



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ  
ТЕХНОЛОГИЙ»

Факультет среднего профессионального образования  
Цикловая комиссия информационных технологий

Специальность 09.02.07 «Информационные системы и программирование»  
(шифр, наименование специальности)

**Отчет**  
**по учебной практике**  
**УП 03.01 МДК 03.01 «Проектирование, разработка и оптимизация веб-приложений»**

Выполнил обучающийся группы ИСП-195

Рубцов И.Е

(ф.и.о.)

(подпись)

Проверил:  
преподаватель

(должность руководителя от организации)

Бавыкина Е. Г.

(ф.и.о.)

(оценка)

(подпись)

М.П.

(дата)

Воронеж - 2022 г.

## Характеристика на обучающегося по освоению общих компетенций в период прохождения учебной практики

Рубцов Илья Евгеньевич

Группы ИСП-195 специальности 09.02.07 «Информационные системы и программирование»

Обучающийся прошёл(-ла) учебную практику на ФСПО ВГУИТ  
(наименование предприятия)

с «03» ноября 2022г. по «16» ноября 2022 г.

Место работы ВГУИТ ФСПО должность \_\_\_\_\_

За время пребывания на практике проявил(-а) себя следующим образом:

Отношение к производственной работе: \_\_\_\_\_ ответственное \_\_\_\_\_

Степень выполнения программы практики: \_\_\_\_\_ в полном объёме \_\_\_\_\_

Выполнение индивидуального задания: \_\_\_\_\_ выполнено \_\_\_\_\_

Трудовая дисциплина и поведение на рабочем месте: \_\_\_\_\_ без нареканий \_\_\_\_\_

Соответствие теоретической подготовки требованиям к специалисту СЗ: соответствует

Освоенные общие компетенции (указать освоенные компетенции из представленного во ФГОС перечня ОК):

- ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
- ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.
- ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.
- ОК 4. Планировать и реализовывать собственное профессиональное и личностное развитие.
- ОК 5. Планировать и реализовывать собственное профессиональное и личностное развитие.
- ОК 6. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей
- ОК 7. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.
- ОК 8. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
- ОК 9. Использовать информационные технологии в профессиональной деятельности.
- ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке
- ОК 11. Планировать предпринимательскую деятельность в профессиональной сфере

Общие замечания и предложения, пожелания:

Результат практики заслуживает оценку: \_\_\_\_\_

Дата « 16 » ноября 2022 г.

Подпись руководителя практики \_\_\_\_\_  
/ФИО, должность/

Подпись ответственного лица организации \_\_\_\_\_

\_\_\_\_\_  
/ФИО, должность/

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение высшего  
образования

ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ  
ТЕХНОЛОГИЙ

Аттестационный лист  
по учебной практике обучающегося

Рубцов Илья Евгеньевич  
(ФИО)

обучающегося (-йся) на курсе 4 по специальности СПО

09.02.07 «Информационные системы и программирование»

код и наименование

успешно прошёл учебную практику-по профессиональному модулю  
ПМ 03.01 Проектирование и разработка информационных систем

в объеме      часов с «03» ноября по «16» ноября 2022 г.

В процессе прохождения учебной практики были освоены следующие  
профессиональные компетенции:

ПК 9.1. Разрабатывать техническое задание на веб-приложение в  
соответствии с требованиями заказчика.

ПК 9.2. Разрабатывать веб-приложение в соответствии с техническим  
заданием.

ПК 9.3. Разрабатывать интерфейс пользователя веб-приложений в  
соответствии с техническим заданием.

ПК 9.4. Осуществлять техническое сопровождение и восстановление  
веб-приложений в соответствии с техническим заданием.

Адрес деканата: 394036, г. Воронеж, Ленинский проспект пр-т, 14;

ВГУИТ, факультет среднего профессионального образования;

телефон (8-473)-249-93-79, факс (8-473)-249-93-79.

наименование организации, юридический адрес

Наименование Профессионального модуля	Виды работ	Освоенные компетенции	№ задания по ФОС	Качество выполнения работ в соответствии с требованиями организации
УП.01.01	Осуществлять сбор предварительных данных для выявления требований к веб-приложению. Определять первоначальные требования заказчика к веб-приложению и возможности их реализации. Подбирать оптимальные варианты реализации задач и согласование их с заказчиком. Оформлять техническое задание	ПК 9.1 ПК 9.2	Блок 1	выполнено
	Выполнять верстку страниц веб-приложений. Использовать специальные готовые технические решения при разработке веб-приложений. Разрабатывать интерфейс пользователя. Разрабатывать анимационные эффекты	ПК 9.3	Блок 2	выполнено
	Кодировать на языках веб-программирования. Использовать инструментальные средства контроля версий и баз данных. Проводить работы по резервному копированию веб-приложений. Выполнять регистрацию и обработку запросов Заказчика в службе технической поддержки.	ПК 9.4	Блок 3	выполнено

## Содержание

<u>1 Общая характеристика практики.....</u>	<u>6</u>
<u>2 Правила техники безопасности.....</u>	<u>8</u>
<u>3 Виды работ по УП 03.01 МДК 03.01 «Проектирование и разработка информационных систем».....</u>	<u>14</u>
<u>3.1 Сбор предварительных данных для выявления требований к веб-приложению.....</u>	<u>14</u>
<u>3.2 Определение требований заказчика к веб-приложению и возможность их реализации.....</u>	<u>16</u>
<u>3.3 Подбор оптимальных вариантов реализации задач и согласование их с заказчиком.....</u>	<u>18</u>
<u>3.4 Оформление технического задания.....</u>	<u>18</u>
<u>3.5 Индивидуальное задание.....</u>	<u>20</u>
<u>3.6 Вёрстка страниц веб-приложений.....</u>	<u>20</u>
<u>3.7 Использование готовых технических решений при разработке веб-приложений.....</u>	<u>22</u>
<u>3.8 Разработка интерфейса пользователя.....</u>	<u>22</u>
<u>3.9 Разработка анимационных эффектов.....</u>	<u>26</u>
<u>3.10 Индивидуальное задание.....</u>	<u>27</u>
<u>3.11 Кодирование на языках программирования.....</u>	<u>29</u>
<u>3.12 Написание скрипта для валидации.....</u>	<u>30</u>
<u>3.13 Разработка игра на JavaScript.....</u>	<u>34</u>
<u>3.14 Использование инструментальных средств контроля версий и баз данных.....</u>	<u>38</u>
<u>3.15 Проведение работ по резервному копированию веб-приложений.....</u>	<u>38</u>
<u>3.15 Регистрация и обработка запросов заказчика в службе технической поддержки.....</u>	<u>39</u>
<u>3.15 Индивидуальное задание.....</u>	<u>39</u>
<u>Вывод.....</u>	<u>40</u>

## 1 Общая характеристика практики

Целями освоения учебной практики является подготовка выпускника к выполнению и решению профессиональных задач в области проектирования, разработке и оптимизации веб-приложений.

Задачи учебной практики направлены на приобретение первоначального практического опыта в:

- использовании специальных готовых технических решений при разработке веб-приложений;
- выполнении разработки и проектирования информационных систем;
- модернизации веб-приложений с учетом правил и норм подготовки информации для поисковых систем;
- реализации мероприятий по продвижению веб-приложений в сети Интернет.

Процесс прохождения практики направлен на освоение следующих компетенций в соответствии с ФГОС СПО по данной специальности:

а) общих (ОК):

ОК 1 Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам

ОК 2 Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 4 Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 5 Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 6 Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей

ОК 7 Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.

ОК 8 Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности

ОК 9 Использовать информационные технологии в профессиональной деятельности.

ОК 10 Пользоваться профессиональной документацией на государственном и иностранном языке

ОК 11 Планировать предпринимательскую деятельность в профессиональной сфере

б) профессиональных (ПК):

ПК 9.1. Разрабатывать техническое задание на веб-приложение в соответствии с требованиями заказчика.

ПК 9.2. Разрабатывать веб-приложение в соответствии с техническим заданием.

ПК 9.3. Разрабатывать интерфейс пользователя веб-приложений в соответствии с техническим заданием.

ПК 9.4. Осуществлять техническое сопровождение и восстановление веб-приложений в соответствии с техническим заданием.

## 2 Правила техники безопасности

Задание 2.1. Пройти вводный инструктаж на рабочем месте. Пройти инструктаж по технике безопасности.

Типовая инструкция по охране труда для студентов/специалистов ИТ отделов

### 1. ОБЩИЕ ТРЕБОВАНИЯ БЕЗОПАСНОСТИ

1.1. Настоящая инструкция по охране труда студента или специалиста, занятого эксплуатацией персональных электронно-вычислительных машин (ПЭВМ) и видео-дисплейных терминалов (ВДТ), разработана с учетом условий его работы в конкретной организации.

1.2. На студента или специалиста могут воздействовать опасные и вредные производственные факторы:

а) физические:

- повышенные уровни электромагнитного излучения;
- повышенные уровни рентгеновского излучения;
- повышенные уровни ультрафиолетового излучения;
- повышенный уровень инфракрасного излучения;
- повышенный уровень статического электричества;
- повышенные уровни запыленности воздуха рабочей зоны;
- повышенное содержание положительных аэроионов в воздухе рабочей зоны;
- пониженное содержание отрицательных аэроионов в воздухе рабочей зоны;
- пониженная или повышенная влажность воздуха рабочей зоны;
- пониженная или повышенная подвижность воздуха рабочей зоны;
- повышенный уровень шума;
- повышенный или пониженный уровень освещенности;
- повышенный уровень прямой блёскости;



- повышенный уровень отраженной блёскости;
- повышенный уровень ослепленности;
- неравномерность распределения яркости в поле зрения;
- повышенная яркость светового изображения;
- повышенный уровень пульсации светового потока;
- повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека;

б) химические:

- повышенное содержание в воздухе рабочей зоны двуокиси углерода, озона, аммиака, фенола, формальдегида и полихлорированных бифенилов;

в) психофизиологические:

- напряжение зрения;
- напряжение внимания;
- интеллектуальные нагрузки;
- эмоциональные нагрузки;
- длительные статические нагрузки;
- монотонность труда;
- большой объем информации, обрабатываемой в единицу времени;
- нерациональная организация рабочего места;

г) биологические:

- повышенное содержание в воздухе рабочей зоны микроорганизмов.

1.3. К работам допускаются:

- лица не моложе 18 лет, прошедшие обязательный при приеме на работу и ежегодные медицинские освидетельствования на предмет пригодности для работы;

- прошедшие вводный инструктаж по охране труда;

- прошедшие обучение безопасным приемам и методам труда по программе, утвержденной руководителем предприятия (работодателем),

разработанной на основе Типовой программы, и прошедшие проверку знаний, в том числе по электробезопасности;

- прошедшие курс обучения на персональном компьютере с использованием конкретного программного обеспечения;

- прошедшие инструктаж по охране труда на конкретном рабочем месте по данной инструкции.

1.4. Студент или специалист должен быть обеспечен СИЗ в соответствии с Межотраслевыми правилами обеспечения работников специальной одеждой, специальной обувью и другими средствами индивидуальной защиты, утвержденными Приказом Минздравсоцразвития России от 01.06.2009 N 290н; выдаваемые работникам средства индивидуальной защиты должны соответствовать характеру и условиям работы и обеспечивать безопасность труда. Не допускаются приобретение и выдача работникам средств индивидуальной защиты без сертификата соответствия.

Характеристика выданных СИЗ (номенклатура, срок выдачи и нормы соответствия) устанавливается из личных карточек работников, занятых на определенном рабочем месте.

Нормативные номенклатура и сроки выдачи СИЗ определяются согласно Типовым отраслевым нормам бесплатной выдачи рабочим и служащим специальной одежды, специальной обуви и других СИЗ.

## 2. ТРЕБОВАНИЯ БЕЗОПАСНОСТИ ПЕРЕД НАЧАЛОМ РАБОТЫ

2.1. Перед началом работы студент или специалист обязан:

- осмотреть и привести в порядок рабочее место;
- отрегулировать освещенность на рабочем месте, убедиться в достаточности освещенности, отсутствии отражений на экране, отсутствии встречного светового потока;

- проверить правильность подключения оборудования в электросеть;

- протереть специальной салфеткой поверхность экрана;

- убедиться в отсутствии дискет в дисководы процессора персонального компьютера;

- проверить правильность установки стола, стула, подставки для ног, пюпитра, положения оборудования, угла наклона экрана, положение клавиатуры и, при необходимости, произвести регулировку рабочего стола и кресла, а также расположение элементов компьютера в соответствии с требованиями эргономики и в целях исключения неудобных поз и длительных напряжений тела.

2.2. При включении компьютера соблюдать правила электробезопасности.

2.3. Студенту или специалисту запрещается приступать к работе при:

- отсутствии на ВДТ гигиенического сертификата, включающего оценку визуальных параметров;

- отсутствии информации о результатах аттестации условий труда на данном рабочем месте или при наличии информации о несоответствии параметров данного оборудования требованиям санитарных норм;

- отключенном заземляющем проводнике защитного фильтра;

- обнаружении неисправности оборудования;

- отсутствии защитного заземления устройств ПЭВМ и ВДТ;

- отсутствии углекислотного или порошкового огнетушителя и аптечки первой помощи;

- нарушении гигиенических норм размещения ВДТ (при однорядном расположении менее 1 м от стен, при расположении рабочих мест в колонну на расстоянии менее 1,5 м, при размещении на площади менее 6 кв. м на одно рабочее место, при рядном размещении дисплеев экранами друг к другу).

### 3. ТРЕБОВАНИЯ БЕЗОПАСНОСТИ ВО ВРЕМЯ РАБОТЫ

3.1. Студент или специалист во время работы обязан:

- выполнять только ту работу, которая ему была поручена и по которой он был проинструктирован;

- в течение всего рабочего дня содержать в порядке и чистоте рабочее место;
- держать открытыми все вентиляционные отверстия устройств;
- при необходимости прекращения работы на некоторое время корректно закрыть все активные задачи;
- выполнять санитарные нормы и соблюдать режимы работы и отдыха;
- соблюдать правила эксплуатации вычислительной техники в соответствии с инструкциями по эксплуатации;
- соблюдать установленные режимом рабочего времени регламентированные перерывы в работе и выполнять в физкультминутках рекомендованные упражнения для глаз, шеи, рук, туловища, ног;
- соблюдать расстояние от глаз до экрана в пределах 60 - 80 см.

3.2. Студенту или специалисту во время работы запрещается: прикасаться к задней панели системного блока при включенном питании; переключать разъемы интерфейсных кабелей периферийных устройств при включенном питании; загромождать верхние панели устройств бумагами и посторонними предметами; допускать захламленность рабочего места бумагой - в целях недопущения накопления органической пыли; производить отключение питания во время выполнения активной задачи; производить частые переключения питания; допускать попадание влаги на поверхность системного блока, монитора, рабочую поверхность клавиатуры, дисководов, принтеров и др. устройств; включать сильно охлажденное (принесенное с улицы в зимнее время) оборудование; производить самостоятельно вскрытие и ремонт оборудования.

#### 4. ТРЕБОВАНИЯ БЕЗОПАСНОСТИ В АВАРИЙНЫХ СИТУАЦИЯХ

##### 4.1. Студент или специалист обязан:

- во всех случаях обнаружения обрыва проводов питания, неисправности заземления и других повреждений электрооборудования, появления запаха гари

немедленно отключить питание и сообщить об аварийной ситуации руководителю и дежурному электрику;

- при обнаружении человека, попавшего под напряжение, немедленно освободить его от действия тока путем отключения электропитания и до прибытия врача оказать потерпевшему первую медицинскую помощь;

- при любых случаях сбоя в работе технического оборудования или программного обеспечения немедленно вызвать представителя инженерно-технической службы эксплуатации вычислительной техники;

- в случае появления рези в глазах, при резком ухудшении видимости - невозможности сфокусировать взгляд или навести его на резкость, появлении боли в пальцах и кистях рук, усилении сердцебиения немедленно покинуть рабочее место, сообщить о происшедшем руководителю работ и обратиться к врачу;

- при возгорании оборудования отключить питание и принять меры к тушению очага пожара при помощи углекислотного или порошкового огнетушителя, вызвать пожарную команду и сообщить о происшествии руководителю работ.

## 5. ТРЕБОВАНИЯ БЕЗОПАСНОСТИ ПОСЛЕ ОКОНЧАНИЯ РАБОТЫ

5.1. По окончании работ студент или специалист обязан соблюдать следующую последовательность выключения вычислительной техники:

- произвести закрытие всех активных задач;
- убедиться, что в разъемах нет внешних накопителей;
- выключить питание системного блока;
- выключить питание всех периферийных устройств;
- отключить блок питания.

5.2. По окончании работ студент или специалист обязан осмотреть и привести в порядок рабочее место, вымыть с мылом руки и лицо.

3 Виды работ по УП 03.01 МДК 03.01 «Проектирование и разработка информационных систем»

3.1 Сбор предварительных данных для выявления требований к веб-приложению

Ниже представлены основные этапы проектирования и разработки веб-ресурса в современном Интернет-агентстве, занесённые в таблицу 1.

Таблица 1 – Основные этапы проектирования и разработки веб-ресурса

№	Этап	Что делается	Кто делает	Что может пойти не так
1	Определение целей разработки	Менеджер обсуждает с заказчиком цели и задачи которые должен выполнять сайт , его функции и, возможно, методы их выполнений	Менеджер со стороны агентства Заказчик	Заказчик запросит невыполнимые требования
2	Составление технической документации, разработка технического задания и требований	Определение технические и тактико-технические характеристик разрабатываемого проекта	Контрактная служба	ТЗ отражает недействительную, ложную информацию о проекте

3	Создание дизайн-макета сайта	Создаются различные макеты сайта , отражающие основной интерфейс и функции, далее утверждается или перерабатывается	Веб-дизайнер	Созданный макет не будет соответствовать необходимому функционалу сайта, или же не будет подходить современным требованиям
4	Верстка, программирование	Техническая работа, разработка кода и скриптов для работы сайта	Отдел программистов	Возникновение программных ошибок
5	Наполнение сайта	заполнение сайта текстом и картинками, а так же создание админ панели для дальнейшего редактирования участия агенства	зависит от договора	Наполнения сайта не будет отражать его действительные функции
6	Запуск сайта в интернете,	Перенос проекта на хостинг  Подключение БД и админ панели	Разработчик	Возникнут сложности с выбором хостинга, БД содержат логические ошибки
7	Тестирование сайта и дальнейшая	Проверка работоспособности сайта, и устранение	Тестировщики и/или	Выполненное тестирование не сможет выявить

	поддержка	неисправностей	программисты	всех проблем работы сайта
--	-----------	----------------	--------------	------------------------------

3.2 Определение требований заказчика к веб-приложению и возможности их реализации

Далее были определены первоначальные требования заказчика к веб-приложению. Полученные результаты были занесены в таблицу 2.

Таблица 2 – Первоначальные требования заказчика к веб-приложению

<b>Общие сведения</b>	
Название	RUBTSOFF
Адрес, телефон, e-mail	Г. Воронеж Фридриха-Энгельса 75/1  89805568495  rubtsoff@gmail.com
Сфера деятельности	Продажа телефонов
Позиционирование на рынке	Стремится к лидирующим позициям в стране
Использование интернета	Активное
Адрес текущего сайта	-
Другие сайты	-
<b>Продукция</b>	
Основные товары/услуги	Смартфоны



Форма реализации	Розница/опт
Продаваемые бренды	Собственный бренд
География реализации	Г. Воронеж
Основные потребители услуг	От 9 до 55 лет
Замечания о продукции	-
<b>Общая характеристика сайта</b>	
Название	RUBTSOFF
Язык	Русский
Что нужно, чтобы увидел пользователь	Каталог, цены
Что нужно, чтобы пользователь сделал	Просмотрел товары, ознакомился с подходящими под себя вариантами
Основная задача	Продажа товаров
Тип-сайта	Интернет магазин
Объём обновлений	В зависимости от выпуска нового типа продукции
<b>Дизайн</b>	
Впечатления от сайта	Простой в использовании и интуитивно понятный
Фирменный стиль	Неоновые градиенты , логотип
Предпочитаемая цветовая гамма	Оранжевый , желтый , черный

Примеры для оформления	Название	Причины
<a href="https://www.cifrus.ru/">https://www.cifrus.ru/</a>	Cifrus	Простая и приятная навигация, хорошая реализация сайта , приятный дизайн

### 3.3 Подбор оптимальных вариантов реализации задач и согласование их с заказчиком

Разработанная схема основных видов веб-сайтов представлена на рисунке 1.

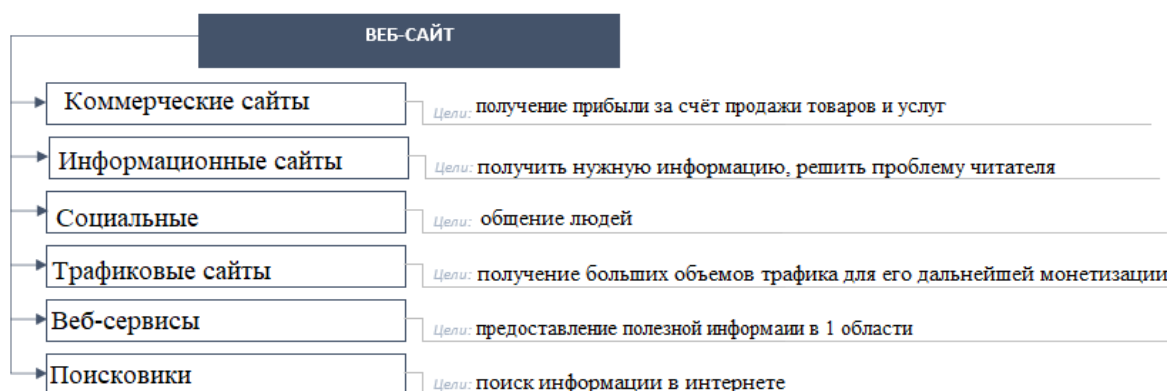


Рисунок 1 - Схема основных видов веб-сайтов

### 3.4 Оформление технического задания

Техническое задание на разработку веб-сайта

1. Имя: [www.rubtsoff.ru](http://www.rubtsoff.ru)
2. Название: Сайт ООО «Рубцофф»
3. Назначение: реализация готовой продукции

4. Язык: русский
5. Основные слова по которым можно произвести поиск сайта в интернете: «rubtsoff», «Купить телефон» «Телефон» и т.п.
6. Возраст аудитории: от 9 лет и старше
7. Возрастное ядро: 12-30 лет
8. Количество страниц сайта: 5+
9. Кнопки управления: определяются веб-дизайнером самостоятельно
10. Объем сайта: не оговаривался
11. Оформление графических материалов: все графические материалы должны быть выполнены в 1 стиле
12. Время загрузки: меньше 15 секунд
13. Основной диапазон разрешения мониторов: от 400р до 4k
14. Минимальное разрешение: 400р
15. Требования к браузеру: IE 5.5 или выше
16. Цветовая палитра: Один цветовой диапазон
17. Общий фон: определяется дизайнером
18. Размер и тип шрифта: не оговаривается
19. Срок разработки: 1 месяц
20. Порядок передачи сайта: удаленно
21. Сопровождение сайта: самостоятельно определяется заказчиком
22. Дополнительные условия: нету

### 3.5 Индивидуальное задание

### 3.6 Вёрстка страниц веб-приложений

Веб-сайты состоят из веб-страниц, которые связаны между собой гиперссылками. Для верстки страниц используется язык HTML, с помощью которого можно описать большое количество элементов. Для оформления элементов и форматирования текстов на странице используется язык CSS. В нем каждому элементу можно задавать правила оформления (CSS).

HTML5 – пятая версия языка для структурирования и представления содержимого веб-сайтов Интернет. С его помощью один и тот же макет можно сверстать множеством способов, однако, если вы хотите, чтобы ваш код был понятен другим разработчикам, необходимо соблюдать правила семантической верстки. Это означает, что с помощью тегов можно разбивать страницу на секции, явно задавать, где меню страницы, а также «шапка» и «подвал». Новые семантические теги языка позволяют определить смысловую нагрузку различных элементов веб-страницы, упростить их индексацию поисковыми системами, улучшить отображение в браузере.

Макет версался в [www.pixso.net](http://www.pixso.net).

Разработанный макет главной страницы сайта отображён на рисунке 2.

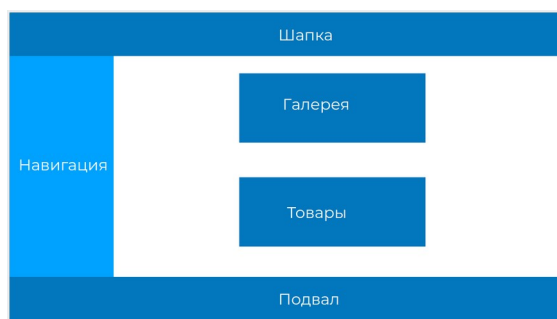


Рисунок 2 – Макет главной страницы сайта

Далее была произведена вёрстка сайта. При вёрстке использовал семантические теги header, nav, ul, il, main, footer, section, article, h1, img и другие. Например, для выделения содержимого контента в целом, был использован тэг section, он обозначает определенную секцию на сайте. Внутри section был применён тэг article, для элементов галереи товаров. Для более корректного отображения страницы были добавлены файлы CSS. Также в процессе вёрстки были использованы flex и grid контейнеры.

Grid элементы использовались для обозначения областей — шапки, боковой панели, основной части сайта и «подвала». Flex контейнеры были использованы для центрирования элементов внутри Grid контейнера.

Стили использовались для указания фонового цвета, и другого.

Использовалась среда разработки — Visual Studio Code.

Полученный результат сверстанного сайта представлен на рисунке 3.



Рисунок 3 – Сверстанный макет сайта с добавлением стилей

### 3.7 Использование готовых технических решений при разработке веб-приложений

Данное разработку была спроектирована и реализована самостоятельно, никаких готовых проектных решений задействовано не было.

### 3.8 Разработка интерфейса пользователя

Макет, на основе которого производилась вёрстка страницы, отражающей каталог магазина представлен на рисунке 4.

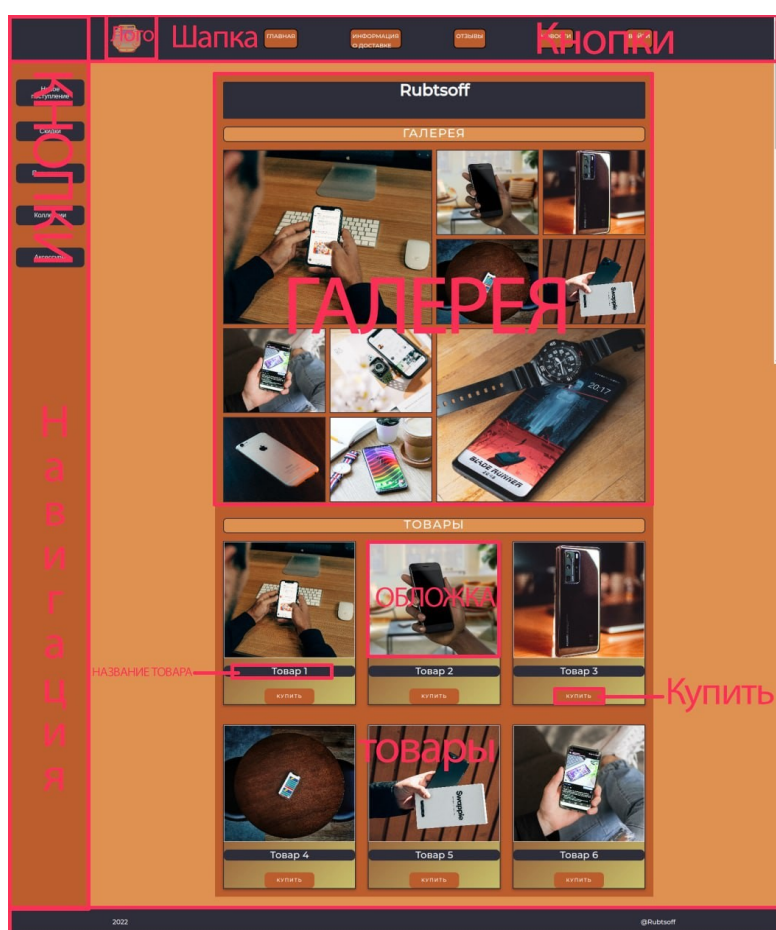


Рисунок 4 – Макет каталога

Макет верстался в графическом редакторе Adobe Photoshop

Результат итоговой верстки главной страницы сайта представлен на рисунке 5.



Рисунок 5 – Главная страница сайта

Для нормальной работоспособности сайта необходимо было разработать страницу покупки. При проектировании этой страницы были использованы теги `label` и `input`, а также кнопки типа `reset` и `submit`. Поле с выбором товара доступно для редактирования, функционал реализован как корзина товаров. Разработанная страница представлена на рисунке 6.

Фамилия

Имя

Отчество

Комментарий

Номер телефона

Почта

Укажите желаемую дату доставки

Первый класс посылки

Второй класс посылки

Адрес доставки

Товар

Товар 2

Товар 3

Товар 4

Товар 5

Товар 6

Выберите позиции который вы хотите заказать

Очистить

Отправить форму

2022

Rubisoff

Рисунок 6 – Форма для отправки данных на сервер

Также для расширения списка устройств, с которых возможна работа с сайтом, была использована адаптивная вёрстка. Страница будет менять свой внешний вид для всех типов устройств. Для создания адаптивности были использованы медиа-запросы. Внешний вид страницы для устройств с мобильным (ширина менее 600 пикселей) разрешением показан на рисунке 7.

Для устройство от 600 до 800 внешний вид показан на рисунке 8.





Рисунок 7- Вид страницы при разрешении экрана менее 600 пикселей



Рисунок 8- Вид страницы при разрешении экрана менее 800 пикселей

Для устройств с маленьким разрешением также было разработано специальное меню для навигации, называемое бургером , представленное на рисунке 9.

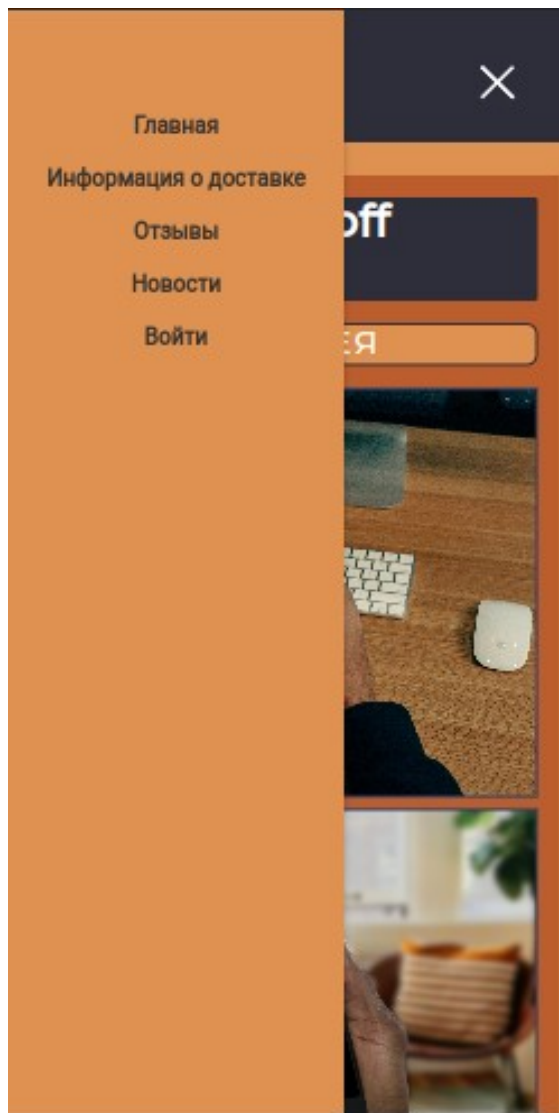


Рисунок 9 – Боковое меню навигации для экрана менее с шириной 800 пикселей

### 3.9 Разработка анимационных эффектов

Для спроектированного сайта были разработаны специальные анимации, которые начинаются на наведение мыши на определённые элементы. Пример на

рисунке 10 . Код на рисунке 11.

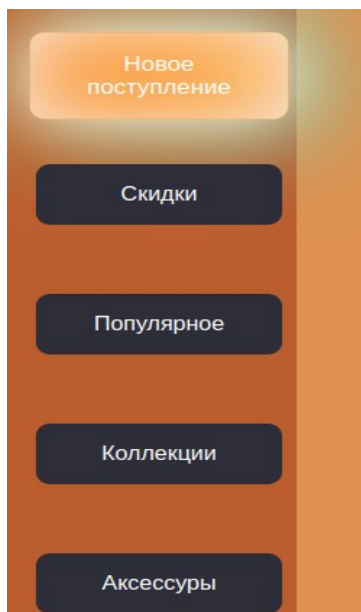


Рисунок 10 - Эффект при наведении на элемент кнопку выпадающего меню

```
.link:hover {  
  transform: scale(1.05);  
  box-shadow:  
    inset 0 0 60px whitesmoke,  
    inset 20px 0 80px #f0f,  
    inset -20px 0 80px #0ff,  
    inset 20px 0 300px #f0f,  
    inset -20px 0 300px #0ff,  
    0 0 50px #fff,  
    -10px 0 80px #f0f,  
    10px 0 80px #0ff;  
  transition: 0.3s;  
  border-radius: 10px;  
}
```

Рисунок 11 - Код для эффекта при наведении на кнопку выпадающего меню

Для "широкого" экрана фото выводится по 3 в ряд, занимая при этом всю ширину (резиновая верстка).

Для узкого экрана – 1 фотография в ряду, растягивается по ширине.

Результат работы представлен на рисунке 11 и 12.



Рисунок 12- Галерея фотографий при ширине экрана более 800 пикселей

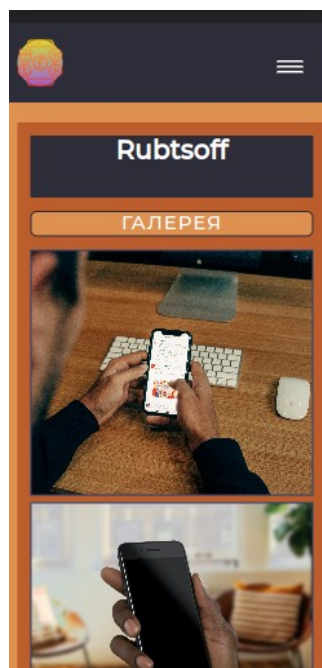


Рисунок 13- Галерея фотографий при ширине экрана более 600 пикселей

### 3.10 Индивидуальное задание

### 3.11 Кодирование на языках программирования

На странице одного из товаров был разработан раскрывающийся список, список его дочерних элементов скрывается или раскрывается. Заголовки, дочерние списки которые скрыты визуально выделяются. Пример выпадающего списка представлен на рисунке 12. Код выпадающего списка представлен на рисунке 13.

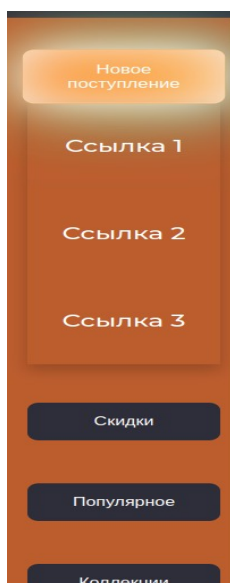


Рисунок 14-Пример выпадающего меню

Для заголовка и элементов списка была добавлена функция обработки события onclick, показанная на рисунке 13.

```

function myFunction1() {
    document.getElementById("myDropdown1").classList.toggle("show");
}
function myFunction2() {
    document.getElementById("myDropdown2").classList.toggle("show");
}
function myFunction3() {
    document.getElementById("myDropdown3").classList.toggle("show");
}
function myFunction4() {
    document.getElementById("myDropdown4").classList.toggle("show");
}
function myFunction5() {
    document.getElementById("myDropdown5").classList.toggle("show");
}

window.onclick = function(event) {
    if (!event.target.matches('.dropbtn')) {
        var dropdowns = document.getElementsByClassName("dropdown-content");
        var i;
        for (i = 0; i < dropdowns.length; i++) {
            var openDropdown = dropdowns[i];
            if (openDropdown.classList.contains('show')) {
                openDropdown.classList.remove('show');
            }
        }
    }
}

```

Рисунок 15-Код для выпадающего меню

### 3.12 Написание скрипта для валидации

Для формы покупки были прописаны специальные скрипты, которые позволяют производить проверку ведёных данных на соответствие принятым стандартам безопасности. Валидация выполняется при нажатии кнопки типа submit. Код скрипта, отвечающего за валидацию формы представлен на рисунке 16, 17 , 18.

```

JS validation.js > form.addEventListener('submit', callback)
You, 3 часа назад | 1 author (You)
let form = document.querySelector('.buy_form')
let validateBtn = form.querySelector('#validateBtn')

form.addEventListener('submit', function (event) {
  event.preventDefault()
  console.log('clicked on validate')
})

form.addEventListener('submit', function (event) {
  event.preventDefault()
  console.log('clicked on validate')
  console.log('from: ', from.value)
  console.log('password: ', password.value)
  console.log('passwordConfirmation: ', passwordConfirmation.value)
  console.log('where: ', where.value)
  console.log('message: ', message.value)
})

let fields = form.querySelectorAll('.field')

form.addEventListener('submit', function (event) {
  event.preventDefault()

  for (let i = 0; i < fields.length; i++) {
    if (!fields[i].value) {
      console.log('field is blank', fields[i])
    }
  }
})

for (let i = 0; i < fields.length; i++) {
  if (!fields[i].value) {
    console.log('field is blank', fields[i])
    let error = document.createElement('div')
    error.className = 'error'
    error.style.color = 'red'
    error.innerHTML = 'Cannot be blank'
    form[i].parentElement.insertBefore(error, fields[i])
  }
}

// удаляю ошибки при начале валидации
form.addEventListener('submit', function (event) {
  event.preventDefault()

  let errors = form.querySelectorAll('.error')

  for (let i = 0; i < errors.length; i++) {
    errors[i].remove()
  }

  for (let i = 0; i < fields.length; i++) {
    if (!fields[i].value) {
      console.log('field is blank', fields[i])
      let error = document.createElement('div')
      error.className = 'error'
      error.style.color = 'red'
      error.innerHTML = 'Cannot be blank'
      form[i].parentElement.insertBefore(error, fields[i])
    }
  }
})

//проверка на ошибки когда не совпадают пароли
form.addEventListener('submit', function (event) {
  event.preventDefault()

  let errors = form.querySelectorAll('.error')

  for (let i = 0; i < errors.length; i++) {
    errors[i].remove()
  }

  for (let i = 0; i < fields.length; i++) {
    if (!fields[i].value) {
      console.log('field is blank', fields[i])
      let error = document.createElement('div')
      error.className = 'error'
      error.style.color = 'red'
      error.innerHTML = 'Cannot be blank'
      form[i].parentElement.insertBefore(error, fields[i])
    }
  }
})

```

Рисунок 16 - Скрипт валидации формы реализованный на JavaScript

```

    if (password.value !== passwordConfirmation.value) {
        console.log('not equals')
        let error = document.createElement('div')
        error.className = 'error'
        error.style.color = 'red'
        error.innerHTML = 'Passwords doesnt match'
        password.parentElement.insertBefore(error, password)
    }
})

//функция которая принимает на вход строку и возвращает в DOM элемент
let generateError = function (text) {
    let error = document.createElement('div')
    error.className = 'error'
    error.style.color = 'red'
    error.innerHTML = text
    return error
}

//переносу в отдельную функцию отчистку ошибок
let removeValidation = function () {
    let errors = form.querySelectorAll('.error')

    for (let i = 0; i < errors.length; i++) {
        errors[i].remove()
    }
}

//вызываем ее
form.addEventListener('submit', function (event) {
    event.preventDefault()

    removeValidation()

    for (let i = 0; i < fields.length; i++) {
        if (!fields[i].value) {
            console.log('field is blank', fields[i])
            let error = generateError('Cant be blank')
            form[i].parentElement.insertBefore(error, fields[i])
        }
    }

    if (password.value !== passwordConfirmation.value) {
        console.log('not equals')
        let error = generateError('Password doesnt match')
        password.parentElement.insertBefore(error, password)
    }
})

//вызову проверку полей на пустоту
let checkFieldsPresence = function () {
    for (let i = 0; i < fields.length; i++) {
        if (!fields[i].value) {
            console.log('field is blank', fields[i])
            let error = generateError('Cant be blank')
            form[i].parentElement.insertBefore(error, fields[i])
        }
    }
}

//вызываем ее
form.addEventListener('submit', function (event) {
    event.preventDefault()

    removeValidation()

    checkFieldsPresence()

    if (password.value !== passwordConfirmation.value) {
        console.log('not equals')
        let error = generateError('Password doesnt match')
        password.parentElement.insertBefore(error, password)
    }
})

//вызову валидация пароля
let checkPasswordMatch = function () {
    if (password.value !== passwordConfirmation.value) {
        console.log('not equals')
        let error = generateError('Password doesnt match')
        console.log(error)
        password.parentElement.insertBefore(error, password)
    }
}

```

Рисунок 17

— Скрипт валидации формы реализованный на JavaScript



```

form.addEventListener('submit', function (event) {
  event.preventDefault()

  removeValidation()

  checkFieldsPresence()

  checkPasswordMatch()
})

```

Рисунок 18 - Скрипт валидации формы реализованный на JavaScript

На рисунке 19 представлен пример валидации формы.

Фамилия

Имя

Отчество

Илья

Евгеньевич

Заполните это поле.

Рисунок 19 - Пример валидации формы на JavaScript

Также была произведена вёрстка формы с галерей фотографий.

В форме регистрации пользователя была добавлена возможность добавления фото, продемонстрированная на рисунке 20.

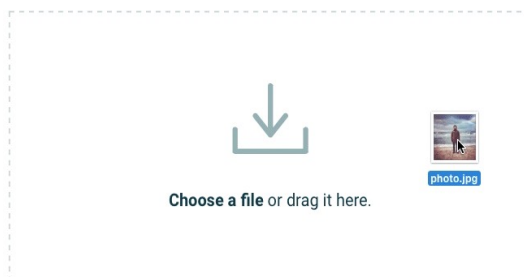


Рисунок 20-Добавление фото

### 3.13Разработка игры на JavaScript

Также в папке сайта была добавлена папка позволяющая начать играть в разработанную мини-игру. Управление в этой игре реализуется посредством клавиш А и D, а так же указателем мыши. Суть игры заключается в том, чтобы отбивать мячи. За каждый сбитый мяч даётся одно очко. За каждый пропущенный мяч отнимается счет,. Нужно отбить как можно больше мячей за определенное время . Результат показан на рисунке 22.

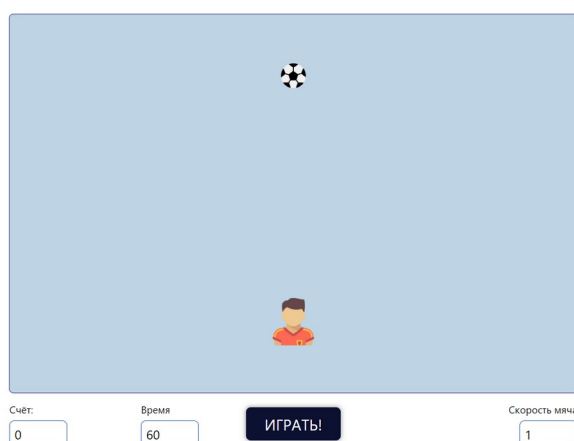


Рисунок 22- Страница с игрой

Код, на основе которого функционирует игра представлен на рисунках

23,

24,

24.

```
JS breakout.js > [x] config > [x] physics > [x] arcade > [x] checkCollision
1  let ball;
2  let paddle;
3  let bricks;
4  let scoreText;
5  let livesText;
6  let startButton;
7  let rotation;
8  let gameOverText;
9  let wonTheGameText;
10
11  let score = 0;
12  let lives = 3;
13
14  const textStyle = {
15    font: 'bold 18px Arial',
16    fill: '#FFF'
17  };
18
19  const config = {
20    type: Phaser.AUTO,
21    width: window.innerWidth,
22    height: window.innerHeight,
23    backgroundColor: '#222',
24    physics: {
25      default: 'arcade',
26      arcade: {
27        // debug: true,
28        checkCollision: {
29          up: true,
30          down: false,
31          left: true,
32          right: true
33        }
34      }
35    },
36    scene: {
37      preload,
38      create,
39      update
40    }
41  };
42
43  const game = new Phaser.Game(config);
44
45  function preload() {
46    this.load.image('paddle', 'img/paddle.png');
47    this.load.image('brick', 'img/brick.png');
48    this.load.image('destroyed', 'img/destroyed.png');
49    this.load.image('ball', 'img/ball.png');
50  }
51
52  function create() {
53    paddle = this.physics.add.image(this.cameras.main.centerX, this.game.config.height - 50, 'paddle')
54    .setImmovable();
55
56    ball = this.physics.add.image(this.cameras.main.centerX, this.game.config.height - 100, 'ball')
57    .setCollideWorldBounds(true)
58    .setBounce(1);
59
60    bricks = this.physics.add.staticGroup({
61      key: 'brick',
62      frameQuantity: 20,
63      gridAlign: { width: 10, cellWidth: 60, cellHeight: 60, x: this.cameras.main.centerX - 277.5, y: 100 }
64    });
65
66    scoreText = this.add.text(20, 20, 'Score: 0', textStyle);
67    livesText = this.add.text(this.game.config.width - 20, 20, 'Lives: '+lives, textStyle).setOrigin(1, 0);
68  }
```

Рисунок 23 - код игры

```

gameOverText = this.add.text(this.cameras.main.centerX, this.cameras.main.centerY, 'Game over!', textStyle)
    .setOrigin(0.5)
    .setPadding(10)
    .setStyle({ backgroundColor: '#111', fill: '#e74c3c' })
    .setVisible(false);

wonTheGameText = this.add.text(this.cameras.main.centerX, this.cameras.main.centerY, 'You won the game!', textStyle)
    .setOrigin(0.5)
    .setPadding(10)
    .setStyle({ backgroundColor: '#111', fill: '#27ae60' })
    .setVisible(false);

startButton = this.add.text(this.cameras.main.centerX, this.cameras.main.centerY, 'Start game', textStyle)
    .setOrigin(0.5)
    .setPadding(10)
    .setStyle({ backgroundColor: '#111' })
    .setInteractive({ useHandCursor: true })
    .on('pointerdown', () => startGame.call(this))
    .on('pointerover', () => startButton.setStyle({ fill: '#f39c12' }))
    .on('pointerout', () => startButton.setStyle({ fill: '#fff' }));

this.physics.add.collider(ball, bricks, brickHit, null, this);
this.physics.add.collider(ball, paddle, paddleHit, null, this);
}

function update() {
    if (rotation) {
        ball.rotation = rotation === 'left' ? ball.rotation - .05 : ball.rotation + .05;
    }

    if (ball.y > paddle.y) {
        lives--;

        if (lives > 0) {
            livesText.setText(`Lives: ${lives}`);

            ball.setPosition(this.cameras.main.centerX, this.game.config.height - 100)
                .setVelocity(300, -150);
        } else {
            ball.destroy();

            gameOverText.setVisible(true);
        }
    }
}

function paddleHit(ball, paddle) {
    var diff = 0;

    if (ball.x < paddle.x) {
        diff = paddle.x - ball.x;
        ball.setVelocityX(-20 * diff);
        rotation = 'left';
    } else if (ball.x > paddle.x) {
        diff = ball.x - paddle.x;
        ball.setVelocityX(20 * diff);
        rotation = 'right';
    } else {
        ball.setVelocityX(2 + Math.random() * 10);
    }
}

```

Рисунок 24- код игры

```

function brickHit(ball, brick) {
    brick.setTexture('destroyed');

    score += 5;
    scoreText.setText(`Score: ${score}`);

    this.tweens.add({
        targets: brick,
        scaleX: 0,
        scaleY: 0,
        ease: 'Power1',
        duration: 500,
        delay: 250,
        angle: 180,
        onComplete: () => {
            brick.destroy();

            if (bricks.countActive() === 0) {
                ball.destroy();

                wonTheGameText.setVisible(true);
            }
        }
    });
}

function startGame() {
    startButton.destroy();
    ball.setVelocity(-300, -150);
    rotation = 'left';

    this.input.on('pointermove', pointer => {
        paddle.x = Phaser.Math.Clamp(pointer.x, paddle.width / 2, this.game.config.width - paddle.width / 2);
    });
}

function keyDownHandler(e) {
    if(e.keyCode == 39) {
        rkey = true;
    } else if(e.keyCode == 37) {
        lkey = true;
    } else if(e.keyCode == 32) {
        spacebar_action();
    }
}

function keyUpHandler(e) {
    if(e.keyCode == 39) {
        rkey = false;
    } else if(e.keyCode == 37) {
        lkey = false;
    }
}

$(document).ready(function(){
    setup();
    draw();

    var strt = document.getElementById('start');
    var pau = document.getElementById('pause');

    document.addEventListener("keydown", keyDownHandler, false);
    document.addEventListener("keyup", keyUpHandler, false);
    document.addEventListener("mousemove", mouseMoveHandler, false);
});

```

Рисунок 25 – код игры

### 3.14 Использование инструментальных средств контроля версий и баз данных

В разработке веб-приложения была использовано средство контроля версий Git Hub. Git — распределенные системы контроля версий, которые помогают обмениваться кодом и «ковать» проекты в команде — отслеживать и контролировать все изменения в коде. GitHub — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Веб-сервис основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc.

Сервис бесплатен для проектов с открытым исходным кодом и (с 2019 года) небольших частных проектов, предоставляя им все возможности, а для крупных корпоративных проектов предлагаются различные платные тарифные планы.

### 3.15 Проведение работ по резервному копированию веб-приложений

Решение, связанное с резервным копированием и последующим восстановлением веб-сайта, позволяет реализовать механизм резервного копирования для веб-сайта на основе хранилища Windows Azure Storage. В ряде случаев решение позволяет обеспечить совместимость на уровне веб-сервера и базы данных, а также реализовать возможность запуска копии веб-сайта непосредственно из «облака».

Суть бэкапа сайта сводится к копированию баз данных, файлов сайта, почты, FTP-аккаунтов и множества других параметров хостинга. Проще говоря, сохраняется весь сайт и его настройки в отдельном месте и, при необходимости, можно вернуть сайт к той версии, которая сохранена. При этом может

осуществляться копирование данных на текущий и бэкапный (дополнительный) сервер, располагающийся отдельно от серверов провайдера, либо в другом дата-центре. Оно производится на случай, если что-то случится с сервером, на котором хранится сайт.

### 3.16 Регистрация и обработка запросов заказчика в службе технической поддержки

Автоматизация отделов технической поддержки пользователей (ОТПП) – задача на сегодняшний день известная и распространенная. Можно говорить об огромном количестве удачно внедренных проектов в различных отраслях, в различных по масштабу компаниях, с применением различных средств автоматизации. Когда говорят об автоматизации ОТПП, подразумевается внедрение автоматизированного средства из группы программного обеспечения HelpDesk. Фактически понятие HelpDesk стало аналогом понятия, что говорит о том, что в настоящее время большинство ИТ-служб в той или иной степени пересмотрели свой подход к структуре и процессам управления ИТ. Это во многом связано с распространением процессного подхода для управления ИТ и применением передовых методологий eTOM, ITSM\ITIL, COBIT. Если произвести анализ данной структуры как большинства крупных и средних ИТ подразделений, то независимо от того используют они эти методологии или нет, выстроена модель управления процессами ИТ или нет - мы обязательно найдем аналог понятию HelpDesk, которое определено в библиотеке передового опыта в области информационных технологий ITIL.

### 3.17 Индивидуальное задание

## Вывод

В ходе прохождения учебной практики УП 03.01 МДК 03.01 «Проектирование, разработка и оптимизация веб-приложений» научился:

1. Осуществлять сбор предварительных данных для выявления требований к веб-приложению.
2. Определять первоначальные требования заказчика к веб-приложению и возможности их реализации.
3. Подбирать оптимальные варианты реализации задач и согласование их с заказчиком.
4. Оформлять техническое задание.
5. Выполнять верстку страниц веб-приложений.
6. Использовать специальные готовые технические решения при разработке веб-приложений.
7. Разрабатывать интерфейс пользователя.
8. Разрабатывать анимационные эффекты.
9. Кодировать на языках веб-программирования.
10. Использовать инструментальные средства контроля версий и баз данных.
11. Проводить работы по резервному копированию веб-приложений.
12. Выполнять регистрацию и обработку запросов Заказчика в службе технической поддержки.





## МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

Факультет среднего профессионального образования

Цикловая комиссия информационных технологий

Специальность 09.02.07 Информационные системы и программирование  
(шифр) (наименование специальности)

### **ДНЕВНИК ОБУЧАЮЩЕГОСЯ ПО ПРАКТИКЕ**

---

(наименование практики, отражающее вид и тип практики, в соответствии с программой практики по специальности)

---

УП 03.01 МДК 03.01 <sup>(группа)</sup> «Проектирование, разработка и оптимизация веб-приложений» <sup>(Ф.И.О.)</sup>

Адрес деканата: 394036, г. Воронеж, Ленинский пр-т, 14, ВГУИТ, факультет среднего профессионального образования; телефон (8-473)-249-93-79, факс (8-473)-249-93-79.

ИСП-195 Рубцов Илья Евгеньевич

Воронеж 2022 г.

Дата	Виды работ	Кол-во часов	Подпись руководителя практики
03.1 1.20 22	Осуществлять сбор предварительных данных для выявления требований к веб-приложению.	6	
04.1 1.20 22	Определять первоначальные требования заказчика к веб-приложению и возможности их реализации.	6	
06.1 1.20 22	Подбирать оптимальные варианты реализации задач и согласование их с заказчиком.	6	
08.1 1.20 22	Оформлять техническое задание	6	
10.1 1.20 22	Выполнять верстку страниц веб-приложений.	6	
11.1 1.20 22	Использовать специальные готовые технические решения при разработке веб-приложений.	6	
12.1 1.20 22	Разрабатывать интерфейс пользователя.	6	
13.1 1.20 22	Разрабатывать анимационные эффекты	6	
14.1 1.20 22	Кодировать на языках веб-программирования.	6	
15.1 1.20 22	Использовать инструментальные средства контроля версий и баз данных.	6	
16.1 1.20 22	Проводить работы по резервному копированию веб-приложений.	6	
16.1 1.20 22	Выполнять регистрацию и обработку запросов Заказчика в службе технической поддержки.	6	