PB #1: Automated Scheduling of Matches

User Story:

As a **league administrator**,

I want to **auto-generate a complete season schedule for all teams and leagues**,

So that **I can ensure every team is assigned a fair number of matches within the available venues and timeslots without manually handling complex logistics**.

Acceptance Criteria:

- The system generates a schedule for all teams in each league without time overlaps.
- Each scheduled match includes team names, week, day, start/end times, league name, and assigned venue.
- Scheduled matches respect the number of fields and availability windows defined by the venues.

Definition of Done:

- Code implemented and integrated into the main application.
- All unit tests and integration tests for scheduling logic pass.
- Schedule output (schedule.csv) matches specified format and includes correct game details.

PB #2: Configurable Number of Games per League

User Story:

As a **league commissioner**,

I want to **set a maximum numberOfGames for each league**,

So that **the system only generates the required number of matchups, rather than scheduling all possible combinations**.

Acceptance Criteria:

- numberOfGames field in league.csv is read and respected by the scheduling logic.
- If numberOfGames is not provided or is invalid, the system defaults to scheduling all team matchups.

- The generated schedule is limited to the set number of games for that league and no more.

Definition of Done:

- Implementation respects numberOfGames from input data.
- Unit tests verify that fewer or more games are not scheduled than specified.
- Test cases 5–8 show correct truncation of matchups based on numberOfGames values.

PB #3: Venue Availability and Non-Overlapping Intervals

User Story:

As a **scheduling coordinator**,

I want to **ensure that no two matches overlap in the same field and that venues are used only during their specified season windows**,

So that **every scheduled match has a confirmed, conflict-free location and time**.

Acceptance Criteria:

- Matches only scheduled during the venue's active weeks (seasonStart to seasonEnd).
- No overlapping intervals on the same field: once a timeslot is taken, the next game must go to a different field or timeslot.
- System checks both day and time to detect and prevent collisions.

Definition of Done:

- IntervalTree or similar structure ensures no overlaps.
- Unit tests confirm that overlapping scenarios fail scheduling.
- Final schedule reviewed to confirm no duplicate field/time usage.

PB #4: Data Import and Export in Specified Formats

User Story:

As a **developer or admin**,

I want to **easily load team, venue, and league data from CSV inputs and produce a final schedule output in**

CSV (and optional JSON),

So that **I can quickly integrate this scheduling solution with external tools and maintain a simple workflow for updates**.

Acceptance Criteria:

- Reads team.csv, league.csv, venue.csv from a given directory.
- Produces schedule.csv and schedule.json (optional) with consistent formatting and headers.
- Handles file not found errors gracefully.

Definition of Done:

- Verified that all required input files are read correctly.
- Test with provided sample data directories.
- Confirm schedule.csv and schedule.json contain correct fields and formatting.