



CLIO — Command Line Intelligent Organizer

Персональний консольний асистент

CLIO (Command Line Intelligent Organizer) — це розумний консольний асистент, розроблений для оптимізації щоденних завдань будь-кого, **хто живе з телефоном у руках**. Наша мета — перетворити щоденні повторювані задачі на швидкі та ефективні рішення.

by RP-15 Development Team



color assest neut.

Cercralaigest foyiche nou guade t,
curtybinty falysion, Retil fenwil,
pecralies. Thte plnenatng to foletiim,
prereires attil melet with mual vellout.

Хто така CLIO?

Я — CLIO (Command Line Intelligent Organizer) Я створена, щоб впорядковувати твої контакти, нотатки та важливі дані. Я допомагаю прибрати хаос і повернути фокус на головне.

Наша Команда: RP-15 Development Team

```
class RP15DevelopmentTeam(Neoversity, OlehAndrus):
    skills = ["flexible", "smart", "friendly", "intelligent", "stress_resistant"]

    def to_learn(self): pass
    def to_communicate(self): pass
    def to_code(self): pass
    def to_support_each_other(self): pass
```



**Oleksandra
Sheronova**
Team Lead



**Dan
Zolotarenko**
Architect



**Khrystyna
Yasenyska**
Developer /
Scrum Master



**Denys
Yanushko**
Developer



**Hlib
Potanin**
Junior
Developer

Ідея Проекту CLIO:

CLIO — це персональний консольний асистент, що допомагає швидко керувати контактами та нотатками, економлячи час та зменшуючи хаос у повсякденних робочих процесах.



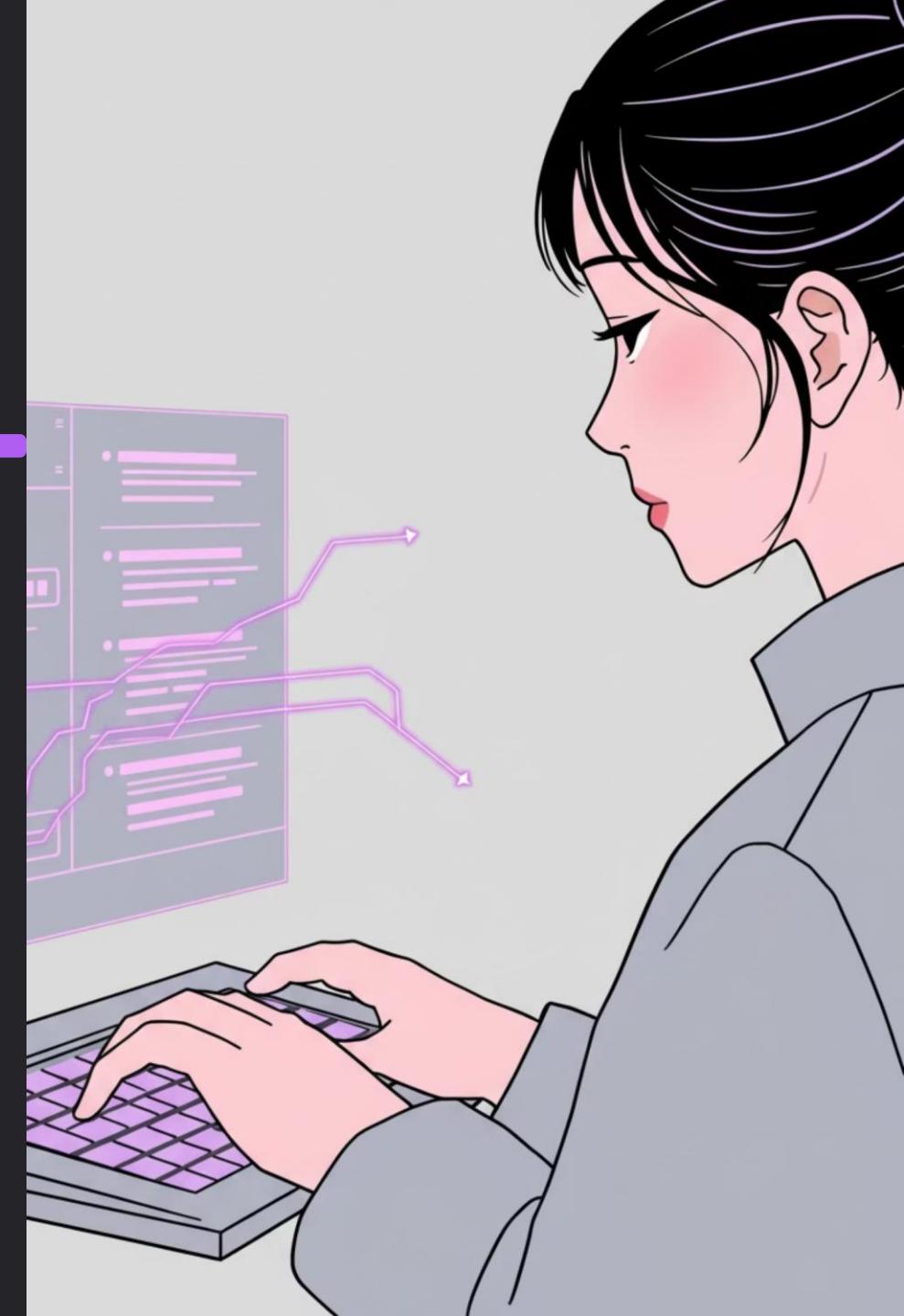
У чому цінність CLIO?

- Спрощує пошук, зберігання та оновлення інформації
- Забезпечує швидку взаємодію без зайвих кліків і відволікань
- Збирає всі робочі дані в одному місці
- Знижує ризик втрати важливої інформації



Особливо корисна для:

- Дизайнерів інтер'єрів
- Студентів та викладачів
- Менеджерів та креаторів
- PR-агентів та рекрутерів





Софія: Від Хаосу до Порядку з CLIO

Дизайнер інтер'єрів: Софія
28 років, «Найкраще місце на Землі»

Було (проблеми):

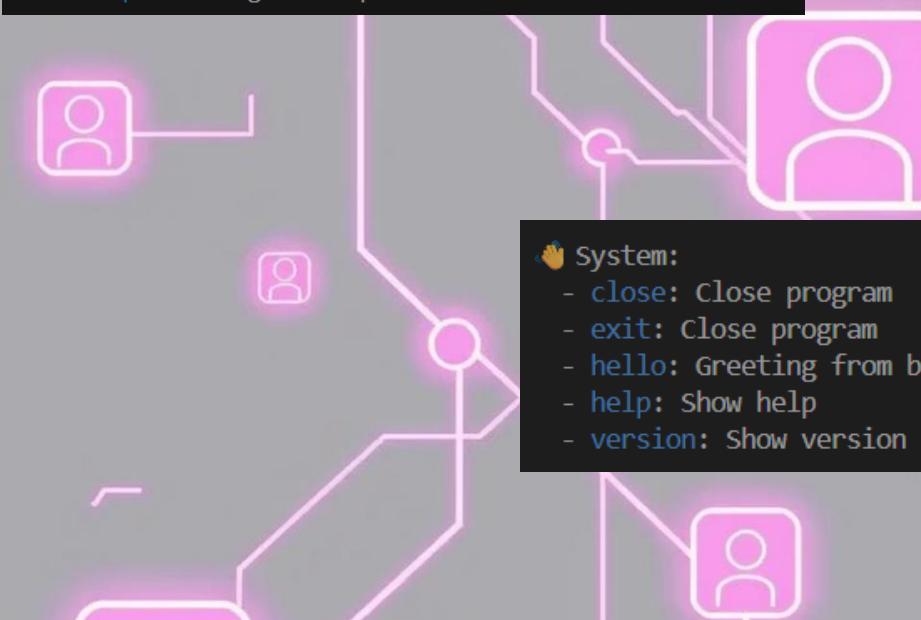
- Контакти в різних месенджерах
- Втрата важливої інформації
- Багато нотаток на папері та в телефоні
- Складно знайти правки
- Хаос у комунікації

Стало (рішення з CLIO):

- Впорядкована адресна книга
- Нотатки з тегами
- Швидкий пошук
- Єдине інформаційне середовище
- Ефективне управління даними

Phonebook:

- **add-address**: Usage: add-address "Name" "Kyiv, ..."
- **add-birthday**: Usage: add-birthday "Name" DD.MM.YYYY
- **add-contact**: Usage: add-contact "Name" [0123456789]
- **add-email**: Usage: add-email "Name" example@mail.com
- **all-contacts**: Show all contacts
- **birthdays**: Show upcoming birthdays (within 7 days)
- **change-phone**: Usage: change-phone "Name" old10 new10
- **delete-address**: Usage: delete-address "Name"
- **delete-contact**: Usage: delete-contact "Name"
- **delete-email**: Usage: delete-email "Name" example@mail.com
- **delete-phone**: Usage: delete-phone "Name" 0123456789
- **find-contact**: Usage: find-contact query
- **show-birthday**: Usage: show-birthday "Name"
- **show-phone**: Usage: show-phone "Name"



Notes:

- **add-note**: Usage: add-note "Title" text...
- **add-tags**: Usage: add-tags "Title" tag1 tag2 ...
- **all-notes**: List all notes (sort by title or created)
- **delete-note**: Usage: delete-note "Title"
- **delete-tag**: Usage: delete-tag "Title" tag
- **edit-note**: Usage: edit-note "Title" new_text...
- **find-note**: Usage: find-note query
- **find-tag**: Usage: find-tag tag

Функціональність CLIO: Контакти та Нотатки

CLIO надає потужний набір інструментів для ефективного управління вашими даними, зосереджуючись на ключових аспектах: контактах та нотатках.

Контакти

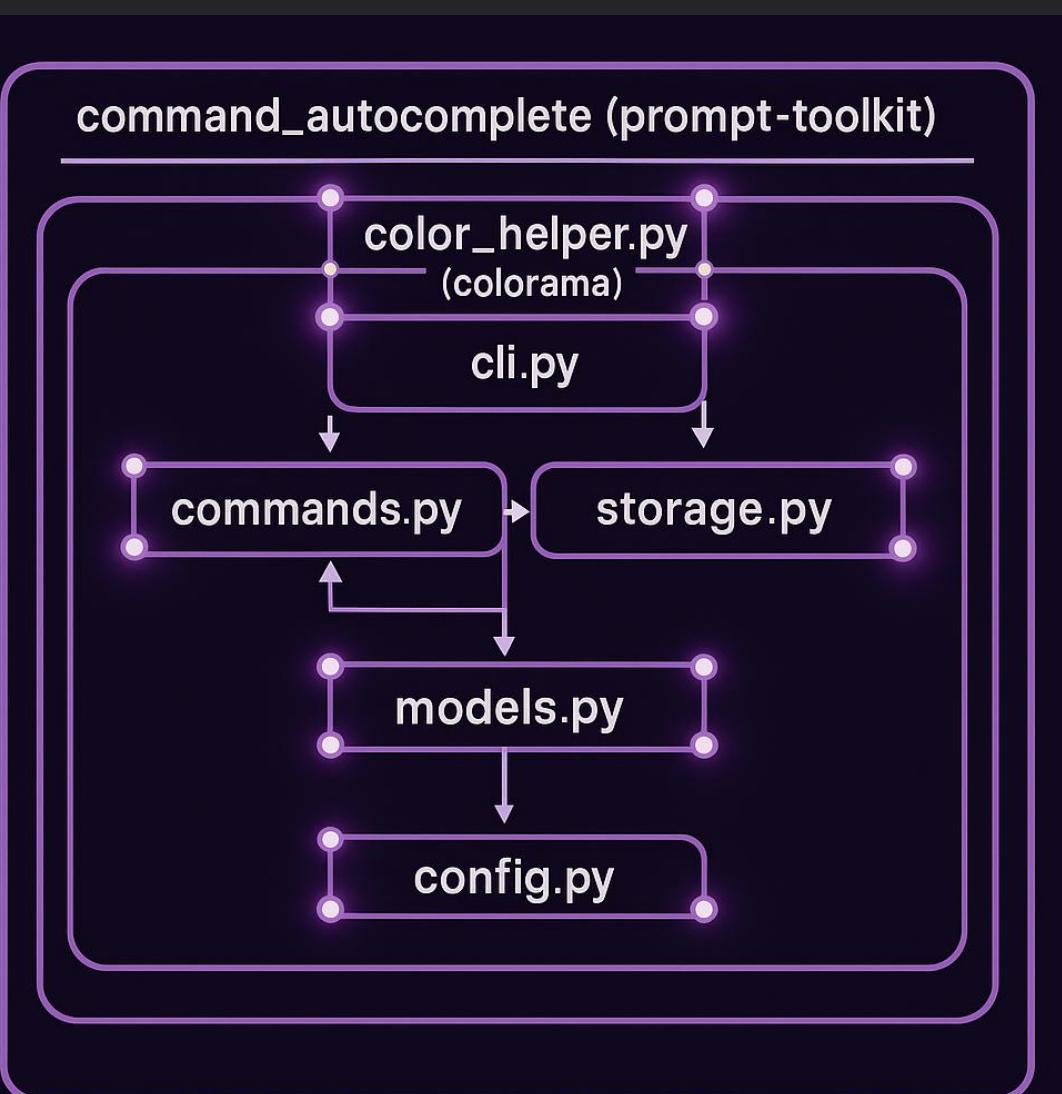
- Додавання нових контактів з детальною інформацією
- Редагування існуючих записів
- Видалення непотрібних контактів
- Швидкий пошук за будь-яким полем
- Зберігання адрес електронної пошти (email)
- Додавання фізичних адрес (address)
- Відстеження днів народження

Нотатки

- Створення нових нотаток з вільним текстом
- Редагування та оновлення змісту нотаток
- Видалення застарілих нотаток
- Присвоєння тегів для категоризації
- Пошук за текстом у нотатках
- Пошук за тегами для швидкого фільтрування

Архітектура Системи та Технологічний Стек CLIO

CLIO побудована на надійній та гнучкій архітектурі, що забезпечує стабільність та можливість подальшого розширення



Ми використовували сучасні технології та інструменти для створення ефективного консольного помічника.

Technologies



Python 3.13

Code written in Python language



Colorama

Library for terminal text coloring



Prompt-Toolkit

Powerful and interactive command-line applications



Regex

Regular expressions for pattern matching



Pickle

Serialization and deserialization of Python object structures

Tools



GitHub



Slack



VS Code



Trello



Google Meet

Backstage: Work in Progress

To Do

- Зробити презентацію проекту в ppt
- Додати валідацію високосного року в методі `get_next_birthday()`
- Рев'ю коду та коментарів

+ Add a card

In Progress

- Керування контактами - валідація Email
- *Optional* Інтелектуальний аналіз: command auto-complete

Done

- запуску - рісکів
- Керування нотатками - `add_notes`
- *Optional* додавання тегів в нотатки

Rework

- Керування контактами
- додавання Address

cli.py
color_helper.py
commands.py
config.py
main.py
models.py
README.md
requirements.txt
storage.py

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

- **find-note:** Usage: `find-note query`
- **find-tag:** Usage: `find-tag tag`

System:

- `close:` Close program
- `exit:` Close program
- `hello:` Greeting from bot
- `help:` Show help
- `version:` Show version

For command details use: `help <command>`
enter the command > `help`

add-address
add-birthday
add-contact
add-email
add-note
add-tags

▼ Docs

- ↓ 0_TZ.md
- ↓ 1_NAVIGATOR.md
- ↓ 2_SUMMARY.md
- ↓ 3_ARCHITECTURE.md
- ↓ 4_COMMANDS.md
- ↓ 5_Available_Bot_s_Commands.md
- ↓ 6_AUTOCOMPLETE.md
- ↓ Project_Q_AND_A.md
- ↓ TESTING.md

Подолання Викликів: Наш Шлях до Успіху

Кожна розробка має свої труднощі, і CLIO не була винятком. Ми зіткнулися з низкою викликів, але завдяки злагодженній командній роботі та інноваційним підходам, успішно їх подолали.

01

Розділення монолітного коду

Перетворення початкового моноліту на модульну структуру для підвищення керованості та масштабованості.

03

Впровадження Registry (фреймворк «ім'я → функція»)

Створення гнучкого механізму для реєстрації та виклику команд, що дозволило уникнути громіздких `if/elif` конструкцій.

05

Освоєння GitHub workflow

Активне використання `branches`, `Pull Requests` та вирішення `merge conflicts` для ефективної командної розробки.

02

Зниження зв'язності та підвищення читабельності

Оптимізація коду для покращення розуміння та спрощення подальшої підтримки.

04

Підтримка 25+ команд, покращений UI/UX (кольорування та автокомпліт команд)

Ефективне управління великою кількістю команд завдяки продуманій архітектурі.

06

Командна взаємодія

Комунікація та взаємна підтримка дозволили успішно пройти всі виклики та досягти поставлених цілей.

A stylized illustration of a woman with dark hair, wearing futuristic, transparent glasses and large purple headphones. She is looking slightly to the left. The background is dark and features several floating, semi-transparent digital icons, including a play button, a camera, a map, and various mobile application symbols.

THANK YOU