**📑 DOCUMENTAZIONE TECNICA PER RIFATTORIZZAZIONE DEL PROGETTO GENCHAT**

**1. 📂 STRUTTURA FINALE DEL PROGETTO**

plaintext

CopiaModifica

crisp/

│

├── \_\_init\_\_.py

│

├── ai\_interfaces/

│ ├── \_\_init\_\_.py

│ ├── genspark\_driver.py

│ ├── browser\_manager.py

│ ├── file\_text\_utils.py

│ └── interaction\_utils.py

│

├── framework/

│ ├── \_\_init\_\_.py

│ ├── crisp\_framework.py

│ ├── crisp\_extractors.py

│ ├── crisp\_utils.py

│ └── analysis/

│ ├── \_\_init\_\_.py

│ ├── market\_analysis.py

│ ├── buyer\_persona.py

│ └── gap\_analysis.py

│

├── ui/

│ ├── \_\_init\_\_.py

│ ├── app\_launcher.py

│ ├── ai\_connection.py

│ ├── chat\_manager.py

│ ├── cooldown\_manager.py

│ ├── book\_builder.py

│ ├── log\_utils.py

│ └── components.py

│

└── main.py

**2. 📜 DETTAGLIO CONTENUTI FILE PER FILE**

**ai\_interfaces/**

| **File** | **Contenuto** | **Fonte Originale** |
| --- | --- | --- |
| genspark\_driver.py | Logica principale connessione a Genspark (setup login, gestione sessione) | genschat.py |
| browser\_manager.py | Solo funzioni di setup e gestione browser Selenium (setup\_browser, ecc.) | genschat.py |
| file\_text\_utils.py | Funzioni per gestione file e testo (sanitize\_filename, split\_text) | genschat.py |
| interaction\_utils.py | Funzioni di invio testo, gestione input box, interazione pagina | genschat.py |

**framework/**

| **File** | **Contenuto** | **Fonte Originale** |
| --- | --- | --- |
| crisp\_framework.py | Logica del framework CRISP | crisp\_framework.py |
| crisp\_extractors.py | Estrazione dati da risposte AI | crisp\_extractors.py |
| crisp\_utils.py | Utility generali CRISP (replace\_variables, ecc.) | crisp\_utils.py |

**framework/analysis/**

| **File** | **Contenuto** | **Fonte Originale** |
| --- | --- | --- |
| market\_analysis.py | Funzioni analisi mercato (BSR, keyword trend) | crisp\_extractors.py |
| buyer\_persona.py | Funzioni creazione buyer persona | crisp\_extractors.py |
| gap\_analysis.py | Funzioni analisi gap (recensioni, criticità) | crisp\_extractors.py |

**ui/**

| **File** | **Contenuto** | **Fonte Originale** |
| --- | --- | --- |
| app\_launcher.py | Avvio dell'app Gradio e routing GUI/CLI | genschat\_gui.py |
| ai\_connection.py | Gestione connessione browser/Ai (connect\_callback) | genschat\_gui.py |
| chat\_manager.py | Gestione contesto chat, backup file, log | genschat\_gui.py |
| cooldown\_manager.py | Gestione cooldown tra richieste | genschat\_gui.py |
| book\_builder.py | Classe AIBookBuilder per costruzione libri CRISP | genschat\_gui.py |
| log\_utils.py | Gestione cronologia log, aggiunta e pulizia | genschat\_gui.py |
| components.py | (Futuro) Componenti UI separati se interfaccia cresce | genschat\_gui.py (inizialmente vuoto o minimale) |

**root/**

| **File** | **Contenuto** | **Fonte Originale** |
| --- | --- | --- |
| main.py | Entry point globale del progetto (avvio GUI o altri moduli futuri) | Nuovo (non esiste nei file caricati) |

**3. 🔄 IMPORT DA MODIFICARE**

**Principali correzioni di import:**

**Esempio di prima:**

python

CopiaModifica

from crisp\_utils import replace\_variables

**Deve diventare:**

python

CopiaModifica

from crisp.framework.crisp\_utils import replace\_variables

**Esempio per CRISPFramework:**

python

CopiaModifica

from crisp import CRISPFramework

**Deve diventare:**

python

CopiaModifica

from crisp.framework.crisp\_framework import CRISPFramework

**Esempio per setup\_browser (prima):**

python

CopiaModifica

from genschat import setup\_browser

**Deve diventare:**

python

CopiaModifica

from crisp.ai\_interfaces.browser\_manager import setup\_browser

**Regola generale:**

| **Se usi una funzione che prima era in** | **Devi importare da** |
| --- | --- |
| genschat.py | crisp.ai\_interfaces.browser\_manager / interaction\_utils / file\_text\_utils |
| crisp\_utils.py | crisp.framework.crisp\_utils |
| crisp\_framework.py | crisp.framework.crisp\_framework |
| crisp\_extractors.py | crisp.framework.crisp\_extractors oppure da analysis se separato |
| genschat\_gui.py | crisp.ui.\* (a seconda della classe/funzione) |

**4. 📈 GRAFICO STRUTTURALE COMPLETO**

plaintext

CopiaModifica

crisp/

│

├── ai\_interfaces/

│ ├── browser\_manager.py

│ ├── interaction\_utils.py

│ ├── genspark\_driver.py

│ ├── file\_text\_utils.py

│

├── framework/

│ ├── crisp\_framework.py

│ ├── crisp\_extractors.py

│ ├── crisp\_utils.py

│ └── analysis/

│ ├── market\_analysis.py

│ ├── buyer\_persona.py

│ └── gap\_analysis.py

│

├── ui/

│ ├── app\_launcher.py

│ ├── ai\_connection.py

│ ├── chat\_manager.py

│ ├── cooldown\_manager.py

│ ├── book\_builder.py

│ ├── log\_utils.py

│ └── components.py

│

└── main.py

**5. 📈 GRAFICO A FLUSSO DEL FUNZIONAMENTO**

plaintext

CopiaModifica

[ main.py ]

↓

[ app\_launcher.py (UI Launcher) ]

↓

[ AIBookBuilder (book\_builder.py) ]

↓

[ CRISP Framework (crisp\_framework.py) ]

↓

[ Analysis Modules (market\_analysis, buyer\_persona, gap\_analysis) ]

↓

[ AI Interfaces (browser\_manager, interaction\_utils, genspark\_driver) ]

↓

[ Browser Chrome + Genspark ]

**6. 🛡 REGOLE OPERATIVE PER IL RIFATTORIZZATORE**

* ❗ **Non modificare il codice interno** alle funzioni o classi: solo spostamenti fisici nei file e correzione import.
* ❗ **Controllare ogni import** dopo il refactoring e testare che non ci siano moduli non trovati.
* ❗ **Mantenere la logica di esecuzione identica**: solo l'organizzazione fisica cambia.
* ❗ **Se una funzione sembra incerta su dove collocarla**, preferire:
  + Se gestisce browser o Genspark → ai\_interfaces/
  + Se analizza dati → framework/analysis/
  + Se è logica principale di costruzione libro → framework/
  + Se è gestione UI, sessione, log → ui/