

## **Explanation of files in repository:**

student\_activities.csv:

This is the CSV file that we have created that follows the criteria for our program. It contains five columns; Event Name , Weekday, Time,Repeating Weekly?, and Event Type. This CSV file ensures that we have all the information required to run the program, and output the right format.

Event name (str): a string of the event

Weekday (str): string of the day in the week

time(int) in the following format: 10:00-10:50

Repeating weekly?(boolean): yes or no

schedule\_planner.py:

This is the main program that contains our project. All the required information to make our project run it's in here. We have different classes and functions that account for the information provided in the CSV file, creating a solution to one of the issues that many people, specifically college students struggle with; time management. This program effectively outputs an organized and concise schedule that it's easy to understand.

schedule\_output.txt:

This is the text file that will contain the complete schedule! This will also appear in the terminal so the user can read the results from either location. The formatting is shown below in the instructions for reading the output.

## **Instructions on how to run program:**

In order to run our program successfully, we had to account for using an external CSV file that contained all the necessary information to run the program. We have created our own CSV file (will be provided in our repository). This file contains the initial parameters of the program such as event name, weekday, time, occurrence, event time.

To be able to run the program successfully, the following prompt should be written into the command line argument:

```
python3 schedule_planner.py student_activities.csv --output schedule_output.txt
```

### **Instructions on how to read program output:**

This prompt should return a weekly schedule that has information about the person's schedule from Monday to Friday. It provides the user with the event name that is scheduled for that day, the starting and ending time, as well as a section for the free time slots.

It should provide the user with a response in the following format:

Tuesday:

Scheduled events:

Rugby Practice (16:00 - 18:00)

Free time:

00:00 - 16:00

18:00 - 24:00

This is helpful, as the user can visualize what the schedule looks like for the day, and for the rest of the week.

Additionally, in the last line of the prompt provided in the command line argument, we have a text file. This text file is the output with the information provided such as the schedule with its respective information. This text file was created when we ran this prompt through the command line argument.

**Attribution:**

Method/Function	Primary Author	Techniques Demonstrated
show_schedule(self)	Nick	With Statements
show_schedule(self)	Nick	F-Strings Containing Expressions
event_time(self,time_str)	Catalina	Conditional expressions
main()	Catalina	The ArgumentParser Class from argparse
event_time(self)	Catalina	Sequence unpacking
read_input_file	Marlon	comprehensions or generator expressions
free_time	marlon	Key function (lambda
parse_events(self)	marlon	Set operations(union)