```
# EcoTrack
print("Welcome to Ecotrack! Let's reduce your carbon footprint")


def entrypoint():


    ch = input("Choose: Login / Signup: ")


    if ch.lower() in ["signup", "sign up"]:
        signup()
    elif ch.lower() == "login":
        login()
    else:
        print("Invalid choice. Please enter Login or Signup")



def signup():
    import pickle
    data = {}


    d = input("Enter your email id: ")
    while "@gmail.com" not in d:
        print("Please enter a valid email address! (name@gmail.com)")
        d = input("Email ID: ")


    e = input("Enter your password: ")
    N = input("Enter nickname: ")
```

```
data["EmailId"] = d
data["Password"] = e
data["Nickname"] = N
data["Vehicles"] = None

# Load existing users
users = []
try:
    with open("user.dat", "rb") as fp:
        try:
            while True:
                users.append(pickle.load(fp))
        except EOFError:
            pass
except FileNotFoundError:
    pass

# Add new user and save
users.append(data)
with open("user.dat", "wb") as fp:
    for u in users:
        pickle.dump(u, fp)

print("Signup successful! Please login to continue.")
login()
```

```
def login():
    import pickle
    import time

    email = input("Enter email: ")
    while "@gmail.com" not in email:
        print("Please enter a valid email address! (name@gmail.com)")
        email = input("Enter email: ")

    try:
        users = []
        with open("user.dat", "rb") as fp1:
            try:
                while True:
                    users.append(pickle.load(fp1))
            except EOFError:
                pass
    except FileNotFoundError:
        print("No users found. Please signup first.")
        entrypoint()
        return

    user = None
    for u in users:
        if u.get("EmailId") == email:
            user = u
            break
```

```
if user is None:
    print("No user found with that email. Please signup first.")
    entrypoint()
    return

# only ask password if user exists
attempts = 0
while attempts < 3:
    password = input("Enter password: ")
    if user["Password"] == password:
        print("Login successful! Welcome,", user["Nickname"])

        if not user.get("Vehicles"):
            user["Vehicles"] = recorddetails()

        # update file
        with open("user.dat", "wb") as fp:
            for u in users:
                if isinstance(u, dict) and u.get("EmailId") == user.get("EmailId"):
                    pickle.dump(user, fp)
                else:
                    pickle.dump(u, fp)

        calculate(user["Vehicles"])
        return
    else:
            attempts += 1
            print("Wrong password. Attempts left:", 3 - attempts)

    print("Too many failed attempts. Please wait 10 seconds.")
    time.sleep(10)
    print("You can try again now.")
    login()



def recorddetails():
    print("Enter your vehicle details, so we can calculate your carbon footprint!")

    vehicles = []
    Num = int(input("\nEnter number of vehicles in your household: "))

    for i in range(Num):
        print(f"\nDetails of Vehicle {i + 1}")

        # fuel type validation loop
        while True:
            fuel_type = input("Enter your fuel type (petrol/diesel/cng): ")

            if fuel_type.lower() == "petrol":
                emfact = 2.31
                break
            elif fuel_type.lower() == "diesel":
                emfact = 2.68
                break
            elif fuel_type.lower() == "cng":
                emfact = 2.69
                break
            else:
                print("Please enter a valid fuel type.")

        mil = float(input("Enter your vehicle's mileage (km/l): "))

        vehicles.append({
            "FuelType": fuel_type,
            "Mileage": mil,
            "EmissionFactor": emfact
        })

    print("\nAll your vehicle details have been recorded successfully!")
    return vehicles



def calculate(vehicles):
    import time

    print("\nCalculating today's carbon footprint...")

    total_emission = 0

    for idx, v in enumerate(vehicles, start=1):
        print(f"\nVehicle {idx}:")
        dis = float(input(f"Enter distance travelled today (km) for Vehicle {idx}: "))

        fuel_used = dis / v["Mileage"]
        emission = fuel_used * v["EmissionFactor"]
        total_emission += emission

        print("Your carbon footprint for this vehicle is:",
              round(emission, 2), "kg CO2")

    print("Analysing your consumption...")
    time.sleep(2)

    print("\nYour TOTAL carbon footprint for today is:",
          round(total_emission, 2), "kg CO2")

    if total_emission < 3:
        print("Low: Great job keeping emissions minimal!")
    elif total_emission < 8:
        print("Moderate: Try walking/cycling short trips.")
    elif total_emission < 15:
        print("High: Consider carpooling or public transport.")
    else:
        print("Very High: Strongly consider EVs, CNG, or fewer trips.")


# start program
```

entrypoint()