
System Requirements Specification Index

For

Python Basics and NumPy, Pandas
Use case No 8
1.0

IIHT Pvt. Ltd.

fullstack@iiht.com

Usecase No 1 Use Case: Student Attendance Management System (student.py)

Dataset

students = {

```
101: {'name': 'Alice Johnson', 'class': '10th Grade'},
102: {'name': 'Bob Smith', 'class': '10th Grade'},
103: {'name': 'Charlie Brown', 'class': '10th Grade'},
104: {'name': 'David Lee', 'class': '10th Grade'},
105: {'name': 'Eve Miller', 'class': '10th Grade'}
```

}

Attendance Records (student_id: list of (date, status) tuples):

attendance = {

```
101: [('2025-02-25', 'Absent'), ('2025-02-26', 'Absent'), ('2025-02-27', 'Absent')],
102: [('2025-02-25', 'Absent'), ('2025-02-26', 'Present'), ('2025-02-27', 'Present')],
103: [('2025-02-25', 'Present'), ('2025-02-26', 'Absent'), ('2025-02-27', 'Present')],
104: [('2025-02-25', 'Present'), ('2025-02-26', 'Present'), ('2025-02-27', 'Present')],
105: [('2025-02-25', 'Absent'), ('2025-02-26', 'Present'), ('2025-02-27', 'Absent')]
```

}

1 Write a Python function to count the number of unique attendance days.

Define: count_number_of_days_using_dates()

The function should:

- Identify the unique dates in the attendance dataset.
- Print the total number of unique dates.

2 Write a Python function to find the student with the most absent days.

Define: find_most_absent_student()

The function should:

- Count the number of absences for each student.
- Identify and print the student with the highest number of absent days.

Use Case No 2: Flight Management System (FlightReservationSystem.py)

Dataset

flights_data = {

```
"Flight Number": ["AI101", "BA202", "DL303"],
"Airline": ["Air India", "British Airways", "Delta Airlines"],
"Total Seats": [150, 180, 200],
"Booked Seats": [120, 160, 190],
"Ticket Price": [8000, 12000, 10000] # Price per ticket in ₹
```

}

A **static dataset** is defined using a **Python dictionary** named flights_data.

It contains information about 3 flights, including:

"Flight Number"

"Airline"

"Total Seats"

"Booked Seats"

"Ticket Price" (in ₹)

This dictionary is converted into a **Pandas DataFrame (flights_df)** for all subsequent operations.

1. Write a Python function to list all flights with details.

Define: list_all_flights()

The function should:

- Display all flight details in a tabular format using Pandas DataFrame.
- Print the list of available flights.
- Return type **DataFrame**

2. Write a Python function to check available seats for a given flight.

Define: available_seats_for_flight(flight_number)

The function should:

- Accepts a **flight number** (e.g., "AI101").
- Looks up the flight in the DataFrame.
- If the flight exists:
 - Retrieves Total Seats and Booked Seats.
 - Uses **NumPy** to compute available_seats = Total Seats - Booked Seats.
 - Returns the available seat count as an **integer**.
- Return the available seat count or an appropriate message if the flight is not found.
- Test case only check first flight other two flights seats are just to display.

3. Write a Python function to calculate the total revenue generated from all flights.

Define: total_revenue_for_all_flights()

The function should:

- Calculate the revenue for each flight using Booked Seats * Ticket Price.
- Compute the total revenue using Pandas operations to sum.
- Return the total revenue value numeric value.

Main function returns the value of dictionary see the below format provided

Sample format output expected

```
--- All Flights ---
Flight Number   Airline Total Seats Booked Seats Ticket Price
0   <Flight1>   <Airline1>   <Seats1>   <Booked1>   <Price1>
1   <Flight2>   <Airline2>   <Seats2>   <Booked2>   <Price2>
2   <Flight3>   <Airline3>   <Seats3>   <Booked3>   <Price3>

--- Available Seats for Each Flight ---
<Flight1>: <Available1> seats available
<Flight2>: <Available2> seats available
<Flight3>: <Available3> seats available

--- Total Revenue for All Flights ---
Total Revenue: ₹<TotalRevenue>
--- Final Result ---
{
  'flights':
    Flight Number   Airline Total Seats Booked Seats Ticket Price Total Revenue
0   <Flight1>   <Airline1>   <Seats1>   <Booked1>   <Price1>   <Revenue1>
1   <Flight2>   <Airline2>   <Seats2>   <Booked2>   <Price2>   <Revenue2>
2   <Flight3>   <Airline3>   <Seats3>   <Booked3>   <Price3>   <Revenue3>,

  'available_seats': {
    '<Flight1>': <Available1>,
    '<Flight2>': <Available2>,
    '<Flight3>': <Available3>
  },

  'total_revenue': <TotalRevenue>
}
```

Use Case No 3: Movie Ticket Booking System (MovieTicketBookingSystem.py)

Dataset is given from movie.json

```
movies = {
    "Avengers: Endgame": {
        "total_seats": 10,
        "booked_seats": 7,
        "ticket_price": 250 # Price per ticket in ₹
    },
    "Inception": {
        "total_seats": 8,
        "booked_seats": 4,
        "ticket_price": 200
    },
    "The Dark Knight": {
        "total_seats": 12,
        "booked_seats": 11,
        "ticket_price": 300
    }
}
```

Write the python code for the following


- 1 Use the file handling method `load_movies()` :
to read the dataset value from movie.json if the file does not exist return false
 - if the file exist create the function to check the `total_tickets_sold()`
 - Sum up all the booked seats across different movies.
 - Return the total count of sold tickets.
- 2. Write a Python function to find the movie that has generated the highest revenue.
Define: `highest_revenue_movie()`
The function should:
 - Compute the total revenue for each movie using `Booked Seats * Ticket Price`.
 - Identify and return the movie with the highest revenue.

Execution Steps to Follow:

- All actions like build, compile, running application, running test cases will be through Command Terminal.
- To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal
- This editor Auto Saves the code
- If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use **CTRL+Shift+B** -command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
- These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
To setup environment:
- You can run the application without importing any packages
To launch application:
python3 student.py
python3 MovieTicketBookingSystem.py
python3 FlightReservationSystem.py
- To run Test cases:
python3 -m unittest
- Before Final Submission also, you need to use **CTRL+Shift+B** - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.

Screen shot to run the program


```
OK
coder@dighe20250227t070305r1fj5p3:/home/myproject/dighegmailcom_20250227T070305$ python3 <<scriptname>>.py
```



To run the application

```
python3 student.py
python3 MovieTicketBookingSystem.py
python3 FlightReservationSystem.py
```

```
• coder@dighe20250227t070305r1fj5p3:/home/myproject/dighegmailcom_20250227T070305$ python3 -m unittest
TestBoundary = Passed
.TestExceptional = Passed
.TestCalculateTotalDonations = Failed
.TestCalculateTotalStockValue = Failed
.TestCheckFrankWhiteDonated = Failed
```



To run the testcase

```
python3 -m unittest
```

The screenshot displays the 'Python-Level1-Template' assessment page. The browser address bar shows the URL: one.techademy.com/assessment-v1/IHTINTERNAL/test-taker/test/NWZlOTFhMWIhMGIyNS00ZDk1T6ZjYiODU5NDU3MTYyNTIhIDAwMz4ZjZlWUwZj.... The page title is 'Python-Level1-Template'. In the top right corner, it indicates 'Time Remaining: 02:54:04' and a 'Submit Assessment' button. The left sidebar shows a 'Project' dropdown with '1' and a list of projects. The main workspace area is titled '1 Python-Level1-Template' and features a large VS Code logo. Below the logo are two input fields: 'URL' with the value 'https://compiler.techademy.com/vsc/cddqw20250224t0114391...' and 'Password' with the value 'mA4DpEBElf'. A note below these fields states: 'Kindly use this password to login into the IDE. All the questions, files and related documents for this assessment will be available within the IDE.' The bottom status bar includes a 'Speed Test' section showing 'Speed Test: >= 10Mbps' and 'Live: 0.41Mbps', along with a 'techademy' logo, a contact email 'techademy@iit.com', a 'Privacy Policy' link, and a footer note 'All Rights Reserved by Techademy Learning Solutions Private Limited'.

- Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on “Submit Assessment” after you are done with code.