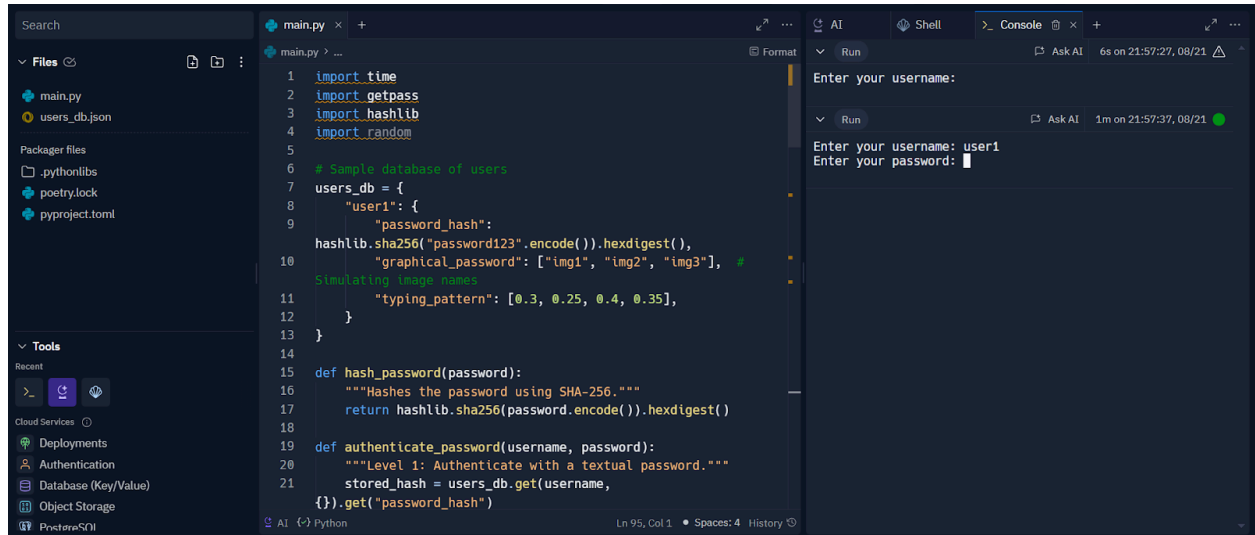


Three-Level Password System

Output:-



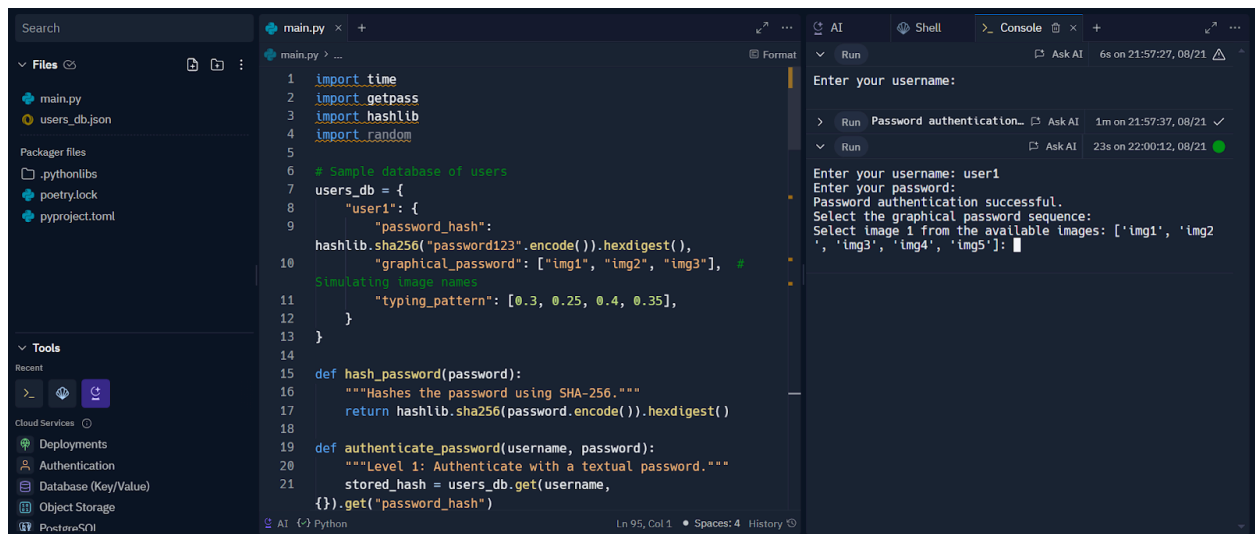
The screenshot shows a code editor with a file explorer on the left, a central code editor, and a console on the right. The file explorer shows files like main.py and users_db.json. The code editor contains a Python script for a three-level password system. The console shows the initial prompt 'Enter your username:' and the user input 'user1'.

```
1 import time
2 import getpass
3 import hashlib
4 import random
5
6 # Sample database of users
7 users_db = {
8     "user1": {
9         "password_hash":
10         hashlib.sha256("password123".encode()).hexdigest(),
11         "graphical_password": ["img1", "img2", "img3"], #
12         "typing_pattern": [0.3, 0.25, 0.4, 0.35],
13     }
14 }
15
16 def hash_password(password):
17     """Hashes the password using SHA-256."""
18     return hashlib.sha256(password.encode()).hexdigest()
19
20 def authenticate_password(username, password):
21     """Level 1: Authenticate with a textual password."""
22     stored_hash = users_db.get(username,
23     {}).get("password_hash")
```

Enter your username:

Enter your username: user1

Enter your password:



The screenshot shows the same code editor as the first screenshot, but with the second part of the Python script and the console output. The console shows the prompt 'Enter your password:' and the user input 'password123'. It then displays the message 'Password authentication successful.' and the prompt 'Select the graphical password sequence:'. The user input 'img1' is shown, and the console displays the list of available images: ['img1', 'img2', 'img3', 'img4', 'img5'].

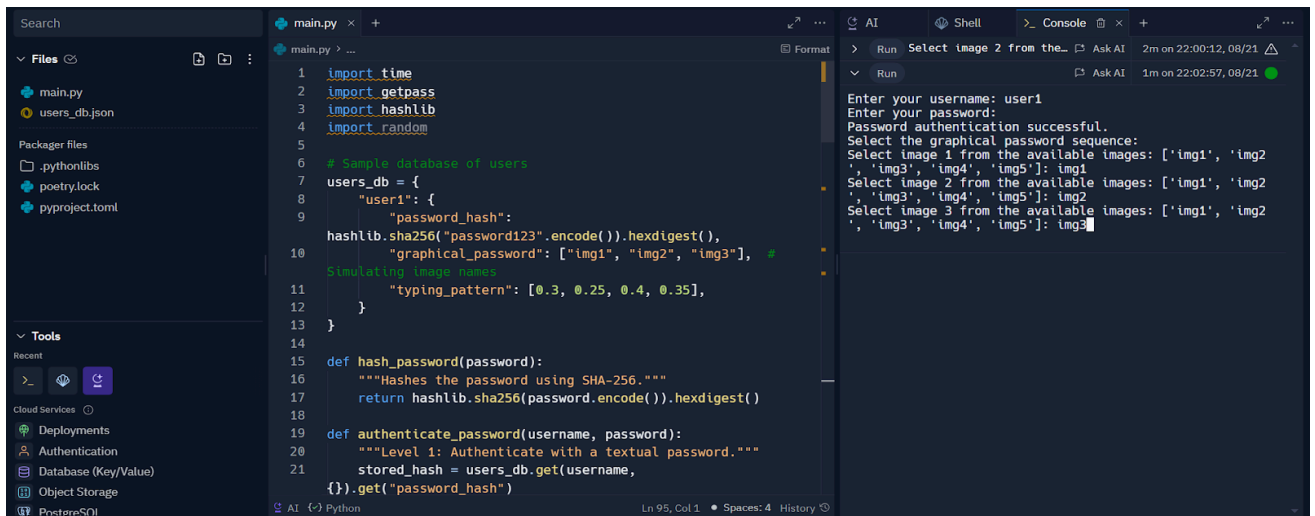
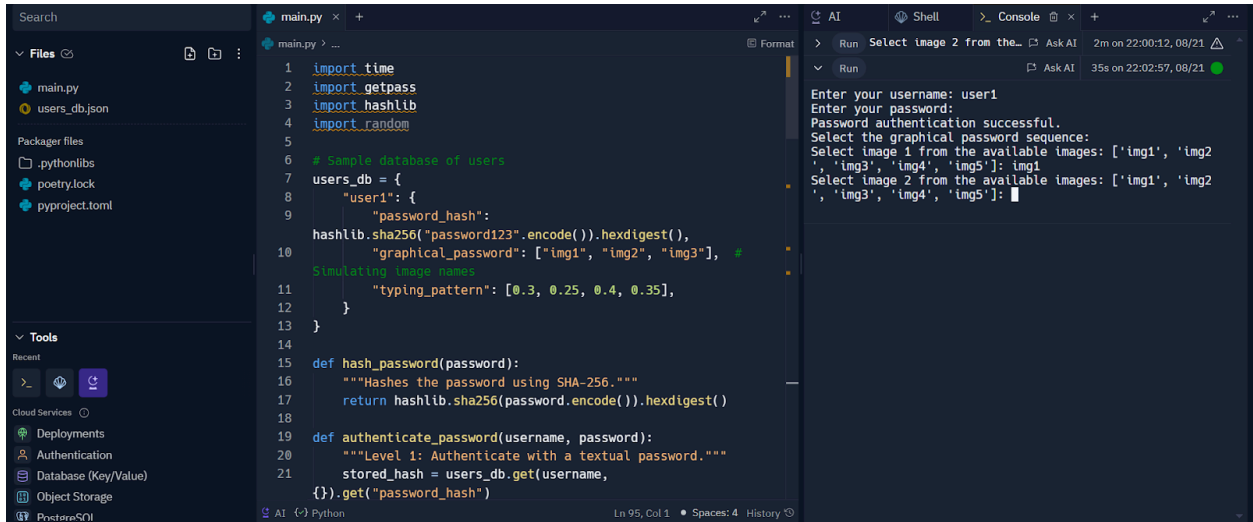
```
1 import time
2 import getpass
3 import hashlib
4 import random
5
6 # Sample database of users
7 users_db = {
8     "user1": {
9         "password_hash":
10         hashlib.sha256("password123".encode()).hexdigest(),
11         "graphical_password": ["img1", "img2", "img3"], #
12         "typing_pattern": [0.3, 0.25, 0.4, 0.35],
13     }
14 }
15
16 def hash_password(password):
17     """Hashes the password using SHA-256."""
18     return hashlib.sha256(password.encode()).hexdigest()
19
20 def authenticate_password(username, password):
21     """Level 1: Authenticate with a textual password."""
22     stored_hash = users_db.get(username,
23     {}).get("password_hash")
```

Enter your password:

Password authentication successful.

Select the graphical password sequence:

Select image 1 from the available images: ['img1', 'img2', 'img3', 'img4', 'img5']:



Search

Files

main.py

users_db.json

Packager files

.pythonlibs

poetry.lock

pyproject.toml

Tools

Recent

Cloud Services

Deployments

Authentication

Database (Key/Value)

Object Storage

PreviewSQL

main.py

main.py > ...

1 import time

2 import getpass

3 import hashlib

4 import random

5

6 # Sample database of users

7 users_db = {

8 "user1": {

9 "password_hash":

10 hashlib.sha256("password123".encode()).hexdigest(),

11 "graphical_password": ["img1", "img2", "img3"], #

12 "typing_pattern": [0.3, 0.25, 0.4, 0.35],

13 }

14 }

15

16 def hash_password(password):

17 """Hashes the password using SHA-256."""

18 return hashlib.sha256(password.encode()).hexdigest()

19

20 def authenticate_password(username, password):

21 """Level 1: Authenticate with a textual password."""

22 stored_hash = users_db.get(username,

23 {}).get("password_hash")

AI

Python

Ln 95, Col 1

Spaces: 4

History

Run

Select image 2 from the...

Ask AI

2m on 22:00:12, 08/21

Run

Ask AI

1m on 22:02:57, 08/21

Enter your username: user1

Enter your password:

Password authentication successful.

Select the graphical password sequence:

Select image 1 from the available images: ['img1', 'img2', 'img3', 'img4', 'img5']: img1

Select image 2 from the available images: ['img1', 'img2', 'img3', 'img4', 'img5']: img2

Select image 3 from the available images: ['img1', 'img2', 'img3', 'img4', 'img5']: img3

Graphical password authentication successful.

Please re-enter your password to analyze your typing pattern.

Re-enter your password:

three level password system

CPU/RAM LIMITED

Stop

Ask AI & search Ctrl K

Invite

PUBLIC

Deploy

?

IT

Search

Files

main.py

users_db.json

Packager files

.pythonlibs

poetry.lock

pyproject.toml

Tools

Recent

Cloud Services

Deployments

Authentication

Database (Key/Value)

Object Storage

PreviewSQL

main.py

main.py > ...

1 import time

2 import getpass

3 import hashlib

4 import random

5

6 # Sample database of users

7 users_db = {

8 "user1": {

9 "password_hash":

10 hashlib.sha256("password123".encode()).hexdigest(),

11 "graphical_password": ["img1", "img2", "img3"], #

12 "typing_pattern": [0.3, 0.25, 0.4, 0.35],

13 }

14 }

15

16 def hash_password(password):

17 """Hashes the password using SHA-256."""

18 return hashlib.sha256(password.encode()).hexdigest()

19

20 def authenticate_password(username, password):

21 """Level 1: Authenticate with a textual password."""

22 stored_hash = users_db.get(username,

23 {}).get("password_hash")

AI

Python

Ln 95, Col 1

Spaces: 4

History

Run

Select image 2 from the...

Ask AI

2m on 22:00:12, 08/21

Run

Ask AI

2m on 22:02:57, 08/21

Enter your username: user1

Enter your password:

Password authentication successful.

Select the graphical password sequence:

Select image 1 from the available images: ['img1', 'img2', 'img3', 'img4', 'img5']: img1

Select image 2 from the available images: ['img1', 'img2', 'img3', 'img4', 'img5']: img2

Select image 3 from the available images: ['img1', 'img2', 'img3', 'img4', 'img5']: img3

Graphical password authentication successful.

Please re-enter your password to analyze your typing pattern.

Re-enter your password:

Press ENTER to simulate next keystroke...

