

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Уральский федеральный университет  
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ  
Школа бакалавриата

## ОТЧЕТ

По проекту  
«Разработка современного сайта для HR-отдела торгового дома»  
по дисциплине «Проектный практикум»

Заказчик: Якименко Е.А.

Куратор: Шестеров М.А.

Руководитель проектного практикума

Студенты команды \_\_\_\_\_

Мельников Н.С.

Тришина У.А.

Устинов Д.Н.

Шаматов Р.Р.

Юшков Ю.А.

Екатеринбург, 2025

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Аналитическая часть.....	5
1.1 Эффективность использования корпоративных сайтов в системе рекрутинга.....	5
1.1.1 Подходы к рекрутингу.....	5
1.1.2 Формирование имиджа работодателя.....	5
1.1.3 Снижение затрат и зависимости от агрегаторов.....	5
1.1.4 Повышение эффективности рекрутинга.....	5
1.1.5 Расширение охвата и SEO-преимущества.....	6
1.1.6 Вывод.....	6
1.2 Целевая аудитория .....	6
1.3 Анализ и сопоставление аналогов разрабатываемого продукта .....	7
1.3.1 Обзор аналогов .....	7
1.3.2 Doubletapp .....	7
1.3.3 НАГ .....	8
1.3.4 Реиннольц .....	9
1.3.5 Выводы и рекомендации .....	9
1.4 Разбор требований.....	10
1.4.1 Сбор требований .....	10
1.4.2 Бизнес-требования .....	11
1.4.3 Пользовательские требования .....	12
1.4.4 Операционные требования.....	13
1.4.5 Совместимость и адаптивность .....	13
1.4.6 Развёртывание и инфраструктура .....	14
1.4.7 Область ответственности и ограничения .....	14
2 Описание методологии разработки .....	15
2.1 Agile .....	15
2.2 Формирование плана разработки .....	16

2.3 Разделение ответственности .....	16
2.4 Процесс разработки .....	18
3 Обзор архитектуры программного продукта .....	20
3.1 Общее описание архитектуры .....	20
3.2 Клиентская часть приложения.....	20
3.3 Серверная часть приложения.....	22
3.3.1 Фреймворк Django .....	22
3.3.2 Модели данных .....	22
3.3.3 Зависимости.....	22
3.4 Общее функционирование .....	23
3.5 Взаимодействие между клиентской и серверной частью приложения .	24
3.5.1 Программный интерфейс .....	24
3.5.2 Процесс запроса (Клиент) .....	24
3.5.3 Процесс ответа (Сервер).....	24
3.5.4 Аутентификация и авторизация .....	25
3.6 Развёртывание веб-приложения .....	25
3.6.1 Контейнеризация.....	25
3.6.2 Docker-контейнеры .....	25
3.6.3 Развёртывание .....	26
4 Документация и инструкции по эксплуатации .....	27
ЗАКЛЮЧЕНИЕ .....	28
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	30
ПРИЛОЖЕНИЕ А (справочное) Требования заказчика .....	31
ПРИЛОЖЕНИЕ В (справочное) Карта пользовательских историй .....	33

## **ВВЕДЕНИЕ**

Целью проекта является разработка функционального прототипа HR-сайта для компании «ТД УЭТ», предназначенного для увеличения числа откликов на вакансии, привлечения квалифицированных соискателей и формирования положительного имиджа работодателя.

Прототип должен обеспечивать следующий набор функций для публичного доступа: отображение актуальных вакансий с подробным описанием, представление информации о компании, возможность отклика на вакансию. Для сотрудников компании, авторизованных в системе, предусматривается доступ к просмотру и обработке откликов, а также к редактированию информации о размещённых вакансиях.

В рамках проектной работы также предполагается разработка полного комплекта технической документации и пользовательских инструкций. Указанные материалы, наряду с исходным кодом программного продукта, будут переданы заказчику по завершении проекта.

Ввод продукта в эксплуатацию, его сопровождение и последующее обновление не входят в рамки данного проекта и осуществляются заказчиком.

Актуальность проекта определяется текущим состоянием российского рынка труда, характеризующегося исторически низким уровнем безработицы, что приводит к дефициту квалифицированных специалистов и росту конкуренции за трудовые ресурсы. В этих условиях компании вынуждены активно использовать цифровые инструменты для привлечения персонала, включая внешние кадровые платформы, агентства, а также корпоративные сайты.

Разрабатываемый программный продукт будет использоваться в качестве внутреннего HR-инструмента компании «ТД УЭТ» для автоматизации процессов подбора персонала и взаимодействия с потенциальными кандидатами, с возможностью последующей адаптации под расширенные потребности организации.

## **1 Аналитическая часть**

### **1.1 Эффективность использования корпоративных сайтов в системе рекрутинга**

#### **1.1.1 Подходы к рекрутингу**

В условиях цифровой трансформации рынка труда компании стремятся оптимизировать стратегии подбора персонала, комбинируя различные каналы привлечения кандидатов. Одним из ключевых инструментов становится корпоративный карьерный сайт, который, наряду с использованием внешних агрегаторов, обеспечивает комплексный и более управляемый подход к рекрутингу.

#### **1.1.2 Формирование имиджа работодателя**

Корпоративный сайт предоставляет компании возможность транслировать ценности, корпоративную культуру и уникальные преимущества работы, способствуя формированию положительного имиджа работодателя. Согласно исследованию CareerArc, 75 % соискателей изучают репутацию компании перед откликом на вакансию, а 69 % отказываются от подачи заявки при отсутствии положительного цифрового взаимодействия (CareerArc, 2020).

#### **1.1.3 Снижение затрат и зависимости от агрегаторов**

Публикация вакансий на внешних платформах сопряжена с высокими финансовыми затратами. Использование собственного сайта позволяет сократить бюджет на найм за счёт прямого привлечения кандидатов. Исследование Jobvite показывает, что 35 % компаний добились снижения стоимости найма благодаря использованию внутренних ресурсов.

#### **1.1.4 Повышение эффективности рекрутинга**

Кандидаты, приходящие на корпоративный сайт, как правило, проявляют более осознанный интерес. По данным аналитики SmashFly, конверсия в отклик на вакансии на корпоративных сайтах на 20–30 % выше по сравнению с внешними платформами.

### **1.1.5 Расширение охвата и SEO-преимущества**

Интеграция корпоративного сайта с агрегаторами через API и кросс-публикации позволяет расширить охват аудитории и улучшить поисковую видимость вакансий. Согласно HR.com, такие компании сокращают среднее время закрытия вакансий на 15–20 %.

### **1.1.6 Вывод**

Использование корпоративного карьерного сайта в сочетании с внешними платформами демонстрирует высокую эффективность за счёт усиления бренда работодателя, снижения затрат, увеличения конверсии и ускорения найма. Данный подход может рассматриваться как приоритетный для компаний, заинтересованных в устойчивом и контролируемом развитии HR-процессов.

## **1.2 Целевая аудитория**

Проект ориентирован на две ключевые группы пользователей: соискателей и HR-менеджеров компаний.

### **Соискатели:**

Основную пользовательскую аудиторию сайта составляют молодые специалисты до 30 лет, заинтересованные в трудоустройстве в сферах продаж промышленного оборудования, энергетики и разработки промышленного ПО. Их основная цель — найти работу, соответствующую уровню квалификации и ожиданиям. Сайт должен обеспечить доступ к актуальным вакансиям,

предоставить развернутую информацию о компании и возможность отправить резюме даже при отсутствии подходящей позиции.

### **HR-менеджеры:**

Внутренними пользователями являются сотрудники HR-отдела, ответственные за подбор персонала. Их задача — эффективно управлять вакансиями, отслеживать отклики и работать с резюме. Для этой категории важно наличие удобного интерфейса, минимальной кривой обучения и инструментов, упрощающих анализ и обработку входящих заявок.

## **1.3 Анализ и сопоставление аналогов разрабатываемого продукта**

### **1.3.1 Обзор аналогов**

Ввиду специфики проектируемого ресурса, ориентированного на выполнение функций корпоративного представительства и эффективное взаимодействие с соискателями, прямые аналоги отсутствуют. Однако для целей сравнительного анализа рассмотрены корпоративные сайты компаний, осуществляющих деятельность в смежной отрасли — в сфере поставок промышленного оборудования и разработки цифровых решений для бизнеса, что соответствует профилю деятельности ООО «ТД УЭТ».

### **1.3.2 Doubletapp**

#### **Визуальное оформление:**

Дизайн выдержан в минималистичном стиле с использованием графики, имитирующей маркерные наброски.

#### **Структура и содержание:**

Главная страница содержит краткое описание компании и блок с актуальными вакансиями. Также присутствуют разделы с корпоративными новостями и реализованными проектами. Раздел «Не нашли подходящую вакансию?» стимулирует пользователей к отправке резюме.

**Формат вакансий:**

Информация представлена стандартно: условия труда, требования к кандидатам, типовые обязанности.

**Преимущества:**

Лаконичный и приятный дизайн, наличие раздела с проектами, частично раскрывающего деятельность компании. Сайт адаптивен для мобильных устройств.

**Недостатки:**

Недостаточная информативность: не указаны график работы, формат занятости, отсутствуют данные о миссии и команде компании.

**1.3.3 НАГ****Визуальное оформление:**

Сдержанная цветовая гамма, большое количество визуальных элементов и инфографики. Информационные блоки логично структурированы.

**Структура и содержание:**

Основная информация представлена на одной прокручиваемой странице. Наиболее выразительные блоки: видеопрезентация команды, инфографика с показателями деятельности, подробный раздел о компании и её ценностях, информация о стажировках, карьерных перспективах, наградах и корпоративных мероприятиях. Реализована форма подачи резюме.

**Формат вакансий:**

Отдельная страница с фильтрами. Указаны условия труда, местоположение, заработка плата, требования и типовые задачи.

**Преимущества:**

Высокая информативность, современное оформление, акцент на ценностях и корпоративной культуре.

**Недостатки:**

Перегруженность главной страницы. Некоторые логические блоки находятся на значительном расстоянии друг от друга (например, кнопка «Отправить резюме» отделена от релевантного блока на два экрана).

### **1.3.4 Рейнольц**

#### **Визуальное оформление:**

Строгая цветовая гамма, неудачно масштабированные элементы интерфейса. Навигация и чтение информации затруднены.

#### **Структура и содержание:**

Раздел с вакансиями размещён на основном сайте компании. Присутствуют блоки с описанием истории и миссии, видеопрезентация, список руководителей с контактами, корпоративные новости и форма отправки резюме.

#### **Формат вакансий:**

Фактически отсутствует. Конкретные предложения о работе и требования к кандидатам не указаны.

#### **Преимущества:**

Создаётся впечатление надёжной крупной компании за счёт хорошо представленного раздела «О компании».

#### **Недостатки:**

Плохая техническая реализация: некорректное отображение элементов (ошибки CSS), слабая структурная организация контента. Вакансии представлены формально, без деталей, что снижает привлекательность ресурса. Сайт лучше воспринимается в мобильной версии, чем на широкоформатном мониторе.

### **1.3.5 Выводы и рекомендации**

Проведённый анализ выявил отсутствие прямых аналогов, обладающих сбалансированным функционалом и оформлением, соответствующим задачам

разрабатываемого продукта. На основе анализа конкурентных решений сформулированы следующие рекомендации:

#### **Структурная организация:**

Сайт должен иметь чёткую и логичную структуру. Важно обеспечить последовательную подачу информации о компании, её миссии, ценностях и направлениях деятельности. Необходимо включить разделы о карьерных возможностях, корпоративной культуре и процессе найма.

#### **Формат вакансий:**

Раздел с вакансиями должен быть максимально информативным. Следует указывать формат занятости, уровень заработной платы, требования к кандидатам, условия работы и типовые задачи. Желательно реализовать систему фильтрации и поиска.

#### **Визуальный стиль:**

Оптимален современный и сдержанный дизайн с акцентом на удобство восприятия информации. Следует избегать как избыточного минимализма, так и перегруженности страницы.

#### **Адаптивность и техническое качество:**

Корректное отображение на всех типах устройств и корректная работа CSS являются критически важными. Следует учитывать недочёты сайтов, подобных «Реиннольц», где технические ошибки мешают восприятию контента.

Таким образом, проектируемый ресурс должен сочетать функциональную полноту, структурную чёткость, визуальную сбалансированность и высокое качество реализации. Эти качества позволяют создать конкурентоспособный и репрезентативный инструмент взаимодействия с потенциальными сотрудниками компании «ТД УЭТ».

### **1.4 Разбор требований**

#### **1.4.1 Сбор требований**

Формирование требований к разрабатываемому программному продукту осуществлялось на основании анализа исходных данных, представленных в проектной документации (Приложение А), а также по результатам очной встречи с представителем заказчика — компании «ТД УЭТ».

Первоначальные требования были сформулированы на этапе постановки задачи и зафиксированы в описании проекта. Впоследствии, в ходе очного взаимодействия с представителем заказчика, уточнён ряд аспектов, касающихся как функционального наполнения продукта, так и особенностей визуального и структурного оформления веб-ресурса. На встрече обсуждались предпочтения в части интерфейсных решений, порядок отображения информации, а также механизмы обработки откликов от соискателей.

Итоговая версия требований представляет собой совокупность формализованных технических и нефункциональных параметров.

Особое внимание в процессе сбора требований было уделено необходимости подготовки подробной эксплуатационной документации. Представитель заказчика в устной форме обозначил ожидания в отношении состава и содержания технических материалов: документация должна включать инструкции по установке и развертыванию продукта, описание архитектуры и функциональных модулей, а также рекомендации по дальнейшему сопровождению. Это обусловлено тем, что внедрение, техническая поддержка и возможное развитие программного продукта предполагаются силами заказчика и выходят за рамки текущего проекта.

#### **1.4.2 Бизнес-требования**

Данный программный продукт разрабатывается в рамках инициативы по цифровизации внутренних HR-процессов компании «ТД УЭТ» и преследует следующие бизнес-цели:

**Увеличение количества откликов на вакансии.** Сайт должен повысить видимость открытых позиций компании и упростить процесс отклика для соискателей.

**Укрепление имиджа работодателя.** Через визуальное оформление, структуру и содержание сайт должен способствовать формированию положительного образа компании как современного и надёжного работодателя.

**Привлечение квалифицированных специалистов.** Возможность полноценной презентации условий, задач и ценностей компании должна способствовать интересу со стороны целевой аудитории кандидатов.

**Автоматизация рутинных HR-процессов.** Внедрение цифрового инструмента сократит время на обработку резюме, фильтрацию откликов и публикацию вакансий.

#### **1.4.3 Пользовательские требования**

##### **1) Для соискателей (неавторизованные пользователи)**

**Просмотр вакансий.** Главная функция сайта — демонстрация актуального перечня открытых вакансий с краткой аннотацией.

**Детальная информация о вакансиях.** Каждая позиция должна иметь отдельную страницу с подробным описанием: требования, обязанности, формат работы, уровень оплаты.

**Информация о компании.** Раздел о компании должен содержать сведения о её профиле, истории, миссии, корпоративной культуре и преимуществах работы.

**Возможность отклика.** Соискатель должен иметь возможность оставить отклик непосредственно через сайт — либо в виде формы, либо путём загрузки резюме.

**Переход на внешние ресурсы.** Ссылки на основной сайт компании и её страницы в социальных сетях.

##### **2) Для менеджеров (авторизованные сотрудники HR-отдела)**

**Просмотр откликов.** Уполномоченные сотрудники должны иметь доступ к интерфейсу просмотра поступивших откликов по каждой вакансии.

**Работа с резюме.** Возможность просмотра и скачивания загруженных кандидатами резюме.

**Управление вакансиями.** Добавление, редактирование и деактивация вакансий через административный интерфейс.

#### **1.4.4 Операционные требования**

**Надёжность и стабильность.** Сайт должен функционировать в круглосуточном режиме, с допустимым временем простоя не более одного часа в месяц (за исключением регламентных работ).

**Производительность.** Ожидается работа с умеренной нагрузкой — до 100 одновременных пользователей, включая соискателей и сотрудников HR-отдела.

**Язык интерфейса.** Вся пользовательская и административная часть реализуется на русском языке. Мультиязычная поддержка не требуется.

**Отказоустойчивость.** При сбоях необходимо обеспечить сохранность всех данных, в частности — откликов и загруженных резюме.

**Безопасность.** Хранение и передача персональных данных должны осуществляться в соответствии с требованиями законодательства РФ, включая шифрование и ограничение доступа.

**Мониторинг.** Необходимо реализовать систему логирования технических ошибок и действий пользователей для целей отладки и аудита.

#### **1.4.5 Совместимость и адаптивность**

Сайт должен корректно отображаться во всех современных браузерах: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari (на macOS и iOS), мобильные версии Chrome и Safari.

Интерфейс должен быть полностью адаптивным: корректная работа и читаемость как на широкоформатных мониторах, так и на мобильных устройствах.

#### **1.4.6 Развёртывание и инфраструктура**

**Контейнеризация.** Проект должен быть подготовлен для развёртывания в Docker-среде с документированной процедурой запуска.

**Хостинг и серверная инфраструктура.** Определяются заказчиком, на стороне разработчика — обеспечение переносимости проекта.

**Внешние зависимости.** Должны быть минимальны, за исключением критичных библиотек и API, необходимых для базовой функциональности.

**Документация.** В комплект поставки входят инструкции по установке, развертыванию и сопровождению.

#### **1.4.7 Область ответственности и ограничения**

**Ввод в эксплуатацию.** Производится заказчиком на основании предоставленных разработчиком инструкций.

**Техническая поддержка.** За пределами текущего проекта. Ответственность за поддержку, обновление и модификацию после приёмки лежит на стороне заказчика.

**Редактирование контента.** Администрирование вакансий, редактирование информации о компании и публикация новостей осуществляется заказчиком после передачи проекта.

## **2 Описание методологии разработки**

### **2.1 Agile**

Для реализации проекта была выбрана методология Kanban, относящаяся к гибкому подходу Agile. Kanban ориентирован на визуализацию процессов и непрерывную поставку результата без жёсткой привязки к спринтам, что позволяет адаптироваться к изменяющимся требованиям заказчика и получать своевременную обратную связь. Такой подход особенно эффективен в условиях неопределённости и необходимости итеративной доработки продукта.

Проектная работа была условно разделена на три последовательных этапа, соответствующих логике жизненного цикла разработки:

#### **1) Аналитический этап**

На данном этапе осуществлялся сбор и структурирование требований к системе. Были изучены цели проекта, целевая аудитория, проведён анализ аналогов и сформулированы функциональные и нефункциональные требования. Результаты были оформлены в виде документации, переданной заказчику на согласование.

#### **2) Этап разработки**

На этом этапе производилась реализация ключевой функциональности программного продукта в соответствии с утверждёнными требованиями. Разработка велась с регулярным тестированием промежуточных версий, позволяющим оперативно выявлять и устранять ошибки.

#### **3) Этап доработки и внедрения**

Заключительная стадия включала в себя устранение выявленных в процессе тестирования недочётов, подготовку пользовательской документации и инструкций по развертыванию, а также финальное тестирование. Продукт был подготовлен к передаче заказчику для последующего ввода в эксплуатацию.

Управление задачами осуществлялось с использованием Kanban-доски, что обеспечивало прозрачность текущего статуса задач и контроль выполнения работ. Для планирования и оценки сроков использовалась диаграмма Ганта, позволившая визуализировать временные зависимости и эффективно распределить ресурсы проекта.

## **2.2 Формирование плана разработки**

Для приоритизации требований и структурирования работ над проектом была использована карта пользовательских историй (Приложение Б). Этот инструмент позволил отразить функциональные потребности пользователей в виде сценариев взаимодействия с системой, сгруппированных по этапам использования и логическим блокам.

Создание карты историй позволило сформировать обоснованный перечень задач, исходя из предполагаемых пользовательских сценариев, определить минимально жизнеспособный продукт (MVP), включающий критически важную функциональность для запуска, разделить последующую разработку на этапы релизов с постепенным наращиванием возможностей системы, установить приоритеты задач и обеспечить их последовательную реализацию.

На основе карты был составлен предварительный план работ, отражающий ключевые этапы разработки. Этот план лег в основу технического и временного управления проектом, позволив организовать работу команды последовательно и в соответствии с приоритетами бизнеса.

## **2.3 Разделение ответственности**

Команда проекта включала пять участников, каждый из которых выполнял определённую роль в соответствии с компетенциями и задачами проекта. Заблаговременное распределение зон ответственности позволило

минимизировать управленческие риски, исключить дублирование усилий, неравномерную загрузку и неопределённость в вопросах ответственности.

Состав команды и роли:

– Тимлид (руководитель команды)

Отвечал за общее планирование проекта, координацию участников, контроль выполнения задач, поддержание коммуникации с заказчиком. Также занимался подготовкой проектной документации, включая материалы для сдачи контрольных точек и инструкции по эксплуатации программного продукта.

– Аналитик

Занимался сбором и формализацией требований, анализом целевой аудитории, описанием пользовательских сценариев, построением пользовательской карты, а также разработкой архитектурных решений на уровне API и баз данных. Участвовал в создании отчётной документации и контроле соблюдения логики взаимодействия компонентов системы.

– Дизайнер

Отвечал за разработку пользовательского интерфейса (UI/UX), создание макетов и интерактивных прототипов. Предлагал улучшения сценариев взаимодействия на основе принципов удобства и логичности пользовательского пути.

– Разработчик клиентской части (Frontend)

Реализовывал интерфейсную часть приложения по предоставленным макетам: вёрстка, маршрутизация страниц, реализация форм, интеграция с API. Обеспечивал адаптивность и кроссбраузерность интерфейса.

– Разработчик серверной части (Backend)

Отвечал за построение серверной логики, настройку и реализацию баз данных, проектирование API, а также подготовку окружения для развёртывания (включая контейнеризацию с использованием Docker).

Такой подход к организации команды обеспечил прозрачность процесса разработки, чёткое понимание зон ответственности и эффективное взаимодействие между участниками на всех этапах проекта.

## 2.4 Процесс разработки

Благодаря компактному составу проектной команды и применению гибких методологий управления (Agile, Kanban), удалось эффективно организовать процесс разработки, минимизировав временные и организационные издержки. Одним из ключевых факторов успеха стала возможность параллельного выполнения задач, не блокирующих друг друга.

Так, на ранних этапах проекта:

- Проектирование пользовательского интерфейса (UI/UX) велось параллельно с формализацией системных требований и проектированием архитектуры базы данных.
- Разработка отдельных модулей сайта осуществлялась одновременно с подготовкой сопроводительной документации по уже завершённым компонентам.

Такая организация труда позволила равномерно распределить загрузку между участниками команды и ускорить выход на этап интеграции.

В процессе работы использовался итеративный подход к выявлению и устранению ошибок: при обнаружении недочётов задачи на доработку оперативно фиксировались и передавались ответственному исполнителю через доску задач.

Для координации работы и актуализации задач, а также решения возникающих несостыковок, еженедельно проводились командные онлайн-созвоны. Эти встречи позволяли уточнять приоритеты, согласовывать доработки и оперативно вносить корректизы в план проекта.

В качестве системы контроля версий использовался git, что позволило эффективно управлять разработкой и модификацией программного кода.

Такой формат взаимодействия доказал свою эффективность при реализации проекта в сжатые сроки и в условиях ограниченных ресурсов.

## **3 Обзор архитектуры программного продукта**

### **3.1 Общее описание архитектуры**

Разработанное программное обеспечение представляет собой веб-приложение, построенное на основе многоуровневой, характеризующейся четким разделением функциональности на различные слои. В рамках данной архитектуры выделяются два основных уровня: клиентский уровень (frontend), отвечающий за обеспечение пользовательского интерфейса, и серверный уровень (backend), обеспечивающий обработку запросов, логику приложения и управление данными.

Клиентский уровень реализован с использованием библиотеки React, которая предназначена для создания динамических и интерактивных пользовательских интерфейсов. React позволяет формировать пользовательский интерфейс из компонентной структуры, что обеспечивает повторное использование элементов и повышает эффективность разработки.

Серверный уровень реализован с использованием фреймворка Django, который является высокоуровневой платформой для разработки веб-приложений на языке Python. Django предоставляет набор инструментов и библиотек для создания и управления базами данных, а также для реализации серверной логики.

Взаимодействие между клиентским и серверным уровнями осуществляется посредством API (Application Programming Interface), что позволяет эффективно обмениваться данными и обеспечивать бесперебойную работу веб-приложения. Выбранная архитектура обеспечивает четкое разделение ответственности между клиентской и серверной частями приложения, что способствует масштабируемости, поддерживаемости и гибкости программного продукта.

### **3.2 Клиентская часть приложения**

Клиентская часть веб-приложения реализована с использованием библиотеки React, представляющей собой JavaScript-библиотеку для построения динамических и интерактивных пользовательских интерфейсов. Для ускорения разработки и управления проектом использовался инструмент Vite в качестве сборщика модулей.

Архитектура клиентской части основывается на компонентном подходе, что позволяет создавать переиспользуемые и независимые элементы пользовательского интерфейса. Каждый компонент отвечает за отображение определенной части страницы, обеспечивая модульность и облегчая сопровождение проекта. Повторное использование компонентов способствует сокращению времени разработки и улучшению согласованности пользовательского интерфейса. Проект организован в соответствии с общепринятыми паттернами для React-приложений, что способствует лучшей читаемости и поддержке кода.

Приложение использует встроенные механизмы React для управления состоянием компонентов, обеспечивая динамическое обновление интерфейса в зависимости от действий пользователя и поступающих данных.

Реализация интерактивных элементов (например, кнопок, полей ввода, форм) обеспечивается с помощью React.

Для управления навигацией между различными страницами приложения используется библиотека react-router-dom.

Маршрутизация также включает в себя использование компонента ProtectedRoute, который обеспечивает защиту доступа к страницам в зависимости от ролей пользователей, тем самым гарантируя безопасность пользовательских данных и соответствие их функциональности.

В качестве дополнительных зависимостей использовались следующие библиотеки:

axios: для осуществления HTTP-запросов к серверной части приложения.

`jwt-decode`: для декодирования токенов JSON Web Tokens, применяемых для аутентификации и авторизации пользователей.

`lucide-react`: для использования векторных иконок.

`react-toastify`: для отображения уведомлений пользователю.

Использование данных библиотек позволяет сократить время разработки, а также повысить качество продукта.

Стилизация: для стилизации компонентов используются отдельные файлы стилей, что позволяет отделить логику компонентов от их визуального представления.

### **3.3 Серверная часть приложения**

#### **3.3.1 Фреймворк Django**

Серверная часть веб-приложения реализована с использованием фреймворка Django, который является высокоуровневым веб-фреймворком на языке Python. Django предоставляет набор инструментов и библиотек для построения сложных веб-приложений, включая управление данными, обработку запросов, аутентификацию и авторизацию пользователей, а также интеграцию с внешними сервисами.

#### **3.3.2 Модели данных**

В основе серверной части лежат модели данных, которые определяют структуру и отношения между различными сущностями в системе. Django ORM (Object-Relational Mapper) используется для представления данных и взаимодействия с базой данных PostgreSQL. Каждая модель представляет собой таблицу в базе данных, а ее поля — столбцы.

Модели Django используются для представления и манипуляции данными в системе.

#### **3.3.3 Зависимости**

Для корректной работы приложения используются следующие дополнительные библиотеки и фреймворки:

Django и Django REST framework: основные фреймворки для разработки бэкенда, предоставляющие инструменты для работы с данными и создания API.

Django filters: библиотека для создания API фильтров, позволяющая фильтровать данные по разным параметрам.

Django CORS headers: обеспечивает возможность междоменных запросов, необходимых для взаимодействия с клиентской частью.

Psycopg2: адаптер для взаимодействия с базой данных PostgreSQL.

PyJWT: библиотека для реализации аутентификации и авторизации с использованием JSON Web Tokens.

### **3.4 Общее функционирование**

Серверная часть веб-приложения функционирует как RESTfull API, предоставляя доступ к данным и ресурсам через HTTP-запросы. Django REST framework упрощает создание API, преобразуя модели данных в JSON-формат для обмена с клиентской частью.

При получении HTTP-запроса, Django обрабатывает его, основываясь на настроенных URL-адресах, и направляет запрос в соответствующее представление (view). Представления (views) выполняют бизнес-логику приложения, запрашивают данные из базы данных, обрабатывают их и формируют ответы.

Для аутентификации и авторизации пользователей используются JWT (JSON Web Tokens). Пользователи, успешно прошедшие аутентификацию, получают токен доступа, который они могут использовать для доступа к защищенным ресурсам.

Связь с базой данных PostgreSQL происходит через Django ORM (Object-Relational Mapper), что упрощает взаимодействие с данными. ORM позволяет работать с данными, не используя напрямую SQL запросы.

Выбор PostgreSQL обусловлен ее надежностью, производительностью и поддержкой широкого спектра возможностей, необходимых для эффективной работы приложения.

### **3.5 Взаимодействие между клиентской и серверной частью приложения**

#### **3.5.1 Программный интерфейс**

Взаимодействие между клиентской частью (React) и серверной частью (Django) осуществляется через RESTfull API, который предоставляется Django. Клиентская часть отправляет HTTP-запросы (например, GET, POST, PATCH, DELETE) к API бэкенда для получения и отправки данных. Бэкенд обрабатывает эти запросы, выполняет бизнес-логику, взаимодействует с базой данных и возвращает ответы в формате JSON.

#### **3.5.2 Процесс запроса (Клиент)**

Axios Instance: Для упрощения отправки HTTP-запросов к API используется библиотека axios. Создан собственный экземпляр axiosInstance, который автоматически добавляет токен доступа в заголовки каждого запроса, если он сохранен в локальном хранилище клиента.

Запросы к API: Клиентская часть отправляет HTTP-запросы к различным URL-адресам, предоставляемым бэкендом. Эти запросы могут быть запросами на получение данных (GET), создание новых записей (POST), изменение существующих (PATCH) или удаление данных (DELETE).

Обработка ответов: после отправки запроса клиентская часть получает ответ от бэкенда, который содержит данные в формате JSON. React-компоненты используют эти данные для отображения информации пользователю или выполнения других операций.

#### **3.5.3 Процесс ответа (Сервер)**

**URL-маршрутизация:** Django URL-маршрутизация определяет, какие представления должны обрабатывать запросы к определенным URL-адресам.

**Представления:** Представления получают HTTP-запрос, выполняют необходимую логику (например, обращаются к базе данных через Django ORM), и возвращают ответ, который может содержать данные в формате JSON или статус код.

**Сериализация данных:** для преобразования данных из формата Django-моделей в формат JSON используются Django REST framework serializers.

### **3.5.4 Аутентификация и авторизация**

**Токены доступа:** для аутентификации и авторизации используются JSON Web Tokens (JWT). При успешной аутентификации бэкенд возвращает токены доступа и обновления (access и refresh).

**Interceptor:** axiosInstance использует interceptor для добавления access токена в каждый запрос. Если бэкенд возвращает ошибку 401 (не авторизован), то interceptor пытается обновить access токен с помощью refresh токена. Если обновление не удалось, то пользователя перенаправляют на страницу авторизации.

## **3.6 Развёртывание веб-приложения**

### **3.6.1 Контейнеризация**

Для обеспечения воспроизводимого и надежного развертывания веб-приложения на сервере используется технология контейнеризации Docker. Docker позволяет упаковать приложение со всеми его зависимостями в стандартизованный контейнер, который можно запускать в любом окружении, где установлен Docker.

### **3.6.2 Docker-контейнеры**

Docker-контейнер представляет собой изолированную среду, содержащую все необходимые для работы приложения компоненты, включая код приложения, библиотеки, зависимости и системные утилиты.

Контейнеры обеспечивают изоляцию приложения от хост-системы, что предотвращает конфликты между различными приложениями и гарантирует стабильность работы.

Благодаря контейнеризации приложение становится портативным и может быть легко развернуто на различных платформах, где установлен Docker.

Для сборки Docker-образа используется Dockerfile, который представляет собой текстовый файл, содержащий инструкции для создания образа.

Dockerfile содержит инструкции по установке зависимостей, копированию файлов приложения и настройке окружения.

### **3.6.3 Развёртывание**

Для развертывания приложения на сервере используется Docker-образ, который запускается в контейнере.

Управление контейнерами осуществляется с помощью Docker CLI или Docker Compose.

Использование Docker упрощает процесс развертывания, обеспечивает стабильную работу приложения и позволяет масштабировать его при необходимости. Также Docker упрощает процесс развертывания, сокращая количество ручных операций и снижая вероятность ошибок.

## **4 Документация и инструкции по эксплуатации**

### **4.1 Состав документации**

Заказчику переданы следующие документы: инструкция по эксплуатации, инструкции по внедрению, техническая документация продукта.

### **4.2 Инструкция по эксплуатации**

Данный документ описывает все функции программного продукта, их расположение и способы доступа. Ориентирована на hr-менеджеров и администраторов. Включает в себя информацию о управлении откликами, вакансиями, новостями и учетными записями пользователей.

### **4.3 Инструкция по внедрению**

Описывает процесс развертывания приложения на серверах. Ввиду использования контейнеризации достаточно краткая. Также включает в себя создание супер-пользователей, для дальнейшего управления учетными записями.

### **4.4 Техническая документация**

Включает в расширенное описание архитектуры программного продукта, включая описание классов и модулей, ответственных за основные модули функциональности, структуры проектов клиентской и серверной части, подробную документацию API в формате OpenApi 3.1.

## **ЗАКЛЮЧЕНИЕ**

В ходе реализации проекта был разработан программный продукт, соответствующий ключевым бизнес-целям и пользовательским ожиданиям, сформулированным на этапе сбора требований. Продукт выполняет основную функцию — служит цифровым представительством компании «ТД УЭТ», ориентированным на взаимодействие с соискателями. Созданная система обеспечивает удобный доступ к вакансиям, информирует о деятельности компании и позволяет пользователям легко откликаться на предложения о работе, что полностью отражает поставленные задачи по повышению вовлеченности соискателей и автоматизации части процессов HR-отдела.

Система реализует все базовые пользовательские сценарии: просмотр информации о компании и вакансиях, фильтрация и отклик на вакансии, а также подача резюме. Менеджеры получают доступ к откликам и резюме, имеют возможность редактировать открытые вакансии. Таким образом, функциональные требования заказчика были выполнены в полном объёме. Также учтены операционные и системные требования — сайт работает стабильно, адаптирован под различные устройства и совместим с современными браузерами.

По результатам внутреннего тестирования можно заключить, что программный продукт обладает достаточной стабильностью, корректно выполняет свои функции и демонстрирует высокую степень готовности к эксплуатации. Выявленные в ходе тестирования ошибки относились к незначительным: в основном — визуальным несоответствиям, мелким проблемам в отображении на разных устройствах или незначительным логическим сбоям в отдельных пользовательских сценариях. Все обнаруженные дефекты были оперативно устранены. На момент завершения проекта критических или блокирующих багов не зафиксировано.

Тем не менее, стоит отметить, что в дальнейшем потребуется наблюдение за поведением системы в реальной эксплуатации, особенно в условиях растущей нагрузки или масштабирования пользовательской базы.

Разработанный продукт представляет собой функциональное и расширяемое решение, которое может служить основой для дальнейшего развития. Потенциальные направления совершенствования включают:

- Интеграцию с системами внутреннего документооборота или CRM для автоматического отслеживания откликов;
- Расширение аналитических функций — добавление отчётности по просмотрам, источникам трафика, эффективности вакансий;
- Внедрение чат-бота для первичного взаимодействия с кандидатами и автоматического ответа на часто задаваемые вопросы.

Также рекомендуется предусмотреть периодическую актуализацию контента и интерфейса в соответствии с изменениями в структуре компании и актуальными трендами в дизайне пользовательских интерфейсов.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Капелюшников Р.И. Российский рынок труда: тенденции, институты, структурные изменения // Центр стратегических разработок. – 2017. – URL: [https://www.csr.ru/uploads/2017/03/Doklad\\_trud.pdf](https://www.csr.ru/uploads/2017/03/Doklad_trud.pdf) (дата обращения: 15.05.2025).
2. CareerArc. Future of Recruiting Study. – 2020. – URL: <https://www.careerarc.com> (дата обращения: 16.05.2025).
3. Jobvite. Recruiter Nation Report. – 2022. – URL: <https://www.jobvite.com> (дата обращения: 16.05.2025).
4. SmashFly. The Source of Hire Report. – 2021. – URL: <https://www.smashfly.com> (дата обращения: 16.05.2025).
5. HR.com. The State of Talent Acquisition Technology. – 2023. – URL: <https://www.hr.com> (дата обращения: 16.05.2025).

## **ПРИЛОЖЕНИЕ А**

### **(справочное)**

#### **Требования заказчика**

Проект должен стать примером инновационного подхода в привлечении молодых специалистов и продемонстрировать возможности современной веб-разработки в рамках HR-коммуникаций торгового дома.

**Концепция и дизайн:**

Сайт должен иметь современный, минималистичный и технологичный дизайн с акцентом на визуальную привлекательность для молодой аудитории.

Использовать современные технологии и тенденции в веб-дизайне (например, адаптивная верстка, интерактивные элементы, плавные анимации).

**Структура и функционал:**

Главная страница: Краткое описание компании, миссии и ценностей, а также акцент на возможности карьерного роста.

Страница вакансий: Интерактивный список актуальных вакансий с фильтрами (по направлениям, уровню опыта, локации и т.д.). Возможность перехода к подробному описанию вакансии.

Страница компании: Информация о компании, корпоративной культуре, преимуществах работы и успехах.

Контактная информация: Форма обратной связи, контактные данные, карта расположения офиса.

Дополнительные разделы (оциально): Блог или раздел новостей, отзывы сотрудников, видеointервью.

**Технические требования:**

Адаптивная верстка для корректного отображения на всех устройствах (ПК, планшеты, смартфоны).

Кроссбраузерная совместимость (основные современные браузеры).

Оптимизация скорости загрузки страниц.

Интеграция с социальными сетями для легкого распространения контента.

Технологический стек:

Использование HTML5, CSS3, JavaScript (библиотеки/фреймворки по выбору студентов – например, React, Vue.js или аналогичные).

Возможное использование CMS или фреймворка для ускорения разработки (по согласованию с руководителем проекта).

Уровни результата (минимальный / базовый / оптимальный):

Минимальный: рабочий прототип с базовыми модулями (вакансии, контакты, информация о компании).

Базовый: адаптивный дизайн, оптимизированная загрузка, интерактивные элементы.

Оптимальный: полная документация, презентация, интеграция с социальными сетями, дополнительные разделы (блог, отзывы).

Рабочий прототип: Полнofункциональный сайт с реализованным дизайном, структурой и основными функциональными модулями.

Исходный код: Полностью документированный исходный код, готовый для дальнейшей поддержки и масштабирования.

Дизайн-макеты: Готовые макеты всех ключевых страниц (можно в виде файлов Figma, Adobe XD или аналогичных инструментов).

Инструкция по установке и эксплуатации: Документация для развертывания сайта, описание используемых технологий и рекомендаций по дальнейшей поддержке.

Презентация проекта: Подготовка презентации, демонстрирующей ключевые особенности сайта, его функциональность и преимущества для HR-отдела.

# ПРИЛОЖЕНИЕ В

## (справочное)

### Карта пользовательских историй

