AI PROJECT-01

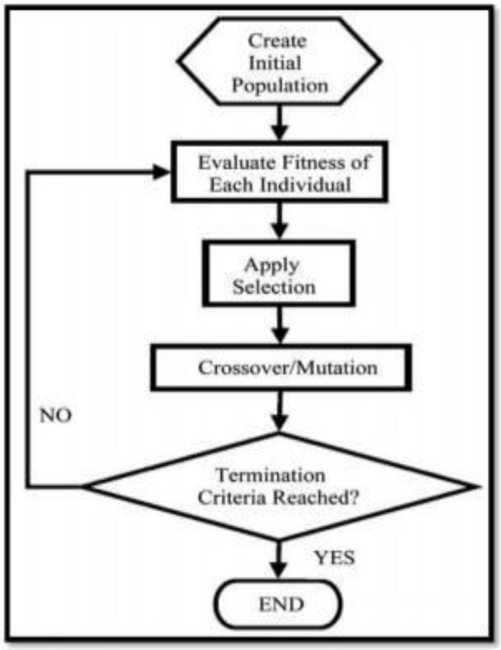
Group Members

Saifullah Dar	i180599
Omar Hayat	i180511

Section: E

Algorithm Implemented: Genetic Algorithm

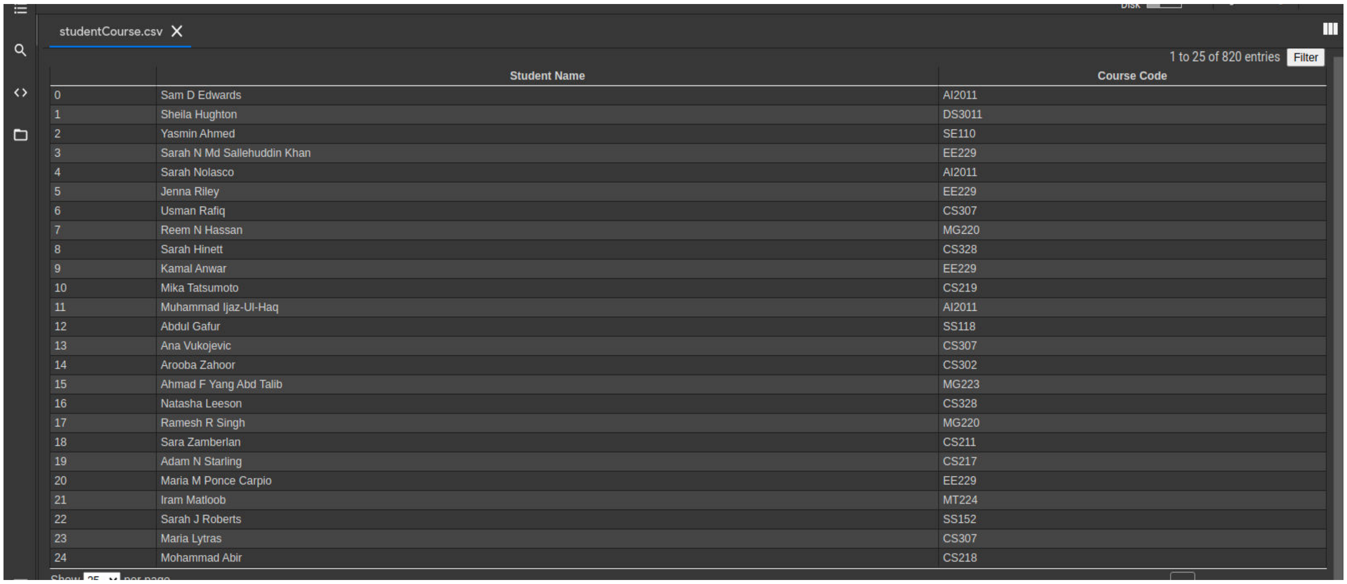
Flowchart:


Create Initial Population
Evaluate Fitness of Each Individual
Apply Selection
Crossover/Mutation
NO
Termination Criteria Reached?
YES
END

Dataset:

1. studentCourse.csv
2. teachers.csv

studentCourse.csv :



1 to 25 of 820 entries

	Student Name	Course Code
0	Sam D Edwards	AI2011
1	Sheila Hughton	DS3011
2	Yasmin Ahmed	SE110
3	Sarah N Md Sallehuddin Khan	EE229
4	Sarah Nolasco	AI2011
5	Jenna Riley	EE229
6	Usman Rafiq	CS307
7	Reem N Hassan	MG220
8	Sarah Hinett	CS328
9	Kamal Anwar	EE229
10	Mika Tatsumoto	CS219
11	Muhammad Ijaz-Ul-Haq	AI2011
12	Abdul Gafur	SS118
13	Ana Vukojevic	CS307
14	Arooba Zahoor	CS302
15	Ahmad F Yang Abd Talib	MG223
16	Natasha Leeson	CS328
17	Ramesh R Singh	MG220
18	Sara Zamberlan	CS211
19	Adam N Starling	CS217
20	Maria M Ponce Carpio	EE229
21	Iram Matloob	MT224
22	Sarah J Roberts	SS152
23	Maria Lytras	CS307
24	Mohammad Abir	CS218

teachers.csv is a csv file containing teacher names in each line.

Representation:

Each exam is represented by a class named as Exam_Slot which is used as gene

Chromosome class comprises of list of exam_slots named as timetable.

Generation:

We generated the total population of 300 chromosomes initially.

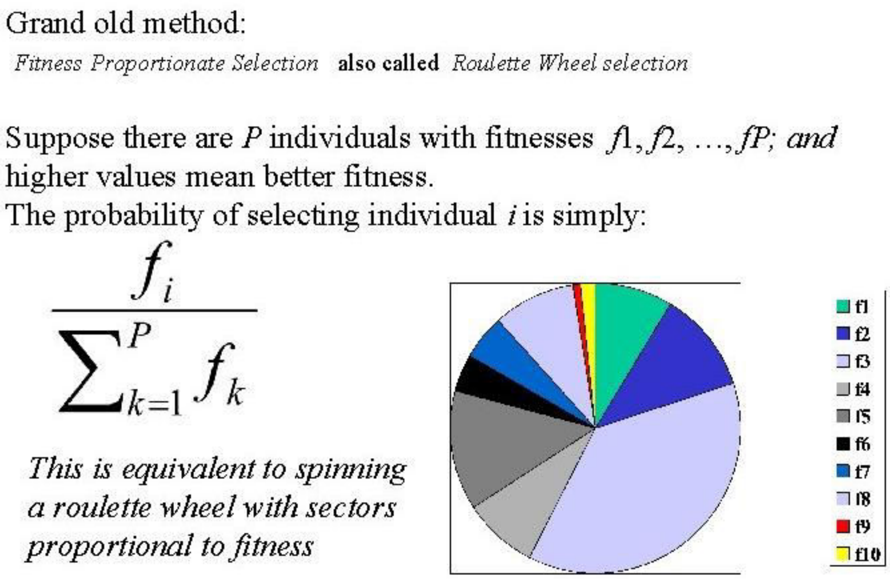
Fitness Function:

In fitness, we analyzed the hard constraints and soft constraints of chromosome and we penalized according to the level of clash. For example, if two exams which have student in common are scheduled at one time, we penalized the chromosome by 5 points.

Selection:

For selection we implemented two methods tournament selection and roulette wheel, but we prefered roulette wheel selection because it was requirement.

Roulette Wheel Selection:



Grand old method:

Fitness Proportionate Selection **also called** *Roulette Wheel selection*

Suppose there are *P* individuals with fitnesses *f*1, *f*2, …, *fP; and* higher values mean better fitness.

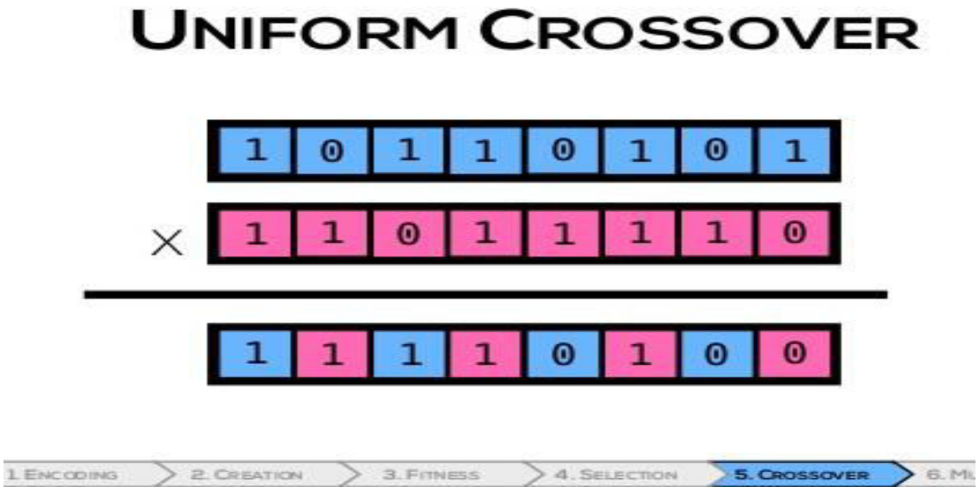
The probability of selecting individual *i* is simply:

$$\frac{f_i}{\sum_{k=1}^{P} f_k}$$

This is equivalent to spinning a roulette wheel with sectors proportional to fitness

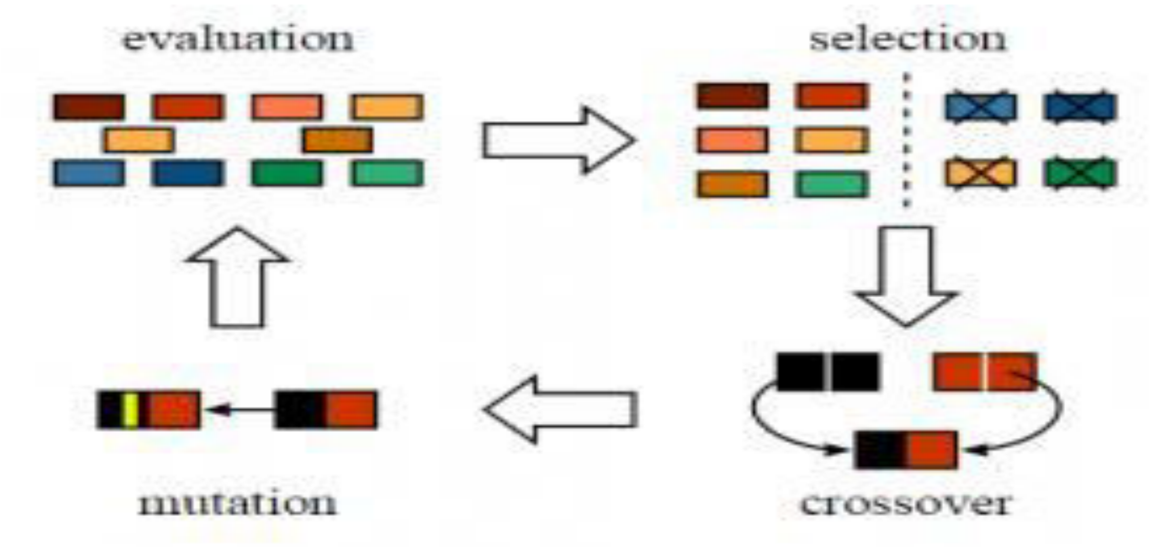
Crossover:

In crossover, we used uniform crossover to overlap the offspring with time,invigilator and rooms of its parent.

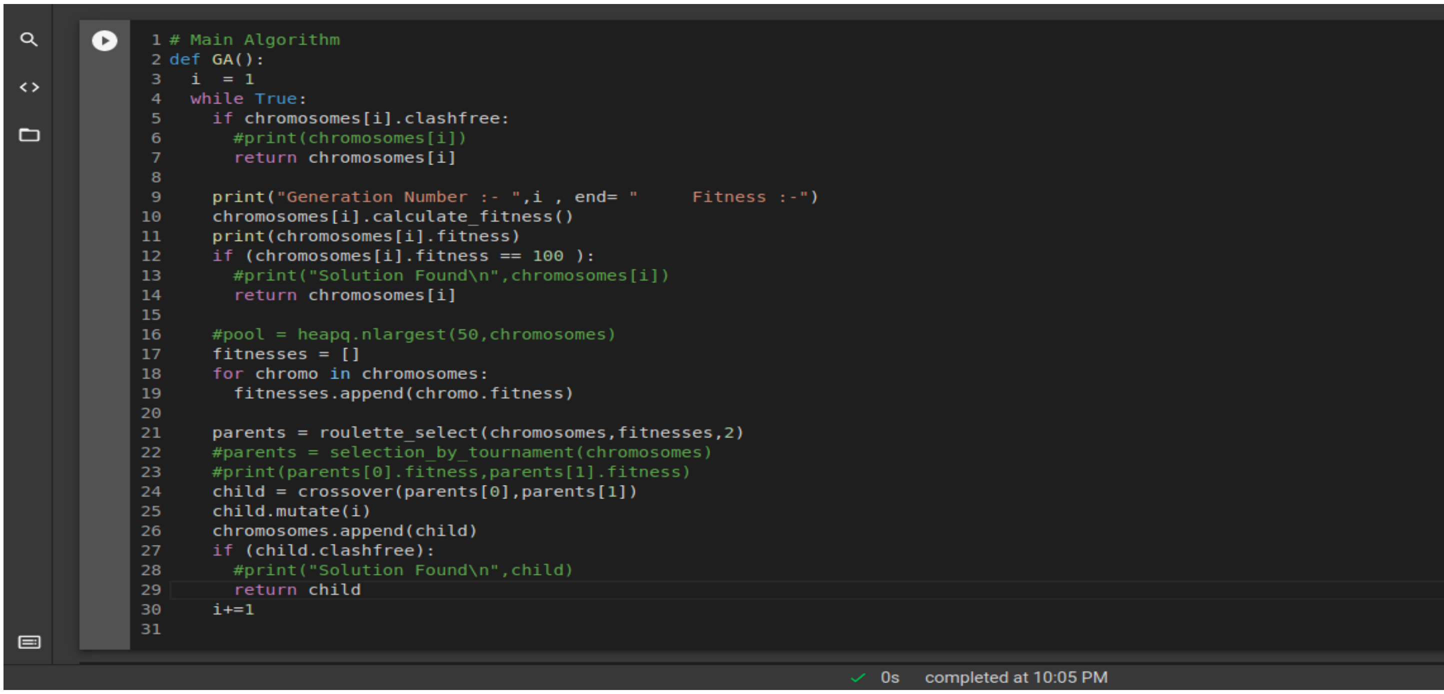


Mutation:

In mutation we fixed the rate of mutation to 0.01 and then mutated the timeslots, teachers and rooms of chromosome keeping in view that where these parameters will enhance the result.

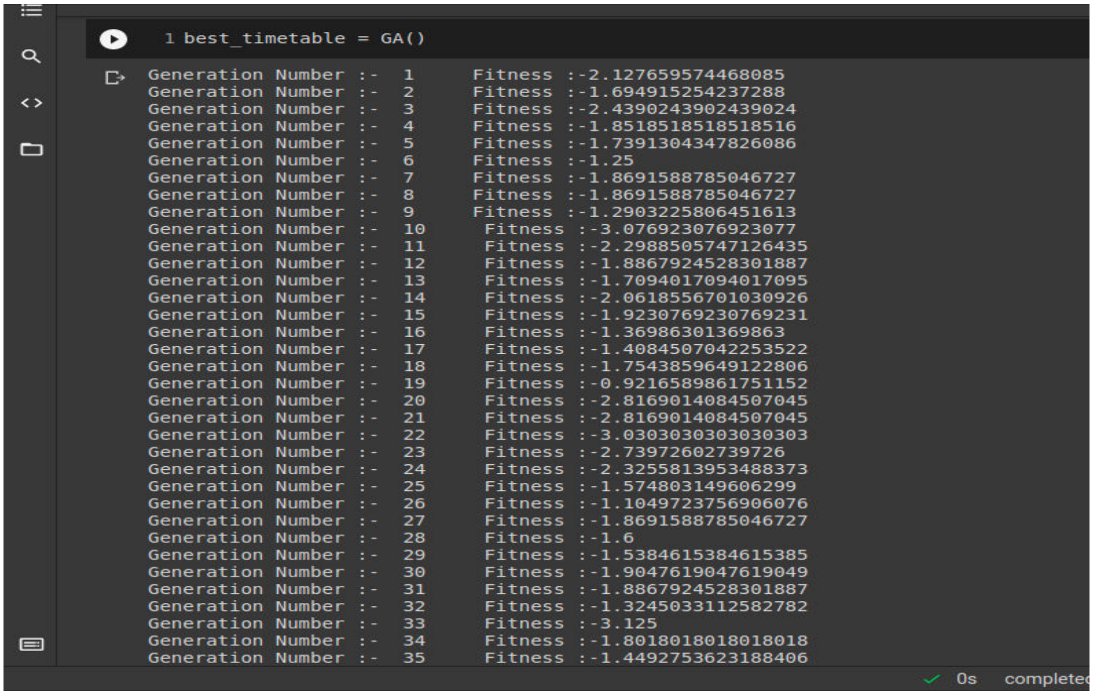


Code of our Main Algorithm:



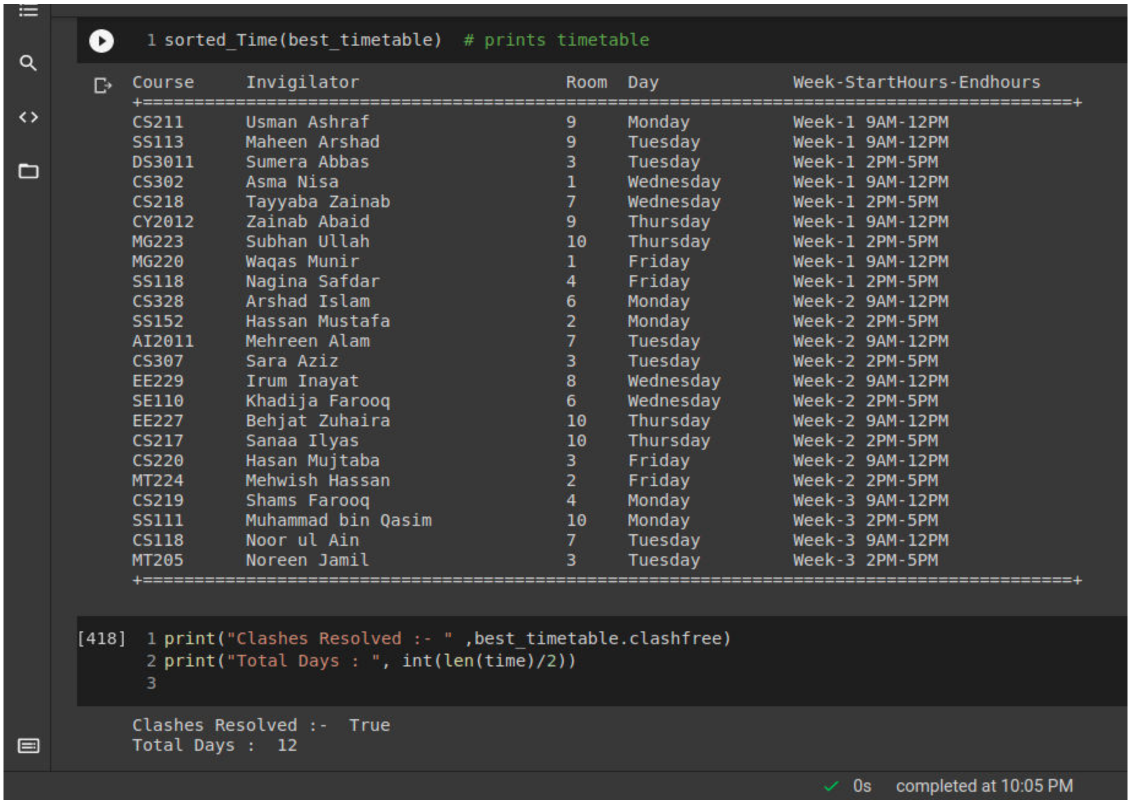
```
# Main Algorithm
def GA():
  i  = 1
  while True:
    if chromosomes[i].clashfree:
      #print(chromosomes[i])
      return chromosomes[i]

    print("Generation Number :- ",i , end= "      Fitness :-")
    chromosomes[i].calculate_fitness()
    print(chromosomes[i].fitness)
    if (chromosomes[i].fitness == 100 ):
      #print("Solution Found\n",chromosomes[i])
      return chromosomes[i]

    #pool = heapq.nlargest(50,chromosomes)
    fitnesses = []
    for chromo in chromosomes:
      fitnesses.append(chromo.fitness)

    parents = roulette_select(chromosomes,fitnesses,2)
    #parents = selection_by_tournament(chromosomes)
    #print(parents[0].fitness,parents[1].fitness)
    child = crossover(parents[0],parents[1])
    child.mutate(i)
    chromosomes.append(child)
    if (child.clashfree):
      #print("Solution Found\n",child)
      return child
    i+=1

```

Output:



```
best_timetable = GA()
```

```
Generation Number :-  1     Fitness :-2.127659574468085
Generation Number :-  2     Fitness :-1.694915254237288
Generation Number :-  3     Fitness :-2.4390243902439024
Generation Number :-  4     Fitness :-1.8518518518518516
Generation Number :-  5     Fitness :-1.7391304347826086
Generation Number :-  6     Fitness :-1.25
Generation Number :-  7     Fitness :-1.8691588785046727
Generation Number :-  8     Fitness :-1.8691588785046727
Generation Number :-  9     Fitness :-1.2903225806451613
Generation Number :-  10     Fitness :-3.076923076923077
Generation Number :-  11     Fitness :-2.2988505747126435
Generation Number :-  12     Fitness :-1.8867924528301887
Generation Number :-  13     Fitness :-1.7094017094017095
Generation Number :-  14     Fitness :-2.0618556701030926
Generation Number :-  15     Fitness :-1.9230769230769231
Generation Number :-  16     Fitness :-1.36986301369863
Generation Number :-  17     Fitness :-1.4084507042253522
Generation Number :-  18     Fitness :-1.7543859649122806
Generation Number :-  19     Fitness :-0.9216589861751152
Generation Number :-  20     Fitness :-2.8169014084507045
Generation Number :-  21     Fitness :-2.8169014084507045
Generation Number :-  22     Fitness :-3.0303030303030303
Generation Number :-  23     Fitness :-2.73972602739726
Generation Number :-  24     Fitness :-2.3255813953488373
Generation Number :-  25     Fitness :-1.574803149606299
Generation Number :-  26     Fitness :-1.1049723756906076
Generation Number :-  27     Fitness :-1.8691588785046727
Generation Number :-  28     Fitness :-1.6
Generation Number :-  29     Fitness :-1.5384615384615385
Generation Number :-  30     Fitness :-1.9047619047619049
Generation Number :-  31     Fitness :-1.8867924528301887
Generation Number :-  32     Fitness :-1.3245033112582782
Generation Number :-  33     Fitness :-3.125
Generation Number :-  34     Fitness :-1.8018018018018018
Generation Number :-  35     Fitness :-1.4492753623188406
```



```
sorted_Time(best_timetable)  # prints timetable
```

```
Course     Invigilator                    Room  Day             Week-StartHours-Endhours
+==========================================================================================+
CS211      Usman Ashraf                   9     Monday          Week-1 9AM-12PM
SS113      Maheen Arshad                  9     Tuesday         Week-1 9AM-12PM
DS3011     Sumera Abbas                   3     Tuesday         Week-1 2PM-5PM
CS302      Asma Nisa                      1     Wednesday       Week-1 9AM-12PM
CS218      Tayyaba Zainab                 7     Wednesday       Week-1 2PM-5PM
CY2012     Zainab Abaid                   9     Thursday        Week-1 9AM-12PM
MG223      Subhan Ullah                   10    Thursday        Week-1 2PM-5PM
MG220      Waqas Munir                    1     Friday          Week-1 9AM-12PM
SS118      Nagina Safdar                  4     Friday          Week-1 2PM-5PM
CS328      Arshad Islam                   6     Monday          Week-2 9AM-12PM
SS152      Hassan Mustafa                 2     Monday          Week-2 2PM-5PM
AI2011     Mehreen Alam                   7     Tuesday         Week-2 9AM-12PM
CS307      Sara Aziz                      3     Tuesday         Week-2 2PM-5PM
EE229      Irum Inayat                    8     Wednesday       Week-2 9AM-12PM
SE110      Khadija Farooq                 6     Wednesday       Week-2 2PM-5PM
EE227      Behjat Zuhaira                 10    Thursday        Week-2 9AM-12PM
CS217      Sanaa Ilyas                    10    Thursday        Week-2 2PM-5PM
CS220      Hasan Mujtaba                  3     Friday          Week-2 9AM-12PM
MT224      Mehwish Hassan                 2     Friday          Week-2 2PM-5PM
CS219      Shams Farooq                   4     Monday          Week-3 9AM-12PM
SS111      Muhammad bin Qasim             10    Monday          Week-3 2PM-5PM
CS118      Noor ul Ain                    7     Tuesday         Week-3 9AM-12PM
MT205      Noreen Jamil                   3     Tuesday         Week-3 2PM-5PM
+==========================================================================================+
```

```
[418] print("Clashes Resolved :- " ,best_timetable.clashfree)
      print("Total Days : ", int(len(time)/2))

```

```
Clashes Resolved :-  True
Total Days :  12
```