

Содержание

Введение.....	3
1 Анализ предметной области	6
2 Проектирование приложения.....	9
3 Разработка программного обеспечения.....	15
3.1 Описание технологического стека разработки	15
3.2 Описание алгоритма работы	18
3.3 Описание интерфейса пользователя	21
4 Тестирование веб-приложения	24
4.1 План тестирования	24
4.2 Оценка результатов проведения тестирования	25
Заключение	31
Список использованных источников	33

					ОКЭИ 09.02.07. 7025. 08 УП				
Изм.	Лист	№ докум.	Подпись	Дата					
Разраб.		Исаева Д.С.			Отчёт по учебной практике				
Провер.		Левченко И.А.							
Реценз									
Н. Контр.									
Утверд.									
					Лит.	Лист	Листов		
							2	38	
					Отделение программирования, гр. 4082				

Введение

Интернет и веб-технологии прошли долгий путь с момента их изобретения. От первых электронных писем и простых веб-страниц мы пришли к тому, что имеем сегодня — к сложным веб-приложениям, мобильным приложениям и социальным сетям.

Изначально интернет использовался только для обмена данными между учеными и исследовательскими центрами. С течением времени он стал доступен для всех, и сейчас мы используем его для общения, поиска информации, покупки товаров и многого другого.

Веб-сайты — это ресурсы, расположенные в интернете и доступные для просмотра через браузер. Они могут содержать различные элементы, такие как текст, изображения, видео, аудио и другие. Веб-сайты используются для различных целей, таких как предоставление информации, продажа товаров, общение и другие.

В современном динамичном мире ресторанной индустрии эффективная коммуникация между компаниями-поставщиками (продукты, оборудование, услуги) и ресторанами играет ключевую роль в успехе обеих сторон. Традиционные методы коммуникации, такие как телефонные звонки и электронная почта, часто оказываются недостаточно эффективными, приводя к задержкам, недопониманиям и потере времени. В условиях растущей конкуренции и необходимости оперативного реагирования на изменения рынка, потребность в оптимизированной и прозрачной системе коммуникации становится особенно актуальной. Данная курсовая работа посвящена разработке информационного веб-портала, призванного улучшить взаимодействие между компаниями и ресторанами, обеспечивая своевременный обмен информацией, повышение эффективности сотрудничества и укрепление деловых отношений.

Портал может значительно упростить процесс коммуникации между компанией и ресторанами. Это особенно актуально в условиях, когда скорость и точность передачи информации критически важны для принятия решений.

На сегодняшний день наблюдается значительный рост числа ресторанов и связанных с ними предприятий, что, в свою очередь, стимулирует конкуренцию на рынке. Успех ресторана все больше зависит не только от качества кухни и обслуживания, но и от эффективного управления ресурсами, включая закупку продуктов, оборудования и привлечение новых клиентов. Для компаний в свою очередь, крайне важно поддерживать тесные контакты с ресторанами, обеспечивая своевременную доставку товаров, предоставление качественного сервиса и гибкое реагирование на запросы клиентов. Отсутствие эффективной коммуникации может привести к ряду негативных последствий, оказывающих значительное влияние как на отдельные проекты, так и на весь бизнес в целом. Прежде всего, это ведёт к потере времени и ресурсов. Когда сотрудники не могут ясно выражать свои мысли или не понимают друг друга, задачи затягиваются, и на их выполнение уходит больше времени; в результате страдают не только результаты работы, но и моральный климат в команде.

Недопонимание и ошибки также становятся привычными при плохой коммуникации. Когда информация не доходит до адресатов или искажается по пути, это неизбежно приводит к ошибкам, которые могут быть дорогостоящими и требовать дополнительных затрат на их исправление. Это может касаться как мелких задач, так и крупных проектов, где неточности могут кардинально изменить результат.

Уменьшение прибыли — ещё один негативный эффект. Неэффективная коммуникация приводит к снижению продуктивности работы, что, в свою очередь, может сказаться на финансовых показателях компании. Процессы становятся менее оптимизированными, а клиенты могут недовольствоваться качеством обслуживания, что негативно влияет на репутацию и, как следствие, на прибыль.

Кроме того, ухудшение деловых отношений также является важным аспектом. Отсутствие взаимопонимания между коллегами, партнёрами и клиентами может вызвать недоверие и напряжённость. Этот фактор, в свою очередь, делает невозможным построение долговременных и взаимовыгодных отношений, которые играют ключевую роль в развитии бизнеса. В конечном итоге, недостаточная коммуникация может поставить под угрозу как отдельные проекты, так и устойчивое существование компании в целом. Потеря времени и ресурсов означает, что запросы и ответы, проходящие через множество посредников, затягивают процесс и увеличивают затраты на коммуникацию.

Недопонимание и ошибки возникают в случае если есть неясная или неполная информация, которая может привести к ошибкам в заказах, поставках и других аспектах сотрудничества.

Задержки и недоразумения приводят к снижению эффективности работы ресторанов и уменьшению прибыли как для них самих, так и для компаний-поставщиков.

Предлагаемый информационный веб-портал направлен на решение вышеупомянутых проблем, предоставляя комплексную платформу для эффективного взаимодействия между компаниями и ресторанами

Внедрение такого портала позволит компаниям и ресторанам значительно сократить время, затрачиваемое на коммуникацию, снизить вероятность ошибок, повысить эффективность сотрудничества и укрепить деловые отношения. В конечном итоге, это приведет к росту прибыли и конкурентоспособности как ресторанов, так и компаний-поставщиков. Данная курсовая работа представляет собой исследование и разработку такого портала, анализ его потенциальной эффективности и описание архитектуры и функциональных возможностей. В ходе работы будут рассмотрены вопросы выбора технологий, разработки пользовательского интерфейса, обеспечения безопасности и масштабируемости.

Разработка информационного веб-портала как средства оптимизации коммуникации между компаниями и ресторанами — это актуальная задача, направленная на повышение эффективности работы обеих сторон и создание благоприятной среды для развития бизнеса в ресторанной индустрии.

Одним из основных средств создания web-страниц являются HTML-теги. HTML (HyperText Markup Language) — это язык разметки гипертекста, который

позволяет структурировать текст и элементы на web-странице. С помощью HTML-тегов можно определить, как будет выглядеть текст, какие элементы будут отображаться, как будут располагаться элементы на странице и т.д.

Объектом исследования является ООО ОренРест, которая занимается деятельностью ресторанов и услуг по доставке продуктов питания.

Предметом исследования является информационный портал для коммуникации между компанией и ресторанами.

Целью данной работы является разработка концепции информационного портала, который будет служить связующим звеном между компаниями и ресторанами. Такой портал может включать в себя функционал для обмена информацией, упрощения процессов заказа, предоставления актуальных новостей и специальных предложений, а также возможность обратной связи.

Для достижения поставленной цели необходимо решение следующих основных задач:

- проведение анализа исходных данных;
- составить техническое задание на разработку сайта;
- разработать дизайн сайта с учетом удобства навигации, информативности, привлекательности и простоты использования;
- создание веб-сайта;
- разработка сайта с использованием современных программных средств;
- создание базы данных для сайта;
- проведение тестирования веб-сайта;
- оценить эффективность разработанного сайта.

Выбор темы «Информационный портал для коммуникации между компанией и ресторанами» обусловлен ее практической значимостью и актуальностью в условиях современного рынка. Разработка подобного портала представляет собой сложную, но высоко востребованную задачу, решение которой может существенно улучшить бизнес-процессы в ресторанной индустрии. Данная работа позволит углубить знания в области веб-разработки, проектирования баз данных и управления проектами, а также получить практический опыт в создании реального веб-приложения.

Таким образом, данный проект позволит не только установить эффективные каналы коммуникации между компаниями и ресторанами, но и создать платформу для развития бизнеса, улучшения качества предоставляемых услуг и удовлетворенности клиентов.

1 Анализ предметной области

С развитием технологий и цифровых платформ веб-сайты становятся неотъемлемой частью бизнеса, особенно в сфере общепита. Одной из актуальных задач в данной области является создание эффективного информационного портала, который будет служить связующим звеном между компанией (например, поставщиком продуктов или услуг) и ресторанами. Такой портал должен обеспечивать удобный доступ к информации о новостях, изменениях в правилах и стандартах, а также предоставлять другие важные данные, способствующие успешной интеграции бизнес-процессов.

Предметная область включает двух ключевых участников: компанию и рестораны. Их потребности и цели во взаимодействии формируют основу для функциональных требований портала.

Компания: компания нуждается в эффективном инструменте для распространения информации среди ресторанов, контроля соблюдения стандартов и оперативного реагирования на запросы. Ключевые потребности включают:

- централизованное управление информацией;
- контроль доступа;
- эффективная обратная связь;
- интеграция с существующими системами[12].

Централизованное управление информацией — это возможность легко публиковать и обновлять новости, правила, стандарты и другую документацию.

Контроль доступа — это управление правами доступа разных пользователей внутри Компании (разделение на администратора, директоров и пользователей).

Эффективная обратная связь — это возможность получать вопросы и отзывы от ресторанов, оперативно на них реагировать.

Рестораны нуждаются в удобном и оперативном доступе к актуальной информации, необходимой для эффективной работы. Ключевые потребности включают:

- быстрый и легкий доступ к информации;
- архив документов;
- простота использования.

Быстрый и легкий доступ к информации означает удобный интерфейс для поиска и просмотра новостей, правил, стандартов и другой документации.

Архив документов — это доступ к истории публикаций для поиска необходимой информации в любое время.

Простота использования значит интуитивно понятный интерфейс, не требующий специальной подготовки.

Предметная область включает следующие ключевые сущности и их взаимосвязи:

- два типа пользователей;
- новости;
- правила;

- стандарты;
- документы;
- роли и права доступа.

Два типа пользователей, которыми являются администратор (Компания) и пользователи ресторанов.

Новостями являются текстовые сообщения с возможностью добавления изображений, содержащие актуальную информацию для ресторанов.

Правила — это документы, регламентирующие взаимоотношения между Компанией и Ресторанами (например, контрактные условия, требования к качеству).

Стандарты — это документы, описывающие требования к качеству продукции, услугам или процессам.

Документы — это общий термин, охватывающий новости, правила и стандарты. Может включать различные типы файлов (PDF, DOCX, JPG и т.д.);

Функциональные требования для компании, администратора и директоров ресторанов:

- добавление, редактирование и удаление новостей (с возможностью загрузки изображений);
- добавление, редактирование и удаление правил и стандартов (с возможностью загрузки файлов);
- управление пользователями (добавление, редактирование, удаление, назначение ролей);
- просмотр статистики использования портала (количество посещений, просмотров новостей и т.д.).

Для авторизированных пользователей:

- просмотр новостей, правил и стандартов;
- возможность поиска и фильтрации информации по ключевым словам, дате публикации и типу документа.

Нефункциональные требования:

- надежность;
- производительность;
- масштабируемость;
- безопасность;
- удобство использования;
- соответствие стандартам.

Надежность значит, что портал должен быть устойчив к отказам и обеспечивать бесперебойную работу. Надежность в контексте портала означает его способность функционировать стабильно и безотказно в течение заданного периода времени и при различных условиях. Это значит, что пользователи должны иметь возможность полагаться на портал для выполнения своих задач без неожиданных сбоев или недоступности.

Производительность значит, что портал должен быстро загружаться и обрабатывать запросы пользователей. Производительность портала характеризует его скорость и эффективность в обработке запросов пользователей и выполнении

задач. Быстрый и отзывчивый портал обеспечивает комфортный пользовательский опыт и повышает эффективность работы.

Масштабируемость означает, что портал должен быть способен обрабатывать большое количество пользователей и данных. Масштабируемость – это способность портала эффективно справляться с возрастающей нагрузкой, будь то увеличение числа пользователей, объема данных или интенсивности трафика. Масштабируемый портал может расти и развиваться вместе с потребностями бизнеса.

Безопасность портала – это комплекс мер, направленных на защиту информации, функциональности и пользователей от несанкционированного доступа, использования, изменения, раскрытия или уничтожения. Безопасность обеспечивает конфиденциальность, целостность и доступность данных.

Удобство использования значит, что UI должен быть интуитивно понятным и удобным для пользователей. Удобство использования (юзабилити) – это степень, в которой портал легок в изучении, эффективен в использовании, приятен и удовлетворителен для пользователей. Удобный портал позволяет пользователям быстро и эффективно достигать своих целей без лишних усилий и раздражения.

Соответствие стандартам значит, что портал должен соответствовать современным стандартам веб-разработки. Соответствие стандартам веб-разработки означает, что портал разработан в соответствии с общепринятыми рекомендациями, спецификациями и стандартами, установленными организациями, такими как W3C (World Wide Web Consortium).

					ОКЭИ 09.02.07. 7025. 08 УП	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

2 Проектирование приложения

Цель работы заключается в разработки веб-портала для эффективной коммуникации между главной компанией и сетью ресторанов, обеспечивающего оперативный обмен информацией о новостях, изменениях правил, стандартов, а также другой важной документацией.

Задачи:

- разработка архитектуры веб-портала, включающей в себя клиентскую и серверную части;
- проектирование базы данных для хранения информации о новостях, правилах, стандартах, ресторанах и пользователях;
- разработка пользовательского интерфейса (UI) для удобного доступа к информации и управления личным кабинетом;
- реализация функционала публикации новостей, правил и стандартов главной компанией;
- реализация функционала доступа к информации для ресторанов, включая поиск, фильтрацию и сортировку;
- реализация системы авторизации и управления доступом пользователей (компания и рестораны);
- тестирование и отладка разработанного портала.

Функциональные требования для администратора:

- добавление новостей;
- добавление коммуникаций;
- добавление пользователей.

Добавление новостей означает, что администратор должен иметь возможность создавать новую новость с указанием заголовка, текста, категории и даты публикации. Для каждой новости предусмотрена функция загрузки изображения. Поддерживаемые форматы должны включать JPEG, PNG и GIF. Администратор должен иметь возможность перемещения новостей в архив, а также полного удаления, с подтверждением действия, чтобы избежать случайного удаления;

Добавление коммуникаций — это возможность создания новых документов, содержащих правила и стандарты с указанием заголовка, текста и даты публикации. Функция загрузки файлов (например, PDF, DOCX) для дополнительных материалов, связанных с правилами и стандартами. Возможность перемещения документов в архив или полного удаления с подтверждением, чтобы предотвратить случайные действия.

Добавление пользователей значит, что администратор может создать нового пользователя, указав необходимые данные: имя, фамилию, email, пароль и роль (например, пользователь, модератор, редактор).

Для авторизованных пользователей:

- просмотр новостей, правил и стандартов;
- авторизация на сайте;

- обратная связь;

Ниже представлены различные требования к дизайну такого сайта.

Требования к пользовательскому интерфейсу (UI):

- простой и интуитивно понятный интерфейс;
- адаптивный дизайн;
- чистый и современный дизайн;
- логотип и брендинг.

Простой и интуитивно понятный интерфейс значит, что дизайн должен быть легким для восприятия, чтобы пользователи могли быстро находить нужную информацию.

Адаптивный дизайн означает, что сайт должен корректно отображаться на различных устройствах (ПК, планшеты, мобильные телефоны).

Планируются использовать минималистичные элементы, чтобы не перегружать пользователей что подразумевает чистый и современный дизайн.

Логотип и брендинг — это наличие яркого логотипа и соблюдение цветовой схемы, соответствующей идентичности компании.

Требования к навигации:

- удобная навигационная панель, ясная и логичная структура меню с выделением главных разделов (Новости, Правила, Стандарты, Контакты и т.д.);
- поиск по сайту, функция поиска по ключевым словам для быстрого доступа к нужной информации.

Требования к контенту:

- актуальная информация, регулярные обновления новостей и материалов;
- подробные разделы, четкое и структурированное представление информации о нововведениях, правилах и стандартах;
- возможность комментирования и обсуждения, опции для пользователей оставлять свои мнения и обмениваться опытом.

Требования к графическим элементам:

- использование изображений и видео;
- иконки и инфографика.

Использование изображений и видео значит, что визуальные элементы должны поддерживать текстовой контент и сделать его более интересным.

Иконки и инфографика помогут быстро передавать информацию и визуализировать данные[3].

Требование к доступности:

- соблюдение стандартов доступности;
- контрастность и шрифты.

Соблюдение стандартов доступности: этот аспект должен учитывать пользователей с ограниченными возможностями (например, наличие текстов для экранных считывателей).

Контрастность и шрифты: необходимо использовать четкие и читаемые шрифты с хорошим контрастом.

Требования к безопасности:

- защита данных;

- SSL-сертификат.

Защита данных — это обеспечение конфиденциальности и безопасности пользовательских данных.

SSL-сертификат — это наличие сертификата для защиты данных пользователей.

Требования к SEO-оптимизации:

- адаптация под поисковые системы.

Оптимизация заголовков, метатегов и контента для повышения видимости в поисковых системах.

Требования к тестированию и обратной связи:

- пользовательское тестирование;
- сбор обратной связи.

Пользовательское тестирование — это регулярное тестирование с реальными пользователями для выявления проблем и улучшения UX.

Сбор обратной связи — это опции для пользователей оставлять отзывы о сайте и его функционале[2].

Технические требования:

- серверная часть на Node.js;
- база данных на SQLite;
- фронтенд на HTML, CSS, JavaScript;
- система контроля версий на Git.

В данном проекте серверная часть будет разворачиваться на платформе Node.js, что обеспечит асинхронную обработку запросов и высокую производительность. Для хранения данных будет использоваться SQLite — легковесная реляционная база данных, идеально подходящая для небольших приложений и быстрой разработки. Она обеспечивает простоту настройки и интеграции с серверной частью на Node.js.

Фронтенд графического интерфейса будет реализован с использованием стандартных технологий — HTML, CSS и JavaScript. Это позволит создать современное и отзывчивое пользовательское пространство, обеспечивая при этом кроссбраузерную совместимость. Структура разметки и стилизация будут выполнены с акцентом на удобство и визуальную привлекательность, а функциональность будет реализована с использованием JavaScript, включая возможные взаимодействия с сервером через AJAX.

Ожидаемый результат:

Работоспособный веб-портал, соответствующий всем функциональным и нефункциональным требованиям, с полной документацией и отчетом о проделанной работе. В отчете должна быть описана архитектура приложения, диаграммы баз данных, описание использованных технологий, результаты тестирования и рекомендации по дальнейшему развитию портала.

Требования к техническому обеспечению:

- сеть с подключение к Интернет процессор Intel Xeon E3 шестого поколение или новее, или AMD Ryzen 1950x или новее; 32 Gb и более оперативной памяти;
- 2 Tb – жесткий диск SSD или HDD с поддержкой SATA 3;

- монитор – FullHD или 4K;
- клавиатура;
- манипулятор типа «мышь».

Требования, предъявляемые к конфигурации клиентских станций:

- процессор, с тактовой частотой не менее 2,4 GHz, не менее 2 ядер;
- 8 Gb оперативной памяти;
- монитор – FullHD;
- клавиатура;
- манипулятор типа «мышь»;
- доступ к подключению – Интернет со скоростью не менее 100 МБ/с.

Требования к программному обеспечению.

На рабочей станции пользователя необходимо установить:

- операционная система: Windows версии не ниже 8.1;
- доступ к широкополосному подключению к сети Интернет;
- браузер для просмотра веб-страниц последней версии.

Общие требования к контенту сайта

- высококачественные изображения и видео;
- уникальный и информативный контент;
- правильная грамматика и орфография;
- оптимизация для поисковых систем (SEO);
- регулярное обновление для обеспечения актуальности и новизны.

Веб-сайт должен быть наполнен высококачественными изображениями и видеоматериалами, которые визуально привлекают пользователей и усиливают восприятие информации. Крайне важен уникальный и информативный контент, предоставляющий ценность для посетителей и отличающийся от материалов на других ресурсах. Текстовое наполнение должно быть безупречным с точки зрения правил грамматики и орфографии, демонстрируя профессионализм и внимание к деталям. Важно проводить оптимизацию для поисковых систем (SEO), чтобы сайт был легко находим в интернете и привлекал органический трафик. Для поддержания интереса аудитории и сохранения актуальности информации необходимо регулярно обновлять контент, добавляя свежие материалы и поддерживая новизну ресурса.

Требования к сопровождению:

- регулярные обновления программного обеспечения;
- усовершенствование функциональности;
- улучшение производительности;
- адаптация к новым устройствам и технологиям;
- постоянная поддержка и обслуживание.

Регулярные обновления программного обеспечения — это обновление программного обеспечения сайта для обеспечения безопасности, производительности и функциональности.

Усовершенствование функциональности — это изучение новых тенденций и технологий веб-разработки и внедрение улучшений для повышения пользовательского опыта и значит отслеживание производительности сайта и

внедрение оптимизации для повышения скорости загрузки страниц и общей отзывчивости.

Адаптация к новым устройствам и технологиям — это проверка совместимости сайта с новыми устройствами и технологиями, такими как мобильные устройства и голосовые помощники.

Постоянная поддержка и обслуживание — это обеспечение постоянной поддержки и обслуживания сайта, чтобы он был постоянно доступным и работал без сбоев[2].

В процессе разработки программного обеспечения ключевое значение имеет этап проектирования, на котором определяется архитектура, функциональные и нефункциональные требования к создаваемому приложению. Грамотно выполненное проектирование заложит прочный фундамент для дальнейшего этапа реализации и обеспечит соответствие продукта ожиданиям пользователей и бизнес-требованиям.

Важным аспектом является описание функциональных требований, которые можно визуализировать с помощью диаграммы прецедентов (use case diagram). Эта диаграмма позволяет наглядно представить взаимодействие пользователей с системой и выявить ключевые сценарии использования приложения (Приложение Б). Прецеденты используются для описания конкретных функций или задач, которые система должна выполнять. Они представляют собой сценарии взаимодействия акторов с приложением и помогают понять, какие функции являются критически важными для пользователей[10].

Нефункциональные требования:

- надежность;
- производительность;
- масштабируемость;
- безопасность;
- удобство использования;
- соответствие стандартам.

Надежность значит, что портал должен быть способен выдерживать различные виды сбоев, таких как отключение сервера или проблемы с сетью, без потери доступности для пользователей. Важно реализовать механизмы резервного копирования и восстановления данных, а также обеспечить безотказную работу ключевых функций. Проведение регулярного мониторинга состояния системы для быстрого реагирования на возможные проблемы.

Производительность означает, что время загрузки страниц должно составлять не более 2-3 секунд для оптимального пользовательского опыта. Портал должен эффективно обрабатывать запрашиваемые данные, минимизируя задержки и время отклика на действия пользователя. Наличие механизма кэширования для ускорения загрузки и обработки часто запрашиваемой информации.

Масштабируемость значит, что портал должен быть способен обработать увеличенное количество пользователей и транзакций без ухудшения производительности. Необходимо предусмотреть возможность расширения инфраструктуры (например, добавление серверов или увеличение ресурсов) при

росте нагрузки. Изоляция компонентов системы для упрощения обновлений и интеграции новых функций.

Безопасность значит, что все данные, передаваемые между пользователями и порталом, должны быть защищены с помощью шифрования (например, использование HTTPS). Реализация многоуровневой аутентификации и авторизации для защиты от несанкционированного доступа. Проведение регулярных аудитов безопасности и обновления системы для защиты от новых уязвимостей.

Удобство использования значит, что интерфейс должен быть интуитивно понятным, позволяя пользователям легко находить необходимую информацию и выполнять действия. Должны быть предусмотрены адаптивные элементы интерфейса для обеспечения возможности работы с порталом на различных устройствах (мобильных, планшетах, ПК). Регулярные пользовательские тестирования для выявления и устранения проблем с удобством использования интерфейса.

Соответствие стандартам означает, что портал должен быть разработан с соблюдением современных стандартов веб-разработки, включая семантическую разметку, доступность (например, WCAG), и лучшие практики безопасности. Необходима кроссбраузерная совместимость, чтобы все пользователи могли получить доступ к функциональности портала независимо от используемого браузера. Следует учитывать требования к локализации и интернационализации, чтобы обеспечить поддержку различных языков и культур.

3 Разработка программного обеспечения

3.1 Описание технологического стека разработки

Технологический стек разработки, состоит из HTML, CSS, JavaScript, SQLite, VisualStudioCode и Figma. Они представляет собой мощный набор инструментов для создания веб-приложений.

HTML — это стандартный язык разметки для создания веб-страниц. Он позволяет структурировать содержание, используя теги для определения заголовков, параграфов, списков, изображений, ссылок и других элементов. HTML — это основа любого веб-документа и обеспечивает семантическую структуру.

Преимущества HTML:

- структурирование контента;
- простота изучения;
- совместимость;
- доступность;
- интерактивность.

HTML имеет простую и интуитивно понятную синтаксическую структуру, что делает его доступным даже для новичков. Люди, не имеющие опыта программирования, могут быстро освоить основы и начать создавать свои собственные веб-страницы.

HTML также гарантирует высокую степень совместимости с современными веб-браузерами и устройствами. Это значит, что страницы, созданные с помощью HTML, будут корректно отображаться на различных платформах, что значительно расширяет аудиторию.

Доступность является еще одним важным аспектом HTML. При правильном использовании он позволяет создавать веб-страницы, которые легко читаются программами для людей с ограниченными возможностями. Это помогает сделать интернет более инклюзивным и доступным для всех пользователей.

Наконец, HTML поддерживает различные типы медиа и форм, что позволяет возвращать интерактивность страниц. Это добавляет динамичности и делает взаимодействие с пользователями более интересным. В сумме, преимущества HTML делают его основополагающим элементом веб-разработки и важным инструментом для создания современных, доступных и интерактивных веб-страниц.

CSS — это язык стилей, который используется для оформления и визуального представления HTML-документов. С помощью CSS можно задавать цвета, шрифты, отступы, размеры и многие другие стилистические параметры. CSS также позволяет адаптировать дизайн под различные устройства и экраны, используя медиа-запросы.

Преимущества CSS:

- разделение содержания и оформления;
- управление стилями;
- адаптивный дизайн;
- кросс-браузерная совместимости;

- переиспользование стилей;
- анимации и переходы.

CSS, или каскадные таблицы стилей, обладает множеством преимуществ, которые делают его неотъемлемой частью веб-разработки. Во-первых, он обеспечивает разделение содержания и оформления, что означает, что структура веб-страницы отделена от её визуального представления. Это делает процесс обновления и поддержки сайта значительно проще, так как изменения в дизайне не требуют переработки HTML-кода.

Кроме того, CSS предоставляет разработчикам мощные инструменты для управления стилями, позволяя легко изменять шрифты, цвета, отступы и размеры элементов на странице. Это делает процесс стилизации более гибким и удобным. Также стоит отметить возможность создания адаптивного и отзывчивого дизайна с помощью CSS, что гарантирует корректное отображение сайта на различных устройствах, включая мобильные телефоны, планшеты и настольные компьютеры.

Кросс-браузерная совместимость является еще одним важным преимуществом CSS. Стандарты CSS поддерживаются большинством современных браузеров, что позволяет обеспечить единый стиль как на разных платформах, так и в различных браузерах.

CSS также позволяет переиспользовать стили, что значительно сокращает объем кода и упрощает его поддержку. Вместо того чтобы повторно писать одни и те же стили для разных элементов или страниц, разработчики могут создать стиль всего один раз и применять его везде, где это необходимо.

Наконец, CSS поддерживает анимации и переходы, которые могут добавить динамику и привлекательность пользовательскому интерфейсу. Это не только улучшает восприятие сайта, но и делает взаимодействие с ним более интересным и увлекательным для пользователей.

SQLite — это легковесная СУБД (система управления базами данных), которая используется для хранения данных в виде файлов. Она не требует установки на сервер и подходит для разработки небольших приложений и прототипов. SQLite идеально вписывается в проекты, где необходима простая и быстрая работа с данными без сложной архитектуры.

Преимущества SQLite:

- легковесность;
- простота использования;
- кросс-платформенность;
- отсутствие необходимости в установке;
- хорошая производительность;
- поддержка SQL.

SQLite представляет собой легковесную и встроенную базу данных, которая не требует отдельного серверного процесса, что делает ее удобной для развертывания и использования. Вся информация хранится в одном файле, что упрощает управление данными. У SQLite интуитивно понятный интерфейс, благодаря которому пользователи могут быстро начать работать с базой данных без

необходимости в сложной настройке. Важным аспектом является поддержка свойств ACID, что гарантирует надежность и целостность работы с данными.

Безусловным плюсом является кросс-платформенность: SQLite может функционировать на различных операционных системах и устройствах, поддерживающих файловую систему. Поскольку это встроенная база данных, пользователям не требуется устанавливать или управлять отдельным сервером, что значительно сокращает время и усилия на настройку. Для небольших и средних приложений SQLite демонстрирует хорошую производительность при обработке запросов. Кроме того, использование стандартного языка SQL облегчает переход от других систем управления базами данных, так как пользователи могут применить свои знания практически без изменений.

JavaScript — это язык программирования, который позволяет добавлять интерактивность на веб-страницы. С его помощью можно обрабатывать события, управлять элементами DOM (Document Object Model), делать асинхронные запросы к серверу (например, через AJAX) и многое другое. JavaScript широко используется для создания динамических и отзывчивых интерфейсов[13].

Преимущества JavaScript:

- кросс-платформенность;
- асинхронное программирование;
- обширная экосистема;
- удобство работы с данными;
- динамическая типизация;
- поддержка событий;
- сообщество и поддержка.

JavaScript обладает рядом значительных преимуществ, которые делают его популярным языком программирования в современной веб-разработке. Одним из главных достоинств является кросс-платформенность: JavaScript работает на всех современных браузерах и может использоваться как на клиентской, так и на серверной стороне благодаря среде выполнения Node.js.

Непосредственно при разработке веб-приложений, JavaScript поддерживает асинхронное программирование. Это позволяет выполнять операции без блокировки интерфейса, что значительно повышает отзывчивость и улучшает пользовательский опыт. Поддержка обширной экосистемы различных библиотек и фреймворков, таких как React, Angular и Vue.js, также упрощает разработку сложных и интерактивных интерфейсов, позволяя программистам сосредотачиваться на функциональности, а не на детальной реализации.

Работа с данными в JavaScript также становится более удобной благодаря его отличной совместимости с форматом JSON, который широко используется при взаимодействии с RESTful API и другими методами обмена данными. Динамическая типизация языка позволяет разработчикам быстрее писать код, однако требует внимательного отношения к типам, чтобы избежать ошибок.

Обработка событий — еще одна сильная сторона JavaScript. Он предоставляет мощные возможности для создания интерактивных интерфейсов, что делает веб-приложения более привлекательными и удобными для пользователей. Наконец,

стоит отметить активное сообщество разработчиков и регулярное обновление языка, что обеспечивает доступ к самым последним технологиям и поддержке, облегчая процесс обучения и разработки. Эти факторы в совокупности делают JavaScript одним из ключевых инструментов в арсенале современных веб-разработчиков.

VS Code — это популярный редактор кода, который поддерживает множество языков программирования и предоставляет многофункциональные инструменты для разработчиков. Он включает в себя поддержку расширений, подсветку синтаксиса, автозаполнение, отладку и интеграцию с системами контроля версий, такими как Git. VS Code является удобной средой для написания кода на HTML, CSS и JavaScript.

Figma — это облачный инструмент для дизайна интерфейсов и прототипирования. С его помощью дизайнеры могут создавать макеты веб-страниц, мобильных приложений и других продуктов. Figma позволяет командам совместно работать над проектами в реальном времени и предоставляет мощные инструменты для создания интерактивных прототипов, что облегчает процесс разработки[11].

Совместное использование этих технологий позволяет создать полноценное веб-приложение:

- HTML используется для создания структуры страницы;
- CSS оформляет страницу, обеспечивая привлекательный вид;
- JavaScript добавляет интерактивность, позволяя пользователям взаимодействовать с приложением;
- SQLite обеспечивает хранение данных, которые могут быть использованы в приложении;
- VS Code служит инструментом для разработки и написания кода;
- Figma помогает в проектировании пользовательского интерфейса и создании прототипов.

3.2 Описание алгоритма работы

Разработка программного обеспечения (ПО) — это процесс создания и поддержки программных приложений, который включает несколько этапов. Один из основных аспектов разработки ПО — это грамотное проектирование алгоритмов.

Алгоритм работы для информационного портала коммуникации между сотрудниками и ресторанами может быть следующим:

- пользователь авторизуется на сайте и попадает на главную страницу;
- пользователь может просматривать новости как своей так и другой компании;
- пользователь может просматривать общие коммуникации для всех ресторанов;
- пользователь может посмотреть доступные или прошедшие мероприятия;
- пользователь сам может заполнить форму и предложить новость.

Хранение данных в приложении будет реализовано с использованием базы данных для хранения информации о мероприятиях, пользователях, коммуникации, новостей. Для этого планируется использовать реляционную базу данных – SQLite.

Так как на сайте и на странице администратора представлены формы для отправки данных в бд, они реализованы следующим образом:

- создание пользовательского интерфейса;
- сбор данных из формы;
- отправка данных на сервер;
- обработка данных на сервере;
- работа с базой данных SQLite;
- получение и отображение данных для администратора;
- отрисовка данных на странице.

На первой стадии мы разработали HTML-форму изображенную на рисунке 1, которая содержит поля для ввода данных и кнопку для отправки данных в бд.

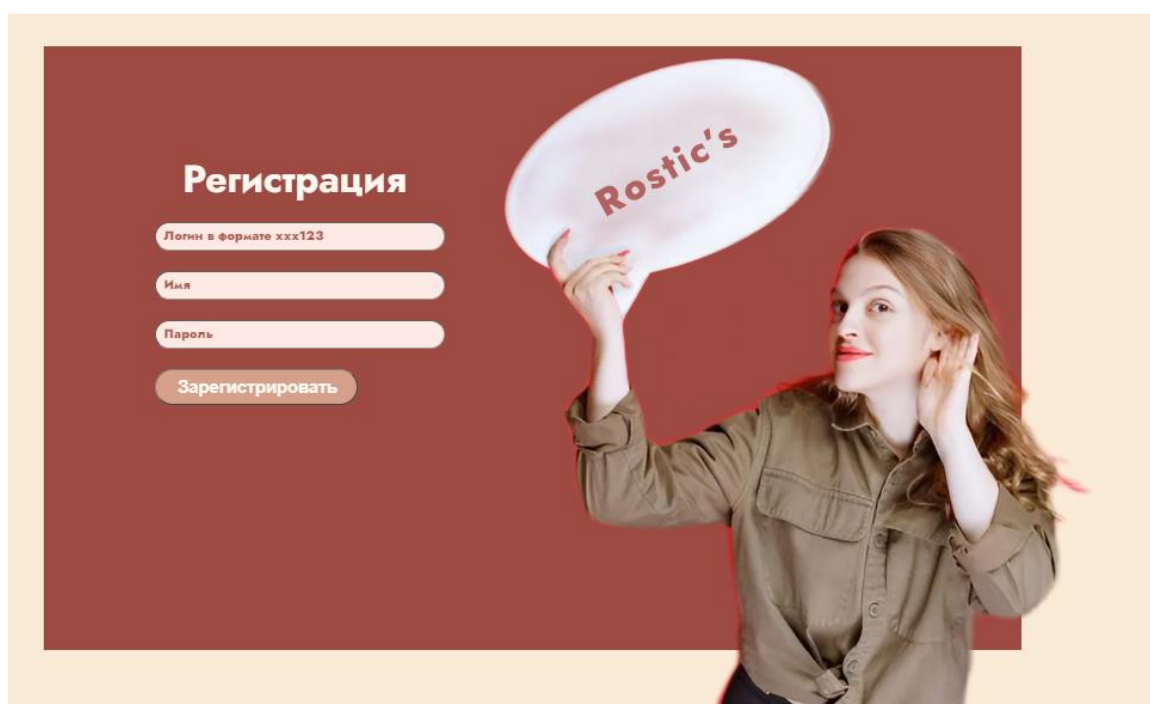


Рисунок 1 – Форма для добавления пользователей

После того как пользователь заполнит форму и нажмет кнопку отправки, JavaScript используется для захвата данных, введенных в полях формы[14]. Этот процесс включает в себя проверку корректности данных перед отправкой.

Для передачи собранных данных на сервер используется AJAX-запрос (обычно через fetch или XMLHttpRequest). Данные отправляются в формате JSON, который удобен для обработки на сервере.

На серверной стороне создается обработчик, который принимает данные от клиента. Этот обработчик может быть реализован на Node.js[15], Python или другом языке программирования, поддерживающем работу с SQLite. Здесь осуществляется подключение к базе данных и выполнение SQL-команды для вставки полученных данных в соответствующие таблицы.

После выполнения SQL-команды данные сохраняются в базе данных SQLite. Обработчик возвращает ответ клиенту, подтверждая успешное добавление данных или сообщая об ошибке.

```

1    const express = require("express");
2    const bodyParser = require("body-parser");
3    const session = require("express-session");
4    const { Sequelize, DataTypes } = require("sequelize");
5    const bcrypt = require("bcrypt");
6    const path = require("path");
7
8    const app = express();
9    const port = 3000;
10
11    // Настройка базы данных SQLite и подключение к ней
12    const sequelize = new Sequelize({
13        dialect: "sqlite",
14        storage: "db.sqlite",
15    });

```

Рисунок 2 – Подключение к БД

Для вывода данных из базы данных на страницу администратора создается еще один AJAX-запрос, который запрашивает данные из таблиц[6]. Сервер обрабатывает этот запрос, извлекает нужные данные из базы и отправляет их обратно клиенту, код для этого представлен на рисунке 3.

```

59    <script>
60        fetch('/users')
61        .then(response => response.json())
62        .then(users => {
63            const tbody = document.querySelector('#usersTable tbody');
64            users.forEach(user => {
65                const row = document.createElement('tr');
66                row.innerHTML = `
67                    <td>${user.id}</td>
68                    <td>${user.login}</td>
69                    <td>${user.name}</td>
70                    <td>${user.roleId}</td>
71                `;
72                tbody.appendChild(row);
73            });
74        })
75        .catch(error => console.error('Ошибка при загрузке пользователей:', error));
76    </script>

```

Рисунок 3 – Код отображения данных на странице

Полученные данные отображаются на странице администратора с помощью JavaScript, представлено на рисунке 4. Это может быть сделано путем динамического создания HTML-элементов или использования библиотек для работы с таблицами (например, DataTables), что позволяет отображать и форматировать данные в удобном для восприятия виде.

Rosfic's					Таблицы	Статистика	Новости	Коммуникации	Регистрация	Выйти
Список пользователей										
ID	Login	Name	Password	Role						
1	admin	admin	\$2b\$10\$13DBBZA2a15Ez71~fP20d6LAAkMF96cB8n~770Cn8TkGk8q7XaIS	2						
2	dan123	Daria	\$2b\$10\$6G97nDfamC4pT59QY2q0XGLyRR56ggrB85Ea8t3ytMOX2	1						
3	Test123	Test	\$2b\$10\$YmboWyVfV0ak510FEB96kO7nTMHjodQdPn0d62ga3~robTILKm	1						

Рисунок 4 – Отображение таблицы на странице администратора

Диаграмма классов (Приложение Г) представляет собой статическую модель проектируемого информационного портала для коммуникации между компанией-поставщиком и сетью ресторанов. Она визуализирует структуру системы, отображая ключевые классы, их атрибуты, методы и взаимосвязи. Диаграмма служит основой для дальнейшей разработки программного обеспечения, обеспечивая ясное понимание архитектуры и структуры данных. В ней представлены основные сущности предметной области и их взаимоотношения, что позволяет оценить целостность и корректность модели.

3.3 Описание интерфейса пользователя

Приложение предоставляет авторизованным пользователям доступ к новостям, важной информации (коммуникациям), форме обратной связи, расписанию мероприятий, информации о компании и её контактными данным.

Первое что видит пользователь заходя на сайт, это страница авторизации, где пользователь должен ввести свои данные, которые предоставляет компания для входа на сайт.

Форма авторизации представляет поле «Логин» – текстовое поле для ввода личного неповторяющегося имени пользователя. Валидация на стороне клиента: обязательное поле. Поле «Имя» – текстовое поле для ввода обычного имени пользователя. Поле «Пароль» – поле для ввода пароля. Тип поля: password для скрытия вводимых символов. Валидация на стороне клиента: обязательное поле. Кнопка «Войти» Активируется после заполнения обоих полей. При нажатии отправляет данные формы на сервер для проверки.

После авторизации пользователь попадает на главную страницу, на которой представлены: шапка сайта (рисунок 5) – логотип компании: при клике перенаправляет на главную страницу; ссылки на остальные страницы сайта ;кнопка «Выход»: расположена справа. При нажатии завершает сеанс пользователя и перенаправляет на страницу авторизации.

Rosfic's					Коммуникации	Новости	О нас	Мероприятия	Выйти
----------	--	--	--	--	--------------	---------	-------	-------------	-------

Рисунок 5 – Шапка сайта

Основной контент главной страницы это главная новость или коммуникация компании (рисунок 6). Так же на нем присутствует кнопка «перейти к ценностям», которая перенаправит пользователя на страницу самой новости.

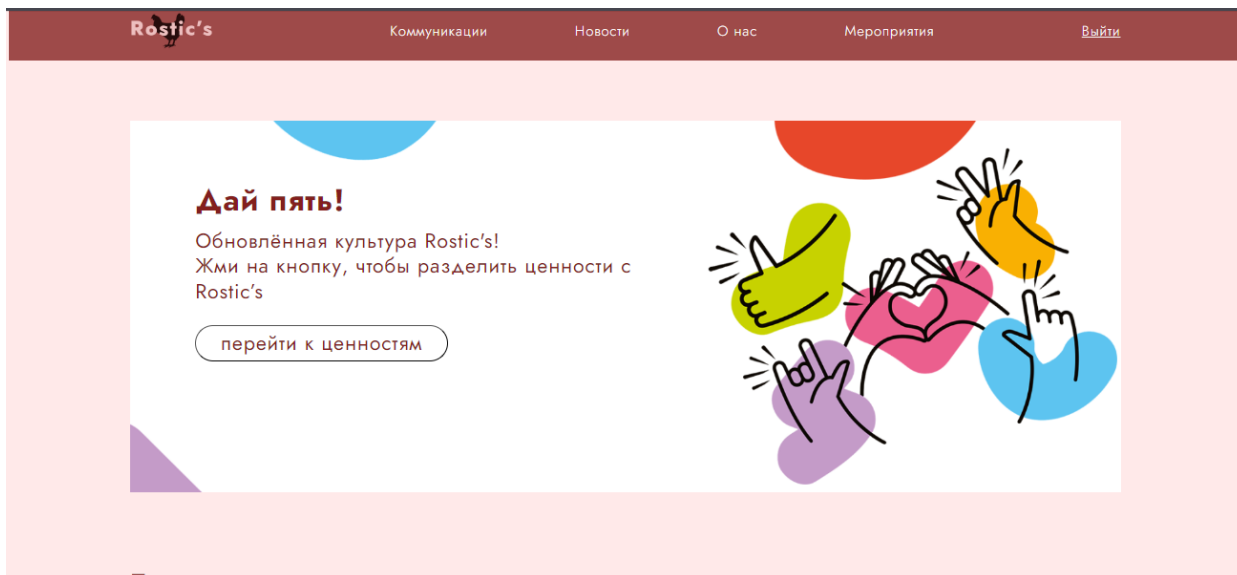


Рисунок 6 – Первый блок главной страницы

Блок «Главное», на котором представлены 3 последние новости компании, каждая новость включает заголовок новости и картинку (Приложение Д).

Блок «О нас» включает краткое описание и историю компании. Контактная информация или подвал сайта предоставляет контактную информацию и реквизиты компании (рисунок 7).

Контакты	Реквизиты
Название компании: ООО "ОренРест"	ИНН: 0278202810
Юр.адрес: 450078, Республика Башкортостан, г.	КПП: 027801001
Уфа, ул. Владивостокская д.2	ОГРН: 1130280050196
Телефон: 13-34-05	Организационно-правовая форма: Общество с
Email: office@grufa.ru	ограниченной ответственностью

Рисунок 7 – Подвал сайта

Краткая карта сайта представлена на рисунке 8.

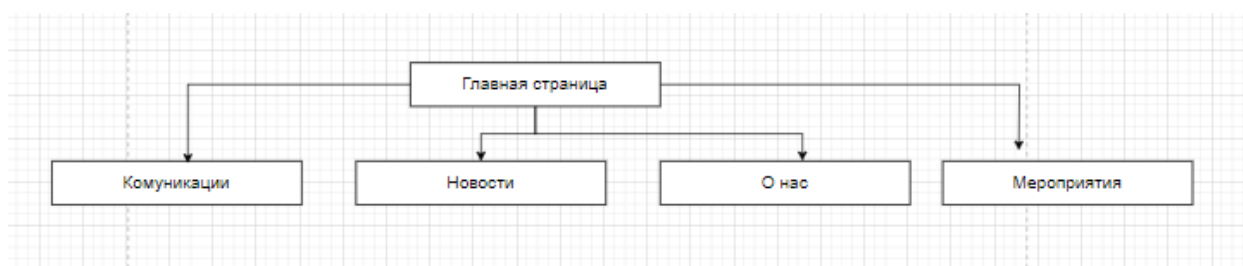


Рисунок 8 – Карта сайта

Также пользователь может оставить обратную связь или задать интересующий вопрос, форма для этого представлена на рисунке 9.

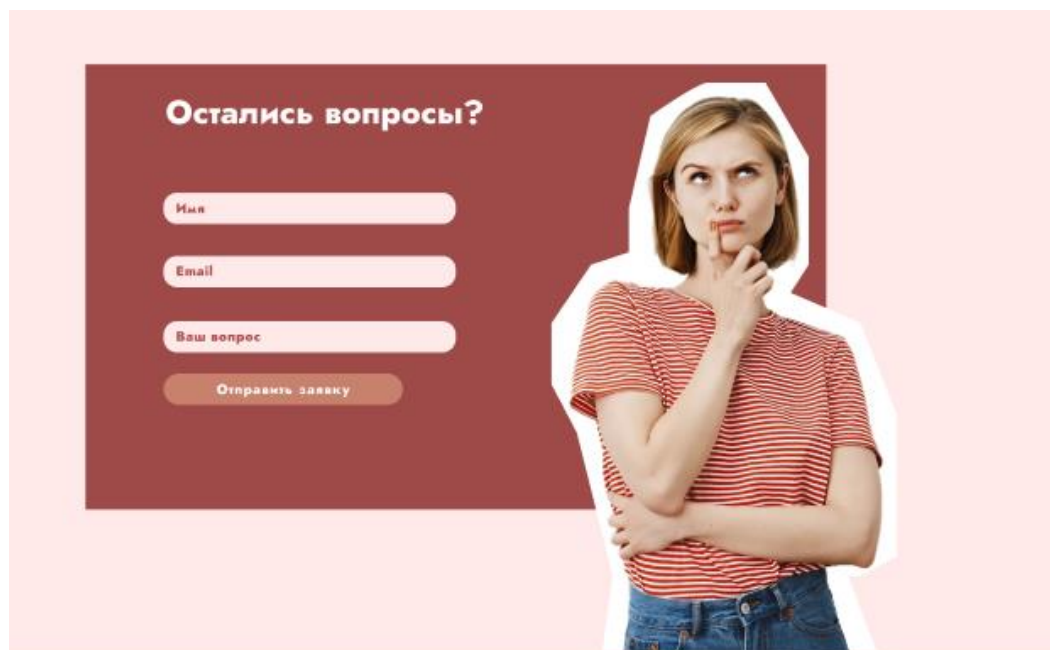


Рисунок 9 – Форма обратной связи

Формы обратной связи играют важную роль в коммуникации между пользователями и организациями. Они позволяют клиентам делиться своими мнениями, задавать вопросы или выражать пожелания, тем самым обеспечивая двустороннее взаимодействие. Основные характеристики таких форм включают простоту и доступность, чтобы пользователь мог быстро и удобно оставить свое сообщение.

В заключении можно отметить, что интерфейс является ключевым элементом взаимодействия пользователей с системой. Он должен быть интуитивно понятным, удобным и привлекательным, что способствует улучшению пользовательского опыта и повышению удовлетворенности. Адекватное оформление, логическая структура и оптимальная навигация играют важную роль в обеспечении эффективной работы с приложением или веб-сайтом.

Качественный интерфейс в значительной степени определяет успех продукта, так как он влияет на то, насколько легко пользователи могут достигать своих целей. Элементы дизайна, такие как цветовая гамма, типографика и размещение кнопок, также должны быть тщательно продуманы для создания гармоничного и функционального пространства.

4 Тестирование веб-приложения

4.1 План тестирования

Тестирование веб-приложения — это процесс проверки приложения для выявления ошибок и обеспечения его правильной работы.

Область тестирования: форма авторизации.

Тестирование функции авторизации имеет целью выявление возможных уязвимостей и ошибок в механизме аутентификации, который позволяет пользователям входить в систему, а также контроля доступа к различным уровням функционала. Эффективное тестирование авторизации включает в себя не только проверку корректности реализованных функций, но и оценку их устойчивости к потенциальным угрозам.

Цель данного плана тестирования — определить подходы и стратегии для тестирования функциональности авторизации на сайте. Это включает в себя разработку unit тестов, тест-кейсов для определенной логики работы системы, а также описание порядка интеграционного тестирования.

Тестирование будет охватывать следующие аспекты функциональности авторизации:

- вход с корректными данными;
- вход с некорректными данным;
- обработка ошибок;
- безопасность (например, защита от брутфорс-атак).

Unit тестирование будет сосредоточено на небольших, изолированных частях кода, таких как функции и методы, которые отвечают за авторизацию. Каждый unit тест должен проверять конкретный сценарий.

Unit-тесты: проверка успешной авторизации пользователя:

- входные данные: корректный логин и пароль;
- ожидаемый результат: возвращаемое значение true.

Тест функции обработки некорректного логина или пароля:

- входные данные: некорректный логин или пароль (например, неправильный пароль);
- ожидаемый результат: возвращаемое значение false.

Проверка авторизации с пустыми полями:

- входные данные;
- ожидаемый результат.

Так же будет проведено интеграционное тестирование. Интеграционное тестирование будет направлено на проверку взаимодействия различных компонентов системы, связанных с авторизацией, таких как база данных, API и интерфейс пользователя.

Порядок интеграционного тестирования:

1. Подготовка среды тестирования.
2. Тестирование работы с базой данных.

3. Тестирование взаимодействия с API.

4. Тестирование пользовательского интерфейса.

Для начала необходимо подготовить среду тестирования, что включает в себя настройку тестовой базы данных, заполненной тестовыми учетными записями пользователей, а также подготовку окружения с доступом к API авторизации. Далее следует протестировать работу с базой данных, удостоверившись в корректном сохранении данных пользователя в процессе регистрации и в верном получении данных пользователя при авторизации. Затем важно проверить взаимодействие с API, убедившись, что API возвращает корректный ответ при успешной авторизации с валидными данными, и правильно обрабатывает и возвращает ошибки в ситуациях использования некорректных данных. На заключительном этапе необходимо протестировать пользовательский интерфейс, чтобы подтвердить, что все элементы интерфейса отображаются и функционируют правильно, включая все поля ввода и интерактивные кнопки. Также важно убедиться, что сообщения об ошибках корректно отображаются на пользовательском интерфейсе в случаях неудачной авторизации[5].

Данный план тестирования предоставляет структурированный подход к тестированию функциональности авторизации на сайте. Выполнение всех тестов и их мониторинг помогут обеспечить.

4.2 Оценка результатов проведения тестирования

Тестирование веб-приложения — это процесс проверки приложения для выявления ошибок и обеспечения его правильной работы.

Область тестирования: форма авторизации.

Тестирование функции авторизации имеет целью выявление возможных уязвимостей и ошибок в механизме аутентификации, который позволяет пользователям входить в систему, а также контроля доступа к различным уровням функционала. Эффективное тестирование авторизации включает в себя не только проверку корректности реализованных функций, но и оценку их устойчивости к потенциальным угрозам.

Для тестирования данной функции используем тест-кейсы, которых представлены в таблицах 1-3.

Администратор, отвечающий за систему управления пользователями, играет ключевую роль в обеспечении функциональности процедуры регистрации. Его задача включает в себя контроль за корректностью собираемых данных, защиту от несанкционированного доступа и предотвращение злоупотреблений. Эффективное тестирование регистрации пользователей позволяет выявить возможные уязвимости системы, ошибки в логике работы, а также улучшить пользовательский опыт.

Для тестирования данной функции используем тест-кейсы, которых представлены в таблицах 4-5.

Протестируем успешную авторизацию пользователя со всеми верными данными, которая представлена в тест-кейсе в таблице 1.

Таблица 1 – Проверка успешной авторизации пользователя

№1	Проверка заполнения поля		
Ссылка на ТЗ	Исаева Д.С.	Высокий	Заявка
1. У пользователя есть зарегистрированная учетная запись. 2. Пользователь знает свои учетные данные (логин и пароль).		1. Открыть страницу сайта. 2. Ввести корректный логин в поле «Логин». 3. Ввести корректный пароль в поле «Пароль». 4. Нажать на кнопку «Войти».	
Пользователь успешно авторизуется и перенаправляется на главную страницу или личный кабинет. На странице отображается сообщение приветствия с именем пользователя.			

На рисунке 10 изображен корректный ввод данных в форму авторизации.

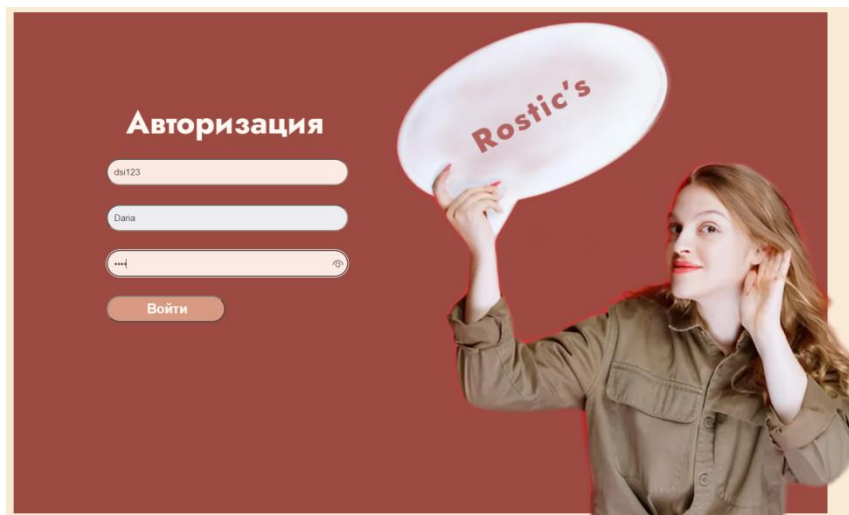


Рисунок 10 – Ввод данных в форму авторизации

После введения данных пользователь переходит на главную страницу сайта. Тестируем авторизацию с неверным логином пользователя. (Таблица 2).

Таблица 2 – Проверка авторизации с некорректным паролем

№2	Проверка заполнения поля		
Ссылка на ТЗ	Исаева Д.С.	Высокий	Заявка
1. У пользователя есть зарегистрированная учетная запись.		1. Открыть страницу сайта. 2. Ввести корректный логин в поле «Логин». 3. Ввести некорректный пароль в поле «Пароль». 4. Нажать кнопку «Войти».	
Появляется сообщение об ошибке, указывающее на некорректные учетные данные. Пользователь остается на странице авторизации.			

На рисунке 11 изображен ввод неверного логина но верного пароля.

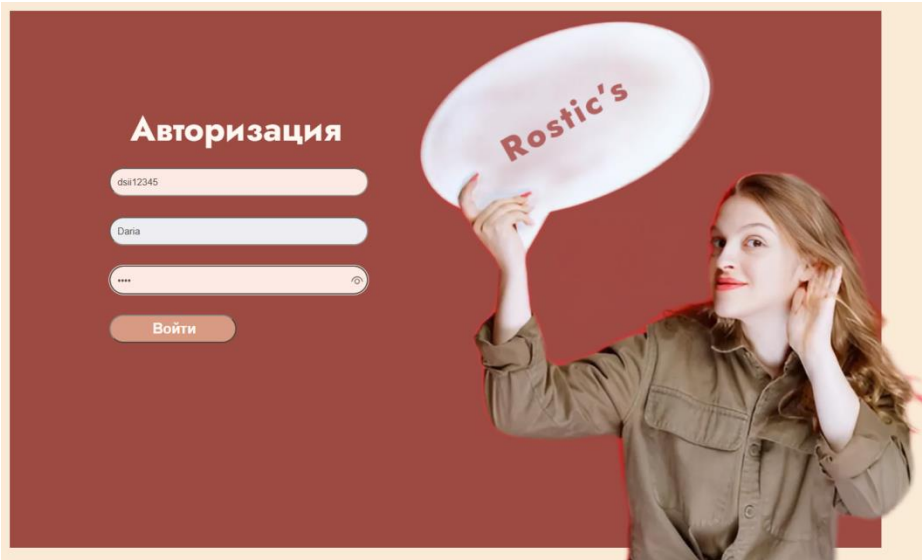


Рисунок 11 – Ввод некорректных данных в форму авторизации

После ввода неверных данных пользователь видит ошибку (рисунок 12).



Рисунок 12 – Ошибка

Протестируем функцию авторизации с пустыми полями. В таблице 3 представлен тест-кейс, который подробно описывает процесс проверки данной функции. Он включает в себя описание тестируемых полей, ожидаемые результаты и фактические выводы системы при попытке авторизации без ввода данных.

Таблица 3 – Проверка авторизации с пустыми полями

№3	Проверка заполнения поля		
Ссылка на ТЗ	Исаева Д.С.	Высокий	Заявка
1. Открытый сайт		1. Открыть страницу сайта. 2. Оставить поле «Логин» пустым. 3. Оставить поле «Пароль» пустым. 4. Нажать кнопку «Войти».	
Появляется сообщение об ошибке, указывающее на необходимость заполнения полей логина и пароля. Пользователь остается на странице авторизации.			

На рисунке 13 показан процесс работы с формой авторизации, где

пользователю предоставляется возможность оставить поля для ввода пустыми. Этот тест служит для проверки, как система обрабатывает отсутствие данных, что является критическим аспектом обеспечения безопасной работы приложения.

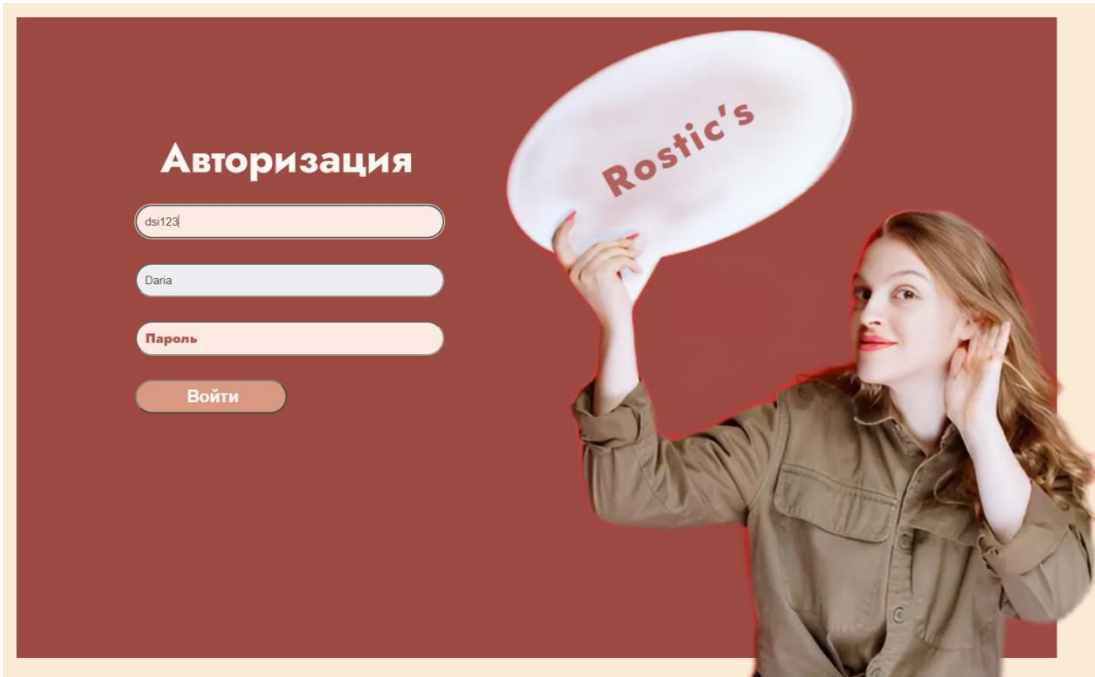


Рисунок 13 – Ввод некорректных данных в форму авторизации

На рисунке 14 мы видим, что пользователю не удалось войти так как он пропустил поле с паролем.

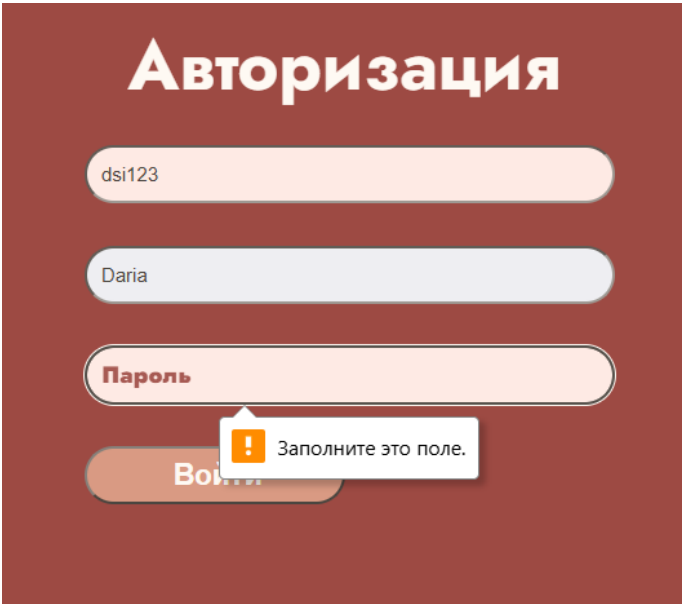


Рисунок 14 – Всплывающая подсказка о пустом поле

Регистрация пользователя является одним из ключевых компонентов любой веб-приложения или мобильного приложения. Эта функция не только позволяет пользователям создавать учетные записи, но и обеспечивает безопасность и управление их данными. Успешная реализация функции регистрации напрямую

влияет на пользовательский опыт, поэтому важно тщательно протестировать ее на наличие ошибок и уязвимостей.

Протестируем функцию регистрации пользователя (Таблица 4). Для успешной регистрации введем валидные данные.

Таблица 4 – Проверка успешной регистрации нового пользователя

№4	Проверка заполнения поля		
Ссылка на ТЗ	Исаева Д.С.	Высокий	Заявка
1. Администратор находится на странице регистрации сайта. 2. Администратор имеет доступ к интернету.		1. Авторизоваться от администратора, войти на страницу регистрации 2. Ввести валидное имя в поле «Имя пользователя». 3. Ввести валидный адрес электронной почты в поле «Логин». 4. Ввести валидный пароль в поле «Пароль». 5. Нажать кнопку «Зарегистрировать».	
Пользователь успешно зарегистрирован. В базе данных добавляется новая запись пользователя с корректными данными.			

На рисунке 15 вводим корректные новые данные.

Рисунок 15 – Введение данных для регистрации

На рисунке 16 представлена визуализация, иллюстрирующая успешную регистрацию пользователя в системе. Мы видим подтверждение того, что данные нового пользователя были корректно сохранены в базе данных. Этот этап является критически важным, так как он свидетельствует о том, что функция регистрации работает должным образом и обеспечивает надежность хранения информации.

Список пользователей

ID	Логин	Имя	Пароль	Роль
1	admin	admin	\$2b\$10\$IDaNB1CjD2zB55FCpPjfm3SUgUAQUBxEg5jVMV8UzZTvhZhcKla	2
2	dsi123	Daria	\$2b\$10\$GgbTbwCe0qD9PvnyIF38ul5Vhpyz6EmjLPsbePONGBbo1q8MkdAW	1
3	test123	test	\$2b\$10\$479dCjigYAgs4wclTvTuPBB2WxZRBmUZhZvoo1sED1nszRZf22	1

Рисунок 16 – Данные в таблице «Пользователи»

В таблице 5 представлен тест-кейс, который описывает процесс проверки функции регистрации при попытке использования существующего логина. Тест-кейс содержит описание условий тестирования, шаги, ожидаемый результат и фактический результат, что позволяет четко оценить корректность работы функции.

Таблица 5 – Проверка регистрации с уже существующим Логин

№5	Проверка заполнения поля		
Ссылка на ТЗ	Исаева Д.С.	Высокий	Заявка
1. Пользователь находится на странице регистрации сайта. 2. Пользователь имеет доступ к интернету. 3. В базе данных должна быть создана новая запись с данными, введенными пользователем. 4. Пользователь может войти в систему с использованием зарегистрированных данных.		1. Авторизироваться от администратора, войти на страницу регистрации 2. Ввести валидное имя в поле «Имя пользователя». 3. Ввести существующий адрес электронной почты в поле «Логин». 4. Ввести валидный пароль в поле «Пароль». 5. Нажать кнопку «Зарегистрировать».	
Ошибка регистрация.			

На рисунке 17 показано, что после попытки регистрации пользователя возникает ошибка. Это может свидетельствовать о том, что введенные данные не соответствуют требованиям системы или уже существует учетная запись с такими же данными.

Рисунок 17 – Вводим уже существующего пользователя

Заключение

Данная работа была посвящена разработке сайта для коммуникации между компанией и ресторанами. В ходе работы была достигнута поставленная цель – разработана концепция и создан прототип информационного портала, обеспечивающего эффективное взаимодействие между двумя целевыми группами: компаниями, заинтересованными в своевременной информированности сотрудников, и самими ресторанами, стремящимися работать по всем правилам и стандартам работы компании. Разработанный сайт призван стать удобным инструментом для обмена информацией, предоставления актуальных новостей и специальных предложений, а также обеспечения эффективной обратной связи.

Работа над проектом началась с проведения анализа предметной области. Были изучены существующие решения в сфере онлайн-коммуникаций между компаниями и ресторанами, выявлены их преимущества и недостатки. Особое внимание уделялось анализу потребностей целевых аудиторий, определению функциональных требований к разрабатываемому сайту. Этот этап позволил сформировать четкое представление о задачах, которые должен решать будущий портал.

На основе проведенного анализа было составлено техническое задание, которое детально описывало функциональные и нефункциональные требования к сайту. В техническом задании были определены основные модули сайта, их функциональность, требования к дизайну, безопасности и производительности. Техническое задание стало основополагающим документом для всех последующих этапов разработки.

Следующим этапом стала разработка дизайна сайта. При этом особое внимание уделялось принципам удобства навигации, информативности, привлекательности и простоты использования. Были разработаны макеты основных страниц сайта, продумана структура меню, выбрана цветовая палитра и шрифты. Дизайн сайта был разработан с учетом современных трендов веб-дизайна и требований целевой аудитории.

Описание хода разработки веб-сайта включало в себя выбор технологий и инструментов разработки, описание архитектуры сайта, а также поэтапное описание процесса создания отдельных модулей и функциональности. В качестве основной технологии разработки был выбран SQLite, HTML, CSS, Node.js, которые обеспечивают высокую производительность, гибкость и масштабируемость.

Процесс создания веб-сайта включал в себя верстку страниц, разработку frontend и backend функциональности, интеграцию с базой данных. На этом этапе были реализованы все основные функции сайта, определенные в техническом задании. Были созданы формы для регистрации пользователей, публикации новостей и специальных предложений, а также модуль обратной связи.

Разработка сайта осуществлялась с использованием современных программных средств, что позволило обеспечить высокое качество кода, оптимизировать производительность и обеспечить безопасность.

Для хранения данных сайта была создана база данных SQLite. Структура базы данных была спроектирована таким образом, чтобы обеспечить эффективное хранение и обработку информации о ресторанах, компаниях, заказах, новостях и других данных.

После завершения разработки был проведен этап тестирования веб-сайта. Тестирование проводилось по различным сценариям использования, с целью выявления и исправления ошибок, а также оценки производительности и безопасности сайта. Было проведено функциональное тестирование. Результаты тестирования показали, что разработанный сайт соответствует требованиям технического задания и готов к использованию.

В заключение следует отметить, что в рамках данной курсовой работы был разработан функциональный прототип сайта для коммуникации между компаниями и ресторанами. Разработанный сайт обладает рядом преимуществ, таких как удобный интерфейс, широкий функционал, высокая производительность и безопасность. В дальнейшем планируется добавить функционал личных кабинетов пользователей. Реализация этой задачи позволит сделать сайт еще более эффективным инструментом для коммуникации между компаниями и ресторанами.

Список использованных источников

- 1 Бейзер ДАО. Шаблоны проектирования доступа к данным / Бейзер, Б. — Санкт-Петербург : Питер, 2019. — 368 с. — Текст : непосредственный.
- 2 Бойков Основы проектирования пользовательских интерфейсов / Бойков, Н. П. — Москва : Издательство «Питер», 2020. — с. — Текст : непосредственный.
- 3 Вигерс, К. Разработка требований к программному обеспечению / К. Вигерс. — Москва : Издательство «Питер», 2021. — 768 с. — Текст : непосредственный.[Дата обращения: 17.12.2024]
- 4 Гарретт, Д. Д. Элементы опыта пользователя. Веб-дизайн для души / Д. Д. Гарретт. — Санкт-Петербург : Символ-Плюс, 2020. — 320 с. — Текст : непосредственный.[Дата обращения: 20.12.2024]
- 5 Гладун, Е. А. Современные технологии веб-дизайна / Е. А. Гладун. — Москва : МГУ, 2021. — 300 с. — Текст : непосредственный.[Дата обращения: 15.01.2025]
- 6 Дергачева, Н. В. Веб-программирование для начинающих / Н. В. Дергачева. — Казань : Казанский университет, 2019. — 280 с. — Текст : непосредственный.[Дата обращения: 5.01.2025]
- 7 Закас, Н. CSS. Каскадные таблицы стилей. Подробное руководство / Н. Закас, Д. Брэдбери. — Санкт-Петербург : Символ-Плюс, 2019. — 1152 с. — Текст : непосредственный. [Дата обращения: 21.12.2024]
- 8 Зиновьев, П. С. Интернет-технологии в бизнесе / П. С. Зиновьев. — Екатеринбург : УралГАУ, 2022. — 200 с. — Текст : непосредственный. [Дата обращения: 21.12.2024]
- 9 Канер, С. Тестирование программного обеспечения / С. Канер, Б. Фолк, Т. Нгуен. — Москва : Диалектика, 2022. — 544 с. — Текст : непосредственный.[Дата обращения: 21.12.2024]
- 10 Кроуфорд, С. JavaScript. Шаблоны / С. Кроуфорд, М. Дибров. — Санкт-Петербург : Питер, 2019. — 256 с. — Текст : непосредственный.[Дата обращения: 31.12.2024]
- 11 Круг, С. Не заставляйте меня думать. Веб-юзабилити и здравый смысл / С. Круг. — Санкт-Петербург : Символ-Плюс, 2019. — 288 с. — Текст : непосредственный.[Дата обращения: 25.12.2024]
- 12 Кузнецов, М. В. Разработка веб-приложений / М. В. Кузнецов, И. В. Симдянов. — Санкт-Петербург : БХВ-Петербург, 2021. — 608 с. — Текст : непосредственный.[Дата обращения: 20.12.2024]
- 13 Остервальдер, А. Построение бизнес-моделей / А. Остервальдер, И. Пинье. — Москва : Альпина Паблишер, 2020. — 288 с. — Текст : непосредственный.[Дата обращения: 17.12.2024]
- 14 Петров, А. В. Web-программирование: теория и практика / А. В. Петров. — Самара : Самарский университет, 2021. — 280 с. — Текст : непосредственный.[Дата обращения: 19.12.2024]
- 15 Соловьев, В. Ю. Разработка сайтов для бизнеса / В. Ю. Соловьев. — Уфа

- : БашГУ, 2020. — 300 с. — Текст : непосредственный.[Дата обращения: 28.12.2024]
- 16 Торнтон, К. Дизайн веб-форм / К. Торнтон, Л. Уилсон. — Санкт-Петербург : Питер, 2021. — 352 с. — Текст : непосредственный.[Дата обращения: 16.12.2024]
- 17 Фаулер, М. Шаблоны корпоративных приложений / М. Фаулер, Д. Райс. — Москва : Вильямс, 2023. — 528 с. — Текст : непосредственный.[Дата обращения: 28.12.2024]
- 18 Фленган, Д. JavaScript. Подробное руководство / Д. Фленган. — Санкт-Петербург : Символ-Плюс, 2019. — 1088 с. — Текст : непосредственный.[Дата обращения: 5.01.2025]
- 19 Хейлз, Д. Изучаем программирование на JavaScript / Д. Хейлз, К. Браун. — Санкт-Петербург : Питер, 2020. — 320 с. — Текст : непосредственный.[Дата обращения: 10.01.2025]
- 20 Хоган, Дж Разработка веб-приложений с помощью Node.js / Дж Хоган. — Москва : Вильямс, 2019. — 480 с. — Текст : непосредственный.[Дата обращения: 17.12.2024]