Core Components Design Plan
Keshav Singh

1.1 User Input & Data Fetch

- User enters their Chess.com username
- Backend fetches game archives using the Chess.com API

Pseudo code for above
User inputs > usernamexyz

Backend runs > https://api.chess.com/pub/player/usernamexyz/games/archives

Returns > all monthly archive URLs from that player's history (for each archive, it fetches the games for that month)

1.2 Opening Detection (Using OpeningTree)

- Use https://github.com/openingtree/openingtree (open-source website previously existing to identify openings) based on PGN moves

Pseudo code for above
Return opening types and probabilities of occurring -> curl https://www.openingtree.com/api/player/usernamexyz (analyze through given months previously fetches)

1.3 PGN Parsing

- Use python-chess(built-in python chess library) to:
    - Parse PGN files for desired month of play
    - Extract move sequences, player color, result
    - Analyze annotations or run Stockfish to detect inaccuracies/blunders (maybe gemini, not sure on how to implement stockfish)

1.4 Data structuring

- Create some sort structured data per game
- Store in a pandas dataframe

Pseudo code for above
{

"game_id": "12345",

Core Components Design Plan
Keshav Singh

"date": "2023-09-15",

"color": "white",

"opening": "Sicilian Defense",

"result": "loss",

"mistakes": ["move_10", "move_22"] (use of gemini, not sure exactly how)

}

1.5 Implementation of AI

- Track Metrics like :
    1. Win/loss ratio per opening
    2. Mistake frequency by move number
    3. Blunder distribution by color
    4. Game length and performance decay
    5. Opponent opening frequency
- Use Random forests to analyze pd data frames to see which factors repeat often when the game ends in a loss.
- Use gemini or a custom rule-based engine to generate plain-language feedback

Pseudo code for above

```
model = genai.GenerativeModel(model_name='gemini-pro')

response = model.generate_content("

User has a  {win_rate} win rate against Sicilian Defense as {color}

Blunders occur frequently around move {10-15}.

Suggest 3 training tips based on this history.

")
to_markdown(response.text)
```

Core Components Design Plan
Keshav Singh

1.6 Presentation

- Frontend: Maybe a web-app, simple to do using a github generated website, not sure on how to translate this into a familiar IDE like xCode for myself

Features:

1. Home: Username input + submit button
2. Dashboard:
    - Chart of win/loss by opening
    - Mistake frequency graph
    - Gemeni personalized advice