

# 1. Общие сведения

Название проекта: IT Start Bot.

Сущность проекта: Учебный проект. Разработка Telegram-бота для автоматизации поиска стажировок, вакансий и конференций.

Заказчик: Крючкова Александра Андреевна, преподаватель СГУ.

Исполнитель: Команда студентов СГУ, факультет КНИИТ, 5 курс, 551 группа, з.о.

Сроки реализации: 3 месяца (MVP).

Бюджет: 0 рублей (в рамках учебного проекта работа осуществляется безвозмездно).

## 2. Цели и задачи проекта

Основная цель:

Создать полезный инструмент для IT-студентов, автоматизирующий поиск релевантных стажировок, вакансий и конференций с перспективой монетизации.

Целевая аудитория:

- Студенты IT-направлений
- Начинающие разработчики, ищащие первый опыт
- Участники конференций и хакатонов

Ключевые задачи:

- Автоматизировать сбор данных из различных источников (hh.ru, корпоративные сайты, Telegram-каналы и т.д.)
- Обеспечить персонализированный поиск с фильтрацией по технологиям, локации и типу возможностей.
- Реализовать систему уведомлений с настраиваемыми подписками и напоминаниями о дедлайнах.
- Создать административную панель для модерации контента и аналитики.
- Обеспечить высокую производительность и надежность (поддержка 5000+ пользователей, 99.9% uptime).

### 3. Требования к продукту

#### 3.1. Функциональные требования

##### 3.1.1. Парсинг и сбор данных

- Категории публикаций: вакансии, стажировки, конференции, хакатоны и контесты.
- Автоматический парсинг из различных источников: hh.ru, Habr Career и др., корпоративные сайты (Яндекс, Тинькофф, VK, Сбер и др.), агрегаторы конференций, Telegram-каналы и чаты с вакансиями.
- Расписание обновлений: каждые 6 часов для вакансий (12:00, 18:00, 00:00, 06:00 МСК), ежедневно для остальных типов публикаций (12:00 МСК).
- Обработка ошибок: повторные попытки парсинга с интервалами 15 мин (2-я попытка) и 45 мин (3-я попытка).
- Управление дубликатами: автоматическое определение по URL + название + компания, без участия администратора.
- Жизненный цикл данных: автоматическое удаление записей старше 3 месяцев.
- Автоматическое извлечение тегов из данных. Возможные теги: тип (стажировка, практика, конференция, вакансия, хакатон, конкурс), локация (онлайн, или город), технологии – например, python, ML, java, android и т.д.

##### 3.1.2. Основной интерфейс Telegram-бота

Основные команды: /start, /help, /internships, /jobs, /conferences, /contests, /subscribe, /unsubscribe

Фильтрация: по тегам (тип: стажировка/практика/конференция/вакансия/хакатон), навыкам, городу/онлайн, технологиям (разработка/ML).

Пример использования команды с фильтрацией:

```
/jobs #ml #remote #saratov #python
```

или

```
/jobs ml remote саратов python
```

Набор тегов для поисковых команд может быть любым и является опциональным.

Язык интерфейса: русский.

Также доступны inline-кнопки, повторяющие функциональность команд.

### 3.1.3. Система уведомлений и рассылок

Персонализированные рассылки: множественный выбор типов уведомлений (стажировки, практики, вакансии, конференции). Настраивается специальной командой /settings.

Частота: по мере поступления новой информации.

Напоминания о дедлайнах: за 3 дня до окончания (опционально, с возможностью подписки/отписки).

Каждый пост должен содержать:

- Название
- Ссылка на оригинал (источник)
- Краткое описание (основные требования, дедлайны, условия)
- Контактные данные (email, форма подачи, ссылка на анкету)
- Теги (например, #стажировка #конференция #backend)

### 3.1.4. Административная WEB-панель

Данные: публикации (вакансии, конференции, стажировки, хакатоны), теги, пользователи системы, источники.

Управление контентом: добавление/редактирование/удаление публикаций.

Управление источниками: блокировка/разблокировка источников парсинга, добавление новых.

Статистика: количество активных пользователей, топ-5 запрашиваемых навыков, процент ошибок парсинга, количество добавленных вакансий по дням.

Система ролей: администратор (полный доступ) и модератор (доступ только к публикациям).

Фильтрация данных по категориям (вакансии, конференции, стажировки, мероприятия) и дате публикации.

Язык интерфейса: русский.

### 3.1.5. Техническая инфраструктура

Резервное копирование: ежедневные бэкапы БД со сроком хранения 3 месяца.

Логирование: журнал всех действий пользователей (Grafana), ошибок парсинга и системных событий (Sentry / Loggly).

Мониторинг: специальная Telegram-группа для разработчиков с уведомлениями о критических ошибках.

### **3.1.6. Экспорт данных**

Экспорт во внешние системы: интеграция с Google Sheets.

Экспорт данных в CSV и MS Excel.

### **3.1.7. Telegram-канал**

Создание Telegram-канала, куда в автоматическом режиме публикуются материалы по мере их появления в базе.

## **3.2. Нефункциональные требования**

### **3.2.1. Производительность**

Время отклика: < 1 секунды на команды пользователей.

Парсинг: < 5 минут на обработку одного источника.

Пропускная способность: обработка 100+ новых вакансий ежедневно.

Нормальная нагрузка: поддержка 5000+ одновременных пользователей.

Кеширование данных.

### **3.2.2. Надежность**

Доступность: 99.9% uptime (не более 8 часов простоя в год).

Отказоустойчивость: автоматическое восстановление после сбоев.

Мониторинг: 24/7 отслеживание состояния системы через Sentry + Grafana.

Резервирование: горячие бэкапы критически важных компонентов.

### **3.2.3. Безопасность**

Шифрование данных: PGP для контактной информации (email/телефоны).

Аутентификация: 2FA для администраторов.

Контроль доступа: админ-панель доступна только через VPN/Whitelist IP.

Валидация: защита от SQL-инъекций и других атак через входящие сообщения.

Аудит: логирование всех административных действий.

### **3.2.4. Масштабируемость**

Модульная архитектура: добавление новых источников парсинга без переписывания существующего кода.

### **3.2.5. UX/UI требования:**

Интуитивность: четкие инструкции для новых пользователей.

Доступность: соответствие принципам универсального дизайна.

Отзывчивость: мгновенная обратная связь на действия пользователя.

### **3.2.6. Технический стек:**

- Python 3.10+ (Aiogram)
- PostgreSQL + Redis
- Docker
- Sentry/Loggly для мониторинга ошибок
- Grafana для метрик
- Git + GitHub

## 4. Организационная структура

Роль	Описание	Основные задачи
Руководитель проекта	Управляет сроками, бюджетом, коммуникацией с заказчиком	<ul style="list-style-type: none"><li>• Формирование Roadmap;</li><li>• Контроль выполнения спринтов;</li><li>• Решение организационных вопросов</li></ul>
Координатор проекта	Помощник руководителя, отвечает за документацию и процессы	<ul style="list-style-type: none"><li>• Ведение backlog;</li><li>• Организация митингов;</li><li>• Отслеживание прогресса</li></ul>
Техлид проекта	Технический лидер, отвечает за архитектуру и код-ревью	<ul style="list-style-type: none"><li>• Распределение задач;</li><li>• Контроль качества кода;</li><li>• Решение технических блокеров</li></ul>
Разработчики	Пишут код бота (Python + Aiogram)	<ul style="list-style-type: none"><li>• Реализация парсеров;</li><li>• Настройка команд бота;</li><li>• Интеграция с БД</li></ul>
Разработчики БД	Проектируют и оптимизируют базу данных	<ul style="list-style-type: none"><li>• Создание схемы PostgreSQL;</li><li>• Настройка индексов;</li><li>• Миграции данных</li></ul>
Девопсы	Настраивают инфраструктуру и CI/CD	<ul style="list-style-type: none"><li>• Развертывание Docker;</li><li>• Настройка Celery для парсинга;</li><li>• Мониторинг (Grafana)</li></ul>
Тестировщики	Проводят ручное тестирование	<ul style="list-style-type: none"><li>• Проверка сценариев использования;</li><li>• Багрепорты;</li><li>• Тест-кейсы для новых фич</li></ul>
Бизнес-аналитики (BA)	Собирают требования, проектируют UX	<ul style="list-style-type: none"><li>• Написание User Stories;</li><li>• Прототипирование;</li><li>• Приоритизация фич</li></ul>

# 5. Ресурсы проекта

## Человеческие ресурсы:

- Команда из 22 студентов;
- Время: 3 месяца активной разработки;
- Еженедельные затраты: ~16 часов на человека.

## Технические ресурсы:

- Сервер: VPS для хостинга бота и БД;
- Стек: Python 3.10+, PostgreSQL;
- Инструменты: Git (Github), Битрикс 24, Telegram API.

## Финансовые ресурсы:

- Хостинг: бесплатный тариф;
- Домен и SSL: бесплатный тариф;
- Инструменты мониторинга: бесплатные тарифы.

## 6. План управления проектом

- Методология: Scrumban;
- Спринты: 2-недельные итерации;
- Доска задач (Битрикс 24): To Do → In Progress → Done;
- Митинги: еженедельные стендапы (2 часа), ретроспективы после спринтов.

### Этапы разработки:

#### Этап 0: Подготовка (2 недели)

- Уточнение требований с заказчиком;
- Формализация и оформление конечных требований к продукту;
- Формирование беклога;
- Настройка окружения разработки.

#### Этап 1: Разработка MVP (6 недель) и дополнительных функций (2 недели)

- Спринт 1: Ядро бота + парсинг сайтов;
- Спринт 2: Подписки и настройки пользователей;
- Спринт 3: Административная панель;
- Спринт 4: Дополнительные функции.

#### Этап 2: Стабилизация (2 недели)

- Исправление багов;
- Настройка мониторинга;
- Подготовка к релизу.

#### Этап 3: Поддержка (постоянно)

- Добавление новых источников;
- Масштабирование при росте нагрузки.

## 7. РИСКИ И ПУТИ ИХ МИНИМИЗАЦИИ

Риск	Вероятность	Влияние	Способы минимизации
Блокировка парсинга сайтами	Высокая	Критическое	<ul style="list-style-type: none"><li>• Ротация прокси;</li><li>• Использование API, где возможно;</li><li>• Резервные источники</li></ul>
Превышение rate limits API	Средняя	Высокое	<ul style="list-style-type: none"><li>• Кеширование данных;</li><li>• Оптимизация запросов;</li></ul>
Недостаток экспертизы в команде	Средняя	Среднее	<ul style="list-style-type: none"><li>• Менторство опытных разработчиков;</li><li>• Обучение в процессе;</li><li>• Использование готовых решений</li></ul>
Изменение требований заказчика	Низкая	Среднее	<ul style="list-style-type: none"><li>• Четкое ТЗ;</li><li>• Регулярные аудиты;</li><li>• Agile-подход</li></ul>
Технические сбои инфраструктуры	Низкая	Высокое	<ul style="list-style-type: none"><li>• Мониторинг 24/7;</li><li>• Автоматические бекапы;</li><li>• Fallback-сценарии</li></ul>
Срыв сроков разработки	Средняя	Высокое	<ul style="list-style-type: none"><li>• Буферное время в спринтах;</li><li>• Регулярные стендапы;</li><li>• Раннее выявление блокеров</li></ul>
Потеря ключевых участников команды	Низкая	Высокое	<ul style="list-style-type: none"><li>• Документирование знаний;</li><li>• Парное программирование;</li><li>• Кросс-функциональные навыки</li></ul>
Правовые ограничения на парсинг	Низкая	Критическое	<ul style="list-style-type: none"><li>• Изучение Terms of Service;</li><li>• Использование официальных API;</li><li>• Юридическая консультация</li></ul>
Низкая пользовательская активность	Средняя	Среднее	<ul style="list-style-type: none"><li>• UX-тестирование;</li><li>• Обратная связь от пользователей;</li><li>Маркетинговые активности</li></ul>

# 8. Критерии успеха

## Технические критерии:

- Бот парсит минимум 5 источников.
- Команды /jobs, /internships, /subscribe работают корректно.
- Рассылка уведомлений функционирует стабильно.
- Время отклика < 1 секунды.

## Бизнес-критерии:

- 500+ активных пользователей за первый месяц.
- 60%+ положительных отзывов пользователей.
- 100+ новых вакансий добавляется ежедневно.
- Uptime 99.9% в течение месяца.
- Готовность к монетизации (техническая основа для премиум-функций).

## 9. Документы и приложения

## Основная документация:

- Техническое задание
  - Критерии приемки

# ЗАШИБИСЬ, УТВЕРЖДАЮ!

Утверждаю: