

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  
«Динамическая модель компетенций»  
по дисциплине «Проектный практикум»

Заказчик: Шадрин Д.Б.

Куратор: Шадрин Д.Б.

Старший преподаватель

Студенты команды Diamond Dogs

Бушуев А.О.

Сорокин Ю.П.

Яковлев А.И.

Черников А.А.

Старцев В.И.

---

---

---

---

---

---

---

---

Екатеринбург, 2025

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 Описание проекта.....	6
1.1 Описание проблемы.....	6
1.2 Анализ целевой аудитории .....	7
1.3 Анализ аналогов .....	10
1.4 Архитектура.....	14
2 Основная часть .....	17
2.1 Планирование работы.....	17
2.2 Отчет по работе Бушуева Арсения Олеговича .....	18
2.3 Отчет по работе Яковлева Алексея Ильича .....	19
2.4 Отчет по работе Старцева Владислава Игоревича .....	25
2.5 Отчет по работе Черникова Александра Алексеевича.....	27
2.6 Отчет по работе Сорокина Юрия Павловича.....	29
ЗАКЛЮЧЕНИЕ .....	32
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	33

## ВВЕДЕНИЕ

IT – сфера является самой популярной сферой деятельности за последние несколько десятков лет, благодаря своей востребованности и перспективности. Однако у данной сферы до сих пор нет качественной, структурированной и регулярно обновляемой статистики по вакансиям в сфере IT. Эта информация была бы полезна трём целевым группам.

Руководителям образовательных IT программ критически важно знать тенденции рынка труда в IT сфере, чтобы изменять учебную программу таким образом, чтобы обучать нужных рынку специалистов. В этом им поможет регулярно обновляемая статистика по рынку IT вакансий.

Для абитуриентов, студентов, обучающихся преимущественно в технических институтах, и людей, желающих найти себе профессию в IT будет полезно знать статистическую информацию о рынке труда в IT, чтобы, опираясь на неё, выбрать себе профессию.

HR – специалисты. Для грамотного и правильного ведения кадровой политики, планирования бюджета, требуется источник, отражающий полноценную актуальную статистику динамики количества и доли вакансий, динамику средних зарплат и информацию о востребованности навыков в IT.

Решить заявленную проблему с отсутствием качественной, структурированной и регулярно обновляемой статистикой можно, собрав данные по вакансиям в IT-сфере для каждой IT-профессии, и представив статистическую информацию с помощью графиков, дополняя статистическую информацию – общей информацией о профессиях.

Исходя из вышеописанного, трём целевым группам, необходим удобный источник информации, который включал бы в себя общую, а также статистическую информацию о профессиях, основанную на анализе IT вакансий с популярных порталов по поиску работы в России.

Данный проект уже находится в стадии реализации. В прошлых семестрах команды провели работы по представлению на веб-сайте «Найти в IT»[1] актуальной, структурированной, регулярно обновляемой статистике по рынку IT вакансий. Основной представленной на сайте аналитикой является: динамика доли и количества вакансий, динамика заработной платы и необходимые навыки для каждой профессии на рынке IT вакансий.

Также для решения проблемы целевой аудитории на сайте были разработаны следующие разделы: «Интерфейс сравнения профессий», позволяющий сравнивать профессии между собой по ключевым параметрам; «Навыки» – раздел со статистикой по динамике востребованности навыка для профессий; «Статьи» – аналитические статьи по статистике, представленной на нашем сайте; «Конструктор графиков» – настраиваемый график, позволяющего индивидуально формировать визуализации наших статистических данных с учетом различных параметров и зависимостей.

В целях помощи в комплексном решении проблем и улучшения ценности проекта для целевой аудитории, было определено, что необходимо провести:

- создание новой модели для кластеризации вакансий, основанной на технологии машинного обучения, чтобы повысить качество обработки данных и тем самым улучшить статистику для пользователей;

- Модернизация архитектуры системы сбора, классификации и аналитики вакансий для повышения масштабируемости, гибкости и расширяемости за счёт перехода к микросервисной, контейнеризованной инфраструктуре.

- Создать современный, легко масштабируемый бэкенд для профилированного теста, полностью совместимый с существующим Vue-фронтом и готовый к работе как в локальной, так и в продакшн-среде.

- Повысить скорость и прозрачность разработки за счёт детализированной документации API и процессов развёртывания;

- Обеспечить воспроизводимую, отказоустойчивую инфраструктуру через стандартизированные Docker-контейнеры;

- разработка нового user-friendly дизайна и модернизация фронтенд архитектуры на модульную, уменьшение бloatа в джаваскрипте, а также множество quality-of-life изменений для облегчения дальнейшей работы по поддержанию проекта;

- Внедрение нового интерактивного сервиса для помощи с профориентацией посредством решения тестов.

Актуальность данной работы заключается в том, чтобы помочь: руководителям образовательных IT программ составить структуру и содержание образовательной программы для обучения востребованным профессиям в IT, студентам и абитуриентам выбрать профессию из сферы информационных технологий, ознакомиться с требуемыми навыками, понять особенности рынка труда для профессий из сферы информационных технологий, HR-специалистам помочь предлагать конкурентные условия и выстраивать эффективную кадровую стратегию с помощью веб-приложения, развёрнутого в сети Интернет, на котором будет содержаться регулярно обновляемая и качественно проанализированная информация по вакансиям профессий сферы IT с 2003 года по настоящее время совместно с общей информации о профессиях.

Целью работы является разработка и внедрения нового функционала на веб-сайт «Найти в IT», с помощью которого целевая аудитория сможет эффективнее решить заявленные ранее проблемы.

Исходя из поставленной цели, в работе определены следующие задачи:

- провести анализ существующих решений проблем целевой аудитории;
- провести анализ текущей версии системы, для определения возможностей по внедрению нового функционала;
- разработать требования к новому функционалу системы;
- разработать и внедрить новый функционал на веб-сайт «Найти в IT».

## **1 Описание проекта**

### **1.1 Описание проблемы**

Основная проблема — это отсутствие полноценной, структурированной, регулярно обновляемой статистики по рынку IT вакансий. Данная проблема раскрывается у трех целевых групп, а именно у руководителей образовательных программ в сфере информационных технологий, студентов, абитуриентов, обучающиеся в технических институтах, которые желают стать IT специалистом и HR-специалистов. Данную проблему мы решили, в рамках работы по дисциплине «Проектный практикум» в предыдущих семестрах. В ходе работы над проектом были выявлены аспекты, требующие улучшения для повышения эффективности сервиса и его ценности для целевой аудитории. Эти аспекты определили направления предстоящих изменений в этом семестре:

- необходимость повышения качества обработки данных для формирования статистики, предоставляемой пользователям, чтобы сделать её более точной и полезной. Планируется разработка новой модели кластеризации вакансий на основе технологий машинного обучения;

- Исходная реализация основанная на CSV-файлах, что затрудняло запуск, конфигурацию и расширение системы. Добавление новых источников вакансий и интеграция с аналитикой являются очень затруднительными из-за монолитной структуры и отсутствия автоматизации. Это препятствует масштабированию и снижает эффективность всей платформы.

- Отсутствие необходимых сервисов для внедрения интеллектуального классификатора вакансий, невозможность быстрого разворачивания системы на новой машине. Планируется модернизация архитектуры на новую, с независимыми сервисами, упакованными по докер-контейнерам

– Перегруженный избыточным JS-кодом фронтенд, имеющий монолитную структуру и устаревший дизайн, что затрудняет его развитие, сопровождение и в целом негативно сказывается на пользовательском опыте. Не хватает архитектурной гибкости и пользовательской доработки.

– Отсутствие единой, удобочитаемой спецификации API и README приводит к задержкам при подключении разработчиков и увеличивало риск ошибок при интеграции;

– Ручное поднятие окружений затягивает разработку и создавало расхождения между локальной и боевой конфигурацией;

– Нехватка автоматических тестов снижает уверенность в стабильности критичных функций (аутентификация, работа с БД);

Исходя из этого, можно выделить третий уровень боли – личный интерес: Третий уровень – это наличие личной мотивации клиента на решение проблемы. Пользователь может получить существенную экономию времени и сохранить психическое здоровье.

## **1.2 Анализ целевой аудитории**

Для создания продукта, максимально соответствующего запросам целевой аудитории, необходимо чётко понимать её потребности и особенности. В рамках исследования была использована методика 5W Марка Шеррингтона для анализа и определения ключевых сегментов целевой аудитории.

Методика 5W предполагает ответ на следующие вопросы:

- 1) кто? – Кто входит в состав целевой аудитории;
- 2) что? – Какие задачи они хотят решить;
- 3) почему? – Почему эти задачи важны;
- 4) когда? – В какие моменты возникает необходимость в продукте;
- 5) где? – Где и как они будут использовать продукт.

Приведем результаты анализа по методике 5W для описанных целевых групп:

1. Руководители образовательных IT программ

Кто?

Это специалисты, ответственные за разработку и актуализацию учебных программ в образовательных учреждениях. Их главная задача — подготовка студентов, которые будут соответствовать требованиям рынка труда в IT на момент выпуска из образовательного учреждения.

Что?

Им необходимы актуальные данные о тенденциях рынка труда, включая востребованные навыки, технологии и направления в IT, чтобы корректировать содержание образовательных программ.

Почему?

Своевременное обновление учебных планов позволяет готовить выпускников, обладающих востребованными на рынке знаниями. Это повышает их конкурентоспособность и трудоустройство.

Когда?

Руководители обращаются к сервису в процессе обновления учебных планов, разработки новых курсов или специализаций, что обеспечивает их соответствие требованиям современного рынка.

Где?

Для удобства взаимодействия с сервисом необходимо обеспечить доступ через веб-платформу, доступную с различных устройств: компьютеров, планшетов, телефонов.

2. Абитуриенты, студенты технических вузов и люди, желающие получить IT-профессию.

Кто?

Эта группа включает тех, кто выбирает профессию, обучается техническим дисциплинам, или планирует смену карьеры для перехода в IT.



Что?

Их интересует информация о востребованных специальностях, навыках, перспективах карьеры и уровне оплаты труда, чтобы правильно выбрать направление для развития.

Почему?

Достоверная информация позволяет делать осознанный выбор учебных программ, курсов переквалификации и карьерных путей, что способствует успешному трудоустройству.

Когда?

Абитуриенты используют сервис при выборе учебного заведения и программы, студенты — для корректировки планов обучения, а люди, меняющие профессию, — на этапе планирования переквалификации.

Где?

Для удобства пользователей сервис должен быть доступен через веб-платформу, поддерживающую все популярные устройства.

### 3. HR-специалисты

Кто?

HR-менеджеры занимаются подбором кадров, управлением персоналом и формированием эффективной команды в IT-компаниях.

Что?

Им нужны данные о зарплатах, востребованных навыках и тенденциях на рынке труда, чтобы планировать кадровую политику и формировать конкурентоспособные предложения для сотрудников.

Почему?

Актуальная информация помогает привлекать квалифицированных специалистов, удерживать сотрудников и грамотно планировать бюджет на оплату труда.

Когда?

Сервис используется для регулярного обновления данных о рынке труда, мониторинга тенденций и корректировки кадровой стратегии.

Где?

HR-специалистам требуется удобный доступ к платформе через веб-приложение, совместимое с различными устройствами.

Методика 5W Марка Шеррингтона позволила выделить три основные группы целевой аудитории: руководители образовательных программ, абитуриенты и студенты, а также HR-специалисты. Обеспечение этих групп актуальной информацией о рынке труда способствует улучшению качества образовательных программ, осознанному выбору профессий и оптимизации кадрового управления в IT-компаниях.

### **1.3 Анализ аналогов**

На данный момент на рынке представлены разнообразные онлайн-сервисы, предоставляющие статистическую информацию о состоянии IT-сферы в России. Среди наиболее известных можно отметить:

- «Навр.Карьера» – цифровая платформа [2];
- «Зарплатомер» – веб-ресурс [3];
- «Роснавык» – интернет-сервис [4].

«Навр.Карьера» представляет собой цифровую платформу, предоставляющую актуальную информацию о зарплатах в IT-сфере. Этот ресурс позволяет пользователям ознакомиться с текущими уровнями оплаты труда для различных специальностей в IT-индустрии, обновляя данные ежедневно. Статистика формируется на основе анонимных данных, предоставляемых самими специалистами. На момент проведения анализа база данных ресурса содержала сведения из 56 832 анкет.

Информация разделена по уровням профессиональной квалификации: Lead, Senior, Middle, Junior и Intern, с указанием зарплат для каждого из них.

Платформа предлагает расширенные инструменты фильтрации, которые позволяют уточнять данные по региону, специализации, ключевым навыкам, компаниям, типу занятости и формату работы.

Преимущества:

- возможность получения данных о зарплатах с учетом уровня квалификации;
- наличие расширенных фильтров для поиска информации: по местоположению, специализации, навыкам, компаниям, типу занятости и формату работы;
- бесплатный доступ, требующий минимальных действий — регистрации и указания информации о своей зарплате;
- применение перцентильного метода для отображения статистики;
- автоматическое обновление данных.

Недостатки:

- поскольку информация предоставляется пользователями, есть вероятность внесения некорректных данных;
- относительно небольшая выборка данных для формирования статистики;
- сервис ограничен исключительно данными о зарплатах, без предоставления других показателей.

«Зарплатомер» от компании «SuperJob» — это электронный ресурс, предоставляющий подробные отчеты по различным профессиям, включая ИТ-сферу. В рамках платного отчета сервис предлагает несколько ключевых данных. Во-первых, отображаются зарплатные индексы «SuperJob», которые демонстрируют ежемесячную динамику изменения средних заработных плат в различных сферах, включая ИТ. Во-вторых, в отчете содержится информация о профессии, включая подробное описание профессиональных обязанностей, требований к кандидатам и портрет идеального соискателя для каждой роли.

Этот подход помогает не только получить представление о текущих зарплатах, но и понять, какие компетенции и навыки наиболее востребованы на рынке.

Тем не менее, стоит отметить, что обновления информации по IT-отрасли происходят сравнительно редко, всего раз в квартал. Это может оказывать влияние на актуальность предоставляемых данных, особенно в условиях быстро изменяющегося рынка труда в IT, где тенденции и требования к специалистам могут существенно меняться в короткие сроки. Такой временной промежуток между обновлениями может ограничивать точность прогнозирования текущих условий на рынке труда, что делает ресурс менее гибким для пользователей, которым необходима более оперативная информация.

Преимущества:

- предоставление детализированных и структурированных отчетов по профессиям, включая IT;
- наличие зарплатных индексов, которые отображают ежемесячную динамику изменения заработных плат;
- предоставление обширной информации о профессиях, включая описание обязанностей, требований и портреты соискателей.

Недостатки:

- обновление данных лишь раз в квартал, что снижает актуальность информации в условиях быстро меняющегося рынка труда;
- высокая стоимость отчетов, которая составляет в среднем около 160 000 рублей, что может быть значительным барьером для большинства пользователей.

«Роснавык» — это интернет-сервис, предназначенный для анализа рынка труда и востребованных навыков, с целью создания образовательных продуктов. Платформа предоставляет данные по всем профессиям в стране, включая информацию о вакансиях, зарплатах, навыках, опыте работы и типах занятости. Анализируется информация с таких источников, как «Работа России», «SuperJob», «hh.ru» и «Работа.ру».

Платформа предлагает удобный дашборд для управления метриками, однако доступ к статистике в бесплатной версии ограничен. Условия подписки и стоимость услуг оговариваются индивидуально. Стоимость выгрузки данных составляет 25 000 рублей, аналитики для образовательных продуктов — 100 000 рублей, а интерактивные дашборды и отчеты — по согласованию.

#### Преимущества:

- обширная выборка данных и множество метрик, охватывающих широкий спектр профессий;
- подробная информация по регионам, специальностям, вакансиям, зарплатам, навыкам, количеству вакансий, опыту работы, типу занятости, графику работы, статусам вакансий, компаниям и отраслям;
- интуитивно понятный дашборд для удобного управления метриками и аналитикой;
- возможность получения комплексной аналитики, что помогает создавать качественные образовательные продукты.

#### Недостатки:

- отсутствие углубленной аналитики, специально ориентированной на IT-отрасль;
- не хватает детализации вакансий, относящихся к конкретным IT профессиям;
- наличие дубликатов вакансий, что может исказить статистику;
- ограничения на фильтры для запросов и доступность статистики только за последние 3 месяца.

Анализ сервисов, предоставляющих статистику по рынку труда в IT, выявил их сильные и слабые стороны. Nabr Карьера предлагает актуальные данные и удобство использования, но ограничена в метриках. Зарплатомер (SuperJob) предоставляет детализированные отчеты и хорошую специализацию на IT, но имеет высокую цену и редкие обновления. Роснавык предлагает множество метрик, но недостаточно ориентирован на IT. Эти выводы будут

использованы при разработке сервиса, который объединит сильные стороны существующих решений и устранил их недостатки, обеспечивая полную и актуальную информацию о рынке труда в IT.

## 1.4 Архитектура

Архитектура системы — это основа, которая определяет функциональность, масштабируемость и гибкость программного продукта. В этом разделе будет представлена структура разрабатываемого сервиса, описаны ключевые модули и их взаимодействие, а также их роль в обеспечении эффективной работы всей системы. Архитектура представлена на Рисунке 1.

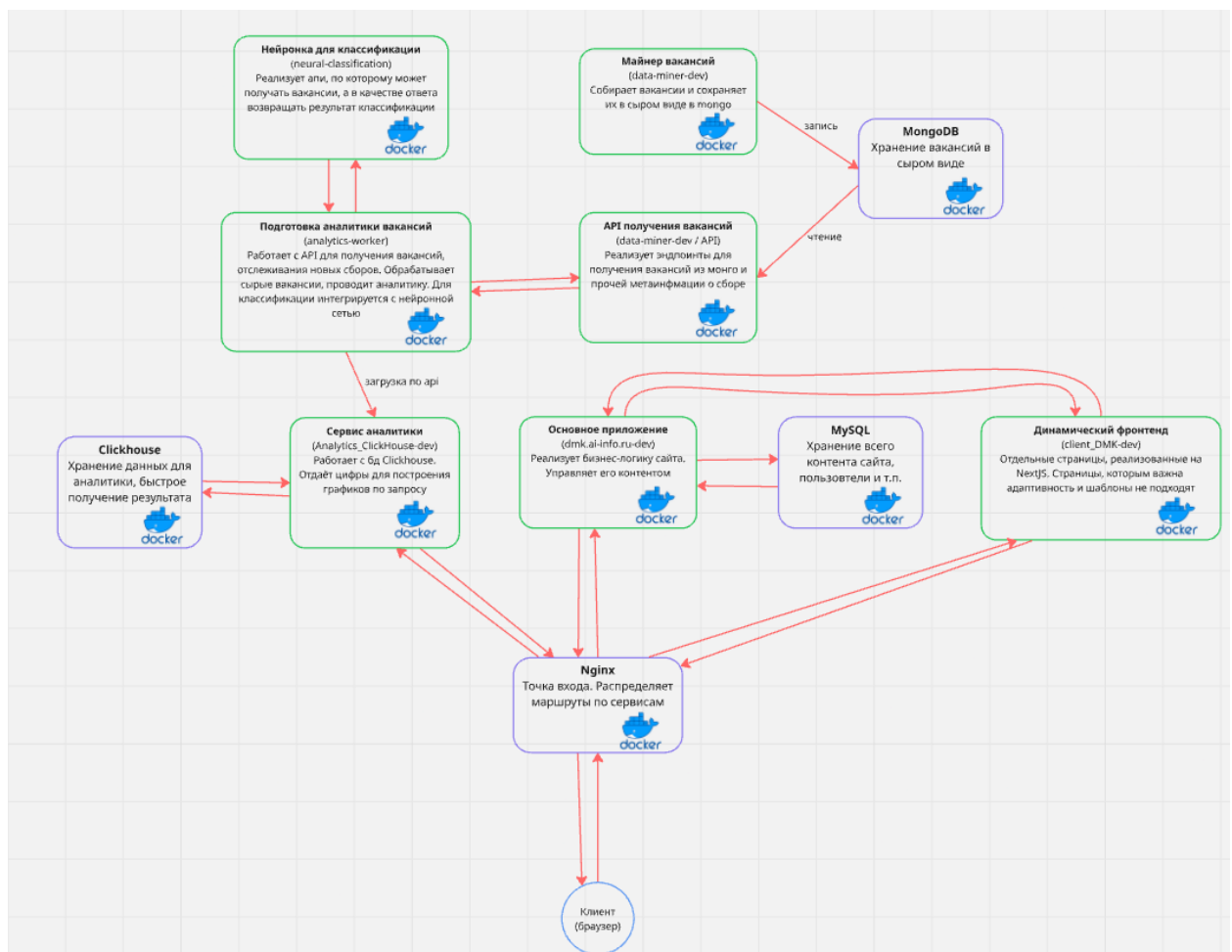


Рисунок 1 – Архитектура системы

Архитектура разрабатываемого сервиса состоит из трех основных компонентов: модуля сбора вакансий (Downloader app), аналитического приложения (Analytics app) и веб-приложения (Web app).

Модуль сбора данных (Downloader app) отвечает за автоматизированный сбор информации о вакансиях в сфере информационных технологий с таких платформ, как hh.ru, trudvsem.ru и SuperJob.ru. Этот компонент интегрируется с внешними API, что обеспечивает регулярное обновление данных. После первичной обработки данные сохраняются для дальнейшего анализа, поддерживая актуальность информации без необходимости ручного вмешательства. Модуль состоит из двух сервисов: майнер (data-miner-dev), собирающий вакансии и сохраняющий их в базу данных, и API для получения вакансии для интеграции со следующим модулем.

Аналитический модуль (Analytics app) обрабатывает данные, полученные от модуля сбора. Его основная задача — преобразовать информацию в удобный для анализа формат, включая агрегацию, классификацию и фильтрацию данных. Этот компонент играет ключевую роль в создании аналитической информации, которая используется для принятия обоснованных решений. Модуль состоит из трёх сервисов. Данные, поступающие в модуль, сначала обрабатываются и классифицируются (сервис analytics-worker). Классификация происходит на основе специально обученной нейронной сети, представленной в виде независимого отдельного сервиса (neural-classification). После обработки данные загружаются в сервис (Analytics\_ClickHouse-dev), предоставляющий API для получения готовой аналитики. В его задачи входит быстрый подсчёт в реальном времени запрашиваемого анализа.

Веб-приложение (Web app) предоставляет интерфейс для пользователей, через который они могут получать доступ к аналитике вакансий, зарплат и навыков. Оно включает удобный и интуитивно понятный интерфейс, а также административную панель для управления данными, их обновления и настройки персонализированных функций для различных групп

пользователей. Веб-приложение является точкой взаимодействия с системой. Модуль представлен сервисом-ядром (dmk.ai-info.ru-dev), реализующий основную бизнес-логику веб-приложения. А также отдельным фронтом (client\_DMK-dev), демонстрирующий новый подход в архитектуре фронтеда, к которой мы стремимся.

Такая модульная архитектура повышает гибкость и масштабируемость системы, позволяя эффективно развивать и поддерживать ее, а также внедрять изменения в один из компонентов без влияния на остальные.

В этом семестре основное внимание будет уделено работе над внедрением новой модульной архитектуры, оптимизацией внесения изменений посредством настройки CI-CD пайплайна, разделением ответственности между зонами сервисов, решением существенной архитектурной проблемы, основанной на работе с CSV-файлами и всеми вытекающими недостатками, в виде высокой связанности, а так же повышение надёжности и автономности системы.



## 2 Основная часть

### 2.1 Планирование работы

В рамках работы над проектом были определены общие цели и задачи, которые включают создание новой модели кластеризации с использованием машинного обучения, внедрения модульной архитектуры, разработку сервиса для тестирования на проф-ориентацию, разработку нового user-friendly дизайна, модернизацию фронтенд архитектуры на модульную, уменьшение бloatа в джаваскрипте. Эти задачи были разбиты на более мелкие этапы и разделены между участниками команды с учетом их компетенций и опыта.

Процесс разработки велся параллельно по нескольким направлениям, что позволяло ускорить выполнение задач и обеспечить интеграцию разных компонентов на разных стадиях проекта. Весь процесс сопровождался активной работой с заказчиком, включая регулярные обсуждения результатов аналитики, разработки и обработку обратной связи. Важно отметить, что формат работы позволил своевременно интегрировать различные компоненты системы и эффективно управлять временем разработки. В процессе работы также происходило постоянное тестирование, доработка компонентов и оперативное внесение изменений в соответствии с замечаниями заказчика, что обеспечивало качественную и своевременную реализацию проекта.

Для реализации проекта были использованы разнообразные ресурсы и инструменты, включая платформы для работы с данными, такие как Python и библиотеки для машинного обучения, а также инструменты для разработки веб-приложений, такие как React, Next.js и Node.js. На backend-е использовались Django, FastAPI, Swagger, ClickHouse и MySQL. Управление задачами велось через платформу YouGile[5], а для связи внутри команды использовался Microsoft Teams. Информация по спринтам велась в Google Sheets, что

обеспечивало прозрачность в процессе выполнения задач и их отслеживание. Интеграция всех этих инструментов позволила создать гибкую и масштабируемую архитектуру приложения, что способствовало эффективной разработке и быстрому реагированию на изменения.

Оценка сроков работы над проектом была произведена на основе принципов Agile[6], с использованием спринтов продолжительностью в две недели. Каждый спринт включал четко определенные задачи, которые должны были быть выполнены за определенный срок. В рамках каждой итерации проводились ревью и ретроспективы, что позволяло вовремя вносить корректировки в сроки и задачи. На основе этой оценки был построен график работы и определены ключевые сроки для каждого этапа проекта.

В результате применения описанного подхода удалось обеспечить высокое качество работы, гибкость в реагировании на изменяющиеся условия, что способствовало успешной реализации проекта в установленные сроки.

## **2.2 Отчет по работе Бушуева Арсений Олеговича**

В рамках работы по дисциплине «Проектный практикум» в 4 семестре была проведена разработка проекта «Найти в IT», название которого в TeamProject – «Динамическая модель компетенций». Роль тимлида в команде включала организацию работы, координацию процессов и управление командой.

В ходе работы была проведена работа по формированию требований к новому функционалу проекта, которые включали создание модели кластеризации на основе методов машинного обучения, разработку новой микросервисной архитектуры, оптимизацию backend-части приложения и совершенствованию фронтенда. Эти требования стали основой для дальнейшей работы команды, поскольку каждый новый функционал вносил изменения в архитектуру системы, что требовало соответствующей адаптации всех компонентов.

Для достижения поставленных целей был разработан подробный план работ, основанный на принципах Agile. Планирование работы, распределение задач по спринтам и назначение ответственных за выполнение задач стало важным этапом разработки. Одним из ключевых моментов было обеспечение четкого понимания каждым участником команды своих обязанностей и ролей, что способствовало повышению эффективности работы и улучшению взаимодействия между участниками команды.

Организация работы команды в ходе проекта включала координацию всех этапов разработки. Поддерживалась постоянная связь с куратором проекта, организовывались рабочие звонки для обсуждения текущих вопросов и обмена информацией, проводилась аналитика по возникающим задачам. Важным элементом работы было систематическое рассмотрение обратной связи от заказчика и команды, что позволило оперативно корректировать направление работы и повышать качество выполнения задач.

Кроме того, внимание уделялось соблюдению сроков и качеству выполнения задач, что включало организацию регулярных ретроспектив выполненных спринтов. Этот процесс способствовал оперативному выявлению слабых мест в работе и своевременному принятию мер по их устранению.

Таким образом, основная задача заключалась в организации и координации работы команды, обеспечении своевременного выполнения задач и их качественной реализации. Взаимодействие с командой и заказчиком позволило адаптировать стратегию работы с учетом изменяющихся условий и успешно реализовать проект.

### **2.3 Отчет по работе Яковлева Алексея Ильича**

В рамках проектной деятельности по дисциплине «Проектный практикум» в 4 семестре были проведены большие архитектурные изменения в backend-части приложения, касающейся всей аналитики по вакансиям.

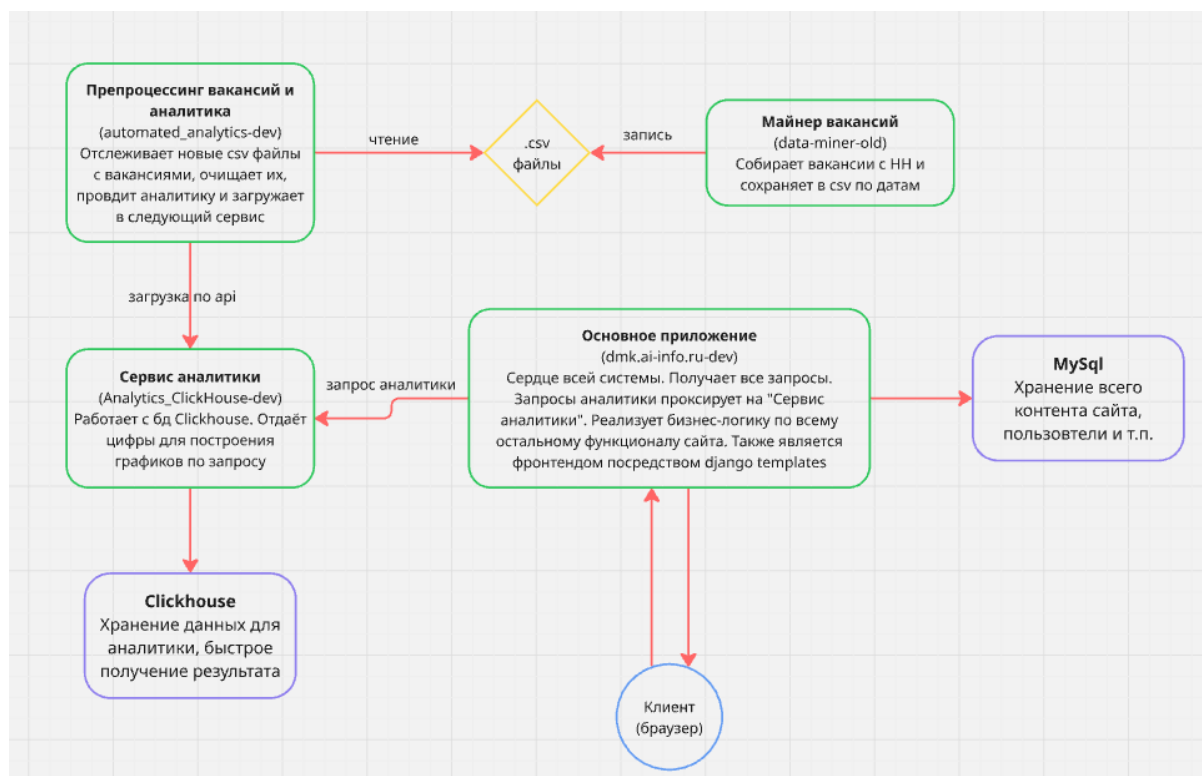


Рисунок 2 - Старая архитектура на CSV-файлах

Самой главной проблемой являлось использование csv файлов, как способ общения и передачи данных между сервисами в системе. Такой подход делал систему излишне сложной, каждый сервис был строго связан с другим, то есть не соблюдался принцип независимости. Также в текущей реализации было сложно модернизировать систему, разрабатывать новый функционал.

Поэтому была разработана новая архитектура, строящаяся на принципе независимости каждого сервиса, то есть компоненты не должны знать о существовании друг друга. Такой подход обеспечивает легкость и быстроту внедрения нового функционала, облегчает поддержку всей системы. Также каждый сервис должен выполнять ровно одну роль.

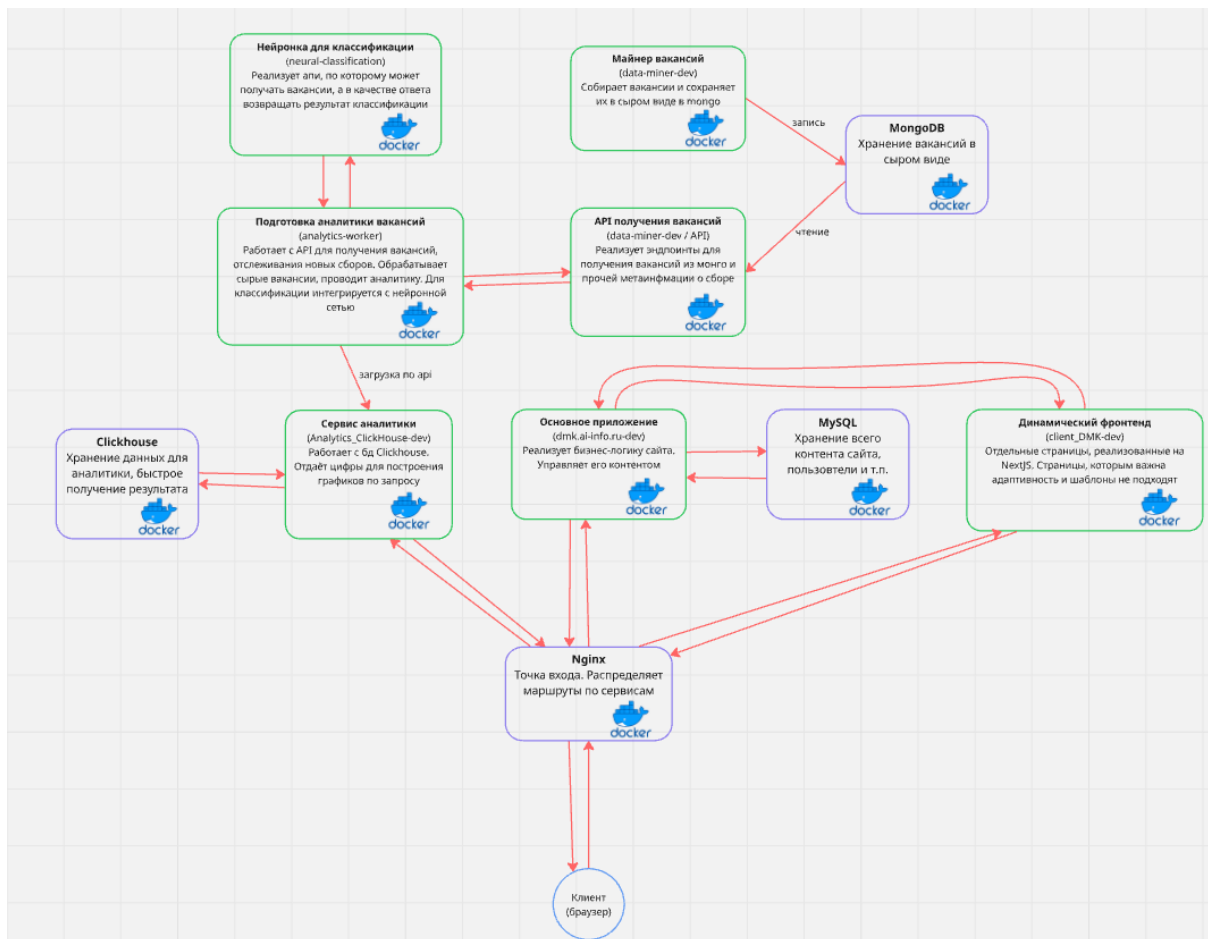


Рисунок 3 - Новая (текущая) архитектура

Был переписан майнер вакансий (data-miner-dev). Теперь он работает без csv файлов, его задача - парсинг вакансий с указанных сайтов и их загрузка в MongoDB. Была выбрана MongoDB, потому что это документо-ориентированная база данных, позволяющая хранить произвольные данные без жестко заданной структуры. Сам по себе сервис не проводит никакой обработки вакансий (обработка проводится только по удалению дубликатов), он просто скачивает их и загружает в бд в сыром виде. Такой подход позволяет хранить полные данные о каждой вакансии за большие промежутки времени, что полезно для внедрения новых видов аналитики, где потребуется информация по вакансиям, не использовавшаяся ранее.

Для интеграции с майнером (data-miner-dev) других компонентов системы разработан API (data-miner-dev / api). Это отдельный сервис, работающий с MongoDB на чтение данных из неё. Был выбран фреймворк FastAPI, как

простой способ написать работающее REST API. В задачи этого сервиса входит выдача вакансий по запросу и также их обработка. Обработка заключается в удалении ненужных полей (настраиваемых параметр) и удаление излишней вложенности данных (линеаризация).

Далее данные необходимо классифицировать, провести обработку. Это входит в задачу сервиса подготовки аналитики (analytics-worker). Его прошлый предшественник (automated\_analytics-dev) работал при помощи csv файлов и мог обрабатывать только вакансии с сайта hh. Новый вариант полностью автономен, для получения вакансий он интегрируется с API майнера (data-miner-dev / api). Его основная задача - классификация вакансий по профессиям, навыков по группам, местоположения по областям. Для классификации вакансий по профессиям использован новый подход на базе обученной нейронной сети, что позволило более точно распознавать профессии, то есть мы улучшили точность аналитики (ранее использовался подход по ключевым словам, основной минус - много вакансий не распознавалось, были коллизии). Для этой задачи был разработан независимый автономный сервис классификации вакансий на основе нейронной сети (neural-classification). Этот сервис предоставляет API для интеграции, использован FastAPI.

Благодаря почти полному переписыванию сбора, подготовки аналитики с учетом прошлых недостатков, удалось добиться такой архитектура, в которой модернизация не является проблемой. Так были добавлены ещё два сервиса для сбора вакансий: trudysem, superjob.

Отдельного внимания заслуживает DevOps. Всего у нас 7 сервисов + 3 баз данных. Раньше это было ещё большой проблемой (хотя и сервисов было меньше), но все они запускались вручную и нужно было не ошибиться с конфигурированием. Чтобы решить эту проблему, каждый сервис упакован в докер контейнер, а развертывание всей системы происходит по одной команде.

Все настройки конфигурации вынесены наружу, чтобы настраивать работу системы можно было из одного места, не влезая в код.

```
[+] Running 19/19
✓Network it_professions_common          Created
✓Volume "it_professions_grafana_storage" Created
✓Volume "it_professions_clickhouse_data" Created
✓Volume "it_professions_mongo_data"      Created
✓Volume "it_professions_analytics_worker_progress" Cre...
✓Volume "it_professions_mysql_data"      Created
✓Container it_professions-miner_mongo-1  Started
✓Container it_professions-dmk_mysql-1    Started
✓Container it_professions-analytics_clickhouse-1 Health...
✓Container it_professions-grafana_promtail-1 Started
✓Container it_professions-grafana_loki-1  Started
✓Container it_professions-grafana-1       Started
✓Container it_professions-dmk-web-1       Started
✓Container it_professions-analytics_clickhouse_api-1 S...
✓Container it_professions-miner_api-1     Started
✓Container it_professions-miner_worker-1  Started
✓Container it_professions-frontend_main_next-1 Started
✓Container it_professions-analytics_worker-1 Started
✓Container it_professions-nginx-1         Started
PS C:\Users\alexe\Desktop\configs> |
```

Рисунок 4 – запуск системы

<input type="checkbox"/>	Name ↓↑	State ↓↑ Filter ▾	Quick Actions	Stack ↓↑	Image ⓘ ↓↑
<input type="checkbox"/>	it_professions-analytics_clickhouse-1	healthy		it_professions	clickhouse/clickhouse-server:24.12.3
<input type="checkbox"/>	it_professions-analytics_clickhouse-dev-1	running		it_professions	ghcr.io/it-professions-ru/analytics_clickhouse-dev:main
<input type="checkbox"/>	it_professions-analytics_worker-1	running		it_professions	ghcr.io/it-professions-ru/analytics-worker:main
<input type="checkbox"/>	it_professions-dmk-web-1	running		it_professions	ghcr.io/it-professions-ru/dmk.ai-info.ru-dev:main
<input type="checkbox"/>	it_professions-dmk_mysql-1	running		it_professions	mysql:8.0
<input type="checkbox"/>	it_professions-frontend_main_next-1	running		it_professions	ghcr.io/it-professions-ru/client_dmk-dev:master
<input type="checkbox"/>	it_professions-grafana-1	running		it_professions	grafana/grafana-enterprise:latest
<input type="checkbox"/>	it_professions-grafana_loki-1	running		it_professions	grafana/loki:latest
<input type="checkbox"/>	it_professions-grafana_promtail-1	running		it_professions	grafana/promtail:latest
<input type="checkbox"/>	it_professions-miner_api-1	running		it_professions	ghcr.io/it-professions-ru/data-miner-dev-api:main

Рисунок 5 – работа сервиса на сервере

Также была проведена работа по наблюдению за системой. Настроен сбор логов с помощью promtail, а для их визуализации выбрана grafana. Этот

стек был выбран как самый популярный, а соответственно легкий в настройке. Также были требования к ресурсам, grafana и promtail написаны на GO и потребляют мало памяти, в отличие от ELK стека, написанного на java.

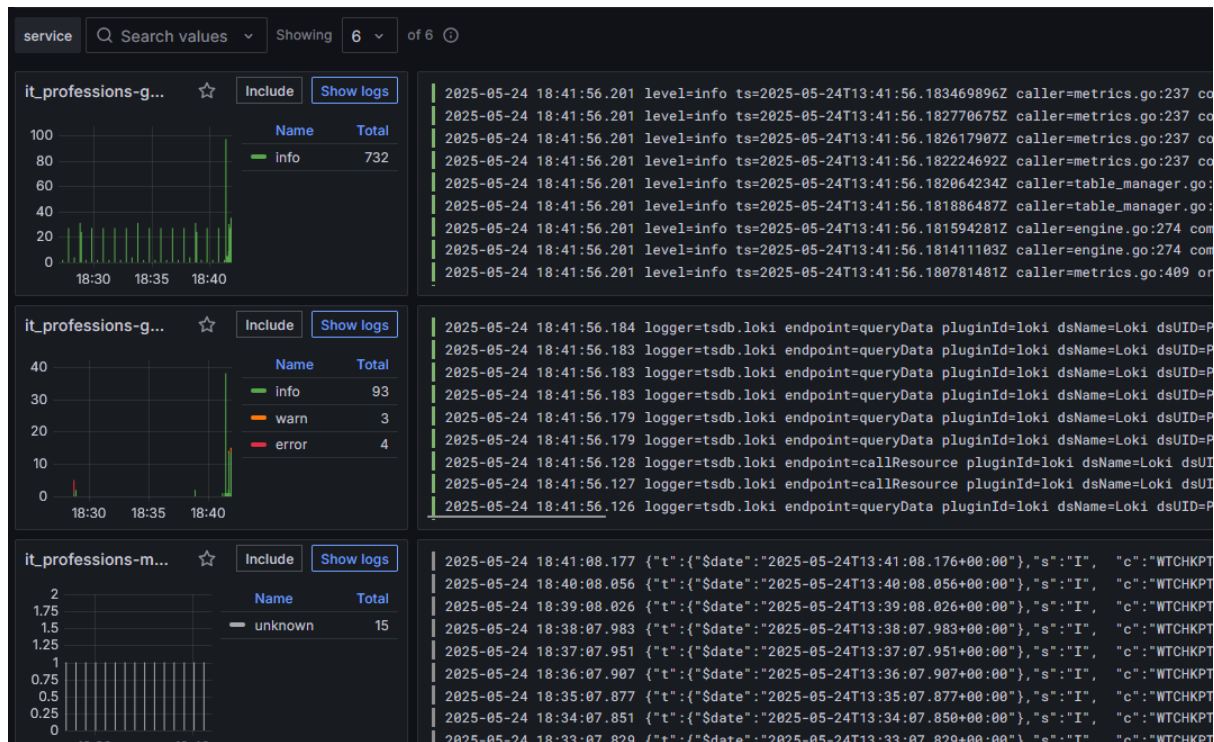


Рисунок 6 – процесс логгирования

Был организован удобный процесс разработки. При коммите в main ветку запускается сборка проекта и развертывание новой версии на сервере. То есть CI/CD



✓ turbopack avoided CI/CD #7: Commit c11e94d pushed by akiraduck	master	2 hours ago 3m 37s	...
✓ fixed bad image src reflection CI/CD #6: Commit 8987595 pushed by akiraduck	master	2 hours ago 2m 43s	...
✗ dynamic metadata CI/CD #5: Commit 7c58e44 pushed by akiraduck	master	2 hours ago 1m 27s	...
✓ added metadata CI/CD #4: Commit ebe018d pushed by akiraduck	master	2 hours ago 3m 7s	...
✓ Merge remote-tracking branch 'origin/master' CI/CD #3: Commit bcef21e pushed by akiraduck	master	2 hours ago 3m 27s	...
✓ setup docker CI/CD #2: Commit d723703 pushed by yakovlev05	master	5 days ago 3m 33s	...
✗ Create docker-publish.yml CI/CD #1: Commit fdd1e9d pushed by yakovlev05	master	5 days ago 30s	...

Рисунок 7 – CI/CD в Github

Таким образом, была улучшена архитектура системы, добавлен новый функционал по аналитике. Организована бесперебойная работа всего сервиса в целом.

## 2.4 Отчет по работе Старцева Владислава Игоревича

В рамках проектной деятельности по дисциплине «Проектный практикум» в 4 семестре была разработана модель классификации вакансий на основе современных трансформеров. Разработка велась в составе команды в проекте «Найти в IT», где моя роль заключалась в реализации и обучении модели на основе предварительно размеченных данных. Целью данной работы стало создание точной и масштабируемой модели для автоматической классификации профессий по названиям вакансий, предназначенной для интеграции в модуль Analytics App системы.

Изначально использовались классические методы обработки текста и машинного обучения, такие как TF-IDF в сочетании с логистической регрессией или деревьями решений, однако они показывали ограниченную обобщающую способность и низкую точность при работе с текстами на разных языках

и с синонимичной лексикой. Для повышения качества классификации было принято решение использовать мультязычную трансформер-модель DistilBERT, способную эффективно обрабатывать текстовые данные на естественном языке. В качестве основы была выбрана предобученная модель `distilbert-base-multilingual-cased`, поддерживающая работу с более чем 100 языками, включая русский. Для подготовки данных использовался размеченный датасет в формате CSV, содержащий названия вакансий и соответствующие им профессиональные категории. Данные были автоматически загружены с помощью библиотеки `datasets` от Hugging Face и разделены на обучающую и тестовую выборки в пропорции 80/20.

Для классификации использовалась модель `AutoModelForSequenceClassification` с числом выходных классов, соответствующим количеству уникальных профессий в датасете. Каждой профессии был сопоставлен уникальный идентификатор, что позволило преобразовать задачу в задачу многоклассовой классификации. Текстовые данные токенизировались с помощью `AutoTokenizer`, с применением `padding` и усечения длины до 128 токенов, обеспечивая единообразную структуру входных данных для обучения. Обучение модели проводилось с использованием `Trainer` — высокоуровневого API из `Transformers`. Основные параметры обучения включали:

Размер батча: 64

Количество эпох: 3 Скорость обучения:  $5e-5$  Штраф за вес: 0.01

Метрика оптимизации: F1-мера

Использование смешанной точности (fp16) на GPU для ускорения обучения

Для оценки производительности модели использовались стандартные метрики классификации: `accuracy`, `precision`, `recall` и F1. Метрики рассчитывались после каждой 1000 итерации, а логирование производилось каждые 500 шагов. Также использовалась методика ранней остановки с сохранением наилучшей версии модели. Пример использованных метрик при финальной

оценке модели: Accuracy: ~0.89 F1-score: ~0.88 Precision: ~0.87 Recall: ~0.88 В ходе разработки было обеспечено: Автоматическое сопоставление профессий и числовых меток Обработка переменной длины текстов с выравниванием Повышение масштабируемости модели с помощью Trainer и встроенных data collator'ов В результате проведённой работы была построена эффективная модель на основе DistilBERT, способная автоматически классифицировать названия вакансий по профессиональной принадлежности с высокой точностью. Это решение позволило значительно улучшить качество аналитических выводов модуля Analytics App в проекте «Найти в IT» и стало основой для внедрения интеллектуальной обработки вакансий на этапе предварительного анализа данных.

Была оформлена подробная swagger-спецификация, заметно облегчающая внешнюю интеграцию и внутренний онбординг, а также была подготовлена подробнейшая readme-инструкция для всех модулей архитектуры. Все ключевые сервисы были покрыты авто-тестами.

Наконец, был разработан новый бэкенд на FastAPI + SQLAlchemy (async) + Pydantic v2: он разворачивается «из коробки», защищён через X-API-Key для POST запросов, включает REST-эндпоинты для загрузки вопросов и единую точку /assessment, а для контента предусмотрена админ-панель SQLAlchemyAdmin. В итоге фронтенд, API и админка работают как автономная система, выдавая пользователям персональные рекомендации по профессиям без лишних усилий.

## **2.5 Отчет по работе Черникова Александра Алексеевича**

В рамках работы по дисциплине «Проектный практикум» в 4 семестре была проведена разработка проекта «Найти в IT», название которого из TeamProject – «Динамическая модель компетенций». Роль дизайнер в команде заключалась в анализе текущего интерфейса, разработке его визуального

обновления и проектировании нововведений. Основной целью выступило повышение пользовательской вовлеченности, улучшение удобства использования и создание атомарной дизайн-системы для дальнейшего масштабирования.

Первым этапом стал отчет по слабостям, где методом анализа интерфейса и исходного файла Figma были выявлены ключевые проблемы: перегруженность картинки цветами и элементами заднего фона, не интуитивное навигация и расположение элементов. Параллельно был сформирован документ «Перспективы развития», включающий четкое выделение целевой аудитории и предложения по введению интерактивных элементов (тест профориентации, чек-листы навыков). Для аргументации перспектив развития, был написан отчет по метрике на основе данных с сервиса Яндекс Метрика (возрастные категории, посещаемость сайта, трафик конкретных разделов и страниц)

Для визуального обновления в Figma было проработано три стилевых направления: необрутализм с акцентом на строгие формы, минимальным использованием графики и яркими цветами; вариант близкий к старому дизайну, но с акцентом на одном цвете; минимализм с приглушенными цветами. После презентации заказчику был утвержден минималистичный стиль, сочетающий структурную четкость с легким не нагружающим визуальным оформлением. В данном стиле были воссозданы макеты всех ключевых разделов, включая адаптивные версии для мобильных устройств. Для обеспечения консистентности была создана дизайн система, по принципам атомарного дизайна были выделены компоненты с вариациями кнопок, карточек и модулей, что ускорило работу. Особое внимание уделено концептам нововведений:

Тест профориентации — интерактивный опросник с прогресс-баром, визуально интегрированный в общий стиль платформы;

Чек-лист навыков — избранные профессии с возможностью отмечать освоенные навыки;

Система написания статей — шанс для пользователей поделиться личным опытом работы в IT.

Для каждого решения подготовлены демонстрационные макеты, иллюстрирующие их интеграцию в текущий интерфейс.

Таким образом, проведение аналитики перед началом работы, глубокое погружение в пользовательские потребности и гибкость в проработке стилей позволили не только модернизировать интерфейс, но и заложить основу для будущих обновлений.

## **2.6 Отчет по работе Сорокина Юрия Павловича**

В рамках работы по дисциплине «Проектный практикум» в 4 семестре была выполнена адаптация существующего JavaScript-фронтенда на современную архитектуру с использованием фреймворка Next.js, языка TypeScript и утилитарного CSS-фреймворка Tailwind CSS. Основной целью выступала оптимизация производительности интерфейса, улучшение структуры проекта и обеспечение строгой типизации данных.

Для перехода на новую архитектуру был выбран современный маршрутизатор App Router, представленный в последних версиях Next.js. Это позволило реализовать удобную систему вложенных маршрутов с общими компонентами (такими как Header и Footer), а также использовать серверный рендеринг (SSR) и инкрементальную статическую генерацию (ISR) в зависимости от требований конкретной страницы. Такая гибкость значительно повысила как показатели SEO, так и скорость первого рендера.

Одним из ключевых этапов стала полная типизация данных. Были разработаны и вынесены в отдельный модуль `types.ts` интерфейсы и типы для основных сущностей приложения, таких как `Profession`, `Skill` и другие. Это решение позволило минимизировать ошибки на этапе выполнения (runtime) и существенно упростило автодополнение и рефакторинг кода в среде разработки.

Также был создан модуль `const.ts` для хранения всех API-эндпоинтов и статических значений, что обеспечило единообразие при работе с API.

Для развертывания новой frontend-платформы был подготовлен отдельный сервер с установленным веб-сервером `nginx`, что обеспечило стабильную работу приложения даже при высокой нагрузке. Такой подход позволил централизованно управлять маршрутизацией, кешированием и безопасностью соединений.

На этапе внедрения были полностью переработаны и реализованы с нуля следующие ключевые страницы приложения:

«Главная страница» - в качестве точки входа и маршрутизации пользователя;

«Все профессии» — для отображения информации о профессиональных направлениях;

«Все навыки» - для представления компетенций и навыков; «Статьи» - для публикации и чтения материалов;

Профориентационный тест - для завлечения аудитории, еще не определившейся с направлением;

Особое внимание было уделено производительности: страницы были адаптированы под SSR или CSR, в зависимости от целевого поведения, а также внедрена динамическая загрузка компонентов с помощью React-хуков. Примером служат такие компоненты, как `SkillsCard`, использующие локальное состояние и минимизирующие количество ререндеров.

Полный отказ от монолитного CSS-файла объемом более 6000 строк в пользу Tailwind CSS сделал разработку более предсказуемой и наглядной. Классы применяются непосредственно в JSX-разметке, обеспечивая визуальную прозрачность стилей. Кроме того, для изображений и ссылок используются оптимизированные компоненты `next/image` и `next/link`, что улучшило загрузку и переходы между страницами.

Результаты тестов Lighthouse и WebPageTest показали сокращение времени загрузки на 30–40%, а также рост SEO-показателей благодаря семантической верстке и оптимизированному рендерингу. Для обеспечения единого кода-стайла в проекте применяются ESLint и Prettier, а TypeScript обеспечивает строгую типизацию на всех уровнях.

Таким образом, переход на Next.js в связке с TypeScript и Tailwind CSS позволил не только модернизировать архитектуру фронтенда, но и создать устойчивую основу для дальнейшего масштабирования. В дальнейших планах — доработка системы аутентификации с использованием токенов и декомпозиция сложных компонентов, включая фильтры и графики.

## ЗАКЛЮЧЕНИЕ

Анализ итогов разработки показал, что обновлённый программный продукт соответствует большинству требований, выдвинутых заказчиком и пользователями. В процессе проектирования и реализации учитывались ключевые запросы целевой аудитории, что позволило создать функциональное и производительное решение. Применение современных технологий, таких как Next.js, Django и ClickHouse, в сочетании с разработанной моделью кластеризации вакансий, обеспечило необходимую гибкость, масштабируемость и аналитическую мощность системы. Модель кластеризации, реализованная с использованием библиотеки scikit-learn и алгоритма MiniBatchKMeans, позволила добиться точной кластеризации вакансий, что существенно улучшило возможности аналитики и работы с данными.

Несмотря на достигнутый высокий уровень функциональности, продукт обладает потенциалом для дальнейшего роста и развития. Одной из ключевых перспектив является расширение области применения разработанных моделей и инструментов. В частности, предложенная модель кластеризации вакансий, успешно задействованная в IT-отрасли, может быть протестирована на данных из других востребованных отраслей, таких как промышленность, финансовый сектор или строительство. Это позволит проверить гипотезу о возможности создания универсальной системы для HR-специалистов, которая будет предоставлять детализированную статистику и аналитику для профессиональной деятельности



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Найти в IT [Электронный ресурс] – URL: <https://it-professions.ru/> (дата обращения: 11.01.2025).
2. Habr.Карьера [Электронный ресурс] – URL: <https://career.habr.com/> (дата обращения: 22.11.2024).
3. Зарплатомер [Электронный ресурс] – URL: <https://www.zarplatomer.ru/> (дата обращения: 10.01.2025).
4. Роснавык [Электронный ресурс] – URL: <https://lk.rosnavyk.ru> (дата обращения: 22.11.2024).
5. YouGile [Электронный ресурс] – URL: <https://ru.yougile.com/> (дата обращения: 11.01.2025).
6. The Agile Coach [Электронный ресурс] – URL: <https://www.atlassian.com/agile> (дата обращения: 10.01.2025).