step by step to configure ramadan timer interface

Download Warp as its simpler to use and gives u autofill if needed

I beleive if brew doesnt exist just add

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/H
```

1️⃣ Install Python (if not installed)

Mac (using Zsh Terminal):

- Open Terminal (`Cmd + Space` → type Terminal → press Enter).
- Type:

  and press Enter.

  ```
  brew install python
  ```

- After installation, check Python’s version:

  If you see a version number, Python is installed! ✅

  ```
  python3 --version
  ```

2️⃣ Create the Python Script

- Open Terminal.

- Navigate to the folder where you want to save the script:

```
cd ~/Desktop
```

- Create a new Python file:

```
touch ramadan_timer.py
```

- Open the file to edit:

```
nano ramadan_timer.py
```

- add this code

```
import time
from datetime import datetime, timedelta

# Set Ramadan start date
ramadan_start = datetime(2025, 3, 2)

# Set daily prayer times
prayer_times = {
    "Fajr": "05:19",
    "Shuruq": "07:12",
    "Dhuhr": "12:41",
    "Asr": "16:16",
    "Maghrib": "18:10",
    "Isha": "19:56",
    "Suhoor": "05:00",
    "Iftar": "18:30"
}

# Convert times to datetime objects
def get_prayer_times():
    return {name: datetime.strptime(time_str, "%H:%M").time() for name, time_str
```

```
prayer_times_today = get_prayer_times()

# Function to increment prayer times daily at midnight
def increment_prayer_times():
    global prayer_times_today
    prayer_times_today = {
        name: (datetime.combine(datetime.today(), time) + timedelta(minutes=1)).tir
        for name, time in prayer_times_today.items()
    }

# Countdown to the next prayer
def countdown_to_event():
    now = datetime.now()
    upcoming_events = sorted([(name, datetime.combine(now.date(), time)) for na

    for name, event_time in upcoming_events:
        if now < event_time:
            return name, event_time - now

    # If all prayers passed, count down to tomorrow’s Fajr
    next_day_fajr = datetime.combine(now.date() + timedelta(days=1), prayer_time
    return "Fajr", next_day_fajr - now

# Display prayer times at the top
def display_prayer_times():
    print("\n🕰 Today's Prayer Times:")
    for name, time in prayer_times_today.items():
        print(f"🪬 {name}: {time.strftime('%H:%M')}")
    print("-" * 30)

# Display dynamic countdown on the same line
def display_countdown():
    last_update_day = datetime.today().day  # Track the day to update times

    while True:
        event_name, time_left = countdown_to_event()

        # Update prayer times at midnight
        if datetime.today().day != last_update_day:
            increment_prayer_times()
            last_update_day = datetime.today().day

        # Print countdown on one line dynamically
        print(f"\r⏳ Time until {event_name}: {str(time_left).split('.')[0]} ", end="", flu

        time.sleep(1)  # Update every second

# Run program
display_prayer_times()
display_countdown()
```

- Copy and paste the script into the file.
- Save the file:
  - Press `CTRL + X`.
  - Press `Y` to confirm.
  - Press `Enter` to save.

3️⃣ Run the Script

- In Terminal, navigate to the folder where the file is saved:

  ```
  cd ~/Desktop
  ```

- Run the script:

  ```
  python3 ramadan_timer.py
  ```

4️⃣ What You'll See

- ✅ At the top: A list of today's prayer times.
- ✅ At the bottom: A countdown timer updating every second on one line.
- ✅ Every midnight: Prayer times increase by 1 minute automatically.

5️⃣ Stop the Script (if needed)

- Press `CTRL + C` to stop the script anytime.