~\Desktop\ST4000CEM\Bank Management System\update_account.py

```
from tkinter import *
root = Tk()
import sqlite3
from tkinter import messagebox
root.title("Update All Details")
root.geometry("750x400")
root.resizable(0, 0)

def clear():
    name_entry.delete(0, END)
    dob_entry.delete(0, END)
    email_entry.delete(0, END)
    phone_number_entry.delete(0, END)
    address_entry.delete(0, END)

def search():
    conn = sqlite3.connect('Bank Management System.db')
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM accounts WHERE AC_No=?", (account_number_entry.get(),))
    account = cursor.fetchone()
    if account:
        fill_details(account)
    else:
        messagebox.showerror("Account Not Found", f"The account number '
{account_number_entry.get()}' does not exist.")

def fill_details(account):
    name_entry.delete(0, END)
    dob_entry.delete(0, END)
    email_entry.delete(0, END)
    phone_number_entry.delete(0, END)
    address_entry.delete(0, END)

    name_entry.insert(0, account[1])
    dob_entry.insert(0, account[3])
    email_entry.insert(0, account[4])
    phone_number_entry.insert(0, account[5])
    address_entry.insert(0, account[6])

def update():
    conn = sqlite3.connect('Bank Management System.db')
    cursor = conn.cursor()
    cursor.execute("UPDATE accounts SET Name=?, DOB=?, Email=?, [Phone_Number]=?, Address=?
WHERE AC_No=?",
                   (name_entry.get().upper(), dob_entry.get().upper(), email_entry.get()
.lower(), phone_number_entry.get(),
                    address_entry.get().upper(), account_number_entry.get()))
    conn.commit()
    conn.close()
    messagebox.showinfo("Success", "Account details updated successfully.")
    import customer_details

account_number_label = Label(root, text="Account Number : ", font=("Arial", 12, "bold"))
account_number_label.place(x=100, y=50)
```

```
52
53  account_number_entry = Entry(root, width=30)
54  account_number_entry.place(x=250, y=50)
55
56  search_btn = Button(root, text="Search", font=("Arial Bold", 12), fg="white", bg="green",
    width=10, cursor="hand2", command=search)
57  search_btn.place(x=500, y=45)
58
59  name_label = Label(root, text="Name : ", font=("Arial", 12, "bold"))
60  name_label.place(x=100, y=90)
61
62  dob_label = Label(root, text="DOB : ", font=("Arial", 12, "bold"))
63  dob_label.place(x=100, y=130)
64
65  email_label = Label(root, text="Email : ", font=("Arial", 12, "bold"))
66  email_label.place(x=100, y=170)
67
68  phone_number_label = Label(root, text="Phone Number : ", font=("Arial", 12, "bold"))
69  phone_number_label.place(x=100, y=210)
70
71  address_label = Label(root, text="Address : ", font=("Arial", 12, "bold"))
72  address_label.place(x=100, y=250)
73
74  name_entry = Entry(root, width=30)
75  name_entry.place(x=250, y=90)
76
77  dob_entry = Entry(root, width=30)
78  dob_entry.place(x=250, y=130)
79
80  email_entry = Entry(root, width=30)
81  email_entry.place(x=250, y=170)
82
83  phone_number_entry = Entry(root, width=30)
84  phone_number_entry.place(x=250, y=210)
85
86  address_entry = Entry(root, width=30)
87  address_entry.place(x=250, y=250)
88
89  update_btn = Button(root, text="Update", font=("Arial Bold", 12), fg="white", bg="green",
    width=10, cursor="hand2", command=update)
90  update_btn.place(x=300, y=320)
91
92  clear_btn = Button(root, text="Clear", font=("Arial Bold", 12), fg="white", bg="red", width=
    10, cursor="hand2", command=clear)
93  clear_btn.place(x=450, y=320)
94
95  root.mainloop()
96
```