HackMerlin Agent – Detailed Documentation

1. Overview
   HackMerlin Agent is an auto-solver that helps beat the HackMerlin challenge.
   It has two modes:

- Manual Mode → User enters password guesses.
- Auto-Solver Mode → Agent asks questions, collects answers, deduces candidates, and auto-submits until solved.

The system keeps logs of:

- All questions and answers asked to Merlin
- All candidate guesses tested

2. Flow of the Program

Main Menu

When the program starts, it shows a menu with three options:

1. Enter password (manual mode)
2. Auto-solver (automatic mode)
3. Exit

Manual Mode

- User enters a password.
  Check if it is 5–7 letters long.
  If not valid → show invalid → back to menu.
  If valid → submit to API.
- If correct → save level and move to next.
- If wrong → back to menu.

Auto-Solver Mode

- Run the solver loop:

    1. Ask the LLM for 10 new smart questions.
    2. Send them to the Merlin API and log all Q/A.
    3. Deduce candidate passwords. Use regex for uppercase hints and ask the LLM for suggestions.
    4. Submit each candidate to the API.
    5. If correct → save level and reset logs.
    6. If wrong → repeat until solved.

3. Code Walkthrough – File by File

main.py

- Entry point of the program

- Loads current level from levels.json
- Shows menu (Enter, Auto, Exit)
- Calls submit_password for manual mode or solve_with_llm for auto-solver

solver.py
- Core auto-solver loop
- Steps: ask LLM → send to Merlin → log → deduce candidates → submit → save if solved

agent.py
- Handles Merlin communication and logging
- agent_ask sends questions to Merlin API
- Logs answers into logs/level_prompt_log.txt
- reset_level_prompt_log clears old logs when solved

game_connector.py
- API connection functions
- ask_merlin sends POST to HackMerlin API and returns text response or error

submitter.py
- Handles sending guesses
- submit_password posts guess to API and returns result

levels.py
- Progress tracker
- load_levels reads current level from levels.json
- save_level updates solved password and moves to next

instructions.py
- Defines question strategies
- Level 1: ask full password
- Levels 2–6: ask letter by letter or in chunks
- Default: fallback rules
- Always 10 unique questions, avoid yes/no

llm.py
- Interface to LLM (DeepSeek)
- ask_llm sends prompts and returns responses

merlin_solver.py
- Deduce candidates from logs

- Reads Q/A logs and extracts uppercase leaks
- Asks LLM for candidate guesses
- Saves guesses to logs/answer_log.txt

config.py

- Loads .env credentials: SESSION, CF_CLEARANCE, DEEPSEEK_API_KEY

Logs

- level_prompt_log.txt stores Merlin Q/A until solved
- answer_log.txt stores candidate guesses until solved
- Both reset after solving a level

4. End-to-End Execution Flow

5. Start program

6. Show menu

7. Choose manual or auto

8. Manual → enter and submit password

9. Auto → ask questions, log answers, deduce candidates, submit, repeat

10. If solved → save level, reset logs, move to next