

# Take-Home Assignment: Customer Service Support System

## Overview

Build an application using **Claude Code** - a Multi-Agent system to support customer service representatives at a bank.

## What You Need to Build

### 1. Knowledge Base - Banking Domain

Create a knowledge base with 5-10 documents covering topics such as:

- Account opening policies
- Loans and mortgage terms
- Fees and refunds
- Credit cards - types and benefits
- Branch hours and contact information
- Common app/website issues and troubleshooting

Transform this Knowledge Base into a searchable RAG system.

### 2. Multi-Agent System

Build a pipeline with three agents working together:

User Question → Reformulation Agent → Search Agent → Validation Agent → Response to Rep

#### Agent 1 - Reformulation Agent

- Receives the raw question from the representative
- Identifies the true intent behind the question
- Reformulates it into an optimized search query

#### Example:

- Rep writes: "Customer is yelling that money was stolen from his card"
- Agent reformulates: "Unauthorized credit card charge - dispute process and refund policy"

#### Agent 2 - Search Agent

- Receives the reformulated query

- Searches the Knowledge Base
- Returns an answer + source reference

### Agent 3 - Validation Agent

- Receives the answer from Agent 2
- Evaluates accuracy and relevance
- Returns a confidence score (percentage)

## 3. User Interface - Two Screens

### Screen 1: Representative View

- Input field for questions
- Display of the answer
- Confidence score (visual - color/icon)
- Reference - which document the answer came from
- Button/link to view the full source document

### Screen 2: Manager Dashboard

- Statistics on human users (e.g., number of queries per rep, topics asked, response usage)
- Statistics on the application and agents (e.g., average confidence scores, response times, most-used documents, low-confidence answers)

## Technical Requirements

- Everything must be built using Claude Code
- The code must run and work
- The UI must be clear and usable - does not need to be beautiful, but must be practical for daily use

## Deliverables

1. Source code (GitHub repo or ZIP file)
2. Brief instructions on how to run the application
3. Demo - short video (2-3 minutes) + live demonstration, showing:
  - A question being asked
  - The reformulated query (visible in UI or logs)
  - Answer returned with confidence score
  - Opening the source document
  - Manager dashboard with statistics

## Time Allocation

**Up to 2 hours** (not including time to learn Claude Code if you are new to the tool)

# Evaluation Criteria

Criteria	What We're Looking For
Multi-Agent	Three agents orchestrated correctly in sequence
RAG + DB	Search returns relevant results from Knowledge Base
Vibe Coding	Everything built through Claude Code
UX	A non-technical rep can use it without explanation
Manager Dashboard	Meaningful statistics on both users and system

Good luck!