

**JSS ACADEMY OF TECHNICAL EDUCATION**  
**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**  
**INNOVATIVE METHODS OF TEACHING**  
**2020-21**

| <b>Sl. No</b> | <b>Faculty Name</b>  | <b>Course</b>                                   | <b>Innovative Method</b>   |
|---------------|--|---|--|
| 1             | Pooja H, Savita S  | Cryptography, Network Security and Cyber Law    | Tool Demonstration on vulnerabilities in computing system and security solutions |
| 2             | K S Rajeshwari   | Data Structures and Applications                | Survey on Applications of data structures in real world                          |
| 3             | K V Shanthala  | Management and Entrepreneurship for IT Industry | An Entrepreneurial Case Study  |
| 4             | Dr. Naidila Sadashiv   | Unix System Programming                         | Activity to execute the shell scripts  |
| 5             | K V Shanthala  | User Interface Design                           | Front end development using Open Source Tools                                    |
| 6             | Snehalatha N   | Data Structures and Applications                | On Spot programming, Hands on session with virtual lab                           |
| 7             | Dr Naidila Sadashiv  | Operating Systems                               | MCQs APP   |
| 8             | Dr. Prabhudev Jagadeesh.<br>M.P, Snehalatha N,<br>Rohitaksha K | Internet of Things                              | IDEATHON & Quiz  |

**JSS ACADEMY OF TECHNICAL EDUCATION**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

**Date: 3/1/2021**

**Faculty Name : Savita S**

**Class/Section: VI 'B'**

**Activity Name:** Exploring some requirements for some cryptographic applications and demonstration for the same using open source tools.

**Academic Year: 2020-21**

**GOAL OF THE ACTIVITY:**

- To make students recognize, categorize, characterize the security requirement and Examine how the key management is conducted for any one cryptographic application.
- Show the demonstration of the same application with the help of some open source Cryptographic tool(like Kali Linux, Cryptool etc)

**DESCRIPTION OF ACTIVITY:**

For any one among the following applications explore

- 1) Examine the security requirements.
- 2) Analyze the application constraints that influence decision making.
- 3) List the cryptographic primitives that are deployed.
- 4) Compare the cryptographic algorithms and key lengths that are Supported.
- 5) Examine how the key management is conducted.

Show the demonstration of the same application with the help of some open source Cryptographic tool(like Kali Linux, Cryptool etc)

Note: Students are free to take up any real time application of their choice.

Few applications are listed below

| SI No. | Topics   |
|--------|--|
| 1      | Ticketing Application  |
| 2      | ATM pin protection   |
| 3      | Payment card key management  |
| 4      | Size of the password space   |
| 5      | Security in pay-TV application                                       |
| 6      | Passwords stored on a computer                                       |
| 7      | Biometric technologies that provide a source of identity information |
| 8      | Cryptography in ID cards   |
| 9      | Cryptography in mobile communication network                         |
| 10     | Security in electronic signatures                                    |
| 11     | Security in subscriber identification module (SIM) card              |
| 12     | Cryptography in payment card services                                |
| 13     | Security for IT support team   |
| 14     | Email security   |
| 15     | HTTP Secure  |
| 16     | Cryptography in Whatsapp   |
| 17     | Disk encryption  |
| 18     | Home security from the perspective of home user securing email.      |

- A report with proper format should be submitted.

**RESULTS/OUTCOME :**

Awareness on how the cryptography plays a supporting role in the different applications by exploring the different parameters in providing security for Information and Network Applications.

Number of students participated 31

**JSS Mahavidyapeetha**

# **JSS Academy of Technical Education**

**Kengeri-Uttarahalli Main Road, Bangalore-560060**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**COURSE NAME:** Information and Network Security

**COURSE CODE:** 17CS743

**TOPIC:** Network Traffic Analysis using Packet Sniffer

**Prepared by:** Karthik Ram M

1JS17CS044

**Semester / Branch:** 7th Sem/CSE

**Marks Scored:**

**Signature of Faculty:**

## **ABSTRACT**

Network traffic analysis (NTA) is the process of intercepting, recording and analysing network traffic communication patterns in order to detect and respond to security threats. Originally coined by Gartner, the term represents an emerging security product category. While other network security tools such as firewalls and intrusion detection system (IDS)/intrusion prevention system (IPS) products focus on monitoring vertical traffic that crosses the perimeter of a network environment, network traffic analysis solutions are focused on all communications – whether those are traditional TCP/IP style packets, “virtual network traffic” crossing a virtual switch (or “vSwitch”), traffic from and within cloud workloads, and API calls to SaaS applications or serverless computing instances. These solutions also focus on operational technology and Internet of things (IoT) networks that are otherwise completely invisible to the security team. Advanced NTA tools are even effective when network traffic is encrypted. The first generation of this technology focused on establishing a baseline of what’s ‘normal’ or ‘good’ and then pinpoint anomalies that could be ‘irregular’ or ‘bad.’ This approach has the downside of being noisy as business and IT evolves all the time for very legitimate reasons. Advanced NTA tools operate in a more intelligent manner by comparing not just to past behaviour but also to other entities in the environment.

# TABLE OF CONTENTS

ABSTRACT .....2  
TABLE OF CONTENTS .....3  
INTRODUCTION .....4  
APPLICABLE QUESTIONS.....5  
METHOD OF IMPLEMENTATION .....6  
RESULTS.....9  
CONCLUSION .....21  
FUTURE SCOPE .....22  
REFERENCES .....22

# INTRODUCTION

Network Traffic Analysis (NTA) systems detect information security threats by analysing events at the level of the network. They allow security specialists to detect attacks at an early stage, effectively isolate threats, and ensure that security guidelines are met. It is essential to monitor network security in order to prevent hackers from conducting an attack inside a network. This is where NTA systems come into play.

One way to implement NTA systems is to use packet analysers or sniffers. When any data has to be transmitted over the computer network, it is broken down into smaller units at the sender's node called data packets, and reassembled at the receiver's node to the original format. The packet is the smallest unit of communication over a computer network. The act of capturing and logging some or all data packet that pass through a computer network, is referred to as packet sniffing. This is done using tools called packet sniffers. They can be either Filtered i.e., when only certain pre-specified type/s of data packets are captured, and Unfiltered i.e., when all the types of packets are to be captured. After capturing the data packets like TCP, UDP, ICMP, etc., the sniffers also provide facilities to extract these data packets. They may also be used for further analysis, such as to monitor network traffic, performance, etc, because of which they are also called packet analysers.

Packet sniffing consists of 2 parts:

First, a network adapter that connects the packet sniffer to an existing network.

Second, a software that provides a way to log, see, or further analyse the data collected. There are two main types of packet sniffers:

- **Hardware Packet Sniffers** - A hardware packet sniffer is designed to be plugged into a network in order to examine it. A hardware packet sniffer is particularly useful when attempting to see traffic of a specific network segment. A hardware packet sniffer either stores the collected packets or forwards them on to a collector that logs the data.
- **Software Packet Sniffers** - Software packet sniffers are more common. While any network interface attached to a network can receive all of the network traffic that flows by, most are configured not to do so. A software packet sniffer changes this configuration so that the network interface passes all network traffic to it also. This configuration is known as promiscuous mode.

Packet sniffers collect entire packets, hence packets that aren't encrypted can be reassembled and read in entirety. For example, intercepted packets from a user accessing a website would include the HTML and CSS of the web pages. So users logging in to network resources across unencrypted transmissions expose their username and password as plain text that can be seen in captured packets.

Packet sniffing can be done by anyone with access to a router. It is generally done by:

1. Internet Service Providers (ISPs), they use packet sniffing to track all activities such as:
  - What is download?
  - Which sites are visited?
  - Streaming events like video, audio, etc.
2. Advertising agencies or internet advertising agencies, this is because they are paid according to the number of ads shown by them, the number of clicks on their ads (also called PPC-Pay per click), etc.

To achieve this target, these agencies use packet sniffing to inject advertisements into the flowing packets.

3. Government agencies to:
  - Ensure security of data over the network.
  - Track an organisation's data transmissions, etc.
  - Conduct mass surveillance.
4. Businesses to:
  - Determine what its competitor's doing (usually by business espionage agents)
  - Steal competitor's information, research, etc.
5. Black hats (people who violate computer security for personal gain, or for pure maliciousness.) Which may be to:
  - Steal credit card numbers
  - Harvesting personal data for sale to identity, etc.

Packet sniffing can be used for:

- Network Troubleshooting
- Analysing network traffic
- Capturing the clear text usernames and passwords
- Capturing and replying VoIP telephone conversations
- Conversion of network traffic into human readable form

## **APPLICABLE QUESTIONS**

Q: Analyze the applications constraints that influence decision making?

Ans: The applications constraints that influence decision making are:

- No matter how good the packet sniffer is, it can only see network traffic, that too of only the network traffic crossing the data lines that the packet sniffer's connected to.
- Most Packet sniffer's don't create metadata about the packets and flows at the time of capture of the Packet. But this they are useful for analysis and correlation, and hence searching for what's needed becomes difficult and an analyst will have to do much of the data analysis/crunching by hand prior to rendering any decisions.
- Just because any data can be captured by a packet sniffer, it doesn't mean it should be captured. Many countries and state governments have strict privacy laws that will be broken by full packet capture, and devising a sniffer accordingly can sometimes turn out to become a big task.
- Personnel having access to the sniffed data must be highly trustworthy and should be vetted to ensure they maintain the highest standards possible in order to avoid misusing the data to which they have access. This itself might turn out to be difficult to fulfil.
- Information that were not encrypted in transit are the only things that can be made use of directly, but in the scenario of encrypted transmission, it is totally blind.



## METHOD OF IMPLEMENTATION

Inspiration to choose this topic is from a tool called Scapy, which is a network monitoring tool from Kali Linux. The output of the Scapy tool is not easily understood due to its complexity. Hence, we have used Python to implement the same. We have used the Socket module of the Python programming language. Linux OS is preferred because Windows OS's inbuilt firewall does not allow 3<sup>rd</sup> party sniffers. Below is the code used to implement the packet sniffing using Python.

```
import socket
import struct
import binascii
import os

# Ethernet Header Extraction
def eth_header(data):
    ethh=struct.unpack("!6s6sH",data)
    destination_mac=binascii.hexlify(ethh[0])
    source_mac=binascii.hexlify(ethh[1])
    eth_protocol=ethh[2]
    data={"Destination Mac":destination_mac,
        "Source Mac":source_mac,
        "Protocol":eth_protocol}
    return data

# ICMP Header Extraction
def icmp_header(data):
    icmp_h=struct.unpack('!BBH',data)
    icmp_type=icmp_h[0]
    code=icmp_h[1]
    checksum=icmp_h[2]
    data={'ICMP Type':icmp_type,
        "Code":code,
        "Checksum":checksum}
    return data

# UDP Header Extraction
def udp_header(data):
    udph=struct.unpack('!HHHH',data)
    source_port=udph[0]
    dest_port=udph[1]
    length=udph[2]
    checksum=udph[3]
    data={"Source Port":source_port,
        "Destination Port":dest_port,
        "Length":length,
        "Checksum":checksum}
    return data
```

```

# IP Header Extraction
def ip_header(data):
    iph=struct.unpack("!BBHHBBH4s4s", data)
    version=iph[0]
    tos=iph[1]
    total_length=iph[2]
    identification=iph[3]
    fragment_Offset=iph[4]
    ttl=iph[5]
    protocol=iph[6]
    v=-1
    if protocol==1 or protocol==6 or protocol==17:
        v=protocol
    header_checksum=iph[7]
    source_address=socket.inet_ntoa(iph[8])
    destination_address=socket.inet_ntoa(iph[9])

    data={'Version':version,
          "Tos":tos,
          "Total Length":total_length,
          "Identification":identification,
          "Fragment":fragment_Offset,
          "TTL":ttl,
          "Protocol":protocol,
          "Header CheckSum":header_checksum,
          "Source Address":source_address,
          "Destination Address":destination_address}
    return (data,v)

# TCP Header Extraction
def tcp_header(data):
    tcph=struct.unpack('!HLLBBHHH', data)
    source_port=tcph[0]
    destination_port=tcph[1]
    sequence_number=tcph[2]
    acknowledge_number=tcph[3]
    offset_reserved=tcph[4]
    tcp_flag=tcph[5]
    window=tcph[6]
    checksum=tcph[7]
    urgent_pointer=tcph[8]

    data={"Source Port":source_port,
          "Destination Port":destination_port,
          "Sequence Number":sequence_number,
          "Acknowledge Number":acknowledge_number,
          "Offset & Reserved":offset_reserved,
          "Tcp Flag":tcp_flag,
          "Window":window,
          "CheckSum":checksum,
          "Urgent Pointer":urgent_pointer}

```

```

return data

# Mac address formating
def mac_formatter(a):
    b="%.2x:%.2x:%.2x:%.2x:%.2x:%.2x" % (a[0],a[1],a[2],a[3],a[4],a[5])
    return b

# if operating system is Windows
if os.name == "nt":
    s = socket.socket(socket.AF_INET,socket.SOCK_RAW,socket.IPPROTO_IP)
    s.bind(("localhost",9999))
    s.setsockopt(socket.IPPROTO_IP,socket.IP_HDRINCL,1)
    s.ioctl(socket.SIO_RCVALL,socket.RCVALL_ON)
    print('On Windows OS :')

# if operating system is non-Windows
else:
    s=socket.socket(socket.PF_PACKET, socket.SOCK_RAW, socket.ntohs(0x0
800))
    print('On non-Windows OS :')

print('Assuming the packets to be IP packets (which is usually that, as
we are sniffing packets only from the internet for this packet sniffer
)\n')

for z in range(15):

    # Capture packets from network
    pkt = s.recvfrom(9999)

    print('Packet no :',z+1)
    print("----- Ethernet Header -----")
    t=eth_header(pkt[0][0:14])
    for i in t:
        if "Mac" in i:
            print(i," : ",mac_formatter(t[i])," |",end=' ')
        else:
            print(i," : ",t[i]," |",end=' ')
    print()
    print("----- IP Header -----")
    t,v=ip_header(pkt[0][14:34])
    for i in t:
        print(i," : ",t[i]," |",end=' ')
    print()

    if v==6:
        st='TCP'
        lld=54
        t=tcp_header(pkt[0][34:54])
    elif v==17:
        st='UDP'
        lld=42

```

```

        t=udp_header(pkt[0][34:42])
elif v==1:
    st='ICMP'
    lld=38
    t=icmp_header(pkt[0][34:38])
else:
    print('Packet is not of type TCP or UDP or ICMP (such packets are
not fully sniffed by this packet sniffer)')
    continue
print('-----',st,'Header -----')
for i in t:
    print(i," : ",t[i]," |",end=' ')
print('\n\nPayload (from Application Layer: not extracted or format
ted) :')
print(pkt[0][lld:])
if lld==38:
    print("['Data' field may contain a part of the ICMP header(this p
art is not provided explicitly; its usage varies based on the 'Type' an
d 'Code' fields]")
    print('\n\n')

```

## RESULTS

The software variant of the packet sniffer has been implemented using python programming language, in which we made use of the socket module to set up a socket, and capture the packets. A socket in general networking terms is an abstract representation (handle) for the local endpoints of a network communication path. We then obtain the packet's, source, destination information, etc details (these can be modified according to the requirement of the software) using socket and other modules (such as struct, binascii, etc).

Below are the sequence of outputs showing the packet sniffer at work. It has been implemented and tested on Google CoLab.

```

On non-Windows OS :
Assuming the packets to be IP packets (which is usually that, as we are
sniffing packets only from the internet for this packet sniffer)

Packet no : 1
----- Ethernet Header -----
Destination Mac : 30:30:30:30:30:30 | Source Mac :
30:30:30:30:30:30 | Protocol : 2048 |
----- IP Header -----
Version : 69 | Tos : 0 | Total Length : 659 | Identification
: 5409 | Fragment : 16384 | TTL : 64 | Protocol : 6 | Header
Checksum : 9538 | Source Address : 127.0.0.1 | Destination
Address : 127.0.0.1 |
----- TCP Header -----
Source Port : 33633 | Destination Port : 52978 | Sequence Number
: 2526275576 | Acknowledge Number : 3008018757 | Offset & Reserved
: 128 | Tcp Flag : 24 | Window : 512 | CheckSum : 136 |
Urgent Pointer : 0 |

```

```
Payload (from Application Layer: not extracted or formatted) :
b'\x01\x01\x08\nv\xee0\xd3v\xeeNc\x01\rstream.stdout\x01\t<IDS|MSG>\x01
\x00\x01\xbb{"version":"5.3","date":"2020-12-
18T17:48:33.542940Z","session":"18665022-
0e36cbcbba37f542def4b568","username":"username","msg_type":"stream","ms
g_id":"acfb1125-
4415c683d933bf1ff6d8a54f"}\x01\xc2{"username":"username","msg_type":"ex
ecute_request","msg_id":"ef01a40848cb41f6d21f0f4c669e0b28","version":"5
.0","session":"2e9f696e12704087d895f3474376ddc5","date":"2020-12-
18T17:48:33.140050Z"}\x01\x02{} \x00\xbc{"name":"stdout","text":"On non-
Windows OS : \n\nAssuming the packets to be IP packets (which is usually
that, as we are sniffing packets only from the internet for this packet
sniffer)\n\n"}'
```

Packet no : 2

```
----- Ethernet Header -----
Destination Mac : 30:30:30:30:30:30 | Source Mac :
30:30:30:30:30:30 | Protocol : 2048 |
----- IP Header -----
Version : 69 | Tos : 0 | Total Length : 52 | Identification :
50465 | Fragment : 16384 | TTL : 64 | Protocol : 6 | Header
Checksum : 30624 | Source Address : 127.0.0.1 | Destination
Address : 127.0.0.1 |
----- TCP Header -----
Source Port : 52978 | Destination Port : 33633 | Sequence Number
: 3008018757 | Acknowledge Number : 2526276183 | Offset & Reserved
: 128 | Tcp Flag : 16 | Window : 503 | CheckSum : 65064 |
Urgent Pointer : 0 |
```

```
Payload (from Application Layer: not extracted or formatted) :
b'\x01\x01\x08\nv\xee0\xd3v\xee0\xd3'
```

Packet no : 3

```
----- Ethernet Header -----
Destination Mac : 30:30:30:30:30:30 | Source Mac :
30:30:30:30:30:30 | Protocol : 2048 |
----- IP Header -----
Version : 69 | Tos : 0 | Total Length : 659 | Identification
: 9619 | Fragment : 16384 | TTL : 64 | Protocol : 6 | Header
Checksum : 5328 | Source Address : 127.0.0.1 | Destination
Address : 127.0.0.1 |
----- TCP Header -----
Source Port : 33633 | Destination Port : 52996 | Sequence Number
: 3829231461 | Acknowledge Number : 2458009640 | Offset & Reserved
: 128 | Tcp Flag : 24 | Window : 512 | CheckSum : 136 |
Urgent Pointer : 0 |
```

```
Payload (from Application Layer: not extracted or formatted) :
b'\x01\x01\x08\nv\xee0\xd3v\xeeNc\x01\rstream.stdout\x01\t<IDS|MSG>\x01
\x00\x01\xbb{"version":"5.3","date":"2020-12-
18T17:48:33.542940Z","session":"18665022-
0e36cbcbba37f542def4b568","username":"username","msg_type":"stream","ms
g_id":"acfb1125-
4415c683d933bf1ff6d8a54f"}\x01\xc2{"username":"username","msg_type":"ex
ecute_request","msg_id":"ef01a40848cb41f6d21f0f4c669e0b28","version":"5
.0","session":"2e9f696e12704087d895f3474376ddc5","date":"2020-12-
18T17:48:33.140050Z"}\x01\x02{} \x00\xbc{"name":"stdout","text":"On non-
Windows OS : \n\nAssuming the packets to be IP packets (which is usually
```

that, as we are sniffing packets only from the internet for this packet sniffer)\n\n"}'

Packet no : 4

----- Ethernet Header -----

Destination Mac : 30:32:34:32:61:63 | Source Mac :  
30:32:34:32:61:63 | Protocol : 2048 |

----- IP Header -----

Version : 69 | Tos : 0 | Total Length : 52 | Identification :  
2785 | Fragment : 16384 | TTL : 64 | Protocol : 6 | Header  
Checksum : 55205 | Source Address : 172.28.0.3 | Destination  
Address : 172.28.0.2 |

----- TCP Header -----

Source Port : 55944 | Destination Port : 9000 | Sequence Number  
: 2530555647 | Acknowledge Number : 3177701435 | Offset & Reserved  
: 128 | Tcp Flag : 16 | Window : 501 | CheckSum : 22628 |  
Urgent Pointer : 0 |

Payload (from Application Layer: not extracted or formatted) :

b'\x01\x01\x08\n\xfa[K<\xa9\xbfW\x14'

Packet no : 5

----- Ethernet Header -----

Destination Mac : 30:32:34:32:61:63 | Source Mac :  
30:32:34:32:61:63 | Protocol : 2048 |

----- IP Header -----

Version : 69 | Tos : 0 | Total Length : 814 | Identification  
: 16179 | Fragment : 16384 | TTL : 64 | Protocol : 6 |  
Header CheckSum : 41049 | Source Address : 172.28.0.3 |  
Destination Address : 172.28.0.2 |

----- TCP Header -----

Source Port : 6000 | Destination Port : 57482 | Sequence Number  
: 2738570308 | Acknowledge Number : 3233590468 | Offset & Reserved  
: 128 | Tcp Flag : 24 | Window : 501 | CheckSum : 23390 |  
Urgent Pointer : 0 |

Payload (from Application Layer: not extracted or formatted) :

b'\x01\x01\x08\n\xfa[K=\xa9\xbfU\xa7\x81~\x02\xf6{"parent\_header":  
{"username": "username", "version": "5.0", "msg\_type":  
"execute\_request", "msg\_id": "ef01a40848cb41f6d21f0f4c669e0b28",  
"session": "2e9f696e12704087d895f3474376ddc5", "date": "2020-12-  
18T17:48:33.140050Z"}, "msg\_type": "stream", "msg\_id": "acfb1125-  
4415c683d933bf1ff6d8a54f", "content": {"text": "On non-Windows OS  
:\n\nAssuming the packets to be IP packets (which is usually that, as we  
are sniffing packets only from the internet for this packet  
sniffer)\n\n", "name": "stdout"}, "header": {"username": "username",  
"version": "5.3", "msg\_type": "stream", "msg\_id": "acfb1125-  
4415c683d933bf1ff6d8a54f", "session": "18665022-  
0e36cbcbba37f542def4b568", "date": "2020-12-18T17:48:33.542940Z"},  
"channel": "iopub", "buffers": [], "metadata": {}}'

Packet no : 6

----- Ethernet Header -----

Destination Mac : 30:30:30:30:30:30 | Source Mac :  
30:30:30:30:30:30 | Protocol : 2048 |

----- IP Header -----

Version : 69 | Tos : 0 | Total Length : 52 | Identification : 51501 | Fragment : 16384 | TTL : 64 | Protocol : 6 | Header CheckSum : 29588 | Source Address : 127.0.0.1 | Destination Address : 127.0.0.1 |

----- TCP Header -----

Source Port : 52996 | Destination Port : 33633 | Sequence Number : 2458009640 | Acknowledge Number : 3829232068 | Offset & Reserved : 128 | Tcp Flag : 16 | Window : 512 | CheckSum : 65064 | Urgent Pointer : 0 |

Payload (from Application Layer: not extracted or formatted) :  
b'\x01\x01\x08\nv\xee0\xfbv\xee0\xd3'

Packet no : 7

----- Ethernet Header -----

Destination Mac : 30:32:34:32:61:63 | Source Mac : 30:32:34:32:62:33 | Protocol : 2048 |

----- IP Header -----

Version : 69 | Tos : 0 | Total Length : 620 | Identification : 42092 | Fragment : 16384 | TTL : 64 | Protocol : 6 | Header CheckSum : 15332 | Source Address : 172.28.0.1 | Destination Address : 172.28.0.2 |

----- TCP Header -----

Source Port : 33692 | Destination Port : 8080 | Sequence Number : 2022589041 | Acknowledge Number : 3787639457 | Offset & Reserved : 128 | Tcp Flag : 24 | Window : 501 | CheckSum : 23194 | Urgent Pointer : 0 |

Payload (from Application Layer: not extracted or formatted) :

b'\x01\x01\x08\n\x01A..\xd8`GET  
/socket.io/?EIO=3&sid=kcvVbgvaJ9lhmYx7AAAA&t=NPszx94&transport=polling  
HTTP/1.1\r\nHost: colab.research.google.com\r\nUser-Agent: Mozilla/5.0  
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)  
Chrome/87.0.4280.88 Safari/537.36,gzip(gfe)\r\nAccept: \*/\*\r\nAccept-  
Encoding: gzip, deflate, br,gzip(gfe)\r\nAccept-Language: en-  
GB,en;q=0.9\r\nReferer: <https://colab.research.google.com/>\r\nSec-Ch-  
Ua: "Google Chrome";v="87", " Not;A Brand";v="99",  
"Chromium";v="87"\r\nSec-Ch-Ua-Mobile: ?0\r\nSec-Fetch-Dest:  
empty\r\nSec-Fetch-Mode: cors\r\nSec-Fetch-Site: same-origin\r\n\r\n'

Packet no : 8

----- Ethernet Header -----

Destination Mac : 30:32:34:32:61:63 | Source Mac : 30:32:34:32:62:33 | Protocol : 2048 |

----- IP Header -----

Version : 69 | Tos : 0 | Total Length : 52 | Identification : 42093 | Fragment : 16384 | TTL : 64 | Protocol : 6 | Header CheckSum : 15899 | Source Address : 172.28.0.1 | Destination Address : 172.28.0.2 |

----- TCP Header -----

Source Port : 33692 | Destination Port : 8080 | Sequence Number : 2022589609 | Acknowledge Number : 3787640560 | Offset & Reserved : 128 | Tcp Flag : 16 | Window : 501 | CheckSum : 22626 | Urgent Pointer : 0 |

Payload (from Application Layer: not extracted or formatted) :  
b'\x01\x01\x08\n\x01A/..\xd9\xe5'

```
Packet no : 9
----- Ethernet Header -----
Destination Mac : 30:30:30:30:30:30 | Source Mac : 
30:30:30:30:30:30 | Protocol : 2048 |
----- IP Header -----
Version : 69 | Tos : 0 | Total Length : 8244 | Identification 
: 5410 | Fragment : 16384 | TTL : 64 | Protocol : 6 | Header 
Checksum : 1952 | Source Address : 127.0.0.1 | Destination 
Address : 127.0.0.1 |
----- TCP Header -----
Source Port : 33633 | Destination Port : 52978 | Sequence Number 
: 2526276183 | Acknowledge Number : 3008018757 | Offset & Reserved 
: 128 | Tcp Flag : 24 | Window : 512 | CheckSum : 7721 | 
Urgent Pointer : 0 |
```

```
Payload (from Application Layer: not extracted or formatted) :
b'\x01\x01\x08\nv\xeeP\x9cv\xeeO\xd3\x01\rstream.stdout\x01\t<IDS|MSG>\x01\x00\x01\xbb{"version":"5.3","date":"2020-12-18T17:48:33.744088Z","session":"18665022-0e36cbcbba37f542def4b568","username":"username","msg_type":"stream","msg_id":"f991d6cf-d4c32e403d39040799d4a98b"}\x01\xc2{"username":"username","msg_type":"execute_request","msg_id":"ef01a40848cb41f6d21f0f4c669e0b28","version":"5.0","session":"2e9f696e12704087d895f3474376ddc5","date":"2020-12-18T17:48:33.140050Z"}\x01\x02{} \x02\x00\x00\x00\x00\x00\x00%a{"name":"stdout","text":"Packet no : 1\n----- Ethernet Header -----\n\nDestination Mac : 30:30:30:30:30:30 | Source Mac : 
30:30:30:30:30:30 | Protocol : 2048 | \n----- IP Header ---
-----\nVersion : 69 | Tos : 0 | Total Length : 659 | 
Identification : 5409 | Fragment : 16384 | TTL : 64 | Protocol 
: 6 | Header CheckSum : 9538 | Source Address : 127.0.0.1 | 
Destination Address : 127.0.0.1 | \n----- TCP Header -----
----\nSource Port : 33633 | Destination Port : 52978 | Sequence 
Number : 2526275576 | Acknowledge Number : 3008018757 | Offset & 
Reserved : 128 | Tcp Flag : 24 | Window : 512 | CheckSum : 
136 | Urgent Pointer : 0 | \n\nPayload (from Application Layer: 
not extracted or formatted)
:\nb'\x01\x01\x08\nv\xeeO\xd3v\xeeNc\x01\rstream.stdout\x01\t<IDS|MSG>\x01\x00\x01\xbb{"version":"5.3","date":"2020-12-18T17:48:33.542940Z","session":"18665022-0e36cbcbba37f542def4b568","username":"username","msg_type":"stream","msg_id":"acfb1125-4415c683d933bf1ff6d8a54f"}\x01\xc2{"username":"username","msg_type":"execute_request","msg_id":"ef01a40848cb41f6d21f0f4c669e0b28","version":"5.0","session":"2e9f696e12704087d895f3474376ddc5","date":"2020-12-18T17:48:33.140050Z"}\x01\x02{} \x00\xbc{"name":"stdout","text":"On non-Windows OS : \n\nAssuming the packets to be IP packets (which is usually that, as we are sniffing packets only from the internet for this packet sniffer)\n\nPacket no : 2\n-----
- Ethernet Header -----\nDestination Mac : 30:30:30:30:30:30 
| Source Mac : 30:30:30:30:30:30 | Protocol : 2048 | \n-----
--- IP Header -----\nVersion : 69 | Tos : 0 | Total 
Length : 52 | Identification : 50465 | Fragment : 16384 | TTL 
: 64 | Protocol : 6 | Header CheckSum : 30624 | Source Address 
: 127.0.0.1 | Destination Address : 127.0.0.1 | \n-----
TCP Header -----\nSource Port : 52978 | Destination Port : 
33633 | Sequence Number : 3008018757 | Acknowledge Number :
```



```
2526276183 | Offset & Reserved : 128 | Tcp Flag : 16 | Window :  
503 | CheckSum : 65064 | Urgent Pointer : 0 | \\n\\nPayload  
(from Application Layer: not extracted or formatted)  
:\\nb'\\x01\\x01\\x08\\nv\\xeeO\\xd3v\\xeeO\\xd3'\\n\\  
\\n\\n\\nPacket no : 3\\n----- Ethernet Header -----  
\\nDestination Mac : 30:30:30:30:30:30 | Source Mac :  
30:30:30:30:30:30 | Protocol : 2048 | \\n----- IP Header ---  
-----\\nVersion : 69 | Tos : 0 | Total Length : 659 |  
Identification : 9619 | Fragment : 16384 | TTL : 64 | Protocol  
: 6 | Header CheckSum : 5328 | Source Address : 127.0.0.1 |  
Destination Address : 127.0.0.1 | \\n----- TCP Header -----  
----\\nSource Port : 33633 | Destination Port : 52996 | Sequence  
Number : 3829231461 | Acknowledge Number : 2458009640 | Offset &  
Reserved : 128 | Tcp Flag : 24 | Window : 512 | CheckSum :  
136 | Urgent Pointer : 0 | \\n\\n\\nPayload (from Application Layer:  
not extracted or formatted)  
:\\nb'\\x01\\x01\\x08\\nv\\xeeO\\xd3v\\xeeNc\\x01\\r  
stream.stdout\\x01\\t<IDS|MSG>\\x01\\x00\\x01\\xb{\\\"versi  
on\\\":\\\"5.3\\\",\\\"date\\\":\\\"2020-12-  
18T17:48:33.542940Z\\\",\\\"session\\\":\\\"18665022-  
0e36cbcbba37f542def4b568\\\",\\\"username\\\":\\\"username\\\",\\\"msg_type\\  
\\\":\\\"stream\\\",\\\"msg_id\\\":\\\"acfb1125-  
4415c683d933bf1ff6d8a54f\\\"}\\x01\\xc2{\\\"username\\\":\\\"username\\  
\\\",\\\"msg_type\\\":\\\"execute_request\\\",\\\"msg_id\\\":\\\"ef01a40848cb41f6  
d21f0f4c669e0b28\\\",\\\"version\\\":\\\"5.0\\\",\\\"session\\\":\\\"2e9f696e12  
704087d895f3474376ddc5\\\",\\\"date\\\":\\\"2020-12-  
18T17:48:33.140050Z\\\"}\\x01\\x02{\\x00\\x00\\x00\\x00{\\\"name\\\":\\\"std  
out\\\",\\\"text\\\":\\\"On non-Windows OS :\\n\\n\\nAssuming the packets  
to be IP packets (which is usually that, as we are sniffing packets  
only from the internet for this packet  
sniffer)\\n\\n\\n}\\n\\n\\n\\n\\nPacket no : 4\\n-----  
- Ethernet Header -----\\nDestination Mac : 30:32:34:32:61:63  
| Source Mac : 30:32:34:32:61:63 | Protocol : 2048 | \\n-----  
--- IP Header -----\\nVersion : 69 | Tos : 0 | Total  
Length : 52 | Identification : 2785 | Fragment : 16384 | TTL  
: 64 | Protocol : 6 | Header CheckSum : 55205 | Source Address  
: 172.28.0.3 | Destination Address : 172.28.0.2 | \\n-----  
TCP Header -----\\nSource Port : 55944 | Destination Port :  
9000 | Sequence Number : 2530555647 | Acknowledge Number :  
3177701435 | Offset & Reserved : 128 | Tcp Flag : 16 | Window :  
501 | CheckSum : 22628 | Urgent Pointer : 0 | \\n\\n\\nPayload  
(from Application Layer: not extracted or formatted)  
:\\nb'\\x01\\x01\\x08\\n\\xfa[K<\\xa9\\xbfW\\x14'\\n\\  
\\n\\n\\nPacket no : 5\\n----- Ethernet Header -----  
\\nDestination Mac : 30:32:34:32:61:63 | Source Mac :  
30:32:34:32:61:63 | Protocol : 2048 | \\n----- IP Header ---  
-----\\nVersion : 69 | Tos : 0 | Total Length : 814 |  
Identification : 16179 | Fragment : 16384 | TTL : 64 |  
Protocol : 6 | Header CheckSum : 41049 | Source Address :  
172.28.0.3 | Destination Address : 172.28.0.2 | \\n----- TCP  
Header -----\\nSource Port : 6000 | Destination Port : 57482  
| Sequence Number : 2738570308 | Acknowledge Number : 3233590468  
| Offset & Reserved : 128 | Tcp Flag : 24 | Window : 501 |  
CheckSum : 23390 | Urgent Pointer : 0 | \\n\\n\\nPayload (from  
Application Layer: not extracted or formatted)  
:\\nb'\\x01\\x01\\x08\\n\\xfa[K=\\xa9\\xbfU\\xa7\\x8  
1~\\x02\\xf6{\\\"parent_header\\\":{\\\"username\\\":\\\"username\\\",  
\\\"version\\\":\\\"5.0\\\",\\\"msg_type\\\":\\\"execute_request\\\",  
\\\"msg_id\\\":\\\"ef01a40848cb41f6d21f0f4c669e0b28\\\",\\\"session\\\":  
\\\"2e9f696e12704087d895f3474376ddc5\\\",\\\"date\\\":\\\"2020-12-  
18T17:48:33.140050Z\\\"}},\\\"msg_type\\\":\\\"stream\\\",\\\"msg_id\\\":  
\\\"acfb1125-4415c683d933bf1ff6d8a54f\\\",\\\"content\\\":{\\\"text\\\":
```

```

\\ "On non-Windows OS :\\\\\\\\\\\\\\\\nAssuming the packets to be IP packets
(which is usually that, as we are sniffing packets only from the
internet for this packet sniffer)\\\\\\\\\\\\\\\\n\\\\\\\\\\\\\\\\n\\\\", \\ "name\\":
\\ "stdout\\", \\ "header\\": {\\ "username\\": \\ "username\\",
\\ "version\\": \\ "5.3\\", \\ "msg_type\\": \\ "stream\\", \\ "msg_id\\":
\\ "acfb1125-4415c683d933bf1ff6d8a54f\\", \\ "session\\": \\ "18665022-
0e36cbcbba37f542def4b568\\", \\ "date\\": \\ "2020-12-
18T17:48:33.542940Z\\", \\ "channel\\": \\ "iopub\\", \\ "buffers\\": [],
\\ "metadata\\": {}}\\'\\\\n\\\\n\\\\n\\\\nPacket no : 6\\\\n----- Ethernet
Header -----\\\\nDestination Mac : 30:30:30:30:30:30 | Source
Mac : 30:30:30:30:30:30 | Protocol : 2048 | \\\\n----- IP
Header -----\\\\nVersion : 69 | Tos : 0 | Total Length :
52 | Identification : 51501 | Fragment : 16384 | TTL : 64 |
Protocol : 6 | Header CheckSum : 29588 | Source Address :
127.0.0.1 | Destination Address : 127.0.0.1 | \\\\n----- TCP
Header -----\\\\nSource Port : 52996 | Destination Port :
33633 | Sequence Number : 2458009640 | Acknowledge Number :
3829232068 | Offset & Reserved : 128 | Tcp Flag : 16 | Window :
512 | CheckSum : 65064 | Urgent Pointer : 0 | \\\\n\\\\nPayload
(from Application Layer: not extracted or formatted)
:\\nb'\\\\\\\\x01\\\\\\\\x01\\\\\\\\x08\\\\\\\\nv\\\\\\\\xeeO\\\\\\\\xfbv\\\\\\\\xeeO\\\\\\\\xd3'\\\\n\\
\\\\n\\\\n\\\\nPacket no : 7\\\\n----- Ethernet Header -----
\\\\nDestination Mac : 30:32:34:32:61:63 | Source Mac :
30:32:34:32:62:33 | Protocol : 2048 | \\\\n----- IP Header ---
-----\\\\nVersion : 69 | Tos : 0 | Total Length : 620 |
Identification : 42092 | Fragment : 16384 | TTL : 64 |
Protocol : 6 | Header CheckSum : 15332 | Source Address :
172.28.0.1 | Destination Address : 172.28.0.2 | \\\\n----- TCP
Header -----\\\\nSource Port'

```

```

Packet no : 10
----- Ethernet Header -----
Destination Mac : 30:30:30:30:30:30 | Source Mac :
30:30:30:30:30:30 | Protocol : 2048 |
----- IP Header -----
Version : 69 | Tos : 0 | Total Length : 52 | Identification :
50466 | Fragment : 16384 | TTL : 64 | Protocol : 6 | Header
CheckSum : 30623 | Source Address : 127.0.0.1 | Destination
Address : 127.0.0.1 |
----- TCP Header -----
Source Port : 52978 | Destination Port : 33633 | Sequence Number
: 3008018757 | Acknowledge Number : 2526284375 | Offset & Reserved
: 128 | Tcp Flag : 16 | Window : 478 | CheckSum : 65064 |
Urgent Pointer : 0 |

Payload (from Application Layer: not extracted or formatted) :
b'\\\\x01\\\\x01\\\\x08\\\\nv\\\\xeeP\\\\x9cv\\\\xeeP\\\\x9c'

```

```

Packet no : 11
----- Ethernet Header -----
Destination Mac : 30:30:30:30:30:30 | Source Mac :
30:30:30:30:30:30 | Protocol : 2048 |
----- IP Header -----
Version : 69 | Tos : 0 | Total Length : 1855 | Identification
: 5411 | Fragment : 16384 | TTL : 64 | Protocol : 6 | Header
CheckSum : 8340 | Source Address : 127.0.0.1 | Destination
Address : 127.0.0.1 |
----- TCP Header -----

```





```

:\nb\'\x01\x01\x08\nv\xee0\xd3v\xee0\xd3'\n\
\n\n\nPacket no : 3\n----- Ethernet Header -----
\nDestination Mac : 30:30:30:30:30:30 | Source Mac :
30:30:30:30:30:30 | Protocol : 2048 | \n----- IP Header ---
-----\nVersion : 69 | Tos : 0 | Total Length : 659 |
Identification : 9619 | Fragment : 16384 | TTL : 64 | Protocol
: 6 | Header CheckSum : 5328 | Source Address : 127.0.0.1 |
Destination Address : 127.0.0.1 | \n----- TCP Header -----
----\nSource Port : 33633 | Destination Port : 52996 | Sequence
Number : 3829231461 | Acknowledge Number : 2458009640 | Offset &
Reserved : 128 | Tcp Flag : 24 | Window : 512 | CheckSum :
136 | Urgent Pointer : 0 | \n\nPayload (from Application Layer:
not extracted or formatted)
:\nb\'\x01\x01\x08\nv\xee0\xd3v\xeeNc\x01\r
stream.stdout\x01\t<IDS|MSG>\x01\x00\x01\xbb{\n"versi
on\":"5.3",\n"date\":"2020-12-
18T17:48:33.542940Z",\n"session\":"18665022-
0e36cbcbba37f542def4b568",\n"username\":"username",\n"msg_type\
":\n"stream",\n"msg_id\":"acfb1125-
4415c683d933bf1ff6d8a54f"}\x01\xc2{\n"username\":"username\
",\n"msg_type\":"execute_request",\n"msg_id\":"ef01a40848cb41f6
d21f0f4c669e0b28",\n"version\":"5.0",\n"session\":"2e9f696e12
704087d895f3474376ddc5",\n"date\":"2020-12-
18T17:48:33.140050Z"}\x01\x02{\x00\xbc{\n"name\":"std
out",\n"text\":"On non-Windows OS :\\\\\\\\\nAssuming the packets
to be IP packets (which is usually that, as we are sniffing packets
only from the internet for this packet
sniffer)\\\\\\\\\n\\\\\\\\\n"}'\n\n\n\n\n\nPacket no : 4\n-----
- Ethernet Header ----- \nDestination Mac : 30:32:34:32:61:63
| Source Mac : 30:32:34:32:61:63 | Protocol : 2048 | \n-----
--- IP Header ----- \nVersion : 69 | Tos : 0 | Total
Length : 52 | Identification : 2785 | Fragment : 16384 | TTL
: 64 | Protocol : 6 | Header CheckSum : 55205 | Source Address
: 172.28.0.3 | Destination Address : 172.28.0.2 | \n-----
TCP Header ----- \nSource Port : 55944 | Destination Port :
9000 | Sequence Number : 2530555647 | Acknowledge Number :
3177701435 | Offset & Reserved : 128 | Tcp Flag : 16 | Window :
501 | CheckSum : 22628 | Urgent Pointer : 0 | \n\nPayload
(from Application Layer: not extracted or formatted)
:\nb\'\x01\x01\x08\nv\xfa[K<\xa9\xbfW\x14'\n\
\n\n\n\n\nPacket no : 5\n----- Ethernet Header -----
\nDestination Mac : 30:32:34:32:61:63 | Source Mac :
30:32:34:32:61:63 | Protocol : 2048 | \n----- IP Header ---
-----\nVersion : 69 | Tos : 0 | Total Length : 814 |
Identification : 16179 | Fragment : 16384 | TTL : 64 |
Protocol : 6 | Header CheckSum : 41049 | Source Address :
172.28.0.3 | Destination Address : 172.28.0.2 | \n----- TCP
Header ----- \nSource Port : 6000 | Destination Port : 57482
| Sequence Number : 2738570308 | Acknowledge Number : 3233590468
| Offset & Reserved : 128 | Tcp Flag : 24 | Window : 501 |
CheckSum : 23390 | Urgent Pointer : 0 | \n\nPayload (from
Application Layer: not extracted or formatted)
:\nb\'\x01\x01\x08\nv\xfa[K=\xa9\xbfU\xa7\x8
1~\x02\x06{\n"parent_header\": {\n"username\":"username",
\n"version\":"5.0", \n"msg_type\":"execute_request",
\n"msg_id\":"ef01a40848cb41f6d21f0f4c669e0b28", \n"session\":"
\n"2e9f696e12704087d895f3474376ddc5", \n"date\":"2020-12-
18T17:48:33.140050Z"}, \n"msg_type\":"stream", \n"msg_id\":"
\n"acfb1125-4415c683d933bf1ff6d8a54f", \n"content\": {\n"text\":"
\n"On non-Windows OS :\\\\\\\\\nAssuming the packets to be IP packets
(which is usually that, as we are sniffing packets only from the
internet for this packet sniffer)\\\\\\\\\n\\\\\\\\\n", \n"name\":"

```

```

\\"stdout\\"}, \\"header\\": {\\"username\\": \\"username\\",
\\"version\\": \\"5.3\\", \\"msg_type\\": \\"stream\\", \\"msg_id\\":
\\"acfb1125-4415c683d933bf1ff6d8a54f\\", \\"session\\": \\"18665022-
0e36cbcbba37f542def4b568\\", \\"date\\": \\"2020-12-
18T17:48:33.542940Z\\", \\"channel\\": \\"iopub\\", \\"buffers\\": [],
\\"metadata\\": {}|\n\n\n\n\n\nPacket no : 6\n----- Ethernet
Header -----
\nDestination Mac : 30:30:30:30:30:30 | Source
Mac : 30:30:30:30:30:30 | Protocol : 2048 | \n----- IP
Header -----
\nVersion : 69 | Tos : 0 | Total Length :
52 | Identification : 51501 | Fragment : 16384 | TTL : 64 |
Protocol : 6 | Header CheckSum : 29588 | Source Address :
127.0.0.1 | Destination Address : 127.0.0.1 | \n----- TCP
Header -----
\nSource Port : 52996 | Destination Port :
33633 | Sequence Number : 2458009640 | Acknowledge Number :
3829232068 | Offset & Reserved : 128 | Tcp Flag : 16 | Window :
512 | CheckSum : 65064 | Urgent Pointer : 0 | \n\n\nPayload
(from Application Layer: not extracted or formatted)
:\nb'\x01\x01\x08\nv\xee0\xfbv\xee0\xd3'\n\n
\n\n\nPacket no : 7\n----- Ethernet Header -----
\nDestination Mac : 30:32:34:32:61:63 | Source Mac :
30:32:34:32:62:33 | Protocol : 2048 | \n----- IP Header ---
-----
\nVersion : 69 | Tos : 0 | Total Length : 620 |
Identification : 42092 | Fragment : 16384 | TTL : 64 |
Protocol : 6 | Header CheckSum : 15332 | Source Address :
172.28.0.1 | Destination Address : 172.28.0.2 | \n----- TCP
Header -----
\nSource Port'

```

Packet no : 14

```

----- Ethernet Header -----
Destination Mac : 30:30:30:30:30:30 | Source Mac :
30:30:30:30:30:30 | Protocol : 2048 |
----- IP Header -----
Version : 69 | Tos : 0 | Total Length : 52 | Identification :
51502 | Fragment : 16384 | TTL : 64 | Protocol : 6 | Header
CheckSum : 29587 | Source Address : 127.0.0.1 | Destination
Address : 127.0.0.1 |
----- TCP Header -----
Source Port : 52996 | Destination Port : 33633 | Sequence Number
: 2458009640 | Acknowledge Number : 3829240260 | Offset & Reserved
: 128 | Tcp Flag : 16 | Window : 478 | CheckSum : 65064 |
Urgent Pointer : 0 |

```

Payload (from Application Layer: not extracted or formatted) :

```

b'\x01\x01\x08\nv\xeeP\x9cv\xeeP\x9c'

```

Packet no : 15

```

----- Ethernet Header -----
Destination Mac : 30:30:30:30:30:30 | Source Mac :
30:30:30:30:30:30 | Protocol : 2048 |
----- IP Header -----
Version : 69 | Tos : 0 | Total Length : 1855 | Identification
: 9621 | Fragment : 16384 | TTL : 64 | Protocol : 6 | Header
CheckSum : 4130 | Source Address : 127.0.0.1 | Destination
Address : 127.0.0.1 |
----- TCP Header -----
Source Port : 33633 | Destination Port : 52996 | Sequence Number
: 3829240260 | Acknowledge Number : 2458009640 | Offset & Reserved

```



- Protocol – Tells the Network layer at the destination host, to which Protocol this packet belongs to, i.e. the next level Protocol.
- Source Address – 32-bit address of the Sender (or source) of the packet.
- Destination Address – 32-bit address of the Receiver (or destination) of the packet.
- Header Checksum – This field is used to keep checksum value of entire header which is then used to check if the packet is received error-free.
- Source Port - The TCP Source Port is the port number used by the computer sending the TCP segment and is usually a number above 1024.
- Destination Port - The TCP Destination Port is the port number used by the computer receiving the TCP packet and is usually a number below 1024.
- Sequence Number - Used for segmentation of application data into TCP segments and reassembling them on the other side.
- Data Offset - The TCP Data Offset indicates number of bytes into the TCP packet where data can be found.
- Acknowledge number - Used during three-way handshake and data transfers.
- Checksum - This field is used by the receiver to verify the integrity of the data in the TCP payload and rejects data that fails the CRC check.
- Window - Number of octets in the TCP header.
- Urgent Pointer - Points to the end of "urgent" data in the packet, but this field only exists if the URG flag is set.

## CONCLUSION

A packet Sniffer can be installed at any point in a network. It is mostly Software but can be a Hardware also and can play same role in in network troubleshooting. For Hardware it can be plugged directly into network and store or forward any data information. It could also be sneakily installed on a server that acts as a gateway. Thinking it only as a hacker's tool is not true as its very useful for big companies and organisations to check weaknesses in the network system and how intruder can break into it, which helps the companies to make their data on the network more secure and track any unencrypted data if present. A Packet Sniffer helps in limiting malware and tells weakness of a network system. And as everything comes with a cost so as Sniffers so one should always use them under permissible condition with robust security software to prevent their misuse otherwise packet sniffer can capture sensitive personal information that can lead to invasion of privacy. Packet Sniffers are used as spying tool but in such cases network can be protected using:

- Anti-Sniff Tools
- Encryption like Secured Socket Layer
- Analyser can be used to understand network pattern from time to time to detect any malicious intrusion of sniffer in the network



## **FUTURE SCOPE**

As we have seen from the uses of sniffer we can say that future scope of sniffing is bright. No matter how big any Company or any Organisation is or how reputed they are, the only thing which matter them the most is the data security and privacy. Data is an asset for every company, it's a base of companies on which they stand. Security is undoubtedly very important for which companies can spend any amount of money and hence they are investing big for its security. Since sniffing provides a great help to big companies and organisation to provide data information present over the network and help them to know where their security system is lagging and how to overcome it, this will make the sniffer tool and the sniffing technique a huge demand in the future. Not only data security but various organisation also require Sniffers to keep track of specific person's data, or trace any intruder or criminal or maybe an illegal hacker.

## **REFERENCES**

- <https://www.slideshare.net/sgouthamroyal/packet-sniffer-implementation-72190951>
- <https://www.binarytides.com/python-packet-sniffer-code-linux/>
- <http://www.bitforestinfo.com/2017/01/how-to-write-simple-packet-sniffer.html>

JSS Mahavidyapeetha

# JSS Academy of Technical Education

Kengeri - Uttarahalli Main Road, Bangalore-560060



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SUBJECT : INFORMATION AND NETWORK SECURITY

SUBJECT CODE: 15CS743

TOPIC: Digital Signature Generation

Done by:

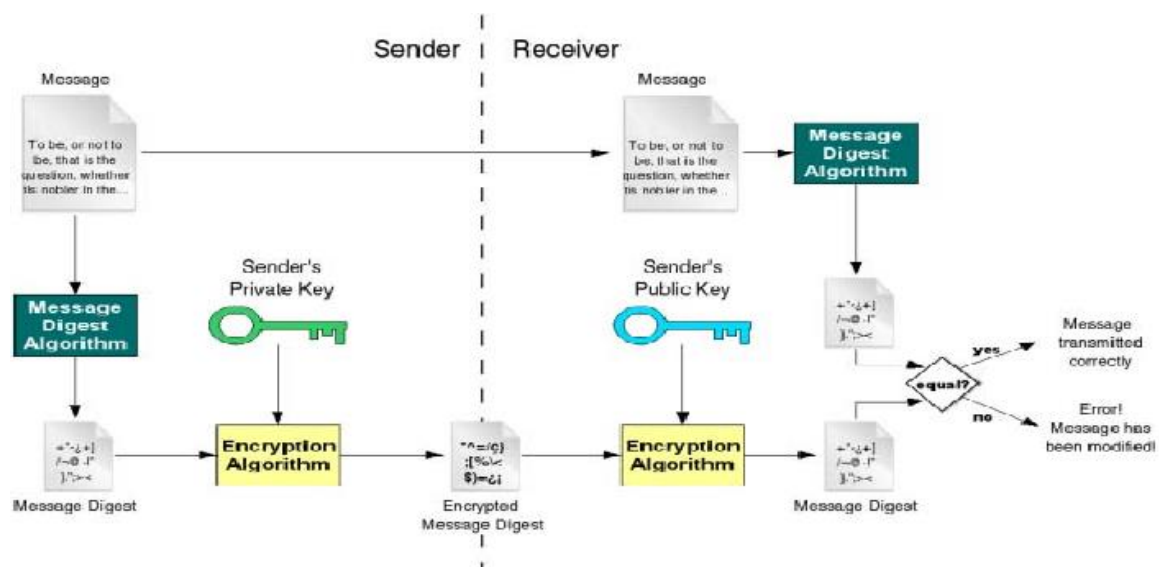
THEJAS KUMAR R  
1JS17CS421

GAGAN GOWDA N  
1JS17CS404

# DIGITAL SIGNATURE GENERATION

A digital signature is a mathematical scheme for verifying the authenticity of digital messages or documents. A valid digital signature, where the prerequisites are satisfied, gives a recipient very strong reason to believe that the message was created by a known sender (authentication), and that the message was not altered in transit (integrity).<sup>[1]</sup>

Digital signatures are a standard element of most cryptographic protocol suites, and are commonly used for software distribution, financial transactions, contract management software, and in other cases where it is important to detect forgery or tampering.



A digital signature scheme typically consists of three algorithms;

- A key generation algorithm that selects a private key uniformly at random from a set of possible private keys. The algorithm outputs the private key and a corresponding public key.
- A signing algorithm that, given a message and a private key, produces a signature.
- A signature verifying algorithm that, given the message, public key and signature, either accepts or rejects the message's claim to authenticity.

The security requirements:

In their foundational paper, Goldwasser, Micali, and Rivest lay out a hierarchy of attack models against digital signatures:<sup>[26]</sup>

1. In a key-only attack, the attacker is only given the public verification key.
2. In a known message attack, the attacker is given valid signatures for a variety of messages known by the attacker but not chosen by the attacker.
3. In an adaptive chosen message attack, the attacker first learns signatures on arbitrary messages of the attacker's choice.

They also describe a hierarchy of attack results:<sup>[26]</sup>

1. A total break results in the recovery of the signing key.
2. A universal forgery attack results in the ability to forge signatures for any message.
3. A selective forgery attack results in a signature on a message of the adversary's choice.
4. An existential forgery merely results in some valid message/signature pair not already known to the adversary.

The strongest notion of security, therefore, is security against existential forgery under an adaptive chosen message attack.

The applications constraints that influence decision making:

### Cryptographic Constraints

While general hash functions are free to transform the input message into the fixed-sized output digest in any manner they see fit, there are a few constraints imposed on cryptographic hash functions used in secure environments. Namely, cryptographic hash functions must have the following properties:

- Deterministic: for a given message, the hash function always results in the same digest.
- Quick: for any given message, computing the digest should be completed in a reasonable time (note that there are cases where it should take an extended period of time to compute the hash, making brute-force trials of all possible messages infeasible, but in the general case, the hash should be computed in a timely manner for a small set of messages).

- Irreversible: given some digest, it is infeasible to compute the message that was used to generate the digest without trying all possible messages.
- Disperse: given two messages, where the second message is generated by changing a small portion of the first message (i.e. adding a single character), the digests of the two messages should not have any perceivable correlation.
- Collision Resistant: it is infeasible to find two messages that hash to the same digest.

The constraints required for a cryptographic hash function provide some realistic bounds on our theoretical hash function. In the general case, we could devise a hash function, albeit impractical, that maps all possible messages to the value 1, thus allowing for an infinite message set. In the case of a cryptographic hash function, the digest must be deterministic and collision resistant. Therefore, when the same message is fed into the hash function, the same digest is always produced, no matter how many times the process is repeated. Furthermore, all digests must be unique in the sense that given two different messages, the same digest must never be produced.

Since our digest length is limited, there is no feasible way to produce a completely collision-resistant cryptographic hash function. Instead, we make the digest length sufficiently long as to make the chance of a collision highly unlikely. For example, if we define our digest length to be 256 bits, we can have  $1.18e^{77}$  possible digests. Thus, assuming a uniform distribution, there is an  $8.64e^{-76}\%$  chance that two messages will result in the same digest.

We must also ensure that given some digest, it is infeasible to compute the original message that created this digest without computing all possible digests. For example, given some digest, it must be infeasible to run the cryptographic hash in reverse and obtain the message that was used to generate the digest. In mathematical terms, this property of cryptographic hash functions makes them a one-way function, where running the function in the forward direction is computationally easy (satisfying our requirement for quick generation) while attempting to reverse the result of the function is computationally hard. In practical terms, a cryptographic hash requires that every possible input message is run through the function until a matching digest is found (since the function is deterministic, the message that caused a matching digest to be generated was the message that maps to the digest of interest).

Lastly, the function must produce digests that are sufficiently dispersed, where a small change in a message results in a largely different digest. For example, if we hashed the contents of a book, adding a period to the end of the book should result in a digest that is wholly unrecognizable from the digest generated from the original contents of the book. Put simply, even the most minuscule changes to a message should result in a large change to the digest.

## Cryptographic primitives that are deployed

Cryptographic primitives are well-established, low-level cryptographic algorithms that are frequently used to build cryptographic protocols for computer security systems. These routines include, but are not limited to, one-way hash functions and encryption functions.

When creating cryptographic systems, designers use cryptographic primitives as their most basic building blocks. Because of this, cryptographic primitives are designed to do one very specific task in a precisely defined and highly reliable fashion.

The cryptographic algorithms that are used in Digital Signature are:

1. The MD5 message-digest algorithm is a widely used hash function producing a 128bit hash value. Although MD5 was initially designed to be used as a cryptographic hash function, it has been found to suffer from extensive vulnerabilities. It can still be used as a checksum to verify data integrity, but only against unintentional corruption. It remains suitable for other non-cryptographic purposes, for example for determining the partition for a particular key in a partitioned database.
2. RSA (Rivest–Shamir–Adleman) is a public-key cryptosystem that is widely used for secure data transmission. It is also one of the oldest. In a public-key cryptosystem, the encryption key is public and distinct from the decryption key, which is kept secret (private). An RSA user creates and publishes a public key based on two large prime numbers, along with an auxiliary value. The prime numbers are kept secret. Messages can be encrypted by anyone, via the public key, but can only be decoded by someone who knows the prime numbers.
3. SHA-1 (Secure Hash Algorithm 1) is a cryptographic hash function which takes an input and produces a 160-bit (20-byte) hash value known as a message digest – typically rendered as a hexadecimal number, 40 digits long. It was designed by the United States National Security Agency, and is a U.S. Federal Information Processing Standard.

## The cryptographic algorithms and key lengths that are Supported

One digital signature scheme (of many) is based on RSA. To create signature keys, generate a RSA key pair containing a modulus,  $N$ , that is the product of two random secret distinct large primes, along with integers,  $e$  and  $d$ , such that  $e d \equiv 1 \pmod{\phi(N)}$ , where  $\phi$  is the Euler phi function. The signer's public key consists of  $N$  and  $e$ , and the signer's secret key contains  $d$ .

To sign a message,  $m$ , the signer computes a signature,  $\sigma$ , such that  $\sigma \equiv m^d \pmod{N}$ . To verify, the receiver checks that  $\sigma^e \equiv m \pmod{N}$ .

Several early signature schemes were of a similar type: they involve the use of a trapdoor permutation, such as the RSA function, or in the case of the Rabin signature scheme, computing square modulo composite,  $N$ . A trapdoor permutation family is a family of permutations, specified by a parameter, that is easy to compute in the forward direction, but is difficult to compute in the reverse direction without already knowing the private key ("trapdoor"). Trapdoor permutations can be used for digital signature schemes, where computing the reverse direction with the secret key is required for signing, and computing the forward direction is used to verify signatures.

Used directly, this type of signature scheme is vulnerable to key-only existential forgery attack. To create a forgery, the attacker picks a random signature  $\sigma$  and uses the verification procedure to determine the message,  $m$ , corresponding to that signature.<sup>[28]</sup> In practice, however, this type of signature is not used directly, but rather, the message to be signed is first hashed to produce a short digest, that is then padded to larger width comparable to  $N$ , then signed with the reverse trapdoor function.<sup>[29]</sup> This forgery attack, then, only produces the padded hash function output that corresponds to  $\sigma$ , but not a message that leads to that value, which does not lead to an attack. In the random oracle model, hash-then-sign (an idealized version of that practice where hash and padding combined have close to  $N$  possible outputs), this form of signature is existentially unforgeable, even against a chosen-plaintext attack.<sup>[19][clarification needed]</sup>

There are several reasons to sign such a hash (or message digest) instead of the whole document.

For efficiency

The signature will be much shorter and thus save time since hashing is generally much faster than signing in practice.

For compatibility

Messages are typically bit strings, but some signature schemes operate on other domains (such as, in the case of RSA, numbers modulo a composite number  $N$ ). A hash function can be used to convert an arbitrary input into the proper format.

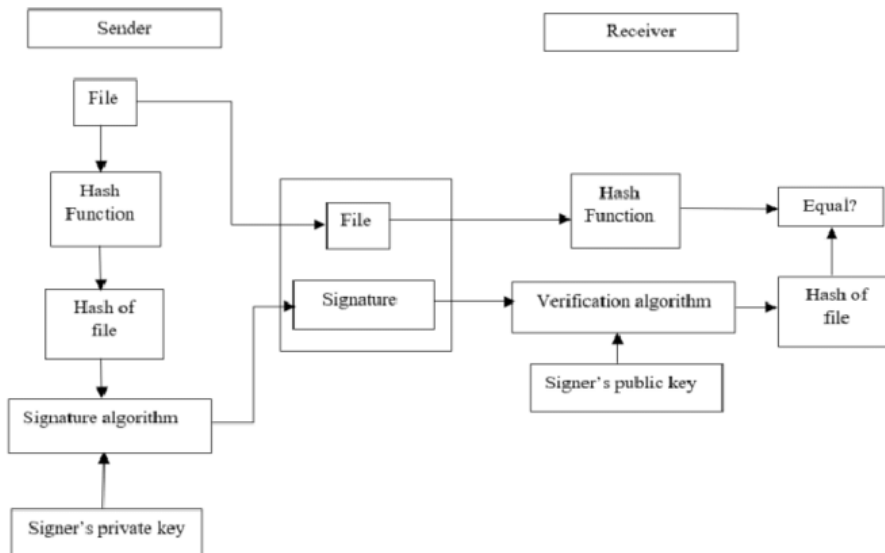
For integrity

Without the hash function, the text "to be signed" may have to be split (separated) in blocks small enough for the signature scheme to act on them directly. However, the receiver of the signed blocks is not able to recognize if all the blocks are present and in the appropriate order.

The key length bits that are supported is 512 bits.

## The key management

As mentioned earlier, the digital signature scheme is based on public key cryptography. The model of digital signature scheme is depicted in the following illustration –



The following points explain the entire process in detail –

- Each person adopting this scheme has a public-private key pair.
- Generally, the key pairs used for encryption/decryption and signing/verifying are different. The private key used for signing is referred to as the signature key and the public key as the verification key.
- Signer feeds data to the hash function and generates hash of data.
- Hash value and signature key are then fed to the signature algorithm which produces the digital signature on given hash. Signature is appended to the data and then both are sent to the verifier.
- Verifier feeds the digital signature and the verification key into the verification algorithm. The verification algorithm gives some value as output.
- Verifier also runs same hash function on received data to generate hash value.
- For verification, this hash value and output of verification algorithm are compared. Based on the comparison result, verifier decides whether the digital signature is valid.
- Since digital signature is created by 'private' key of signer and no one else can have this key; the signer cannot repudiate signing the data in future.

It should be noticed that instead of signing data directly by signing algorithm, usually a hash of data is created. Since the hash of data is a unique representation of data, it is sufficient to

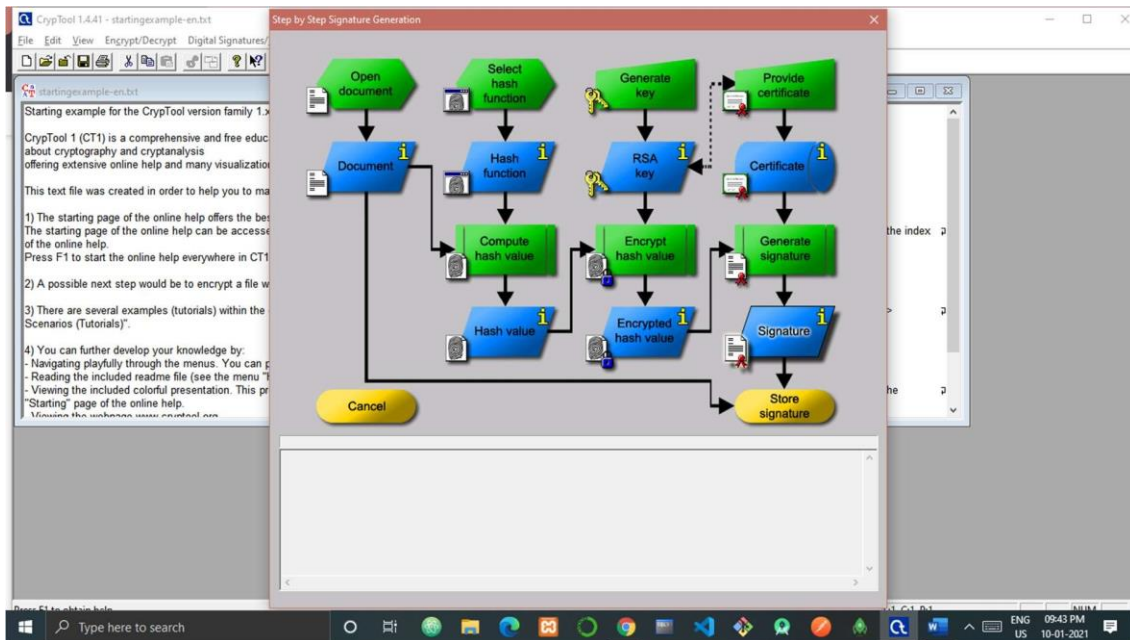


sign the hash in place of data. The most important reason of using hash instead of data directly for signing is efficiency of the scheme.

Let us assume RSA is used as the signing algorithm. As discussed in public key encryption chapter, the encryption/signing process using RSA involves modular exponentiation.

Signing large data through modular exponentiation is computationally expensive and time consuming. The hash of the data is a relatively small digest of the data, hence signing a hash is more efficient than signing the entire data.

### Output Snapshots:



The flowchart illustrates the process: **Open document** (green arrow) → **Select hash function** (green arrow) → **Generate key** (red arrow) → **Provide certificate** (red arrow).

**Generate RSA Key** dialog box:

Choose two prime numbers  $p$  and  $q$ . The number  $N = pq$  is the public RSA modulus and  $\phi(N) = (p-1)(q-1)$  is the Euler phi function. Public key  $e$  is coprime to  $\phi(N)$ . The private key  $d = e^{-1} \pmod{\phi(N)}$  is calculated from this.

Prime number entry:

- Prime number  $p$ : 2423163384625797737617864374887
- Prime number  $q$ : 1434298724673852070184430816748

Generate prime numbers... (button)

$p$  and  $q$  are prime numbers.

RSA parameter:

- Length: 304 bit
- RSA modulus  $N$ : 3475540152245156575672595655541 (public)
- $\phi(N) = (p-1)(q-1)$ : 3475540152245156575672595655541 (secret)
- Public key  $e$ :  $2^{16}+1$  (e does not divide phi (N)).
- Private key  $d$ : 9391918472963627104561037133166

Store key (button)      Cancel (button)

**CrypTool 1.4.41 - RSA (MDS) signature of <startingexample-en.txt>**

File Edit View Encrypt/Decrypt Digital Signatures/PKI Indiv.Procedures Analysis Options Window Help

Document content (hex dump):

```

00000000 63 69 67 6E 61 74 75 72 65 3A 20 20 20 20 20 04 17 62 E4 83 45 7A 95 F2 6A 19 D4 6F 7D
0000001F 0D 12 87 F9 12 5B 67 A5 57 20 95 AC 5E 42 E9 A2 21 B4 96 63 60 DA D9 9D 20 20 20 20 20 20
0000003E 20 20 20 20 20 20 20 20 20 20 53 69 67 6E 61 74 75 72 65 20 6C 65 6E 67 74 68 3A 20 20 33 30 34
0000005D 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 41 6C 67 6F 72 69 74 68 6D 3A 20 20 20 20 20
0000007C 20 52 53 41 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 48 61 73 68 20 66 75 6E 63 74 69
0000009B 6F 6E 3A 20 20 20 20 44 35 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
000000BA 20 20 20 20 5B 63 68 69 72 75 5D 5B 63 68 61 6E 64 61 6E 5D 5B 52 53 41 2D 33 30 34 5D 5B 31
000000D9 36 31 30 33 35 32 35 31 33 5D 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 4D 65 73 73 61
000000F8 67 65 3A 20 20 20 20 20 20 20 20 20 20 53 74 61 72 74 69 6E 67 20 65 78 61 6D 70 6C 65 20 66 6F 72 20 74
00000117 68 65 20 43 72 79 70 54 6F 6F 6C 20 76 65 72 73 69 6F 6E 20 66 61 6D 69 6C 79 20 31 2E 70 20
00000136 28 43 54 31 29 0D 0A 0D 0A 43 72 79 70 54 6F 6F 6C 20 31 20 28 43 54 31 29 20 69 73 20 61 20
00000155 63 6F 6D 70 72 65 68 65 6E 73 69 76 65 20 61 6E 64 20 66 72 65 65 20 65 64 75 63 61 74 69 6F
00000174 6E 61 6C 20 70 72 6F 67 72 61 6D 0D 0A 61 62 6F 75 74 20 63 72 79 70 74 6F 67 72 61 70 68 79
00000193 20 61 6E 64 20 63 72 79 70 74 61 6E 61 6C 79 73 69 73 0D 0A 6F 66 66 65 72 69 6E 67 20 65 78
000001B2 74 65 6E 73 69 76 65 20 6F 6E 6C 69 6E 65 20 68 65 6C 70 20 61 6E 64 20 6D 61 6E 79 20 76 69
000001D1 73 75 61 6C 69 74 61 74 69 6F 6E 73 2E 0D 0A 0D 0A 54 68 69 73 20 74 65 78 74 20 66 69 6C 65
000001F0 20 77 61 73 20 63 72 65 61 74 65 64 20 69 6E 20 6F 72 64 65 72 20 74 6F 20 68 65 6C 70 20 79
0000020F 6F 75 20 74 6F 20 6D 61 6B 65 20 79 6F 75 72 20 66 69 72 73 74 20 73 74 65 70 73 20 67 69 74
0000022E 63 20 43 54 31 2E 0D 0A 0D 0A 31 29 20 54 68 65 20 73 74 61 72 74 69 6E 67 20 70 61 67 65 20
0000024D 6F 66 20 74 68 65 20 6F 6E 6C 69 6E 65 20 68 65 6C 70 20 6F 66 66 65 72 73 20 74 68 65 20 62
0000026C 65 73 74 20 6F 76 65 72 73 69 67 68 74 20 6F 66 20 43 54 31 27 73 20 63 61 70 61 63 69 74 79
0000028B 2E 20 46 72 6F 6D 20 74 68 65 20 73 74 61 72 74 69 6E 67 20 70 61 67 65 20 79 6F 75 20 63 61
000002AA 2E 72 65 61 63 68 20 61 6C 6C 20 65 73 65 6E 74 69 61 6C 20 66 75 6E 63 74 69 6F 6E 73
000002C9 20 76 69 61 20 6C 69 6E 6B 73 2E 0D 0A 54 68 65 20 73 74 61 72 74 69 6E 67 20 70 61 67 65 20
000002E8 6F 66 20 74 68 65 20 6F 6E 6C 69 6E 65 20 6D 68 65 6C 70 20 63 61 6E 20 62 65 20 61 63 63 65 73
00000307 73 65 64 20 76 69 61 20 74 68 65 20 6D 68 65 6C 75 20 22 48 65 6C 70 20 2D 3E 20 53 74 61 61 72 74
00000326 69 6E 67 20 50 61 67 65 22 20 61 74 20 74 68 65 20 74 6F 20 72 69 67 68 74 20 6F 66 20 74
00000345 68 65 20 6D 61 69 6E 20 77 69 6E 64 6F 77 20 6F 72 20 62 79 20 75 73 69 6E 67 20 74 68 65 20
00000364 73 65 61 72 63 68 20 6B 65 79 77 6F 72 64 20 22 53 74 61 72 74 69 6E 67 20 70 61 67 65 22 20
00000383 77 69 74 68 69 6E 20 74 68 65 20 69 6E 64 65 78 20 6F 66 20 74 68 65 20 6F 6E 6C 69 6E 65 20
000003A2 68 65 6C 70 2E 0D 0A 50 73 65 73 20 46 31 20 74 6F 20 73 74 61 72 74 20 74 68 65 20 6F 6E
000003C1 6C 69 6E 65 20 68 65 6C 70 20 65 76 65 72 79 77 68 65 72 65 20 69 6E 20 43 54 31 2E 0D 0A 0D
    
```

Signature details window:

```

Signature: .....b..Ez..j..o)
.....[g.W...B...C...
Signature length: 304
Algorithm:
RSA Hash functi
on: MD5 Key:
[chiru][chandan][RSA-304][1
610352513] Messa
ge: Starting example for t
he CrypTool version family 1.x
(CTI)... CrypTool 1 (CTI) is a
comprehensive and free educatio
nal program about cryptography
and cryptanalysis, offering ex
tensive online help and many vi
sualizations....This text file
was created in order to help y
ou to make your first steps wit
h CTI....(1) The starting page
of the online help offers the b
est oversight of CTI's capacity
. From the starting page you ca
n reach all essential functions
via links...The starting page
of the online help can be acces
sed via the menu "Help -> Start
ing Page" at the top right of t
he main window or by using the
search keyword "Starting page"
within the index of the online
help. Press F1 to start the on
line help everywhere in CTI....
    
```

```

Hash value encrypted with the private key of the signer
Padding string: 01 FF 00
Algorithm ID: 30 20 30 0C 06 08 2A 86 48 86 F7 0D 02 05 05 00 04 10
Hash value: 66 10 7C 7A 5B 19 71 A0 6D 12 43 F8 26 E1 A0 45
ASN-1 hash value: 01 FF 00 30 20 30 0C 06 08 2A 86 48 86 F7 0D 02 05 05 00 04 10
Length in bits: 296
Encrypted hash value: 04 17 62 E4 83 45 7A 95 F2 6A 19 D4 6F 7D 0D 12 87 F9 12 5B 67
Length in bits: 304

```

**CrypTool 1.441 - RSA (MDS) signature of <abc.txt>**

Step by Step Signature Generation

Open document → Select hash function → Generate key → Provide certificate → Generate signature → Store signature

**Congratulations!**  
 You have successfully created an RSA signature of the document abc.txt.  
 For this purpose you have generated a new RSA key and stored it together with a certificate in the PSE file [path].  
 As the hash function you have selected MDS.

MDS signature of abc.txt

```

00000 53 69 67 6E 61 74 75 72 65 3A 20 20 20 20 20 20 09
00012 24 29 B2 61 C6 D1 3A 9C 9A 64 19 5E EF C2 E5 A9 21 7C
00024 45 55 AD DE 8B DD C6 E9 8C E9 0C 1D 87 BF 6B E3 98 10
00036 3E 20 20 20 20 20 20 20 20 20 20 20 20 20 20 53
00048 69 67 6E 61 74 75 72 65 20 6C 65 8E 67 74 68 3A 20 20
0005A 33 30 34 20 20 20 20 20 20 20 20 20 20 20 20 20
0006C 20 41 6C 67 6E 72 69 74 68 6D 3A 20 20 20 20 20 52
0007E 53 41 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00090 48 61 73 68 20 66 75 6E 63 74 69 6F 6E 3A 20 20 20 4D
000A2 44 35 20 20 20 20 20 20 20 20 20 20 20 20 20 20 D5
000B4 4B 65 79 3A 20 20 20 20 20 20 20 20 20 20 20 20 20 20

```

Signature: [hex data]  
 Algorithm: RSA  
 Hash function: MDS  
 Key: [hex data]

**PuTTY Key Generator**

File Key Conversions Help

**Key**

Public key for pasting into OpenSSH authorized\_keys file:

```
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAIBtWH/RU5kLRYnO/y2S9Fi1dXhRB2oqxqTZV3i0SXlcrWk/2TH9D+//Gn9xHNcE//7xnhZNYvd9Yxu0QQ+WmJ2GK94mtMmd45auyYhy2UtItPIKw2Mq8dHoAAyR7X4P7PaCQEMtdWv88CPes9MLUwqzSkBLYFIXHUtrzLShUl/nU7w== rsa-key-20061213
```

Key fingerprint: ssh-rsa 1023 0e:38:42:25:cd:cc:94:d4:fa:04:32:ee:60:99:55:9c

Key comment: rsa-key-20061213

Key passphrase: [masked]

Confirm passphrase: [masked]

**Actions**

Generate a public/private key pair Generate

Load an existing private key file Load

Save the generated key Save public key Save private key

**Parameters**

Type of key to generate:  
 SSH-1 (RSA)  SSH-2 RSA  SSH-2 DSA

Number of bits in a generated key: 1024



# JSS Academy of Technical Education, Bangalore-60

## Department of Computer Science & Engineering

### Survey on Applications of data structures in real world

**Announcement Date:** 18/01/2021

**Submission Date:** 01/02/2021

**Semester:** 3<sup>rd</sup> CSE 'A'

**Course Name:** Data Structures and Applications (18CS32)

#### Course Outcomes (Cos):

| COs    | Description   | RBT Levels |
|--------|---|------------|
| C202.1 | Utilize the fundamental concepts of data structures and their applications for problem solving. | L3         |
| C203.2 | Choose appropriate linear data structures to solve given problem.                               | L3         |
| C203.3 | Make use of non linear data structures for problem solving.                                     | L3         |
| C203.4 | Apply sorting, searching and hashing techniques for processing data.                            | L3         |

#### Assignment Rules:

- ➔ Specify the real world application name for the assigned data structure.
- ➔ A survey report on application.
- ➔ Duplication of application will be rejected. (Application priority will be FCFS)
- ➔ **The following is a suggested guideline for preparing your reports:**
  - 1) Key Issues/Goals/Problems
  - 2) Decision Criteria & Assumptions.
  - 3) Context.
  - 4) Program implementation in C.
  - 5) Results/Outcomes.

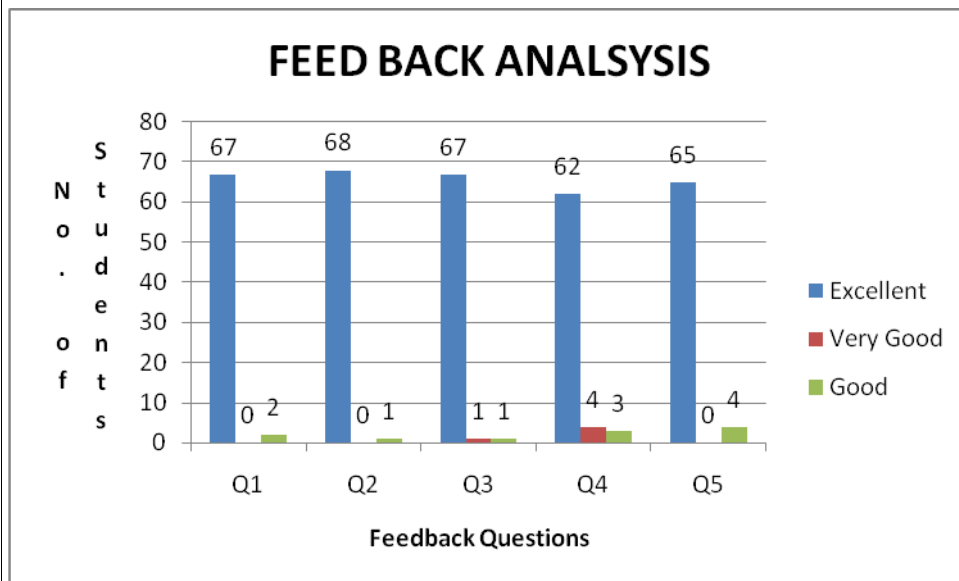
#### Rubrics for the evaluation :

- 1) Presentation(power point presentation) – 5 Marks
- 2) Submission Report - 5 Marks

Prepare the detail report on application about 7 to 8 pages including front page and cover page.  
(soft copy to be uploaded to google classroom and hard copy to be submitted later)

## Feedback Analysis

- Q1. Knowledge and information gained from the Innovative teaching method.  
Q2. Have you gain the knowledge of usage of data structures in real world.  
Q3. Understanding the concept.  
Q4. The Method was useful in enhancing the learning capabilities of the subject.  
Q5. Overall rating of the method.



## Outcome of the activity:

- Understand the concept of Dynamic memory management, data types.
- Understand basic data structures such as arrays, linked lists, stacks and queues.
- Describe the hash function and concepts of collision and its resolution methods.
- Solve problem involving graphs, trees and heaps.
- Apply Algorithm for solving problem using C.

**Faculty In charge**

Mrs. K S Rajeshwari

### Allotment of Data structures to each team

| Group number | SI No. | NAME              | USN        | Data Structures      |
|--------------|--------|-------------------|------------|----------------------|
| 1            | 1      | deepak            | 1js19cs049 | Hashing              |
|              | 2      | ankit             | 1js19cs027 |                      |
|              | 3      | chirag            | 1js19cs046 |                      |
| 2            | 1      | Adithya.S         | 1JS19CS010 | File organization    |
|              | 2      | Abhay TM          | 1JS19CS003 |                      |
|              | 3      | Ashutosh Wodeyar  | 1JS19CS036 |                      |
|              | 4      | Girish Kumar DV   | 1JS19CS057 |                      |
| 3            | 1      | Harsh Bhaskar     | 1JS19CS064 | Singly Linked List   |
|              | 2      | Aryan Rai         | 1JS19CS034 |                      |
|              | 3      | Abhijeet kumar    | 1JS19CS004 |                      |
|              | 4      | Anand Rai         | 1JS19CS024 |                      |
| 4            | 1      | Chiranjeevi R     | 1JS19CS047 | File organization    |
|              | 2      | Dhanush Kumar K   | 1JS19CS051 |                      |
|              | 3      | Gagana Chandana M | 1JS19CS055 |                      |
|              | 4      | Girish S N        | 1JS19CS058 |                      |
| 5            | 1      | Aishwarya MB      | 1JS19CS013 | Stack                |
|              | 2      | Sahana TE         | 1JS18CS134 |                      |
|              | 3      | Ayushi B          | 1JS18CS029 |                      |
| 6            | 1      | Amulya.K          | 1JS19CS022 | Strings              |
|              | 2      | Harika.A          | 1JS19CS063 |                      |
| 7            | 3      | Disha Ramesh      | 1JS19CS052 | Hashing              |
|              | 1      | Anushka. Jha      | 1JS19CS030 |                      |
|              | 2      | Apoorva.v         | 1JS19CS031 |                      |
|              | 3      | Ashutosh Mishra   | 1JS19CS035 |                      |
| 8            | 1      | Aishwarya BT      | 1JS19CS011 | Doubly Linked List   |
|              | 2      | Ammineni Mayukha  | 1JS19CS021 |                      |
|              | 3      | Ginni Singh       | 1JS19CS056 |                      |
| 9            | 1      | Dhananjay A Patel | 1JS19CS050 | Stack                |
|              | 2      | Abhishek S P      | 1JS19CS008 |                      |
|              | 3      | Ajay Biradar      | 1JS19CS014 |                      |
| 10           | 1      | Aishwarya KS      | 1JS19CS012 | Circular Linked List |
|              | 2      | Anshu Upadhyay    | 1JS19CS028 |                      |
|              | 3      | Bhoomika P        | 1JS19CS042 |                      |
|              | 4      | Hari Chandana P   | 1JS19CS062 |                      |
| 11           | 1      | Aneesh K.         | 1JS17CS013 | Queue                |
| 12           | 1      | Akash Rao M B     | 1JS19CS018 | Binary Search Trees  |
|              | 2      | Amit Kadekar      | 1JS19CS020 |                      |
| 13           | 1      | Anaghashree Nanda | 1JS19CS023 | Graphs               |
|              | 2      | Ananya G          | 1JS19CS025 |                      |
|              | 3      | Deeksha S         | 1JS19CS048 |                      |
|              | 4      | Aayushi Singh     | 1JS19CS001 |                      |
| 14           | 1      | Gagan Karanth N   | 1JS19CS054 | BFS                  |
|              | 2      | Bhargav M         | 1JS19CS041 |                      |
|              | 3      | Atul C Anil       | 1JS19CS039 |                      |

|    |   |                          |            |                                    |
|----|---|--------------------------|------------|------------------------------------|
|    | 4 | Chandan Kumar R          | 1JS19CS045 |                                    |
| 15 | 1 | Abdullah Nihad           | 1JS19CS002 | DFS                                |
|    | 2 | Akshay Prakash           | 1JS19CS019 |                                    |
|    | 3 | Anupam Ashok             | 1JS19CS029 |                                    |
|    | 4 | Arjun BR                 | 1JS19CS033 |                                    |
| 16 | 1 | C Mukund reddy           | 1JS19CS043 | Trees                              |
|    | 2 | Chaithanya kumar H D     | 1JS19CS044 |                                    |
|    | 3 | Akash H                  | 1JS19CS017 |                                    |
| 17 | 1 | Akanksha V G             | 1JS19CS016 | Sorting types :<br>Radix sort      |
|    | 2 | Ashwin                   | 1JS19CS037 |                                    |
|    | 3 | Abhijna B C              | 1JS19CS005 |                                    |
| 18 | 4 | Ankit Kumar<br>Upadhyay  | 1JS19CS026 | Searching types :<br>Binary search |
|    | 1 | Sushma S<br>Kalasannavar | 1JS19CS192 |                                    |
|    | 2 | Vibhuti Bajaj            | 1JS19CS191 |                                    |
| 19 | 3 | Kumar R                  | 1JS18CS068 | Binary trees                       |
|    | 1 | Adarsh Narayan           | 1JS19CS009 |                                    |
|    | 2 | Ayush Kumar              | 1JS19CS040 |                                    |
|    | 3 | Arham Jain               | 1JS19CS032 |                                    |
| 20 | 4 | Asif Nawaz               | 1JS19CS038 | Queue                              |
|    | 1 | Hamsashree Srinivas      | 1JS19CS060 |                                    |
|    | 2 | H Vamshi                 | 1JS19CS059 |                                    |
|    | 3 | Ajay Singh Raju          | 1JS19CS015 |                                    |
|    | 4 | Hareesh N                | 1JS19CS061 |                                    |

J S S ACADEMY OF TECHNICAL EDUCATION,  
BENGALURU



**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**Data Structures and Applications (18CS32)**

**Collaborative Learning Activity**

An Activity Report On

**Air Traffic Controller Simulator: Application of File  
Organization Principles in C**

Submitted by

|                   |            |
|-------------------|------------|
| Chiranjeevi R     | 1JS19CS047 |
| Dhanush Kumar K   | 1JS19CS051 |
| Gagana Chandana M | 1JS19CS055 |
| Girish S N        | 1JS19CS058 |

Under the guidance of  
**Mrs. K S Rajeshwari**  
Assistant Professor,  
Department of Computer Science, JSSATE



## **ABSTRACT**

This activity report bears the details of a simulator running on the functional guidelines of an ATC (Air Traffic Controller). Air Traffic Controller is an essential body of Air Transportation and Management, that is responsible for maintaining functional air routes across the controlled airspace of the organization or a country for that matter be, or communicating and aiding the aircrafts in other controlled airspaces, regulating air traffic, maintaining ground clearance – mainly maneuvering the aircraft across the runways, escorting passengers and goods to the baggage claim areas of the airport etc.

Our code mainly focuses on a significant issue – Traffic. These days the airfare is quite reasonable and people travel around the world in considerable numbers and frequencies. As they rightfully say the world today is a global village. As and when cities grow and flourish, they are connected to the world through air. Resource constraints are imposed ever effortlessly, so means to optimize the case with what is available is the call of the hour. Moreover, despite of the terrestrial resource of capital, an arial resource is sufficiently needed. The air routes are growing out of bounds and every time a new one is needed, it shall surpass the existing and intertwining networks, mastering the time management techniques. In simple terms be it the deficiency of land or airspace an alternative is needed to maintain and efficiently utilize the facility of air transport. That is where we think of Traffic Control over the distributed resources. Quite the task our software is attempting to commit.

Apart from that the code focuses on File handling principles. A simulator is close to nothing if it can't retain the details of every simulation ever, as the core ideology of a simulation is to acquaint, prevent or act. The idea is to create a log file that will make an entry of simulation start time, simulation end time along with a summary of the simulation. The summary will be notably the statistics of the execution.

## ACKNOWLEDGEMENT

First of all, we would like to render our sincere gratitude to our beloved Principal, **Prof. Mrithyunjaya V Latte** sir, for bestowing an opportunity to undertake software development activities, for the students of JSSATE and also extending all possible aid during the process. Would really feel contented if we could do something in return to the college.

Then we would like to express our gratitude to our HOD, **Dr. Naveen N C** sir, for the support and ease of approach, that is available to all the students and the Department of Computer Science as a whole, creating a community for likeminded software developers like us and our guiding professors. Looking forward for some great projects ahead.

Most important of all we would like to thank our lovely guide and Assistant Professor, **Mrs. K S Rajeshwari** ma'am for her dedicated guidance in making this project a reality, every step of the way. She always inspired us with competitive attitude and led us to opt this challenging project at this stage of learning. Would like to venture into more challenging avenues in your guidance ma'am, with best of our efforts.

We would also like to appreciate the role of other professors along the way, as we admit the fact that software development is a team game and the adequate knowledge across the subjects is equally important to us.

Finally, I would like to thank my teammates for working together as a team and accomplishing this task amidst all the odds. Really grateful for being able to do this together.

## TABLE OF CONTENTS

| Sl.No. | Content                | Page No. |
|--------|------------------------|----------|
| 1.     | Introduction           | 1        |
| 2.     | Literary Survey        | 2        |
| 2.1    | ATC: Application       | 2-3      |
| 2.2    | Simulation             | 3-4      |
| 2.3    | File Organization      | 4        |
| 3.     | Program Implementation | 5-14     |
| 4.     | Results                | 15-18    |
| 5.     | Conclusion             | 19-20    |
| 6.     | Reference              | 21       |

## 1. INTRODUCTION

This software is basically about simulating the process of Traffic Control managed by the personnel at an ATC tower aided with RADAR, visual and other means of electronically enabled tools of mapping and projections in regulating air traffic and also within it the responsibility to protect all the pilots, in-flight staff and the passengers included. This software plays a dual role in helping run possibilities to learn from and also to avoid adversities beforehand. The functionality remains the same though the situation of employing the use of simulation certainly changes.

The software that we have developed utilizes a basic logical function for random selection and decision-making elements of the simulator. A simulator indeed works on deterministic algorithms but the results are expected to change subject to other contributing factors of the system, that are random in the scheme of variation. We have incorporated the randomness in the arrival of different aircrafts at different times, but the ways of handling the traffic are quite the same. The traffic will be transferred across the runways whenever a given runway's limit is reached.

There are inbuilt changes in this field that need expertise not only in computer simulation but also in the very field of electronics, aviation, communication and other related technologies. We have picked the computer simulation part of this entire process and it is well implied that a system of the above technological domains is requisite for a functional simulator that can be a real-life software undoubtedly.

To mimic the operation of communication and transmission of signals between the ground base of operation and the aircraft we utilize a randomly generated input sequence of planes that are requesting to land or takeoff and appropriately clear after they have left. We also assume that the operator is in uncompromisable visual communication with the aircraft in the visual human range and for the rest of the trajectory the aircraft is on the RADAR. If things are not desirably perfect then they are accounted with time delays induced in the system to speak of. At least that is where we can start from in the process of ideating the creation of a simulator.

For simplicity we will just let the normal execution time be the standard.

## 2. LITERARY SURVEY

### 2.1. ATC: APPLICATION

Air traffic control is a necessary responsibility in the ever-growing field of aircraft travel. Conducted by ground-based personnel, who have been trained and certified by the respective national authority, air traffic control responsibilities center on the monitoring of air traffic in a given area, specifically tower control, approach and departure control, and enroute control. Air traffic controller specialists are the trained personnel who guide pilots, their planes and numerous daily passengers from taxi to takeoff, through the air and back safely on the ground.

Air traffic control is a large human-machine system. The future may bring intelligent knowledge-based systems, expert systems, very accurate satellite-derived navigational information, automated speech synthesis and speech recognition, touch-sensitive input devices, glorious color—a whole panoply of technological advances which can now be foreseen, if not implemented. Many of these innovations imply increased automation of functions or more computer assistance for controllers. These changes, if well-chosen and sensibly introduced, can produce many benefits, but it is important to be aware of some of their human factors consequences which may not be so welcome, the prevention of which should influence the particular forms which automation takes. Many of the prospective benefits of progressive automation may prove to be elusive unless all its human factors implications are anticipated and allowed for.

The role of an air traffic controller is quite complex. They're the person who works from the control tower, giving clearance for aircraft to take off and land safely in the airport. The air traffic controller works within a system, coordinating patterns to ensure aircraft keep a safe distance in the air and on the ground. The main goal of an air traffic controller is to ensure the safety of aircraft, pilots, flight attendants, and of course, the airline passengers.

An air traffic controller will communicate with pilots throughout their entire flight, relaying information back and forth between incoming and outgoing flights. They use a large system of computers, radars, and visual references during this communication. They must provide pilots with an in-depth explanation about the weather and should be prepared for any necessary flight path changes.

There are seven phases in air traffic control: preflight, takeoff, departure, in the air, descent, approach, and landing. Preflight is when the weather forecast is communicated from the air traffic control tower to the pilot and clearance is provided for the flight's route. Takeoff is when the tower gives the airplane clearance to lift off the ground. Departure occurs when the plane is five miles beyond the airport and flight control is transferred to a Terminal Radar. In the air describes when the oversight is given to an Air Route Traffic Control Center (ARTCC), which is a radar system supervising flights within the area. Descent is when the plane is within 50 miles of its destination airport. During the approach, the TRACON controller fuses several streams of descending airplanes into one even pace. Finally, landing is when the local controller gives clearance for landing and directs pilots through taxiways.

Our code is mainly based on the fact that radar and visual communications are already set in place. We only deal with the software implications of the entire procedure.

## 2.2. SIMULATION:

A simulation is an approximate imitation of the operation of a process or system over time. Simulation is used in many contexts, such as simulation of technology for performance tuning or optimizing, safety engineering, testing, training, education, and video games. Often, computer experiments are used to study simulation models. Simulation is also used with scientific modelling of natural systems or human systems to gain insight into their functioning, as in economics. Simulation can be used to show the eventual real effects of alternative conditions and courses of action. It is also used when the real system cannot be engaged, because it may not be accessible, or it may be dangerous or unacceptable to engage, or it is being designed but not yet built, or it may simply not exist. The code that we have developed indeed mimics the course of action needed to be taken by an ATC personnel in real life emergencies. Of course such a

process is not expected to be experienced and then learnt from, it is undoubtedly dangerous and undesirable.

Key issues in simulation include the acquisition of valid sources of information about the relevant selection of key characteristics and behaviors, the use of simplifying approximations and assumptions within the simulation, and fidelity and validity of the simulation outcomes. As previously noted, a course of action needs to be reconstructed but, virtually - with the opportunity to predict, learn and overcome.

Procedures and protocols for model verification and validation are an ongoing field of academic study, refinement, research and development in simulations technology or practice, particularly in the work of computer simulation. Our code is mainly aimed at producing a reality on your desktop that puts your skills to the test, to solve and handle a unprecedented scenario. All of this as a part of the training to become an ATC personnel or to just utilize the software directly. Though it is widely debated that a simulation cannot beat human intuition.

### 2.3.FILE ORGANIZATION:

File organization refers to the way data is stored in a file. File organization is very important because it determines the methods of access, efficiency, flexibility and storage devices to use.

Our code utilizes the aspect of storing data on a file, retrieving from a file and also transferring data back and forth, across the code execution files and the simulation log file. We utilize pointers of the datatype “file” to read and write through the files. Functions like fscanf, fprintf, fgetc, fopen, fclose are used as a part of this file organisation technique. The files by default are text based so all digital content except images and graphics are allowed in it.

The idea of files help culminating the data of simulation as an organized entity and enables a better access. We have included the login and logout time of the simulation along with a brief summary of the entire process.

### 3. PROGRAM IMPLEMENTATION

#### PROGRAM STATEMENT

Air Traffic Control: Managing an airport with three runways one always restricted for landing, one always restricted for takeoff and the other one depending on the priority and queue status. High priority for landing and low priority for takeoff.

The code makes use of the concept of Singly Linked List to realize a queue that holds the number of requests that can be processed in a single runway at a given amount of time preferably before the queue overflow (dependent on the system processing time). The same works for the runway 1 and 2 irrespective of their priorities. If in case any of these queues overflow that content shall be printed or in terms of the application the request is serviced. A random number generator is utilized for the situation of providing input to the simulator. Finally, the entire process is briefly tabulated in the file as a part of simulation logging for future interests.

#### SOURCE CODE

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include "random.h"
#include "cur_time.h"
#include "read_this_file.h"
#define MAX 3
void simulation_log();
void summary();
void insert_front(int);
int delete_end();
int count_nodes();
int out,del,runw1[100],runw3[100],req,run1,run3,units,ch;
```



```
int takeoff[100],dely[100];
FILE *fp;
char fname[20];
char c;
typedef struct node
{
    int air_number;
    struct node *link;
}NODE;
NODE *start=NULL;
int main()
{
    int num,i,n,found,num1,j,k,d,t;
    j=k=d=t=0;
    printf("\n*****\n");
    printf("\n\t\tAIR TRAFIC CONTROLER\n");
    printf("\n*****\n");
    printf("\nEnter the number of units : ");
    scanf("%d",&units);
    printf("\n Enter the name of the file that will store the simulation log for you:");
    scanf("%s",fname);
    fp=fopen(fname,"a");
    fprintf(fp,"\nSimulation started at:\n");
    fclose(fp);
    currenttime(fname);
    fp=fopen(fname,"a");
    fprintf(fp,"\n");
```

```
fclose(fp);
    for(i=0;i<units;i++)
    {
printf("%d : \n",i+1);
        num=random()%(999-100+1)+100;
        num1=random()%(999-100+1)+100;
printf("\n\tPlan %d requested for landing\n",num);
insert_front(num);
printf("\n\tPlan %d requested for landing\n",num1);
insert_front(num1);
        if(count_nodes(<MAX)
        {
printf("\n\tPlan %d landed on runway 1\n",num);
                runw1[j++]=num;
                run1++;
                n=delete_end();
printf("\n\tPlan %d take off on runway 2\n",n);
                takeoff[t++]=n;
printf("\n\tplan %d landed on runway 1\n",num1);
                run1++;
                runw1[j++]=num1;
        }
        else
        {
printf("\n\tPlan %d landed on runway 1\n",num1);
                runw1[j++]=num1;
                run1++;
        }
    }
}
```

```
printf("\n\tPlan %d delayed for landing\n",num);
dely[d++]=num;
    del++;
insert_front(num);
printf("\n\tPlan %d landed on runway 3\n",num);
    runw3[k++]=num;
    run3++;
    n=delete_end();
printf("\n\tPlan %d take off in runway 2\n",n);
    takeoff[t++]=n;
    found=random()%(1000-0+1)+0;
    if(found%2==0)
    {
        n=delete_end();
printf("\n\tPlan %d take off in runway 3\n",n);
        takeoff[t++]=n;
    }
}
}
printf("\n\tWould you like to see the summary???\n");
printf("\n\tEnter 1 for YES 0 for NO : ");
scanf("%d",&ch);
    if(ch==1)
    {
simulation_log();
printf("\n You can see the contents of this file now\n");
printf(file(fname));
```

```
    }  
    else  
    {  
fp=fopen(fname,"a");  
fprintf(fp,"\nSimulation ended at:\n");  
fclose(fp);  
currenttime(fname);  
printf("\n You did not choose to open the file but it is updated");  
exit(0);  
    }  
}  
  
void insert_front(int num)  
{  
    NODE *newnode,*curptr=NULL;  
newnode=(NODE*)malloc(sizeof(NODE));  
newnode->air_number=num;  
newnode->link=start;  
    start=newnode;  
    req++;  
}  
  
int delete_end()  
{  
    NODE *curptr=NULL,*nextptr=NULL;  
    int n;  
    if(start->link==NULL)  
    {  
curptr=start;
```

```
    n=curptr->air_number;
    free(curptr);
}
else
{
curptr=start;
    while(curptr->link!=NULL)
    {
nextptr=curptr;
curptr=curptr->link;
    }
    n=curptr->air_number;
    free(curptr);
nextptr->link=NULL;
}
out++;
return n;
}
int count_nodes()
{
    NODE *curptr=NULL;
    int count=0;
    if(start==NULL)
    {
        return 0;
    }
else
```

```

    {
curptr=start;

    while(curptr!=NULL)
    {
curptr=curptr->link;

        count++;

    }

}

return count;
}

void summary()
{
    int i;

fp=fopen(fname,"a");

fprintf(fp,"n*****\n");

fprintf(fp,"n\t\tSUMMARY OF %d units\n",units);

fprintf(fp,"n*****\n");

fprintf(fp,"n\tThe plans requested for landing    : %d\n",req);

fprintf(fp,"n\tNumbers of plans landed on runway 1 : %d\n",run1);

fprintf(fp,"n\tNumbers of plans landed on runway 3 : %d\n",run3);

fprintf(fp,"n\tNumber of Plans take off          : %d\n",out);

fprintf(fp,"n\tNumber of plans delayed            : %d\n",del);

fprintf(fp,"n\tThe Total Plans landed                : %d\n",run1+run3);

fprintf(fp,"n*****\n");

fprintf(fp,"n\tThe plans landed on runway 1 are : \n");

fprintf(fp,"n");

    for(i=0;i<run1;i++)

```

```
{
fprintf(fp, "\t\tPlan %d\n", runw1[i]);
}
if(run3!=0)
{
fprintf(fp, "\n*****\n");
fprintf(fp, "\n\tThe plans landed on runway 3 are : \n");
fprintf(fp, "\n");
    for(i=0; i<run3; i++)
    {
fprintf(fp, "\t\tPlan %d\n", runw3[i]);
    }
}
fprintf(fp, "\n*****\n");
fprintf(fp, "\n\tThe plans takeoff on runway 2 are : \n");
fprintf(fp, "\n");
    for(i=0; i<out; i++)
    {
fprintf(fp, "\t\tPlan %d\n", takeoff[i]);
    }
fprintf(fp, "\n*****\n");
fclose(fp);
}
void simulation_log()
{
summary();
fp=fopen(fname, "a");
```

```
fprintf(fp, "\nSimulation ended at\n");  
fclose(fp);  
currenttime(fname);  
}
```

## HEADER FILES

```
1. #ifndef CUR_TIME  
#define CUR_TIME  
int currenttime(char fname[20])  
{  
    FILE *fp;  
    time_t curtime;  
    fp=fopen(fname, "a");  
    time(&curtime);  
    fprintf(fp, "\t%s\n", ctime(&curtime));  
    fclose(fp);  
}  
#endif // CUR_TIME
```

```
2. #ifndef RANDOM  
#define RANDOM  
int random()  
{  
    static int next = 3251;  
    next=((next*next)/100)% 10000;
```



```
        return next;
    }
#endif // RANDOM

3. #ifndef READ_THIS_FILE
#define READ_THIS_FILE

void printfile(char fname[20])
{
    char c;
    FILE *fp;
    fp=fopen(fname,"r");
    if(fp==NULL)
    {
        printf("Cannot open the file!\n");
    }
    c=fgetc(fp);
    while(c!=EOF)
    {
        printf("%c",c);
        c=getc(fp);
    }
    fclose(fp);
}

#endif // READ_THIS_FILE
```

## 4. RESULTS

```

C:\Users\just call me gcm\Desktop\code blocks\123 atc.exe
*****
AIR TRAFIC CONTROLLER
*****
Enter the number of units : 3
Enter the name of the file that will store the simulation log for you:atcsim
1 :
    Plan 390 requested for landing
    Plan 261 requested for landing
    Plan 390 landed on runway 1
    Plan 390 take off on runway 2
    plan 261 landed on runway 1
2 :
    Plan 651 requested for landing
    Plan 254 requested for landing
    Plan 254 landed on runway 1
    Plan 651 delayed for landing
    Plan 651 landed on runway 3
    Plan 261 take off in runway 2
    Plan 651 take off in runway 3
3 :
    Plan 598 requested for landing
    Plan 208 requested for landing
    Plan 208 landed on runway 1
    
```

**Figure 1: Simulation of 3 units-taking input from user and displaying the simulation in the file named atcsim**

```

C:\Users\just call me gcm\Desktop\code blocks\123 atc.exe
    Plan 208 landed on runway 1
    Plan 598 delayed for landing
    Plan 598 landed on runway 3
    Plan 254 take off in runway 2
    Plan 651 take off in runway 3
    Would you like to see the summary??
    Enter 1 for YES 0 for NO : 1
    You can see the contents of this file now
    Simulation started at:
    Mon Feb 01 20:34:09 2021
*****
SUMMARY OF 3 units
*****
    The plans requested for landing      : 8
    Numbers of plans landed on runway 1 : 4
    Numbers of plans landed on runway 3 : 2
    Number of Plans take off             : 5
    Number of plans delayed               : 2
    The Total Plans landed               : 6
*****
    The plans landed on runway 1 are :
    Plan 390
    
```

**Figure 2: Simulation of 3 units-displaying the summary on user request in the file atcsim**

## AIR TRAFFIC CONTROLLER SIMULATOR

```
"C:\Users\just call me gcm\Desktop\code blocks\123 atc.exe"
Plan 390
Plan 261
Plan 254
Plan 208
*****
The plans landed on runway 3 are :
Plan 651
Plan 598
*****
The plans takeoff on runway 2 are :
Plan 390
Plan 261
Plan 651
Plan 254
Plan 651
*****
Simulation ended at
Mon Feb 01 20:34:12 2021
Process returned 0 (0x0) execution time : 13.327 s
Press any key to continue.
```

**Figure 3: Simulation of 3 units-summary involving the details of the plane landing and taking off at different runways as in file atcsim**

```
"C:\Users\just call me gcm\Desktop\code blocks\123 atc.exe"
*****
AIR TRAFFIC CONTROLLER
*****
Enter the number of units : 1
Enter the name of the file that will store the simulation log for you:atcsim
1 :
Plan 390 requested for landing
Plan 261 requested for landing
Plan 390 landed on runway 1
Plan 390 take off on runway 2
plan 261 landed on runway 1
Would you like to see the summary??
Enter 1 for YES 0 for NO : 0
You did not choose to open the file but it is updated
Process returned 0 (0x0) execution time : 9.620 s
Press any key to continue.
```

**Figure 4 : Simulation of 1 unit executed and no summary displayed, as per user request but the same is updated in the file atcsim**

# AIR TRAFFIC CONTROLLER SIMULATOR

```
"C:\Users\just call me gcm\Desktop\code blocks\123 atc.exe"
*****
AIR TRAFIC CONTROLLER
*****
Enter the number of units : 0
Enter the name of the file that will store the simulation log for you:atcsim
Would you like to see the summary??
Enter 1 for YES 0 for NO : 1
You can see the contents of this file now
Simulation started at:
Mon Feb 01 20:34:09 2021
*****
SUMMARY OF 3 units
*****
The plans requested for landing : 8
Numbers of plans landed on runway 1 : 4
Numbers of plans landed on runway 3 : 2
Number of Plans take off : 5
Number of plans delayed : 2
The Total Plans landed : 6
*****
The plans landed on runway 1 are :
```

**Figure 5: Simulation of 0 units-history of all the simulations entered in file atcsim displayed**

```
"C:\Users\just call me gcm\Desktop\code blocks\123 atc.exe"
The plans landed on runway 1 are :
Plan 390
Plan 261
Plan 254
Plan 208
*****
The plans landed on runway 3 are :
Plan 651
Plan 598
*****
The plans takeoff on runway 2 are :
Plan 390
Plan 261
Plan 651
Plan 254
Plan 651
*****
Simulation ended at:
Mon Feb 01 20:34:12 2021
Simulation started at:
Mon Feb 01 20:38:19 2021
Simulation ended at:
Mon Feb 01 20:38:23 2021
Simulation started at:
Mon Feb 01 20:41:25 2021
```

**Figure 6: The complete history of simulations individually executed for 3,1 and 0 units respectively**

# AIR TRAFFIC CONTROLLER SIMULATOR

```
"C:\Users\just call me gcm\Desktop\code blocks\123 atc.exe"
*****
SUMMARY OF 0 units
*****
The plans requested for landing      : 0
Numbers of plans landed on runway 1 : 0
Numbers of plans landed on runway 3 : 0
Number of Plans take off            : 0
Number of plans delayed              : 0
The Total Plans landed              : 0
*****
The plans landed on runway 1 are :
*****
The plans takeoff on runway 2 are :
*****
Simulation ended at
Mon Feb 01 20:41:27 2021

Process returned 0 (0x0)   execution time : 11.103 s
Press any key to continue.
```

**Figure 7: Continued history**

## 5. CONCLUSION

This project idea has been there among us since first semester but we couldn't ideate well and understand how it actually worked. Having an opportunity to get back to it and complete it today is truly a remark of the commitment and work we have put into this brainchild. It is one of the projects we will always look back to for resilience, perseverance, determination and strength when we will need it. Thankful to all my teammates for their teamwork and dedication and to teaching faculty for their constant support. The entire process has seen moments of intensive work, at times immobilized and at times tiring, nevertheless the more we worked for it the more we became deserving. The team building exercises aided well as we were a bunch of acquaintances when we started, but now I know them well as good friends. All in all, it was a good experience, would undeniably look for such projects with my fellow batchmates.

Though our code is simple it can still be deployed to utility in real life with slight modifications (would be suitable and recommended for small domestic airports). However, as a simulator it is quite basic in terms of estimating possible scenarios, their relative permutations, combinations and in proposing solutions, but it is relevant in the above context of a domestic airport. The software certainly has the potential to serve busier airports with significant efforts in software evolution. The system constraint pertaining to the environment of deploying the use of C language can significantly affect the real time application of the code. C running in non-Linux environment do not have the privileges of time manipulations as a variable to execute instructions. At this point the code saturates and cannot be further improved in C language as it will lose compatibility. Here we propose the code that is of requisite functionality and accordance to the language of C. Given the systems operating in C, the code is suitable and compatible of use. That is the advantage of the code- to make the ends meet, that is with a fairly equipped software support we can run quite complicated software such as that of a simulator.

C is certainly a high-level language but it tends to lean a little towards the low-level side when it comes to the execution of complicated or advanced logic. Our code particularly works for the simulation that predicts the cases in normal environments of the systems that are utilizing it, there is no real time execution in this case. That can prove to be a significant drawback considering that a simulation might be employed at crucial scenarios where time is the game. The code breaks at this point. However, we can update this software to be functional at a level

that it mimics reality as well as predicting one, in advanced programming environments such as JavaScript or a more convenient option of Python. These languages support time manipulation and incorporation of permutations and combinations at a fair level than C.

So, there is a fairly open chance to develop a better and apt software true to its name of a simulator. Out of the scope of this course that is outside C this code has potential of recording furthermore updates, and for better functionality it can be evolved. Finally, we would like to say that we will undertake another project and potentially develop an updated and far more “real” software. Looking forward for the day when you all could see it.

## 6. REFERENCES

[1] Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2<sup>nd</sup> Ed, Universities Press, 2014.

[2] Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1<sup>st</sup> Ed, McGraw Hill, 2014.

### Website links:

<https://stackoverflow.com/questions/5141960/get-the-current-time-in-c>

<https://indianairforce.nic.in/>

<https://www.facebook.com/IndianAirForce/>

<https://www.sheffield.com/articles/how-air-traffic-control-systems-work#:~:text=The%20main%20goal%20of%20an,between%20incoming%20and%20outgoing%20flights.>

<https://www.quora.com/What-is-the-importance-of-ATC-in-an-airport>



**JSSATE**  
**Department of Computer Science and Engineering**  
**Management and Entrepreneurship for IT Industry (18CS51)**

**An Entrepreneurial Case Study**

**(Collaborative Learning)**

**Academic Year 2020-21 (September-January)**

**CLASS: 5 B**

**FACULTY : SHANTHALA KV**

**Theme: ONLINE BUSINESS**

E-commerce has always occupied an important space in the marketplace and the pandemic showed us both the resilience of businesses selling online and the demand of community members looking to support local and small businesses. Online stores are the best avenue for aspiring retailers.

Assume that you are an Entrepreneur planning to start an online business of your choice. Prepare a report on Opportunity Analysis and a Business plan with the following components:

1. Idea.
2. Objectives.
3. Planning.
4. SWOT Analysis.
5. Feasibility Study.(Also Challenges)
6. Structure of Management. (Also Among Employees)
7. Staff Recruitment.
8. Marketing.
9. Maintenance/Support
10. ERP solution

A PPT presentation (with appropriate diagrams/figures) and a report of 10-12 pages by team members of the group

**Rubrics:**

**1. Presentation – 4 Marks. 2. Content - 4 Marks 3. Report – 2 Marks**

**PO's mapping:**

| PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|
| -   | -   | -   | -   | -   | √   | √   | √   | √   | √    | √    | -    |

### Schedule for the Presentation:

| TEAM ID | TEAM MEMBERS                | DATE                 | TIME      |
|---------|-----------------------------|----------------------|-----------|
|         | PALLAVI MEHERWAD 1JS18CS100 | 8-1-2021<br>FRIDAY   | 10AM-11AM |
|         | PAVITHRA J 1JS18CS102       |                      |           |
|         | P. YASASWINI 1JS18CS103     |                      |           |
|         | RASHMI KESHARI 1JS18CS128   |                      |           |
| 2       | NIKIL SAVALGI 1JS18CS095    |                      |           |
|         | PRAJWAL D R 1JS18CS109      |                      |           |
|         | PRAJWAL HULAMANI 1JS18CS110 |                      |           |
|         | RAJATH CR 1JS18CS124        |                      |           |
| 3       | PAVAN P 1JS18CS101          |                      |           |
|         | NISHANT KETU 1JS18CS097     |                      |           |
|         | RAJAT GUPTA 1JS18CS123      |                      |           |
|         | MOHIT KUMAR 1JS18CS085      |                      |           |
| 4       | MANIKANTA B 1JS18CS075      | 9-1-2021<br>SATURDAY | 11AM-12PM |
|         | MANJUNATH BHOVI 1JS18CS076  |                      |           |
|         | NIRANJANA K 1JS18CS096      |                      |           |
|         | RAKSHITH N KUMAR 1JS18CS126 |                      |           |
|         | LEELANANDA NAIK 1JS18CS071  |                      |           |
| 5       | MEGHANA J 1JS18CS079        |                      |           |
|         | MERLLIN M 1JS18CS082        |                      |           |
|         | NIDHI MADYALKAR 1JS18CS092  |                      |           |
|         | PRAJNA B R 1JS18CS108       |                      |           |
| 6       | KUSHI S 1JS18CS070          |                      |           |
|         | LIPIKA B A 1JS18CS072       |                      |           |
|         | MEGHA V SALIMATH 1JS18CS078 |                      |           |
|         | NEHA R 1JS18CS090           |                      |           |
| 7       | YASH MAKADIA 1JS18CS074     | 11-1-2021<br>MONDAY  | 10AM-11AM |
|         | KUSHAGRA AGRAWAL 1JS18CS069 |                      |           |
|         | MILAN PANDEY 1JS18CS083     |                      |           |
|         | NISHANTH 1JS17CS063         |                      |           |
|         | PUNEET YADAV 1JS18CS117     |                      |           |
| 8       | RAHUL R 1JS18CS122          |                      |           |
|         | PRASHANTH BHAT 1JS18CS113   |                      |           |
|         | PRASHANTH DS 1JS18CS112     |                      |           |
|         | MOHAMMED AYAAN 1JS18CS084   |                      |           |
| 9       | JYOTHSNA R 1JS18CS066       |                      |           |
|         | MEGHANA K 1JS18CS080        |                      |           |
|         | N.YOGITHAA NAG 1JS18CS086   |                      |           |
|         | P SANDHYA RANI 1JS18CS099   |                      |           |
|         | PRAGATHI M R 1JS18CS107     |                      |           |
|         | PINTU D 1JS18CS104          |                      |           |
|         | ABHISHEK S M 1JS17CS003     |                      |           |

|    |                                      |                        |                  |
|----|--------------------------------------|------------------------|------------------|
|    | NEHANA 1JS18CS091                    | 13-2-2021<br>WEDNESDAY | 10AM-11AM        |
|    | RANJITHA 1JS18CS127                  |                        |                  |
|    | POOJA 1JS18CS106                     |                        |                  |
| 11 | DISHA PANJWANI 1JS18CS197            |                        |                  |
|    | ANANDI SAGAR 1JS18CS200              |                        |                  |
|    | NIRANJAN 1JS19CS408                  |                        |                  |
|    | PRASANNA SHETTY 1JS18CS111           |                        |                  |
|    | PRITHVI 1JS18CS114                   |                        |                  |
| 12 | MEGHANA R 1JS18CS081                 |                        |                  |
|    | NEHA NAZRE 1JS18CS089                |                        |                  |
|    | NIDHI S 1JS18CS093                   |                        |                  |
|    | MEGHANA S 1JS19CS406                 |                        |                  |
|    | PAVITHRA K S 1JS19CS410              |                        |                  |
| 13 | MUDASSIR MAZHAR 1JS18CS077           | 15-1-2021<br>FRIDAY    | 10AM-11AM        |
|    | PRIOESH AGRAWAL 1JS18CS 116          |                        |                  |
|    | KARAN KUMAR 1JS18CS067               |                        |                  |
|    | PIYUSH AGNIHOTRI 1JS18CS105          |                        |                  |
| 14 | RACHIT GUPTA 1JS18CS119              |                        |                  |
|    | RAJPUROHIT KALPESHKUMAR B 1JS18CS125 |                        |                  |
|    | PRIYANK BALIYAN 1JS18CS115           |                        |                  |
| 15 | MRUTHYUNJAYA M R 1JS19CS407          | 16-1-2021<br>SATURDAY  | 11AM-<br>12.30PM |
|    | NITHISH KUMAR G 1JS19CS409           |                        |                  |
|    | PRATHAPA S 1JS19CS411                |                        |                  |
|    | MADAN.M 1JS18CS073                   |                        |                  |
| 16 | NAVNEETH.M 1JS18CS087                |                        |                  |
|    | NEERAJ 1JS18CS088                    |                        |                  |
|    | RAHUL.B.N 1JS18CS121                 |                        |                  |
|    | NIKHIL 1JS18CS094                    |                        |                  |
| 17 | DIPEN REDDY                          |                        |                  |



**JSS ACADEMY OF TECHNICAL EDUCATION, BANGALORE-560060**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
ACADEMIC YEAR: 2020-2021**

**SUBJECT: MANAGEMENT AND ENTREPRENEURSHIP FOR IT  
INDUSTRY (18CS51)**

**ACTIVITY – 02 (ONLINE GIFT STORE)**

**REPORT ON  
TINYTREATS [TINY TINY HAPPINESS IN IT]**

**SUBMITTED BY:**

| S no. | Name                    | USN        |
|-------|-------------------------|------------|
| 1.    | PALLAVI MEHARWADE       | 1JS18CS100 |
| 2.    | PAVITHRA J              | 1JS18CS102 |
| 3.    | PEDDA N REDDY YASASWINI | 1JS18CS103 |
| 4.    | RASHMI KESHARI          | 1JS18CS128 |

**NAME OF THE FACULTY INCHARGE: SHANTHALA KV**

## **ABSTRACT**

Online shopping has obtained very important position in the 21<sup>st</sup> century as most of the people are busy, loaded with hectic schedule and also now this pandemic situation which made peoples to only come out for limited works with all safety and precautions. In such a situations online shopping became the easiest and most suitable mode for their shopping. Internet has changed the way of consumer's store, and has rapidly developed into a global perspective. "Online Gift Store" is a web-based Business of Gift Shop where peoples can send gift boxes filled with happiness to their loved ones by sitting relaxed in their house for any occasions.

A wedding, the birth of a child, a new home, an important birthday... All these are events that mark our lives forever and should be celebrated with a gift you can cherish for a lifetime. Every person is in search of attractive and innovative products for purpose of exchanging gifts. So idea of this online business is to provide a large variety of innovative and creative gift boxes having dozens of happiness packed in it.

The name of our business "**TINYTREATS**" which is an effort to provide exceptional gifts for special celebrations and occasions to our customers.

## TABLE OF CONTENT

| <b>S.NO</b> | <b>TOPICS</b>  |
|-------------|--|
| 1.          | INTRODUCTION   |
| 1.1<br>1.2  | OBJECTIVE<br>PLANNING  |
| 2.          | SWOT ANALYSIS<br><br>(S)STRENGTHS<br>(W)WEAKNESS<br>(O)OPPORTUNITIES<br>(T)THREATS |
| 3.          | OUR PREVIOUS ORDERS  |
| 4.          | MARKETING  |
| 4.1         | MARKETING STRATEGY   |
| 5.          | ERP  |

# 1.INTRODUCTION

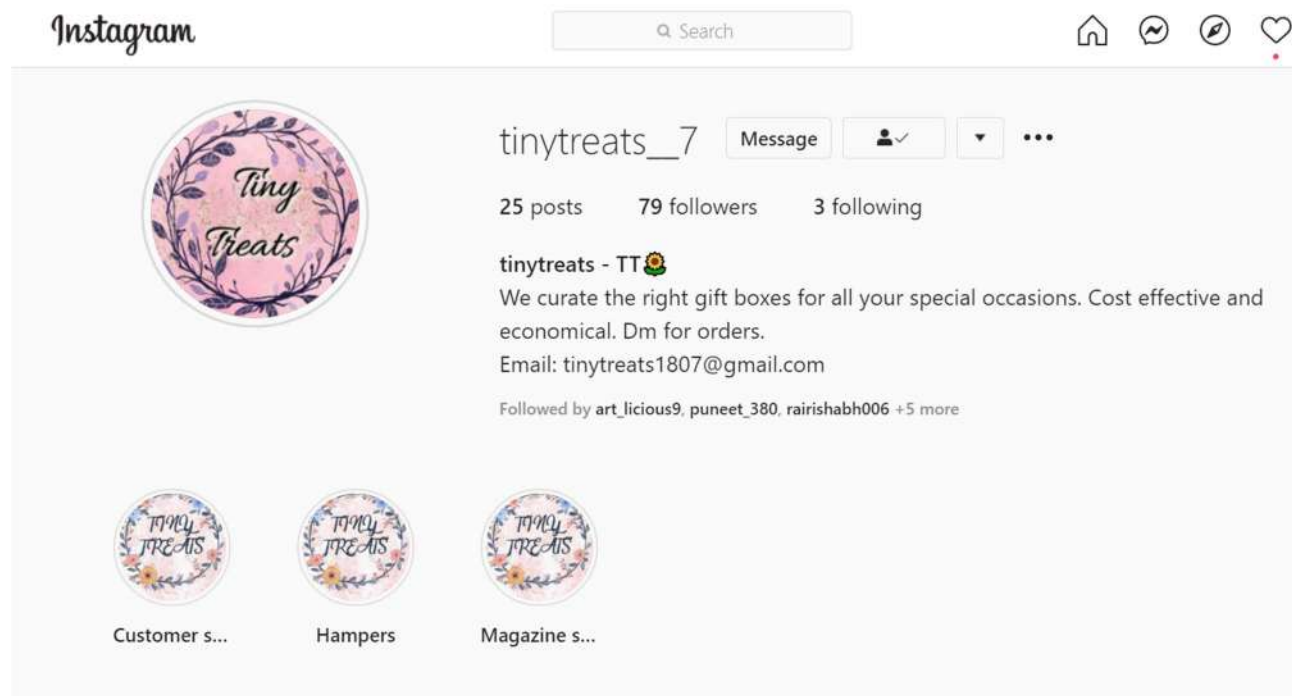
The idea of 'Tinytreats' is to provide a large variety of innovation and creative self-made gift boxes under one roof. The name of our organisation is 'Tinytreats' which is actually an effort to provide exceptional items for special celebrations and occasions.

It is basically a manufacturing as well as merchandising business in which we are going to present a wide assortment of self-made gift boxes. This idea mainly came by seeing this pandemic issue where everyone is locked in home and not able to celebrate the special occasions with their loved ones so it's a way of sending a right gift boxes with full of love and happiness to them even in this situation.

Email: [tinytreats1807@gmail.com](mailto:tinytreats1807@gmail.com)

**Our Instagram page:**

[https://instagram.com/tinytreats\\_\\_7?igshid=idewcawigkod](https://instagram.com/tinytreats__7?igshid=idewcawigkod)



## **1.1. OBJECTIVE**

Main objective of this TINYTREATS is to glorify the different occasions of our customers by providing exclusive gift boxes full of tiny gifts in it. Our modish style reflects a sense of taste that urges the customer to express his or her feelings by selecting a right box for their dear ones.

There are numerous businesses offering a variety of gift items for customers to fulfil their needs, but as a personal experience of ours we had seen how we want the gift to be and that's why we had chosen this business idea so our objective is that we can make a right gift boxes for the right occasions. We believe in extending such incredible service to all customers that they feel real sense of creativity in our establishment. We constantly anticipate, meet, and exceed our customers' expectations.

## **1.2. PLANNING**

The work of the business will be divided among the team members like marketing, making of gift hampers, sales etc.

Marketing is the beginning level of any business, so our marketing aspect should be effective and for online business social media marketing is the best in this pandemic situation. First level of our planning is to make our page to include all the feedbacks, previous orders placed and many sample orders also so that customers wishes to order the gifts after visiting our social media page and next is to make our business page to reach as far as possible among people.

Then to prepare our gift boxes we should purchase the boxes, chocolates, candies, accessories etc to make a big box of happiness. To make magazines we should design it well with the photos given by customers and messages also. We should plan the production according to the customers delivery time when they want it to get deliver.

Financial planning is the most important aspect that we do not go into loss and we can satisfy the customers within a specific price (not too high for customers and not too low for our business).



## 2. SWOT ANALYSIS

To run a successful business, you should regularly analyse your processes to ensure you are operating as efficiently as possible. One of the most effective methods is to conduct a SWOT analysis. A SWOT (strengths, weaknesses, opportunities, and threats) analysis is a planning process that helps your company overcome challenges and determine what new leads to pursue.

The primary objective of a SWOT analysis is to help organizations develop a full awareness of all the factors involved in making a business decision. A SWOT analysis focuses on the four elements of the acronym, allowing companies to identify the forces influencing a strategy, action or initiative. Knowing these positive and negative elements can help company more effectively communicate what parts of a plan need to be recognized,

The SWOT Analysis for “TINYTREATS”:

### (S) STRENGTHS:

- Innovative and passionate in making gifts.
- Good customer experience.



- Customer oriented.
- Self-gift boxes with handmade item purchasing facilities.

### (W) WEAKNESS:

- Low web developing skills.
- No sponsors yet.
- Lack of awareness among public regarding the business.
- Uncertainty about the business success.
- The range of items offered is limited compared to supermarkets which limits the options available to consumer.

**(O) OPPORTUNITIES:**

- Opportunity to become an entrepreneur.
- To curate gifts and wedding invites for rich people.
- To curate gift hampers for MNC's.
- No direct competitor.
- High number of repeat customers may enable increased sales through the addition of complimentary product lines.

**(T) THREATS:**

- Competition is vast.
- Lack of marketing skills.
- Rapid use of readymade gift items in the society.
- Losing customers means losing great part of population.

### **3.OUR PREVIOUS ORDERS**

Some of the previous orders placed by customers :



## **4. MARKETING**

tinytreats will develop effective marketing & sales strategies by focusing on the following key areas:

- Prices
- Promotions
- Distribution
- Channels
- Customer relations
- Products

### **4.1. MARKETING STRATEGY**

#### **PRICING STRATEGY**

Although tinytreats products are high-end and stylish, our pricing structure remains affordable and in-line with other specialty gift retailers. Our pricing structure is also based on the value the customers place on these products.

#### **PROMOTION STRATEGY**

Tinytreats will host several seasonal open houses offering special discounts. This promotional strategy will showcase new products and liquidate slow moving merchandise.

## 5. ERP

- Supervise sales information from all areas and locations
- Insert consumer information at the point of sale screen
- Develop customer support and consistent business with the establishment of loyalty programs
- Establish targeted sales and promotions to boost transactions
- Maintain customer and sales information in a single database and have a real-time access to all data
- Sell materials that contain several different items like gift baskets and gift kits of numerous types
- Simple and swift item and order look-up
- Copy and print gift custom card messages, notes, and covers
- Individual mail and telephone order record options
- Export the equivalent gift choice to various receivers
- Completely unified and traced gift cards and certificates



# ONLINE GIFT STORE

# TINY TREATS

(Tiny Tiny happiness in it)



# OUR PAGE

The image shows a screenshot of an Instagram profile for 'tinytreats\_\_7'. The profile picture is a circular logo with a pink background, a purple leafy wreath, and the text 'Tiny Treats' in a cursive font. The bio includes the text 'tinytreats - TT' with a sunflower emoji, followed by 'We curate the right gift boxes for all your special occasions. Cost effective and economical. Dm for orders.' and 'Email: tinytreats1807@gmail.com'. Below the bio, it says 'Followed by art\_licious9, puneet\_380, rairishabh006 +5 more'. At the bottom, there are three circular thumbnails with the 'TINY TREATS' logo, each with a caption: 'Customer s...', 'Hampers', and 'Magazine s...'. The Instagram interface includes the 'Instagram' logo, a search bar, and navigation icons for home, messages, post, and likes.

Instagram

Search

Home Messages Post Likes

tinytreats\_\_7 Message [Profile] [Dropdown] [More]

25 posts 79 followers 3 following

tinytreats - TT 🌻  
We curate the right gift boxes for all your special occasions. Cost effective and economical. Dm for orders.  
Email: tinytreats1807@gmail.com

Followed by art\_licious9, puneet\_380, rairishabh006 +5 more

Customer s... Hampers Magazine s...

## IDEA BEHIND TINYTREATS



- THE IDEA OF 'TINYTREATS' IS TO PROVIDE A LARGE VARIETY OF INNOVATION AND CREATIVE SELF MADE GIFT BOXES UNDER ONE ROOF. THE NAME OF OUR ORGANISATION IS 'TINYTREATS' WHICH IS ACTUALLY AN EFFORT TO PROVIDE EXCEPTIONAL ITEMS FOR SPECIAL CELEBRATIONS AND OCCASIONS.
- IT IS BASICALLY A MANUFACTURING AS WELL AS MERCHANDISING BUSINESS IN WHICH WE ARE GOING TO PRESENT A WIDE ASSORTMENT OF SELF MADE GIFT BOXES.
- THIS IDEA MAINLY CAME BY SEEING THIS PANDEMIC ISSUE WHERE EVERYONE IS LOCKED IN HOME AND NOT ABLE TO CELEBRATE THE SPECIAL OCCASIONS WITH THEIR LOVED ONES SO ITS A WAY OF SENDING A RIGHT GIFT BOXES WITH FULL OF LOVE AND HAPPINESS TO THEM EVEN IN THIS SITUATION.



- MAIN OBJECTIVE OF THIS TINYTREATS IS TO GLORIFY THE DIFFERENT OCCASIONS OF OUR CUSTOMERS BY PROVIDING EXCLUSIVE GIFT BOXES FULL OF TINY GIFTS IN IT. OUR MODISH STYLE REFLECTS A SENSE OF TASTE THAT URGES THE CUSTOMER TO EXPRESS HIS OR HER FEELINGS BY SELECTING A RIGHT BOX FOR THEIR DEAR ONES.



- THERE ARE NUMEROUS BUSINESSES OFFERING A VARIETY OF GIFT ITEMS FOR CUSTOMERS TO FULFIL THEIR NEEDS, BUT AS A PERSONAL EXPERIENCE OF OURS WE HAD SEEN HOW WE WANT THE GIFT TO BE AND THAT'S WHY WE HAD CHOSEN THIS BUSINESS IDEA SO OUR OBJECTIVE IS THAT WE CAN MAKE A RIGHT GIFT BOXES FOR THE RIGHT OCCASIONS.
- WE BELIEVE IN EXTENDING SUCH INCREDIBLE SERVICE TO ALL CUSTOMERS THAT THEY FEEL REAL SENSE OF CREATIVITY IN OUR ESTABLISHMENT. WE CONSTANTLY ANTICIPATE, MEET, AND EXCEED OUR CUSTOMERS EXPECTATIONS.

# PLANNING

THE WORK OF THE BUSINESS WILL BE DIVIDED AMONG THE TEAM MEMBERS LIKE MARKETING, MAKING OF GIFT HAMPERS, SALES ETC.



MARKETING IS THE BEGINNING LEVEL OF ANY BUSINESS, SO OUR MARKETING ASPECT SHOULD BE EFFECTIVE AND FOR ONLINE BUSINESS SOCIAL MEDIA MARKETING IS THE BEST IN THIS PANDEMIC SITUATION. FIRST LEVEL OF OUR PLANNING IS TO MAKE OUR PAGE TO INCLUDE ALL THE FEEDBACKS, PREVIOUS ORDERS PLACED AND MANY SAMPLE ORDERS ALSO SO THAT CUSTOMERS WISHES TO ORDER THE GIFTS AFTER VISITING OUR SOCIAL MEDIA PAGE AND NEXT IS TO MAKE OUR BUSINESS PAGE TO REACH AS FAR AS POSSIBLE AMONG PEOPLE.

THEN TO PREPARE OUR GIFT BOXES WE SHOULD PURCHASE THE BOXES, CHOCOLATES, CANDIES, ACCESSORIES ETC TO MAKE A BIG BOX OF HAPPINESS. TO MAKE MAGAZINES WE SHOULD DESIGN IT WELL WITH THE PHOTOS GIVEN BY CUSTOMERS AND MESSAGES ALSO. WE SHOULD PLAN THE PRODUCTION ACCORDING TO THE CUSTOMERS DELIVERY TIME WHEN THEY WANT IT TO GET DELIVER.

FINANCIAL PLANNING IS THE MOST IMPORTANT ASPECT THAT WE DO NOT GO INTO LOSS AND WE CAN SATISFY THE CUSTOMERS WITHIN A SPECIFIC PRICE (NOT TOO HIGH FOR CUSTOMERS AND NOT TOO LOW FOR OUR BUSINESS).

# SWOT ANALYSIS

## STRENGTHS

- INNOVATIVE AND PASSIONATE IN MAKING GIFTS.
- GOOD CUSTOMER EXPERIENCE.



- CUSTOMER ORIENTED.
- SELF GIFT BOXES WITH HANDMADE ITEM PURCHASING FACILITIES.

# WEAKNESS

- LOW WEB DEVELOPING SKILLS.
- NO SPONSORS YET.
- LACK OF AWARENESS AMONG PUBLIC REGARDING THE BUSINESS.
- UNCERTAINTY ABOUT THE BUSINESS SUCCESS.
- THE RANGE OF ITEMS OFFERED IS LIMITED COMPARED TO SUPERMARKETS WHICH LIMITS THE OPTIONS AVAILABLE TO CONSUMER.

- OPPORTUNITY TO BECOME AN ENTREPRENEUR.
- TO CURATE GIFTS AND WEDDING INVITES FOR RICH PEOPLE.
- TO CURATE GIFT HAMPERS FOR MNC'S.
- NO DIRECT COMPETITOR.
- HIGH NUMBER OF REPEAT CUSTOMERS MAY ENABLE INCREASED SALES THROUGH THE ADDITION OF COMPLIMENTARY PRODUCT LINES.

## THREATS

- COMPETITION IS VAST.
- LACK OF MARKETING SKILLS.
- RAPID USE OF READYMADE GIFT ITEMS IN THE SOCIETY.
- LOSING CUSTOMERS MEANS LOSING GREAT PART OF POPULATION.

## FEASIBILITY STUDY

THE GIFTS GETS PREPARED BY THE SELF CREATIVE IDEAS OF THE TEAM, WHICH INCLUDES CANDIES, CHOCOLATES, ACCESSORIES, PERFUME, ETC ACCORDING TO THE OCCASION, THERE IS MAGAZINES ALSO WHICH INCLUDES PHOTOS AND MESSAGES OF THE CUSTOMERS ALONG WITH THE GIFT BOX.



**CHALLENGES:** IT INCLUDES ONLY SOCIAL MEDIA MARKETING SO THE MAIN CHALLENGE IS GETTING LESS CUSTOMERS BECAUSE OF THE LIMITED REACH OF PEOPLE, THERE ARE NO SPONSORS YET SO NO FINANCIAL SUPPORT.

# MARKETING

TINYTREATS WILL DEVELOP EFFECTIVE MARKETING & SALES STRATEGIES BY FOCUSING ON THE FOLLOWING KEY AREAS:

PRICES

PROMOTIONS

DISTRIBUTION

CHANNELS

CUSTOMER RELATIONS

PRODUCTS



# **MARKETING STRATEGY**

## **PRICING STRATEGY**

ALTHOUGH TINYTREATS PRODUCTS ARE HIGH-END AND STYLISH, OUR PRICING STRUCTURE REMAINS AFFORDABLE AND IN-LINE WITH OTHER SPECIALTY GIFT RETAILERS. OUR PRICING STRUCTURE IS ALSO BASED ON THE VALUE THE CUSTOMERS PLACE ON THESE PRODUCTS.

## **PROMOTION STRATEGY**

TINYTREATS WILL HOST SEVERAL SEASONAL OPEN HOUSES OFFERING SPECIAL DISCOUNTS. THIS PROMOTIONAL STRATEGY WILL SHOWCASE NEW PRODUCTS AND LIQUIDATE SLOW MOVING MERCHANDISE.



- SUPERVISE SALES INFORMATION FROM ALL AREAS AND LOCATIONS
- INSERT CONSUMER INFORMATION AT THE POINT OF SALE SCREEN
- DEVELOP CUSTOMER SUPPORT AND CONSISTENT BUSINESS WITH THE ESTABLISHMENT OF LOYALTY PROGRAMS
- ESTABLISH TARGETED SALES AND PROMOTIONS TO BOOST TRANSACTIONS
- MAINTAIN CUSTOMER AND SALES INFORMATION IN A SINGLE DATABASE AND HAVE A REAL-TIME ACCESS TO ALL DATA
- SELL MATERIALS THAT CONTAIN SEVERAL DIFFERENT ITEMS LIKE GIFT BASKETS AND GIFT KITS OF NUMEROUS TYPES
- SIMPLE AND SWIFT ITEM AND ORDER LOOK-UP
- COPY AND PRINT GIFT CUSTOM CARD MESSAGES, NOTES, AND COVERS
- INDIVIDUAL MAIL AND TELEPHONE ORDER RECORD OPTIONS
- EXPORT THE EQUIVALENT GIFT CHOICE TO VARIOUS RECEIVERS
- COMPLETELY UNIFIED AND TRACED GIFT CARDS AND CERTIFICATES

# THANK YOU



**DO VISIT OUR PAGE:**

[https://instagram.com/tinytreats\\_\\_7?igshid=idewcawigkod](https://instagram.com/tinytreats__7?igshid=idewcawigkod)

# JSS ACADEMY OF TECHNICAL EDUCATION

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Date: 22/03/2021

**Faculty Name :** Dr. Naidila Sadashiv

**Class/Section:** Vth B

**Activity Name:** 1. Activity to execute the shell scripts  
2. Programming Assignment on Unix Programming Concepts.

**Academic Year: 2020-21**

## **GOAL OF THE ACTIVITY:**

1. To enhance student's learning skills in terms of Unix shell scripting and programming concepts.
2. To enable students to understand different Unix commands and API's.
3. To make them apply the concepts learnt in the course into action.

## **DESCRIPTION OF ACTIVITY:**

Unix shell commands and API's were discussed in the class along with some demonstration.

Students were given a set of Unix commands and were informed to show the working of those commands with all the different options it supported. Programming assignments were given to make them apply the Unix API's and develop simple programs to know how the process is created, how they communicate and coordinate within the system.

## **USE OF APPROPRIATE METHODS:**

Use of CLI to show the execution of Unix commands by developing simple scripts and programs.

## **RESULTS/OUTCOME :**

Students were able to understand the concepts more clearly by working on shell scripts and API's and will be able to develop applications with knowledge gained.

**Signature of the Faculty**

**HOD, CSE**

# User Interface Design (17CS832)

## Assignment 3

### Designing User Interface for a given Case Study

(Collaborative Learning)

Academic Year 2021 (April-August)

CLASS: 8 Semester

#### Description of the Activity:

1. A team of 4 students are formed.
2. A case Study is assigned to each group. Each group has to design a User Interface with respect to the topic assigned using a Web UI Mock tool (Eg: Figma, Fluid etc)
3. Function of Design Elements in each screen should be briefly explained.
4. Also standard Guidelines followed should be highlighted.(Font, colour, alignment, selection control etc)
5. A report of 5-6 pages containing all the above details should be submitted.

#### Rubrics:

- 1. Presentation – 2 Marks.
- 2. Content - 4 Marks
- 3. Report – 4 Marks

#### PO's mapping:

| PO 1 | PO 2 | PO 3 | PO 4 | PO 5 | PO 6 | PO 7 | PO 8 | PO 9 | PO1 0 | PO1 1 | PO1 2 |
|------|------|------|------|------|------|------|------|------|-------|-------|-------|
| √    | √    | √    | √    | √    | -    | -    | -    | -    | -     | -     | -     |

**User Interface  
Design(17CS832)  
UID GROUP ACTIVITY  
Faculty: SHANTHALA KV  
Semester -8**

| <b>GROUP Id</b> | <b>Team Members Name</b> | <b>USN</b> | <b>Topic</b>  |
|-----------------|--------------------------|------------|---|
| <b>1</b>        | Anusha S                 | 1JS17CS018 | Design a User interface for an ATM Machine which enables customer with an account to withdraw amount , deposit amount and check the balance.  |
|                 | Prajwala G               | 1JS17CS034 |   |
|                 | N R Usha                 | 1JS16CS127 |   |
|                 | Madhu G A                | 1JS15CS131 |   |
| <b>2</b>        | Mohammed Arshed Ahmed    | 1JS17CS055 | Design a user interface for an Online clothing store, which offers variety of cloth lines to Men, Women and Kids. Assume that Currently Diwali offer is going on. Add discounts and combos to the UI.   |
|                 | Ritik Dubey              | 1JS17CS079 |   |
|                 | Rakesh G                 | 1JS16CS079 |   |
|                 | Parikshith Raj Ellur     | 1JS17CS065 |   |
| <b>3</b>        | Saransh Thukral          | 1JS16CS090 | Design a user interface for an online Resort reservation system. Keep the Covid Pandemic Situation in mind and add a screen which offers extra care taken by the resort.  |
|                 | Azfar Ahmad              | 1JS16CS025 |   |
|                 | Anil Vibhu               | 1JS17CS014 |   |
|                 | Chiranjeevi C R          | 1JS17CS030 |   |
|                 | Kishan Terdal            | 1JS17CS046 |   |
| <b>4</b>        | Anusha V                 | 1js18cs400 | Design a User interface for an ERP system for a college which can be used to manage the information of student attendance and internal marks.   |
|                 | chandu N                 | 1js18cs401 |   |
|                 | Deepa R                  | 1js18cs402 |   |
|                 | Manasa T V               | 1js18cs405 |   |
| <b>5</b>        | M Nishchitha             | 1js16cs051 | Design a User interface for an online government portal , where interested sellers can register , pay a nominal amount of fees to showcase handloom arts and crafts and Sarees from various parts of the country in a fair called Desi Duniya. Customers can also view and book any items of their interest in advance and know more about the history of the craft work. |
|                 | Reha S                   | 1js16cs082 |   |
|                 | Achala S Pandit          | 1js17cs004 |   |
|                 | Anagha K                 | 1js17cs010 |   |
| <b>6</b>        | Abhirup Kulkarni         | 1js17cs002 | Design a user interface for an online campaign for Covid vaccination. Let the system enable registration and Schedule for Vaccination to the citizens of India. Include appropriate Authentication and verification screens.  |
|                 | Brinda Raj               | 1js17cs026 |   |
|                 | Chetana M                | 1js17cs029 |   |
|                 | Prachi Sabarad           | 1jd17cs070 |   |
| <b>7</b>        | Dhanush GC               | 1JS17CS031 | Design a User interface for a system to be used in all police stations in the city. Officers should be able to login and access a database containing criminal records and reports. The system also allows citizens to file an online complaint for minor theft cases.  |
|                 | Manoj Athreya            | 1JS17CS052 |   |
|                 | kushal BH                | 1JS17CS049 |   |
|                 | Punit k                  | 1JS17CS074 |   |
| <b>8</b>        | Ramya K                  | 1JS17CS077 | Design a User interface for an application which helps a user to learn a new Language and improve reading, writing, and speaking skills. It also helps user to Work toward his/her language learning goals with playful rewards and achievements.   |
|                 | Rathi Ramachandran       | 1JS17CS078 |   |

|           |                       |            |  |
|-----------|-----------------------|------------|--|
| <b>9</b>  | Mahaveer. b           | 1js17cs051 | Design a User interface for an online system BreadnBfast that connects people who want to rent out their homes with people who are looking for accommodations in that locale.  |
|           | kushal chaturvedi N H | 1js17cso50 |  |
| <b>10</b> | Harshith R Shekar     | 1JS17CS039 | Design a User interface for an application which facilitates an audiobook subscription service. It also provides books for free on credit earned with adjustable narration speed and a sleep timer.  |
|           | Manoj A M             | 1JS18CS406 |  |
|           | Keerthana R           | 1JS17CS045 |  |
|           | Monica V              | 1JS17CS056 |  |
| <b>11</b> | Apoorva M             | 1JS17CS019 | Design a User Interface for a start-up online Music Streaming site which allows user to listen to music as well as podcasts ,lyrics and translations to other languages. And also recommends songs based on user's interest.   |
|           | Ashmitha C            | 1JS17CS022 |  |
|           | Chaitanya SM          | 1JS17CS027 |  |
| <b>12</b> | Rajiv G M             | 1js17cs075 | Design a User interface for the online portal of a chain of jewellery stores having an exquisite collection of jewellery in Gold, Diamond and Silver. Store also offers a savings scheme in which customers can make a monthly payment for 11 months and can buy at the end of the term along with sale of jewellery |
|           | Harshitha L           | 1js17cs040 |  |
|           | Amit Kumar            | 1js17cs009 |  |
|           | Aishwarya BP          | 1js17cs005 |  |
|           | Aishwarya R           | 1js17cs006 |  |
| <b>13</b> | Arcot Sarvani         | 1js17cs020 | Design a User interface for a cab service portal, which allows cabs to be booked for commute within bangalore city through a mobile app and the service accepts both cash and cashless payments  |
|           | Bindushree .R         | 1js17cs025 |  |
| <b>14</b> | Pranav Vatsa          | 1js17cs071 | Design a car display system car which keeps the passengers entertained during their journey and also to help in the proper navigation of the car.  |

**JSS MAHAVIDYAPEETHA, BENGALURU**  
**JSS ACADEMY OF TECHNICAL EDUCATION**

JSS Campus, Uttarahalli-Kengeri Main Road, Bengaluru - 560060



**UID ASSIGNMENT REPORT**  
**ON**  
**User Interface to Learn a Language**  
**“LINGUALIFT”**

*Submitted in partial fulfilment of the requirements for the User Interface Design  
(17CS832) course of 8th semester*

**BACHELOR OF ENGINEERING**  
**IN**  
**COMPUTER SCIENCE AND ENGINEERING**

Submitted By  
RATHI RAMACHANDRAN(1JS17CS078)  
RAMYA K(1JS17CS077)

Under the guidance of  
Ms. SHANTHALA K V  
Associate Professor,CSE

# TABLE OF CONTENTS

|                         |   |
|-------------------------|---|
| 1. Introduction.....    | 2 |
| 2. Platform.....        | 3 |
| 3. Design elements..... | 5 |
| 4. Functionality.....   | 6 |



# INTRODUCTION

User interface for an application which helps a user to learn a new Language and improve reading, writing, and speaking skills. It also helps users to Work toward his/her language learning goals with playful rewards and achievements. This user interface was designed with the help of a tool Adobe XD. The good part of XD is that it can do both — Visual Design and Animated Prototypes and it is a vector-based tool, unlike Photoshop. The main reason we used Adobe XD is because of its simplicity. Simplicity and usability matter because it is hard to design rich interfaces if using a tool or software which is complex to understand and frustrating to use.

We have designed a user interface prototype for the user to learn a new language and improve his/her speaking and writing skills. The user will first have to sign up to access the courses and then login. The user will then be allowed to pick a language to begin the course and he/she will have to undergo a survey. After which the user is redirected to his/her dashboard where the user can find his/her progress of their learning. The learning process has three levels i.e., basic, intermediate and advanced level which the user has to complete one after the other. In each level of the course the user will have to undergo many assessments and will be given rewards. This keeps the user engaged throughout the learning process and the user will never get bored because of the interactive design.

## **PLATFORM: ADOBE XD**

Adobe XD (also known as Adobe Experience Design) is a vector-based user experience design tool for web apps and mobile apps, developed and published by Adobe Inc. It is available for macOS and Windows, although there are versions for iOS and Android to help preview the result of work directly on mobile devices. Adobe XD supports website wireframing and creating click-through prototypes.

Adobe XD creates user interfaces for mobile and web apps. Main features are:

- Repeat grid

Helps create a grid of repeating items such as lists, and photo galleries.

- Prototype and animation

Creates animated prototypes through linking artboards. These prototypes can be previewed on supported mobile devices.

- Interoperability

XD supports and can open files from Illustrator, Photoshop, Photoshop Sketch, and After Effects. In addition to the Adobe Creative Cloud, XD can also connect to other tools and services such as Slack and Microsoft Teams to collaborate. XD is also able to auto-adjust and move from macOS to Windows. For security, prototypes can be sent with password protection to ensure full disclosure.

- Content-Aware Layout

Design and edit components without the nudging or the tinkering. Content-Aware Layout aligns and evenly-spaces as you add, remove, or resize objects. Make adjustments with smart controls and get back to exploring.

- Voice design

Apps can be designed using voice commands. In addition, what users create for smart assistants can be previewed as well.

- Components

Users can create components (previously known as symbols) to create logos, buttons and other assets for reuse. Their appearance can change with the context where they are used.

- Responsive resize


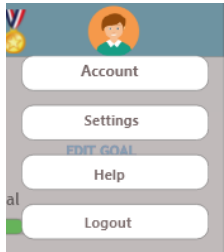

Responsive resize automatically adjusts and sizes pictures and other objects on the artboards. This allows the user to have their content automatically adjusted for different screens for different sized platforms such as mobile phones and PCs.

- Plugins


XD is compatible with custom plugins that add additional features and uses. Plugins range from design to functionality, automation and animation.

# DESIGN ELEMENTS


## Input Controls

| Element                | Description   | Usage in App  |
|------------------------|---|---|
| <b>Buttons</b>         | A button indicates an action upon touch and is typically labeled using text, an icon, or both.                    |    |
| <b>Dropdown Button</b> | The dropdown button consists of a button that when clicked displays a drop-down list of mutually exclusive items. |    |
| <b>Text fields</b>     | Text fields allow users to enter text. It can allow either a single line or multiple lines of text.               |  |

## Navigational Components

|              |   |   |
|--------------|---|---|
| <b>Icons</b> | An icon is a simplified image serving as an intuitive symbol that is used to help users to navigate the system. Typically, icons are hyperlinked. |  |
|--------------|---|---|

## Information Components

|                      |  |   |
|----------------------|--|---|
| <b>Progress Bars</b> | A progress bar indicates where a user is as they advance through a series of steps in a process. Typically, progress bars are not clickable. |  |
|----------------------|--|---|

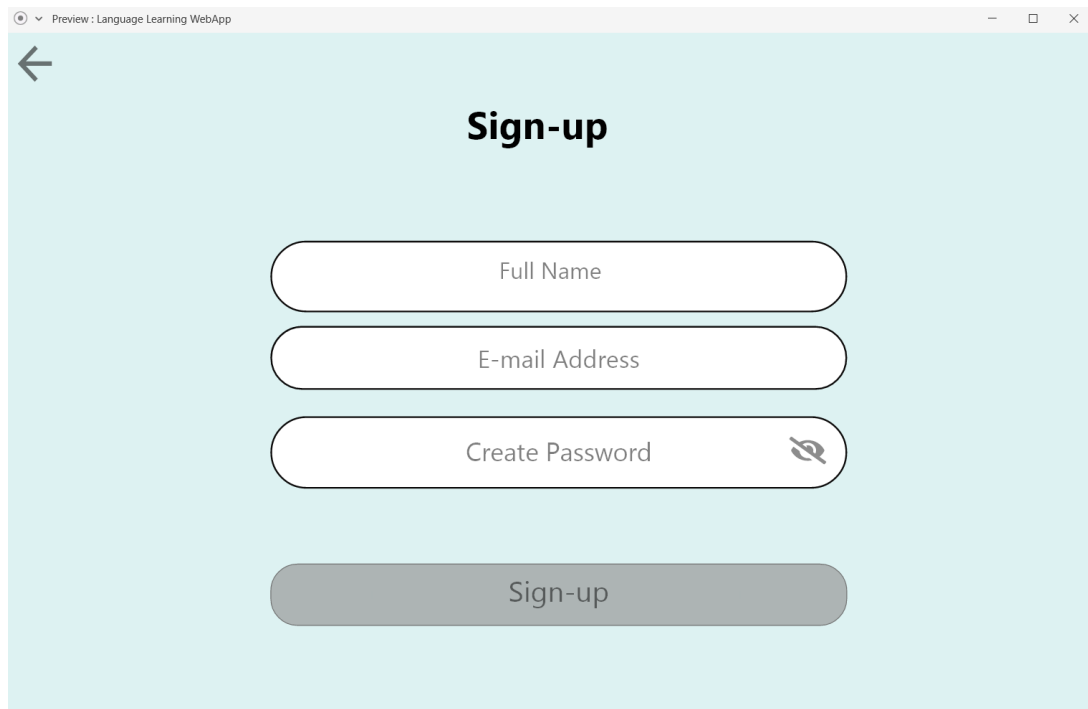
# FUNCTIONALITY

- 1) Home page of a web app that can be scrollable. The user can have access to options like Log in, Sign up and Get Started.



## 2) Sign-up

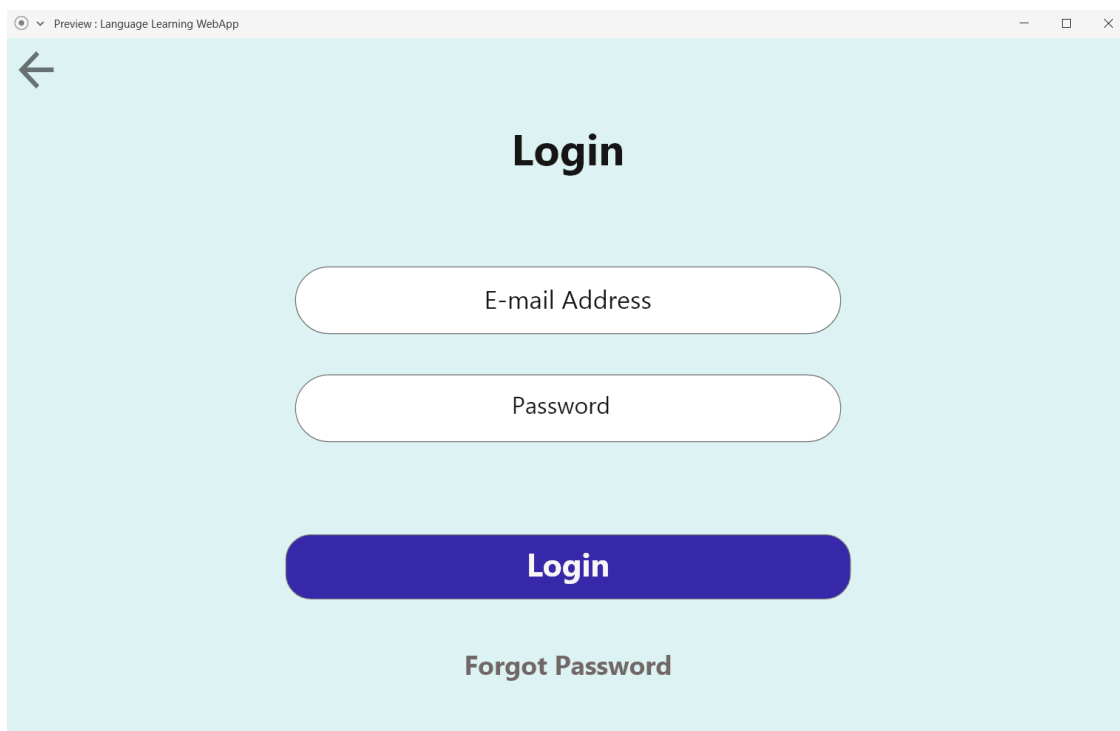
Users will be able to sign-up to the web app by using his/her name, E-mail address and by creating a password.



The screenshot shows a web browser window titled "Preview: Language Learning WebApp". The page has a light blue background and a back arrow in the top left corner. The main heading is "Sign-up". Below the heading are three input fields: "Full Name", "E-mail Address", and "Create Password". The "Create Password" field has a small icon of a crossed-out key on the right side. At the bottom of the form is a grey "Sign-up" button.

## 3) Login page

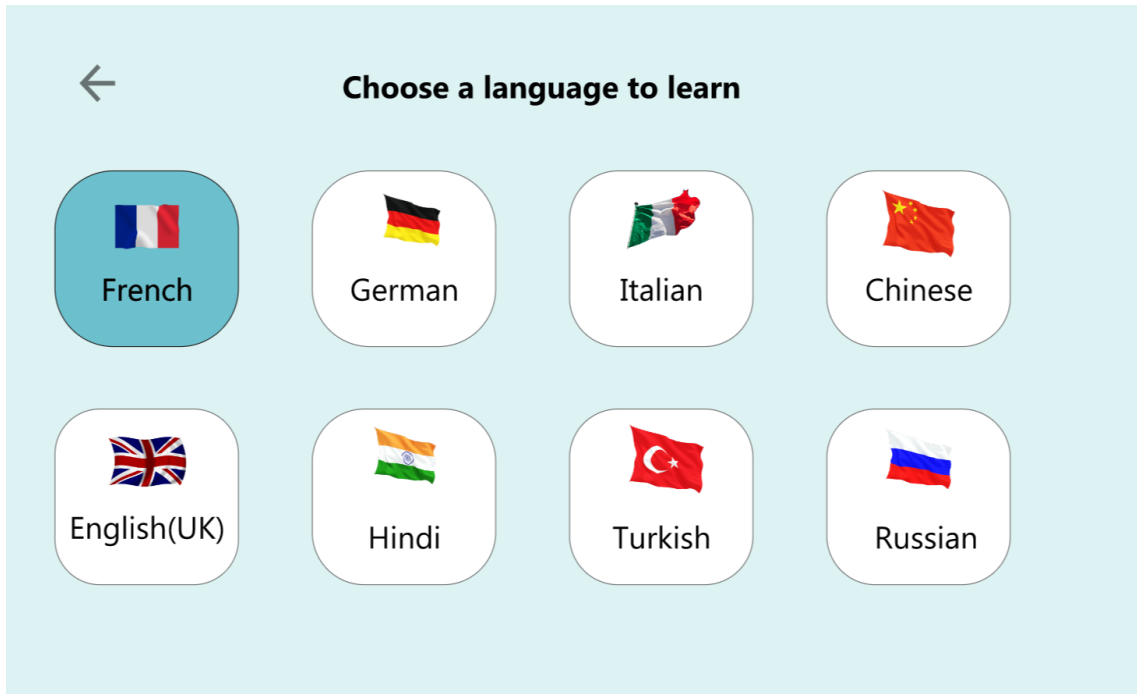
Users can log in to the web app after sign-up, using e-mail address and password. Users can even reset the password by clicking on the forgot password button.



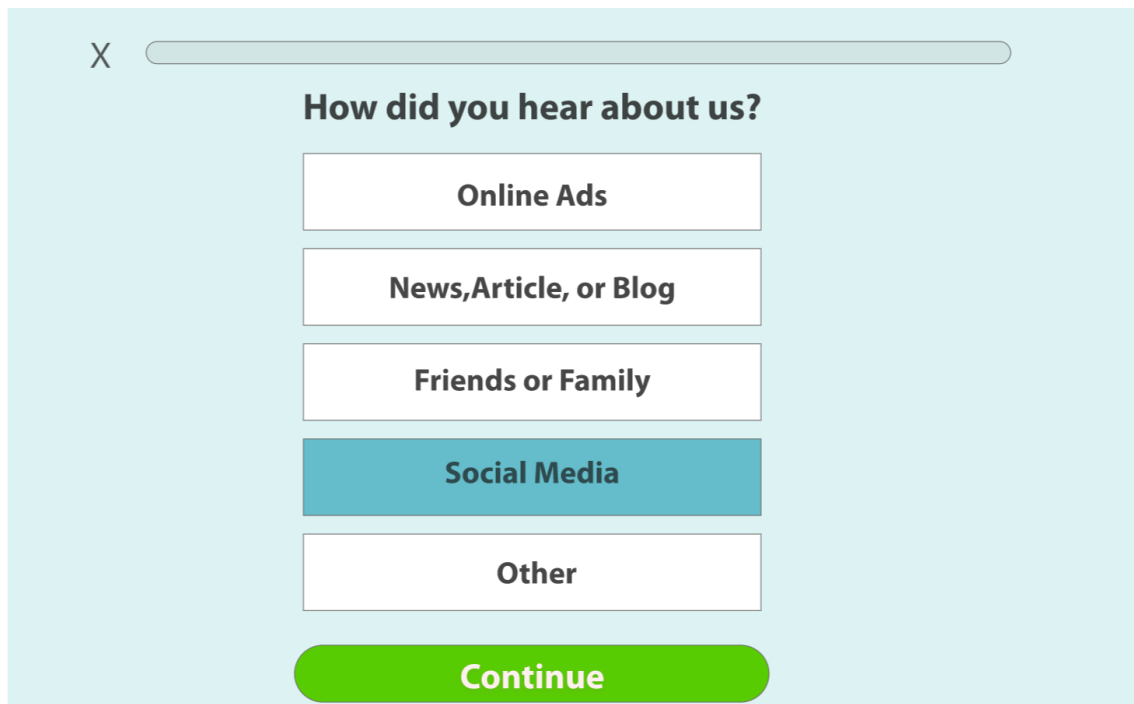
The screenshot shows a web browser window titled "Preview: Language Learning WebApp". The page has a light blue background and a back arrow in the top left corner. The main heading is "Login". Below the heading are two input fields: "E-mail Address" and "Password". At the bottom of the form is a dark blue "Login" button and a "Forgot Password" link.

4) Get started

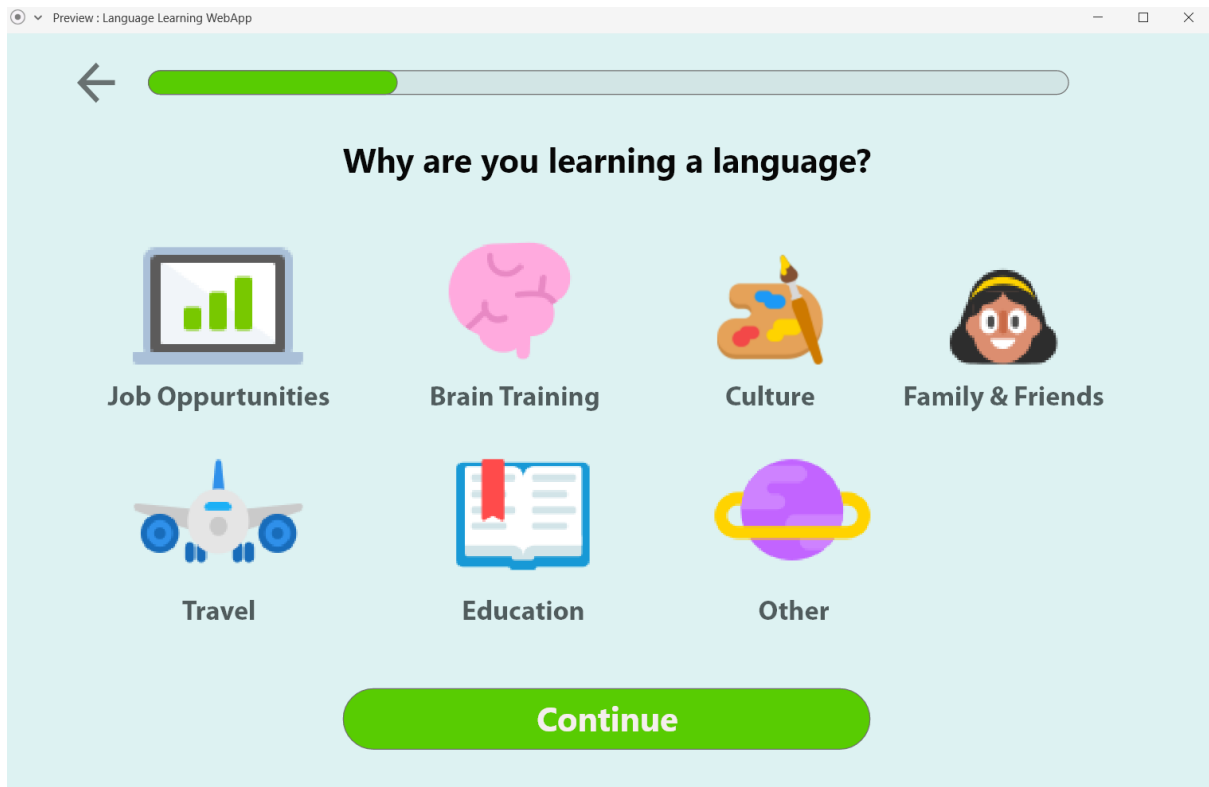
Once the user clicks on the Get started button in the home page, the user will be directed to the following page where the user will have to select a language that he/she wants to learn.



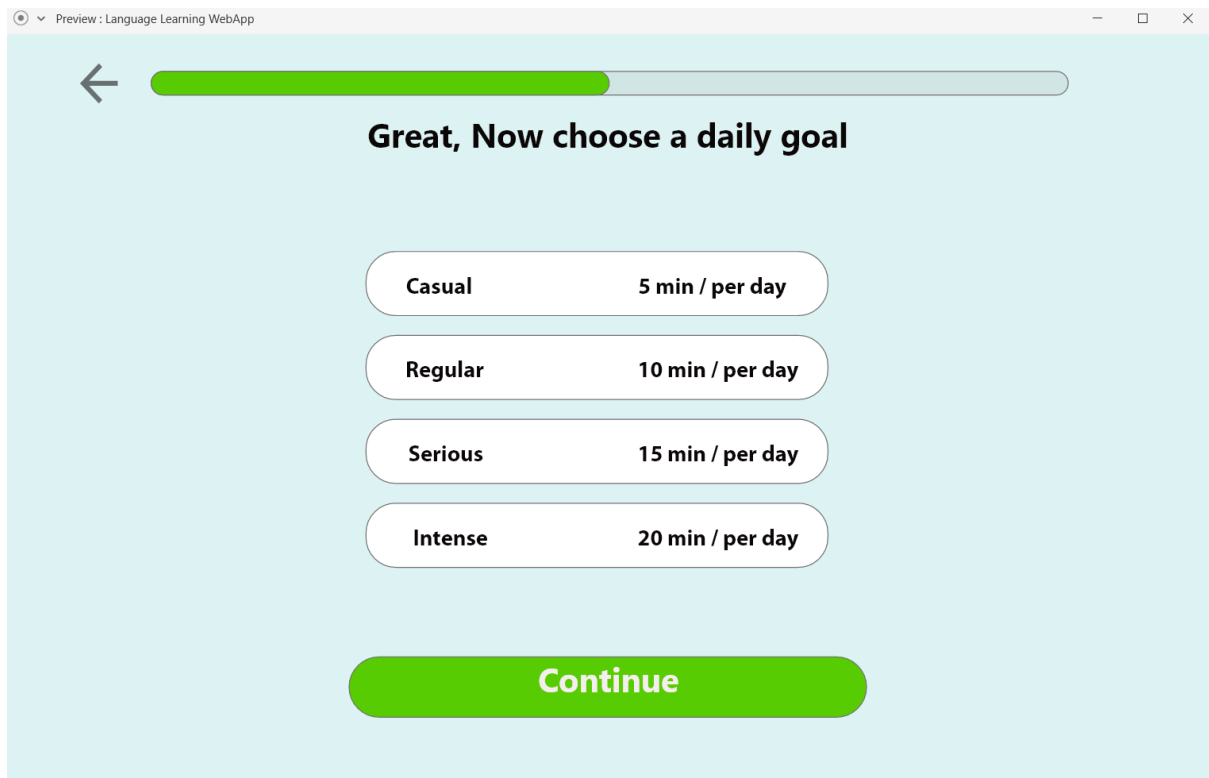
5) Once the user clicks on a button, he/she will be redirected to the following page which is mainly for the survey



6) After the user selects an option, he/she will be redirected to the next page

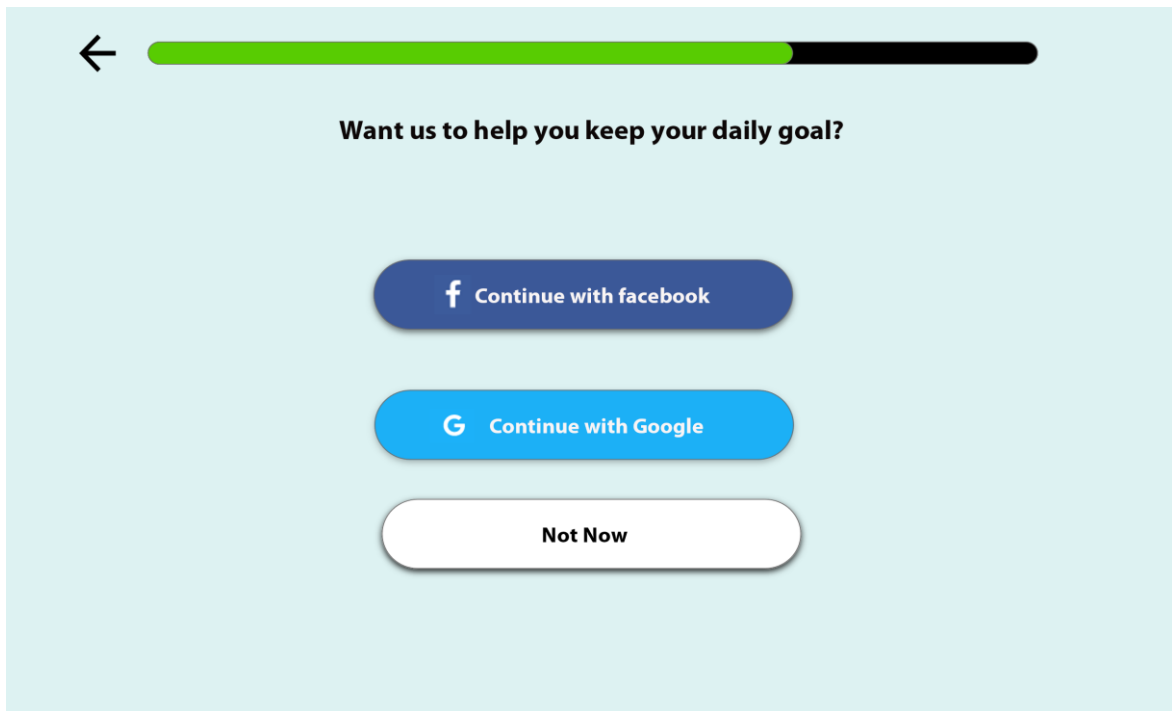


7) Now user will be made to select the type of course he/she wants to undergo

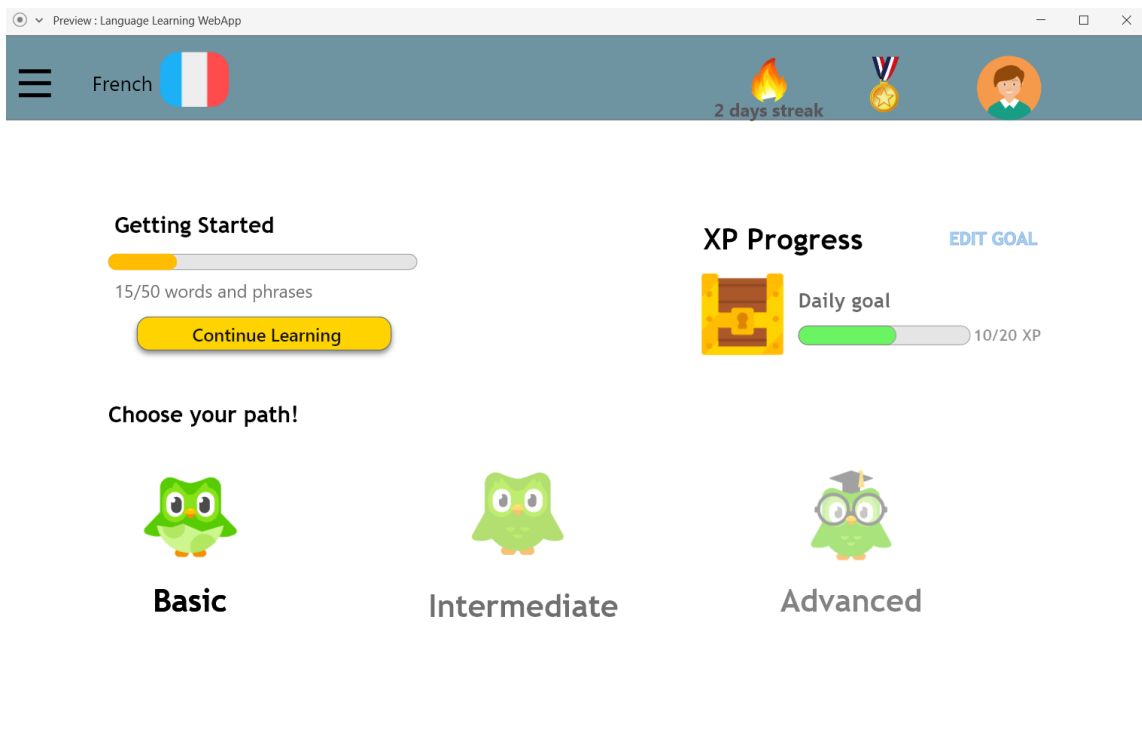




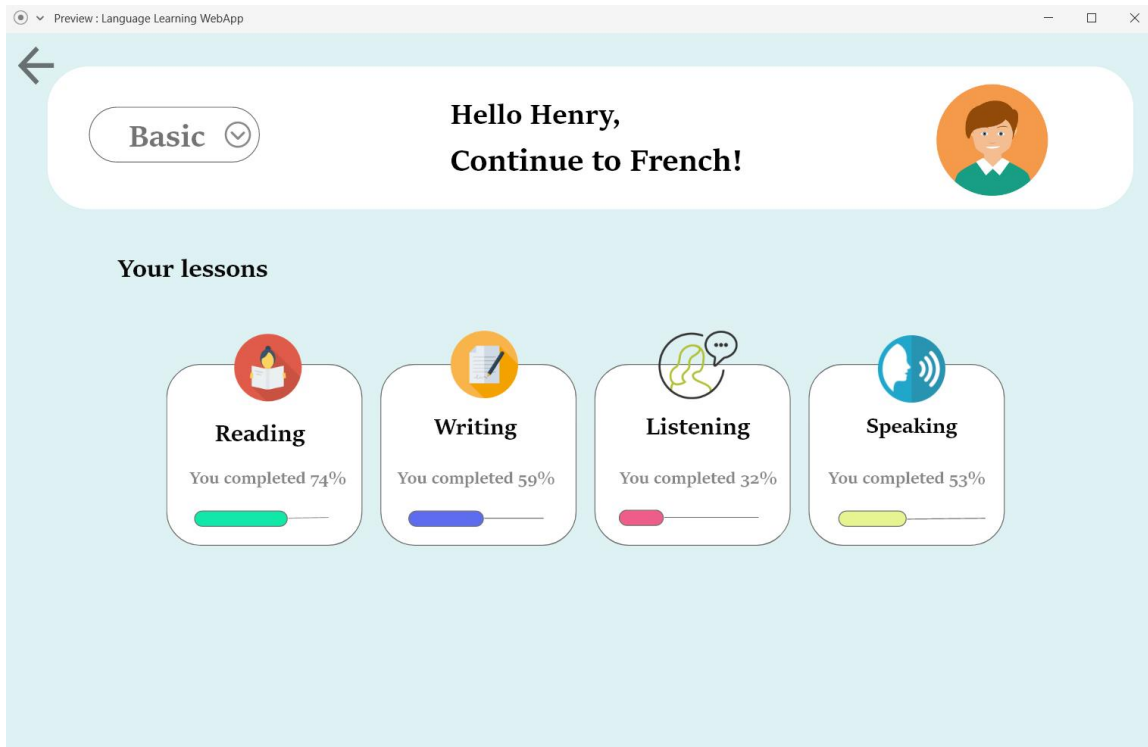
8) As a final step to the learning journey the user will be redirected to the following page.



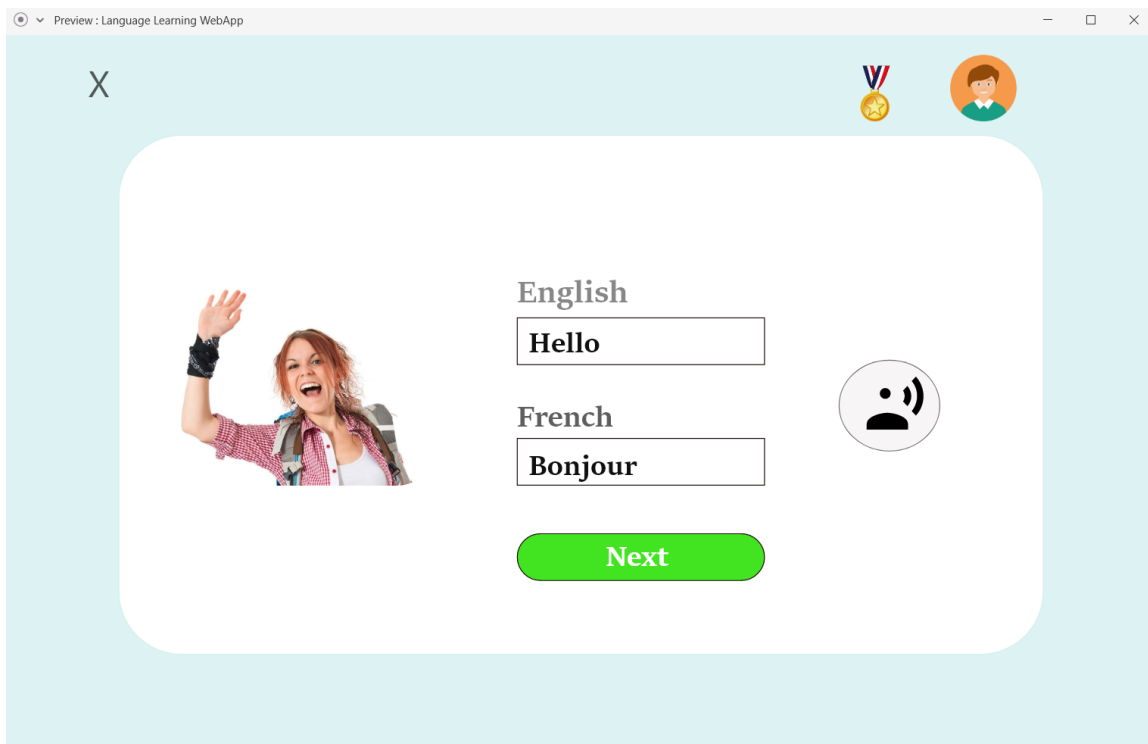
9) The user will then be redirected to his/her dashboard which will have a track of the user's progress , points scored and their profile can be viewed on the navigation bar.



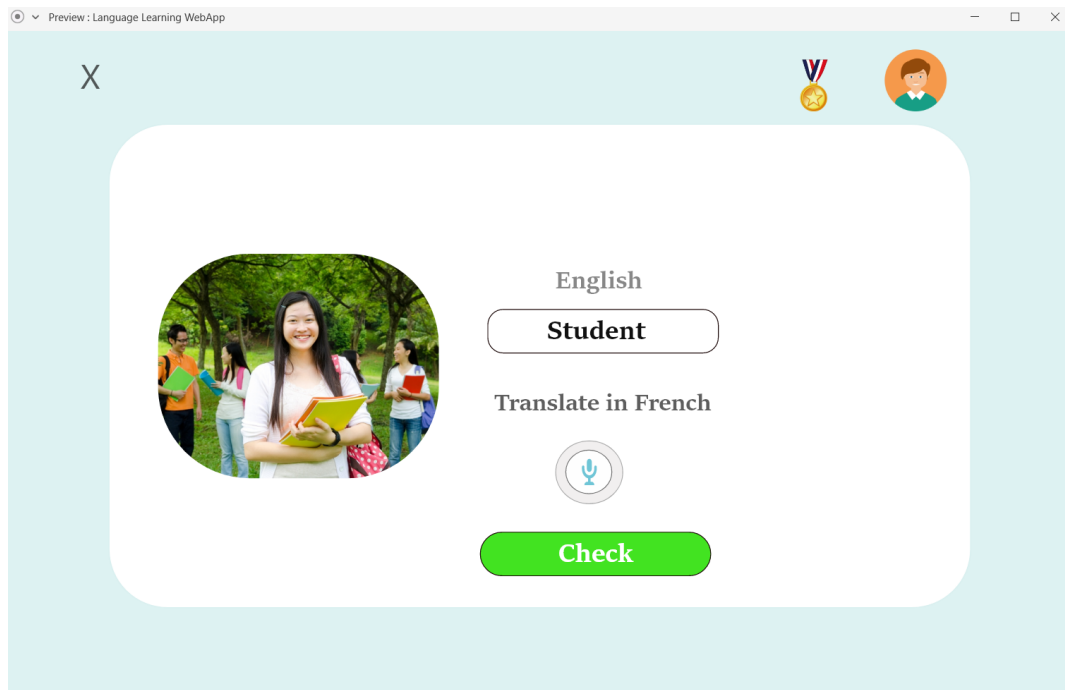
10) Once the user clicks on “Basic” button, the user can access the course and continue with their learning



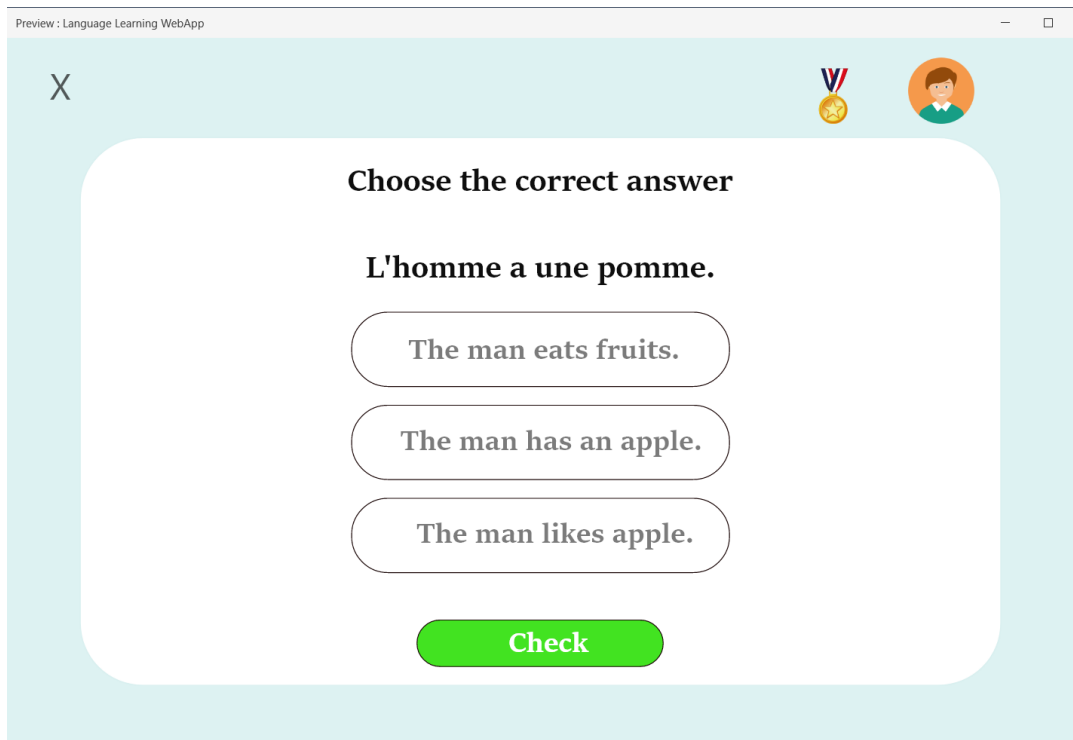
11) Reading section is where users can learn to read a particular language. The word is translated into the language picked by the user and the word is also spelled by the system.



12) The user can access the next page where he/she can listen to each word after translation.



13) After learning the user can click on check where there will be series of assignments as below



14) The assessment consists of different types like MCQs, fill up the blanks etc and these assessments are evaluated by clicking on check.



15) There will be a pop-up that would say how well the user has performed.



16) If the user did well in the assessments the following is the output.



## Access to the prototype

The prototype of the Language app, Lingualift can be accessed using this link:

<https://xd.adobe.com/view/9277c150-c1b4-4dfc-884f-fa1836b12260-c724/>

Please feel free to drop in your feedback on our work.

**JSS ACADEMY OF TECHNICAL EDUCATION**  
**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

Date: 10.1.2021

Faculty Name : N.Snehalatha

Class/Section: III CSE 'B'

Activity Name: Hands-on session with virtual Data Structures lab

Academic Year: 2020-21

**GOAL OF THE ACTIVITY:**

- To enhance the knowledge in Data Structures and its applications.
- To explore the features and implementation of different linear and non linear data structures
- To experience hands-on session.
- To build team work and communication skills.

**DESCRIPTION OF ACTIVITY:**

- A team of 5 students was formed.
- Each team has to do virtual experiments to get the opportunity for learning and better understanding of data structures using algorithms.
- Prepare a report on the list of the experiments , their Objective , experiment results and quiz

**USE OF APPROPRIATE METHODS:**

Virtual labs on Data structures

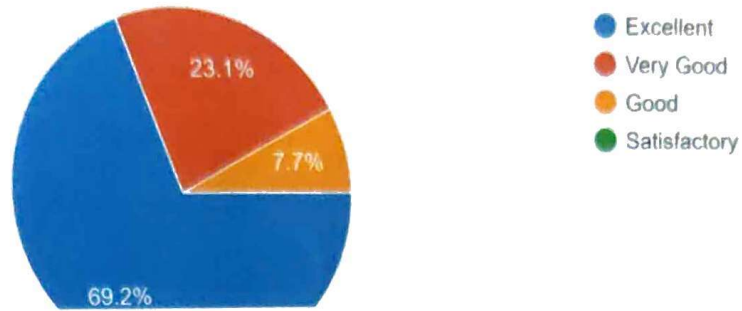
**RESULTS/OUTCOME :**

Students were able to visualize the concepts of Data structures and conduct handson experiment along with quiz

Feedback:

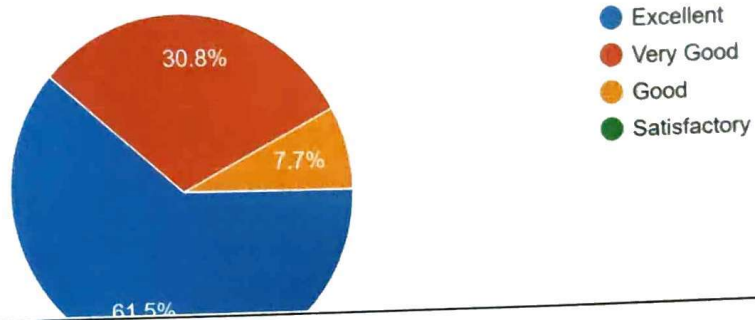
Extent to which the visualization tool- VisuAlgo helped you to enhance the know structures

13 responses



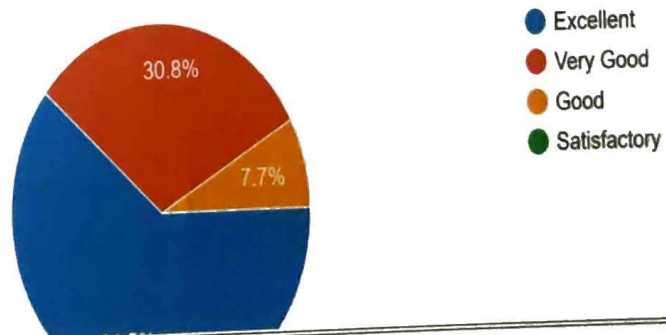
Extent to which the visualization tools helped in improving the knowledge in pro

13 responses



To what extent the virtual lab hands-on session helped in improving the learnin data structures.

13 responses



*Snet*

Signature of the Faculty

*[Signature]*  
HOD, CSE

Faculty Name: N. Snehalatha

Class/Section: VIII CSE 'C'

Activity Name: Ideathon: IoT applications

Academic Year: 2020-21

### GOAL

- Build a community of problem solvers .
- Creates a platform for students to brainstorm and share new innovative ideas
- Identify unique ideas that can make a difference to the broader community
- Provide a platform to develop an active community that will learn from each other and work toward a common goal to raise public awareness about research and reasons to participate in social trials
- Raises awareness of emerging technologies and their impact in different IoT applications

### **About the Activity:**

Team of 4-5 students were formed. Each team were allotted a time of at most 10 minutes for presentation. The presentation was followed by 2 minutes of a Q & A session. Students presented their ideas on different IoT topics with a prototype/Model

### **USE OF APPROPRIATE METHODS:**

- Idea was evaluated on the following main criteria:
- a. Innovation - How 'innovative' is the idea? Is this idea creative and progressive?
- b. Feasibility of Implementation - How feasible is the idea?
- c. Sustainability - How sustainable is the idea? Is this idea cost-effective? Where does this have market potential and scalability?
- d. Impact - How does this idea raise awareness for diverse or vulnerable populations in a range of social conditions



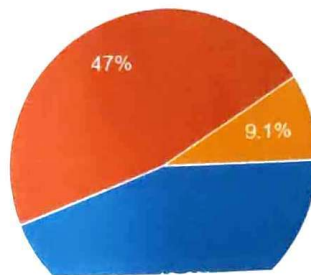
## RESULTS/OUTCOME :

- Students were able to explore and present new innovative ideas in IoT technologies.

### Feedback:

How would you rate this session was informative

66 responses



Whether the session was useful to enhance your subject skills

66 responses

