

Частное профессиональное образовательное учреждение  
«Серпуховский городской открытый колледж»

# Курсовая работа

по дисциплине/профессиональному модулю:

---

**Тема:**

---

Выполнил:  
ИС-03-2021 Со  
Сердюк С.С.

Серпухов, 2025

<b>ВВЕДЕНИЕ.....</b>	<b>3</b>
<b>1. Теоретическая часть.....</b>	<b>5</b>
1.1. Роль веб-сайта в деятельности торговой организации.....	5
1.2. Обоснование выбора технологий.....	7
1.3. Архитектурный подход к разработке.....	8
1.4. Обеспечение безопасности веб-приложений.....	10
1.5. SEO и индексация сайта.....	12
1.6. Сравнение хостингов.....	13
<b>2. Практическая часть.....</b>	<b>16</b>
2.1. Общая структура проекта.....	16
2.2. Главные разделы и страницы сайта.....	17
2.3. Реализация формы заявки.....	20
2.4. Панель администратора.....	22
2.5. Реализация безопасности.....	24
2.6. Генерация sitemap.xml и robots.txt.....	25
2.7. Размещение сайта на хостинге.....	25
2.8. Интеграция с Яндекс.Вебмастер.....	28
<b>Заключение.....</b>	<b>31</b>
<b>Библиографический список.....</b>	<b>32</b>
<b>Приложение А.....</b>	<b>33</b>

## **ВВЕДЕНИЕ**

В современных условиях цифровизации и роста конкуренции среди торговых организаций наличие собственного веб-сайта становится необходимым элементом эффективного ведения бизнеса. Электронное представительство позволяет расширить клиентскую аудиторию, представить ассортимент товаров в доступной форме, автоматизировать часть взаимодействия с потребителями, а также повысить имидж компании. Особенно важным является наличие канала обратной связи, позволяющего клиентам направлять запросы напрямую через сайт.

В качестве предметной области данной работы рассматривается торговая организация — предприятие, специализирующееся на продаже бытовой, офисной и сопутствующей техники. Для моделирования практической части был создан проект компании ООО «ТехноМир», осуществляющей деятельность на рынке более 10 лет. Организация ориентирована на предоставление широкого ассортимента современной техники по доступным ценам, сопровождая продажи высоким уровнем сервиса и индивидуальным подходом к заказу. Основные товарные категории включают бытовую технику, химию, сантехнику, посуду и электронику.

Целью курсовой работы является разработка и запуск функционального веб-сайта для торговой организации с учётом требований безопасности, доступности и удобства взаимодействия. Сайт должен быть размещён на бесплатном хостинге, реализовывать форму обратной связи с сохранением данных в базу, содержать защищённый административный раздел для обработки заявок, а также включать SEO-компоненты: файл robots.txt, карту сайта sitemap.xml и подключение к сервису «Яндекс.Вебмастер».

Для достижения поставленной цели сформулированы следующие задачи:

Провести анализ технологий и архитектурных решений, подходящих для реализации проекта на бесплатном хостинге;

Обосновать выбор хостинг-платформы с учётом технических ограничений и потребностей проекта;

Разработать структуру веб-приложения с применением архитектурного шаблона MVC;

Реализовать механизм защиты данных и авторизации с использованием современных подходов (JWT, CSRF, rate-limit);

Создать SEO-файлы и обеспечить соответствие требованиям индексации поисковыми системами;

Развернуть проект на сервере и осуществить попытку интеграции с Яндекс.Вебмастером.

Настоящая работа охватывает как теоретические обоснования применённых решений, так и практическую реализацию сайта, ориентированного на задачи коммерческой деятельности.

# **1. Теоретическая часть**

## **1.1. Роль веб-сайта в деятельности торговой организации**

Торговая организация представляет собой юридическое лицо, осуществляющее реализацию товаров в розницу или оптом, ориентируясь на конечного потребителя, других предприятий или коммерческих партнёров. Торговая деятельность предполагает наличие товарного ассортимента, выполнение операций купли-продажи и стремление удовлетворить существующий потребительский спрос. В условиях цифровизации экономики и усиления конкуренции классические формы торговли всё чаще трансформируются в гибридные или полностью электронные, где значительную роль начинают играть веб-технологии.

Наличие собственного сайта становится важнейшим элементом цифровой инфраструктуры торгового предприятия. Такой сайт позволяет предоставлять клиентам круглосуточный доступ к информации о товарах и услугах, упрощает взаимодействие за счёт форм онлайн-обращений, повышает доверие к бренду за счёт прозрачности, демонстрирует технологичность компании и способствует формированию позитивного имиджа. Кроме того, сайт может служить каналом для приёма заказов, демонстрации ассортимента, публикации акций и новостей, а также точкой входа для дальнейшей цифровой трансформации бизнеса. Его наличие позволяет использовать инструменты веб-аналитики и SEO-продвижения, усиливая конкурентные позиции компании в интернете.

Для целей данной курсовой работы в качестве предметной области была выбрана модельная компания — общество с ограниченной ответственностью «ТехноМир». Организация позиционируется как универсальный поставщик бытовой и офисной техники, а также сопутствующих товаров, включая сантехнику, электронику, кухонные принадлежности и профессиональное оборудование. Компания имеет более десяти лет опыта на рынке и

zareкомендовала себя как надёжный партнёр для частных клиентов и малого бизнеса. Основной миссией «ТехноМира» является обеспечение населения и организаций качественной техникой по доступным ценам при высоком уровне клиентского обслуживания. География клиентов включает как индивидуальных покупателей, так и корпоративных заказчиков. Ассортимент включает широкий выбор бытовой техники (в том числе холодильники, стиральные машины, микроволновые печи), сантехнические товары и элементы наполнения кухонь, электронику и аксессуары, профессиональную технику, а также посуду и бытовую химию.

В качестве ключевых конкурентных преимуществ компании можно выделить актуальный и постоянно пополняющийся ассортимент, ориентацию на клиента с быстрым реагированием на запросы, гибкую ценовую политику, наличие собственного склада, отлаженную логистику и стабильную репутацию на рынке.

Разработка веб-сайта для такой торговой организации призвана выполнять несколько взаимосвязанных задач. Сайт должен обеспечивать информирование клиентов о товарах, условиях покупки, новостях и акциях; представлять каталог продукции в структурированном и доступном виде; предоставлять удобные формы для отправки заявок; обеспечивать доступ к административной панели, где сотрудники могут просматривать и обрабатывать обращения; соответствовать базовым требованиям к безопасности и стабильности, предъявляемым к публичным веб-приложениям.

Таким образом, веб-сайт выполняет не только представительскую функцию, но и становится инструментом цифровой автоматизации бизнес-процессов. В случае ООО «ТехноМир» проект сайта направлен на демонстрацию архитектурных и технических решений, применимых в условиях современной розничной торговли.

## **1.2. Обоснование выбора технологий**

Разработка веб-сайта торговой организации осуществлялась с применением технологий, сочетающих функциональную достаточность, доступность для обучения и совместимость с условиями бесплатного хостинга. При проектировании учитывались как ограничения инфраструктуры, так и необходимость обеспечения устойчивой и расширяемой архитектуры. Основной упор был сделан на использование проверенных и широко распространённых решений, не требующих дорогостоящих серверных ресурсов или специализированных платформ.

В качестве серверной технологии был выбран язык программирования PHP. Он остаётся одним из наиболее популярных инструментов в сфере веб-разработки, благодаря низкому порогу вхождения, широкой поддержке со стороны хостинг-провайдеров и активному сообществу. PHP позволяет эффективно обрабатывать запросы, генерировать динамические страницы и взаимодействовать с базой данных, что делает его оптимальным выбором для сайтов с пользовательским вводом и административной панелью. Отказ от использования фреймворков был осознанным решением, направленным на демонстрацию базовых принципов архитектуры и ручную реализацию всех ключевых компонентов системы.

В качестве системы управления данными была выбрана реляционная база данных MySQL. Она предоставляет надёжный механизм хранения структурированной информации, поддерживает масштабируемые запросы и полностью совместима с PHP. Использование SQL позволило реализовать хранение заявок пользователей, данных учётных записей и других элементов бизнес-логики в форме связанных таблиц.

Для клиентской части использовались HTML и CSS в качестве базовых технологий разметки и стилизации интерфейса. Эти языки являются стандартом для создания визуальной структуры страниц и оформления контента. Для

ускорения разработки и обеспечения адаптивного дизайна применён CSS-фреймворк Bootstrap, который позволяет быстро реализовать современный внешний вид интерфейса и корректное отображение на различных устройствах.

Язык JavaScript был задействован для реализации интерактивного поведения элементов интерфейса. Его использование позволяет реагировать на действия пользователя без необходимости перезагрузки страниц. Технология AJAX (Asynchronous JavaScript and XML) дополняла JavaScript и использовалась для отправки запросов к серверу в фоновом режиме, что особенно актуально при работе с фильтрами в каталоге товаров и отправке форм без полной перезагрузки страницы.

Выбор перечисленного технологического стека обусловлен его широкой распространённостью, документированностью, совместимостью с бесплатными хостингами (в частности, InfinityFree) и возможностью полной реализации всех необходимых функций — от отображения интерфейса до взаимодействия с базой данных и реализации защиты пользовательских данных. Эти технологии обеспечивают надёжную и контролируемую основу для создания учебного проекта, близкого по архитектуре к реальному коммерческому веб-приложению.

### **1.3. Архитектурный подход к разработке**

Для обеспечения структурированности, масштабируемости и удобства сопровождения веб-приложения в рамках проекта была применена архитектурная модель MVC (Model–View–Controller). Этот подход широко используется в веб-разработке и предполагает разделение логики приложения на три независимых компонента: модель, представление и контроллер. Каждая из этих частей отвечает за строго определённый круг задач, что упрощает отладку, тестирование и развитие проекта.

Компонент *Model* реализует работу с данными, включая взаимодействие с базой данных, бизнес-логику и валидацию входной информации. В контексте



разработанного веб-сайта модели обеспечивают сохранение и извлечение информации о заявках, пользователях, а также реализацию логики авторизации и ограничения доступа. Компонент *View* отвечает за генерацию HTML-страниц, предоставляемых пользователю. Представления инкапсулируют визуальную часть интерфейса и получают данные от контроллеров для отображения. Компонент *Controller* выступает в роли посредника между моделью и представлением, обрабатывает входящие HTTP-запросы, управляет маршрутизацией и определяет логику обработки пользовательских действий.

Применение шаблона MVC позволило избежать смешения логики обработки данных и представления, что является частой проблемой в проектах с монолитной структурой. Такой подход также облегчает внедрение новых функций и позволяет различным участкам приложения развиваться независимо друг от друга.

Для организации системы авторизации в проекте использовалась технология JWT (JSON Web Token). Этот механизм обеспечивает безопасную и гибкую передачу информации об аутентификации между клиентом и сервером. После успешного ввода логина и пароля сервер генерирует токен, содержащий информацию о пользователе и срок действия, и передаёт его клиенту. Впоследствии клиент использует этот токен для подтверждения своей личности при доступе к защищённым разделам сайта. Такой подход позволяет отказаться от традиционных PHP-сессий, а также упростить масштабирование и реализацию REST-совместимых API в будущем.

Дополнительно в проекте реализована защита от межсайтовых подделок запросов (CSRF — Cross-Site Request Forgery). Эта уязвимость возникает, когда злоумышленник инициирует нежелательные действия от имени авторизованного пользователя. Для предотвращения подобных атак в проекте используется механизм CSRF-токенов: при генерации каждой формы сервер добавляет скрытое поле с уникальным токеном, связанным с текущей сессией.

При отправке формы сервер проверяет наличие и корректность этого токена, и в случае несоответствия отклоняет запрос.

Таким образом, архитектура проекта сочетает проверенные подходы и современные механизмы защиты, что позволяет не только удовлетворить функциональные требования, но и обеспечить высокий уровень безопасности и структурной целостности веб-приложения. Применение MVC, JWT и CSRF на практике формирует основу для создания надёжного, расширяемого и поддерживаемого сайта торговой организации.

#### **1.4. Обеспечение безопасности веб-приложений**

Веб-приложения, работающие с пользовательскими данными и предоставляющие административный функционал, неизбежно сталкиваются с рисками, связанными с угрозами безопасности. Наиболее распространёнными типами атак являются SQL-инъекции, межсайтовое выполнение скриптов (XSS), подделка межсайтовых запросов (CSRF), подбор паролей, автоматическая отправка форм и другие действия, направленные на несанкционированное вмешательство в работу сайта или компрометацию данных. В условиях публичного размещения сайта торговой организации важно обеспечить защиту всех его компонентов, даже при демонстрационном характере проекта.

Одним из ключевых элементов защиты является механизм аутентификации и разграничения доступа. В рамках данного проекта реализована авторизация администратора, основанная на передаче зашифрованного JWT-токена, который создаётся после успешного ввода логина и пароля. При этом пароли хранятся в базе данных в виде хешей, что исключает возможность их восстановления при компрометации базы. Таким образом, даже при утечке данных реальный пароль остаётся недоступным.

Для защиты от массовых автоматизированных запросов, атак перебора и спама применяется механизм ограничения частоты запросов (rate-limiting). Он реализован на серверной стороне и отслеживает количество обращений от одного IP-адреса в определённый временной интервал. В случае превышения заданного порога система временно блокирует доступ, предотвращая нагрузку на сервер и повышая устойчивость сайта к простейшим формам атак.

Безопасность ввода данных обеспечивается за счёт фильтрации и валидации информации, поступающей от пользователей. Все поля форм проверяются на корректность формата, допустимую длину и наличие потенциально опасных символов. Вывод текстов на страницы сопровождается экранированием, что исключает возможность внедрения вредоносных скриптов (XSS). Кроме того, при работе с базой данных используются подготовленные выражения (prepared statements), что предотвращает SQL-инъекции независимо от формы введённых данных.

Отдельное внимание уделено защите от CSRF-атак. Каждая форма, способная вызвать изменения на сервере (например, отправка заявки или вход в админ-панель), снабжается уникальным токеном, привязанным к текущей сессии. Этот токен проверяется при отправке формы, и в случае его отсутствия или несоответствия запрос отклоняется. Такой подход надёжно защищает от подделки запросов через сторонние сайты.

В рамках реализации также применяется система логирования действий. В журнал записываются ключевые события, включая попытки входа в систему, отправки форм, обращения к защищённым маршрутам, а также возникающие ошибки. Это позволяет оперативно отслеживать подозрительную активность, проводить аудит безопасности и ускорять отладку при выявлении сбоев.

Реализация вышеперечисленных мер защиты позволяет обеспечить надёжную и устойчивую работу веб-приложения, соответствующую требованиям к безопасности, предъявляемым к публичным онлайн-сервисам. Даже в условиях

учебного проекта были применены подходы, аналогичные тем, что используются в коммерческой практике. Это делает разработанный сайт не только функциональным, но и устойчивым к распространённым угрозам.

### **1.5. SEO и индексация сайта**

Эффективность присутствия компании в интернете во многом определяется тем, насколько её сайт виден для поисковых систем. Для этого необходимо обеспечить корректную индексацию всех ключевых страниц ресурса, чтобы они отображались в результатах поиска по релевантным запросам. Веб-сайт, не проиндексированный поисковыми роботами, фактически остаётся невидимым для большинства потенциальных клиентов. В контексте торговой организации это означает потерю части трафика и недополученную прибыль.

Для управления процессом индексации в проектах веб-разработки применяются два ключевых файла — robots.txt и sitemap.xml. Оба размещаются в корне сайта и служат для взаимодействия с поисковыми системами, но выполняют различные функции.

Файл robots.txt представляет собой текстовый документ, содержащий директивы для поисковых роботов. Он определяет, какие разделы сайта разрешены или запрещены к сканированию. Это позволяет исключить из индексации технические маршруты, административные панели, временные страницы или другие ресурсы, не предназначенные для публичного доступа. Кроме того, в robots.txt обычно указывается путь к карте сайта (sitemap.xml), чтобы поисковые системы могли быстро найти её и воспользоваться структурированной информацией о страницах ресурса.

Файл sitemap.xml — это XML-документ, содержащий список URL-адресов сайта, предназначенных для индексации. Для каждой страницы может быть указана дата последнего обновления, желаемая частота сканирования и относительный приоритет. Это позволяет поисковым системам эффективно

обходить сайт, даже если некоторая часть страниц не имеет внутренних ссылок. Наличие актуальной карты сайта особенно важно для крупных или динамически обновляемых сайтов, таких как интернет-каталоги. В проекте «ТехноМир» sitemap.xml генерируется автоматически, включая как статические страницы (например, "О компании", "Как купить"), так и карточки товаров, создаваемые на основе данных из базы.

Для контроля индексации и получения статистики по сайту используются инструменты веб-мастеров, такие как «Яндекс.Вебмастер» или Google Search Console. Эти сервисы позволяют проверить корректность файлов robots.txt и sitemap.xml, отправить сайт на переобход, отследить ошибки сканирования, проанализировать поисковые запросы и общее состояние индексации. Подтверждение прав на сайт в данных системах открывает доступ к широкому спектру аналитических и технических функций, важных для поддержки и продвижения веб-проекта.

Таким образом, обеспечение корректной SEO-настройки сайта — это необходимый этап даже для учебного проекта. Это демонстрирует готовность ресурса к реальной работе в условиях конкурентной цифровой среды и позволяет проверить его соответствие базовым требованиям поисковой оптимизации.

## **1.6. Сравнение хостингов**

Одним из первых этапов при запуске веб-сайта является выбор хостинг-платформы — сервера, на котором размещаются файлы сайта, обрабатываются HTTP-запросы и обеспечивается доступность ресурса в сети Интернет. В рамках курсового проекта было принято решение использовать бесплатный хостинг, что позволяет сократить затраты и сосредоточиться на реализации функционала без привязки к платной инфраструктуре. Вместе с тем, подобные платформы зачастую сопровождаются техническими ограничениями,

поэтому выбор конкретного провайдера осуществлялся на основании нескольких критериев.

Основные критерии оценки хостинга:

- поддержка PHP и MySQL;
- возможность подключения собственного .htaccess;
- стабильность работы и скорость отклика сервера;
- отсутствие принудительной рекламы;
- доступ к панели управления и протоколам FTP;
- возможность верификации сайта в Яндекс.Вебмастере.

В таблице ниже представлено сравнение протестированных бесплатных хостинг-платформ:

Хостинг	Поддержка PHP	База данных	.htaccess	Верификация Яндекс	Удобство / стабильность
InfinityFree	Да	MySQL	Полная поддержка	Нет (403/404 ошибки)	Стабильный, без рекламы, VistaPanel
000webhost	Да	SQLite / MySQL	Частичная	Нет (robots.txt закрыт)	Ограничения, баннеры
ByetHost	Да	MySQL	Частичная	Нет (ошибки DNS/SSL)	Перебои, сложная панель
AwardSpace	Да (устаревший)	MySQL	Частичная	Нет (403/404 ошибки)	Медленно, баги
GitHub Pages	Нет	Нет	Нет	Да (через HTML-зеркало)	Высокая скорость, без бэкенда

Несмотря на предпринятые попытки, найти бесплатный хостинг с полной совместимостью с системой Яндекс.Вебмастер не удалось. В каждом протестированном случае попытка размещения верификационного файла и его

последующей проверки заканчивалась ошибкой 403 или 404, либо файл игнорировался из-за политики платформы. В частности, возникали проблемы с доступностью файла для поискового робота даже при наличии прямой ссылки, а перенаправления через GitHub Pages с HTML-зеркалом также не дали результата.

Попытки продолжить подбор альтернативных платформ были прекращены по причине отсутствия времени: большинство хостингов требует длительного ожидания активации или появления домена в системе, а сайт при этом остаётся недоступным либо полностью неработоспособным. В условиях ограниченного срока выполнения проекта подобный подход оказался непрактичным.

В связи с этим, в качестве основной платформы был выбран хостинг InfinityFree — как наиболее сбалансированный по функциональности, стабильности и удобству. Он поддерживает все необходимые технологии, позволяет гибко настраивать серверное поведение через .htaccess, не накладывает навязчивой рекламы и предоставляет понятную панель управления. Несмотря на ограничение с Яндекс.Вебмастером, по совокупности технических характеристик он оказался оптимальным решением для реализации учебного проекта.

## 2. Практическая часть

### 2.1. Общая структура проекта

Разработанный сайт организован в виде многоуровневой файловой структуры, соответствующей парадигме MVC (Model-View-Controller). Это обеспечивает логическое разделение кода по ответственностям и упрощает сопровождение. Основной точкой входа в приложение является файл `index.php`, обрабатывающий все HTTP-запросы после маршрутизации через `.htaccess`.

Логика маршрутизации реализована в файле `web.php`, расположенном в каталоге `public/routes/`. Все запросы перенаправляются на этот файл через правило `RewriteRule` в `.htaccess`, после чего определяются маршруты и вызываются соответствующие контроллеры. В контроллерах содержится логика обработки запросов, а также вызов нужных моделей и представлений.

Модели, реализующие работу с базой данных, располагаются в каталоге `core/Models/`. Представления (HTML-шаблоны) и макеты страниц вынесены в папку `views/`, где находятся как страницы пользовательской части сайта, так и шаблоны административной панели.

Безопасность реализована с помощью отдельных компонентов, размещённых в `core/Security/`, включая модули аутентификации, CSRF-защиты, управления JWT-токенами и ограничения частоты запросов. Конфигурация подключения к базе данных вынесена в `core/Config/Database.php`, а параметры среды задаются в `.env`. Сокращенная схема структуры проекта представлена ниже.



```

├─ index.php           // Единая точка входа в приложение
├─ .htaccess          // Правила маршрутизации, защита и редиректы
├─ routes/
│   └─ web.php         // Все маршруты проекта, соответствующие REST-подходу
├─ core/
│   └─ Config/
│       └─ Database.php // Подключение к базе данных (PDO)
│   └─ Controllers/
│       └─ ...         // Логика обработки пользовательских запросов
│   └─ Models/
│       └─ ...         // Работа с базой данных (заявки, пользователи, товары)
│   └─ Security/
│       └─ ...         // RateLimiter, CSRFManager, JWTManager, AuthManager
│   └─ init.php        // Инициализация компонентов и автозагрузка
├─ utils/
│   └─ http.php        // Универсальный abort(), respond(), redirect()
│   └─ helpers.php     // Разные утилитарные функции
├─ views/
│   └─ layouts/        // Шаблоны: base, auth, error
│   └─ admin/          // Интерфейсы админ-панели
│   └─ pages/          // Статические публичные страницы (О нас, Как купить и пр.)
│   └─ request/        // Форма заявки и страница успеха
│   └─ errors/         // Кастомные страницы ошибок (401-503)
├─ robots.txt         // SEO-настройки для поисковых систем
├─ sitemap.xml         // Генератор карты сайта
└─ yandex_...html     // Верификационный файл Яндекс.Вебмастера

```

### схема структуры проекта

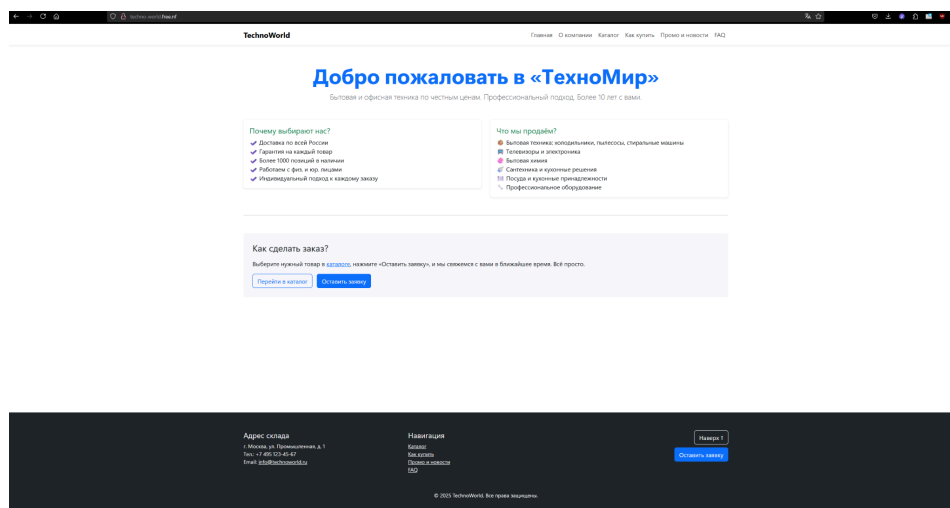
Полная схема находится в **Приложении А**.

Такое структурное построение обеспечивает масштабируемость и прозрачность, позволяя в дальнейшем легко расширять проект новыми модулями, разделами и сервисами.

## 2.2. Главные разделы и страницы сайта

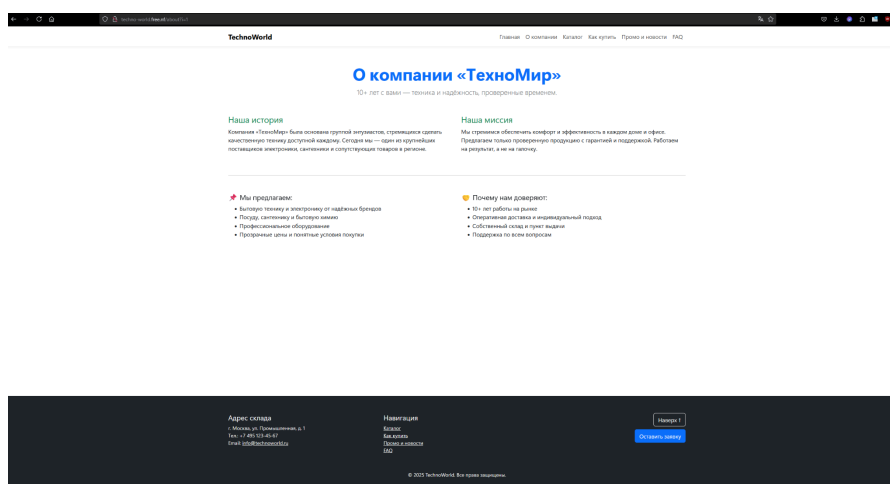
Структура пользовательской части сайта построена по классической схеме, обеспечивающей интуитивно понятную навигацию для клиента. Каждая из страниц реализует конкретную функцию в рамках взаимодействия с потенциальным покупателем и соответствует задачам, поставленным при проектировании веб-сайта для торговой организации.

Главная страница открывает сайт и содержит краткое описание компании, ссылки на ключевые разделы, а также визуальные элементы, отражающие тематику и ассортимент продукции. Это основная точка входа для большинства пользователей, поэтому она оптимизирована по скорости загрузки и содержит призыв к действию.



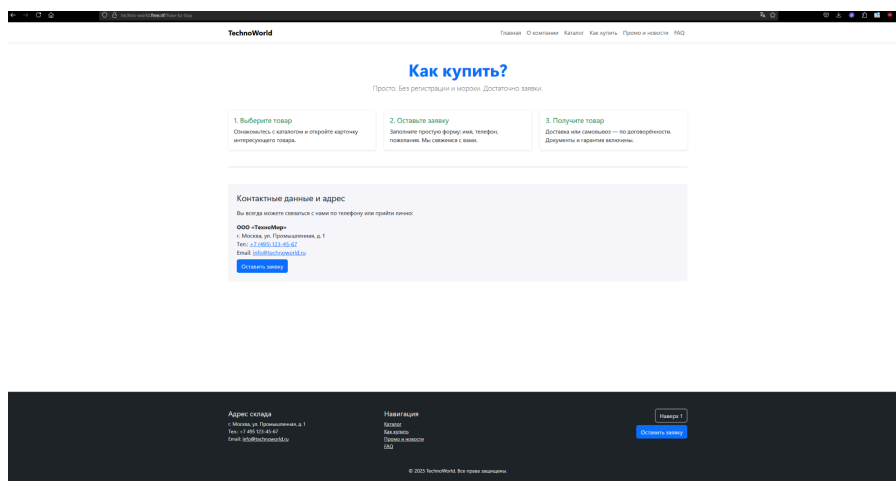
Изображение 1 - главная страница

Раздел «О компании» предоставляет текстовое описание истории ООО «ТехноМир», его миссии, направлений деятельности и конкурентных преимуществ. Здесь также расположены контактные данные и ссылки на другие страницы, повышающие доверие пользователей к ресурсу.



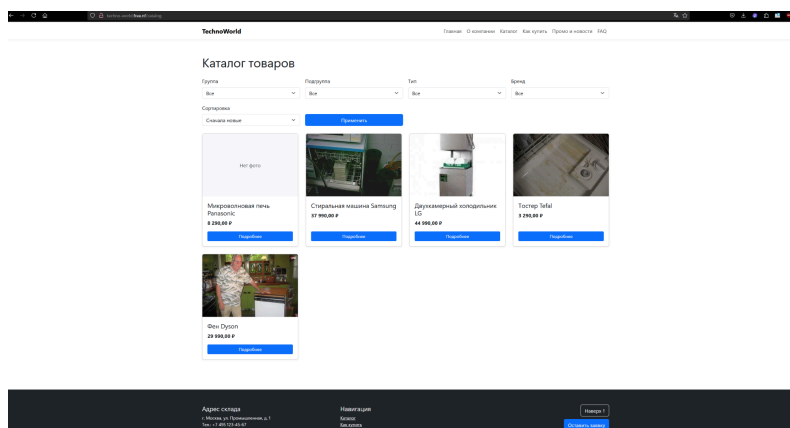
Изображение 2 - страница “О компании”

**Страница «Как купить»** служит справочным разделом, описывающим способы оформления заказа, возможные формы оплаты, доставку и условия возврата. Эта информация позволяет потенциальным клиентам получить полное представление о процессе покупки и исключить сомнения перед оформлением заявки.

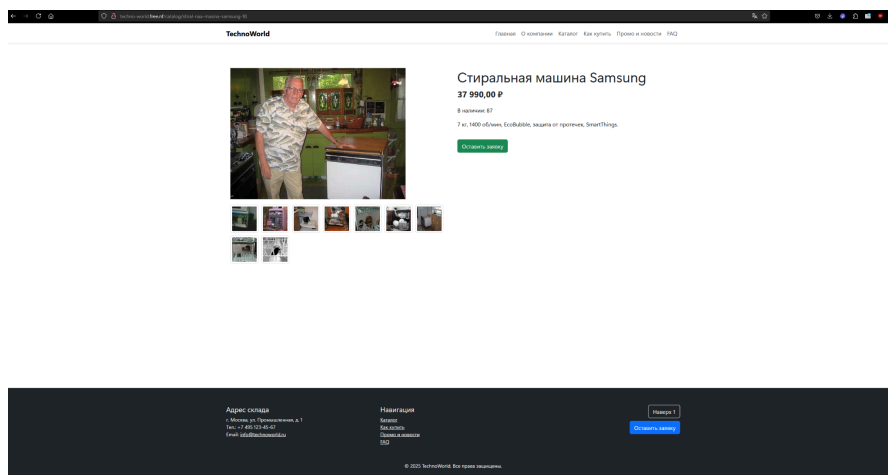


*Изображение 3 - страница “Как купить”*

**Каталог товаров** структурирован в виде карточек с описаниями и динамически формируется на основе данных из базы. Каждый товар представлен с наименованием, краткими характеристиками, изображением и уникальной ссылкой. Названия автоматически трансформируются в человеко-понятные URL с помощью функции slugify.



*Изображение 4 - каталог*



*Изображение 5 - карточка товара*

Каждая из перечисленных страниц разработана с учётом доступности, адаптивности и SEO-требований. Вёрстка выполнена с применением Bootstrap 5, что обеспечивает корректное отображение на различных устройствах. Контент загружается динамически, при необходимости применяются элементы AJAX (например, при фильтрации каталога), что позволяет избежать избыточной перезагрузки страниц.

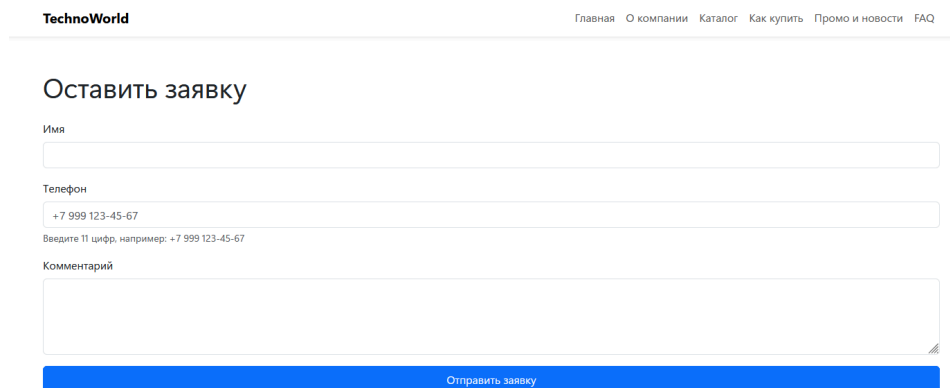
### **2.3. Реализация формы заявки**

Форма заявки является одним из ключевых интерактивных элементов сайта, обеспечивающих сбор клиентских обращений. Она размещена в публичной части проекта и доступна по маршруту /request.

Форма содержит следующие поля:

- имя клиента (name);
- контактный телефон;
- текст обращения;
- скрытое поле с CSRF-токеном;
- кнопка отправки.

Разметка формы реализована с использованием Bootstrap и минимального JavaScript. Отправка осуществляется через AJAX-запрос методом POST, что позволяет избежать полной перезагрузки страницы.



The screenshot shows a web form titled "Оставить заявку" (Leave a request) on the TechnoWorld website. The form includes fields for "Имя" (Name), "Телефон" (Phone) with a pre-filled number "+7 999 123-45-67" and a hint "Введите 11 цифр, например: +7 999 123-45-67", and a "Комментарий" (Comment) text area. A blue button at the bottom is labeled "Отправить заявку" (Send request). The website header includes the logo "TechnoWorld" and navigation links: "Главная", "О компании", "Каталог", "Как купить", "Промо и новости", and "FAQ".

*Изображение 6 - страница заявки*

При отправке данные проверяются на клиенте и сервере. После успешной обработки пользователь перенаправляется на страницу подтверждения (/request/success), где отображается сообщение о принятии заявки.

На сервере данные сохраняются в таблицу requests базы данных. Пример SQL-структуры таблицы:

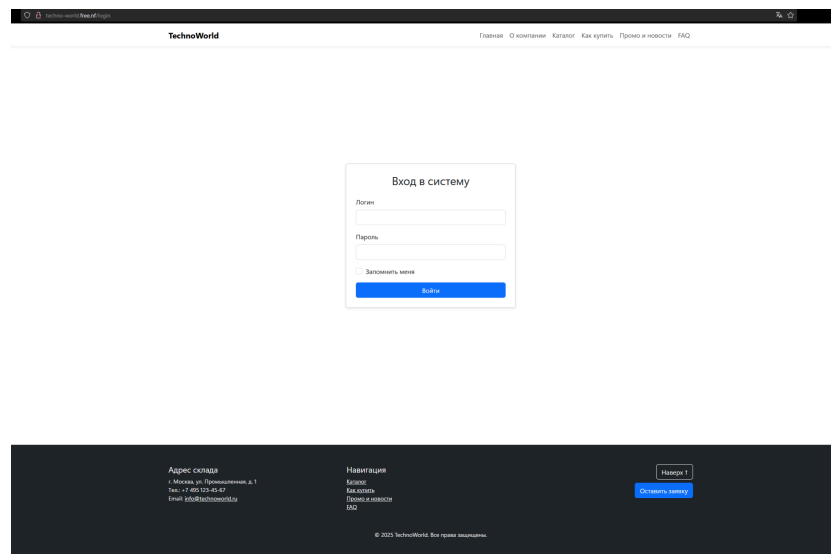
```
CREATE TABLE requests (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    client_name VARCHAR(255),  
    contact VARCHAR(255),  
    comment TEXT,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

Контроллер обрабатывает входящие данные, выполняет проверку токена CSRF, ограничивает частоту запросов и сохраняет результат с логированием события.

## 2.4. Панель администратора

Административный раздел доступен по маршруту `/admin` и защищён механизмом авторизации. Для входа используется форма логина с полями `username` и `password`. После успешной авторизации пользователю выдаётся JWT-токен, хранящийся в `cookie`.

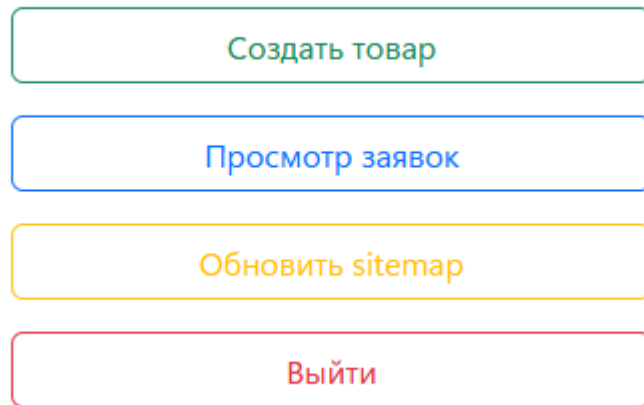
Контроллер `AuthController` проверяет введенные данные, сверяет их с записями в таблице `users` и формирует подпись токена, содержащую ID и уровень доступа. Доступ к админ-панели разрешён только пользователям с ролью `admin`.



*Изображение 7 - вход в админ-панель*

Главный интерфейс админ-панели содержит:

- кнопку создания товара;
- переход к списку заявок;
- кнопку генерации sitemap;
- кнопку выхода из аккаунта.



*Изображение 8- админ-панель*

Страница /admin/requests отображает таблицу всех клиентских обращений. Записи сортируются по дате, отображаются имя отправителя, контакты, текст и дата получения. Администратор может просматривать заявки. Система рассчитана на добавление дополнительных действий (например, удалить заявку, отметить как обработанную).

#### Заявки от клиентов

#	Имя	Телефон	Сообщение	Дата
8	Денис	+7 999 123-45-67	заявочка	2025-04-21 13:17:27
7	Дмитрий	+7 999 123-45-67	Заявка	2025-04-21 13:17:16
6	Александр	+7 999 123-45-67	ывфывфы	2025-04-21 06:25:21

*Изображение 9 - просмотр заявок в админ-панели*

## 2.5. Реализация безопасности

Сайт реализует комплекс базовых механизмов защиты, соответствующих типовым рекомендациям для малых веб-приложений.

CSRF-защита. Каждая форма, отправляющая данные, содержит скрытое поле с CSRF-токеном. Токен генерируется на сервере и привязывается к сессии пользователя. При отправке формы значение проверяется на соответствие. При недействительном токене возвращается ошибка 419.

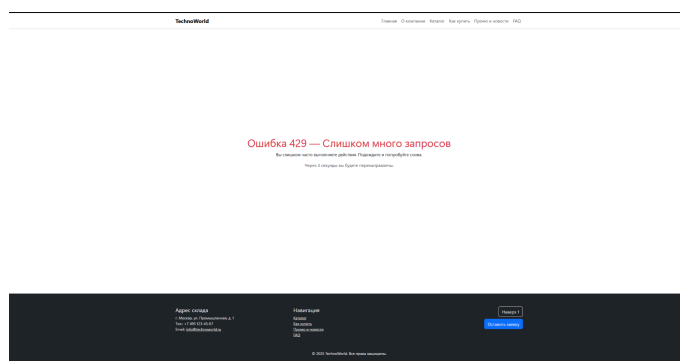
Ограничение частоты. Для защиты от спама и перебора паролей реализована система rate-limit, ограничивающая частоту выполнения конкретных действий по IP-адресу. Повторный запрос блокируется, если выполнен слишком быстро после предыдущего.

Логирование. Все ключевые действия, включая вход в систему, отправку формы, ошибки доступа и действия администратора, записываются в лог. Это позволяет отслеживать поведение пользователей и выявлять аномалии.

Обработка ошибок. На сайте реализованы пользовательские страницы для ошибок 403, 404, 419, 500. Ошибки логируются с деталями запроса.

Дополнительно использовались:

- фильтрация ввода и экранирование HTML;
- подготовленные SQL-запросы (PDO::prepare);
- защита от XSS в шаблонах вывода.



Изображение 10 - страница ошибки



## 2.6. Генерация sitemap.xml и robots.txt

Файл robots.txt размещён в корневом каталоге и содержит инструкции для поисковых систем. Он разрешает индексацию публичных страниц и запрещает доступ к административным маршрутам.

```
User-agent: *  
  
Disallow: /admin/  
  
Disallow: /login  
  
Disallow: /logout  
  
Disallow: /request/success  
  
Disallow: /error/  
  
Disallow: /error  
  
  
Allow: /  
  
  
Sitemap: https://techno-world.free.nf/sitemap.xml
```

*файл robots.txt*

Файл sitemap.xml создаётся автоматически через админ-панель кнопкой «Обновить sitemap». Генерация выполняется контроллером SitemapController. В sitemap включаются:

- все основные статические страницы;
- страницы товаров, полученные из базы данных;  
даты последнего изменения страниц;
- частота обновления и приоритет.

## 2.7. Размещение сайта на хостинге

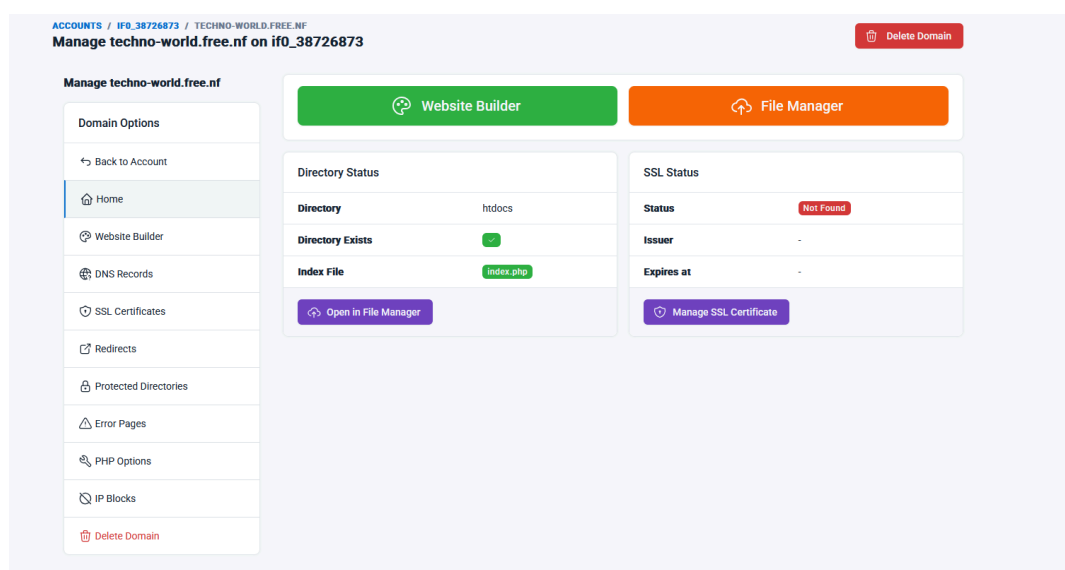
Публикация проекта в сети Интернет осуществлялась с использованием бесплатного хостинга InfinityFree, выбранного в соответствии с техническими требованиями проекта и ограничениями учебного формата. Данная платформа обеспечивает поддержку PHP, MySQL, а также загрузку пользовательских

.htaccess-файлов, что делает её пригодной для развёртывания сайтов с пользовательской логикой и маршрутизацией.

Процесс размещения включал несколько этапов:

Регистрация и создание сайта.

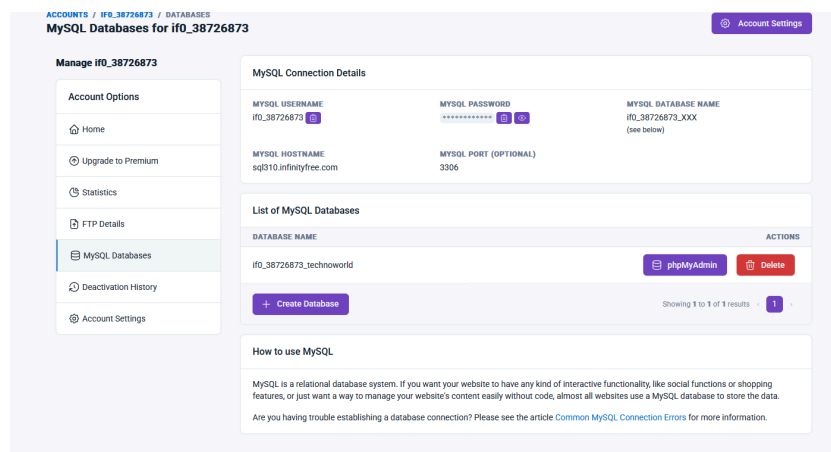
На официальном сайте InfinityFree была пройдена регистрация, после чего в панели управления (VistaPanel) был создан новый сайт с техническим доменом techno-world.free.nf. На этом этапе также были сгенерированы параметры доступа к FTP и базе данных MySQL.



*Изображение 11 - Панель управления infinityfree*

Настройка базы данных.

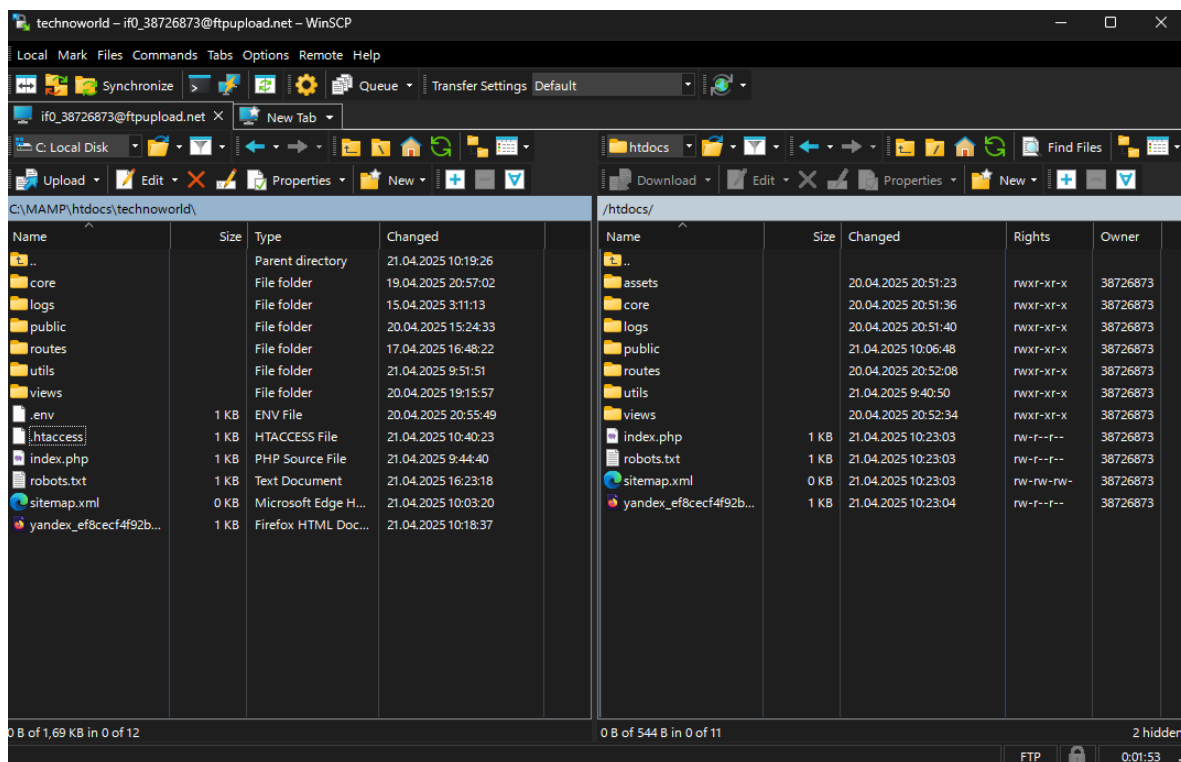
В интерфейсе панели был создан экземпляр базы данных, содержащий имя, логин и пароль. Эти параметры впоследствии были указаны в конфигурационном классе проекта, отвечающем за подключение к базе через PDO.



*Изображение 12 - Панель управления базами данных infinityfree*

Передача файлов на сервер.

Для загрузки файлов использовался FTP-клиент WinSCP, обеспечивший стабильную передачу и поддержку визуальной синхронизации содержимого между локальной директорией и сервером. Альтернативный инструмент FileZilla был отклонён из-за конфликтов с антивирусным ПО.



*Изображение 13 - WinSCP FTP Client*

Все файлы проекта были размещены в корневой папке `htdocs`, которая автоматически соответствует базовому маршруту сайта. После завершения передачи ресурсов и базовой конфигурации, спустя трое суток сайт стал доступен по адресу:

<https://techno-world.free.nf/>

Таким образом, проект был успешно развёрнут в условиях ограниченного хостинг-окружения, без дополнительных серверных компонентов. Несмотря на отсутствие полноценного SSL-сертификата, сайт функционирует в полной мере, включая маршрутизацию, хранение данных и административный интерфейс.

## **2.8. Интеграция с Яндекс.Вебмастер**

После публикации сайта была предпринята попытка интеграции с сервисом Яндекс.Вебмастер с целью передачи данных для индексации и анализа SEO-показателей. Стандартная процедура включала подтверждение прав на сайт и отправку `sitemap`-файла.

Использованные методы верификации

В рамках проекта были реализованы различные поддерживаемые Яндексом методы подтверждения:

HTML-файл — на сервер был размещён файл `yandex_ef8cecf4f92b4583.html` в корневой директории. При переходе по прямой ссылке файл открывался корректно в браузере, но попытка верификации через Вебмастер завершалась ошибкой 403 или 404.

Мета-тег — тег вида `<meta name="yandex-verification"`

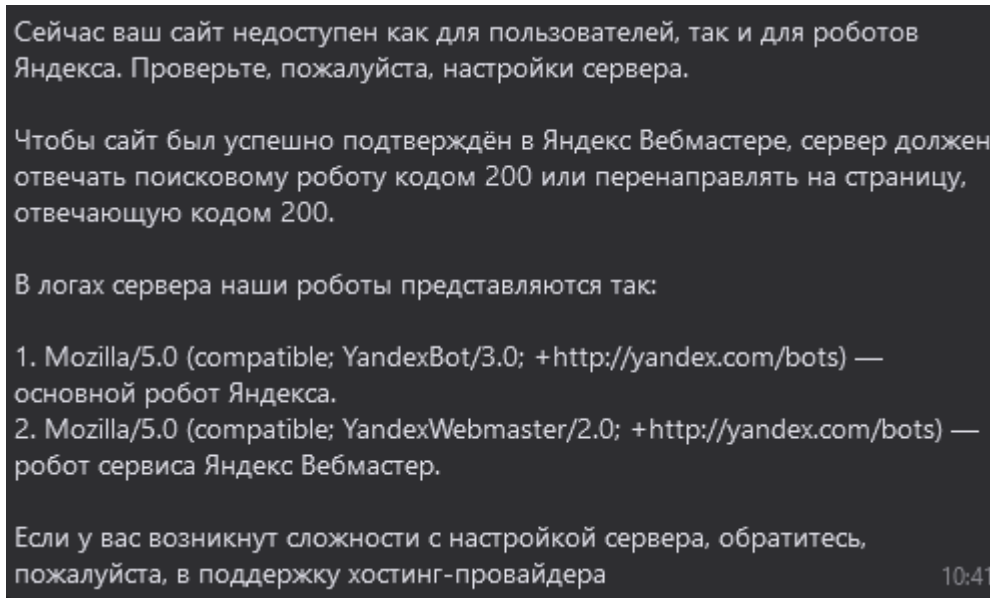
`content="ef8cecf4f92b4583" />` был вставлен в `<head>` главной страницы.

Несмотря на наличие тега в исходном HTML-коде, Яндекс сообщал, что он недоступен для проверки.

Зеркало через GitHub Pages.

Для обхода ограничений хостинга была предпринята попытка верифицировать сайт через промежуточное зеркало. На платформе GitHub Pages был создан отдельный сайт, содержащий HTML-файл верификации Яндекса, а также настроен автоматический редирект на основной домен проекта. После этого зеркало было добавлено в Яндекс.Вебмастер как временный сайт, а основной сайт `techno-world.free.nf` — предложено прикрепить к нему как зеркало. Тем не менее, система Яндекса отказалась признавать основной домен зеркалом GitHub-сайта, из-за чего метод не дал результата. Попытка обойти фильтрацию роботов таким способом не увенчалась успехом.

Согласно данным технической поддержки Яндекса, сайт `techno-world.free.nf` оказался недоступен для их роботов. Проверка HTTP-заголовков показала, что основной бот YandexBot получает ответ 403 Forbidden при попытке обращения даже к главной странице сайта. Это говорит о том, что InfinityFree фильтрует или ограничивает доступ к ресурсу на уровне IP или User-Agent, препятствуя нормальной работе поисковых краулеров.



*Изображение 14 - Ответ техподдержки Yandex*

Дополнительным осложнением стал тот факт, что многие альтернативные бесплатные хостинги также требуют значительного времени на активацию сайта, регистрацию в DNS и не гарантируют доступность без вложений. В связи с тем, что работа над курсовым проектом велась в условиях ограниченного времени, дальнейшие попытки подключить другие платформы были прекращены.

В текущих условиях подтвердить владение сайтом через Яндекс.Вебмастер на базе бесплатного хостинга не удалось. Все используемые методы оказались неэффективными, что связано с техническими ограничениями InfinityFree. Тем не менее, функциональность сайта и его SEO-структура были реализованы полностью, включая robots.txt, sitemap.xml, понятные URL и логичную архитектуру. При дальнейшем переносе на коммерческий хостинг вопрос индексации может быть решён без доработок.

## Заключение

В рамках курсовой работы была спроектирована и реализована полнофункциональная версия веб-сайта для модельной торговой организации ООО «ТехноМир». Сайт охватывает базовые потребности коммерческого веб-присутствия: от информационных страниц и каталога товаров до формы заявок с последующей административной обработкой.

В процессе разработки были реализованы следующие аспекты:

- архитектура MVC с разделением логики, шаблонов и маршрутизации;
- форма обратной связи с сохранением данных в базу;
- административная панель с входом по JWT и отображением заявок;
- защита от CSRF и злоупотреблений через токены и ограничение частоты запросов;
- генерация файлов sitemap.xml и robots.txt;
- попытка интеграции с сервисом «Яндекс.Вебмастер».

Проект был развёрнут на бесплатном хостинге InfinityFree. Несмотря на его ограниченность, он обеспечил стабильную работу, поддержку PHP, MySQL и .htaccess. Основная трудность заключалась в том, что ни один из протестированных хостингов не позволил пройти верификацию в Яндекс.Вебмастере — запросы робота блокировались или искажались. Попытки решения через зеркало и редиректы также не дали результата, и работа с индексацией была ограничена.

Весь функционал, запланированный в начале проекта, был реализован. Созданный сайт может быть доработан и расширен: добавить полноценную корзину, REST-интерфейс, поиск по товарам, поддержку нескольких ролей пользователей, или перенесён на коммерческий хостинг с полноценной индексацией и аналитикой.

## Библиографический список

1. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 : пер. с англ. / Р. Никсон. — 5-е изд. — М. : Addison-Wesley, 2020. — 720 с.
2. Макфарланд, Д. Новая большая книга CSS3 : пер. с англ. / Д. Макфарланд. — М. : O'Reilly Media, 2019. — 512 с.
3. Ташков, П. Веб-мастеринг на 100%: HTML, CSS, JavaScript, PHP, CMS, AJAX, раскрутка / П. Ташков. — М. : Реальное издание, 2018. — 384 с.
4. Веллинг, Л. Разработка Web-приложений на PHP и MySQL : пер. с англ. / Л. Веллинг, Л. Томсон. — М. : Addison-Wesley, 2021. — 832 с.
5. Симпсон, К. Вы не знаете JS. Замыкания и объекты : пер. с англ. / К. Симпсон. — М. : O'Reilly Media, 2019. — 160 с.
6. Дакетт, Д. HTML и CSS. Разработка и дизайн веб-сайтов : пер. с англ. / Д. Дакетт. — М. : Wiley, 2020. — 512 с.
7. Стефанов, С. JavaScript. Шаблоны : пер. с англ. / С. Стефанов. — М. : O'Reilly Media, 2020. — 240 с.
8. Эспозито, Д. Разработка современных веб-приложений. Анализ предметных областей и технологий : пер. с англ. / Д. Эспозито. — М. : Microsoft Press, 2019. — 368 с.
9. Пьюривал, С. Основы разработки веб-приложений : пер. с англ. / С. Пьюривал. — М. : No Starch Press, 2020. — 304 с.
10. Седерхольм, Д. CSS ручной работы : пер. с англ. / Д. Седерхольм. — М. : Peachpit Press, 2021. — 208 с.



## Структура файлов и каталогов веб-сайта торговой организации ООО “ТехноМир”

```

— index.php // Точка входа в приложение, обрабатывает все входящие запросы
— .htaccess // Конфигурация веб-сервера: ЧПУ, защита .env, перенаправления, кастомные ошибки
— robots.txt // Директивы для поисковых роботов
— sitemap.xml // XML-карта сайта для ускорения индексации
— yandex_ef8cecf4f92b4583.html // Файл верификации Яндекс.Вебмастера
— .env // Конфигурация окружения: DSN, логин, пароль БД и секреты

— core/
  — Config/
    — Database.php // Класс подключения к базе данных через PDO
  — Controllers/ // Контроллеры — логика обработки HTTP-запросов
    — AdminController.php
    — AdminDashboardController.php
    — AdminGoodsController.php
    — AdminRequestsController.php
    — AuthController.php
    — BaseController.php
    — CatalogController.php
    — InfoPagesController.php
    — RequestController.php
    — SitemapController.php
  — Models/ // Модели — взаимодействие с БД
    — Good.php
    — GoodImage.php
    — Request.php
    — User.php
  — Security/ // Компоненты безопасности
    — AuthManager.php
    — CSRFManager.php
    — JWTManager.php
    — RateLimiter.php
  — logs/
    — website.log // Файл логирования событий сайта
  — init.php // Инициализация базовых компонентов проекта
  — utils/
    — helpers.php // Вспомогательные функции
    — http.php // Обработка ошибок и HTTP-ответов

— public/
  — routes/
    — web.php // Файл маршрутов: связывает URL и контроллеры

— views/ // Представления (шаблоны)
  — admin/ // Интерфейс административной панели
    — createGood.php
    — dashboard.php
    — requests.php
    — sitemapResult.php
  — errors/ // Шаблоны страниц ошибок по HTTP-кодам
    — 401.php, 403.php, ..., 503.php
  — layouts/ // Лэйауты (обертки) для разных типов страниц
    — auth.php
    — base.php
    — error.php
  — pages/ // Статические страницы
    — about.php
    — home.php
    — how-to-buy.php
  — partials/ // Частичные шаблоны
    — catalogResults.php
    — footer.php
    — header.php
    — productCard.php
  — request/ // Страницы для взаимодействия с формами
    — form.php
    — success.php
    — catalog.php
    — loginForm.php
    — product.php

```