**Problem Statement:**

The problem is that detecting customer frustration in real-time is challenging and often relies on manual observation. This can lead to delayed responses and potentially escalations. The task is to build a DevRev snap-in that can analyze customer conversations and automatically trigger alerts for support agents when frustration is detected.

**Core Features:**

1. **Sentiment Analysis:**
   * The snap-in should continuously analyze the text of customer conversations to determine the sentiment expressed.
   * It should be able to identify signs of frustration, such as negative language, sarcasm, or urgency.
2. **Threshold Configuration:**
   * Allow users to set thresholds for when an alert should be triggered. This could be based on specific words, phrases, or overall sentiment scores.
3. **Alert Mechanism:**
   * When frustration is detected, the snap-in should notify support agents through various channels like email, in-app notifications, or even integrate with messaging platforms like Slack.
4. **Tracking and Reporting:**
   * The snap-in should keep a record of instances of detected frustration and generate reports to help teams analyze trends and improve their support strategies.

**Tools and Technologies:**

* **APIs:** Use DevRev APIs for fetching and posting notifications and comments, as well as email/Slack APIs for sending alerts to external services.
* **Languages:** Use TypeScript for development.
* **Platform:** Build the snap-in on the DevRev platform.

**Tips for Participants:**

* **Avoid Hardcoding:** Don't hardcode settings like Slack channels. Use DevRev snap-in inputs to allow users to configure them.
* **Design for Versatility:** Create a solution that can be used by any customer, not tied to specific data or scenarios.

**In essence:**

The goal is to build a tool that can proactively identify frustrated customers and alert support agents in real-time. This will enable timely intervention, prevent escalations, and improve overall customer satisfaction.

**My Github files follow this order**

**Customer-Frustration-Alerting-System/**

**├── src/**

**│ ├── index.ts # Main TypeScript file to run the app**

**│ ├── sentimentAnalysis.ts # Module for sentiment analysis**

**│ ├── alertMechanism.ts # Module for Slack and email alerts**

**│ ├── config.json # Configuration for thresholds**

**│ └── mockData.ts # Mock data for testing**

**├── dist/ # Compiled JavaScript files after running `npx tsc`**

**│ └── (auto-generated files here)**

**├── tsconfig.json # TypeScript configuration**

**├── package.json # Node.js dependencies**

**├── README.md # Project documentation**

**└── demo-data/ # Folder for demo or test data**

**└── demo-conversations.json**