Клавиатурный тренажер - это программа или онлайн-сервис, предназначенный для обучения набору на компьютерной клавиатуре.

Клавиатурный тренажер представляет собой интерактивный текстовый редактор для набора определенного фрагмента, с функцией подсчёта различных метрик печати (скорость печати, количество допущенных опечаток и т.п.).

Данная программа может быть рассчитана на людей любого возраста. Сфера деятельности также неограниченна. Эту программу могут использовать для своих целей, для развития скорости и четкости печати. Может быть полезна пользователям, часто использующим клавиатуру в рабочих целях.

Утилита может использоваться в детских развивающих центрах, в школах с уклоном в информационные технологии и т.п.

Первый клавиатурный тренажер появился в начале XX века, когда механические печатные машинки стали широко использоваться в офисах. Одним из первых методов обучения была система "слепого десятипальцевого набора", разработанная Фрэнком МакГурриным в 1888 году. С развитием компьютеров появились цифровые тренажеры, а с приходом интернета – онлайн-сервисы, делающие обучение доступным для всех.

Интересный факт: Рекорд по скорости набора на клавиатуре принадлежит Барбаре Блэкберн – в 2005 году она достигла скорости 212 слов в минуту на модифицированной клавиатуре Dvorak!"

Обзор аналогов:

Аналог №1 - «Keyboard\_Typing\_Game»

Программное обеспечение «Keyboard\_Typing\_Game» (рис. 1) от разработчика «prerit85jain» выпущено в 2025 на языке разработки Python.

Ссылка: https://github.com/prerit85jain/Keyboard\_Typing\_Game

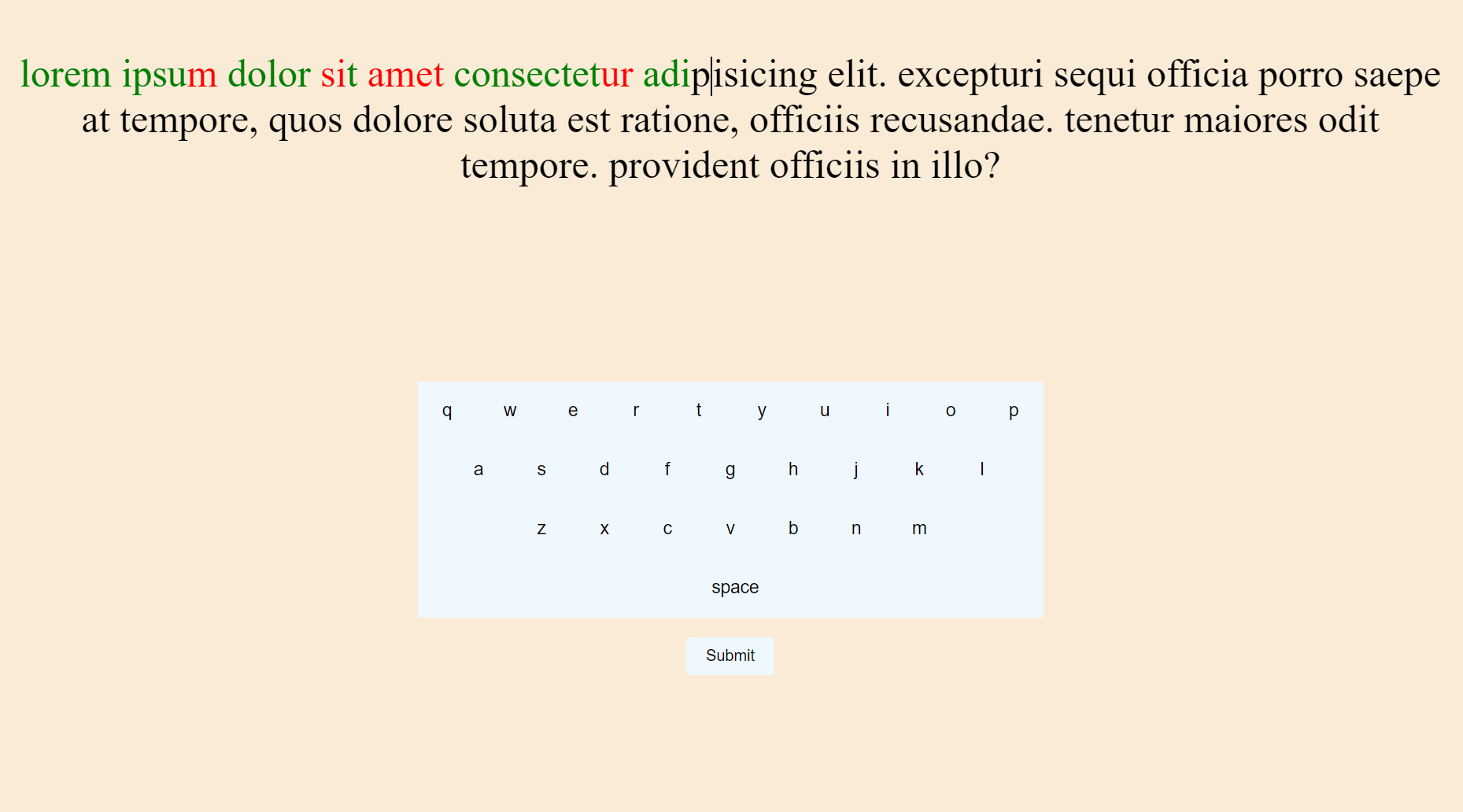


Рисунок 1 – Скриншот интерфейса «Keyboard\_Typing\_Game»

Преимуществом данного программного обеспечения является простой интерфейс.

Недостатки:

* отсутствие режимов набора текста: один фиксированный текст;
* отсутствие каких-либо настроек;
* интуитивно непонятный набор: курсор некорректно отображает положение набора.

Аналог №2 - «Keyboard-pro»

Программное обеспечение «Keyboard-pro» (рис. 2) от разработчика «hardeepnarang10» выпущено в 2021 на языке разработки Python.

Ссылка: https://github.com/hardeepnarang10/keyboard-pro

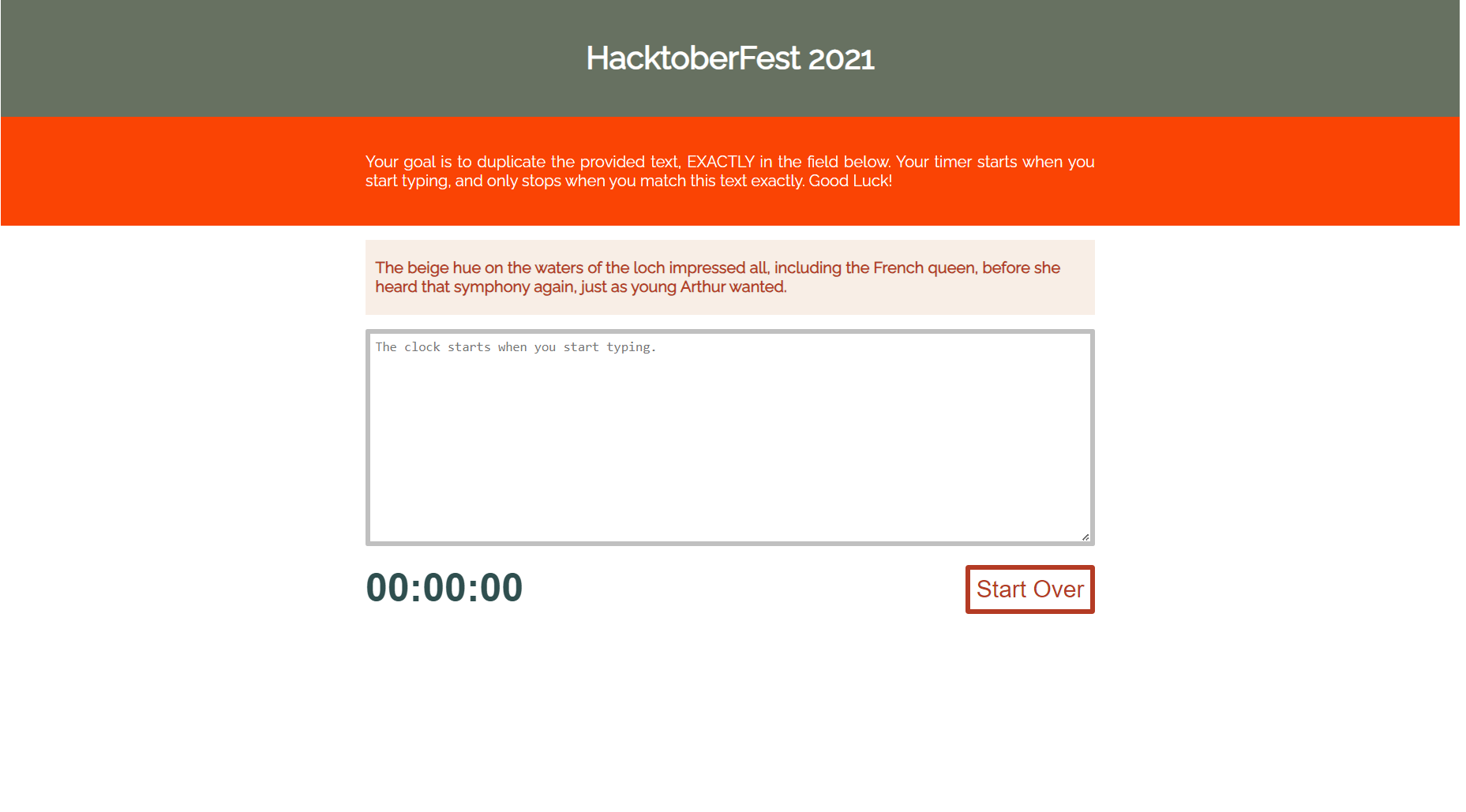


Рисунок 2 – Скриншот интерфейса «keyboard-pro»

Преимущества:

* реализован таймер;
* простой запуск: не требуется предварительной установки приложения на ваш компьютер.

Недостатки:

* программа не выполняет свой функционал: не обрабатывает набранный текст;
* одна вариация набора текста;
* программа не обновляется: отсутствие обновлений 4 года.

4.3 Аналог №3 - «Keyboard»

Программное обеспечение «Keyboard» (рис. 3) от разработчика «Dearian27» выпущено в 2022 на языке разработки Python.

Ссылка: https://github.com/Dearian27/keyboard

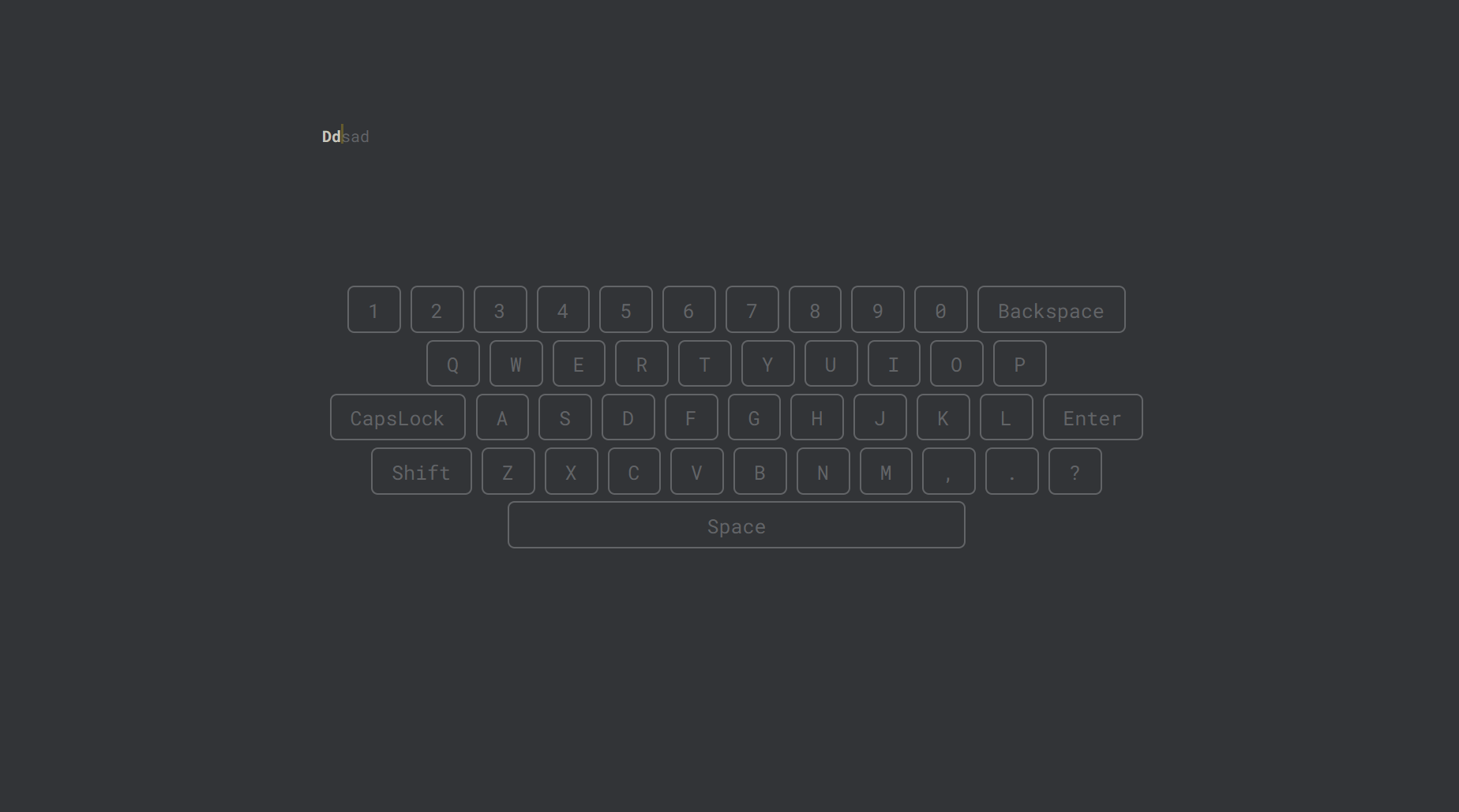


Рисунок 3 – Скриншот интерфейса «keyboard»

Преимуществом данного программного обеспечения является простой интерфейс.

Недостатки:

* устаревшая программа: отсутствие обновлений на протяжении двух лет;
* отсутствие разнообразных упражнений: две вариации набора текста;
* отсутствие каких-либо настроек интерфейса.

Основные проблемы аналогов — неактуальные версии программ, отсутствие разнообразных упражнений, некорректное выполнение и ограниченный функционал. При разработке собственного тренажера важно учитывать несколько ключевых аспектов.

Интерфейс должен быть простым и удобным, без лишних элементов и интуитивно понятным расположением функциональных кнопок. Наличие нескольких языков расширит аудиторию, а бесплатность или разумная модель монетизации снизит барьер входа. Современный дизайн и регулярные обновления обеспечат актуальность продукта. Учитывая эти принципы, можно создать востребованный, эффективный и удобный клавиатурный тренажер.

Описание работы программы:

Программа "Клавиатурный тренажер" предназначена для обучения пользователей навыкам быстрого и точного набора текста на клавиатуре. Она предоставляет интерактивные упражнения, которые включают набор предложений, слов или отдельных символов.

Основные функции программы:

1. Набор текста с различным уровнем сложности (буквы, слова, предложения).
2. Подсчет скорости печати (знаков в минуту) и точности.
3. Анализ и отображение статистики ошибок для их устранения.

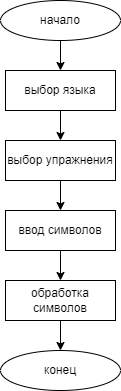


Рисунок 4 – Схема пользовательского пути

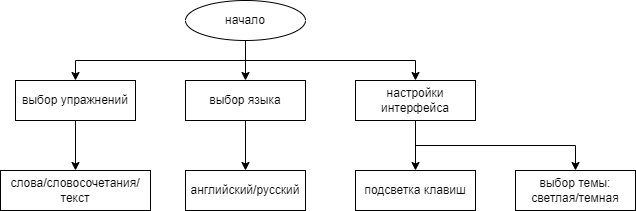


Рисунок 5 – Схема функционала приложения

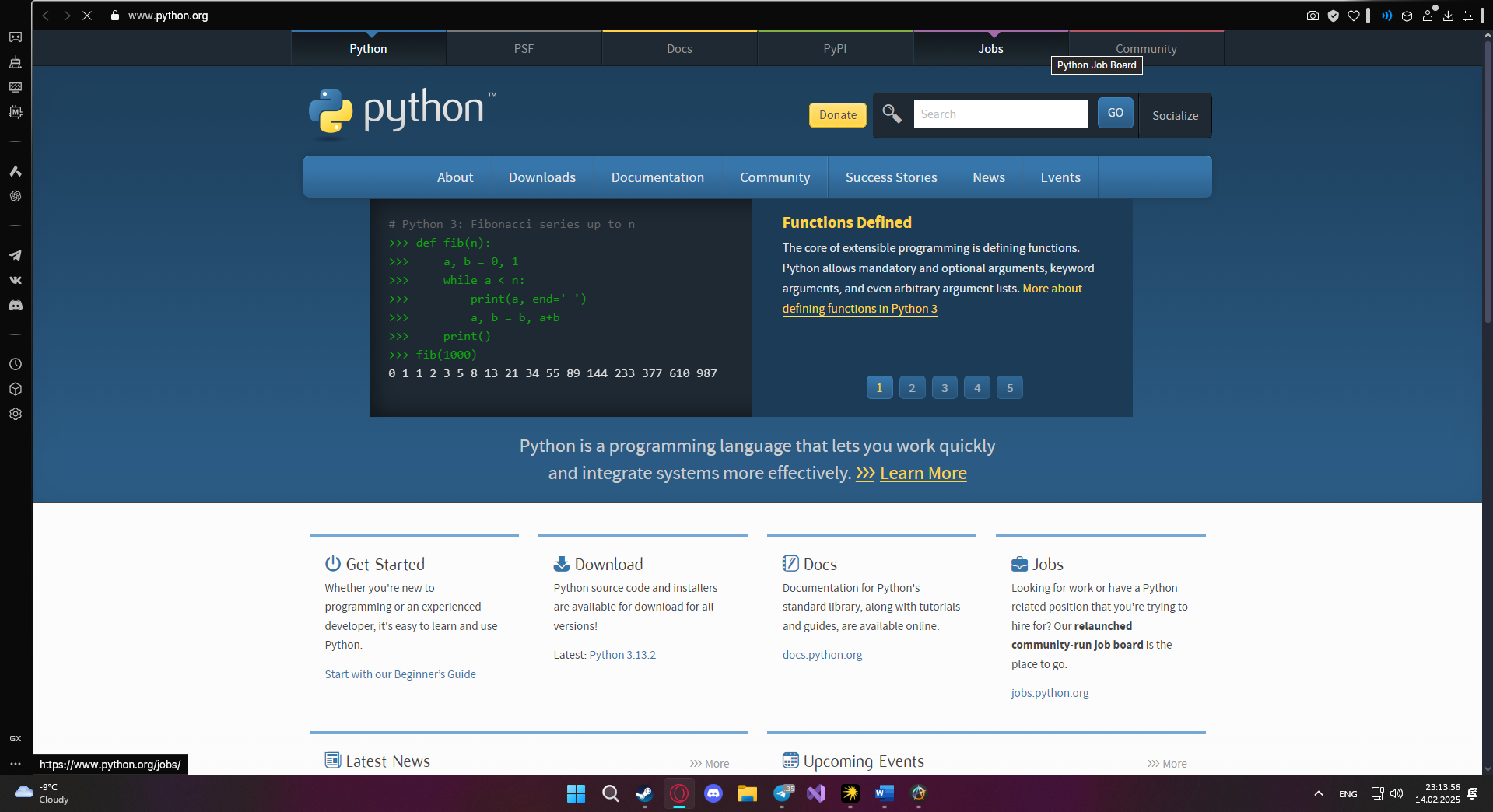
НАСТРОЙКА РАБОЧЕГО ОКРУЖЕНИЯ

Выбор языка программирования и среда разработки

Python — это высокоуровневый, интерпретируемый язык программирования с динамической типизацией. Он популярен благодаря своей простоте, читаемости кода и широкому сообществу разработчиков.

Критерии выбора Python:

* Простота и читаемость — понятный синтаксис, легкий для изучения.
* Кроссплатформенность — работает на Windows, macOS и Linux.
* Широкая стандартная библиотека — включает множество встроенных модулей.
* Популярность и поддержка — большое сообщество и обширная документация.
* Универсальность — подходит для веб-разработки, анализа данных, машинного обучения, автоматизации и других задач.

  
Рисунок 4 – Скриншот официального сайт Python

После нажатия кнопки установочный файл сохранится в папке «Загрузки» на вашем компьютере. Открыв его, вы увидите экран установки Python (рис. 5).

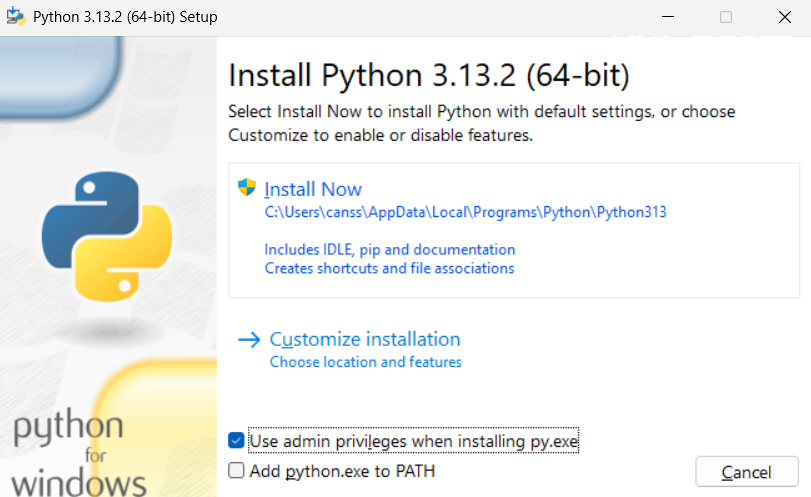


Рисунок 5 – Скриншот окна установки Python

Кнопка «Install Now» позволяет выполнить установку с рекомендуемыми настройками без дополнительных действий, а «Customize installation» предоставляет возможность настроить параметры вручную. В большинстве случаев предпочтителен вариант установки по умолчанию. После выбора параметров начнется процесс установки (рис. 6).

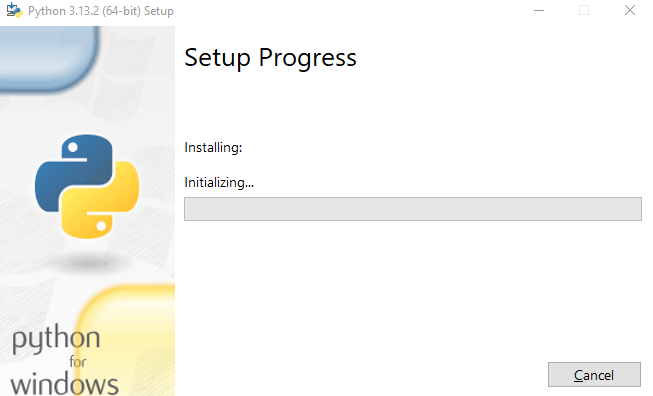


Рисунок 6 – Скриншот окна установки Python

Используемая среда разработки: PyCharm  
PyCharm — это мощная IDE (интегрированная среда разработки) от компании JetBrains, предназначенная для разработки на Python.

Основные преимущества PyCharm:

* Удобный редактор кода с подсветкой синтаксиса и автодополнением.
* Отладчик и средства тестирования для быстрого выявления ошибок.
* Встроенная система контроля версий (Git, SVN).
* Поддержка виртуальных окружений для управления зависимостями.
* Инструменты анализа кода и рефакторинга для улучшения качества кода.

PyCharm позволяет удобно разрабатывать, тестировать и отлаживать Python-приложения, делая процесс разработки более эффективным.

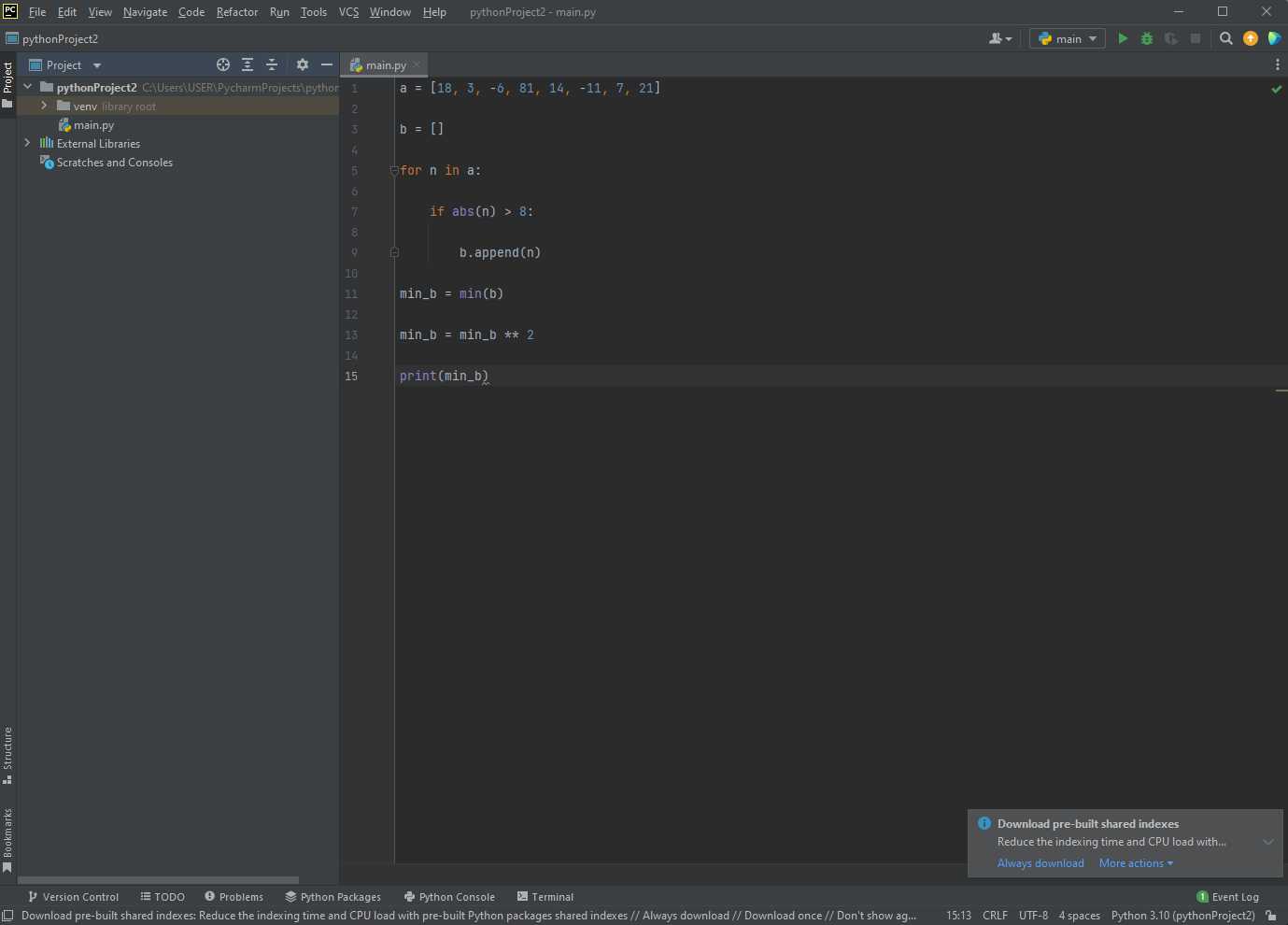


Рисунок 7 – Скриншот интерфейса среды разработки PyCharm

Чтобы установить PyCharm, необходимо зайти на официальный сайт и выбрать Community Edition. Также есть платная версия IDE, ориентированная на более опытных и многопрофильных разработчиков.

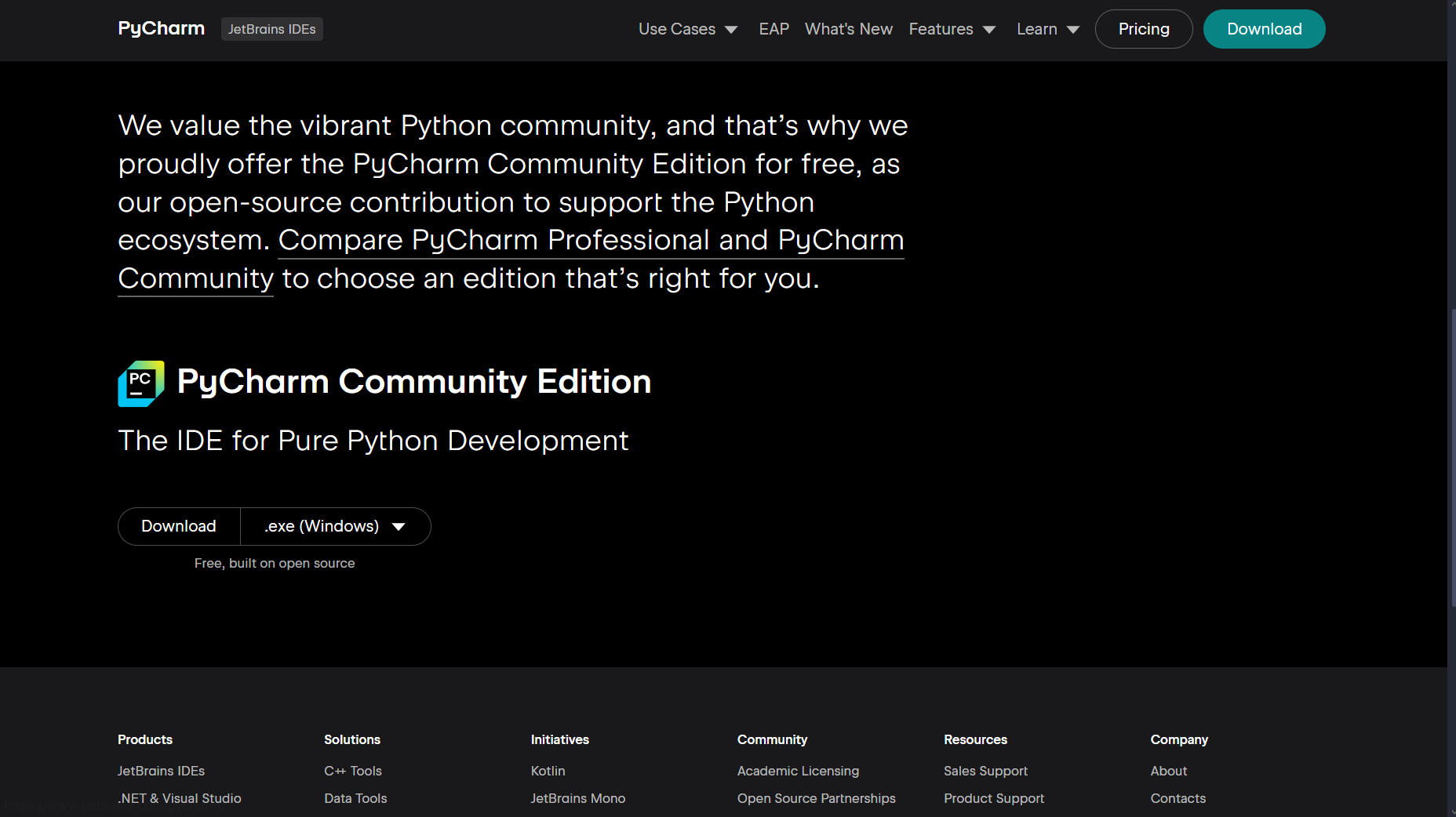


Рисунок 8 – Скриншот официального сайта PyCharm

После нажатия кнопки установочный файл сохранится в папке «Загрузки» на вашем компьютере. Открыв его, вы увидите экран установки PyCharm (рис. 9).

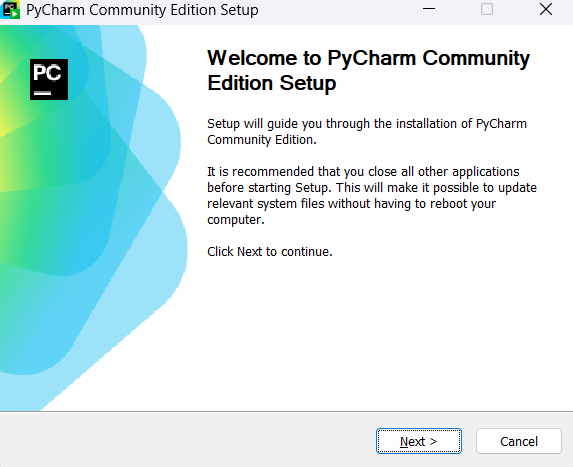


Рисунок 9 – Скриншот установочного окна PyCharm

Далее потребуется выбрать установочный путь для IDE, также можно выбрать дополнительные опции при установке. После предлагается создать или выбрать существующую папку для создания ярлыка программы. (рис 10-12.)

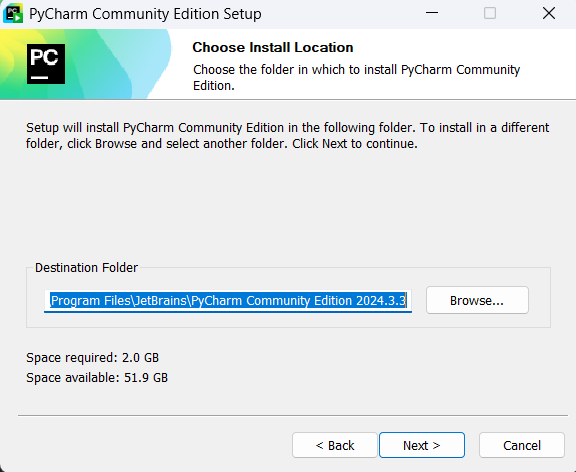


Рисунок 10 – Скриншот установочного окна PyCharm

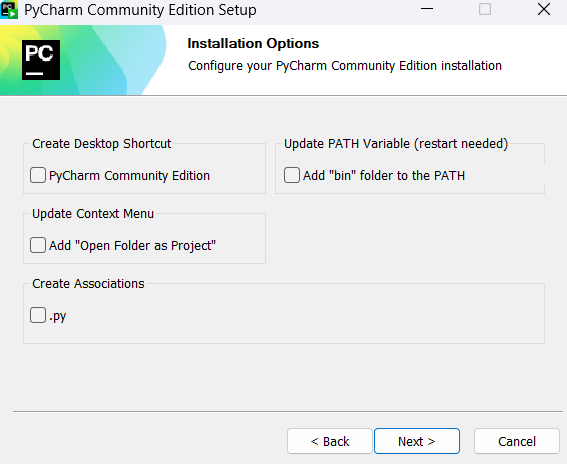


Рисунок 11 – Скриншот установочного окна PyCharm

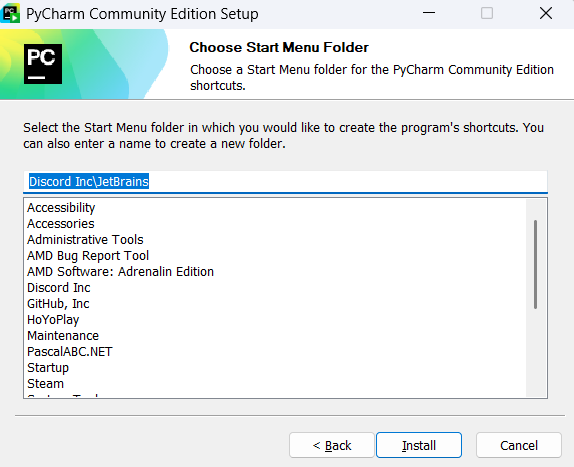


Рисунок 12 – Скриншот установочного окна PyCharm

Затем необходимо создать новый проект в PyCharm. При выборе «New Project» откроется диалоговое окно, где нужно указать название проекта, папку для его хранения и дополнительные параметры виртуального окружения.

Виртуальное окружение (virtual environment) в Python — это инструмент, позволяющий изолировать зависимости проекта от глобальной среды, предотвращая возможные конфликты между библиотеками. Для его настройки потребуется выбрать версию интерпретатора, установленную ранее.

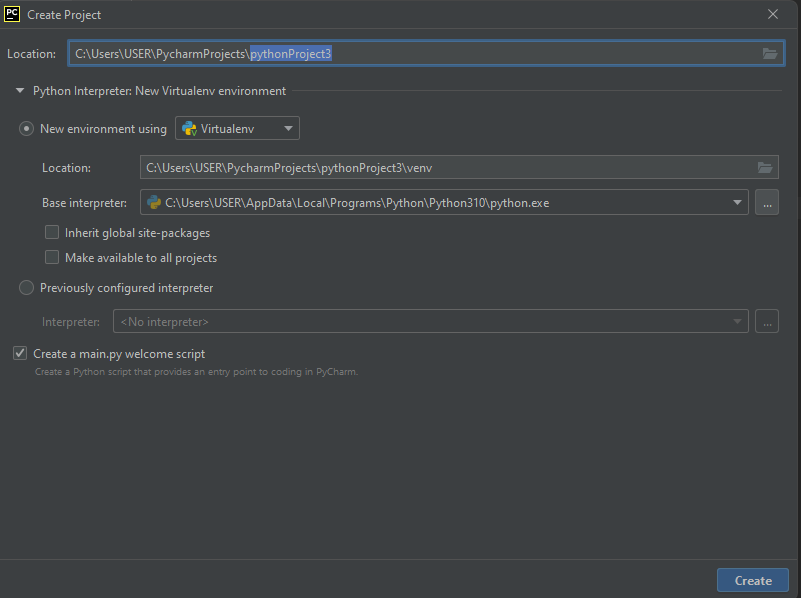


Рисунок 13 – Скриншот окна создания нового Python проекта

Для удобства разработки, управления версиями, а также для возможности работать над проектом с любого устройства — будь то домашний компьютер или ноутбук в другом месте — необходимо установить систему контроля версий Git.

Git представляет собой консольную утилиту, предназначенную для отслеживания и ведения истории изменений файлов в проекте. Чаще всего она используется для работы с кодом, но может применяться и для других типов файлов. С помощью Git можно откатывать проект к более ранним версиям, сравнивать изменения, анализировать их и загружать обновления в репозиторий.

Репозиторий — это хранилище кода и всей истории его изменений. Git работает локально, сохраняя репозитории в определенных папках на жестком диске. Однако можно хранить их и в интернете, используя, например, сервис GitHub.

Каждая сохраненная версия проекта называется коммитом (commit). У каждого коммита есть уникальный идентификатор (hash) и комментарий. Коммиты формируют ветку, которая представляет собой историю изменений. Ветки имеют собственные названия, а один репозиторий может содержать несколько веток, которые создаются на основе других или объединяются с ними.

Работа с Git и его командами осуществляется через специальный терминал Git Bash или встроенный терминал в среде разработки.

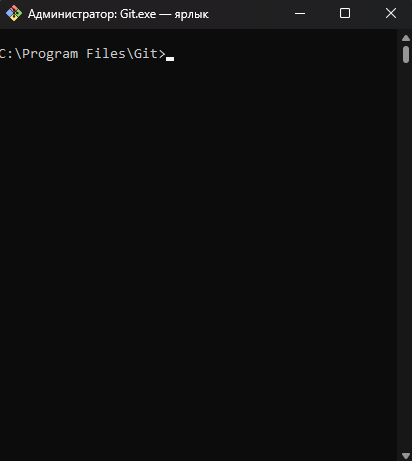


Рисунок 14 – Скриншот интерфейса GitBash

Теперь можно переходить к установке системы контроля версий. Для этого необходимо скачать Git с официального сайта: https://git-scm.com/downloads/win. На странице загрузки следует выбрать операционную систему, установленную на вашем компьютере, и нажать соответствующую кнопку. После этого установочный файл автоматически загрузится в папку «Загрузки».

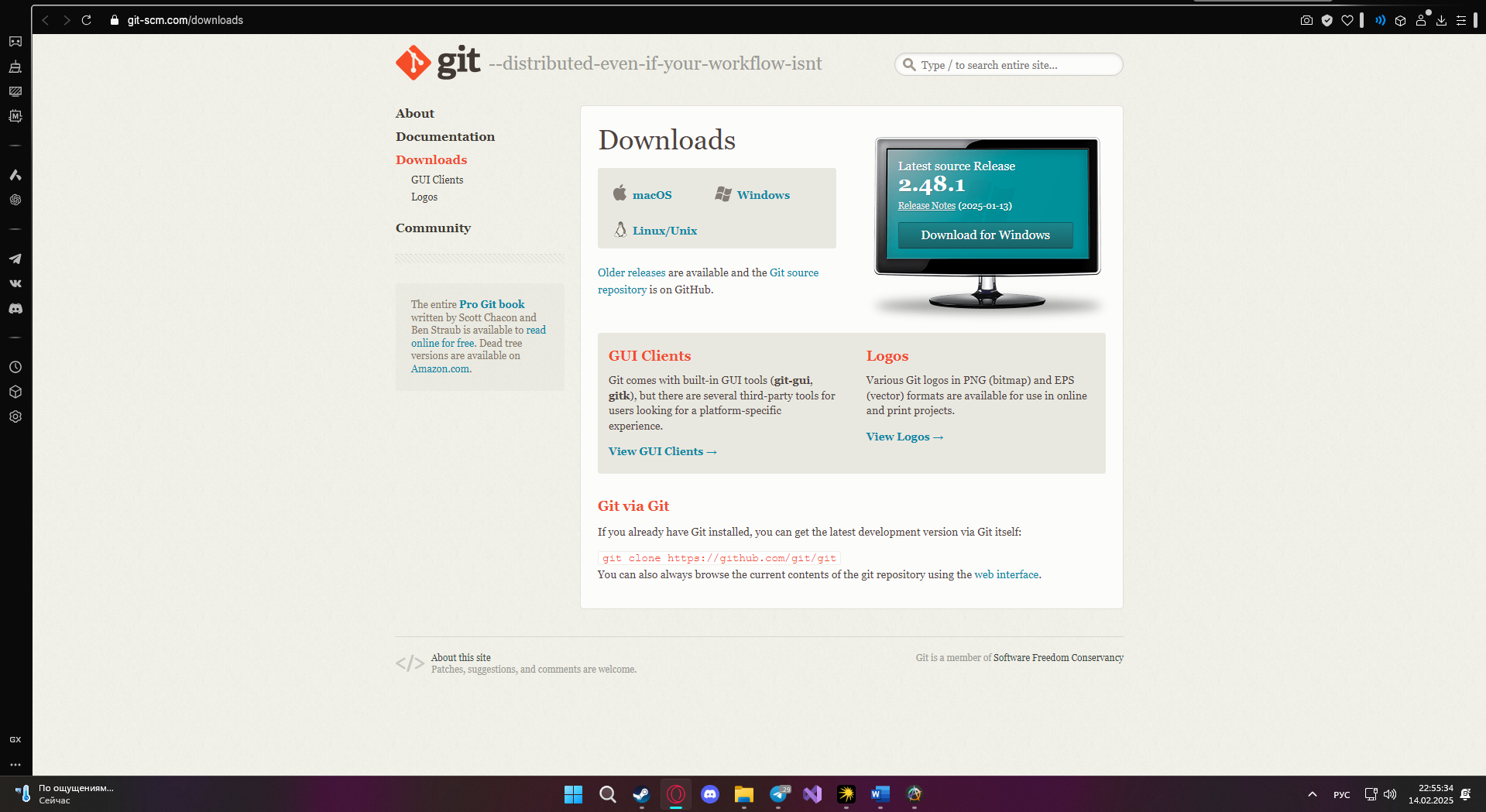
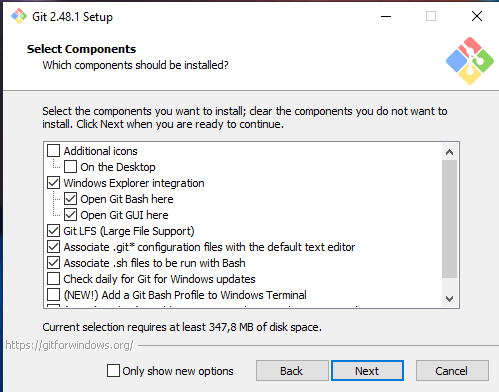
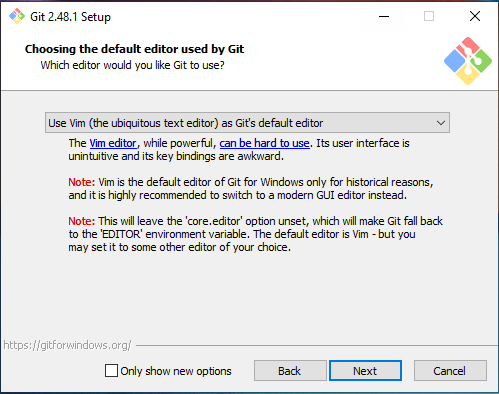


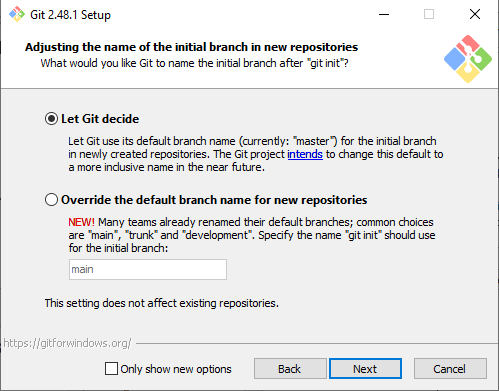
Рисунок 15 – Скриншот официального сайта Git

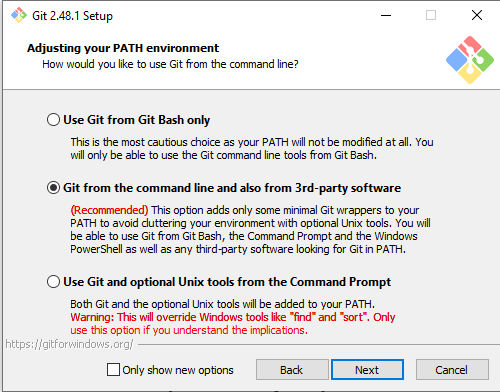
После запуска программы установщика Git (рис. 14) можно выбрать выбор всех параметров установки по умолчанию (рис. ).

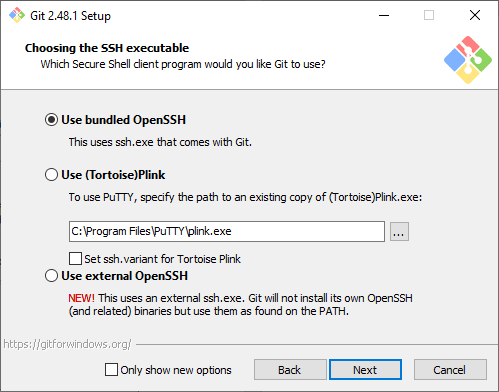
  
Рисунок 16 – Скриншот окна установки Git

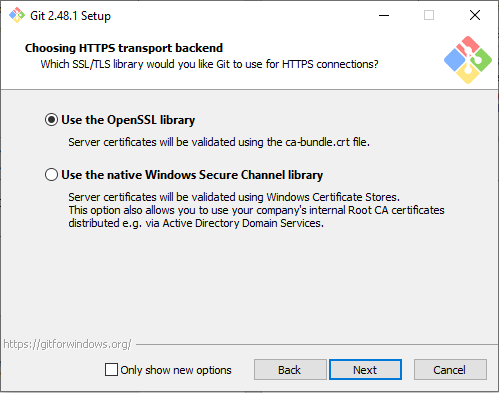
  
Рисунок 17 – Скриншот окна настройки установки Git

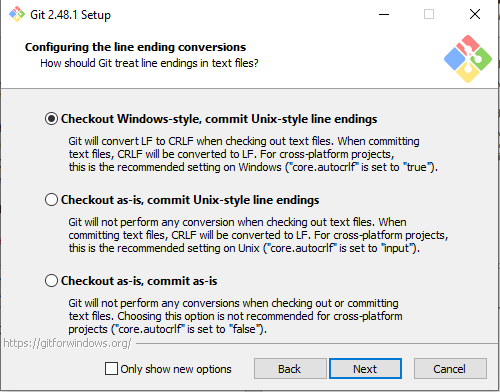
  
Рисунок 18 – Скриншот окна настройки установки Git

  
Рисунок 19 – Скриншот окна настройки установки Git

  
Рисунок 20 – Скриншот окна настройки установки Git

  
Рисунок 21 – Скриншот окна настройки установки Git

  
Рисунок 22 – Скриншот окна настройки установки Git

  
Рисунок 23 – Скриншот окна настройки установки Git

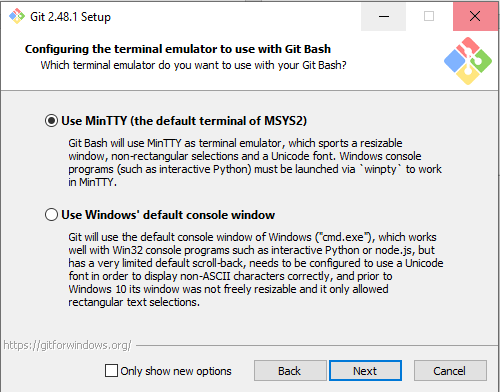
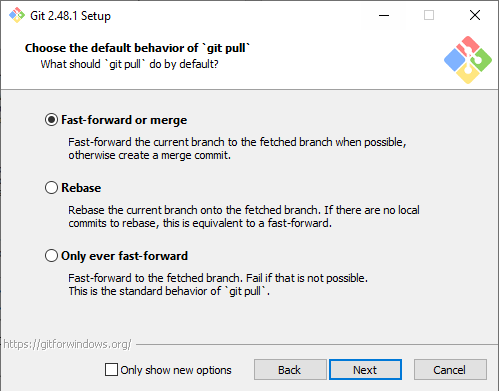


Рисунок 24 – Скриншот окна настройки установки Git

  
Рисунок 25 – Скриншот окна настройки установки Git

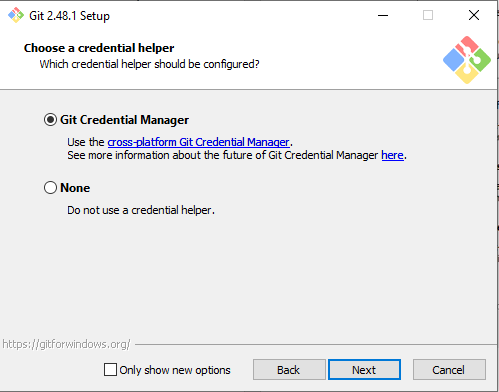
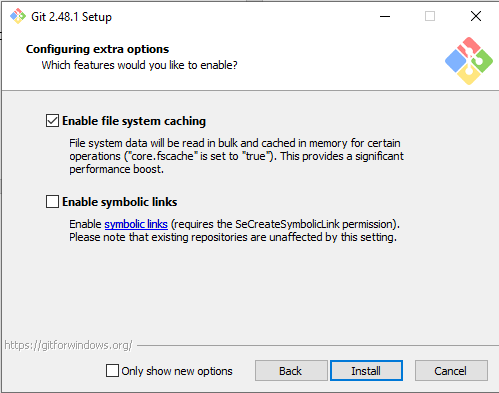


Рисунок 26 – Скриншот окна настройки установки Git

  
Рисунок 27 – Скриншот окна настройки установки Git

Как уже упоминалось ранее, в ходе работы над проектом будет использоваться онлайн-сервис GitHub — облачная платформа, которая упрощает работу с Git, предоставляет удобный веб-интерфейс, инструменты для командной работы, систему управления задачами и многое другое. Перед началом работы с GitHub необходимо зарегистрироваться на официальном сайте: https://github.com/signup (рис. 22)

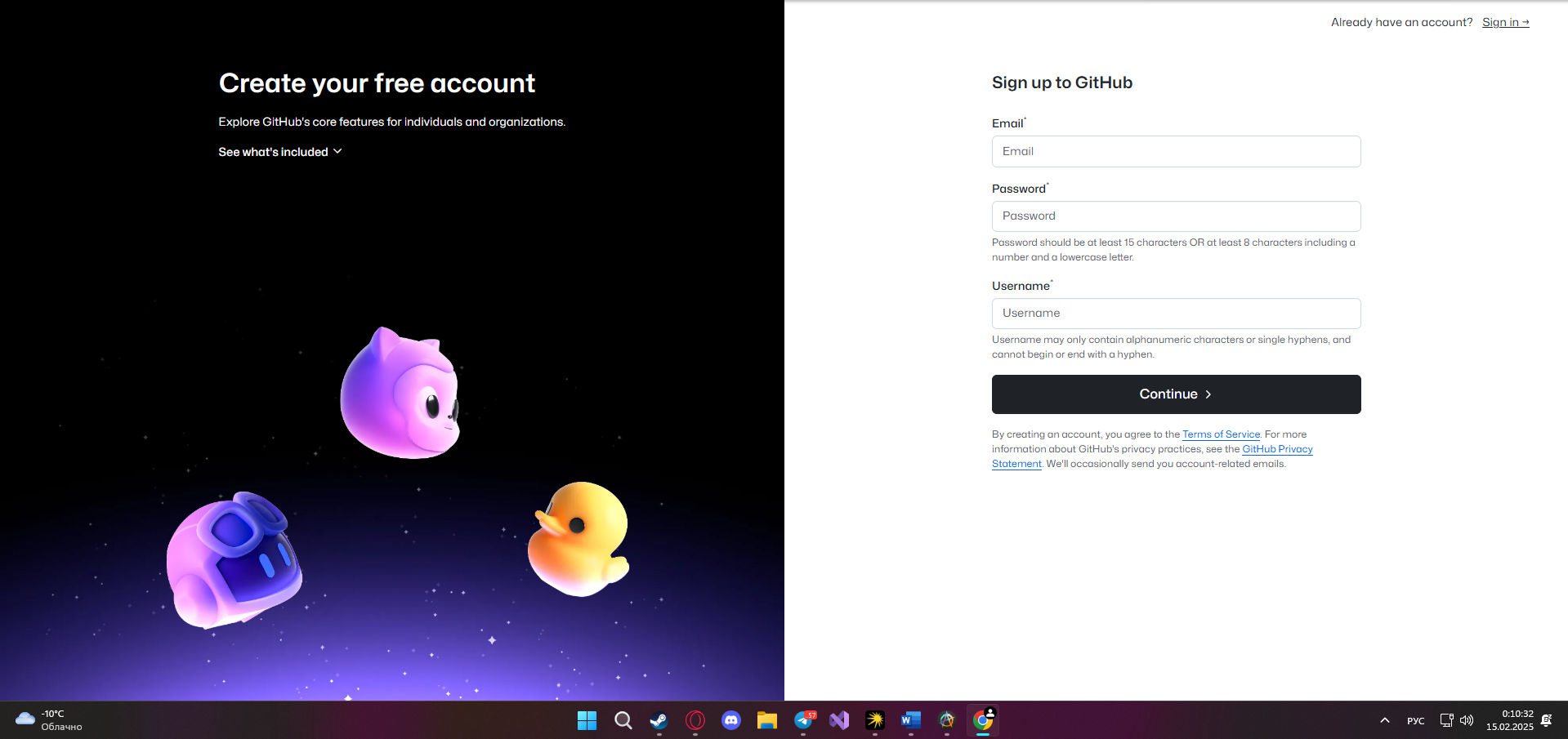


Рисунок 28 – Скриншот официального сайта GitHub

После регистрации пользователь попадает на страницу своего профиля (рис. 29), где отображаются имя, никнейм, аватар, список достижений, закрепленные репозитории и график активности.

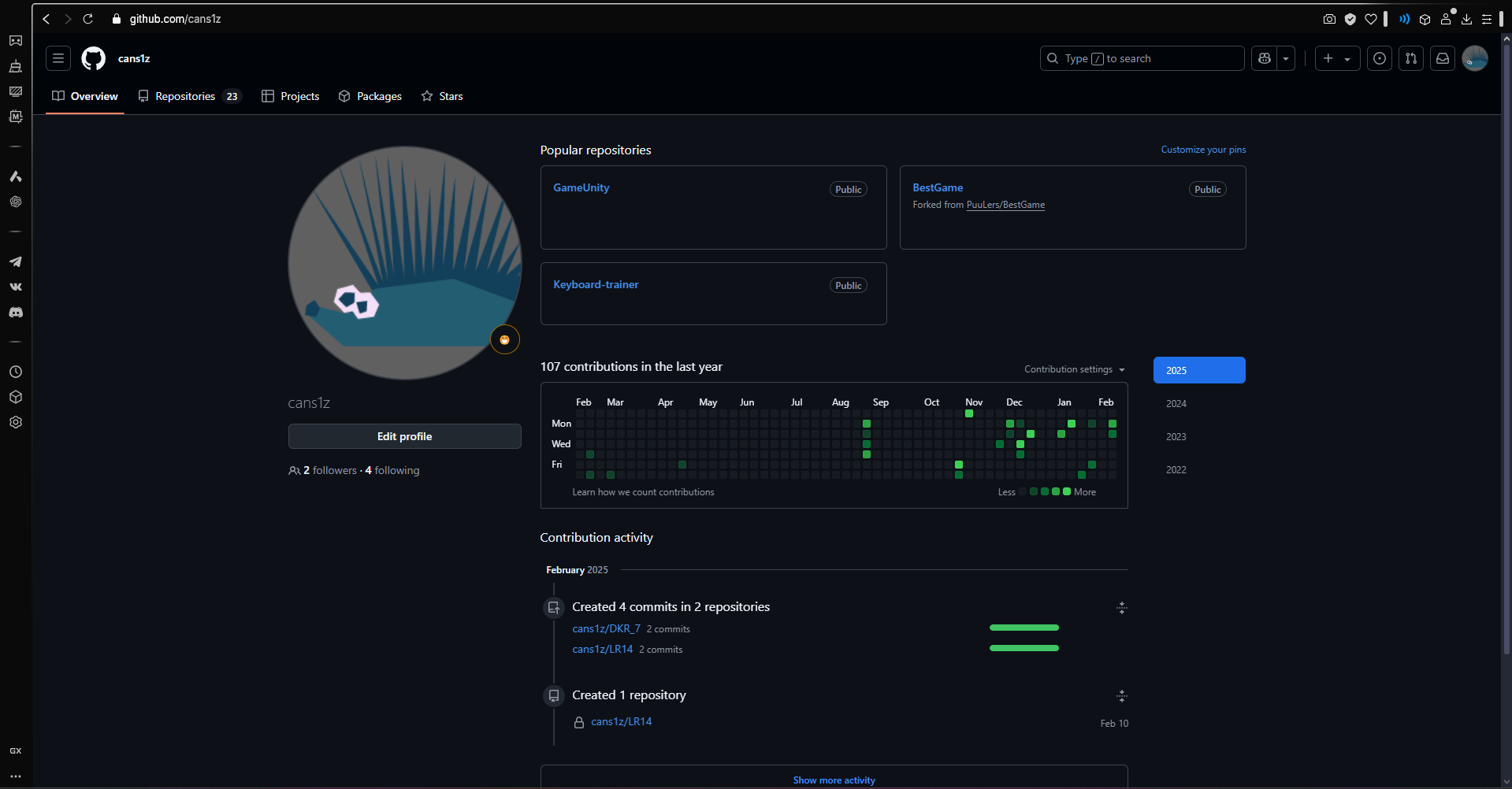


Рисунок 29 – Скриншот страницы пользователя

Для создания нового репозитория необходимо перейти во вкладку с репозиториями «Repositories» (рис. 30).

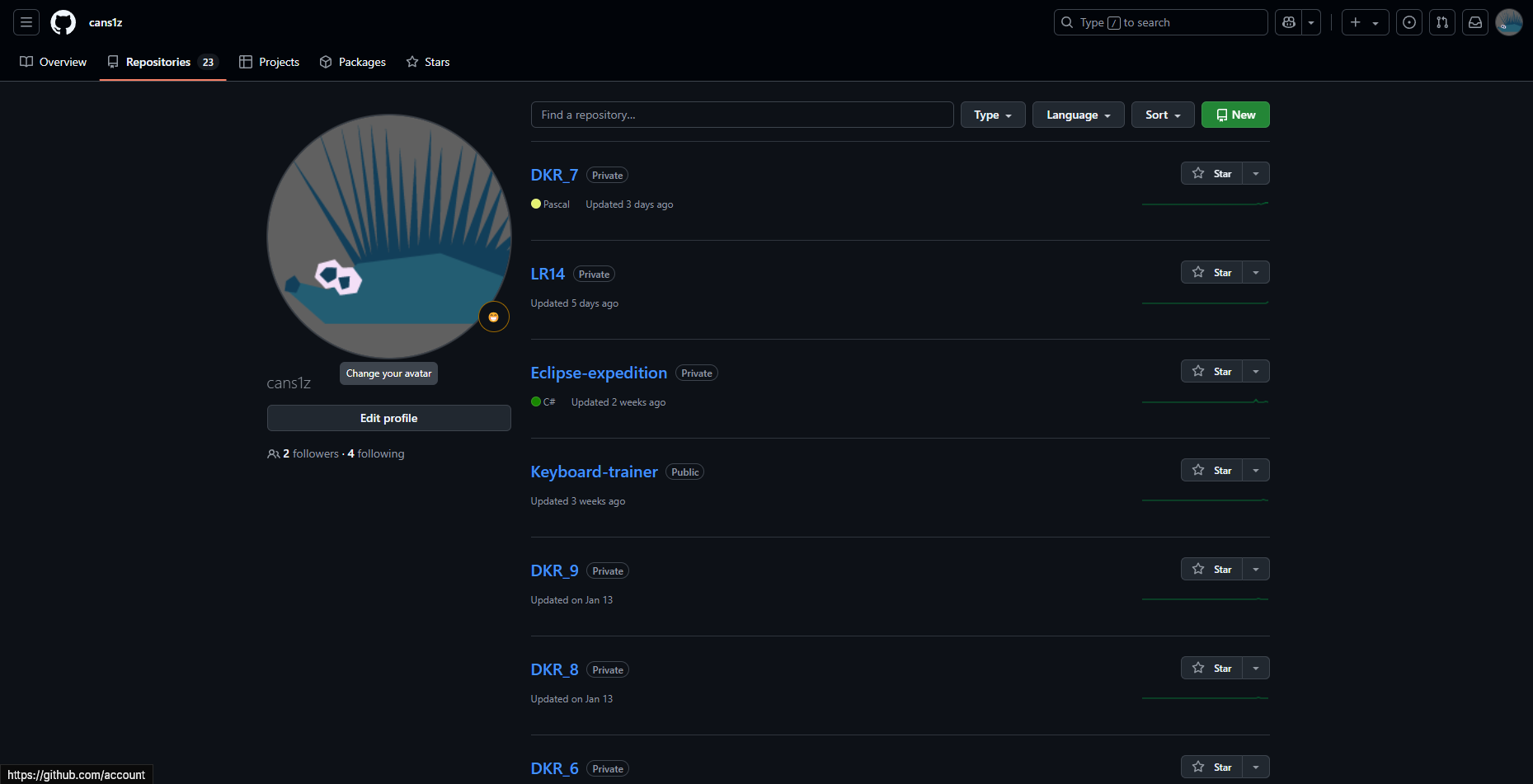


Рисунок 30 – Скриншот репозиториев пользователя

Далее необходимо нажать кнопку «New» для создания нового репозитория. После этого откроется вкладка (рис. 31 ), где следует указать владельца репозитория, его название, а также уровень доступа (приватный или открытый).

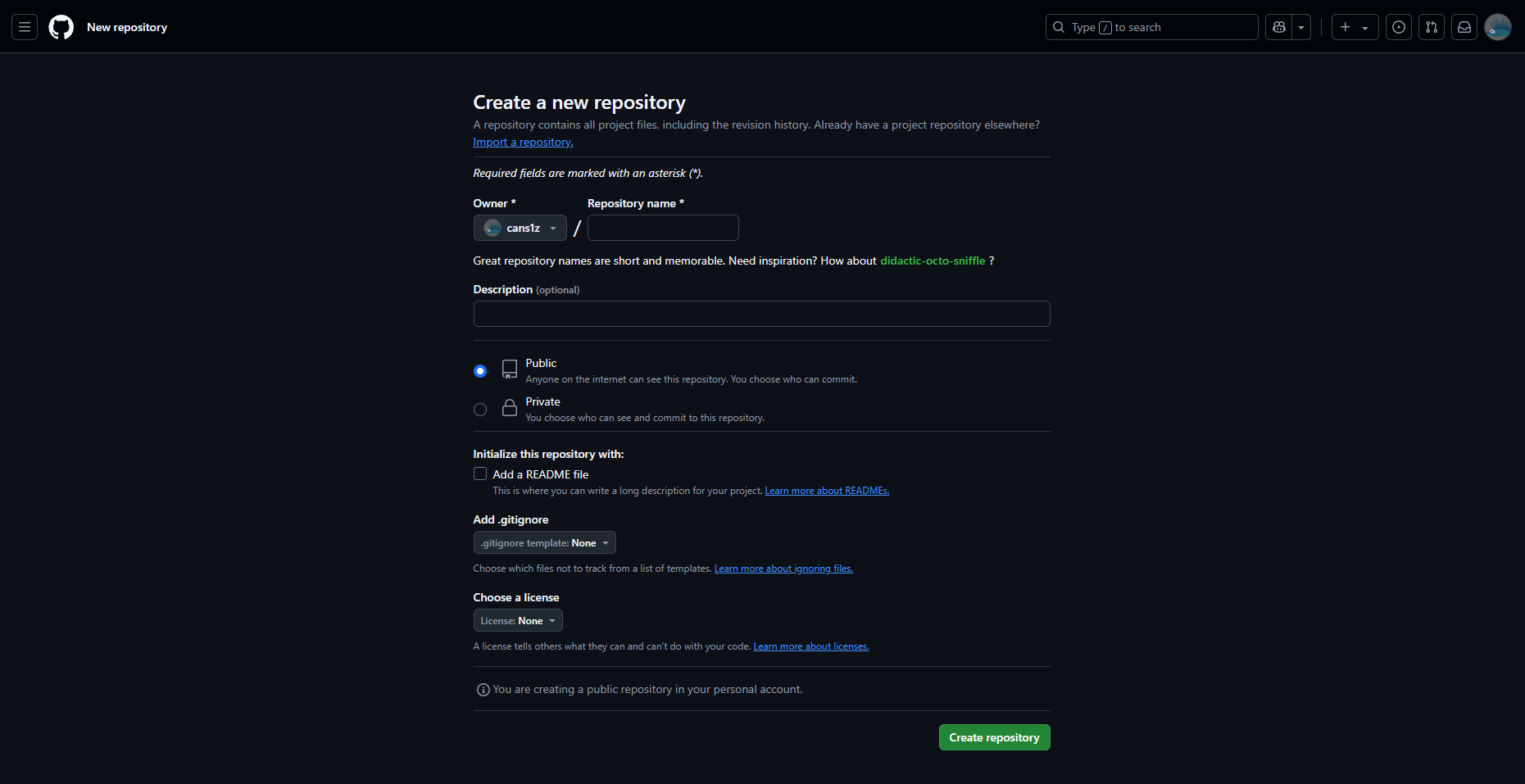


Рисунок 31 – Скриншот окна создания репозитория

Вывод:

В процессе настройки рабочего окружения в качестве основного языка программирования был выбран Python благодаря его удобству и высокой скорости разработки. Для работы с кодом использовалась среда разработки PyCharm. Кроме того, был создан удаленный репозиторий на GitHub для хранения и управления кодом проекта, а локальный репозиторий был привязан к нему. Это позволяет удобно отслеживать изменения, организовывать совместную работу и обеспечивать надежное хранение данных.