

CMU-Africa Campus Assistant - Local Deployment Guide

Deployment Status:  **FULLY OPERATIONAL**

Last Updated: October 15, 2025



Current Deployment Status

Services Running

-  **Backend API:** Running on `http://localhost:8001`
-  **Frontend App:** Running on `http://localhost:3000`
-  **Vector Database:** Pinecone (8 vectors indexed)
-  **AI Model:** OpenAI GPT-4 with embeddings

Health Check Results

```
{  
  "status": "healthy",  
  "rag_pipeline": "initialized",  
  "vector_store_stats": {  
    "total_vectors": 8,  
    "dimension": 1536  
  }  
}
```

1 How to Access the Application

Frontend (User Interface)

URL: `http://localhost:3000`

Open this URL in your web browser to access the full chat interface with:

- Interactive chat interface with AI assistant
- Quick suggestion pills for common queries
- Collapsible source citations
- Follow-up question recommendations
- Mobile-responsive design

Backend API (Development/Testing)

URL: `http://localhost:8001`

- **Root Endpoint:** `http://localhost:8001/`
- **API Documentation:** `http://localhost:8001/docs` (Interactive Swagger UI)
- **Alternative Docs:** `http://localhost:8001/redoc` (ReDoc format)

Quick Test

Open your browser and navigate to:

```
http://localhost:3000
```

You should see the CMU-Africa Campus Assistant welcome screen with a robot mascot and suggestion cards.

2 Available API Endpoints

Base URL

```
http://localhost:8001
```

Endpoints Overview

Root Endpoint

```
GET /
```

Response:

```
{
  "message": "CMU-Africa Campus Assistant API",
  "version": "1.0.0",
  "status": "active"
}
```

Chat Query (Main Endpoint)

```
POST /api/chat
```

Request Body:

```
{
  "message": "What programs does CMU-Africa offer?",
  "user_profile": {
    "program": "MSIT",
    "year": 2
  },
  "session_id": "optional-session-id"
}
```

Response:

```
{
  "answer": "CMU-Africa offers three Master's degree programs: \n- Master of Science in Information Technology (MSIT)\n- Master of Science in Electrical and Computer Engineering (MSECE)\n- Master of Science in Engineering Artificial Intelligence (MSEAI)",
  "sources": [
    {
      "id": "degrees_masters_1",
      "title": "Master's Programs at CMU-Africa",
      "snippet": "CMU-Africa offers three Master's degree programs...",
      "category": "Academic Programs"
    }
  ],
  "suggestions": [
    {
      "id": "housing_options",
      "label": "🏠 Housing",
      "prompt": "What housing options are available?"
    }
  ],
  "follow_up": "Would you like to see the course curriculum details?"
}
```

cURL Example:

```
curl -X POST http://localhost:8001/api/chat \
-H "Content-Type: application/json" \
-d '{"message": "What are the library hours?"}'
```

Health Check

[GET /api/health](#)

Response:

```
{
  "status": "healthy",
  "rag_pipeline": "initialized",
  "vector_store_stats": {
    "total_vectors": 8,
    "dimension": 1536
  }
}
```

Index Statistics

[GET /api/index/stats](#)

Response:

```
{
  "total_vectors": 8,
  "dimension": 1536
}
```

Add Documents to Index

`POST /api/index/documents`

Request Body:

```
[
  {
    "id": "unique_doc_id",
    "title": "Document Title",
    "content": "Full document content here...",
    "category": "Academic Programs",
    "keywords": ["keyword1", "keyword2"]
  }
]
```

Response:

```
{
  "message": "Successfully indexed 1 documents",
  "indexed_count": 1
}
```

3 Test Queries to Try

Quick Test Queries

1. Academic Programs

What programs does CMU-Africa offer?

Expected: Information about MSIT, MSECE, and MSEAI programs

2. Library Hours

What are the **library** hours?

Expected: Monday-Friday 8:00 AM - 10:00 PM, Weekends 10:00 AM - 8:00 PM

3. Transportation

What are the shuttle bus timings?

Expected: Shuttle operates weekdays 7:00 AM - 10:00 PM with departure times

4. Housing Information

Tell me about housing options

Expected: Information about on-campus and off-campus housing

5. Campus Events

What events are happening this week?

Expected: Information about tech talks, cultural events, and activities

6. Administration Contact

How can I contact the administration office?

Expected: Office location, hours, and contact details

Testing via cURL

```
# Test 1: Programs Query
curl -X POST http://localhost:8001/api/chat \
-H "Content-Type: application/json" \
-d '{"message": "What programs does CMU-Africa offer?"}' | jq .'
```



```
# Test 2: Library Hours
curl -X POST http://localhost:8001/api/chat \
-H "Content-Type: application/json" \
-d '{"message": "What are the library hours?"}' | jq .'
```



```
# Test 3: Health Check
curl http://localhost:8001/api/health | jq .'
```

Testing via Browser

1. Open <http://localhost:3000>
2. Click on any suggestion card (e.g., “What programs does CMU-Africa offer?”)
3. View the response with expandable sources
4. Try follow-up suggestions that appear below
5. Type custom queries in the input field

4 How to Stop/Restart Services

Stop Services

Stop Backend

1. Go to the terminal running the backend
2. Press `Ctrl + C`
3. Wait for graceful shutdown

4. (Optional) Kill process if needed:

```
bash
# Find the process
ps aux | grep unicorn
# Kill it
kill <PID>
```

Stop Frontend

1. Go to the terminal running the frontend

2. Press `Ctrl + C`

3. Wait for graceful shutdown

4. (Optional) Kill process if needed:

```
bash
# Find the process
ps aux | grep node
# Kill it
kill <PID>
```

Stop All Services at Once

```
# Kill all backend processes
pkill -f unicorn

# Kill all frontend processes
pkill -f "react-scripts start"
```

Restart Services

Method 1: Using Start Scripts (Recommended)

Start Backend:

```
cd /home/ubuntu/code_artifacts/cmu-africa-campus-assistant
./start_backend.sh
```

Start Frontend (in a new terminal):

```
cd /home/ubuntu/code_artifacts/cmu-africa-campus-assistant
./start_frontend.sh
```

Method 2: Manual Start

Backend:

```
cd /home/ubuntu/code_artifacts/cmu-africa-campus-assistant/backend
source venv/bin/activate
python main.py
# Or: unicorn main:app --reload --host 0.0.0.0 --port 8001
```

Frontend:

```
cd /home/ubuntu/code_artifacts/cmu-africa-campus-assistant/frontend
npm start
```

Check if Services are Running

```
# Check ports
netstat -tulnp | grep -E "8001|3000"

# Expected output:
# tcp 0.0.0.0:3000  LISTEN <PID>/node
# tcp 0.0.0.0:8001  LISTEN <PID>/python
```

5 Configuration & Customization

Environment Variables

Backend Configuration

Location: `/home/ubuntu/code_artifacts/cmu-africa-campus-assistant/backend/.env`

```
OPENAI_API_KEY=sk-proj-...
PINECONE_API_KEY=pcsk_...
PINECONE_ENVIRONMENT=us-east-1
```

To Change:

1. Edit the `.env` file
2. Restart the backend service
3. Changes take effect immediately

Frontend Configuration

Location: `/home/ubuntu/code_artifacts/cmu-africa-campus-assistant/frontend/.env`

```
REACT_APP_API_BASE_URL=http://localhost:8001
```

To Change Backend URL:

1. Edit `REACT_APP_API_BASE_URL` to point to a different backend
2. Restart frontend with `npm start`

Adding New Knowledge to the System

Step 1: Prepare Your Data

Create a JSON file with your knowledge base:

```
[
  {
    "id": "unique_id",
    "title": "Document Title",
    "category": "Category Name",
    "content": "Full content of the document...",
    "keywords": ["keyword1", "keyword2"]
  }
]
```

Step 2: Load via API

```
curl -X POST http://localhost:8001/api/index/documents \
-H "Content-Type: application/json" \
-d @your_data.json
```

Step 3: Verify

```
curl http://localhost:8001/api/index/stats
# Should show increased vector count
```

Customizing UI Colors

Edit: `frontend/tailwind.config.js`

```
module.exports = {
  theme: {
    extend: {
      colors: {
        cmu: {
          red: '#C41230', // Change to your brand color
          gray: '#6B6B6B',
        }
      }
    }
  }
}
```

Restart frontend to see changes.

Modifying AI Behavior

Edit: `backend/rag_pipeline.py`

Change System Prompt:

```
system_prompt = f"""You are the CMU-Africa Campus Assistant...
[Modify this to change AI personality and instructions]
"""
```

Adjust Response Style:

- Line ~150: Modify suggestion generation logic
- Line ~200: Adjust follow-up question generation
- Line ~50: Change vector search parameters

Restart backend after changes.

Adding New Suggestion Categories

Edit: `backend/rag_pipeline.py`, function `_generateSuggestions()`

```
suggestion_templates = {
    'YourNewCategory': [
        {
            'id': 'action_id',
            'label': '🔥 Your Label',
            'prompt': 'Full prompt text here'
        },
        # Add more suggestions...
    ]
}
```



Troubleshooting

Backend Issues

Issue: Failed to initialize Pinecone index

- **Solution:** Check Pinecone API key and ensure you have index capacity

Issue: OpenAI API Error

- **Solution:** Verify OpenAI API key and check account credits

Issue: Port 8001 already in use

- **Solution:**

```
bash
# Find process using port
lsof -i :8001
# Kill it
kill -9 <PID>
```

Frontend Issues

Issue: Failed to get response

- **Solution:** Ensure backend is running on port 8001
- Check: `curl http://localhost:8001/api/health`

Issue: CORS errors

- **Solution:** Backend already configured for localhost:3000
- If using different port, update CORS in `backend/main.py`

Issue: `npm start` fails

- **Solution:**

```
bash
rm -rf node_modules package-lock.json
npm install
npm start
```

Project File Structure

```
cmu-africa-campus-assistant/
├── backend/
│   ├── main.py                                # FastAPI application
│   ├── rag_pipeline.py                         # RAG logic and AI integration
│   ├── requirements.txt                        # Python dependencies
│   ├── .env                                    # Environment variables (API keys)
│   ├── load_knowledge_base.py                 # Script to load sample data
│   └── venv/                                   # Virtual environment
|
├── frontend/
│   ├── src/
│   │   ├── components/                         # React components
│   │   │   ├── ChatMessage.tsx
│   │   │   ├── ChatInput.tsx
│   │   │   ├── SuggestionPills.tsx
│   │   │   └── WelcomeScreen.tsx
│   │   ├── services/                           # API client
│   │   │   └── api.ts
│   │   ├── App.tsx                            # Main app
│   │   └── index.tsx                          # Entry point
│   ├── public/
│   ├── package.json                           # Frontend config
│   └── .env
|
└── data/
    └── sample_knowledge_base.json # Sample CMU-Africa data
|
├── start_backend.sh                         # Backend startup script
├── start_frontend.sh                        # Frontend startup script
├── setup.sh                                 # Initial setup script
└── README.md                                # Main documentation
```

Key Features Implemented

Frontend Features

- 💬 Real-time chat interface with message history
- 🎯 Smart suggestion pills above input box
- 📚 Collapsible source citations with categories
- 🔍 AI-generated follow-up questions as buttons
- 🖼 CMU-branded design (red: #C41230)
- 🛡️ Mobile-responsive layout
- ⚡ Fast, smooth animations

Backend Features

-  Strict RAG pipeline (no hallucination)
-  Vector search with Pinecone
-  OpenAI GPT-4 integration
-  Structured JSON responses
-  Error handling and fallbacks
-  High-performance async API
-  Automatic API documentation

Knowledge Base

- 8 vectors indexed covering:
 - Master's degree programs (MSIT, MSECE, MSEAI)
 - Library hours and facilities
 - Shuttle bus services and routes
 - Housing options (on-campus and off-campus)
 - Campus events and student activities
 - Administration contact information
-

Security Notes

API Keys

- API keys are stored in `.env` files (not committed to git)
- **Never** share or commit `.env` files
- Rotate keys regularly for production use

Production Deployment

For production, consider:

- Using environment variables instead of `.env` files
 - Setting up HTTPS with SSL certificates
 - Implementing authentication and rate limiting
 - Using production builds (`npm run build`)
 - Deploying to cloud platforms (AWS, Azure, GCP)
-

System Requirements

Minimum Requirements

- **OS:** Linux, macOS, or Windows with WSL
- **Python:** 3.8 or higher
- **Node.js:** 16 or higher
- **RAM:** 4GB minimum (8GB recommended)
- **Disk Space:** 2GB for dependencies and data

Network Requirements

- Active internet connection for:
 - OpenAI API calls
 - Pinecone vector database
 - npm package installation
-



Next Steps & Recommendations

Immediate Improvements

1. Add More Knowledge

- Expand the knowledge base with more CMU-Africa information
- Add course catalogs, faculty profiles, research areas
- Include FAQs from student services

2. User Authentication

- Implement student login system
- Personalize responses based on user profile
- Track conversation history per user

3. Analytics Dashboard

- Track most asked questions
- Monitor system performance
- Analyze user engagement

Medium-Term Enhancements

1. Advanced Features

- Multi-language support (French, Kinyarwanda)
- Voice input/output capabilities
- Document upload for question answering
- Integration with campus systems (LMS, calendar)

2. Performance Optimization

- Implement caching for common queries
- Add response streaming for faster UX
- Optimize vector search parameters

3. Testing & Quality

- Add unit tests for backend
- Add integration tests for API
- Implement E2E tests for frontend
- Set up CI/CD pipeline

Long-Term Vision

1. Production Deployment

- Deploy to cloud platform (AWS/Azure/GCP)
- Set up load balancing
- Implement auto-scaling
- Add monitoring and logging (DataDog, CloudWatch)

2. Mobile Applications

- Develop native iOS app
- Develop native Android app
- Progressive Web App (PWA)

3. Advanced AI Features

- Fine-tune models on CMU-Africa data
 - Implement multi-modal search (images, documents)
 - Add reasoning capabilities for complex queries
-

Support & Resources

Documentation

- **Main README:** README.md
- **Quick Start Guide:** QUICK_START.md
- **API Docs:** <http://localhost:8001/docs>

Useful Commands

```
# Check service status
netstat -tulnp | grep -E "8001|3000"

# View backend logs
cd backend && python main.py

# View frontend logs
cd frontend && npm start

# Test API
curl http://localhost:8001/api/health

# Restart services
./start_backend.sh # Terminal 1
./start_frontend.sh # Terminal 2
```

Common Tasks

Adding New Knowledge:

```
curl -X POST http://localhost:8001/api/index/documents \
-H "Content-Type: application/json" \
-d '[{"id": "new_doc", "title": "New Info", "content": "...", "category": "Category"}]'
```

Checking Vector Count:

```
curl http://localhost:8001/api/index/stats
```

Testing Chat:

```
curl -X POST http://localhost:8001/api/chat \
-H "Content-Type: application/json" \
-d '{"message":"Your question here"}'
```

Deployment Checklist

- [x] Backend server running on port 8001
- [x] Frontend app running on port 3000
- [x] Pinecone vector database initialized
- [x] 8 vectors indexed in knowledge base
- [x] OpenAI API integration working
- [x] Chat functionality tested and working
- [x] Source citations displaying correctly
- [x] Suggestion pills working
- [x] Follow-up questions generating
- [x] API documentation accessible
- [x] Health check endpoint responding

Conclusion

Your CMU-Africa Campus Assistant is **fully deployed and operational!**

Access it now at: <http://localhost:3000>

The system is ready to:

- Answer questions about CMU-Africa
- Provide information on programs, facilities, and services
- Assist students with campus navigation
- Offer personalized suggestions and follow-ups

Enjoy your AI-powered campus assistant! 🚀

Version: 1.0.0

Deployment Date: October 15, 2025

Status:  Production Ready (Local)