Project Title: MTG Pack Opening Simulator

Project Goal: To develop a web application that simulates the experience of opening Magic: The Gathering booster packs, providing a fun and insightful tool for enthusiasts.

Development Approach: Agile methodology with iterative sprints, focusing on delivering a functional product increment at the end of each sprint.

Sprint Breakdown & Deliverables:

- **Sprint 1: Core Simulation (MVP)**
    - **Goal:** To create a basic but functional simulation of opening a single MTG set (Foundations - 4BB) and displaying the results.
    - **Backend (Java/Spring Boot):** Will establish the foundation by setting up the project and integrating with the Scryfall API to fetch card data (name, image, price) for the chosen set. A simple API endpoint will be created to serve this data.
    - **Frontend (React - Basic UI):** A rudimentary React interface will be built to allow users to "open" a Foundations pack (via a button click) and display the names, images, and prices of the simulated cards.
- **Sprint 2: User Accounts & Data Persistence**
    - **Goal:** To enable user registration and login, and to begin storing basic user data related to their pack openings.
    - **Backend (Java/Spring Boot):** Will integrate existing login/authentication code with a PostgreSQL database to manage user accounts. The database schema for user accounts and basic pack opening history (set, date) will be implemented.
    - **Frontend (React - User Management):** React components for user registration and login will be developed, allowing users to create accounts and log in. Basic UI elements to indicate the logged-in state will be included.
- **Sprint 3: Enhanced UI & Engagement**
    - **Goal:** To significantly enhance the user experience through visual elements and interactive features.
    - **Frontend (React):** Will focus on creating a more engaging UI with a "drag-to-rip" pack opening animation (using a library like GSAP), a dynamic card display (potentially with a physics-based effect using Matter.js), and the initial implementation of a return-on-investment (ROI) meter to show the value of pulled cards versus the pack cost. Basic set selection (for the initial supported sets) will be implemented in the UI.

Technology Stack:

- **Backend:** Java with Spring Boot

- **Database:** PostgreSQL
- **Frontend:** React
- **API Integration:** Scryfall API

Key Features Demonstrated by End of Sprint 3:

- Simulation of opening MTG packs from a specific set with card details and pricing.
- Basic user account creation and login functionality.
- Storage of user's pack opening history.
- Visually engaging pack opening animation and card display.
- Initial feedback on the potential value of opened cards.

Educational Value:

This project will allow for the practical application of:

- Backend development with Java and Spring Boot.
- Frontend development with React.
- Database design and interaction with PostgreSQL.
- RESTful API design and integration.
- User authentication and security principles.
- Agile development methodologies and sprint-based work.
- UI/UX considerations and implementation of interactive elements.

This sprint-based approach ensures a manageable workload for each phase and allows for continuous progress and demonstrable results. The project leverages industry-standard technologies and methodologies, providing valuable learning experience.