NAME: TEJA SINGURU

COURSE:PYTHON(3MONTHS)INTERNSHIP

DEVTERN TYPING TEST PYTHON PROJECT

INTRODUCTION :

1. OBJECTIVE OF THE TASK
    - This typing test python project has to be buid in python
    - We use GUI applications also to build graphical interface
    - This game makes user to practice the typing speed
    - This projet is so fun and interesting to make it
2. RELEVANCE
    - This project is practicle
    - We can use it to improve user own typing skills
    - It can be modified as per needs
    - Using tkinter to create graphical user interfaces

TASKDESCRIPTION :

3. PROBLEM STATEMENT
    - This typing test python project is build for own typing speed development
    - In this project it displays words on the screen for the user to type
    - This game gives user 60 seconds to type as many words as possible
    - It tracks users accuracy and speed
    - Displays the final score, including the number of wrong words entered
4. REQUIREMENTS
    - I need some more basic knowledge on GUI application
    - I am currently improving my python coding skill

METHODOLOGY:

5. APPROACH

- Firstly we need to install tkinter in visual studio
- By pip install tk
- After I followed the code in given task vedio
- I learned it and implemented successfully

6. TOOLS & TECHNOLOGIES

- I build complete project in visual studio
- And installed tkinter to make GUI application
- Used some of google resources to build knowledge on it

IMPLEMENTATION:

7.CODE/ALGORITHM DETAILS

```
from tkinter import *
import random
from tkinter import messagebox


words = ['example', 'typing', 'test', 'quick', 'brown', 'fox', 'jumps',
'over', 'lazy', 'dog', 'python', 'programming', 'developer', 'keyboard',
'interface', 'function', 'variable', 'loop', 'condition', 'algorithm']

Mainscreen = Tk()
Mainscreen.geometry('800x600')
Mainscreen.title('Typing Test By Teja Singuru')
Mainscreen.config(bg="skyblue")

score = 0
missed = 0
time = 60
count1 = 0
movingwords = ''

def movingtest():
    global count1, movingwords
    floatingtext = 'Typing Test By Teja Singuru'
    if count1 >= len(floatingtext):
        count1 = 0
        movingwords = ''
    movingwords += floatingtext[count1]
```

```
    count1 += 1
    findlabels.configure(text=movingwords)
    findlabels.after(150, movingtest)

def giventime():
    global time, score, missed
    if time > 0:
        time -= 1
        timercount.configure(text=time)
        timercount.after(1000, giventime)
    else:
        gameinstruction.configure(text=f'Hit={score} | Miss={missed} | Total
Score={score - missed}')
        rr = messagebox.askretrycancel('Notification', 'Do you want to play
again?')
        if rr:
            reset_game()

def reset_game():
    global score, missed, time
    score = 0
    missed = 0
    time = 60
    timercount.configure(text=time)
    update_word()
    scorelabelcount.configure(text=score)
    wordentry.delete(0, END)
    gameinstruction.configure(text='Hit enter button after typing the word')
    startlabel.configure(text='Start Typing')
    wordentry.focus_set()

def game(event):
    global score, missed
    if time == 60:
        giventime()
    gameinstruction.configure(text='')
    startlabel.configure(text='')
    entered_word = wordentry.get().strip()
    current_word = labelforward['text']
    if entered_word.lower() == current_word.lower():
        score += 1
        scorelabelcount.configure(text=score)
    else:
        missed += 1
        scorelabelcount.configure(text=score)
    update_word()
    wordentry.delete(0, END)
```

```
def update_word():
    global words, labelforward
    labelforward.configure(text=random.choice(words))


findlabels = Label(Mainscreen, text='', font=('arial', 20, 'italic bold'),
bg="skyblue", fg="black")
findlabels.place(x=250, y=150)

startlabel = Label(Mainscreen, text='Start Typing', font=('arial', 30, 'italic
bold'), bg='black', fg='white')
startlabel.place(x=275, y=50)

labelforward = Label(Mainscreen, text=random.choice(words), font=('arial', 45,
'italic bold'), fg='green')
labelforward.place(x=250, y=240)

scorelabel = Label(Mainscreen, text='Your Score:', font=('arial', 25, 'italic
bold'), fg='red')
scorelabel.place(x=10, y=100)

scorelabelcount = Label(Mainscreen, text=score, font=('arial', 25, 'italic
bold'), fg='blue')
scorelabelcount.place(x=150, y=180)

labelfortimer = Label(Mainscreen, text='Time Left:', font=('arial', 25,
'italic bold'), fg='red')
labelfortimer.place(x=600, y=100)

timercount = Label(Mainscreen, text=time, font=('arial', 25, 'italic bold'),
fg='blue')
timercount.place(x=600, y=180)

gameinstruction = Label(Mainscreen, text='Hit enter button after typing the
word', font=('arial', 25, 'italic bold'), fg='grey')
gameinstruction.place(x=150, y=500)

wordentry = Entry(Mainscreen, font=('arial', 25, 'italic bold'), bd=10,
justify='center')
wordentry.place(x=250, y=330)
wordentry.focus_set()

Mainscreen.bind('<Return>', game)

movingtest()

mainloop()
```

EXPALNATION OF THE ABOVE CODE :

- **Imports and Setup**
  - **Modules Imported**:
    - `tkinter`: For creating the GUI.
    - `random`: To select random words from the list.
    - `messagebox` from `tkinter`: To display dialog boxes for user interactions.
  - **Main Window (`Mainscreen`)**:
    - Size: 800x600 pixels.
    - Title: "Typing Test By Teja Singuru".
    - Background Color: Sky blue.

- **Global Variables**
  - `words`: A list of words used in the typing test.
  - `score`: Tracks the number of correct entries.
  - `missed`: Tracks the number of incorrect entries.
  - `time`: Countdown timer set to 60 seconds.
  - `count1` & `movingwords`: Used for creating a moving text effect.
  - `timer_started`: A flag to ensure the timer starts only once.

- **Functions**
  - **`movingtest()`**:
    - Creates a scrolling text effect for the label `findlabels`.
    - Continuously updates the label to display moving text.
  - **`giventime()`**:
    - Manages the countdown timer.
    - Updates the timer display every second.
    - Ends the game when time reaches zero, displaying the user's performance and prompting to retry.
  - **`reset_game()`**:
    - Resets all game variables (`score`, `missed`, `time`).
    - Updates the GUI to its initial state for a new game session.
  - **`game(event)`**:
    - Triggered when the user presses the Enter key.
    - Starts the timer on the first entry.
    - Checks if the entered word matches the displayed word.
    - Updates `score` or `missed` accordingly.
    - Provides visual feedback by changing the word color.
    - Selects a new word for the next entry.
  - **`update_word()`**:
    - Selects a new random word from the `words` list.
    - Removes the selected word to prevent immediate repetition.
    - Resets the `words` list once all words are used.

- **GUI Components**

  - **Labels**:
    - `findlabels`: Displays the moving text animation.
    - `startlabel`: Initial prompt "Start Typing".
    - `labelforward`: Shows the current word to type.
    - `scorelabel` & `scorelabelcount`: Display the user's score.
    - `misslabel` & `misscount`: Display the number of missed entries.
    - `labelfortimer` & `timercount`: Show the remaining time.
    - `gameinstruction`: Provides instructions to the user.
  - **Entry Field**:
    - `wordentry`: Where the user types the displayed word.
    - Automatically focused for immediate input.
  - **Event Binding**:
    - The Enter key (`<Return>`) is bound to the `game` function to submit entries.

- **Application Flow**

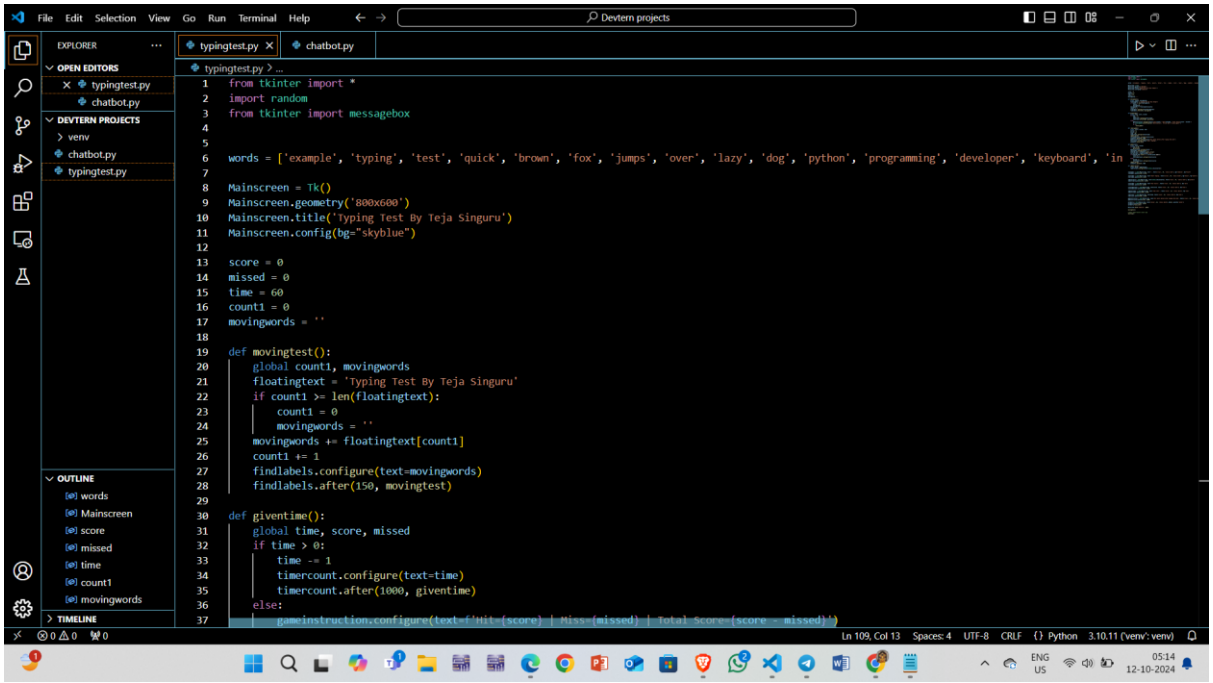
  1. **Initialization**:
     - Main window setup with labels and entry field.
     - Moving text animation starts.
  2. **Gameplay**:
     - User types the displayed word and presses Enter.
     - Timer starts on the first entry.
     - The application checks the correctness of the input.
     - Scores are updated, and new words are displayed.
  3. **End of Game**:
     - When the timer runs out, performance statistics are shown.
     - User is prompted to retry or exit.
  4. **Reset**:
     - If the user chooses to retry, the game resets to its initial state.

8.CHALLENGES AND SOLUTIONS

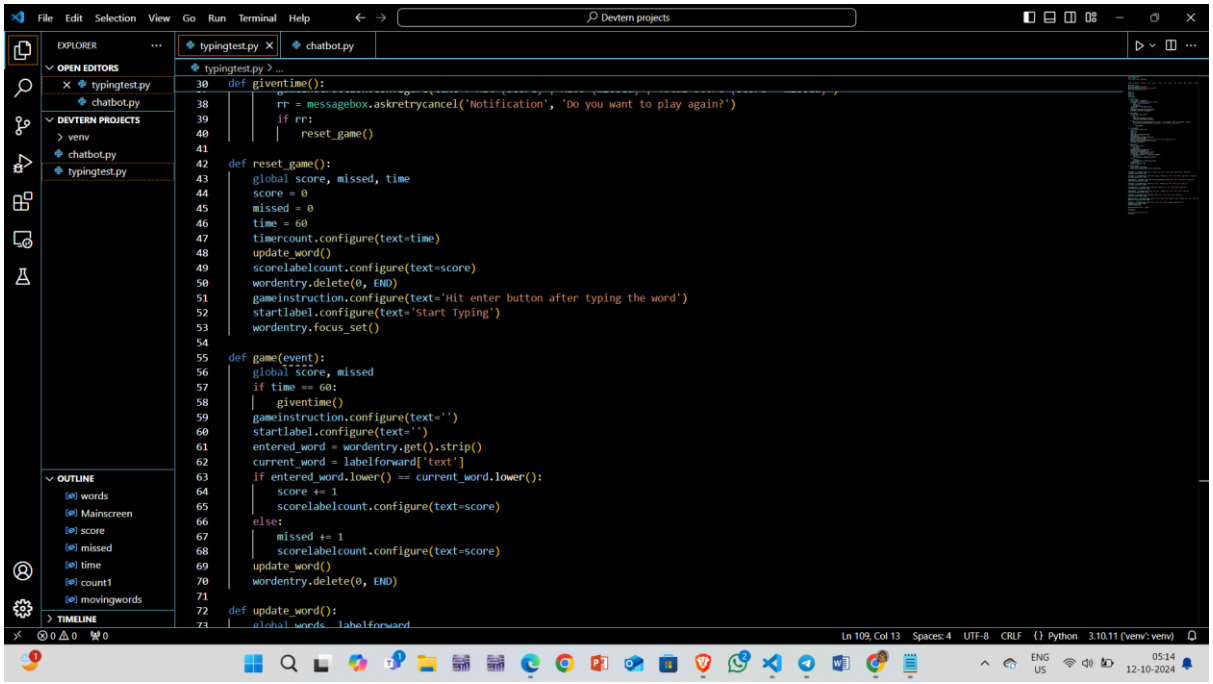
- As I have minor typing errors and code rules
- But as I go through the code again I resolvd witht the basic python knowledge

WORKING CONDITIONS :

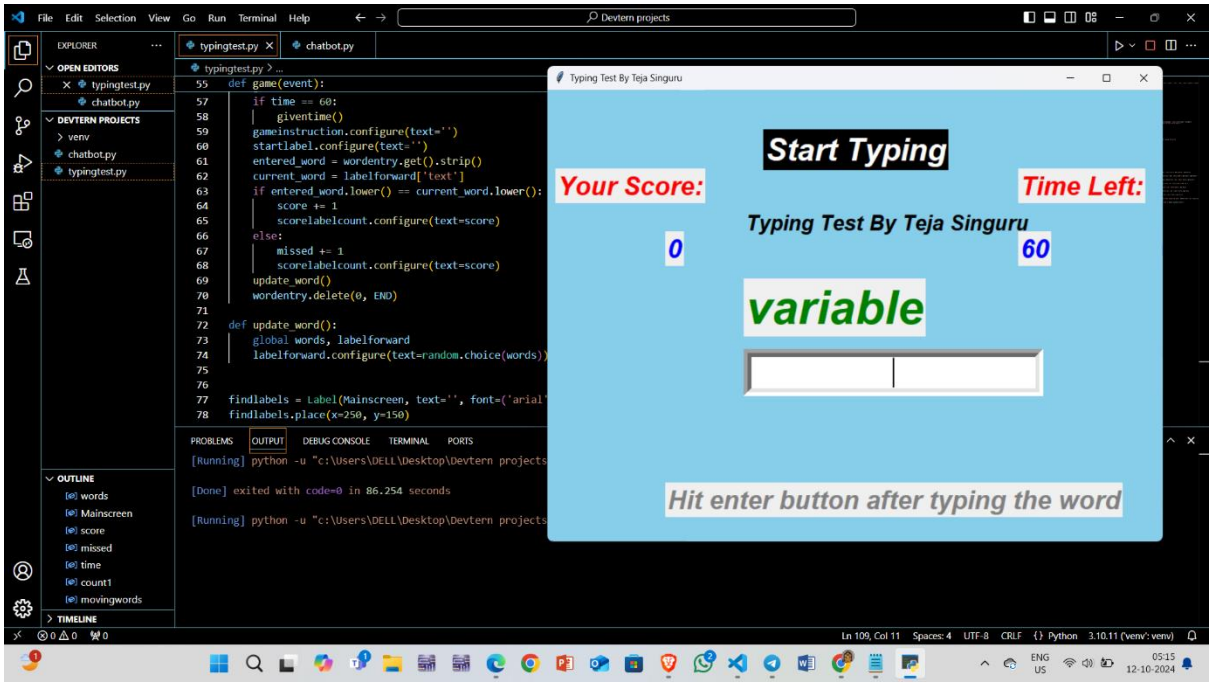
9.RESULTS

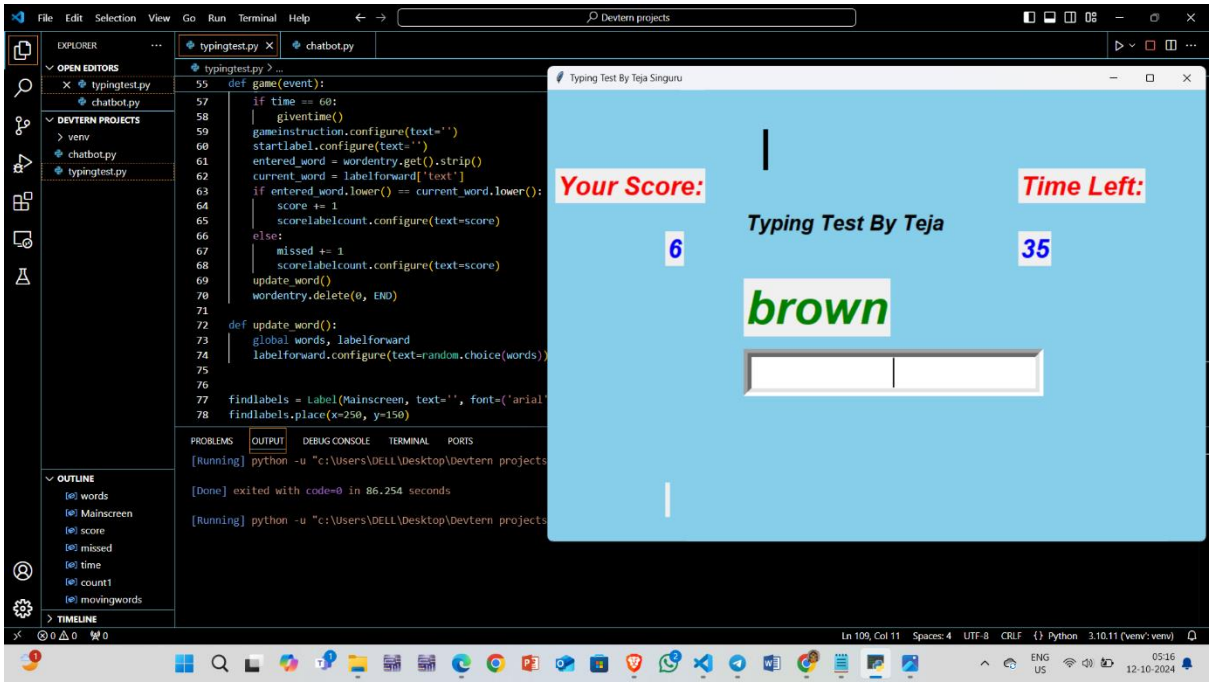


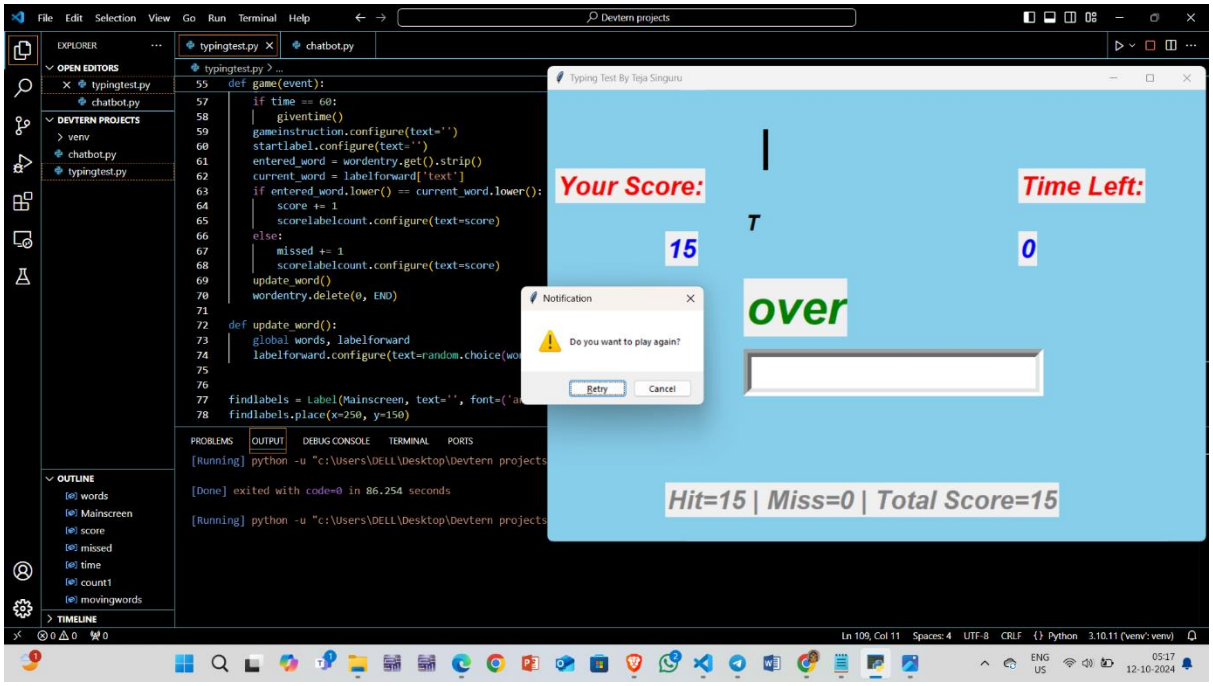
```python
from tkinter import *
import random
from tkinter import messagebox


words = ['example', 'typing', 'test', 'quick', 'brown', 'fox', 'jumps', 'over', 'lazy', 'dog', 'python', 'programming', 'developer', 'keyboard', 'in

Mainscreen = Tk()
Mainscreen.geometry('800x600')
Mainscreen.title('Typing Test By Teja Singuru')
Mainscreen.config(bg="skyblue")

score = 0
missed = 0
time = 60
count1 = 0
movingwords = ''

def movingtest():
    global count1, movingwords
    floatingtext = 'Typing Test By Teja Singuru'
    if count1 >= len(floatingtext):
        count1 = 0
        movingwords = ''
    movingwords += floatingtext[count1]
    count1 += 1
    findlabels.configure(text=movingwords)
    findlabels.after(150, movingtest)

def giventime():
    global time, score, missed
    if time > 0:
        time -= 1
        timercount.configure(text=time)
        timercount.after(1000, giventime)
    else:
        gameinstruction.configure(text=f"Hit={score} | Miss={missed} | Total Score={score - missed}")
```



```python
def giventime():
        rr = messagebox.askretrycancel('Notification', 'Do you want to play again?')
        if rr:
            reset_game()

def reset_game():
    global score, missed, time
    score = 0
    missed = 0
    time = 60
    timercount.configure(text=time)
    update_word()
    scorelabelcount.configure(text=score)
    wordentry.delete(0, END)
    gameinstruction.configure(text='Hit enter button after typing the word')
    startlabel.configure(text='Start Typing')
    wordentry.focus_set()

def game(event):
    global score, missed
    if time == 60:
        giventime()
    gameinstruction.configure(text='')
    startlabel.configure(text='')
    entered_word = wordentry.get().strip()
    current_word = labelforward['text']
    if entered_word.lower() == current_word.lower():
        score += 1
        scorelabelcount.configure(text=score)
    else:
        missed += 1
        scorelabelcount.configure(text=score)
    update_word()
    wordentry.delete(0, END)

def update_word():
    global words, labelforward
```


Typing Test By Teja Singuru
Start Typing
Your Score:
Time Left:
Typing Test By Teja Singuru
0
60
variable
Hit enter button after typing the word


Typing Test By Teja Singuru
Your Score:
Time Left:
Typing Test By Teja
6
35
brown



CONCLUSION:

10.SUMMARY

- This code creates a typing test game where users try to type words as they appear on the screen. The game tracks the score, missed attempts, and time left, and provides an option to play again when the time runs out.
- Using tkinter in python

11.LEARNINGS

- Overall, this code provides a good introduction to building interactive applications with Tkinter.
- It demonstrates fundamental GUI programming concepts, including widget creation, event handling, and timer-based updates.
- The project also offers insights into game development basics, such as state management, user input validation, and game flow control.

APPENDICES:

How to Make a Typing Speed Tester with Tkinter in Python - The Python Code

https://thepythoncode.com/article/how-to-make-typing-speed-tester-in-python-using-tkinter