**Руководство программиста**

**«**Информационной системы телеграмм-бота по поиску аналогов игр»

1. **Инсталляция программы:**
   1. **Пользовательский вариант**
      1. Скачать файлы для работоспособности программы, которые находится по ссылке <https://github.com/Deserag/Telegram-bot_for_analogov>.
      2. Для установки необходимо запустить файл main. В нем требуется указать токен telegram-bot для его работы.
      3. При отсутствии токена, создать его с помощью телеграмм-бота BotFather: <https://t.me/BotFather>.
      4. Проверить бота.
   2. **Вариант поддержания и модернизации**
2. Убедитесь, что на вашем компьютере установлен Python, с библиотекой Telebot.
3. Убедитесь что на вашем смартфоне или персональном компьютере установлен telegram или вы можете войти в него в web версии.
4. Склонируйте репозиторий программы с GitHub с помощью команды git clone https://github.com/Deserag/Telegram-bot\_for\_analogov
5. Перейдите в папку вашего проекта с помощью команды cd repository.
6. Установите зависимости проекта, выполнив команду yarn install.
7. Запустите программу для проверки успешности установки командой yarn start.
8. Запустите телеграмм и перейдите в общение к боту по ссылке <https://t.me/poisk_analogov_game_bot>.
9. **Проверка работоспособности чат-бота**

Для проверки работоспособности требуется запустить общение с чат-ботом с помощью команд (“старт”, “start”), также пользователю требуется ввести собственный никнейм для чат-бота, а также названия игр или ее жанров.

1. **Поддержка программы**

Регулярно проверяйте обновления библиотеки telebot, Python, Express.js и правил платформы Телеграмм по работе ботов.

Отслеживайте сообщения об ошибках, поступающие от пользователей, и предоставляйте им поддержку, анализируя и исправляя проблемы.

При изменении функционала или добавлении новых функций, уведомлять пользователей с помощью сообщений в чат-боте.

Ведите документацию по программе, включая инструкции по устранению проблем и ответы на часто задаваемые вопросы, используя Markdown или другой удобный формат, и храните ее в репозитории на GitHub.

Мониторьте производительность программы и проводите оптимизацию кода, включая Python, и базы данных при необходимости.

Регулярно резервируйте данные программы, чтобы предотвратить потерю информации в случае сбоев или сбоев системы, используя соответствующие инструменты и методы, и храните их в безопасном месте, отдельно от репозитория.

1. **Модернизация**

Анализируйте обратную связь пользователей и их потребности, чтобы определить возможности для улучшения программы.

Создайте новую ветку в репозитории для разработки новых функций или улучшений с помощью команды git checkout -b feature-branch.

Разрабатывайте и реализуйте новые функции или улучшения, основываясь на исследованиях и требованиях пользователей, используя Python для разработки логики программы.

Тестируйте новые функции перед их выпуском, чтобы убедиться в их правильной работе и отсутствии ошибок, используя соответствующие инструменты для тестирования Python кода.

Обновляйте документацию программы, чтобы отразить новые функции и изменения, и убедитесь, что она соответствует актуальной версии программы, и отправьте изменения в репозиторий с помощью команды git push origin feature-branch.

Проводите регулярное обслуживание программы, включая оптимизацию кода, в том числе Python и telebot, и обновление зависимостей, используя команду yarn upgrade.