# CSI3007 - ADVANCED PYTHON PROGRAMMING

## LAB ACTIVITY – 20

## Local Containerization (Docker Desktop)

**GOKULA J**

**22MID0127**

**Goal:** To containerize a **Streamlit-based Blockchain Authentication App** into a portable Docker image, ensuring consistent execution across all systems and enabling easy cloud or local deployment.

**Project Overview**

Title-**Blockchain-Based Password Strength Classifier Using MD5**

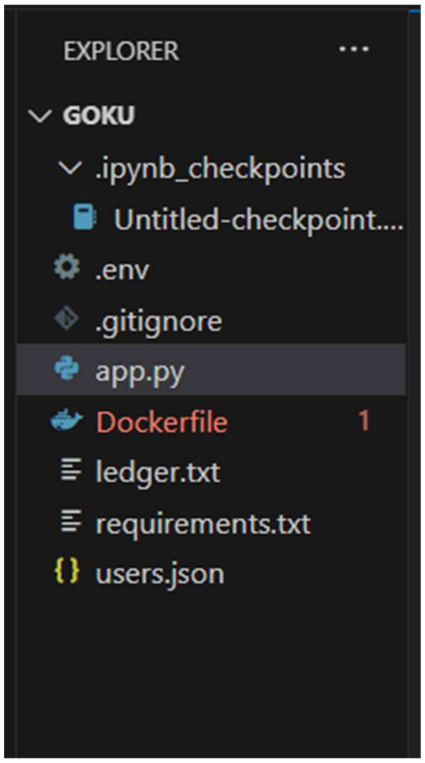
**Aim:** The aim of this project is to design a secure, blockchain-integrated password authentication system that classifies password strength, validates user credentials, and securely stores hashed passwords on a blockchain ledger.

**Features:**

- **User Registration & Login** – Allows users to register and authenticate securely.
- **Password Strength Classification** – Checks passwords against length, case, digit, and special character rules.
- **Blockchain Ledger** – Stores user data and hashed passwords as immutable blocks.
- **MD5 Hashing Algorithm** – Converts plain-text passwords into irreversible hashes for security.
- **Secret Key Access** – Protects the ledger download feature via a developer-controlled secret key.
- **Deployable Web Interface** – Built using Streamlit, easily deployable locally or on cloud platforms.

**Project Structure:**

```
BLOCKCHAIN_AUTH_APP/
|
├── app.py            # Main Streamlit application
├── Dockerfile         # Docker configuration
├── requirements.txt     # Python dependencies
├── .env              # Secret key environment variable
├── .gitignore          # Ignored files (ledger, secrets)
├── ledger.txt          # Blockchain ledger (auto-generated)
└── users.json          # User database (auto-generated)
```



**Core Component: Dockerfile**

```
# -----------------------------
# Dockerfile for Blockchain Auth App
# -----------------------------

# Use official lightweight Python image
FROM python:3.10-slim

# Set working directory
WORKDIR /app

# Copy all project files into container
```

```
COPY . /app

# Install dependencies
RUN pip install --no-cache-dir -r requirements.txt

# Expose Streamlit default port
EXPOSE 8501

# Command to run the Streamlit app
CMD ["streamlit", "run", "app.py", "--server.port=8501", "--server.address=0.0.0.0"]
docker build -t blockchain-auth-app .
```

**Build Process:**

Step 1: Build Docker Image

```
docker build -t blockchain-auth-app .
```

Step 2: Run the Docker Container

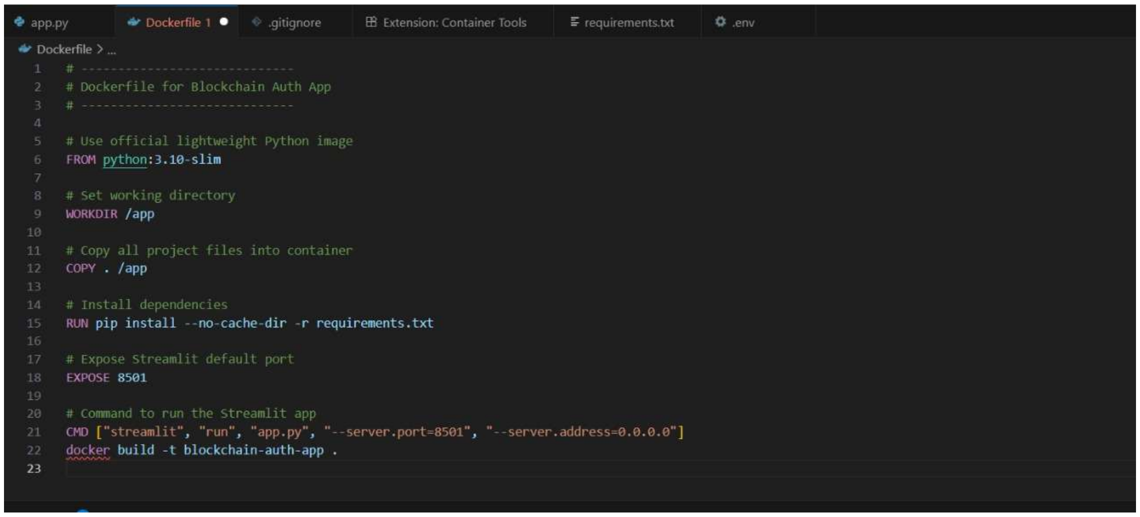
```
docker run -p 8501:8501 blockchain-auth-app
```

Step 3: Access the Application

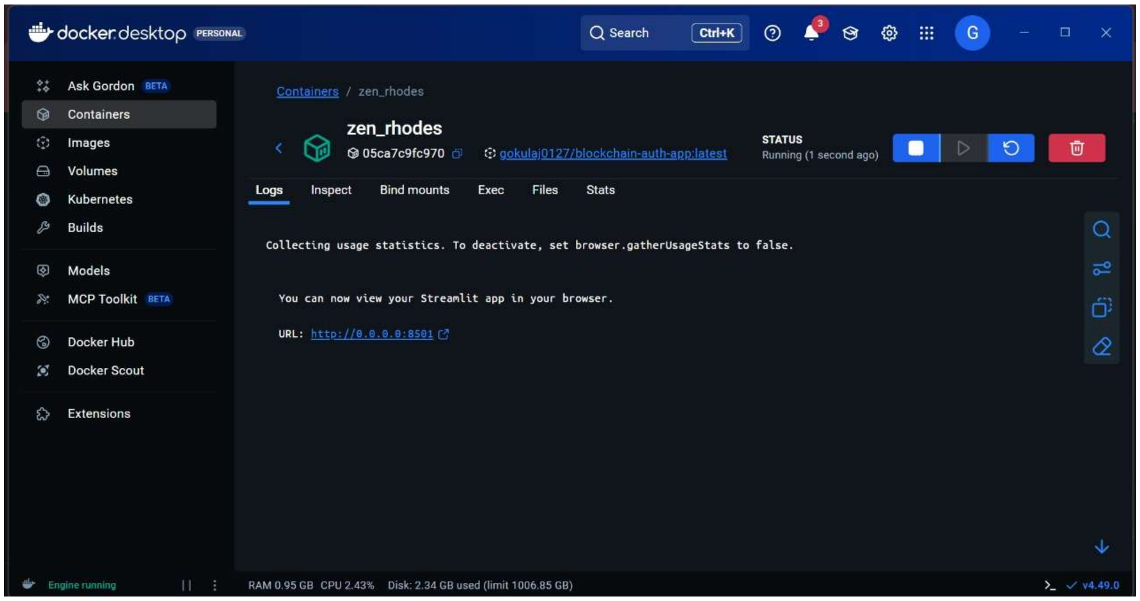
http://localhost:8501

Step 4: Push to Docker Hub

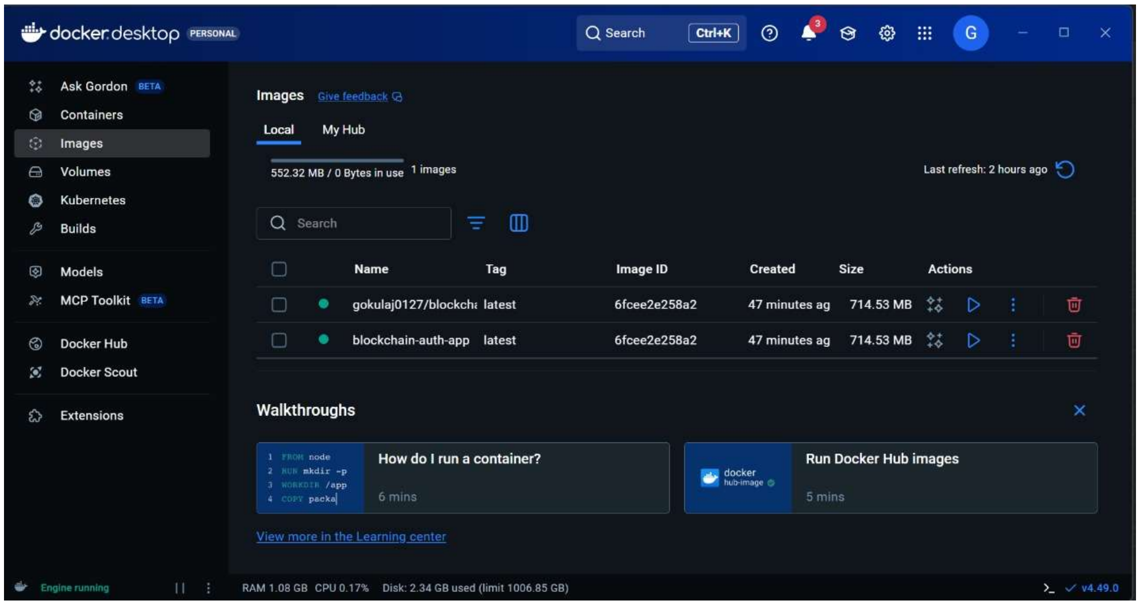
```
docker tag blockchain-auth-app gokulaj0127/blockchain-auth-app:latest
docker push gokulaj0127/blockchain-auth-app:latest
```



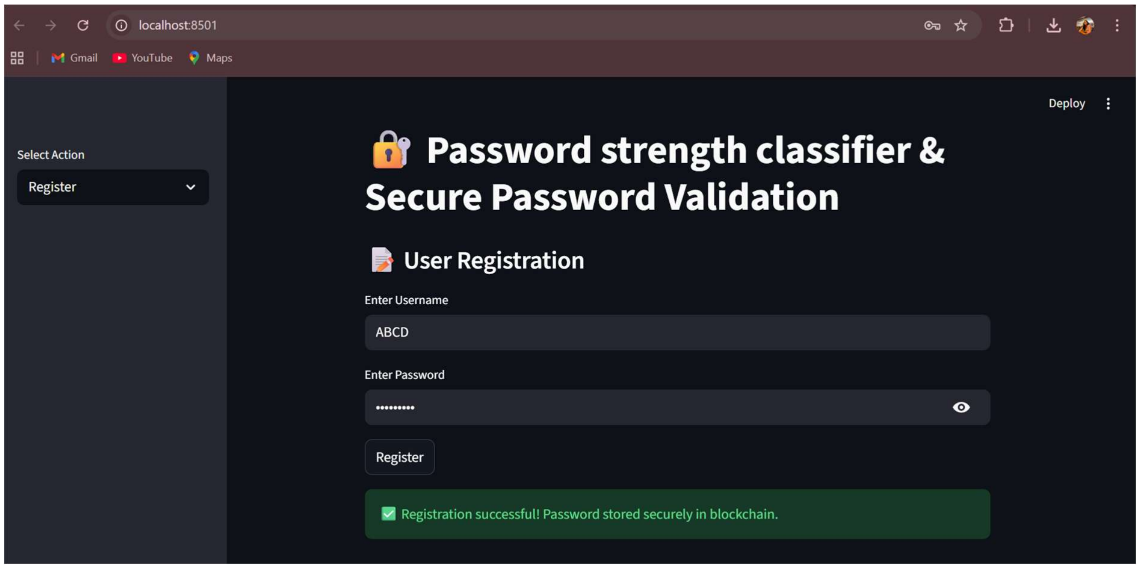
```
# ----------------------------
# Dockerfile for Blockchain Auth App
# ----------------------------

# Use official lightweight Python image
FROM python:3.10-slim

# Set working directory
WORKDIR /app

# Copy all project files into container
COPY . /app

# Install dependencies
RUN pip install --no-cache-dir -r requirements.txt

# Expose Streamlit default port
EXPOSE 8501

# Command to run the Streamlit app
CMD ["streamlit", "run", "app.py", "--server.port=8501", "--server.address=0.0.0.0"]
docker build -t blockchain-auth-app .
```



```
Collecting usage statistics. To deactivate, set browser.gatherUsageStats to false.


  You can now view your Streamlit app in your browser.

  URL: http://0.0.0.0:8501
```



| Name | Tag | Image ID | Created | Size | Actions |
|---|---|---|---|---|---|
| gokulaj0127/blockchi | latest | 6fcee2e258a2 | 47 minutes ag | 714.53 MB | |
| blockchain-auth-app | latest | 6fcee2e258a2 | 47 minutes ag | 714.53 MB | |



Select Action

Register

🔐 Password strength classifier & Secure Password Validation

📝 User Registration

Enter Username

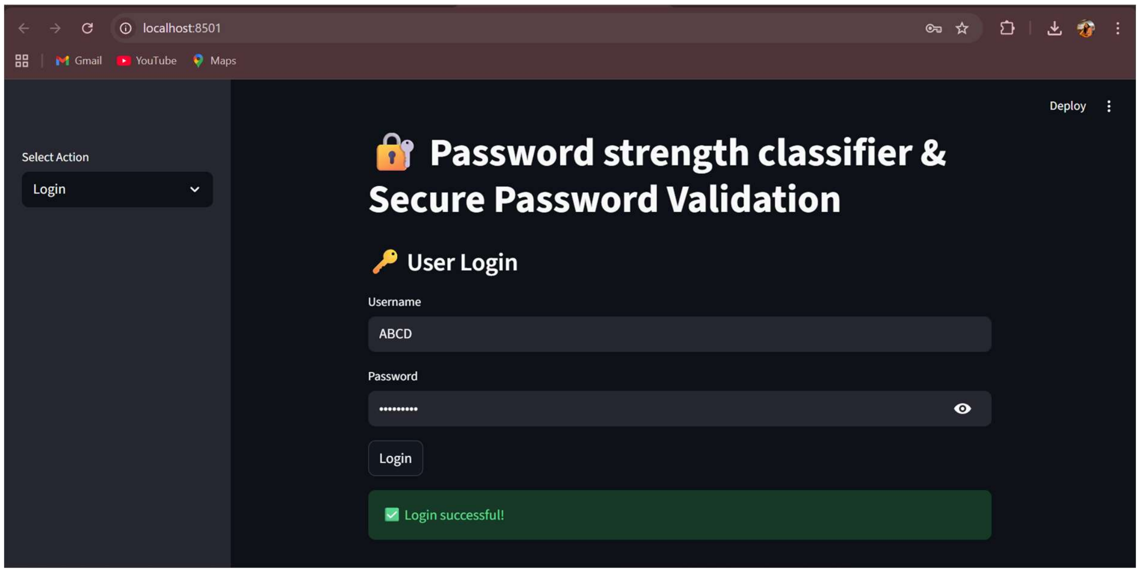
ABCD

Enter Password

•••••••••

Register

✅ Registration successful! Password stored securely in blockchain.



Select Action

Login

🔐 Password strength classifier & Secure Password Validation

🔑 User Login

Username

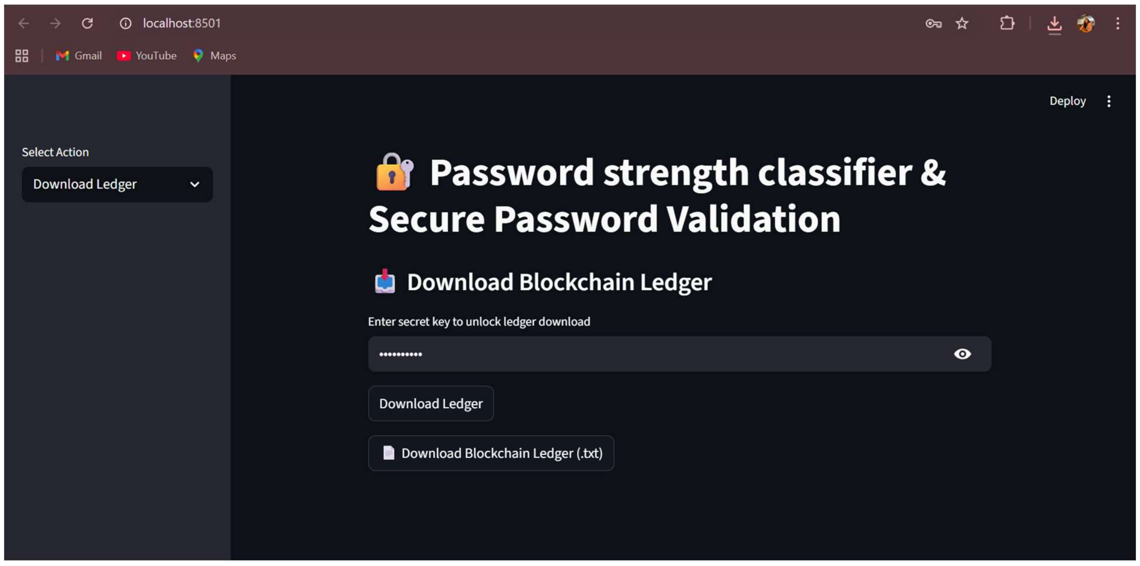
ABCD

Password

•••••••••

Login

✅ Login successful!



Select Action

Download Ledger

🔐 Password strength classifier & Secure Password Validation

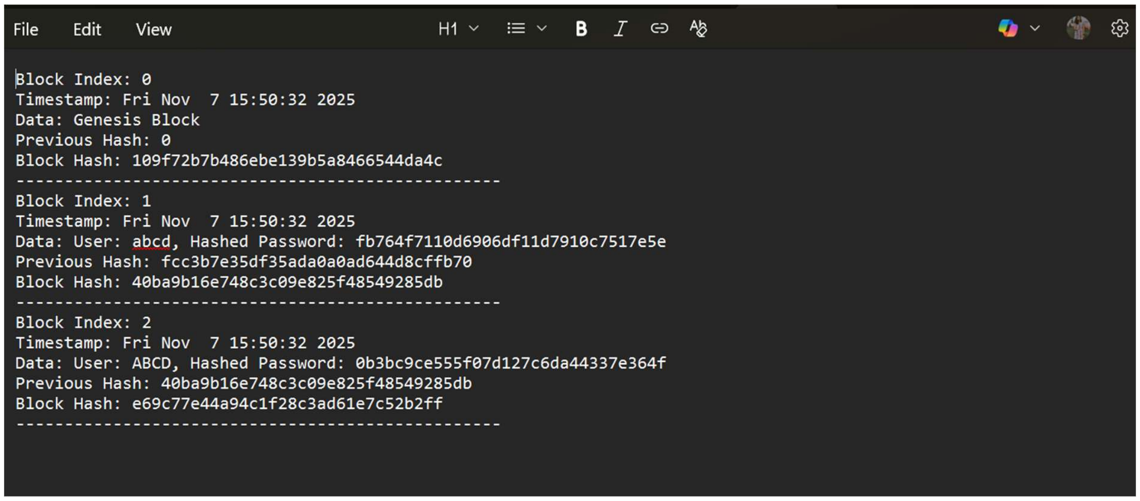
📥 Download Blockchain Ledger

Enter secret key to unlock ledger download

••••••••••

Download Ledger

📄 Download Blockchain Ledger (.txt)



```
Block Index: 0
Timestamp: Fri Nov  7 15:50:32 2025
Data: Genesis Block
Previous Hash: 0
Block Hash: 109f72b7b486ebe139b5a8466544da4c
--------------------------------------------------
Block Index: 1
Timestamp: Fri Nov  7 15:50:32 2025
Data: User: abcd, Hashed Password: fb764f7110d6906df11d7910c7517e5e
Previous Hash: fcc3b7e35df35ada0a0ad644d8cffb70
Block Hash: 40ba9b16e748c3c09e825f48549285db
--------------------------------------------------
Block Index: 2
Timestamp: Fri Nov  7 15:50:32 2025
Data: User: ABCD, Hashed Password: 0b3bc9ce555f07d127c6da44337e364f
Previous Hash: 40ba9b16e748c3c09e825f48549285db
Block Hash: e69c77e44a94c1f28c3ad61e7c52b2ff
--------------------------------------------------
```

**Conclusion:**

The process of containerizing the Blockchain Authentication App using Docker successfully achieved:

- **Environmental Consistency:**
  The Docker container locks the Python version and dependencies, ensuring the same behavior on all systems.
- **Application Portability:**
  The complete app runs identically on any machine supporting Docker, aligning with the “Build once, run anywhere” concept.
- **Security Enhancement:**
  By using **MD5 hashing** and blockchain, the system ensures password integrity, traceability, and immutability.