VCU MDA Weekend AI Chatbot Documentation

Document Version: 2.0

Created Date: 2025-10-30

Author: AI Assistant

# Table of Contents

*Table of Contents will be generated automatically...*

# 1. Executive Summary

## 1.1 Overview

The VCU MDA Weekend AI Chatbot is an intelligent virtual assistant designed to provide prospective students with comprehensive information about the Master of Decision Analytics Weekend program at Virginia Commonwealth University. The system leverages advanced Natural Language Processing (NLP) and Artificial Intelligence (AI) technologies to deliver accurate, context-aware responses in real-time.

## 1.2 Key Features

* **AI-Powered Natural Language Understanding**
* **Semantic Search with Sentence Transformers**
* **Real-time Content Fetching from VCU Sources**
* **Intelligent Response Generation using OpenAI GPT**
* **Conversation Memory with Chroma DB**
* **Multi-source Information Integration**
* **User Intent Recognition and Classification**
* **Advanced Entity Extraction**

## 1.3 Technology Stack

**Frontend:** Streamlit - Python web framework for interactive applications

**NLP Processing:** spaCy - Industrial-strength natural language processing

**AI Models:** OpenAI GPT, Sentence Transformers - Intelligent response generation

**Vector Database:** Chroma DB - Persistent conversation storage and retrieval

**Web Scraping:** BeautifulSoup4 - Content extraction from VCU websites

**Machine Learning:** scikit-learn - Similarity calculations and clustering

# 2. System Architecture

## 2.1 High-Level Architecture

The chatbot follows a modular microservices architecture with the following components:

* **User Interface Layer** - Streamlit-based web interface handling user interactions
* **NLP Processing Layer** - Intent recognition, entity extraction, and semantic analysis
* **AI Response Layer** - OpenAI integration and rule-based response generation
* **Data Layer** - Chroma DB for conversation storage and vector embeddings
* **Content Layer** - Real-time data fetching from VCU websites and bulletin

## 2.2 Data Flow

1. 1. User submits query through Streamlit interface
2. 2. Query processed by NLP layer for intent and entity extraction
3. 3. System searches Chroma DB for similar past conversations
4. 4. Real-time content fetched from VCU sources if needed
5. 5. AI response generator creates context-aware answer
6. 6. Response displayed to user and stored in conversation history
7. 7. Conversation metadata saved to Chroma DB for future reference

# 3. Installation Guide

## 3.1 Prerequisites

* Python 3.8 or higher
* pip package manager
* Stable internet connection
* VCU website access (for content fetching)
* OpenAI API key (optional, for enhanced AI features)

## 3.2 Installation Steps

### Step 1: Clone or Download the Code

Save the chatbot code as VCU\_MDA\_V7\_AI.py in your preferred directory.

### Step 2: Create Requirements File

Create a requirements.txt file with the following content:

streamlit>=1.28.0  
requests>=2.31.0  
beautifulsoup4>=4.12.0  
pandas>=2.0.0  
numpy>=1.24.0  
spacy>=3.7.0  
sentence-transformers>=2.2.0  
scikit-learn>=1.3.0  
chromadb>=0.4.0  
python-docx>=1.1.0

### Step 3: Install Dependencies

Open terminal/command prompt and run:

pip install -r requirements.txt

### Step 4: Download spaCy Model

Install the English language model:

python -m spacy download en\_core\_web\_sm

### Step 5: Configure API Keys (Optional)

Create a .streamlit/secrets.toml file for OpenAI API key:

[openai]  
api\_key = "your-openai-api-key-here"

# 4. Configuration

## 4.1 Primary URLs

|  |  |  |
| --- | --- | --- |
| Variable | URL | Description |
| PRIMARY\_BASE | https://business.vcu.edu/graduate-programs/mda-weekend/ | Main program page |
| BULLETIN\_URL | https://bulletin.vcu.edu/graduate/school-business/decision-analytics-programs/decision-analytics-mda-pro/ | Official curriculum and requirements |
| APPLY\_URL | https://gradadmissions.vcu.edu/portal/apply?\_gl=1\*qxhpby\*\_gcl\_au\*NzE4MTA2NDk1LjE3NjE3ODc2NTE.\*\_ga\*Mjk1NjQ2NzY5LjE3NTk3NjQ3MTg.\*\_ga\_WMHV0FXMBD\*czE3NjE4NjAxNDEkbzEkZzEkdDE3NjE4NjAzMzAkajYwJGwwJGhw | Application portal |
| REFERRAL\_URL | https://business.vcu.edu/graduate-programs/mda-weekend/referral-award/ | Alumni referral program |

## 4.2 AI Model Configuration

**Sentence Transformer Model:** all-MiniLM-L6-v2 - Semantic similarity calculations

**OpenAI Model:** gpt-3.5-turbo - Intelligent response generation

**spaCy Model:** en\_core\_web\_sm - NLP processing and entity recognition

**Similarity Threshold:** 0.3 - Minimum similarity score for context matching

**Response Temperature:** 0.3 - AI creativity control (lower = more factual)

# 5. User Guide

## 5.1 Getting Started

To launch the chatbot:

1. Open terminal/command prompt in the project directory
2. Run the command: streamlit run VCU\_MDA\_V7\_AI.py
3. The application will open in your default web browser
4. Wait for the AI components to initialize (indicated by status messages)

## 5.2 Using the Chatbot

### Basic Operations

**Ask Questions:** Type natural language questions in the chat input box

**Quick Actions:** Use sidebar buttons for common questions

**Load Content:** Click 'Load AI Context' to fetch latest program information

**View Analysis:** Enable 'Show Analysis' to see NLP processing details

## 5.3 Example Questions

### Admission Questions

* What are the application deadlines?
* What qualifications do I need?
* How do I apply to the program?
* Is GMAT required for admission?

### Program Questions

* How long is the program?
* What is the program format?
* What will I learn in the curriculum?
* Tell me about the practicum experience

### Financial Questions

* What is the total cost?
* Are there scholarships available?
* Tell me about the alumni referral award
* What payment options are available?

### Advanced AI Questions

* How does this program compare to other analytics programs?
* What career outcomes can I expect?
* How does the weekend format benefit working professionals?
* What makes this program unique?

# 6. Technical Details

## 6.1 Core Classes

### AINLPProcessor

Responsibilities:  
- Natural Language Processing operations  
- Intent recognition and classification  
- Entity extraction using spaCy  
- Semantic similarity calculations  
  
Key Methods:  
- extract\_entities(text): Extract named entities from text  
- analyze\_intent(query): Determine user intent from query  
- semantic\_similarity(text1, text2): Calculate similarity score

### AIResponseGenerator

Responsibilities:  
- Generate intelligent responses using AI  
- Manage knowledge base and context  
- Handle rule-based and AI-generated responses  
  
Key Methods:  
- generate\_ai\_response(query, context): Main response generation  
- find\_most\_relevant\_context(query, chunks): Semantic context matching  
- generate\_rule\_based\_response(): Fallback response system

### AIChromaDBChatHistory

Responsibilities:  
- Persistent conversation storage  
- Semantic search across chat history  
- Conversation metadata management  
  
Key Methods:  
- add\_message(): Store conversation with metadata  
- get\_similar\_conversations(): Find related past conversations  
- initialize\_db(): Chroma DB setup and configuration

## 6.2 Data Structures

**Knowledge Base:** Hierarchical dictionary organizing program information by categories

**Conversation Metadata:** JSON structure storing query, response, intent, timestamp, and sources

**Content Chunks:** List of processed sentences from VCU websites for context

**Entity Objects:** Structured entities with text, label, and position information

# 7. API Reference

## 7.1 Method Specifications

### AINLPProcessor.analyze\_intent(query)

Parameters:  
- query (str): User's input question  
  
Returns:  
- str: Detected intent category ('admission', 'curriculum', 'financial', 'deadline', 'general\_info')  
  
Description:  
Analyzes user query to determine primary intent using keyword matching and pattern recognition.

### AIResponseGenerator.generate\_ai\_response(query, context\_chunks)

Parameters:  
- query (str): User's input question  
- context\_chunks (list): List of relevant content sentences  
  
Returns:  
- str: AI-generated or rule-based response  
  
Description:  
Main response generation method that combines AI capabilities with contextual information.

### AIChromaDBChatHistory.get\_similar\_conversations(query, limit=3)

Parameters:  
- query (str): Current user question  
- limit (int): Maximum number of similar conversations to return  
  
Returns:  
- list: Similar past conversations with similarity scores  
  
Description:  
Finds semantically similar past conversations using vector similarity search.

# 8. Troubleshooting

## Chatbot not starting

* Check Python version (requires 3.8+)
* Verify all dependencies are installed
* Ensure no port conflicts (Streamlit uses port 8501)

## AI features not working

* Verify spaCy model is downloaded
* Check internet connection for API calls
* Validate OpenAI API key in secrets.toml
* Check console for import errors

## Content fetching failures

* Verify VCU websites are accessible
* Check network connectivity
* Review error messages in console
* Try loading content manually using sidebar button

## Slow response times

* Reduce number of simultaneous context chunks
* Disable semantic search if not needed
* Use rule-based responses instead of AI
* Check system resources and memory

# 9. Maintenance

## 9.1 Regular Maintenance Tasks

* Update dependencies periodically
* Verify VCU URLs are still valid
* Refresh content context regularly
* Monitor AI API usage and costs
* Backup Chroma DB conversation history
* Review and update response templates

## 9.2 Update Procedures

1. 1. Pull latest code changes
2. 2. Update requirements.txt with new dependencies
3. 3. Run dependency installation: pip install -r requirements.txt
4. 4. Test all chatbot features
5. 5. Update documentation if needed
6. 6. Deploy updated version