**Руководство программиста**

**«**Информационной системы по сбору статистики достижений STEAM»

1. **Инсталляция программы:**
   1. **Пользовательский вариант**
      1. Скачать инсталятор, который находится по ссылке <https://github.com/Errormacr/steam_achievement_norm/releases/tag/steam> . Файлы необходимые для скачивания отображены на рисунке 1.
      2. Для установки необходимо запустить my-app-1.0.0.Setup.exe и ожидать окончания установки,
      3. Проверьте программу.
   2. **Вариант поддержания и модернизации**
2. Убедитесь, что на вашем компьютере установлен Node.js и yarn (вместо npm).
3. Склонируйте репозиторий программы с GitHub с помощью команды git clone https://github.com/Errormacr/steam\_achievement\_norm
4. Перейдите в папку вашего проекта с помощью команды cd repository.
5. Установите зависимости проекта, выполнив команду yarn install.
6. Запустите программу для проверки успешности установки командой yarn start.
7. Проверьте программу
8. **Проверка работоспособности программы**

Для проверки работоспособности требуется ввести API ключ, который можно получить по ссылке https://steamcommunity.com/dev/apikey зайдя в Steam аккаунт или создав новый, ввести steamId этого аккаунта, после чего сравнить выдаваемую статистику с статистикой того же аккаунта на сайте: https://completionist.me/ . Количество игр может различаться из-за не учитывания некоторых игр программой (инструменты модификации игр и тд), но кол-во достижений, общий процент достижений, последнии достижения и игры должны совпадать при условии удачной загрузки данных.

1. **Поддержка программы**

Регулярно проверяйте обновления Electron Forge, TypeScript, Express.js и других зависимостей вашего проекта и обновляйте их при необходимости, используя команду yarn upgrade.

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

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 TypeScript, 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 TypeScript для разработки логики программы.

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

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 TypeScript и SCSS, и обновление зависимостей, используя команду yarn upgrade.