HP Business Edge Project Deployment Manual

This guide provides a comprehensive, step-by-step walkthrough for deploying the HP Business Edge RAG (Retrieval-Augmented Generation) system either through Docker or by running the Python source code directly.

🛠️ Prerequisites

Before running the project, ensure you have the following tools and configurations set up on your system:

✅ General Requirements

- **Operating System: Ubuntu 22.04+, MacOS, or Windows (with Docker and WSL2 installed)**
- **Hardware Requirements:**
  - **Minimum 8 GB RAM (16 GB recommended for optimal performance)**
  - **At least 4 CPU cores**
  - **Sufficient disk space (~10 GB) for Docker images and model files**

✅ For Docker-Based Deployment

- **Docker installed and running**
  - **Download Docker Desktop**
  - **Make sure Docker Engine is active and has internet access**

✅ For Local Python Execution

- **Python 3.8+ installed and added to system PATH**
  - **Verify with:** `python3 --version`

- **pip for package management**
- **Ollama to run LLMs locally**
  - **[Install Ollama](#) based on your OS**
- **Gradio, LangChain, ChromaDB, pdfplumber, and camelot-py installed via `requirements.txt`**

✅ **Network Configuration**

- **Ensure the following ports are available:**
  - **`7860` for Gradio UI**
  - **`11434` for Ollama service**

🐳 Deployment Using Docker (Recommended)

✅ Step 1: Load or Pull the Docker Image

Choose **one** of the following methods:

- **To load the image from a `.tar` file:**

```
docker load -i business-edge.tar
```

- **To pull the image from Docker Hub:**

```
docker pull umar747/business-edge:latest
```

✅ Step 2: Run the Docker Container

Run the container and expose the necessary ports:

```
docker run -d -p 7860:7860 -p 11434:11434 umar747/business-edge:latest
```

✅ Step 3: Monitor the Container Logs

To ensure everything is initializing correctly:

```
docker ps                      # Find the Container ID
docker logs -f [CONTAINER_ID]  # Follow logs live
```

✅ Step 4: Access the Gradio Interface

Once the setup is complete, open your browser and go to:

```
http://localhost:7860
```

This will launch the Gradio UI, where you can start interacting with the RAG system.

🧪 Deployment Using Raw Python Files (Local Setup)

✅ Step 1: Install Ollama

Llama 3.1 8B is run locally via Ollama.

- **Linux**:

```
curl -fsSL https://ollama.com/install.sh | sh
```

- **Windows**: Install Ollama for Windows
- **macOS**: Install Ollama for macOS

✅ Step 2: Start the Ollama Server

```
ollama serve
```

✅ Step 3: Pull the Llama 3.1 8B Model

```
ollama pull llama3.1:8B
```

✅ Step 4: Install Python Dependencies

Make sure you're in the project root directory and run:

```
pip install -r requirements.txt
```

✅ Step 5: Run the Application

Launch the app with:

```
python app.py
```

After execution, the terminal will display a link like `http://localhost:7860`. Click or copy-paste it into your browser to begin using the interface.

📌 Notes

- **Ports**:
  - `7860`: Gradio interface (Frontend)
  - `11434`: Ollama API service
- **Container Startup Time**: Initial setup (model pulling, Ollama startup) may take several minutes on first run.

- **Model Download**: Ensure your machine has sufficient storage and RAM to download and run Llama 3.1 8B (~4GB+ model).
- **Best Practice**: Use Docker for consistent, environment-independent deployments.

For any issues or questions, contact the project maintainers:

- ali.abbas.23@ucl.ac.uk
- omar.nazir.23@ucl.ac.uk
- umar.ali.23@ucl.ac.uk