

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РЕСПУБЛИКИ КАЗАХСТАН
СЕВЕРО - КАЗАХСТАНСКИЙ УНИВЕРСИТЕТ ИМ. М. КОЗЫБАЕВА
Факультет инженерии и цифровых технологий
Кафедра «Информационно-коммуникационные технологии»

СЕМЕСТРОВЫЙ ПРОЕКТ

На тему: «Разработка информационной системы «Обслуживание заказов
клиентов»

Выполнили студенты
группы АПО-20-1

Сактаганов Э. К.
Рахметов М. М.
Ленгард С. С.
Ерболат Д. О.

Доктор PhD,
доцент кафедры ИКТ

Астапенко Н.В.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. АНАЛИТИЧЕСКАЯ ЧАСТЬ.....	4
1.1 Выбор методологии Agile.....	4
1.2 Создание структуры проекта в coda.io	6
1.3 Планирование процесса разработки и распределение работы	9
2 ПРОЕКТНАЯ ЧАСТЬ	13
2.1 Разработка UserStory	13
2.2 Структура сайта.....	15
ЗАКЛЮЧЕНИЕ.....	19
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ:	20

ВВЕДЕНИЕ

Сегодня, с развитием интернет-технологий, онлайн-шопинг становится все более популярным. В этой связи, создание веб-приложения для онлайн-магазина тортов является весьма перспективным проектом. Цель данного проекта - разработать удобное и простое в использовании веб-приложение для заказа тортов.

Для успешной реализации проекта будет использоваться методология Scrum. Эта методология подходит для быстрой разработки и управления проектами, особенно в сфере ИТ. С ее помощью команда сможет быстро реагировать на изменения и адаптироваться к новым требованиям заказчика, а также повысить эффективность работы.

Цели и задачи проекта:

Цель проекта - создание веб-приложения для онлайн-магазина тортов, которое позволит заказчику удобно и быстро выбрать торт по своим предпочтениям и осуществить заказ с доставкой на дом.

Для достижения цели проекта необходимо выполнить следующие задачи:

- Разработать функциональные требования к приложению;
- Разработать дизайн приложения;
- Реализовать функционал выбора и заказа торта;
- Реализовать функционал оплаты заказа;
- Реализовать функционал отслеживания статуса заказа;
- Протестировать приложение на соответствие требованиям;
- Запустить приложение в работу и обеспечить его поддержку.

При выполнении задач команда проекта будет придерживаться методологии Scrum, что позволит ей быстро реагировать на изменения в процессе и обеспечить высокое качество конечного продукта.

Объект исследования:

Объектом исследования является процесс создания веб-приложения для онлайн-магазина тортов с использованием методологии Scrum. В рамках проекта будут рассмотрены все этапы создания приложения, начиная с формирования требований и заканчивая запуском приложения в работу.

Предмет исследования:

Предметом исследования является разработка удобного и функционального веб-приложения для заказа тортов с использованием методологии Scrum. В рамках проекта будут рассмотрены процессы реализации функционала выбора и заказа торта, оплаты заказа, а также управления ассортиментом тортов. Основной целью исследования является создание приложения, которое будет удобным и простым в использовании для пользователей, а также позволит магазину эффективно управлять процессом заказов.

1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1 Выбор методологии Agile

Scrum — это методология управления проектами, которая была разработана для улучшения эффективности командной работы в области разработки программного обеспечения. Она основывается на итеративном и инкрементальном подходе к разработке, который позволяет быстро реагировать на изменения в процессе и обеспечивает более высокое качество конечного продукта.

Основной идеей Scrum является организация работы команды в виде коротких спринтов, продолжительностью обычно от 1 до 4 недель. В начале каждого спринта команда обсуждает свои цели и планирует работу на этот период, а в конце каждого спринта происходит демонстрация результатов и обсуждение дальнейших шагов.

Преимущества Scrum методологии включают:

- Гибкость. Scrum позволяет быстро реагировать на изменения в процессе разработки и адаптироваться к новым требованиям заказчика.
- Улучшенная коммуникация. Scrum акцентирует внимание на коммуникации в команде и регулярные обсуждения проблем, что помогает снизить вероятность ошибок и повысить эффективность работы.
- Прозрачность. Scrum методология позволяет заказчику быть в курсе состояния проекта на каждом этапе разработки и более четко контролировать процесс.
- Улучшенное качество продукта. Scrum ставит перед собой цель постоянного улучшения качества продукта и процесса разработки.

В целом, Scrum является эффективным методом управления проектами, который позволяет быстро достигать целей и улучшать качество конечного продукта. Однако, выбор методологии зависит от конкретных потребностей и характера проекта, поэтому Scrum может быть не подходящим в некоторых случаях.

Для успешной реализации данного проекта была применена методология Scrum - гибкий фреймворк управления проектами, который позволяет эффективно организовать работу команды и достичь результатов в условиях быстро меняющихся требований. Команда проекта была разбита на три ключевые роли: Владелец проекта, Скрам-мастер и два Разработчика.

Владелец проекта был ответственен за определение целей и приоритетов проекта, управление бюджетом и обеспечение соответствия результата ожиданиям заказчика. Скрам-мастер занимался организацией процесса разработки, контролировал выполнение задач и устранял препятствия на пути команды. Разработчики же занимались прямой работой по созданию продукта.

Каждый участник команды выполнял свою роль и был ответственен за достижение результатов в соответствии со своей областью компетенции. Распределение ролей и их функциональные обязанности были представлены на

рисунке 1.1, что обеспечило ясность и понимание каждым членом команды своей роли и вклада в проект.



Рисунок 1.1 Команда разработчиков

Для упорядочивания рабочего процесса команда использовала Scrum-доску (рисунок 1.2), на которой отображались задачи, выполненные работой, и те, на которые следовало обратить внимание в ближайшее время. Это позволило команде быстро и эффективно управлять своими задачами и не терять важные детали при выполнении проекта.

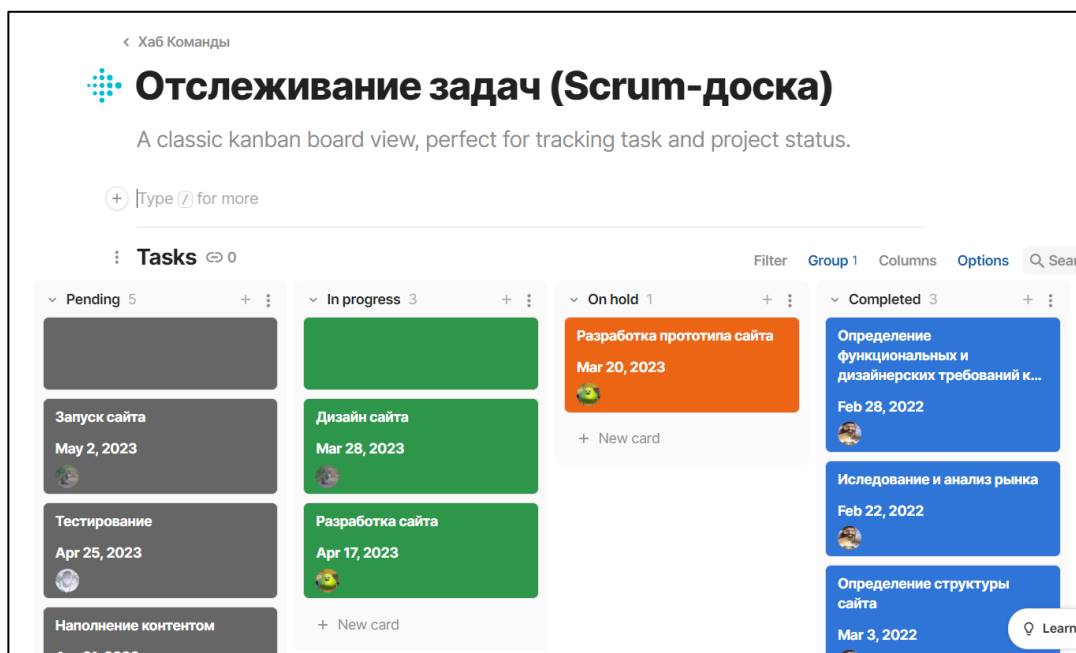


Рисунок 1.2 Scrum-доска

Для упорядочивания рабочего процесса команда использовала Scrum-доску, которая была разделена на 4 столбца: "В ожидании", "В процессе", "На удержании" и "Выполнено". Каждая задача была представлена в виде карточки, которая содержала информацию о том, какая работа должна быть выполнена, кто отвечает за ее выполнение и когда она должна быть завершена.

Когда задача была начата, карточка перемещалась в столбец "В процессе". Если выполнение задачи было приостановлено, она перемещалась в столбец "На удержании". Когда задача была завершена, карточка перемещалась в столбец "Выполнено". Это позволяло команде эффективно отслеживать прогресс работы и понимать, на какие задачи следует сделать упор в ближайшее время.

1.2 Создание структуры проекта в coda.io

В начале разработки проекта будет использован сайт coda.io для контроля проекта. Основные участники группы будут указаны, после чего будут распределены роли участников для реализации проекта. Будут определены основные индивидуальные задачи, установлены сроки и определен уровень выполненной работы.

Coda.io — это универсальный инструмент для создания документов, таблиц и баз данных, которые можно настраивать под свои нужды. Создание хаба проекта, отслеживание выполнения задач и таймлайн проекта на Coda.io может включать следующие шаги:

Создание документа проекта (рис 1.3): Начните с создания нового документа в Coda.io и назовите его в соответствии с названием проекта (например, "Хаб проекта тортов"):

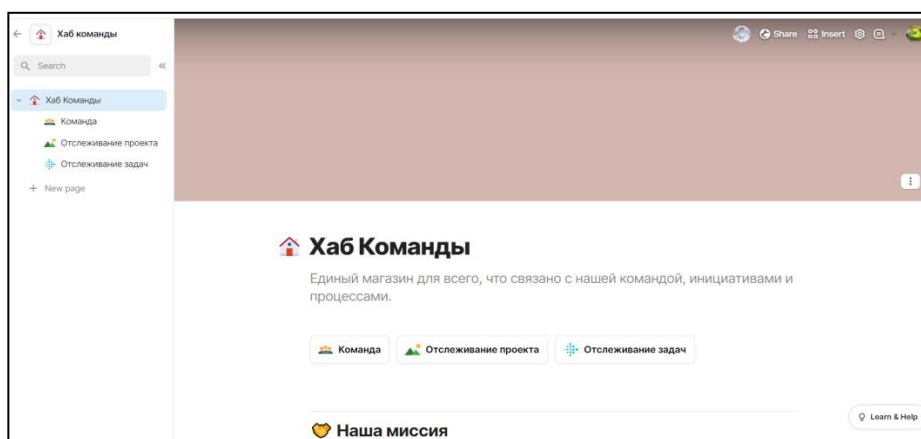


Рисунок 1.3 Хаб команды

Ознакомительная информация о команде (рис 1.4): на первой странице документа добавьте краткое описание проекта, его цели и основных задач. Также добавьте список:

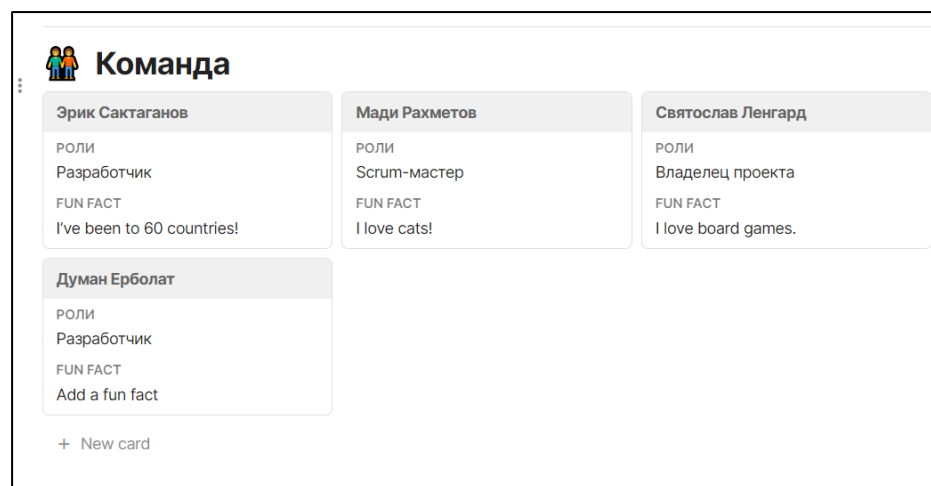


Рисунок 1.4 Ознакомительная информация о команде

Команда проекта (рис 1.5): Создайте новую страницу в документе и назовите ее "Команда проекта". Здесь можно создать таблицу с информацией о каждом члене команды, включая имя, роль, контактные данные, навыки и опыт работы.

Состав команды

1

Filter

Sort

Columns 1

Options

Search





	Имя	Фамилия	Имя фамилия	Роли	Дата начала	Email	
1	Эрик	Сактаганов	Эрик Сактаганов	Разработчик	Mar 16, 2023	 kitlian5050@gmail.com	
2	Мади	Рахметов	Мади Рахметов	Scrum-мастер	Mar 16, 2023	 gastonmra@gmail.com	
3	Святослав	Ленгард	Святослав Ленга...	Владелец проекта	Mar 16, 2023	 mcscovorodka@gmail.com	
4	Думан	Ерболат	Думан Ерболат	Разработчик	Mar 16, 2023	 duman.sko.kz@gmail.com	
4 Count							

Рисунок 1.5 Команда проекта

Задачи для создания проекта (рис 1.6): Создайте еще одну страницу и назовите ее "Задачи проекта". На этой странице создайте таблицу со столбцами: "Задача", "Ответственный", "Статус", "Срок выполнения" и "Примечания". Заполните таблицу всеми задачами, связанными с проектом, и назначьте ответственных.

Имя задачи	Исполни	Начало	Конец	Стадия	Прогресс	Details
1 Исследование и анализ рынка		Feb 19	Feb 22	Выполнен	100	
2 Определение функциональных и дизай...		Feb 23	Feb 28	Выполнен	100	
3 Определение структуры сайта		Mar 1	Mar 3	Выполнен	100	
4 Разработка прототипа сайта		Mar 6	Mar 20	Выполнен	100	
5 Дизайн сайта		Mar 21	May 28	Выполнен	100	
6 Разработка сайта		Mar 24	Apr 14	Выполнен	100	
7 Наполнение контентом		Apr 17	Apr 21	Выполнен	100	
8 Тестирование		Apr 12	Apr 25	Выполнен	100	
9 Маркетинг и продвижение		Apr 16	Apr 29	Выполнен	100	
10 Запуск сайта		May 1	May 1	Действующий	50	

Рисунок 1.6 Задачи для создания проекта

Отслеживание выполнения задач (рис 1.7): на странице "Задачи проекта" можно использовать фильтры и сортировку для отслеживания прогресса выполнения задач. Например, вы можете добавить фильтр который покажет только невыполненные задачи или задачи, которые уже просрочены.

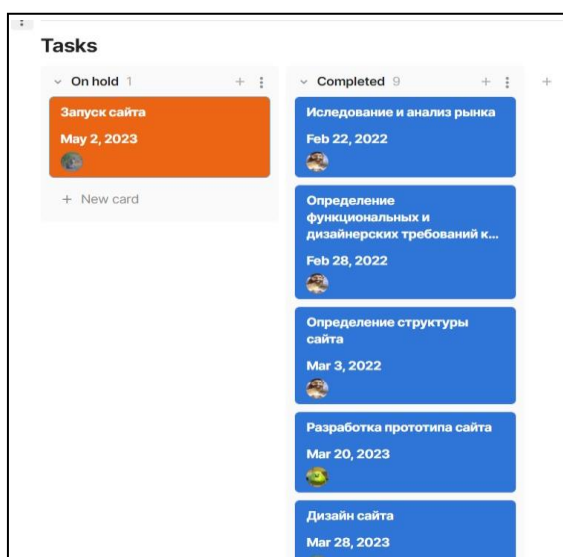


Рисунок 1.7 Отслеживания выполнения задач

Таймлайн проекта (рис 1.8): Создайте новую страницу с названием "Таймлайн проекта". Здесь вы можете использовать гantt-диаграмму для визуализации сроков выполнения задач, зависимостей между ними и прогресса. Для этого создайте таблицу со столбцами "Задача", "Начало", "Конец" и "Ответственный". Затем, используя функцию "Timeline" (Таймлайн) в Coda.io, преобразуйте таблицу в гantt-диаграмму.

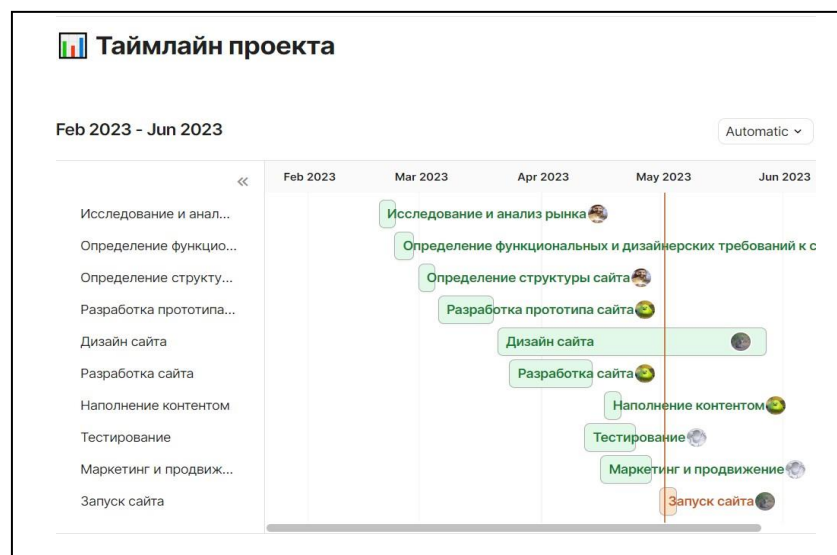


Рисунок 1.8 Таймлайн проекта

1.3 Планирование процесса разработки и распределение работы

Процесс разработки проекта разделен на 8 шагов, каждый из которых содержит своё описание.

Шаг 1: Понимание потребностей бизнеса

- Встреча с владельцем, чтобы узнать его требования и ожидания от сайта. В ходе встречи можно обсудить такие вопросы, как цели сайта, его главную аудиторию, функциональные требования и дизайн.
- Определить основные цели сайта и метрики, которые будут использоваться для измерения успеха. Например, это может быть увеличение продаж, увеличение посетителей на сайте, улучшение конверсии и т.д.

Шаг 2: Исследование рынка

- Изучить конкурентов и их сайты, чтобы понять, что делает их успешными или неуспешными. Стоит обратить внимание на дизайн, функциональность, пользовательский опыт, ценообразование и прочие факторы.
- Определить, какие функции и возможности необходимы для конкурентоспособности. На основе анализа конкурентов можно сформировать список функций и возможностей, которые необходимы для того, чтобы сайт был конкурентоспособным.

Шаг 3: Проектирование сайта

- Создать макет сайта с помощью инструментов Tilda. На этом этапе можно определить общий стиль дизайна, расположение элементов, основные цвета и шрифты.
- Определить все разделы сайта, их функции и взаимосвязь. Например, это могут быть разделы "Каталог", "О нас", "Контакты", "Корзина" и т.д.
- Создать списки задач для всех разделов. Для каждого раздела можно

создать список задач, необходимых для его реализации.

Шаг 4: Разработка сайта

- Разделить работу между двумя разработчиками, определив задачи для каждого из них. Например, один разработчик может работать над фронтендом, а другой - над бэкендом.

- Начать работу с главной страницы и переходить к другим разделам по мере готовности. Важно своевременно делать обзоры и согласовывать изменения с владельцем проекта.

- Своевременно делать обзоры, чтобы убедиться, что проект соответствует требованиям владельца.

Шаг 5: Тестирование и корректировка (продолжение)

- Протестировать все функции сайта, убедившись, что они работают должным образом. Для этого можно использовать автоматизированные и ручные тесты, чтобы проверить работу сайта в разных браузерах и на разных устройствах.

- Определить все недочеты и ошибки и скорректировать их. Если в ходе тестирования были обнаружены ошибки, их нужно исправить и провести повторное тестирование, чтобы убедиться в работоспособности сайта.

- Протестировать сайт с помощью реальных клиентов. Можно предложить тестирование сайта небольшой группе реальных пользователей, чтобы получить обратную связь и исправить выявленные проблемы.

Шаг 6: Запуск сайта

- Развернуть сайт на хостинге. Для этого нужно выбрать подходящий хостинг и настроить его для размещения сайта.

- Протестировать работу сайта после развертывания на хостинге. После развертывания на хостинге нужно убедиться, что сайт работает должным образом и доступен для всех пользователей.

- Запустить сайт на продакшен. После успешного тестирования и развертывания сайта на хостинге, его можно запустить на продакшене и начать привлекать пользователей.

Шаг 7: Поддержка и обслуживание

- Убедиться, что сайт работает без проблем после запуска. Необходимо регулярно проверять работу сайта и мониторить его производительность, чтобы предотвратить возможные проблемы.

- Обновлять сайт регулярно и добавлять новые функции по мере необходимости. На этом этапе можно добавлять новые функции, улучшать дизайн и выполнять другие изменения, которые улучшат работу сайта.

- Следить за метриками и результатами, чтобы оптимизировать сайт и улучшать его производительность. Можно использовать различные инструменты аналитики, чтобы отслеживать метрики, такие как конверсия, посещаемость и другие, и принимать меры для их улучшения.

Шаг 8: Оценка успеха (продолжение)

- Оценить метрики, установленные на шаге 1, и сравнить их с

результатами после запуска сайта. Необходимо провести анализ метрик и сравнить их с теми, которые были определены на этапе планирования. Если метрики не достигли требуемого уровня, нужно выявить причины и принять меры для их улучшения.

- Разработать планы дальнейшего развития. На основе оценки результатов можно сформировать планы дальнейшего развития сайта, включающие в себя новые функции, улучшение пользовательского опыта и прочее. Важно также учитывать мнение пользователей и их обратную связь, чтобы улучшать сайт в соответствии с их потребностями.

На каждом этапе разработке задействуется определенный круг лиц.

Ниже приведено, какие лица присутствуют на каждом этапе.

Шаг 1: Понимание потребностей бизнеса.

- Владелец - определял требования и ожидания от сайта.
- Скрам-мастер - участвовал во встрече и обеспечивал коммуникацию между командой и владельцем.
- Разработчики - принимали участие в обсуждении и понимании требований, чтобы определить, какие функции и возможности необходимы для сайта.

Шаг 2: Исследование рынка.

- Владелец - предоставлял информацию о конкурентах и их сайтах.
- Разработчики - изучали конкурентов и анализировали их сайты, чтобы понять, что делает их успешными или неуспешными.
- Скрам-мастер - обеспечивал связь между командой и владельцем, чтобы уточнить требования и обсудить результаты исследования.

Шаг 3: Проектирование сайта.

- Разработчики - создавали макет сайта и определяли все разделы сайта, их функции и взаимосвязь.
- Скрам-мастер - координировал работу команды и обеспечивал связь между разработчиками и владельцем.
- Владелец - утверждал макет сайта и давал рекомендации по его улучшению, если это необходимо.

Шаг 4: Разработка сайта.

- Разработчики - разрабатывали сайт, выполняли задачи и координировали свою работу.
- Скрам-мастер - обеспечивал связь между разработчиками и владельцем, управлял процессом разработки и координировал работу команды.
- Владелец - утверждал каждый этап разработки сайта, делал замечания и рекомендации, если это необходимо.

Шаг 5: Тестирование и корректировка.

- Разработчики - тестировали сайт, исправляли ошибки и дорабатывали функционал.
- Скрам-мастер - обеспечивал контроль качества и координировал работу команды.

- Владелец - проверял работу сайта и давал обратную связь, чтобы помочь команде исправить ошибки и доработать функционал.

Шаг 6: Запуск сайта (продолжение).

- Разработчики - разворачивали сайт на хостинге, проводили финальное тестирование, чтобы убедиться в работоспособности сайта.
- Скрам-мастер - обеспечивал контроль качества и координировал работу команды.
- Владелец - утверждал запуск сайта на продакшен, проверял его работу и давал обратную связь, если необходимо.

Шаг 7: Поддержка и обслуживание.

- Разработчики - следили за работой сайта и мониторили его производительность, обновляли сайт и добавляли новые функции, если это необходимо.
- Скрам-мастер - координировал работу команды и обеспечивал связь между разработчиками и владельцем.
- Владелец - давал рекомендации по улучшению сайта и оценивал его успешность на основе метрик, установленных на предыдущих этапах.

Шаг 8: Оценка успеха.

- Владелец - оценивал результаты проекта и проводил анализ метрик, сравнивая их с теми, которые были определены на этапе планирования. На основе этой оценки владелец определял успешность проекта и давал рекомендации по дальнейшему развитию сайта
- Разработчики и скрам-мастер - обеспечивали поддержку и обслуживание сайта, реализовывали изменения и добавляли новые функции в соответствии с рекомендациями владельца.

2 ПРОЕКТНАЯ ЧАСТЬ

В данном отчете представлен анализ проекта по созданию и внедрению сайта для продажи тортов, разработанного с использованием платформы Tilda. Эта платформа была выбрана благодаря ее удобству, гибкости и интуитивно понятному интерфейсу, что позволяет быстро создавать привлекательные и функциональные сайты без необходимости знания программирования. В рамках отчета описываются основные этапы проекта, достижения и результаты, а также особенности работы с платформой Tilda.

2.1 Разработка UserStory

User Story 1: Каталог товаров

Как пользователь, я хочу просмотреть имеющиеся на сайте каталог тортов, чтобы я мог увидеть фотографии каждого торта и выбрать тот торт, который наиболее подходит для меня.

Порядок действий:

- Открыть сайт;
- Нажать на кнопку «Торты»;
- Перейти к разделу тортов на сайте;
- Посмотреть результаты, где будет название торта и изображение, отображающий соответствующий торт.

Критерии приемки: когда пользователь открывает, он должен видеть список доступных тортов с их фотографиями. Он должен видеть подробное описание торта.

User Story 2: Покупка торта

Как пользователь, я хочу купить торт, чтобы заказать и получить его в удобном для меня месте.

Порядок действий:

- Открыть сайт;
- Нажать на кнопку «Торты»;
- Перейти к разделу тортов на сайте;
- Нажать на карточку «Торта»;
- Нажать кнопку «Добавить в корзину»;
- При необходимости свернуть форму «Заказа» и добавить следующий торт;
- Перейти на иконку «Корзины»;
- Заполнить форму заказа торта, утвердив количество тортов, записав данные в формы «Ваше имя», «Ваш Email», «Ваш телефон», «Ваш комментарий»;
- Нажать кнопку «Отправить заказ».

Критерии приемки: когда я нахожу торт, который мне нравится, я

должен иметь возможность добавить его в корзину, заполнить форму, и отправить мой заказ.

User Story 3: Информация о товаре

Как пользователь, я хочу увидеть список ингредиентов торта, срок годности и условия хранения, чтобы убедиться, что он соответствует моим предпочтениям и требованиям.

Порядок действий:

- Открыть сайт;
- Нажать на кнопку «Торты»;
- Перейти к разделу тортов на сайте;
- Нажать на карточку «Торта»;
- Увидеть на форме изображение торта и соответствующее описание торта.

Критерии приемки: когда я открываю описание торта, я должен видеть список ингредиентов, используемых в торте. Я должен иметь возможность узнать, какой срок годности и условия хранения данного торта.

User Story 4: Информация о доставке

Как пользователь, я хочу увидеть информацию о доставке, чтобы выбрать удобный для меня способ получения торта или узнать время работы доставки.

Порядок действий:

- Открыть сайт;
- Нажать на кнопку «Доставка»;
- Перейти к разделу «Доставка» на сайте;
- Ознакомиться с информацией о доставке: стоимости и сроках доставки.

Критерии приемки: когда я оформляю заказ, я должен иметь возможность узнать стоимость доставки. Я должен получить информацию о сроках доставки моего заказа.

User Story 5: Связь с магазином

Как пользователь, я хочу иметь возможность связаться с кондитерской, чтобы получить ответы на мои вопросы и решить возможные проблемы.

Порядок действий:

- Открыть сайт;
- Нажать на кнопку «Контакты»;
- Перейти к разделу «Контакты» на сайте;
- Посмотреть номер телефона магазина или электронную почту, на которую можно обратиться.

Критерии приемки: на сайте кондитерской я должен иметь возможность найти контактную информацию для связи с магазином, такую как телефон, адрес электронной почты или любая другая форма обратной связи. Я должен получить быстрый и профессиональный ответ на свои вопросы.

2.2 Структура сайта

Проект «Магазин тортов» состоит из одной главной страницы, разделенной на 4 блока. Мы разделили её так для удобства представления в отчете и для удобной навигации по сайту. Первый блок состоит из названия магазина, навигационных кнопок и небольшого описания. Блок представлен на рисунке 2.1.

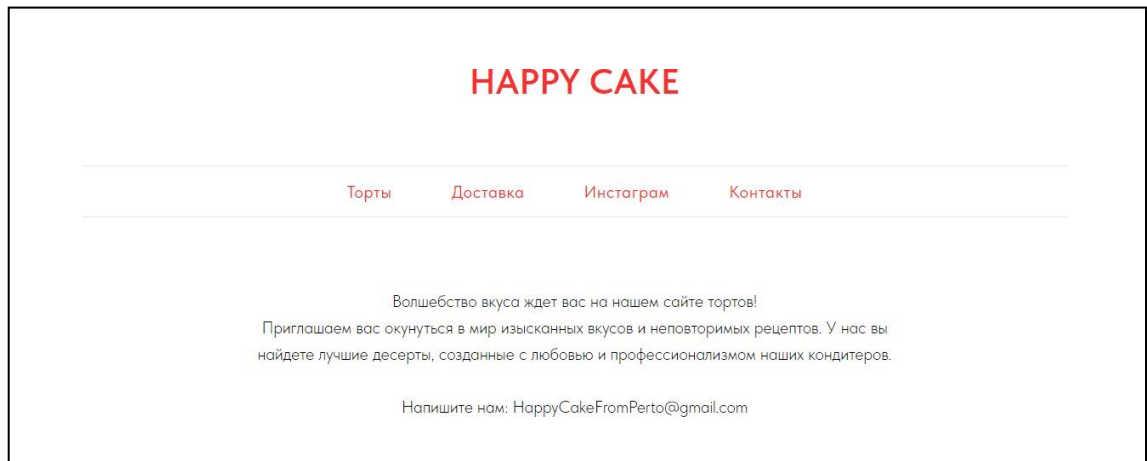


Рисунок 2.1 Шапка сайта

Второй блок магазина содержит каталог предлагаемых товаров. Блок занимает большую половину сайта, так как содержит много карточек с тортами. Сама карточка состоит из картинки, название, цена и кнопки

«Добавить в корзину». Каталог представлен в соответствии с рисунком 2.2.

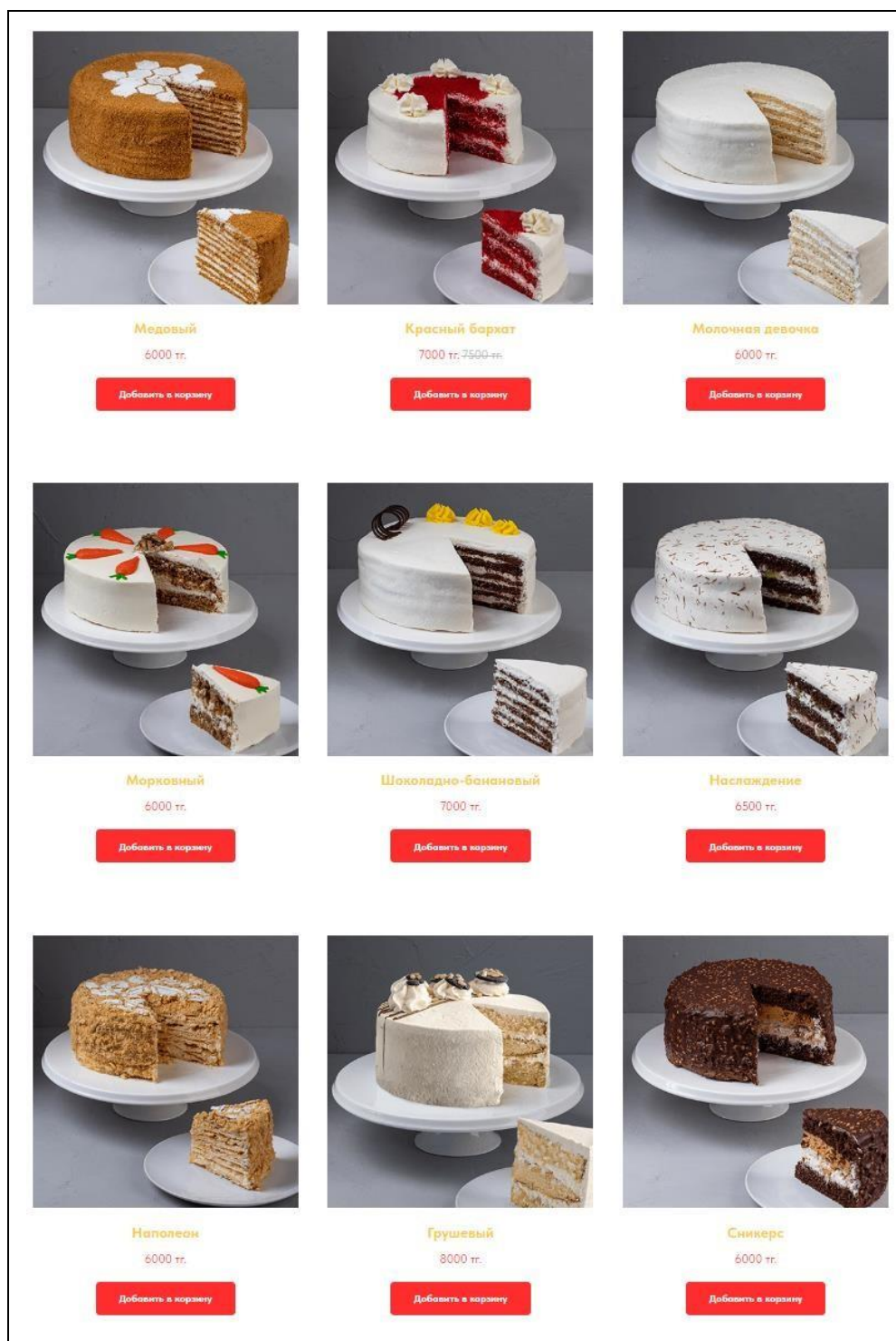


Рисунок 2.2 Каталог магазина

Следующий, третий блок содержит в себе ссылку на инстаграм, фотогалерею и информацию о доставке. В блок ведут две кнопки из навигационного меню, а именно кнопки «Инстаграм» и «Доставка». Блок представлен на рисунке 2.3.

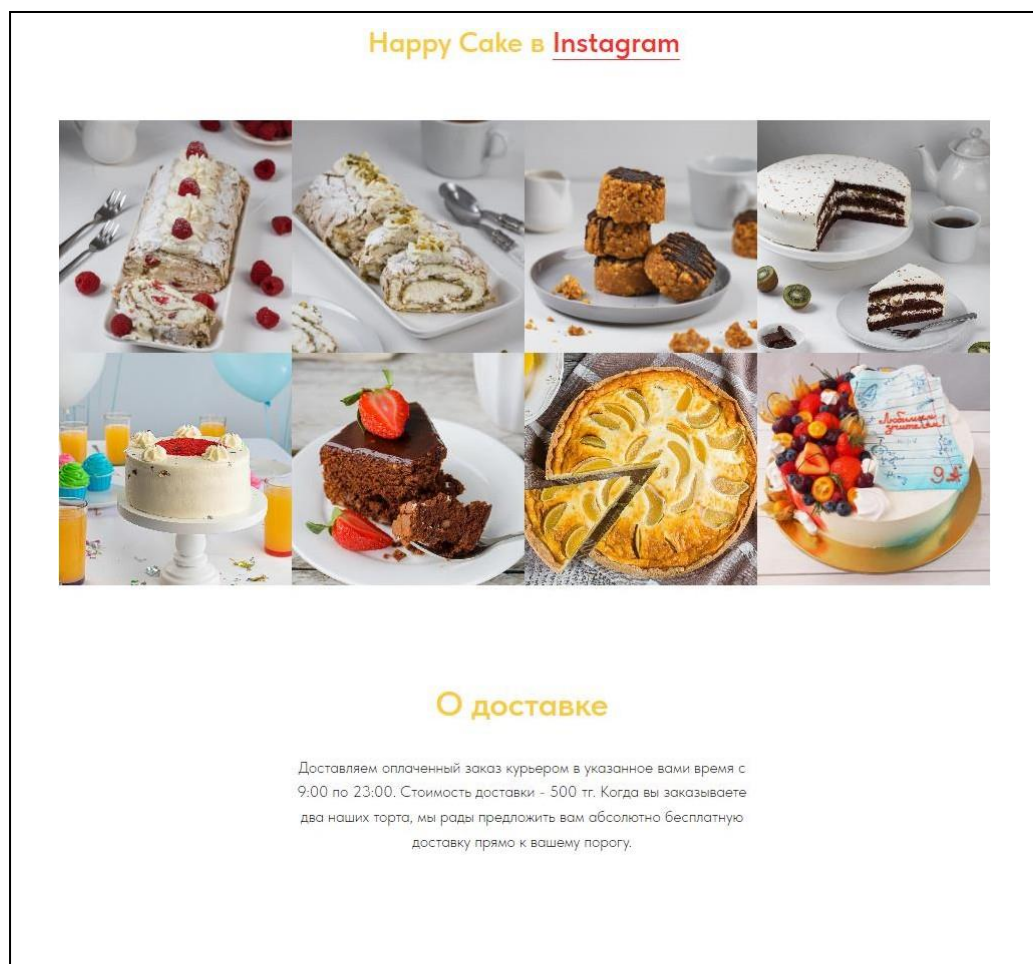


Рисунок 2.3 Инстаграм, фотогалерея и доставка

Последний блок является футером сайта. Содержит в себе контакты для связи с магазином и copyright строку. В него так же ведет кнопка из навигационного меню, называемая «Контакты». Футер представлен на рисунке 2.4.

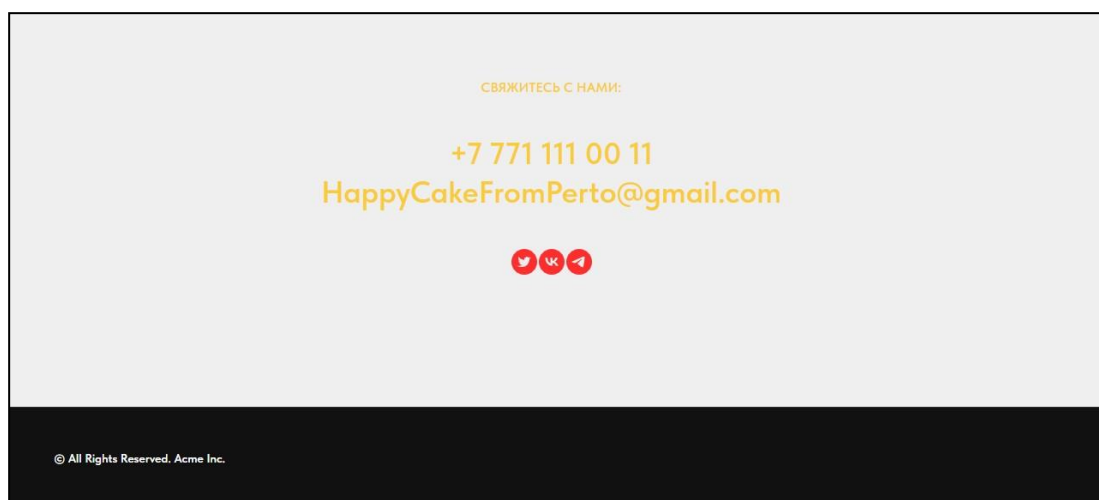


Рисунок 2.4 Футер сайта

При нажатии на кнопку «Добавить в корзину» нас встречает всплывающее окно с полным описанием торта, включая его состав, срок годности, пищевую, энергетическую ценность и условия хранения. В окне также дублируется кнопка «Добавить в корзину» и изображение товара. Окно представлено на рисунке 2.5.

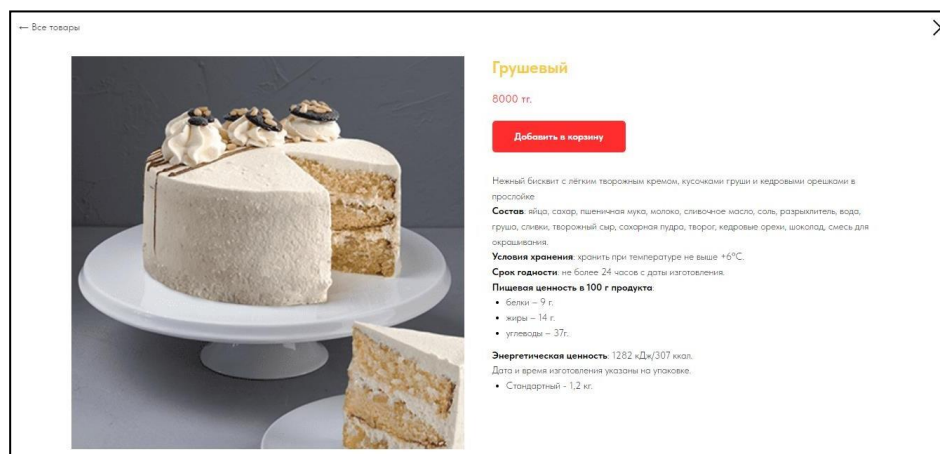


Рисунок 2.5 Окно с подробной информацией о товаре

Далее, после нажатия на кнопку, нас переводит в корзину, где мы можем заполнить данные и оформить заказ. По желанию, мы можем выйти и добавить в корзину еще пару тортов. Все это представлено в соответствии с рисунком 2.6.

Ваш заказ:

	Красный бархат	1	7 000 тг.	✕
	Молочная девочка	1	6 000 тг.	✕
	Наслаждение	1	6 500 тг.	✕

Сумма: 19 500 тг.

Ваше имя

Ваш Email

Ваш телефон

Ваш комментарий

Оформить заказ

Рисунок 2.6 Корзина

ЗАКЛЮЧЕНИЕ

В настоящее время многие компании стремятся удовлетворить потребности своих клиентов и расширить свой бизнес, используя онлайн-магазины. Создание веб-приложения для онлайн-магазина тортов с использованием методологии Scrum может быть отличным решением для этой цели.

Проект направлен на разработку функционального приложения для заказа тортов с доставкой на дом. Для успешного выполнения проекта необходимо пройти несколько этапов, начиная с формирования требований, дизайна и разработки приложения, тестирования и запуска приложения в работу. Одним из ключевых моментов в процессе реализации является обеспечение поддержки приложения после его запуска.

Использование методологии Scrum обеспечивает гибкость и быструю реакцию на изменения и позволяет команде быстро адаптироваться к новым требованиям заказчика, что в свою очередь может существенно повысить эффективность работы и сократить время разработки.

Результатом выполнения проекта станет создание приложения, которое будет обеспечивать удобство и простоту в использовании для пользователей. Приложение позволит клиентам легко и быстро выбирать и заказывать торты с доставкой на дом, а магазину - эффективно управлять процессом заказов и улучшать свой сервис.

Таким образом, создание веб-приложения для онлайн-магазина тортов с использованием методологии Scrum является актуальным и перспективным проектом, который может принести значительную выгоду и удобство для клиентов и компании в целом.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ:

1. Сазерленд, Д. (2015). "Скрам: Революционный метод управления проектами." Манн, Иванов и Фербер.
2. Швабер, К., & Сазерленд, Д. (2018). "Руководство по Скрам: Официальное пособие по Agile-методологии Scrum." Scrum.org (Русская версия).
3. Рубин, К. С. (2016). "Скрам. Эффективное управление проектами." ООО "Издательство Символ-Плюс."
4. Воитель, Д. (2017). "Scrum и Agile-методологии: Просто о сложном." Эксмо.
5. Фриденфельд, А. (2019). "Мастер Скрам: Как стать успешным менеджером по Agile-проектам." Манн, Иванов и Фербер.
6. Голосов, В. (2018). "Скрам. Самоучитель." Эксмо.
7. Паттерсон, Д., & Гранич, К. (2017). "User Stories: Гибкое проектирование продуктов с Agile-методологиями." Манн, Иванов и Фербер.
8. Мастеров, В. (2017). "Scrum: Настольная книга." Эксмо.
9. Черный, А. (2020). "Tilda Publishing: Разработка сайтов без программирования." Эксмо.
10. Зайцева, М. (2020). "Путеводитель по Tilda: Создай свой сайт без знания кода." Издательство Символ-Плюс.
11. Белоусова, Е. (2019). "NoCode-платформы: Создание веб-приложений без программирования." Эксмо.
12. Глушко, В. (2018). "Создание сайтов на Tilda: Практическое пособие." Манн, Иванов и Фербер.
13. Колесников, А. (2016). "Web-дизайн на платформе Tilda: От идеи до реализации." Эксмо.
14. Рыжков, С. (2021). "Веб-разработка без кода: Путь дизайнера в NoCode-среде." Манн, Иванов и Фербер.
15. Чумакова, О. (2020). "Введение в Scrum: Основы Agile-методологии и ее применение." Эксмо.