**1. Basic Understanding: What is Epic, Story, Task, Sub-task?**

| **Term** | **Meaning** | **Example** |
| --- | --- | --- |
| **Epic** | A big body of work that can be broken down into smaller parts (Stories/Tasks). | "Build AI-powered Technical Support Chatbot" |
| **Story** | A feature or requirement that delivers some business value. | "User can ask questions and get AI answers" |
| **Task** | A piece of work needed to complete a Story. | "Integrate OpenAI model with UI" |
| **Sub-task** | A smaller action needed to complete a Task. | "Write API call to OpenAI", "Handle API errors" |

* **Epic** = Large Objective
* **Story** = Feature / Requirement
* **Task** = Work to achieve feature
* **Sub-task** = Detailed steps of a Task

**2. Sample Project:**

Let's take a project example:

🛠 **Project Name:**  
**"AI-Powered Technical Support Chatbot with RAG System"**

🧩 **Objective:**  
Develop a chatbot that can answer technical queries by fetching accurate information from a private document database (using Retrieval Augmented Generation - RAG).

**3. Requirements Document (Draft)**

**Requirements for "AI-powered Technical Support Chatbot":**

* The bot should accept user queries in text form.
* The bot should search internal documents for accurate answers (RAG system).
* Integrate OpenAI GPT-4 API.
* Provide a Web UI for chat.
* Admin panel to upload new documents.
* Maintain chat history for users.
* Add fallback to human support if bot confidence is low.

**4. EPIC → STORY → TASK → SUB-TASK (Draft)**

**📌 Epic 1: Build the Core Chatbot System**

Objective: Core functioning of chatbot that can respond based on document search + OpenAI.

**Story 1.1: Set up Retrieval Augmented Generation (RAG) backend**

* **Task 1.1.1:** Set up Document Storage (e.g., FAISS, ElasticSearch, Pinecone)
  + **Sub-task 1.1.1.1:** Install FAISS and configure server
  + **Sub-task 1.1.1.2:** Define document schema for storage
  + **Sub-task 1.1.1.3:** Set up document indexing pipeline
* **Task 1.1.2:** Integrate RAG pipeline
  + **Sub-task 1.1.2.1:** Write code to retrieve top relevant docs
  + **Sub-task 1.1.2.2:** Send documents and user query to GPT API
  + **Sub-task 1.1.2.3:** Parse and format GPT response

**Story 1.2: Develop Chat API**

* **Task 1.2.1:** Design Chat REST API
  + **Sub-task 1.2.1.1:** Define API endpoints (POST /chat, GET /history)
  + **Sub-task 1.2.1.2:** Implement authentication
* **Task 1.2.2:** Implement Chat Session Management
  + **Sub-task 1.2.2.1:** Create session tokens
  + **Sub-task 1.2.2.2:** Link chat history to session

**📌 Epic 2: Build Frontend User Interface**

Objective: Provide an easy and responsive Web UI for users to chat with the bot.

**Story 2.1: Design Chatbot UI**

* **Task 2.1.1:** Create Frontend using ReactJS
  + **Sub-task 2.1.1.1:** Setup project with React + TailwindCSS
  + **Sub-task 2.1.1.2:** Create Chat Input and Output Components
* **Task 2.1.2:** Connect UI to Backend APIs
  + **Sub-task 2.1.2.1:** Axios integration for Chat API
  + **Sub-task 2.1.2.2:** Show loading spinners during response wait

**Story 2.2: Implement Chat History View**

* **Task 2.2.1:** Display previous conversations
  + **Sub-task 2.2.1.1:** Fetch history via API
  + **Sub-task 2.2.1.2:** Show in chat window with timestamps

**📌 Epic 3: Admin Panel for Document Management**

Objective: Admins should be able to upload new documents to improve bot responses.

**Story 3.1: Build Admin Upload Feature**

* **Task 3.1.1:** Create Admin Authentication System
  + **Sub-task 3.1.1.1:** Build login page for Admin
  + **Sub-task 3.1.1.2:** Set up JWT-based authentication
* **Task 3.1.2:** Implement Document Upload UI
  + **Sub-task 3.1.2.1:** File input to upload PDFs
  + **Sub-task 3.1.2.2:** Send uploaded docs to backend for indexing

**📌 Epic 4: Human Support Escalation System**

Objective: If AI fails or has low confidence, escalate to human support.

**Story 4.1: Implement Low Confidence Detection**

* **Task 4.1.1:** Add Confidence Score Analysis to Bot
  + **Sub-task 4.1.1.1:** Define threshold score for escalation
  + **Sub-task 4.1.1.2:** Modify API to return confidence flag
* **Task 4.1.2:** Redirect User to Human Support
  + **Sub-task 4.1.2.1:** Show human support contact form

**🎯 Summary of Draft:**

| **Epic** | **Story** | **Task** | **Sub-Task** |
| --- | --- | --- | --- |
| Build Core Bot | Setup RAG backend | Setup Document Storage | Install FAISS |
|  |  |  | Define Schema |
|  |  |  | Index Pipeline |
|  | Integrate RAG pipeline | Code Retrieval | Retrieve Top Docs |
| Build Frontend | Chatbot UI | Create Frontend | Setup React |
|  |  |  | Create Chat Components |
| Admin Panel | Upload Feature | Admin Auth | Login Page |
|  |  |  | JWT Setup |
| Human Escalation | Confidence Handling | Analyze Score | Define Threshold |