User Profile Creation Should be atomic above all This is very crucial.

Name\*

Email address

Social accounts multiple addable and password save able.

Contact details Optional

Job

Designation

Work place

House address

Friends names

Relatives names

Etc.

UI /UX

* User on right side Ai on left side.
* The Al text should be in continue writing word by word start appearing next second as user sends input.
* Ui Should look good and colour should be better.
* The typing of user or should I say input style should be like MS-word in which the character appears first then cursor moves.

1. Offline on Windows works offline responses offline.
2. Always available on screen with a beautiful UI.
3. Internet access is provided can search stuff online.
4. Has a Separate memory for keeping users preferences and whereabouts like name age liking disliking routine and other stuff and also if user give any command to remember keeping it in memory and using as a condition defined by user.
5. Demo (Clean Temp folder if it has size of temp folder exceeding 1 GB).
6. Can do text file operations (Create txt file, read txt file, write txt file).
7. Has access to files and folder structure.

Architecture  
  
AI AGENT/

├── run.py

├── agent/

│ ├── agent\_core.py

│ ├── memory\_bus.py ✅ Handles profile load/save/update

│ ├── session\_state.py

│ ├── command\_router.py

│ └── profile\_form.py ✅ (NEW) One-time GUI/CLI form for first run

├── interface/

│ ├── ui\_tk.py

│ └── event\_dispatcher.py

├── llm/

│ ├── engine.py

│ ├── prompt\_builder.py ✅ Includes profile in prompt

│ ├── stream\_parser.py

│ ├── model\_selector.py

│ ├── template\_formatter.py

│ └── instructions.py

├── tools/

│ ├── file\_search.py

│ ├── media\_downloader.py

│ ├── schedule\_manager.py

│ └── profile\_utils.py ✅ (NEW) Auto-extraction utils from chat

├── config/

│ ├── settings.yaml

│ ├── settings.py

│ └── templates.yaml

├── data/

│ ├── logs/

│ ├── sessions/

│ └── profile.json ✅ Stores current user profile

└── utils/

├── logger.py

└── async\_utils.py