

A Project Report

on

“Olympic player’s performance Analysis”

*Submitted in partial fulfillment of the
requirement for the award of the degree of*

B. TECH (COMPUTER SCIENCE & ENGINEERING)



(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

**Under The Supervision of
DR. SHRADDHA SAGAR
PROFESSOR**

Submitted By

**MAYADHAR YADAV
18SCSE101022**

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA
DECEMBER, 2021**

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled “**OLYMPIC PLAYER’S PERFORMANCE ANALYSIS**” in partial fulfillment of the requirements for the award of B. TECH submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of **DR. SHRADDHA SAGAR** Designation, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

MAYADHAR YADAV

18SCSE1010122

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

DR. Shraddha Sagar
PROFESSOR

TABLE_OF CONTENTS

S. No.	Particulars	Page No.
1	Abstract	4
2	List of Tables	5
3	Introduction	6
4	Literature Survey	7
5	Requirements	9
6	Feasibility Analysis	15
7	Objectives	16
8	Functionality	17
9	Implementation	18
10	Design	36
11	Diagrams	37
12	Result, Conclusion & Future scope	47
13	List of figures	47
14	References	48

ABSTRACT

Data is very important for one's life, because we are human being so we can't remember everything.

This is the main problem because players can't remember everything as a long time, as you know If players don't able to see, where he has done mistakes then how he can improve itself, and also, he doesn't know about past match like where he played which ground and we can say who is against team or opposite team players, then players can't do better performance in next game.

AND NOT ONLY PLAYERS THEY CAN USE EVERY USER'S WHO WANT TO KNOW ABOUT HOW MUCH COUNTRY IS THER AND HOW MUCH COUNTRY WON GOLD, SILVER AND BRONZE IN WHICH YEAR, THIS IS HELPFUL FOR ALL USER WHO WANT TO KNOW OR LEARN TO GENERAL KNOWLEDGE

Essential data can be used as information for many things as analysis, that's why I am making this project, and store data of players' and provide them post-match analysis and his achievement and rewards

And enhance their performance through machine learning, which will helpful for Player's. This can be also used in All sports as well, providing an overall review of one's performance. Helping one to find their areas of improvement by looking into their past performance data and records with country wise.

It will helpful for every player in future because for all players to improve themselves, they will need their past data

Ex. Input: if you want know how much country is there.

Input: `dff['region'].unique().shape`

Output:- **206**

The tools I have used in My project are:

"PyCharm" for the python coding.

"Dialogflow" which is a Google's product for training my Olympic data Analysis

"Heroku", which is platform to make the product online.

The purpose of making this project is to provide everyone a solution of the problem mentioned above. So, the result of this project is a Olympic data Analysis which is easily provide all information to user's

TABLES

Table for Student Data:

S. No	Name	Enrollment Number	Admission Number	Program / Branch	Sem
1	MAYADHAR YADAV	18021011370	18SCSE1010122	B. Tech/ CSE	7

Table for Faculty Data:

	SUPERVISOR	REVIEWER
NAME	Dr Shraddha Sagar	Ms. Lalita Varma
PROFESSION	Professor	Professor

INTRODUCTION

The ancient Olympic game was held at Olympia, Greece, from 776 BC through 393 AD. It returned after 1503 years. The first modern Olympic was held in Athens, Greece in 1896. The 'modern Olympics' comprises all the Games from Athens 1896 to Rio 2016. Baron Pierre de Coubertin presented the idea in 1894.

There are two long periods without any Games between 1912-1920 and 1936-1948, corresponding to WWI and WWII.

Perhaps, the most significant benefit of visual analytics is to ease the understanding of complex data, while representing it in correct, concise, and appropriate way. This manuscript proposes a handy visualization analysis of Olympic games between 1896 to 2016, which comprises in four levels of design system.

This application utilizes concrete analysis examples and claim to provide efficient, effective, functional, and convenient model for users. Discussion, conclusion, future work are summarized along with requesting recommendation for improvement of the limitations, where it was surveyed in evaluation as another section.

Essential data can be used as information for many things as analysis, that's why I am making this project, and store data of players' and provide them post-match analysis and his achievement and rewards

And enhance their performance through machine learning, which will helpful for Player's. This can be also used in All sports as well, providing an overall review of one's performance. Helping one to find their areas of improvement by looking into their past performance data and records with country wise.

LITERATURE SURVEY

1. “**Recruitment Olympic data Analysis**”, **International Research Journal of Engineering and Technology (IRJET)**, vol. 5, Issue: 08, Aug 2018[1].
Authors: Akash Balachandar, Anusha D Kulkarni

In this paper, authors have explained how the **Olympic data Analysis** behaving as a human conversational partner are designed to comprehend a conclusive human response. In today’s world, it is difficult to collect correct information easily while hiring the right candidate. Using simply a **Olympic data Analysis** can be a solution to this problem. Recruiters can use this in day-to-day life to automate time-consuming tasks.

SYSTEM DESIGN

Describing the designing process of interaction between the **Olympic data Analysis** and the user. It uses dialogues systems, and they are of two types:

- 1) Goal Oriented Dialogue Systems.
- 2) General conversation Dialogue Systems.

We use Generative and Selective approaches in recruitment **Olympic data Analysis** which needs a general conversational dialog system. The Machine Learning principle is a core philosophy for both these approaches: Build it, train it, and test it. By using bot characteristics, constraints, dialogue dataset, access flow, and Sequence tokens this model is built. Figure.1: Data-driven Dialogue System.

2. **Implementation of Olympic data Analysis in Online Commerce, and Open Innovation.**
Authors: María D. Illescas-Manzano, Noé Vicente López, Nuno Afonso González and Carmen Cristofol Rodríguez.

In this paper the study describes the **Olympic data Analysis** journey and focuses on its implementation within the digital marketing strategy . The main goal was to apply a **Olympic data Analysis** via all browser’s platform to increase the number of leads, comparing the **Olympic data Analysis** with the previous strategy used by the user’s to obtain contact information. This research work takes a step further and shows that implementing a **Olympic data Analysis** through the Many platform by a user’s that markets online has a positive impact on the capturing of leads, as opposed to the results obtained by authors such as Luo et al. and Leung et al. A **Olympic data Analysis** platform used with the intention of obtaining leads seems to be an agile and powerful tool; in fact; the main conclusion of this work is that including this method can be one of the main axes of obtaining information about consumers with the aim of performing marketing actions in a two-way communication that facilitates by user.

3. Literature survey on various Olympic data analysis.

Authors: Harsha Pariyani, Anshika Sinha, Preeti Bhat, Roshni Rote, Asst. Prof. N. A. Mulla.

In this paper they describe the various study about Olympic data analysis like what are Olympic data analysis? how they can communicate? How their working done? What are their types?

A/c to them a Olympic data analysis is AI computer software that can act as a conversation through textual or auditory methods. In simple words a Olympic data analysis can conduct smart search data either via text or voice. The core of the Olympic data analysis analyses a customer's data using the AI which gives responses to them. They are trained through various data using machine learning which can interact with humans and become more accurate with each interaction. They divide the Olympic data analysis in categories on the basis of what are support, what are their skills and what type of work they assist.

In support part they use to master a single domain.

A skill based Olympic data analysis does not require a lot pf contextual awareness. And an assistant Olympic data analysis is the middle of between a skill and support Olympic data analysis. They know a little about the variety of topics.

4. "Intelligent Olympic data analysis for easy web-analytics".

Author: Ravi R.

In this paper, a comparison is done among Olympic data analysis based on their ease of usage, by using different analytic tools. They describe that the Olympic data analysis is build using Artificial Intelligence Markup Language contains analytics' raw data and the required data is fetched from analytics tool's raw data. They said every website note all the details user made. AIML comprises of possible queries and their responses. It consists of 3 elements such as template, categories and pattern. Each category contains pattern and a template. Patterns are the possible queries that the bot-user may type in and the template is the response to the respective pattern. They describe 3 types of queries can be considered Domain related query, General Query and None of them. The user can type to web analytics their query related to and will get an immediate reply. This Olympic data analysis tools are mastered to avoid the time taking task.

They have also mentioned some application of Olympic data analysis which are:

- *Experience and Services: Communicate in multiple language, can handle customer service request, manage account setting.*
- *Recruitment: Scheduling meeting, Candidate's interaction, Answer & Questions.*
- *Healthcare: Provide status on staffing, Communication with doctors.*
- *Government: Resolve parking tickets, Visa applications management.*
- *Smart Home Devices: control entertainment devices, control heating, manage security.*
- *Transportation: Remotely lock vehicles, retrieve information.*

REQUIREMENTS

Concept Used:

- Machine Learning

Coding Platforms:

- PyCharm Community Edition IDE

Training Platform:

- Dialogflow

Languages Used:

- Python

Deployment Platform Used:

- Heroku
-

DESCRIPTION OF REQUIREMENTS

MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

But, using the classic algorithms of machine learning, text is considered as a sequence of keywords; instead, an approach based on semantic analysis mimics the human ability to understand the meaning of a text.

Some Machine Learning Methods

Machine learning algorithms are often categorized as supervised or unsupervised.

Supervised machine learning algorithms can apply what has been learned in the past to new data using labelled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

In contrast, **unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labelled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabelled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabelled data.

Semi-supervised machine learning algorithms fall somewhere in between supervised and unsupervised learning, since they use both labelled and unlabelled data for training – typically a small amount of labelled data and a large amount of unlabelled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labelled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabelled data generally doesn't require additional resources.

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behaviour within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information.

PyCharm Community

PyCharm is a hybrid-platform developed by JetBrains as an IDE for Python. It is commonly used for Python application development. Some of the unicorn organizations such as Twitter, Facebook, Amazon, and Pinterest use PyCharm as their Python IDE!

It supports two versions: v2.x and v3.x.

We can run PyCharm on Windows, Linux, or Mac OS. Additionally, it contains modules and packages that help programmers develop software using Python in less time and with minimal effort. Further, it can also be customized according to the requirements of developers.

Features of PyCharm:

1. Intelligent Code Editor:

- ❖ It helps us write high-quality codes!
- ❖ It consists of colour schemes for keywords, classes, and functions. This helps increase the readability and understanding of the code.
- ❖ It helps identify errors easily.
- ❖ It provides the autocomplete feature and instructions for the completion of the code.

2. Code Navigation:

- ❖ It helps developers in editing and enhancing the code with less effort and time.
- ❖ With code navigation, a developer can easily navigate to a function, class, or
- ❖ file.

Reinforcement machine learning algorithms is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behaviour within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information.

PyCharm Community

PyCharm is a hybrid-platform developed by JetBrains as an IDE for Python. It is commonly used for Python application development. Some of the unicorn organizations such as Twitter, Facebook, Amazon, and Pinterest use PyCharm as their Python IDE!

It supports two versions: v2.x and v3.x.

We can run PyCharm on Windows, Linux, or Mac OS. Additionally, it contains modules and packages that help programmers develop software using Python in less time and with minimal effort. Further, it can also be customized according to the requirements of developers.

Features of PyCharm:

1. Intelligent Code Editor:

- ❖ It helps us write high-quality codes!
- ❖ It consists of colour schemes for keywords, classes, and functions. This helps increase the readability and understanding of the code.
- ❖ It helps identify errors easily.
- ❖ It provides the autocomplete feature and instructions for the completion of the code.

2. Code Navigation:

- ❖ It helps developers in editing and enhancing the code with less effort and time.
- ❖ With code navigation, a developer can easily navigate to a function, class, or
- ❖ file.

- ❖ A programmer can locate an element, a symbol, or a variable in the source code within no time.

A programmer can locate an element, a symbol, or a variable in the source code within no time.

3. Refactoring:

- ❖ It has the advantage of making efficient and quick changes to both local and global variables.
- ❖ Refactoring in PyCharm enables developers to improve the internal structure without changing the external performance of the code.
- ❖ It also helps split up more extended classes and functions with the help of the extract method.

4. Assistance for Many Other Web Technologies:

- ❖ It helps developers create web applications in Python.
- ❖ It supports popular web technologies such as HTML, CSS, and JavaScript.
- ❖ Developers have the choice of live editing with this IDE. At the same time, they can preview the created/updated web page.
- ❖ The developers can follow the changes directly on a web browser.
- ❖ PyCharm also supports AngularJS and NodeJS for developing web applications.

5. Support for Popular Python Web Frameworks:

- ❖ PyCharm supports web frameworks such as Django.
- ❖ It provides the autocomplete feature and suggestions for the parameters of Django.
- ❖ It helps in debugging the codes of Django.
- ❖ It also assists web2py and Pyramid, the other popular web frameworks.

6. Assistance for Python Scientific Libraries:

- ❖ PyCharm supports Python's scientific libraries such as Matplotlib, NumPy, and Anaconda.
- ❖ These scientific libraries help in building projects of Data Science and Machine Learning.
- ❖ It consists of interactive graphs that help developers understand data.

- ❖ It is capable of integrating with various tools such as IPython, Django, and Pytest. This integration helps innovate unique solutions.

Python

Nowadays, Python is in great demand. It is widely used in the software development industry. There are 'n' number of reasons for this.

High-level object-oriented programming language: Python includes effective symbolism.

Rapid application development: Because of its concise code and literal syntax, the development of applications gets accelerated. The reason for its wide usability is its simple and easy-to-master syntax. The simplicity of the code helps reduce the time and cost of development.

Dynamic typescript: Python has high-level incorporated data structures blended with dynamic typescript and powerful binding.

Features of Python:

- ❖ Python supports code reusability and modularity.
- ❖ It has a quick edit-inspect-debug cycle.
- ❖ Debugging is straightforward in Python programs.
- ❖ It has its own debugger written in Python itself, declaring to Python's reflective power.
- ❖ Python includes a plethora of third-party components present in the Python Package Index (PyPI)

Data Preparation

120 Years of Olympic Games is being examined for representation after extensive research into a variety of datasets. Its dataset has several advantages, including being a large-scale dataset that spans the years 1896 to 2016. Last but not least, the Olympic Games dataset is deemed to be worth studying from the standpoint of visual representation and giving the optimal user experience.

```
# LOAD ATHLETES EVENTS DATA
dataOlympics <-
read_csv("datasets/athleteEvents.csv", col_types = cols(
  ID = col_character(),
  Name = col_character(),
  Sex = col_factor(levels = c("M", "F")),
  Age = col_integer(),
  Height = col_double(),
  Weight = col_double(),
  Team = col_character(),
  NOC = col_character(),
  Games = col_character(),
  Year = col_integer(),
  Season = col_factor(levels =
c("Summer", "Winter")),
  City = col_character(),
  Sport = col_character(),
  Event = col_character(),
  Medal = col_factor(levels =
c("Gold", "Silver", "Bronze"))
)
)glimpse(dataOlympics)
```

Which nations took home the most medals?

This is a second question that you can answer, and it would be interesting to also be able to visualize the proportion of gold, silver, and bronze medals that each nation has accumulated.

Let's say you want to get the top 30.

```
# THE TOTAL NUMBER OF MEDALS GIVEN TO EACH TEAMmedalCounts <-
dataOlympics %>% filter(!is.na(Medal))%>%
  group_by(NOC, Medal, Event, Games) %>%
  summarize(isMedal=1)medalCounts <- medalCounts %>%
  group_by(NOC, Medal) %>%
  summarize(Count= sum(isMedal))medalCounts <-
left_join(medalCounts, NOCs, by= "NOC" )medalCounts <- medalCounts
%>%
mutate (Team = region)medalCounts <- medalCounts %>% select(
Medal, Team, Count)# ORDERING TEAM BY TOTAL MEDAL COUNTlevelsTeam
<- medalCounts %>%
  group_by(Team) %>%
  summarize(Total=sum(Count)) %>%
  arrange(desc(Total)) %>%
  select(Team) %>%
  slice(30:1)medalCounts$Team <- factor(medalCounts$Team,
levels=levelsTeam$Team)medalCounts <- medalCounts %>% filter(Team
!= "NA")# PLOT MEDAL COUNTSggplot(medalCounts, aes(x=Team, y=Count,
fill=Medal)) +
  geom_col() +
  coord_flip() +
  scale_fill_manual(values=c("gold3","gray64","sienna")) +
  labs(x = "Nations", y = "Count",
       title="Top 30 - Nations with the most medals won in
history",
       subtitle = "Olympic Games from 1896 to 2016")
```


Heroku

Heroku is a cloud service platform whose popularity has grown in recent years. Heroku is so easy to use that it's a top choice for many development projects.

With a special focus on supporting customer-focused apps, it enables simple application development and deployment. Since the Heroku platform manages hardware and servers, businesses that use Heroku are able to focus on perfecting their apps. And not the infrastructure that supports them.

Heroku, a Platform-as-a-Service solution, is generally easy-to-use. But it's most beneficial to businesses in specific situations. Heroku has a free service model for small projects. Also, tiered service packages exist for cases where more complex business needs must be addressed.

The Heroku cloud service platform is based on a managed container (called dynos within the Heroku paradigm) system. It has integrated data services and a powerful ecosystem for deploying and running modern applications.

Features of Heroku:

1. Heroku Accommodates Many Development Languages:

Heroku supports several programming languages that are used as a web application deployment model. As one of the first cloud platforms, Heroku has been in development since June 2007. Back then, it supported only the Ruby programming language.

But now it also supports Java, Node.js, Scala, Clojure, Python, PHP, and Go. This

means a variety of developers can look to Heroku for an inexpensive way to scale their application, no matter their preferred development language.

2. Heroku Supports Diverse Solutions:

Heroku also provides custom buildpacks, where developers can deploy apps in any other programming language. For this reason, Heroku is a polyglot platform. It lets the developer build, run, and scale applications in a similar manner across all programming languages.

Polymorphism and scalability are reasons why Heroku is often seen as a preferred platform amongst developers.

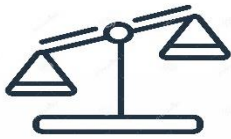
3. Heroku Dynos Enable Easy Development and Better Usability:

Applications that are run on Heroku typically have unique domain names, which are used to route HTTP requests to the correct container. Applications as services use application containers. Containers are designed to package and run services. Each of the application containers is a smart container on a reliable, fully-managed runtime environment.

4. Heroku Lets Developers Scale Applications Instantly:

This is accomplished either by increasing the number of dynos or by changing the type of dyno in which the app runs. When the application can scale so easily, the user can always expect more speed when using that application.

FEASIBILITY ANALYSIS



dreamstime.com



QUITE
FEASIBLE

FREE
OF
COST

PYTHON BASED
PROJECT



alamy



EASY
TO
USE

WORK
VERY
SMOOTHLY

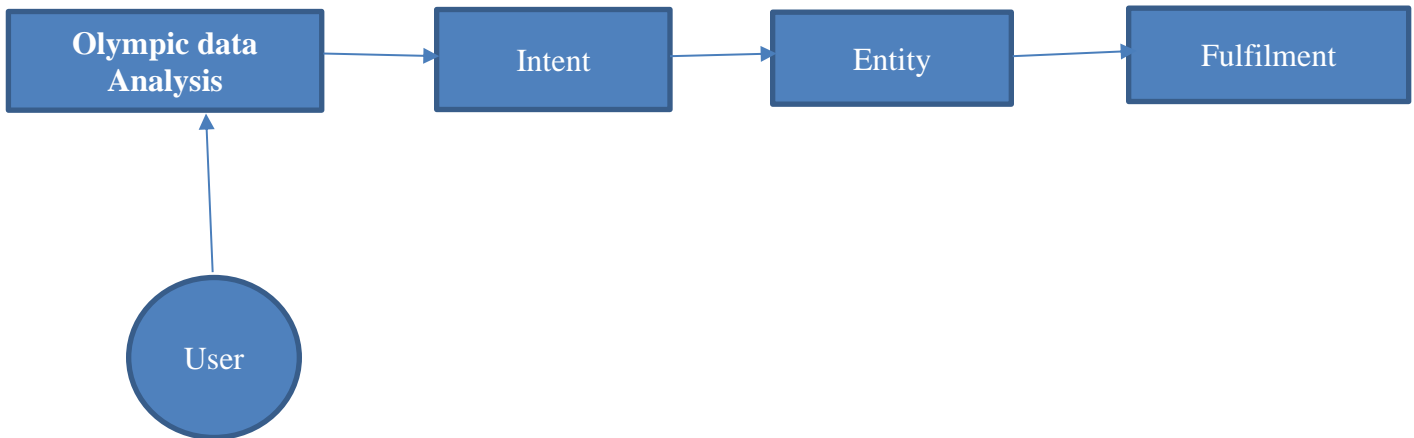
DEPLOYED
WITH
HEROKU

OBJECTIVE

- ❖ The main aim of this project is to search a “DATA ANALYSIS” Which can Search the data of any country in single command, and it can search with us too both via text messages and voice message as well.
- ❖ Purpose of creating this olympic data analysis is to provide easy way and basic facilities which the user’s need.
- ❖ With this system, the tasks would be performed in appropriate amount of time and more efficiently.
- ❖ Additional benefit is that we can search normally with the olympic data analysis as a friend.
- ❖ When user search the data or intent what they wanted to be done, the output will be their desirable result.
- ❖ The main purpose is to provide the basic facilities which the users need.
- ❖ With this system, the tasks would be performed in appropriate amount of time and more efficiently.

Functionality & Design

In this project Dialogflow is used. It is a **Olympic data Analysis** building framework. The query done by user and according to that the **Olympic data Analysis** extract the intent of user.



- ❖ User search with Olympic data Analysis.
- ❖ In intent part query done by the user. (Intent means user telling to Olympic data Analysis or want some work done by the Olympic data Analysis.
- ❖ Behind the scene Olympic data Analysis extract the entity from intent.
- ❖ Depending on the Olympic data Analysis entity can be one or more.
- ❖ Most important stage is fulfilment, in this stage we are making an app using python or flask API. API which work is to do search data.
- ❖ Flask API do search and result send to **Olympic data Analysis** as a reply.
- ❖ This how the entire things will work.
- ❖ So, in user side there is a Olympic data Analysis, which can be on any platform like: chrome, yahoo, or any browser etc.
- ❖ So, the user search with Olympic data Analysis, Olympic data Analysis understand the intent and extract the entity and send to our flask web app where search done and result sent back to user as a reply.

IMPLEMENTATION

Step 1: Importing libraries

```
# importing all necessary libraries
import pandas as pd
import numpy as np
```

Step 2: Understanding database

When dealing with Olympic data, we have two CSV files. One containing outturn sports-related costs of the Olympic Games of all years. And other is containing the information about athletes of all years when they participated with information.

CSV data file can be download from here: [Datasets](#)

Step 3: Data cleaning and formatting

We imported both the datasets using the [.read_csv\(\)](#) method into a dataframe using pandas and displayed the first 5 rows of each dataset.

```
data = pd.read_csv('athlete_events.csv') # read file

# data.head() display first 5 entry
# data.describe describe about model
# data.info give info about data
print(data.head(), data.describe(), data.info())
```

Output:

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 271116 entries, 0 to 271115
Data columns (total 15 columns):
#   Column  Non-Null Count  Dtype
---  ---      -
0   ID       271116 non-null  int64
1   Name     271116 non-null  object
2   Sex      271116 non-null  object
3   Age      261642 non-null  float64
4   Height   210945 non-null  float64
5   Weight   208241 non-null  float64
6   Team     271116 non-null  object
7   NOC      271116 non-null  object
8   Games    271116 non-null  object
9   Year     271116 non-null  int64
10  Season   271116 non-null  object
11  City     271116 non-null  object
12  Sport    271116 non-null  object
13  Event    271116 non-null  object
14  Medal    39783 non-null   object
dtypes: float64(3), int64(2), object(10)
memory usage: 31.0+ MB

```

```

   ID      Name  ...      Event  Medal
0    1  A Dijiang  ...  Basketball Men's Basketball  NaN
1    2  A Lamusi  ...  Judo Men's Extra-Lightweight  NaN
2    3  Gunnar Nielsen Aaby  ...  Football Men's Football  NaN
3    4  Edgar Lindenau Aabye  ...  Tug-Of-War Men's Tug-Of-War  Gold
4    5  Christine Jacoba Aaftink  ...  Speed Skating Women's 500 metres  NaN

```

```

[5 rows x 15 columns]
count  271116.000000  261642.000000  ...  208241.000000  271116.000000
mean    68248.954396    25.556898  ...    70.702393    1978.378480
std     39022.286345     6.393561  ...    14.348020     29.877632
min       1.000000    10.000000  ...    25.000000    1896.000000
25%     34643.000000    21.000000  ...    60.000000    1960.000000
50%     68205.000000    24.000000  ...    70.000000    1988.000000
75%    102097.250000    28.000000  ...    79.000000    2002.000000
max    135571.000000    97.000000  ...    214.000000    2016.000000

```

```

[8 rows x 5 columns] None
   ID      Name Sex  ...  Medal  region  notes
3    4  Edgar Lindenau Aabye  M  ...  Gold  Denmark  NaN
42  17  Paavo Johannes Aaltonen  M  ...  Gold  Finland  NaN
44  17  Paavo Johannes Aaltonen  M  ...  Gold  Finland  NaN
48  17  Paavo Johannes Aaltonen  M  ...  Gold  Finland  NaN
60  20    Kjetil Andr Aamodt  M  ...  Gold  Norway  NaN

```

```

[5 rows x 17 columns]
>>> |

```

Step 4: Merging two DataFrame

Here we are going to merge two dataframe using `pandas.merge()` in python.

```
country_noc_data_csv_file
```

```
regions = pd.read_csv('datasets_31029_40943_noc_regions.csv')  
print(regions.head())
```

```
# merging to data and regions frame
```

```
merged = pd.merge(data, regions, on='NOC', how='left')  
print(merged.head())
```

Output:

```
   NOC   region      notes  
0  AFG  Afghanistan      NaN  
1  AHO   Curacao  Netherlands Antilles  
2  ALB   Albania      NaN  
3  ALG   Algeria      NaN  
4  AND   Andorra      NaN  
   ID      Name Sex ... Medal   region notes  
0   1      A Dijiang  M ...   NaN     China  NaN  
1   2      A Lamusi  M ...   NaN     China  NaN  
2   3  Gunnar Nielsen Aaby  M ...   NaN   Denmark  NaN  
3   4  Edgar Lindenau Aaby  M ...   Gold   Denmark  NaN  
4   5  Christine Jacoba Aaftink  F ...   NaN  Netherlands  NaN  
  
[5 rows x 17 columns]  
>>> |
```

Data analysis of Gold medalists

Creating a new data frame including only gold medalists.

```
# creating goldmedal dataframes
goldMedals = merged[(merged.Medal == 'Gold')]
print(goldMedals.head())
```

Output :

```
===== RESTART: C:/Users/Rahul/Downloads/athlete_events.csv/olympicblog.py =====
   ID      Name Sex  ... Medal  region  notes
3   4  Edgar Lindenau Aabye  M  ...   Gold  Denmark  NaN
42  17 Paavo Johannes Aaltonen  M  ...   Gold  Finland  NaN
44  17 Paavo Johannes Aaltonen  M  ...   Gold  Finland  NaN
48  17 Paavo Johannes Aaltonen  M  ...   Gold  Finland  NaN
60  20   Kjetil Andr Aamodt  M  ...   Gold  Norway  NaN

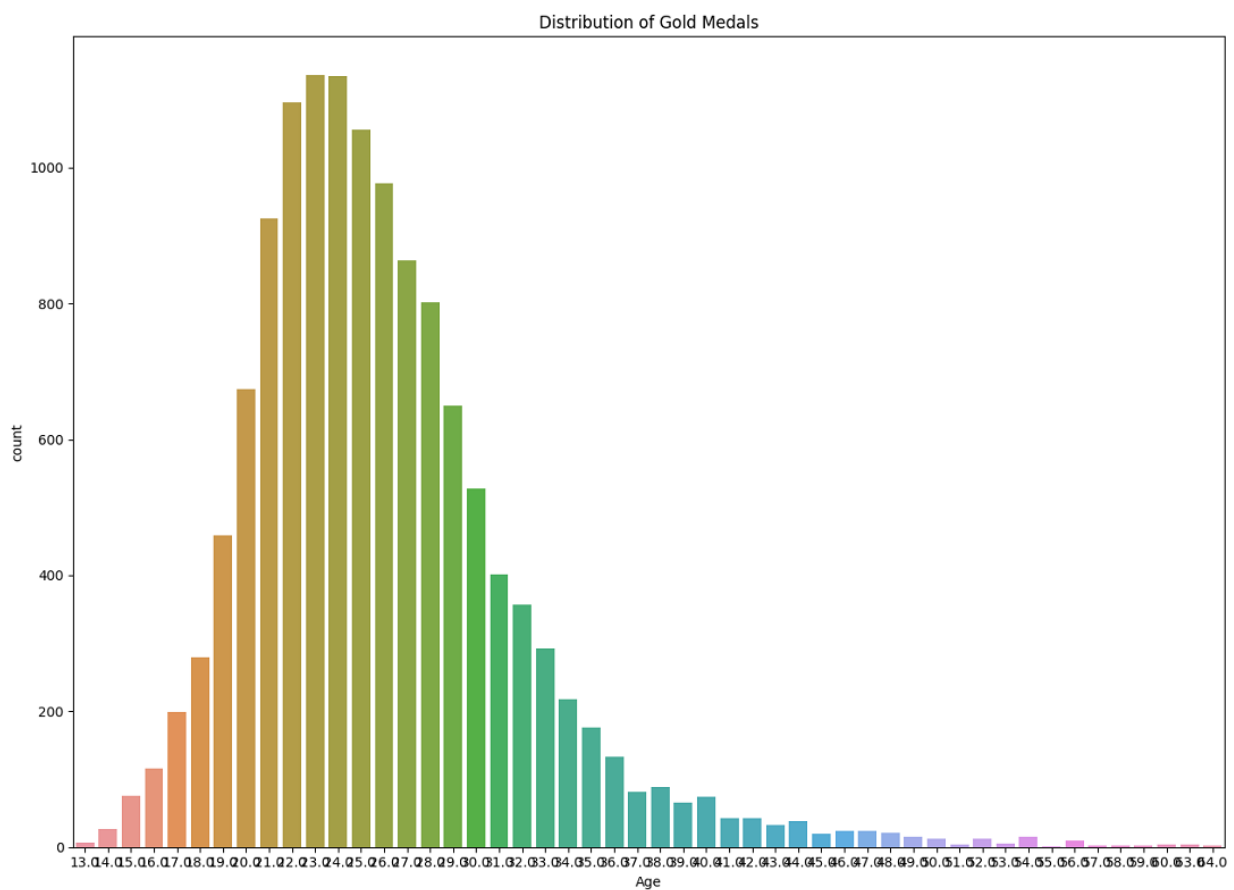
[5 rows x 17 columns]
>>> |
```

Gold medalist in respect of age:

Here we are going to create a graph of the number of gold medals with respect to age. For this, we will create [countplot](#) for graph representation which shows the X-axis as the age of the players and the Y-axis represent the number of medals.


```
plt.figure(figsize=(20, 10))
plt.title('Distribution of Gold Medals')
sns.countplot(goldMedals['Age'])
plt.show()
```

Output :



Number of athletes who are gold medalists and whose age is greater than 50 with their info.

```
goldMedals = merged[(merged.Medal == 'Gold')]
print('The no of athletes is',
      goldMedals['ID'][goldMedals['Age'] > 50].count(), '\n')
print(goldMedals[goldMedals['Age'] > 50])
```

Output :

```
===== RESTART: C:/Users/Rahul/Downloads/athlete_events.csv/olympicblog.py =====
The bo of athletes is 65

   ID      Name Sex ... Medal  region  notes
5078  2894  Derek Swithin Allhusen  M ...  Gold    UK    NaN
7961  4404  Johan August Anker      M ...  Gold   Norway  NaN
13394 7272  Nikolaus "Klaus" Balkenhol  M ...  Gold  Germany  NaN
13396 7272  Nikolaus "Klaus" Balkenhol  M ...  Gold  Germany  NaN
24897 13033 Gustaf Adolf Boltenstern, Jr.  M ...  Gold   Sweden  NaN
...    ...      ... .. ...    ...    ...    ...
261102 130662  Robert W. Williams, Jr.  M ...  Gold    USA    NaN
261671 130936  Walter Winans           M ...  Gold    USA    NaN
261675 130936  Walter Winans           M ...  Gold    USA    NaN
262865 131533  Emily Woodruff (Smiley-)  F ...  Gold    USA    NaN
266293 133226  Mahonri Mackintosh Young  M ...  Gold    USA    NaN

[65 rows x 17 columns]
>>> |
```

Women who can play in summer

Display all women athletes who have played in the summer season and it show the increase in women athletes after a long period via graphical representation.

```
womenInOlympics = merged[(merged.Sex == 'F') &
                          (merged.Season == 'Summer')]
print(womenInOlympics.head(10))

sns.set(style="darkgrid")
plt.figure(figsize=(20, 10))
sns.countplot(x='Year', data=womenInOlympics)
plt.title('Women medals per edition of the Games')
plt.show()
```

Output :

```
===== RESTART: C:/Users/Rahul/Downloads/athlete_events.csv/olympicblog.py =====
   ID      Name Sex ... Medal      region notes
26   8  Cornelia "Cor" Aalten (-Strannood)  F ...   NaN Netherlands  NaN
27   8  Cornelia "Cor" Aalten (-Strannood)  F ...   NaN Netherlands  NaN
32  13      Minna Maarit Aalto  F ...   NaN Finland      NaN
33  13      Minna Maarit Aalto  F ...   NaN Finland      NaN
79  21  Ragnhild Margrethe Aamodt  F ...  Gold Norway      NaN
80  22      Andreea Aanei  F ...   NaN Romania      NaN
91  29  Willemien Aardenburg  F ... Bronze Netherlands  NaN
105 37      Ann Kristin Aarnes  F ... Bronze Norway      NaN
135 49      Moonika Aava  F ...   NaN Estonia      NaN
136 49      Moonika Aava  F ...   NaN Estonia      NaN
```

```
[10 rows x 17 columns]
```

Top 5 countries who won the most medals

Here we are going to print the top 5 countries and show them in the graph with [catplot](#).

```
print(goldMedals.region.value_counts().reset_index(name='Medal').head())
```

```
totalGoldMedals = goldMedals.region.value_counts()
```

```

.reset_index(name='Medal').head(5)
g = sns.catplot(x="index", y="Medal", data=totalGoldMedals,
               height=6, kind="bar", palette="muted")
g.despine(left=True)
g.set_xlabels("Top 5 countries")
g.set_ylabels("Number of Medals")
plt.title('Medals per Country')
plt.show()

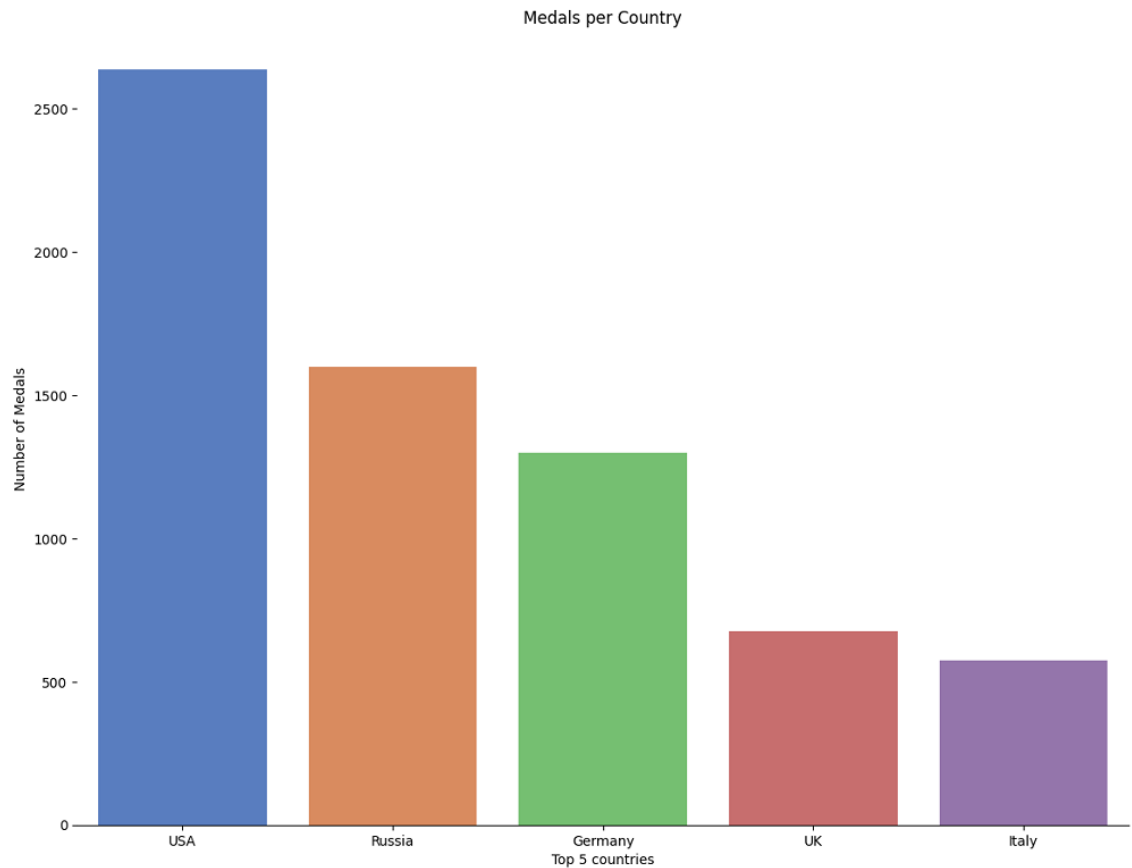
```

output:

ID	Name	Sex	...	Medal	region	notes
26	Cornelia "Cor" Aalten (-Strannood)	F	...	NaN	Netherlands	NaN
27	Cornelia "Cor" Aalten (-Strannood)	F	...	NaN	Netherlands	NaN
32	Minna Maarit Aalto	F	...	NaN	Finland	NaN
33	Minna Maarit Aalto	F	...	NaN	Finland	NaN
79	Ragnhild Margrethe Aamodt	F	...	Gold	Norway	NaN
80	Andreea Aanei	F	...	NaN	Romania	NaN
91	Willemien Aardenburg	F	...	Bronze	Netherlands	NaN
105	Ann Kristin Aarnes	F	...	Bronze	Norway	NaN
135	Moonika Aava	F	...	NaN	Estonia	NaN
136	Moonika Aava	F	...	NaN	Estonia	NaN

[10 rows x 17 columns]

	index	Medal
0	USA	2638
1	Russia	1599
2	Germany	1301
3	UK	678
4	Italy	575



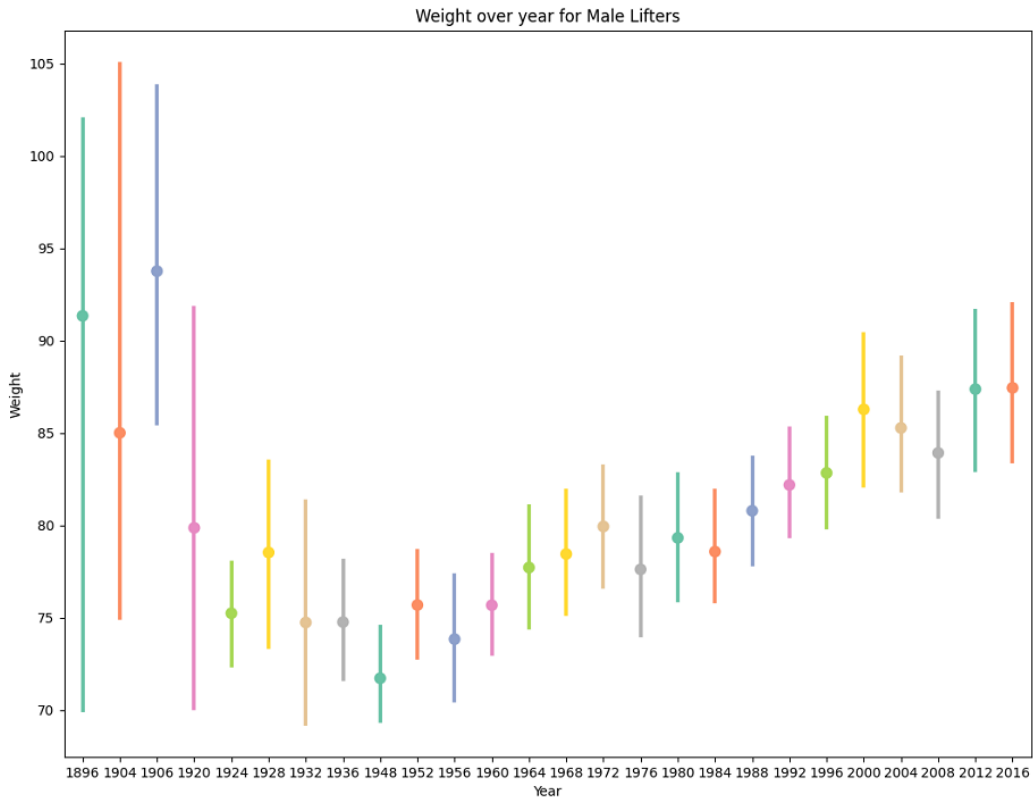
Players weight Analysis

Here we are going to see how weight over year for Male Lifters via graphical representation using [pointplot](#).

```
MenOverTime = merged[(merged.Sex == 'M') &
                      (merged.Season == 'Summer')]
w1MenOverTime = MenOverTime.loc[MenOverTime['Sport'] == 'Weightlifting']


plt.figure(figsize=(20, 10))
sns.pointplot('Year', 'Weight', data=w1MenOverTime, palette='Set2')
plt.title('Weight over year for Male Lifters')
plt.show()
```

Output :



TRAINING IMAGES.

Olympics Analysis



Select an Option

- Medal Tally
- Overall Analysis
- Country-wise Analysis
- Athlete wise Analysis

Medal Tally

Select Year

Overall ▼

Select Country

Overall ▼

Overall Tally

	region	Gold	Silver	Bronze	total
0	USA	1035	682	788	2545
1	Russia	592	498	487	1577
2	Germany	444	457	491	1392
3	UK	278	317	388	895
4	France	234	256	287	777
5	China	228	163	154	545
6	Italy	219	191	198	608
7	Hungary	178	154	172	504
8	Sweden	158	175	188	513
9	Australia	158	171	197	518
10	Japan	142	134	161	437
11	Finland	104	86	128	318

Olympics Analysis



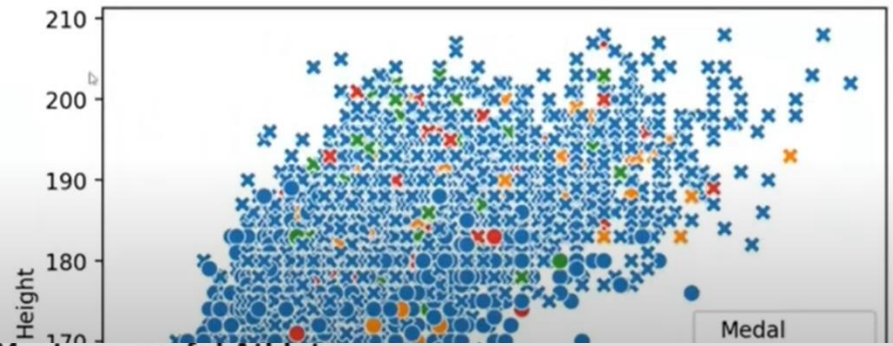
Select an Option

- Medal Tally
- Overall Analysis
- Country-wise Analysis
- Athlete wise Analysis

Height Vs Weight

Select a Sport

Overall



Most successful Athletes

Select a Sport

Athletics

	Name	Medals	Sport	region
0	Paavo Johannes Nurmi	12	Athletics	Finland
12	Raymond Clarence "Ray" Ewry	10	Athletics	USA
22	Frederick Carlton "Carl" Lewis	10	Athletics	USA
32	Martin Joseph Sheridan	9	Athletics	USA
47	Allyson Michelle Felix	9	Athletics	USA
57	Merlene Joyce Ottey-Page	9	Athletics	Jamaica
76	Veronica Angella Campbell-Brown	8	Athletics	Jamaica
87	Viljo Eino "Ville" Ritola (Koukkari-)	8	Athletics	Finland
96	Justin St. Leo	8	Athletics	Jamaica
106	Irena Szewiska-Kirszenstein	7	Athletics	Poland

Olympics Analysis



Select an Option

- Medal Tally
- Overall Analysis
- Country-wise Analysis
- Athlete wise Analysis

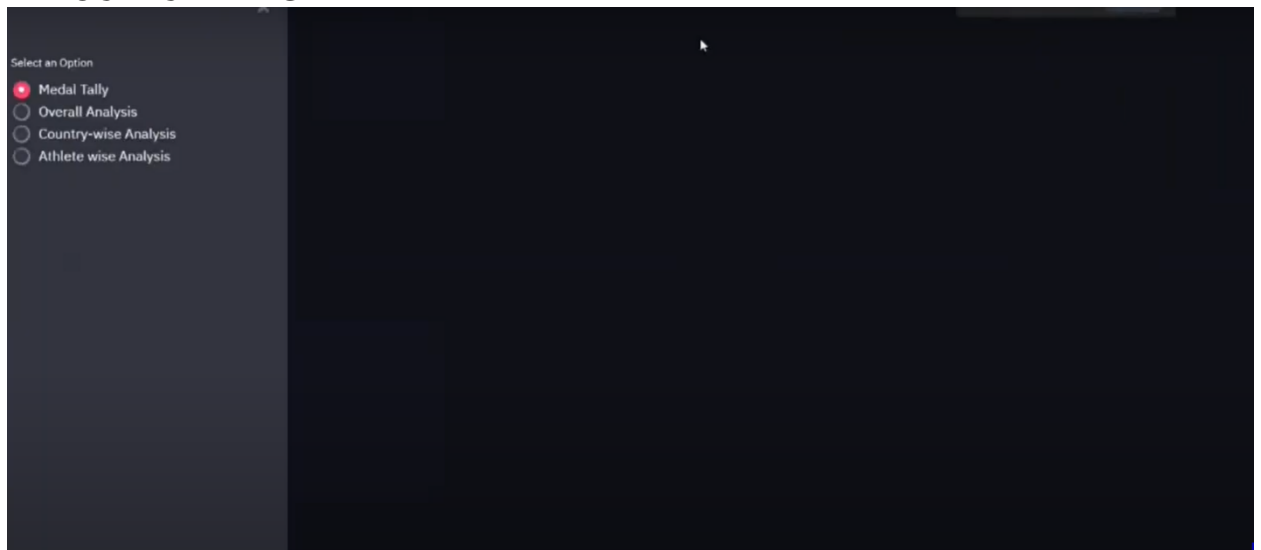
FLASK API IMAGE

```
import streamlit as st
import pandas as pd

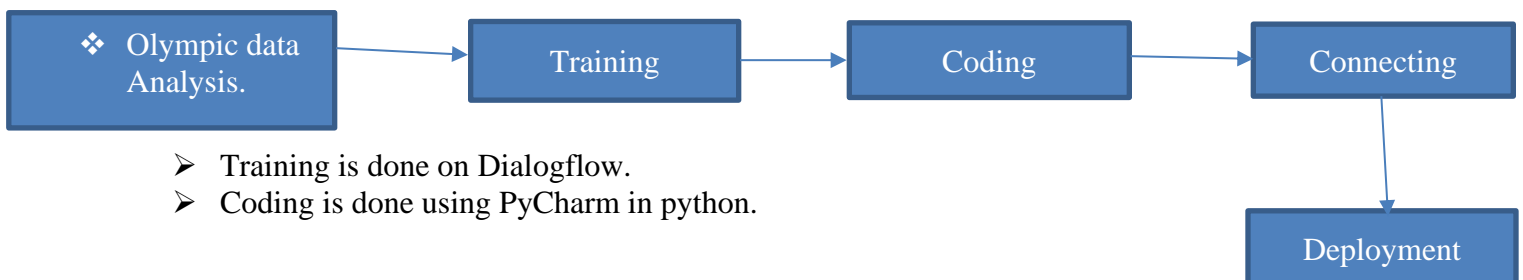
st.sidebar.radio(
    'Select an Option',
    ('Medal Tally', 'Overall Analysis', 'Country-wise Analysis', 'Athlete wise Analysis')
)

7.0 pyparsing-2.4.7 pyrsistent-0.18.0 python-dateutil-2.8.2 pytz-2021.1 pywin32-301 pywinpty-1.1.3 pyzm
2.1.0 requests-2.26.0 six-1.16.0 smmap-4.0.0 streamlit-0.85.0 terminado-0.10.1 testpath-0.5.0 toml-0.10
toolz-0.11.1 tornado-6.1 traitlets-5.0.5 typing-extensions-3.10.0.0 tzlocal-2.1 urllib3-1.26.6 validato
3.18.2 watchdog-2.1.7 wwidh-0.2.5 webencodings-0.5.1 widetextextension-3.5.1 zipp-3.5.0
```

API OUTPUT IMAGE



DESIGN



- ❖ After that our app is going to connect with Olympic data Analysis.
- .
- ❖ Then we deploy our Olympic data Analysis.
- on Browser.

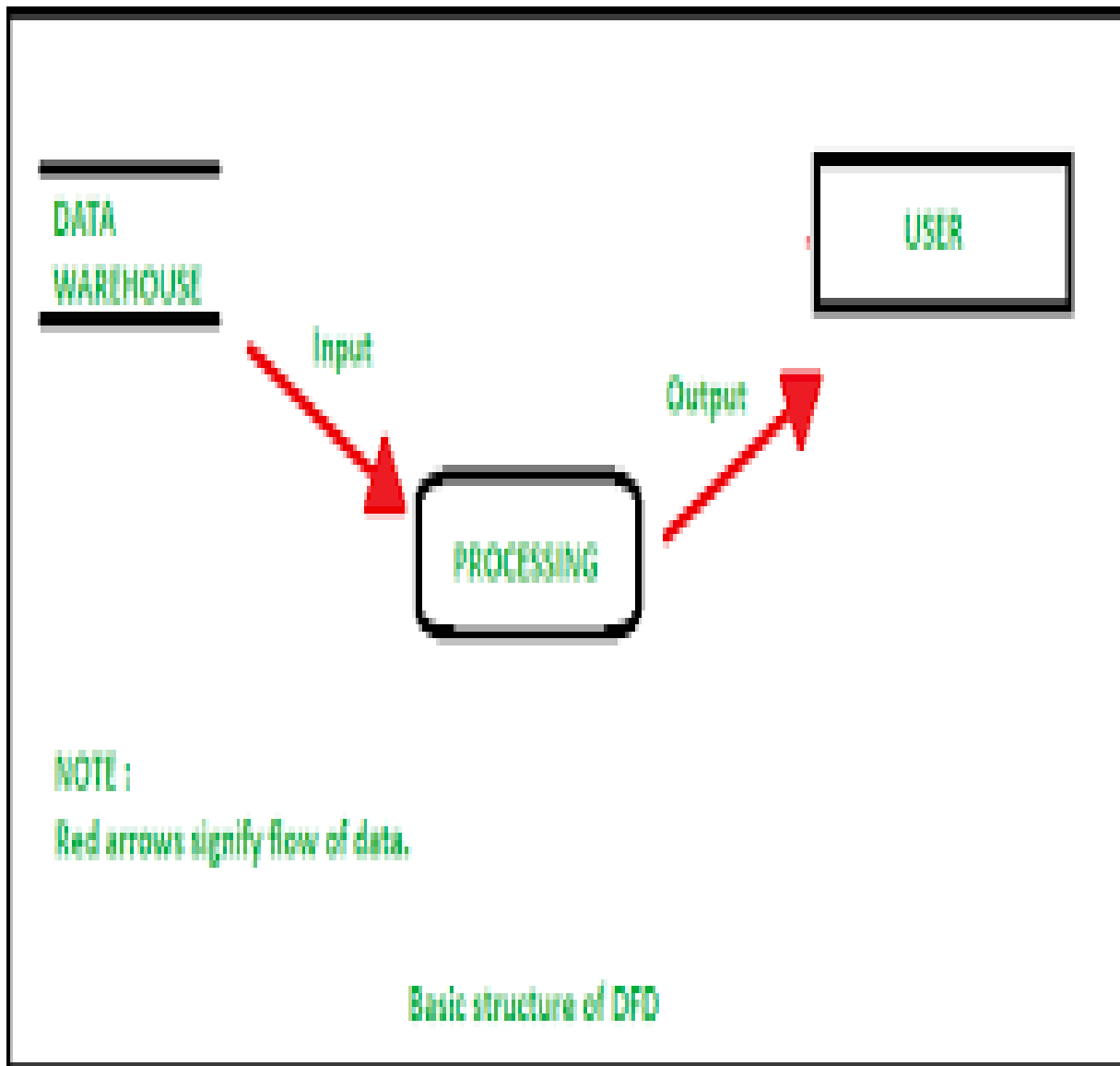
Data Flow Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyse an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That’s why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.



Rules for creating DFD

- ❖ The name of the entity should be easy and understandable without any extra assistance (like comments).
- ❖ The processes should be numbered or put in ordered list to be referred easily.
- ❖ The DFD should maintain consistency across all the DFD levels.
- ❖ A single DFD can have maximum processes up to 9 and minimum 3 processes.



Advantages of DFD

- ❖ It helps us to understand the functioning and the limits of a system.
- ❖ It is a graphical representation which is very easy to understand as it helps visualize contents.
- ❖ Data Flow Diagram represent detailed and well explained diagram of system components.
- ❖ It is used as the part of system documentation file.
- ❖ Data Flow Diagrams can be understood by both technical or nontechnical person because they are very easy to understand.

Disadvantages of DFD

- ❖ At times DFD can confuse the programmers regarding the system.
- ❖ Data Flow Diagram takes long time to be generated, and many times due to this reasons analysts are denied permission to work on it.

Activity Diagram:

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

Purpose of Activity Diagrams

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behaviour of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as –

- ❖ Draw the activity flow of a system.
- ❖ Describe the sequence from one activity to another.
- ❖ Describe the parallel, branched and concurrent flow of the system.

How to Draw an Activity Diagram?

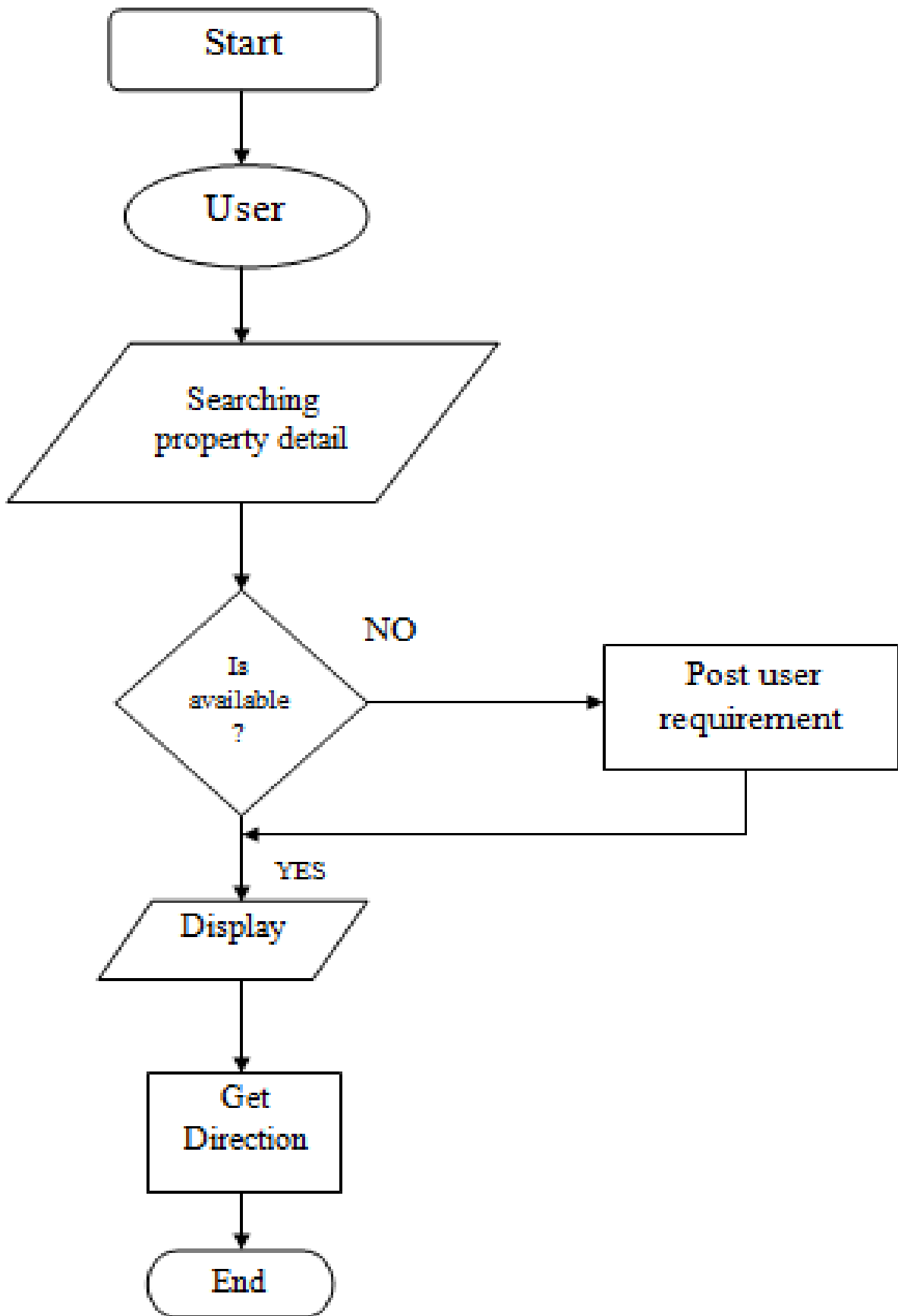
Activity diagrams are mainly used as a flowchart that consists of activities performed by the system. Activity diagrams are not exactly flowcharts as they have some additional capabilities. These additional capabilities include branching, parallel flow, swimlane, etc.

Before drawing an activity diagram, we must have a clear understanding about the elements used in activity diagram. The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions.

Before drawing an activity diagram, we should identify the following elements –

- ❖ Activities
- ❖ Association
- ❖ Conditions
- ❖ Constraints

Once the above-mentioned parameters are identified, we need to make a mental layout of the entire flow. This mental layout is then transformed into an activity diagram.



Where to Use Activity Diagrams?

The basic usage of activity diagram is similar to other four UML diagrams. The specific usage is to model the control flow from one activity to another. This control flow does not include messages.

Activity diagram is suitable for modelling the activity flow of the system. An application can have multiple systems. Activity diagram also captures these systems and describes the flow from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues, or any other system.

Activity diagram can be used for –

- ❖ Modelling work flow by using activities.
- ❖ Modelling business requirements.
- ❖ High level understanding of the system's functionalities.
- ❖ Investigating business requirements at a later stage.

Class Diagram:

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

Purpose of Class Diagrams

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as –

- ❖ Analysis and design of the static view of an application.
- ❖ Describe responsibilities of a system.
- ❖ Base for component and deployment diagrams.
- ❖ Forward and reverse engineering.

How to Draw a Class Diagram?

Class diagrams are the most popular UML diagrams used for construction of software applications. It is very important to learn the drawing procedure of class diagram.

Class diagrams have a lot of properties to consider while drawing but here the diagram will be considered from a top-level view.

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. A collection of class diagrams represents the whole system.

The following points should be remembered while drawing a class diagram –

- ❖ The name of the class diagram should be meaningful to describe the aspect of the system.
- ❖ Each element and their relationships should be identified in advance.
- ❖ Responsibility (attributes and methods) of each class should be clearly identified
- ❖ For each class, minimum number of properties should be specified, as unnecessary properties will make the diagram complicated.
- ❖ Use notes whenever required to describe some aspect of the diagram. At the end of the drawing, it should be understandable to the developer/coder.
- ❖ Finally, before making the final version, the diagram should be drawn on plain paper and reworked as many times as possible to make it correct.

Where to Use Class Diagrams?

Class diagram is a static diagram and it is used to model the static view of a system. The static view describes the vocabulary of the system.

Class diagram is also considered as the foundation for component and deployment diagrams. Class diagrams are not only used to visualize the static view of the system but they are also used to construct the executable code for forward and reverse engineering of any system.

Generally, UML diagrams are not directly mapped with any object-oriented programming languages but the class diagram is an exception.

Class diagram clearly shows the mapping with object-oriented languages such as Java, C++, etc. From practical experience, class diagram is generally used for construction purpose.

In a nutshell it can be said, class diagrams are used for –

- ❖ Describing the static view of the system.
- ❖ Showing the collaboration among the elements of the static view.
- ❖ Describing the functionalities performed by the system.
- ❖ Construction of software applications using object-oriented languages.

Use Case Diagram:

To model a system, the most important aspect is to capture the dynamic behaviour. Dynamic behaviour means the behaviour of the system when it is running/operating.

Only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML, there are five diagrams available to model the dynamic

nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

Hence to model the entire system, a number of use case diagrams are used.

Purpose of Use Case Diagrams

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and Statechart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analysed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modelled to present the outside view.

In brief, the purposes of use case diagrams can be said to be as follows –

- ❖ Used to gather the requirements of a system.
- ❖ Used to get an outside view of a system.

- ❖ Identify the external and internal factors influencing the system.
- ❖ Show the interaction among the requirements are actors.

How to Draw a Use Case Diagram?

Use case diagrams are considered for high level requirement analysis of a system. When the requirements of a system are analysed, the functionalities are captured in use cases.

We can say that use cases are nothing but the system functionalities written in an organized manner. The second thing which is relevant to use cases are the actors. Actors can be defined as something that interacts with the system.

Actors can be a human user, some internal applications, or may be some external applications. When we are planning to draw a use case diagram, we should have the following items identified.

Functionalities to be represented as use case

- ❖ Actors
- ❖ Relationships among the use cases and actors.

Use case diagrams are drawn to capture the functional requirements of a system.

Where to Use a Use Case Diagram?

There are five diagrams in UML to model the dynamic view of a system. Now each and every model has some specific purpose to use. Actually, these specific purposes are different angles of a running system.

To understand the dynamics of a system, we need to use different types of diagrams. Use case diagram is one of them and its specific purpose is to gather system requirements and actors.

Use case diagrams specify the events of a system and their flows. But use case diagram never describes how they are implemented. Use case diagram can be imagined as a black box where only the input, output, and the function of the black box is known.

These diagrams are used at a very high level of design. This high-level design is refined again and again to get a complete and practical picture of the system. A well-structured use case also describes the pre-condition, post condition, and exceptions. These extra elements are used to make test cases when performing the testing.

Although use case is not a good candidate for forward and reverse engineering, still they are used in a slightly different way to make forward and reverse engineering. The same is true for reverse engineering. Use case diagram is used differently to make it suitable for reverse engineering.

In forward engineering, use case diagrams are used to make test cases and in reverse engineering use cases are used to prepare the requirement details from the existing application.

Use case diagrams can be used for –

- ❖ Requirement analysis and high-level design.
- ❖ Model the context of a system.
- ❖ Reverse engineering.
- ❖ Forward engineering.

Result, Conclusion & Future Scope

- ❖ The purpose of making this project is to provide everyone an easy way to the problem of data search. So, the result of this project is a Olympic data Analysis. which can search Olympic related data into our desirable data.

- ❖ And finally, it came up with a friendly Olympic data Analysis.
- ❖ . The Olympic data Analysis. is friendly because we can search with easier as compare to other's future, it can search the all type of data and many more option will be there.

List of Table

S.No.	Caption	Page No.
1	Student's Data	4
2	Faculty's Data	4

List of Figures

S.No.	Title	Page No.
1	Implementing of data	21 &23
2	Merging two dataframe	23
3	Creating new intent	24
4	Data Analysis of Gold medal	25
5	Responses creation	26
6	Search data training	27
7	Search data customisation	28
8,9	Player's weight Analysis	29
10	Flask API	30
11	Fulfilment section	31
12	Webhook in fulfilment	32
13	Flask app & search factor	33
14	API output	34
15	Integrations part	30
16,17	Web demo	35
18	Testing of Olympic data analysis	36
19	Browser uploading	37
20	Confirmation data	38
21,22	Browser token & search data	39

REFERENCES

1. “Recruitment Olympic player’s performance Analysis”, International Research Journal of Engineering and Technology (IRJET), vol. 5, Issue: 08, Aug 2018[1].
Authors: Akash Balachandar, Anusha D Kulkarni
2. Implementation of Olympic data analysis in Online Commerce, and Open Innovation.
Authors: María D. Illescas-Manzano, Noé Vicente López, Nuno Afonso González and Carmen Cristofol Rodríguez.
3. Literature survey on various Olympic data analysis.
Authors: Harsha Pariyani, Anshika Sinha, Preeti Bhat, Roshni Rote, Asst. Prof. N. A. Mulla.
4. “Intelligent Olympic data analysis for easy web-analytics”.
Author: Ravi R.
5. “Task-based Interaction Olympic data analysis”, EEE521 final year project Report school of computing, Engineering & Intelligent System[4]. Authors: Dr. Kevin Curran, Dr. Daniel Kelly
6. “Classification Technique of Interviewer-Bot Result using Naïve Bayes an Phrase Reinforcement Algorithms,” International Journal of Emerging Technologies in Learning (iJET), 13(02), 33-47, 2018[2]. Authors: Sarosa, M., Junus, M., Hoesny, M. U., Sari, Z., & Fatnuriyah, M.
7. Deshpande, A., Shahane, A., Gadre, D., Deshpande, M., & Joshi, P. M, 2017. A Survey of various Olympic data analysis implementation techniques. International Journal of Computer Engineering and Applications, Volume XI, Special Issue, ISSN 2321-3469.

8. [.https://chatbotsmagazine.com/what-is-the-working-of-a-chatbot-e99e6996f51c](https://chatbotsmagazine.com/what-is-the-working-of-a-chatbot-e99e6996f51c)
9. [https://en.wikipedia.org/wiki/ Olympic data analysis](https://en.wikipedia.org/wiki/Olympic_data_analysis)