A Project report on

**CRIC ALERTS**

A Dissertation submitted in partial fulfillment of the academic

requirements for the award of the degree.

**Bachelor of Technology**

**In**

**Computer Science and Engineering**

Submitted by

A.SAI PRASANTH 20H51A05A9

B.TIRYAGISH PRAYAD 20H51A05B8

G.HARDHIK 20H51A05C1

Under the esteemed guidance of

Dr.S.Kirubakaran

Associate Professor



**Department of Computer Science and Engineering**

**CMR COLLEGE OF ENGINEERING & TECHNOLOGY**

(An Autonomous Institution, Approved by AICTE, Affiliated to JNTUH, NAAC ’A+’)

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2020- 2024

### **CMR COLLEGE OF ENGINEERING & TECHNOLOGY**

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



#### **CERTIFICATE**

This is to certify that the Project report on “Cric Alerts” being submitted by A.Sai Prashanth (20H51A05A9), B.TriyagishPrayad (20H51A05B8), G.Hardhik (20H51A05C1) in partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering is a record of bonafide work carried out his/her under my guidance and supervision.

e

The results embodies in this project report have not been submit any other University or Institute for the award of any Degree.

Dr.S.Kirubakaran Dr.S.Siva Skandha

Associate Professor Associate Professor and HOD

Dept. of CSE Dept. of CSE

##### Acknowledgement

With great pleasure we want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **Dr.S.Kirubakaran,** Dept of Computer Science and Engineering for his valuable suggestions and guidance during the execution of this project work.

We would like to thank **Dr. S.Siva Skandha,** Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete my project work successfully.

We are very grateful to **Dr. Vijaya Kumar Koppula**, Dean-Academic, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Dr.VA Narayana**, Principal CMRCET for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the Teaching & Non- teaching staff of Department of Computer Science and Engineering for their co-operation

Finally we express my sincere thanks to **Mr. Ch. Gopal Reddy,Secretary**, CMR Group of Institutions, for his continuous care. I sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

###### A.Sai Prashanth (20H51A05A9)

B.Triyagish Prayad (20H51A05B8)

G.Hardhik (20H51A05C1)

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **CHAPTERS** | | **DESCRIPTION** | **PAGE No** |
|  |  | **ABSTRACT** | 7 |
| **1** |  | **INTRODUCTION** | 8 |
|  | 1.1 | Objective | 8 |
| **2** |  | **BACKGROUND WORK** | 9-12 |
|  | 2.1 | Existing Systems | 10-12 |
| **3** |  | **Proposed System** | 13-16 |
|  | 3.1 | Proposed solution | 13 |
|  | 3.2 | Source Code | 14-15 |
|  | 3.3 | System Requirements | 16 |
| **4** |  | **DESIGNING** | 17-18 |
|  | 4.1 | Proposed System Architecture | 17 |
|  | 4.2 | Diagrams of components | 18 |
| **5** |  | **Results and Discussions** | 19-21 |
|  | 5.1 | Results | 19-20 |
|  | 5.2 | Discussions | 21 |
| **6** |  | **Conclusion And Future Work** | 22 |
|  | 6.1 | **Conclusion** | 22 |
|  | 6.2 | **Future work** | 22 |
| **7** |  | **References** | 23 |
|  | 7.1 | References | 23 |

**List Of Figures**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PAGE NO.** |
| **2.1.1-2.1.4** | Existing Systems | **10-12** |
| **4.1.1** | Proposed System Architecture | **17** |
| **4.2.1** | Diagram of Components | **18** |
| **5.1.1-5.1.4** | Results | **19-20** |
| **5.2.1** | Discussions | **21** |

**ABSTRACT**

As we all know we watch cricket we admire we get thrilled and sometimes we also want to share the cricket match scores. But for that we have to type the score manually or we have to say our friends please watch match is wonderful their batting or bowling excellently. So why can’t a platform be there where everyone knows the score and we can connect and share our reactions to that.

So by taking this as a problem statement for our mini Projects we have come with a solution.

For this we have to select a platform so we have selected telegram as a as our platform this is very simple and reliable application. This Product Architecture consists of 3 stages as follows:1. Fetch live score from espncricinfo.2. Data Parser to extract critical information.3. Notification mechanism to send data to telegram

This would be helpful for the people who cannot watch the match online and it consumes less amount of data rather than streaming online.

CHAPTER 1

# INTRODUCTION

We all like the cricket game, especially India vs Pakistan and India vs Australia matches are very interesting and we enjoy the match .As many people want to watch the match live in the stadium, but only some people can get the chance of watching the live, due to some reasons like not having enough seats in the stadium, money problem etc. So people will search in the websites to watch the live score, but sometimes it takes some time for the process and etc. We took this problem as opportunity and we want to solve this problem by creating the telegram bot. This bot can show the live score of live matches and by and this telegram bot can show the every detail of the match live. But the thing with these websites is that you have to keep staring at the page or refreshing it to get the latest score which can be quite unproductive if you are in the middle of important work. Moreover, every time you want to share the joy of a SIX or a wicket with your friend, you have to manually share it via a messaging app.

CHAPTER 2

BACKGROUND WORK

Web scraping is an automatic method to obtain large amounts of data from websites. Most of this data is unstructured data in an HTML format which is then converted into structured data in a spreadsheet or a database so that it can be used in various applications. There are many different ways to perform web scraping to obtain data from websites. These include using online services, particular API’s or even creating code for web scraping from scratch. Many large websites, like Google, Twitter, Facebook, StackOverflow, etc. have API’s that allow you to access their data in a structured format. This is the best option, but there are other sites that don’t allow users to access large amounts of data in a structured form or they are simply not that technologically advanced. In that situation, it’s best to use Web Scraping to scrape the website for data.

This cricket bot which is deployed in Telegram application will allow us to get the details of the live scores of the live matches going across the world. Once we started chatting with the Cricket bot by sending /start or press the START button it will start fetching the ongoing live matches and it will send us the details. We can select the required match. Then bot will look into the API and it will throw us the details of the match which we have selected.

2.1 Existing solutions:

CRICBUZZ:-

Cricbuzz is an Indian cricket news website owned by Times Internet. It features, news, articles and live coverage of cricket matches including videos, text commentary, player stats and team rankings. Their website also offers a mobile application and also telegram

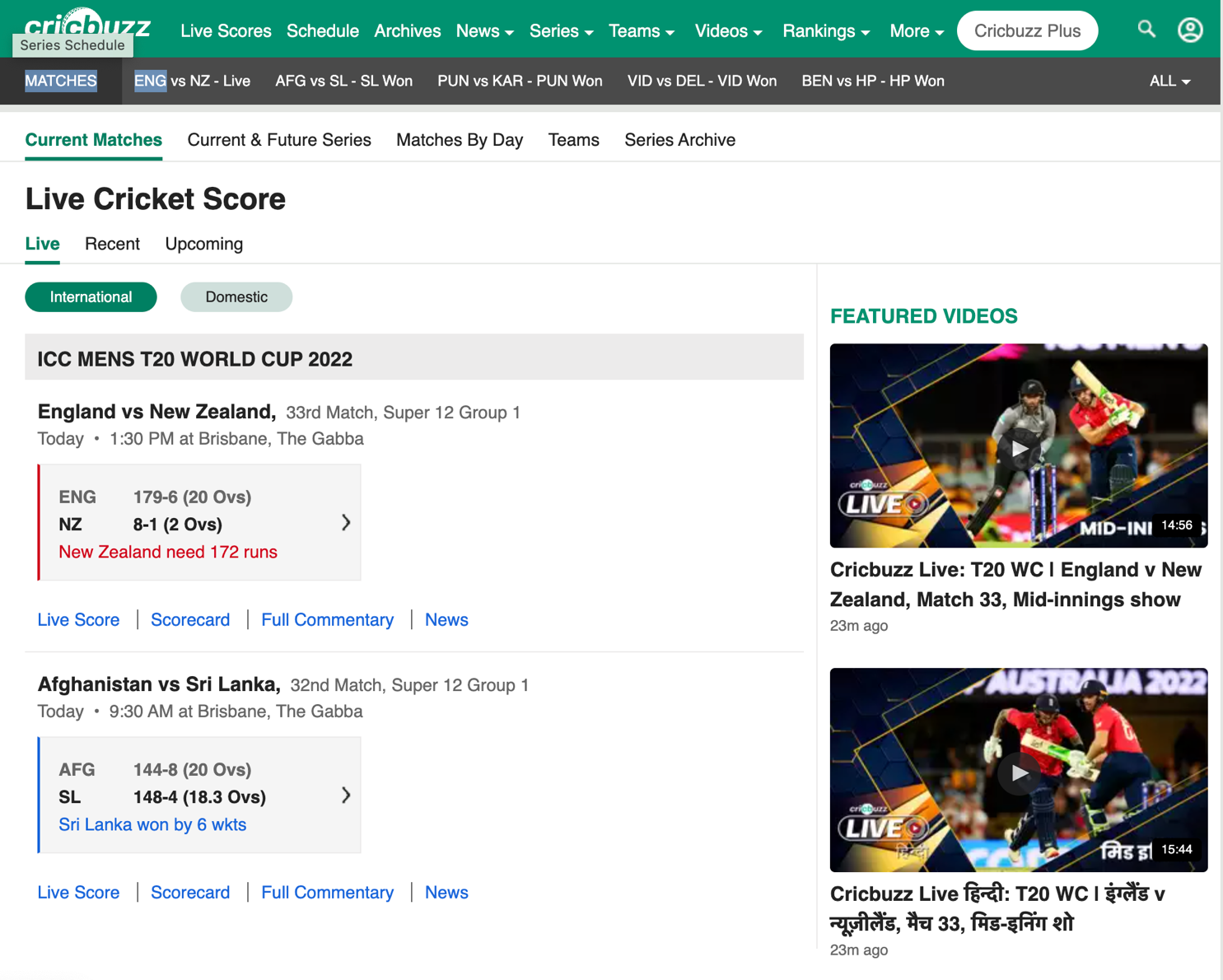


Fig no-(2.1.1)

This figure represents the cricbuzz website where scores are displayed

CRICKET SOCIAL MEDIA BOTS

These Bots scrape the data by using api services and send the data to the user. The services are offered by many cricket websites and individual’s who contribute to opensource in creating bots .

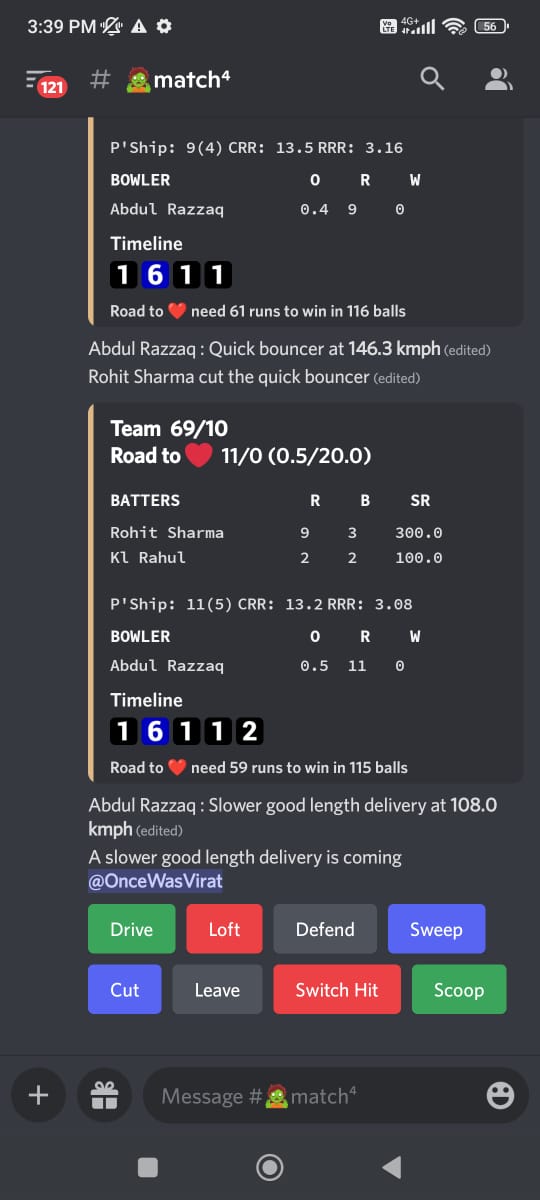
 

Fig no-(2.1.2) Fig no-(2.1.3)

These figures represent a discord cricket bot.

Fig no-(2.1.3) Fig no-(2.1.4)

These figures represent a twitter cricket bot.

CHAPTER 3

PROPOSED SYSTEM

**3.1Proposed Solution :**

This cricket bot which is deployed in Telegram application will allow us to get the details of the live scores of the live matches going across the world. Once we started chatting with the Cricket bot by sending /score or press the score button it will start fetching the ongoing live matches and it will send us the details.Then bot will look into the API and it will throw us the details of the matches.

**3.2 Source Code:**

import logging

import signal

import requests

import sys

import time

import json

from aiogram import Bot, Dispatcher, executor, types

API\_TOKEN = '5605931750:AAH5I2yMkqiKIXLobMjXPD6KCRLoP09lExQ'

#API\_TOKEN = '5164815705:AAHZuIrXz2CA-7dRPVfyBmX4HDK6NjjwPYA'

# Configure logging

logging.basicConfig(level=logging.INFO)

# Initialize bot and dispatcher

bot = Bot(token=API\_TOKEN)

dp = Dispatcher(bot)

def signal\_handler(signal, frame):

global interrupted

interrupted = True

signal.signal(signal.SIGINT, signal\_handler)

def wait\_time(g\_time):

interrupted = False

for remaining in range(g\_time, 0, -1):

sys.stdout.write("\r")

sys.stdout.write("{:2d} seconds remaining.".format(remaining))

sys.stdout.flush()

time.sleep(1)

if interrupted:

print("Gotta go")

break

scorelist = [0]\*100

prevdata = " "

@dp.message\_handler(commands=['score','scor'])

async def echo(message: types.Message):

while True:

count = 0

response = requests.get('https://hs-consumer-api.espncricinfo.com/v1/pages/matches/live?')

time.sleep(1)

data = json.loads(response.text)

if prevdata != data:

for liveInning in data['content']['matches']:

if liveInning['liveInning'] != None:

msg = ""

txt = liveInning['slug']

x = txt.upper()

msg = msg + " " + x + " " + "\n"

msg = msg + liveInning['state'] + "\n"

msg = msg + liveInning['statusText'] + "\n"

flag = False

for isLive in liveInning['teams']:

if isLive['isLive'] == False:

if isLive['score'] == None:

msg = msg + "Bowling..." + "\n"

else:

msg = msg + isLive['team']['name'] + " scored: " + isLive['score'] + "\n"

if isLive['isLive'] == True:

if scorelist[count] == 0:

scorelist.insert(count, isLive['score'])

flag = True

elif scorelist[count] != isLive['score']:

flag = True

scorelist[count] = isLive['score']

else:

flag = False

msg = msg + isLive['team']['name'] + " current score: " + isLive['score'] + "\n"

msg = msg + "Score Info: " + isLive['scoreInfo'] + "\n"

if flag == True:

await message.reply(msg)

await bot.send\_message('1778193674',msg)

count = count + 1

if \_name\_ == '\_main\_':

executor.start\_polling(dp,skip\_updates=True)

**3.3** **SYSTEM REQUIREMENTS**

**HARDWARE REQUIREMENTS:**

PROCESSOR: DUAL CORE 2 DUOS and above

RAM: 2GB DD HARD DISK: 250 GB

**SOFTWARE REQUIREMENTS:**

OPERATING SYSTEM: WINDOWS 8/8.1/10/11, MACOS, LINUX.

PLATFORM: PYCHARM

PROGRAMMING LANGUAGE: PYTHON

CHAPTER 4

DESIGNING

4.1Proposed System Architecture:

The proposed solution uses a systematic approach to display scores through telegram

with new functionalities and commands.

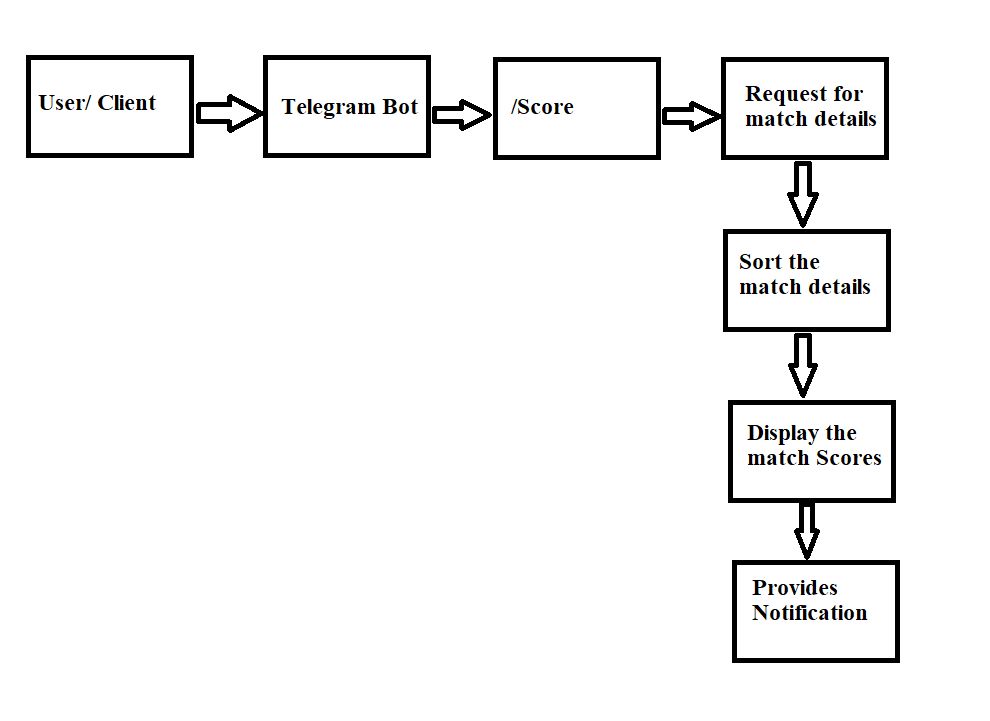


Fig no-(4.1.1)

This figure represents the architecture of the system.

4.2 Diagram of Components

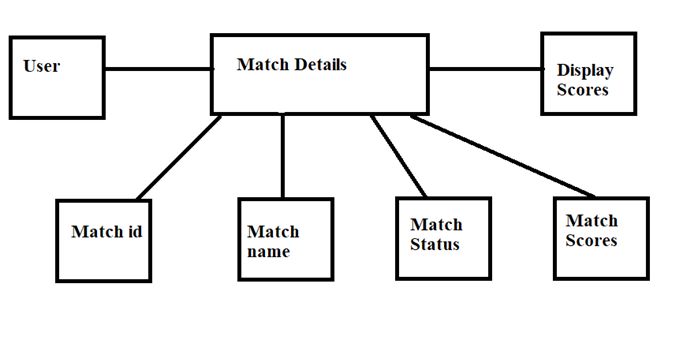


Fig no-(4.2.1)

This figure represents the components like match details etc.

CHAPTER 5

RESULTS AND

DISCUSSION

5.1 Results

Output :-when we execute the program this the output on the terminal/Console.

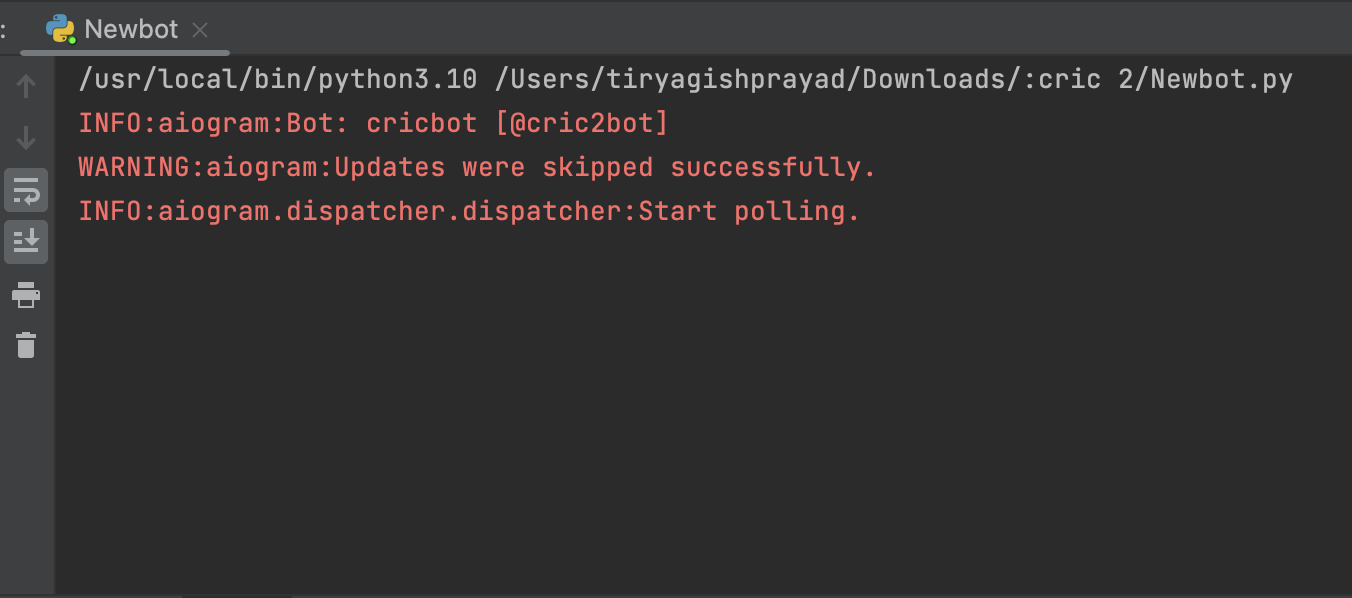


Fig no-(5.1.1)

This figure represents output console which is shown when we execute the code.

When we click on telegram app and click cricbot bot there is a feature where the bot can be added in groups and friends can discuss about the scores.

To start displaying scores there is a command :-“/score” which produces scores of all matches which are live.



Fig no-(5.1.2)

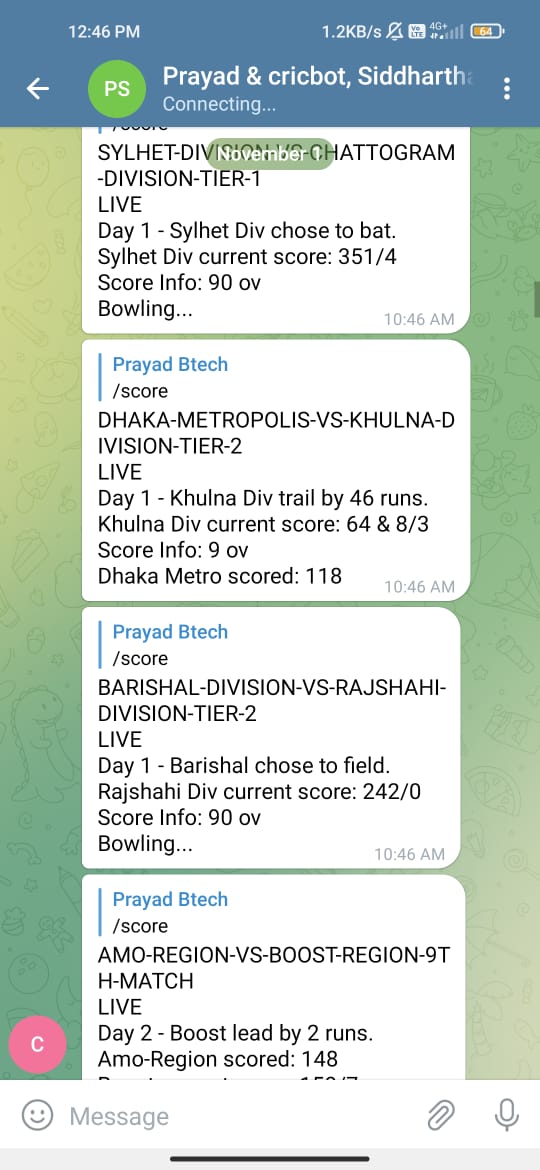


Fig no-(5.1.3)

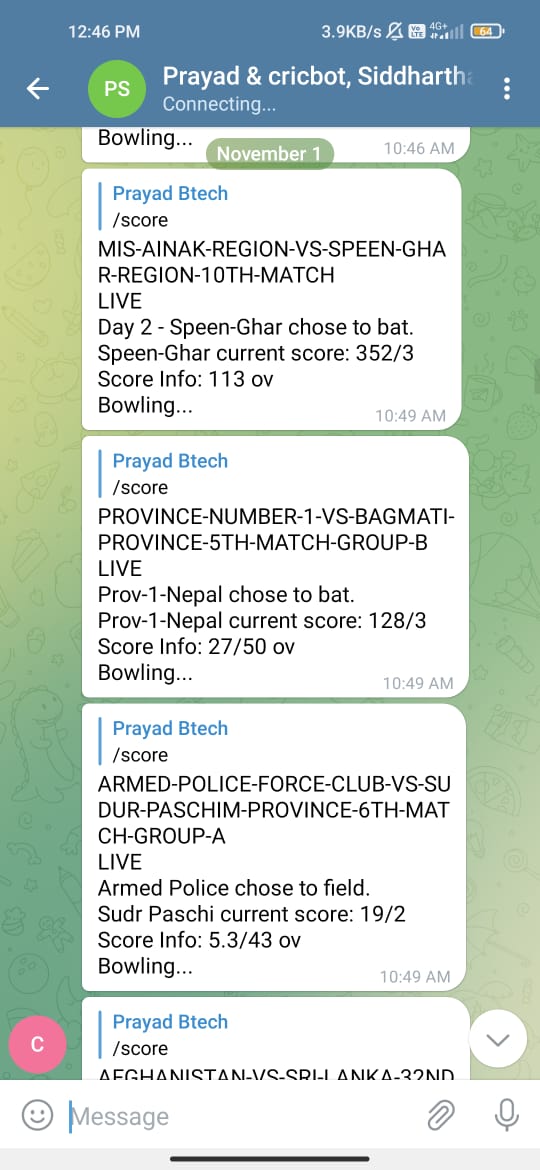


Fig no-(5.1.4)

These figures represent the score of the matches displayed when /start or /score command is given

5.2 Discussion

This cricket bot is made up of several components. One of the major component is Aiogram is a pretty simple and fully asynchronous framework for Telegram Bots  written in Python 3.7 with asyncio and aiohttp. It helps you to make your bots faster and simpler.

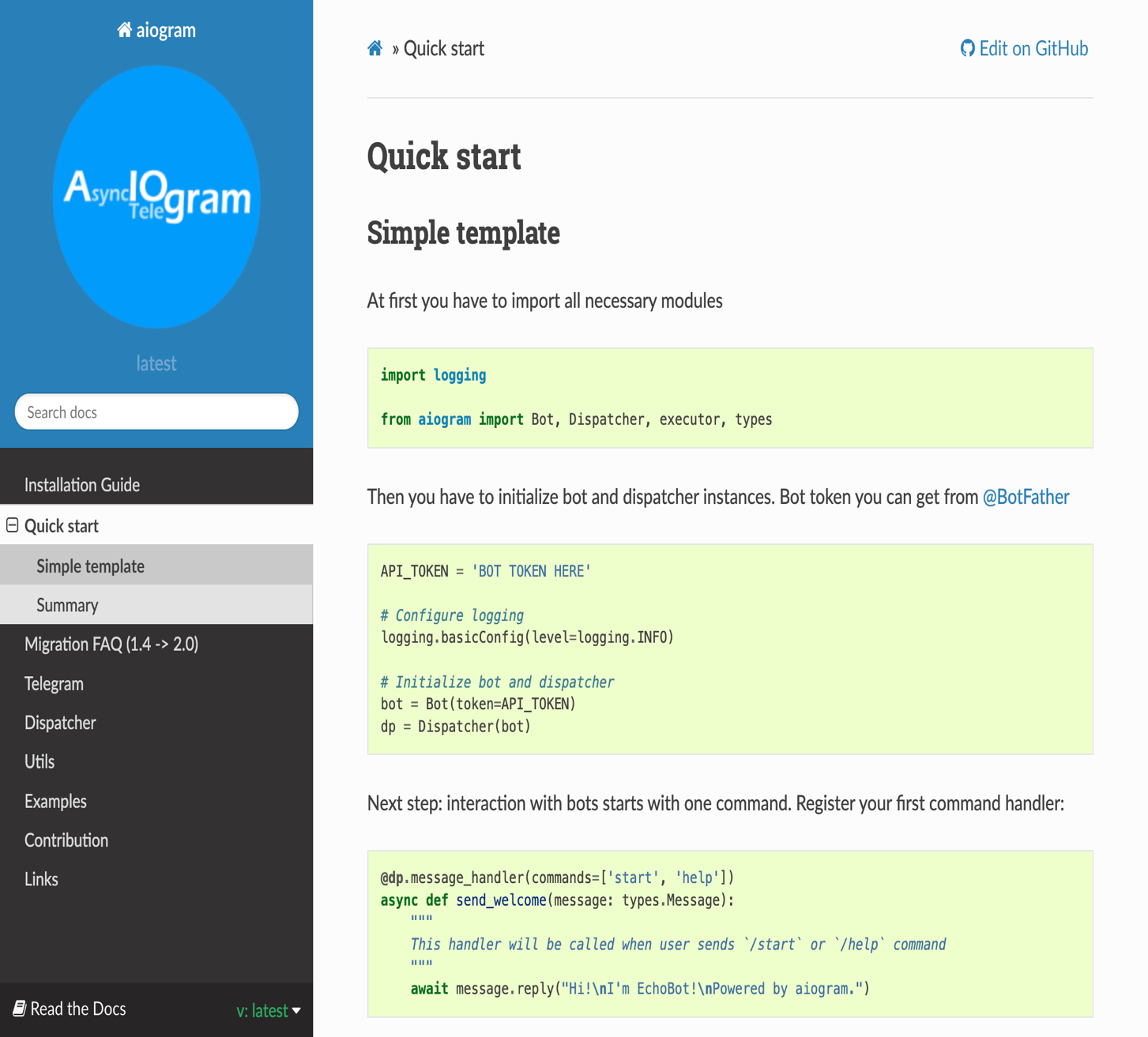


Fig no-(5.2.1)

This figure represents the template of aiogram doc’s

CHAPTER 6

CONCLUSION AND

FUTURE WORK

6.1 Conclusion :

The bot which we have developed is capable of sending the cricket updates in the telegram. The live cricket updates will be automatically. It extracts the live updates from the espncric info web and organises the json data and sends the data through the telegram chat. Such that people can chat and get updates of the cricket at the same time.This would be helpful for the people who cannot watch the match online and it consumes less amount of data rather than streaming online.

6.2 Future Work :

At present our bot sends the score which is scrapped through api and sends the data to the

user. We can improve the accuracy by Fast API’s which receive the data and send to the

user. We can also update by adding all sports and their live match scores. So that user will be

able to read all scores whatever they liked.

REFERENCES

[1] <https://medium.com/@ajayececit/telegram-bot-for-cricket-scores-5e0cf2338f07>

[2] <https://influencermarketinghub.com/top-telegram-bots/>

[3] <https://core.telegram.org/bots/api>

[4] [https://docs.aiogram.dev/en/latest/install.html#](https://docs.aiogram.dev/en/latest/install.html)

[5] <https://docs.aiogram.dev/en/latest/quick_start.html>