**THE FOOD APP**

**MINOR PROJECT REPORT**

*Submitted by*

**Abhay Goyal**

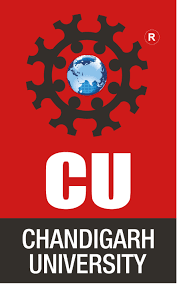
**20BCS6079**

**Abhigyan Mishra**

**20BCS6066**

BACHELOR OF ENGINEERING

CSE



APEX INSTITUTE OF TECHNOLOGY

CHANDIGARH UNIVERSITY, GHARUAN

MOHALI PUNJAB

**ACKNOWLEDGEMENT**

### The project work in this report is an outcome of continuous work a period and drew intellectual support from various sources. I would like to articulate my profound gratitude and indebtedness to those persons who helped me in completion of the project. I take this opportunity to express my sincere thanks and deep gratitude to all those people who extended their wholehearted co-operation and have helped me in completing this project successfully.

### I would also like to thank my parents and other fellow mates for guiding and encouraging me throughout the duration of the project.

**LIST OF CONTENTS**

### About The Project i

[Acknowledgement iii](#_TOC_250001)

[List Of Figures iv](#_TOC_250000)

### Introduction To Activites ………………………………………………. 3

### Project details…………………………………………………………………. 4

### Prerequisites……………………….. 4

### Steps to create the checker…………………………………………4

### Result and outcome of product……………………………7

### Why use connectivity test………………………………………8

### Conclusion and future scope…………………………………….9

# **1.Python Website Connectivity Checker Project**

### **What is a Site Connectivity Checker?**

A site connectivity checker is an application that helps us to monitor whether a website is available or not. There are many times when we open a url and the site isn’t available. To avoid such troubles we can use a website connectivity checker. The user just needs to input the url and it will be checked and availability will be visible.

### **2.Python Website Connectivity Checker Project Details**

The objective is to create a GUI based project which will have an input field where we have to enter the url. If the url is available “Available” will be displayed else “Not Available” will be displayed. To build this project we need a basic understanding of Python and Tkinter Module.

**3- Prerequisites:**

To build this project we need to install the following libraries:

1. **Tkinter Module** – This module is for creating an easy GUI in Python. To install Tkinter, use the following command:

**pip install tk**

1. **Urllib Module** – This module is used while working with URLs. To install urllib Module, use the following command:

**pip install urllib**

### **4.Steps to Create Python Website Connectivity Checker Project**

The following are the steps to create the site connectivity checker project-

1. Importing the required libraries.
2. Making the GUI.
3. Function to check the URL.

Let’s look at the steps in detail.

#### **1. Importing the Required Libraries:**

**a. Tkinter Module** – Tkinter Module is for creating an easy GUI in python. We will be using built methods in this library.

**b. Urllib Library** – Urllib library is used to do operations with urls in python. In this project, we will be using urllib.request to monitor what is the http code of the url.

#### **2. Creating GUI for Python Website Connectivity Checker Project:**

window=Tk()

* This command uses the Tk() method to create a GUI window named window.

window.geometry("700x350")

window.title("PythonGeeks")#give title to the window

head=Label(window, text="Website Connectivity Checker", font=('Calibri 15'))# a label

head.pack(pady=20)

* Now that we have created a window, we will be specifying some attributes to it. To give a specific size to the window, we use the geometry() method. We give a title to the window using the title() method.
* Now we want to add a heading to our window. For this we create a label named head using the Label() method. While using a Label() method, we can specify the text, font, background color, foreground color, etc. To display this Label on the window, we use the pack() method.

Entry(window, textvariable=url).place(x=200,y=80,height=30,width=280)# enter a website url

#create a button

Button(window, text="Check",command=check).place(x=320,y=160)

window.mainloop()#main command

* Now we create a text area using the Entry() method in the tkinter library. This will be the area on the window where the user can put in the urls that are to be tested. Textvariable is an attribute that specifies that whatever input is given will be saved in a variable named url. To display this input area, we use the place() method. Inside a place method, x and y coordinates need to be specified.
* We create a button using the Button() method. command=check ensures that when the button is clicked the check function is evoked. To place the button we use place() method and specify the x and y coordinates within.

#### **3. Creating Check Function to check URL:**

url=tk.StringVar()# url is of string type

* Whatever url is entered in the Entry field was saved in the variable url. Now we need to make sure that the url variable is String Type. StringVar() method helps you to manage strings and get its value when needed.

**def** check():

web= (url.get())

status\_code = urllib.request.urlopen(web).getcode()

website\_is\_up = status\_code == 200

**if** website\_is\_up==**TRUE**:

Label(window, text="Website Available", font=('Calibri 15')).place(x=260,y=200)

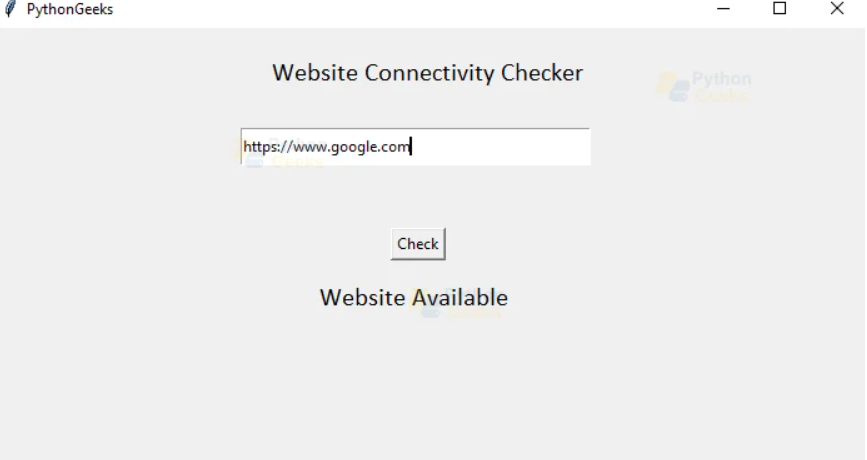
**else**:

Label(window, text="Website Not Available", font=('Calibri 15')).place(x=260,y=200)

* We create a check() function which will perform a url checking for us. Whenever the button will be clicked this function will be evoked.
* We create a variable named web. Using the get() method we extract the value of the url and save the value on the web.
* Urllib.request helps us to use the urlopen which will open that url for us. Once the url is open we check the status code for that url by using the getcode() method. We saved this status code of the web in a variable status\_code.
* As we want to check if the website is working or not, we want the source code to be 200. (Status Code 200 means OK. It specifies that a website is working). Now we check if status\_code is equal to 200 or not. A boolean value ( true or false ) is saved in a variable website\_is\_up.
* Now using the if-else loop, we check if the website\_is\_up is true then we make a label to display “Website Available”. If it is false then we display a label “Website Not Available”.

### **5.Python Site Connectivity Checker Output**

Here we have checked if a website is available or not:



## 6:Why use Connectivity Tests?

Connectivity Tests can help you troubleshoot the following network connectivity issues:

* Unintended inconsistent configurations
* Obsolete configurations caused by network configuration changes or migrations
* Configuration errors for a variety of network services and functions

When testing Google-managed services, Connectivity Tests can also help you determine whether there is an issue in your VPC network or in the Google-owned VPC network used for the service resources.

## How Connectivity Tests analyzes configurations

When analyzing network configurations, Connectivity Tests uses an [abstract state machine](https://wikipedia.org/wiki/Abstract_state_machine) to model how a VPC network should process packets. Google Cloud processes a packet in several logical steps.

Because of the variety of VPC network services and features that the configuration analysis supports, a test packet traversing a VPC network configuration can take many possible paths.

The following diagram shows a model for how the configuration analysis simulates trace traffic between two Compute Engine virtual machine (VM) instances—one on the left and another on the right.

Depending on your Google Cloud network and resource configurations, this traffic might go through a Cloud VPN tunnel, a Google Cloud load balancer, or a peered VPC network before reaching the destination VM instance.

**7:Conclusion and future scope**

Connectivity Tests is a diagnostics tool that lets you check connectivity between network endpoints. It analyzes your configuration and, in some cases, performs live data plane analysis between the endpoints. An endpoint is a source or destination of network traffic, such as a VM, Google Kubernetes Engine (GKE) cluster, load balancer forwarding rule, or an IP address on the internet.

To analyze network configurations, Connectivity Tests simulates the expected forwarding path of a packet through your Virtual Private Cloud (VPC) network, Cloud VPN tunnels, or VLAN attachments. Connectivity Tests can also simulate the expected inbound forwarding path to resources in your VPC network.

For some connectivity scenarios, Connectivity Tests also performs live data plane analysis. This feature sends packets over the data plane to validate connectivity and provides baseline diagnostics of latency and packet loss. If the route is supported for the feature, each test that you run includes a live data plane analysis result.