CSI3007 - ADVANCED PYTHON PROGRAMMING

LAB ACTIVITY – 20

Local Containerization (Docker Desktop)

NAME : SONALI R

REG NO : 22MID0061

Goal: To containerize a Python Flask web application (the chatbot App) into a portable Docker image, ensuring a consistent execution environment across all local systems.

Project Overview

Title – SERVICE MIND AI

Aim: ServiceMind-AI is an intelligent customer chatbot application designed to assist users with various services by handling multiple types of customer inquiries (Service Inquiry, Cancel Request, Reschedule Request, Working Hours Inquiry, Pricing Inquiry, Greeting Response) and providing relevant information in real time.

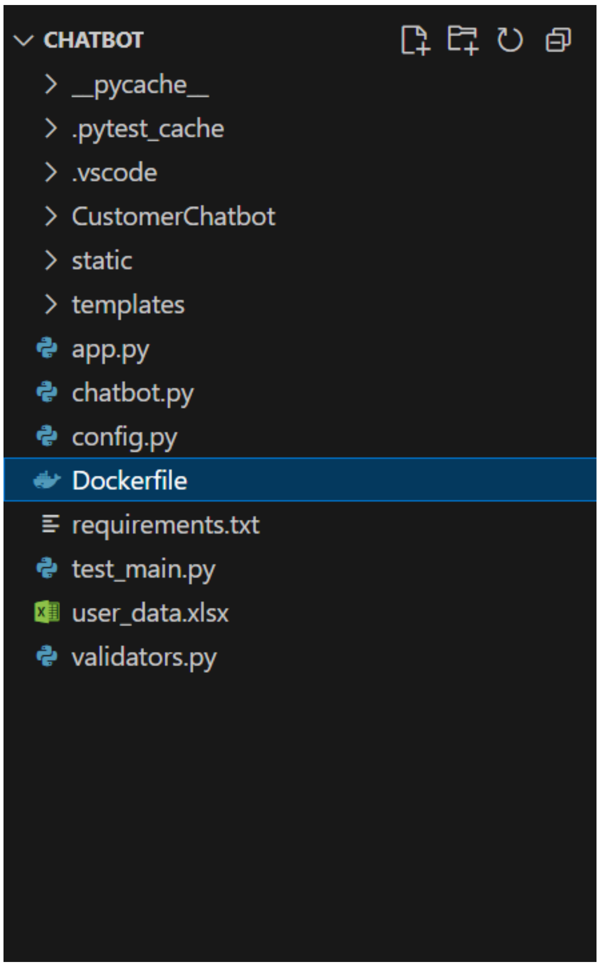
Features:

- Automated Customer Support – Provides real-time responses to user queries.
- Web-Based Interface – Interacts with users through a front-end interface.
- User Data Management – Stores and processes customer queries securely.
- Validation Mechanisms – Ensures valid inputs using custom validation scripts.
- Modular Architecture – Well-structured code for scalability and maintainability.

Project Structure:

```
CHATBOT/
├── templates/
    │-- chatbot.html     # HTML file for the chatbot interface
│-- chatbot.py          # Main Python script for chatbot logic
│-- config.py           # Configuration file for chatbot settings
│-- main.py             # Entry point for running the chatbot application
│-- test_main.py        # Unit tests for chatbot functionalities
│-- validators.py       # Python script for input validation
```

```
| -- requirements.txt      # List of dependencies for the project
|
├── CustomerChatbot/      # Virtual environment and dependencies
|
├── static/
    ├── Images/            # Directory containing image assets
    | -- chatbotScript.js     # JavaScript file for chatbot interactions
    | -- chatbotStyles.css    # CSS file for chatbot styling
|
└── user_data.xlsx       # Excel file containing user data
```


```
CHATBOT
  > __pycache__
  > .pytest_cache
  > .vscode
  > CustomerChatbot
  > static
  > templates
  app.py
  chatbot.py
  config.py
  Dockerfile
  requirements.txt
  test_main.py
  user_data.xlsx
  validators.py
```

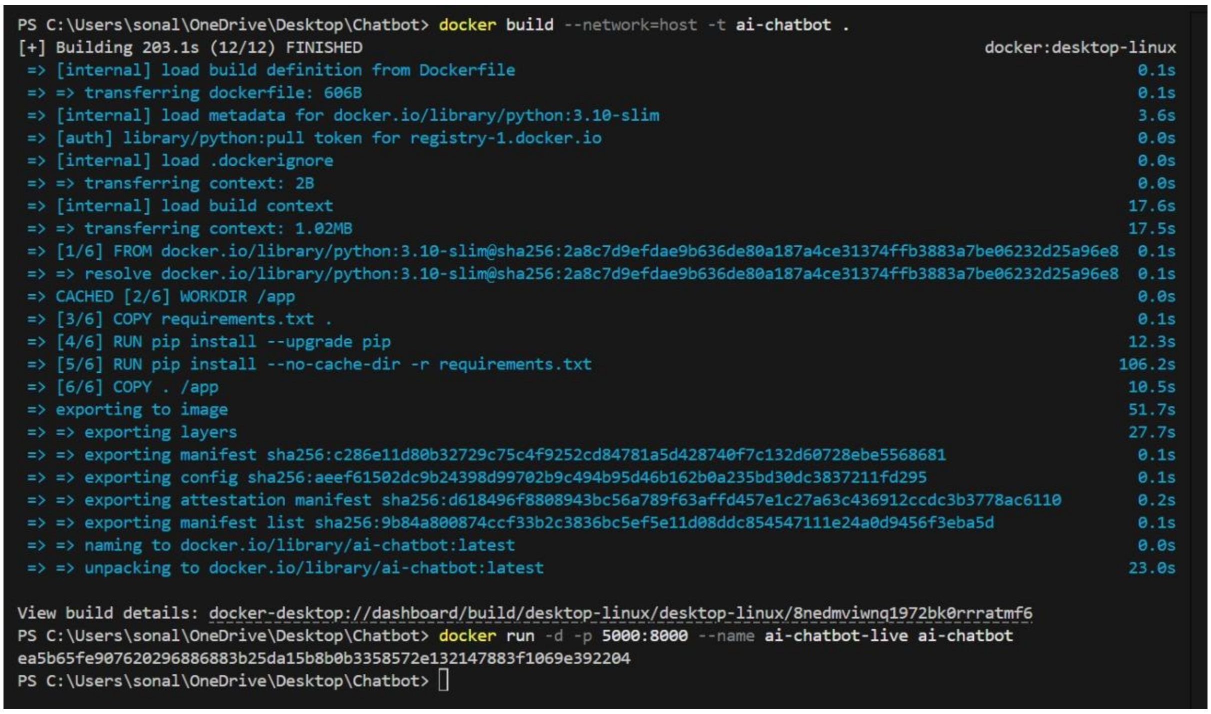
Github Link for Project: https://github.com/SonaliRajaram/ServiceMind-AI

Core Component: Dockerfile

```
FROM python:3.10-slim
# Prevent Python buffering
ENV PYTHONDONTWRITEBYTECODE=1
ENV PYTHONUNBUFFERED=1

# Set working directory
WORKDIR /app

# Copy and install dependencies
COPY requirements.txt .
RUN pip install --upgrade pip
RUN pip install --no-cache-dir -r requirements.txt

# Copy the project files
COPY . /app

# Expose Flask port
EXPOSE 8000

# Environment variables
ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0
ENV OLLAMA_HOST=http://host.docker.internal:11434

# Run Flask
CMD ["flask", "run", "--host=0.0.0.0", "--port=8000"]
```

Build Process:

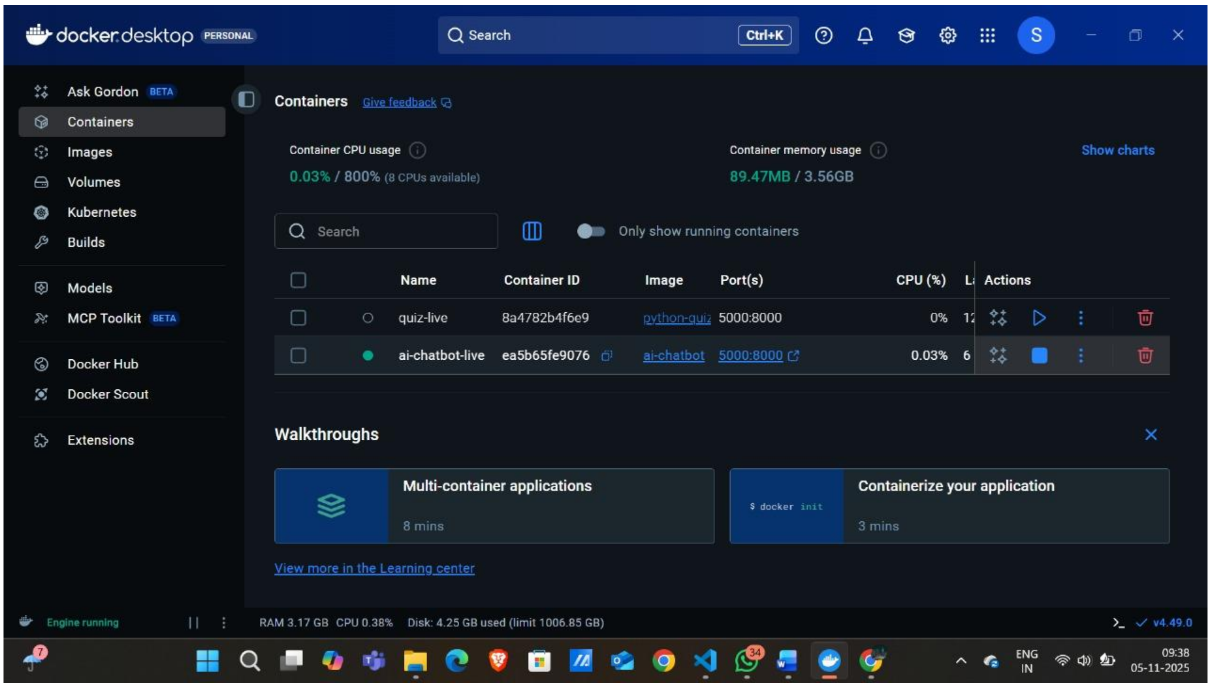
- To build Docker Desktop: docker build --network=host -t ai-chatbot .
- To deploy and run Docker Desktop: docker run -d -p 5000:8000 --name ai-chatbot-live ai-chatbot



```
PS C:\Users\sonal\OneDrive\Desktop\Chatbot> docker build --network=host -t ai-chatbot .
[+] Building 203.1s (12/12) FINISHED                                                          docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                           0.1s
 => => transferring dockerfile: 606B                                                                           0.1s
 => [internal] load metadata for docker.io/library/python:3.10-slim                                            3.6s
 => [auth] library/python:pull token for registry-1.docker.io                                                  0.0s
 => [internal] load .dockerignore                                                                              0.0s
 => => transferring context: 2B                                                                                0.0s
 => [internal] load build context                                                                             17.6s
 => => transferring context: 1.02MB                                                                           17.5s
 => [1/6] FROM docker.io/library/python:3.10-slim@sha256:2a8c7d9efdae9b636de80a187a4ce31374ffb3883a7be06232d25a96e8  0.1s
 => => resolve docker.io/library/python:3.10-slim@sha256:2a8c7d9efdae9b636de80a187a4ce31374ffb3883a7be06232d25a96e8  0.1s
 => CACHED [2/6] WORKDIR /app                                                                                  0.0s
 => [3/6] COPY requirements.txt .                                                                              0.1s
 => [4/6] RUN pip install --upgrade pip                                                                       12.3s
 => [5/6] RUN pip install --no-cache-dir -r requirements.txt                                                 106.2s
 => [6/6] COPY . /app                                                                                         10.5s
 => exporting to image                                                                                        51.7s
 => => exporting layers                                                                                       27.7s
 => => exporting manifest sha256:c286e11d80b32729c75c4f9252cd84781a5d428740f7c132d60728ebe5568681              0.1s
 => => exporting config sha256:aeef61502dc9b24398d99702b9c494b95d46b162b0a235bd30dc3837211fd295               0.1s
 => => exporting attestation manifest sha256:d618496f8808943bc56a789f63affd457e1c27a63c436912ccdc3b3778ac6110  0.2s
 => => exporting manifest list sha256:9b84a800874ccf33b2c3836bc5ef5e11d08ddc854547111e24a0d9456f3eba5d         0.1s
 => => naming to docker.io/library/ai-chatbot:latest                                                           0.0s
 => => unpacking to docker.io/library/ai-chatbot:latest                                                       23.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/8nedmviwnq1972bk0rratmf6
PS C:\Users\sonal\OneDrive\Desktop\Chatbot> docker run -d -p 5000:8000 --name ai-chatbot-live ai-chatbot
ea5b65fe907620296886883b25da15b8b0b3358572e132147883f1069e392204
PS C:\Users\sonal\OneDrive\Desktop\Chatbot> 
```

Troubleshooting challenge (DNS/Network failure) and the final solution used (docker build --network=host)

- Docker Desktop Container is Created for ai-chatbot:



Containers

Container CPU usage
0.03% / 800% (8 CPUs available)

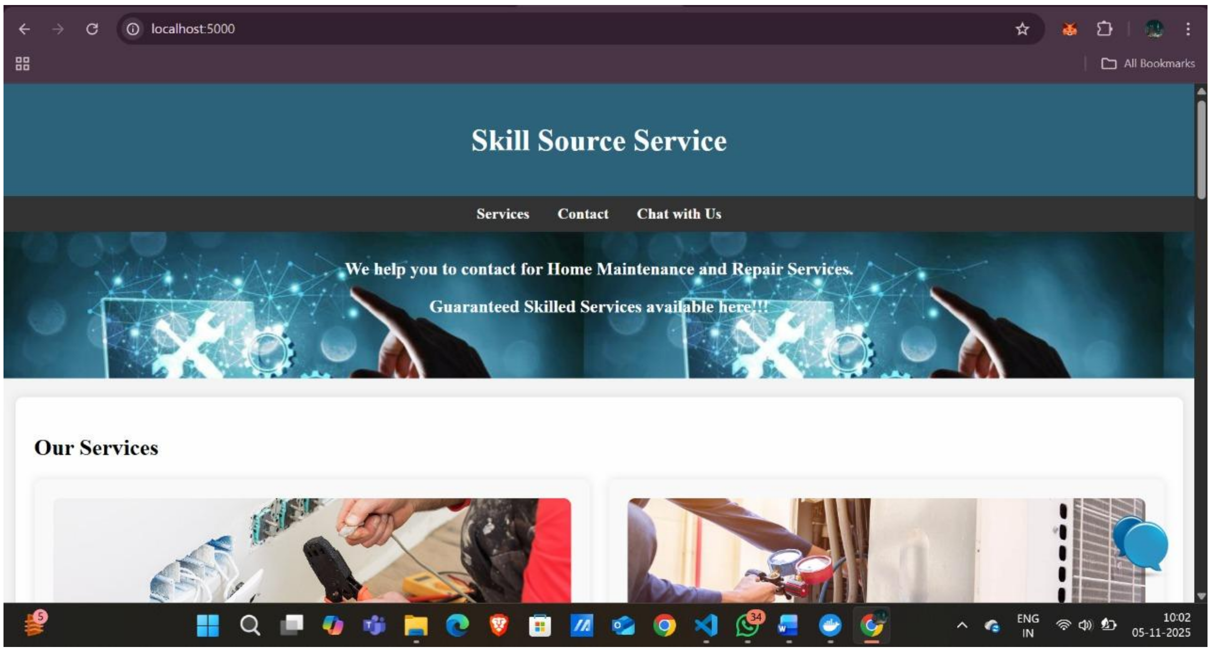
Container memory usage
89.47MB / 3.56GB

Name	Container ID	Image	Port(s)	CPU (%)	L	Actions
quiz-live	8a4782b4f6e9	python-quiz	5000:8000	0%	12	
ai-chatbot-live	ea5b65fe9076	ai-chatbot	5000:8000	0.03%	6	

RAM 3.17 GB CPU 0.38% Disk: 4.25 GB used (limit 1006.85 GB)

- Verify by accessing http://localhost:5000

Skill Source Service website and chatbot conversation snapshots:



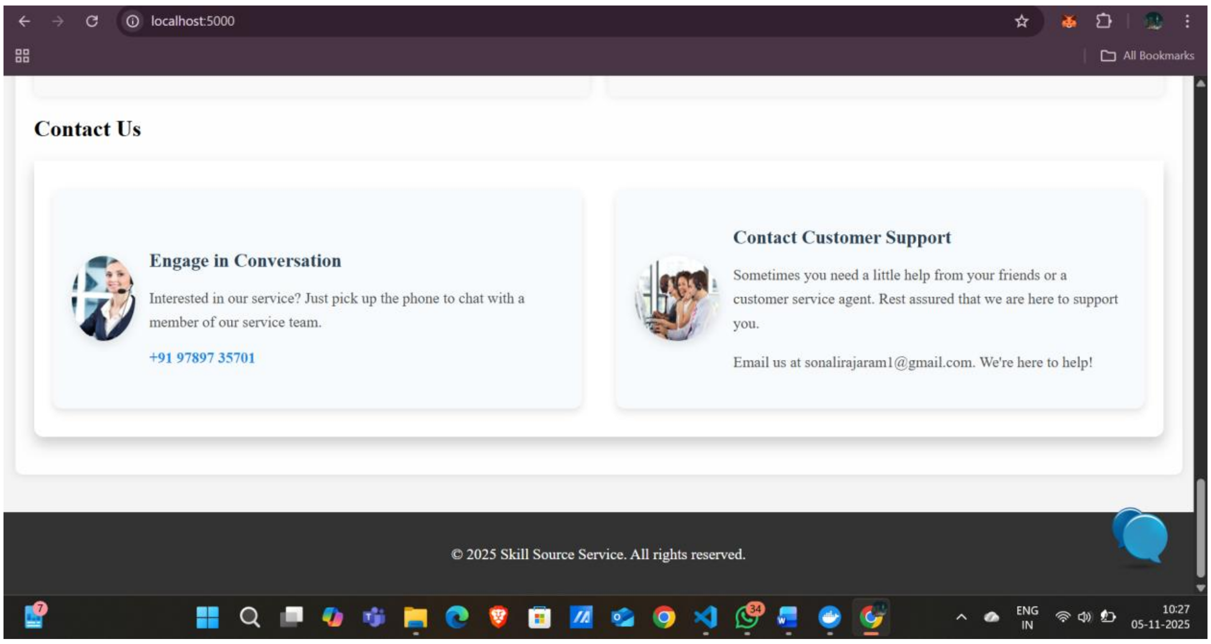
Skill Source Service

Services Contact Chat with Us

We help you to contact for Home Maintenance and Repair Services.

Guaranteed Skilled Services available here!!!

Our Services



Contact Us

Engage in Conversation

Interested in our service? Just pick up the phone to chat with a member of our service team.

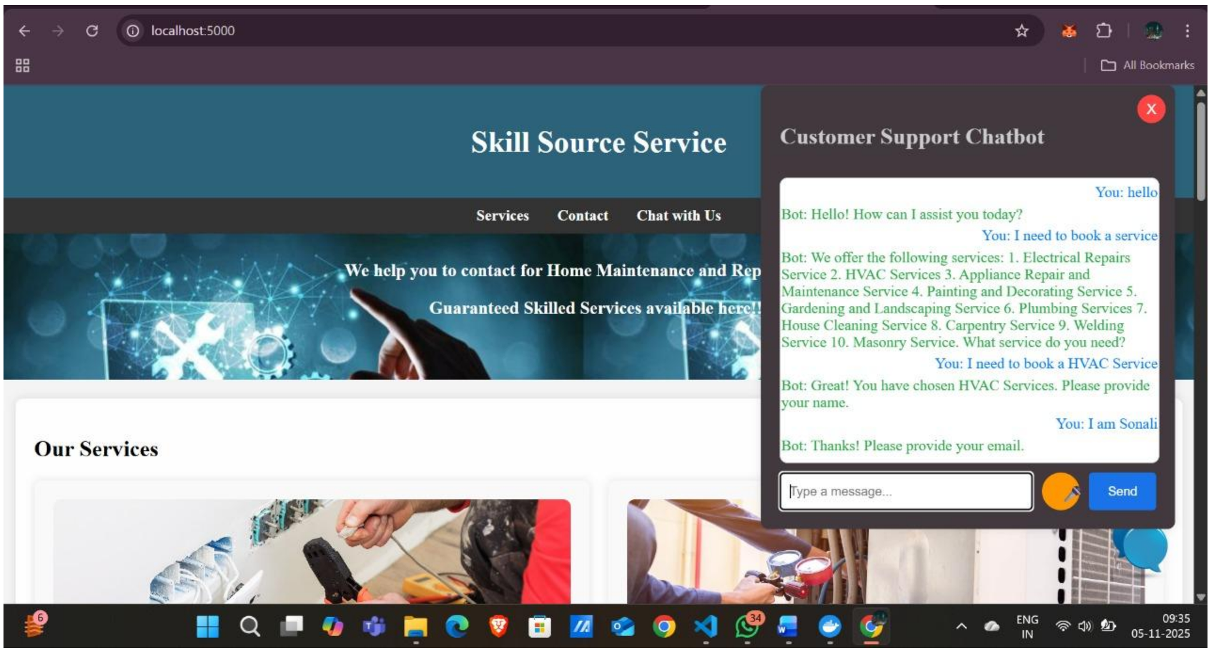
+91 97897 35701

Contact Customer Support

Sometimes you need a little help from your friends or a customer service agent. Rest assured that we are here to support you.

Email us at sonalirajaram1@gmail.com. We're here to help!

© 2025 Skill Source Service. All rights reserved.



You: hello

Bot: Hello! How can I assist you today?

You: I need to book a service

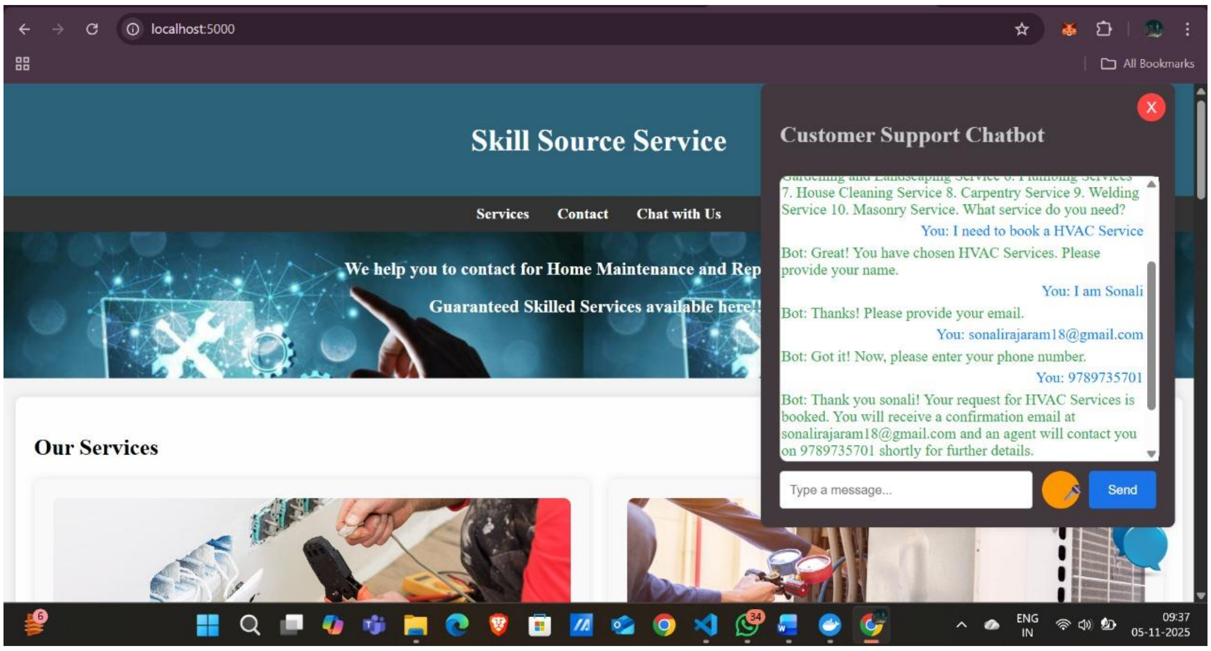
Bot: We offer the following services: 1. Electrical Repairs Service 2. HVAC Services 3. Appliance Repair and Maintenance Service 4. Painting and Decorating Service 5. Gardening and Landscaping Service 6. Plumbing Services 7. House Cleaning Service 8. Carpentry Service 9. Welding Service 10. Masonry Service. What service do you need?

You: I need to book a HVAC Service

Bot: Great! You have chosen HVAC Services. Please provide your name.

You: I am Sonali

Bot: Thanks! Please provide your email.



7. House Cleaning Service 8. Carpentry Service 9. Welding Service 10. Masonry Service. What service do you need?

You: I need to book a HVAC Service

Bot: Great! You have chosen HVAC Services. Please provide your name.

You: I am Sonali

Bot: Thanks! Please provide your email.

You: sonalirajaram18@gmail.com

Bot: Got it! Now, please enter your phone number.

You: 9789735701

Bot: Thank you sonali! Your request for HVAC Services is booked. You will receive a confirmation email at sonalirajaram18@gmail.com and an agent will contact you on 9789735701 shortly for further details.

Conclusion

The process of containerizing the AI Chatbot App using Docker successfully achieved the primary aims of establishing environmental consistency and application portability for the local development cycle.

Environmental Consistency (Solving "Works on my machine")

By defining the application environment within the Dockerfile, we locked in the exact operating system, Python version (python:3.10-slim), and the specific version of the Flask library (requirements.txt).

- **Result:** This eliminated the common dependency conflicts inherent in local development. Regardless of whether the host machine ran Windows, macOS, or Linux, the code executed inside the container's isolated environment was guaranteed to be identical, ensuring the application behaved consistently.

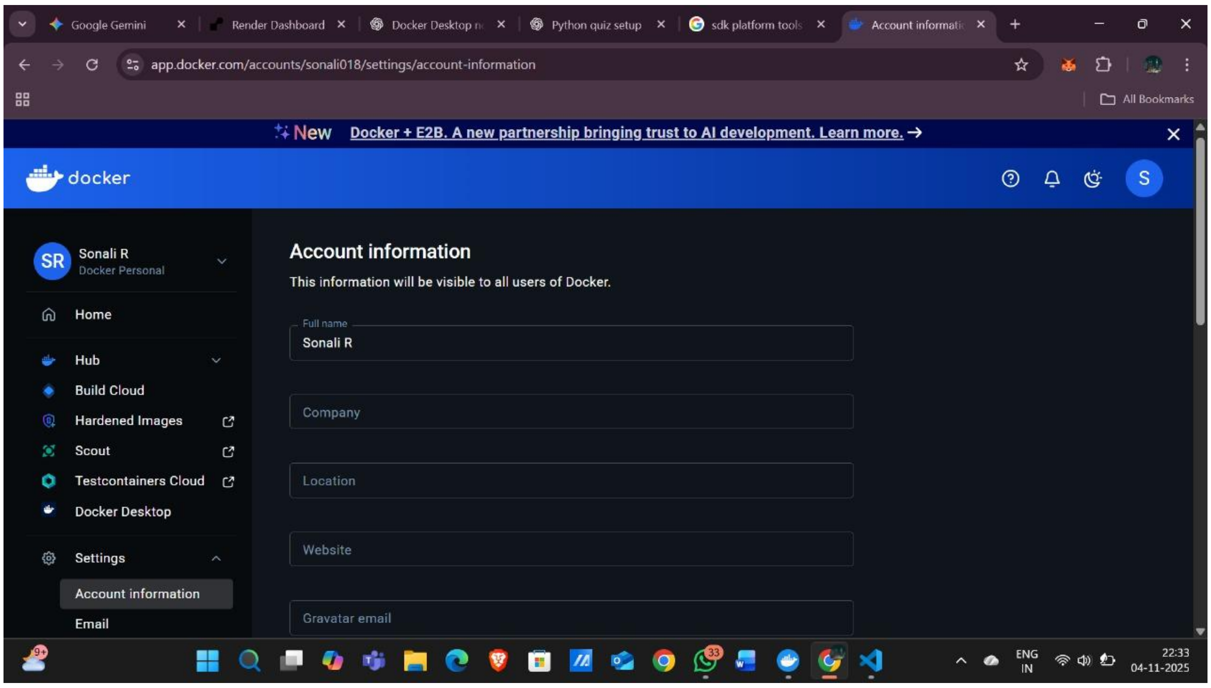
Application Portability (Ease of Transfer)

The successful creation of the Docker image consolidated all application code, dependencies, and configuration into a single, immutable package.

- **Result:** This single artifact, the Docker image, could be seamlessly transferred and run on any machine with Docker installed (the **"Build once, run anywhere"** principle). This portability was verified by the initial local deployment on Docker Desktop, which prepared the application for the next stage of public cloud deployment.

In summary, Docker did not just run the application; it packaged the entire environment, transforming the Flask app into a robust, easily shared, and perfectly isolated component ready for the full DevOps pipeline.

Docker Desktop Snapshots:



app.docker.com/accounts/sonali018/settings/account-information

New Docker + E2B. A new partnership bringing trust to AI development. Learn more. →

Account information

This information will be visible to all users of Docker.

Full name: Sonali R

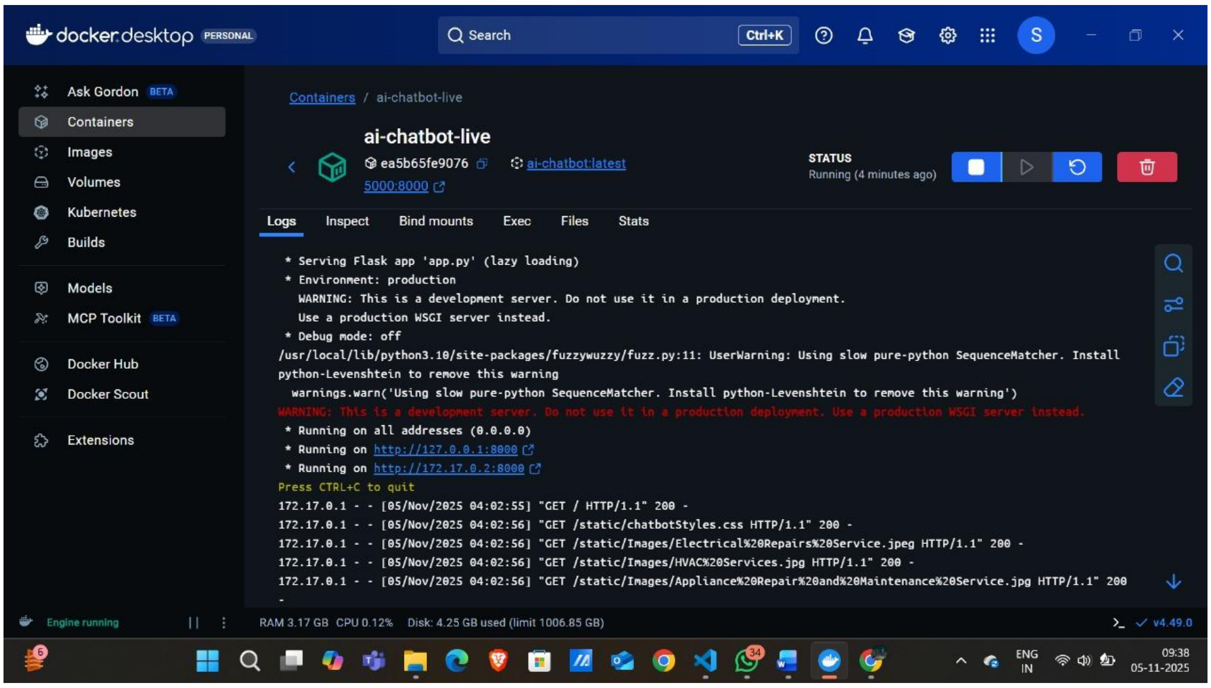
Company

Location

Website

Gravatar email

Ai-chatbot-live container in Docker Desktop:



Containers / ai-chatbot-live

ai-chatbot-live
ea5b65fe9076 ai-chatbot:latest
5000:8000

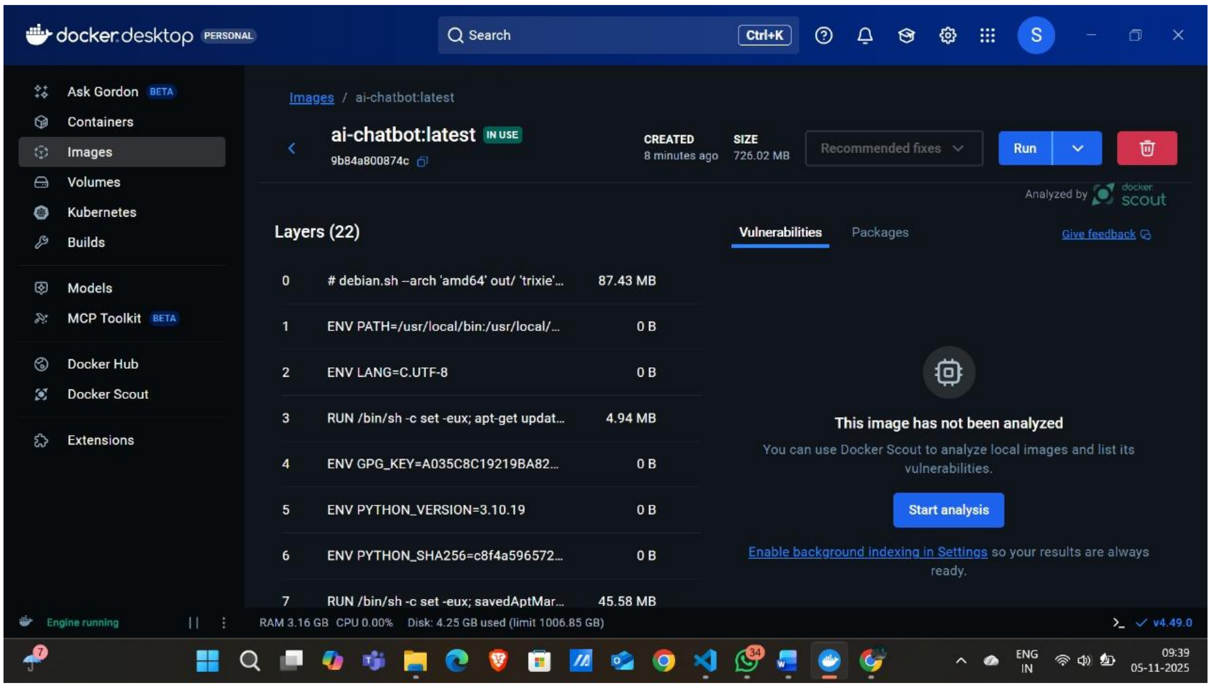
STATUS
Running (4 minutes ago)

```
 * Serving Flask app 'app.py' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
/usr/local/lib/python3.10/site-packages/fuzzywuzzy/fuzz.py:11: UserWarning: Using slow pure-python SequenceMatcher. Install
python-Levenshtein to remove this warning
  warnings.warn('Using slow pure-python SequenceMatcher. Install python-Levenshtein to remove this warning')
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:8000
 * Running on http://172.17.0.2:8000
Press CTRL+C to quit
172.17.0.1 - - [05/Nov/2025 04:02:55] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [05/Nov/2025 04:02:56] "GET /static/chatbotStyles.css HTTP/1.1" 200 -
172.17.0.1 - - [05/Nov/2025 04:02:56] "GET /static/Images/Electrical%20Repairs%20Service.jpeg HTTP/1.1" 200 -
172.17.0.1 - - [05/Nov/2025 04:02:56] "GET /static/Images/HVAC%20Services.jpg HTTP/1.1" 200 -
172.17.0.1 - - [05/Nov/2025 04:02:56] "GET /static/Images/Appliance%20Repair%20and%20Maintenance%20Service.jpg HTTP/1.1" 200
-
```

Engine running   RAM 3.17 GB CPU 0.12% Disk: 4.25 GB used (limit 1006.85 GB)   v4.49.0

ENG IN 09:38 05-11-2025

Ai-chatbot Image in Docker Desktop:



Images / ai-chatbot:latest

ai-chatbot:latest IN USE
9b84a800874c

CREATED
8 minutes ago

SIZE
726.02 MB

Layers (22)

0	# debian.sh --arch 'amd64' out/ 'trixie'...	87.43 MB
1	ENV PATH=/usr/local/bin:/usr/local/...	0 B
2	ENV LANG=C.UTF-8	0 B
3	RUN /bin/sh -c set -eux; apt-get updat...	4.94 MB
4	ENV GPG_KEY=A035C8C19219BA82...	0 B
5	ENV PYTHON_VERSION=3.10.19	0 B
6	ENV PYTHON_SHA256=c8f4a596572...	0 B
7	RUN /bin/sh -c set -eux; savedAptMar...	45.58 MB

This image has not been analyzed

You can use Docker Scout to analyze local images and list its vulnerabilities.

Enable background indexing in Settings so your results are always ready.

Engine running   RAM 3.16 GB CPU 0.00% Disk: 4.25 GB used (limit 1006.85 GB)   v4.49.0

ENG IN 09:39 05-11-2025

****