

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfilment of the requirements for the
award of Bachelor of Engineering Degree in Computer
Science and Engineering

By
VISHANTHAN SHREE R
REG. NO. 39111107



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119,
TAMILNADU
APRIL 2022



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI- 600119
www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **VISHANTHAN SHREE R** (Reg. No: **39111107**) who carried out the project entitled "**WHATSAPP AUTOMATION USING ANDROID DEBUGGING BRIDGE**" under my supervision from December 2021 to April 2022.

Internal Guide

Dr.S. Prayla Shyry,

Head of the Department

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I, **VISHANTHAN SHREE R** hereby declare that the project report
“WHATSAPP AUTOMATION USING ANDROID DEBUGGING BRIDGE”
done By me under the guidance of **DR.S. PRAYLA SHYRY** is submitted
in partial fulfilment of the requirements for the award of Bachelor of
Engineering Degree in Computer Science and Engineering.

DATE:

PLACE:



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D**, Dean, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D.** and **Dr. L. Lakshmanan, M.E., Ph.D., Head of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.S. Prayla Shyry** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of **the Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

This is a long course, with just about every fact about Python and ADB. Here's a brief subset of the topics covered in the course:

- Master the basics of the language, easily understanding variables, objects, arrays, and functions
- Understand how to design the structure of the code you write, leading to beautiful and easy-to-read programs
- Leverage Python's built-in methods to increase your productive regardless of what libraries or frameworks you use
- Observe how the Python and browser work together, and how to increase the performance of PY code
- Fetch and manage information from third-party ADB commands

Staying up-to-date with relevant information is a key potential to note when working technically with Automation on the app that we needed. This creates a three-way route for most users to know both general chat related details, a precise image shown how well the image is sent and can be done.

This Project **“WHATSAPP AUTOMATION USING ANDROID DEBUGGING BRIDGE”** brings all the above scenarios of general users into a single picture, where they can get every possible. information about word meanings and the examples.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No
	ABSTACT	5
	LIST OF ABBREVIATIONS	
	LIST OF FIGURES	
1	INTRODUCTION	1
2	AIM AND SCOPE OF THE PRESENT INVESTIGATION	2
	2.1 AIM	2
	2.2 SCOPE	2
	2.3 OBJECTIVE	2
	2.4 ARCHITECTURE	2
	2.5 METHODOLOGY	2
3	EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMS USED	4
	3.1 MATERIALS	
	3.2 METHODS AND ALGORITHMS USED	
	3.3 COMPONENTS AND FUNCTIONS USED	
	3.3.1 FUNCTIONS DEFINED	
4	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	12
	4.1 RESULTS	12
5	SUMMARY AND CONCLUSIONS	17
	REFERENCES	18
	APPENDIX	19
	A. SCREENSHOTS	19-21
	B. SOURCE CODE	22-38

LIST OF ABBREVIATIONS

S.NO	SHORTCUT	ABBREVIATION
1.	ADB	Android Debugging Bridge
2.	IPNYB	Jupyter Python Notebook
3.	CLI	Command Line Interface
4.	CSV	Comma-separated values file
5.	EU	End-User
6.	INP	Input
7.	SUBPROCESS	Subprocess management
8.	SDK	Software Development Kit

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
1.0	<i>ARCHITECTURE</i>	11
2.0	<i>METHODOLOGY</i>	11
4.0	<i>HOMEPAGE</i>	17
4.1	<i>WHATSAPP PAGE</i>	18
4.2	<i>SEND SCREEN</i>	19
4.3	<i>THE IMAGE HAS BEEN SENT SUCCESSFULLY</i>	20
4.1.0	<i>PYLINT SCORE</i>	21

CHAPTER-1

INTRODUCTION

A “**WHATSAPP AUTOMATION USING ANDROID DEBUGGING BRIDGE**” uses Android Debug (ADB).

Android Debug Bridge (adb) is a versatile command-line tool that lets you communicate with a device. The adb command facilitates a variety of device actions, such as installing and debugging apps, and it provides access to a Unix shell that you can use to run a variety of commands on a device. It is a client-server program that includes three components:

A client, which sends commands. The client runs on your development machine. You can invoke a client from a command-line terminal by issuing an adb command.

A daemon (adbd), which runs commands on a device. The daemon runs as a background process on each device.

A server, which manages communication between the client and the daemon. The server runs as a background process on your development machine.

CHAPTER-2

AIM AND SCOPE OF THE PRESENT INVESTIGATION:

2.1. AIM:

The main aim of the project is to automate the WhatsApp messaging system with this efficient script.

2.2. SCOPE:

Using Subprocess, CSV and multiple ADB commands to request and do the automation. Reusable components, and lifecycle hooks are used to speed Automation Script. Making the code-reusability to a maximum level. Testing and analysing the script using various tools for high performance.

2.3. OBJECTIVE:

Scaffolding a starter project using PyCharm to make it highly fast and accessible. Using ADB to request multiple ADB commands to fetch the events, and render it to the Script. Finally, bundling the source files for static deployment on any Cloud platforms.

2.4. ARCHITECTURE:

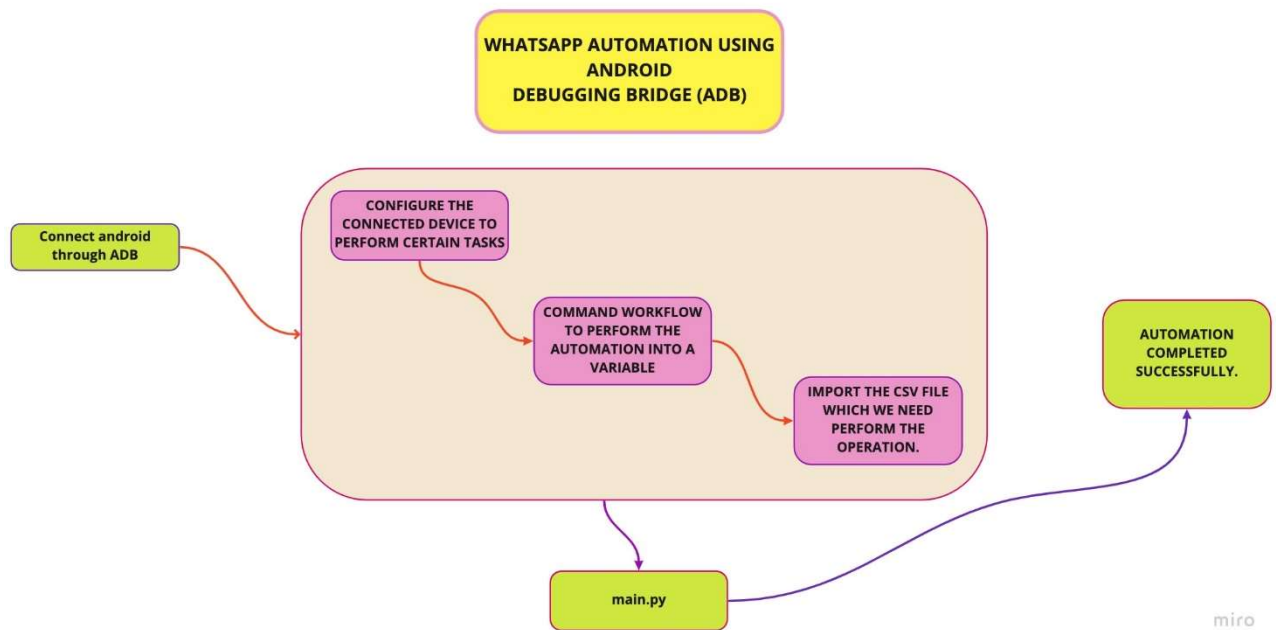


Fig 1.0 Architecture

2.5. METHODOLOGY:

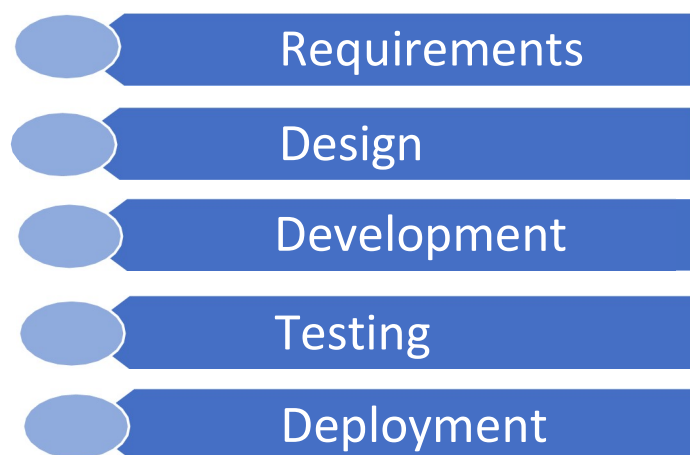


Fig 2.0 Methodology

CHAPTER-3

EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMS USED

3.1. MATERIALS

3.1.1. Minimum 3G/WIFI stable internet connection.
(Recommended)

3.1.2. Android 6.0 or later device. (Recommended)

3.2. METHODS AND ALGORITHMS USED

- Create a project and open main.py using PyCharm IDE.
- Import some required packages at the beginning and install the package.
- Define a function for adb click event and pass the co-ordinates as arguments.
- Define a function for sending message and pass the Phone number and message as arguments.
- Define a function for sending message with image and pass the name, image name, image path as arguments.

- Define a function for running the adb commands and pass the adb commands as arguments.
- Get the co-ordinates of the search icon of WhatsApp using co-ords app and save the co-ordinates in a variable and pass it in click function.
- Get the co-ordinates of the contact's name of WhatsApp using co-ords app and save the co-ordinates in a variable and pass it in click function.

Get the co-ordinates of the Send icon of WhatsApp using co-ords app and save the

coordinates in a variable and pass it in click function.

Get the co-ordinates of the Text box of WhatsApp using co-ords app and save the co-ordinates in a variable and pass it in click function.

- Get the co-ordinates of the Send icon of WhatsApp using co-ords app and save the co-ordinates in a variable and pass it in click function.

3.3. COMPONENTS AND FUNCTIONS USED:

3.3.1. Data.csv

3.3.2. Subprocess

3.3.3. Csv package

3.3.4. Android debugging bridge

3.3.1. Data.csv

The so-called CSV (Comma Separated Values) format is the most common import and export format for spreadsheets and databases. CSV format was used for many years prior to attempts to describe the format in a standardized way in RFC 4180.

This is the input csv file which contains Data set of the students. This contains student name, WhatsApp number, WhatsApp contact name and corresponding certificate ID.

3.3.2. Subprocess

The subprocess module allows you to spawn new processes, connect to their input/output/error pipes, and obtain their return codes.

3.3.3. CSV package

The csv module implements classes to read and write tabular data in CSV format. It allows programmers to say, “write this data in the format preferred by Excel,” or “read data from this file which was generated by Excel,” without knowing the precise details of the CSV format used by Excel. Programmers can also describe the CSV formats understood by other applications or define their own special-purpose CSV formats.

3.3.4. Android Debug Bridge (adb):

Android Debug Bridge is a versatile command-line tool that lets you communicate with a device. The adb command facilitates a variety of device actions, such as installing and debugging apps, and it provides

access to a Unix shell that you can use to run a variety of commands on a device.

There are several functions defined in the main.py script:

They are:

3.4.1. def click(tap_x, tap_y)

3.4.2. def adb(command)

3.4.3. def send_msg(phone,message)

3.4.4. def send_img(name,img_name,img_path)

3.4.5. getting coordinates using co-ords app

3.4.1. def click(tap_x, tap_y)

In the function for adb **click** event and pass the co-ordinates as arguments and it will process the Click event into adb command as **adb("adb shell input tap {} {}".format(tap_x, tap_y))**.

3.4.2. def adb(command)

In the function for running the adb commands and pass the adb commands as arguments as

```
proc = subprocess.Popen(command.split(' '), stdout=subprocess.PIPE, shell=True).
```

3.4.3. def send_msg(phone,message)

In the function for sending message and pass the Phone number and message as arguments and it will pass the arguments into the adb

command as **adb(f'adb shell am start -a android.intent.action.VIEW -d "https://api.whatsapp.com/send?phone={phone}").**

3.4.4. def send_img(name,img_name,img_path)

In the function for sending message with image and pass the name, image name, image path as arguments and abd commands as **adb(f"adb shell am start -a android.intent.action.SEND -t text/plain -e jid '+919566341405@s.whatsapp.net' --eu android.intent.extra.STREAM file:///storage/emulated/0/{img_name} -p com.whatsapp")**

3.4.5. getting coordinates using co-ords app

Get the co-ordinates of the search icon of WhatsApp using co-ords app and save the co-ordinates in a variable and pass it in click function.

Get the co-ordinates of the contact's name of WhatsApp using co-ords app and save the co-ordinates in a variable and pass it in click function.

Get the co-ordinates of the Send icon of WhatsApp using co-ords app and save the coordinates in a variable and pass it in click function.

Get the co-ordinates of the Text box of WhatsApp using co-ords app and save the co-ordinates in a variable and pass it in click function.

Get the co-ordinates of the Send icon of WhatsApp using co-ords app and save the co-ordinates in a variable and pass it in click function.

Finally, all the files are stored in /public folder.To preview the Script locally we just open the main.py and run in the android device.

CHAPTER-4

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

4.1 RESULTS AND DISCUSSION:

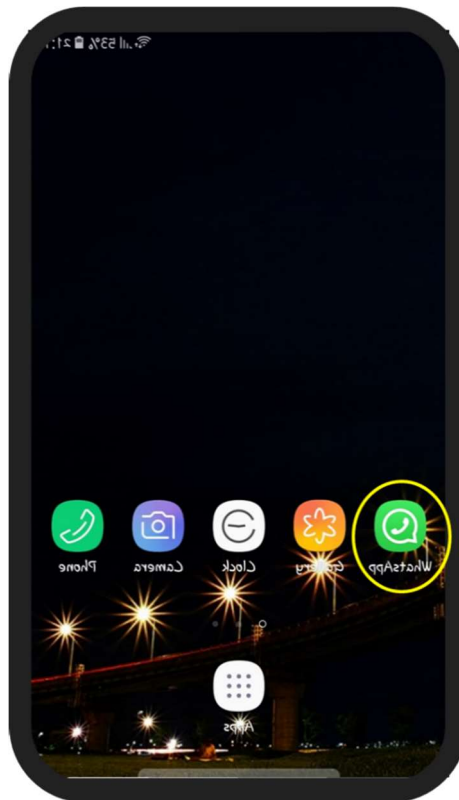


FIG: 4.0 : HOMEPAGE

- This is the first process of our project
- The above output shows the final output of how the homepage looks and works
- This image shows the app that we've automated using android debugging bridge

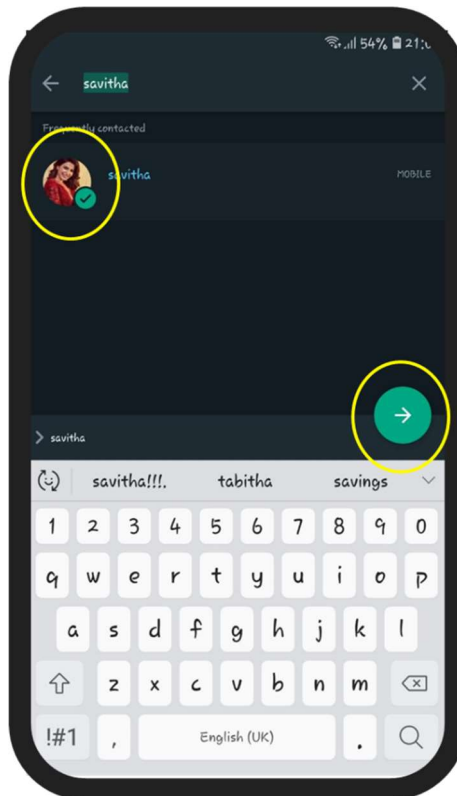


FIG: 4.1 WHATSAPP PAGE

This is the second major process in our automation

- In this image the requested person is selected in WhatsApp.
- The person's name is typed and searched through an automation Script

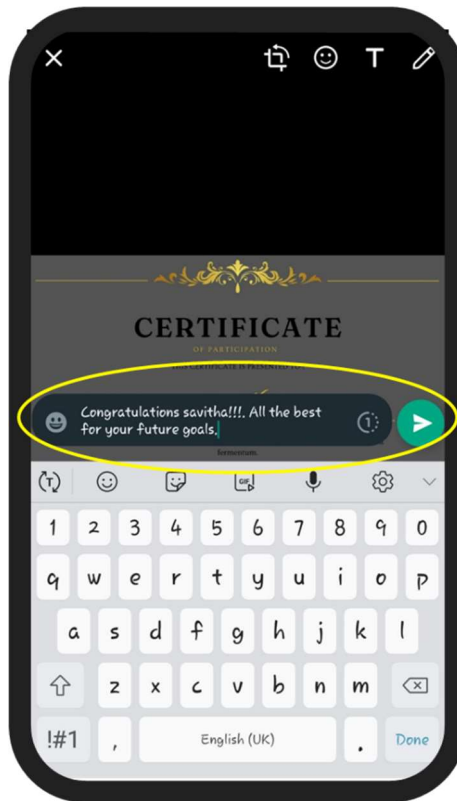


FIG : 4.2 SEND SCREEN

- This is the third major process in our automation
- This is the process where the requested image will be selected to the requested person.
- And also, the requested message will be typed through an automation script.
- And also, this is the process where the send button is pressed, so it is one of the most important processes of all.



FIG: 4.3 THE IMAGE HAS BEEN SENT SUCCESSFULLY

- This is the third major process in our automation
- This is the process where the requested image will be selected to the requested person.
- And also the requested message will be typed through an automation script.
- And also this is the process where the send button is pressed, so it is one of the most important processes of all.

4.1 PERFORMANCE ANALYSIS:

- The performance of this script is very efficient on stable processing speeds (i3 Recommended).
- On i5 processors CPU typically takes less than 30 seconds to complete one loop.
- On i3 processors CPU typically takes less than 45 seconds to complete one loop.
- The script scored 8.49 points in pylint python testing tool.

```
samp.py:23:28: W0621: Redefining name 'img_name' from outer scope (line 60) (redefined-outer-name)
samp.py:23:38: W0621: Redefining name 'img_path' from outer scope (line 61) (redefined-outer-name)
samp.py:23:47: W0621: Redefining name 'name' from outer scope (line 56) (redefined-outer-name)
samp.py:38:8: W1309: Using an f-string that does not have any interpolated variables (f-string-without-interpolation)
samp.py:43:5: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
samp.py:45:4: C0103: Constant name "a" doesn't conform to UPPER_CASE naming style (invalid-name)
samp.py:53:8: C0103: Constant name "a" doesn't conform to UPPER_CASE naming style (invalid-name)
samp.py:55:0: C0206: Consider iterating with .items() (consider-using-dict-items)
samp.py:61:4: C0103: Constant name "img_path" doesn't conform to UPPER_CASE naming style (invalid-name)

-----
Your code has been rated at 8.49/10 (previous run: 8.29/10, +0.20)
```

Fig: 4.1.0 Pylint score

CHAPTER-5

SUMMARY AND CONCLUSIONS

- To summarize, this automation is built using ADB Framework, with python and csv as dependency.
- The automation poses as a one stop destination for sending multiple messages to multiple people.
- The accessibility and responsiveness of the automation, guarantee a large user base on various android device levels.
- The use of python, proves to be lightweight on the machine, and enables fast loading.
- To conclude, the project meets the aim for which it was built. Proving to be positive in all aspects of managing the time in sending multiple WhatsApp messages .

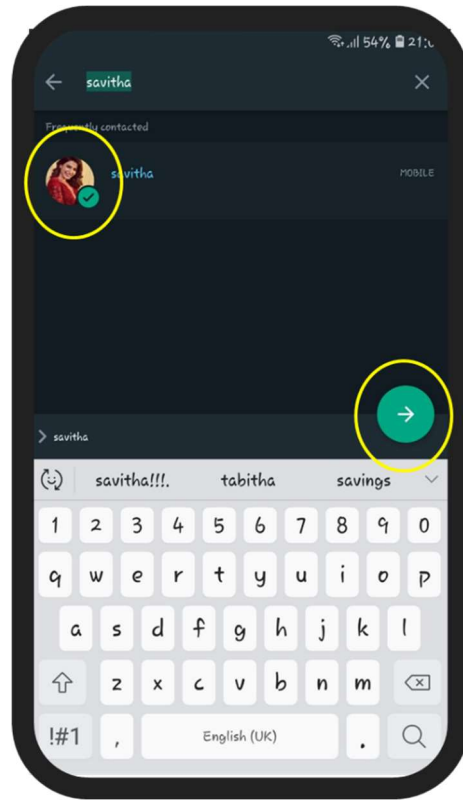
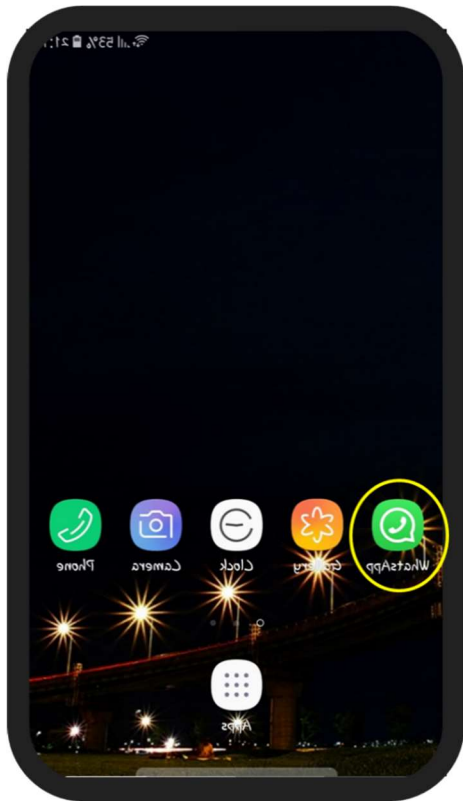
REFERENCES

1. <https://developer.android.com/studio/command-line/adb> : Official documentation for Android Debugging Bridge (ADB)
2. <https://docs.python.org/3/library/subprocess.html> : Official documentation for module Subprocess
3. <https://docs.python.org/3/library/csv.html> : Official documentation for module Comma Separated Values (CSV)
4. <https://developer.android.com/studio/command-line/sdkmanager> : Official documentation for Android SDK Manager
5. <https://docs.python.org/3/> : Official documentation for python3

APPENDIX

A. Screenshots:





B. SOURCE CODE

File name: main.py

File size: 2.28 KB

```
import subprocess
import csv
def click(tap_x, tap_y):
    adb("adb shell input tap {} {}".format(tap_x, tap_y))
def adb(command):
    proc = subprocess.Popen(command.split(' '),
stdout=subprocess.PIPE, shell=True)
    (out, _) = proc.communicate()
    return out.decode('utf-8')
def send_msg(phone,message):
    adb(f'adb shell am start -a android.intent.action.VIEW -d
"https://api.whatsapp.com/send?phone={phone}"')
    adb('ping 127.0.0.1 -n 2 > nul')
    adb(f'adb shell input text "{message}"')
    adb(f'adb shell input keyevent 22 adb shell input keyevent 22
adb shell input keyevent 66')
    print(f'Message sent to {phone}')

def send_img(whatsapp_name,img_name,img_path,name):
    adb(f'adb push {img_path}\\{img_name} /sdcard/')
    adb(f'adb shell am start -a android.intent.action.SEND -t
text/plain -e jid '+919566341405@s.whatsapp.net' --eu
android.intent.extra.STREAM
file:///storage/emulated/0/{img_name} -p com.whatsapp")
    adb('ping 127.0.0.1 -n 1 > nul')
    click(1003,129)
    adb(f'adb shell input text "{whatsapp_name}"')
    adb('ping 127.0.0.1 -n 1 > nul')
```

```

click(309,398)
adb('ping 127.0.0.1 -n 1 > nul')
click(953,1281)
adb('ping 127.0.0.1 -n 2 > nul')
click(456,2127)
adb('ping 127.0.0.1 -n 1 > nul')
adb(f'adb shell input text "Congratulations *{name}*!!!.All
the best for your Future Goals"')
adb('ping 127.0.0.1 -n 1 > nul')
adb(f'adb shell input keyevent 61 adb shell input keyevent
61 adb shell input keyevent 66"')
adb('ping 127.0.0.1 -n 1 > nul')
print(f"sent to {name}")
# driver code
d={}
with open('Data.csv', mode='r') as inp:
    reader = csv.reader(inp)
    a=True
    for i in reader:
        if a==True:
            header = i
            header[0] = header[0].replace("ï»¿", "")
        else:
            i[5] = i[5].replace('","', '"')
            d[i[1]] =
{header[2]:i[2],header[3]:i[3],header[4]:i[4],header[5]:i[5]}
            a=False

for i in d:
    name = d[i]['Student-Name']
    whatsapp_name = d[i]['Whatsapp-Name']
    whatsapp_number = d[i]['Whatsapp-Number']
    message = f"Hello {name}"

```

```
img_name = d[i]['Certificate-ID']
img_path =
'C:\\Users\\visha\\Documents\\PPP\\Certificates'
send_img(whatsapp_name,img_name,img_path,name)
# send_msg(whatsapp_number,message)
```