**Project Title: Multi-turn Conversational Chatbot with RAG and Tool Integration**

### **Project Aims:**

* Implement multi-turn conversational chatbot with robust state management.
* Develop agentic planning for decision-making and actions.
* Integrate external tools (calculator API, custom FastAPI endpoints).
* Build retrieval-augmented generation (RAG) pipeline with vector-store (FAISS or Pinecone) and Text2SQL.
* Ensure chatbot robustness through handling unhappy flows gracefully.

### **System Architecture Flow:**

User Input  
 │  
 ▼  
Intent & Entity Parsing (LangChain) ────────┐  
 │ │  
 ▼ │  
Conversation State Management (LangChain) ◄─┘  
 │  
 ▼  
Agentic Planning & Controller (LangChain)  
 │  
 ├─────────► Missing Information?  
 │ │  
 │ └─ Yes ──► Clarifying Question (LangChain) ────┐  
 │ │  
 │ ▼  
 │ User Response  
 │ │  
 │◄────────────────────────────────────────────────────────────────┘  
 │  
 ▼  
Select Appropriate Action (LangChain)  
 │  
 ├───────────────────────────┐  
 │ │  
 ▼ ▼  
Calculator API Custom API & RAG Integration (FastAPI)  
Integration (FastAPI) ├─────────┬─────────┐  
 │ │ │ │  
 ▼ ▼ ▼ ▼  
Arithmetic Product Outlets Text2SQL Query  
Calculations Query Query (Outlet DB)  
(FastAPI) (FAISS/Pinecone) (FastAPI) (FastAPI)  
 │  
 ▼  
 SQL Execution (Outlet DB)  
 │  
 ▼  
 Query Results (FastAPI Response)  
 │ │ │ │  
 └───────────┐ │ │ │  
 │ │ │ │  
 Response Parsing (LangChain)│ │  
 │ │ │ │  
 ▼ ▼ ▼ ▼  
 Chatbot Response Generation (LangChain, using Retrieved Information)  
 │  
 ▼  
 Robustness and Error Handling (LangChain & FastAPI)  
 ├─────────┬─────────┐  
 │ │ │  
 ▼ ▼ ▼  
 Missing API Down Malicious  
 Parameters Time Inputs  
 (LangChain) (FastAPI) (FastAPI)  
 │ │ │  
 ▼ ▼ ▼  
Clear Error Graceful Security  
 Message Degradation Response  
(LangChain) (LangChain) (FastAPI)  
 │ │ │  
 └─────────┴─────────┘  
 │  
 ▼  
 Final Response to User (LangChain)

### **Project Timeline and Tasks:**

#### **Day 1: Initial Setup and Planning**

* Planning and Research:
  + Thoroughly read assessment requirements.
  + Setup GitHub repository.
  + Choose frameworks/tools (LangChain, Flowise, FastAPI, FAISS/Pinecone).
* **Deliverables:**
  + Project directory structure initialized.
  + README.md with preliminary architecture description.

#### **Day 2: Sequential Conversation Implementation (Part 1)**

* Define conversational state and memory management.
* Implement state management in LangChain/Flowise.
* Ensure 3-turn conversation flow is demonstrable.
* **Deliverables:**
  + Initial conversation state management system.
  + Automated tests for happy and interrupted conversation paths.

#### **Day 3: Agentic Planning and Tool Integration (Parts 2 & 3)**

* Develop planner/controller logic to parse intents and detect missing info.
* Integrate calculator API for arithmetic operations.
* Handle and test calculator API responses, including failures.
* **Deliverables:**
  + Planner/controller Python module.
  + Calculator integration module and test cases.
  + Short write-up on decision-making logic.

#### **Day 4: Custom API and RAG Setup (Part 4 Preparation)**

* Create FastAPI backend for product knowledge base and outlet SQL DB.
* Scrape and ingest ZUS product data into FAISS/Pinecone.
* Design initial Text2SQL schema for outlets.
* **Deliverables:**
  + FastAPI endpoints (/products, /outlets) skeleton.
  + Data ingestion scripts.
  + Draft OpenAPI spec.

#### **Day 5: Custom API and RAG Integration (Part 4 Completion)**

* Develop AI summarization for /products.
* Build Text2SQL translation mechanism for /outlets.
* Integrate chatbot with both endpoints.
* **Deliverables:**
  + Functional RAG pipeline.
  + Integrated and tested Text2SQL execution.
  + Example chatbot transcripts.

#### **Day 6: Robustness Testing and Error Handling (Part 5)**

* Implement error handling for edge cases.
* Develop test cases covering robustness (SQL injection, API downtime).
* **Deliverables:**
  + Complete robustness test suite.
  + Summary document on error handling and security.

#### **Day 7: Final Submission and Documentation**

* Finalize and review codebase.
* Host chatbot demo (Heroku/Vercel).
* Prepare documentation.
* **Deliverables:**
  + Public GitHub repository.
  + Live chatbot demo URL.
  + Documentation:
    - Setup and run instructions.
    - Architectural diagrams/screenshots.
    - Detailed API specifications.

### **Final Submission Checklist:**

* ✅ Public GitHub Repo link
* ✅ Hosted Demo URL
* ✅ README.md with setup instructions and architecture
* ✅ API documentation for RAG & Text2SQL endpoints
* ✅ Flow diagrams/screenshots of chatbot interactions

### **Tools & Resources:**

**Frameworks and Tools:**

* LangChain
* Flowise
* FastAPI
* FAISS / Pinecone (vector store)

**Documentation & Tutorials:**

* OpenAPI Specification Guide
* FastAPI Docs
* Heroku Deployment Guide
* Vercel Deployment Guide

**API & Data Sources:**

* ZUS Coffee Product Data
* ZUS Coffee Outlets Data

**Best Practices:**

* Robustness Testing
* Prompt Engineering for Text2SQL