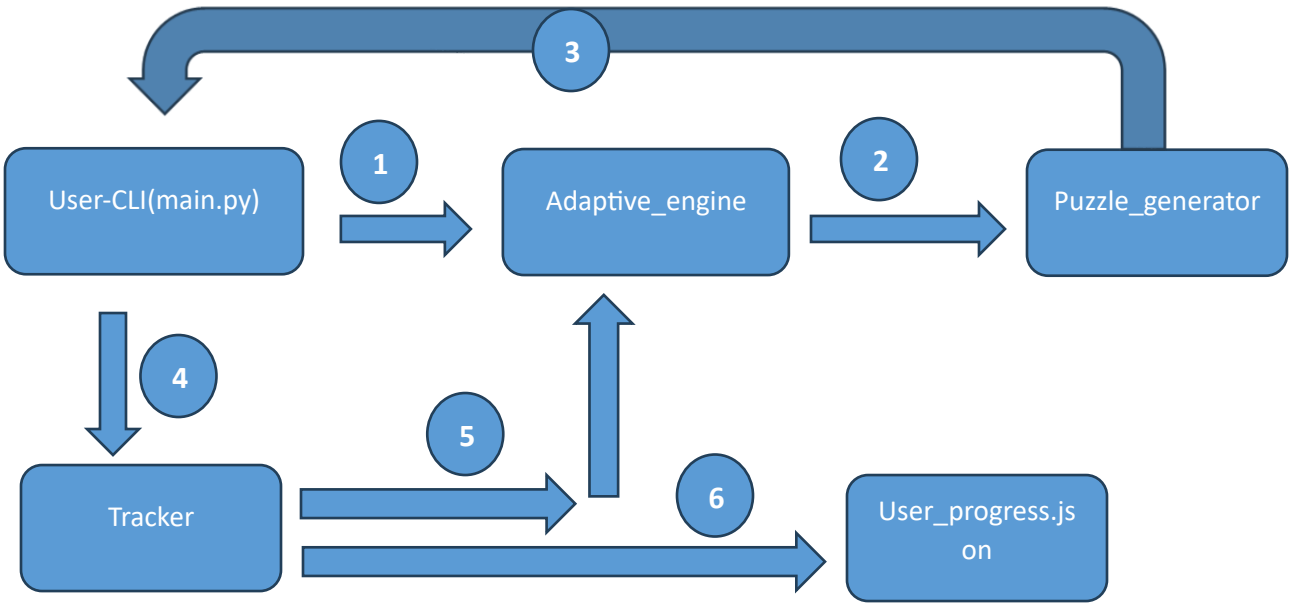
1. Architecture / Flow diagram.



2. Explanation of adaptive logic used

Summary: hybrid — rule-based primary layer with an ML fallback.

*. Detailed behavior (from `src/adaptive_engine.py`):

1. Primary, rule-based heuristics (checked first):

- Increase difficulty by one level if the last 3 answers are all correct.
- Decrease difficulty by one level if the last 2 answers are both incorrect.

2. ML fallback/assistant:

- A `LogisticRegression` model maps features to next difficulty index.
- Features: [previous difficulty index, time_taken, correct (0/1)].
- The file seeds the model with a small `X_dummy` / `y_dummy` dataset.
- When there is enough variation in labels from the user's history, the code refits the model on history-derived
- The model predicts next difficulty from the last answer's features.

3. Reason for hybrid approach

- Rules are deterministic and interpretable (safe default). ML provides nuance when streak rules don't apply.

3. Key metrics tracked and how they influence difficulty?

Tracked fields (saved by `src/tracker.py` via `record_answer`):

- `correct` (bool) — whether the user was correct
- `time_taken` (float seconds) — time spent answering
- `difficulty` (string) — difficulty label of the question asked

How these metrics influence difficulty in the current system:

A. Correctness (`correct`):

- Used directly by streak rules: 3 consecutive correct → move up; 2 consecutive incorrect → move down.

B. Response time (`time_taken`):

- Included as a raw scalar in the ML feature vector. The trained model coefficients (if any) determine whether faster responses increase difficulty.

C. Previous difficulty (`difficulty`):

- Encoded to an index and used both in rules (indirectly via `current_difficulty_index`) and in the ML features.

4. Why this approach was chosen (rationale)

Rationale for hybrid (rules + ML):

The streak and logic-based adaption is good for simple scenarios where the variables like time_taken, correctness and difficulty are predefined, hence the first execution is logic based. But this approach has flaws and cannot be personalized. The Logistic regression approach comes in handy, if we want to design the questionnaire according to a particular user's behaviour. ML approach is good when the pattern of user session is non-linear, as if, cannot be precisely hard coded. ML algorithms can be initially trained on certain datasets to get an idea of the mindset and then can finetune themselves realtime on a specific user.