1. Rapidly Automating a Banking Process (Workflow & API Integration):

Workflow Design

The dispute resolution process can be structured into the following workflow:

1. **User Submission**: Customers submit dispute details through a web form (e.g., Nodejs backend and integrated into an existing banking app).

2. **AI Classification**: An AI model (e.g., NLP-based classification using OpenAI's GPT) categorizes disputes into predefined types (e.g., fraud, billing error, service issue).

3. **Priority Assignment**: Based on customer transaction history and previous disputes:

    - **High Priority**: High-value transactions, VIP customers, repeat disputes.

    - **Medium Priority**: Mid-value transactions, single-time disputes.

    - **Low Priority**: Small transactions, first-time issues.

4. **Routing & Notification**:

    - Disputes are routed to the appropriate team (e.g., fraud team, customer service, billing team).

    - Support agents receive AI-generated recommendations for resolution.

Approach Explanation

(a) AI-Assisted or Low-Code Implementation

If I had to implement this with minimal custom code, I would leverage AI-assisted and low-code tools to streamline the process while integrating automation features. Here's how:

1. **User Input & Data Collection:**

    - Use **Google Forms** to collect dispute details without writing custom frontend code.

    - Integrate form submissions with an AI-powered workflow tool like **Zapier**, **Power Automate**

2. **AI-Based Dispute Classification:**
    - Use **OpenAI API** to classify disputes based on historical data without manually training models.
    - Example: A Zapier webhook can send dispute descriptions to an OpenAI API, which returns the category (e.g., fraud, billing, service).
3. **Priority Assignment & Routing:**
    - Use **low-code business process automation (BPA) tools** like **ServiceNow**, to assign priority levels based on transaction value, customer history, and AI insights.
    - Example: A ServiceNow workflow can flag high-priority disputes and route them directly to a fraud detection team.
4. **Notifications & Recommendations:**
    - Automate dispute notifications using **Slack,** via Zapier or Power Automate.
    - Example: AI-generated recommendations (from OpenAI/GPT-4) can be included in notifications to support agents for faster dispute resolution.

By using these AI-assisted and low-code tools, the implementation effort is significantly reduced while ensuring scalability and automation.

(b) Continuous Improvement with AI-Assisted Recommendations

Over time, AI can enhance dispute handling in several ways:

1. **Better AI Models for Classification:**
    - Collect real dispute data and train a **custom NLP model** (e.g., Hugging Face transformers).
    - Improve classification accuracy based on **customer feedback**.
2. **Predictive Analytics for Proactive Resolution:**

- Use **Machine Learning (ML)** to predict **high-risk customers** or **frequent dispute types** before they escalate.

- Example: If a customer has multiple disputes in the past 3 months, flag future disputes for high-priority review automatically.

3. **Sentiment Analysis for Prioritization:**

    - AI can analyze **customer sentiment** from dispute descriptions and escalate urgent cases based on emotion detection.

    - Example: If a dispute contains words like *"very angry," "legal action,"* or *"urgent,"* it can be auto-escalated.

4. **Continuous Feedback Loop:**

    - Use **A/B testing** to evaluate AI-generated recommendations and refine decision-making rules over time.

    - Implement **human-in-the-loop AI** where support agents can adjust AI suggestions to improve model accuracy.