**Epic: Augur System Implementation**

**User Story 1: User Dashboard**

* As a user, I want to access a browser-based dashboard to view metrics for PepsiCo distributors.
* Tasks:
  + Create a web-based dashboard.
  + Design the dashboard interface.
  + Implement user authentication.
  + Integrate with the Augur backend.

**User Story 2: Display Distributor Metrics**

* As a user, I want the dashboard to display metrics for each PepsiCo distributor, including name, year-to-date average quantity of goods shipped, quantity of goods shipped last month, and forecasted quantity of goods to ship this month.
* Tasks:
  + Define the data structure for distributor metrics.
  + Develop the frontend components to display metrics.
  + Implement backend APIs to retrieve and serve distributor metrics.

**User Story 3: Predictive Pipeline Integration**

* As a user, I want Augur to integrate with the existing predictive shipping pipeline to access monthly shipment quantities and next month's forecasts for each distributor.
* Tasks:
  + Define the data structure for pipeline integration.
  + Establish communication with messaging queues.
  + Develop services to fetch and process pipeline data.

**User Story 4: Data Storage**

* As a user, I want historical shipment data and forecasts to be stored in a data repository.
* Tasks:
  + Choose a suitable database system.
  + Design the database schema.
  + Implement data storage and retrieval operations.

**User Story 5: Azure Deployment**

* As a user, I want the Augur system to be deployed on the Azure cloud platform.
* Tasks:
  + Set up Azure infrastructure.
  + Configure server and networking environments.
  + Deploy system components to Azure.

**User Story 6: Extendibility**

* As a user, I want the system to be designed for extendibility to accommodate future growth and additional functionalities.
* Tasks:
  + Architect the system with modularity in mind.
  + Document system architecture and APIs.
  + Implement coding and naming conventions for maintainability.

These engineering tickets represent the major tasks and objectives for implementing the Augur system. You can organize these tickets on a Trello board or a similar project management tool, assign them to team members, and track their progress during sprints to build the system incrementally.

Top of Form

Bottom of Form