

COMPUTER SCIENCE

INVESTIGATORY

PROJECT



DELHI PUBLIC SCHOOL, WHITEFIELD

UNDER THE AEGIS OF THE DELHI PUBLIC SCHOOLS SOCIETY, DELHI
AFFILIATED TO CBSE, NEW DELHI


PRAXY FERNANDES
MEMORIAL TRUST

Delhi Public School Whitefield

BY:

SOHAN VARIER(ROLL NO. 31)

AND

ADHVAITH RAJESH(ROLL NO.3)

CERTIFICATE

This is to certify that the Project entitled Basic TimeTable Generator is a bonafide work done by _____ of class XII Session 20-21 in partial fulfillment of CBSE's AISSCE Examination 2021 and has been carried out under my direct supervision and guidance. This report or a similar report on the topic has not been submitted for any other examination and does not form a part of any other course undergone by the candidate.

.....
Signature of Student

.....
Signature of Teacher/Guide Name:

Ms.Monika Kohli and Pavithra NR

Roll No.:

Design.: PGT Comp.Sc.

ACKNOWLEDGEMENT

I would like to express my special thanks to my teacher Mrs. Pavithra NR as well as our principal Mrs. Kamava Bopana who gave me the golden opportunity to do this wonderful project on the topic Timetable Generator. It helped me learn and apply new concepts that I otherwise would not have had the chance to. I am really thankful to them.

Secondly, I would also like to thank my parents, and friends who were indispensable to the completion of this project within the limited time frame.

INDEX

1.Introduction

2.Objective and Scope of the Project

3.Theoretical Background

4. Event Coding

5.Output

6.Bibliography

INTRODUCTION

This software project is developed to create timetables with ease. It takes in a specific set of values including number of days, periods etc. and generates a timetable to be available for download in JPG format.

The generator consists of a main front end GUI programmed using the Tkinter module of python. The generated timetable is displayed in a new popup window. The timetable can also be saved directly to your desktop.

It is simple in design and does not require training to use. Any person regardless of age can use it as long as they are old enough to understand.

OBJECTIVE AND SCOPE OF THE **PROJECT**

The objective of this project is to make the creation timetables easier for schools and colleges. Just by entering a few relevant sets of data, a time table can be generated within seconds, which can be available for download in JPG format. This saves time and resources of the administration of institutions, who could use that time for more important purposes.

SYSTEM IMPLEMENTATION

Hardware used -

- 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz
- 16 GB RAM
- Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz
- 8 GB RAM

Software Required -

- Python 3.9
- Pillow, Tabulate, Tkinter, Itertools, Random modules should be downloaded.

THEORETICAL BACKGROUND

This project was coded on Python 3.9, and multiple modules were used. The modules used were Itertools, Pillow, Random, Tkinter and Tabulate.

- 1) Itertools - itertools is an inbuilt module of python that provides common iterative functions that are memory efficient and time saving. The functions provided are basic iteration tools that are used commonly across programming projects. Some examples of itertools functions are - count(), cycle(), chain(), permutations(), combinations() etc. In this project, permutations() and combinations() have been used.
- 2) Pillow - Pillow is a fork of PIL (Python Image Library). It provides us with multiple image processing capabilities. It can be used to manipulate images of a large variety of image file types. For example, using Pillow, the file type of images can be changed, the image can be rotated, and various other such functions can be performed. In this program, PIL is used to make the final timetable in an image form.

- 3) Random - Random function is a very commonly used module in python programming. It's functions return random numbers based on what's required. Examples of functions are `randrange()`, `randint()`, etc. In this project, we have made use of the `sample()` function, which returns a specified number of random selections from a sequence.
- 4) Tkinter - Tkinter is Python's de-facto standard GUI(Graphical User Interface) package. Using Tkinter, the root window for the application is developed which includes all the necessary input and output fields and buttons. In this project, Tkinter is used to create the window where the timetable generator program is displayed along with the required buttons and input fields.
- 5) Tabulate - Tabulate is a python module that helps tabulating data, with a simple function call. It contains different table formats which can be used as required. In this project, we have made use of the `tabulate()` function to tabulate the information entered by the user and make it into a timetable format.

EVENT CODING

main.py :

```
from tkinter import * #importing tkinter
from PIL import ImageTk, Image
from timetable import timetable_main as tt
from download import download_button
import os.path

#root window
root = Tk()
root.title('Timetable Generator')
root.configure(bg = 'black')

#functions
def display_timetable(): #function to open a pop-up window

    days = int(daysEntry.get()) # receives values entered by user in the GUI
    subjects = (subjectsEntry.get()).split(',')
    periods = len(subjects)

    if days > 7:
        display_error()
    else:
        pop_up = Toplevel() # pop-up window for timetable
        pop_up.title('Timetable Generator')
        pop_up.configure(bg = 'black')
        pop_up.grid()

        tt(days,periods,subjects)

        global timetable_img
        timetable_img = ImageTk.PhotoImage(Image.open("tt-new.jpg"))

        image_label = Label(pop_up, image = timetable_img)
        image_label.grid(row = 0, column = 0)

def display_instructions(): #pop-up window for displaying instructions
    pop_up2 = Toplevel()
    pop_up2.title('Timetable Generator Instructions')
    pop_up2.configure(bg = 'black')
    pop_up2.grid()
    f1 = open('Instructions.txt', 'r')
    global text_label
```

```

    text_label = Label(pop_up2, text = f1.read(), fg = 'white', bg = 'black', font =
('Arial', 12))
    text_label.grid(row = 1, column = 1)

def display_error():    # pop-up window to display error
    pop_up3 = Toplevel()
    pop_up3.title('Error Message!')
    pop_up3.configure(bg = "black")
    pop_up3.grid()
    f2 = open('error.txt','r')
    global text_label2
    text_label2 = Label(pop_up3, text = f2.read(), fg = "white", bg = "black", font =
('Arial', 12))
    text_label2.grid(row = 1, column = 1)

#Labels definitions
daysLabel = Label(root, text = 'Enter number of days:', fg = 'white', bg = 'black', font =
("proxima nova", 12))

subjectsLabel = Label(root, text = 'Enter the subjects separated by a comma:', fg =
'white', bg = 'black', font = ("proxima nova", 12))

#Buttons
generate = Button(root, text = 'Generate Timetable', width = 25, bg = 'green', fg =
'white', font = ("uni sans",12), command = display_timetable)

download = Button(root, text = 'Download TimeTable to Desktop',width = 25,  bg = 'green',
fg = 'white', font = ("uni sans",12), command = download_button)

quit = Button(root, text = 'QUIT', width = 25, bg = 'red', fg = 'white', font = ("uni
sans",12), command = root.destroy)

instructions = Button(root, text = 'Instructions', width = 25, bg = 'green', fg = 'white',
font = ("uni sans",12), command = display_instructions)

#taking input
daysEntry = Entry(root, text = 'Enter number of days', width = 25, fg = 'black', bg =
'grey', bd = 5, font = ('Arial',12))

subjectsEntry = Entry(root, text = 'Enter the subjects separated by a comma', width = 25,
fg = 'black', bg = 'grey', bd = 5, font = ('Arial',12))

#formatting the page
daysLabel.grid(row = 0, column = 0)

daysEntry.grid(row = 0, column = 1, padx = 35, pady = 10, columnspan = 1)

subjectsLabel.grid(row = 1, column = 0)

subjectsEntry.grid(row = 1, column = 1, padx = 35, pady = 10, columnspan = 1)

instructions.grid(row = 3, column = 0, padx = 20, pady = 20, columnspan = 1)

```

```
generate.grid(row = 2, column = 0, padx = 20, pady = 20, columnspan = 1)

download.grid(row = 2, column = 1, padx = 20, pady = 20, columnspan = 1)

quit.grid(row = 3, column = 1, padx = 20, pady = 20, columnspan = 1)

root.mainloop()
```

timetable.py :

```
import itertools as it
import random
from tabulate import tabulate
from image_gen import gen_img as img

def timetable_main(days,subject_n,subjects):

    perm = list(it.permutations(subjects)) # all possible arrangements of periods in a
list
    if len(perm)<days: # in case the number of permutations are less than number of days
        perm *= days

    days_of_the_week = ["Monday", "Tuesday",
"Wednesday", "Thursday", "Friday", "Saturday", "Sunday"] #list that stores days of the week
    head = ["Day"]
    for i in range(1,subject_n + 1):
        head.append(i)

    final = random.sample(perm,days) # choosing 5 random arrangements (depending on number of working days)
    for j in range(days):
        final[j] = (days_of_the_week[j],)+final[j]

    table = tabulate(final, headers= head, tablefmt="grid") # putting it into table form
    img(table)
```

image_gen.py :

```
from PIL import Image, ImageFont, ImageDraw

def gen_img(table):      #function to draw the table on the blank image
    img = Image.open("blank.jpg")
    font = ImageFont.truetype("font/Anonymous_Pro/AnonymousPro-Regular.ttf", 34)

    draw = ImageDraw.Draw(img)

    text = table

    draw.multiline_text((50,50), text,fill = "black", font=font)

    img.save("tt-new.jpg")
```

download.py :

```
from PIL import Image
import os.path

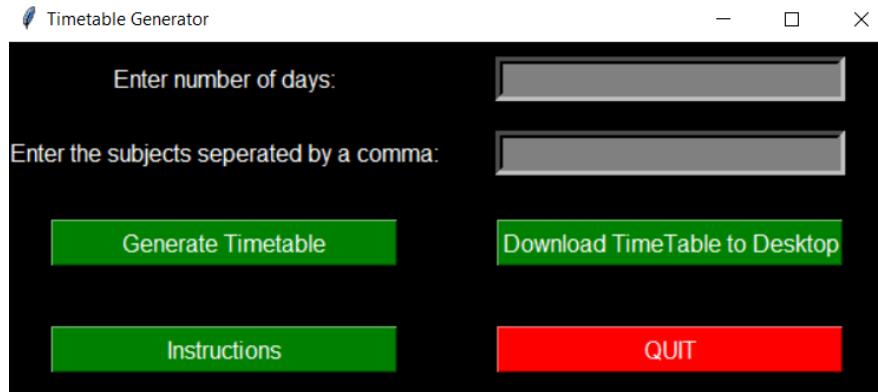
def download_button():    # function to save timetable image on desktop
    img = Image.open("tt-new.jpg")

    file_name = "timetable.jpg"
    complete_name = os.path.join(os.path.expanduser('~'),'Desktop', file_name) # fetches
full path for 'Desktop' on any user

    img.save(complete_name)
```

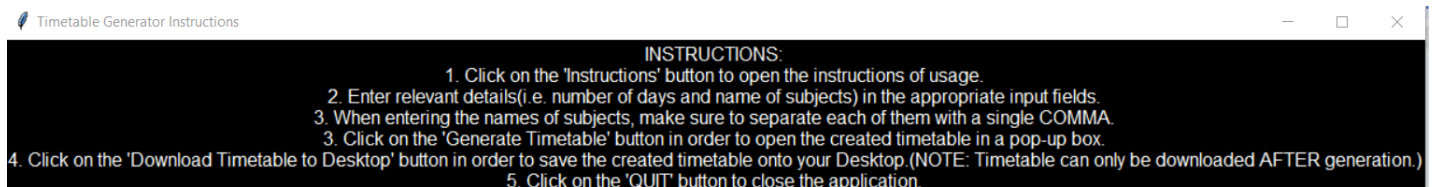
OUTPUT

GUI Front page -



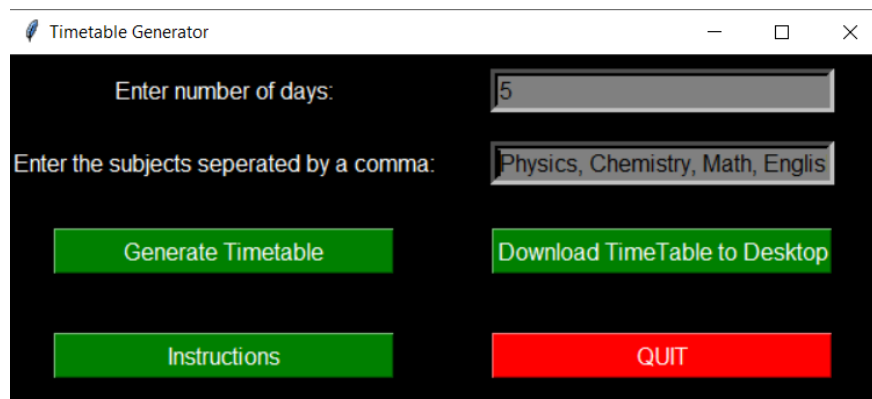
The screenshot shows the main window of the 'Timetable Generator' application. It has a black background with white text. At the top, the title bar reads 'Timetable Generator' with standard window controls. The main area contains two input fields: 'Enter number of days:' and 'Enter the subjects seperated by a comma:'. Below these are four buttons: 'Generate Timetable' (green), 'Download TimeTable to Desktop' (green), 'Instructions' (green), and 'QUIT' (red).

Instructions popup -



The screenshot shows a 'Timetable Generator Instructions' popup window. It has a black background with white text. The title bar reads 'Timetable Generator Instructions'. The main content is titled 'INSTRUCTIONS:' and lists five steps: 1. Click on the 'Instructions' button to open the instructions of usage. 2. Enter relevant details(i.e. number of days and name of subjects) in the appropriate input fields. 3. When entering the names of subjects, make sure to separate each of them with a single COMMA. 4. Click on the 'Generate Timetable' button in order to open the created timetable in a pop-up box. 5. Click on the 'Download Timetable to Desktop' button in order to save the created timetable onto your Desktop. (NOTE: Timetable can only be downloaded AFTER generation.)

Enter the relevant details -



The screenshot shows the 'Timetable Generator' application window with input details. The title bar reads 'Timetable Generator'. The 'Enter number of days:' field now contains the value '5'. The 'Enter the subjects seperated by a comma:' field now contains the text 'Physics, Chemistry, Math, Englis'. The buttons remain the same: 'Generate Timetable' (green), 'Download TimeTable to Desktop' (green), 'Instructions' (green), and 'QUIT' (red).

Generated timetable pops up -

Day	1	2	3	4	5
Monday	Physics	Chemistry	English	Math	Computers
Tuesday	English	Computers	Chemistry	Math	Physics
Wednesday	English	Math	Computers	Physics	Chemistry
Thursday	Physics	Computers	English	Chemistry	Math
Friday	Physics	Math	Computers	Chemistry	English

BIBLIOGRAPHY

1. Tabulate module info - <https://github.com/astanin/python-tabulate>
2. Itertools module info - <https://www.geeksforgeeks.org/permutation-and-combination-in-python/>
3. Tkinter module info - <https://docs.python.org/3.9/library/tkinter.html#module-tkinter>
4. Python documentation - <https://docs.python.org/3/>