

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 ОБЩАЯ ЧАСТЬ.....	6
1.1 Цель разработки.....	6
1.2 Анализ технологий и возможных средств решения проблемы	7
1.3 Выбор средств и технологий	8
2 СПЕЦИАЛЬНАЯ ЧАСТЬ	13
2.1 Моделирование диаграммы потоков данных	13
2.2 Проектирование моделей в Contentful CMS	14
2.3 Проектирование веб-сервиса.....	15
2.4 Тестирование.....	16
3 ЭКОНОМИЧЕСКАЯ ЧАСТЬ.....	20
3.1 Расчет себестоимости.....	20
3.2 Определение трудоемкости	21
3.2.1 Расчет материальных затрат	22
3.2.2 Расчет затрат на электроэнергию и амортизацию оборудования	23
3.2.3 Расчет расходов на заработную плату	23
3.2.4 Расчет себестоимости и цены разработки	24
3.2.5 Расчет экономического эффекта.....	24
4 ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ.....	26

					09.02.03.ИТ.ДП.00.000.ПЗ			
Изм	Лист	№ докум.	Подп.	Дата				
Разраб.		Притыкин			Проектирование и разработка веб-сервиса приема и оплаты заказов на примере сервисного центра «Папа-Принтер»	Лит.	Лист	Листов
Пров.		Лебедева				У	2	69
Конс.		Тарасова				Колледж НТИ НИЯУ МИФИ ЦМК ИТ КПР-47Д		
Н.контр								
Утв.		Горницкая						

4.1 Разработка мер защиты информации от несанкционированного доступа	26
4.2 Мероприятия по обеспечению безопасности жизнедеятельности	26
ЗАКЛЮЧЕНИЕ	29
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	30
ПРИЛОЖЕНИЕ А.....	32
ПРИЛОЖЕНИЕ В.....	36
ПРИЛОЖЕНИЕ С	41
ПРИЛОЖЕНИЕ D.....	44

ВВЕДЕНИЕ

Последнее время, в связи с развитием цифровой экономики и технологий, любому предприятию без представления себя в интернете, не обойтись. В связи с этим, разработка веб-сервисов для предприятий актуальна и востребована.

Основой разработки настоящего веб-сервиса является проблема отсутствия модуля создания и оплаты заказа на предприятии «Фото-Новик», являющегося партнером «Папа-Принтер». Проблема в виде реальных неудобств и жалоб клиентов, которые приходили на предприятие и хотели сразу получить желаемый товар. Однако, прежде чем его получить, необходимо было сначала создать макет для печати, поэтому получить заказ можно было только в следующий раз, в назначенную дату. Безусловно создание макета и реализация товара отнимала много времени, а клиентопоток только увеличивался, что в результате сказывалось на качестве продукции и качестве обслуживания.

Благодаря данной разработке предприятие сможет повысить качество производимой продукции за счет большего внимания к ней, что позволит достичь высоких технико-экономических показателей, экономить ресурсы, и более полно удовлетворить общественные потребности [1].

Таким образом, после внедрения результатов выпускной квалификационной работы на предприятие, клиенты смогут создавать макет для печати в удобном для них месте, оплачивать, после чего получить товар от доставщика в назначенную администратором предприятия дату или забрать его лично.

Ранее для увеличения конверсии на предприятиях «Папа-Принтер» и «Фото-Новик» был создан веб-ресурс для привлечения клиентов, а в настоящий момент в него будет внедрен веб-сервис для создания и оплаты заказов.

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		4

Безоговорочно то, что данный веб-сервис увеличит количество клиентов, и будет мотивировать заказывать куда больше, чем при личной встрече на предприятии, ведь когда человек в комфортной обстановке, он волен придумывать потрясающие глаз и воображение шедевры.

Подводя итоги, можно сказать, что данный вид веб-сервисов является актуальной разработкой, на которую очень высок спрос у заказчиков.

Для создания ресурса будут использованы гибкие технологии разработки, современные инструментальные средства и методики расчета, что позволит обеспечить универсальность, надежность и высокое качество веб-сервиса. Программный продукт будет удобен в сопровождении и обновлении, а также легок в управлении и настройке.

Веб-сервис приема и оплаты заказов будет являться удобным средством работы для клиентов на предприятии, что значительно улучшит его работу.

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		5

1 ОБЩАЯ ЧАСТЬ

1.1 Цель разработки

Сформулируем цель разработки: «Распределить нагрузку на предприятии за счет веб-сервиса создания и оплаты заказов онлайн без участия сотрудников».

Предприятие «Фото-Новик» является партнером «Папа-Принтер» и занимается печатью на различных предметах, например, кружка или футболка.

Для выявления недостатков или преимуществ, произведем сравнение с веб-сервисами, которые находятся на первой позиции в поисковых системах. Одним из таких веб-сервисов является «vsemaykiru», предоставляющий печать практически на всём, что только можно себе представить: возможность создавать свой собственный дизайн, создавать заказ онлайн и ждать доставки.

Наверняка, развитию сервиса «vsemaykiru» помогли следующие факторы: рост качества выпускаемой продукции, удобство создания заказа клиента и быстрая доставка – это то, чего не достает предприятию «Фото-Новик».

Таким образом, в проекте должны быть реализованы функции оформления заказа, которые заключаются в создании макета для печати и заполнении контактных данных, а также функции оплаты и отправки чека на почту. Подробнее список функций отображен в списке ниже:

- Добавление текста на макет;
- Добавление изображений на макет;
- Перемещение текста и изображений;
- Изменение размера текста и изображений;
- Изменение цвета текста;

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		6

- Ввод email, телефона и имени для оформления заказа;
- Оплата используя карту или qiwi кошелек;
- Отправка письма с заказом на почту администратору;
- Отправка письма с чеком на почту клиенту.

Описанные выше цели были выявлены в процессе разработки, поскольку она велась с использованием гибких подходов.

Выбор методологии разработки основывался на размере предприятия на том, что классические методологии подходят для более крупных проектов и не являются оптимальным решением поставленных задач. Выбор методологии будет подробнее описан в пункте 1.2.

Поскольку предприятие «Фото-Новик» не большое, то принято решение разрабатывать проект основываясь на гибких методологиях разработки, таких как Scrum [3] и Kanban [4], суть которых можно описать цитатой Клода Адриана Гельвеция: «Знание некоторых принципов легко возмещает незнание некоторых фактов» [5].

Благодаря использованию данных методологий разработки, удастся достичь необходимой гибкости в разработке программного продукта. Таким образом, проект будет готов практически к любым желаемым изменениям, и что более важно, намного быстрее получится функциональный продукт.

1.2 Анализ технологий и возможных средств решения проблемы

Исходя из сформулированной цели разработки, можно сделать вывод, что наиболее подходящей методологией разработки является гибкая методология с использованием двух подходов: Scrum и Kanban.

Поверхностно опишем для чего нужны данные подходы:

- Scrum – для разделения разработки на рабочие части;
- Kanban – для визуализации рабочего процесса.

Таким образом, разработка будет разбита на независимые части.

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		7

Использование иных методологий является заведомо проигрышным решением, поскольку в условиях ограниченного времени и непостоянности в желаниях заказчика, они могут сгубить проект.

Например, при использовании каскадной модели разработки, проект может получиться нерасширяемым, как монолит. Спиральная модель отлично подходит для дорогих и крупных проектов, где нет права на ошибку, так как на каждом этапе оцениваются риски [6].

Гибкая методология включает в себе преимущества всех ранее существующих и позволяет использовать различные подходы одновременно, даже при изменении различных этапов, благодаря чему, может подойти практически под любой небольшой или средний проект.

1.3 Выбор средств и технологий

Разрабатываемый веб-сервис будет внедрен в уже существующий веб-ресурс, который написан с использованием следующих технологий:

- Gatsby;
- GraphQL;
- ContentfulCMS;
- Netlify.

Веб-ресурс разрабатывался, с использованием подхода JAMstack [7]. Если описывать коротко, то данный подход подразумевает разработку веб-приложения на желаемом фреймворке и управление контентом из CMS. В результате получатся HTML страницы, которые были сгенерированы с использованием рекомендаций SEO, PWA и Accessibility (A11y).

Александр Товмач в своем выступлении на JS Fest 2019 года предоставил сравнение трех подходов в веб-разработке, которые будут кратко представлены ниже [16].

Ранее использовался PHP сервер-рендеринг позволяющий извлекать данные из базы данных и, используя конструкции языка, вставлять их HTML шаблоны, которые в последствии предоставлялись пользователю в браузере. Схематично, это отображено на рисунке 1.

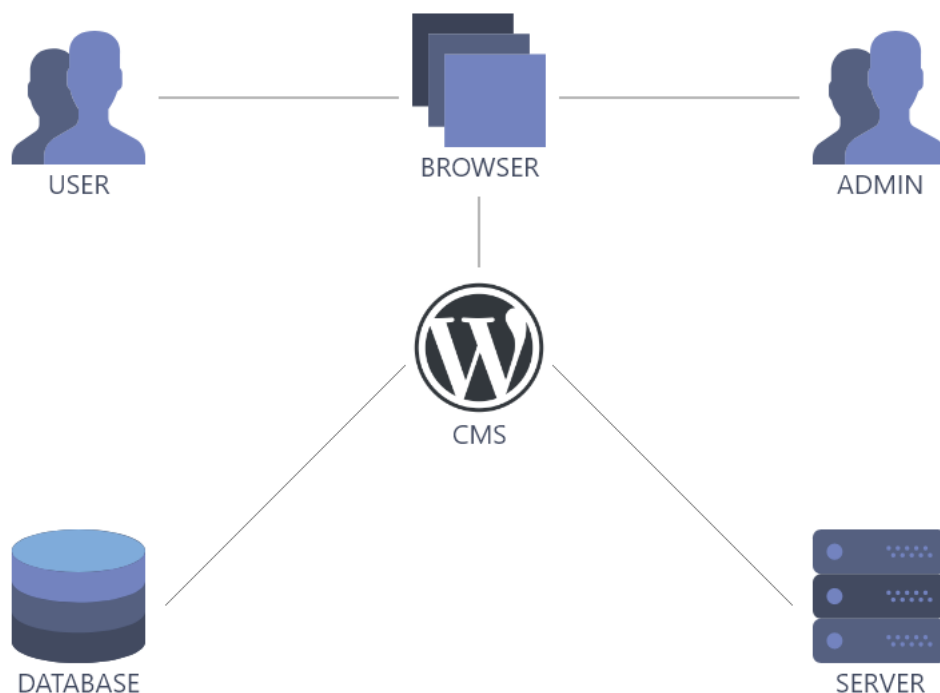


Рисунок 1 – PHP сервер-рендеринг

Впоследствии появилась JavaScript библиотека React, а также фреймворки Vue и Angular. Они предоставляют возможность разрабатывать быстрые динамичные интерфейсы, логика которых исполнялась на клиенте, а доступ к серверу осуществлялся через различные API. Подход отображен в виде схемы на рисунке 2.

На рисунке 3 представлен JAM подход, который предполагает отказ от рендеринга на стороне сервера и клиента, компиляцию всего сайта в готовые HTML страницы единой раз при изменении контента, а также размещение на абсолютно любом файловом хостинге. Таким образом, при грамотной разработке, производительность измеряется только в скорости интернет-соединения пользователя.

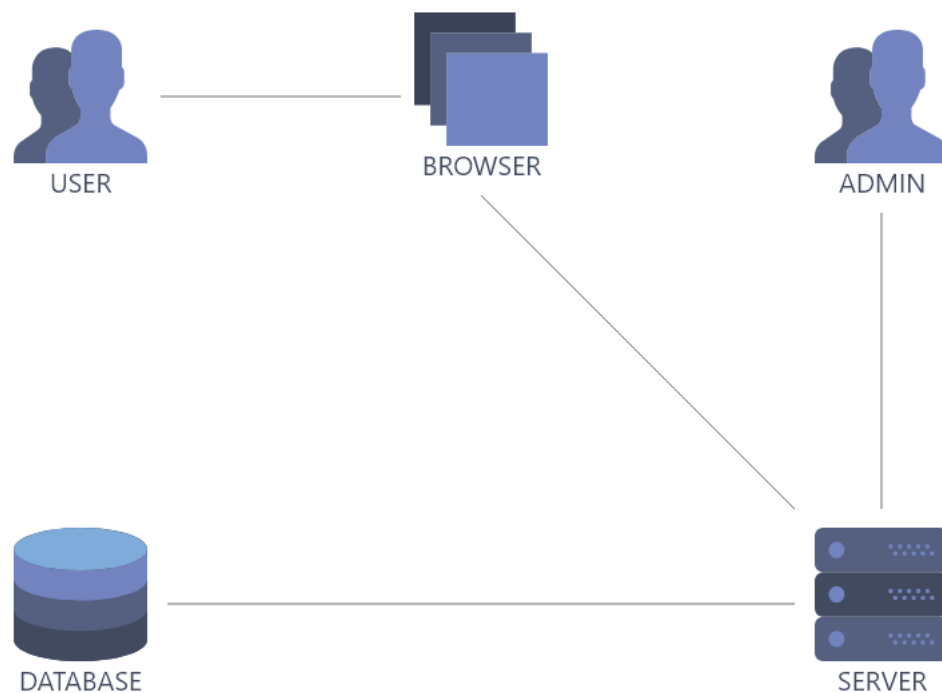


Рисунок 2 – Подход с использованием React, Vue или Angular

Существует множество различных генераторов статических сайтов, один из которых используется в веб-ресурсе – это Gatsby [9]. Он совмещает использование React и GraphQL. Вероятно, выбор данной технологии был сделан, основываясь на статистике.

Contentful – это система управления контентом на статических страницах, которая отлично работает совместно с Gatsby и Netlify. Вместо создания связей между шаблоном и базой данных, она меняет данные в самих HTML файлах [10].

Netlify – это система, позволяющая создавать интуитивно понятный рабочий процесс на основе Git, используя мощную бессерверную платформу для создания, развертывания и совместной работы над веб-приложениями [11].

Исходя из используемых технологий во внедряемом веб-ресурсе, подберем наиболее подходящие библиотеки для реализации поставленных задач.

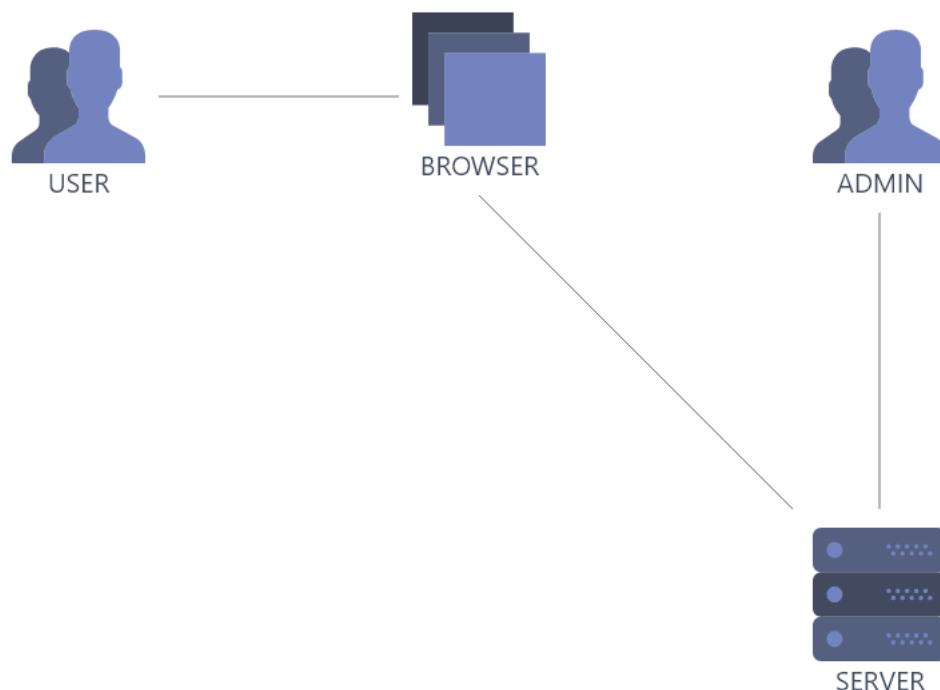


Рисунок 3 – JAM

Исходя из поставленных в пункте 1.1 задач, можно сделать вывод, что необходимо разработать три независимо работающих сервиса: создание макета, рассылка писем на почту, оплата.

Для реализации первого сервиса, будет использоваться библиотека Konva.js [12]. Данная библиотека позволяет использовать canvas.js через декларативные компоненты библиотеки React. Использование чистого canvas.js нецелесообразно при наличии данного решения.

В качестве библиотек для отправки писем будет использоваться бесплатный Gmail API через библиотеку googleapis и nodemailer [13] для отправки электронных писем через SMTP протокол.

Так как необходимо реализовать сервис оплаты, то будет использоваться QIWI кошелек [14]. Кроме выбранного сервиса существует YooMoney, ранее известный, как Яндекс Деньги, разработанный компанией Яндекс, славящейся своей любовью к монетизации всего, к чему может прикоснуться. Так, например, за перевод на карту YooMoney, необходимо будет заплатить

комиссию в 30 рублей, даже за перевод одного рубля. QIWI кошелек предлагает свои услуги бесплатно, например, комиссия за перевод на QIWI кошелек с карты Сбербанка составит 0%.

Сервис создания макета будет внедрен непосредственно в веб-ресурс, а оставшиеся сервисы необходимо будет разместить на отдельном сервере, который будет выступать в роли API. Сервер можно разместить на бесплатном хостинге Heroku.

В результате использования выбранных технологий, возможно будет достичь необходимой скорости в разработке поставленных задач.

Для непосредственной разработки будут использоваться следующие, зарекомендовавшие себя временем, средства:

- Редактор кода VS Code;
- Браузер Google Chrome.

Редактор кода VS Code поддерживает разработку на множестве языков, в том числе и полную поддержку TypeScript, поскольку разработан той же компанией, что и язык.

Браузер Google Chrome является монополистом в мире браузеров. Он поддерживает все современные технологии веб-разработки и привносит новые.

Использование современных средств разработки позволяет разрабатывать и сопровождать качественные программные продукты.

Кроме вышеперечисленных средств, также необходимо система контроля версий Git. Для ее использования достаточно будет сервиса GitHub. Поскольку веб-ресурс, в который будет внедряться разработка, уже находится на репозитории GitHub, то разработка будет вестись в нем.

2 СПЕЦИАЛЬНАЯ ЧАСТЬ

2.1 Моделирование диаграммы потоков данных

Перед непосредственной разработкой программных продуктов принято создавать диаграммы потоков данных. Для реализации поставленных задач необходимо создать такую диаграмму потоков данных, которая смогла бы в достаточной степени описать и структурировать данные, с которыми взаимодействуют сущности в системе. На рисунке 4 отображена контекстная диаграмма потоков данных. На представленном рисунке видно, что есть две сущности: клиент и администратор, один процесс: уточнение требований, данные: требования, оплата и заказ-макет, а также хранилище данных: заказ-макет.



Рисунок 4 – Контекстная диаграмма потоков данных

При детализации процесса, имеющего проблему, можно представить её в виде, отображенном на рисунке 5. После детализации становится видно, что администратор занимается двумя делами вместо одного, в результате чего падает качество работы и увеличивается нагрузка на данную сущность.

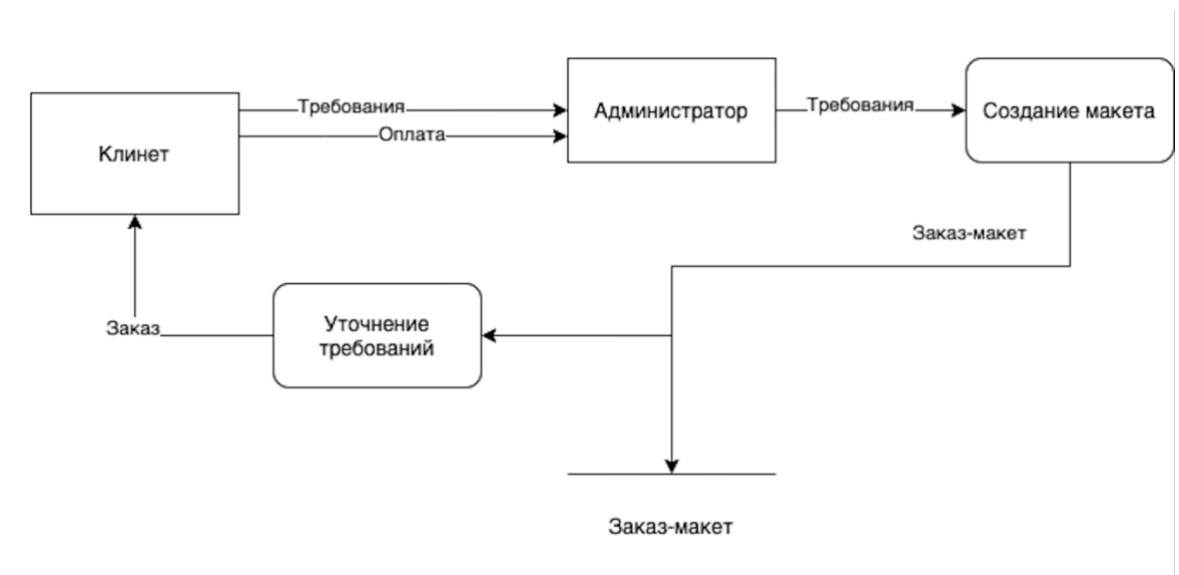


Рисунок 5 – Детализированная диаграмма потоков данных

Таким образом, после детализации стало видно, что предприятие имеет утечку производительности в области создания заказа. Данные действия можно возложить на клиента, предоставив ему удобные инструменты, благодаря которым, тот сможет сам создавать свои собственные макеты.

После внедрения разработки, администратору нужно будет лишь отвечать на письма заказов и передавать макеты на печать, тем самым он станет посредником между производителем и клиентом.

2.2 Проектирование моделей в Contentful CMS

В ContentfulCMS создаются модели контента. Для данной разработки спроектируем модель, необходимую для создания макета на стороне клиента.

Ниже представлен список с необходимыми полями:

- 1) Заголовок;
- 2) Задний фон;
- 3) Шаблон;
- 4) Точка начала области печати по X;
- 5) Точка начала области печати по Y;

- 6) Высота области печати;
- 7) Ширина области печати;
- 8) Размер товара;
- 9) Цена товара.

Данную модель необходимо интегрировать в уже имеющуюся модель посредством связи один ко многим.

Таким образом после создания контента в Contentful CMS можно будет получить данные в корректном виде для отображения редактора макета и формы создания заказа в целом.

2.3 Проектирование веб-сервиса

Можно смоделировать диаграмму последовательностей, которая в достаточной степени отобразит последовательность процессов, необходимых для достижения поставленных задач. На рисунке 6 представлена диаграмма последовательностей веб-сервиса.

На рисунке 6 можно наблюдать 6 сущностей:

- Клиент;
- Сервер;
- БД (база данных);
- CMS;
- QIWI;
- EMAIL.

Между представленными сущностями установлены взаимодействия, в результате которых пользователь может создать и оплатить заказ.

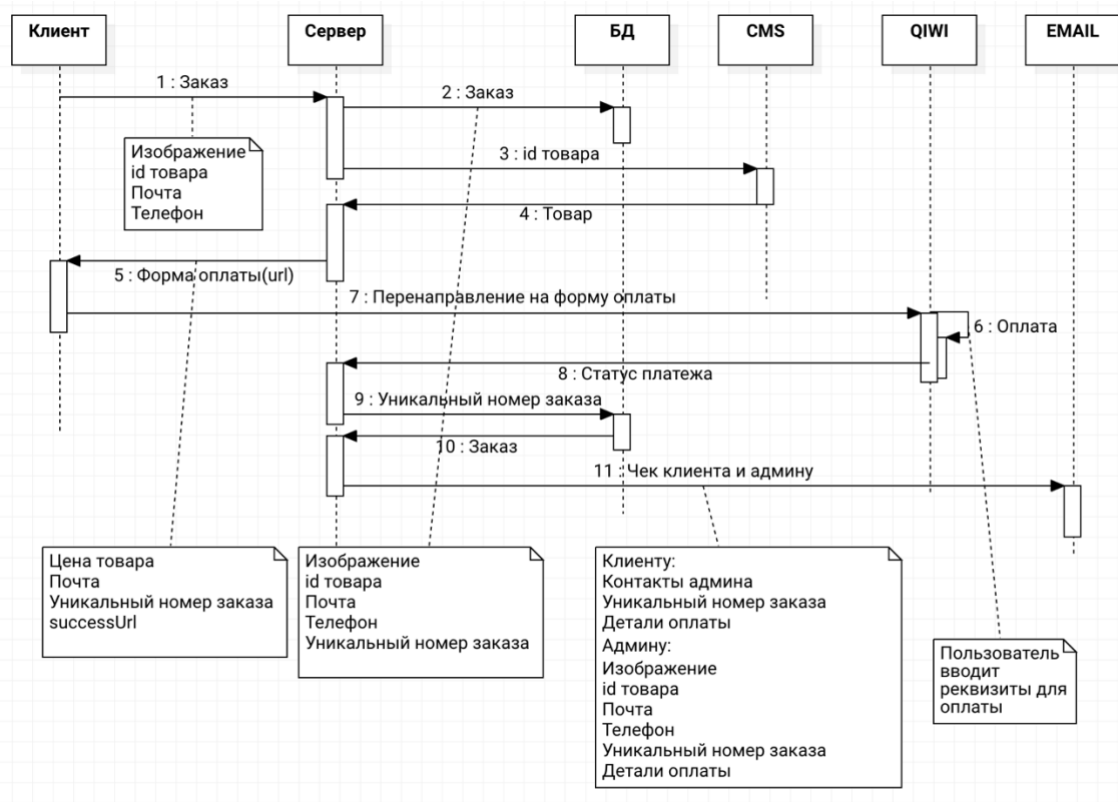


Рисунок 6 – диаграмма последовательностей веб-сервиса

2.4 Тестирование

При разработке веб-сервиса не выявилось ни одной ошибки, в результате чего, необходимость в отладке пропадает. Отсутствие ошибок вызвано использованием языка TypeScript, который в отличие от JavaScript является языком со статической типизацией, которая проверяется в момент компиляции. Таким образом, язык TypeScript не только позволяет избежать некоторых ошибок в runtime, но и ошибок в compile time, посредством вывода подсказок при написании кода, которые вызваны установкой типов для каждой функции, объекта или константы.

Однако ошибки могут возникнуть, поскольку TypeScript не является языком, который следит за типизацией в runtime, следовательно вследствие обращения к API могут возникнуть ошибки.

Ошибки могут отобразиться во время разработки в виде сообщения или некорректной работы. Зачастую сообщения об ошибках отображают место и причину, а иногда, вследствие особенностей языка JavaScript, они могут не отображаться, то есть быть неявными, которые можно исправить лишь используя Debugger.

Во избежание возможных ошибок, проводят различные тесты, например, unit-тестирование, тестирование методом белого или черного ящика.

Для тестирования разрабатываемого веб-сервиса выбран метод тестирования черного и белого ящика, поскольку время на написание Unit-тестов отсутствует. При тестировании достигается необходимая гарантия того, что веб-сервис будет работать корректно при различных условиях.

Метод черного ящика заключается в тестировании, не предполагающим знания внутреннего устройства системы. Благодаря данному тестированию можно не только сказать, что система работает верно, но и то, что пользователь сможет с легкостью разобраться в ней.

Тестирование методом черного ящика проводилось между закрытым кругом лиц, являющимися добровольцами в прохождении тестирования. В таблице 1 отображены результаты тестирования методом черного ящика.

Таблица 1 – Результаты теста методом черного ящика

Описание теста	Ожидаемый результат	Результат
Открытие редактора макета	Открытый редактор макета	Пройдено
Выбор макета через селектор	Выбранный макет	Пройдено
Добавление текста на макет	Отображение текста на макете	Пройдено
Добавление нескольких изображений на макет	Отображение изображений на макете	Пройдено
Перемещение текста	Перемещенный текст	Пройдено
Перемещение изображений	Перемещенное изображение	Пройдено
Изменение цвета текста	Измененный цвет текста	Пройдено

Описание теста	Ожидаемый результат	Результат
Изменение размера текста	Измененный размер текста	Пройдено
Изменение размера изображений	Измененный размер изображений	Пройдено
Ввод email, телефона и имени в форму	Введенные email, телефон и имя	Пройдено
Оплата используя карту или qiwі кошелек	Оплата используя карту или qiwі кошелек	Пройдено
Получение письма на почту администратором с изображениями, товаром, email, телефоном, именем	Полученное письмо на почте администратора	Пройдено
Получение письма с чека на почту клиента	Получение письма с чеком на почте клиента	Пройдено

Тестирование методом белого ящика будет использоваться для сравнения требований, описанных в техническом задании, и реализованного функционала. В таблице 2 отображены результаты тестирования методом белого ящика.

Таблица 2 – Результаты теста методом белого ящика

Описание теста	Ожидаемый результат	Результат
Открытие редактора макета	Открытый редактор макета	Пройдено
Выбор макета через селектор	Выбранный макет	Пройдено
Добавление текста на макет	Отображение текста на макете	Пройдено
Добавление нескольких изображений на макет	Отображение изображений на макете	Пройдено
Перемещение текста	Перемещенный текст	Пройдено
Перемещение изображений	Перемещенное изображение	Пройдено
Изменение цвета текста	Измененный цвет текста	Пройдено

Описание теста	Ожидаемый результат	Результат
Изменение размера текста	Измененный размер текста	Пройдено
Изменение размера изображений	Измененный размер изображений	Пройдено
Ввод email, телефона и имени в форму	Введенные email, телефон и имя	Пройдено
Оплата используя карту или qiwі кошелек	Оплата используя карту или qiwі кошелек	Пройдено
Получение письма на почту администратором с изображениями, товаром, email, телефоном, именем	Полученное письмо на почте администратора	Пройдено
Получение письма с чека на почту клиента	Получение письма с чеком на почте клиента	Пройдено

По результатам тестов можно сказать, что веб-сервис работает согласно требованиям и готов к внедрению.

3 ЭКОНОМИЧЕСКАЯ ЧАСТЬ

3.1 Расчет себестоимости

Существуют множество техник и подходов в оценке проектов. Основных моделей оценки два, которые тесно связаны с основными подходами к управлению. Так, например, в статье Виталия Салахмира, являющимся опытным руководителем проектов, про оценку стоимости разработки проекта в достаточной полной форме описаны данные модели [17].

Ниже представлены две основные модели оценки проекта:

- 1) Fix-price;
- 2) Time and material.

По модели оценки Fix-price производится оценка сроков и стоимости всего проекта целиком. Зачастую данная модель используется тогда, когда существуют строго определенное техническое задание.

При использовании модели Time and material оценка сроков и стоимости происходит во время разработки.

Согласно модели Time and material, оценка происходит путем выяснения того, сколько нужно времени на разработку и умножением его на стоимость часа работы специалиста.

Оценка стоимости проекта по модели Fix-price происходит один раз, таким образом заказчик экономит время на операциях, связанных с оценкой каждой задачи, а также заплатить больше или заплатить меньше, чем стоит проект. Однако в некоторых случаях оценка по данной модели может пересматриваться, так как угадать реальную цену зачастую не удастся, и тогда необходимо повторное оценивание, но обычно, обсуждается лишь добавочное время, нежели доплата.

Модель Time and material предполагает оплату за затраченное время, например, при гибкой разработке получение оплаты происходит после выполнения каждой задачи или же перед выполнением задачи производится

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
						20
Изм	Лист	№ Докум.	Подп.	Дата		

оценка и оплата. Данная модель позволяет оценивать стоимость конкретных задач. Оценка стоимости каждой задачи точно также является задачей, имеющая свою стоимость. При данном подходе производятся исследования более явно, чем при Fix-price.

Разработка настоящего веб-сервиса является бесплатной, однако оценивается по модели Time and material, поскольку она в достаточной степени позволит оценить выполненную работу.

Согласно выбранной модели оценки, она может производиться менеджером проекта, тим-лидом, руководителем отдела или непосредственным исполнителем. Таким образом, оценка производится непосредственным исполнителем, ввиду отсутствия вышеперечисленных.

При оценке непосредственным исполнителем достигается максимальная точность, потому что он разбирается в нюансах задачи, а также именно он будет её реализовывать и нести ответственность за выполнение.

Поскольку разработчик настоящего веб-сервиса является достаточно опытным, чтобы считать его Middle-разработчиком, поэтому оценка будет вестись с учётом профессионального уровня специалиста.

3.2 Определение трудоемкости

Выделим используемых для реализации проекта сотрудников:

- Программист;
- Дизайнер;
- Верстальщик;
- Тестировщик;
- Менеджер.

В таблице 3 отображены потраченные часы и стоимость часа каждого сотрудника.

Таблица 3 – Стоимость проекта

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		21

Сотрудник	Часы	Стоимость часа	Стоимость
Программист	38	800	30400
Дизайнер	8	100	800
Верстальщик	8	300	2400
Тестировщик	2	100	200
Менеджер	24	500	12000

Время, затраченное на разработку, равняется 10 дням по 8 часов работы в день.

Поскольку работа велась в бесплатном выделенном помещении, используя бесплатную электроэнергию, то стоимость часа исходит исключительно из чистой работы без надбавок, однако в стоимость часа включаются затраты на оборудование.

Цена за час взята, как средняя по рынку веб-разработки, учитывая опыт и навыки сотрудника.

3.2.1 Расчет материальных затрат

Материальные затраты исходят из затрат на используемое оборудование, а также иные нужды, какие укажет разработчик в разумных пределах. Например, затраты на питание по средней цене в столовых.

Вышеперечисленные затраты отображены в таблице 4.

Таблица 4 – Расчет материальных затрат

Материальные затраты	Стоимость затрат в день в рублях
MacBook Pro	10
Иные нужды	250

Стоимость затрат на MacBook Pro рассчитывалась как стоимость замены батареи, деленная на гарантийный срок службы. Иные нужды указываются как средние затраты в день. Таким образом стоимость материальных затрат за всё время разработки будет 2150 рублей.

3.2.2 Расчет затрат на электроэнергию и амортизацию оборудования

Поскольку электричество во время разработки являлось бесплатным, то затраты на него не учитываются при вычислении финальной цены разработки. Однако при платной электроэнергии, расчет производился бы путем умножения стоимости электричества в час на количество выработанных часов программистом, дизайнером, верстальщиком, тестировщиком и менеджером.

Расчет амортизации оборудования исходит из окупаемости Macbook Pro, которая рассчитывается как стоимость покупки, деленная на гарантийный срок службы, что в итоге равно 50 рублям в день. Поскольку разработка велась 10 дней, то за время всей разработки, амортизация оборудования составила 500 рублей.

3.2.3 Расчет расходов на заработную плату

Подведем итоги по заработной плате согласно таблице 3, в которой указана стоимость работы сотрудника в час. В таблице 4 приведена заработная плата в рублях для каждого сотрудника, принимающего участие в проекте.

Таблица 4 – Заработная плата

Сотрудник	Заработная плата в рублях
Программист	30400
Дизайнер	800
Верстальщик	2400
Тестировщик	200
Менеджер	12000

Исходя из данных в таблице 4, общая сумма всей заработной платы сотрудников равна 45800 рублям.

3.2.4 Расчет себестоимости и цены разработки

Согласно проведенным вычислениям, просуммируем все затраты на разработку. Будем использовать формулу:

$$CP = (ЗП + МЗ + АО) \times Н + ЗП + МЗ + АО, \quad (1)$$

где CP – стоимость разработки в рублях;

ЗП – заработная плата;

МЗ – материальные затраты;

АО – амортизация оборудования;

Н – наценка.

Подставив в формулу значения, рассчитанные выше, учитывая наценку в 10% получим:

$$(45800 + 2100 + 500) \times 0,1 + 45800 + 2100 + 500 = 53240 \quad (2)$$

По итогам расчета, финальная цена разработки с учетом всех ее составляющих расходов составила 53240 рублей.

3.2.5 Расчет экономического эффекта

Из произведенных исследований, на данный момент предприятие обслуживает в среднем 20 клиентов в день, из них только 5 оставляют заказ на одну единицу товара.

Благодаря внедрению разработки, предприятие увеличит за месяц клиентопоток с 20 человек в день до 22. Увеличит количество клиентов, оставивших один заказ с 5 до 6, а также появятся 1 клиент, который сделает в среднем два заказа за раз.

Исходя из предсказаний, число клиентов увеличилось на 10%, а количество заказов на 60%.

Спустя 6 месяцев, при должной рекламной работе, предприятие сможет увеличить клиентопоток с 20 клиентов в день до 32, а количество заказов с 5 до 23.

Таким образом за 6 месяцев предприятие увеличит клиентопоток на 60%, а количество заказов на 360%.

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		25

4 ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ

4.1 Разработка мер защиты информации от несанкционированного доступа

В разрабатываемом веб-сервисе присутствует сервис оплаты, который обязан поддерживать передачу данных по протоколу HTTPS. Оплата происходит максимально безопасным методом, через заведомо надежный сервис онлайн платежей QIWI Кошелек.

При заполнении заявки пользователем на клиенте, данные отправляются на сервер, который отправляет пользователя на страницу оплаты, созданную QIWI Кошельком. После успешной или безуспешной оплаты, QIWI Кошелек отправляет чек об оплате на секретный эндпоинт сервера.

Таким образом, можно с уверенностью заявить, что оплата на разрабатываемом веб-сервисе безопасна. Также стоит отметить, что данные пользователя, указанные им при составлении заявки, хранятся на сервере лишь до окончания оплаты, тем самым уменьшается риск утечки данных.

4.2 Мероприятия по обеспечению безопасности жизнедеятельности

Во время разработки необходимо не только задумываться о безопасности реализуемого проекта, будь то архитектурное решение или написанный код, также, что более важно, необходимо уделять внимание безопасности жизнедеятельности при работе за компьютером.

В общем доступе через глобальную сеть интернет представлено множество техник, обеспечивающих безопасность жизнедеятельности.

Согласно множеству источников, ниже представлены шесть основных опасных и вредных факторов:

- 1) Повышенный уровень электромагнитных излучений;
- 2) Повышенный уровень статического электричества;

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		26

- 3) Пониженная ионизация воздуха;
- 4) Физические перегрузки в статическом положении;
- 5) Перенапряжение глаз;
- 6) Недостаточная освещенность.

Исходя из представленных вредных и опасных факторов, необходимо организовать мероприятия по уменьшению воздействия данных факторов.

Например, в начале работы необходимо производить настройку и регулировку рабочего места для уменьшения нагрузки в статическом положении.

Во время работы важно делать физические разминки глаз и тела, а также проветривать помещения, и иметь ионизаторы воздуха.

Недостаточную освещенность можно предотвратить наличием прямого доступа солнечных лучей на рабочее место, однако излишняя освещенность может навредить. При работе в темное время суток, устанавливаются специальные лампы, создающее фоновое освещение с мощностью от 40 до 60 Вт, например, светодиодные и галогенные лампы. Желательно располагать лампы под монитором на высоте около 35 см [19].

По окончанию работы обязательным мероприятием должно быть приведение рабочего места в изначальное состояние, выключаются рабочие устройства и используемые предметы убираются на место. Таким образом, уменьшаются излучения в момент бездействия.

Разработка велась с учетом вышеперечисленных правил. При работе за персональным компьютером соблюдались правила расположения относительно стола и монитора. Во время работы и перерыва был включен озонатор воздуха, проветривались помещения, а также, в темное время суток, использовалось фоновое дневное освещение. В перерывах между работой проводилась тренировка, представленная на YouTube канале Dev Ed, имеющем 647 тысяч подписчиков [20].

Исходя из проводимых мероприятий по обеспечению безопасности жизнедеятельности при работе за компьютером, можно сказать, что здоровье разработчика сохраняется успешно.

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		28

ЗАКЛЮЧЕНИЕ

Согласно пункту 1.1, цель сформулирована следующая: «Распределить нагрузку на предприятии за счет веб-сервиса создания и оплаты заказов онлайн без участия сотрудников». По результатам выполненных работ, можно утверждать, что проект готов к внедрению на предприятие, поставленная цель достигнута.

Внедренный веб-сервис позволит предприятию увеличить качество производимой продукции за счет составления заказа онлайн, а именно, создание макета и заполнение контактных данных, а также его последующей оплаты, что значительно снизит нагрузку на администратора. Увеличение качества продукции и обслуживания увеличивает клиентопоток, что в свою очередь, увеличивает прибыль компании. Данное суждение подтверждается пунктом 3.2.5.

Предоставление клиенту возможности составлять заказы удаленно, повысит уровень доверия к предприятию, что увеличит процент повторных заказов. Реализовывая свой творческий потенциал, клиенты смогут создавать макеты в удобном месте, совместно со своими знакомыми или родными.

За счет использования Contentful CMS, администратор может добавлять столько макетов, сколько необходимо, что является огромным преимуществом, за счет своей цены и скорости добавления.

После внедрения возможно дальнейшее развитие разработки, поскольку она велась с использованием гибких методологий, что позволило сохранить гибкую архитектуру. Возможно добавление нового функционала в создание макета, а также добавление различных способов оплаты, например, Apple Pay и Google Pay.

Разработанный веб-сервис расположен по адресу <https://novoservice.netlify.app/photo>.

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		29

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Условия повышения качества продукции [Электронный ресурс] – Режим доступа: <http://www.stroitelstvo-new.ru/sudostroenie/brigadir/povyshenie-kachestva-produktsii.shtml> (дата обращения 26.05.21).

2 Сервис печати [Электронный ресурс] – Режим доступа: <https://www.vsemayki.ru/> (дата обращения 27.05.21).

3 Agile, scrum, kanban: в чем разница и для чего использовать? [Электронный ресурс] — Режим доступа: <https://rb.ru/story/agile-scrum-kanban> (Дата обращения: 08.05.21) .

4 Что такое Agile? Канбан – это Agile? [Электронный ресурс] — Режим доступа: <https://rb.ru/story/kanban-agile/> (Дата обращения: 27.05.21) .

5 Цитаты известных личностей [Электронный ресурс] — Режим доступа: <https://ru.citaty.net/tsitaty/472306-klod-adrian-gelvetsii-znanie-nekotorykh-printsipov-legko-vozmeshchaet-neznani/> (Дата обращения: 27.05.21).

6 Методологии разработки [Электронный ресурс] — Режим доступа: <https://issoft.by/blog/podkhody-k-razrabotke-po-kak-pravilno> (Дата обращения: 30.05.21).

7 Что такое Jamstack [Электронный ресурс] — Режим доступа: <https://jamstack.org/what-is-jamstack/> (Дата обращения: 30.05.21).

8 Jamstack [Электронный ресурс] — Режим доступа: <https://habr.com/ru/post/439232/> (Дата обращения: 30.05.21).

9 Gatsby [Электронный ресурс] — Режим доступа: <https://www.gatsbyjs.com/> (Дата обращения: 30.05.21).

10 Contentful [Электронный ресурс] — Режим доступа: <https://www.contentful.com/> (Дата обращения: 30.05.21).

11 Netlify [Электронный ресурс] — Режим доступа: <https://www.netlify.com/> (Дата обращения: 30.05.21).

12 Konvajs [Электронный ресурс] — Режим доступа: <https://konvajs.org/> (Дата обращения: 30.05.21).

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		30

13 Nodemailer [Электронный ресурс] — Режим доступа: <https://nodemailer.com/about/> (Дата обращения: 30.05.21).

14 QIWI [Электронный ресурс] — Режим доступа: <https://qiwi.com/> (Дата обращения: 30.05.21).

15 Heroku [Электронный ресурс] — Режим доступа: <https://heroku.com> (Дата обращения: 30.05.21).

16 JS Fest 2019 [Электронный ресурс] — Режим доступа: <https://www.youtube.com/watch?v=wCNSK4iFCuE> (Дата обращения: 30.05.21)

17 Оценка себестоимости разработки [Электронный ресурс] — Режим доступа: <https://salakhmir.ru/оценка-стоимости-разработки-проекта> (Дата обращения: 30.05.21).

18 Услуги программиста [Электронный ресурс] — Режим доступа: <https://www.kadrof.ru/articles/46641> (Дата обращения: 30.05.21)

19 Освещение за компьютером [Электронный ресурс] — Режим доступа: <https://wd-x.ru/osveshhenie-pri-rabote-za-kompyuterom/> (Дата обращения: 30.05.21).

20 Тренировка для разработчика [Электронный ресурс] — Режим доступа: <https://www.youtube.com/watch?v=qL1YjIRDQEI> (Дата обращения: 30.05.21).

21 ГОСТ 19.505-79 Единая система программной документации (ЕСПД). Руководство оператора. Требования к содержанию и оформлению [Электронный ресурс] — Режим доступа: <https://docs.cntd.ru/document/1200007676> (Дата обращения: 30.05.21).

ПРИЛОЖЕНИЕ А

(обязательное)

Техническое задание

1 Введение

Наименование веб-ресурса: «Novoservice».

Наименование веб-сервиса: «Веб-сервис приема и оплаты заказов».

Основная цель разрабатываемого модуля – предоставить пользователю возможность создать заказ, оплатить его и получить чек на указанный им при создании заказа email.

Задачи «Веб-сервис приема и оплаты заказов»:

- Предоставление пользователю конструктора макетов для создания макета;
- Предоставления возможности оплаты созданного макета;
- Рассылка чека оплаты исполнителю и заказчику.

2 Основания для разработки

Основанием для разработки программного продукта является задание на дипломную практику от 19 марта 2021 года, согласованное заместителем директора по управляющей работе Балакиной И.А. и председателем ЦМК Горницкой И.И.

Наименование темы разработки: «Проектирование и разработка веб-сервиса приема и оплаты заказов на примере сервисного центра «Папа-Принтер».

3 Назначение разработки

Основное назначение разработки – предоставление пользователю удобного интерфейса получения услуг.

3.1 Функциональное назначение

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		32

Путем добавления модуля уже рабочий веб-сервис предоставит пользователю конструктор макета для создания макета с последующей его онлайн-оплатой.

3.2 Эксплуатационное назначение

Клиенты получают возможность оформлять заказ удаленно и в дальнейшем забирать его в назначенное исполнителем время.

4 Требования к программе

Веб-сервис должен содержать следующий функционал:

- Добавление текста на макет;
- Добавление изображений на макет;
- Перемещение текста и изображений;
- Изменение размера текста и изображений;
- Изменение цвета текста;
- Ввод email, телефона и имени для оформления заказа;
- Оплата используя карту или qiwi кошелек;
- Отправка письма с заказом на почту администратору;
- Отправка письма с чеком на почту клиенту.

4.1 требования к функциональным характеристикам

Веб-сервис состоит из клиентской части, написанной на технологии Gatsby, который получает данные с CMS Contentful. Модуль должен использовать необходимые данные из используемой веб-сервисом CMS Contentful.

4.1.1 Требования к серверной части

Серверная часть должна быть написана на NodeJS.

4.1.2 Требования к взаимодействию клиентской и серверной частей

Взаимодействие между серверной и клиентской частью должно происходить по протоколу HTTPS.

4.1.3 Требования к клиентской части

Клиентская часть должна быть реализована с использованием технологий, уже используемых в веб-сервисе, и при надобности, использовать необходимые библиотеки.

4.2 Требования к надежности

4.2.1. Требования к обеспечению надежного (устойчивого) функционирования программы

Требованием к надежности является обеспечение передачи данных с использованием протокола HTTPS, а также отправки заказчику и исполнителю чека об успешной оплате.

4.2.2. Время восстановления после отказа

Время восстановления после отказа не предусмотрено.

4.2.3. Отказы из-за некорректных действий оператора

Действия оператора не предусматриваются.

4.3 Условия эксплуатации

4.3.1. Климатические условия эксплуатации

Требований к климатическим условиям эксплуатации не предъявляется.

4.3.2. Требования к видам обслуживания

Обслуживание не предусматривается. Предусматривается только разработка.

4.3.3. Требования к численности и квалификации персонала

Управление необходимыми настройками должно производиться через CMS Contentful.

4.4 Требуемая квалификация и уровень подготовки пользователя

Внедряемый в веб-сервис модуль должен быть прост, чтобы в нем можно было легко разобраться в короткое время, людям, умеющим взаимодействовать со смартфоном или персональным компьютером.

4.5 Требования к составу и параметрам технических средств

Внедряемый в клиентскую часть веб-сервиса модуль должен соответствовать составу и параметрам технических средств, используемых в веб-сервисе.

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		34

Серверная часть может располагаться на веб-хостинге heroku.

4.6 Требования к информационной и программной совместимости

Веб сервис доступен через современные веб-браузеры. Внедряемый модуль должен так же быть доступен в них.

4.6.1. Требования к исходным кодам и языкам программирования

Исходные коды модуля внедряемого в клиентскую часть веб-сервиса должны быть написаны на языке Typescript.

Сервер должен быть написан на NodeJS.

4.7 Требования к маркировке и упаковке

Требования к маркировке и упаковке не предусмотрены.

4.8 Требования к транспортированию и хранению

Требования к транспортированию и хранению не предусмотрены.

4.9 Специальные требования

Специальные требования не предусмотрены.

5 Требования к программной документации

Требования к программной документации устанавливаются индивидуальным заданием на производственную (преддипломную) практику, а также заданием на дипломную практику.

6 Техничко-экономические показатели

Внедряемый модуль позволит клиентам быстрее создать заказ, оплатить и получить его в назначенную дату.

7 Стадии и этапы разработки

Стадии и этапы разработки определяются разработчиком самостоятельно.

Исполнитель: Притыкин Михаил Вячеславович.

Заказчик: ИП Фомин Виктор Дмитриевич

8 Порядок контроля и приемки

Контроль и приемка разработки осуществляются после сравнительных тестов реализованного функционала и поставленных задач в настоящем техническом задании.

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		35

ПРИЛОЖЕНИЕ В

(обязательное)

Руководство пользователя (для клиента)

1. Введение

1.1 Область применения

Веб-сервис применяется внутри веб-ресурса «Novoservice» и предназначен для создания макета с последующим составлением заказа.

1.2 Краткое описание возможностей

Ниже приведен список, являющийся кратким описанием возможностей:

- Добавление текста на макет;
- Добавление изображений на макет;
- Перемещение текста и изображений;
- Изменение размера текста и изображений;
- Изменение цвета текста;
- Оформление заказа;
- Оплата используя карту или qiwi кошелек;
- Получение письма с чеком на почту.

1.3 Уровень подготовки пользователя

Для работы с веб-сервисом необходимы базовые навыки работы с компьютером или смартфоном.

1.4 Перечень эксплуатационной документации

Перечень эксплуатационных документов, с которым необходимо ознакомиться:

- Настоящее руководство пользователя (для клиента).

2 Назначение и условия применения

Назначение веб-сервиса заключается в онлайн составлении заказов.

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		36

Условия, при соблюдении которых обеспечивается применение веб-сервиса, являются: наличие современного браузера и наличие базовых знаний в его использовании.

2.1 Виды деятельности, функции

Веб-сервис предназначен для быстрого и удобного создания макета для печати с дальнейшей его оплатой.

Для стабильного функционирования необходим современный браузер.

3.1 Запуск редактора макета

Для запуска редактора макета, последовательно выполните следующие действия:

- После перехода на веб-ресурс, перейдите в раздел Фотоновик, который находится по ссылке <https://novoservice.netlify.app/photo>;
- Опустившись ниже, во вторую секцию с выбором товара, нажмите на ссылку «Или создайте свой макет».

3.2 Проверка работоспособности системы

Если при открытии отображается селектор выбора макета, то это значит, что всё работает корректно.

4 Описание операций

4.1 Операция Выбор шаблона

Наименование: Выбор шаблона

Условия: запущен редактор макета

Основные действия в требуемой последовательности: в представленном селекторе выбрать шаблон. Дождаться загрузки и отображения шаблона.

4.2 Операция Добавление текста в макет

Наименование: Добавление текста в макет

Условия: выбран шаблон

Основные действия в требуемой последовательности: В область ввода текста, введите текст. После ввода текста, нажмите на кнопку с текстом «Добавить текст». После нажатия, текст отобразится на шаблоне.

4.3 Операция Изменение размера текста

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		37

Наименование: Изменение размера текста

Условия: добавлен текст

Основные действия в требуемой последовательности: на панели редактирования желаемого текста в поле ввода введите желаемый размер. При вводе размера, он сразу применяется к тексту.

4.4 Операция Изменение цвета текста

Наименование: Изменение цвета текста

Условия: добавлен текст

Основные действия в требуемой последовательности: на панели редактирования желаемого текста, нажмите на селектор цвета для его открытия. Выберите нужный цвет. Выбранный цвет сразу применится к тексту.

4.5 Операция Изменение области отображения текста

Наименование: Изменение области отображения текста

Условия: добавлен текст

Основные действия в требуемой последовательности: Нажмите на текст, добавленный на шаблон, появятся точки взаимодействия. Если потянуть за них, то можно изменить область отображения текста.

4.6 Операция Изменение положения текста на шаблоне

Наименование: Изменение положения текста на шаблоне

Условия: добавлен текст

Основные действия в требуемой последовательности: Потяните за текст на шаблоне и перетащите в нужное место.

4.7 Операция Удаление текста

Наименование: Удаление текста

Условия: добавлен текст

Основные действия в требуемой последовательности: нажмите на кнопку с текстом «Удалить».

4.8 Операция Добавление изображения в макет

Наименование: Добавление изображения в макет

Условия: выбран шаблон

Основные действия в требуемой последовательности: Нажмите на кнопку с текстом «Выберите файл». Используя встроенный функционал браузера, выберите необходимое изображение. После выбора изображения, оно отобразится на шаблоне.

4.9 Операция Изменение размера изображения

Наименование: Изменение области изображения

Условия: добавлено изображение

Основные действия в требуемой последовательности: Нажмите на изображение, добавленное на шаблон, появятся точки взаимодействия. Если потянуть за них, то можно изменить его размер.

4.10 Операция Изменение положения изображения на шаблоне

Наименование: Изменение положения изображения на шаблоне

Условия: добавлено изображение

Основные действия в требуемой последовательности: Потяните за изображение на шаблоне и перетащите в нужное место.

4.11 Операция Удаление текста

Наименование: Удаление текста

Условия: добавлено изображение

Основные действия в требуемой последовательности: нажмите на кнопку с текстом «Удалить», находящуюся на панели изображения.

4.12 Операция Изменение размера товара

Наименование: Изменение размера товара

Условия: выбран шаблон

Основные действия в требуемой последовательности: в селекторе выбора размер выберите размер.

4.13 Операция Оформление заказа

Наименование: Оформление заказа

Условия: выбран шаблон

Основные действия в требуемой последовательности: нажмите на кнопку с текстом «Купить», находящуюся на панели изображения. Заполните данные указанные в открывшейся форме данные и нажмите на кнопку «Купить». Дождитесь открытия формы оплаты.

4.14 Операция Оплата заказа

Наименование: Оплата заказа

Условия: открыта форма оплаты

Основные действия в требуемой последовательности: следуйте инструкциям, отображенным в форме оплаты.

5 Аварийные ситуации

В случае возникновения аварийных ситуаций, необходимо немного подождать.

6 Рекомендации по освоению

Для освоения не требуется специальных навыков.

Руководство пользователя составлялось на основе ГОСТ 19.505-79 ЕСПД [22].

ПРИЛОЖЕНИЕ С

(обязательное)

Руководство пользователя (для администратора)

1. Введение

Данное руководство позволит администратору с легкостью овладеть необходимыми навыками управления контентом на веб-сервисе.

1.1 Область применения

Управление веб-сервисом происходит через систему управления контентом Contentful CMS.

1.2 Краткое описание возможностей

Ниже приведен список, являющийся кратким описанием возможностей:

- Добавление шаблонов;
- Добавление названия шаблона;
- Установка цены шаблона;
- Изменение области печати;
- Изменение размера изделия.

1.3 Уровень подготовки пользователя

Для управления веб-сервисом необходимо владеть навыками создания, редактирования, публикации, удаления контента в Contentful CMS.

1.4 Перечень эксплуатационной документации

Перечень эксплуатационных документов, с которым необходимо ознакомиться:

- Настоящее руководство пользователя (для администратора).

2 Назначение и условия применения

Назначение Contentful CMS заключается в управлении контентом веб-сервиса.

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		41

Условия, при соблюдении которых обеспечивается применение веб-сервиса, являются: ознакомление и применение инструкций, приведенных в настоящем руководстве пользователя (для администратора).

2.1 Виды деятельности, функции

Contentful CMS предназначена для управления контентом на веб-сервисе.

Для стабильного функционирования необходим современный браузер.

3.1 Вход в Contentful CMS

Для запуска Contentful CMS, последовательно выполните следующие действия:

- Перейти по ссылке <https://www.contentful.com/>;
- Провести авторизацию.

3.2 Проверка работоспособности Contentful CMS

Если после входа отображается Contentful CMS, то это значит, что всё работает корректно.

4 Описание операций

4.1 Операция Выбор типа контента

Наименование: Выбор типа контента

Условия: открыта вкладка «Content»

Основные действия в требуемой последовательности: выбрать тип контента: «Layout Maker».

4.2 Операция Добавление шаблона

Наименование: Добавления шаблона

Условия: Выбран тип контента: «Layout Maker»

Основные действия в требуемой последовательности: нажать на кнопку «Add Layout Maker».

4.3 Операция Заполнения шаблона необходимыми данными

Наименование: Редактирование шаблона

Условия: Добавлен шаблон

Основные действия в требуемой последовательности: заполнить все представленные поля во вкладке «Editor».

4.4 Операция Публикация шаблона

Наименование: Публикация шаблона

Условия: Шаблон создан и отредактирован

Основные действия в требуемой последовательности: нажать на кнопку «Publish».

5 Аварийные ситуации

В случае возникновения аварийных ситуаций, необходимо немного подождать.

6 Рекомендации по освоению

Для более легкого освоения, желательно изучить базовый набор навыков управления контентом в Contentful CMS.

Руководство пользователя составлялось на основе ГОСТ 19.505-79 ЕСПД [22].

ПРИЛОЖЕНИЕ Д

(обязательное)

Код программы

Файл LayoutSelector.tsx

```
export default function LayoutSelector({
  layoutMakerData,
  welcomeText,
}: LayoutSelectorProps) {
  if (!layoutMakerData) {
    return null
  }
  const [isLayoutMakerOpen, setIsLayoutMakerOpen] = useState(false)
  const [selectedLayout, setSelectedLayout] = useState<LayoutType |
null>(null)
  const [order, setOrder] = useState<OrderType>()
  const [formVisible, setFormVisible] = useState(false)
  const [loading, setLoading] = useState(false)
  const [form, setForm] = useState({
    email: "",
    phone: "",
    name: "",
  })
  function onFormChange(e: React.ChangeEvent<HTMLInputElement>) {
    setForm(prevForm => ({ ...prevForm, [e.target.name]: e.target.value }))
  }
  function closeLayoutMaker() {
    document.body.style.overflow = 'auto'
    setIsLayoutMakerOpen(false)
  }
```

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		44

```

}

function openLayoutMaker() {
  document.body.style.overflow = 'hidden'
  setIsLayoutMakerOpen(true)
}

function openForm() {
  setFormVisible(true)
}

function closeForm() {
  setFormVisible(false)
}

function onExport(
  e: {
    file: any
    data: {
      name: string
      type: 'layout' | 'asset'
      width: number
      height: number
      x: number
      y: number
    }
  }[]
) {
  if (!selectedLayout) return
  const data = e.map(file => ({ ...file.data }))
  const images = e.map(file => file.file)
  setOrder({ data, images, id: selectedLayout.id })
  setSelectedLayout(null)
  openForm()
}

```

```

}

function sendOrder(e: React.FormEvent) {
  e.preventDefault()
  async function send() {
    if (!order) return
    const imagesCount = order.images.length ?? 0
    const formData = new FormData()
    for (let i = 0; i < imagesCount; i++) {
      const img = order.images[i]
      const name = order.data[i].name
      const blob = await (await fetch(img)).blob()
      const file = new File([blob], name)
      formData.append('images', file)
    }
    const API_URL = 'https://photonovik-api.herokuapp.com/order'
    formData.append('data', JSON.stringify(order.data))
    formData.append('id', order.id)
    formData.append('email', form.email)
    formData.append('phone', form.phone)
    formData.append('name', form.name)
    formData.append('size', sizeSelectValue)
    try {
      setLoading(true)
      const { url } = await (
        await fetch(API_URL, { method: 'POST', body: formData })
      ).json()
      window.open(url, '_self')
    } catch (e) {
      console.log(e)
    } finally {

```

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		46

```

        setLoading(false)
    }
}
send()
}

function onSizeChange(e: React.ChangeEvent<HTMLSelectElement>) {
    setSizeSelectValue(e.target.value)
}

const [sizeSelectValue, setSizeSelectValue] = useState("")
return (
    <AnimateSharedLayout type='crossfade'>
        <motion.h3
            onClick={openLayoutMaker}
            layoutId='maketMaker'
            className='text-2xl underline text-center pt-3 cursor-pointer'
            transition={{ ease: 'easeInOut' }}
        >
            {welcomeText}
        </motion.h3>
        {isLayoutMakerOpen && (
            <AnimatePresence>
                <div className='fixed z-10 inset-0 overflow-y-auto'>
                    <div className='flex justify-center servicesData-center min-h-screen
pt-4 px-4 pb-20 text-center'>
                        <motion.div
                            initial={{ opacity: 0 }}
                            animate={{ opacity: 1 }}
                            exit={{ opacity: 0 }}
                            className='fixed inset-0 bg-gray-500 bg-opacity-75 transition-
opacity'

```

```

        aria-hidden='true'
        onClick={(e: React.MouseEvent) => e.stopPropagation()}
      />
    <motion.div
      layoutId='maketMaker'
      className='container inline-block align-bottom bg-white rounded-
lg text-left shadow-xl transform sm:my-8 sm:align-middle sm:max-w-lg sm:w-full
h-full py-5 px-5'
    >
      <div>
        <div className='flex justify-between'>
          <Close onClick={closeLayoutMaker} />
          {formVisible && (
            <button
              className='underline focus:outline-none hover:text-gray-400'
              onClick={closeForm}
            >
              Назад
            </button>
          )}
        </div>
        <motion.h4 className='text-2xl text-gray-900 text-center'>
          {formVisible
            ? 'Заполните данные, чтобы после оплаты Вам пришел чек и
мы могли с Вами связаться'
            : 'Выберите макет'}
        </motion.h4>
        {formVisible && (
          <form onSubmit={sendOrder} className='grid gap-3 mt-3'>

```

```
<label className='relative w-full bg-white border border-gray-300 rounded-md shadow-sm px-3 py-2 text-left grid gap-1'>
```

Как к Вам обращаться?

```
<input
  type='text'
  name='name'
  required
  value={form['name']}
  onChange={onFormChange}
  className={
    'relative bg-white border border-gray-300 rounded-md
shadow-sm px-3 py-2 text-left'
  }
/>
```

Почта

```
<input
  type='email'
  name='email'
  required
  value={form['email']}
  onChange={onFormChange}
  className={
    'relative bg-white border border-gray-300 rounded-md
shadow-sm px-3 py-2 text-left'
  }
/>
```

Номер телефона

```
<input
  type='tel'
  name='phone'
```



```

        required
        value={form['phone']}
        onChange={onFormChange}
        className={
            'relative bg-white border border-gray-300 rounded-md
shadow-sm px-3 py-2 text-left'
        }
    />
</label>
<button
    className={`text-white w-full self-end max-h-9 py-1 px-2 bg-
green-500 rounded border border-green-600 focus:outline-none focus:ring-1
hover:bg-green-600 ${
        loading
        ? 'disabled:opacity-50 cursor-not-allowed'
        : ''
    }`}
    onClick={() => ({})}
    disabled={loading}
>
    {loading ? 'Обработка...' : 'Купить'}
</button>
</form>
)}
{!formVisible && (
    <Listbox
        value={selectedLayout}
        onChange={setSelectedLayout}
    >
        {{{ open }}} => (

```

```

<>

<div className='mt-1 relative'>

  <Listbox.Button className='relative w-full bg-white border
border-gray-300 rounded-md shadow-sm pl-3 pr-10 py-2 text-left cursor-default
focus:outline-none focus:ring-1 focus:ring-yellow-500 focus:border-yellow-500'>

    <span className='flex items-center'>
      {selectedLayout ? (
        <>
          <span className='ml-3 block truncate'>
            {selectedLayout.title} -{' '}
            {selectedLayout.price}₽
          </span>
        </>
      ) : (
        <span className='ml-3 block'>
          Выберите макет
        </span>
      )}
    </span>

    <span className='ml-3 absolute inset-y-0 right-0 flex
items-center pr-2 pointer-events-none'>
      <SelectorIcon className='h-5 w-5 text-gray-400' />
    </span>
  </Listbox.Button>

  <Transition
    show={open}
    as={Fragment}
    leave='transition ease-in duration-100'
    leaveFrom='opacity-100'

```

```

        leaveTo='opacity-0'
    >
    <Listbox.Options
        static
        className='absolute z-10 mt-1 w-full bg-white shadow-
lg max-h-56 rounded-md py-1 text-base ring-1 ring-black ring-opacity-5 overflow-
auto focus:outline-none '
    >
        {layoutMakerData.map(layout => (
            <Listbox.Option
                key={layout.id}
                className={{({ active }) =>
                    classNames(
                        active
                        ? 'text-white bg-yellow-600'
                        : 'text-gray-900',
                        'cursor-default select-none relative py-2 pl-3 pr-9'
                    )
                }
                value={layout}
            >
                {{{ selected, active }} => (
                    <>
                        <div className='flex items-center'>
                            <GatsbyImage
                                image={
                                    layout.layout.gatsbyImageData
                                }
                                alt=""
                                className='flex-shrink-0 h-6 w-6 rounded-full'

```

```

/>
<span
  className={classNames(
    selected
      ? 'font-semibold'
      : 'font-normal',
    'ml-3 block truncate'
  )}
>
  {layout.title} - {layout.price}₽
</span>
</div>

```

```

{selected ? (
  <span
    className={classNames(
      active
        ? 'text-white'
        : 'text-yellow-600',
      'absolute inset-y-0 right-0 flex items-center pr-4'
    )}
  >
    <CheckIcon
      className={classNames(
        active
          ? 'text-white'
          : 'text-yellow-600',
        'h-5 w-5'
      )}
    />

```

```

        </span>
        ) : null}
    </>
    })
    </Listbox.Option>
    )))
    </Listbox.Options>
    </Transition>
    </div>
    </>
    })
    </Listbox>
    )}{''}
</div>
{selectedLayout?.sizes.length && (
    <
    Размер{' '}
    <select
        onChange={onSizeChange}
        value={sizeSelectValue}
        name='size'
        className='cursor-pointer relative w-full bg-white border
border-gray-300 rounded-md shadow-sm px-3 py-2 text-left'
    >
        {selectedLayout?.sizes.map(size => (
            <option value={size}>{size}</option>
        ))}
    </select>
    </>
    )}

```

```

    {selectedLayout && (
      <LayoutMaker
        clip={{
          clipX: selectedLayout.clipX,
          clipY: selectedLayout.clipY,
          clipHeight: selectedLayout.clipHeight,
          clipWidth: selectedLayout.clipWidth,
        }}
        onExport={onExport}
        backgroundImage={selectedLayout.background.file.url}
        layoutImage={selectedLayout.layout.file.url}
      />
    )}
  </motion.div>
</div>
</div>
</AnimatePresence>
)}
</AnimateSharedLayout>
)
}

```

Файл LayoutMaker.jsx

```

export default function LayoutMaker({
  onExport,
  backgroundImage,
  layoutImage,
  clip,
}) {
  const [stageWidth, setStageWidth] = useState(0)
  const [stageHeight, setStageHeight] = useState(0)

```

```

const [selectedId, selectShape] = useState(null)
const [files, setFiles] = useState([])
const [orderInputs, setOrderInputs] = useState([nanoid()])
const [isExporting, setIsExporting] = useState(false)
const stageRef = useRef(null)
const stageContainerRef = useRef(null)
const [maxStageWidth, setMaxStageWidth] = useState(0)
const [textValue, setTextValue] = useState("")
useEffect(() => {
  if (!stageContainerRef.current) return
  const stageContainerStyle = window.getComputedStyle(
    stageContainerRef.current,
    null
  )
  setMaxStageWidth(
    stageContainerStyle.width.substring(
      0,
      stageContainerStyle.width.length - 2
    )
  )
}, [stageContainerRef])
useEffect(() => {
  if (!isExporting) return
  exportStageToPng()
  setIsExporting(false)
}, [isExporting])
function updateOrderInputs() {
  setOrderInputs(prev => [...prev, nanoid()])
}
function onAddText(orderNumber) {

```

```

if (!textValue) return
setFiles(prev => [
  ...prev,
  {
    orderNumber,
    x: clip.clipX,
    y: clip.clipY,
    id: nanoid(),
    height: 50,
    width: 100,
    value: textValue,
    type: 'text',
    fontSize: 15,
    color: '#000000',
  },
])
setTextValue("")
updateOrderInputs()
}

function changeFontSizeByOrderNumber(e, orderNumber) {
  setFiles(prevFiles =>
    prevFiles.map(prevFile => {
      if (prevFile.orderNumber === orderNumber) {
        return {
          ...prevFile,
          fontSize: +e.target.value,
        }
      }
    })
  return prevFile
})

```



```

    )
  }
function changeColorByOrderNumber(e, orderNumber) {
  setFiles(prevFiles =>
    prevFiles.map(prevFile => {
      if (prevFile.orderNumber === orderNumber) {
        return {
          ...prevFile,
          color: e.target.value,
        }
      }
      return prevFile
    })
  )
}
function checkDeselect(e) {
  if (e.target?.attrs?.name === 'bg') {
    selectShape(null)
  }
}
function moveForward(e) {
  const id = e.target.id()
  setFiles(prevFiles => {
    const file = prevFiles.find(f => f.id === id)
    const fileIndex = prevFiles.indexOf(file)
    return [
      ...prevFiles.slice(0, fileIndex),
      ...prevFiles.slice(fileIndex + 1),
      file,
    ]
  })
}

```

```

    })
  }
  function onInputChange(e, orderNumber) {
    // get file from input
    const file = e.target.files[0]
    // if file input cancel then delete them
    if (!file) {
      deleteFileByOrderNumber(orderNumber)
      return
    }
    const reader = new FileReader()

    // check if file exist
    const fileAlreadyExist = files.find(f => f.orderNumber === orderNumber)
    if (fileAlreadyExist) {
      reader.onload = () => {
        const img = new window.Image()
        img.src = reader.result
        // replace prevfile by newfile
        setFiles(prevFiles =>
          prevFiles.map(prevFile => {
            if (prevFile.orderNumber === orderNumber) {
              return {
                ...prevFile,
                orderNumber,
                x: clip.clipX,
                y: clip.clipY,
                img: img,
                name: file.name,
                file: reader.result,

```

```

        }
    }
    return prevFile
  })
)
}
} else {
  reader.onload = () => {
    const img = new window.Image()
    img.src = reader.result
    // append new file
    setFiles(prev => [
      ...prev,
      {
        orderNumber,
        x: clip.clipX,
        y: clip.clipY,
        img: img,
        id: nanoid(),
        name: file.name,
        file: reader.result,
        type: 'image',
      },
    ])
  }
  updateOrderInputs()
}
reader.readAsDataURL(file)
}

function deleteFileByOrderNumber(orderNumber) {

```

					09.02.03.ИТ.ДП.00.000.ПЗ	Лист
Изм	Лист	№ Докум.	Подп.	Дата		60

```

        setFiles(prevFiles => prevFiles.filter(f => f.orderNumber !==
orderNumber))

        setOrderInputs(prev => prev.filter(o => o !== orderNumber))
    }

    function handleExport() {
        selectShape(null)
        setIsExporting(true)
    }

    function exportStageToPng() {
        stageRef.current.children.shift()
        console.log(stageRef.current)
        const exported = [
            {
                data: {
                    type: 'layout',
                    name: 'layout.png',
                    width: stageRef.current.attrs.width,
                    height: stageRef.current.attrs.height,
                },
                file: stageRef.current.toDataURL(),
            },
            ...files
                .filter(f => f.type !== 'text')
                .map((file, i) => {
                    return {
                        data: {
                            type: 'asset',
                            name: `${i}-${file.name}`,
                            width: file.width,
                            height: file.height,

```

```

        x: file.x,
        y: file.y,
      },
      file: file.file,
    }
  )),
]
onExport(exported)
}

return (
  <div>
    <button
      onClick={handleExport}
      className='mt-2 text-white w-full py-1 px-2 bg-green-500 rounded
border border-green-600 focus:outline-none focus:ring-1 hover:bg-green-600'
    >
      Купить
    </button>
    <div className='grid gap-1'>
      {orderInputs
        .slice(0)
        .reverse()
        .map(orderNumber => {
          const currentFile = files.find(f => f.orderNumber === orderNumber)
          if (currentFile) {
            if (currentFile.type === 'image') {
              return (
                <div key={orderNumber} className='grid gap-1 py-2 px-0'>
                  <input

```

```

        accept='image/*'
        type='file'
        id={`input-file-${orderNumber}`}
        onChange={e => onInputChange(e, orderNumber)}
        style={{ display: 'none' }}
      />
      <label
        htmlFor={`input-file-${orderNumber}`}
        className='cursor-pointer relative w-full bg-white border
border-gray-300 rounded-md shadow-sm px-3 py-2 text-left grid gap-1'
      >
        {currentFile.name}

        <img
          className='input-file-preview w-full h-auto'
          src={currentFile.img.src}
          alt={currentFile.name}
        />

        <button
          className='text-white w-full self-end max-h-9 py-1 px-2 bg-
red-500 rounded border border-red-600 focus:outline-none focus:ring-1 hover:bg-
red-600'
          onClick={() => deleteFileByOrderNumber(orderNumber)}
        >
          Удалить
        </button>
      </label>
    </div>
  )

```

```

    }
    if (currentFile.type === 'text') {
      return (
        <div key={orderNumber} className='grid gap-1 py-2 px-0'>
          <label
            htmlFor={`input-file-${orderNumber}`}
            className='cursor-pointer relative w-full bg-white border
border-gray-300 rounded-md shadow-sm px-3 py-2 text-left grid gap-1'
          >
            {currentFile.value}
            <input
              type='text'
              value={currentFile.fontSize}
              onChange={e =>
                changeFontSizeByOrderNumber(e, orderNumber)
              }
              className={
                'relative bg-white border border-gray-300 rounded-md
shadow-sm px-3 py-2 text-left'
              }
            />
            <input
              type='color'
              value={currentFile.color}
              onChange={e => changeColorByOrderNumber(e,
orderNumber)}
            />
            <button

```

```
className='text-white w-full self-end max-h-9 py-1 px-2 bg-
red-500 rounded border border-red-600 focus:outline-none focus:ring-1 hover:bg-
red-600'
```

```
onClick={() => deleteFileByOrderNumber(orderNumber)}
```

```
>
```

```
Удалить
```

```
</button>
```

```
</label>
```

```
</div>
```

```
)
```

```
}
```

```
}
```

```
return (
```

```
<div key={orderNumber} className='grid gap-1 py-2 px-0'>
```

```
<div className='grid grid-cols-2 gap-1'>
```

```
<button
```

```
onClick={e => onAddText(orderNumber)}
```

```
className='cursor-pointer relative bg-white border border-gray-
300 rounded-md shadow-sm px-3 py-2 text-left'
```

```
>
```

```
Добавить текст
```

```
</button>
```

```
<input
```

```
type='text'
```

```
id={`input-text-${orderNumber}`}
```

```
onChange={e => setTextValue(e.target.value)}
```

```
value={textValue}
```

```
placeholder='Введите текст'
```

```
className='relative bg-white border border-gray-300 rounded-
md shadow-sm px-3 py-2 text-left'
```



```

        />
    </div>
    <div>
        <input
            accept='image/*'
            type='file'
            id={`input-file-${orderNumber}`}
            onChange={e => onInputChange(e, orderNumber)}
            style={{ display: 'none' }}
        />
        <label
            htmlFor={`input-file-${orderNumber}`}
            className='cursor-pointer relative w-full bg-white border
border-gray-300 rounded-md shadow-sm px-3 py-2 text-left grid gap-1'
        >
            Выберите файл
        </label>
    </div>
</div>
)
)}}
</div>

<div className='flex justify-center' ref={stageContainerRef}>
    <Stage
        id='stage'
        ref={stageRef}
        width={stageWidth}
        height={stageHeight}
        onMouseDown={checkDeselect}

```

```

onTouchStart={ checkDeselect }
name='bg'
style={{ border: '1px solid black' }}
>
<Layer layerName='background'>
  <BackgroundImage
    width={ stageWidth }
    height={ stageHeight }
    imageUrl={ backgroundImage }
  />
</Layer>
<Layer layerName='layout'>
  <LayoutImage
    name='bg'
    setStageHeight={ setStageHeight }
    setStageWidth={ setStageWidth }
    imageUrl={ layoutImage }
    maxWidth={ maxStageWidth }
  />
</Layer>
<Layer
  layerName='resizableImages'
  clipX={ clip.clipX }
  clipY={ clip.clipY }
  clipHeight={ clip.clipHeight }
  clipWidth={ clip.clipWidth }
>
  { files.map((file, i) => {
    if (file.type === 'text')
      return (

```

```

<ResizableText
  text={file}
  key={file.id}
  onSelect={e => {
    moveForward(e)
    selectShape(file.id)
  }}
  onChange={newAttrs => {
    setFiles(prevFiles => {
      return prevFiles.map(f => {
        if (f.id === newAttrs.id) {
          return { ...f, ...newAttrs }
        }
        return f
      })
    })
  }}
  isSelected={file.id === selectedId}
/>
)
if (file.type === 'image')
  return (
    <ResizableImage
      stageHeight={clip.clipHeight}
      stageWidth={clip.clipWidth}
      key={file.id}
      file={file}
      isSelected={file.id === selectedId}
      onSelect={e => {
        moveForward(e)

```

```

        selectShape(file.id)
    }}
    onChange={newAttrs => {
        setFiles(prevFiles => {
            return prevFiles.map(f => {
                if (f.id === newAttrs.id) {
                    return { ...f, ...newAttrs }
                }
                return f
            })
        })
    }}
    />
)
}}
</Layer>
</Stage>
</div>
</div>
)
}

```