Mini – Project CFL

Roll no : 512

Name : Sejal Pawar

Class : M.S c CS P2

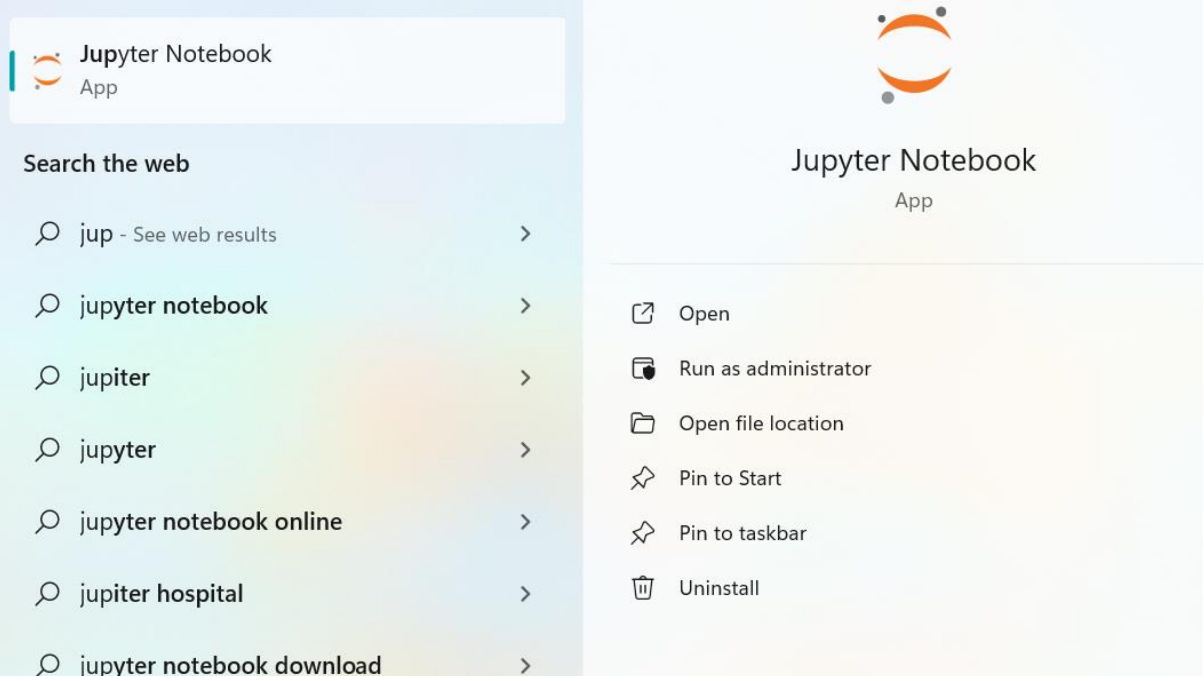
Subject : CFL

Topic : Mini Project

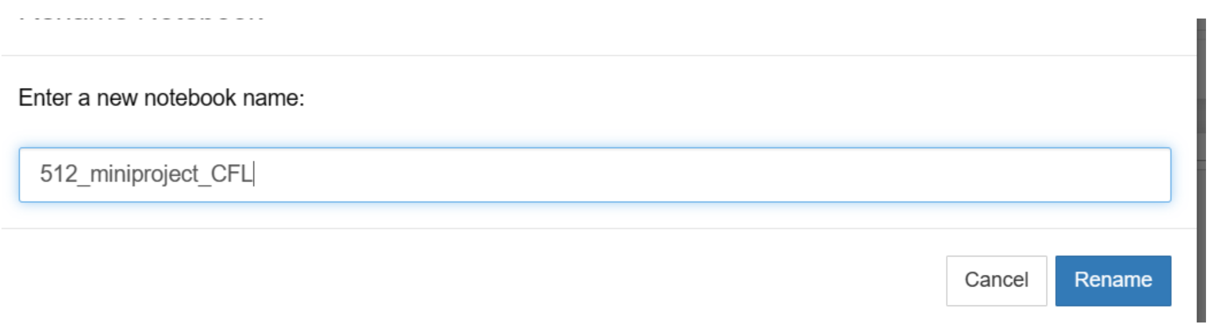
AIM : **User Authentication System using python built-in functions**

Steps :

1) Open Jupyter Notebook :



2) Create and Give Name to file for program code



Enter a new notebook name:

512_miniproject_CFL

3) Importing of libraires is not required as this program is made on built In functions

In the code provided using several built-in functions and features of Python:

1. `input()`: This built-in function is used to get user input from the console.
2. `print()`: Another built-in function used for displaying output to the console.
3. `if`, `else`: These are control flow statements, not functions, but they are built-in features of Python that allow you to perform conditional branching.
4. `in`: The `in` operator is used to check if a value exists in a sequence, such as a list or dictionary.
5. `str` concatenation: You're using string concatenation with the `+` operator to build strings for the output messages.

These are standard features of Python that are available without the need to import any external libraries.

The aim of the program you provided is to demonstrate a simple authentication system. Here are the main objectives of this program:

1. **User Authentication:** The program allows users to enter their username and password.
2. **Authentication Check:** It checks whether the provided username exists in a predefined user database and if the entered password matches the stored password for that username.
3. **Authentication Result:** Based on the authentication check, the program provides feedback to the user:
    - If the authentication is successful, it welcomes the user by their username.
    - If the authentication fails (either due to an incorrect username or password), it informs the user of the failure.

In summary, the program's aim is to simulate a basic login system, where users can enter their credentials, and the program checks if those credentials are valid. It serves as a simple demonstration of user authentication without storing or handling user data in a secure or production-ready manner.

Program Code :

Program code to Create and verify the authencation of user :

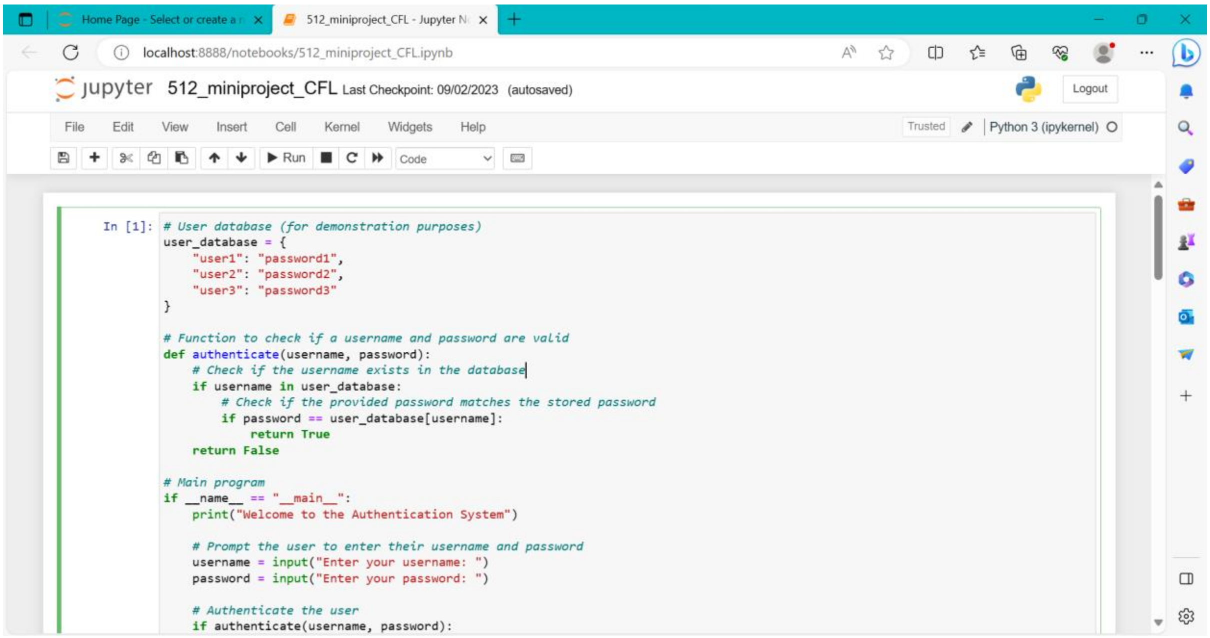
```
# User database (for demonstration purposes)

user_database = {

    "user1": "password1",

    "user2": "password2",

    "user3": "password3"

}


# Function to check if a username and password are valid

def authenticate(username, password):

    # Check if the username exists in the database

    if username in user_database:

        # Check if the provided password matches the stored
password

        if password == user_database[username]:

            return True

    return False


# Main program
```

```
if __name__ == "__main__":
    print("Welcome to the Authentication System")


    # Prompt the user to enter their username and password
    username = input("Enter your username: ")
    password = input("Enter your password: ")


    # Authenticate the user


    if authenticate(username, password):
        print("Authentication successful. Welcome, " + username + "!")
    else:
        print("Authentication failed. Invalid username or password.")
```

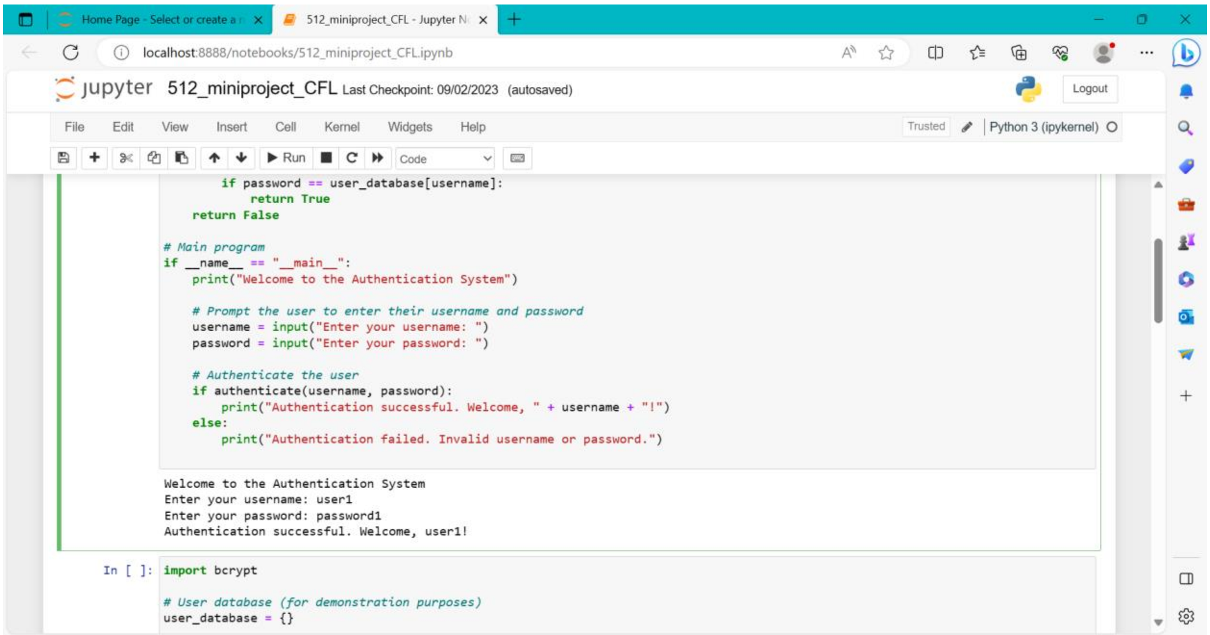
Program to Add Register New user and verify the authencation :

```
import bcrypt

# User database (for demonstration purposes)
user_database = {}

# Function to register a new user
def register(username, password):
    # Check if the username already exists
    if username in user_database:
        return False  # Username is already taken
    else:
        # Hash the password before storing it
        hashed_password = bcrypt.hashpw(password.encode('utf-8'), bcrypt.gensalt())
        user_database[username] = hashed_password
        return True  # Registration successful

# Function to check if a username and password are valid
def authenticate(username, password):
    # Check if the username exists in the database
    if username in user_database:
        # Verify the provided password against the hashed password
        if bcrypt.checkpw(password.encode('utf-8'),
user_database[username]):
            return True
    return False


# Main program
if __name__ == "__main__":
    print("Welcome to the Authentication System")


    while True:
        print("\nOptions:")
        print("1. Register a new user")
        print("2. Authenticate")
        print("3. Quit")


        choice = input("Enter your choice (1/2/3): ")


        if choice == "1":
            username = input("Enter a username: ")
            password = input("Enter a password: ")
            if register(username, password):
                print("Registration successful.")
            else:
                print("Username already exists. Please choose another.")
        elif choice == "2":
            username = input("Enter your username: ")
            password = input("Enter your password: ")
            if authenticate(username, password):
                print("Authentication successful. Welcome, " + username + "!")
            else:
                print("Authentication failed. Invalid username or password.")
        elif choice == "3":
            print("Exiting the program.")
            break
        else:
            print("Invalid choice. Please select 1, 2, or 3.")
```

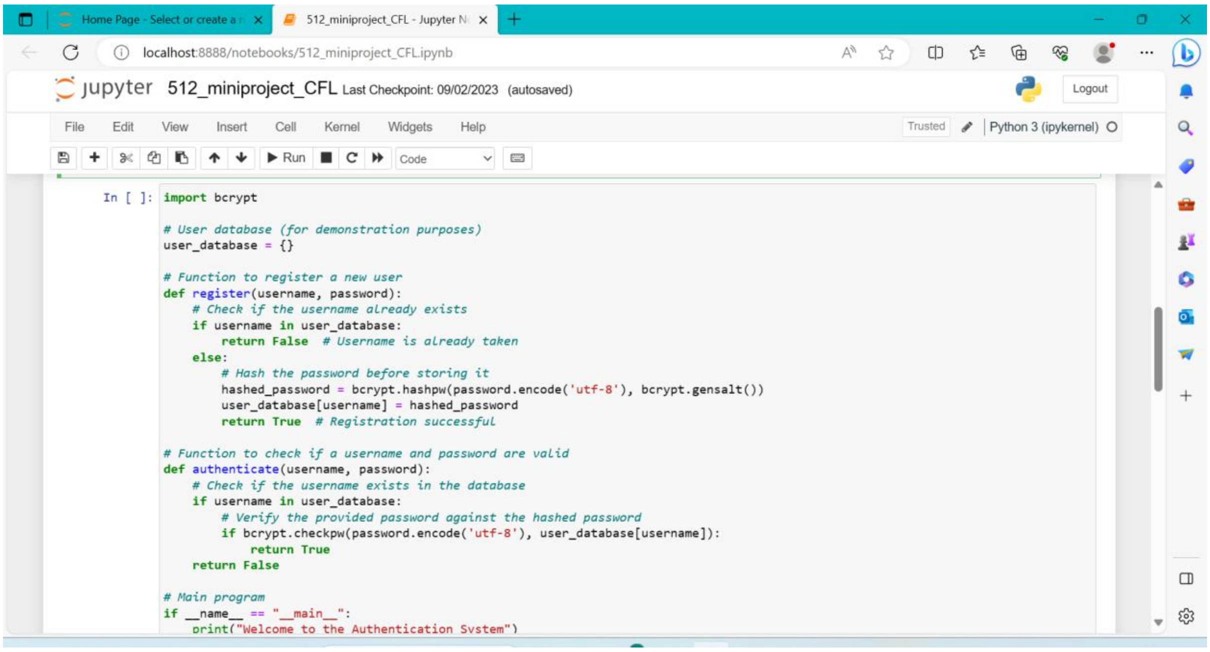
Output of the program :


Home Page - Select or create a
512_miniproject_CFL - Jupyter N
localhost:8888/notebooks/512_miniproject_CFL.ipynb
jupyter 512_miniproject_CFL Last Checkpoint: 09/02/2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

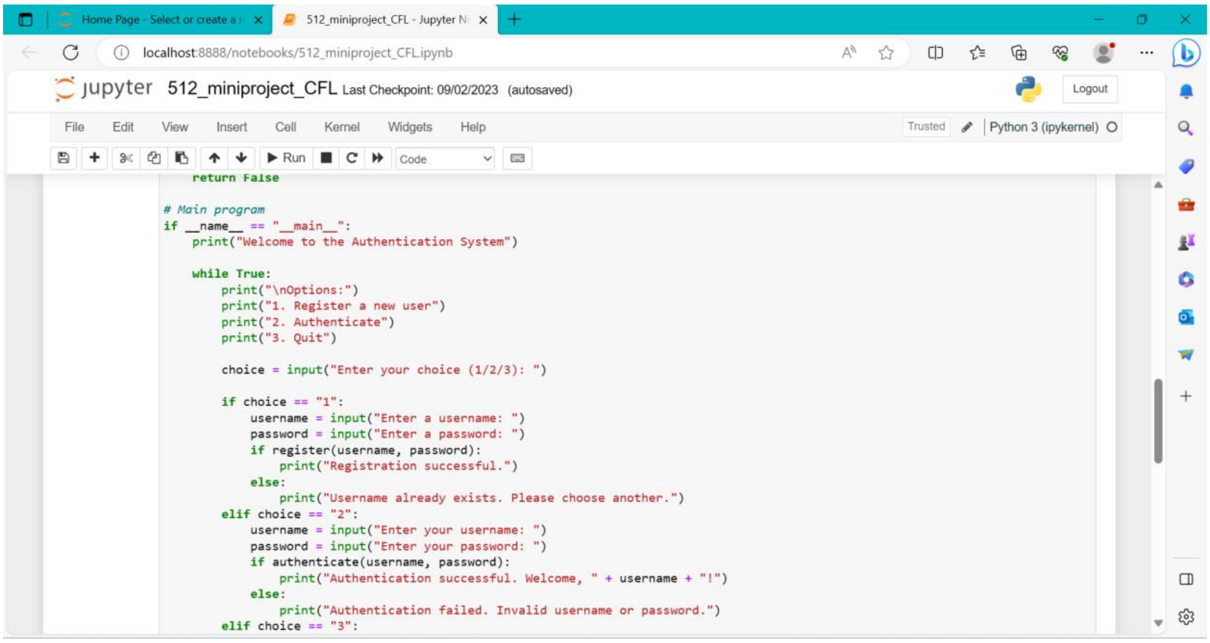
```
In [1]: # User database (for demonstration purposes)
        user_database = {
            "user1": "password1",
            "user2": "password2",
            "user3": "password3"
        }

        # Function to check if a username and password are valid
        def authenticate(username, password):
            # Check if the username exists in the database
            if username in user_database:
                # Check if the provided password matches the stored password
                if password == user_database[username]:
                    return True
            return False

        # Main program
        if __name__ == "__main__":
            print("Welcome to the Authentication System")

            # Prompt the user to enter their username and password
            username = input("Enter your username: ")
            password = input("Enter your password: ")

            # Authenticate the user
            if authenticate(username, password):
```


Home Page - Select or create a
512_miniproject_CFL - Jupyter N
localhost:8888/notebooks/512_miniproject_CFL.ipynb
jupyter 512_miniproject_CFL Last Checkpoint: 09/02/2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

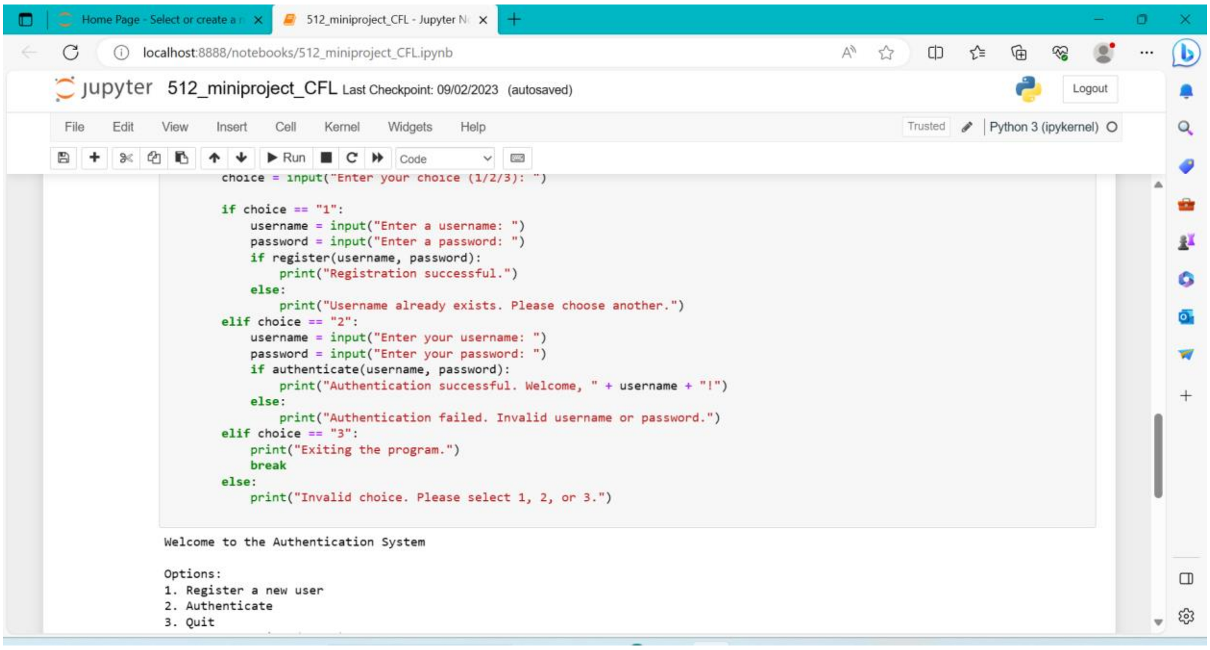
```
                if password == user_database[username]:
                    return True
            return False

        # Main program
        if __name__ == "__main__":
            print("Welcome to the Authentication System")

            # Prompt the user to enter their username and password
            username = input("Enter your username: ")
            password = input("Enter your password: ")

            # Authenticate the user
            if authenticate(username, password):
                print("Authentication successful. Welcome, " + username + "!")
            else:
                print("Authentication failed. Invalid username or password.")

        Welcome to the Authentication System
        Enter your username: user1
        Enter your password: password1
        Authentication successful. Welcome, user1!

In [ ]: import bcrypt

        # User database (for demonstration purposes)
        user_database = {}
```


Home Page - Select or create a n
512_miniproject_CFL - Jupyter N
localhost:8888/notebooks/512_miniproject_CFL.ipynb
jupyter 512_miniproject_CFL Last Checkpoint: 09/02/2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [ ]: import bcrypt

        # User database (for demonstration purposes)
        user_database = {}

        # Function to register a new user
        def register(username, password):
            # Check if the username already exists
            if username in user_database:
                return False  # Username is already taken
            else:
                # Hash the password before storing it
                hashed_password = bcrypt.hashpw(password.encode('utf-8'), bcrypt.gensalt())
                user_database[username] = hashed_password
                return True  # Registration successful

        # Function to check if a username and password are valid
        def authenticate(username, password):
            # Check if the username exists in the database
            if username in user_database:
                # Verify the provided password against the hashed password
                if bcrypt.checkpw(password.encode('utf-8'), user_database[username]):
                    return True
            return False

        # Main program
        if __name__ == "__main__":
            print("Welcome to the Authentication System")
```

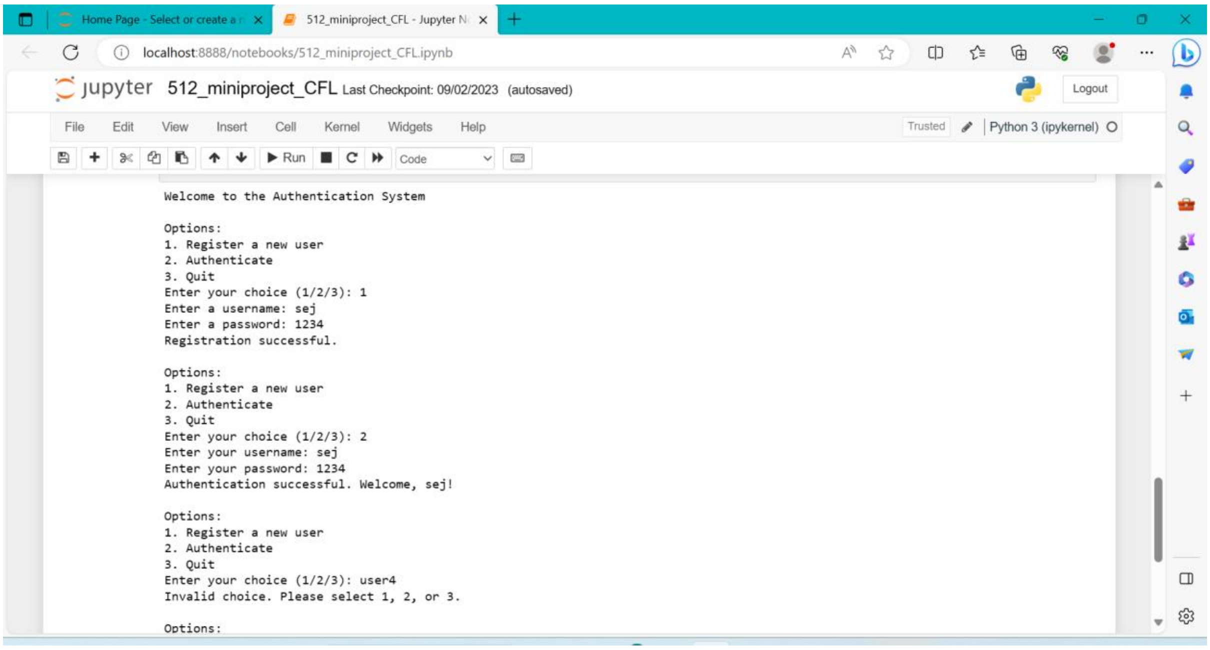

Home Page - Select or create a n
512_miniproject_CFL - Jupyter N
localhost:8888/notebooks/512_miniproject_CFL.ipynb
jupyter 512_miniproject_CFL Last Checkpoint: 09/02/2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
    return False

# Main program
if __name__ == "__main__":
    print("Welcome to the Authentication System")

    while True:
        print("\nOptions:")
        print("1. Register a new user")
        print("2. Authenticate")
        print("3. Quit")

        choice = input("Enter your choice (1/2/3): ")

        if choice == "1":
            username = input("Enter a username: ")
            password = input("Enter a password: ")
            if register(username, password):
                print("Registration successful.")
            else:
                print("Username already exists. Please choose another.")
        elif choice == "2":
            username = input("Enter your username: ")
            password = input("Enter your password: ")
            if authenticate(username, password):
                print("Authentication successful. Welcome, " + username + "!")
            else:
                print("Authentication failed. Invalid username or password.")
        elif choice == "3":
```


Home Page - Select or create a n
512_miniproject_CFL - Jupyter N
localhost:8888/notebooks/512_miniproject_CFL.ipynb
jupyter 512_miniproject_CFL Last Checkpoint: 09/02/2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
        choice = input("Enter your choice (1/2/3): ")

        if choice == "1":
            username = input("Enter a username: ")
            password = input("Enter a password: ")
            if register(username, password):
                print("Registration successful.")
            else:
                print("Username already exists. Please choose another.")
        elif choice == "2":
            username = input("Enter your username: ")
            password = input("Enter your password: ")
            if authenticate(username, password):
                print("Authentication successful. Welcome, " + username + "!")
            else:
                print("Authentication failed. Invalid username or password.")
        elif choice == "3":
            print("Exiting the program.")
            break
        else:
            print("Invalid choice. Please select 1, 2, or 3.")
```

```
Welcome to the Authentication System

Options:
1. Register a new user
2. Authenticate
3. Quit
```


Home Page - Select or create a n
512_miniproject_CFL - Jupyter N
localhost:8888/notebooks/512_miniproject_CFL.ipynb
jupyter 512_miniproject_CFL Last Checkpoint: 09/02/2023 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
Welcome to the Authentication System

Options:
1. Register a new user
2. Authenticate
3. Quit
Enter your choice (1/2/3): 1
Enter a username: sej
Enter a password: 1234
Registration successful.

Options:
1. Register a new user
2. Authenticate
3. Quit
Enter your choice (1/2/3): 2
Enter your username: sej
Enter your password: 1234
Authentication successful. Welcome, sej!

Options:
1. Register a new user
2. Authenticate
3. Quit
Enter your choice (1/2/3): user4
Invalid choice. Please select 1, 2, or 3.

Options:
```