Phase 6 v1 — UI + Ollama

1. Purpose of Phase 6

In Phase 6, we focus on **how you interact** with the prediction engine rather than how the predictions are generated.

We will:

• Wrap the core engine (database + feature logic + models) in a user interface that is convenient for you to use while managing your fantasy team.
• Integrate Ollama so the system can produce natural-language explanations and reports instead of just numbers.

The underlying math and modeling won't change here; this phase is about usability and presentation.

2. Target User Experience (Desktop-Oriented)

Your goal is to have something like a small desktop app or floating window you can open while your Sleeper league page is open in a browser. Conceptually, the experience might look like:

• You choose a week or "this upcoming week."
• You see your roster with projected fantasy points.
• You click two players and ask: "Who should I start?"
• The app shows a recommendation plus a short explanation.
• You click a "Waivers" tab and see ranked pickup recommendations.
• You click a "Matchups" tab and see predicted game scores.

We will not get bogged down in advanced UI libraries at first. The initial UI can be minimal but functional, built with technologies you are comfortable with.

3. UI Technology Options (Conceptual)

For a desktop-style experience, there are a few reasonable options:

1) Local web app:
   • Backend: Flask or FastAPI in Python.
   • Frontend: Simple HTML/CSS/JavaScript (or a basic framework like React later).
   • You run the app locally (e.g., http://localhost:5000) and open it in a browser window next to Sleeper.

2) True desktop GUI:
   • Use Python GUI frameworks like PyQt, PySide, or Tkinter.
   • These create windows that run natively on your OS.

3) Hybrid:
   • Run a local web server (Flask/FastAPI) and use something like Electron or Tauri later to wrap it into a desktop window.

Given your existing comfort with Flask and web tech, a **local web app** is probably the most straightforward starting point. You can treat it as a web UI, but since it runs locally, it behaves much like a desktop app for your purposes.

4. Basic UI Pages/Views to Implement

You do not need a huge, complex interface from day one. A small set of pages/views is enough:

1) Home / Dashboard
   • Shows an overview of:
     - This week's matchups for your team.
     - Quick summary of top start/sit decisions.
     - A link to detailed views.

2) Roster View
   • Shows your full roster (pulled from Sleeper).
   • For each player:
     - Projected fantasy points this week.
     - Optional ROS value.
   • Buttons or controls to:
     - Compare two players (start/sit).
     - Ask for an explanation via Ollama.

3) Waivers View
   • Shows waiver candidates with projected rest-of-season and short-term value.
   • Lets you filter by position, week range, etc.

4) Game Predictions View
   • Shows predicted scores and spreads for upcoming NFL games.
   • You can click into a matchup to see key players and their projections.

These views can be implemented as simple HTML pages rendering data from the backend.

5. Backend API Layer

Under the hood, the UI will call a simple API layer that sits on top of your existing prediction engine. Conceptually, this API will have endpoints such as:

- /api/projections/week/<week_number>
- /api/player/<player_id>/projection
- /api/start_sit?playerA=<id>&playerB=<id>&week=<week>
- /api/waivers
- /api/games/week/<week_number>/predictions

Each endpoint will:

1) Query the database for necessary features (or call precomputed tables).
2) Load the trained model(s) from disk.
3) Run the prediction logic to get statlines and fantasy points.
4) Return the results as JSON to the frontend.

This keeps your logic and data in one place (the backend) and the UI as a relatively thin presentation layer.

6. Integrating Ollama for Natural-Language Reports

Ollama will be used to turn raw prediction data into human-readable advice. Conceptually, the flow might look like:

1) The backend computes a player's projected stats, fantasy points, and some explanatory features (e.g., last3 form, opponent defense rank, etc.).
2) The backend builds a structured prompt for Ollama, such as:
   - "Explain why Player A is projected for X points this week versus Player B. Include matchup factors and recent performance."
3) Ollama processes this prompt and returns a paragraph or two of text.
4) The frontend displays this text alongside the numeric projections.

Possible use cases:

- Detailed start/sit explanation for two players.
- Weekly summary report for your roster (e.g., "Your biggest upside this week is at WR with Player X…")
- Waiver pickup justification (e.g., "Player Y is a strong pickup because…").

The key is that Ollama is not used for stat prediction; it is used for generating explanations and narratives based on the outputs of your stat models.

7. Privacy and Local-Only Considerations

Because this tool is initially for your personal use, you can run everything locally:

- The database lives on your machine.
- The models run on your machine.
- The web server/UI runs only on localhost.
- Ollama runs locally on your machine as well.

This means:

- No sensitive league or login data needs to be sent to remote servers.
- You retain full control over your data.
- Performance is generally fast because everything is local and there is no internet latency for most operations (except when updating data from Sleeper).

8. Iterative UI Development Strategy

You do not need to build the “perfect” UI in one shot. A good iterative strategy is:

1) Start with a simple page that lists projections for your roster for a single week, in a table.
2) Add start/sit comparison as a simple form where you choose two players from dropdowns and see the numeric comparison.
3) After that works, integrate a call to Ollama for a natural language explanation of that comparison.
4) Then add waivers and game prediction pages gradually.

This way, you always have something usable and can layer on complexity as you gain confidence and ideas.

9. What Success Looks Like for Phase 6

Phase 6 is successful when:

- You can open a local UI (in a browser or small window) while managing your Sleeper fantasy team.
- You can see projections for your players, waivers, and game matchups without digging into code or running manual scripts.
- You can ask the system “Who should I start?” and get both numeric and natural-language answers.

• You feel the system is easy enough to use that you actually want to use it each week during the season.

At that point, the project has evolved from a pure engineering/analytics experiment into a practical personal tool that supports real-world decisions in a user-friendly way.