**ARTIFICIAL INTELLIGENCE**

**INT - 404**

PROJECT:- INTELLIGENT TIME-TABLE PREPARATION

Submitted By:-

|  |  |  |
| --- | --- | --- |
| NAME | REGISTRATION NO. | ROLL NO. |
| P Smithin Reddy | 11807366 | A17/K18GT |
| Srijan Singh | 11807017 | A19/K18GT |
| Maddala Narmada | 11807192 | A20/K18GT |

Submitted to:- Amandeep Kaur Mam

**Format Of The Project Report**

1. Format Of Report
2. Abstract
3. Introduction
4. Objectives
5. Methodology
6. Results
7. Conclusion
8. Future Scope
9. GitHub Link
10. **Abstract:**

This project defines and says that the total fundamental basics we used in the implementation of this application. And we approach successfully and creates a time-table according to needs we provide. The overview and outlook of this project makes the student to develop a creative planning task in the further. By modifying this application we can also able to management and create the different types of time-tables according to department. And here we can highlight the process to approach of the project.

1. **Introduction:**

For any university or college or institution to run successfully the main theme should be a good time-table management and make a reliable.

Here time-table management evolves an approach in which involves the subjects list and duration of each hour. The university can provide their related subjects in the list and hour after that it going to give a time-table. And the best example we can discuss is the LPU here all streams are well- planned and time-table is provided in every field respective like classes and examinations, scrutiny. With the help of these management we can minimize the work to the humans.

1. **Objectives:**

The objective we can achieve with this project is the output comes from this one a timetable in which there is an subjectslist and duration hour. With the help of providing a max hour to each subject we can distinguish each subjects and provided with specific hour. And also it is very helpful. By mentioning the student id and faculty id it can give us a time-table.

1. **Methodology:**

In the time-table preparation we have written a programme in such a way that that the user will elated and can able to have the time-table in a perfect manner.

Here first we have to provide the total days in a week and subjects in the list in which we want and duration of hour to each class is provided.

And here we are using the csv file and pathlib in the code in which we used to get the project.

And here we will show the Excel file which represents in an effective way and we show in that days in excel file and duration of each hour of the subject.

And we are using an inbuilt package of sys in python of the command line argument.

The main thing of any project is allocation that means a perfect allocation of each subject to that related hour and well planned task.

Coding in python helps us a well mannered readable structure. And AI is suitable for implementing the any project code.

And here the main part which is def to define the definition and here we are giving the maximum hour to the subject.

And for saving the time-table we can create and save into csv file.

And by defining by the variables like subjectslist and start hour and next hour we can start the code.

1. **Results:**

Implementation of code:

import pathlib

import csv

# Let's start by defines variables

subjects\_list = []

start\_hour = 8 # school start at 8.am

next\_hour = 9 # 1rst next hour is 9.am

school\_days = [

'monday',

'tuesday'

]

time\_slot\_list = [] # get list of time slot

subject\_per\_slot = {}

MAX\_HOUR\_PER\_SUBJECT = 6 # use capital letter because it's a constant variable

subject\_hour\_count = {}

def fill\_in\_subjects\_list():

"""Ask user subjects and fill in subjects list"""

enter\_another\_subject = True

while enter\_another\_subject:

subject = input('Type another subject: ')

subject = subject.capitalize()

if not subject in subjects\_list:

subjects\_list.append(subject)

subject\_hour\_count[subject] = MAX\_HOUR\_PER\_SUBJECT

else:

print(f'You\'ve already type {subject} in list.')

question = input('Enter another subject (type "n" to exit)?')

if question.lower() == 'n':

enter\_another\_subject = False

def fill\_out\_subjects\_list():

"""Ask user subjects and fill in subjects list"""

subjects = input('Type all subjects you want add in subjects list\

and separate them by comma: ') # we collect all subjects

the\_subjects = subjects.replace(', ', ',') # remove space after comma

# Split all subjects in order to put them into a list

the\_subjects = the\_subjects.split(',')

for subject in the\_subjects:

subject = subject.capitalize()

if not subject in subjects\_list:

subjects\_list.append(subject)

subject\_hour\_count[subject] = MAX\_HOUR\_PER\_SUBJECT

def ask\_hour():

"""Ask hour to user"""

print(f'Subjects list: {subjects\_list}')

print(f'Planning time: {start\_hour}h-{next\_hour}h')

user\_answer = input('What\'s subject do you want put here? ')

return user\_answer

def fill\_in\_timetable():

"""Display an hour & ask user which subject he want to put there"""

global start\_hour

global next\_hour

for day in school\_days:

# Reset start and next hour

the\_hour = {}

time = 0

start\_hour = 8 # we suppose that school start at 8.am

next\_hour = 9

print('\n---------------------------')

print(f'{day.capitalize()} timetable')

print('---------------------------\n')

while time < 4: # Suppose we've 4hours course/day (you can change it)

hour\_format = f'{start\_hour}h-{next\_hour}h' # format time slot

# it's represent 8 hours/per day for school

if time == 2: # if it's a midday (12.am), make a break

# Add a break in timetable with 'Break time' as inscription

subject\_per\_slot[hour\_format] = ['Break time']

# Add hour format while making sure we avoid duplicate

if not hour\_format in time\_slot\_list:

time\_slot\_list.append('hour\_format')

else:

chosen\_subject = ask\_hour().capitalize()

print(f'start\_hour: {start\_hour}')

print(f'next\_hour: {next\_hour}')

# Check that subject chosen by user is in subjects list

while not chosen\_subject in subjects\_list:

print(f'{chosen\_subject} is not in subjects list.')

print('Choose another subject.')

chosen\_subject = ask\_hour().capitalize()

# Add hour format while making sure we avoid duplicate

if not hour\_format in time\_slot\_list:

time\_slot\_list.append(hour\_format)

subject\_per\_slot[hour\_format] = [chosen\_subject]

else:

subject\_per\_slot[hour\_format] += [chosen\_subject]

# Check that chosen subject max hours didn't reached

for subject, max\_hour in subject\_hour\_count.items():

if chosen\_subject == subject:

# remove one hour on subject max hour

subject\_hour\_count[chosen\_subject] = max\_hour - 1

# go to next hour

start\_hour += 1

next\_hour += 1

time += 1

fill\_out\_subjects\_list()

fill\_in\_timetable()

print(f'Subject per slot: {subject\_per\_slot}')

timetable\_path = pathlib.Path.cwd() / 'timetable.csv'

# Now, let's write process to save timetable into a csv file

with open(timetable\_path, 'w') as timetable\_file:

timetable\_writing = csv.writer(timetable\_file)

# Write headers into csv file

csv\_headers = ['Hours']

csv\_headers.extend(school\_days)

timetable\_writing.writerow(csv\_headers)

# Write content into csv file

for time\_slot, concerned\_subjects in subject\_per\_slot.items():

time\_line = [time\_slot]

concerned\_subjects\_list = []

if concerned\_subjects == ['Break time']:

for x in range(0, len(school\_days)):

concerned\_subjects\_list.append('Break time')

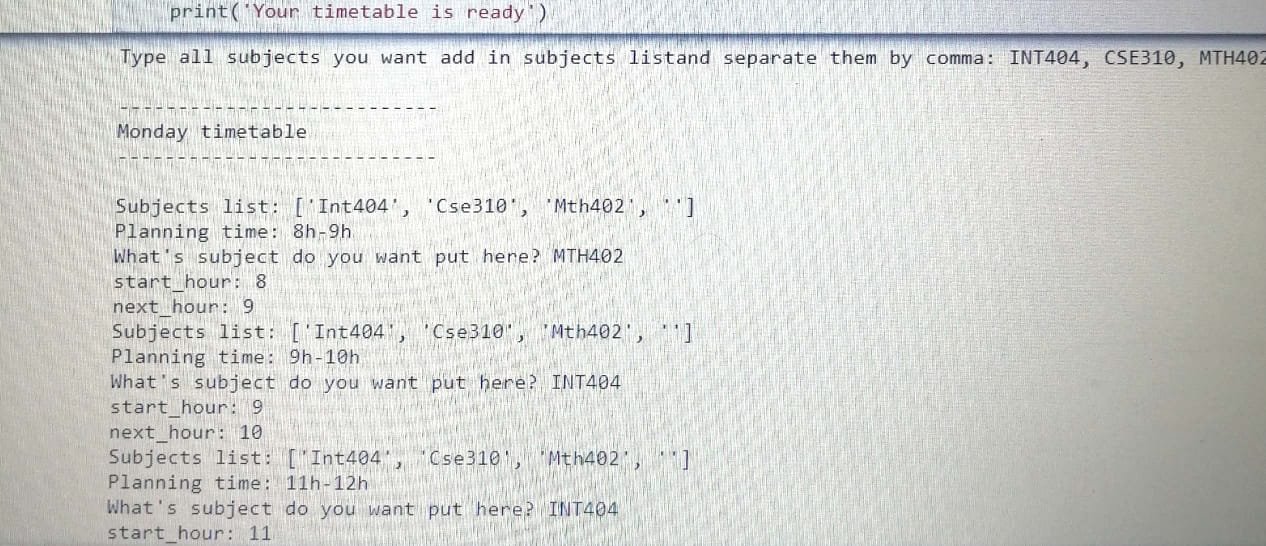
else:

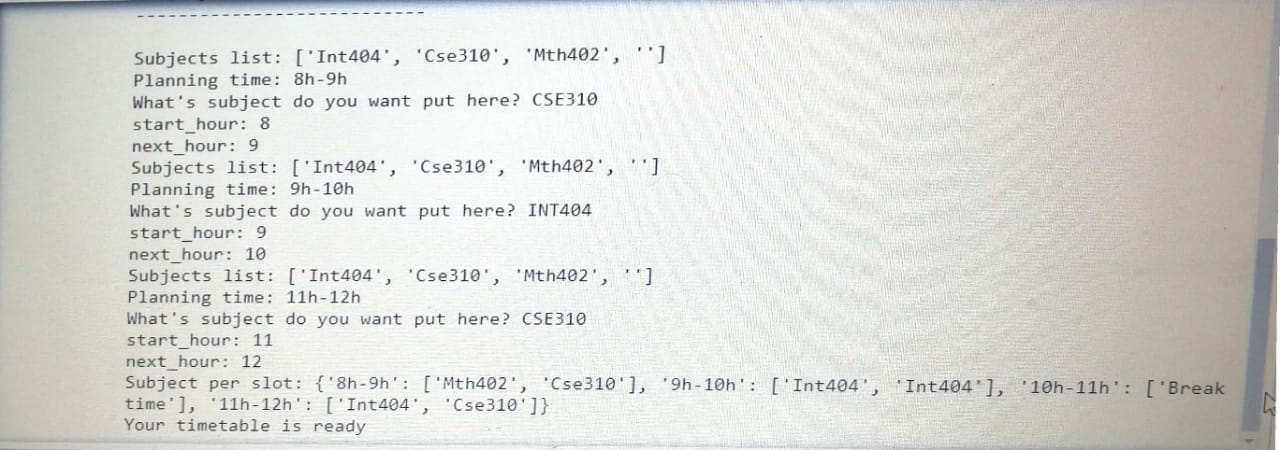
concerned\_subjects\_list = concerned\_subjects

final\_line = time\_line + concerned\_subjects\_list

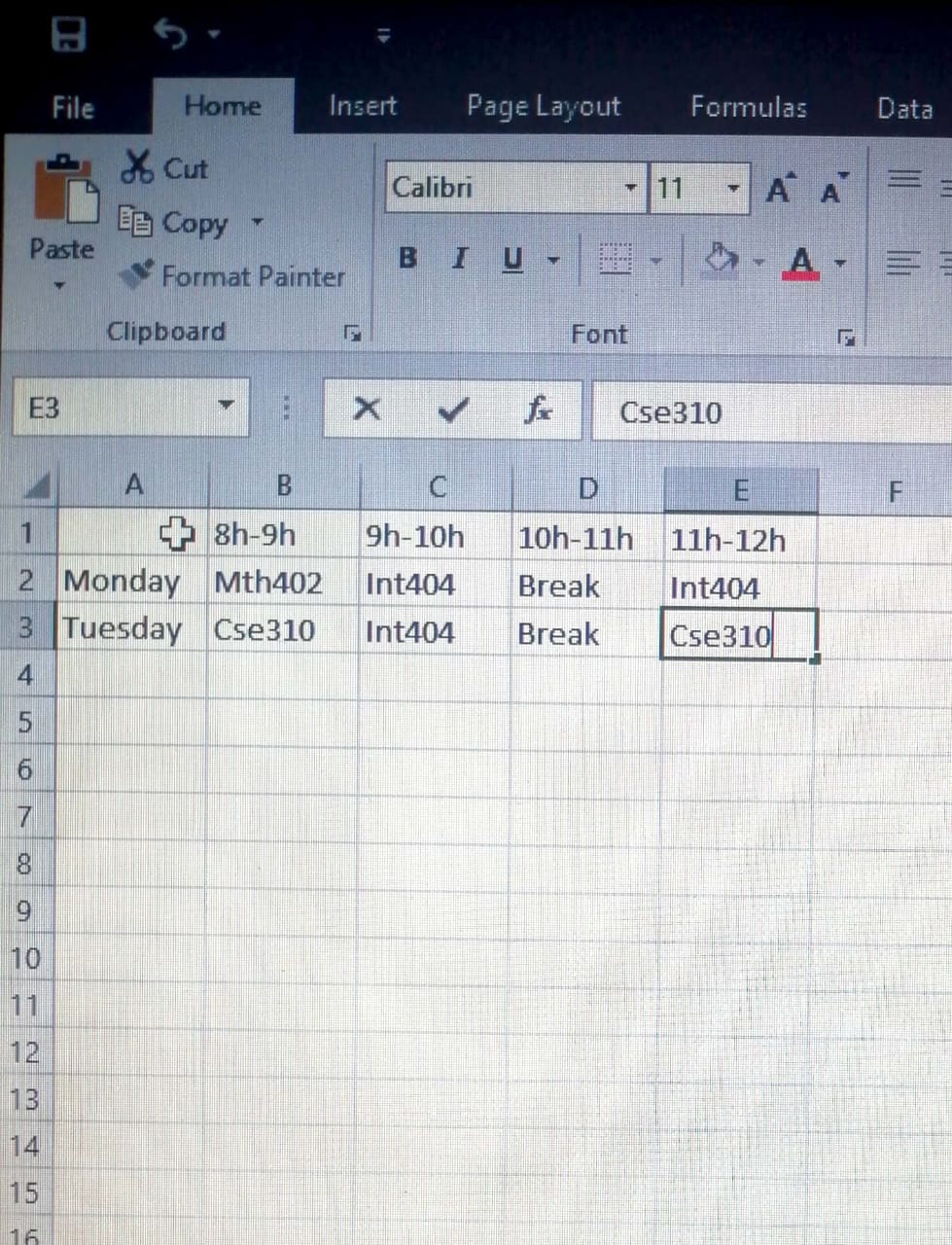
timetable\_writing.writerow(final\_line)

print('Your timetable is ready')





The picture which was shown above is the output in which after compiling it was going to ask the user type the name of the subjects and duration of each other after providing the details it will show the time-table.



And here is the csv file in which the data will be saved and shown in an effective way.

1. **Future Scope:**

As every project had done will be atleast have some effect on future. With help of this project one can able to prepare and manage their own time-table. As preparing the time-table manually is not an easy task when there is strength in thousands but, with the help of these application it is possible and reduce strain and minimize human work.

1. **GitHub Link:**

* <https://github.com/MaddalaNarmada/AI_project1>