Hotel Booking AI Agent - Ubuntu Setup Guide

Prerequisites

Before starting, make sure you have:

- Ubuntu 18.04 or later
- Internet connection
- Admin/sudo privileges

Step 1: Update System and Install Python

```bash
# Update package list
sudo apt update && sudo apt upgrade -y

# Install Python 3 and pip if not already installed
sudo apt install python3 python3-pip python3-venv git curl -y

# Verify Python installation
python3 --version
pip3 --version
```

Step 2: Clone or Create Project Directory

```bash
# If you have a git repository, clone it:
# git clone <your-repo-url>
# cd <project-directory>

# Or create a new directory for the project:
mkdir hotel-booking-agent
cd hotel-booking-agent

# If you have project files, copy them here
```

Step 3: Set Up Python Virtual Environment

```bash
# Create virtual environment
python3 -m venv venv

# Activate virtual environment
source venv/bin/activate

# Your terminal prompt should now show (venv)
```

Step 4: Install Project Dependencies

```bash
# If you have requirements.txt file:
pip install -r requirements.txt

# If you don't have requirements.txt, install common dependencies:
pip install fastapi uvicorn langchain langgraph google-generativeai sqlite3 requests p
```

Step 5: Environment Configuration

```bash
# Copy environment template (if exists)
cp .env.template .env

# Or create .env file manually
nano .env
```

Add the following to your `.env` file:

```bash
GEMINI_API_KEY=your_gemini_api_key_here
INSTAGRAM_ACCESS_TOKEN=your_instagram_token
INSTAGRAM_APP_SECRET=your_app_secret
INSTAGRAM_PAGE_ID=your_page_id
```

Step 6: Get Gemini API Key

1. Open your browser and go to Google AI Studio
2. Sign in with your Google account
3. Click "Create API Key"
4. Copy the generated API key
5. Add it to your `.env` file

bash

```
# Edit .env file
nano .env
# Add: GEMINI_API_KEY=your_actual_api_key_here
```

Step 7: Create Required Project Structure

bash

```
# Create necessary directories
mkdir -p agents database data

# Create basic files if they don't exist
touch main.py test_agent.py
touch agents/hotel_agent.py
touch database/db_manager.py
touch data/hotel_data.json
```

Step 8: Test the Setup

bash

```
# Make sure virtual environment is active
source venv/bin/activate

# Test the agent (if test file exists)
python test_agent.py

# Or test Python imports
python3 -c "import fastapi, langchain; print('Dependencies installed successfully!')"
```

Step 9: Run the Server

bash

```
# Start the FastAPI server
python main.py

# Or if using uvicorn directly:
uvicorn main:app --host 0.0.0.0 --port 8000 --reload
```

The server should start and show:

```
INFO:     Uvicorn running on http://0.0.0.0:8000
```

Step 10: Test the API

Open a new terminal and test:

bash

```bash
# Test the chat endpoint
curl -X POST "http://localhost:8000/chat" \
  -H "Content-Type: application/json" \
  -d '{"user_id": "test_user", "message": "I want to book a room"}'
```

Step 11: Instagram Integration Setup (Optional)

Install ngrok for webhook testing:

bash

```bash
# Download and install ngrok
curl -s https://ngrok-agent.s3.amazonaws.com/ngrok.asc | sudo tee /etc/apt/trusted.gpg
echo "deb https://ngrok-agent.s3.amazonaws.com buster main" | sudo tee /etc/apt/source
sudo apt update && sudo apt install ngrok

# Authenticate ngrok (sign up at ngrok.com for free token)
ngrok config add-authtoken YOUR_NGROK_TOKEN

# Expose your local server
ngrok http 8000
```

Configure Facebook Developer App:

1. Go to Facebook Developers
2. Create new app
3. Add Instagram Basic Display product
4. Set webhook URL to your ngrok URL: `https://your-ngrok-url.ngrok.io/webhook`

Step 12: Database Setup

bash

```bash
# The SQLite database will be created automatically when you first run the app
# Verify database creation
ls -la *.db

# If you need to reset the database:
rm -f hotel_bookings.db
```

Troubleshooting

Common Issues and Solutions:

1. Permission Denied Errors:

bash

```
# Fix Python/pip permissions
sudo chown -R $USER:$USER ~/.local
```

2. Port Already in Use:

bash

```
# Find process using port 8000
sudo lsof -i :8000
# Kill the process
sudo kill -9 <PID>
```

3. Virtual Environment Issues:

bash

```
# Deactivate and recreate virtual environment
deactivate
rm -rf venv
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
```

4. Gemini API Key Issues:

bash

```
# Verify .env file is loaded
python3 -c "import os; from dotenv import load_dotenv; load_dotenv(); print(os.getenv(
```

Project Structure

Your final project structure should look like:

```
hotel-booking-agent/
├── venv/
├── agents/
│   └── hotel_agent.py
├── database/
│   └── db_manager.py
├── data/
│   └── hotel_data.json
├── .env
├── .env.template
├── main.py
├── test_agent.py
├── requirements.txt
└── hotel_bookings.db (created automatically)
```

Next Steps

1. **Test all endpoints** using the provided curl commands
2. **Customize hotel data** in `data/hotel_data.json`
3. **Set up Instagram integration** if needed
4. **Deploy to production** using Heroku, AWS, or other cloud platforms

Keeping the Server Running

For production, use a process manager:

bash

```bash
# Install PM2
sudo npm install -g pm2

# Start server with PM2
pm2 start "python main.py" --name hotel-agent

# Make PM2 start on boot
pm2 startup
pm2 save
```

The Hotel Booking AI Agent should now be running successfully on your Ubuntu system!