Weekly Python Assessment

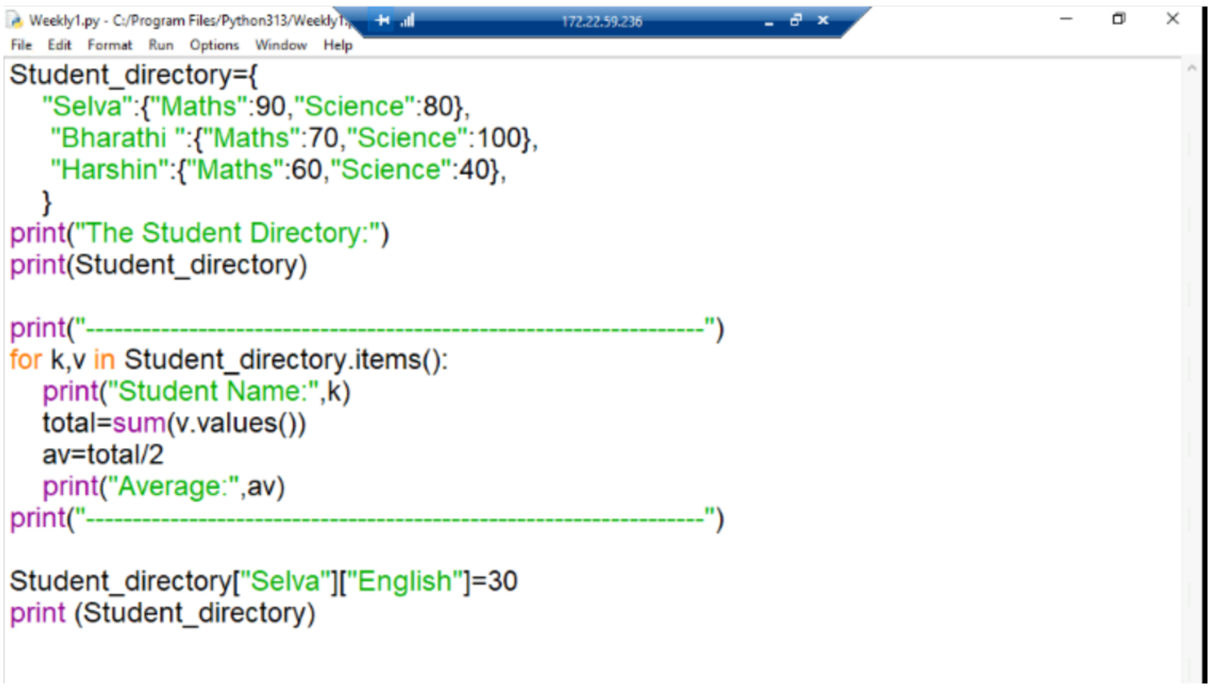
Question1:

Student Grades and Courses Your school's administration wants to keep track of student grades across different courses.

Tasks:

1. Create a dictionary called students where keys are student names (e.g., "Alice", "Bob") and values are dictionaries. Each inner dictionary should contain keys representing courses (e.g., "Math", "Science") and values representing their grades in those courses. Include at least three students and two courses per student.

As Mentioned I created the student dictionary for 3 students and with 2 Subjects as inner directory.


Weekly1.py - C:/Program Files/Python313/Weekly1
File Edit Format Run Options Window Help
172.22.59.236

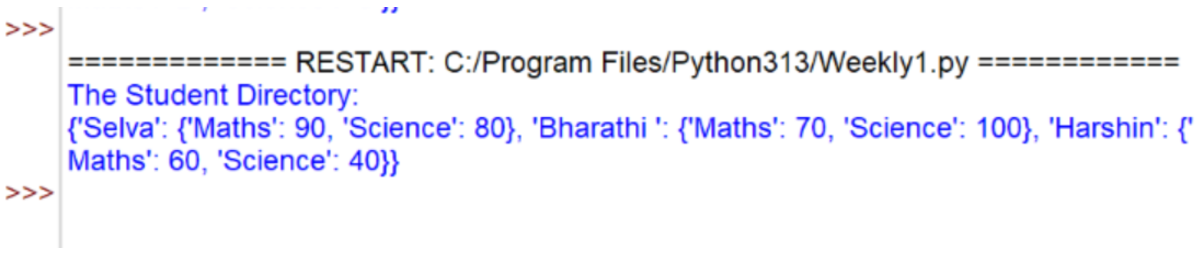
```
Student_directory={
    "Selva":{"Maths":90,"Science":80},
     "Bharathi ":{"Maths":70,"Science":100},
     "Harshin":{"Maths":60,"Science":40},
    }
print("The Student Directory:")
print(Student_directory)

print("--------------------------------------------------------------------")
for k,v in Student_directory.items():
    print("Student Name:",k)
    total=sum(v.values())
    av=total/2
    print("Average:",av)
print("--------------------------------------------------------------------")

Student_directory["Selva"]["English"]=30
print (Student_directory)
```

2. Print the students dictionary.

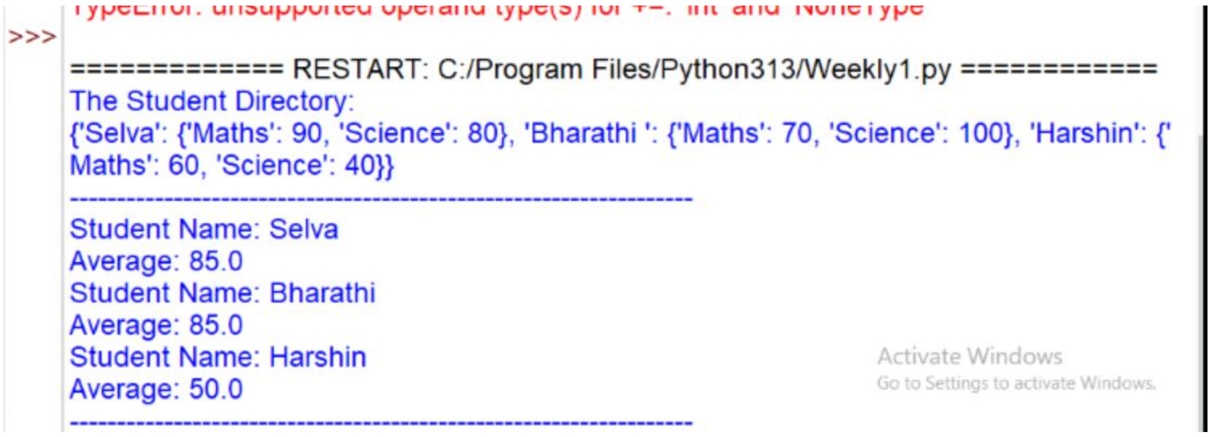
Printing the students dictionary



```
>>>
    ============= RESTART: C:/Program Files/Python313/Weekly1.py ============
    The Student Directory:
    {'Selva': {'Maths': 90, 'Science': 80}, 'Bharathi ': {'Maths': 70, 'Science': 100}, 'Harshin': {'
    Maths': 60, 'Science': 40}}
>>>
```

3. Iterate through the students dictionary and print each student's name and their average grade across all courses.

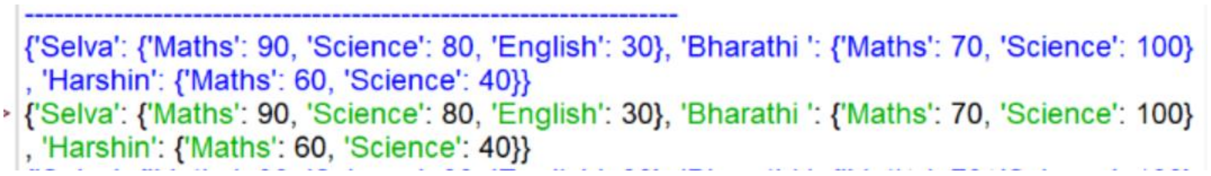
Iterating through the students dictionary



```
TypeError: unsupported operand type(s) for +=: 'int' and 'NoneType'
>>>
============= RESTART: C:/Program Files/Python313/Weekly1.py ============
The Student Directory:
{'Selva': {'Maths': 90, 'Science': 80}, 'Bharathi ': {'Maths': 70, 'Science': 100}, 'Harshin': {'
Maths': 60, 'Science': 40}}
------------------------------------------------------------------
Student Name: Selva
Average: 85.0
Student Name: Bharathi
Average: 85.0
Student Name: Harshin
Average: 50.0
------------------------------------------------------------------
```

4. Add a new course and grade for one of the existing students.

Adding one subject "English " to the student "Selva"



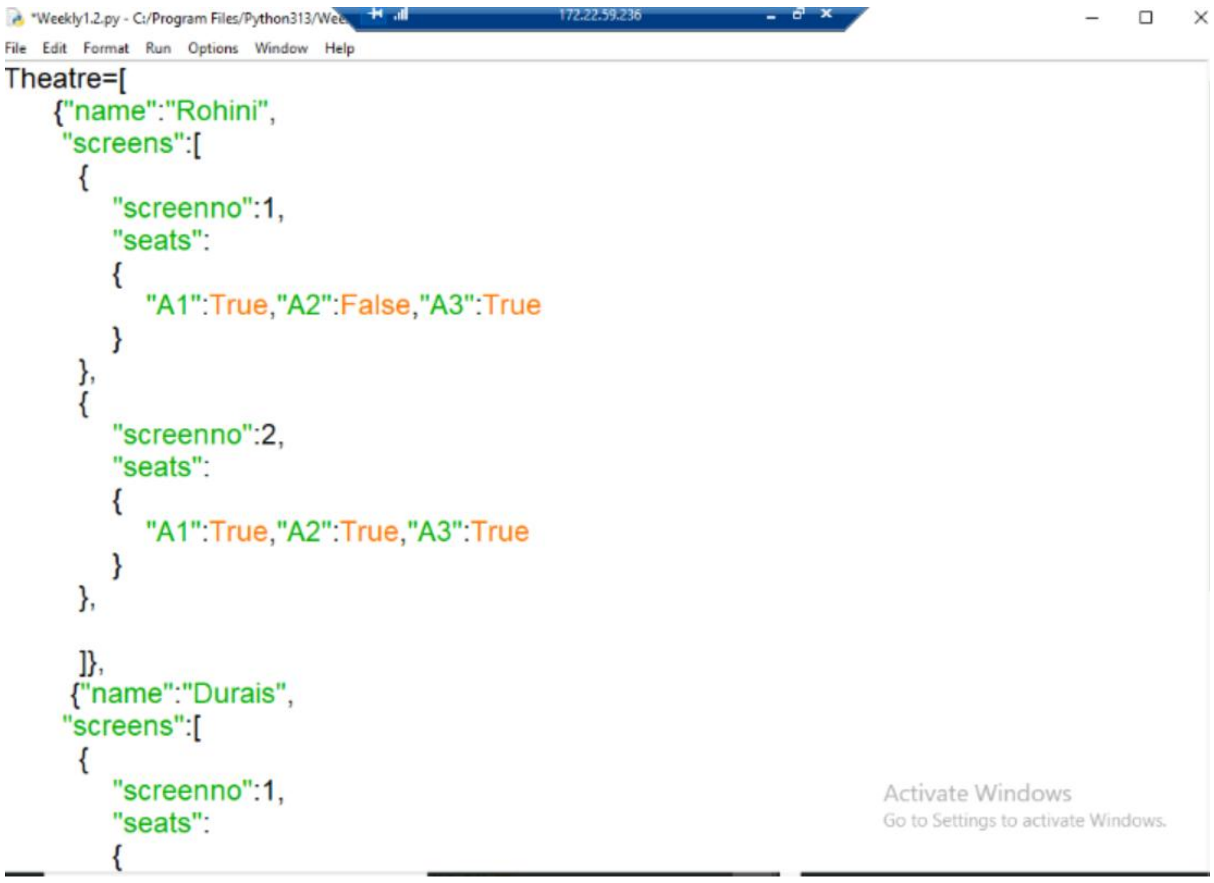
```
------------------------------------------------------------------
{'Selva': {'Maths': 90, 'Science': 80, 'English': 30}, 'Bharathi ': {'Maths': 70, 'Science': 100}
, 'Harshin': {'Maths': 60, 'Science': 40}}
{'Selva': {'Maths': 90, 'Science': 80, 'English': 30}, 'Bharathi ': {'Maths': 70, 'Science': 100}
, 'Harshin': {'Maths': 60, 'Science': 40}}
```

Question2: Movie Ticket Booking

1. **Data Setup o Create a list called theaters. Each element is a dictionary with keys:**
   - **"name" (theater name)**
   - **"screens" (a nested list of screen dictionaries)**

   o Each screen dictionary has:
   - **"screen_number" (int)**
   - **"seats" (dict mapping seat IDs like "A1" to True/False for booked status).**

   The data has been created as per requested



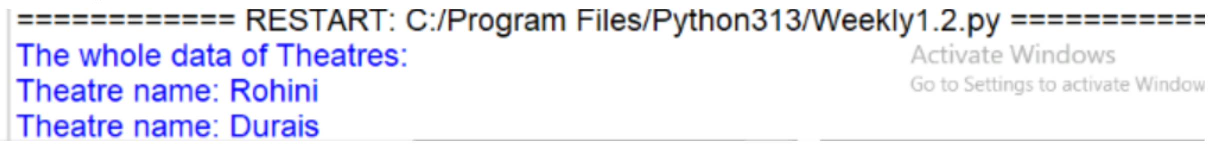
*Weekly1.2.py - C:/Program Files/Python313/We
172.22.59.236
File Edit Format Run Options Window Help

```
Theatre=[
    {"name":"Rohini",
     "screens":[
      {
         "screenno":1,
         "seats":
         {
            "A1":True,"A2":False,"A3":True
         }
      },
      {
         "screenno":2,
         "seats":
         {
            "A1":True,"A2":True,"A3":True
         }
      },

      ]},
     {"name":"Durais",
     "screens":[
      {
         "screenno":1,
         "seats":
         {
```

2. **Print Theaters o Print the full theaters list.**

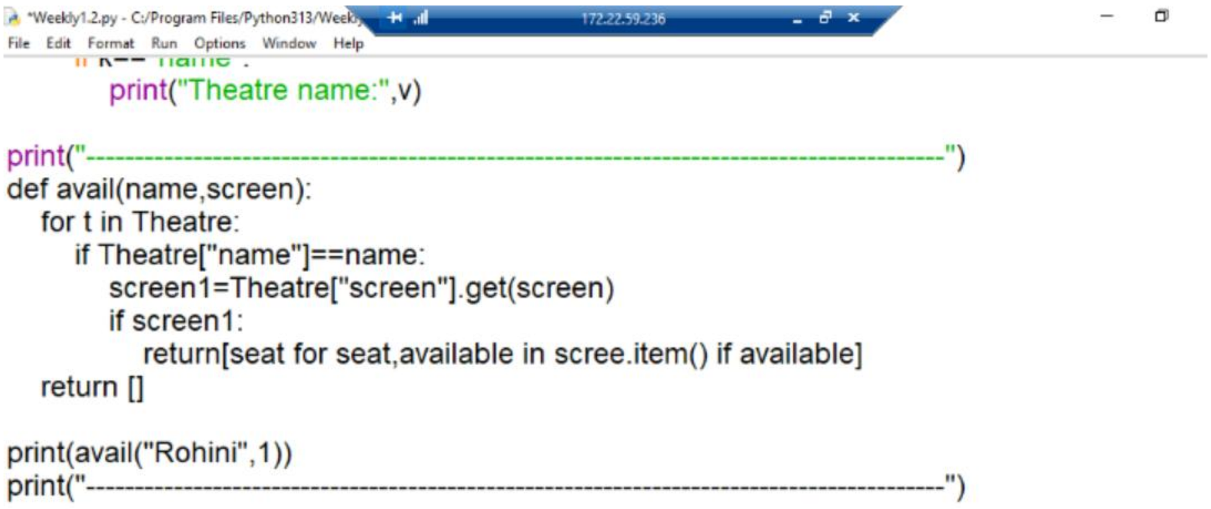
   Printing the data with loops



```
============ RESTART: C:/Program Files/Python313/Weekly1.2.py ==========
The whole data of Theatres:
Theatre name: Rohini
Theatre name: Durais
```

3. **Show Availability o For a given theater and screen, list all unbooked seats.**

   The function to print the list of available seats



*Weekly1.2.py - C:/Program Files/Python313/Weekly
172.22.59.236
File Edit Format Run Options Window Help

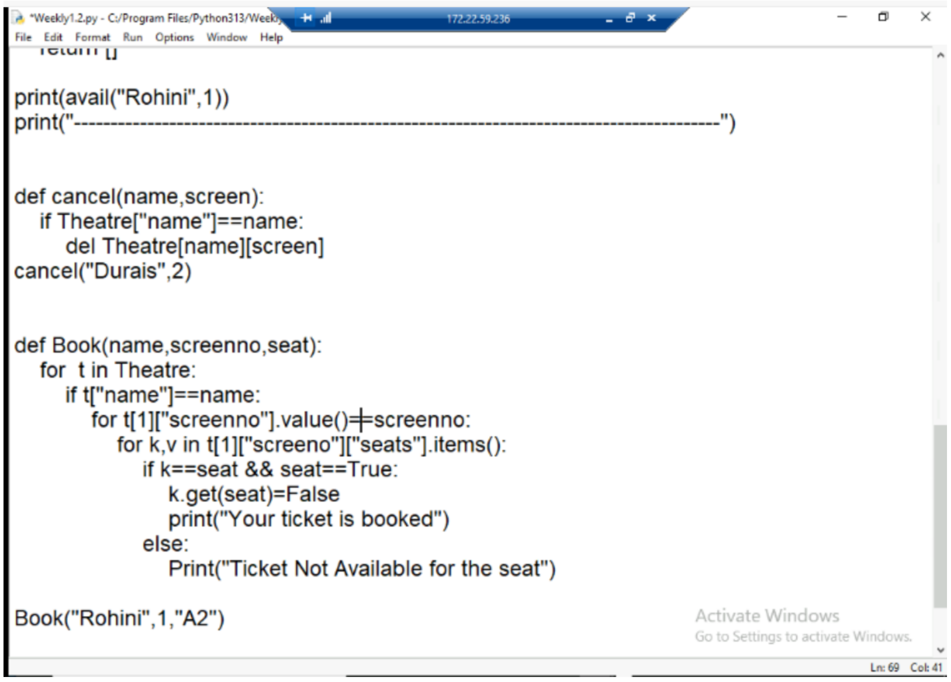
```
        print("Theatre name:",v)

print("-------------------------------------------------------------------------------------")
def avail(name,screen):
    for t in Theatre:
        if Theatre["name"]==name:
            screen1=Theatre["screen"].get(screen)
            if screen1:
                return[seat for seat,available in scree.item() if available]
    return []

print(avail("Rohini",1))
print("-------------------------------------------------------------------------------------")
```

4. **Book a Seat o Write a function that takes theater name, screen number, and seat ID and marks it booked.**

   Program to book a seat



*Weekly1.2.py - C:/Program Files/Python313/Weekly
172.22.59.236
File Edit Format Run Options Window Help
Activate Windows
Go to Settings to activate Windows.
Ln: 69 Col: 41

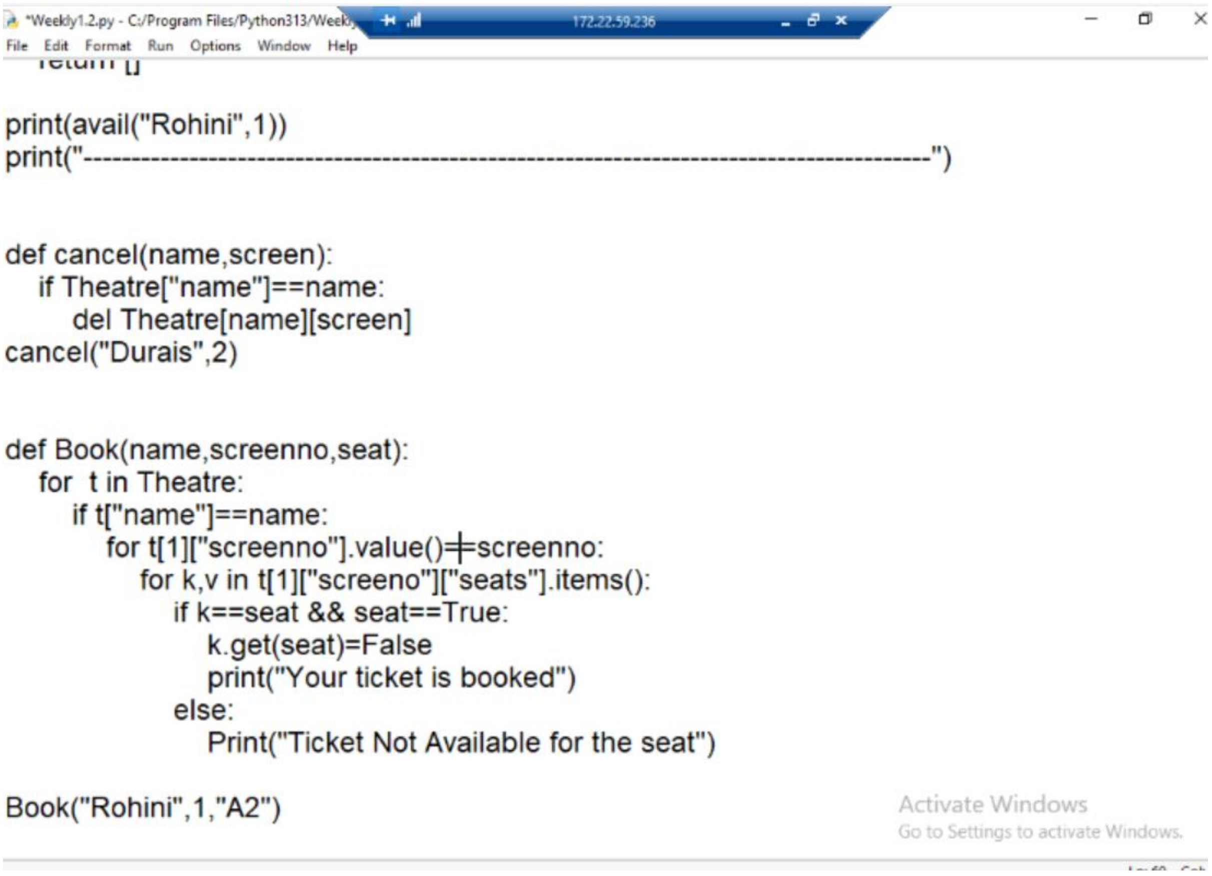
```
print(avail("Rohini",1))
print("-------------------------------------------------------------------------------------")


def cancel(name,screen):
    if Theatre["name"]==name:
        del Theatre[name][screen]
cancel("Durais",2)


def Book(name,screenno,seat):
    for  t in Theatre:
        if t["name"]==name:
            for t[1]["screenno"].value()==screenno:
                for k,v in t[1]["screeno"]["seats"].items():
                    if k==seat && seat==True:
                        k.get(seat)=False
                        print("Your ticket is booked")
                    else:
                        Print("Ticket Not Available for the seat")

Book("Rohini",1,"A2")
```

5. **Cancel a Screen o Remove one screen dictionary from a selected theater.**
   Function to delete a screen



*Weekly1.2.py - C:/Program Files/Python313/Weekly

File Edit Format Run Options Window Help

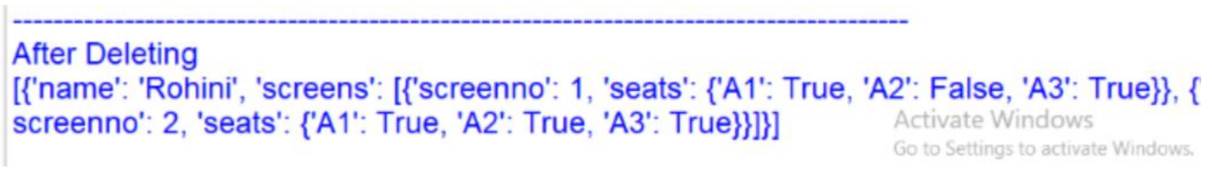
```
print(avail("Rohini",1))
print("-------------------------------------------------------------------------------------------")


def cancel(name,screen):
   if Theatre["name"]==name:
      del Theatre[name][screen]
cancel("Durais",2)


def Book(name,screenno,seat):
   for  t in Theatre:
      if t["name"]==name:
         for t[1]["screenno"].value()==screenno:
            for k,v in t[1]["screeno"]["seats"].items():
               if k==seat && seat==True:
                  k.get(seat)=False
                  print("Your ticket is booked")
               else:
                  Print("Ticket Not Available for the seat")

Book("Rohini",1,"A2")
```

Activate Windows
Go to Settings to activate Windows.



```
-------------------------------------------------------------------------------------------
After Deleting
[{'name': 'Rohini', 'screens': [{'screenno': 1, 'seats': {'A1': True, 'A2': False, 'A3': True}}, {
screenno': 2, 'seats': {'A1': True, 'A2': True, 'A3': True}}]}]
```

Activate Windows
Go to Settings to activate Windows.