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

Институт радиоэлектроники и информационных технологий – РТФ
Школа бакалавриата

ОТЧЕТ

По проекту
«Разработка парсера для сбора информации о стажировках»

по дисциплине «Проектный практикум»

Заказчик: **Смирнов Д. С.**
Куратор: **Пушкарь Ю.А.**
Студенты команды: **Polymatrix**
Ергин И.К
Стишенко Е.В
Ефтеев С.В.
Микрюков Д.А.
Судоплатов В. М.

Екатеринбург, 2025

Содержание

1. Введение

Целью проекта является разработка парсера, способного собирать информацию о стажировках с различных ресурсов по заданным параметрам. Результатом должен стать работающий прототип, реализующий парсинг по ключевым словам с выгрузкой всех найденных стажировок в. json файл.

Проект инициирован компанией, занимающейся организацией стажировок, с целью автоматизировать сбор информации о внешних стажировках и тем самым увеличить посещаемость своих цифровых ресурсов.

2. Основная часть

2.1 Вклад участников команды

Ергин Игорь (Тим-лид):

- Построение рабочего процесса команды, ведение бэклога с четко описанными задачами и выставление по ним дедлайнов.
- Прововедение контроля по соблюдению сроков исполнения задач и качеству их выполнения.
- Организация связи с заказчиком и куратором для своевременного уточнения направления разработки в случае возникновения вопросов.
- Организация регулярных внутрикомандных собраний для поддержания вовлеченности всех участников в разработку проекта и нахождения решений по возникающим вопросам.
- Защита наработок команды: составление презентации, запись видеоотчетов.

Ефтеев Станислав (Аналитик):

- Постановка цели проекта, анализ целевой аудитории и ожиданий заказчика;
- Составление декомпозиции бизнес-процесса: определение этапов сбора, обработки и представления информации;
- Проведение анализа и сопоставления агрегаторов: hh.ru, trudvsem.ru, Habr Career, SuperJob;
- Выработка критериев выбора источников: уникальность контента, наличие API, полнота данных, актуальность;
- Участие в разработке user flow — последовательности действий пользователя от запуска бота до получения результатов;

- Тестирование Telegram-бота: проверка работы фильтров, обработки команд, стабильности под нагрузкой.
- Обоснование выбора hh.ru и trudvsem.ru как основных источников для сбора стажировок;

Стишенко Евгений (Аналитик, дизайнер):

- Подготовка анализа сайтов крупных компаний, публикующих стажировки, с акцентом на город Екатеринбург;
- Составление перечня сайтов, непригодных для парсинга, и обоснование причин (отсутствие фильтров по городу, низкий объём стажировок);
- Участие в разработке и визуализации user flow для обычного пользователя и администратора;
- Проектирование интерфейсов Telegram-бота в Figma (в том числе состояния фильтрации, выбор сайтов, админ-панель);
- Формулировка вопросов для заказчика, необходимых для уточнения требований и будущих точек публикации;
- Создание критериев тестирования, включая фильтрацию, обработку ошибок API и проверку стабильности;

Микрюков Денис (Разработчик):

- Разработка структуры проекта: следил за структурой проекта и взаимодействием разных модулей.
- Разработка телеграм-бота: работал над разделом фильтров, подключал функции бота к БД.
- Создание реляционной базы данных: создал БД с необходимыми таблицами и связями и внедрил python-код для работы с БД.
- Разработка API через FastAPI: работал над функциями API для взаимодействия удаленных запросов с БД.

- Контейнеризация всего проекта с помощью Docker: контейнезировал проект, составил docker-compose, проверил работу на удаленном сервере.

Судоплатов Владислав (Разработчик):

- Разработка телеграм-бота: работал над первоначальной версией телеграм-бота, создал шаблон для всех страниц.
- Разработка парсера с агрегатора hh.ru: изучил документацию API hh.ru и создал асинхронный парсер с сайта hh.ru с использование API. Столкнулся с проблемой блокировки запросов (ошибка 403). Для её решения использовал прокси-сервера, что позволило распределять нагрузку и избегать IP-банов.
- Разработка парсера с агрегатора trudvsem.ru: изучил документацию API trudvsem.ru и реализовал асинхронный парсер с сайта trudvsem.ru с использование API.

2.2 Требования заказчика и backlog

Требования заказчика:

1. возможность сбора стажировок с сайтов;
2. возможность выгрузки результатов в виде .json файла;
3. простота использования;

Backlog включал:

- анализ агрегаторов и сайтов компаний;
- разработка user-flow;
- реализация фильтров;
- разработка интерфейса Telegram-бота;
- подключение парсера и базы данных;
- тестирование на корректность формирования и вывода .json файла на стабльном уровне;

- анализ целевой аудитории;
- разработка бизнес-процесса.

2.3 Анализ аналогов

Были рассмотрены следующие сайты:

- **hh.ru** — основной источник, содержит фильтры, подробную информацию о стажировках и удобный API.
- **trudvsem.ru** — содержит государственные стажировки, поддерживает региональные фильтры.
- **Habr Career** — ориентирован на ИТ-специальности, доступен парсинг через HTML.
- **SuperJob** — требует API-ключ, используется ограниченно.

Вывод: hh.ru и trudvsem.ru - эти сайты будут использоваться для парсинга, так как имеют большое количество разнообразных и уникальных стажировок, качественную фильтрацию, полное и удобное описание каждой стажировки, а также они доступны для парсинга.

2.4 Архитектура программного продукта

Система построена на архитектуре клиент-сервер:

- **Клиент** - **Telegram-бот** на базе **aiogram** с кнопками и вводом данных, предназначенный для взаимодействия с пользователем.
- **Сервер** - Python-скрипт с логикой фильтрации и парсинга или API эндпоинты
- **Хранилище** - Реляционная база данных (БД) для хранения информации о стажировках, с возможностью масштабирования и работы с большими объёмами данных

Важным компонентом архитектуры является разработка RESTful API с использованием FastAPI. Это решение обеспечило возможность удалённого взаимодействия с БД через HTTP-запросы, что позволяет системе быть гибкой и масштабируемой, а также позволяет взаимодействовать с другими

веб-ресурсами, такими как сайты компаний и агрегаторы вакансий. API включает несколько ключевых функций:

- Получение стажировок по фильтрам, указанным пользователем (например, профессия, оклад, тип занятости и др.);
- Удаление старых и добавление новых стажировок в БД, при этом все запросы обрабатываются асинхронно, что позволяет снижать нагрузку на сервер;
- Обработка запросов от сторонних сервисов, что позволяет взаимодействовать с БД через API с любых источников;
- Обработка ошибок и логирование запросов, что позволяет отслеживать сбои и своевременно их устранять.

Разработка API была необходима для эффективного взаимодействия между БД и внешними сервисами. Использование FastAPI позволило быстро разработать высокопроизводительное решение с поддержкой асинхронных запросов, что оптимизирует работу с БД и повышает скорость отклика.

Выбор архитектуры обусловлен простотой, возможностью масштабирования и минимальными затратами. В разработке использовались модули requests для парсинга, aiogram для создания бота в Telegram, fastapi для разработки API, а также mysql для взаимодействия модулей с БД.

2.5 Методология и процесс разработки

Использовалась **итерационная модель разработки**:

- каждый этап проходил через анализ - реализацию - тестирование;
- применялась Google-таблица задач (backlog);
- Результативность отслеживалась и фиксировалась в Протоколе собраний
- общение происходило в Telegram-чате / Microsoft Teams.

Промежуточное тестирование проводилось по следующим критериям:

1. корректность работы фильтров;
2. стабильность при множественных запросах;
3. отображение ошибок при сбое API;
4. правильная выгрузка файлов .json.

Были выявлены и устранены баги с фильтрацией и некорректным выводом данных в JSON

2.6 Планирование и распределение задач

Планирование велось по задачам:

1. на первом этапе — аналитика и прототипирование;
2. далее — параллельная работа над парсером и Telegram-ботом;
3. в финальных итерациях — тестирование и отладка.

Распределение:

- аналитики — исследование источников, документации, формирование требований;
- разработчики — техническая реализация логики парсинга и интерфейса;
- дизайнер — создание Figma-макетов;
- тимлид — общее управление, ведение документации и взаимодействие с заказчиком.

3. Заключение

В результате проделанной работы удалось реализовать программный продукт, соответствующий ключевым требованиям заказчика: реализована возможность сбора стажировок с нескольких источников, реализованы основные фильтры, предусмотрена выгрузка результатов в .json файл. Функциональность, необходимая для правильной работы продукта реализована в полном объеме.

Тестирование показало, что продукт работает стабильно при множественных запросах, корректно обрабатывает ошибки, а также сохраняет и выводит результаты без искажений. Были выявлены отдельные недочеты в логике фильтрации и отображения данных, которые были устранены в ходе итераций.

В качестве направления для развития проекта можно выделить следующие пункты:

- расширение числа источников;
- переход от файловой БД к полноценному серверному хранилищу;
- доработка разнообразия фильтров;

Таким образом, продукт имеет потенциал для масштабирования, уже решает основную задачу по агрегации стажировок и может быть внедрен как внутренняя система для поиска студентами актуальных предложений.

Успешный результат был достигнут благодаря четкому распределению ролей в команде, грамотному планированию и постоянной обратной связи с заказчиком.

Приложения А

Название приложения

FIGMA:

https://www.figma.com/design/nywWBvLs5ntyNu1pAeGijm/Untitled?node-id=0-1&t=EXr3XLIR7R7C8o1a-1

Протокол собраний:

https://docs.google.com/document/d/1tTAeFgS3qKa3Dd5abcp8NYZO3-NaxL8i2iEFO9KJRQ4/edit?tab=t.9fn16k9qy3gd

Файлы команды:

https://drive.google.com/drive/folders/1RyWHOVt9NLaa_QnCGSmKuRfH9zR0rsIH