

Отчет о проверке на заимствования №1



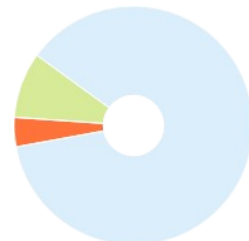
Автор: Царулкова Анастасия Витальевна
Проверяющий: Государев Илья Борисович (igossoudarev@herzen.spb.ru / ID: 10407)
Организация: Российский государственный педагогический университет им. А. И. Герцена
Отчет предоставлен сервисом «Антиплагиат» - <http://herzen.antiplagiat.ru>

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 80
Начало загрузки: 24.05.2022 16:59:29
Длительность загрузки: 00:00:08
Корректировка от 24.05.2022 17:58:30
Имя исходного файла: Диплом Царулкова АВ. Оформленный.docx
Название документа: Разработка клиентского интерфейса онлайн-микросервиса по редактированию изображений на основе импортозамещенных веб-технологий
Размер текста: 1 кБ
Тип документа: Выпускная квалификационная работа
Символов в тексте: 48309
Слов в тексте: 6064
Число предложений: 449

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)
Начало проверки: 24.05.2022 16:59:38
Длительность проверки: 00:01:50
Комментарии: не указано
Поиск с учетом редактирования: да
Модули поиска: ИПС Адилет, Библиография, Сводная коллекция ЭБС, Интернет Плюс, Сводная коллекция РГБ, Цитирование, Переводные заимствования (RuEn), Переводные заимствования по eLIBRARY.RU (EnRu), Переводные заимствования по Интернету (EnRu), Переводные заимствования издательства Wiley (RuEn), eLIBRARY.RU, СПС ГАРАНТ, Модуль поиска "РГПУ им.А.И.Герцена", Медицина, Диссертации НББ, Перефразирования по eLIBRARY.RU, Перефразирования по Интернету, Перефразирования по коллекции издательства Wiley, Патенты СССР, РФ, СНГ, СМИ России и СНГ, Шаблонные фразы, Кольцо вузов, Издательство Wiley, Переводные заимствования



ЗАИМСТВОВАНИЯ

3,54% ■

САМОЦИТИРОВАНИЯ

0% ■

ЦИТИРОВАНИЯ

8,55% ■

ОРИГИНАЛЬНОСТЬ

87,91% ■

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.
Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.
Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.
Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Доля в тексте	Источник	Актуален на	Модуль поиска	Блоков в отчете	Блоков в тексте	Комментарии
[01]	7,07%	7,07%	не указано	13 Янв 2022	Библиография	1	1	
[02]	0,41%	2,45%	m131303 Маркеева Д.С.	13 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	2	6	
[03]	0,29%	2,05%	b120508	24 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	2	5	
[04]	0,24%	1,69%	b117938	13 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	1	5	
[05]	0%	1,52%	70826 http://e.lanbook.com	09 Мар 2016	Сводная коллекция ЭБС	0	3	
[06]	0%	1,52%	Введение в проектирование интеллектуальных интерфейсов https://e.lanbook.com	22 Янв 2020	Сводная коллекция ЭБС	0	3	
[07]	1,47%	1,47%	не указано	13 Янв 2022	Шаблонные фразы	20	20	
[08]	0%	1,45%	Тифлопедагогическая технология коррекции и развития тонко координированных формообразующих движений и действий у старших дошкольников с амблиопией и косоглазием	19 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	0	3	
[09]	0,15%	1,43%	ВКР1.rar/Кадырова (Соловьянюк).docx	28 Дек 2015	Кольцо вузов	1	3	
[10]	0,07%	1,35%	54_252_42_0_0.600_75021620 http://window.edu.ru	13 Фев 2022	Интернет Плюс	1	9	
[11]	0%	1,35%	Сергеев С.Ф., Падерно П.И., Назаренко Н.А. Введение в проектирование интеллектуальных интерфейсов. - СПб.: СПбГУ ИТМО, 2011	23 Апр 2017	Интернет Плюс	0	9	

			http://novsu.ru				
[12]	0,25%	1,3%	Тифлопедагогические приемы развития представлений об эмоциях у младших слабовидящих школьников	21 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	1	3
[13]	0%	1,24%	m130726	17 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	0	4
[14]	0%	1,17%	b120601	13 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	0	2
[15]	0,11%	1,12%	Принципы и правила проектирования пользовательского интерфейса Статья в журнале «Молодой ученый» https://moluch.ru	03 Мая 2021	Интернет Плюс	1	7
[16]	0%	1,11%	Принципы и правила проектирования пользовательского интерфейса Статья в журнале «Молодой ученый» https://moluch.ru	24 Мая 2022	Интернет Плюс	0	7
[17]	0%	1,11%	Принципы и правила проектирования пользовательского интерфейса Статья в журнале «Молодой ученый» https://moluch.ru	03 Мая 2021	Интернет Плюс	0	7
[18]	0%	1,07%	2020 ИЗО 09.03.02 БР Альтман Анастасия Сергеевна	05 Июн 2020	Кольцо вузов	0	3
[19]	0%	1,04%	Своеобразие освоения рельефно-точечного шрифта Брайля слепыми младшими школьниками	19 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	0	2
[20]	0%	0,95%	b120540	24 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	0	2
[21]	0%	0,88%	РАЗВИТИЕ ЛИНГВОСТРАНОВЕДЧЕСКОЙ КОМПЕТЕНЦИИ УЧАЩИХСЯ СТАРШИХ	14 Мая 2022	Модуль поиска "РГПУ им.А.И.Герцена"	0	2
[22]	0%	0,88%	https://books.ifmo.ru/file/pdf/768.pdf https://books.ifmo.ru	19 Янв 2022	Интернет Плюс	0	4
[23]	0%	0,88%	https://books.ifmo.ru/file/pdf/768.pdf https://books.ifmo.ru	02 Мая 2022	Интернет Плюс	0	4
[24]	0%	0,84%	Информационные технологии http://ibooks.ru	09 Дек 2016	Сводная коллекция ЭБС	0	1
[25]	0%	0,84%	Компьютерная графика http://ibooks.ru	09 Дек 2016	Сводная коллекция ЭБС	0	1
[26]	0%	0,84%	rs101003387491.txt http://dlib.rsl.ru	14 Июн 2019	Сводная коллекция РГБ	0	1
[27]	0%	0,84%	Влияние интерфейса на состояние и здоровье оператора. http://elibrary.ru	28 Авг 2014	eLIBRARY.RU	0	1
[28]	0,84%	0,84%	Биотехносфера № 06.2009 http://studentlibrary.ru	19 Дек 2016	Медицина	1	1
[29]	0%	0,79%	2014_ФЗО_ИВТ_ТОРС_БР_Мурзин_Павел_Владимирович.docx	03 Апр 2015	Кольцо вузов	0	2
[30]	0%	0,77%	Современные операционные системы. Учебное пособие http://bibliorossica.com	26 Мая 2016	Сводная коллекция ЭБС	0	1
[31]	0%	0,77%	228775 http://biblioclub.ru	19 Апр 2016	Сводная коллекция ЭБС	0	1
[32]	0%	0,77%	БИОНИЧЕСКИЙ АЛГОРИТМ ПРИНЯТИЯ РЕШЕНИЙ В САПР. http://elibrary.ru	10 Фев 2020	eLIBRARY.RU	0	1
[33]	0%	0,77%	MilovzorovSV_Текстовая часть.docx	25 Фев 2013	Кольцо вузов	0	1
[34]	0%	0,77%	Автоматизация обработки заказов в компьютерном сервисном центре «Оранже»	06 Июн 2016	Кольцо вузов	0	1
[35]	0%	0,77%	Диплом Блинов БИ-122	10 Окт 2018	Кольцо вузов	0	1
[36]	0%	0,76%	Учебное пособие_Проектирование_ЧМИ_.docx	13 Окт 2016	Кольцо вузов	0	2
[37]	0%	0,75%	ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ ДРУЖЕСТВЕННЫХ ИНТЕРФЕЙСОВ ДЛЯ ПОЛЬЗОВАТЕЛЕЙ-ЭКОНОМИСТОВ. http://elibrary.ru	28 Июн 2019	eLIBRARY.RU	0	2
[38]	0%	0,73%	АВТОМАТИЗИРОВАННАЯ СИСТЕМА ПРОДАЖИ ЖЕЛЕЗНОДОРОЖНЫХ БИЛЕТОВ - PDF https://docplayer.ru	13 Авг 2019	Интернет Плюс	0	4
[39]	0%	0,73%	Осинцева_Проектирование_информационных_систем.docx	15 Ноя 2016	Кольцо вузов	0	2
[40]	0%	0,73%	61ИТ_КарповичИЭ_отчет.docx	26 Июн 2020	Кольцо вузов	0	2
[41]	0%	0,73%	Разработка информационной системы аптечной базы	19 Июн 2018	Кольцо вузов	0	2
[42]	0%	0,72%	Интерфейс пользователя - это... Что такое Интерфейс пользователя?	24 Мая 2022	Интернет Плюс	0	4

			https://dic.academic.ru				
[43]	0%	0,72%	Разработка web-приложения и базы данных интернет-магазина. Дипломная (ВКР). Информационное обеспечение, программирование. 2015-07-10 https://bibliofond.ru	24 Апр 2022	Интернет Плюс	0	4
[44]	0%	0,7%	http://venec.ulstu.ru/lib/disk/2017/460.pdf http://venec.ulstu.ru	05 Апр 2022	Интернет Плюс	0	4
[45]	0,39%	0,67%	Тео Мандел Разработка пользовательского интерфейса : [Пер. с англ.] М. 2001 http://dlib.rsl.ru	01 Янв 2001	Сводная коллекция РГБ	1	2
[46]	0%	0,62%	https://elar.urfu.ru/bitstream/10995/107336/1/m_e.n.isakov_2021.pdf https://elar.urfu.ru	14 Янв 2022	Интернет Плюс	0	4
[47]	0%	0,6%	МУ КР Технология разработки информационного и программного обеспечения https://imsit.ru	10 Июл 2020	Интернет Плюс	0	4
[48]	0%	0,6%	МУ КР Проектирование ИС https://imsit.ru	11 Окт 2021	Интернет Плюс	0	3
[49]	0%	0,6%	МУ КР Проектный практикум https://imsit.ru	21 Янв 2021	Интернет Плюс	0	3
[50]	0%	0,59%	https://xn--d1abbusdciv.xn--p1ai/wp-content/uploads/2021/06/%D0%94%D0%B8%D0%BF%D0%BB%D0%BE%D0%BC%D0%BD%D0%B0%D1%8F-%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0-%D0%93%D0%BE%D1%80%D0%BB%D0%BE%D0%B2-%D0%9C.%D0%98.-20.06.21.pdf https://xn--d1abbusdciv.xn--p1ai	23 Фев 2022	Интернет Плюс	0	2
[51]	0%	0,54%	Разработка пользовательского интерфейса http://studentlibrary.ru	27 Ноя 2017	Сводная коллекция ЭБС	0	2
[52]	0,28%	0,54%	Разработка пользовательского интерфейса http://studentlibrary.ru	19 Дек 2016	Медицина	1	2
[53]	0%	0,53%	ВОПРОСЫ ПРИВЛЕЧЕНИЯ К АДМИНИСТРАТИВНОЙ ОТВЕТСТВЕННОСТИ В УСЛОВИЯХ РЕЖИМА ПОВЫШЕННОЙ ГОТОВНОСТИ – тема научной статьи по праву читайте бесплатно текст научно-исследовательской работы в электронной библиотеке КиберЛенинка https://cyberleninka.ru	19 Апр 2022	Интернет Плюс	0	2
[54]	0%	0,52%	ВОЛОНТЕРСКОЕ УЧАСТИЕ МОЛОДЕЖИ В ПЕРИОД ПАНДЕМИИ: ПОРТРЕТ ГЕРОЯ СЛОЖНОГО ВРЕМЕНИ – тема научной статьи по социологическим наукам читайте бесплатно текст научно-исследовательской работы в электронной библиотеке КиберЛенинка https://cyberleninka.ru	24 Мая 2022	Интернет Плюс	0	2
[55]	0%	0,52%	Правовой минимум медицинского работника (врача) http://studentlibrary.ru	26 Янв 2018	Медицина	0	2
[56]	0%	0,52%	Документационное обеспечение деятельности Федерального агентства по делам национальностей	29 Мар 2022	СМИ России и СНГ	0	2
[57]	0%	0,5%	Выпускная магистерская квалификационная работа "Театральные технологии как средство развития социальной активности у младших школьников" https://infourok.ru	25 Апр 2022	Интернет Плюс	0	2
[58]	0%	0,47%	https://books.ifmo.ru/file/pdf/1363.pdf https://books.ifmo.ru	24 Янв 2022	Интернет Плюс	0	2
[59]	0%	0,46%	МУ по выполнению КР УпрИТпр https://imsit.ru	27 Апр 2021	Интернет Плюс	0	2
[60]	0%	0,46%	МУ по выполнению КР ПИС https://imsit.ru	10 Ноя 2020	Интернет Плюс	0	2
[61]	0%	0,46%	Программное обеспечение, ГОСТы https://standartgost.ru	25 Янв 2021	Интернет Плюс	0	2
			ГОСТ Р 51904-2002 Программное обеспечение встроенных систем.				

[62]	0%	0,46%	Общие требования к разработке и документированию, ГОСТ Р от 25 июня 2002 года №51904-2002 http://docs.cntd.ru	22 Мая 2022	Интернет Плюс	0	2
[63]	0%	0,45%	72179 http://e.lanbook.com	10 Мар 2016	Сводная коллекция ЭБС	0	1
[64]	0%	0,45%	https://physics.herzen.spb.ru/wp-content/docs/trebovaniya-vkr-2020.pdf https://physics.herzen.spb.ru	23 Мая 2022	Интернет Плюс	0	1
[65]	0%	0,42%	Набор волонтеров на Международную математическую олимпиаду для школьников http://sanktpeterburg.bezformata.com	20 Мая 2020	СМИ России и СНГ	0	1
[66]	0%	0,38%	Распоряжение Правительства РФ <Об утверждении перечней организаций, подведомственных Министерству науки и высшего образования Российской Федерации, Министерству просвещения Российской Федерации, Рособнадзора и признании утратившими силу актов Правитель...	19 Мая 2020	Интернет Плюс	0	2
[67]	0%	0,38%	Юзабилити информационных систем в образовании: основные методы юзабилити-тестирования. http://elibrary.ru	24 Дек 2016	Перефразирования по eLIBRARY.RU	0	1
[68]	0%	0,37%	Приказ Министерства образования и науки Российской Федерации (Минобрнауки России) от 30 августа 2017 г. N 866 г. Москва "Об утверждении перечня олимпиад школьников и их уровней на 2017/18 учебный год"	27 Дек 2018	СМИ России и СНГ	0	2
[69]	0%	0,36%	Основы интернет-технологий: учебное пособие https://e.lanbook.com	22 Янв 2020	Сводная коллекция ЭБС	0	1
[70]	0%	0,36%	Г. Б. Барская ; Российская Федерация, Министерство образования и науки, ФГБОУ ВПО Тюменский государственный университет, Институт дистанционного образования, Институт математики и компьютерных наук Web-технологии и Web-дизайн учебное пособие Тюмень 2014 http://dlib.rsl.ru	05 Авг 2019	Сводная коллекция РГБ	0	1
[71]	0%	0,34%	Туристско-рекреационный потенциал как основа развития въездного туризма Китая http://dep.nlb.by	16 Янв 2020	Диссертации НББ	0	1
[72]	0%	0,34%	Список использованных источников http://reftrend.ru	29 Янв 2017	Перефразирования по Интернету	0	1
[73]	0%	0,33%	Басов, Олег Олегович Модели и метод синтеза полимодальных инфокоммуникационных систем : диссертация ... доктора технических наук : 05.13.01 Орел 2016 http://dlib.rsl.ru	27 Дек 2019	Сводная коллекция РГБ	0	1
[74]	0%	0,33%	Глобализация образования и проблема формирования экологической культуры http://dep.nlb.by	06 Дек 2018	Диссертации НББ	0	1
[75]	0%	0,31%	Эргономика пользовательских интерфейсов в информативных системах https://book.ru	03 Июл 2017	Сводная коллекция ЭБС	0	1
[76]	0%	0,31%	Бородовская, Анастасия Юрьевна Дизайн электронных образовательных ресурсов в контексте когнитивного восприятия текста читателями : диссертация ... кандидата педагогических наук : 05.25.03 Казань 2016 http://dlib.rsl.ru	27 Дек 2019	Сводная коллекция РГБ	0	1
[77]	0%	0,28%	Замануни университет web сайты Статья в журнале «Молодой ученый» https://moluch.ru	24 Мая 2022	Интернет Плюс	0	1
[78]	0%	0,24%	Национальный стандарт РФ ГОСТ Р 53195.5-2010 "Безопасность функциональная связанных с безопасностью зданий и сооружений систем. Часть 5. Меры по снижению риска, методы оценки" (утв. приказом Федерального агентства по	18 Апр 2017	СПС ГАРАНТ	0	1

			техническому регулированию и метроло... http://ivo.garant.ru					
[79]	0,23%	0,23%	[Б. С. Алешин и др.]; под общ. ред. Б. С. Алешина, К. К. Веремеенко, А. И. Черноморского Ориентация и навигация подвижных объектов : современные информационные технологии : [монография] Москва 2006 http://dlib.rsl.ru	03 Ноя 2017	Сводная коллекция РГБ	1	1	
[80]	0%	0,23%	Президент запретил open source? // Указ об обеспечении технологической независимости и его толкование https://zakon.ru	01 Apr 2022	СМИ России и СНГ	0	1	
[81]	0%	0,22%	Исследование функциональности программ для разработки дизайна сайтов и мобильных приложений Статья в журнале «Молодой ученый» https://moluch.ru	24 Apr 2022	Интернет Плюс	0	2	
[82]	0,16%	0,22%	Дипломная работа: Проектирование информационной технологии APM - BestReferat.ru https://bestreferat.ru	21 Окт 2020	Интернет Плюс	2	2	
[83]	0%	0,22%	https://fgosvo.ru/uploadfiles/Projects_PO_OP/MAG/09.04.01_2.pdf https://fgosvo.ru	15 Фев 2022	Интернет Плюс	0	1	
[84]	0%	0,22%	https://knastu.ru/media/files/page_files/page_1425/SBORNIK_GOTOVYY_3.pdf https://knastu.ru	17 Мая 2022	Интернет Плюс	0	1	
[85]	0%	0,22%	https://dspace.tltsu.ru/bitstream/123456789/3754/1/%D0%97%D1%83%D0%B5%D0%B2%D0%B0%20%D0%A1.%D0%95_%D0%94%D0%98%D0%97%D0%B1-1301.pdf https://dspace.tltsu.ru	13 Мая 2022	Интернет Плюс	0	1	
[86]	0,11%	0,21%	Литвинюк, Ярослав Валерьевич Совершенствование управления сбалансированностью федерального бюджета Российской Федерации в современных условиях : автореферат дис. ... кандидата экономических наук : 08.00.10 Москва 2007 http://dlib.rsl.ru	07 Окт 2010	Сводная коллекция РГБ	1	1	
[87]	0%	0,2%	Максимова, Марина Владимировна Практико-ориентированная образовательная среда как средство развития учебной мотивации обучающихся колледжа : диссертация ... кандидата педагогических наук : 13.00.08 Орел 2017 http://dlib.rsl.ru	15 Окт 2019	Сводная коллекция РГБ	0	1	Источник исключен. Причина: Маленький процент пересечения.
[88]	0%	0,2%	https://elar.urfu.ru/bitstream/10995/100808/1/m_th_i.n.savichev_2021.pdf https://elar.urfu.ru	22 Янв 2022	Интернет Плюс	0	2	Источник исключен. Причина: Маленький процент пересечения.
[89]	0%	0,19%	не указано	13 Янв 2022	Цитирование	0	1	Источник исключен. Причина: Маленький процент пересечения.
[90]	0%	0,19%	http://elar.uspu.ru/bitstream/uspu/5231/1/mon00053.pdf http://elar.uspu.ru	24 Мая 2022	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[91]	0%	0,19%	https://nauchkor.ru/uploads/documents/5c1a76ee7966e104f6f85b2c.pdf https://nauchkor.ru	29 Apr 2022	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[92]	0%	0,18%	Государственная поддержка инновационного предпринимательства. http://elibrary.ru	15 Янв 2017	eLIBRARY.RU	0	1	Источник исключен. Причина: Маленький процент пересечения.
[93]	0%	0,17%	https://www.bsuir.by/m/12_100229_1_141272.pdf https://bsuir.by	06 Мая 2022	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[94]	0%	0,17%	Разработка страницы клиента IT-компании, разрабатывающей визуальные решения для застройщиков ООО «Технософт» [Курсовая №14301] https://evkova.org	17 Мая 2022	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[95]	0%	0,17%	https://elar.rsvpu.ru/bitstream/123456789/29073/1/RVSPU_2019_069.pdf https://elar.rsvpu.ru	06 Apr 2022	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[96]	0%	0,14%	Первая программа на C# GeekBrains - образовательный портал https://gb.ru	19 Мая 2022	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
			Применение возможностей MS Excel					

[97]	<div>0%</div>	0,13%	для разработки служебной документации по эксплуатации и ремонту бронетанкового вооружения и техники. http://elibrary.ru	12 Мар 2019	eLIBRARY.RU	0	1	Источник исключен. Причина: Маленький процент пересечения.
[98]	<div>0%</div>	0,13%	Justering i css. Juster tekst lodret og centrér CSS. Absolut placering af barnet https://halzen.ru	24 Мая 2022	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[99]	<div>0%</div>	0,13%	CSS: выравнивание по центру http://w3.org	24 Мая 2022	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[100]	<div>0%</div>	0,11%	https://codepen.io/shshaw/pen/apwMwM.less https://codepen.io	24 Мая 2022	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.
[101]	<div>0%</div>	0,1%	Развитие социально-регулятивных компетенций обучающихся в процессе коллективной музыкально-творческой деятельности – тема научной статьи по наукам об образовании читайте бесплатно текст научно-исследовательской работы в электронной библиотеке КиберЛенинка https://cyberleninka.ru	07 Апр 2022	Интернет Плюс	0	1	Источник исключен. Причина: Маленький процент пересечения.

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ ³ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ⁷
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ⁷
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»



⁷ Направление подготовки/специальность
09.03.01 Информатика и вычислительная техника

направленность (профиль)/специализация
«Технологии разработки программного обеспечения»

² **Выпускная квалификационная работа**

Разработка клиентского интерфейса онлайн-микросервиса по редактированию
изображений на основе импортозамещенных веб-технологий

Обучающегося 4 курса
очной формы обучения ⁴
Царулковой Анастасии Витальевны

Руководитель ⁷ выпускной квалификационной
работы:
кандидат педагогических наук, доцент ³
Государев Илья Борисович

Санкт-Петербург
2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И СУЩЕСТВУЮЩИХ РЕШЕНИЙ. ПОДГОТОВКА К РАЗРАБОТКЕ	6
1.1. Анализ клиентских веб-интерфейсов и современных трендов	6
1.2. Анализ существующих зарубежных аналогов	13
1.3. Анализ программных средств, использованных при разработке	16
ГЛАВА 2. РАЗРАБОТКА КЛИЕНТСКОГО ИНТЕРФЕЙСА ОНЛАЙН ГРАФИЧЕСКОГО РЕДАКТОРА	23
2.1 Создание макета интерфейса	23
2.2. Кодирование клиентского интерфейса	27
ЗАКЛЮЧЕНИЕ	39
СПИСОК ЛИТЕРАТУРЫ	41

ВВЕДЕНИЕ

Пользовательские интерфейсы являются неотъемлемой частью жизни современного человека. Они рассматриваются как инструменты управления устройствами, приложениями, программами. Профессионалы и обычные пользователи ежедневно взаимодействуют с ними, общаясь в социальных сетях, заказывая товары через интернет, смотря фильмы или сериалы онлайн, а также занимаясь своей профессиональной деятельностью, например, разработкой программного обеспечения, ретушью фотографий и другим. Учитывая факты, можно сделать вывод, что при разработке пользовательских интерфейсов необходим серьезный и глубокий исследовательский подход. Успех приложения или программы во многом зависит от качества проработки пользовательского интерфейса. В профессиональных редакторах, таких, как например Adobe Photoshop, большое количество функций и инструментов, они распределены и сгруппированы по всему экрану, и обычному человеку достаточно сложно разобраться, какой инструмент за что отвечает и как реализовать свою идею с помощью многообразия всего предоставляемого функционала. С данной точки зрения, чтобы создать продукт, который сможет использовать любой желающий без глубоких знаний о конкретном программном обеспечении и получать хороший результат, требуется найти баланс в количестве функций и инструментов и расположить их наиболее удобным для пользователя образом. Также стоит отметить, что на данный момент существует не так много доступных для быстрого освоения онлайн графических редакторов российского производства, которые функционируют на российских серверах.

Таким образом, актуальность дипломной работы связана с небольшим выбором среди веб-сервисов для обработки изображений с простым интерфейсом и находящихся на серверах Российской Федерации.

Цель заключается в разработке клиентского интерфейса онлайн-микросервиса по редактированию изображений на основе импортозамещающих веб-технологий.

Объект исследования – пользовательские интерфейсы.

Предмет исследования – клиентский интерфейс онлайн-микросервиса по редактированию изображений.

В соответствии с поставленной целью необходимо решить следующие задачи:

- Провести анализ трендов современных клиентских веб-интерфейсов;
- Провести анализ существующих зарубежных аналогов;
- Определить необходимый функционал клиентской части онлайн-микросервиса по редактированию изображений;
- Создать макет интерфейса;
- Разработать клиентскую часть сервиса в соответствии с созданным макетом.

Практическая значимость заключается в получении удобного интерфейса для онлайн-микросервиса по редактированию изображений, который позволит быстро и легко ориентироваться в функционале микросервиса.

В первой главе представлен анализ предметной области, исследование последних тенденций в разработке интерфейсов и выбор основных функций сервиса.

Во второй главе представлены прикладные аспекты разработки интерфейса – техническое задание, макет, прототип, используемые библиотеки и собственно разработка клиентской части.

Выпускная квалификационная работа имеет следующую структуру: введение, две главы, заключение, список литературы, включающий 21 источник.

Общий объем работы включает 44 страницы. В работе представлено 24 рисунка.

ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И СУЩЕСТВУЮЩИХ РЕШЕНИЙ. ПОДГОТОВКА К РАЗРАБОТКЕ

1.1. Анализ клиентских веб-интерфейсов и современных трендов

Качество пользовательских интерфейсов совершенствуется с каждым годом. При разработке очень важно не отставать от современных идей и следить за ситуацией на рынке. Создание «сильного» интерфейса для любой компании требует много времени и средств, но в будущем он станет одним из ключевых критериев выбора именно этого продукта пользователем, а не другого. Но часто разработчики забывают и «сосредотачиваются на технологии и порождают решения, слабоуправляемые и неудобные в применении» [1].

Если пользователю будет не удобно взаимодействовать с интерфейсом какого-то продукта, его будет отталкивать внешний вид, то он быстро закроет и удалит этот продукт, а цель эффективного интерфейса, удержать пользователя в сервисе как можно дольше. Кроме того, неправильно спроектированный интерфейс может стать источником дискомфорта, стресса человека. С целью понять, как этого лучше всего этого избежать, в данном параграфе будут проанализированы существующие клиентские веб-интерфейсы и тенденции последних лет.

Рассмотрим основные используемые понятия.

Клиент – это часть программного обеспечения, которая работает на компьютере пользователя и взаимодействует с сервером.

В клиентскую сторону приложения входит интерфейс и часть функций. Данные функции, обычно, могут быть расположены на клиенте по

нескольким причинам. Они, в соответствии с поставленными задачами, должны обрабатываться быстро, или же для их работы нет необходимости затрачивать ресурсы сервера.

Клиентский (пользовательский) интерфейс – элементы и компоненты программы, способные оказывать влияние на взаимодействие пользователя с программным обеспечением. К ним относятся:

- Средства отображения информации, отображаемая информация;
- Язык «пользователь-интерфейс»;
- Устройства и технологии ввода данных;
- Диалоги взаимодействия и транзакции между пользователем и компьютером;
- Обратная связь с пользователем и др. [2]

Фронтенд – это непосредственно сам клиентский код. В нем происходит синтез трёх технологий:

- HTML (HyperTextMarkupLanguage) – код, который позволяет структурировать и отобразить веб-страницу и её контент. Также это язык гипертекстовой разметки, взаимодействующий с браузером на стороне клиента [3].
- CSS (Cascading Style Sheets) – каскадные таблицы стилей – код, благодаря которому можно придать странице форму, оригинальный вид. Один из инструментов для создания комфортной среды пользователя.
- JavaScript – язык программирования, который позволяет взаимодействовать с данными в HTML и CSS, обеспечить логику взаимодействия. С его помощью можно добавить на страницу или приложение интерактивные элементы.

Познакомимся с принципами разработки интерфейса – концепциями высокого уровня, которые могут использоваться при проектировании программного обеспечения. После чего определим, какой из них наиболее ⁴⁵ важен в данной разработке. Кратко они звучат так:

1. Контроль пользователем интерфейса;
2. Уменьшение загрузки памяти пользователя;
3. Последовательность пользовательского интерфейса ⁵² [4].

В соответствии с первым принципом, интерфейс должен дать пользователю возможность контроля, удовлетворять потребности и желания пользователя. Следовать данному принципу помогают определенные правила. Например, возможность пользователя свободно ориентироваться в интерфейсе: при движении вглубь интерфейса должна быть возможность легко вернуться обратно. Если пути движения для пользователя неочевидны, то нужно вводить контекстные подсказки. Он должен не бояться нажимать на интерактивные элементы, переходить на другие экраны. Также нужно уметь сфокусировать внимание пользователя и не заставлять его заканчивать цепочку действий раньше времени и теряться. Важно дать человеку возможность сохранить произведенные действия.

По второму принципу не следует заставлять пользователя держать в голове большое количество информации, а переложить данную задачу на клиентский интерфейс. Для этого система должна запоминать и хранить введенную в неё информацию (например форма адреса или пароля). Лучше всего при возможности предоставлять пользователю возможность выбора из списка заранее заготовленных данных, вместо ввода их вручную. Визуально интерфейс должен выглядеть как можно более структурировано: данные и объекты сгруппированы, наиболее важная информация выделена особым образом и общее количество демонстрируемой информации не перегружает человека.

Интерфейсы, спроектированные по третьему принципу, должны предоставить пользователю возможность применить свой прошлый опыт взаимодействия с интерфейсами. Возможно обучение пользователя идеям программного обеспечения на новых ситуациях. Следует предупреждать человека, если результат его действий будет отличаться, от того, что он может ожидать. Важно в таком случае добавить отмену совершенных действий и их воспроизведения, чтобы избавить человека от страха ошибки. Подобный дружественный интерфейс создается с применением различных способов поощрения его изучения.

В данной работе были применены первое и третье правила. Пользователь сможет легко перемещаться по интерфейсу и функционалу, выбирать инструменты, а сам интерфейс будет разработан таким образом, чтобы человек смог применять в нем свои навыки, полученные в прошлом. Инструменты и функции будут всем знакомы, а их поведение ожидаемо и интуитивно.

Теперь перейдем к элементам дизайна интерфейса. Рассмотрим имеющуюся информацию за последние несколько лет.

В 2021 году был актуален глассоморфизм (стекломорфизм) — стиль, в котором имитируется стеклянная поверхность. Пример использования этого стиля на рисунке 1. К его характерным особенностям относятся:

- Эффект матового стекла с использованием размытия фона;
- Многослойность, объекты, плавающие в пространстве;
- Яркие цвета;
- Тонкая светлая граница на стеклянных объектах.

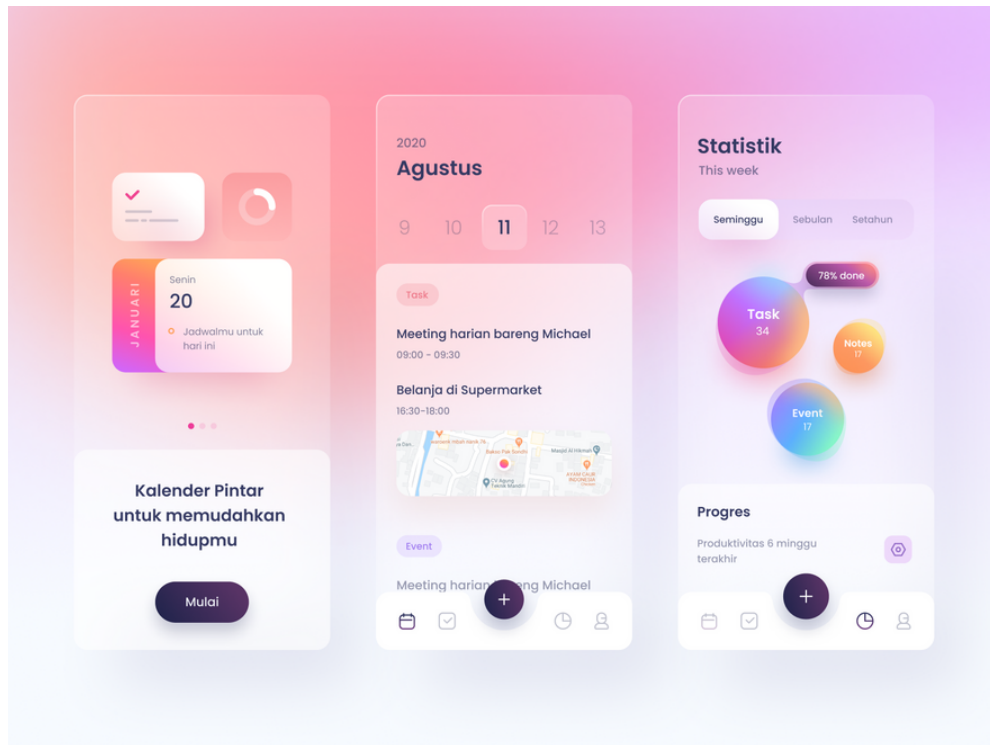


Рисунок 1 — Пример интерфейса в стиле стекломорфизм, автор Ghani Pradita

Этот стиль достаточно сложно правильно использовать в дизайне пользовательских интерфейсов, так как необходимо найти такое сочетание его элементов, чтобы в будущем он не затруднял чтение информации для пользователя. Также он не подойдет для графического редактора, потому что будет сильно мешать пользователю, работающему с изображением, которое может иметь множество цветов. Таким образом общая картинка на экране станет рябящей, что безусловно вызовет дискомфорт у человека.

Следующий актуальный стиль — минимализм, главная идея которого ясна из названия — сведение к минимуму элементов, деталей, цветов. Такой стиль позволяет пользователю легко и быстро ориентироваться в интерфейсе, находить нужные функции, не отвлекаться от поставленных целей. Пример использования минимализма показан на рисунке 2. Характерные черты данного стиля:

1. Крупные, читаемые заголовки
2. Продуманное использование цветов

3. Сосредоточение внимания на контрасте
4. Ограниченное использование эффектов [5]

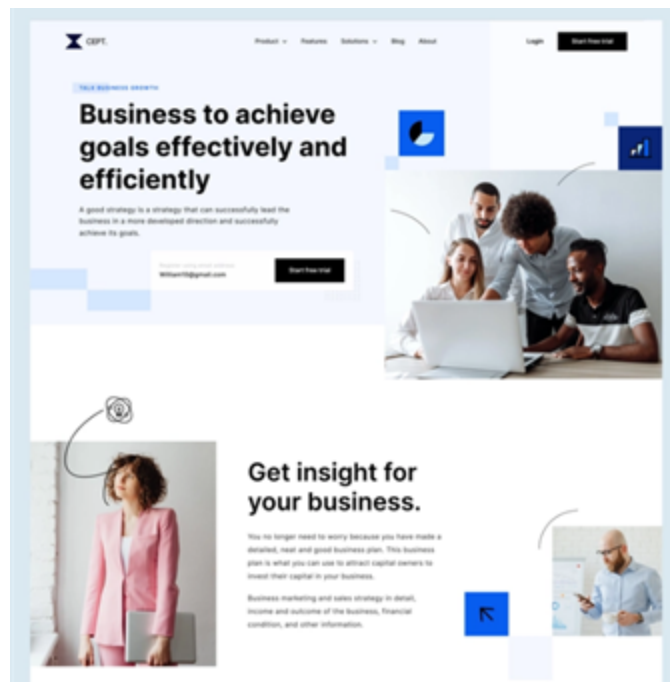


Рисунок 2 — Пример интерфейса в стиле минимализм, источник google.com

Брутализм молодой тренд, отличительными чертами которого являются:

- четкие контуры,
- черный или белый фон,
- мало изображений, они не обработаны,
- ошибки в дизайне,
- один шрифт для всего,
- отсутствие анимации,
- минимум оформления.

Цель брутализма произвести эмоциональный эффект, вывести функциональную часть на первый план, осуществить провокацию, чтобы как можно больше людей узнало о продукте. Данный стиль используют для дизайна всего продукта крайне редко, как эксперимент или высказывание. В реальных проектах могут использоваться отдельные его элементы для

придания чувства новизны. На рисунке 3 пример интерфейса в стиле брутализм.



Рисунок 3 — Пример интерфейса в стиле брутализм, источник: <https://infogra.ru/ui/9-glavnyh-trendov-v-ux-ui-dizajne-v-2021-godu?hcb=1>

Из всех описанных выше стилей наиболее подходящим для данного проекта будет минималистичный, так как в графическом редакторе на первом месте стоит функционал. Ничто не должно отвлекать пользователя от работы. Этому способствует удобство навигации, однородность иконок. Все нужные инструменты должны быть под рукой и их расположение должно быть легко считываемо. В основной рабочей области центральным элементом безусловно будет изображение, загруженное пользователем, поэтому там будет спокойный однотонный фон.

После выбора стиля оформления интерфейса, необходимо определить функционал, который будет работать на стороне клиента. К нему относятся базовые функции и инструменты графического редактора, такие как:

- отрисовка (рендер) загруженного изображения,
- сохранение изображения
- кисть,
- палитра,
- ластик,
- геометрические фигуры: круга, прямоугольника, треугольника,

- отправка запросов на сервер для осуществления загрузки изображения, поворота.

Инструменты, связанные с поворотом изображения, его обрезкой и наложением фильтров будут обрабатываться на стороне сервера. На стороне клиента будут кнопки для отправки запроса на сервер. Также посредством интерфейса будет реализована возможность загрузки изображения на сервер и сохранение его на локальный диск.

1.2. Анализ существующих зарубежных аналогов

На зарубежном рынке существует множество продуктов относящихся к онлайн-редакторам изображений. Проведем анализ некоторых из них, чтобы выяснить, какие слабые и сильные стороны в них есть, какие интересные идеи были использованы, и что ещё можно привнести нового в данную работу.

Первым для анализа выбран онлайн-фоторедактор Pixlr E. На официальном сайте сообщается, что это сервис №1 для редактирования фото. Данный сервис имеет бесплатную версию. В ней рядом с интерфейсом редактора на странице находится рекламный блок и есть ограничения некоторых функций, и доступны только базовые инструменты. Для отключения рекламного блока можно приобрести Премиум подписку, которая также дает доступ к большому библиотеке декоративных элементов.

Интерфейс редактора выполнен в нейтральных серо-чёрных тонах с яркими акцентами на активных элементах, как представлено на рисунке 4. Слева находится список инструментов в виде иконок в количестве 26 штук. При наведении на иконку всплывает подсказка по инструменту. Ниже расположены два круга с цветами — основным и дополнительным, которые

по нажатию открывают палитру выбора цвета. Наверху помещена навигация по фильтрам и дополнительным функциям редактирования. При нажатии на вкладку открывается выпадающее меню. Чуть ниже расположена полоса параметров выбранного инструмента.

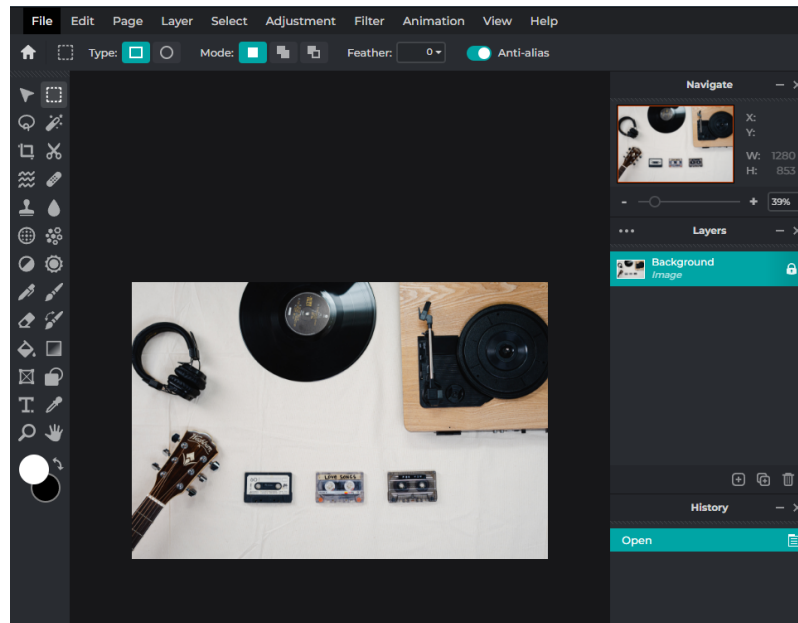


Рисунок 4 — интерфейс Pixlr E

В левой части экрана находится навигация по изображению, список слоёв и блок истории действий. Центральной частью интерфейса является рабочая область. К достоинствам данного интерфейса можно отнести:

- минималистичность;
- привычное расположение инструментов и настроек, для тех, кто уже имел опыт работы с графическими редакторами;
- всплывающие подсказки, для тех, кто только знакомится инструментами подобного рода.

Главной особенностью данного сервиса можно назвать широкий функционал для глубокой обработки изображений. Таким образом, он представляет собой бесплатную альтернативу Adobe Photoshop. Инструментов очень много, что не подходит для решения задачи, поставленной в данной выпускной

7
квалификационной работе, но общую конструкцию можно взять в пример для разработки.

Второй онлайн-редактор под названием Befunky. Позиционируется компанией как сервис для того, чтобы можно было просто и быстро создавать профессиональные фотографии и дизайны без специального образования. Интерфейс представлен на рисунке 5. Он выполнен в тёмных тонах с небольшими яркими цветовыми акцентами. Сервис предлагает несколько разных вариантов для загрузки изображений, как с помощью технологии Drag and Drop, так и стандартным образом, а также можно выбрать что-то из списка предзагруженных изображений.

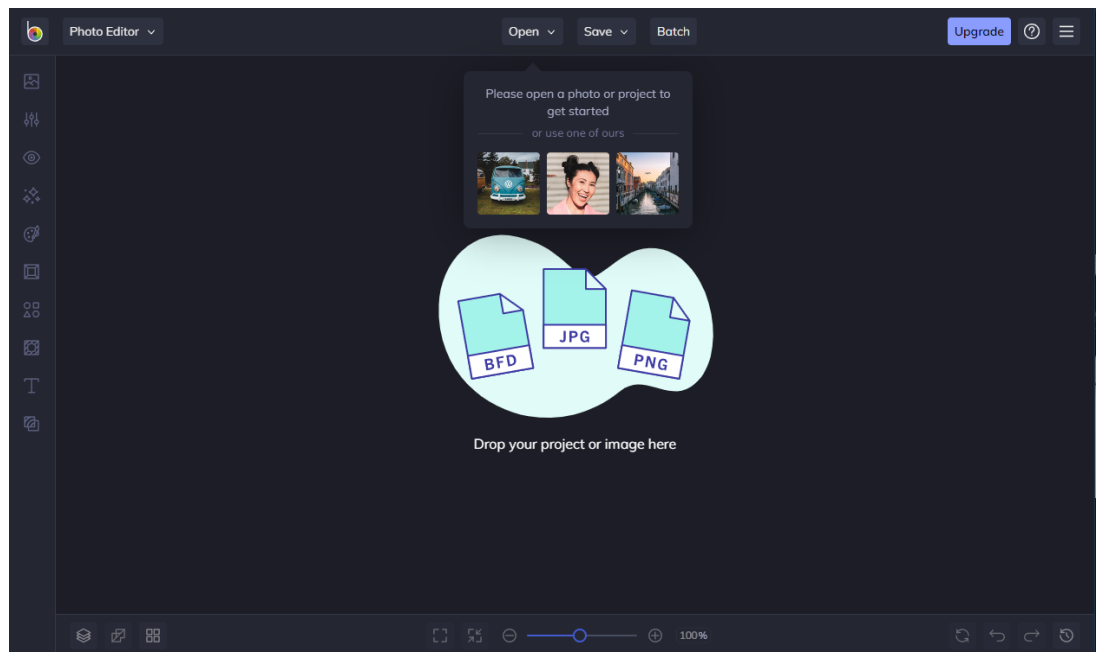


Рисунок 5 — интерфейс Befunky

После загрузки изображения раскрывается обширный список инструментов слева. По нажатию на выбранный инструмент вкладка навигации сменяется на вкладку расширенных настроек. Кроме инструментов присутствуют многочисленные фильтры, маски, стикеры и др. В нижней части экрана находится настройка масштаба, кнопки движения по истории действий и другие полезные функции, но часть из них доступна

только после подключения платной подписки. Основную часть экрана занимает рабочая область.

К достоинствам данного интерфейсам можно отнести:

- минимальную загроуженность экрана: все лишние меню, навигации можно скрыть;
- минималистичный дизайн;
- интерактивность: все изменения быстро и удобно применяются;
- подсказки к инструментам для новичков.

В данном редакторе также очень много различных инструментов и фильтров, в которых нужно разбираться и ориентироваться. Интерфейс построен не совсем стандартно, к расположению компонентов нужно время, чтобы привыкнуть и запомнить.

Подводя итог, можно сделать вывод, что при создании интерфейсов графических редакторов часто используются определенные паттерны, для того чтобы знакомство пользователя с новым для себя интерфейсом происходило как можно легче и быстрее. Довольно распространено использование широкого списка инструментов и готовых фильтров, но часть функций доступна только за дополнительную плату. Цветовая гамма интерфейсов в основном тёмная.

1.3. Анализ программных средств, использованных при разработке

При разработке был использован Visual Studio Code — редактор исходного кода, разработанный Microsoft. Это легкий и бесплатный редактор для разных платформ. Он переведен на множество языков, в том числе и на русский. В нем можно комфортно и эффективно работать с проектами любого размера и направленности благодаря возможности устанавливать нужные

расширения, создаваемые сообществом. В данный редактор также входит отладчик, подсветка синтаксиса, автодополнение кода, инструменты работы с Git. Таким образом Visual Studio Code идеально подходит для данной разработки.

Дополнительно при разработке применялись браузеры Mozilla Firefox и Opera для тестирования работоспособности клиентского интерфейса в браузерах на разных движках (Mozilla Firefox на движке Quantum, Opera на Chromium).

Язык программирования, на котором был написан клиентский код — JavaScript. Выбор данного языка программирования обусловлен тем, что он представляет собой инструмент для управления элементами страниц, написания клиентских сценариев и разработку кроссплатформенных программ. Он изначально заточен под работу с браузерами и имеет большое количество библиотек для решения широкого спектра задач.

Из всего многообразия библиотек для JavaScript для разработки пользовательских интерфейсов была выбрана библиотека React. Это одна из самых известных и широко используемых JavaScript-библиотек. В React разрабатываемые интерфейсы состоят из отдельных компонентов, что позволяет сократить количество возможных ошибок в будущем. Данная библиотека основана на декларативном подходе в программировании, т. е. описании конечного результата, она предоставляет мощную ментальную модель. Она использует функциональные и объектно-ориентированные концепции программирования. [6] Также у нее есть хорошо проработанная документация и множество материалов, статей, курсов и т. Д. Важной особенностью React является использование JSX. React использует возможности современного JavaScript для многих своих шаблонов. JSX расширяет синтаксис ECMAScript, так что HTML-подобный код может сосуществовать рядом с ним. Но браузер не распознает JSX, поэтому этот код

необходимо преобразовывать с помощью препроцессоров или транспилаторов, например, Babel. Таким образом, используя JSX можно писать краткие структуры, похожие на HTML (древовидные структуры, как DOM), в том же файле, что и основной код JavaScript. После Babel преобразует эти структуры в обычный JavaScript-код.

Преимущества JSX:

- JSX может легко конвертировать HTML-теги в элементы React.
- Быстрее, чем обычный JavaScript.
- Позволяет помещать HTML-элементы в DOM без использования метода `appendChild()` или `createElement()`.
- Делает проще написание HTML в React.
- Так как JSX — это выражение, мы можем использовать его внутри операторов `if` и циклов `for`, присваивать его переменным, принимать в качестве аргументов или возвращать из функций.
- JSX предотвращает атаки XSS (межсайтовый скриптинг), широко известные как инъекционные атаки.
- Он типобезопасен (type-safe), то есть надёжен в применении его системы типов, и большинство ошибок можно обнаружить во время компиляции.

К недостаткам использования библиотеки можно отнести:

- средний порог вхождения,
- увеличение размера приложения,
- приложение может запускаться медленнее, так как нужно выполнить код в браузере,
- затрата ресурсов на хранение виртуальных деревьев DOM. [7]

Изучив статьи и приведенную в них статистику [8, 9, 10], можно сделать вывод о том, что React, на протяжении уже многих лет не только не

теряет свою популярность и распространенность, но и наращивает их. В 2020 году был проведен опрос на сайте Stack Overflow, на котором программисты могут задавать друг другу вопросы по своей работе и, следовательно, это сайт, который имеет большое и живое сообщество разработчиков. В нём приняли участие 57 378 человек, 68,9% из которых ответили, что заинтересованы в продолжении работы с React.js. На рисунке 6 представлен график самых используемых в мире фреймворков за 2021 год.

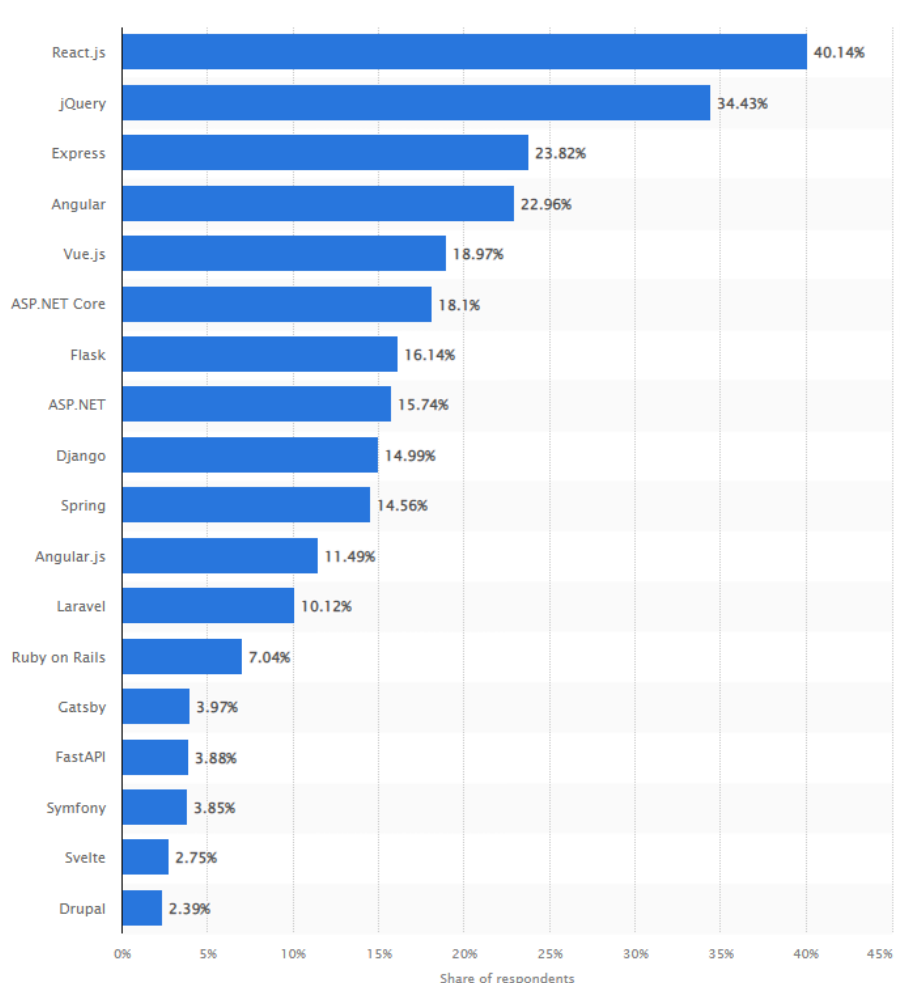


Рисунок 5 График самых используемых в мире фреймворков за 2021 год

При создании проекта использовалась Node.js — асинхронная исполнительная платформа JavaScript, которая управляется событиями и включает стандартную библиотеку [11]. С её помощью можно писать js-код вне среды браузера. В неё также включено средство для установки и обновления пакетов — Node Package Manager (npm).

Create React App. Это инструмент командной строки, который автоматически генерирует проект на React с шаблонами для быстрого начала работы. Благодаря ему не нужно осуществлять ручную настройку Webpack, Babel и Eslint [12]. Также он оптимизирует приложение для финальной стадии разработки, делает сам процесс разработки более комфортным посредством:

- включения всех пакетов JavaScript, необходимых для запуска React-проекта,
- возможности одной командой запустить локальный сервер с приложением, который будет автоматически обновляться при сохранении изменений в проекте
- возможности одной командой осуществить готовую для работы сборку приложения и др.

Пакет react-dom. Предназначен для сопряжения с основным пакетом react. В него входят специальные методы для DOM, используемые на верхнем уровне приложения.

Пакет react-router-dom. Содержит привязки для использования React Router. В свою очередь React Router — это легкая, полнофункциональная библиотека маршрутизации для React. С её помощью в проекте происходит переключение страницы авторизации на рабочее пространство.

Пакет react-scripts. Этот пакет содержит скрипты и конфигурацию, используемые Create React App, именно он устанавливает Babel, Webpack, и Eslint и автоматически настраивает их.

Библиотека **reactstrap.** Это библиотека react-компонентов со стилями Bootstrap 5. В отличие от некоторых библиотек компонентов, Reactstrap не встраивает свои собственные стили, а вместо этого зависит от CSS-фреймворка Bootstrap. Это позволяет иметь согласованные стили для

собственных компонентов на основе React и статических частей сайта, а также позволяет при необходимости включать собственную тему Bootstrap. В отличие от использования Bootstrap в HTML, Reactstrap автоматически экспортирует все правильные классы Bootstrap, и ему не нужно использовать или включать файлы из данной библиотеки или добавлять атрибуты данных для активации функций. Вместо этого компоненты определяются в дружественных к React компонентах с соответствующими параметрами, которыми можно управлять.

Пакет react-sidenav. В него входит компонент боковой навигации с интерактивными элементами: открытием и закрытием по нажатию на кнопку «гамбургер» / «крестик», меню, разворачивающимся при наведении на раздел.

Пакет react-colorful. Компонент с простой палитрой цветов. Она легкая и быстрая, потому что:

- не имеет зависимостей (никаких рисков с точки зрения уязвимостей, никаких неожиданных изменений размера пакета);
- построена только с помощью хуков и функциональных компонентов (для них нет классов и полифиллов);
- включает только минимальное количество оптимизированных вручную алгоритмов преобразования цветов (в то время как большинство популярных сборщиков импортируют целые библиотеки цветовых манипуляций, которые увеличивают размер пакета более чем на 10 КБ и замедляют работу приложения).

Выводы по главе 1:

1. Проведен анализ требований к клиентским веб-интерфейсам и современные в их оформлении. Сделан вывод о том, что при разработке необходимо применять правило контроля пользователем интерфейса и

последовательность в опыте взаимодействия с данным интерфейсом. Пользователь должен иметь возможность применять свой предыдущий опыт и свободно контролировать действия. В отношении дизайна следует применять принципы минимализма.

2. Проведен анализ существующих зарубежных аналогов. Сделан вывод об общих тенденциях в разработке интерфейсов графических редакторов: тёмная цветовая гамма, слева меню инструментов, сверху дополнительные функции, справа или снизу навигация.
3. Определён функционал и инструментарий, который будет работать на основе клиентской части приложения. Это основные инструменты графического редактора.
4. Проанализированы и определены инструменты и библиотеки, которые будут использоваться и при разработке. Среда для написания кода Visual Studio Code, язык программирования JavaScript, библиотеки React.js и Reactstrap, а также дополнительные зависимости.

ГЛАВА 2. РАЗРАБОТКА КЛИЕНТСКОГО ИНТЕРФЕЙСА ОНЛАЙН-МИКРОСЕРВИСА ПО РЕДАКТИРОВАНИЮ ИЗОБРАЖЕНИЙ

Разработка программного продукта проводилась на основе процесса разработки в соответствии с жизненным циклом ПО, взятого из ГОСТ Р 51904-2002. Он включает в себя такие этапы, как [13]:

- процесс определения требований к ПО;
- процесс проектирования ПО;
- процесс кодирования ПО;
- процесс интеграции.

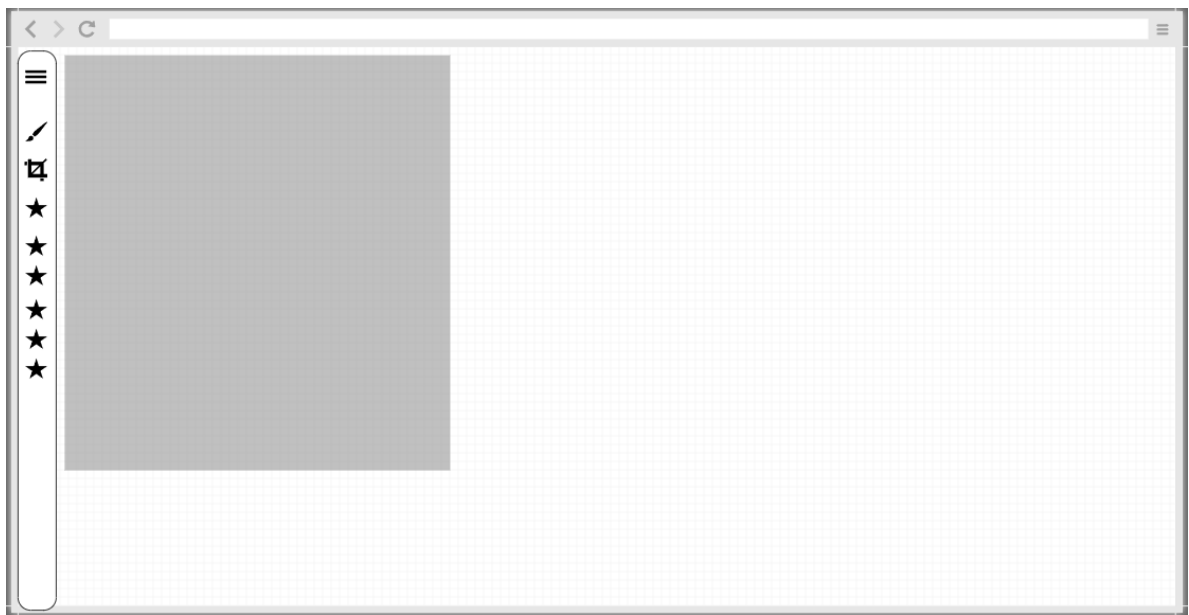
На первом этапе было составлено техническое задание на основе ГОСТ 19.201-78 [14]. Полный текст технического задания находится в приложении А.

2.1 Создание макета интерфейса

Второй этап разработки проводился посредством создания макета в сервисе Pidoco. Pidoco позволяет создавать интерактивные макеты, каркасы и прототипы графического интерфейса без программирования. Он работает в веб-браузере. Также, своей работой можно поделиться с коллегами, рецензентами и пользователями тестирования для редактирования, обсуждения и утверждения в режиме реального времени. Подобный подход улучшает процесс разработки пользовательских интерфейсов для веб-, мобильных и настольных приложений.

По требованиям из технического задания был создан макет интерфейса, представленный на рисунке 7. В состав интерфейса будут входить такие элементы как:

- рабочая область,
- меню инструментов,
- дополнительные модальные окна.



Пользователь может раскрыть меню, нажав на кнопку «Гамбургер», и увидеть подписи-подсказки к инструментам, что видно на рисунке 8.



Некоторые инструменты или функции будут иметь возможность настройки. При нажатии на соответствующую кнопку будет открываться модальное окно. Для каждого инструмента будет спроектировано свое окно. Заккрытие любого модального окна происходит по нажатию на любую точку области вокруг окна. Рабочая область во время открытия модального окна становится неактивной.

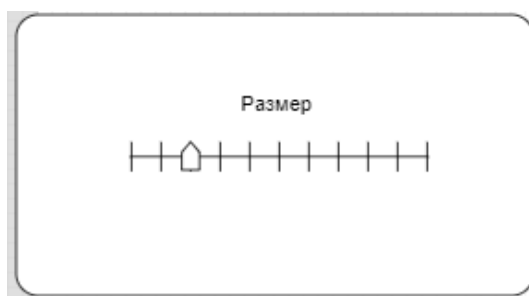
Первое проектируемое окно — это окно загрузки изображения, оно изображено на рисунке 9. В нём будет несколько вариантов загрузки. Пользователь сможет выбрать наиболее удобный для себя в конкретной ситуации. Первый способ стандартный — загрузка через выбор файла в окне Проводника своей операционной системы. Второй способ «Drag and Drop» — загрузка с помощью перетаскивания нужного файла в специально выделенное поле.



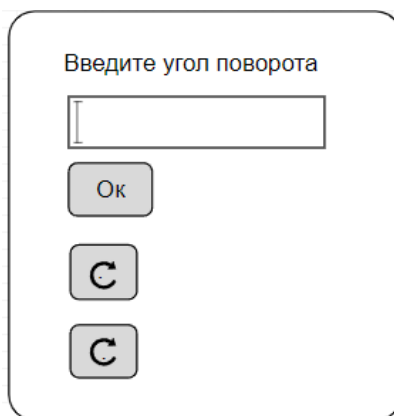
Следующее окно — окно настроек кисти, его вид представлен на рисунке 10. В нем будет расположен бегунок, позволяющий изменить размер кисти, и область палитры, в которой посредством передвижения указателей щелчком мыши или перетаскиванием можно будет выбрать нужный пользователю оттенок. Для подтверждения смены цвета нужно будет нажать кнопку «Поменять цвет».



Модальное окно для настройки ластика будет иметь только одну настройку — размер. Оно представлено на рисунке 11.



В окно с возможностью поворота изображения будет входить поле для ввода целочисленного значения угла поворота, кнопка для подтверждения ввода и две кнопки для быстрого поворота на 90° по часовой стрелке и против часовой, что представлено на рисунке 12.

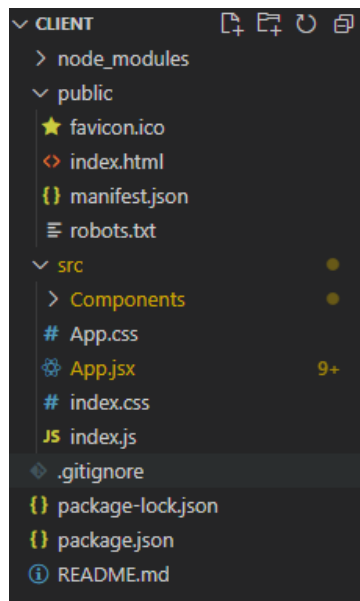


На этом завершается этап создания макета данного программного продукта.

Неотъемлемой частью разработки интерфейса является выбор цветовой палитры, в которой он будет исполнен. Основным цветом был выбран #8A2BE2 (оттенок фиолетового). Он не слишком яркий и не слишком тусклый. Цвет фона рабочей области должен быть нейтральным, не перетягивающим внимание на себя. Таким образом, был выбран цвет #EDEEF7 (оттенок серого). Текст должен быть хорошо читаем на фоне основного цвета интерфейса, для этого идеально подходит чисто белый цвет.

2.2. Кодирование клиентского интерфейса

Этап непосредственного написания кода клиентской части продукта начинается с создания проекта и установки необходимых зависимостей, описанных в главе 1, с помощью `npm`. Базовый проект был создан с применением инструмента `create-react-app`. Структура проекта клиентской части отображена на рисунке 13.



Все установленные пакеты хранятся в папке `node_modules`. В папке `public` хранятся статические файлы, связанные со страницей, на которой будет отрисовываться наш интерфейс и сам файл этой страницы — `index.html`. Файл `favicon.ico` является иконкой, которая будет отображаться на вкладке браузера при загрузке страницы. Она представлена на рисунке 14. Иконка создана под стиль графического редактора, с использованием фирменных цветов и первой буквы названия.



Файл `manifest.json` — это конфигурационный файл, который содержит метаданные в соответствии с критериями Прогрессивных Веб Приложений

(PWA — Progressive Web Apps) [15]. И последний файл в данной папке — это robots.txt, который отвечает за предоставление поисковым системам информации о том, в какие разделы поисковой робот может заходить, а в какие нет [16].

Рассмотрим индексную начальную страницу приложения index.html на рисунке 15.

```
public > < index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="utf-8" />
6    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
7    <meta name="viewport" content="width=device-width, initial-scale=1" />
8    <meta name="theme-color" content="#000000" />
9    <meta name="description" content="Онлайн-фоторедактор от студентов РГПУ им. А. И. Герцена" />
10   <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
11   <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
12   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css" />
13   <title>Палитра – онлайн-фоторедактор</title>
14 </head>
15
16 <body>
17   <noscript>Вам нужно включить JavaScript, чтобы запустить это приложение.</noscript>
18   <div id="root"></div>
19 </body>
20
21 </html>
```

Здесь нас интересуют строки 13, 17, 18.

- На строке 13 находится html-тэг заголовка вкладки.
- На строке 17 находится тэг noscript, который выводит поясняющий текст, если у пользователя не поддерживается или отключена работа JavaScript.
- На строке 18 находится тэг div с id="root". Именно в это место будет помещаться наш интерфейс.

Итоговое содержание зависимостей и скриптов находится в файле package.json, который хранит всю основную информацию о проекте, представлено на рисунке 16.

```

"name": "front-diplom",
"version": "0.1.0",
"private": true,
"dependencies": {
  "@trendmicro/react-sidenav": "^0.5.0",
  "axios": "^0.24.0",
  "bootstrap": "^5.1.3",
  "bootstrap-icons": "^1.8.0",
  "react": "^17.0.2",
  "react-colorful": "^5.5.1",
  "react-dom": "^17.0.2",
  "react-router-dom": "^6.2.2",
  "react-scripts": "4.0.3",
  "reactstrap": "^9.0.0",
  "web-vitals": "^1.1.2"
},
"scripts": {
  "start": "react-scripts start",
  "build": "react-scripts build",
  "test": "react-scripts test",
  "eject": "react-scripts eject"
}

```

Самая главная папка проекта — src. В ней находятся такие файлы, как:

- App.jsx — содержит главный компонент приложения.
- App.css — содержит стили компонента App.jsx.
- index.js — содержит стартовую точку приложения.
- index.css — хранит базовые стили приложения.

Также была создана папа Components, в которой хранятся все специально написанные компоненты, которые будут использоваться в интерфейсе. Всего для проекта разработано три компонента:

- MyCanvas.jsx — основное рабочее пространство, на котором будет отрисовываться загружаемое изображение, и на котором будут происходить манипуляции с этим изображением.
- MySideNav.jsx — боковая навигация по инструментам и функциям графического редактора.
- Modal.jsx — модальное окно для отображения расширенных параметров инструментов и функций.

Для каждого компонента существует свой файл с каскадными стилями: MyCanvas.module.css, MySideNav.module.css и modal.css.

Модальное окно представляет собой два блока `div`, вложенных друг в друга. Каждый из них имеет несколько состояний, которые передаются внутрь компонента. Внешний блок отвечает за перекрытие всей рабочей области и её затемнение. Это необходимо для направления фокуса пользователя на окно с параметрами, а также для возможности закрывать появившееся окно по нажатию на затемненную область.

Открытие и закрытие модального окна реализовано с помощью состояний React и отслеживания событий. В компонент передается:

- переменная `active`, которая хранит состояние `true` или `false`, в зависимости от которого будет применен тот или иной класс `css` (строки 7 и 11 на рисунке 17);
- функция `setActive`, с помощью которой изменяется состояние переменной `active`;
- переменная `children`. В ней содержатся все элементы, которые находятся между тэгами данного компонента.

В событие `onClick` на строке 8 помещается функция `setActive`, которая меняет состояние на `false`, что означает закрытие модального окна. Но так как внутри находится блок с контентом, необходимо вылавливать событие `onClick`, когда нажатие происходит именно в этом блоке.

```
1  import React from 'react'
2  import './modal.css';
3
4  const Modal = ({active, setActive, children}) => {
5    return (
6      <div
7        className={active ? "modal active" : "modal"}
8        onClick={() => setActive(false)}
9      >
10        <div
11          className={active ? "modal__content active" : "modal__content"}
12          onClick={e => e.stopPropagation()}
13        >
14          {children}
15        </div>
16      </div>
17    );
18  };
19
20  export default Modal
```

Для этого существует метод `stopPropagation`, который прекращает движение события по фазам [17]. Последний уровень вложенности, это переменная `children`, с помощью которой помещается нужное содержимое окна.

В стилях `modal.css` прописаны свойства обоих блоков для каждого из двух состояний. Блок с затемнением должен быть во весь экран, поэтому применяются свойства `height: 100v;` и `width: 100vw;`. Сам эффект затемнения создается за счет заливки фона черным цветом с непрозрачностью в 20%. Каждое модальное окно должно появляться по центру экрана, поэтому применяются свойства:

- `display: flex`
- `align-items: center;`
- `justify-content: center;`

Для того чтобы все созданные окна не отображались до момента их активации, добавляется свойство прозрачности `opacity` со значением 0. Свойство `transition: 0.5s` управляет скоростью появления элементов, тем самым добавляя плавности. `Pointer-events` отвечает за управление событиями курсора. В скрытом режиме это свойство имеет значение `none`, запрещая какие-либо события. При активации элемента происходит переключение на подкласс `.modal.active`, в котором выставляется значение прозрачности 1 и разрешаются все события курсора, т. е. все кнопки становятся активными.

Блоку с контентом применяются свойства минимального размера, с целью сохранения формы при уменьшении размера экрана, применяется выбранный в первом параграфе основной фоновый цвет и закругляются углы свойством `border-radius`. Закругления позволяют сделать дизайн более органичным [18], дружелюбным. Также, как и во внешнем блоке, здесь используются `flex`-свойства для центрирования внутренних элементов и свойства, создающие небольшую анимацию при появлении:

- `transform: scale(0.5);` — масштабирование элемента
- `transition: 0.4s all;` — скорость появления, которая применяется ко всем дочерним элементам

При активации происходит переключение на подкласс `.modal__content.active`, в котором свойство `transform` принимает значение `scale(1)`, приводя элементы к изначально заданным размерам.

Следующий компонент отвечает за меню инструментов — `MySideNav.jsx`, представлен на рисунке 18. В нём применялся готовый компонент навигации из зависимости `@trendmicro/react-sidenav`. К нему было добавлено отслеживание состояния (булева переменная `isVisible` и функция изменения состояния `setVisible`), для того чтобы меню могло раскрываться, и возможность передавать внутрь дочерние элементы.

```
import "@trendmicro/react-sidenav/dist/react-sidenav.css";
import React, { useState } from 'react';
import classes from './MySideNav.module.css';
import SideNav from "@trendmicro/react-sidenav";

export const MySideNav = ({children}) => {
  const [isVisible, setVisible] = useState(true);

  return (
    <SideNav className={classes.sideNav}>
      <SideNav.Toggle
        onClick={() => {
          setVisible(!isVisible);
        }}
      />
      <SideNav.Nav>
        {children}
      </SideNav.Nav>
    </SideNav>
  );
}

export default MySideNav;
```

К примененным стилям относится основной цвет и фиксированное расположение меню. Так, при прокрутке страницы, меню всегда будет находиться удобно в зоне досягаемости.

Самый главный и основной компонент интерфейса — это `MyCanvas.jsx` — холст, на котором будет происходить работа с изображением, и меню с

инструментами. Условно его можно разделить на две части: функциональную и то, что будет отрисовано на странице.

Разработка началась с создания холста, код представлен на рисунке 19. Было решено, что сервис будет работать с двумя слоями: фоновым и верхним, слоем для рисования. Таким образом можно рисовать кистью или с помощью фигур, не портя исходное изображение, так как все рисунки легко стираются ластиком.

```
<div className="paint-canvas">
  <canvas id='bg' className={classes.cnvs_bg}></canvas>
  <canvas id='fg' className={classes.cnvs_fg}></canvas>
  <canvas id='readyimg' className={classes.cnvs_save}></canvas>
</div>
```

Каждый слой является отдельным элементом canvas. Третий холст необходим для сохранения итогового изображения. На нем происходит объединение двух созданных слоёв. Следующим шагом было добавлено меню и кнопки для вызова инструментов и функций. Пример того, как выглядит код кнопки на рисунке 20.

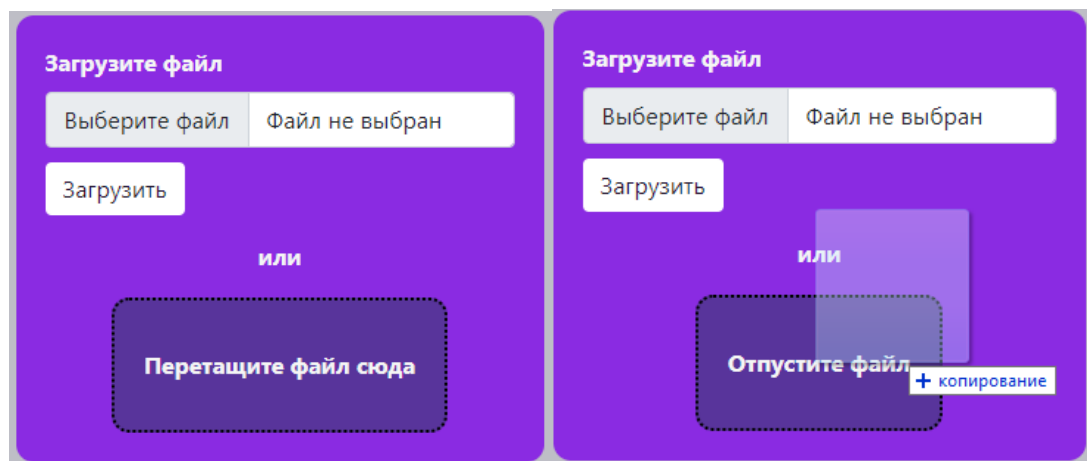
```
<NavItem className="navItem" onClick = {( ) => setLoadModalActive(true)}>
  <NavIcon><img className='icon1' alt='icon1' src={iconLoad}></img></NavIcon>
  <NavText>Загрузить</NavText>
</NavItem>
```

Кнопка представляет собой элемент навигации, к которому прикреплён слушатель события onClick. Он в свою очередь запускает работу соответствующей функции. У кнопки загрузки изображения это функция активации модального окна. Внутри этого элемента находится блок с иконкой и блок с заголовком, который видно только при раскрытии бокового меню.

Модальное окно загрузки изображения, в соответствии с макетом состоит из формы загрузки файла и поля, в которое загружаемый файл можно перетащить. На форме висит слушатель события onSubmit, которое срабатывает при нажатии пользователем кнопки, относящейся к этой форме,

то есть при подтверждении отправки выбранного изображения. Сначала, при выборе файла запускается асинхронная функция, которая конвертирует изображение в понятный для сервера формат. После этого запускается асинхронная функция, которая уже отправляет результат работы предыдущей функции непосредственно на сервер. Далее клиент обращается к серверу для отрисовки полученного изображения на фоновом холсте. На последнем шаге используется такой объект JavaScript как Promise (промис). Промис — это объект, замещающий значение, которое станет доступным в какой-то момент в будущем. Это позволяет другому коду, использующему этот промис, подготовиться к обработке ожидаемого значения [19].

Поле Drag and Drop реализовано с помощью тернарного оператора React и слушателей событий. Тернарный оператор определяет, какой текст будет показываться пользователю при определенных условиях. В данном случае условием является наличие над полем файла или его отсутствие, о котором сигнализирует состояние булевой переменной `drag`. Когда файла нет, поле подсказывает пользователю, что в него нужно перетащить файл (рисунок 21, а), а когда файл появляется, подсказка сменяется на “Отпустите файл” (рисунок 21, б). После этого происходят аналогичные действия с конвертацией картинки и отправкой её на сервер. По окончании загрузки, для удобства пользователя окно автоматически закрывается.

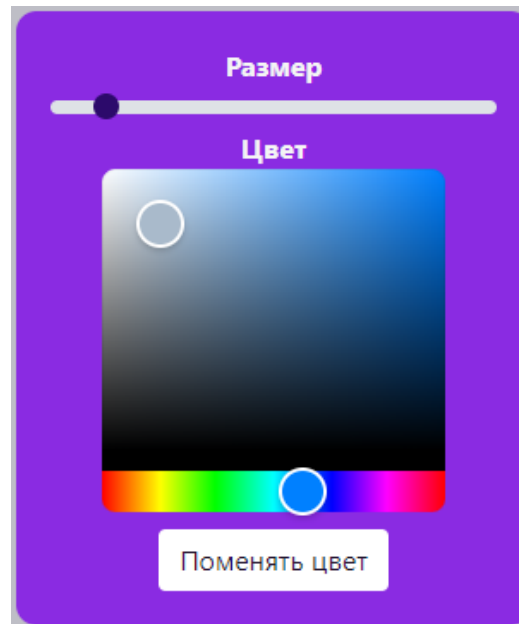


Сохранение изображение на компьютер происходит по нажатию соответствующей кнопки в меню. Она запускает функцию `saveImgBtn`, которая совмещает два изображения на третьем невидимом холсте и создает ссылку, которая в свою очередь автоматически активируется, и браузер сохраняет изображение по этой ссылке.

После кнопки сохранения в меню идет кнопка первого инструмента — кисть. Данный инструмент реализован с помощью трёх обработчиков событий: когда кнопка мыши была нажата (событие `mousedown`), когда она удерживалась (`mouseover`), и когда она была отпущена (`mouseup`). С каждым из событий в массив записываются данные с координатами курсора. В момент, когда клавиша мыши нажимается, запускается процесс рисования при обращении к методу контекста холста `beginPath()`, после чего перо перемещается из своей изначальной точки $[0, 0]$ в точку, в которой произошло нажатие. При этом используется метод `moveTo()` [20]. При удержании клавиши мыши применяется метод `stroke()`, который отрисовывает линию по заданным координатам.

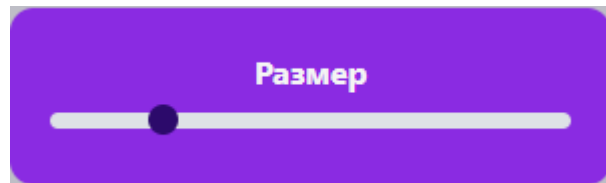
Все три обработчика прикрепляются к соответствующим слушателям событий в момент нажатия кнопки в меню инструментов. Также во время нажатия на кнопку в меню свойству контекста холста `globalCompositeOperation` присваивается значение `"source-over"`. Данное свойство отвечает за режимы наложения графики. Для кисти должно сохраняться значение по умолчанию — `"source-over"`. При этом выбранном свойстве новая фигура накладывается на старые и перекрывает их [21].

Выбор данного инструмента в меню также вызывает открытие модального окна с настройкой его параметров: цвета и толщины, как показано на рисунке 22.



Настройка размера кисти представлена в виде формы типа «Ползунок» (`type="range"`). В ней при возникновении события `onChange`, т. е. при изменении положения ползунка происходит изменение состояния переменной, хранящей размер кисти (`brushVal`), и присвоение этого значения свойству контекста холста, отвечающей за толщину линии — `lineWidth`. Начальное значение размера задано как «10». Настройка цвета кисти осуществляется посредством использования компонента `HexColorPicker` из зависимости `react-colorful`. Он также изменяет переменную состояния, только заданной для хранения цвета (`color`). Но для того, чтобы выбранный цвет был применен к контексту холста, необходимо подтвердить выбор нажатием кнопки «Применить цвет». В контексте холста за цвет отвечает свойство `strokeStyle`. Начальное значение цвета задано как «#aabbcc».

Инструмент Ластик использует те же обработчики событий, что и кисть, но при его активации свойство `globalCompositeOperation` принимает значение `"destination-out"`, что значит, что отображается часть фигуры, на которую не накладывается новая. Новая фигура не выводится. При выборе данного инструмента в меню активируется модальное окно с настройкой размера ластика, как на рисунке 23. Она реализована аналогично настройке размера кисти, но изменяет другую переменную состояния — `eraseVal`.



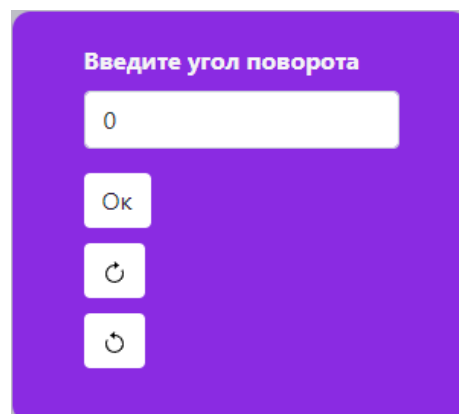
Инструмент Прямоугольник после его выбора в меню считывает координаты двух нажатых точек, вычисляет ширину и длину прямоугольника. Потом, используя метод контекста холста `strokeRect()`, происходит отрисовка контура вычисленной фигуры на холсте. В конце массив точек очищается и нужно активировать инструмент заново.

Инструмент Треугольник рисует треугольник по трём выбранным точкам, которые соответствуют расположению трёх углов, используя стандартные методы canvas: `beginPath()`, `moveTo()`, `lineTo()`, `closePath()`, `stroke()`.

Инструмент Круг рисует фигуру круга по двум точкам: первая точка — это центр круга, вторая точка — это размер радиуса круга. За отрисовку круга на холсте отвечает метод `arc()`.

Все инструменты для рисования фигур за раз позволяют добавить только одну фигуру. Цвет контура фигуры и его толщина зависят от выбранных параметров у кисти или ластика.

Инструмент Поворот при выборе в меню активирует небольшое модальное окно, как на рисунке 24, соответствующее по своему содержанию макету из первого параграфа.



Форма для ввода угла поворота изменяет переменную состояния. При нажатии кнопки «Ок» происходит подтверждение ввода и отправка запроса о повороте изображения угол, равный введенному значению, на сервер. Две кнопки со стрелками ниже являются дополнительными и нужны для быстрого совершения популярного действия. Они поворачивают изображение на 90 градусов по часовой и против часовой стрелки.

Таким образом, разработанный клиентский интерфейс решает задачи, поставленные в первой главе третьем параграфе данной выпускной квалификационной работы.

Выводы по главе 2:

1. Создан макет интерфейса в соответствии техническим заданием и обозначенными функциями.
2. Разработан код клиентского интерфейса в соответствии с макетом, который решает поставленные задачи.

ЗАКЛЮЧЕНИЕ

Цель разработать клиентской интерфейс онлайн-микросервиса по редактированию изображений на основе импортозамещенных веб-технологий, которая была поставлена в данной выпускной квалификационной работе, была выполнена.

В процессе выполнения данной работы были решены следующие задачи:

1. Осуществлен анализ существующих принципов разработки клиентских интерфейсов и современных трендов в дизайне. В результате анализа было определено, что для данного программного продукта наиболее подходящим будет правило контроля пользователем интерфейса и последовательность в опыте взаимодействия с данным интерфейсом. В дизайне интерфейса будут применяться принципы минимализма.
2. Проведен анализ существующих зарубежных аналогов, сделан вывод о применении определенных паттернов построения интерфейсов, использовании преимущественно тёмной цветовой гаммы и большого числа функций и инструментов.
3. Определены инструменты и функции, которые должны быть реализованы на клиентском интерфейсе. К функциям относятся: отрисовка изображения на рабочей области, возможность сохранить изображение на компьютер, отправка запросов на сервер для осуществления загрузки изображения, поворота. К инструментам относятся: кисть, ластик, фигуры (прямоугольник, треугольник, круг).
4. Выбрана для работы среда для написания кода Visual Studio Code, язык программирования JavaScript, библиотеки React.js и Reactstrap, а также дополнительные зависимости.

5. Создан макет будущего клиентского интерфейса на основе требований и заявленного функционала.
6. Разработан код клиентского интерфейса онлайн-микросервиса по редактированию изображений на основе импортозамещенных веб-технологий.

СПИСОК ЛИТЕРАТУРЫ

1. Купер А. Алан Купер об интерфейсе. Основы проектирования взаимодействия / Купер А., Рейман Р., Кронин Д. – Пер. с англ. – Санкт-Петербург: Символ-Плюс, 2009. – 688 с.
2. Сергеев С. Ф. Введение в проектирование интеллектуальных интерфейсов: Учебное пособие / Сергеев С. Ф., Падерно П. И., Назаренко Н.А. – Санкт-Петербург: СПбГУ ИТМО, 2011. – 108 с.
3. Основы HTML - Изучение веб-разработки | MDN – URL: https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/HTML_basics (дата обращения: 13.03.2022) – Режим доступа: свободный.
4. Принципы и правила проектирования пользовательского интерфейса / В. В. Астафьева. – Текст: непосредственный // Молодой ученый. – 2020. – № 1 (291). – С. 10-14. – URL: <https://moluch.ru/archive/291/66058/> (дата обращения: 05.04.2022) – Режим доступа: свободный.
5. Руководство по Modern Minimal стилю UI дизайна – UX PUB – URL: <https://ux.pub/editorial/rukovodstvo-po-modern-minimal-stiliu-ui-dizaina-4m2h> (дата обращения: 05.04.2022) – Режим доступа: свободный.
6. Тиленс Т. М. React в действии / Тиленс Т. М. – Санкт-Петербург: Питер, 2018. – 368 с.
7. Что такое React и как его освоить? — Академия Яндекса – URL: <https://academy.yandex.ru/posts/chto-takoe-react-i-kak-ego-osvoit> (дата обращения: 07.04.2022) – Режим доступа: свободный.

8. React Usage Statistics – URL: <https://trends.builtwith.com/javascript/React> (дата обращения: 09.04.2022) – Режим доступа: свободный.
9. Infographic: React.js stats for 2020 – URL: <https://www.thinslices.com/blog/infographic-react.js-statistics> (дата обращения: 09.04.2022) – Режим доступа: свободный.
10. Usage Statistics and Market Share of React for Websites, May 2022 – URL: <https://w3techs.com/technologies/details/js-react> (дата обращения: 09.04.2022) – Режим доступа: свободный.
11. Янг А. Node.js в действии. 2-е издание / Янг А., Мек Б., Кантелон М. – Санкт-Петербург: Питер, 2018. – 432 с.
12. Бэнкс А. React: современные шаблоны для разработки приложений 2-е издание / Бэнкс А., Порселло Е. – Санкт-Петербург: Питер, 2021. – 320 с.
13. ГОСТ Р 51904-2002. Программное обеспечение встроенных систем. Общие требования к разработке и документированию // Электронный фонд правовых и нормативно-технических документов – URL: <https://docs.cntd.ru/document/1200030195> (дата обращения: 25.04.2022) – Режим доступа: свободный.
14. ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению // Электронный фонд правовых и нормативно-технических документов – URL: <https://docs.cntd.ru/document/1200007648> (дата обращения: 25.04.2022) – Режим доступа: свободный.
15. Chiarelli A. Beginning React: Simplify Your Frontend Development Workflow and Enhance the User Experience of Your Applications with React / Chiarelli A. – Великобритания: Packt Publishing, 2018. – 96 с.
16. Новицкий И. Интернет-маркетинг. Продвижение CEO / Новицкий И. – Россия: ЛитРес, 2019. – 180 с.

17. Чиннатхамби К. JavaScript с нуля / Чиннатхамби К. – Санкт-Петербург: Питер, 2022. – 400 с.
18. Берд Дж. Веб-дизайн. Руководство разработчика / Берд Дж. – Санкт-Петербург: Питер, 2012. – 224 с.
19. Ришкуция В. Програмируй & типизируй. / Ришкуция В. – Санкт-Петербург: Питер, 2021. – 352 с.
20. Дронов В. А. JavaScript. 20 уроков для начинающих / Дронов В. А. – Санкт-Петербург: БХВ-Петербург, 2020. – 352 с.
21. Прохоренок Н. А. HTML, JavaScript, PHP и MySQL.
Джентльменский набор Web-мастера, 5 издание / Прохоренок Н. А.,
Дронов В. А. – Санкт-Петербург: БХВ-Петербург, 2019. – 912 с.