

Описательная документация информационной системы по контролю за проектами нормативных правовых актов LEGISFLOW.TECH (MVP)

1. Введение

LEGISFLOW.TECH – автоматизированная система мониторинга и анализа законопроектов, предназначенная для упрощения работы юристов и повышения их эффективности. Система собирает данные с государственных порталов, анализирует их и предоставляет пользователям персонализированные уведомления о важных изменениях в законодательстве.

2. Обоснование выбора технологий

2.1. Язык программирования: Python

Обоснование выбора:

- Широкая экосистема библиотек для парсинга и обработки текстов
- Простота интеграции с различными API
- Поддержка асинхронного программирования
- Большое сообщество разработчиков и обширная документация
- Кроссплатформенность

2.2. Основные библиотеки и фреймворки

Для парсинга данных:

- Selenium: Автоматизация взаимодействия с динамическими веб-страницами

Обоснование:

Данная библиотека обеспечивает надежный сбор информации даже с защищенных или динамически формируемых страниц государственных порталов.

Для работы с документами:

- PyMuPDF: Извлечение текста из PDF-документов
- striprtft: Обработка документов формата rft

Обоснование:

Государственные порталы публикуют законопроекты преимущественно в этих форматах, что делает данные библиотеки необходимыми.

Для работы с базой данных:

- sqlite3 – для работы с СУБД

Обоснование:

Sqlite3 был выбран как надежная реляционная СУБД, что критично для работы с законодательными документами.

Для реализации Telegram-бота:

- Aiogram: Асинхронный фреймворк для Telegram API

Обоснование:

Aiogram предоставляет удобный интерфейс для работы с Telegram Bot API и поддерживает асинхронную модель программирования.

2.3. База данных: SQLite

Обоснование выбора:

- Надежность и отказоустойчивость
- Возможность масштабирования
- Простота использования
- Бесплатная лицензия

2.4. Хостинг: RuVDS

Обоснование выбора:

- Простота развертывания Python-приложений
- Гибкость в настройке расписания задач
- Доступная стоимость
- Надежная инфраструктура

3. Архитектура системы

3.1. Компоненты системы

1. Модуль сбора данных:

- Реализован на Python с использованием selenium
- Запускается по расписанию (каждые 12 часов)
- Сохраняет сырые данные в базу

2. Модуль обработки:

- Анализирует и структурирует полученные данные
- Применяет фильтры по ключевым словам
- Генерирует краткие сводки

3. Модуль уведомлений:

- Отправляет персонализированные сообщения через Telegram
- Формирует ежедневные/еженедельные отчеты

4. База данных:

- Хранит все собранные законопроекты
- Сохраняет пользовательские настройки и фильтры

3.2. Взаимодействие компонентов

[Гос. порталы] → [Парсер] → [База данных] → [Анализатор] → [Telegram-бот] → [Пользователь]

4. Особенности реализации

4.1. Обработка различных форматов документов

Система поддерживает:

- PDF-документы (обработка через Rumpypdf)
- RTF-файлы (чтение через striprtft)

4.2. Механизм фильтрации

Реализована многоуровневая фильтрация:

1. По ключевым словам (точное соответствие)
2. По категориям законодательства
3. По дате публикации
4. По статусу законопроекта

4.3. Система уведомлений

Особенности:

- Персонализированные настройки для каждого пользователя
- Гибкое расписание отправки
- Разные форматы уведомлений (краткие/подробные)

5. Заключение

Выбранный стек технологий обеспечивает:

- Надежность работы системы
- Простоту дальнейшего развития
- Хорошую производительность
- Удобство сопровождения и модификации

Все компоненты системы имеют открытые лицензии, что делает решение экономически эффективным и не создает юридических рисков.