**Вячеслав Дворак**

**Python-разработчик / AI-инженер**

[Ссылка на GitHub](https://github.com/dvvmoz/legal-bot-backup) [Юридический консультант РБ](https://t.me/legal_rb_rag_bot)  Город: Минск, Беларусь

**Краткая сводка**

Высококвалифицированный Python-разработчик с подтвержденным опытом проектирования, разработки и развертывания сложных интеллектуальных систем. Специализируюсь на создании AI-ассистентов с использованием архитектуры RAG, классического машинного обучения и современных бэкенд-технологий. Обладаю глубокими знаниями в области контейнеризации (Docker), создания веб-панелей (Flask) и автоматизации тестирования (pytest).

**Ключевые навыки**

* **Языки программирования:** Python (продвинутый уровень)
* **AI / Машинное обучение:**
  + **NLP и RAG:** Проектирование и реализация RAG-пайплайнов, семантический поиск, промпт-инжиниринг.
  + **Библиотеки:** Scikit-learn, Pandas, NumPy.
  + **Векторные БД:** ChromaDB.
  + **LLM API:** OpenAI.
* **Бэкенд и API:**
  + **Фреймворки:** Flask, Aiogram (для Telegram-ботов).
  + **Веб-сокеты:** Flask-SocketIO для real-time приложений.
* **Инструменты и DevOps:**
  + **Контейнеризация:** Docker, Docker Compose.
  + **Тестирование:** Pytest.
  + **Системы контроля версий:** Git, GitHub.
* **Обработка данных:** Веб-скрапинг (BeautifulSoup, Requests), обработка файлов (PDF, DOCX).

**Опыт работы**

**Python-разработчик / AI-инженер** | Проект: Юридический AI-ассистент «ЮрПомощник РБ»

Разработал «с нуля» и подготовил к развертыванию многокомпонентного Telegram-бота, предоставляющего юридические консультации на основе законодательства РБ.

**Ключевые достижения и обязанности:**

* **Разработка архитектуры:** Спроектировал и реализовал микросервисную архитектуру с использованием Docker Compose, включающую сервис бота и сервис административной панели.
* **AI-ядро (RAG):**
  + Реализовал полный RAG-цикл: поиск релевантных документов в векторной базе данных **ChromaDB** и подача их в качестве контекста в языковую модель **OpenAI (GPT)**.
  + Разработал сложный **гибридный фильтр** для классификации входящих вопросов, комбинируя эвристические методы и модель на базе **Scikit-learn**.
  + Внедрил механизм **самообучения** бота через динамический поиск на внешних ресурсах при отсутствии ответа в локальной базе.
* **Бэкенд и админ-панель:**
  + Создал асинхронного Telegram-бота с помощью **Aiogram**, реализовав обработку команд и сложную логику диалогов.
  + Разработал защищенную веб-панель администратора на **Flask** с аутентификацией, мониторингом системных ресурсов и возможностью удаленного запуска скриптов.
  + Интегрировал **WebSockets** для отслеживания статуса фоновых задач в реальном времени.
* **Система управления данными:**
  + Написал интеллектуальный веб-скрапер для сбора данных, а также **инкрементальный скрапер** для отслеживания и обработки только измененных страниц.
  + Реализовал модули для извлечения и семантической обработки текста из файлов .pdf и .docx.
* **Качество и надежность:**
  + Настроил CI/CD пайплайн с использованием **Dockerfile** и **Docker Compose** для легкого развертывания.
  + Написал набор автоматизированных тестов с помощью **Pytest** для обеспечения стабильности и надежности приложения.

**Образование**

Университет «Зерокодер»