Expected Product:

- Backend system that continuously pulls live ESPN fantasy data
- User-facing interface (dashboard or API) with analytics engine powered fantasy tools
    - Waiver & Injury Replacement Finder (Advanced)
        - Identify top waiver-wire players.
        - Detect injuries and impacted roster slots to recommend replacements.
    - Trade Analyzer (Advanced)
        - Evaluate proposed trades on short-term vs long-term value
        - Simulate post-trade outcomes
    - Boom / Bust Watch (Easy–Medium)
        - Track players significantly outperforming or underperforming preseason projections
    - Overall Player Rankings (Easy)
        - Rank players based on Season-to-date performance
    - Scheduling Tracker (Medium)
        - Display games-per-week for a selected NBA team
        - Display games-per-week for all NBA teams in a selected week
        - Helps optimize streaming and lineup decisions.
- Open-source GitHub repository with:
    - Data ingestion & update logic
    - Analytics modules for each tool
    - Database schema & documentation
- Final presentation with live demo using an active fantasy league

Dataset / Data Acquisition: ESPN Fantasy Basketball API (via espn-api Python library)
Reference Material

- ESPN Fantasy API: https://github.com/cwendt94/espn-api
- NBA Schedule Data: https://www.nba.com/schedule
- Pandas Time-Series Docs: https://pandas.pydata.org/docs/user_guide/timeseries.html
- AWS EC2 / Lambda (optional): https://aws.amazon.com/
- Google Cloud Platform (GCP): https://docs.cloud.google.com/docs
- Flask / FastAPI Docs: https://fastapi.tiangolo.com/

Week 2 – Onboarding & System Design

1. System Architecture Diagram – Decision Guidelines

Produce a **high-level architecture diagram** that clearly explains how data flows through the system and supports scalable fantasy analytics. The diagram and accompanying notes should address the questions below.

A. Data Ingestion

Define how fantasy and NBA data enters the system.

- What ESPN Fantasy data will be pulled?
  - Players, rosters, matchups, injuries, transactions, projections
- How will data be fetched?
  - On-demand
  - Scheduled (daily vs hourly)
  - Event-aware (game days vs off days)
- How will API limits, failures, and retries be handled?

Guideline:
Design ingestion to support frequent updates while remaining loosely coupled from analytics logic.

B. Data Storage

Define what data is stored and how it is structured.

- What data is persisted vs computed dynamically?
- Will the system store:
  - Historical player stats
  - Weekly matchup snapshots
  - Injury status over time
- What database type is used (relational, local file-based, hybrid)?

Guideline:
Choose a schema that supports historical analysis, trend detection, and "what-if" simulations.

C. Analytics Layer

Define how fantasy tools are implemented and shared.

- Are analytics modules independent services or part of a shared library?
- How do tools (waiver finder, trade analyzer, rankings) access common data?
- How is time-series data handled for trends and projections?

Guideline:
Analytics modules should be reusable, composable, and extensible as feature complexity increases.

D. Backend Interface

Define how analytics are exposed to users or other systems.

- Will the system expose:
  - REST API endpoints
  - Internal Python modules
  - A mix of API and dashboard views?
- How are concurrent requests and long-running simulations handled?

Guideline:
Adopt an API-first backend design, even if the UI is minimal or optional.

E. Deployment & Updates

Define how the system runs and updates over time.

- Will it run locally, in the cloud (AWS/GCP), or both?
- How are scheduled updates triggered?
- How are secrets and environment variables managed?

2. League Format & Scoring Rules – Required Decisions

Objective

Although this project builds a **fantasy basketball assistant** (not a league), the system must operate against a **fully specified league configuration**.
Define a complete league format and scoring rule set that the assistant will accept as input. No category may be left undecided.

A. League Format — Required Choices

Teams must select **one option** for each item and provide a short justification.

1. Competition Format

Choose how teams compete and how results are determined.

- ☐ Head-to-Head

- ☐ Rotisserie

Explain how this choice supports analytics such as trade evaluation, waiver prioritization, and performance trends.

2. Scoring Structure

Choose how fantasy value is calculated.

- ☐ Points-based scoring
- ☐ Category-based scoring

Explain how this structure impacts player rankings, boom/bust analysis, and trade simulations.

3. League Size

Define the number of teams (e.g., 10, 12, 14).

Justify how league size affects player scarcity and waiver-wire value.

4. Lineup Locking Rules

Define how often lineups can be changed.

- ☐ Daily lineup changes
- ☐ Weekly lineup locks

Explain how this choice impacts schedule tracking, streaming decisions, and injury replacements.

B. Scoring Rules — Required Definitions

Define **all scoring rules**, aligned with their chosen format.

1. Statistical Categories / Inputs

- List all NBA stats that count toward fantasy scoring.
- Justify why these stats represent player value and support meaningful analytics.

2. Stat Valuation Method

- Points leagues: assign weights to each stat.
- Category leagues: define each scoring category explicitly.

Explain how these choices affect player archetypes and trade balance.

3. Negative-Value Stats

- Explicitly include or exclude negative stats (e.g., turnovers).
- Justify how this impacts risk evaluation and boom/bust detection.

4. Roster & Position Rules

Define:

- Required starting slots
- Utility or flex positions
- Bench size
- Position eligibility rules

Explain how roster construction affects positional scarcity and replacement logic.

5. Scoring Timeframe

Define how scoring is aggregated:

- Daily totals
- Weekly totals
- Season-to-date accumulation

Justify how this timeframe supports short-term vs long-term analytics.

Required Documentation

- A system architecture diagram with labeled components and data flow
- A fully specified league format
- A complete scoring rule set covering all required categories
- Written justification for every decision
- Confirmation that all rules are compatible with ESPN Fantasy Basketball data