

BASAVARAJESWARI GROUP OF INSTITUTIONS

Ballari Institute of Technology & Management

AUTONOMOUS INSTITUTE UNDER VISVESVARAYA TECHNOLOGICAL UNIVERSITY JNANASANGAMA,

BELAGAVI 590018

INTERNSHIP

Report On

Team Merchandising Revenue System

Submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Engineering

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

R RAGHAVENDRA

3BR22CS132

Internship Carried Out

By

EZ TRAININGS & TECHNOLOGIES PVT.LTD

HYDERABAD

Internal Guide

Mrs. MADHURI A

Assistant Professor ,CSE

Ms. SAMEENA YASMEEN

Supervisor ,CSE

External Guide

Mr. BALAJI SRINIVASAN

Sr. Faculty

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NACC Accredited Institution*

(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)

"Jnana Gangotri" Campus, No. 873/2, Ballari-

Hospet Road, Allipur, Ballari-1

3104 (Karnataka) (India) Ph: 08392-

237100/237190, Fax: 08392-237197

2023-2024

BASAVARAJESWARI GROUP OF INSTITUTIONS
BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

Autonomous institute under VISVESVARAYA TECHNOLOGICAL UNIVERSITY JNANASANGAMA,

BELAGAVI 590018

NACC Accredited Institution*

(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)

"Jnana Gangotri" Campus, No. 873/2, Ballari-Hospet Road, Allipur,
Ballari-583104 (Karnataka) (India)

Ph: 08392-237100/237190, Fax: 08392-237197



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Internship entitled **"Team Merchandising Revenue System"** has been successfully completed by **R RAGHAVENDRA** bearing USN **3BR22CS132** a bonafide student of Ballari Institute of Technology and Management, Ballari. For the partial fulfillment of the requirements for the **Bachelor's Degree in Computer Science and Engineering** of the VISVESVARAYA TECHNOLOGICAL UNIVERSITY, Belagavi during the academic year 2023-2024.

Signature of Internship

Co-ordinators

Mrs. MADHURI A

Assistant Professor ,CSE

&

Ms. SAMEENA YASMEEN

Supervisor ,CSE

Signature of HOD

Dr. R N KULKARNI

Professor & HOD(CSE)

DECLARATION

I, R RAGHAVENDRA, second year student of Computer Science and Engineering, Ballari Institute of Technology, Ballari, declare that Internship entitled **Team Merchandising Revenue System** is a part of Internship Training successfully carried out by **EZ TECHNOLOGIES & TRAININGS PVT.LTD, Hyderabad** at “**BITM, BALLARI**”. This report is submitted in partial fulfillment of the requirements for the award of the degree, Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi.

Date : 04 - 05 - 2024
Place : BALLARI

Signature of the Student

ACKNOWLEDGEMENT

The satisfactions that a company the successful completion of my internship on “**Team Merchandising Revenue System**” would be incomplete without the mention of people who made it possible, whose noble gesture, affection, guidance, encouragement and support crowned my efforts with success. It is my privilege to express my gratitude and respect to all those who inspired me in the completion of my internship.

I am grateful to my respective coordinators “**Mrs.Madhuri A(Asst.prof,CSE) and Ms.Sameena Yasmeen (Supervisor,CSE)**” for their noble gesture ,support co-ordination and valuable suggestions given to me in the completion of Internship.

I also thank **Dr. R N Kulkarni**, HOD , Department of **Computer Science and Engineering** for extending all his valuable support and encouragement.

Table of Contents

Chapter No.	Chapter Name	Page No.
1	Company Profile	1
2	Day to day activity(student diary extract)	
3	Abstract	2
4	Introduction of the project	3
5	Description	4 - 6
6	Flowchart	7
7	Output	8 - 11
8	Conclusion	12
9	References	13

CHAPTER – 1:

COMPANY PROFILE

Company Name : EZ Trainings and Technologies Pvt. Ltd.

Introduction:

EZ Trainings and Technologies Pvt. Ltd. is a dynamic and innovative organization dedicated to providing comprehensive training solutions and expert development services. Established with a vision to bridge the gap between academic learning and industry requirements, we specialize in college trainings for students, focusing on preparing them for successful placements. Additionally, we excel in undertaking development projects, leveraging cutting-edge technologies to bring ideas to life.

Mission:

Our mission is to empower the next generation of professionals by imparting relevant skills and knowledge through specialized training programs. We strive to be a catalyst in the career growth of students and contribute to the technological advancement of businesses through our development projects.

Services:

College Trainings:

- Tailored training programs designed to enhance the employability of students.
- Industry-aligned curriculum covering technical and soft skills.
- Placement assistance and career guidance.

Development Projects:

- End-to-end development services, from ideation to execution.
- Expertise in diverse technologies and frameworks.
- Custom solutions to meet specific business needs.

Locations:

Hyderabad | Delhi NCR At EZ Trainings and Technologies Pvt. Ltd., we believe in transforming potential into excellence

CHAPTER – 3:

ABSTRACT

→ **Team Merchandising Revenue System POC** is a python-based solution designed to automate the merchandise sales of different teams. The system aims to optimize tasks like data entry, sales tracking, and revenue analysis, and it also improve decision-making processes.

It is developed using object-oriented programming (OOP), Inbuilt methods, Functions and CRUD operations.

→ **Merchandise Class:** This class represents individual merchandise items and their attributes such as item ID, name, team ID, revenue, and description. It has methods to initialize and represent merchandise objects.

→ **RevenueReport Class:** This class manages a collection of merchandise items. It has methods to create, read, update, and delete merchandise items. Additionally, it has methods to track sales for a specific team and analyze revenue streams.

→ **Unit Tests:** The unit tests verify the functionality of the RevenueReport class methods. It checks if items can be created, read, updated, and deleted correctly. It also tests the methods to track team merchandise sales and analyze revenue streams.

CHAPTER – 4:

INTRODUCTION OF THE PROJECT

Introduction:

***Project name:* - Team Merchandising Revenue System POC**

In today's market, effective management of merchandise is crucial for businesses to thrive. A merchandise management system enables organizations to efficiently handle inventory, track sales, and analyze revenue streams.

The **Team Merchandising Revenue System** project aims to automate the management of merchandise sales of different teams.

It provides functionalities to input, modify, and delete merchandise data, along with capabilities to analyze sales data of each team and generate comprehensive revenue reports. It allows for the creating, reading, updating, and deleting (CRUD) operations on merchandise items, as well as tracking team merchandise sales and analysing revenue streams.

Thus, the system aims to optimize tasks like data entry, sales tracking, and revenue analysis, and it also improve decision-making processes.

CHAPTER – 5:

MODULE DESCRIPTION

class Merchandise:

```
def __init__(self, item_id, name, team_id, revenue, description):
```

```
    self.item_id = item_id
```

```
    self.name = name
```

```
    self.team_id = team_id
```

```
    self.revenue = revenue
```

```
    self.description = description
```

```
def __str__(self):
```

```
    return f'Item ID: {self.item_id}, Name: {self.name}, Team ID: {self.team_id},  
           Revenue: {self.revenue}, Description: {self.description}'
```

→ In Merchandise class, instance of the Merchandise class contains merchandise item, including its item ID, name, team ID, revenue, and description. when the parameter are passed then they are assigned to the variables.

The f-string formatting syntax is used to embed the values of the object's attributes into the string. The `__str__` method returns a string representation of the merchandise item.

class RevenueReport:

```
def __init__(self):
```

```
    self.merchandise = {}
```

```
def create(self, item_id, name, team_id, revenue, description):
```

```
    if item_id in self.merchandise:
```

```
        raise ValueError('Item ID already exists')
```

```
    self.merchandise[item_id] = Merchandise(item_id, name, team_id, revenue,  
                                             description)
```

→ In **RevenueReport** Class Contains a merchandise dictionary and Methods of CRUD operations.

Create:

In create method, It verifies the item_id, If it is already present then it rises value error. If not it adds a new merchandise item to the collection.

```
def read(self, item_id):  
    if item_id not in self.merchandise:  
        raise ValueError("Item ID does not exist")  
    return self.merchandise[item_id]
```

→ Read:

In read method, It verifies the item_id, If it is not present then it rises value error. if not it retrieves a merchandise item based on its ID.

```
def update(self, item_id, name=None, team_id=None, revenue=None,  
           description=None):  
    if item_id not in self.merchandise:  
        raise ValueError("Item ID does not exist")  
    item = self.merchandise[item_id]  
    if name:  
        item.name = name  
    if team_id:  
        item.team_id = team_id  
    if revenue:  
        item.revenue = revenue  
    if description:  
        item.description = description
```

→ Update:

In update method, It verifies the item_id, If it is not present then it rises value error else it modifies and updates the attributes of a merchandise item by newly Provided name, team ID, revenue, and description.

```
def delete (self, item_id):  
    if item_id not in self.merchandise:  
        raise ValueError("Item ID does not exist")  
    del self.merchandise[item_id]
```

→ In Delete method, when the item_id is not present in the merchandise dictionary then it raises value error or else it removes a merchandise item_id and its information from the collection.

```
def track_team_merchandise_sales(self, team_id):
    total_sales = 0
    for item in self.merchandise.values():
        if item.team_id == team_id:
            total_sales += item.revenue
    return total_sales
```

→Track team merchandise sales:

This method calculates the total sales for a given team based on their team_id.

- For each item, it checks if the team ID matches the provided team_id.
- If a match is found, it adds the revenue of that item to the total_sales by iterating.
- Finally, it returns the total sales for the specified team.

```
def analyze_revenue_streams(self, revenue_data):
    total_revenue = sum(revenue_data)
    average_revenue = total_revenue / len(revenue_data) if revenue_data else 0
    max_revenue = max(revenue_data) if revenue_data else 0
    min_revenue = min(revenue_data) if revenue_data else 0
    return total_revenue, average_revenue, max_revenue, min_revenue
```

→Analyze revenue streams:

- This method analyzes the revenue data provided as a list of revenue values.
- It calculates the total revenue, average revenue, maximum revenue, and minimum revenue from the provided data by using built-in methods.

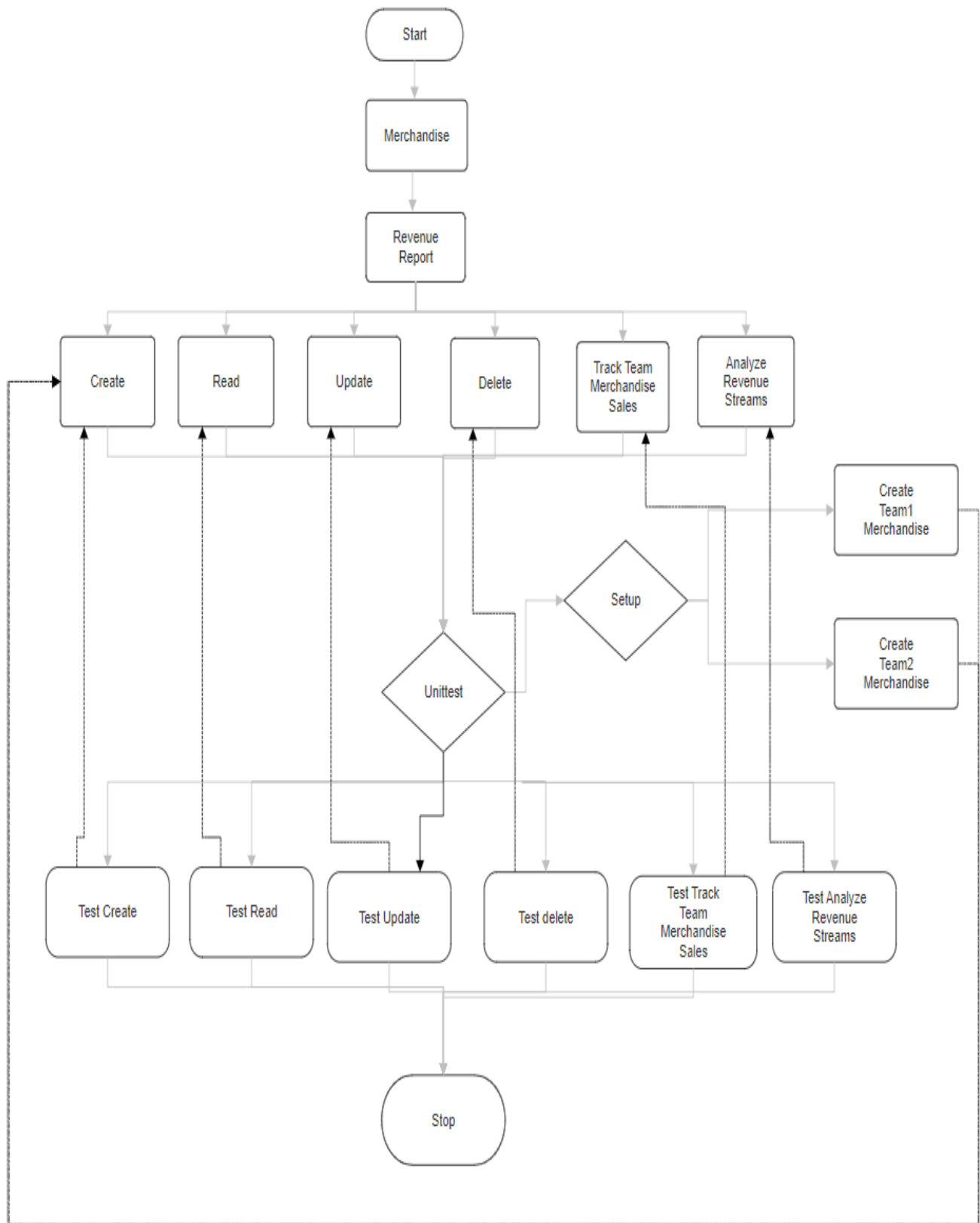
```
if __name__ == '__main__':
    unittest.main(verbosity=10)
```

→It is a main method when the function is called. This runs the unit tests defined in the script with detailed output (verbosity level 10).

Running the unit tests in this way allows you to verify that the functionality of the classes and methods in the code is working as expected.

CHAPTER – 6:

FLOWCHART



CHAPTER – 7:

OUTPUT

➤ Test Case:

```
test_analyze_revenue_streams ( __main__.TestMerchandiseManager) ... ok
test_create ( __main__.TestMerchandiseManager) ... ok
test_delete ( __main__.TestMerchandiseManager) ... ok
test_read ( __main__.TestMerchandiseManager) ... ok
test_track_team_merchandise_sales ( __main__.TestMerchandiseManager) ...
Calculated total sales for team 1: 132500
Item: Computer, Revenue: 40000
Item: Smartphone, Revenue: 30000
Item: Earbuds, Revenue: 2500
Item: Television, Revenue: 60000
ok
test_update ( __main__.TestMerchandiseManager) ... ok

-----
Ran 6 tests in 0.001s

OK

...Program finished with exit code 0
Press ENTER to exit console.[]
```

➤ File Handling:

Shell ×

```
>>> %Run 'Final project with file delete3.py'

Do you want to add revenue data? (yes/no): yes
Enter item ID: 1
Enter item name: Computer
Enter item description: 12th Gen Computer
Enter team ID: 1
Enter revenue: 40000
Revenue data added successfully.
Do you want to add more revenue data? (yes/no): yes
Enter item ID: 2
Enter item name: Smartphone
Enter item description: 100 megapixel camera phone
Enter team ID: 1
Enter revenue: 30000
Revenue data added successfully.
Do you want to add more revenue data? (yes/no): yes
Enter item ID: 3
Enter item name: Earbuds
Enter item description: Noise cancelation Earbuds
Enter team ID: 1
Enter revenue: 2500
Revenue data added successfully.
Do you want to add more revenue data? (yes/no): yes
Enter item ID: 4
Enter item name: Television
Enter item description: 4k HD display T.V.
Enter team ID: 1
Enter revenue: 60000
Revenue data added successfully.
Do you want to add more revenue data? (yes/no): yes
Enter item ID: 5
Enter item name: Cricket Kit
Enter item description: MRF-Company Cricket Kit
Enter team ID: 2
Enter revenue: 15000
Revenue data added successfully.
```

Revenue data added successfully.

Do you want to add more revenue data? (yes/no): yes

Enter item ID: 6

Enter item name: Furniture

Enter item description: Wooden sofa set

Enter team ID: 2

Enter revenue: 30000

Revenue data added successfully.

Do you want to add more revenue data? (yes/no): yes

Enter item ID: 7

Enter item name: Bag

Enter item description: Flexible and lightweight Bag

Enter team ID: 2

Enter revenue: 1800

Revenue data added successfully.

Do you want to add more revenue data? (yes/no): yes

Enter item ID: 8

Enter item name: Smart Watch

Enter item description: Health Tracking Smart Watch

Enter team ID: 2

Enter revenue: 3000

Revenue data added successfully.

Do you want to add more revenue data? (yes/no): no

Do you want to update any item name, description, and revenue? (yes/no): yes

Enter item ID to update name, description, and revenue: 6

Enter new item name: Laptop

Enter new item description: Gaming Laptop

Enter new revenue: 75000

Item name, description, and revenue updated successfully.

Do you want to update more item name, description, and revenue? (yes/no): no

Do you want to delete any item? (yes/no): yes

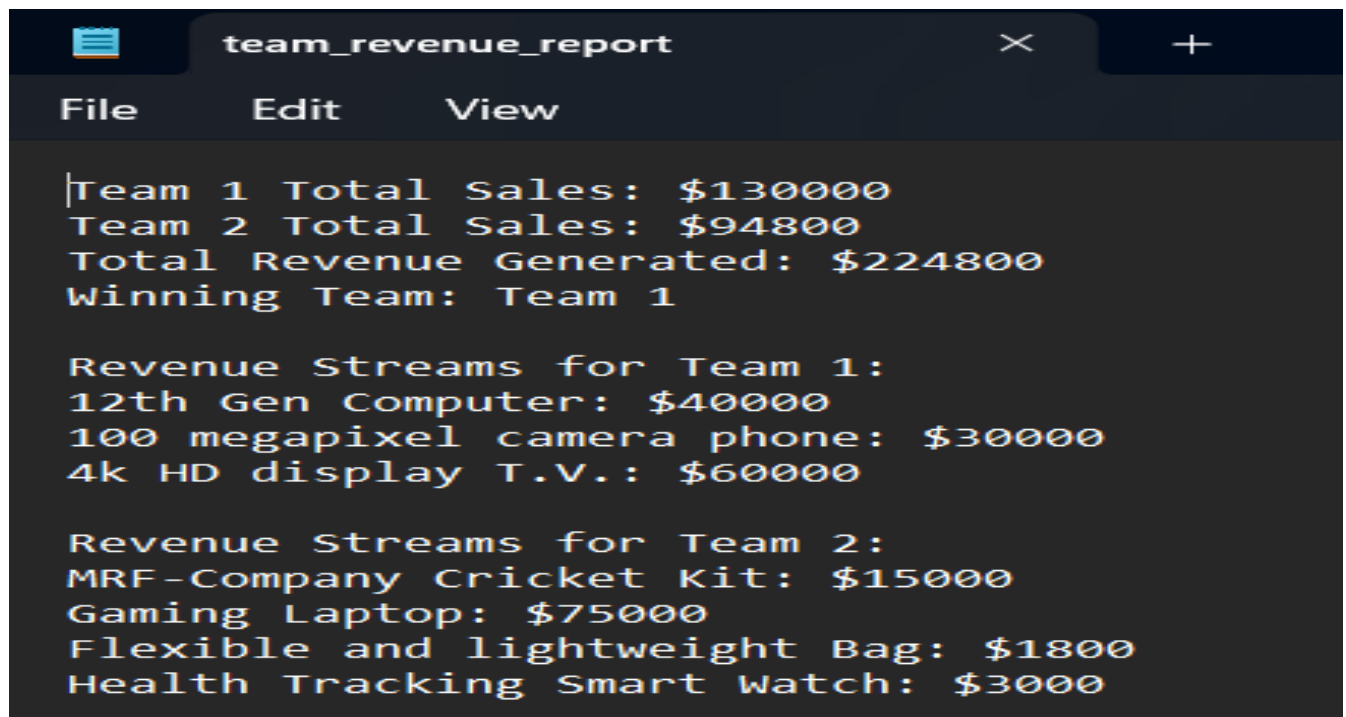
Enter item ID to delete: 3

Item deleted successfully.

Do you want to delete more items? (yes/no): no

Report has been updated and saved to 'team_revenue_report.txt'

>>>



```
Team 1 Total Sales: $130000
Team 2 Total Sales: $94800
Total Revenue Generated: $224800
Winning Team: Team 1

Revenue Streams for Team 1:
12th Gen Computer: $40000
100 megapixel camera phone: $30000
4k HD display T.V.: $60000

Revenue Streams for Team 2:
MRF-Company Cricket Kit: $15000
Gaming Laptop: $75000
Flexible and lightweight Bag: $1800
Health Tracking Smart Watch: $3000
```


CHAPTER – 8:

CONCLUSION

The provided code establishes a system to manage merchandise data and generate revenue reports for a sports team. It employs classes such as `Merchandise`, `MerchandiseManager`, and `RevenueReport` to organize data, track sales by team, and analyze revenue streams. Through this structured approach, the code facilitates efficient management of merchandise sales and revenue analysis for the organization.

CHAPTER – 9:

REFERENCES

- ☐ [Google](#)
- ☐ <https://www.programiz.com/>
- ☐ <https://slidesgo.com/>
- ☐ <https://docs.python.org/3/library/json.html>
- ☐ <https://www.w3schools.com/python/>
- ☐ <https://lucid.app/documents#/templates>

GITHUB ACCOUNT

- ☐ <https://github.com/RRaghu12>