

# **CRICKET ALERTS**

Project submitted to the  
SRM University – AP, Andhra Pradesh  
for the partial fulfillment of the requirements to award the degree of

**Bachelor of Technology/Master of Technology**

In

**Computer Science and Engineering  
School of Engineering and Sciences**

Submitted by

**GROUP-OG13**

**VRIJESHWAR SINGH**

**(AP21110010922)**

**P.DHEERAJ**

**(AP21110010935)**

**K.AKHIL**

**(AP2111010931)**

**SAI AKASH**

**(AP21110010945)**



Under the Guidance of

**Mohammad Miskeen Ali**

**SRM University–AP  
Neerukonda, Mangalagiri, Guntur  
Andhra Pradesh – 522 240  
December, 2022**

# Certificate

Date: 7/16/2024

This is to certify that the work present in this Project entitled “**CRICKET ALERTS**” has been carried out by **VRIJESHWAR SINGH, P.DHEERAJ,SAI AKASH,K.AKHIL** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in the **School of Engineering and Sciences**.

## Supervisor

(Signature)

Prof. Mohammad Miskeen Ali

Designation,

Affiliation.

## Acknowledgments

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crown efforts with success.

We would like to express our special thanks of gratitude to our teacher Mohammad Miskeen Ali, who gave us the golden opportunity to do this wonderful project on the topic which also helped us in doing a lot of research and we came to know about so many new things. We are really thankful to them.

Secondly, we would also like to thank our friends who helped our team a lot in finishing this project within a limited time.

We are making this project not only for marks but also to increase our knowledge.

THANKS AGAIN TO ALL WHO HELPED US.

# Table of Contents

|                           |     |
|---------------------------|-----|
| Certificate               | i   |
| Acknowledgments           | iii |
| Table of Contents         | v   |
| Abstract                  | vii |
| 1. Introduction           | 1   |
| 1.1                       |     |
| Code.....                 | 2   |
| 1.2 Output                | 1   |
| 2. Methodology            | 3   |
| 2.1 Modules               | 3   |
| 2.1.1 SYS                 |     |
| Modules.....              |     |
| 2.1.2                     |     |
| Request.....              | 4   |
| 2.1.3 Time Modules.....   | 6   |
| 2.1.4                     |     |
| Warnings.....             | 8   |
| 2.1.5 Beautiful           |     |
| Soup.....                 | 10  |
| 2.1.6 Bootstrap.....      | 8   |
| 2.1.7 Class created.....  |     |
| 2.1.8 Functions used..... |     |
| 4. Concluding Remarks     | 7   |
| 5. Future Work            | 9   |
| References                | 11  |

## **Abstract**

The project entitled "Cricket Alerts" which is utilized by the user with an update of the cricket even when the user is not watching the match. Each and every match details such as the description about the team and team members will be stored in the repository system in the form of database. That database could be utilized by the mislaid by the users. Each and every match can be updated lively using this software. As soon as someone checks the scoreboard, details of a particular player can be viewed by a single click on his name any news other than cricket will also be updated. This software is error free and anyone can use this software. You can download this software by clicking on download below. So, use this software and get more benefit from this. This project aims to make a cricket score board that will update the scores of a match along with commentary as it happens. Cricket being a special part of the lives of many people, there will be many takers for such a system and the ability to follow the match without seeing the video will make it interesting for many. A user who cannot watch the event like someone busy with their work can easily check the commentary regularly to get updates on what is happening. The system will keep posting updated scores and the team line-up during the match. The admin will store upcoming match details and ensure that the team information is posted on the upcoming events; this will help the admin easily load information at the time of the match. This project will help the people who need to improve their performance at the event which will help to progress the betterment.

# 1. Introduction

Cricket fans, raise your foam fingers if you have been looking for a quicker way to share scores and updates with your friends. Even if you aren't a sports enthusiast, think about the number of fans who are out there looking for an option like this on their phones. Sports websites do offer live scores on their page which is what the majority of the people follow to stay updated on live updates if they are not watching a match.

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.

But now, you can build a better solution for the millions of sports fans (including yourself if you raised your foam finger) and even learn important concepts in that process.

## 1.1 CODE

```

import sys
import warnings
import requests
from time import sleep
from bs4 import BeautifulSoup
warnings.filterwarnings('ignore')
from tkinter import Tk
from tkinter.messagebox import Message
from _tkinter import TclError

class Cricketinfo:

    def __init__(self):#constructor
        self.url = "http://static.cricinfo.com/rss/livescores.xml"

    def display_score(self,score):
        TIME_TO_WAIT = 20000 # in milliseconds
        root = Tk()
        root.withdraw()
        try:
            root.after(TIME_TO_WAIT, root.destroy)
            Message(title="my score board", message=score, master=root).show()
        except TclError:
            pass

    def live_matches_details(self):
        sno = 1
        match_guid = []#empty list created store the guids of the matches
        #response object is created to store the scraped data from that url
        #send the get request

```



```

#get function is used to scrap the data from the url
response = requests.get(self.url)
#soup object is created
#usually, the scraped data is in the html format
#To convert to the structured xml format,BeautifulSoup function is used
soup = BeautifulSoup(response.text,'lxml')
print('\n\n=====')
print('Live Cricket Matches:')
print('=====\\n')
#findAll returns the list containing all matches
#findall returns the list of all the items in the xml
#to iterate the every item in list, we used the forloop
for item in soup.findAll('item'):
    print(str(sno) + ' ' + item.find('description').text)
    #find fucntion returns the description of every match
    #list of the current matches will get displayed by using the print statement
    match_guid.append(item.find('guid').text)
    #add the guids of the matches to the empty list which has been created
earlier
    sno = sno + 1#sno incremented by 1
return match_guid#this function returns the match_guid list

def validate_user_input(self):
    print('Enter match number or enter 0 to exit:')
    while True:
        try:
            userInput = int(input())
        except NameError:#to handle the nameerror exception
            print('Invalid input. Try Again!')
            continue
        except SyntaxError:#to handle syntax error exception

```

```

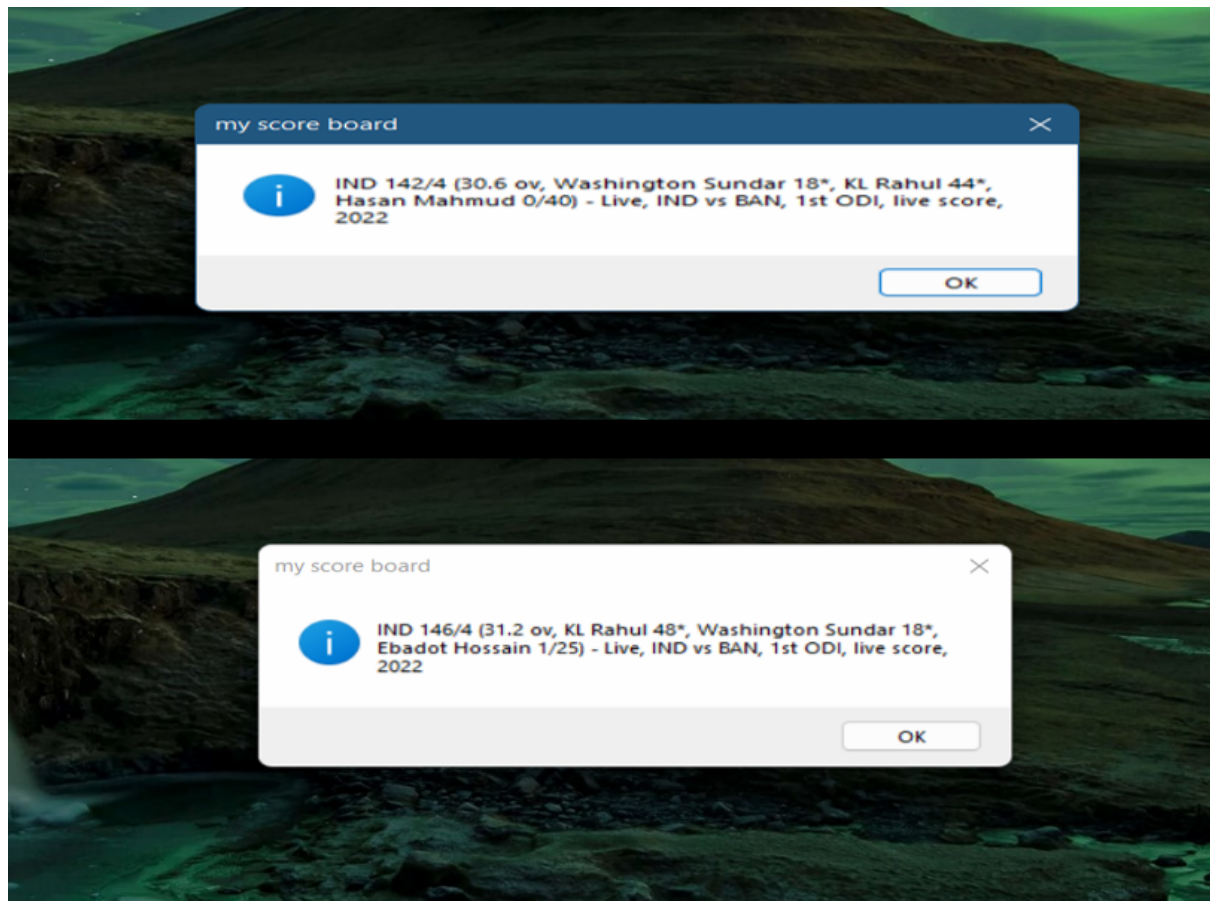
        print('Invalid input. Try Again!')
    if userInput < 0 or userInput > 30:#sno starts from 1
        print('Invalid input. Try Again!')
        continue
    elif userInput == 0:#to exit
        sys.exit()
    return userInput

def get_live_matches(self):
    match_guid= self.live_matches_details()
    userInput = self.validate_user_input()
    try:
        while True:
            matchUrl = match_guid[userInput - 1]#since,list index values start from
the 0,selected match's guid is initilized to variable matchUrl
            #r object is created to scrap the data of selected match
            r = requests.get(matchUrl)
            soup = BeautifulSoup(r.text,'lxml')#selected match's content is
formatted to xml
            #in that soup we need to find 'title' in which the score of the match is
getting stored and updated
            score = soup.findAll('title')
            try:
                r.raise_for_status()
            except Exception as exc:
                print ('Connection Failure. Try again!')
                continue
            self.display_score(score[0].text)
    except KeyboardInterrupt:
        print('interrupted!')

```

```
m=Cricketinfo()  
m.get_live_matches()
```

## **1.2 OUTPUT**



ASH/OneDrive/Desktop/python/python project code.py"



```
=====
Live Cricket Matches:
=====
```

1. Australia 598/4 & 182/2 v West Indies 333/10 & 283/10 \*
  2. Victoria 283/9 & 209/10 v New South Wales 163/10 & 260/10 \*
  3. Queensland 295/6 & 97/7 \* v Western Australia 427/7
  4. Pakistan 579/10 v England 657/10 & 95/2 \*
  5. Mashonaland Eagles 581/10 v Mid West Rhinos 455/10 & 48/2 \*
  6. Matabeleland Tuskers 212/10 & 366/10 v Southern Rocks 337/10 & 125/7 \*
  7. New Zealand Women 148/4 v Bangladesh Women 111/8 \*
  8. Bauhinia Stars Women 66/3 \* v Jade Jets Women 65/10
  9. Bangladesh v India 142/4 \*
  10. India Women Under-19s v New Zealand Women Under-19s
  11. Nepal 7/1 \* v Scotland
  12. Samp Army v Team Abu Dhabi
  13. Deccan Gladiators v New York Strikers
- Enter match number or enter 0 to exit:

9



## 2. Methodology

The project entitled with "Cricket Alerts" is divided into numerous modules. The detail description about the whole modules will be explained in below.

### 2.1 MODULES

#### 2.1.1 SYS MODULES

The **sys module** in Python provides various functions and variables that are used to manipulate different parts of the Python runtime environment. It allows operating on the interpreter as it provides access to the variables and functions that interact strongly with the interpreter.

##### *1.1.1.1 Input and Output using sys*

The sys modules provide variables for better control over input or output. We can even redirect the input and output to other devices. This can be done using three variables –

- `stdin`
- `stdout`
- `stderr`

##### *1.1.1.2 stdin*

It can be used to get input from the command line directly. It is used for standard input. It internally calls the `input()` method. It, also, automatically adds '\n' after each sentence.

##### *1.1.1.3 stdout:*

A built-in file object that is analogous to the interpreter's standard output stream in Python. `stdout` is used to display output directly to the screen console. Output can be of any form, it can be output from a print statement, an expression statement, and even a prompt direct for input. By default, streams are in text mode. In fact, wherever a print function is called within the code, it is first written to `sys.stdout` and then finally on to the screen.

##### *1.1.1.4 stderr*

Whenever an exception occurs in Python it is written to `sys.stderr`.

Commanding line arguments are those which are passed during the calling of the program along with the calling statement. To achieve this using the

sys module, the sys module provides a variable called sys.argv. It's main purpose are:

- It is a list of command-line arguments.
- len(sys.argv) provides the number of command-line arguments.
- sys.argv[0] is the name of the current Python script.

**sys.exit([arg])** can be used to exit the program. The optional argument arg can be an integer giving the exit or another type of object. If it is an integer, **zero is considered “successful termination”**.

sys.path is a built-in variable within the sys module that returns the list of directories that the interpreter will search for the required module.

When a module is imported within a Python file, the interpreter first searches for the specified module among its built-in modules. If not found it looks through the list of directories defined by **sys.path**.

**sys.modules** return the name of the Python modules that the current shell has imported.

**sys.getrefcount()** method is used to get the reference count for any given object. This value is used by Python as when this value becomes 0, the memory for that particular value is deleted.

[sys.setrecursionlimit\(\)](#)

sys.setrecursionlimit() method is used to set the maximum depth of the Python interpreter stack to the required limit.

[sys.getrecursionlimit\(\)  
method](#)

sys.getrecursionlimit() method is used to find the current recursion limit of the interpreter or to find the maximum depth of the Python interpreter stack.

[sys.settrace\(\)](#)

It is used for implementing debuggers, profilers and coverage tools. This is thread-specific and must register the trace using threading.settrace(). On a higher level, sys.settrace() registers the traceback to the Python interpreter

[sys.setswitchinterval\(\)  
method](#)

sys.setswitchinterval() method is used to set the interpreter's thread switch interval (in seconds).

[sys.maxsize\(\)](#)

It fetches the largest value a variable of data type Py\_ssize\_t can store.

[sys.maxint](#)

maxint/INT\_MAX denotes the highest value that can be represented by an integer.

[sys.getdefaultencoding\(\)  
method](#)

sys.getdefaultencoding() method is used to get the current default string encoding used by the Unicode implementation.

## 2.1.2 REQUEST

Requests library is one of the integral part of Python for making HTTP requests to a specified URL. Whether it be REST APIs or Web Scrapping, requests is must to be learned for proceeding further with these technologies. When one makes a request to a URI, it returns a response. Python requests provides inbuilt functionalities for managing both the request and response.

### Installing Requests

Requests installation depends on type of operating system on eis using, the basic command anywhere would be to open a command terminal and run,

```
pip install requests
```

### *Making a Request*

Python requests module has several built-in methods to make Http requests to specified URI using GET, POST, PUT, PATCH or HEAD requests. A Http request is meant to either retrieve data from a specified URI or to push data to a server. It works as a request-response protocol between a client and a server.

|                             |                                                                                                                                                                                                                                                                    |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#"><u>GET</u></a>  | GET method is used to retrieve information from the given server using a given URI.                                                                                                                                                                                |
| <a href="#"><u>POST</u></a> | POST request method requests that a web server accepts the data enclosed in the body of the request message, most likely for storing it                                                                                                                            |
| <a href="#"><u>PUT</u></a>  | The PUT method requests that the enclosed entity be stored under the supplied URI. If the URI refers to an already existing resource, it is modified and if the URI does not point to an existing resource, then the server can create the resource with that URI. |



|                               |                                                                                                                                    |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#"><u>DELETE</u></a> | The DELETE method deletes the specified resource                                                                                   |
| <a href="#"><u>HEAD</u></a>   | The HEAD method asks for a response identical to that of a GET request, but without the response body.                             |
| <a href="#"><u>PATCH</u></a>  | It is used for modify capabilities. The PATCH request only needs to contain the changes to the resource, not the complete resource |

### 2.1.3 TIME MODULE :

In this article, we will discuss the time module and various functions provided by this module with the help of good examples.

As the name suggests Python time module allows to work with time in Python. It allows functionality like getting the current time, pausing the Program from executing, etc. So before starting with this module we need to import it.

#### 1.1.1.5 *Importing time module*

The time module comes with Python's standard utility module, so there is no need to install it externally. We can simply import it using the [import statement](#).

```
import time
```

The epoch is the point where the time starts and is platform-dependent. On Windows and most Unix systems, the epoch is January 1, 1970, 00:00:00 (UTC), and leap seconds are not counted towards the time in seconds since the epoch. To check what the epoch is on a given platform we can use [time.gmtime\(0\)](#).

#### *1.1.1.6 Python time.time()*

The `time()` function returns the number of seconds passed since epoch.

#### *1.1.1.7 Python time.ctime()*

The `time.ctime()` function takes seconds passed since epoch as an argument and returns a string representing local time.

#### *1.1.1.8 Python time.sleep()*

The `sleep()` function suspends (delays) execution of the current thread for the given number of seconds.

#### *1.1.1.9 time.struct\_time Class*

Several functions in the `time` module such as `gmtime()`, `asctime()` etc. either take `time.struct_time` object as an argument or return it.

## **2.1.4 WARNINGS**

Warnings are provided to warn the developer of situations that aren't necessarily exceptions. Usually, a warning occurs when there is some obsolete of certain programming elements, such as keyword, function or class, etc. A warning in a program is distinct from an error. Python program terminates immediately if an error occurs. Conversely, a warning is not critical. It shows some message, but the program runs. The `warn()` function defined in the 'warning' module is used to show warning messages. The warning module is actually a subclass of Exception which is a built-in class in Python.

- **Warning Class:** It is the super class of all warning category classes and a subclass of the Exception class.
- **UserWarning Class:** `warn()` function default category.
- **DeprecationWarning Class:** Base category for alerts regarding obsolete features when those warnings are for other developers (triggered by code in `__main__` unless ignored).
- **SyntaxWarning Class:** Base class for warnings of suspicious syntactic attributes.
- **RuntimeWarning Class:** Base class for warnings of suspicious run time attributes.
- **FutureWarning Class:** Base class for warnings on obsolete features when certain warnings are meant for end-users of Python-written programs.
- **PendingDeprecationWarning Class:** Base class for warnings of an outdated attribute.
- **ImportWarning Class:** Base class for warnings caused during a module importation process.
- **UnicodeWarning Class:** Base class for Unicode based warnings.
- **BytesWarning Class:** Base class for bytes and bytearray based warnings.
- **ResourceWarning Class:** Base class for resource-related warnings.

#### 1.1.1.10 *Warning Filters*

The warning filter in Python handles warnings (presented, disregarded or raised to exceptions). The warnings filter establishes an organized list of filter parameters, any particular warnings are matched on each filter requirement throughout the list till the match is made, the filter determines the match arrangement. Every entry is indeed a tuple (action, message, category, module, lineno) of the form in which:

- The **action** can be any of the following strings:

| String    | Explanation                                                                    |
|-----------|--------------------------------------------------------------------------------|
| “default” | Displays the first matching warnings for each position                         |
| “error”   | Converts warnings to raise exceptions                                          |
| “ignore”  | Never display warnings which match                                             |
| “always”  | Always display the warnings which match                                        |
| “module”  | Displays the first matching warnings per module                                |
| “once”    | Display just the first matching warnings, regardless of where they are located |

- The **message** is a string that has a regular expression that must match the beginning of the warning. (The expression compiled is always case-insensitive)
- The **category** is a class (warning subclass) of which the warning class must be a subclass for matching.
- The **module** is a string with a regular expression which must match the module name (The expression compiled is always case-insensitive).
- The **lineno** is an integer to match the number of the line in which the warning appeared, or 0 to match any number of the line.

### 2.1.5 BEAUTIFUL SOUP

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

#### 1.1.1.11 *Install BeautifulSoup4 using Source code*

One can install beautifulsoup, using source code directly, install beautifulsoup tarball from here – [download the Beautiful Soup 4 source tarball](#) after downloading cd into the directory and run,

Python setup.py install

#### *1.1.1.12 Verifying Installation*

To check whether the installation is complete or not, let's try implementing it using python

### **2.1.6 BOOTSTRAP4(BS4)**

**Bootstrap is a free front-end framework for faster and easier web development**

**Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins**

**Bootstrap also gives you the ability to easily create responsive designs**

### **2.1.7 CLASS CREATED**

Cricketinfo() - This class contains all required methods

### **2.1.8 FUNCTIONS USED**

disply\_score() - function to dispaly the gui of named 'my score board',which contains the score .

live\_matches\_details - function to scrap the details of the live matches from the website,and also it display the list of the current matches the current matches.

`validate_user_input()`- function to get the input from the user  
and validating the interfering erros with  
exception handlings.

`get_live_matches()` - function to evaluate and scrap the details  
of user selected match.

### **3. Concluding Remarks**

It is concluded that the application works well and satisfy the end users. The application is tested very well and errors are properly debugged. The application is simultaneously accessed from more than one system. Simultaneous login from more than one place is tested. This system is user friendly so everyone can use easily. Proper documentation is provided. The end user can easily understand how the whole system is implemented by going through the documentation. The system is tested, implemented and the performance is found to be satisfactory. All necessary output is generated. Thus, the project is completed successfully. Further enhancements can be made to the application, so that the application functions very attractive and useful manner than the present one. The speed of the transactions become more enough now.

## **4. Future Work**

There is scope for future development of this project. The world of computer fields is not static; it is always subject to be dynamic. The technology which is famous today becomes outdated the very next day. To keep the abstract of technical improvements, the system may be further refined. So, it is not concluded. Yet it will improve with further enhancements. Enhancements can be done efficiently. We can even update the same with further modification establishment and can be integrated with minimal modification. Thus the project is flexible and can be enhanced at any time with more advanced features.



## References

1. <https://www.w3schools.com/>
2. <https://www.w3schools.com/>
3. <http://omz-software.com/>
4. Lectures of Mohammad Miskeen Ali Sir