Intro (Show Final Clock Running)

"Hey everyone 👋, welcome to my channel.
Today I'm really excited to show you a project that looks simple but is very useful — a **digital clock.**

This project is great for beginners who want to learn **Tkinter** for GUI and also how to work with the **time module** in Python.
First, let me explain how the logic behind this app works, and then we'll code it together step by step."

Step 1 – Excalidraw Explanation :

"So in this project, we mainly use **two modules**.

- The time module is used to get the system time and day.
- The tkinter module helps us create the app window and display text on it."

"Next, we have **two functions**.

- The first one is `my_clock()`. This function's job is to **keep updating the time** on the screen. Normally, if you just print time once, it won't change. But here we use a method called `after(1000, my_clock)` which means: after 1 second, `this` fuction will execute again and again. So it keeps refreshing every second, just like a real clock.
- The second one is `what_to_say()`. This function checks the **current hour** and based on that it shows a greeting.
  - If it's morning, it says Good Morning.
  - If it's afternoon, Good Afternoon.
  - Evening, Good Evening.
  - And at night, it shows Good Night."

"Finally, all this information is displayed using **labels in Tkinter**.
We create one label for the **day**, one for the **time**, and one for the **greeting**. These labels are styled with fonts and colors to make the clock look good."

"So if I show you the flow:
The time module gives the data → functions process it → and finally labels display it.
That's the whole logic of this project in just a few steps."

Step 2 – Writing the Code (Line-by-Line Script)

"Okay, now let's open a new Python file and start writing the code step by step."

```
#  import required modules
import time                  # 'time' module helps us to get the current system time and day
import tkinter as tk           # 'tkinter' is used to create the GUI

# now, we will create the main window
root = tk.Tk()               # this will initializes the tkinter window
root.title("Clock")           #will sets the title of the window as "Clock"
root.geometry("600x400")       #will sets the size of the window
root.configure(bg="black")      # makes the background color black
root.resizable(False, False)    # disables resizing so window size stays fixed

# now , we will create the labels
# d_label to display the current day (like Mon, Tue, etc.)
d_label = tk.Label(
    root,
    text="",                #this is temporary text, which we will update later
    font=("Arial", 20, "bold"), #To set font style and size
    fg="lightgreen",            #To set text color to light green
    bg="black"                  # To set background color same as window
)
d_label.place(x=30, y=55)       # place the label at a fixed position (x=30, y=55)

# create a label to display the digital clock (time)
time_label = tk.Label(
    root,
    text="",                    # start with empty, will update later
    font=("Arial", 65, "bold"),  # large font for time
    fg="lightgreen",             # text color light green
    bg="black",                  # background black
    relief="ridge"               # gives a bordered effect around time
)
time_label.pack(pady=100)       # pack the label and add vertical padding for spacing

# create a label for greetings like Good Morning / Afternoon etc.
wts_label = tk.Label(
    root,
    text="",                    # initially empty, will update later
    font=("Arial", 25, "bold"),
    fg="lightgreen",
    bg="black"
)
wts_label.place(x=150, y=250)    # place the greeting label near bottom of the window

#now, we will write function to keep updating the clock every second
def my_clock():
    day = time.strftime("%a")                # get current day name (like Mon, Tue, Wed)
    d_label.config(text=day)                 # updates day label with current day

    current_time = time.strftime("%H : %M : %S") # get current time in hours:minutes:seconds format
    time_label.config(text=current_time)      # update time label with current time

    # root.after() runs the function again after given milliseconds
    # here we call my_clock again after 1000ms = 1 second
    root.after(1000, my_clock)

# now we will move to the next function to decide greeting based on current hour
def what_to_say():
    current_hr = int(time.strftime("%H"))      # get current hour in 24-hour format
    if 6 < current_hr < 12:                # if time is between 7am to 11am
        wts_label.config(text="Good Morning")
    elif 12 <= current_hr < 18:            # if time is between 12pm to 5pm
        wts_label.config(text="Good Afternoon")
    elif 18 <= current_hr < 20:            # if time is between 6pm to 7:59pm
        wts_label.config(text="Good Evening")
    else:                          # for all other times (night)
        wts_label.config(text="Good Night")

# call the greeting function once at start
what_to_say()

# call the clock function to start updating time
my_clock()

# start the tkinter event loop so the window keeps running
root.mainloop()
```

Step 3 – Run the Code

🎙️:
"Now let's run the program… and here's our digital clock!
The time is updating every second, today's day is visible, and the greeting also changes based on the time. That means everything is working perfectly."

🎬 Outro

🎙️:
"So that's how you can build your own digital clock app in Python.
It's a fun project, and you can improve it further by adding date, changing themes, or even making it an alarm clock.

I'll also be uploading a video where I explain the **time module and Tkinter basics** in detail, so you can use them confidently in your own projects.

Make sure to **like this video, subscribe to my channel, and drop a comment** if you enjoyed this tutorial.
Thanks for watching, see you in the upcoming video 👋."