🚀 Quick Start Guide

Prerequisites Check

Before starting, ensure you have:
- ✅ Python 3.8+ installed (`python3 --version`)
- ✅ Node.js 16+ installed (`node --version`)
- ✅ npm installed (`npm --version`)
- ✅ OpenAI API key (from https://platform.openai.com/api-keys)
- ✅ Pinecone API key (from https://www.pinecone.io/)

Step-by-Step Setup (5 minutes)

1️⃣ Configure API Keys

```
cd /home/ubuntu/code_artifacts/cmu-africa-campus-assistant/backend
```

Edit the `.env` file and add your API keys:

```
nano .env
# or
vim .env
```

Replace the placeholder values:

```
OPENAI_API_KEY=sk-your-actual-openai-key
PINECONE_API_KEY=your-actual-pinecone-key
PINECONE_ENVIRONMENT=us-east-1
```

Save and exit.

2️⃣ Load Sample Knowledge Base

```
cd /home/ubuntu/code_artifacts/cmu-africa-campus-assistant/backend
source venv/bin/activate
python load_knowledge_base.py
```

Expected output:

```
============================================================
CMU-Africa Campus Assistant - Knowledge Base Loader
============================================================
Loading knowledge base from: ../data/sample_knowledge_base.json
Loaded 8 documents
Initializing RAG pipeline...
RAG pipeline initialized successfully!
Indexing documents into Pinecone...
✅ Successfully indexed 8 documents!
```

3️⃣ Start Backend (Terminal 1)

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

Expected output:

```
Starting CMU-Africa Assistant Backend
Starting FastAPI server on http://localhost:8000
INFO:     Started server process
INFO:     Uvicorn running on http://0.0.0.0:8000
```

Test: Visit http://localhost:8000/api/health

4️⃣ Start Frontend (Terminal 2)

Open a **new terminal** and run:

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

Expected output:

```
Starting CMU-Africa Assistant Frontend
Starting React development server on http://localhost:3000
Compiled successfully!
```

The app will automatically open at: **http://localhost:3000** 🎉

Testing the Application

Try These Questions:

1. **“What are the shuttle bus timings?”**
   - Should return information about bus schedules
2. **“What programs does CMU-Africa offer?”**
   - Should list MSIT, MSECE, and MSEAI programs
3. **“What are the library hours?”**
   - Should return library operating hours
4. **“Tell me about housing options”**
   - Should describe on-campus and off-campus housing

Expected Features:

✅ Chat interface with message history
✅ Suggestion pills ABOVE the input box
✅ Collapsible sources below each response
✅ Follow-up questions as clickable buttons
✅ Smooth animations and transitions
✅ Responsive design (try on mobile size)

API Testing (Optional)

You can also test the API directly using curl:

```
# Health check
curl http://localhost:8000/api/health

# Send a chat message
curl -X POST http://localhost:8000/api/chat \
  -H "Content-Type: application/json" \
  -d '{
    "message": "What are the shuttle bus timings?",
    "user_profile": {"program": "MSIT", "year": 2}
  }'
```

Troubleshooting

Backend won't start?

Check 1: Are API keys configured?

```
cat backend/.env
```

Check 2: Is virtual environment activated?

```
cd backend
source venv/bin/activate
python main.py
```

Check 3: Are dependencies installed?

```
cd backend
source venv/bin/activate
pip install -r requirements.txt
```

Frontend won't start?

Check 1: Are dependencies installed?

```
cd frontend
npm install
```

Check 2: Is port 3000 available?

```
lsof -i :3000
# Kill if needed: kill -9 <PID>
```

API Connection Error in Frontend?

Check 1: Is backend running on port 8000?

```
curl http://localhost:8000/api/health
```

Check 2: Check browser console for errors
- Open DevTools (F12)
- Look at Console and Network tabs

Knowledge Base Not Loading?

Issue: "I don't have verified information about that"

Solution: Reload knowledge base

```
cd backend
source venv/bin/activate
python load_knowledge_base.py
```

Stopping the Services

Stop Backend:

Press `Ctrl+C` in the backend terminal

Stop Frontend:

Press `Ctrl+C` in the frontend terminal

Next Steps

1. ✅ Add more documents to `data/sample_knowledge_base.json`
2. ✅ Customize suggestion generation in `backend/rag_pipeline.py`
3. ✅ Modify UI colors in `frontend/tailwind.config.js`
4. ✅ Add user authentication (optional)
5. ✅ Deploy to production (AWS, Heroku, etc.)

Need Help?

- 📖 Read the full README.md (./README.md)
- 🐛 Check the troubleshooting section above
- 📧 Contact the development team

Happy Assisting! 🎓✨