

Министерство науки и высшего образования Российской Федерации
Лысьвенский филиал федерального государственного автономного
образовательного
учреждения высшего образования
**«Пермский национальный исследовательский политехнический
университет»**

Факультет профессионального образования

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему «Разработка web-сайта для ООО «АНДСЕР-КСБ»»

студента группы КСК9-18-1спо по специальности 09.02.01 Компьютерные
системы и комплексы

Тимощука Данилы Борисовича _____

Руководитель работы: _____ А.С. Иванова

Консультант по
экономической части: _____ К.В. Кондратьева

Консультант по промышленной
экологии и охране труда: _____ А.К. Тороцин

Рецензент: _____ (_____)

Допуск к защите: _____ М.Н. Апталаев

Лысьва, 2022 г.

Министерство науки и высшего образования и Российской Федерации
Лысьвенский филиал федерального государственного автономного
образовательного учреждения высшего образования
«Пермский национальный исследовательский политехнический
университет»
ПЦК «Естественнонаучных дисциплин»

Утверждаю:
Председатель ПЦК
_____ М.Н. Апталаев
«__» _____ 2022 г.

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

студенту ТИМОЩУКУ Даниле Борисовичу курса 4
группы КСК9-18-1спо
специальность 09.02.01 Компьютерные системы и комплексы
Тема задания «Разработка Web-сайта для ООО «АНДСЕР-КСБ»»

Структура выпускной квалификационной работы такова:

а) Введение. Аргументировать актуальность выбранной темы, ее теоретическое значение и практическую значимость, сформулировать цель и конкретные задачи исследований. Конкретизировать объект и предмет исследований.

б) Исследовательский раздел. Обзор аналогов и технологий разработки. Описание предметной области web-сайта. Функциональная модель предметной области web-сайта. Спецификация на уровне взаимодействия подсистем.

в) Конструкторский раздел. Обоснование выбора программного обеспечения по созданию сайта. Разработка функционального внешнего пользовательского интерфейса. Разработка графического макета web-сайта. Описание разработки сайта.

г) Организационно-экономический раздел. Выполнение технико-экономической оценки разработанного сайта.

д) Охрана труда и промышленная экология. Анализ вредных и опасных факторов на рабочем месте инженера-программиста. Разработка средств защиты от воздействия выбранного ВиОПФ. Экологические требования к утилизации вычислительной и оргтехники, а также их расходных материалов.

е) Заключение. Краткое изложение решенных задач, актуальность работы, соответствие полученных результатов теме и заданию ВКР.

ж) Список использованных источников.

з) Приложения.

Дата выдачи _____
Срок окончания _____

Руководитель ВКР
_____ /А.С. Иванова/
«__» _____ 2022 г.

Задание утверждено на заседании ПЦК «Естественнонаучных дисциплин» протокол № _____ 2022 г.

Председатель ПЦК _____ /М.Н. Апталаев/
«__» _____ 2022 г.

Министерство науки и высшего образования Российской Федерации
 Лысьвенский филиал федерального государственного автономного
 образовательного учреждения высшего образования
 «Пермский национальный исследовательский политехнический
 университет»

ПЦК «Естественнонаучных дисциплин»

Утверждаю

Председатель ПЦК

_____ М.Н. Апталаев

«__» _____ 2022 г.

График выполнения выпускной квалификационной работы
 студента группы КСК9-18-1сно
 специальности 09.02.01 Компьютерные системы и комплексы

Тимощука Данилы Борисовича

(фамилия, имя, отчество)

Содержание	Сроки	Отметка о выполнении	Дата										Подпись	
			08.03-20.04	21.04-30.04	01.04-17.05	18.05-25.05	26.05-30.05	31.05-06.06	07.06-15.06	16.06-23.06				
Введение 1. Теоретическая часть	08.01 – 20.02													
2. Охрана труда и БЖД	20.02 – 30.02													
3. Промышленная экология	01.03 – 17.03													
4. Инженерно-технической часть	18.03- 25.03													

« _____ » _____ 2022 г.

Согласовано:

Председатель ПЦК «Естественнонаучных дисциплин»

_____ / М.Н. Апталаев/

« _____ » _____ 2022 г.

РЕФЕРАТ

Выпускная квалификационная работа 4 главы, 65 стр., 29 рисунков, 5 таблиц, 21 источник, 1 приложение.

ПЕРСОНАЛЬНЫЕ ДАННЫЕ, ИНФОРМАЦИОННАЯ СИСТЕМА ПЕРСОНАЛЬНЫХ ДАННЫХ, РАЗРАБОТКА, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, WEB-САЙТ.

Объектом исследования является разработка web-сайта для ООО «АНДСЕР-КСБ»

Предметом исследования является технология проектирования и программирования web-сайта.

Целью выпускной квалификационной работы является разработка web-сайта, предназначенного для взаимодействия студентов и менеджеров, которые ищут стажеров.

Результатом работы является web-сайт для ООО «АНДСЕР-КСБ». Сайт находится на стадии тестирования.

Практическая значимость работы заключается в необходимости для компаний сохранения времени взаимодействия с неподходящими на их вакансию людей, путем добавления дополнительной анкеты, благодаря которой, студент сможет оставить данные о своих навыках и опыте работы.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	9
1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ	11
1.1 Обзор аналогов и технологий разработки	11
1.2 Описание предметной области web-сайта	13
1.3 Информационная структура web-сайта	14
1.4 Основные возможности и требования к сайту	18
2 КОНСТРУКТОРСКИЙ РАЗДЕЛ	20
2.1 Выбор программного обеспечения	20
2.2 Разработка функционального внешнего пользовательского интерфейса ..	22
2.3 Описание разработки сайта	29
3 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ	40
3.1 Расчет затрат на разработку web-сайта	40
3.2 Расчет затрат на внедрение web-сайта	43
3.3 Расчет эксплуатационных текущих затрат по web-сайту	45
3.4 Расчет экономической целесообразности разработки и внедрения информационных технологий	48
4 Охрана труда и промышленная экология	54
4.1 Анализ вредных и опасных производственных факторов на рабочем месте web-дизайнера	54
4.2 Мероприятия по оптимизации освещенности на рабочем месте	55
4.3 Экологические требования к утилизации вычислительной и оргтехники, а также их расходных материалов	58
ЗАКЛЮЧЕНИЕ	63
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	65
ПРИЛОЖЕНИЕ А – Техническое задание	68

ВВЕДЕНИЕ

У любой современной компании существует собственный web-сайт. Это один из элементов репутации, ведь именно в Интернете потенциальные клиенты будут в первую очередь искать информацию о компании [1].

Сайт – это набор страниц в интернете, позволяющий пользователю получить определённую информацию, либо осуществить действие – например, заказать товар и услугу. Состоит из одной либо нескольких логически связанных между собой страниц, объединённых дизайном, общей тематикой и гиперссылками [2].

Актуальность создания сайта – это донесение информации максимально быстро до большого количества человек. Для того, чтобы это осуществить, лучше, чем собственный сайт, способа не найти.

Web-сайт позволяет представить информацию о компании или организации и ее товарах или услугах сжато и одновременно полноценно. Также сайт может сообщать о новостях фирмы, об изменениях в представляемой продукции или режиме работы, содержать отзывы клиентов.

Не для каждой фирмы необходим огромный web-ресурс. Иногда бывает достаточно небольшого сайта-визитки, для заказа которого можно обратиться к фрилансерам или же сделать самому [1].

Сайты могут различаться между собой по многим параметрам, поэтому пока не существует единой классификации. Эксперты разделяют сайты по наиболее значимым признакам, например, по тематике, структуре, дизайну, способу и цели создания.

Узкотематическим называют интернет-ресурс, посвящённый одной теме. Если же он охватывает широкий круг проблем – общим. В зависимости от ориентированности на получение прибыли различают коммерческие и некоммерческие сайты, а по количеству страниц – одностраничные и многостраничные [3].

Объектом исследования выпускной квалификационной работы является разработка web-сайта для ООО «АНДСЕР-КСБ»

Предметом исследования выпускной квалификационной работы является технология проектирования и программирования web-сайта.

Целью выпускной квалификационной работы является разработка web-сайта, предназначенного для взаимодействия студентов и менеджеров, которые ищут стажеров.

Задачи:

- Изучить основные принципы технологий по разработке web-сайтов;
- Проанализировать аудиторию сайта и ее потребностей, относящихся к освещаемой деятельности;
- Обосновать выбор программного обеспечения по созданию сайта;
- Разработать функциональный внешний пользовательский интерфейс;
- Разработка программно-административной части web-сайт;
- Выполнить технико-экономическую оценку разработанного сайта;
- Изучить требования охраны труда и влияние опасных факторов.

1 ИССЛЕДОВАТЕЛЬСКИЙ РАЗДЕЛ

1.1 Обзор аналогов и технологий разработки

Один из многочисленных плюсов интернета является возможность использования информации и услуг, расположенными в сети, а также размещение в ней своей информации, предоставив ее в виде Web-страницы.

Веб-страница – это файл определенного формата. Этот формат определяет внешний вид документа, расположение графической, мультимедийной и текстовой информации. Экранное представление web-страницы разрабатывается с помощью специальных команд на специальном языке, которые интерпретируются браузером [4].

Любой сайт состоит из двух частей – программной и визуальной

Программную часть разрабатывают Backend-специалисты, а визуальную Frontend-разработчики. Для Backend-специалистов больше всего подходят такие языки программирования как Python, PHP, Ruby. В то время как Frontend-разработчики используют такие технологии как CSS, HTML, JavaScript.

Так же существуют специальные CMS. Content Management System – это система создания и управления сайтом. В ней визуальное удобное управление, с помощью которого можно добавлять и редактировать содержимое сайта [5].

Несомненными плюсами CMS можно считать:

- возможность сэкономить на штате программистов,
- простоту и удобство использования,
- высокую скорость создания сайта,

Минусы:

- CMS не подходят для нестандартных сайтов,
- бесплатные CMS могут быть уязвимы.

Одними из популярных являются Tilda и WordPress, которые в несколько раз упрощают процедуру создания сайта. На рисунке 1 представлен интерфейс WordPress.

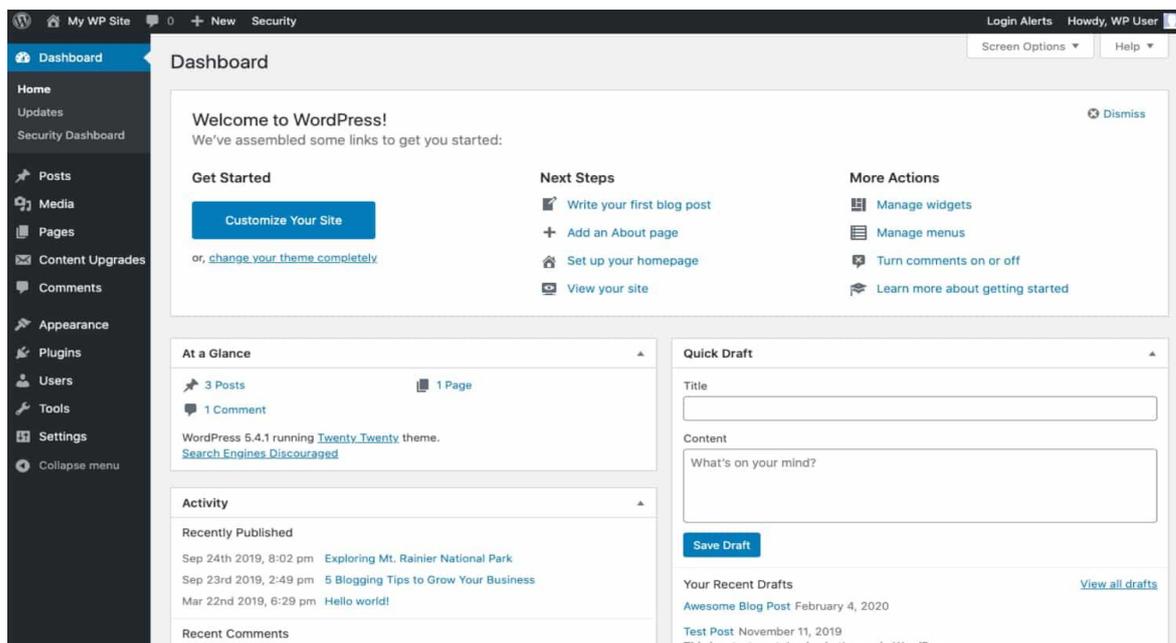


Рисунок 1 – Интерфейс web-конструктора WordPress

Перечислим основные технологии разработки сайтов:

- TypeScript, JavaScript – языки программирования, заставляющие скрипты выполняться;
- HTML, CSS – это технологии разметки страницы (HTML) и внешнего вида страницы (CSS);
- база данных PostgreSQL – отвечает за хранение текстовой информации.

На сегодняшний день для любителей и профессионалов создавать сайты существуют полезные инструменты, в разы упрощающие процесс разработки.

Перечень полезных инструментов, для создания макета будущего сайта:

- Photoshop – приложение для работы с графикой, включая векторную, а также создания и редактирования изображений;
- Figma – онлайн-сервис для дизайнеров, web-разработчиков и маркетологов. В редакторе можно настроить совместную работу, вносить и обсуждать правки, причем как в браузере, так и через приложение на компьютере.

На рисунке 2 представлен интерфейс Figma [6].

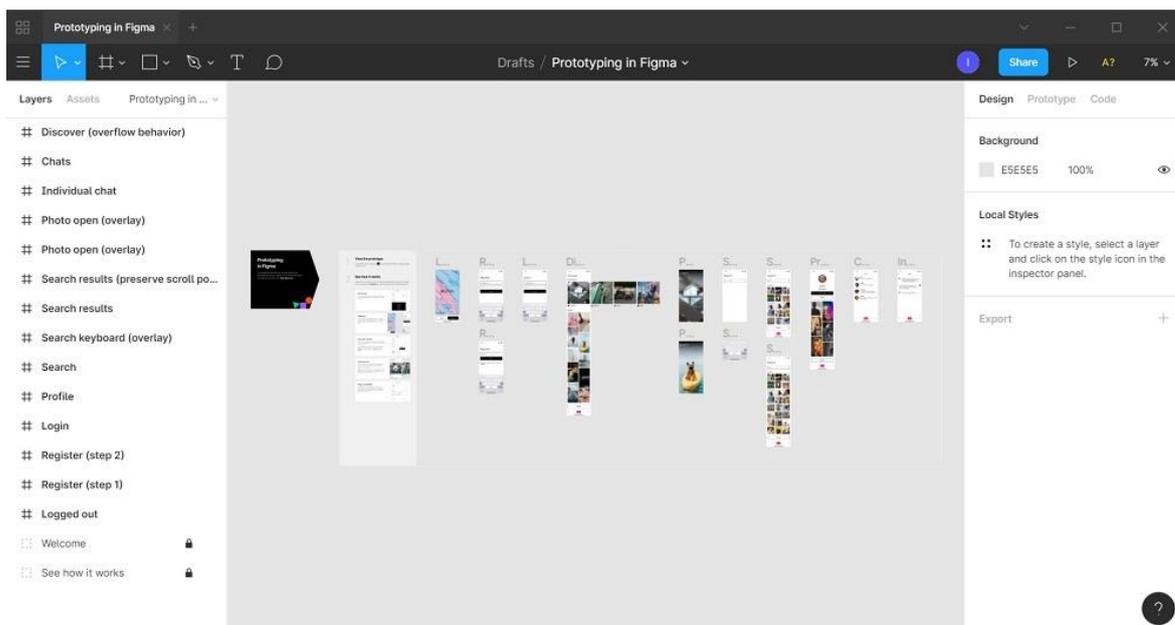


Рисунок 2 – Интерфейс графического редактора Figma

1.2 Описание предметной области web-сайта

На первоначальной стадии проектирования web-сайта следует осуществить анализ предметной области, т.е. определить объекты предметной области, а также взаимосвязь между объектами.

Созданный web-сайт представляет собой концепцию поиска стажировок и стажеров и позволяет отслеживать количество людей, откликнувшихся на вакансию.

При разработке проекта с целью предоставления его наибольшей производительности очень важно принять во внимание характерные черты информационной среды, в которой будет происходить эксплуатация.

Пользователи имеют ограниченные права на сайте. Не зарегистрированные пользователи не смогут создавать, либо откликаться на вакансии. В свою очередь зарегистрированные пользователи смогут редактировать только информацию, которую они создали.

На сайте существует два зарегистрированных вида пользователя:

- Компания, создает вакансии для студентов.
- Студенты, могут откликаться на созданные компаниями вакансии.

1.3 Информационная структура web-сайта

Структура сайта – это схема построения страниц сайта с распределением по папкам и категориям. С точки зрения пользователя структура сайта – это путь, навигация по тому или иному сайту [7].

1.3.1 Принципы построения структуры сайта

Главным образующим звеном каждого web-сайта считается документ – это страница web-сайта, оригинальный адрес в интернете. Документы могут быть как простыми информационными страницами, так и карточками товара или, к примеру, листингами товаров в интернет-магазине.

Каждый web-сайт представляет собой древовидную структуру, где за «ствол» можно представить главную страницу, а разделы и статьи – ветки и листья. Насколько обширной будет структура сайта, зависит от области применения и типа сайта, но даже сайт, состоящий из одной страницы, уже представляет собой древовидную структуру [8].

Принцип построения связей между документами зависит от типа сайта, который нам необходим.

Можно выделить несколько типов сайтов:

1. Сайт-визитка. Данный тип сайтов становится менее популярным, но свою практичность он не теряет. Пример структуры представлен на рисунке 3.

Такой вариант строения предоставляет всю необходимую для пользователя информацию в понятном и кратком виде.



Рисунок 3 – Пример структуры web-сайта визитки

2. Интернет магазин. Такие web-сайты самые сложные и многоуровневые по структуре, с довольно большим количеством страниц.

Из-за огромного количества товаров структура здесь является очень значимым элементом. Особенностью является то, что количество страниц уменьшено, путем создания фильтров.

Фильтры, в свою очередь, имеют большую необходимость именно для покупателя, для облегчения поиска и выбора необходимого товара. Например, фильтр на размер обуви, цвет одежды и т.д. Краткая схема интернет магазина представлена на рисунке 4.

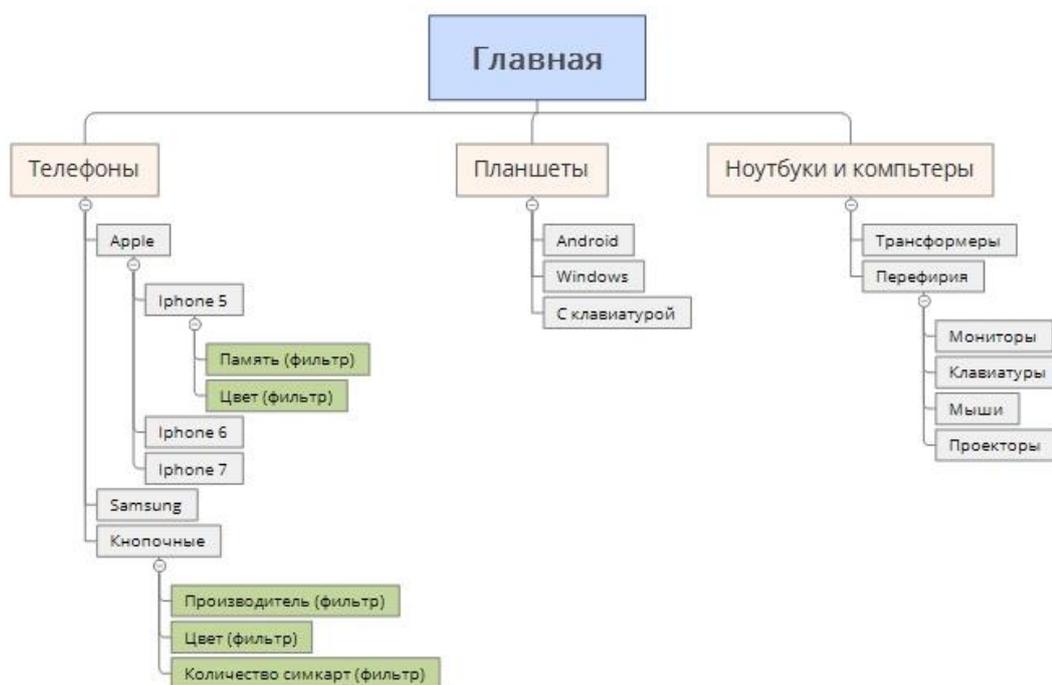


Рисунок 4 – Пример структуры интернет магазина

3. Информационный сайт. Структуры web-сайтов не такие разветвленные как структуры интернет магазинов. Чаще всего они включают в себя разделы, посвященные тому или иному направлению.

Страница интернет магазина отличается от раздела тем, что в разделах представлен список (листинг) всех документов на тему раздела, без дальнейшей структуризации.

Информационные сайты можно разделить два вида, от которых будет зависеть их структура:

1. Сайты с узкой тематикой. Пример информационного сайта с узкой тематикой фитнеса представлен на рисунке 5.



Рисунок 5 – Упрощенная структура информационного сайта о здоровом образе жизни

В данном случае древовидная структура, в которой каждая статья связана с разделом

2. Сайты-порталы. Это такой же информационный сайт, но с более широкими направлениями. Порталы включают в себя широкие, массивные разделы и обладают разветвлённой структурой. Представлен на рисунке 6.

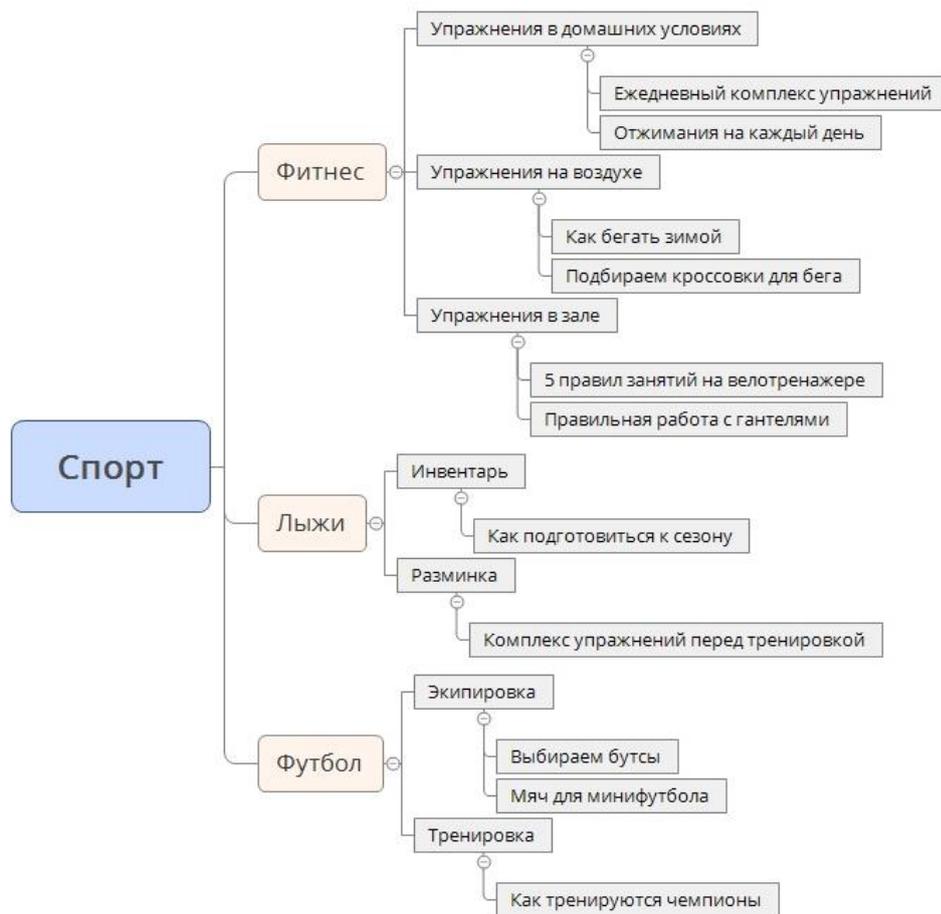


Рисунок 6 – Многоуровневая структура сайта портала с большим количеством рубрик

3. Коммерческий магазин

В данном случае необходимо объяснить и показать пользователю, чем занимается бизнес, что он может предоставить, а также свои особенности.

Пример структуры сайта по ремонту квартир представлен на рисунке 7.



Рисунок 7 – Пример структуры коммерческого сайта

На этот раз количество связей имеет более разветвленный вариант и из раздела «Услуги» мы можем перейти в рубрику «Ремонт квартир», а уже из неё попасть на документ «Капитальный ремонт» [9].

1.4 Основные возможности и требования к сайту

Необходимо выделить функциональные и нефункциональные возможности и требования:

Нефункциональные:

- удобство и простота интерфейса;
- удобная навигация по web-сайту;
- небольшой объем страниц.

Функциональные:

- авторизация, регистрация, аутентификация;
- возможность добавления и удаления стажировки;
- заполнение профиля компании и студента;
- поиск стажировок по названию и по компании;
- отклики от студентов;
- подсчет откликов на стажировки.

На рисунке 8 изображена диаграмма вариантов использования

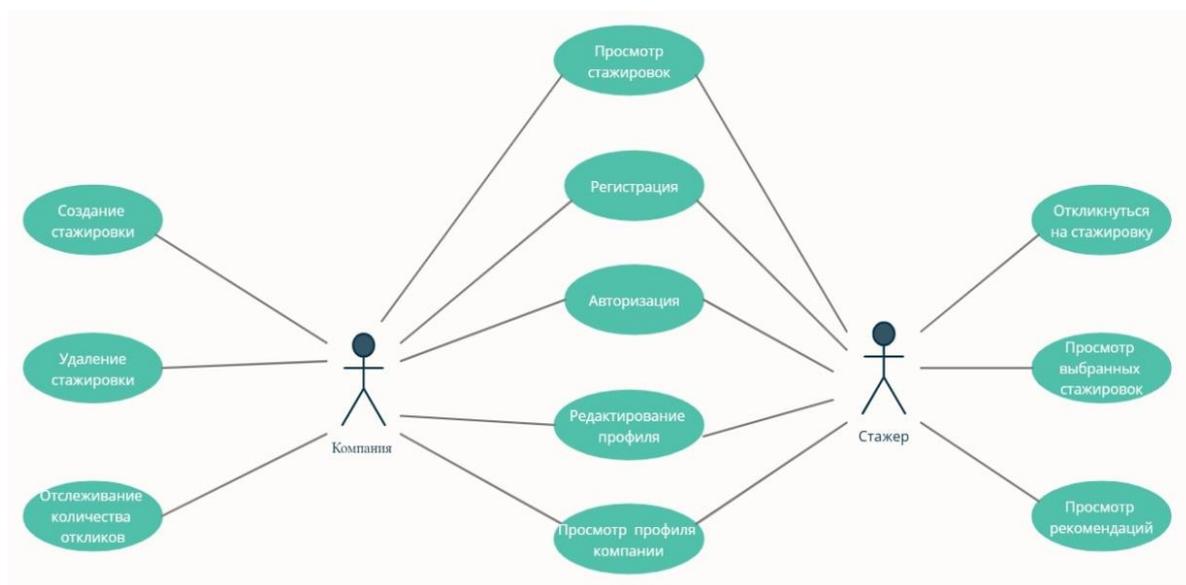


Рисунок 8 – Диаграмма вариантов использования

Вывод по разделу.

В соответствии с темой выпускной квалификационной работы были рассмотрены аналоги и технологии разработки. Из доступных технологий были выбраны Java, PostgreSQL, HTML, CSS, TypeScript.

Были расписаны информационные структуры web-сайтов. В зависимости от целей, сайты условно можно разделить на несколько категорий:

- сайт-визитка
- интернет магазин
- информационный сайт

Также расписаны предметная область, основные требования и возможности web-сайта.

2 КОНСТРУКТОРСКИЙ РАЗДЕЛ

2.1 Выбор программного обеспечения

Для разработки web-сайта рассмотрим несколько видов фреймворков. Фреймворк – каркас приложения, который состоит из набора библиотек. Вместе они дают возможность пользователю выстроить свою функциональность на некотором фундаменте. Фреймворк задаёт структуру и подход к архитектуре приложения, он диктует, как необходимо писать код [10].

2.1.1 Vue.js

Фреймворк для создания Single Page Application (SPA), реализует шаблон Model-View-ViewModel (MVVM), имеет открытый исходный код.

SPA – это web-приложение или web-сайт, использующий единственный HTML-документ как оболочку для всех web-страниц и организующий взаимодействие с пользователем через динамически подгружаемые HTML, CSS, JavaScript. Другими словами, это сайт, для работы которого не требуется обновление страницы, потому что все данные загружаются с помощью скриптов.

MVVM – это шаблон проектирования архитектуры приложения. Используется для разделения модели и её представления, что необходимо для их изменения отдельно друг от друга. Например, разработчик задаёт логику работы с данными, а дизайнер работает с пользовательским интерфейсом.

JavaScript-фреймворк с открытым исходным кодом для создания пользовательских интерфейсов. Легко интегрируется в проекты с использованием других JavaScript-библиотек. Может функционировать как web-фреймворк для разработки одностраничных приложений в реактивном стиле [11].

Разработчики называют Vue.js прогрессивным и постепенно адаптируемым по сравнению с другими web-фреймворками.

Это позволяет разработчику настроить структуру приложения в соответствии с собственными требованиями. Разработчики считают Vue.js

более простым в освоении, чем Angular 2, поскольку API построен намного проще в освоении. В Vue.js можно использовать только знания JavaScript и HTML. Возможно применение Typescript [12].

2.1.2 Ember.js

Это ещё один фреймворк для создания SPA с открытым кодом. Сам же фреймворк построен на классической архитектуре Model-View-Controller (MVC). У Ember очень дружное сообщество, которое развивает инструмент и не останавливается, даже в 2020 году у фреймворка есть цели по развитию [10].

Ember.js – свободный каркас web-приложений на JavaScript, реализующий шаблон MVC и нацеленный на упрощение создания масштабируемых одностраничных web-приложений.

Основные внутренние концепты, используемые каркасом – маршруты, модели, контроллеры и шаблоны. Ember.js – это фреймворк, ориентированный исключительно на фронтенд. Он поддерживает множество способов взаимодействия с различными бэкендами, но всё, что относится к серверному коду, не входит в сферу ответственности Ember.

Фронтенд (англ. front-end) – клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса.

Бэкенд (англ. back-end) – программно-аппаратная часть сервиса, отвечающая за функционирование его внутренней части.

2.1.3 Angular 2

Angular 2 – фреймворк для создания SPA, включает в себя инструменты для разработки и тестирования, реализует архитектуры MVC и MVVM, имеет открытый исходный код.

Для описания интерфейса используется декларативное программирование, а бизнес-логика отделена от кода интерфейса, что позволяет улучшить тестируемость и расширяемость приложений.

Другой отличительной чертой фреймворка является двустороннее связывание, позволяющее динамически изменять данные в одном месте

интерфейса при изменении данных модели в другом. Таким образом, Angular 2 синхронизирует модель и представление [13].

Кроме того, Angular 2 поддерживает такие функциональности, как Ajax, управление структурой Document Object Model (DOM), анимация, шаблоны, маршрутизация и так далее.

Ajax – это подход к построению интерактивных пользовательских интерфейсов web-приложений, заключающийся в «фоновом» обмене данными браузера с web-сервером. В результате при обновлении данных web-страница не перезагружается полностью, и web-приложения становятся быстрее и удобнее [14].

Мощь фреймворка, наличие богатого функционала во многом повлияла на то, что он находит свое применение во все большем количестве web-приложений, являясь на данный момент, одним из самых популярных javascript-фреймворков.

В проекте выбран Angular 2, поскольку он является фреймворком с низким порогом вхождения. Изначально он использует TypeScript, что также является большим преимуществом. На данный момент является одним из самых популярных фреймворков, поэтому на него существует большое количество документаций и пояснений.

2.2 Разработка функционального внешнего пользовательского интерфейса

Структурная схема является первым этапом в разработке проекта. Представляет собой общую характеристику проекта, его составные части и их взаимодействие, представленная на рисунке 8.

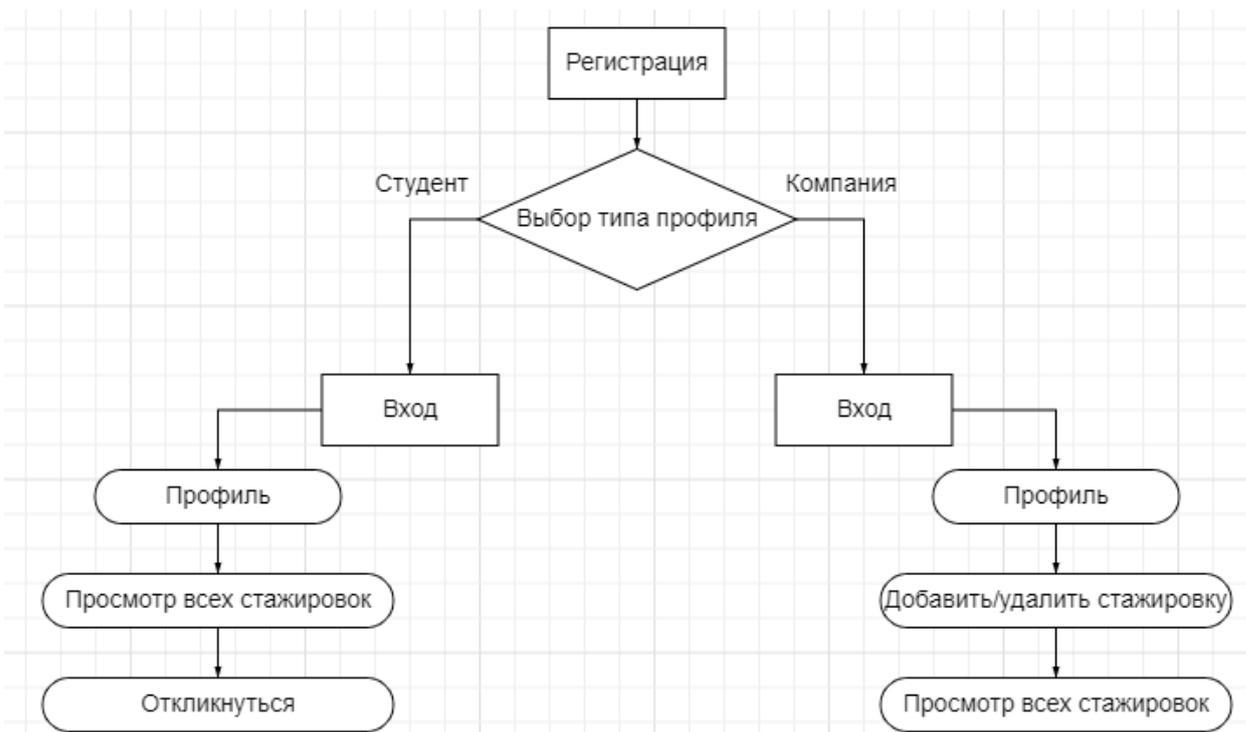


Рисунок 8 – Структурная схема разрабатываемого проекта

После разработки структурной схемы приступаем к разработке функциональной. Она описывает процессы, происходящие внутри отдельных частей проекта, а также их взаимодействие.

На главной странице отображаются вакансии. В шапке страницы находятся кнопки регистрации и входа в аккаунт. На рисунке 9 отображена шапка сайта.

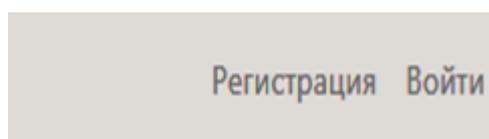


Рисунок 9 – Шапка Web-сайта

Для того чтобы откликнуться или создать вакансию, необходимо зарегистрироваться и войти в свой профиль

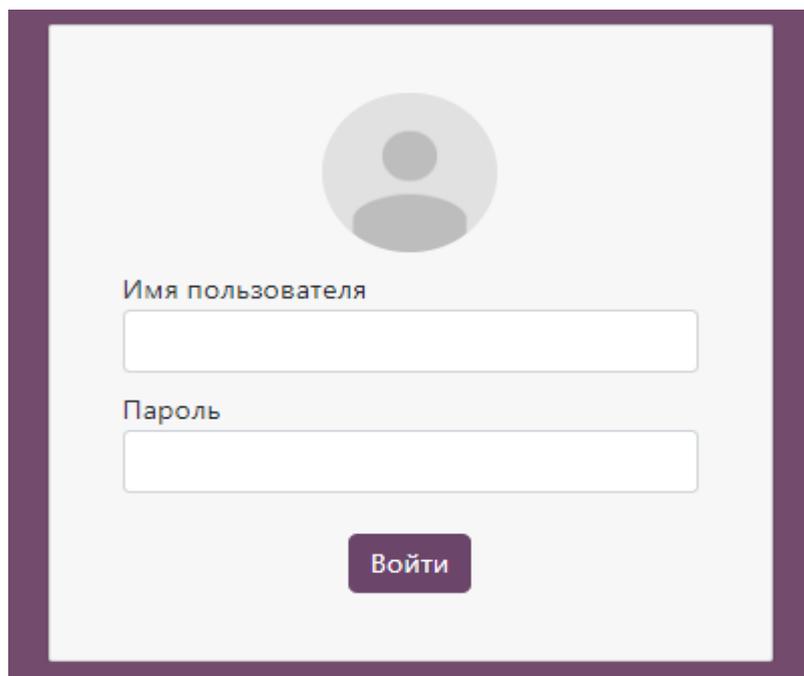
При переходе по любой из кнопок пользователь попадает на страницу с соответствующей названию информацией. При нажатии на пункт «Регистрация» пользователь попадает на страницу с формой для заполнения данными о пользователе. На Рисунке 10 изображена страница с формой регистрации.

The image shows a registration form with a purple border. At the top center is a grey circular placeholder for a user profile picture. Below it are four input fields, each with a red border and a red exclamation mark icon on the right side, indicating they are required. The first field is labeled 'Имя пользователя' (Username). The second is 'Пароль' (Password). The third is 'Подтверждение пароля' (Confirm password). The fourth is 'Роль' (Role), which is a dropdown menu with two options: 'Студент' (Student) and 'Компания' (Company). Below the dropdown is a purple button labeled 'Регистрация' (Registration).

Рисунок 10 – Страница «Регистрация»

На данной странице можно выбрать тег студента или компании. От этого зависит дальнейшее взаимодействие с функционалом сайта.

При нажатии на кнопку в шапке сайта «Войти» пользователь попадает на страницу, в которой представлены формы заполнения. На Рисунке 11 изображена страница с формой регистрации



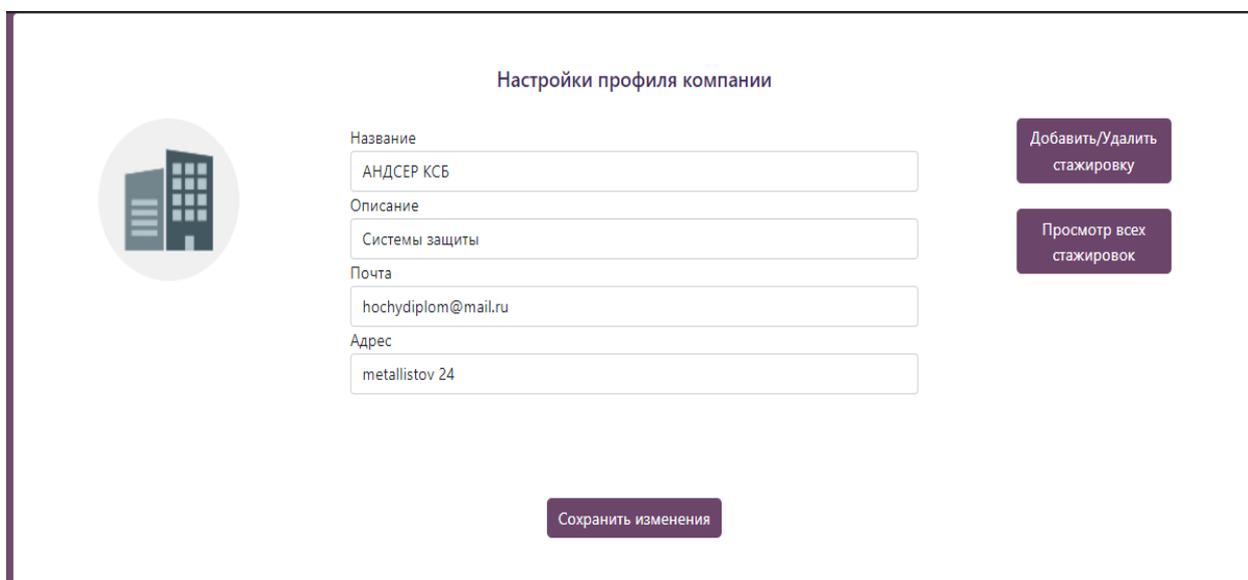
Имя пользователя

Пароль

Войти

Рисунок 11 – Страница «Войти»

После входа на зарегистрированный аккаунт пользователь попадает на страницу с редактированием данных профиля. Страница с данными об компании представлена на рисунке 12.



Настройки профиля компании

Название

Описание

Почта

Адрес

Добавить/Удалить стажировку

Просмотр всех стажировок

Сохранить изменения

Рисунок 12 – Страница «Настройка профиля компании»

Профиль с тегом «Компания» может создавать и удалять вакансии на стажировку, а также просматривать количество откликов на ту или иную вакансию. Форма заполнения вакансии представлена на рисунке 13.

Название

Описание

Ссылка на форму

Тэг

Добавить

Рисунок 13 – Форма заполнения вакансии

Компания так же может указать тег, чтобы у студента была возможность сразу понять ориентированность вакансии. Окно выбора тегов показано на рисунке 14.

- Административный персонал
- Безопасность
- Высший менеджмент
- Закупки
- Информационные технологии

Рисунок 14 –Окно выбора тегов

На рисунке 15 изображено окно информации о созданных вакансиях. Так же отображено количество откликов.

Количество откликов: 2

Теги: Информационные технологии, Управление персоналом

Стажер "Помощник"

АНДСЕР КСБ

Если ты:

Обладаешь знаниями основ программирования на языках Python/Java/C++;
Имеешь уверенные навыки работы в Excel, SQL;
Понимаешь принципы построения современных IT-продуктов и имеешь аналитический склад ума.
Мы предлагаем получить уникальный опыт на стажировке с возможностью дальнейшего трудоустройства в одном из приоритетных подразделений компании.

Тебя ждет:

Работа над реальным продуктом для рынка;
Обучение от экспертов и команда наставников из ТОП-менеджмента;
3 месяца работы в комбинированном формате и конкурентный уровень заработной платы.

Удалить

Рисунок 15 –Окно созданной вакансии

Страница с данными о студенте представлена на Рисунке 16.

Настройки профиля стажера



Имя
Леви

Фамилия
Акерман

Отчество
Иванович

Почта
rapich@yandex.ru

Номер телефона
88005553535

Просмотр всех стажировок

Сохранить изменения

Рисунок 16 –Страница «Настройка профиля студента»

На странице с поиском вакансий размещено окно, в котором можно выбрать тег, чтобы уменьшить количество выводимых вакансий, а также строка поиска, в которой по ключевым словам можно найти необходимую вакансию. Ниже этих строк расположены кнопки, с помощью которых можно посмотреть на отмеченные пользователем вакансии. На рисунке 17

изображено окно редактирования поиска.

Поиск..

▼

Все стажировки Мои стажировки Рекомендации

Рисунок 17 – Окно редактирования поиска

На Рисунке 18 изображено окно с примером вакансии

Теги: Информационные технологии, Управление персоналом

Стажер "Помощник" Вы уже откликнулись

АНДСЕР КСБ

Если ты:

- Обладеешь знаниями основ программирования на языках Python/Java/C++;
- Имеешь уверенные навыки работы в Excel, SQL;
- Понимаешь принципы построения современных IT-продуктов и имеешь аналитический склад ума.

Мы предлагаем получить уникальный опыт на стажировке с возможностью дальнейшего трудоустройства в одном из приоритетных подразделений компании.

Тебя ждет:

- Работа над реальным продуктом для рынка;
- Обучение от экспертов и команда наставников из ТОП-менеджмента;
- 3 месяца работы в комбинированном формате и конкурентный уровень заработной платы.

[Откликнуться](#)

Рисунок 18 – Окно с примером вакансии

При нажатии на кнопку «Откликнуться» студент переходит на сайт с онлайн-формой, заранее прикрепленной к вакансии компанией, которую необходимо заполнить. Пример онлайн-формы изображен на Рисунке 19.

The image shows a web form titled "Контактная информация" (Contact Information). It consists of four vertically stacked input fields, each with a label and a "Краткий ответ" (Short answer) placeholder. The first field is for "Описание" (Description). The second field is for "Имя*" (Name), with an asterisk indicating it is required. The third field is for "Адрес электронной почты*" (Email address), also with an asterisk. The fourth field is for "Номер телефона" (Phone number). The form has a light green border and a white background.

Рисунок 19 – Онлайн форма

2.3 Описание разработки сайта

2.3.1 Архитектура web-приложения

Архитектура приложения представляет собой два микро сервиса: Angular-приложение на стороне клиента и SpringBoot-приложение на стороне сервера.

В нашем приложении для Backend-части используется классическая 3-х слойная архитектура. Запрос поступает на слой контроллеров, затем он переходит на слой бизнес-логики, откуда он обращается к слою данных, который в свою очередь обращается к системе управления базами данных (СУБД). На рисунке 20 изображена диаграмма развертывания.

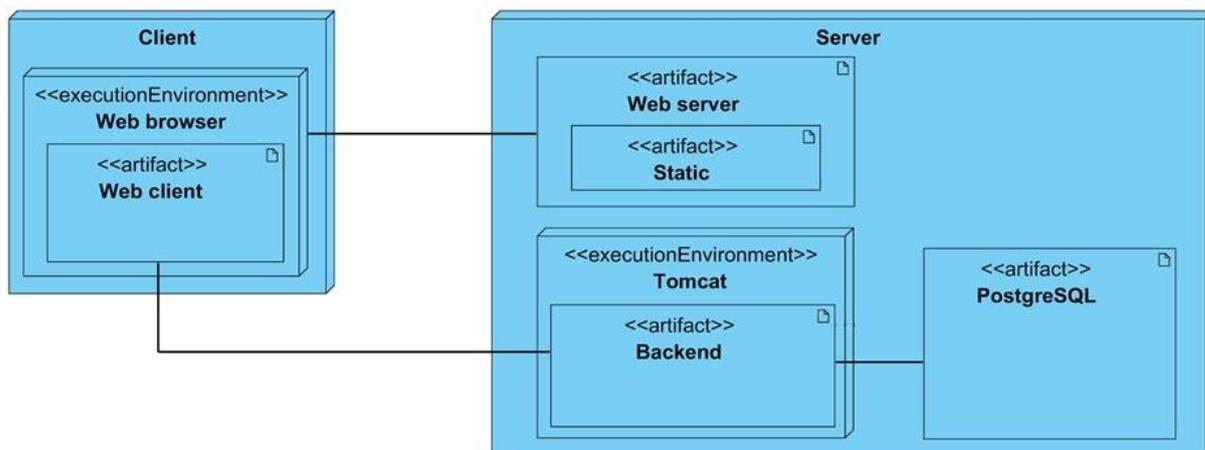


Рисунок 20–Диаграмма развертывания

2.3.2 Описание схемы базы данных

Разберем схему базы данных web-сайта. На рисунке 21 изображена схема базы данных.

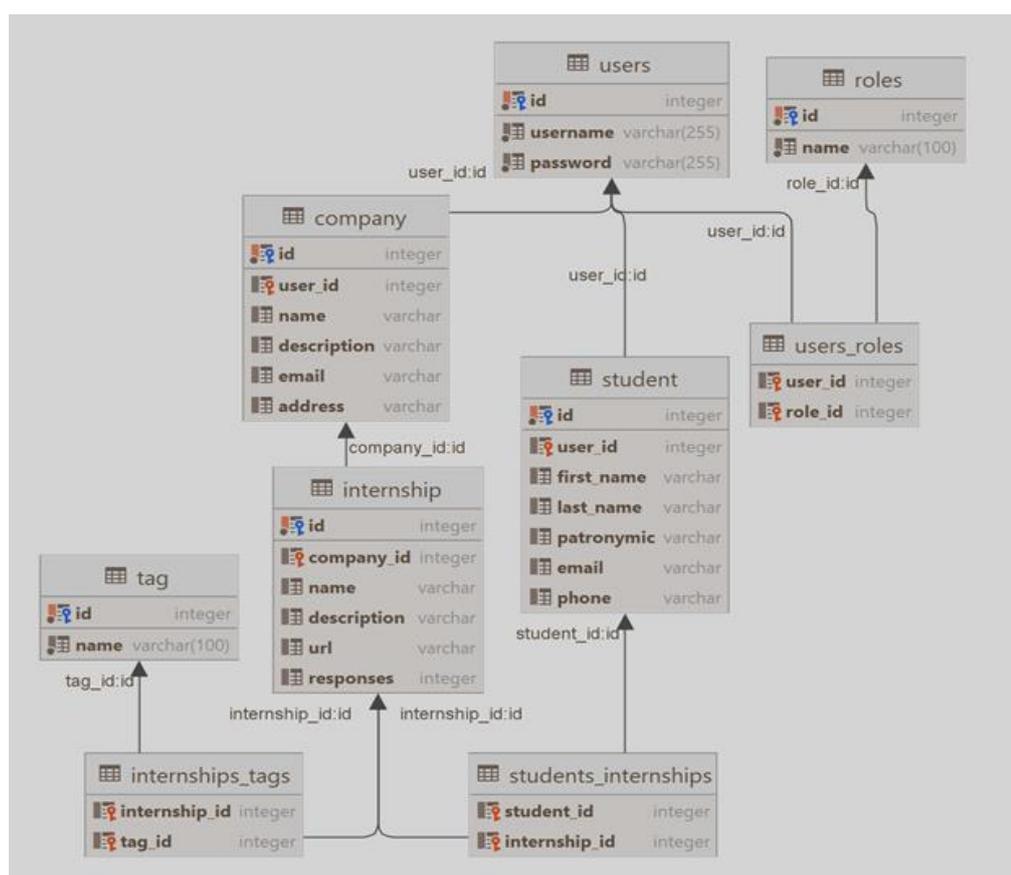


Рисунок 21 - Схема базы данных

Основные сущности web-приложения – это студент, компания, стажировка.

В таблице «student» хранится информация о пользователях, которые ищут стажировки. В ней содержатся имя, фамилия, id, почта, телефон.

Студенты и компании – это пользователи системы. У них есть одинаковые поля, поэтому информация была вынесена в таблицу «users».

В таблице «internship» хранится информация о стажировках, опубликованные компаниями.

В таблице «roles» содержится информация о возможных ролях student и company

Таблица «usersroles» – это таблица связка, соединяющая пользователя с конкретной ролью, хранящую информацию о роли пользователя. Автоматически создается компания или студент в зависимости от выбранной роли при регистрации пользователя.

В таблице «tag» перечислены всевозможные теги, которые можно навесить на стажировку

Таблица «internshiptags» связывает стажировки и теги. На каждую стажировку может быть наложено несколько тегов. Одни и те же теги могут быть наложены на разные стажировки.

Таблица «studentsinternship» хранит информацию о том, на какие стажировки откликнулись студенты

2.3.3 Разработка серверной части приложения

В рамке этой части были реализованы следующие REST-API.

REST API – это способ взаимодействия сайтов и web-приложений с сервером. Термин состоит из двух аббревиатур, которые расшифровываются следующим образом. Application Programming Interface (API) – это код, который позволяет двум приложениям обмениваться данными с сервера. На русском языке его принято называть программным интерфейсом приложения. Representational State Transfer (REST) – это способ создания API с помощью протокола HTTP. На русском его называют «передачей состояния представления».

1. AuthRestController

```
@RestController
@RequestMapping("/api/auth")
public class AuthRestController {
    @PostMapping("/signin")
    public ResponseEntity<?> authenticateUser(@Valid @RequestBody UserDTO userDTO) {
        JwtResponse jwtResponse = this.authService.createJWTResponse(userDTO);
        return ResponseEntity.ok(jwtResponse);
    }

    @PostMapping("")
    public ResponseEntity<?> registerUser(@Valid @RequestBody UserDTO userDTO) {
        if (userRepository.existsByUsername(userDTO.getUsername())) {
            return new ResponseEntity<>(HttpStatus.FORBIDDEN);
        }
        User user = userDTO.toUser(roleRepository, encoder);
        userRepository.save(user);
        return new ResponseEntity<>(HttpStatus.OK);
    }
}
```

Рисунок 22 – Фрагмент кода контроллера

Этот контроллер отвечает за регистрацию и авторизацию пользователя в системе.

– Метод `authenticateUser` – метод авторизации пользователя. Он доступен по API: `api/auth/signin`. Этот метод POST и в теле запроса отправляется сущность `user` в поля `username` и `password`;

– Метод `registerUser` – метод регистрации пользователя. Он доступен по API: `api/auth`. Этот метод POST и в теле запроса отправляется сущность `user` (поля `username`, `password`, `confirmPassword` и `role`).

2. CompanyRestController

```

@RestController
@RequestMapping("/api/company")
public class CompanyRestController {

    private final CompanyServiceImpl companyService;

    @RequestMapping(value = "/company/{id}", method = RequestMethod.GET, produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<Company> getCompanyById(@PathVariable("id") Long id) {...}

    @RequestMapping(value = "", method = RequestMethod.GET, produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<List<Company>> getAllCompanies() {...}

    @RequestMapping(value = "/{id}", method = RequestMethod.GET, produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<Company> getCompanyByUserId(@PathVariable("id") Long userId) {...}

    @PreAuthorize("#id == authentication.principal.id")
    @RequestMapping(value = "/{id}", method = RequestMethod.POST, produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<Company> saveCompany(@PathVariable("id") Long id, @RequestBody CompanyDTO companyDTO) {...}

    @RequestMapping(value = "", method = RequestMethod.PATCH, produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<Company> updateCompany(@RequestBody Company company) {...}

    @RequestMapping(value = "/{id}", method = RequestMethod.DELETE, produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<Company> deleteCompany(@PathVariable("id") Long id) {...}
}

```

Рисунок 23 – Фрагмент кода контроллера

Котроллер `CompanyRestController` отвечает за методы API работы с компаниями

- Метод `getCompanyById` – это метод GET, который принимает в качестве параметра `id` компании и возвращает всю сущность `company`. Он доступен по API: `api/company/company/<id>`;

- Метод `getAllCompanies` – это метод GET, который возвращает все компании. Он доступен по API: `api/company`;

- Метод `getCompanyByUserId` – это метод GET, который принимает в качестве параметра `id` пользователя и возвращает сущность `company`, которая доступна конкретно этому пользователю. Этот метод доступен по API: `api/company/<id>`;

- Метод `saveCompany` – это метод POST, который принимает в качестве параметра `id` пользователя и в теле которого, отправляется сущность компании. По сути это метод создания компании. Он доступен по API: `api/company/<id>`;

– Метод `updateCompany` - это метод `PATCH`, в теле которого отправляется сущность компании. По сути это метод изменяет компанию. Он доступен по API: `api/company`;

– Метод `deleteCompany` - это метод `DELETE`, который принимает в качестве параметра `id` компании. По сути это метод удаляет сущность компании. Он доступен по API: `api/company/<id>`.

3. InternshipRestController

```
@RequestMapping(value = "/api/internship")
public class InternshipRestController {

    @PreAuthorize("#id == authentication.principal.id")
    @RequestMapping(value = "/{id}", method = RequestMethod.POST, produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<Internship> saveInternship(@PathVariable("id") Long id, @RequestBody InternshipDTO internshipDTO) {...}

    @RequestMapping(value = "/{id}", method = RequestMethod.PATCH, produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<Internship> updateInternship(@PathVariable Long id, @RequestBody InternshipDTO internshipDTO) {...}

    @RequestMapping(value = "/{id}", method = RequestMethod.DELETE, produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<Internship> deleteInternship(@PathVariable("id") Long id) {...}

    @RequestMapping(value = "", method = RequestMethod.GET, produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<List<InternshipDTO>> getAllInternships() {...}

    @PreAuthorize("#id == authentication.principal.id")
    @RequestMapping(value = "/{id}", method = RequestMethod.GET, produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<List<InternshipDTO>> getAllInternshipsByUserId(@PathVariable("id") Long id) {...}

    @RequestMapping(value = "/company/{id}", method = RequestMethod.GET, produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<List<InternshipDTO>> getAllInternshipsByCompanyId(@PathVariable("id") Long id) {...}

    @PreAuthorize("#id == authentication.principal.id")
    @RequestMapping(value = "/student/{id}", method = RequestMethod.GET, produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<Set<InternshipDTO>> getAllInternshipsByStudentId(@PathVariable("id") Long id) {...}
}
```

Рисунок 24 –Фрагмент кода котроллера

Котроллер `InternshipRestController` отвечает за методы API работы со стажировками.

– Метод `saveInternship` – это метод `POST`, который принимает в качестве параметра `id` пользователя и в теле которого, отправляется сущность стажировки. По сути это метод создания стажировки. Он доступен по API: `api/intership/<id>`;

– Метод `updateInternship` - это метод `PATCH`, который принимает в качестве параметра `id` стажировки и в теле которого отправляется сущность

стажировки. По сути это метод изменяет стажировку. Он доступен по API: `api/intership/<id>`;

– Метод `deleteInternship` - это метод DELETE, который принимает в качестве параметра `id` стажировки. По сути это метод удаляет сущность стажировки. Он доступен по API: `api/intership/<id>`;

– Метод `getAllInternships` – это метод GET, который возвращает все стажировки. Он доступен по API: `api/intership`;

– Метод `getAllInternshipsByUserId` – это метод GET, который принимает в качестве параметра `id` пользователя и возвращает все стажировки доступные конкретно этому пользователю. Этот метод доступен по API: `api/intership/<id>`;

– Метод `getAllInternshipsByCompanyId` – это метод GET, который принимает в качестве параметра `id` компании и возвращает все стажировки этой компании. Этот метод доступен по API: `api/intership/company/<id>`;

– Метод `getAllInternshipsByStudentId` – это метод GET, который принимает в качестве параметра `id` студента и возвращает все стажировки, на которые откликнулся студент. Этот метод доступен по API: `api/intership/student/<id>`.

4. StudentRestController

```
@RestController
@RequestMapping("/api/student")
public class StudentRestController {

    private final StudentServiceImpl studentService;

    private final InternshipServiceImpl internshipService;

    @RequestMapping(value = "/{id}", method = RequestMethod.GET, produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<StudentDTO> getStudentByUserId(@PathVariable("id") Long userId) {...}

    @RequestMapping(value = "", method = RequestMethod.POST, produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<Student> saveStudent(@RequestBody Student student) {...}

    @PreAuthorize("#id == authentication.principal.id")
    @RequestMapping(value = "/{id}", method = RequestMethod.PATCH, produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<StudentDTO> updateStudent(@PathVariable("id") Long id, @RequestBody StudentDTO studentDTO) {...}

    @RequestMapping(value = "/{id}", method = RequestMethod.DELETE, produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<Student> deleteStudent(@PathVariable("id") Long id) {...}

    @RequestMapping(value = "", method = RequestMethod.GET, produces = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<List<Student>> getAllStudents() {...}
}
```

Рисунок 25 –Фрагмент кода контроллера

Котроллер `StudentRestController` отвечает за методы API работы со студентом.

– Метод `getStudentByUserId` – это метод GET, который принимает в качестве параметра `id` пользователя и возвращает сущность студента, соответствующую этому пользователю. Этот метод доступен по API: `api/student/<id>`;

– Метод `saveStudent` – это метод POST, который принимает в качестве параметра `id` пользователя и в теле которого, отправляется сущность студента. По сути это метод создания студента. Он доступен по API: `api/student/<id>`;

– Метод `updateStudent` - это метод PATCH, в теле которого отправляется сущность студента и который принимает в качестве параметра `id` пользователя. По сути это метод изменяет студента. Он доступен по API: `api/student/<id>`;

– Метод `deleteStudent` - это метод DELETE, который принимает в качестве параметра `id` студента. По сути это метод удаляет сущность студента. Он доступен по API: `api/student/<id>`;

– Метод `getAllStudents` – это метод GET, который возвращает всех студентов. Он доступен по API: `api/student`.

2.3.4 Разработка клиентской части приложения

Фронтенд часть приложения написана с помощью фреймворка Angular и с использованием HTML, CSS, TS.

Архитектура этой части системы выделяет 3 основных направления: `services` отвечают за взаимодействие с бэкендом, `models` – описание сущностей, `components` – элементы интерфейса.

Каждая страница интерфейса – это отдельный элемент, для реализации которого было создано и имплементировано 3 файла: `css`, `html`, `ts`.

В рамках работы были реализованы следующие элементы:

- страница регистрации;
- страница авторизации;
- страница компании;
- страница со списком стажировок. Существует 2 вариации этой страницы в зависимости от наличия роли STUDENT, для авторизованных

пользователей с ролью студент на этой странице дополнительно присутствуют возможности поиска и фильтрации стажировок, отклика на стажировку, просмотр списка стажировок, на которые откликнулся пользователь и просмотр рекомендуемых стажировок;

– элемент добавления новой стажировки.

При авторизации в системе фронтенд сохраняет токен пользователя в `sessionStorage`. `SessionStorage` – это хранилище браузера, которое очищается в момент окончания сессии текущей страницы. Сессия страницы остаётся активной все время пока окно браузера открыто и сохраняется между перезагрузками страниц.

После авторизации в системе и момента очищения `sessionStorage` все запросы к бэкенду выполняются с токеном пользователя.

Для того, чтобы не писать добавление токена к каждому запросу был реализован интерцептор, код которого приведен ниже на рисунке 26.

```
const TOKEN_HEADER_KEY = 'Authorization';

@Injectable()
export class AuthInterceptor implements HttpInterceptor {
  constructor(private token: TokenStorageService, private router: Router) {}

  intercept(req: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<any>> {
    let authReq = req;
    const token = this.token.getToken();
    if (token !== null) {
      authReq = req.clone({headers: req.headers.set(TOKEN_HEADER_KEY, 'Bearer ' + token)});
    }
    return next.handle(authReq).pipe(catchError((err: any) => {
      if (err.status !== 403) {
        this.router.navigate(['/error'])
      }
      return throwError(err);
    }));
  }
}

export const authInterceptorProviders = [
  {provide: HTTP_INTERCEPTORS, useClass: AuthInterceptor, multi: true}
];
```

Рисунок 26 – Фрагмент кода интерцептора

Также в рамках интерцептора (ко всем запросам обрабатывает ответы) реализована обработка ошибок бэкенда: при получении любой ошибки за исключением 403 пользователь перенаправляется на отдельную страницу с сообщением, что в системе что-то пошло не так. Страница с сообщением представлена на рисунке 27.

Произошла какая-то ошибка.
Попробуйте зайти на сайт позже.



Рисунок 27 – Страница ошибки

Ошибка 403 - это отсутствие прав доступа к операции, она обрабатывается отдельно и только в определенных случаях, например, всплывает сообщение, предлагающее войти в системы для выполнения конкретных операций. Пример ошибки представлен на рисунке 28.

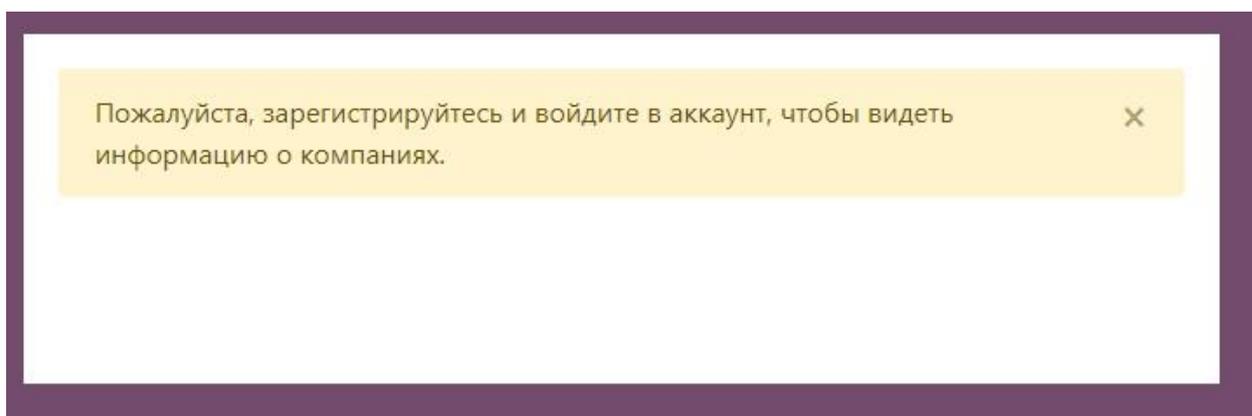


Рисунок 28 – Пример ошибки

Вывод по разделу.

В соответствии с темой выпускной квалификационной работы выбрали программное обеспечение.

Было представлено несколько видов фреймворков:

- Vue.js
- Ember.js
- Angular 2

Был выбран Angular 2, поскольку он является фреймворком с низким порогом вхождения, а из-за его популярности существует большое количество

документаций.

Разработан внешний пользовательский интерфейс. А так же расписана разработка архитектуры web-приложения.

В рамках работы были реализованы следующие элементы:

- страница регистрации
- страница авторизации
- страница компании
- страница со списком стажировок
- элемент добавления новой стажировки

3 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

3.1 Расчет затрат на разработку web-сайта

Затраты на разработку web-сайта рассчитываются по следующей формуле (1):

$$Z_{РПР} = Z_{ФОТР} + Z_{ОВФ} + Z_{ЭВМ} + Z_{СПП} + Z_{ХОИ} + P_H, \quad (1)$$

где $Z_{ФОТР}$ – общий фонд оплаты труда разработчиков программы,

$Z_{ОВФ}$ – начисления на заработную плату разработчиков программы во внебюджетные фонды,

$Z_{ЭВМ}$ – затраты, связанные с эксплуатацией техники,

$Z_{СПП}$ – затраты на специальные программные продукты, необходимые для работы с кодом,

$Z_{ХОИ}$ – затраты на хозяйственно-операционные нужды (бумага, литература, носители информации и т.п.),

P_H – накладные расходы ($P_H = 30\%$ от $Z_{ФОТР}$).

При разработке web-сайта общее время разработки составило 1 месяца.

Фонд оплаты труда за время работы над web-сайтом (2):

$$Z_{ФОТР} = \sum_{j=1}^m O_{Pj} \cdot T_{РПРj} \cdot (1 + k_D)(1 + k_Y), \quad (2)$$

где O_{Pj} – оклад j -го разработчика. В разработке участвовал 1 человек, его оклад составляет 15000 руб.,

$T_{РПРj}$ – общее время работы над программой в месяцах, $T_{РПР} = 1$,

k_D – коэффициент дополнительной зарплаты, $k_D = 20\% = 0,2$,

k_Y – районный коэффициент, $k_Y = 0,15$.

Таким образом,

$$Z_{ФОТР} = 15000 \cdot 1 \cdot (1 + 0,2) \cdot (1 + 0,15) = 20700 \text{руб.}$$

Страховой взнос во внебюджетные фонды складывается из взносов на

обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний.

Страховые взносы складываются из обязательного пенсионного страхования (ОПС), отчислений в фонд социального страхования и отчислений в фонд обязательного медицинского страхования.

Значения всех используемых ставок приведены в таблице 1.

Таблица 1 – Значения ставок страховых взносов

№	Наименование внебюджетного фонда	Размер ставок, %
1	Пенсионный фонд	22
2	Фонд социального страхования	2,9
3	Федеральный фонд обязательного медицинского страхования	5,1
	Итого:	30

Сумма начислений на заработную плату во внебюджетные фонды составляет (3):

$$Z_{\text{ОВФ}} = 0,3 \cdot Z_{\text{ФОТР}}, \quad (3)$$

$$Z_{\text{ОВФ}} = 0,3 \cdot 20700 = 6210 \text{ руб.}$$

Затраты, связанные с использованием вычислительной и оргтехники (4):

$$Z_{\text{ЭВМ}} = T_{\text{МРПР}} \cdot k_{\Gamma} \cdot n \cdot C_{\text{М-ч}}, \quad (4)$$

где k_{Γ} – коэффициент готовности ЭВМ, $k_{\Gamma} = 0,95$,

n – количество единиц техники, равно 1,

$C_{\text{М-ч}}$ – себестоимость машино–часа, $C_{\text{М-ч}} = 10$ руб.,

$T_{\text{МРПР}}$ – машинное время работы над программой, равно 1 мес.

Перевод рабочего времени в часы осуществляется по формуле (5):

$$T_{\text{час}} = T_{\text{мес}} \cdot \text{Ч}_{\text{РД}} \cdot T_{\text{см}} \cdot K_{\text{см}}, \quad (5)$$

где $T_{\text{час}}$ – рабочее время, ч,

$T_{\text{мес}}$ – рабочее время, мес, ($T_{\text{мес}} = 2$),

$Ч_{рд}$ – число рабочих дней, ($Ч_{рд} = 22$),

$Т_{см}$ – продолжительность рабочей смены, ($Т_{см} = 8$ ч),

$К_{см}$ – количество рабочих смен, ($К_{см} = 1$).

Таким образом, время на разработку web-сайта с использованием ЭВМ составляет:

$$T_{\text{час}} = 2 \cdot 22 \cdot 8 \cdot 1 = 352 \text{ часа,}$$

$$З_{\text{ЭВМ}} = 352 \cdot 0,95 \cdot 1 \cdot 10 = 3344 \text{ руб.}$$

Затраты на специальные программные продукты, необходимые для разработки web-сайта рассчитываются по формуле (6):

$$З_{\text{СПП}} = \sum_{p=1}^n Ц_p, \quad (6)$$

где $Ц_p$ – цена p -го специального программного продукта.

Перечень программных продуктов специального назначения приведен в таблице 2.

Таблица 2 – Программные продукты специального назначения

№	Название ПП	Цена, руб.
1	GitHub	0
2	Notepad++	0
3	Angular 2	0
4	PostgreSQL	0
	Итого:	0

Использованные программные продукты бесплатны, поэтому:

$$З_{\text{СПП}} = 0 \text{ руб.}$$

Затраты на хозяйственно–организационные нужды приведены в таблице 3 и вычисляются по формуле (7):

$$Z_{\text{ХОИ}} = \sum_{\tau=1}^n C_{\tau} \cdot K_{\tau}, \quad (7)$$

где C_{τ} – цена τ -го товара, руб.,

K_{τ} – количество τ -го товара.

Таблица 3 – Затраты на хозяйственно–организационные нужды

Наименование	Цена за единицу (руб.)	Кол-во (шт.)	Всего (руб.)
USB–флеш–накопитель	600	1	600
Бумага	1.2	500	600
Итого:			1200

$$Z_{\text{ХОИ}} = 600 \cdot 1 + 1.2 \cdot 500 = 1200 \text{ руб.}$$

Накладные расходы (8):

$$P_{\text{Н}} = Z_{\text{ФОТР}} \cdot k_{\text{НР}}, \quad (8)$$

$$P_{\text{Н}} = 20700 \cdot 0,3 = 6210 \text{ руб.}$$

Таким образом, затраты на разработку web-сайта рассчитанные по формуле 1 составят:

$$Z_{\text{ПП}} = 20700 + 6210 + 3344 + 0 + 1200 + 6210 = 37664 \text{ руб.}$$

3.2 Расчет затрат на внедрение web-сайта

Затраты на внедрение web-сайта рассчитываются по формуле (9):

$$Z_{\text{ВПР}} = Z_{\text{КТС}} \cdot (1 + k_{\text{ТУН}}) + Z_{\text{ПО}} + Z_{\text{ФОТВ}} + Z_{\text{ОФВ}} + Z_{\text{ЭВМ}} + P_{\text{КОМ}} + P_{\text{Н}}, \quad (9)$$

где $Z_{\text{КТС}}$ – затраты на приобретение комплекса технических средств, руб.,

$Z_{\text{ПО}}$ – затраты на приобретение программного обеспечения (включают стоимость разработанной программы, а также других существующих ПП, необходимых для функционирования системы), руб.,

$Z_{\text{ФОТВ}}$ – затраты на оплату труда работников, занятых внедрением проекта, руб.,

$Z_{\text{ОВФ}}$ – страховой взнос во внебюджетные фонды с заработной платы работников, занятых внедрением проекта, руб.,

$Z_{\text{ЭВМ}}$ – затраты, связанные с эксплуатацией ЭВМ при внедрении проектного решения, руб.,

$R_{\text{КОМ}}$ – командировочные расходы, руб.,

$R_{\text{Н}}$ – накладные расходы, руб.,

$k_{\text{ТУН}}$ – коэффициент транспортирования, установки и наладки комплекса технических средств, определяется действующими нормативами организации, а также спецификой конкретного проекта.

Дополнительного приобретения компьютеров или других КТС не требуется, следовательно, $Z_{\text{КТС}} = 0$.

Затраты на приобретение программного обеспечения в данном случае равны затратам на разработку и составляют $Z_{\text{ПО}} = 37514$ руб.

Внедрением занят один программист с окладом 15000 руб. Время внедрения – 0,5 месяц. По формуле рассчитываем затраты на оплату труда и страховой взнос во внебюджетные фонды.

$$Z_{\text{ФОТВ}} = 15000 \cdot 0,5 = 7500 \text{ руб.}$$

$$Z_{\text{ОВФ}} = 7500 \cdot 0,3 = 2250 \text{ руб.}$$

Затраты, связанные с эксплуатацией ЭВМ при внедрении проектного решения составят:

$$Z_{\text{ЭВМ}} = 0,5 \cdot 22 \cdot 8 \cdot 10 = 880 \text{ руб.}$$

Командировочные расходы при внедрении web-сайта не планируются, следовательно, $R_{\text{КОМ}} = 0$.

Так как коэффициент накладных расходов по данным организации составляет $k_{\text{НР}} = 0,3$, то величина накладных расходов равна 880 руб.

Суммарные затраты на внедрение для первой единицы проекта составят:

$$Z_{\text{ВНР}} = 0 + 37514 + 7500 + 2250 + 880 + 0 + 880 = 49024 \text{ руб.}$$

Увеличение затрат для последующей установки системы будет зависеть от количества единиц продукта и его характеристик (длина проводов, которая не включена в первичные расходы). Конечные затраты на внедрение системы составят:

$$Z_{ВПР} + \text{кол} - \text{воедиципродукта}.$$

3.3 Расчет эксплуатационных текущих затрат по web-сайту

Годовые затраты на обработку результатов до внедрения web-сайта рассчитываются по формуле (10):

$$C_1 = ЗП_1 + ОТ_{ВН1} + З_{ЭВМ1} + М_{31} + НР_1, \quad (10)$$

где $ЗП_1$ – затраты на оплату труда сотрудника на выполнение функций до внедрения проектного решения,

$ОТ_{ВН1}$ – страховой взнос во внебюджетные фонды,

$З_{ЭВМ1}$ – затраты, связанные с эксплуатацией ЭВМ,

$М_{31}$ – годовые материальные затраты на хостинг web-сайта составляют 3000 руб.,

$НР_1$ – накладные расходы.

Временные затраты работы сотрудника в месяцах рассчитываются по формуле (11):

$$T_{1мес} = \frac{T_{1час}}{Ч_{рд} \cdot Ч_{рч}}, \quad (11)$$

где $T_{1мес}$, $T_{1час}$ – время, затрачиваемое сотрудником на обработку результатов, в месяцах и часах соответственно ($T_{1час} = 564$ часов),

$Ч_{рд}$ – число рабочих дней в месяц,

$Ч_{рч}$ – число рабочих часов в день.

$$T_{1мес} = \frac{564}{22 \cdot 8} = 3,2 \text{ мес},$$

Тогда затраты на оплату труда сотрудника составят (12):

$$ЗП_1 = O_c \cdot T_{1мес} \cdot (1 + k_D) \cdot (1 + k_Y), \quad (12)$$

где O_c – оклад сотрудника (оклад составляет 15000 руб.),

$$ЗП_1 = 15000 \cdot 3,2 \cdot (1 + 0,2) \cdot (1 + 0,15) = 66240 \text{ руб.}$$

Страховой взнос до внедрения вычисляются по формуле (13):

$$OT_{ВН1} = ЗП_1 \cdot 0,3, \quad (13)$$

$$OT_{ВН1} = 66240 \cdot 0,3 = 19872 \text{ руб.}$$

Рассчитываем затраты, связанные с эксплуатацией ЭВМ до внедрения по формуле (14):

$$З_{ЭВМ1} = T_{1час} \cdot C_{М-Ч}, \quad (14)$$

$$З_{ЭВМ1} = 564 \cdot 10 = 5640 \text{ руб.}$$

Подставив соответствующие значения в формулу 10, получим:

$$C_1 = 66240 + 19872 + 5640 + 3000 = 94752 \text{ руб.}$$

Годовые затраты на эксплуатацию системы после внедрения web-сайта рассчитываются аналогично по формуле (15):

$$C_2 = ЗП_2 + OT_{ВН2} + З_{ЭВМ2} + M_{32} + НР_2, \quad (15)$$

где $ЗП_2$ – затраты на оплату труда сотрудника после внедрения,

$OT_{ВН2}$ – страховой взнос во внебюджетные фонды,

$З_{ЭВМ2}$ – затраты, связанные с эксплуатацией ЭВМ после внедрения,

M_{32} – материальные затраты, годовые материальные затраты хостинг web-сайта составляют 3000 руб.,

$НР_2$ – накладные расходы.

Временные затраты работы сотрудника в месяцах (16):

$$T_{2\text{мес}} = \frac{T_{2\text{час}}}{\text{Ч}_{\text{рд}} \cdot \text{Ч}_{\text{рч}}}, \quad (16)$$

где $T_{2\text{мес}}$, $T_{2\text{час}}$ – время, затрачиваемое сотрудником на обработку результатов, в месяцах и часах соответственно ($T_{2\text{час}} = 300$ часов),

$\text{Ч}_{\text{рд}}$ – число рабочих дней в месяц,

$\text{Ч}_{\text{рч}}$ – число рабочих часов в день.

$$T_{2\text{мес}} = \frac{300}{22 \cdot 8} = 1,70 \text{ мес.}$$

Тогда затраты на оплату труда сотрудника (17):

$$ЗП_2 = O_c \cdot T_{2\text{мес}} \cdot (1 + k_d) \cdot (1 + k_y), \quad (17)$$

где O_c – оклад сотрудника (оклад составляет 15000 руб.)

$$ЗП_2 = 15000 \cdot 1,70 \cdot (1 + 0,2) \cdot (1 + 0,15) = 35190 \text{ руб.}$$

Страховой взнос после внедрения вычисляются по формуле (18):

$$ОТ_{\text{ВН2}} = ЗП_2 \cdot 0,3, \quad (18)$$

$$ОТ_{\text{ВН2}} = 35190 \cdot 0,3 = 10557 \text{ руб.}$$

Рассчитываем затраты, связанные с эксплуатацией ЭВМ после внедрения по формуле (19):

$$З_{\text{ЭВМ2}} = T_{2\text{час}} \cdot C_{\text{М-Ч}}, \quad (19)$$

$$З_{\text{ЭВМ2}} = 300 \cdot 10 = 3000 \text{ руб.}$$

Подставив соответствующие значения в формулу, получим:

$$C_2 = 35190 + 10557 + 3000 + 3000 = 51747 \text{ руб.}$$

Таким образом, текущие затраты на содержание системы до внедрения разработанного web-сайта составляют 94752 руб., после внедрения 51747 руб.

3.4 Расчет экономической целесообразности разработки и внедрения информационных технологий

Для разрабатываемого проекта расчет экономической эффективности производится исходя из следующих условий:

- годовые текущие затраты до внедрения web-сайта, $C_1 = 94752$ руб.,
- годовые текущие затраты после внедрения web-сайта, $C_2 = 51747$ руб.,
- горизонт расчета принимается исходя из срока использования разработки, $T = T_n = 2$ годам,
- шаг расчета равен одному году, $t = 1$ году,
- капитальные вложения на создание первой единицы системы, $K = 52474$ руб.,
- норма дисконта равна норме дохода на капитал, $E = 12\%$.

Ожидаемая условно-годовая экономия от внедрения системы рассчитывается по формуле (20):

$$\mathcal{E}_{уг} = C_1 - C_2 + \sum \mathcal{E}_i, \quad (20)$$

где $\mathcal{E}_{уг}$ – ожидаемая условно-годовая экономия, руб.,

C_1 – годовые текущие затраты до внедрения web-сайта, руб.,

C_2 – годовые текущие затраты после внедрения web-сайта, руб.,

$\Sigma \mathcal{E}_i$ – ожидаемый дополнительный эффект от различных факторов, руб.

Так как основным фактором, по которому производится расчет экономической эффективности от внедрения web-сайта, является уменьшение времени обработки результатов тестирования и дополнительный эффект не учитывается, то $\Sigma \mathcal{E}_i = 0$.

Подставив вычисленные выше значения в формулу, получим:

$$\mathcal{E}_{уг} = 94752 - 51747 = 43005 \text{ руб.}$$

где $\mathcal{E}_{уг}$ – ожидаемая условно-годовая экономия, руб.

Величина ожидаемого годового экономического эффекта от внедрения ИС рассчитывается по формуле (21):

$$\mathcal{E}_r = \mathcal{E}_{уг} - K \cdot E_n, \quad (21)$$

где \mathcal{E}_r – ожидаемый годовой экономический эффект, руб.,

$\mathcal{E}_{уг}$ – ожидаемая условно-годовая экономия, руб.,

K – капитальные вложения (равны затратам на создание ИС), руб.,

E_n – нормативный коэффициент экономической эффективности капитальных вложений.

Нормативный коэффициент экономической эффективности капитальных вложений определяется по формуле (22):

$$E_n = \frac{1}{T_n}, \quad (22)$$

где T_n – нормативный срок окупаемости капитальных вложений, лет.

Подставив вычисленные выше значения в формулу, получим:

$$\mathcal{E}_r = 43005 - 49024 \cdot 0,5 = 18493 \text{ руб.}$$

Расчетный коэффициент экономической эффективности капитальных вложений составляет (23):

$$E_p = \frac{\mathcal{E}_{уг}}{K}, \quad (23)$$

где E_p – расчетный коэффициент экономической эффективности капитальных вложений,

$\mathcal{E}_{уг}$ – ожидаемая условно-годовая экономия, руб.,

K – капитальные вложения на создание системы, руб.

Подставив вычисленные выше значения в формулу, получим:

$$E_p = \frac{43005}{49024} = 0,88$$

Расчетный срок окупаемости капитальных вложений составляет (24):

$$T_p = \frac{1}{E_p}, \quad (24)$$

где E_p – коэффициент экономической эффективности капитальных вложений.

Подставив вычисленные выше значения в формулу, получим:

$$T_p = \frac{1}{0,88} = 1,4 \text{ год.}$$

Срок окупаемости первой единицы продукта без дисконтирования 1,4 года.

Чистый дисконтированный доход (ЧДД) определяется как сумма текущих эффектов за весь расчетный период, приведенная к начальному шагу, или как превышение интегральных результатов над интегральными затратами.

Если в течение расчетного периода не происходит инфляционного изменения цен или расчет производится в базовых ценах, то величина ЧДД для постоянной нормы дисконта вычисляется по формуле (25):

$$\text{ЧДД} = \sum_{t=1}^T (P_t - Z_t) \cdot \frac{1}{(1 + E)^t} - K, \quad (25)$$

где P_t – ожидаемые результаты от внедрения предложенной ИС, руб.,

Z_t – ожидаемые затраты (капитальные и текущие) на создание и эксплуатацию ИС, руб.,

$E_t = (P_t - Z_t)$ – эффект, достигаемый на t -м шаге расчета,

K – капитальные вложения,

t – номер шага расчета ($t = 1,4$),

T – горизонт расчета,

E – постоянная норма дисконта, 12%.

$E_t = (P_t - Z_t) = \text{Эуг} = 43005$ руб. В том случае, если текущие затраты (Z_t) на весь срок использования разработки равны 0.

$t = 1,4$ год, т.к. предполагается, что результат от внедрения предложенной ИС будет с текущего года внедрения ИС.

Если ЧДД инвестиционного проекта положителен, то проект является эффективным (при данной норме дисконта).

Тогда суммарный чистый дисконтированный доход за весь горизонт расчета равен (26):

$$\text{ЧДД} = \mathcal{E}_1 \cdot \frac{1}{(1 + E)} + \mathcal{E}_2 \cdot \frac{1}{(1 + E)^2}, \quad (26)$$

$$\text{ЧДД} = \frac{43005}{(1 + 0,12)} + \frac{43005}{(1 + 0,12)^2} - 49024 = 23657 \text{ руб.}$$

Положительное значение чистого дисконтированного дохода, $\text{ЧДД} > 0$, свидетельствует о том, что инвестирование целесообразно и данная ИС может приносить прибыль в установленном объеме.

Индекс доходности (ИД) представляет собой отношение суммы приведенных эффектов к величине капитальных вложений и определяется по формуле (27):

$$\text{ИД} = \frac{1}{K} \sum_{t=1}^T (P_t - Z_t) \cdot \frac{1}{(1 + E)^t}, \quad (27)$$

где K – величина капиталовложений или стоимость инвестиций.

$$\text{ИД} = \frac{72681}{49024} = 1,5$$

Инвестиции считаются эффективными, если индекс доходности выше единицы, $\text{ИД} > 1$, следовательно, инвестиции в данную ИС, эффективны.

Внутренняя норма доходности (ВНД) (28, 29):

при $E_1 \rightarrow \text{ЧДД}_1 > 0$

$E_2 \rightarrow \text{ЧДД}_2 < 0$

$$ВНД = E_1 + \frac{ЧДД_1}{ЧДД_1 - ЧДД_2} \cdot (E_2 - E_1), \quad (28)$$

$$\text{при } E_1 \rightarrow ЧДД_1 > 0$$

$$E_2 \rightarrow ЧДД_2 > 0$$

$$ВНД = E_1 + \frac{ЧДД_1}{ЧДД_1 + ЧДД_2} \cdot (E_2 - E_1) \quad (29)$$

$$E_1 = 0,11$$

$$ЧДД_1 = \frac{43005}{(1 + 0,11)} + \frac{43005}{(1 + 0,11)^2} - 49024 = 24621 \text{ руб.}$$

$$E_1 = 0,13$$

$$ЧДД_2 = \frac{43005}{(1 + 0,13)} + \frac{43005}{(1 + 0,13)^2} - 49024 = 22713 \text{ руб.}$$

$$E_1 \rightarrow ЧДД_1 > 0$$

$$E_2 \rightarrow ЧДД_2 > 0$$

$$ВНД = 0,10 + \frac{24621}{24621 + 22713} \cdot (0,13 - 0,10) = 0,116.$$

Таким образом, норма дисконта должна быть в пределах 10% – 13%.

Показатели экономической целесообразности разработки и внедрения web-сайта сведены в таблицу 5.

Таблица 4 – Показатели экономической целесообразности разработки и внедрения web-сайта в объёме одной единицы продукта

Наименование показателя	Значения
Затраты на разработку и внедрение ПП, руб.	49024
Ожидаемая экономия от внедрения ПП, руб.	43005
Чистый дисконтированный доход, руб.	23657

Индекс доходности	1,5
Внутренняя норма доходности	0,116
Дисконтированный срок окупаемости, лет	1,4
Срок морального старения, года	2

Вывод по разделу.

Произведенные расчеты свидетельствуют, что внедрение, разработанной в ВКР web-сайта, позволит сократить временные затраты на обработку результатов тестирования, что приведет к сокращению годовых текущих затрат на 43005 рублей

Опираясь на оценку экономической эффективности можно сделать вывод о том, что разработка и внедрение предлагаемого web-сайта является экономически обоснованной и целесообразной.

4 Охрана труда и промышленная экология

4.1 Анализ вредных и опасных производственных факторов на рабочем месте web-дизайнера.

Опасные и вредные производственные факторы по природе возникновения делятся на следующие группы:

- физические;
- химические;
- психофизиологические;
- биологические [15].

В помещении на программиста могут негативно действовать следующие физические факторы:

- повышенная и пониженная температура воздуха;
- чрезмерная запыленность и загазованность воздуха;
- повышенная и пониженная влажность воздуха;
- недостаточная освещенность рабочего места;
- превышающий допустимые нормы шум;
- повышенный уровень электромагнитных полей;
- повышенный уровень статического электричества;
- опасность поражения электрическим током;
- блеклость экрана дисплея [16].

Биологические вредные производственные факторы в данном помещении отсутствуют.

К психологически вредным факторам, воздействующим на программиста в течение его рабочей смены, можно отнести следующие:

- нервно – эмоциональные перегрузки;
- умственное напряжение;
- перенапряжение зрительного анализатора.

4.2 Мероприятия по оптимизации освещенности на рабочем месте

Правильное освещение рабочего места – это одна из основных составляющих успешной деятельности и хорошего самочувствия. Если не придерживаться данного положения, могут возникнуть проблемы со зрением и значительно снизится производительность труда.

Существуют разные классификации освещения. Так, по локализации оно бывает таких типов:

– общее – предполагает равномерное освещение помещения без наличия темных или более светлых зон. Присутствие только такого освещения обычно наблюдается в зонах, где рабочий процесс осуществляется неполное время.

– местное – локальная подсветка помогает дополнительно осветить определенные рабочие зоны: компьютерный или школьный стол, технику и станки. Оно предполагает установку различных осветительных приборов в непосредственной близости от рабочего места.

Применение исключительно локального освещения согласно нормам недопустимо, поскольку в помещении неизбежно будут присутствовать перепады света – от глубокого затемнения до яркого. Это вызовет проблемы со зрением работников. По источникам света освещение также делится на несколько видов:

– естественное освещение – создается силами природы: прямыми солнечными лучами, а также диффузным (отраженным) светом небосвода. Отсутствие естественного света неблагоприятно для человека, ведь именно к нему глаза лучше всего приспособлены. Такой свет зависит от времени года и периода суток, в этом его основной недостаток. Но качество и объем поступающего естественного света зависит и от конструкции здания, количества и размера окон.

– искусственное освещение – без искусственного освещения в сумеречное время суток, в пасмурный день или зимой, когда рано темнеет,

нормальный рабочий процесс невозможен. В качестве дополнительных источников света выступают лампы, светильники, торшеры, бра и прочие электроприборы. Обычно в офисы и на производство приобретают галогенные и светодиодные лампы. Обычные лампочки накаливания сейчас применяются редко, поскольку они тратят много электроэнергии, быстро выходят из строя [17].

Чаще всего освещение бывает смешанным, когда естественное сочетается с искусственным. Чтобы грамотно нормировать освещение, нужно строго следовать СанПиН и иным нормативным документам.

На рабочих местах, где отсутствует естественное освещение необходимо проведение следующих мероприятий:

- защита временем (ограниченное пребывание работника в помещении без естественного освещения менее 25% рабочей смены);
- улучшение условий, создаваемых искусственным освещением
- использование в осветительных установках общего и местного освещения источников света с коррелированной цветовой температурой от 2400К до 6500К.

При недостатке на рабочем месте естественного освещения можно выполнить следующие мероприятия:

- защита временем (ограниченное пребывание работника в помещении с недостаточным естественным освещением менее 50% рабочей смены);
- улучшение условий, создаваемых искусственным освещением (при фактическом обеспечении повышенной на ступень нормируемой освещенности и надлежащем качестве искусственного освещения);
- анализ степени загрязнения стекол в светопроемах, своевременная их чистка;
- если недостаток естественного освещения обусловлен затенением зелеными насаждениями, обеспечение сноса деревьев;
- в случае наличия в помещении зон с достаточным и недостаточным естественным освещением изменение расположения рабочих мест с их

перемещением в зону с достаточным естественным освещением;

- косметический ремонт помещения с использованием светлых отделочных материалов [18].

Требуемые уровни освещенности внутри помещений зависят от характера зрительных работ и варьируются в широких пределах. Системы внутреннего освещения объектов выполняются, как правило, светильниками с люминесцентными лампами, светодиодными светильниками, а также применяются осветительные приборы с лампами накаливания (в том числе и с галогенными). Мероприятия по обеспечению нормируемых уровней освещенности в установках внутреннего освещения в большинстве случаев сводятся к:

- установке дополнительного количества светильников;
- установке дополнительного светильника для местного освещения рабочей поверхности;
- использования маломощных и более эффективных светодиодных ламп.
- использование ламп с высоким качеством цветопередачи
- использованию источников освещения, имеющих примерно одну и ту же цветовую температуру [18].

Также существует ряд эпидемиологических документов – СанПиН, которые тоже регулируют вопрос видов освещенности, норм и основных требований к организации рабочего места. Требования СанПиН учитывают и при разработке СНиП, технической документации. СанПиН распространяется как на уже существующие промышленные и офисные помещения, так и на проектируемые, строящиеся. Санитарно-гигиенические требования к свету указаны на рисунке 29.

Характеристика Зрительной работы	Наименьший размер объекта распознавания, мм	Разряд зрительной работы	Искусственное освещение		Естественное освещение		Комбинированн ое	
			Освещенность, лк		КПО, %			
			При комбинированном освещении	При общем освещении	При верхнем или комбинированном освещении	При боковом освещении	При верхнем или комбинированном освещении	При боковом освещении
Высокой точности	0,3 – 0,5	III	2000 – 400	500 – 200	5	2	3	1,2
Средней точности	0,5 – 1,0	IV	750 – 300	300 – 150	4	1,5	2,4	0,9
Малой точности	1 – 5	V	300 – 200**	200 – 100	3	1	1,8	0,6
Общее наблюдение за ходом производственног о процесса	-	VIII	-	75* – 30	1*	0,3*	0,7*	0,2*

Рисунок 29 – Санитарно–гигиенические требования к свету

4.3 Экологические требования к утилизации вычислительной и оргтехники, а также их расходных материалов.

ИТ-оборудование утилизируется в конце его жизненного цикла. Существуют определенные законодательные требования к этой процедуре для юридических лиц и обычных граждан, так как некоторые детали компьютера представляют серьезную опасность для здоровья человека и окружающей среды.

Любая организация, которая использует в своей деятельности материальные ценности, содержащие драгоценные металлы, должна вести учет драгоценных металлов в установленном законом порядке.

Организации, не выполняющие правила использования, обращения, учета и хранения драгоценных металлов или содержащих их изделий, могут быть привлечены к административной ответственности.

Если организация не имеет лицензии на работу с вредными отходами, то для утилизации оргтехники организации необходимо привлечь специализированную организацию [20].

Большинство видов оргтехники (в том числе компьютеров) состоят из

деталей, содержащих драгоценные металлы, которые согласно нормам действующего законодательства подлежат учету.

Порядок учета установлен:

- Правилами учета и хранения драгоценных металлов..., утв. постановлением Правительства РФ от 28.09.2000 г. N 731 (далее – Правила);
- Инструкцией о порядке учета и хранения драгоценных металлов..., утв. приказом Минфина России от 09.12.2016 N 231н (далее – Инструкция).

Согласно положениям п. 6 Правил и п. 5 Инструкции организации обязаны вести учет драгоценных металлов во всех видах и состояниях, включая драгоценные металлы, входящие в состав основных средств.

Данные о наименовании, массе и количестве драгоценных металлов, содержащихся в соответствующих объектах учета, отражаются в первичной учетной документации на основании сведений, указанных в технической документации (паспортах, формулярах, этикетках, руководствах по эксплуатации, справочниках), либо при отсутствии этих сведений (импортное, устаревшее отечественное оборудование и т.п.) - по данным организаций, разработчиков, изготовителей или комиссионно на основе аналогов, расчетов (п. 8 Инструкции).

Если определить содержание драгоценных металлов невозможно из-за отсутствия данных о наличии драгоценных металлов или аналогов, в учетных документах делается запись, что в данном оборудовании могут находиться драгоценные металлы, содержание которых будет определено после списания и утилизации.

С учетом положений, указанных выше нормативных правовых актов, при списании предметов оргтехники, содержащих драгоценные металлы и их сплавы, организация должна совершить следующие действия:

- изъять из предметов оргтехники детали, содержащие драгоценные металлы;
- определить содержание драгоценных металлов в деталях;
- оприходовать полученные в результате разбора техники детали [19].

Нюансы утилизации:

Если вы выбросите некоторые части вашего компьютера на свалку, вы можете нанести непоправимый ущерб природе. В компьютерах и оргтехнике находятся олово, свинец, цинк и ртуть, которые приводят к гибели живых организмов и отравлению атмосферы, водоемов и почвы.

Некоторые детали могут содержать драгоценные и полудрагоценные металлы: серебро, золото, палладий, платину. Они и другие вещества часто используются в производстве после обработки. Физические и юридические лица могут быть привлечены к административной ответственности за незаконное изъятие старых компьютеров.

Опасность определённых деталей

В утиль сдают такую компьютерную технику, как персональные модели, кластеры, КПК, рабочие станции. Их детали могут оказывать разное влияние на окружающую среду, поэтому нельзя весь ПК отнести к одному классу опасности. Разделяют такие классификации:

- 1 класс – ртутные лампы из мониторов компьютеров;
- 2 класс – аккумуляторы и платы, содержащие олово, кадмий и свинец;
- 3 класс – провода и трансформаторы;
- 4 класс – безопасные металлические запчасти.

Пункты приема:

Обработка компьютерной техники происходит по-разному. Владелец ПК может утилизировать свой ПК следующими способами:

- обратиться в фирму, в перечень услуг которой входит утиль;
- сдать некоторые запчасти в пункты приёма металлолома;
- использовать специальную программу переработки в магазинах техники;
- продать компьютер по деталям.

Специализированные компании предлагают демонтаж и разборку ПК. Они предоставляют продавцу разрешение и паспорт отходов, заключают

соответствующие договоры, а разбором оборудования занимается квалифицированный специалист.

Хозяйственные в обмен на старые модели могут предложить существенную скидку на покупку нового ПК или сертификат определенной стоимости на покупку других товаров.

Предприятия без специального разрешения не могут заниматься вывозом и продажей техники, им необходимо заключать договоры на утилизацию ПК со специализированными фирмами. Переработка техники юридическим лицом проходит в несколько этапов:

- на первом создают внутреннюю комиссию для определения техники, которую нужно списать;
- на втором составляют заключение экспертов и подтверждение невозможности использования списанных компьютеров;
- затем технику отражают в бухгалтерском балансе;
- на заключительной стадии ПК утилизируют с помощью соответствующих компаний и получают от них подтверждающие документы.

Утиль ПК и компьютерного оборудования – это сложный и длительный процесс, но он необходим для поддержания экологической ситуации на приемлемом уровне [21].

Вывод по разделу.

В заключении хотелось бы сделать вывод, что вопрос охраны труда является одним из важнейших на современном этапе жизни нашего общества, в период, когда работодатели ставят для себя основной задачей как можно быстрее и с минимальным вложением средств извлечь наибольшее количество прибыли.

Охрана труда имеет социальное, экономическое и правовое значения. Социальное значение охраны труда состоит в том, что охрана труда способствует укреплению (сохранению) здоровья работников от вредных и опасных производственных факторов. Экономическое значение охраны труда реализуется в росте производительности труда, подъеме экономики,

увеличении производства. Правовое значение охраны труда состоит в правовом регулировании работы по способностям с учетом тяжести условий труда, физиологических особенностей организма, организма подростков и трудоспособности инвалидов.

ЗАКЛЮЧЕНИЕ

В настоящее время сайт - необходимое средство коммуникации для любой компании, для любого учреждения, человека. Каждой компании - будь то новичок на рынке или огромная корпорация - необходимо доносить до своих клиентов информацию.

Постановка задачи данного дипломного проекта включала в себя следующие основные этапы:

- анализ аналогов и технологий разработки;
- обоснование выбора проектных решений по техническому обеспечению;
- разработка техническая и программная части проекта;
- разработка внешний пользовательский интерфейс;
- рассмотрение требований по охране труда.

Формирование направлений работы с сайтом включает в себя составление плана действий, обзор материала для сайта.

Выбор инструментального средства для сайта включает в себя обзор программных средств для реализации сайта.

Web-сайт – это визитная карточка для любой организации. Информативность, дизайн – все это должно знакомить пользователя с необходимой ему информацией. Чтобы обеспечить нормальное функционирование web-сайта необходимо профессиональное техническое обслуживание.

Каждый разработчик использует те технологии создания сайтов, которые считает наиболее подходящими для себя. Однако грамотный специалист выбирает технологию в соответствии с задачами, которые перед ним возникают в процессе разработки, и которые он должен максимально эффективно решить.

Для разработки фронтенд части был выбран Angular 2, поскольку он является фреймворком с низким порогом вхождения. Он является одним из

самых популярных фреймворков, поэтому на него существует большое количество документации.

Фреймворк – каркас приложения, который состоит из набора библиотек. Вместе они дают возможность пользователю выстроить свою функциональность на некотором фундаменте. Фреймворк задаёт структуру и подход к архитектуре приложения, он диктует, как необходимо писать код

В рамках работы были реализованы следующие элементы:

- страница регистрации;
- страница авторизации;
- страница компании;
- страница со списком стажировок;
- элемент добавления новой стажировки.

Главная задача при создании информационного ресурса - при помощи визуального оформления максимально быстро и эффективно донести до посетителя суть информации о предприятии.

Подсчитано, что чистый дисконтированный доход составит 23657 рублей, а разработка окупит свои затраты в течении 1,4 года.

Следовательно, можно считать, что разработка web-сайта является экономически эффективна.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Владислав У.Р., Богуславская А.Т. Актуальность создания сайта.URL: <https://cetera.ru/about/articles/the-relevance-of-creating-the-website> (дата обращения 6.05.2022)
2. Новиков Ю.В. Современные разработки web-сайтов.–М.: 2001. URL:https://studbooks.net/2227158/informatika/sovremennye_tehnologii_razrabotki_saytov (дата обращения: 10.05.2022).
3. Рамис К.В. Классификация видов сайтов. Компоненты и технологии. 2012. №1 (126). URL: https://altblog.ru/vidy_sajtov/ (дата обращения: 12.05.2022).
4. Шевкопляс Б.В. Использование web-технологий в деятельности коммерческих предприятий.URL: https://studbooks.net/2227158/informatika/sovremennye_tehnologii_razrabotki_saytov (дата обращения 12.05.2022)
5. Калабеков Б.А.Что такое CMS.URL: <https://timeweb.com/ru/community/articles/chto-takoe-cms> (дата обращения 29.04.2022)
6. Джемилева А.Ш. Что такое Figma. URL: <https://union-z.ru/articles/dlya-chego-nuzhna-figma-saytu.html> (дата обращения 17.04.2022)
7. Таймазова С.М. Структура сайта. URL: <https://semyadro.pro/struktura>(дата обращения: 25.04.2022).
8. Алексеев Ю.М. Виды и структуры сайта. URL: <https://web-revenue.ru/seo/struktura-sayta>(дата обращения: 25.04.2022).
9. Хамзина Л.В. Разработка структуры сайта и принципы навигации. URL: <https://topuch.ru/razrabotka-strukturi-sajta-i-principa-navigacii/index.html> (дата обращения: 25.04.2022).
10. Чудинов Д.В. Что нужно знать о популярных JS-фреймворках.URL:https://habr.com/ru/company/yandex_praktikum/blog/533702/(дата обращения: 25.04.2022).

11. Белунцов В.Г. Какой фреймворк изучать JS-разработчику. URL:<https://academy.mediasoft.team/article/vue-react-ili-angular-kakoi-freimvork-izuchat-js-razrabotchiku> (дата обращения: 14.05.2022).
12. Суханкин М.Д. Обзор фреймворка Vue.js. URL: <https://timeweb.com/ru/community/articles/obzor-vue-js-1> (дата обращения: 25.04.2022).
13. БраунД.Г. Введение в Angular2. URL: <https://metanit.com/web/angular/1.1.php>(дата обращения: 25.04.2022).
14. Гончаров Л.В. Что такое Ajax. URL: <https://www.dokwork.ru/2012/07/ajax.html>(дата обращения: 25.04.2022).
15. Воронцова С.В. Все про опасные и вредные производственные факторы. URL: <https://oxrana-truda.ru/osnovnye-opasnye-i-vrednye-proizvodstvennye-factory>(дата обращения: 25.04.2022).
16. Гончарова А.Ю. БЖД программиста. URL: <https://www.turboreferat.ru/life-safety/bzhd-programmista/63340-323376-page1.html> (дата обращения: 25.04.2022).
17. Садовский В.В. Типы и нормы освещения при организации рабочего места. URL: <https://220.guru/osveshhenie/istochniki-sveta/normy-i-trebovaniya-osveshhenosti-rabochego-mesta.html>(дата обращения: 25.04.2022).
18. Панин Р.З. Как улучшить освещение рабочих мест. URL:<https://sakhgig.ru/631-kak-uluchshit-osveschenie-rabochih-mest.html>. (дата обращения: 25.04.2022).
19. Соколова О.Н. Нормы и требования к освещению рабочего места на производстве. URL:<https://mfc365.ru/nedvizhimost/normy-i-trebovaniya-k-osveshheniyu-rabochego-mesta-v-ofise-i-na-proizvodstve> (дата обращения: 25.04.2022).
20. Дунаев В.Л. Списание компьютеров в утиль. URL:<https://www.audit-it.ru/articles/account/assets/a8/975600.html>(дата обращения: 25.04.2022).

21. Компания «Гарант» Списание компьютеров в утиль. URL: <https://vtothod.ru/utilizatsiya/utilizatsiya-kompyuternoj-tehnikiikomplektuyushhih-pk> (дата обращения: 25.04.2022).

ПРИЛОЖЕНИЕ А

Техническое задание

СОГЛАСОВАНО

Преподаватель высшей категории

_____ Апталаев М.Н.
« » _____ 2022

УТВЕРЖДАЮ

Преподаватель высшей категории

_____ Апталаев М.Н.
« » _____ 2022

**WEB-САЙТ, ПРЕДНАЗНАЧЕННЫЙ ДЛЯ ВЗАИМОДЕЙСТВИЯ
СТУДЕНТОВ И МЕНЕДЖЕРОВ.**

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Разработал студент гр. КСК9-18-1СПО

_____ Тимошук Д.Б.
« » _____ 2022

Общие сведения

1. Полное наименование системы

Полное наименование: web-сайт для ООО «АНДСЕР-КСБ» предназначенного для взаимодействия студентов и менеджеров.

2. Краткое наименование системы

Краткое наименование: web-сайт.

3. Основания для проведения работ

Работа выполняется на основании учебного плана направления «09.02.01 - информатика и вычислительная техника», профиль: «Компьютерные системы и комплексы» и темы преддипломной практики, согласованной с доцент и.о. зав. кафедрой ТД ЛФ ПНИПУ, Сошиной Т.О.

4. Заказчик

Заказчик: ООО «Андсер-КСБ»

Адрес фактический: г. Лысьва ул. Металлистов 26

Телефон / Факс: +7 (34249) 3-02-86.

5. Разработчик

Разработчик: Тимошук Д.Б. КСК9-18.

Телефон / Факс: +7 9526517790.

6. Порядок оформления и предъявления заказчику результатов работ

Работы по созданию ГСЭ сдаются разработчиком поэтапно в соответствии с календарным планом проекта. По окончании каждого из этапов работ разработчик сдает заказчику соответствующие отчетные документы этапа.

Общие требования

Разработка web-сайта для ООО «АНДСЕР-КСБ», предназначенного для взаимодействия студентов и менеджеров, которые ищут стажеров (далее web-сайт) должна быть выполнена в виде программно-аппаратным комплекса, состоящего из:

1. Внешнего пользовательского интерфейса;
2. Программного обеспечения.

Назначение и цели системы

Web-сайт предназначен для создания вакансий.

Целью системы является: обеспечение необходимой информацией пользователя.

Требования к функциональным возможностям системы

- Авторизация, регистрация, аутентификация;
- Возможность добавления и удаления стажировки;
- Заполнение профиля компании и студента;
- Поиск стажировок по названию и по компании;
- Отклики от студентов;
- Подсчет откликов на стажировки.

Технические требования к web-сайту

Web-сайт должно быть сертифицирован, обеспечен гарантийными обязательствами, а также соответствовать действующим нормам и стандартам, предъявляемым к каждому типу продукции.

Требования к приемке работ по стадиям представлены в таблице 6.

Таблица 6 – Требования к приёмке работ по стадиям

Стадия испытаний	Участники испытаний	Место и срок проведения	Порядок согласования документации	Статус приемочной комиссии
Предварительные испытания	Организация заказчика и разработчика	На территории заказчика	Выявление недостатков с последующим их устранением	Экспертная группа
Опытная эксплуатация	Организация заказчика и разработчика	На территории заказчика	Проведение опытной эксплуатации	Группа тестирования
Приемочные испытания	Организация заказчика и разработчика	На территории заказчика	Проведение приемочных испытаний	Приемочная комиссия

Требования к документированию

Документация выпускной квалификационной работы будет предоставлена в соответствии с ГОСТ 7.32-2001. Требования представлены в таблице 7.

Таблица 7 – Требования к документированию

Этап	Документ
Проектирование. Разработка эскизного проекта. Разработка технического проекта.	Эскизный проект ГСЭ ТЗ для ГСЭ
Разработка рабочей документации. Адаптация программ	Описание системы Программный код системы ER-диаграмма Руководство пользователя
Ввод в действие	Акт приёмки в эксплуатацию Акт завершения работ