Name : Lokesh Patil

PRN : 21520006

Course : ADS Lab

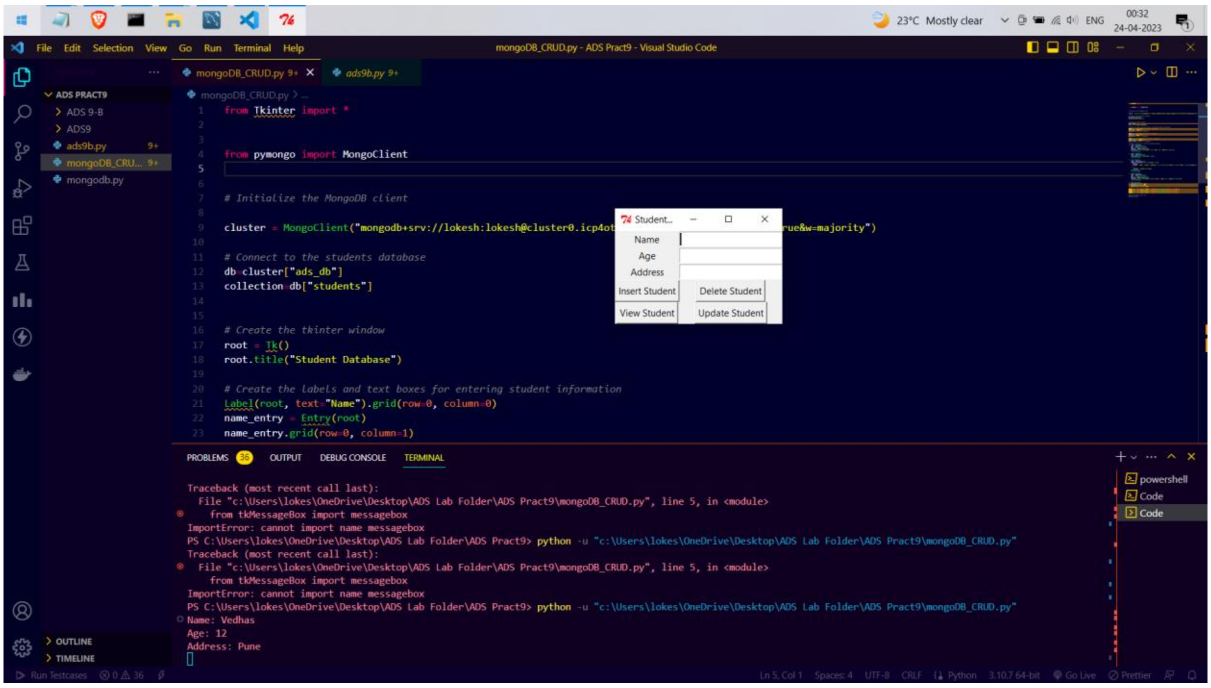
Practical No : 9

Creating databse with mongoDb and Using with python GUI :

```
from Tkinter import *


from pymongo import MongoClient


# Initialize the MongoDB client

cluster = 
MongoClient("mongodb+srv://lokesh:lokesh@cluster0.icp4ote.mongodb.net/?retryWr
ites=true&w=majority")

# Connect to the students database
db=cluster["ads_db"]
collection=db["students"]


# Create the tkinter window
root = Tk()
root.title("Student Database")

# Create the labels and text boxes for entering student information
Label(root, text="Name").grid(row=0, column=0)
name_entry = Entry(root)
name_entry.grid(row=0, column=1)

Label(root, text="Age").grid(row=1, column=0)
age_entry = Entry(root)
age_entry.grid(row=1, column=1)

Label(root, text="Address").grid(row=2, column=0)
address_entry = Entry(root)
address_entry.grid(row=2, column=1)

# Create the buttons for performing CRUD operations
def insert_student():
    name = name_entry.get()
    age = int(age_entry.get())
    address = address_entry.get()
    collection.insert_one({"name": name, "age": age, "address": address})
    print("Record  Inserted  ")
```

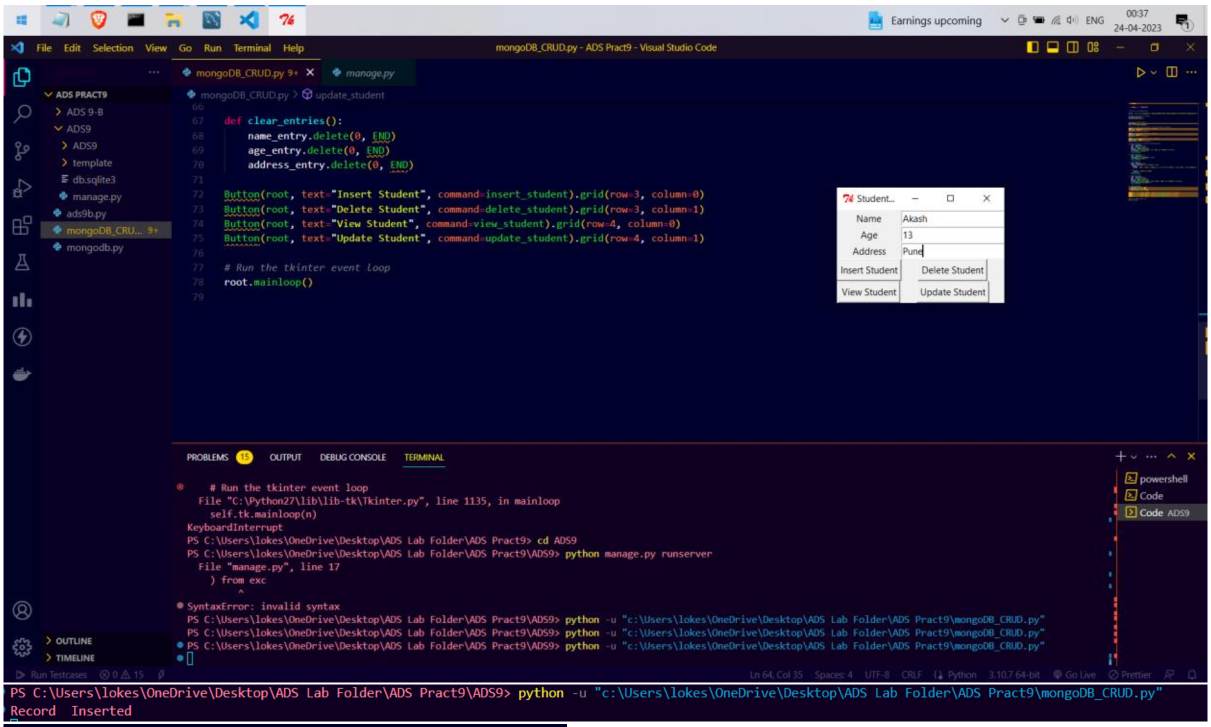
```python
    clear_entries()

def delete_student():
    name = name_entry.get()
    collection.delete_one({"name": name})
    print("Student deleted ")
    clear_entries()

def view_student():
    name = name_entry.get()
    student =collection.find_one({"name": name})
    if student:
       message = "Name: {}\nAge: {}\nAddress: {}".format(student['name'],
student['age'], student['address'])

    else:
        message = "Student not found"
    print(message)
    clear_entries()

def update_student():
    name = name_entry.get()
    age = int(age_entry.get())
    address = address_entry.get()
    collection.update_one({"name": name}, {"$set": {"age": age, "address":
address}})
    print("Student Record Updated  ")
    clear_entries()

def clear_entries():
    name_entry.delete(0, END)
    age_entry.delete(0, END)
    address_entry.delete(0, END)

Button(root, text="Insert Student", command=insert_student).grid(row=3,
column=0)
Button(root, text="Delete Student", command=delete_student).grid(row=3,
column=1)
Button(root, text="View Student", command=view_student).grid(row=4, column=0)
Button(root, text="Update Student", command=update_student).grid(row=4,
column=1)

# Run the tkinter event loop
root.mainloop()
```

Output :



```python
from Tkinter import *


from pymongo import MongoClient


# Initialize the MongoDB client

cluster = MongoClient("mongodb+srv://lokesh:lokesh@cluster0.icp4ot          rue&w=majority")

# Connect to the students database
db=cluster["ads_db"]
collection=db["students"]


# Create the tkinter window
root = Tk()
root.title("Student Database")

# Create the labels and text boxes for entering student information
Label(root, text="Name").grid(row=0, column=0)
name_entry = Entry(root)
name_entry.grid(row=0, column=1)
```

```
Traceback (most recent call last):
  File "c:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\mongoDB_CRUD.py", line 5, in <module>
    from tkMessageBox import messagebox
ImportError: cannot import name messagebox
PS C:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9> python -u "c:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\mongoDB_CRUD.py"
Traceback (most recent call last):
  File "c:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\mongoDB_CRUD.py", line 5, in <module>
    from tkMessageBox import messagebox
ImportError: cannot import name messagebox
PS C:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9> python -u "c:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\mongoDB_CRUD.py"
Name: Vedhas
Age: 12
Address: Pune
```



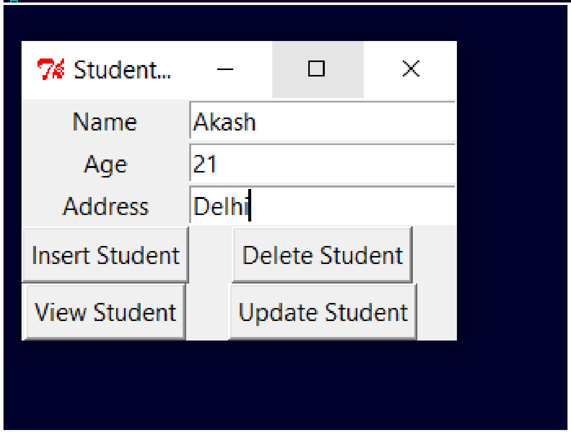
mongoDB_CRUD.py - ADS Pract9 - Visual Studio Code

```python
def clear_entries():
    name_entry.delete(0, END)
    age_entry.delete(0, END)
    address_entry.delete(0, END)

Button(root, text="Insert Student", command=insert_student).grid(row=3, column=0)
Button(root, text="Delete Student", command=delete_student).grid(row=3, column=1)
Button(root, text="View Student", command=view_student).grid(row=4, column=0)
Button(root, text="Update Student", command=update_student).grid(row=4, column=1)

# Run the tkinter event loop
root.mainloop()
```

Name	Akash
Age	13
Address	Pune

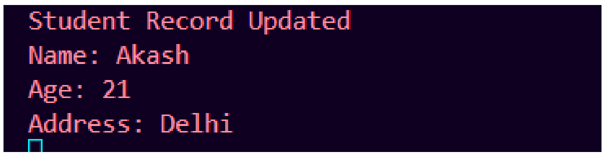
```
    # Run the tkinter event loop
  File "C:\Python27\lib\lib-tk\Tkinter.py", line 1135, in mainloop
    self.tk.mainloop(n)
KeyboardInterrupt
PS C:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9> cd ADS9
PS C:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\ADS9> python manage.py runserver
  File "manage.py", line 17
    ) from exc
     ^
SyntaxError: invalid syntax
PS C:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\ADS9> python -u "c:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\mongoDB_CRUD.py"
PS C:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\ADS9> python -u "c:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\mongoDB_CRUD.py"
PS C:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\ADS9> python -u "c:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\mongoDB_CRUD.py"
```

```
PS C:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\ADS9> python -u "c:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\mongoDB_CRUD.py"
Record  Inserted
```



Student...

Name	Akash
Age	21
Address	Delhi



```
Student Record Updated
Name: Akash
Age: 21
Address: Delhi
```

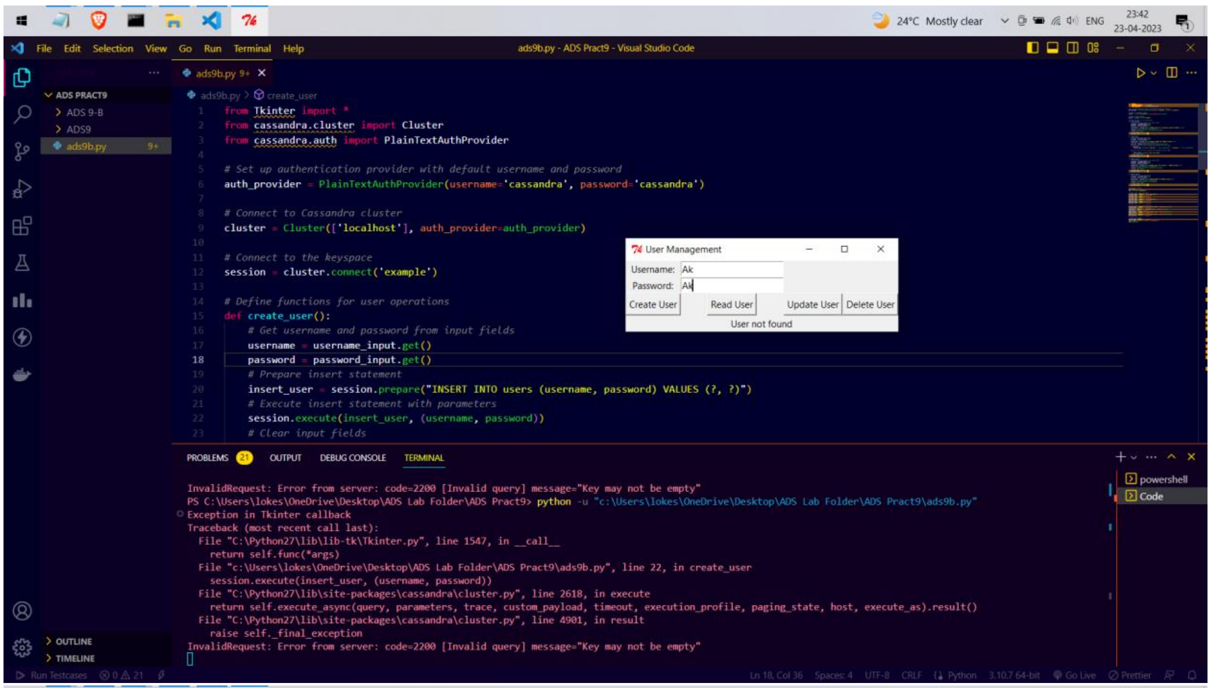
Using cassandra database with python GUI :

```
from Tkinter import *
from cassandra.cluster import Cluster
from cassandra.auth import PlainTextAuthProvider

# Set up authentication provider with default username and password
auth_provider = PlainTextAuthProvider(username='cassandra',
password='cassandra')

# Connect to Cassandra cluster
cluster = Cluster(['localhost'], auth_provider=auth_provider)

# Connect to the keyspace
session = cluster.connect('example')

# Define functions for user operations
def create_user():
    # Get username and password from input fields
    username = username_input.get()
    password = password_input.get()
    # Prepare insert statement
    insert_user = session.prepare("INSERT INTO users (username, password)
VALUES (?, ?)")
    # Execute insert statement with parameters
    session.execute(insert_user, (username, password))
    # Clear input fields
    username_input.delete(0, END)
    password_input.delete(0, END)

def read_user():
    # Get username from input field
    username = username_input.get()
    # Prepare select statement
    select_user = session.prepare("SELECT * FROM users WHERE username = ?")
    # Execute select statement with parameters
    user_row = session.execute(select_user, (username,)).one()
    # If user exists, display username and password in result label
    if user_row:
        result_label.config(text="Username: " + user_row.username + ",
Password: " + user_row.password)

    # If user does not exist, display error message in result label
    else:
        result_label.config(text="User not found")
    # Clear input fields
    username_input.delete(0, END)
    password_input.delete(0, END)
```

```python
def update_user():
    # Get username and password from input fields
    username = username_input.get()
    password = password_input.get()
    # Prepare update statement
    update_user = session.prepare("UPDATE users SET password = ? WHERE
username = ?")
    # Execute update statement with parameters
    session.execute(update_user, (password, username))
    # Clear input fields
    username_input.delete(0, END)
    password_input.delete(0, END)

def delete_user():
    # Get username from input field
    username = username_input.get()
    # Prepare delete statement
    delete_user = session.prepare("DELETE FROM users WHERE username = ?")
    # Execute delete statement with parameters
    session.execute(delete_user, (username,))
    # Clear input fields
    username_input.delete(0, END)
    password_input.delete(0, END)

# Create main window
root = Tk()
root.title("User Management")

# Create input fields and labels
username_label = Label(root, text="Username:")
username_label.grid(row=0, column=0)
username_input = Entry(root)
username_input.grid(row=0, column=1)
password_label = Label(root, text="Password:")
password_label.grid(row=1, column=0)
password_input = Entry(root)
password_input.grid(row=1, column=1)

# Create buttons for user operations
create_button = Button(root, text="Create User", command=create_user)
create_button.grid(row=2, column=0)
read_button = Button(root, text="Read User", command=read_user)
read_button.grid(row=2, column=1)
update_button = Button(root, text="Update User", command=update_user)
update_button.grid(row=2, column=2)
delete_button = Button(root, text="Delete User", command=delete_user)
delete_button.grid(row=2, column=3)
```

```
# Create result label
result_label = Label(root, text="")
result_label.grid(row=3, column=0, columnspan=4)
root.mainloop()
```

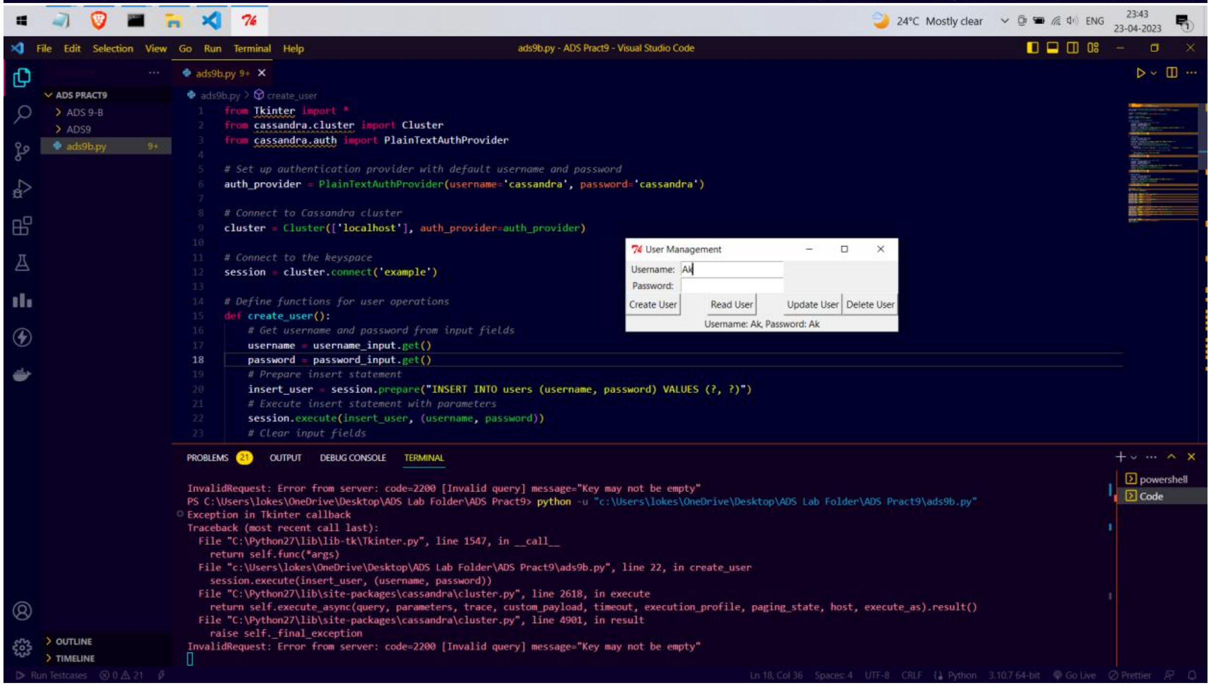
Output :



```
from Tkinter import *
from cassandra.cluster import Cluster
from cassandra.auth import PlainTextAuthProvider

# Set up authentication provider with default username and password
auth_provider = PlainTextAuthProvider(username='cassandra', password='cassandra')

# Connect to Cassandra cluster
cluster = Cluster(['localhost'], auth_provider=auth_provider)

# Connect to the keyspace
session = cluster.connect('example')

# Define functions for user operations
def create_user():
    # Get username and password from input fields
    username = username_input.get()
    password = password_input.get()
    # Prepare insert statement
    insert_user = session.prepare("INSERT INTO users (username, password) VALUES (?, ?)")
    # Execute insert statement with parameters
    session.execute(insert_user, (username, password))
    # Clear input fields
```

User Management
Username: Ak
Password: Ak
User not found

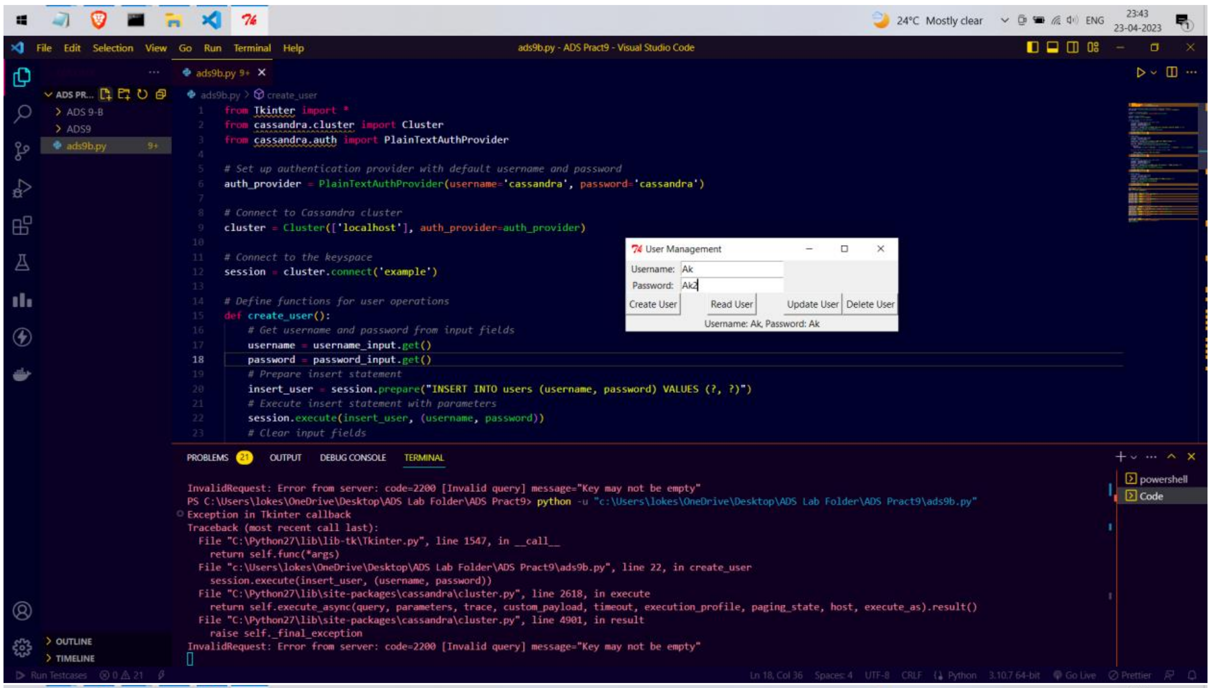
```
InvalidRequest: Error from server: code=2200 [Invalid query] message="Key may not be empty"
PS C:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9> python -u "c:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\ads9b.py"
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Python27\lib\lib-tk\Tkinter.py", line 1547, in __call__
    return self.func(*args)
  File "c:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\ads9b.py", line 22, in create_user
    session.execute(insert_user, (username, password))
  File "C:\Python27\lib\site-packages\cassandra\cluster.py", line 2618, in execute
    return self.execute_async(query, parameters, trace, custom_payload, timeout, execution_profile, paging_state, host, execute_as).result()
  File "C:\Python27\lib\site-packages\cassandra\cluster.py", line 4901, in result
    raise self._final_exception
InvalidRequest: Error from server: code=2200 [Invalid query] message="Key may not be empty"
```



```
from Tkinter import *
from cassandra.cluster import Cluster
from cassandra.auth import PlainTextAuthProvider

# Set up authentication provider with default username and password
auth_provider = PlainTextAuthProvider(username='cassandra', password='cassandra')

# Connect to Cassandra cluster
cluster = Cluster(['localhost'], auth_provider=auth_provider)

# Connect to the keyspace
session = cluster.connect('example')

# Define functions for user operations
def create_user():
    # Get username and password from input fields
    username = username_input.get()
    password = password_input.get()
    # Prepare insert statement
    insert_user = session.prepare("INSERT INTO users (username, password) VALUES (?, ?)")
    # Execute insert statement with parameters
    session.execute(insert_user, (username, password))
    # Clear input fields
```

User Management
Username: Ak
Password:
Username: Ak, Password: Ak

```
InvalidRequest: Error from server: code=2200 [Invalid query] message="Key may not be empty"
PS C:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9> python -u "c:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\ads9b.py"
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Python27\lib\lib-tk\Tkinter.py", line 1547, in __call__
    return self.func(*args)
  File "c:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\ads9b.py", line 22, in create_user
    session.execute(insert_user, (username, password))
  File "C:\Python27\lib\site-packages\cassandra\cluster.py", line 2618, in execute
    return self.execute_async(query, parameters, trace, custom_payload, timeout, execution_profile, paging_state, host, execute_as).result()
  File "C:\Python27\lib\site-packages\cassandra\cluster.py", line 4901, in result
    raise self._final_exception
InvalidRequest: Error from server: code=2200 [Invalid query] message="Key may not be empty"
```



ads9b.py - ADS Pract9 - Visual Studio Code

```python
from Tkinter import *
from cassandra.cluster import Cluster
from cassandra.auth import PlainTextAuthProvider

# Set up authentication provider with default username and password
auth_provider = PlainTextAuthProvider(username='cassandra', password='cassandra')

# Connect to Cassandra cluster
cluster = Cluster(['localhost'], auth_provider=auth_provider)

# Connect to the keyspace
session = cluster.connect('example')

# Define functions for user operations
def create_user():
    # Get username and password from input fields
    username = username_input.get()
    password = password_input.get()
    # Prepare insert statement
    insert_user = session.prepare("INSERT INTO users (username, password) VALUES (?, ?)")
    # Execute insert statement with parameters
    session.execute(insert_user, (username, password))
    # Clear input fields
```

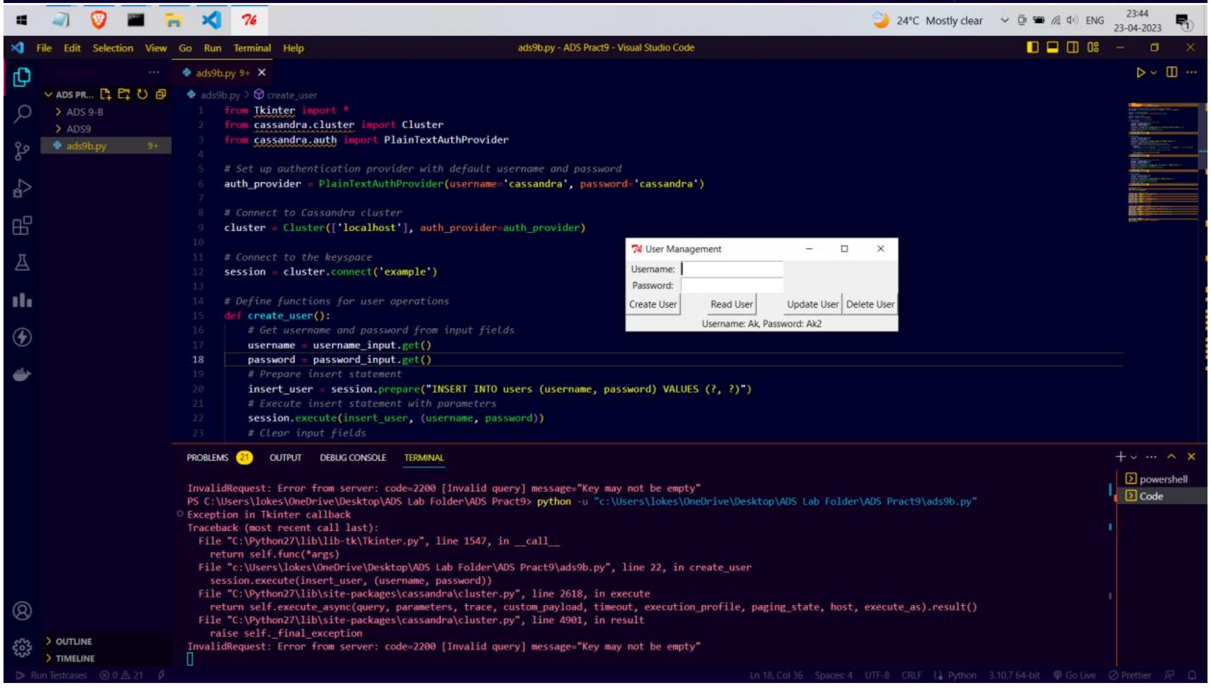
User Management

Username: Ak
Password: Ak2

Create User | Read User | Update User | Delete User

Username: Ak, Password: Ak

```
InvalidRequest: Error from server: code=2200 [Invalid query] message="Key may not be empty"
PS C:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9> python -u "c:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\ads9b.py"
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Python27\lib\lib-tk\Tkinter.py", line 1547, in __call__
    return self.func(*args)
  File "c:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\ads9b.py", line 22, in create_user
    session.execute(insert_user, (username, password))
  File "C:\Python27\lib\site-packages\cassandra\cluster.py", line 2618, in execute
    return self.execute_async(query, parameters, trace, custom_payload, timeout, execution_profile, paging_state, host, execute_as).result()
  File "C:\Python27\lib\site-packages\cassandra\cluster.py", line 4901, in result
    raise self._final_exception
InvalidRequest: Error from server: code=2200 [Invalid query] message="Key may not be empty"
```



ads9b.py - ADS Pract9 - Visual Studio Code

```python
from Tkinter import *
from cassandra.cluster import Cluster
from cassandra.auth import PlainTextAuthProvider

# Set up authentication provider with default username and password
auth_provider = PlainTextAuthProvider(username='cassandra', password='cassandra')

# Connect to Cassandra cluster
cluster = Cluster(['localhost'], auth_provider=auth_provider)

# Connect to the keyspace
session = cluster.connect('example')

# Define functions for user operations
def create_user():
    # Get username and password from input fields
    username = username_input.get()
    password = password_input.get()
    # Prepare insert statement
    insert_user = session.prepare("INSERT INTO users (username, password) VALUES (?, ?)")
    # Execute insert statement with parameters
    session.execute(insert_user, (username, password))
    # Clear input fields
```

User Management

Username:
Password:

Create User | Read User | Update User | Delete User

Username: Ak, Password: Ak2

```
InvalidRequest: Error from server: code=2200 [Invalid query] message="Key may not be empty"
PS C:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9> python -u "c:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\ads9b.py"
Exception in Tkinter callback
Traceback (most recent call last):
  File "C:\Python27\lib\lib-tk\Tkinter.py", line 1547, in __call__
    return self.func(*args)
  File "c:\Users\lokes\OneDrive\Desktop\ADS Lab Folder\ADS Pract9\ads9b.py", line 22, in create_user
    session.execute(insert_user, (username, password))
  File "C:\Python27\lib\site-packages\cassandra\cluster.py", line 2618, in execute
    return self.execute_async(query, parameters, trace, custom_payload, timeout, execution_profile, paging_state, host, execute_as).result()
  File "C:\Python27\lib\site-packages\cassandra\cluster.py", line 4901, in result
    raise self._final_exception
InvalidRequest: Error from server: code=2200 [Invalid query] message="Key may not be empty"
```