**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа «Программная инженерия»

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Доцент департамента программной инженерии факультета компьютерных наук, кандидат наук  А.В.Меликян  « » 2025 г. | УТВЕРЖДЕНО  Академический руководитель ОП  «Программная инженерия», старший преподаватель департамента программной инженерии  Н.А. Павлочев  « » 2025 г. |

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. № |  |
| Взам. Инв. |  |
| Подп. и дата |  |
| Инв. № |  |

# Программа для анализа рынка труда аналитиков данных

**Пояснительная записка ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.05.10-01 81 01-1-ЛУ**

Исполнитель: Студент группы БПИ2310

/ А.М. Муслин /

« » 2025 г.

УТВЕРЖДЕН

RU.17701729.05.10-01 81 01-1-ЛУ

# Программа для анализа рынка труда аналитиков данных

**Пояснительная записка**

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл |  |

**RU.17701729.05.10-01 81 01-1-ЛУ**

**Листов 57**

# АННОТАЦИЯ

Данный программный документ представляет собой пояснительную записку к курсовому проекту под названием «Программа для анализа рынка труда аналитиков данных».

В «Введении» указано наименование программного продукта, а также документ-основание, в соответствии с которым осуществляется разработка, с упоминанием организации, утвердившей данный документ.

Раздел «Назначение и область применения» раскрывает основные функциональные задачи и эксплуатационные цели программы, а также содержит краткое описание сферы её практического использования.

В разделе «Технические характеристики» рассматриваются следующие компоненты: формулировка задачи, описание логики работы программы, её функциональные особенности, обоснование методов организации ввода и вывода данных, а также подбор и аргументация выбора технических и программных средств.

Раздел «Ожидаемые технико-экономические показатели» содержит прогноз по предполагаемому спросу на программный продукт и описывает его преимущества по сравнению с существующими отечественными и зарубежными аналогами.

Программный документ разработан в соответствии с требованиями:

1. ГОСТ 19.101-77 Виды программ и программных документов [1];
2. ГОСТ 19.102-77 Стадии разработки [2];
3. ГОСТ 19.103-77 Обозначения программ и программных документов [3];
4. ГОСТ 19.104-78 Основные надписи [4];
5. ГОСТ 19.105-78 Общие требования к программным документам [5];
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [10].

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78 [13], ГОСТ 19.604-78 [14].

# СОДЕРЖАНИЕ

[АННОТАЦИЯ 2](#_Toc195206648)

[СОДЕРЖАНИЕ 3](#_Toc195206649)

[1. ВВЕДЕНИЕ 6](#_Toc195206650)

[1.1. Наименование программы 6](#_Toc195206651)

[1.2. Документ, на основании которого ведется разработка 6](#_Toc195206652)

[2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 7](#_Toc195206653)

[2.1. Назначение программы 7](#_Toc195206654)

[2.1.1. Функциональное назначение 7](#_Toc195206655)

[2.1.2. Эксплуатационное назначение 7](#_Toc195206656)

[2.2. Краткая характеристика и область применения 8](#_Toc195206657)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 9](#_Toc195206658)

[3.1. Постановка задачи на разработку программы 9](#_Toc195206659)

[3.2 Описание и обоснование выбора состава технических и программных средств 13](#_Toc195206660)

[3.2.1 Состав технических и программных средств 13](#_Toc195206661)

[3.2.2 Обоснование выбора технических и программных средств 13](#_Toc195206662)

[3.3 Описание и обоснование архитектуры программы 14](#_Toc195206663)

[3.4 Описание и обоснование выбора метода организации входных и выходных данных 15](#_Toc195206664)

[4 ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 19](#_Toc195206665)

[4.2 Ориентировочная экономическая эффективность 19](#_Toc195206666)

[4.3 Предполагаемая потребность 19](#_Toc195206667)

[4.4 Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами 19](#_Toc195206668)

[5 ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ 22](#_Toc195206705)

[ПРИЛОЖЕНИЕ 1 ОПИСАНИЕ И НАЗНАЧЕНИЕ ФАЙЛОВ, ИСПОЛЬЗУЕМЫХ ПРИ ПАРСИНГЕ 25](#_Toc195206706)

[ПРИЛОЖЕНИЕ 2 ОПИСАНИЕ И НАЗНАЧЕНИЕ ФАЙЛОВ, ИСПОЛЬЗУЕМЫХ НА СЕРВЕРЕ 28](#_Toc195206707)

[ПРИЛОЖЕНИЕ 3 ОПИСАНИЕ И НАЗНАЧЕНИЕ ФАЙЛОВ, ИСПОЛЬЗУЕМЫХ НА КЛИЕНТСКОЙ ЧАСТИ (ФРОНТ) 39](#_Toc195206708)

[ПРИЛОЖЕНИЕ 4 ОПИСАНИЕ СТРУКТУРЫ ТАБЛИЦЫ ВАКАНСИЙ В БАЗЕ ДАННЫХ 46](#_Toc195206709)

[ПРИЛОЖЕНИЕ 5 ОПИСАНИЕ СТРУКТУРЫ ТАБЛИЦЫ РЕЗЮМЕ В БАЗЕ ДАННЫХ 48](#_Toc195206710)

[ПРИЛОЖЕНИЕ 6 ОПИСАНИЕ СТРУКТУРЫ ТАБЛИЦЫ ВАЛЮТ В БАЗЕ ДАННЫХ 50](#_Toc195206711)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 51](#_Toc195206712)

# 

# ВВЕДЕНИЕ

## Наименование программы

Наименование программы – «Веб-приложение DataCareerInsight».

Наименование программы на английским языке – «DataCareerInsight web application».

Краткое наименование программы – «DataCareerInsight».

## Документ, на основании которого ведется разработка

Разработка ведется на основании учебного плана подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденной академическим руководителем программы темы курсового проекта.

# НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

## Назначение программы

## Функциональное назначение

«Веб-приложение DataCareerInsight» предоставляет пользователям инструменты для анализа текущего состояния рынка труда специалистов в области аналитики данных в России.

Программа предназначена для сбора, обработки и визуализации данных о вакансиях и резюме аналитиков данных с hh.ru

Основные функциональные возможности программы включают:

1. Сбор данных о вакансиях и резюме, фильтрация по ключевым параметрам (зарплата, опыт работы, возраст, компания, условия работы и др.);
2. Интерактивные графики для визуализации статистики по выбранным критериям;
3. Хранение и обработка данных в базе данных с помощью SQL, что позволяет эффективно управлять информацией и предоставлять быстрый доступ к данным для пользователей;
4. Прогнозирование заработной платы с помощью линейной регрессии на основе введенных пользователем параметров;

## Эксплуатационное назначение

«Веб-приложение DataCareerInsight» будет использоваться как инструмент для анализа трудового рынка аналитиков данных для различных категорий пользователей:

1. Соискатели могут использовать программу для оценки рыночных условий, прогнозирования своей заработной платы и оценки востребованных навыков в сфере аналитики данных.
2. Работодатели могут анализировать текущее состояние рынка труда, прогнозировать зарплатные ожидания кандидатов, а также оптимизировать требования и условия для вакансий, чтобы привлечь более квалифицированных специалистов
3. Интересующиеся могут использовать программу для понимания рынка труда в сфере аналитики данных.

## Краткая характеристика и область применения

“Веб-приложение DataCareerInsight ” - сайт, который позволяет визуализировать информацию о вакансиях и резюме специалистов по аналитике данных в России. Основное применение веб-приложения – предоставление агрегированной информации о рынке труда аналитиков данных, включая разные параметры. Важной частью веб-приложения является интерактивная визуализация, которая дает пользователям возможность выбора критериев для анализа с последующим графическим представлением, а также добавление элементов машинного обучения, а именно линейной регрессии, для прогнозирования ожидаемой зарплаты на основе введенных пользователем параметров.

Основная цель программы — предоставить пользователю возможность самостоятельно анализировать текущее состояние рынка труда аналитиков данных в России, выбирая параметры, которые его интересуют, и получая наглядную и детализированную информацию.

# ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

## Постановка задачи на разработку программы

“Веб-приложение DataCareerInsight ” должно предоставлять следующие функции:

1. Главная страница с информацией о каждой вкладке сайта и их возможностях
2. Страница с вакансиями, которая содержит таблицу с вакансиями и 4 типа интерактивных графиков
3. Страница с резюме, которая содержит таблицу с резюме и 4 типа интерактивных графиков
4. Страница с предсказыванием зарплаты по значениям из таблицы вакансий/ резюме
5. Возможность просматривать таблицу со всеми собранными вакансиями и резюме
6. Возможность свободного перемещения между страницами в любой точке сайта
7. Фильтры для строк таблицы
   * Возможность выбора разных столбцов, операторов сравнения и значений
   * Возможность добалвять несколько значений в поле фильтра для столбца
   * Возможность выбирать агрегирующие значения для сравнения столбца с ними
   * Возможность добавлять неограниченное количество фильтров
   * Возможность удаления отдельно взятого фильтра
8. Группировка столбцов
   * Возможность выбрать столбец/столбцы для группировки и получить соответсвующую таблицу
   * Возможность удаления отдельно взятого столбца из группировки
   * Возможность удаления всех столбцов из группировки
9. Агрегированные столбцы
   * Возможность выбирать столбец и агрегирющее значение для него и получать соответсвующую таблицу
   * Возможность удаления отдельно взятого столбца из агрегации
   * Возможность удаления всех столбцов из агрегации
10. Фильтр по агрегированным полям
    * При наличии группировки и агрегации давать возможность отфильтровать любой из агрегирющих столбцов аналогично фильтрам для обычных столбцов
    * Возможность добалвять несколько значений в поле фильтра для агрегированного столбца
    * Возможность удаления отдельно взятого фильтра по агрегированному столбцу
    * Возможность удаления всех фильтров по агрегированным столбцам
11. Фильтр по ненулевым значениям
    * Возможность добавления фильтра по ненулевым значениям для столбца, который оставляет только те строки таблицы, где в выбранном столбце есть значения
    * Возможность удаления отдельно взятого фильтра по ненулевым значениям для отдельно взятого столбца
    * Возможность удаления всех фильтров по ненулевым значениям
12. Сортировка
    * Возможность выбрать любой столбец, который сейчас отображается в таблице, и отсортировать по нему строки в таблице (по убыванию/ по возрастанию)
    * Возможность удаления сортировке по отдельно взятому столбцу
    * Возможность удаления всех сортировок
13. Настройка отображения столбцов
    * Возможность скрыть столбец и не показывать его
    * Возможность убрать из скрытых столбец, чтобы он снова отображался в таблице
14. Возможность очистки всех фильтров/группировок и т.д. одной кнопкой
15. Возмоность скрыть настройки фильтров для компактного отображения
16. Возможность выбирать количество отображаемых строк на странице
17. Возможность перемещаться между таблицами страницы
18. Возможность увидеть, сколько всего строк в текущей таблице
19. Возможность изменять высоту каждой отдельной строки путем наведения на ее низ и протаскивания вверх/вниз
20. Интерактивный график с количественным распределением для вакансий и резюме
    * Возможность увидеть подробности про параметры и отображение графика
    * Возможность выбора столбца и получение по нему график, где ось X – значение из данного столбца, ось Y – количество соответствующих значений в данном столбце
    * Возможность выбора типа графика: столбчатая диагармма, круговая диаграмма, точечная диаграмма
    * Возможность выбрать количество отображаемых значений столбца (берутся самые часто встречающиеся)
    * Возможность добавить аналогичный таблице фильтр по столбцам для построения графика только с нужными строками таблицы
    * Возможность увидеть, сколько всего есть различных значений у выбранного столбца
    * Возможность при наведении на значение на графике увидеть соотв. ему значение из выбранного столбца, а также кол-во таких значений
21. Интерактивный график с распределением с течением времени для вакансий и резюме
    * Возможность увидеть подробности про параметры и отображение графика
    * Возможность выбрать столбец и соотв. ему агргегирющее значений, которое будет взято для распределния данного параметра с течением времени (по месяцам)
    * Возможность выбрать тип графика: линейный, столбчатый, точечный
    * Возможность при наведении на конкретное значение увидеть период времени, конкретное значение параметра, а также кол-во наблюдений с таким значением (если агрегация - мода) или кол-во наблюдений за этот период, взятых для агрегации
    * Возможность добавить аналогичный таблице фильтр по столбцам для построения графика только с нужными строками таблицы
    * Возможность увидеть, сколько всего есть взято периодов/ месяцев
22. Интерактивный график со сравнением выбранного показателя по данному признаку для вакансий и резюме
    * Возможность увидеть подробности про параметры и отображение графика
    * Возможность выбрать метрический столбце и соотв. ему агрегацию в качестве показателя
    * Возможность выбрать столбец в качестве признака
    * Возможность увидеть по каждому отдельно взятому значению из столбца признака его выбранный показатель
    * Возможность выбрать количество отображаемых значений столбца-признака (берутся самые часто встречающиеся)
    * Возможность выбрать тип графика: линейный, столбчатый, круговой
    * Возможность добавить аналогичный таблице фильтр по столбцам для построения графика только с нужными строками таблицы
    * Возможность видеть, сколько всего есть различных значений у столбца-признака
    * Возможность при наведении на конкретное значение увидеть значение признака, показателя, а также кол-во наблюдений с данным значением признака в таблице
23. Интерактивный график с распредением выбранного метрического столбца по диапазонам для вакансий и резюме
    * Возможность увидеть подробности про параметры и отображение графика
    * Возможность выбрать метрический столбец и количество диапазонов и увидеть, количество наблюдений данного столбца в каждом из диапазонов
    * Возможность выбрать тип графика: столбчатый, круговой
    * Возможность при наведении на конкретное значение увидеть соотв. ему диапазон и кол-во наблюдений, попавших в этот диапазон
    * Возможность добавить аналогичный таблице фильтр по столбцам для построения графика только с нужными строками таблицы
24. Предсказывание зарплаты
    * Возможность выбрать значения по каждому из предикторов модели линейной регрессии и получить заработную плату, на которую пользователь может расчитывать исходя из собранной информации
    * Возможность выбора предсказывающей модели (по таблице с вакансиями или с резюме)
    * Возможность увидеть подробную информацию о моделе: метрики, топ значимых предикторов, топ незначимых предикторов, кол-во предикторов модели, уравнение линейной регрессии
    1. **Описание и обоснование выбора состава технических и программных средств**
       1. **Состав технических и программных средств**

**Клиентская часть:**

* Веб-браузер с поддержкой современных стандартов HTML5 и CSS3
* Устройство с разрешением экрана не менее 1280×720 пикселей
* 2 ГБ оперативной памяти или больше
* Подключение к сети интернет со скоростью не менее 5 Мбит/с

**Серверная часть:**

* Процессор с 2+ ядрами
* 4 ГБ оперативной памяти или больше
* 20 ГБ свободного дискового пространства
* Python 3.9+
* Node.js 16+
* PostgreSQL 12+

**Основные программные компоненты:**

* Фронтенд: React 18+, Material-UI 5+, Chart.js 3+
* Бэкенд: FastAPI 0.68+, SQLAlchemy 1.4+, Pandas 1.3+
* Парсер: requests 2.26+, psycopg2 2.9+
* База данных: PostgreSQL 12+
* Внешние API: hh.ru API
  + 1. **Обоснование выбора технических и программных средств**

**Клиентская часть:**

* Современные веб-браузеры обеспечивают полную поддержку всех функций React и Chart.js, что гарантирует корректное отображение интерфейса и графиков.
* Разрешение экрана 1280×720 выбрано как минимальное для комфортного отображения сложных графиков и таблиц с данными.
* 2 ГБ оперативной памяти обеспечивают плавную работу интерфейса при обработке больших наборов данных.

**Серверная часть:**

* Python 3.9+ обеспечивает поддержку всех современных функций FastAPI и асинхронного программирования.
* PostgreSQL выбран за его надежность, производительность и расширенные возможности работы с данными, включая поддержку массивов и JSON.

**Программные компоненты:**

* React и Material-UI позволяют создавать современные интерфейсы.
* FastAPI обеспечивает высокую производительность API и автоматическую генерацию документации.
* SQLAlchemy предоставляет удобный ORM-слой для работы с PostgreSQL.
* API hh.ru является официальным и наиболее полным источником данных о вакансиях и резюме в России.

**Безопасность:**

* Использование .env-файлов для хранения конфиденциальных данных
  1. **Описание и обоснование архитектуры программы**

Веб-приложение "DataCareerInsight" представляет собой трехзвенную архитектуру, состоящую из:

1. **Клиентского приложения** (Frontend):
   * Реализовано на React с использованием функциональных компонентов и хуков
   * Material-UI обеспечивает единый стиль и адаптивность интерфейса
   * Chart.js используется для визуализации интерактивных графиков
   * React Router управляет навигацией между разделами приложения
2. **Серверной части** (Backend):
   * Построена на FastAPI, обеспечивающем высокую производительность
   * Использует асинхронные endpoints для обработки запросов
3. **Сервиса сбора данных** (Parser):
   * Консольное приложение на Python
   * Использует официальное API hh.ru для получения данных

**Обоснование архитектурных решений:**

1. **Разделение клиента и сервера** позволяет независимо развивать интерфейс и бизнес-логику, а также обеспечивает возможность создания мобильных клиентов в будущем.
2. **Использование FastAPI** обеспечивает:
   * Высокую производительность за счет асинхронности
   * Автоматическую генерацию документации API
   * Простую интеграцию с SQLAlchemy
3. **Модульная структура парсера** позволяет:
   * Легко добавлять новые источники данных
   * Изолировать ошибки парсинга
4. **PostgreSQL** выбран за:
   * Надежность и стабильность
   * Поддержку сложных типов данных (массивы, JSON)
   * Возможность создания триггеров и хранимых процедур
   * Хорошую интеграцию с Python-экосистемой

**Потоки данных в системе:**

1. Пользователь взаимодействует с React-интерфейсом
2. Клиентское приложение отправляет запросы к FastAPI
3. Сервер обрабатывает запросы и взаимодействует с PostgreSQL
4. Парсер периодически обновляет данные в БД через API hh.ru
5. Изменения в БД становятся доступны через API сервера

Такая архитектура обеспечивает масштабируемость, надежность и простоту дальнейшего развития системы.

## Описание и обоснование выбора метода организации входных и выходных данных

**Входные данные**

1. Для эндпоинтов таблиц (vacancies\_table.py и resumes\_table.py):

GET /resume/table/

GET /vacancies/table/

Параметры запроса:

* + specific\_fields - строка с перечислением полей через запятую “[поле],[поле],…”
  + filters - строка с условиями фильтрации в формате “[поле][оператор][and/or][:значение]”

значения внутри фильтра разделяются с помощью тильды, фильтры для разных столбцов разделяются с помощью точки с запятой

* + group\_by - поля для группировки через запятую “[поле],[поле],…”
  + having - условия фильтрации после группировки через тильду в формате “[поле]:[ функция][оператор][значение]”
  + aggregates - агрегатные функции в формате "[поле]:[функция]", разделенные запятой
  + not\_null - поля, которые не должны быть NULL в формате “[поле],[поле],…”, разделенные запятой
  + sort\_by - поле и направление сортировки разделяются запятой
    - для сортировки по агрегированному столбцу формат “[поле]:[функция]:[asc/desc]”
    - для сортировки по обычному столбцу формат “[поле]:[asc/desc]”
  + limit и offset – пагинация и лимит отображаемых значений в формате “[значение]”

1. Для эндпоинтов графиков (charts.py):

GET /charts/тип\_графика/

* Общие параметры:
  + model - "vacancies" или "resume"
  + column - анализируемое поле
  + chart\_type - тип графика
  + filters - аналогично таблицам
* Специфические параметры:
  + Для /column\_count/ - limit (лимит значений) в формате
  + Для /time\_distribution/ - aggregates (функция агрегации) в формате
  + Для /metric\_column/ - metric\_column и aggregations
  + Для /metric\_distribution/ - number\_range (количество диапазонов)

3. Для эндпоинтов зарплат (salary.py):

* Для вакансий:

POST /salary/vacancies\_salary/

* + min\_experience - минимальный опыт
  + type\_of\_employment, work\_format, address - категориальные признаки
  + skills - словарь навыков
* Для резюме:

POST /salary/resume\_salary/

* + total\_experience, count\_additional\_courses и др. числовые признаки
  + Множество категориальных признаков в виде словарей

4. Для получения структуры модели (get\_model\_structure.py):

GET / prediction/model\_structure/

* base\_model - "vacancies" или "resume"

**Выходные данные**

1. Таблицы:

JSON с полями:

* + results - массив записей
  + total\_count - общее количество записей

2. Графики:

Единый формат ответа:

{

"chart\_type": "тип графика",

"total\_count": "число измерений",

"data": {

"labels": [...],

"values": [...],

"count\_values": [...] для некоторых типов

},

“limit”:”лимит” , для некоторых типов

“column”: “столбец”,

“aggregates” : “функция”, для некоторых типов

“metric\_column”: “метрический столбец” , для некоторых типов

}

3. Прогноз зарплаты:

Для обоих эндпоинтов:

{

"predicted\_salary": значение,

"model\_info": статистика модели, словарь с информаицей о модели лин. регрессии

"used\_parameters": примененные параметры, словарь с параметрами и значениями

}

4. Структура модели:

Словарь со словарями для каждого из столбцов-предикторов

**Обоснование выбора форматов**

1. Гибкость фильтрации и передачи данных

Для GET-запросов (таблицы, графики, структура модели):

Строковый формат параметров (filters, group\_by и др.) позволяет:

* + Передавать сложные условия через URL

Для POST-запросов (прогноз зарплаты):

Передача данных в теле запроса (JSON) обеспечивает:

* + Поддержку сложных структур (вложенные словари, массивы)
  + Отсутствие ограничений на длину параметров

2. Расширяемость

GET-эндпоинты:

* + Добавление новых параметров без изменения URL-структуры
  + Поддержка новых типов графиков через chart\_type

POST-эндпоинты:

* Возможность расширения тела запроса новыми полями
* Обратная совместимость за счет опциональных параметров

# ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

## Ориентировочная экономическая эффективность

В рамках курсового проекта расчёт экономической эффективности программного продукта не производился.

## Предполагаемая потребность

На данный момент на рынке нет интерактивных инструментов, позволяющих глубоко анализировать российский рынок труда аналитиков данных. Основные существующие ресурсы предоставляют данные только в формате статичных вакансий и резюме или статей, в которых нет возможности визуализировать интересующую пользователю информацию о рынке труда.

Разрабатываемый продукт:

* Упрощает процесс анализа данных за счет интерактивных визуализаций
* Предоставляет пользователям возможность находить актуальную информацию о рынке труда без необходимости вручную обрабатывать многочисленные вакансии

## Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

# Большинство сервисов, предназначенных для анализа рынка аналитиков данных, представляют собой статьи с проанализированной ситуацией, статичными графиками и выводами, однако такие сервисы не позволяют пользователю настраивать информацию под индивидуальные запросы.

# Такими сервисами являются habr.com, newhr.org ,navigator.hse.ru и т.д.

# Однако также существует и аналоги, такие как зарубежный сервис public.tableau.com, позволяющие персонализировать анализ под себя.

# Ниже представлена сравнительная таблица функционала аналогов:

Таблица 1. Сравнение функциональных характеристик с аналогами

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Критерий | habr.com | newhr.org | navigator.hse.ru | public.tableau.com | DataCareerInsight |
| Статичные графики: Наличие графиков без возможности изменения данных |  |  |  |  |  |
| Интерактивные графики: Возможность менять параметры визуализации (например, фильтрация, столбцы) |  |  |  |  |  |
| Возможность настройки фильтров: По таким параметрам, как регион, зарплата, опыт работы, навыки |  |  |  |  |  |
| Прогнозирование зарплаты: Расчет зарплат на основе введенных параметров (например, возраст, опыт, компетенции) |  |  |  |  |  |
| Обновляемость данных: Есть или же нет (статичные) |  |  |  |  |  |
| Доступ к данным о навыках: Анализ востребованных навыков в профессии |  |  |  |  |  |
| Анализ вакансий: Предоставление информации о текущих вакансиях (зарплата, требования, условия) |  |  |  |  |  |
| Анализ резюме: Возможность изучать данные о кандидатах (например, образование, ожидаемая зарплата) |  |  |  |  |  |
| Географический анализ: Визуализация распределения вакансий по регионам |  |  |  |  |  |
| Информация о работодателях: Предоставление данных о компаниях, предлагающих вакансии (название, тип компании — российская или иностранная) |  |  |  |  |  |
| Поддержка разных типов графиков: Гистограммы, круговые диаграммы, линейные графики |  |  |  |  |  |
| Сравнительные графики: Возможность сравнивать данные по различным критериям |  |  |  |  |  |
| Интуитивно понятный интерфейс: Простота использования для пользователей без технических навыков |  |  |  |  |  |
| Доступность на русском языке: Локализация для пользователей из России |  |  |  |  |  |

# Примечание:

# - присутствие следующего функционала

# - остутствие следующего функционала

# - частичное присутствие следующего функционала

# 

# Ключевые преимущества веб-сервиса DataCareerInsight:

# Широкий функционал с акцентом на интерактивность: В отличие от большинства статичных сервисов, таких как habr.com[[1]](#СПИСОК_ИСПОЛЬЗУЕМОЙ_ЛИТЕРАТУРЫ), newhr.org[[15]](#_bookmark50) или navigator.hse.ru[[16]](#_bookmark50), DataCareerInsight предоставляет пользователю возможность самостоятельно выбирать параметры анализа. Интерактивные графики и фильтры позволят адаптировать результаты под индивидуальные запросы.

# Учет резюме: Важной отличительной чертой станет наличие данных не только о вакансиях, но и о резюме аналитиков данных. Это позволит анализировать не только требования работодателей, но и предложения со стороны кандидатов, создавая полную картину рынка труда.

# Прогнозирование зарплат: Расчет ожидаемого диапазона зарплат на основе пользовательских данных, чего нет у большинства аналогов, включая Tableau[[2]](#СПИСОК_ИСПОЛЬЗУЕМОЙ_ЛИТЕРАТУРЫ).

# Актуальность веб-сервиса DataCareerInsight:

# Tableau Public[[2]](#СПИСОК_ИСПОЛЬЗУЕМОЙ_ЛИТЕРАТУРЫ) — единственный конкурент с мощным интерактивным функционалом, но он не предоставляет данных о резюме и не включает инструментов прогнозирования.

# Остальные сервисы значительно уступают по функционалу, так как предлагают только статичные графики и готовые отчеты, без возможности анализа данных в реальном времени.

# Таким образом, веб-сервис DataCareerInsight займет важное место среди существующих инструментов анализа рынка труда, так как он сочетает интерактивность, широкий функционал и ориентированность на локальный рынок.

# ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

1. ГОСТ 19.101-77: Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77: Стадии разработки. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77: Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78: Основные надписи. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78: Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78: Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.201-78: Техническое задание. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.301-79: Программа и методика испытаний. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.401-78: Текст программы. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ 19.404-79: Пояснительная записка. Требования к содержанию и оформлению.

// Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

1. ГОСТ 19.505-79: Руководство оператора. Требования к содержанию и оформлению.

// Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

1. ГОСТ 19.602-78: Правила дублирования, учета и хранения программных документов, выполненных печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.603-78: Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.604-78: Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. Исследование рынка труда аналитиков данных – 2023 [Электронный ресурс] / NewHR. – Режим доступа: <https://newhr.org/data/research-analysts-2023#technologies_and_skills> (дата обращения: 03.12.2024).
5. Навигатор образовательных программ [Электронный ресурс] / Национальный исследовательский университет «Высшая школа экономики». – Режим доступа: <https://navigator.hse.ru/> (дата обращения: 03.12.2024).
6. hh.ru [Электронный ресурс]. – Режим доступа: <https://hh.ru/> (дата обращения: 03.12.2024).
7. Анализ рынка труда аналитиков данных [Электронный ресурс] / Практикум от Яндекса. – Режим доступа: <https://habr.com/ru/companies/yandex_praktikum/articles/818111/> (дата обращения: 03.12.2024).
8. Визуализация данных о рынке труда [Электронный ресурс] / Tableau Public. – Режим доступа: <https://public.tableau.com/app/profile/roman4734/viz/BI_15968874028130/sheet0> (дата обращения: 03.12.2024).

# ПРИЛОЖЕНИЕ 1 ОПИСАНИЕ И НАЗНАЧЕНИЕ ФАЙЛОВ, ИСПОЛЬЗУЕМЫХ ПРИ ПАРСИНГЕ

Таблица 2. Описание и назначений файлов, используемых при парсинге

|  |  |  |
| --- | --- | --- |
| **Название** | **Назначение** | **Функционал** |
| database.py | Управление подключением к PostgreSQL | Создает соединение с БД через psycopg2  Принимает логин/парль (не хэшированные)  Подключается к локальной БД "vacancies\_db" |
| hash\_function.py | Безопасное хэширование данных | Реализует SHA-256 хэширование строк  Используется для защиты учетных данных  Преобразует пароли в 64-символьные хэши |
| token\_authentication.py | OAuth-аутентификация в API hh.ru | Получает CLIENT\_ID/SECRET из .env  Открывает браузер для авторизации пользователя  Обрабатывает callback с кодом авторизации  Обменивает код на access token |
| user\_authentication.py | Верификация пользователей парсера | Сравнивает хэши введенных данных с .env  Использует hash\_function.py для проверки  Возвращает True/False в зависимости от результата |
| currency\_table.py | Работа с курсами валют ЦБ РФ | Получает актуальные курсы через API Центробанка  Обновляет таблицу currency в PostgreSQL:  Добавляет новые валюты  Обновляет существующие курсы |
| date\_list\_generation.py | Генерация временных интервалов для парсинга с фильтром по дате | Создает список дат с заданным шагом (по умолчанию 2 дня)  Формат: YYYY-MM-DD  Используется для разделения запросов к API на периоды |
| get\_resume.py | Парсинг резюме с hh.ru | get\_full\_resume(): Получение полных данных по ID  escape\_array\_elements(): Обработка спецсимволов в массивах  get\_resumes\_with\_dates(): Основной цикл парсинга, где собирается вся информация о резюме и заносится в БД |
| get\_vacancies.py | Парсинг вакансий с hh.ru | Похожая логика с get\_resume: занесение вакансий в БД |
| constants.py | Хранение ключевых слов и списка профессий для фильтрации данных | Содержит предопределенные списки:  keywords (53 элемента) – фразы для поиска в названиях (например, "аналитик", "data scientist").  role\_names (20 элементов) – профессиональные роли (например, "BI-аналитик", "ML Analyst").  Используется для первичной фильтрации вакансий/резюме на этапе парсинга. |
| parser.py | Главный управляющий скрипт для запуска парсера | Аутентификация:  Получает OAuth-токен hh.ru (get\_access\_token).  Проверяет учетные данные БД через get\_user\_access.  Интерактивный режим:  Запрашивает у пользователя:  Логин/пароль для БД.  Тип данных для сбора (вакансии v, резюме r, валюты c).  Парсинг данных:  Для вакансий/резюме:  Итерируется по списку role\_names.  Запускает парсинг с разбивкой по датам (generate\_date\_list).  Сохраняет результаты в БД через get\_vacancies\_with\_dates или get\_resumes\_with\_dates.  Для валют: обновляет курсы через update\_currency\_table.  Отчетность:  Выводит количество добавленных записей.  Закрывает соединение с БД. |

# ПРИЛОЖЕНИЕ 2 ОПИСАНИЕ И НАЗНАЧЕНИЕ ФАЙЛОВ, ИСПОЛЬЗУЕМЫХ НА СЕРВЕРЕ

Таблица 3. Описание и назначений файлов, используемых на сервере

|  |  |  |
| --- | --- | --- |
| **Название** | **Назначение** | **Функционал** |
| models.py | Определение структуры таблиц базы данных (ORM-модели) | Содержит классы Vacancy и Resume, которые описывают структуру таблиц "vacancies" и "resume"  Определяет типы данных для каждого поля (String, Integer, ARRAY, Boolean и т.д.)  Задает первичные ключи и индексы  Позволяет SQLAlchemy создавать и управлять таблицами в PostgreSQL |
| db\_connection.py | Управление подключением к базе данных | Загружает параметры подключения из переменных окружения (.env)  Создает строку подключения (DATABASE\_URL) для PostgreSQL  Инициализирует движок SQLAlchemy (create\_engine)  Создает фабрику сессий (sessionmaker) для работы с БД  Предоставляет функцию get\_db() для получения и управления сессиями БД (используется в зависимостях FastAPI) |
| vacancies\_table.py | API-роутер для работы с таблицей вакансий | Предоставляет GET-эндпоинт /table/ с поддержкой:  Выбора конкретных полей (specific\_fields)  Сложной фильтрации с операторами >, <, =, != и логическими AND/OR (filters)  Группировки данных по одному или нескольким полям (group\_by)  Условий для агрегированных данных (having)  Агрегатных функций (avg, sum, count, max, min и др.)  Сортировки по одному или нескольким полям (sort\_by), а также фильтр на not null значения в столбце  Пагинации (limit/offset)  Использование модели Vacancy для типизированных запросов к БД |
| resumes\_table.py | API-роутер для работы с таблицей резюме | Аналогичный vacancies\_table функционал для таблицы резюме |
| fetch\_table.py | Ядро системы запросов к БД. Обрабатывает параметры и формирует SQL-запросы. | Унифицированная обработка параметров запроса:  Парсинг строковых параметров в структурированные данные  Валидация полей модели  Построение сложных SQL-запросов с:  Динамическим выбором полей  Множественными условиями фильтрации  Группировкой и агрегацией  Сортировкой по простым и агрегированным полям  Поддержка пагинации с подсчетом общего количества записей (total\_count)  Преобразование результатов в словари для JSON-сериализации  Обработка ARRAY-полей (развертывание для группировки, специальные фильтры) |
| sql\_functions.py | Библиотека вспомогательных функций для работы с SQLAlchemy | Парсинг сложных условий:  Фильтров (поддержка field>value, field=value1~value2~value3, AND/OR логики)  HAVING-условий для агрегированных данных  Работа с агрегатными функциями:  10+ встроенных функций (avg, sum, count, median, mode и др.)  Поддержка агрегации по массивам  Специальная обработка для статистических функций  Применение фильтров:  Для обычных полей (числа, строки, даты)  Для ARRAY-полей (поиск в массивах)  С учетом типов данных  Группировка данных:  Автоматическое развертывание массивов для группировки  Поддержка множественной группировки  Сортировка:  По простым и агрегированным полям  Поддержка NULLS LAST  Множественная сортировка  Дополнительные функции:  Проверка на NULL-значения  Обработка пустых массивов  Генерация правильных SQL-выражений |
| charts.py | Роутер FastAPI для генерации данных для визуализации (графиков) на основе данных из таблиц vacancies и resume | /column\_count/ – Генерация данных для столбчатых, круговых и точечных диаграмм, отображающих распределение значений в выбранном столбце.  Поддерживает модели vacancies и resume.  Группирует данные по указанному столбцу (column) и подсчитывает количество записей в каждой группе.  Возвращает данные в формате:  {  "chart\_type": "bar",  "data": {  "labels": ["Python", "SQL", "Excel"],  "values": [150, 120, 80]  }  }  Параметры:  model – выбор таблицы (vacancies или resume).  column – анализируемый столбец (например, skills).  chart\_type – тип графика (bar, pie, scatter).  filters – фильтры (аналогичные /table/).  limit – ограничение на количество групп.  /time\_distribution/ – Анализ временных трендов для метрического столбца (например, динамика зарплат по месяцам).  Поддерживает агрегации: avg, max, min, mode, median.  Возвращает данные в формате:  {  "data": {  "labels": [{"year": 2024, "month": 1}, {"year": 2024, "month": 2}],  "values": [120000, 125000], # Средние зарплаты  "count\_values": [50, 60] # Количество записей  }  }  Параметры:  column – анализируемый столбец (например, salary).  aggregates – агрегация (avg, max, min и др.).  Для вакансий используется published\_at, для резюме – updated\_at.  /metric\_column/ – Сравнение метрик (например, средняя зарплата по городам).  {  "data": {  "labels": ["Москва", "Санкт-Петербург"],  "values": [150000, 130000], # Средние зарплаты  "count\_values": [200, 150] # Количество вакансий  }  }  Параметры:  metric\_column – числовой столбец (например, salary).  aggregations – агрегация (avg, max, sum и др.).  column – столбец для группировки (например, address).  /metric\_distribution/ – Распределение числовых данных по диапазонам (гистограмма и круговая).  Разбивает значения на number\_range интервалов.  {  "data": {  "labels": [{"min\_value": 0, "max\_value": 50000}, ...],  "values": [10, 20, ...] # Количество записей в каждом диапазоне  }  }  column – числовой столбец (например, salary).  number\_range – количество диапазонов. |
| chart\_functions.py | Вспомогательные функции для обработки данных перед визуализацией | aggregate\_with\_count(arr, aggregate)  Вычисляет агрегированные значения (mode, max, min) с учетом частоты встречаемости.  array\_aggregate\_funcs  Словарь агрегационных функций для массивов:  avg, sum, max, min, median, stddev, variance, mode.  get\_data\_for\_time\_distribution()  Группирует данные по месяцам/годам и вычисляет агрегированные значения.  Поддерживает массивы (например, skills).  get\_data\_for\_metric\_distribution()  Разбивает числовые данные на диапазоны (для гистограмм).  get\_data\_for\_metric\_column()  Подготавливает данные для сравнения метрик (например, зарплата по городам).  is\_number()  Проверяет, является ли значение числом (поддержка int, float, строковых чисел). |
| constants.py | Хранение констант и настроек для регрессионных моделей | Содержит стандартные параметры для моделей (default\_vacancies\_data, default\_resume\_data)  Списки ключевых навыков (selected\_vacancies\_skills)  Столбцы для исключения из моделей (vacancies\_drop\_columns, resume\_drop\_columns)  Названия целевых переменных (vacancies\_y\_column, resume\_y\_column) |
| dataframe\_functions.py | Утилиты для работы с данными в Pandas DataFrame | load\_table() - загрузка данных из БД в DataFrame  Поддержка SQL-запросов через SQLAlchemy engine |
| model\_interpretation.py | Анализ и интерпретация результатов регрессионных моделей | get\_model\_results() - генерация полного отчета о модели:  Базовые метрики (R², Adjusted R², RMSE)  Анализ остатков (нормальность, гетероскедастичность)  Анализ значимости признаков  Формирование уравнения модели  Статистические тесты:  Anderson-Darling (нормальность)  Breusch-Pagan (гетероскедастичность) |
| regression\_functions.py | Основные функции для работы с регрессионными моделями | get\_model() - обучение модели с:  Автоматическим отбором значимых признаков (p-value < 0.05)  Контролем мультиколлинеарности (VIF < 10)  predict\_salary() - прогнозирование зарплаты на основе параметров  calculate\_vifs() - расчет фактора инфляции дисперсии |
| resume\_regression.py | Подготовка данных резюме для регрессионного анализа | prepare\_df\_for\_resume\_model():  Очистка и преобразование данных  Кодирование категориальных переменных  Обработка навыков, образования, опыта работы  Удаление выбросов (99-й перцентиль)  get\_structure\_for\_resume() - описание структуры модели для API |
| vacancies\_regression.py | Подготовка данных вакансий для регрессионного анализа | prepare\_df\_for\_vacancies\_model():  Расчет средней зарплаты  Группировка редких категорий  Векторизация навыков и адресов  Удаление неинформативных столбцов  get\_structure\_for\_vacancies() - описание структуры модели для API |
| get\_model\_structure.py | API-роутер для получения структуры регрессионных моделей | GET /model\_structure/:  Возвращает описание параметров модели  Поддержка моделей vacancies и resume  Интеграция с vacancies\_regression и resume\_regression |
| salary.py | API-роутер для прогнозирования зарплат | POST /vacancies\_salary/:  Прогноз по параметрам вакансии  Обработка: опыт, тип занятости, навыки, город  Возвращает предсказанную зарплату и метрики модели  POST /resume\_salary/:  Прогноз по параметрам резюме  Обработка: опыт, курсы, язык, навыки, ВУЗ и др.  Детальная информация о модели |
| main.py | Основной файл приложения FastAPI | Создание экземпляра FastAPI  Настройка CORS (разрешение всех доменов)  Подключение роутеров:  vacancies\_table - работа с вакансиями  resumes\_table - работа с резюме  charts - генерация графиков  get\_model\_structure - структура моделей  salary - прогнозирование зарплат  Запуск сервера через uvicorn |
| test\_chart.http | Тестирование эндпоинтов для работы с графиками | Тестирование 4 типов графиков:  column\_count - распределение значений по столбцам  time\_distribution - временные распределения  metric\_distribution - распределение числовых значений  metric\_column - сравнение метрик  Проверка работы с моделями vacancies и resume  Тестирование фильтрации данных  Проверка агрегаций (avg, mode, count) |
| test\_prediction.http | Тестирование эндпоинтов для прогнозирования зарплат | Получение структуры моделей (model\_structure)  Прогнозирование зарплат для вакансий (vacancies\_salary)  Прогнозирование зарплат для резюме (resume\_salary) |
| test\_resume\_table.http | Тестирование работы с таблицей резюме | Проверка всех параметров API:  Фильтрация (filters)  Группировка (group\_by)  Сортировка (sort\_by)  Агрегация (aggregates)  Пагинация (limit, offset)  Тестирование сложных запросов:  Множественная группировка  Комбинированные фильтры  Условия HAVING |
| test\_vacancy\_table.http | Тестирование работы с таблицей вакансий | Аналогично test\_resume\_table.http, но для вакансий |

# ПРИЛОЖЕНИЕ 3 ОПИСАНИЕ И НАЗНАЧЕНИЕ ФАЙЛОВ, ИСПОЛЬЗУЕМЫХ НА КЛИЕНТСКОЙ ЧАСТИ (ФРОНТ)

Таблица 4. Описание и назначений файлов, используемых на клиентской части (фронт)

|  |  |  |
| --- | --- | --- |
| **Название** | **Назначение** | **Функционал** |
| resume.config.js | Конфигурация полей и операторов для работы с таблицей резюме | Поля (fields):  Список всех доступных полей с русскоязычными labels  Операторы (operators):  Поддерживаемые операторы сравнения (=, >, <, !=, == для поиска подстрок).  Типы полей:  Числовые (numericFields): age, salary, total\_experience.  Даты (dateFields): created\_at, updated\_at.  Агрегации:  Для чисел: avg, sum, median.  Для нечисловых данных: mode, count.  Динамические операторы:  Функция getFilteredOperators(field) возвращает допустимые операторы для выбранного поля (например, для дат исключает ==). |
| vacancies.config.js | Аналогичная конфигурация для таблицы вакансий | Все тоже самое, только другие поля/столбцы |
| Filters.js | Компонент для добавления и управления фильтрами таблицы | Интерфейс:  Выпадающие списки для выбора поля, оператора и логики (AND/OR).  Поле ввода значений (с поддержкой нескольких значений через запятую).  Кнопки быстрого выбора агрегаций (например, avg, max).  Логика:  Динамическая фильтрация операторов (например, для дат нельзя выбрать ==).  Обработка дат через type="date".  Поддержка сложных условий (например, skills==SQL,Python).  Параметры из Filters напрямую передаются в API-запросы  Визуализация активных фильтров:  Чипсы с возможностью удаления.  Анимация раскрытия/сворачивания (Accordion). |
| GroupBy.js | Компонент для группировки данных по выбранным столбцам | Выбор столбцов:  Модальное окно с чекбоксами.  Поддержка множественного выбора.  Управление:  Удаление отдельных столбцов из группировки.  Кнопка сброса.  Параметры из GroupBy напрямую передаются в API-запросы  Визуализация:  Чипсы с названиями выбранных столбцов.  Accordion для компактного отображения. |
| Aggregates.js | Компонент для настройки агрегаций (аналитических вычислений) по полям таблицы | Выбор столбца и типа агрегации (среднее, сумма, медиана и др.)  Динамическое обновление доступных агрегаций в зависимости от типа поля (числовые/даты/текст)  Добавление/удаление агрегаций  Визуализация активных агрегаций в виде чипсов  Поддержка сброса всех агрегаций  Форматирование отображаемых названий полей и агрегаций  Интеграция с API через колбэк onApplyAggregates |
| DynamicTable.js | Компонент для отображения таблицы с данными | Адаптивное отображение таблицы с sticky header  Управление видимостью столбцов через меню  Быстрое переключение столбцов через чипсы  Форматирование агрегированных полей (название\_поля\_агрегация → "Название (Агрегация)")  Обработка сложных типов данных (массивы, объекты)  Отображение состояния "нет данных"  Поддержка русскоязычных названий полей |
| HavingFilters.js | Компонент для фильтрации по агрегированным полям (HAVING в SQL) | Выбор агрегированного поля, оператора и значения  Динамическая фильтрация доступных операторов  Добавление/удаление условий HAVING  Форматирование отображаемых условий (поле (агрегация) оператор значение)  Поддержка сброса всех фильтров  Визуализация активных фильтров в виде чипсов  Интеграция с API через колбэки onAdd/onRemoveHavingFilter |
| NotNullFilter.js | Фильтр для исключения строк с пустыми значениями | Выбор столбцов через выпадающее меню с чекбоксами  Управление фильтром через чипсы (добавление/удаление)  Поддержка сброса фильтра  Форматирование русскоязычных названий полей  Интеграция с API через колбэк onApplyNotNullFilter  Индикация количества выбранных столбцов |
| SortBy.js | Компонент для настройки сортировки данных | Выбор столбца и направления сортировки (по возрастанию/убыванию)  Поддержка сортировки по агрегированным полям  Добавление нескольких условий сортировки  Удаление отдельных условий  Форматирование отображаемых условий (поле ↑/↓)  Поддержка сброса всех условий  Интеграция с API через колбэк onApplySortBy  Визуализация активных условий в виде чипсов |
| GenericTable.js | Универсальный компонент для отображения и управления табличными данными | Получение данных с сервера через API с поддержкой пагинации  Интеграция всех компонентов фильтрации (Filters, GroupBy, Aggregates и др.)  Управление видимостью столбцов  Поддержка изменения высоты строк (resizable rows)  Настройка количества отображаемых строк  Цветовые темы для разных типов данных  Обработка и отображение агрегированных данных  Локализация названий полей  Отображение состояния загрузки  Полноценная пагинация с настройкой  Возможность сброса всех фильтров и настроек |
| ResumeTable.js | Специализированный компонент таблицы для отображения данных резюме | Использует GenericTable с предустановленной конфигурацией для резюме  Задает API-эндпоинт для получения данных резюме  Применяет конфигурацию полей из resumeFieldsConfig  Скрывает технические поля по умолчанию (id\_resume, created\_at и др.)  Устанавливает цветовую тему "resume" |
| VacanciesTable.js | Специализированный компонент таблицы для отображения данных вакансий | Использует GenericTable с предустановленной конфигурацией для вакансий  Задает API-эндпоинт для получения данных вакансий  Применяет конфигурацию полей из vacanciesFieldsConfig  Скрывает технические поля по умолчанию (id, currency, experience и др.)  Использует стандартную цветовую тему |
| ResizableTableRow.js | Компонент строки таблицы с возможностью изменения высоты | Поддержка перетаскивания для изменения высоты строки  Визуальные индикаторы при наведении и изменении размера  Ограничение минимальной и максимальной высоты  Колбэк для оповещения об изменении высоты  Плавные переходы и анимации  Стилизация через MUI |
| useRowResize.js | Кастомный хук для управления изменением высоты строк таблицы | Отслеживание состояния изменения высоты строк  Обработка событий мыши (нажатие, перемещение, отпускание)  Ограничение высоты строк (min/max)  Управление курсором во время изменения размера  Поддержка hover-эффектов  Отписка от событий при размонтировании  Возвращает текущие высоты строк и методы управления |
| SalaryPredictorTabs.js | Компонент для переключения между калькуляторами зарплат (вакансии и резюме) | Переключение между вкладками "Вакансии" и "Резюме"  Анимация плавного перехода между вкладками (Fade)  Стилизованные кнопки с индикацией активной вкладки  Интеграция компонентов VacancySalaryPredictor и ResumeSalaryPredictor |
| SalaryPredictionResult.js | Отображение результатов предсказания зарплаты | Визуализация предсказанной зарплаты с доверительным интервалом  Детализация метрик модели (R², RMSE и др.)  Анализ остатков (нормальность, гетероскедастичность)  Таблицы влияющих факторов (положительные/отрицательные)  Метаинформация о модели (название, выборка, уравнение)  Форматирование чисел для удобного чтения |
| vacancy\_salary.js | Калькулятор зарплаты для вакансий | Загрузка структуры модели с сервера  Форма ввода параметров (опыт, тип занятости, навыки и др.)  Множественный выбор навыков (чечекбоксы)  Валидация числовых полей  Отправка данных на сервер для предсказания  Сброс формы  Интеграция с SalaryPredictionResult для отображения результатов |
| resume\_salary.js | Калькулятор зарплаты для резюме | Загрузка структуры модели с сервера  Расширенная форма ввода (опыт, курсы, язык, водительские права и др.)  Слайдер для уровня английского с маркерами (A1-C2)  Множественный выбор для навыков, графиков, вузов и др.  Переключатель для водительских прав  Отправка данных на сервер и обработка результатов  Сброс формы  Интеграция с SalaryPredictionResult |
| useChartData.js | Универсальный хук для работы с API графиков | Получает данные с сервера по указанному эндпоинту  Поддерживает фильтрацию, пагинацию и параметры запроса  Обрабатывает состояния загрузки и ошибок  Позволяет повторно запрашивать данные (refetch) |
| useChartFilters.js | Управление фильтрами для графиков | Добавление, удаление и сброс фильтров  Преобразование фильтров в строку для API-запроса  Интеграция с компонентами визуализации |
| translations.js | Централизованное хранение переводов. | Хранение переводов, необходимых для русскоязычного отображения параметров графиков |
| ColumnCountChart.js | Компонент для отображения количественного распределения данных | Выбор столбца для анализа  Переключение типа графика (столбчатый/круговой/точечный)  Настройка лимита отображаемых значений  Фильтрация данных  Адаптивная цветовая схема для разных моделей |
| ColumnCountChartVisualizations.js | Визуализации для количественного анализа | Столбчатая диаграмма (BarChart)  Круговая диаграмма (PieChart)  Точечная диаграмма (ScatterChart)  Настройка цветовых схем  Кастомизация отображения (подписи, легенды, tooltips)  Интеграция с useChartData/useChartFilters |
| TimeDistributionChartVizualizations.js | Компонент для визуализации временного распределения данных (динамики изменений) с поддержкой различных типов графиков. | Типы графиков:  Bar, Line, Scatter  Особенности данных:  Поддержка числовых (isNumeric) и категориальных данных.  Форматирование дат в читаемый вид (например, "Январь 2023").  Отображение дополнительной информации в tooltips:  Для числовых данных: значение и количество наблюдений.  Для категориальных: мода (наиболее частое значение) и количество наблюдений.  Кастомизация:  Цветовые схемы для разных типов данных (вакансии – синяя, резюме – фиолетовая).  Гибкие настройки осей, заголовков и подписей.  Адаптивные подписи (поворот на 45° для длинных текстов).  Интерактивность:  Информативные всплывающие подсказки (tooltips).  Настройка внешнего вида точек/столбцов (радиус, цвет при наведении). |
| TimeDistributionChart.js | Визуализация динамики изменений данных вакансий/резюме по времени | Типы графиков: линейный, столбчатый, точечный  Поддержка агрегаций (avg, sum, max для чисел; mode для текста)  Фильтрация данных через компонент Filters  Цветовые схемы: синяя (вакансии), фиолетовая (резюме)  Локализация названий полей и агрегаций  Интеграция с useChartData/useChartFilters |
| MetricColumnChart.js | Визуализация агрегированных данных по выбранному признаку | Типы графиков: столбчатый, линейный, круговой  Агрегации: avg, sum, max, min, median  Фильтрация данных  Лимит отображаемых значений (5-50)  Цветовые схемы для вакансий/резюме  Интеграция с useChartData/useChartFilters |
| MetricColumnChartVizualizations.js | Отрисовка графиков сравнения показателей | BarChart: столбчатая диаграмма с настройкой осей  LineChart: линейный график с заполнением  PieChart: круговая диаграмма с цикличной палитрой  Интерактивные подсказки с данными |
| MetricDistributionChart.js | Анализ распределения числовых данных по диапазонам | Настройка количества диапазонов (2-20)  Типы графиков: столбчатый, круговой  Фильтрация данных  Динамическое обновление при изменении параметров  Интеграция с useChartData/useChartFilters |
| MetricDistributionChartVizualizations.js | Отрисовка графиков распределения | BarChart: отображение количества значений в диапазонах  PieChart: круговая диаграмма с автоматической палитрой  Форматирование подписей диапазонов  Контекстные подсказки |
| Header.js | Переключение между страницами приложения | Кнопки для главной, вакансий, резюме и прогнозирования  Подсветка активной страницы (NavLink с active-стилем)  Минималистичный дизайн |
| ScrollToTop.js | Автоматический скролл страницы при переходе между маршрутами | Следит за изменением pathname через useLocation  Вызывает window.scrollTo(0, 0) при смене маршрута |
| Home.js | Страница с вводной информацией о проекте | Hero-секция с градиентным фоном  Карточки возможностей (анализ вакансий/резюме, прогнозирование)  Статистика данных  Декоративные "кляксы" для визуального акцента |
| Vacancies.js | Страница с визуализацией вакансий | Таблица (VacanciesTable)  4 типа графиков:  ColumnCountChart (количественный анализ)  TimeDistributionChart (динамика изменений)  MetricColumnChart (сравнение показателей)  MetricDistributionChart (распределение значений)  Голубая тема с градиентным заголовком |
| Resumes.js | Страница с визуализацией данных соискателей | Таблица (ResumeTable)  Те же 4 типа графиков, что и для вакансий  Фиолетовая цветовая схема |
| LinearRegression.js | Страница прогнозирования зарплат | Контейнер для SalaryPredictorTabs (вакансии/резюме)  Описание модели (учет опыта, навыков, региона)  Зеленая цветовая схема с анимированным фоном |
| App.js | Корневой компонент | Подключение всех страниц через react-router-dom  Общий контейнер с Header и ScrollToTop  Маршруты: /, /vacancies, /resumes, /linear-regression |

# ПРИЛОЖЕНИЕ 4 ОПИСАНИЕ СТРУКТУРЫ ТАБЛИЦЫ ВАКАНСИЙ В БАЗЕ ДАННЫХ

Таблица 5. Описание структуры таблицы вакансий в базе данных

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Пояснение** |
| id | Integer | Идентификатор вакансии (первичный ключ) |
| title | Text | Название вакансии |
| company\_name | Text | Название компании |
| currency | Text | Валюта |
| experience | Text | Опыт («нет опыта», «от 1 до 3 лет», «от 3 до 6 лет», «более 6 лет») |
| type\_of\_employment | Text | Тип занятости («Полная занятость», «Частичная занятость», …) |
| work\_format | Text | Формат работы («Полный день», «Гибкий график», …) |
| description | Text | Описание вакансии (на сайт по итогу не попало, но в дальнейшем можно использовать для извлечения навыков) |
| skills | Text[] | Список навыков |
| address | Text | Город |
| published\_at | Timestamp | Время публикации вакансии |
| archived | Boolean | Архивирована ли вакансия или нет |
| url | Text | Ссылка на вакансию |
| min\_experience | Integer | Мин. опыт (автоматически с помощью триггера проставляется из experience) |
| max\_experience | Integer | Макс. опыт (автоматически с помощью триггера проставляется из experience) |
| salary\_to | Integer | От какой суммы начинается заработная плата |
| salary\_from | Integer | До какой суммы заработная плата |
| russian\_salary\_to | Numeric | От какой суммы начинается заработная плата, в рублях (с помощью триггера и таблицы currency автоматически добавляются и изменяются значения ) |
| russian\_salary\_from | Numeric | До какой суммы заработная плата, в рублях (с помощью триггера и таблицы currency автоматически добавляются и изменяются значения ) |

# ПРИЛОЖЕНИЕ 5 ОПИСАНИЕ СТРУКТУРЫ ТАБЛИЦЫ РЕЗЮМЕ В БАЗЕ ДАННЫХ

Таблица 6. Описание структуры таблицы резюме в базе данных

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Пояснение** |
| id\_resume | Text | Идентификатор резюме (первичный ключ) |
| title | Text | Название резюме |
| created\_at | Timestamp | Время создания резюме |
| updated\_at | Timestamp | Время последнего обновления |
| age | Integer | Возраст (в годах) |
| gender | Text | Пол |
| salary | Numeric | Зарплата |
| russian\_salary | Numeric | Зарплата в рублях (аналогично таблице с вакансиями вычисляется автоматически за счет триггера и таблице currency) |
| currency | Text | Валюта |
| photo | Boolean | Есть фото или нет |
| total\_experience | Integer | Общий опыт (в месяцах) |
| citizenship | Text[] | Гражданство (может быть несколько) |
| area | Text | В каком городе живет |
| level\_education | Text | Уровень образования |
| university | Text[] | В каких университетах учился (может быть несколько) |
| count\_additional\_courses | Integer | Кол-во пройденных дополнительных курсов для повышения квалификации |
| employments | Text[] | Типы занятости («Полная занятость», «Частичная занятость») |
| experience | Text[] | В каких компаниях работал |
| language\_eng | Text | Уровень английского языка (a1-c2) |
| language\_zho | Text | Уровень китайского языка (a1-c2) |
| schedules | Text[] | График работы («Полный день», «Гибкий график», …) |
| skill\_set | Text[] | Список навыков |
| is\_driver | Boolean | Есть водительское удостоверение или нет |
| professional\_roles | Text[] | Профессиональные роли |
| url | Text | Ссылка на резюме |

# ПРИЛОЖЕНИЕ 6 ОПИСАНИЕ СТРУКТУРЫ ТАБЛИЦЫ ВАЛЮТ В БАЗЕ ДАННЫХ

Таблица 7. Описание структуры таблицы валют в базе данных

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Пояснение** |
| name\_currency | Text | Название валюты (первичный ключ) |
| course\_to\_russia | Numeric | Курс валюты к рублю (при добавлении новой валюты или при изменении курса существующей, все зарплаты в таблицах вакансий и резюме пересчитываются на новый курс благодаря триггеру) |

# ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий  № сопроводит ельного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулирован ных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |