

**دانشگاه اصفهان**

**دانشکده مهندسی کامپیوتر**

**پروژه پایانی درس تجهیزات IOT**

گردآورندگان **:**

**عادل کر یمی 993613050**

**دانیال توکلی 993613017**

**امیرحسین بهرامی 993613011**

فهرست

[سوال اول: روشن و خاموش کردن LED روی بورد وای فای را از طریق مرورگر 3](#_Toc170649253)

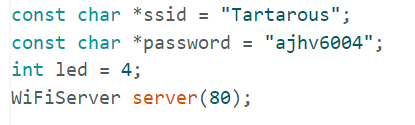
[سوال های دوم و سوم و پنجم: راه اندازی یک وب سرور روی اینترنت و روشن و خاموش کردن LED به واسطه ان 4](#_Toc170649254)

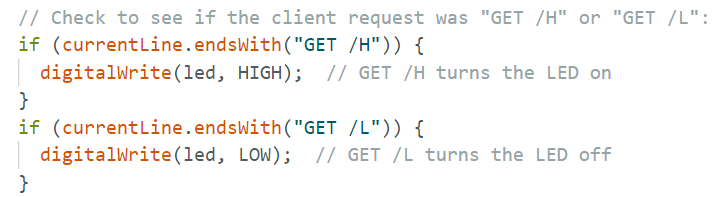
[سوال چهارم:خاموش و روشن کردن یک وسیله منزل از طریق اینترنت با استفاده از ماژول رله 8](#_Toc170649255)

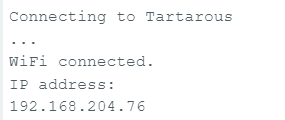
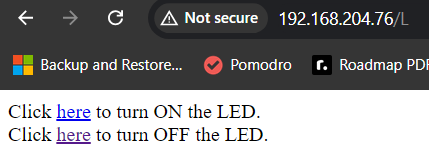
[سوال ششم: دریافت فرمان از طریق تلگرام 9](#_Toc170649256)

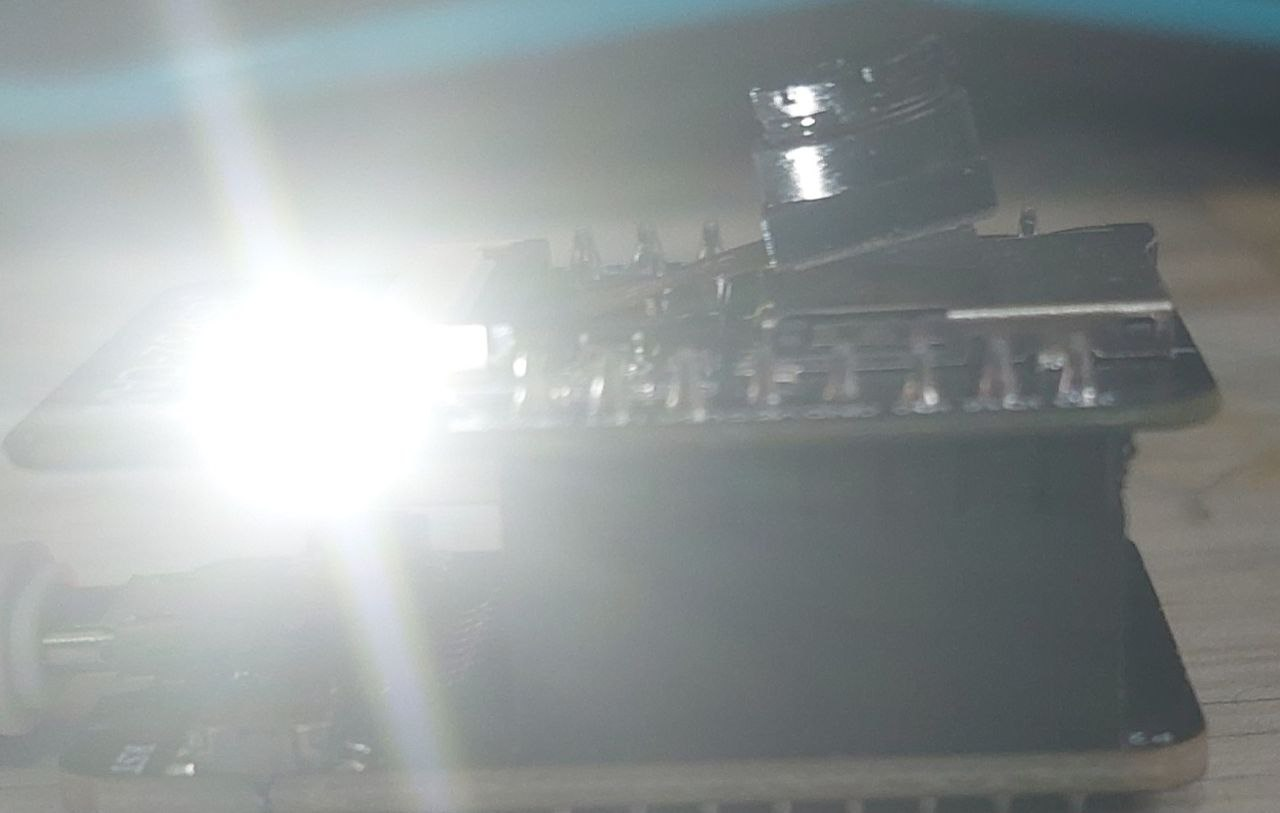
# سوال اول: روشن و خاموش کردن LED روی بورد وای فای را از طریق مرورگر

تغییرات اعمال شده بر روی پروژه طبق اسلاید های درس.







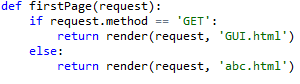
# سوال های دوم و سوم و پنجم: راه اندازی یک وب سرور روی اینترنت و روشن و خاموش کردن LED به واسطه ان

به دلیل پیچیدگی و زمان‌بر بودن فرآیند ساخت URL، برای راه‌اندازی یک وب سرور، تصمیم گرفتیم از سرویس [pythonanywhere](https://www.pythonanywhere.com/) بهره بگیریم. در این پروژه از فریمورک Django استفاده کرده‌ایم و آدرس <https://danial.pythonanywhere.com/> را ایجاد کرده‌ایم. با وارد کردن این آدرس، می‌توانید به صفحه اصلی سایت دسترسی پیدا کنید و اطلاعات مورد نیاز از آنجا دریافت شود.

این وب سرور از طریق APIهای مختلف به کاربران امکان می‌دهد تا وضعیت دستگاه‌ها را بررسی و کنترل کنند و زمان‌های روشن و خاموش شدن دستگاه‌ها را برنامه‌ریزی کنند.

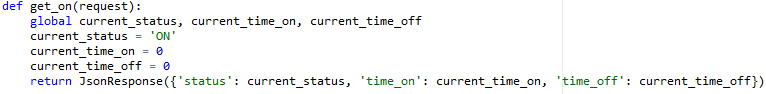
## **۱. صفحه اصلی (Home Page)**

زمانی که کاربر به URL اصلی وب سرور دسترسی پیدا می‌کند، صفحه اصلی (که به عنوان GUI.html شناخته می‌شود) نمایش داده می‌شود. این صفحه شامل یک فرم برای تنظیم زمان‌های روشن و خاموش شدن دستگاه و دکمه‌هایی برای روشن و خاموش کردن دستگاه به صورت دستی است. همچنین وضعیت کنونی دستگاه و زمان‌های تنظیم شده نمایش داده می‌شوند.



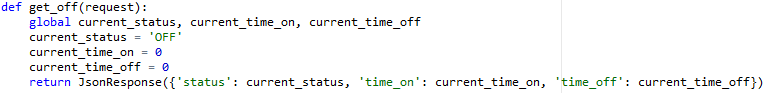
## **۲. API روشن کردن دستگاه (Turn On)**

وقتی کاربر دکمه "Turn On" را فشار می‌دهد، یک درخواست به URL /on ارسال می‌شود. این درخواست وضعیت دستگاه را به "ON" تغییر می‌دهد و زمان‌های روشن و خاموش را به صفر برمی‌گرداند. پاسخ به کاربر شامل وضعیت جدید دستگاه یعنی "ON")) و زمان‌های به‌روزرسانی شده است.



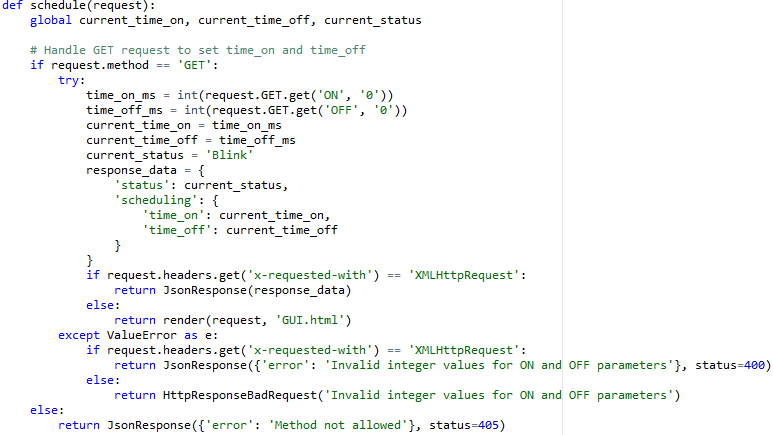
## **۳. API خاموش کردن دستگاه (Turn Off)**

با فشار دادن دکمه "Turn Off"، یک درخواست به URL /off ارسال می‌شود. این درخواست وضعیت دستگاه را به "OFF" تغییر می‌دهد و زمان‌های روشن و خاموش را به صفر برمی‌گرداند. پاسخ شامل وضعیت جدید دستگاه یعنی "OFF")) و زمان‌های به‌روزرسانی شده است.



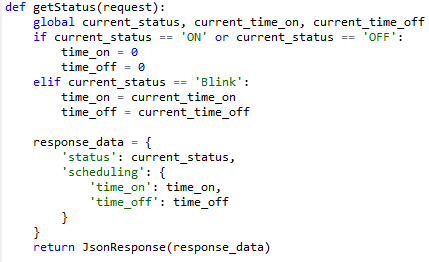
## **۴. API تنظیم زمان‌بندی (Scheduling)**

کاربر می‌تواند زمان‌های روشن و خاموش شدن دستگاه را از طریق فرم موجود در صفحه اصلی تنظیم کند. زمانی که فرم ارسال می‌شود، یک درخواست به URL /schedule با پارامترهای ON و OFF ارسال می‌شود. این درخواست زمان‌های روشن و خاموش شدن دستگاه را تنظیم می‌کند و وضعیت دستگاه را به "Blink" تغییر می‌دهد. پاسخ شامل وضعیت جدید دستگاه یعنی "Blink")) و زمان‌های تنظیم شده است.



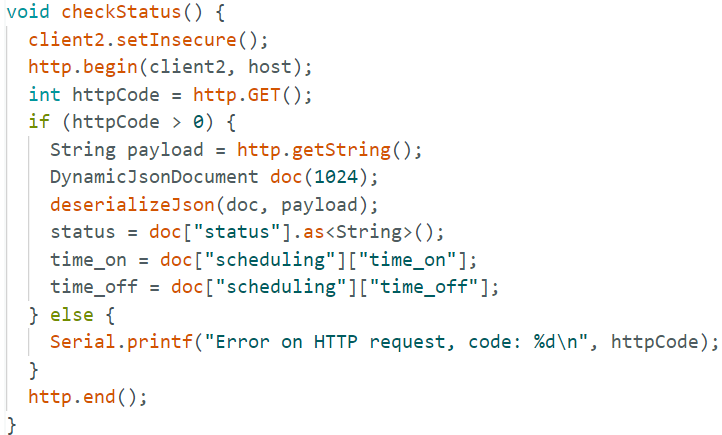
## **۵. API بررسی وضعیت (Status Check)**

برای بررسی وضعیت کنونی دستگاه و زمان‌های روشن و خاموش شدن، یک درخواست به URL /getStatus ارسال می‌شود. پاسخ شامل وضعیت کنونی دستگاه و زمان‌های تنظیم شده است. این اطلاعات به کاربر نمایش داده می‌شود تا بتواند به راحتی وضعیت دستگاه را مشاهده کند.

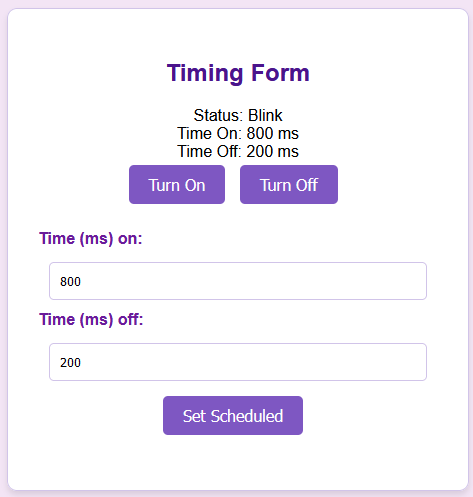


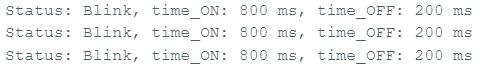
## برنامه سمت client:

با استفاده از تابع checkStatus() ، به صورت دائم به وضعیت سرور بررسی میشود و از طریق کتابخانه ی ArduinoJson، اطلاعات مربوط به وضعیت on, off, Blink و time\_on, time\_off دریافت میشود.



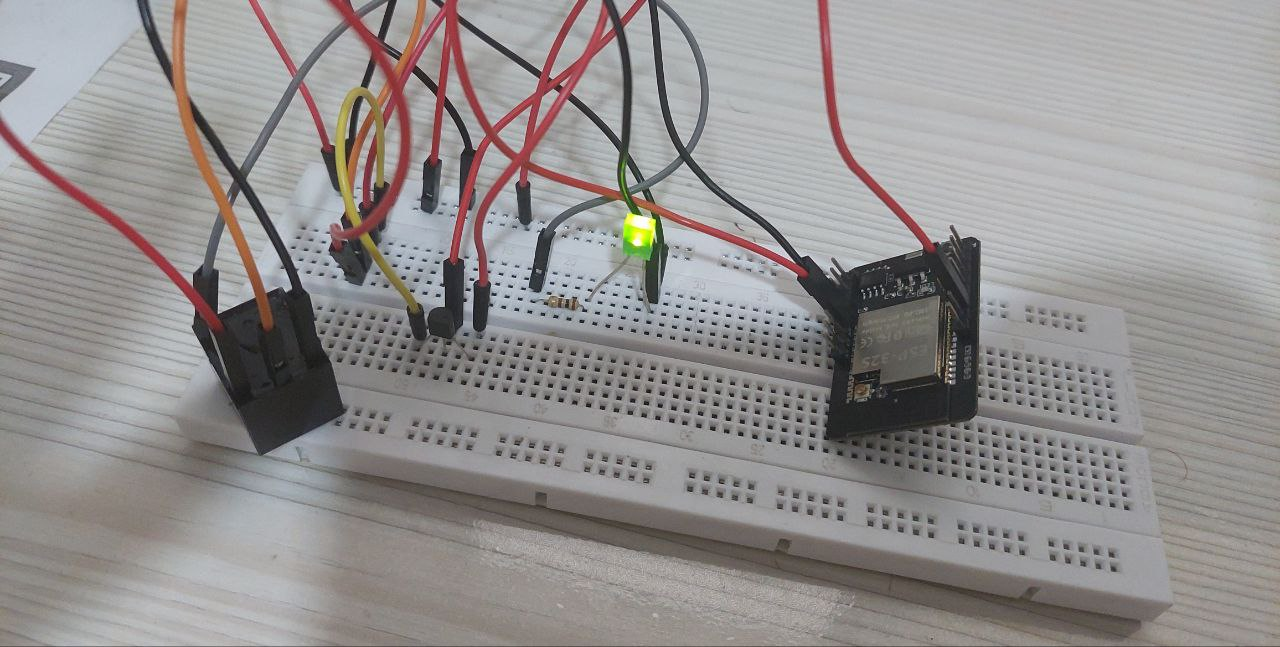
## نمونه اجرا:





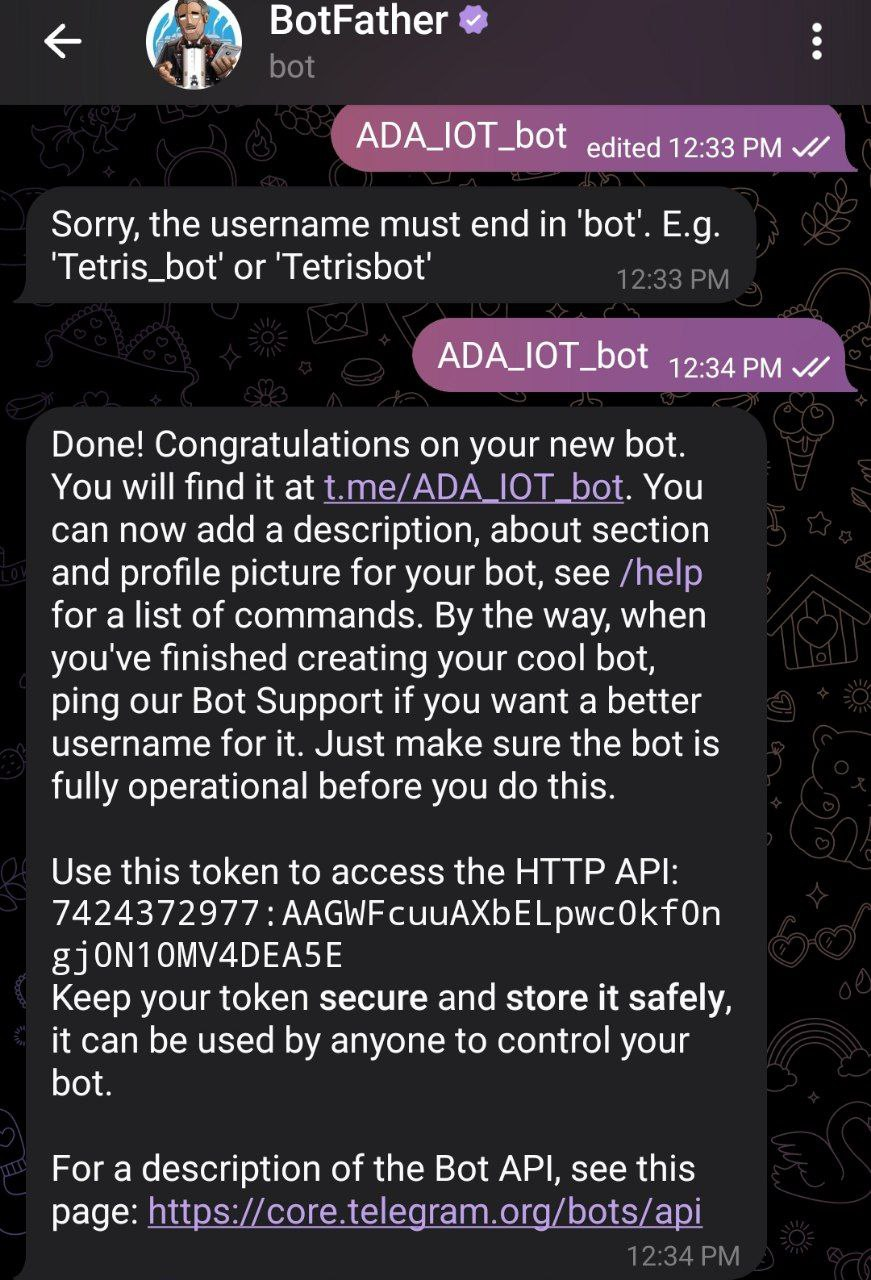
# سوال چهارم:خاموش و روشن کردن یک وسیله منزل از طریق اینترنت با استفاده از ماژول رله

از برنامه ی اتصال از طریق اینترنت در قسمت های قبلی برای برنامه ریزی استفاده کردیم. بجای استفاده از ماژول رله ، از یک رله و ترانزیستور 2N2222 که از قبل داشتیم ، استفاده کردیم.ابتدا تغذیه ی ماژول را با استفاده از منبع تغذیه تامین میکنیم. سپس base ترانزیستور را به پایه ESP متصل کرده و با اتصال مسیر جریان از پایه های Collector و Emitter و اتصال آن به پایه ی تغذیه ی رله (برای فعال سازی آن) ، مسیر کنترل رله را تشکیل دادیم. بعد از پیدا کردن پایه ی NO، آنرا به مقاومت و LED متصل کردیم. در ادامه همه ی GND ها را مشترک کردیم. بعد با ارسال فرمان به ESP، ترانزسیتور رله را فعال کرده و LED روشن میشود.



# سوال ششم: دریافت فرمان از طریق تلگرام

از ربات @BotFather برای ایجاد یک ربات شخصی استفاده کردیم و Token مورد نظر را داخل کد پایتون ربات و کتابخانه ی telebot استفاده کردیم.



**عملکرد کلی ربات:**

## تعریف وظایف اصلی:

* :Update Status نمایش وضعیت کنونی دستگاه و زمان‌بندی تعیین شده.
* Turn On : روشن کردن دستگاه.
* :Turn Offخاموش کردن دستگاه.
* :Set Timingتنظیم زمان‌بندی برای روشن و خاموش شدن دستگاه.

## استفاده از دکمه‌های شیشه‌ای (InlineKeyboard):

برای سهولت کاربران، از دکمه‌های شیشه‌ای جهت انتخاب کارهای مختلف استفاده می‌شود، که این دکمه‌ها با استفاده از InlineKeyboardMarkup ایجاد می‌شوند.

## برقراری ارتباط با سرور از راه دور:

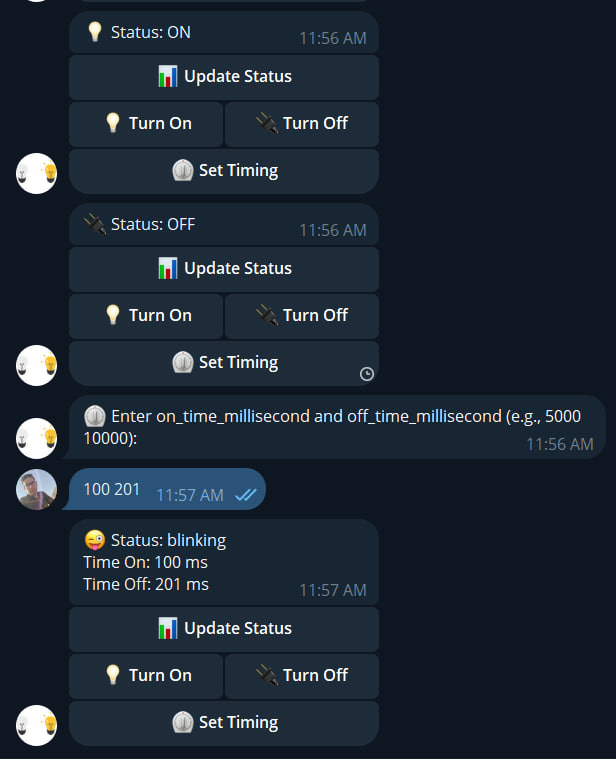
ارتباط با سرور برای ارسال درخواست‌ها به URL‌های مختلف مانند روشن و خاموش کردن دستگاه و تنظیم زمان‌بندی با استفاده از requests.get() انجام می‌شود.

## پردازش دستورات کاربر:

وقتی که کاربر یک دستور از طریق دکمه‌های شیشه‌ای ارسال می‌کند، این دستور توسط handle\_callback پردازش می‌شود و درخواست متناظر به سرور ارسال می‌شود. برای دستور Set Timing، کاربر باید زمان‌های روشن و خاموش کردن دستگاه را به صورت میلی‌ثانیه وارد کند، که این داده‌ها نیز به سرور ارسال می‌شود.

## پیام های خطا:

در صورتی که هنگام ارسال یا دریافت اطلاعات مشکلی به وجود بیاید، پیام‌های خطا به کاربر نمایش داده می‌شود تا اوضاع بهتر مشخص شود.



## **ساختار کلی کد**

کد ربات شامل بخش‌های زیر است:

## وارد کردن کتابخانه‌ها:

* telebot برای ساخت ربات تلگرام.
* Requests برای ارسال درخواست‌های HTTP.
* telebot.types برای استفاده از دکمه‌های شیشه‌ای (Inline Keyboard).

## تعریف متغیرهای اصلی:

* توکن ربات تلگرام.
* آدرس‌های URL برای ارسال درخواست‌های مربوط به روشن و خاموش کردن دستگاه، تنظیم زمان‌بندی و به‌روزرسانی وضعیت.

## تعریف ربات:

ایجاد یک نمونه از کلاس TeleBot با استفاده از توکن.

## ایجاد دکمه‌های شیشه‌ای:

تابع create\_inline\_keyboard برای ایجاد و بازگرداندن دکمه‌های شیشه‌ای.

## مدیریت درخواست‌های کاربر:

تابع handle\_callback برای پردازش درخواست‌های کاربر بر اساس داده‌های برگشتی از دکمه‌های شیشه‌ای.

## پردازش زمان‌بندی:

تابع process\_timing\_step برای دریافت و پردازش زمان‌های وارد شده توسط کاربر جهت تنظیم زمان‌بندی روشن و خاموش شدن دستگاه.

## مدیریت دستورات start و help:

تابع handle\_start\_help برای نمایش پیام خوشامدگویی و دکمه‌های شیشه‌ای به کاربر.

## شروع به کار ربات:

فراخوانی bot.polling() برای شروع دریافت و پردازش پیام‌ها.

## **توضیح توابع و فرآیندها**

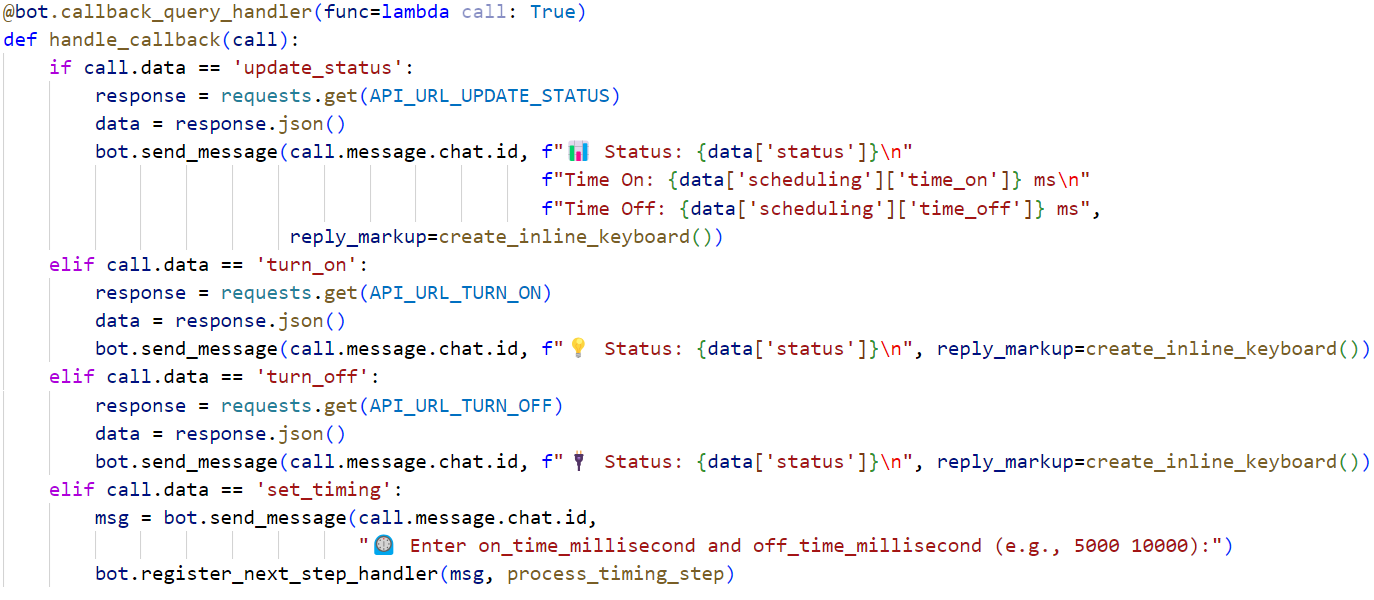
تابعcreate\_inline\_keyboard:

این تابع یک InlineKeyboardMarkup ایجاد می‌کند و دکمه‌های مختلفی را به آن اضافه می‌کند:

* دکمه "Update Status" برای به‌روزرسانی وضعیت دستگاه.
* دکمه "Turn On" برای روشن کردن دستگاه.
* دکمه "Turn Off" برای خاموش کردن دستگاه.
* دکمه "Set Timing" برای تنظیم زمان‌بندی.

## تابعhandle\_callback:

این تابع برای پردازش درخواست‌های کاربر زمانی که یکی از دکمه‌های شیشه‌ای فشرده می‌شود استفاده می‌شود. برای هر درخواست، یک درخواست HTTP به URL متناظر ارسال می‌شود و پاسخ دریافت شده به کاربر نمایش داده می‌شود.

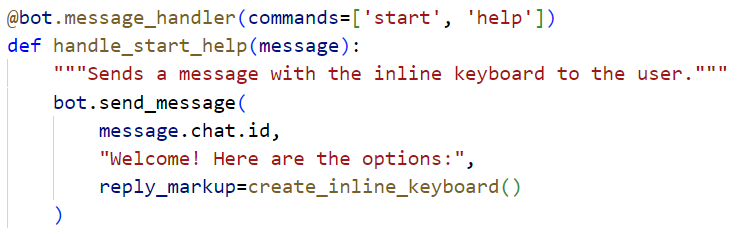


## تابع process\_timing\_step:

این تابع زمان‌های وارد شده توسط کاربر را پردازش می‌کند و آن‌ها را به سرور ارسال می‌کند تا زمان‌بندی روشن و خاموش شدن دستگاه تنظیم شود. در صورت موفقیت‌آمیز بودن درخواست، وضعیت جدید به کاربر نمایش داده می‌شود.

## تابع handle\_start\_help:

این تابع پیام خوشامدگویی و دکمه‌های شیشه‌ای را به کاربر نمایش می‌دهد.



## شروع به کار ربات:

با فراخوانی bot.polling()، ربات شروع به گوش دادن به پیام‌های دریافتی و پردازش آن‌ها می‌کند.

## لینک گیتهاب پروژه:

<https://github.com/adel4780/IOT-Final>

## لینک وب سرور:

<https://danial.pythonanywhere.com/>

## لینک بات: ابتدا فایل برنامه + فیلترشکن اجرا شود

<http://t.me/ADA_IOT_bot>