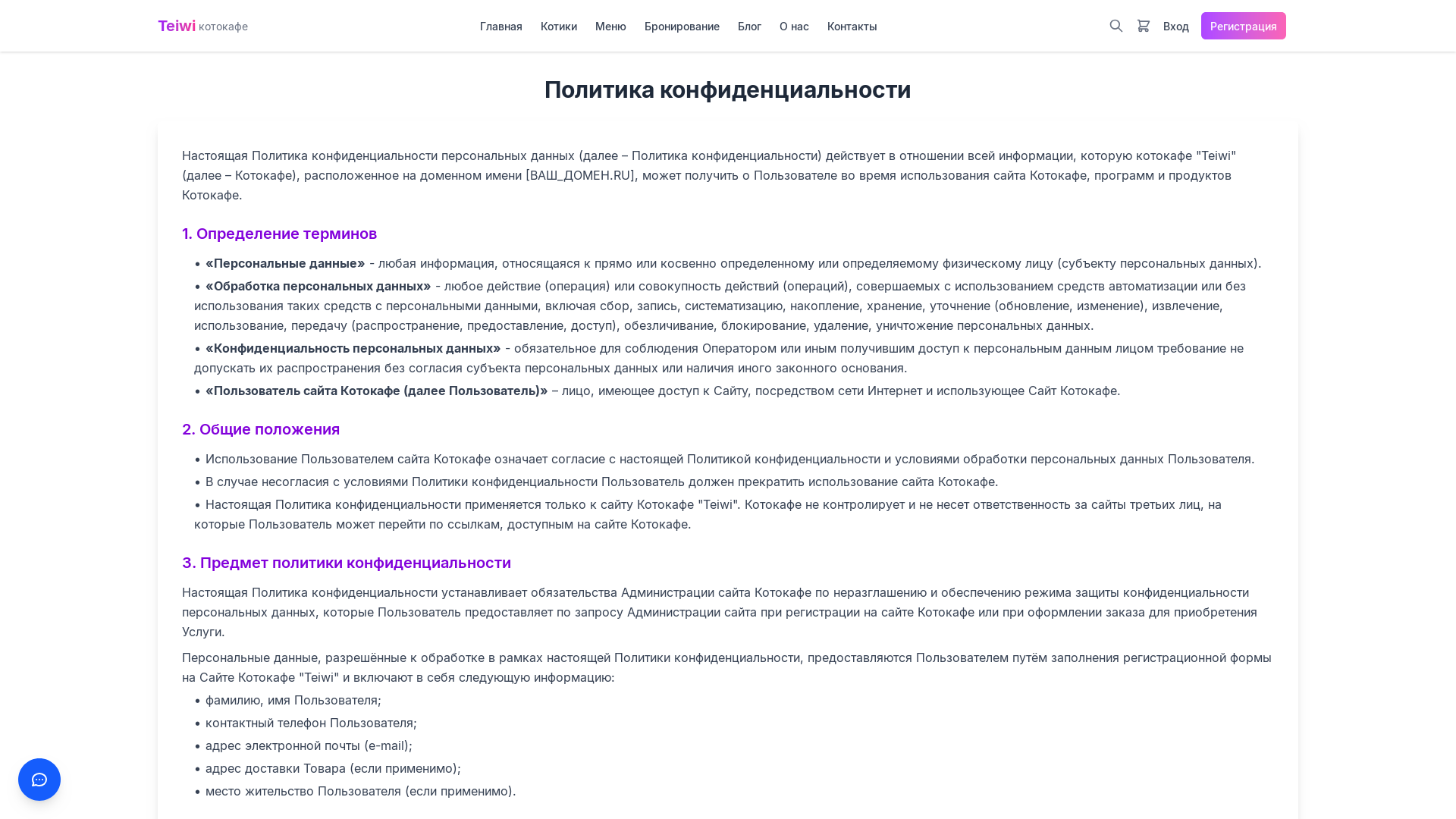
* Страница не найдена (404). Отображается при попытке перехода по несуществующему адресу. Содержит сообщение об ошибке и ссылку на главную страницу. Не взаимодействует с бэкендом.

**Рисунок 10 - Страница не найдена**

* Способы оплаты. Информирует пользователей о доступных вариантах оплаты заказов и услуг котокафе. Содержит текстовое описание возможных способов. Не взаимодействует с бэкендом.

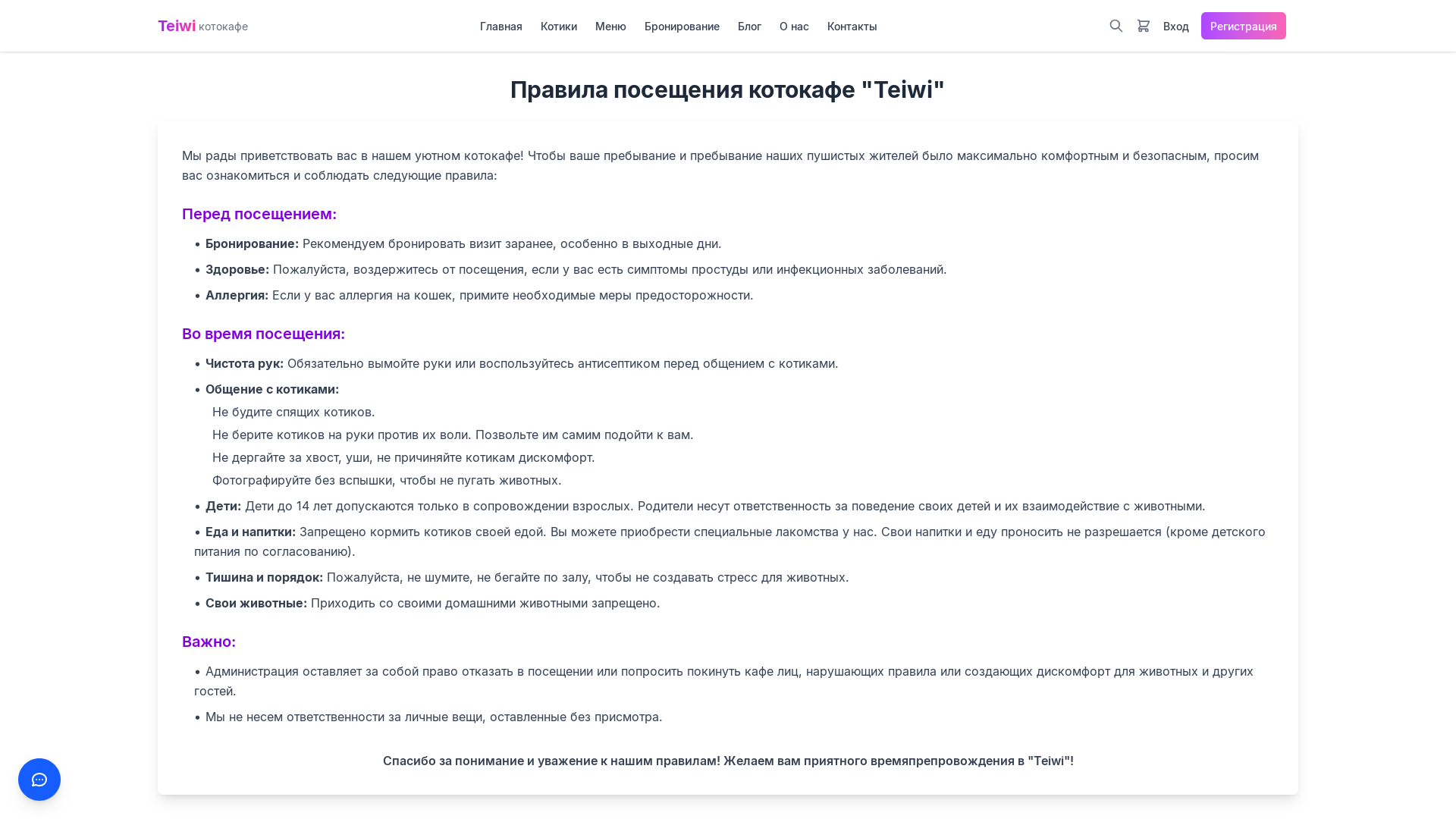
**Рисунок 11 - Способ оплаты**

* Политика конфиденциальности. Раздел с подробным описанием правил сбора, обработки и хранения персональных данных пользователей. Содержит исключительно текстовый контент. Не взаимодействует с бэкендом.



**Рисунок 12 - Политика конфиденциальности**

* Правила посещения. Содержит информацию о правилах поведения в котокафе, важных для комфорта посетителей и безопасности животных. Представляет собой структурированный текстовый контент. Не взаимодействует с бэкендом.



**Рисунок 13 - Правила посещения**

1. **Описание плагинов и модулей**

В процессе разработки серверной части веб-приложения на базе October CMS был задействован ряд плагинов, расширяющих стандартные возможности системы и реализующих специфический функционал проекта. Плагины позволяют добавлять новые типы данных, создавать пользовательские интерфейсы в административной панели и интегрировать сторонние сервисы.

Основными плагинами, использованными в проекте, являются:

* **RainLab.User**: Данный плагин предоставляет базовую функциональность для управления пользователями. Он позволил реализовать процессы регистрации новых пользователей, аутентификацию (вход в систему) и управление профилями пользователей через административную панель October CMS. В рамках проекта использовался API этого плагина для взаимодействия с системой пользователей из фронтенд-приложения. Стандартные возможности плагина были использованы без существенных модификаций.
* **RainLab.Builder:** Этот мощный плагин значительно упрощает процесс разработки для October CMS. Он предоставляет графический интерфейс и инструменты для быстрого создания основных структурных элементов плагина: моделей данных (связанных с таблицами в базе данных), миграций для управления схемой базы данных, контроллеров для обработки запросов и формирования страниц административной панели, а также пунктов меню для доступа к созданным разделам в админке. RainLab.Builder активно использовался для создания всех сущностей данных, описанных в разделе 4 (Product, Orders, Bookings и др.), и формирования интерфейса для их управления в административной части CMS.
* **Teiwi.Cafe:** Это кастомный плагин, разработанный специально для данного проекта котокафе. Он объединяет в себе всю специфическую логику и структуру данных, созданную с помощью RainLab.Builder и вручную написанный код, связанный с уникальными функциями кафе. В состав плагина Teiwi.Cafe входят модели данных для всех ключевых сущностей (меню, заказы, бронирования, кошки и т.д.), миграции для создания соответствующих таблиц в базе данных, а также контроллеры и определения меню административной панели для управления этими данными. Функционал обработки бизнес-логики (например, создание заказа, обработка бронирования) реализован на бэкенде в рамках этого плагина, а его представление и взаимодействие с пользователем осуществляется на фронтенде. По сути, данный плагин является ядром серверной части приложения, отвечающим за работу с данными и логикой котокафе.

Помимо перечисленных, другие сторонние плагины October CMS в проекте не использовались. Вся уникальная функциональность реализована в рамках кастомного плагина Teiwi.Cafe.

1. **Описание тестирования**

Тестирование разработанного веб-приложения является неотъемлемой частью процесса разработки, направленной на выявление и устранение ошибок, а также обеспечение соответствия функциональным и нефункциональным требованиям. В рамках работы над проектом котокафе "Teiwi" проводились различные виды тестирования для оценки работоспособности как серверной (October CMS, API), так и клиентской (Vue.js) частей приложения.

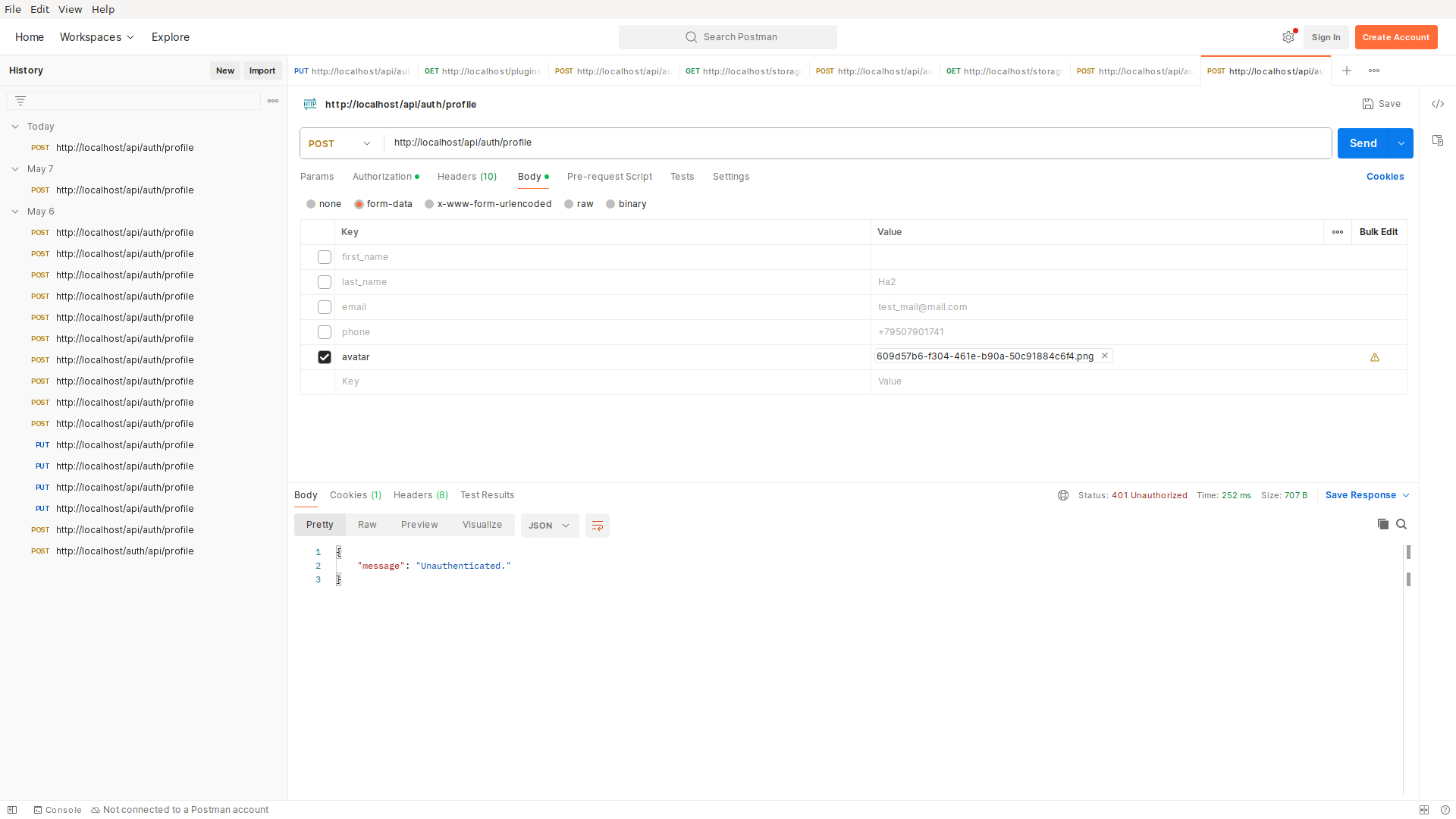
Основные направления тестирования включали:

* **Функциональное тестирование:** Проверка корректности работы всех основных функций приложения с точки зрения конечного пользователя. Тестировались ключевые пользовательские сценарии:
  + Регистрация и авторизация пользователей.
  + Просмотр меню кафе, каталога кошек и записей блога.
  + Поиск по сайту с различными параметрами.
  + Добавление блюд в корзину и управление ее содержимым.
  + Оформление заказа на доставку еды (заполнение данных, выбор адреса).
  + Процесс бронирования столика (выбор даты, времени, количества персон).
  + Оставление отзывов (если функционал был предусмотрен).
  + Работа с личным кабинетом: просмотр и редактирование профиля, управление адресами доставки, просмотр истории заказов.

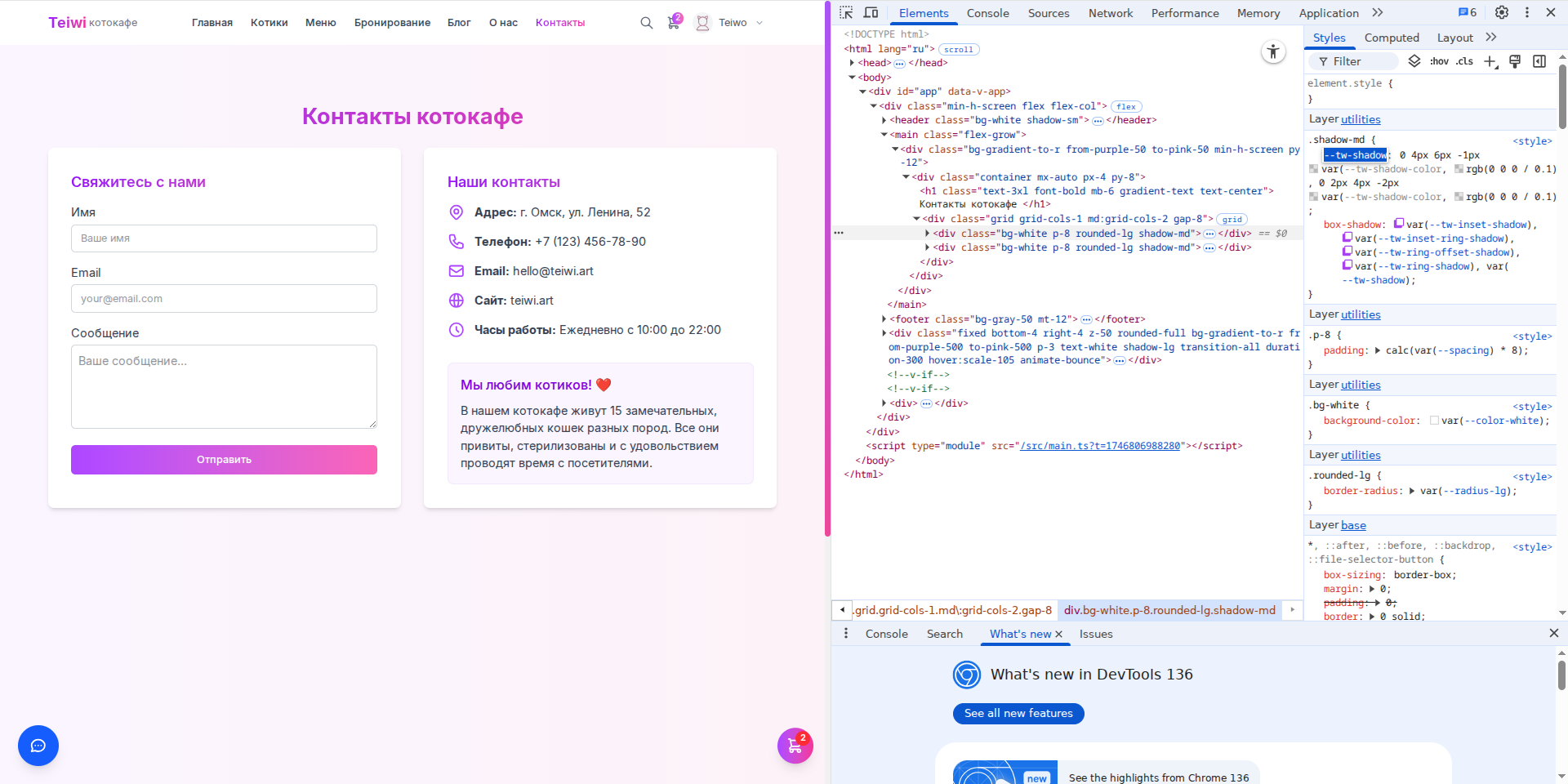
Проверялось как стандартное выполнение этих операций, так и обработка исключительных ситуаций (например, попытка заказать без авторизации, ввод некорректных данных).

* **Тестирование API:** Поскольку фронтенд и бэкенд взаимодействуют через REST API, проводилось тестирование эндпоинтов API, предоставляемых October CMS. Проверялась корректность:
  + Отправки запросов от фронтенда к бэкенду.
  + Получения ожидаемых данных в ответе от бэкенда (в формате JSON).
  + Обработки ошибок на стороне бэкенда и возврата соответствующих HTTP-кодов состояния (например, 200 OK, 400 Bad Request, 401 Unauthorized, 404 Not Found, 422 Unprocessable Entity, 500 Internal Server Error).

Для проверки отдельных API-запросов могли использоваться инструменты разработчика браузера (вкладка Network) или специализированные утилиты, имитирующие запросы к API.

**Рисунок 14 - Тестирование работы API в Postman**

* **Тестирование верстки и адаптивности (Frontend):** Оценка корректного отображения пользовательского интерфейса на различных устройствах и размерах экранов, а также в разных веб-браузерах. Проверялось, что элементы страницы не накладываются друг на друга, сохраняется читабельность текста и удобство использования независимо от условий просмотра. Использовались инструменты разработчика в браузере для эмуляции различных устройств.

**Рисунок 15 - Тестирование верстки пользовательского интерфейса**

* **Тестирование пользовательского интерфейса и удобства (UI/UX):** Оценка общего впечатления от использования сайта, интуитивности интерфейса и легкости выполнения основных действий. Проверялась понятность навигации, удобство заполнения форм, логичность переходов между страницами.
* **Базовое тестирование безопасности:** Выполнялись простые проверки, направленные на выявление очевидных уязвимостей, таких как:
  + Проверка ввода некорректных или потенциально опасных данных в поля форм для оценки реакции системы на валидацию.
  + Попытки доступа к защищенным разделам сайта (личный кабинет, админ-панель) без предварительной авторизации.

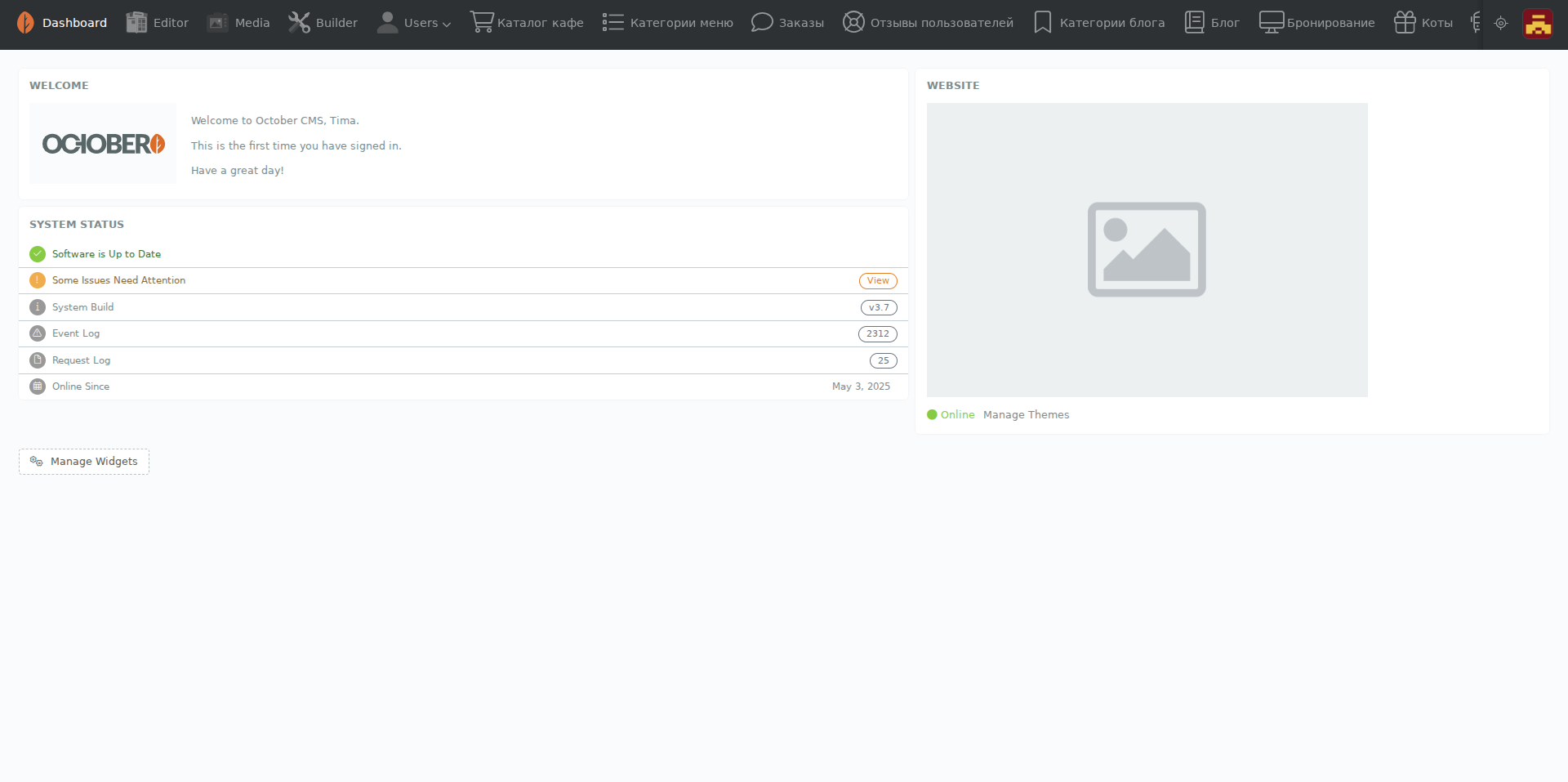
В ходе тестирования все выявленные недочеты и ошибки фиксировались и оперативно устранялись. Процесс тестирования проводился итеративно, по мере готовности отдельных модулей и функций приложения.

1. **Руководство администратора**

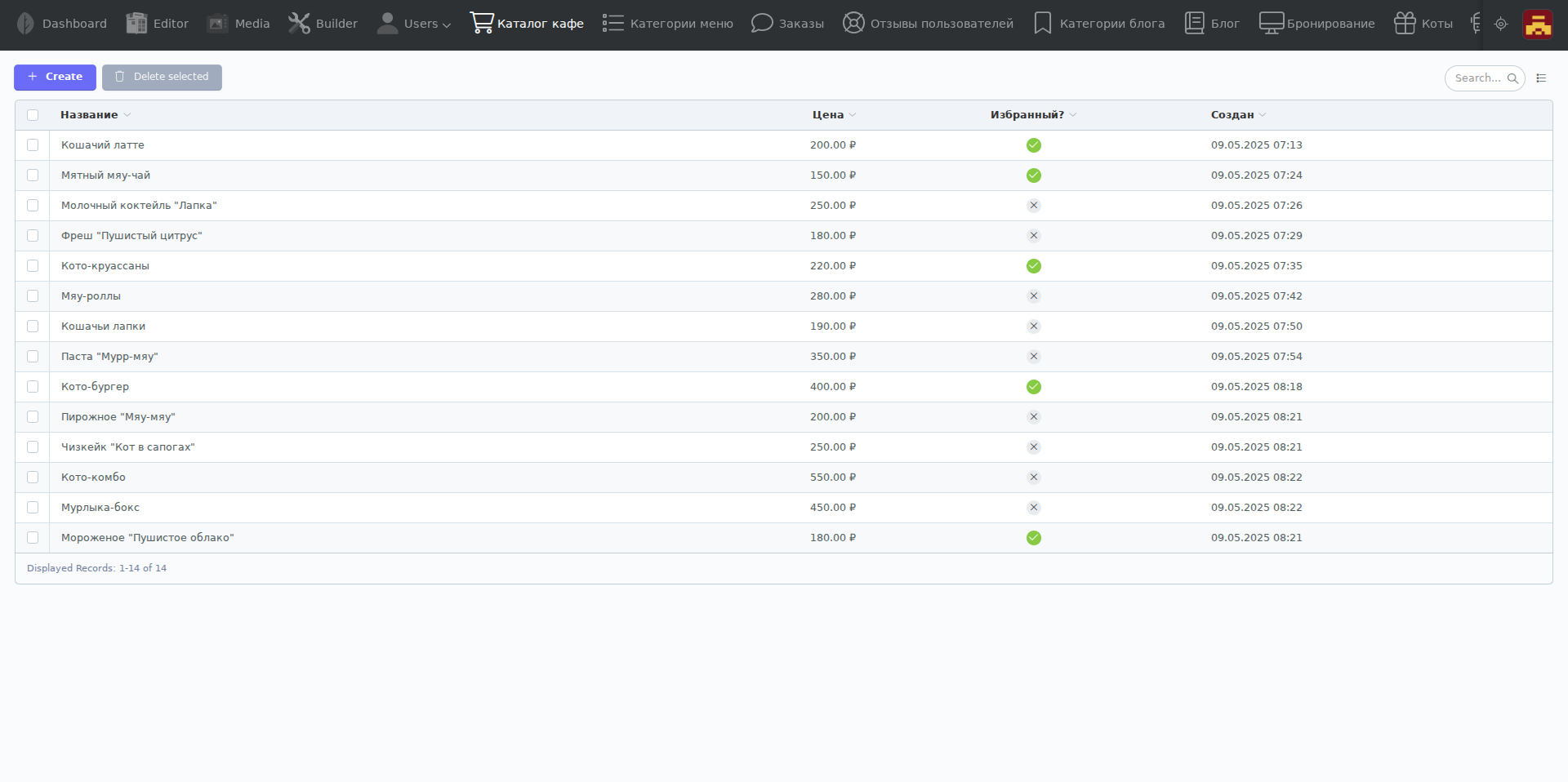
Управление контентом и данными веб-приложения котокафе "Teiwi" осуществляется через удобную административную панель October CMS. Доступ к панели управления администратор получает, перейдя по адресу ***teiwi.art/admin*** и пройдя процедуру аутентификации.

Административная панель предоставляет интуитивно понятный интерфейс, с помощью которого можно выполнять все необходимые операции без прямого взаимодействия с кодом или базой данных. Основные разделы меню административной панели, используемые для управления контентом котокафе, включают:

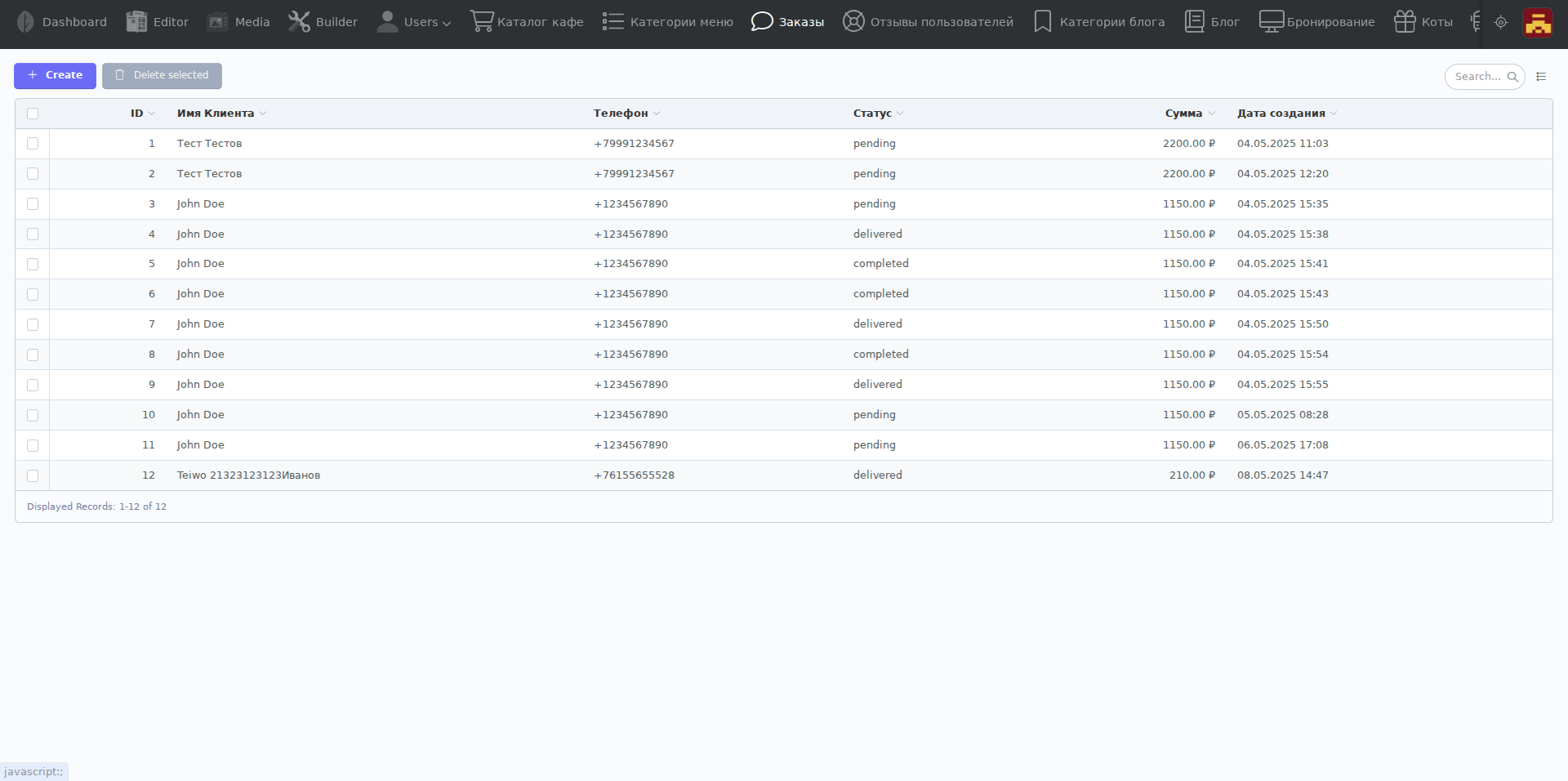
* **Users:** Управление учетными записями зарегистрированных пользователей сайта. Здесь администратор может просматривать список пользователей, их данные, а также при необходимости блокировать или удалять аккаунты.
* **Каталог Меню:** Полное управление ассортиментом кафе. Администратор имеет возможность добавлять новые блюда и напитки в меню, редактировать информацию о существующих позициях (название, описание, цена, изображение, принадлежность к категории), а также удалять неактуальные пункты.
* **Категории Меню:** Управление категориями, по которым фильтруются блюда в пользовательской части сайта (например, "Напитки", "Десерты", "Основные блюда"). Администратор может создавать, редактировать и удалять категории, а также привязывать к ним пункты меню.
* **Заказы:** Раздел для просмотра и обработки заказов, сделанных пользователями. Здесь отображается список всех заказов с деталями: состав заказа, информация о клиенте, адрес доставки, выбранный способ оплаты, текущий статус. Администратор отслеживает выполнение заказов и изменяет их статус (например, "Принят", "Готовится", "В пути", "Выполнен").
* **Отзывы пользователей:** Управление отзывами, оставленными посетителями. Администратор может просматривать опубликованные отзывы, одобрять новые перед публикацией или удалять неприемлемые комментарии.
* **Категории блога и Блог:** Управление контентом раздела новостей или статей о жизни кафе. Администратор создает и редактирует категории для статей, а также добавляет, изменяет и удаляет сами публикации в блоге.
* **Бронирование:** Раздел для работы с заявками на бронирование столиков или мест в котокафе. Администратор просматривает все входящие заявки, проверяет их детали (дата, время, количество гостей) и управляет статусом брони (например, "Подтверждено", "Отклонено", "Завершено").
* **Коты:** Управление информацией о кошках, проживающих в кафе. Администратор может добавлять данные о новых обитателях (имя, порода, описание), загружать их фотографии, редактировать существующую информацию.
* **Адреса пользователей:** Просмотр списка адресов доставки, сохраненных пользователями в своих профилях.

Благодаря структуре October CMS и инструментам, предоставленным плагинами RainLab.Builder и Teiwi.Cafe, администрирование сайта не требует навыков программирования. Все изменения, внесенные через административную панель, автоматически применяются и взаимодействуют с данными в базе данных, а также доступны для отображения на фронтенде через API.

**Рисунок 16 - Главная страница административной панели**



**Рисунок 17 - Управление меню в административной панели**



**Рисунок 18 - Управление заказами в административной панели**

**ЗАКЛЮЧЕНИЕ**

В рамках учебной практики УП.08.02 "Практика программирования" была выполнена работа по разработке веб-приложения для котокафе "Teiwi". Проект представляет собой современное веб-приложение с разделенной архитектурой (decoupled), состоящее из серверной части на базе October CMS, используемой как Headless CMS, и клиентской части (фронтенда), разработанной с применением технологий Node.js, Vue.js и библиотеки стилей Tailwind CSS.

* В ходе выполнения практики были достигнуты поставленные цели и задачи, включая формирование и закрепление практических навыков в области веб-разработки. Были освоены ключевые аспекты работы с:
* Системой управления контентом October CMS, в частности, ее использование в качестве Headless CMS для структурированного хранения данных и предоставления их через API.
* Современным фронтенд-фреймворком Vue.js, включая построение пользовательского интерфейса с использованием компонентов, управление состоянием приложения (Pinia) и взаимодействие с бэкендом.
* Библиотекой стилей Tailwind CSS для быстрой и адаптивной верстки пользовательского интерфейса.
* Принципами построения REST API для обеспечения взаимодействия между серверной и клиентской частями приложения.

В ходе практики была разработана концептуальная модель сайта и структура ролей пользователей, спроектирована структура данных в October CMS, реализованы ключевые сущности (меню, заказы, бронирования, информация о котах и др.) с помощью плагина RainLab.Builder и кастомного плагина Teiwi.Cafe. Были спроектированы и сверстаны основные страницы фронтенд-приложения, реализована их логика и взаимодействие с разработанным API. Проведено тестирование разработанных компонентов и функций.

Несмотря на то, что проект на момент завершения практики может требовать дальнейшего наполнения контентом и доработки отдельных функций, основная архитектура приложения реализована, и ключевые механизмы взаимодействия между фронтендом и бэкендом налажены.

Практика позволила получить ценный опыт работы с актуальным стеком технологий, применяемым в современной веб-разработке, углубить понимание принципов построения распределенных веб-приложений и закрепить навыки самостоятельной работы над проектом. В дальнейшем возможна доработка функционала бронирования и заказов, внедрение онлайн-оплаты, расширение раздела блога и улучшение пользовательского интерфейса.

Прохождение практики способствовало формированию профессиональных компетенций, связанных с разработкой дизайна веб-приложений, и стало важным шагом в профессиональном становлении.

**ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ ИНФОРМАЦИИ**

*Программное обеспечение:*

1. October CMS 3.7
2. Node.js 20
3. Vue 3
4. Tailwind CSS 4
5. VS Code
6. PostgreSQL
7. Caddy
8. Microsoft Visio
9. Git

*Интернет ресурсы:*

1. October CMS Documentation [Электронный ресурс]. – Режим доступа: https://docs.octobercms.com/ (Дата обращения: 04.05.2025)
2. rainlab/user-plugin: Front-end user management. - GitHub [Электронный ресурс]. – Режим доступа: https://github.com/rainlab/user-plugin (Дата обращения: 04.05.2025)
3. Builder plugin - October CMS [Электронный ресурс]. – Режим доступа: https://octobercms.com/plugin/rainlab-builder (Дата обращения: 04.05.2025)
4. Getting started with Vue - Learn web development | MDN [Электронный ресурс]. – Режим доступа: https://developer.mozilla.org/en-US/docs/Learn\_web\_development/Core/Frameworks\_libraries/Vue\_getting\_started (Дата обращения: 04.05.2025)
5. Vue 1 documentation - DevDocs [Электронный ресурс]. – Режим доступа: https://devdocs.io/vue~1/ (Дата обращения: 04.05.2025)
6. Express/Node introduction - Learn web development | MDN [Электронный ресурс]. – Режим доступа: https://developer.mozilla.org/en-US/docs/Learn\_web\_development/Extensions/Server-side/Express\_Nodejs/Introduction (Дата обращения: 04.05.2025)
7. Node.js tutorial in Visual Studio Code [Электронный ресурс]. – Режим доступа: https://code.visualstudio.com/docs/nodejs/nodejs-tutorial (Дата обращения: 04.05.2025)
8. Tailwind CSS documentation - DevDocs [Электронный ресурс]. – Режим доступа: https://devdocs.io/tailwindcss/ (Дата обращения: 04.05.2025)
9. Guides: Tailwind CSS | Next.js [Электронный ресурс]. – Режим доступа: https://nextjs.org/docs/app/guides/tailwind-css (Дата обращения: 04.05.2025)
10. The Ultimate Guide for Using Vue.js with Laravel - Vue School Articles [Электронный ресурс]. – Режим доступа: https://vueschool.io/articles/vuejs-tutorials/the-ultimate-guide-for-using-vue-js-with-laravel/ (Дата обращения: 04.05.2025)