

# Case Study Submission

## TRAIN BOOKING SYSTEM

*CB.EN.U4CSE19139 -Sneha Nair*

### List of Departments in the case study with the purpose

<b>Department Name</b>	<b>Purpose</b>
Banking department	The department takes care of the payment transitions that take place during the booking or cancellation of online tickets
Tatkal department	The department is used for booking journeys at very short notice
Reservation department	The department is responsible for the booking or cancellation of train tickets
Customer service	The main objective of the department is to help passengers in need. To clarify doubts and help in the reservation process

### GROUP DETAILS

<b>Roll number</b>	<b>Name of the Student</b>	<b>Identified Role</b>
CB.ENU4CSE19121	KESA VARSHITHA	Banking department
CB.EN.U4CSE19135	NDN JAHNAVI	Tatkal department
CB.EN.U4CSE19139	R SNEHA NAIR	Reservation department
CB.EN.U4CSE19148	SITHARA J P	Customer service

## **CASE STUDY**

<b>Topic</b>	<b>Page No</b>	<b>Supporting file name</b>
Problem Statement of department	3	
Why is Networking required?	3	
Network Architecture Diagram	3	
Performance Parameters	4	
Socket programming	6	19139_CN_casestudy_Socket programming
GoBack N and Selective Repeat protocol	20	cb.en.u4cse19139_reservation_server.java cb.en.u4cse19139_reservation_client.java
Solving the IP address allocation process for each subnetwork	23	
Cisco packet tracer – Virtual Local Area Network and OSPF	29	cb.en.u4cse19139 Reservation _ vlan_ospf(pkt
Cisco packet tracer – Virtual Local Area Network and RIP	40	cb.en.u4cse19139 Reservation cs _vlan_rip(pkt
Cloud and Cisco packet tracer	46	cb.en.u4cse19139 Reservation Cloud(pkt
Cloud concepts	48	
Cloud and Networking	49	
How is cloud related to your application?	52	

## **Problem statement**

The Anywhere Any Time Advance Reservation system is the online ticket reserving system where the passengers can reserve the tickets for their travel, cancel the reserved ticket and they can view the status of the ticket before traveling. The reservation option enables a person to reserve a ticket in an easier and efficient manner. Once the train is available the person will be asked to enter his details and his requirements regarding the coach. After this, the reservation database is updated with the personal details, train name, and also the source and destination place, and a unique PNR number is generated and given as the reference number for the user. The View option allows the user to enter the PNR number and get the current details about the ticket. The software ensures the safety and security of details about the passengers, their payment details and ensures the integrity of the data being processed

## **Why is Networking required?**

Presently the population has rocketed to a great extent and hence is the booking in the Railways. About thousands of passengers, each day do bookings and cancel their tickets and reservations. Now through this computer networking system, the workload and pressure of Railway employee's reservation counters have come down to a great extent. Because of this, it is very easy to manage the reservations and ticketing system. It also helps to access many vital things in one go. The passengers will be easily notified about their status, and so less pressure will be on ticket counters. Moreover, this will help passengers to access the websites and book tickets of their need from wherever they want. And this will also eliminate the corruption in reservation systems, like agents who take a huge amount of money for booking tickets.

## **Network Architecture diagram**



## **Some key components of a network management system include:**

- **Network Device Discovery** — the ability to identify what devices are present on a network.
- **Network Device Monitoring** — the ability to monitor at the device level to determine the health of network components and the extent to which their performance matches capacity plans and intra-enterprise service-level agreements (SLAs).
- **Network Performance Analysis** — the ability to track performance indicators such as bandwidth utilization, packet loss, latency, availability, and uptime of routers, switches, and other Simple Network Management Protocol(SNMP) enabled devices.
- **Intelligent Notifications** – the ability to configurable alerts that will respond to specific network scenarios by paging, emailing, calling, or texting a network administrator.
- **Mobile and Cloud Support** – the ability to offer mobile and cloud support are important for the financial industry because users require 24/7 access to their financial data no matter where they are.
- **Integration** – the ability to easily integrate with a variety of technologies in place at the institution and work seamlessly together.
- **Automated Intelligence** – the ability to eliminate the need for IT staff to directly administer challenging and time-consuming tasks such as patch management, anti-malware updates, and reporting. Automating these functions saves time while ensuring all patches are up to date. It also reduces device exposure through server hardening.
- **Centralized Monitoring Console** – should include remote control access and monitoring capabilities.
- **Dual Factor Authentication** — enabling secure log-in to the system.
- **Enhanced Reporting Functions** — featuring reporting based on FFIEC requirements for IT audits.

- **Security services** — to protect the institution servers.

## **Performance Parameters**

### **1) Transmission Time**

Transmission time is a time based on how long it takes to send the signal down the transmission line

Formula:

$$\text{Transmission Time} = \text{Message size} / \text{Bandwidth}$$

### **2) Propagation Time**

It is the time required for a bit to travel from the source to the destination

Formula:

$$\text{Propagation Time} = \text{Distance} / \text{Propagation speed}$$

### **3) Latency**

It is defined as the total time taken for a complete message to arrive at the destination, starting with the time when the first bit of the message is sent out from the source and ending with the time when the last bit of the message is delivered at the destination

Formula:

$$\text{Latency} = \text{Propagation Time} + \text{Transmission Time} + \text{Queuing Time} + \text{Processing Delay}$$

### **4) Queuing Time**

Queuing time is a time based on how long the packet has to sit around in the router

### **5) Throughput**

Throughput is the number of messages successfully transmitted per unit time. It

is controlled by available bandwidth, the available signal-to-noise ratio, and hardware limitations. It is measured in bps, Bps, KBps, MBps, Gbps

Formula:

$$\text{Throughput} = (\text{number of frames per minute} \times \text{number of bits in each frame}) / 60$$

## **6)Processing Delay**

Processing delay is the delay based on how long it takes the router to figure out where to send the packet

## **7)Packet loss**

Packet loss examines how many data packets are dropped during data transmissions on your network. The more data packets that are lost, the longer it takes for a data request to be fulfilled.

## **8)Bandwidth**

Bandwidth determines how rapidly the webserver can upload the requested information. Bandwidth is the maximum data transmission ratepossible on a network “Bandwidth” means “Capacity”

# **Socket Programming concepts**

## **List of operations completed with the File**

1. The customer can book train tickets online and their ticket details are stored
2. The number of tickets available when booked is updated
3. The number of tickets available when canceled are updated
4. Money is refunded and the number of tickets available is updated after cancellation
5. View ticket details with the help of the PNR number
6. Update username and password in case of login and ticket booking issue

## Client-Side program with output for each operation

### SERVER CODE

```
import socket
from _thread import *
import pandas as pd
import datetime
import random
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
ThreadCount = 0
try:
    s.bind((socket.gethostname(), 2000))
except socket.error as e:
    print(str(e))
print('Waiting for a Connection..')
s.listen(5)

def threaded_client(clt):
    while True:
        df = pd.read_csv(r"C:\Users\SNEHA\Downloads\network\archive\train_info.csv")
        df1 = pd.read_csv(r"C:\Users\SNEHA\Downloads\network\archive\train_schedule2
edited.csv")
        df4 = pd.read_csv(r"C:\Users\SNEHA\Downloads\network\archive\bank_details.csv")
        df6 = pd.read_csv(r"C:\Users\SNEHA\Downloads\network\archive\waiting_list.csv")
        df5 =
pd.read_csv(r"C:\Users\SNEHA\Downloads\network\archive\Passenger_info.csv")
        ch = clt.recv(100).decode("utf-8")
        if ch == 'Booking':
            clt.send(bytes("server.csv", "utf-8"))
            source = clt.recv(1000).decode("utf-8")
            destination = clt.recv(1000).decode("utf-8")
            d2 = df.loc[(df['Source_Station_Name'] == source) &
(df['Destination_Station_Name'] == destination)]
            d2.to_csv('info.csv', index=False)
            clt.send(bytes("info.csv", "utf-8"))
            train_no = clt.recv(1000).decode("utf-8")
            coach = clt.recv(50).decode("utf-8")
            no_of_ticket = clt.recv(1000).decode("utf-8")
            if coach == 'Sleeper':
                a = df1.loc[(df1['Station_Name'] == source) & (df1['Train_No'] == int(train_no))][
                    'price_sleeper'].values
            elif coach == '2 - tier':
                a = df1.loc[(df1['Station_Name'] == source) & (df1['Train_No'] ==
int(train_no))]['Price_2-tier'].values
            elif coach == '3 - tier':
                a = df1.loc[(df1['Station_Name'] == source) & (df1['Train_No'] ==
int(train_no))]['Price_3-tier'].values
            else:
```

```

a = df1.loc[(df1['Station_Name'] == source) & (df1['Train_No'] == int(train_no))][
    'price_general'].values
station_code = \
    df1.loc[(df1['Station_Name'] == source) & (df1['Train_No'] == int(train_no))][
        'Station_Code'].values
arrival_time = \
    df1.loc[(df1['Station_Name'] == source) & (df1['Train_No'] == int(train_no))][
        'Arrival_time'].values
departure_Time = \
    df1.loc[(df1['Station_Name'] == source) & (df1['Train_No'] == int(train_no))][
        'Departure_Time'].values
dept_Date = df1.loc[(df1['Station_Name'] == source) & (df1['Train_No'] ==
int(train_no))][
    'departure_DATE'].values
arrival_date = df1.loc[(df1['Station_Name'] == source) & (df1['Train_No'] ==
int(train_no))][
    'arrival_DATE'].values
day = df.loc[
    (df['Source_Station_Name'] == source) & (df['Train_No'] == int(train_no)) & (
        df['Destination_Station_Name'] == destination)][['days']].values
train_name = df.loc[
    (df['Source_Station_Name'] == source) & (df['Train_No'] == int(train_no)) & (
        df['Destination_Station_Name'] == destination)][['Train_Name']].values
station_code = str(station_code).lstrip('\'').rstrip('\'')
arrival_time = str(arrival_time).lstrip('\'').rstrip('\'')
departure_Time = str(departure_Time).lstrip('\'').rstrip('\'')
day = str(day).lstrip('\'').rstrip('\'')
train_name = str(train_name).lstrip('\'').rstrip('\'')
dept_Date = str(dept_Date).lstrip('\'').rstrip('\'')
arrival_date = str(arrival_date).lstrip('\'').rstrip('\'')
d = df1.loc[(df1['Station_Name'] == source) & (df1['Train_No'] ==
int(train_no))][['coach']].values
dt = datetime.date.today()
diff = d - int(no_of_ticket)
diff = str(diff).lstrip('\'').rstrip('\'')
diff = int(diff)
PNR = random.randint(100000, 999999)
number = random.randint(1000, 9999)
if diff >= 0:
    clt.send(bytes("Seats Available", "utf-8"))

    tot_amount = a * int(no_of_ticket)
    t = str(tot_amount).lstrip('\'').rstrip('\'')
    clt.send(bytes("Total amount:", "utf-8"))
    clt.send(t.encode("utf-8"))

    list1 = [[int(train_no), source, station_code, destination, arrival_time,
    departure_Time, day, coach,
    int(no_of_ticket), float(t), train_name, dt, PNR, number, dept_Date,
    arrival_date]]

```

```

df5 = pd.DataFrame(list1,
                   columns=['TRAIN NO', 'STATION NAME', 'STATION CODE',
'DESTINATION', 'ARRIVAL TIME',
'DEPARTURE TIME', 'DAY', 'COACH', 'NO OF TICKETS',
'AMOUNT', 'TRAIN NAME',
'DATE_TIME', 'PNR NUMBER', 'booking id', 'departure_DATE',
'arrival_DATE'])

df5.to_csv("C:/Users/SNEHA/Downloads/network/archive/Passenger_info.csv",
mode='a', header=False,
index=False)

clt.send(str(number).encode("utf-8"))
else:
    clt.send(bytes("No of seats available are: ", "utf-8"))
    x = int(no_of_ticket) + diff
    clt.send(str(x).encode("utf-8"))

z = clt.recv(1000).decode("utf-8")
if z == 'yes':
    clt.send(bytes("yes", "utf-8"))
    tot_amount = a * x
    t = str(tot_amount).lstrip('[').rstrip(']')
    clt.send(bytes("Total amount:", "utf-8"))
    clt.send(t.encode("utf-8"))
    list1 = [
        [int(train_no), source, station_code, destination, arrival_time, departure_Time,
day, coach,
x, float(t), train_name, dt, PNR, number, dept_Date, arrival_date]]

df5 = pd.DataFrame(list1,
                   columns=['TRAIN NO', 'STATION NAME', 'STATION CODE',
'DESTINATION',
'ARRIVAL TIME',
'DEPARTURE TIME', 'DAY', 'COACH', 'NO OF TICKETS',
'AMOUNT',
'TRAIN NAME',
'DATE_TIME', 'PNR NUMBER', 'booking id',
'departure_DATE',
'arrival_DATE'])

df5.to_csv("C:/Users/SNEHA/Downloads/network/archive/Passenger_info.csv",
mode='a',
header=False, index=False)

tot_amount1 = a * abs(diff)
t1 = str(tot_amount1).lstrip('[').rstrip(']')

list2 = [

```

```

    [int(train_no), source, station_code, destination, arrival_time, departure_Time,
day, coach,
     abs(diff), abs(float(t1)), train_name, dt, PNR, number, dept_Date,
arrival_date]]]

df6 = pd.DataFrame(list2, columns=['TRAIN NO', 'STATION NAME',
'STATION CODE', 'DESTINATION',
'ARRIVAL TIME', 'DEPARTURE TIME', 'DAY',
'COACH',
'NO OF TICKETS', 'AMOUNT', 'TRAIN NAME',
'DATE_TIME',
'PNR NUMBER', 'booking id', 'departure_DATE',
'arrival_DATE'])

df6.to_csv("C:/Users/SNEHA/Downloads/network/archive/waiting_list.csv",
mode='a',
header=False, index=False)

clt.send(bytes(
    "Unavailable tickets are enter into waiting list \nwaiting for payment for
available "
    "tickets\nYour booking id",
    "utf-8"))
clt.send(str(number).encode("utf-8"))

else:
    clt.send(bytes("No", "utf-8"))
    tot_amount = a * int(no_of_ticket)
    t = str(tot_amount).lstrip('[').rstrip(']')
    list1 = [
        [int(train_no), source, station_code, destination, arrival_time, departure_Time,
day, coach,
         int(no_of_ticket), float(t), train_name, dt, PNR, number, dept_Date,
arrival_date]]]

df6 = pd.DataFrame(list1, columns=['TRAIN NO', 'STATION NAME',
'STATION CODE', 'DESTINATION',
'ARRIVAL TIME', 'DEPARTURE TIME', 'DAY',
'COACH',
'NO OF TICKETS', 'AMOUNT', 'TRAIN NAME',
'DATE_TIME',
'PNR NUMBER', 'booking id', 'departure_DATE',
'arrival_DATE'])

df6.to_csv("C:/Users/SNEHA/Downloads/network/archive/waiting_list.csv",
mode='a',
header=False, index=False)

clt.send(bytes("Ticket entered in waiting list", "utf-8"))

elif ch == 'Cancellation':
    clt.send(bytes("Enter the PNR NUMBER: ", "utf-8"))

```

```

pnr_no = clt.recv(1000).decode("utf-8")
pnr_no = int(pnr_no)
if pnr_no in df5.values:
    clt.send(bytes("Cancel ticket", "utf-8"))
    if clt.recv(1000).decode("utf-8") == 'yes':
        clt.send(bytes("yes", "utf-8"))
        no_ticket = df5.loc[(df5['PNR NUMBER'] == pnr_no)]['NO OF
TICKETS'].values
        CLASS = df5.loc[(df5['PNR NUMBER'] == pnr_no)]['COACH'].values
        price = df5.loc[(df5['PNR NUMBER'] == pnr_no)]['AMOUNT'].values
        train = df5.loc[(df5['PNR NUMBER'] == pnr_no)]['TRAIN NO'].values
        source_c = df5.loc[(df5['PNR NUMBER'] == pnr_no)]['STATION
NAME'].values

        source_c = str(source_c).lstrip('[').rstrip(']')
        train = str(train).lstrip('[').rstrip(']')
        price = str(price).lstrip('[').rstrip(']')
        no_ticket = str(no_ticket).lstrip('[').rstrip(']')
        CLASS = str(CLASS).lstrip('[').rstrip(']')

        price = float(price)
        tot_price = price * 0.95

        df5.loc[(df5['PNR NUMBER'] == pnr_no), 'AMOUNT'] = float(tot_price)
        df5.loc[(df5['PNR NUMBER'] == pnr_no), 'Cancelled'] = 'yes'
        df5.to_csv("C:/Users/SNEHA/Downloads/network/archive/Passenger_info.csv",
mode='w', index=False)

        d = df1.loc[(df1['Station_Name'] == source_c) & (df1['Train_No'] ==
int(train))][
            CLASS].values

        d = str(d).lstrip('[').rstrip(']')
        df1.loc[(df1['Station_Name'] == source_c) & (df1['Train_No'] == int(train)),
CLASS] = int(d) + int(
            no_ticket)

        df1.to_csv("C:/Users/SNEHA/Downloads/network/archive/train_schedule2
edited.csv", mode='w',
            index=False)

        clt.send(bytes("Ticket cancellation successful", "utf-8"))
    else:
        clt.send(bytes("no", "utf-8"))
    else:
        clt.send(bytes("Invalid PNR NUMBER", "utf-8"))
elif ch == 'VIEW':
    clt.send(bytes("C:/Users/SNEHA/Downloads/network/archive/Passenger_info.csv",
"utf-8"))
    clt.close()

```

```

while True:
    client, adr = s.accept()
    print(f"Connection established to {adr} established")
    start_new_thread(threaded_client, (client,))
    ThreadCount += 1
    print('Thread Number: ' + str(ThreadCount))

s.close()

```

## OUTPUT

```

C:\Windows\system32\cmd.exe - py Booking_Cancel_Server.py
Microsoft Windows [Version 10.0.19043.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SNEHA>cd PycharmProjects\pythonProject1

C:\Users\SNEHA\PycharmProjects\pythonProject1>py Booking_Cancel_Server.py
Waiting for a Connection..
Connection established to ('192.168.51.179', 12813) established
Thread Number: 1
Connection established to ('192.168.51.179', 12815) established
Thread Number: 2
Connection established to ('192.168.51.179', 12818) established
Thread Number: 3
Connection established to ('192.168.51.179', 12830) established
Thread Number: 4

```

## CLIENT 1

```

import socket
import pandas as pd
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
print('Waiting for connection')
try:
    s.connect((socket.gethostname(), 2000))
except socket.error as e:
    print(str(e))
df = pd.read_csv(r"C:\Users\SNEHA\Downloads\network\archive\train_schedule2
edited.csv")
s.send(bytes("Booking", "utf-8"))
msg = s.recv(665285).decode("utf-8")
print(pd.read_csv(msg))
source = input("Enter source: ")
s.send(source.encode("utf-8"))
destination = input("Enter destination: ")
s.send(destination.encode("utf-8"))
print(pd.read_csv(s.recv(665285).decode("utf-8")))
train_no = int(input("Enter the train number: "))
s.send(str(train_no).encode("utf-8"))
d = df.loc[(df['Station_Name'] == source) & (df['Train_No'] == train_no)]
print(d)

```

```

coach = input("Enter coach number: ")
s.send(coach.encode("utf-8"))
no_ticket = int(input("Enter number of ticket: "))
s.send(str(no_ticket).encode("utf-8"))
msg2 = s.recv(1000).decode("utf-8")
if msg2 == 'Seats Available':
    print(s.recv(1000).decode("utf-8"), end=' ')
    print(s.recv(1000).decode("utf-8"))
    print("Waiting for payment\nYour booking id is:")
    print(s.recv(1000).decode("utf-8"))

else:
    print(msg2, end=' ')
    print(s.recv(2000).decode("utf-8"))
    choice = input("Proceed with booking the tickets available\nyes or no?: ")
    s.send(choice.encode("utf-8"))
    m = s.recv(1000).decode("utf-8")
    if m == 'yes':
        print(s.recv(1000).decode("utf-8"))
        print(s.recv(1000).decode("utf-8"))
        print(s.recv(1000).decode("utf-8"))
        print(s.recv(1000).decode("utf-8"))
    else:
        print(s.recv(1000).decode("utf-8"))

s.close()

```

## OUTPUT

### Ticket booking with seats available

```

C:\Users\SNEHA>cd PycharmProjects\pythonProject1
C:\Users\SNEHA>PycharmProjects\pythonProject1>py railway_booking_client.py
Waiting for connection
      Source_Station_Name      Destination_Station_Name
0       SAWANTWADI ROAD          MADGAON JN.
1       MADGAON JN.           SAWANTWADI ROAD
2       MADGAON JN.           CHHATRAPATI SHASHI MAHARAJ TERMINUS
3       DELHI-SAFDAR JANG          DELHI-SAFDAR JANG
4       AURANGABAD             VARANASI JN.
...
11108      PUNE JN.            TALEGAON
11109      TALEGAON            SHIVAJINAGAR
11110      PUNE JN.            TALEGAON
11111      TALEGAON            PUNE JN.
11112      PUNE JN.            TALEGAON
[11113 rows x 2 columns]
Enter source: TALEGAON
Enter destination: PUNE JN.
      Train_No  Train_Name Source_Station_Name Destination_Station_Name      days
0      99903   TGN-PUNE EMU      TALEGAON            PUNE JN. Wednesday
1      99907     EMU          TALEGAON            PUNE JN. Thursday
Enter the train number: 99907
      SN Train_No Station_Code Sleeper 2-tier 3-tier General_Station_Name price_sleeper Arrival_time Departure_Time Proce_3-tier Price_2-tier price_general departure_DATE arrival_DATE
285  1   99907      TGN    104    100    100    TALEGAON      300  16:38:00  16:38:00      1350      2200      160.0  25-03-2021  25-03-2021
Enter coach number: 3-tier
Enter number of ticket: 3
Total amount: 480
Waiting for payment
Your booking id is:
1859

```

## After insertion

TRAIN NO	STATION N	STATION C	DESTINATI	ARRIVAL T	DEPARTURE	DAY	COACH	NO OF TIC	AMOUNT	TRAIN N	DATE	TIME	PNR NUM	booking id	departure_DATE	arrival_DATE	ACCOUNT NAME	PHONE NC	Cancelled	
1	99908	PUNE JN.	PU	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	11	2244	EMU	14-08-2021	675035	7847	14-08-2021	14-08-2021	1.92E+08	Sneha Nair	9.46E+09	No
2	99908	PUNE JN.	PU	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	2	408	EMU	14-08-2021	935169	7531	15-08-2021	15-08-2021	9.23E+08	Riya Jain	7.46E+09	No
4	99908	PUNE JN.	PU	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	2	408	EMU	14-08-2021	135988	1295	16-08-2021	16-08-2021	1.29E+08	Sithara JP	8.96E+09	No
5	99908	PUNE JN.	PU	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	2	408	EMU	14-08-2021	621782	2428	17-08-2021	17-08-2021	1.23E+08	Varshitha I	8.8E+09	No
6	99908	PUNE JN.	PU	TALEGAON	20:00:00	00:00:00	Sunday	2-tier	2	208	EMU	14-08-2021	451961	7413	18-08-2021	18-08-2021	1.92E+08	Sneha Nair	9.46E+09	No
7	99908	PUNE JN.	PU	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	3	612	EMU	14-08-2021	830771	4194	19-08-2021	19-08-2021	8.21E+08	Harry style	9.86E+09	No
8	99908	PUNE JN.	PU	TALEGAON	20:00:00	00:00:00	Sunday	3-tier	4	416	EMU	14-08-2021	584726	4771	20-08-2021	20-08-2021	1.23E+08	Varshitha I	8.8E+09	No
9	99908	PUNE JN.	PU	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	3	612	EMU	14-08-2021	101160	4693	21-08-2021	21-08-2021	7.29E+08	NDN Jhan	8.97E+09	No
10	99908	PUNE JN.	PU	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	3	612	EMU	14-08-2021	597275	4876	22-08-2021	22-08-2021	1.92E+08	Sneha Nair	9.46E+09	No
11	99908	PUNE JN.	PU	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	11	2244	EMU	14-08-2021	957786	2140	23-08-2021	23-08-2021	8.21E+08	Harry style	9.86E+09	No
12	99908	PUNE JN.	PU	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	3	612	EMU	14-08-2021	715840	5992	24-08-2021	24-08-2021	1.29E+08	Sithara JP	8.96E+09	No
13	99908	PUNE JN.	PU	TALEGAON	20:00:00	00:00:00	Sunday	2-tier	3	312	EMU	14-08-2021	486475	7216	25-08-2021	25-08-2021	9.23E+08	Riya Jain	7.46E+09	No
14	99908	PUNE JN.	PU	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	3	612	EMU	14-08-2021	127007	7250	26-08-2021	26-08-2021	1.92E+08	Sneha Nair	9.46E+09	No
15	99907	TALEGAON	TGN	PUNE JN.	16:38:00	16:38:00	Thursday	Sleeper	4	486.4	EMU	14-08-2021	739782	6057	25-03-2021	25-03-2021	1.92E+08	Sneha Nair	9.46E+09	yes
16	99908	PUNE JN.	PU	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	5	1020	EMU	14-08-2021	513322	9145	27-08-2021	27-08-2021				
17	99907	TALEGAON	TGN	PUNE JN.	16:38:00	16:38:00	Thursday	3-tier	3	480	EMU	14-08-2021	443326	1859	25-03-2021	25-03-2021				
18																				
19																				
20																				

**Ticket booking with few seats available and others are added to the waiting list with choice**

```
C:\Users\SNEHA\PycharmProjects\pythonProject>py railway_booking_client.py
Waiting for connection
Source_Station Name          Destination_Station Name
0      SAWANTWADI ROAD           MADGOAN JN.
1      MADGOAN JN.              SAWANTWADI ROAD
2      MADGOAN JN. CHHATRAPATI SHAHU MAHARAJ TERMINUS
3      DELHI-SAFDAR JANG        DELHI-SAFDAR JANG
4      AURANGABAD              VARANASI JN.
...
...
11108     PUNE JN.                TALEGAON
11109     TALEGAON               SHIVAJINAGAR
11110     PUNE JN.                TALEGAON
11111     TALEGAON               PUNE JN.
11112     PUNE JN.                TALEGAON
[11113 rows x 2 columns]
Enter source: PUNE JN.
Enter destination: TALEGAON
Train No  Train_Name Source_Station Name Destination_Station Name    days
0      99904  PUNE-TGN EMU      PUNE JN.          TALEGAON   Tuesday
1      99906  EMU                 PUNE JN.          TALEGAON   Wednesday
2      99908  EMU                 PUNE JN.          TALEGAON   Sunday
Enter the train number: 99908
SN Train No Station Code Sleeper 2-tier 3-tier General Station Name price_sleeper Arrival_time Departure_Time Proce_3-tier Price_2-tier price_general departure_DATE arrival_DATE
292 1  99908  PUNE 136 5 100 100 PUNE JN. 204 19:18:00 19:19:00 2100 2100 104.0 28-03-2021 28-03-2021
Enter coach number: 2-tier
Enter number of ticket: 7
No of seats available are: 5
Proceed with booking the tickets available
yes or no?: yes
Total amount:
520
Unavailable tickets are enter into waiting list
waiting for payment for available tickets
Your booking id
6266
```

## After insertion – passenger info

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
10	99908	PUNE JN.	PUNE	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	3	612	EMU	14-08-2021	597275	4876	22-08-2021	22-08-2021	1.92E+08	Sneha Nair	9.46E+09	No		
11	99908	PUNE JN.	PUNE	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	11	2244	EMU	14-08-2021	957786	2140	23-08-2021	23-08-2021	8.21E+08	Harry style	9.86E+09	No		
12	99908	PUNE JN.	PUNE	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	3	612	EMU	14-08-2021	715840	5992	24-08-2021	24-08-2021	1.29E+08	Sithara JP	8.96E+09	No		
13	99908	PUNE JN.	PUNE	TALEGAON	20:00:00	00:00:00	Sunday	2-tier	3	312	EMU	14-08-2021	486475	7216	25-08-2021	25-08-2021	9.23E+08	Riya Jain	7.46E+09	No		
14	99908	PUNE JN.	PUNE	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	3	612	EMU	14-08-2021	127007	7250	26-08-2021	26-08-2021	1.92E+08	Sneha Nair	9.46E+09	No		
15	99907	TALEGAON	TGN	PUNE JN.	16:38:00	16:38:00	Thursday	Sleeper	4	486.4	EMU	14-08-2021	739782	6057	25-03-2021	25-03-2021	1.92E+08	Sneha Nair	9.46E+09	yes		
16	99908	PUNE JN.	PUNE	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	5	1020	EMU	14-08-2021	513322	9145	27-08-2021	27-08-2021						
17	99907	TALEGAON	TGN	PUNE JN.	16:38:00	16:38:00	Thursday	3-tier	3	480	EMU	14-08-2021	443326	1859	25-03-2021	25-03-2021						
18	99908	PUNE JN.	PUNE	TALEGAON	19:18:00	19:19:00	Sunday	2-tier	5	520	EMU	14-08-2021	703035	6266	28-03-2021	28-03-2021						
19																						
20																						
21																						

## After insertion- waiting list

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	TRAIN NO	STATION N	STATION C	DESTINATI	ARRIVAL T	DEPARTU	FDAY	COACH	NO OF TIC	AMOUNT	TRAIN NA	DATE _TIME	PNR NUM	booking id	departure _C	arrival DATI	ACCOUNT NAME	PHONE NO				
2	99908	PUNE JN.	PUNE	TALEGAON	00:00:00	23:00:00	Sunday	Sleeper	5	1020	EMU	11-08-2021	637966	7234	14-08-2021	14-08-2021	1.92E+08	Sneha Nair	9.46E+09			
3	99908	PUNE JN.	PUNE	TALEGAON	00:00:00	23:00:00	Sunday	Sleeper	4	816	EMU	12-08-2021	637966	7234	15-08-2021	15-08-2021	9.23E+08	Riya Jain	7.46E+09			
4	99908	PUNE JN.	PUNE	TALEGAON	20:00:00	00:00:00	Sunday	2-tier	2	208	EMU	13-08-2021	637966	7234	16-08-2021	16-08-2021	1.29E+08	Sithara JP	8.96E+09			
5	99908	PUNE JN.	PUNE	TALEGAON	20:00:00	00:00:00	Sunday	3-tier	104	10816	EMU	13-08-2021	637966	7234	17-08-2021	17-08-2021	1.23E+08	Varshitha I	8.8E+09			
6	99908	PUNE JN.	PUNE	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	1	204	EMU	14-08-2021	637966	7847	18-08-2021	18-08-2021	1.92E+08	Sneha Nair	9.46E+09			
7	99908	PUNE JN.	PUNE	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	3	612	EMU	14-08-2021	637966	2140	19-08-2021	19-08-2021	8.21E+08	Harry style	9.86E+09			
8	99908	PUNE JN.	PUNE	TALEGAON	19:18:00	19:19:00	Sunday	2-tier	2	208	EMU	14-08-2021	703035	6266	28-03-2021	28-03-2021						
9																						
10																						
11																						

## Ticket booking – Entered no of seats unavailable so added to waiting list with choice

```
C:\Users\SNEHA\PycharmProjects\pythonProject1>py railway_booking_client.py
Waiting for connection
Source_Station_Name Destination_Station_Name
0 SAMANTWADI ROAD MADGAON JN.
1 MADGAON JN. SAMANTWADI ROAD
2 MADGAON JN. CHHATRAPATI SHASHI MAHARAJ TERMINUS
3 DELHI-SAFDAR JANG DELHI-SAFDAR JANG
4 AURANGABAD VARANASI JN.
...
...
11108 PUNE JN. TALEGAON
11109 TALEGAON SHIVAJINAGAR
11110 PUNE JN. TALEGAON
11111 TALEGAON PUNE JN.
11112 PUNE JN. TALEGAON

[11113 rows x 2 columns]
Enter source: TALEGAON
Enter destination: PUNE JN.
Train No Train Name Source_Station_Name Destination_Station_Name days
0 99903 TGN-PUNE EMU TALEGAON PUNE JN. Wednesday
1 99907 EMU TALEGAON PUNE JN. Thursday
Enter the train number: 99907
SN Train_No Station_Code Sleeper 2-tier 3-tier General Station_Name price_sleeper Arrival_time Departure_Time Proce_3-tier Price_2-tier price_general departure_DATE arrival_DATE
285 1 99907 TGN 108 100 100 TALEGAON 300 16:38:00 16:38:00 1350 2200 160.0 25-03-2021 25-03-2021
Enter coach number: General
Enter number of ticket: 102
No of seats available are: 100
Proceed with booking the tickets available
yes or no?: no
Ticket entered in waiting list
```

## After insertion- waiting list

1	TRAIN NO	STATION IN	STATION OUT	ARRIVAL TIME	DEPARTURE TIME	DAY	COACH	NO OF TIC	AMOUNT	TRAIN NAI	DATE	TIME	PNR NUM	booking id	departure_DATE	arrival_DATE	ACCOUNT NAME	PHONE NO
2	99908	PUNE JN.	PUNE	TALEGAON	00:00:00	23:00:00	Sunday	Sleeper	5	1020	EMU	11-08-2021	637966	7234	14-08-2021	14-08-2021	1.92E+08	Sneha Nair 9.46E+09
3	99908	PUNE JN.	PUNE	TALEGAON	00:00:00	23:00:00	Sunday	Sleeper	4	816	EMU	12-08-2021	637966	7234	15-08-2021	15-08-2021	9.23E+08	Riya Jain 7.46E+09
4	99908	PUNE JN.	PUNE	TALEGAON	20:00:00	00:00:00	Sunday	2-tier	2	208	EMU	13-08-2021	637966	7234	16-08-2021	16-08-2021	1.29E+08	Sithara JP 8.96E+09
5	99908	PUNE JN.	PUNE	TALEGAON	20:00:00	00:00:00	Sunday	3-tier	104	10816	EMU	13-08-2021	637966	7234	17-08-2021	17-08-2021	1.23E+08	Varshitha I 8.8E+09
6	99908	PUNE JN.	PUNE	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	1	204	EMU	14-08-2021	637966	7847	18-08-2021	18-08-2021	1.92E+08	Sneha Nair 9.46E+09
7	99908	PUNE JN.	PUNE	TALEGAON	20:00:00	00:00:00	Sunday	Sleeper	3	612	EMU	14-08-2021	637966	2140	19-08-2021	19-08-2021	8.21E+08	Harry style 9.86E+09
8	99908	PUNE JN.	PUNE	TALEGAON	19:18:00	19:19:00	Sunday	2-tier	2	208	EMU	14-08-2021	703035	6266	28-03-2021	28-03-2021		
9	99907	TALEGAON	TGN	PUNE JN.	16:38:00	16:38:00	Thursday	General	102	16320	EMU	14-08-2021	480277	7623	25-03-2021	25-03-2021		
10																		
11																		
12																		
13																		
14																		
15																		

## CLIENT 2 -TICKET CANCELLATION

```

import socket
import pandas as pd
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
print('Waiting for connection')
try:
    s.connect((socket.gethostname(), 2000))
except socket.error as e:
    print(str(e))

df = pd.read_csv(r'C:\Users\SNEHA\Downloads\network\archive\train_schedule2
edited.csv")
s.send(bytes("Cancellation", "utf-8"))
pnr = int(input(s.recv(665285).decode("utf-8")))
s.send(str(pnr).encode("utf-8"))
msg9 = s.recv(1000).decode("utf-8")
if msg9 == 'Cancel ticket':
    c1 = input("Cancel ticket??\nyes or no: ")
    s.send(c1.encode("utf-8"))
    msg1 = s.recv(1000).decode("utf-8")
    if msg1 == 'yes':
        print(s.recv(1000).decode("utf-8"))
    else:
        exit()
else:
    print(msg9)
    exit()
s.close()

```

## OUTPUT

**Ticket cancellation – 95% of the money is refunded and the ticket count is increased and cancelled column is assigned with value Yes**

```
C:\windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SNEHA>cd PycharmProjects\pythonProject1

C:\Users\SNEHA\PycharmProjects\pythonProject1>py Railway_Cancel_Client.py
Waiting for connection
Enter the PNR NUMBER: 739782
Cancel ticket??
yes or no: yes
Ticket cancellation successful

C:\Users\SNEHA\PycharmProjects\pythonProject1>
```

## CONTENT BEFORE CANCELLATION

Passenger_info - Excel																			
File	Home	Insert	Page Layout	Formulas	Data	Review	View	Help	?	Tell me what you want to do									
R. Sneha Nair - [CB.ENU.ACSE19139]																			
1	TRAIN NO	STATION I	STATION II	DESTINATI	ARRIVAL T	DEPARTURE	UF DAY	COACH	NO OF TIC	AMOUNT	TRAIN NAI	DATE _TIME	PNR NUM	booking id	departure _arrival	DATI	ACCOUNT NAME	PHONE NC	Cancelled
2	99908	PUNE JN.	PUNE	TALEGAO	20:00:00	00:00:00	Sunday	Sleeper	11	2244 EMU	14-08-2021	675035	7847	14-08-2021	14-08-2021	1.92E+08	Sneha Nali	9.46E+09	No
3	99908	PUNE JN.	PUNE	TALEGAO	20:00:00	00:00:00	Sunday	Sleeper	2	408 EMU	14-08-2021	935169	7531	15-08-2021	15-08-2021	9.23E+08	Riya Jain	7.46E+09	No
4	99908	PUNE JN.	PUNE	TALEGAO	20:00:00	00:00:00	Sunday	Sleeper	2	408 EMU	14-08-2021	135988	1299	16-08-2021	16-08-2021	1.29E+08	Sithara JP	8.96E+09	No
5	99908	PUNE JN.	PUNE	TALEGAO	20:00:00	00:00:00	Sunday	Sleeper	2	408 EMU	14-08-2021	621782	2428	17-08-2021	17-08-2021	1.23E+08	Varshitha I	8.8E+09	No
6	99908	PUNE JN.	PUNE	TALEGAO	20:00:00	00:00:00	Sunday	2-tier	2	208 EMU	14-08-2021	451961	7413	18-08-2021	18-08-2022	1.92E+08	Sneha Nali	9.46E+09	No
7	99908	PUNE JN.	PUNE	TALEGAO	20:00:00	00:00:00	Sunday	Sleeper	3	612 EMU	14-08-2021	830771	4199	19-08-2021	19-08-2021	8.21E+08	Harry style	9.86E+09	No
8	99908	PUNE JN.	PUNE	TALEGAO	20:00:00	00:00:00	Sunday	3-tier	4	416 EMU	14-08-2021	584726	4771	20-08-2021	20-08-2021	1.23E+08	Varshitha I	8.8E+09	No
9	99908	PUNE JN.	PUNE	TALEGAO	20:00:00	00:00:00	Sunday	Sleeper	3	612 EMU	14-08-2021	101160	4693	21-08-2021	21-08-2021	7.29E+08	NDH Jham	8.97E+09	No
10	99908	PUNE JN.	PUNE	TALEGAO	20:00:00	00:00:00	Sunday	Sleeper	3	612 EMU	14-08-2021	597275	4876	22-08-2021	22-08-2021	1.92E+08	Sneha Nali	9.46E+09	No
11	99908	PUNE JN.	PUNE	TALEGAO	20:00:00	00:00:00	Sunday	Sleeper	11	2244 EMU	14-08-2021	957786	2140	23-08-2021	23-08-2021	8.21E+08	Harry style	9.86E+09	No
12	99908	PUNE JN.	PUNE	TALEGAO	20:00:00	00:00:00	Sunday	Sleeper	3	612 EMU	14-08-2021	715840	5999	24-08-2021	24-08-2021	1.29E+08	Sithara JP	8.96E+09	No
13	99908	PUNE JN.	PUNE	TALEGAO	20:00:00	00:00:00	Sunday	2-tier	3	312 EMU	14-08-2021	486475	7219	25-08-2021	25-08-2021	9.23E+08	Riya Jain	7.46E+09	No
14	99908	PUNE JN.	PUNE	TALEGAO	20:00:00	00:00:00	Sunday	Sleeper	3	612 EMU	14-08-2021	127007	7259	26-08-2021	26-08-2021	1.92E+08	Sneha Nali	9.46E+09	No
15	99907	TALEGAO TGN	PUNE JN.		16:38:00	00:00:00	Thursday	Sleeper	4	512 EMU	14-08-2021	739782	6057	25-03-2021	25-03-2021	1.92E+08	Sneha Nali	9.46E+09	No
16	99908	PUNE JN.	PUNE	TALEGAO	20:00:00	00:00:00	Sunday	Sleeper	5	1020 EMU	14-08-2021	513322	9145	27-08-2021	27-08-2021				
17																			
18																			
19																			
20																			
21																			

## AFTER CANCELLATION

Passenger\_info - Excel

R. Sneha Nair - (C.B.EN.U4CSE19139)

**Possible Data Loss:** Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

V26

1	TRAIN NO	STATION	STATION	DESTINATI	ARRIVAL T	DEPARTU	DAY	COACH	NO OF TIC	AMOUNT	TRAIN NAI	DATE_ TIME	PNR NUM	booking id	departure _Arrival	DATI	ACCOUNT NAME	PHONE NC	Cancelled	
2	99908	PUNE JN.	PUNE	TALEGAON	20:00:0	00:00:0	Sunday	Sleeper	11	2244	EMU	14-08-2021	675035	7847	14-08-2021	14-08-2021	1.92E+08	Sneha Nair	9.46E+09	No
3	99908	PUNE JN.	PUNE	TALEGAON	20:00:0	00:00:0	Sunday	Sleeper	2	408	EMU	14-08-2021	935169	7511	15-08-2021	15-08-2021	9.23E+08	Riya Jain	7.46E+09	No
4	99908	PUNE JN.	PUNE	TALEGAON	20:00:0	00:00:0	Sunday	Sleeper	2	408	EMU	14-08-2021	135988	1295	16-08-2021	16-08-2021	1.29E+08	Sithara JP	8.96E+09	No
5	99908	PUNE JN.	PUNE	TALEGAON	20:00:0	00:00:0	Sunday	Sleeper	2	408	EMU	14-08-2021	621782	2428	17-08-2021	17-08-2021	1.23E+08	Varsitha I	8.8E+09	No
6	99908	PUNE JN.	PUNE	TALEGAON	20:00:0	00:00:0	Sunday	2-tier	2	208	EMU	14-08-2021	451961	7413	18-08-2021	18-08-2021	1.92E+08	Sneha Nair	9.46E+09	No
7	99908	PUNE JN.	PUNE	TALEGAON	20:00:0	00:00:0	Sunday	Sleeper	3	612	EMU	14-08-2021	830771	4191	19-08-2021	19-08-2021	8.21E+08	Harry style	9.86E+09	No
8	99908	PUNE JN.	PUNE	TALEGAON	20:00:0	00:00:0	Sunday	3-tier	4	416	EMU	14-08-2021	584726	4771	20-08-2021	20-08-2021	1.23E+08	Varsitha I	8.8E+09	No
9	99908	PUNE JN.	PUNE	TALEGAON	20:00:0	00:00:0	Sunday	Sleeper	3	612	EMU	14-08-2021	101160	4693	21-08-2021	21-08-2021	7.29E+08	NDN Jham	8.97E+09	No
10	99908	PUNE JN.	PUNE	TALEGAON	20:00:0	00:00:0	Sunday	Sleeper	3	612	EMU	14-08-2021	597275	4876	22-08-2021	22-08-2021	1.92E+08	Sneha Nair	9.46E+09	No
11	99908	PUNE JN.	PUNE	TALEGAON	20:00:0	00:00:0	Sunday	Sleeper	11	2244	EMU	14-08-2021	957786	2140	23-08-2021	23-08-2021	8.21E+08	Harry style	9.86E+09	No
12	99908	PUNE JN.	PUNE	TALEGAON	20:00:0	00:00:0	Sunday	Sleeper	3	612	EMU	14-08-2021	715840	5992	24-08-2021	24-08-2021	1.29E+08	Sithara JP	8.96E+09	No
13	99908	PUNE JN.	PUNE	TALEGAON	20:00:0	00:00:0	Sunday	2-tier	3	312	EMU	14-08-2021	486475	7218	25-08-2021	25-08-2021	9.23E+08	Riya Jain	7.46E+09	No
14	99908	PUNE JN.	PUNE	TALEGAON	20:00:0	00:00:0	Sunday	Sleeper	3	612	EMU	14-08-2021	830771	4191	19-08-2021	19-08-2021	8.21E+08	Harry style	9.86E+09	No
15	99907	TALEGAON	TGN	PUNE JN.	16:38:00	16:38:00	Thursday	Sleeper	4	486.4	EMU	14-08-2021	739782	6057	25-03-2021	25-03-2021	1.92E+08	Sneha Nair	9.46E+09	Yes
16	99908	PUNE JN.	PUNE	TALEGAON	20:00:0	00:00:0	Sunday	Sleeper	5	1020	EMU	14-08-2021	513322	9145	27-08-2021	27-08-2021				
17																				
18																				
19																				
20																				

## NO OF TICKETS BEFORE CANCELLING

Passenger\_info - Excel

R. Sneha Nair - (C.B.EN.U4CSE19139)

**Possible Data Loss:** Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

D287

280	1	99906	PUNE	100	100	100	100	PUNE JN.	450	15:40:00	15:40:00	2100	1200	176	#####	#####	#####	#####
281	2	99906	SVJR	110	108	106	104	SHIVAJINA	30	15:44:00	15:45:00	2200	1400	188	#####	#####	#####	#####
282	8	99906	AKRD	195	176	157	138	AKURDI	100	16:09:00	16:10:00	1250	2000	108	#####	#####	#####	#####
283	9	99906	DEHR	220	196	172	148	DEHU RO/	90	16:14:00	16:15:00	1900	2400	116	#####	#####	#####	#####
284	10	99906	BGWI	240	212	184	156	BEGDAEW	150	16:18:00	16:19:00	1500	2800	124	#####	#####	#####	#####
285	11	99906	GRWD	255	224	193	162	GHORAWI	200	16:21:00	16:22:00	1200	2100	140	#####	#####	#####	#####
286	12	99906	TGN	270	236	202	168	TALEGAON	120	16:30:00	16:30:00	1250	220	152	#####	#####	#####	#####
287	1	99907	TGN	100	100	100	100	TALEGAON	300	16:38:00	16:38:00	1350	2200	160	#####	#####	#####	#####
288	2	99907	GRWD	115	112	109	106	GHORAWI	250	16:40:00	16:41:00	1450	2500	168	#####	#####	#####	#####
289	3	99907	BGWI	130	124	118	112	BEGDAEW	108	16:43:00	16:44:00	1600	1200	188	#####	#####	#####	#####
290	9	99907	DAPD	230	204	178	152	DAPODI	168	17:07:00	17:08:00	1500	1400	220	#####	#####	#####	#####
291	10	99907	KK	240	212	184	156	KHADKI	188	17:12:00	17:13:00	1500	2000	224	#####	#####	#####	#####
292	11	99907	SVJR	255	224	193	162	SHIVAJINA	196	17:18:00	17:19:00	1500	2020	232	#####	#####	#####	#####
293	12	99907	PUNE	270	10	202	168	PUNE JN.	200	18:18:00	18:19:00	1800	1800	100	#####	#####	#####	#####

## NUMBER OF TICKETS AFTER CANCELLING

Passenger\_info - Excel

R. Sneha Nair - (C.B.EN.U4CSE19139)

**Possible Data Loss:** Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

D287

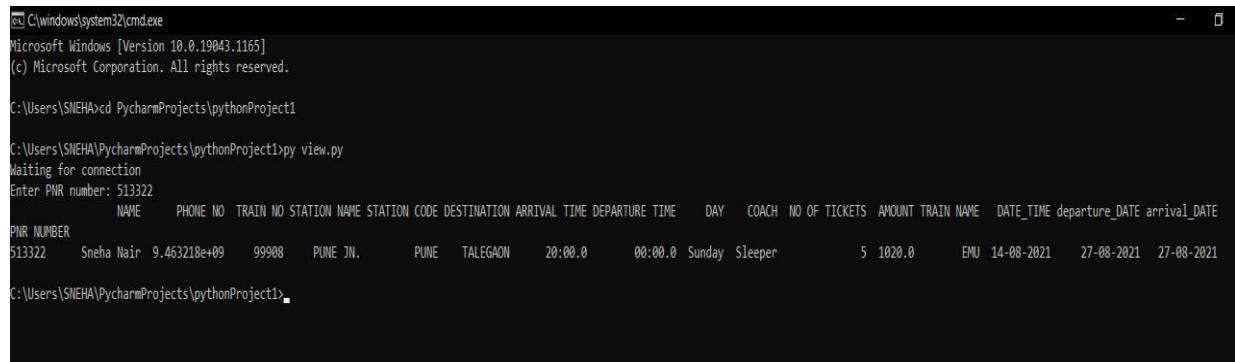
275	7	99905	PMP	195	176	157	138	PIMPRI	200	10:19:00	10:20:00	1200	2200	128	#####	#####	#####	#####
276	8	99905	KSWD	210	188	166	144	KASARWA	500	10:22:00	10:23:00	1400	2000	132	#####	#####	#####	#####
277	9	99905	DAPD	230	204	178	152	DAPODI	600	10:26:00	10:27:00	2000	1000	140	#####	#####	#####	#####
278	10	99905	KK	240	212	184	156	KHADKI	700	10:30:00	10:31:00	2020	1500	160	#####	#####	#####	#####
279	11	99905	SVJR	255	224	193	162	SHIVAJINA	350	10:40:00	10:40:00	1800	1100	168	#####	#####	#####	#####
280	1	99906	PUNE	100	100	100	100	PUNE JN.	450	15:40:00	15:40:00	2100	1200	176	#####	#####	#####	#####
281	2	99906	SVJR	110	108	106	104	SHIVAJINA	30	15:44:00	15:45:00	2200	1400	188	#####	#####	#####	#####
282	8	99906	AKRD	195	176	157	138	AKURDI	100	16:09:00	16:10:00	1250	2000	108	#####	#####	#####	#####
283	9	99906	DEHR	220	196	172	148	DEHU RO/	90	16:14:00	16:15:00	1900	2400	116	#####	#####	#####	#####
284	10	99906	BGWI	240	212	184	156	BEGDAEW	150	16:18:00	16:19:00	1500	2800	124	#####	#####	#####	#####
285	11	99906	GRWD	255	224	193	162	GHORAWI	200	16:21:00	16:22:00	1200	2100	140	#####	#####	#####	#####
286	12	99906	TGN	270	236	202	168	TALEGAON	120	16:30:00	16:30:00	1250	220	152	#####	#####	#####	#####
287	1	99907	TGN	104	100	100	100	TALEGAON	300	16:38:00	16:38:00	1350	2200	160	#####	#####	#####	#####
288	2	99907	GRWD	115	112	109	106	GHORAWI	250	16:40:00	16:41:00	1450	2500	168	#####	#####	#####	#####
289	3	99907	BGWI	130	124	118	112	BEGDAEW	108	16:43:00	16:44:00	1600	1200	188	#####	#####	#####	#####
290	9	99907	DAPD	230	204	178	152	DAPODI	168	17:07:00	17:08:00	1500	1400	220	#####	#####	#####	#####
291	10	99907	KK	240	212	184	156	KHADKI	188	17:12:00	17:13:00	1500	2000	224	#####	#####	#####	#####
292	11	99907	SVJR	255	224	193	162	SHIVAJINA	196	17:18:00	17:19:00	1500	2020	232	#####	#####	#####	#####
293	12	99907	PUNE	270	10	202	168	PUNE JN.	200	18:18:00	18:19:00	1800	1800	100	#####	#####	#####	#####

## CLIENT 3- Viewing ticket details

```
import socket
import pandas as pd
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
print('Waiting for connection')
try:
    s.connect((socket.gethostname(), 2000))
except socket.error as e:
    print(str(e))
s.send(bytes("VIEW", "utf-8"))
file = s.recv(1000).decode("utf-8")
pnr = int(input("Enter PNR number: "))
df = pd.read_csv(file)
x = df.loc[df['PNR NUMBER'] == pnr]
info = x[['PNR NUMBER', 'NAME', 'PHONE NO', 'TRAIN NO', 'STATION NAME',
          'STATION CODE', 'DESTINATION', 'ARRIVAL TIME', 'DEPARTURE TIME', 'DAY',
          'COACH', 'NO OF TICKETS', 'AMOUNT', 'TRAIN NAME', 'DATE_TIME',
          'departure_DATE', 'arrival_DATE']]
info.set_index('PNR NUMBER', inplace=True)
print(info.head())
s.close()
```

## Output

### client



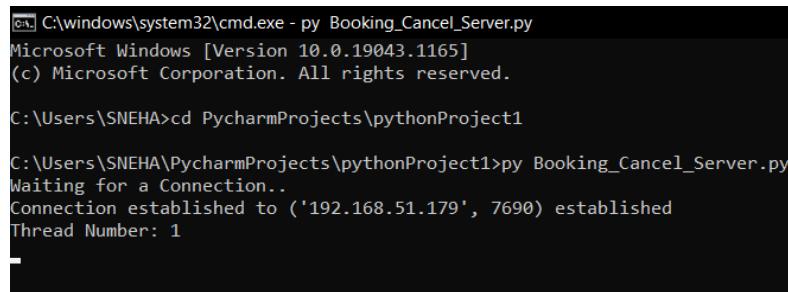
```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SNEHA>cd PycharmProjects\pythonProject1

C:\Users\SNEHA\PycharmProjects\pythonProject1>py view.py
Waiting for connection
Enter PNR number: 513322
      NAME   PHONE NO  TRAIN NO STATION NAME STATION CODE DESTINATION ARRIVAL TIME DEPARTURE TIME   DAY COACH NO OF TICKETS AMOUNT TRAIN NAME DATE_TIME departure_DATE arrival_DATE
PNR NUMBER
513322  Sneha Nair  9.463218e+09  99908   PUNE JN.        PUNE  TALEGAON    20:00:0    00:00:0 Sunday  Sleeper      5 1020.0   EMU 14-08-2021  27-08-2021  27-08-2021

C:\Users\SNEHA\PycharmProjects\pythonProject1>
```

### server



```
C:\Windows\system32\cmd.exe - py Booking_Cancel_Server.py
Microsoft Windows [Version 10.0.19043.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SNEHA>cd PycharmProjects\pythonProject1

C:\Users\SNEHA\PycharmProjects\pythonProject1>py Booking_Cancel_Server.py
Waiting for a Connection..
Connection established to ('192.168.51.179', 7690) established
Thread Number: 1
```

## GoBackN:

Server code:

```
package gobackn;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.net.ServerSocket;
import java.net.Socket;
import java.net.SocketException;

public class Server {
    static ServerSocket Serversocket;
    static DataInputStream dis;
    static DataOutputStream dos;

    public static void main(String[] args) throws SocketException {

        try {
            int a[] = { 30, 40, 50, 60, 70, 80, 90, 100, 110};
            Serversocket = new ServerSocket(8011);
            System.out.println("waiting for connection");
            Socket client = Serversocket.accept();
            dis = new DataInputStream(client.getInputStream());
            dos = new DataOutputStream(client.getOutputStream());
            System.out.println("The number of packets sent is:" + a.length);
            int y = a.length;
            dos.write(y);
            dos.flush();
            for (int i = 0; i < a.length; i++) {
                dos.write(a[i]);
                dos.flush();
            }

            int k = dis.read();
            dos.write(a[k]);
            dos.flush();

        } catch (IOException e) {
            System.out.println(e);
        } finally {
            try {
                dis.close();
                dos.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```
}
```

Client code:

```
package gobackn;
import java.lang.System;
import java.net.*;
import java.io.*;

public class Client {
    static Socket connection;

    public static void main(String a[]) throws SocketException {
        try {
            int v[] = new int[9];
            int n = 0;
            InetAddress addr = InetAddress.getByName("localhost");
            System.out.println(addr);
            connection = new Socket(addr, 8011);
            DataOutputStream out = new DataOutputStream(
                connection.getOutputStream());
            DataInputStream in = new DataInputStream(
                connection.getInputStream());
            int p = in.read();
            System.out.println("No of frame is:" + p);

            for (int i = 0; i < p; i++) {
                v[i] = in.read();
                System.out.println(v[i]);
            }
            v[5] = -1;
            for (int i = 0; i < p; i++)
            {
                System.out.println("Received frame is: " + v[i]);
            }
            for (int i = 0; i < p; i++)
                if (v[i] == -1) {
                    System.out.println("Request to retransmit packet no "
                        + (i+1) + " again!!");
                    n = i;
                    out.write(n);
                    out.flush();
                }
            System.out.println();

            v[n] = in.read();
            System.out.println("Received frame is: " + v[n]);
            System.out.println("quiting");
        } catch (Exception e) {
```

```
        System.out.println(e);
    }

}
}
```

### Server Output:

```
C:\windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SNEHA>cd desktop

C:\Users\SNEHA\Desktop>javac Server.java

C:\Users\SNEHA\Desktop>java Server.java
waiting for connection
The number of packets sent is:9

C:\Users\SNEHA\Desktop>
```

### Client Output:

```
C:\Users\SNEHA\Desktop>javac Client.java

C:\Users\SNEHA\Desktop>java Client.java
Localhost/127.0.0.1
No of frame is:9
20
30
40
50
60
70
80
90
100
Received frame is: 20
Received frame is: 30
Received frame is: 40
Received frame is: 50
Received frame is: 60
Received frame is: -1
Received frame is: 80
Received frame is: 90
Received frame is: 100
Request to retransmit packet no 6 again!

Received frame is: 70
quiting

C:\Users\SNEHA\Desktop>
```

# Implementation With Packet Tracer

Address Block: 121.37.12.64 /26

## Computer Networks Lab Evaluation - 3

Given Address Block is - 121.37.12.64/26

Let the number of required Subnets be 4

Since the number of required subnets are 4 we need 2 bits to identify each subnet.

So the subnet mask becomes

$$\begin{aligned} &= 2^6 + 2 \\ &= 2^8 \\ &= 121.37.12.64/28 \end{aligned}$$

As 28 bits are reserved for the network part of the IP Address only 4 bits will be reserved for the host.

$\therefore 2^4 = 16$  IP address can be used in each subnet

Subnet 1 - 00

Subnet 2 - 01

Subnet 3 - 10

Subnet 4 - 11

### Subnet 1

1st 26 bits are fixed

next 2 bits are (00) → which is used to identify subnet 1

Since there are 16 Addresses. To point to the 1st address all the 4 non fixed bits should be 0

→ So the 1<sup>st</sup> address of subnet 1 is

121.37.12.01 00 0000/28  
fixed To identify Subnet 1

$$\therefore = 121.37.12.64/28$$

→ Since 16 address are possible adding (16-1) gives the last address

$$\therefore 127.37.12.(64+15) = 127.37.12.79$$

$$\text{last address} = 127.37.12.79/28$$

### Subnet 2

1st 26 bits are fixed

next 2 bits are (01) → which is used to identify subnet 2

→ The 1<sup>st</sup> address of subnet 2 is

121.37.12.0101 0000/28  
fixed used to identify subnet 2

$$\text{First address} = 121.37.12.80/28$$

→ last address can be found by adding 15

$$= 121.37.12.(80+15)$$

$$= 121.37.12.95$$

$$\text{last address} = 121.37.12.95/28$$

Subnet 3:-

1st 26 bits are fixed

Next 2 bits are (10) → used to identify subnet 3

→ 1<sup>st</sup> address of subnet 3 is

121.37.12.0110 0000/28

fixed      used to identify subnet

→ first address = 121.37.12.96/28.

→ last address can be found by adding 15

$$= 121.37.12(96+15)$$

$$= 126.37.12.111$$

last address = 126.37.12.111/28

Subnet 4:-

first 26 bits are fixed

next 2 bits are (11) → used to identify subnet 4

→ 1<sup>st</sup> address of subnet 4 is

121.37.12.0111 0000/28

fixed      used to identify subnet

First address = 121.37.12.111/28



→ last address can be obtained by adding 15

$$= 121 \cdot 37 \cdot 12 \cdot (112 + 5)$$

$$= 121 \cdot 37 \cdot 12 \cdot 127$$

$$\text{last address} = 121 \cdot 37 \cdot 12 \cdot 127 / 28$$

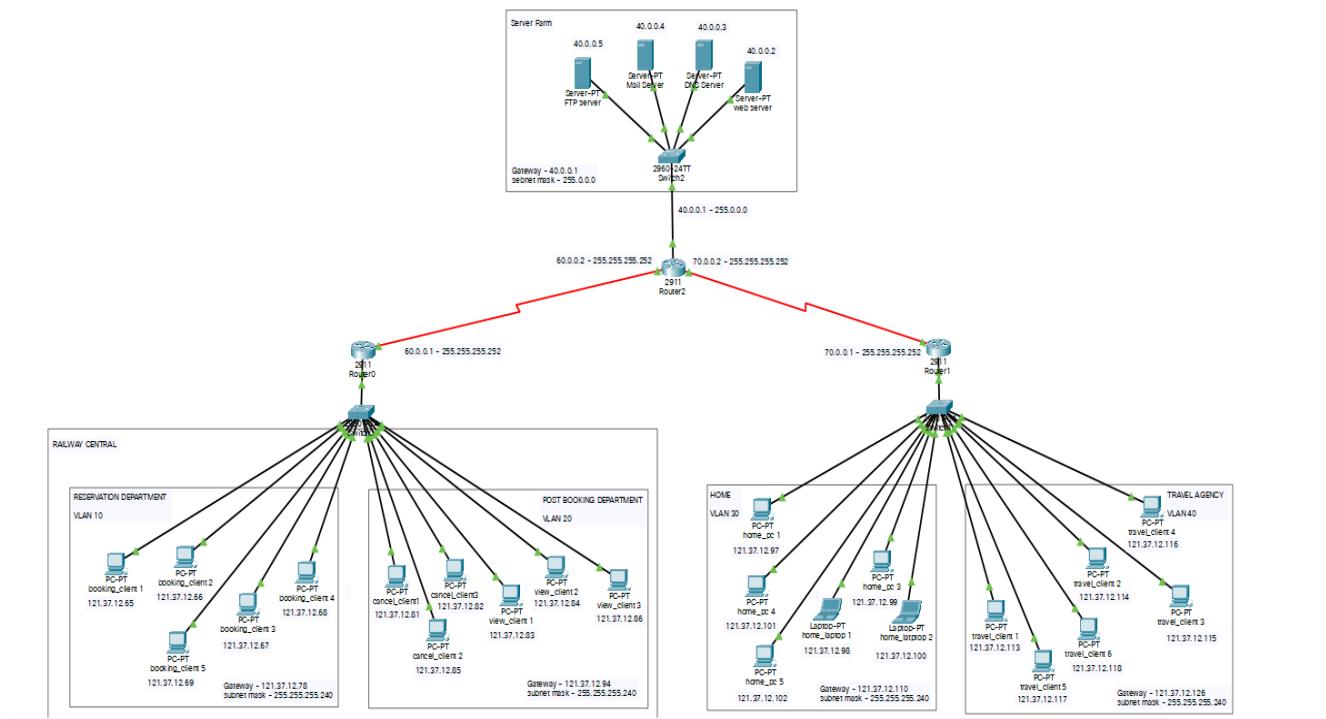
<u>Subnet</u>	<u>Start Address</u>	<u>End Address</u>
1	127.37.12.64	127.37.12.79
2	127.37.12.80	127.37.12.95
3	127.37.12.96	127.37.12.111
4	127.37.12.112	127.37.12.127

Department Name	Subnet Mask	Wild Card Mask	Network Address	Broadcast Address	Usable addresses	Gateway	Host IP Address
Reservation	255.255.255.240	0.0.0.15	121.37.12.64	121.37.12.79	14	121.37.12.78	121.37.12.65-booking_clinet 1 121.37.12.66-booking_clinet 2 121.37.12.67-booking_clinet 3 121.37.12.68-booking_clinet 4 121.37.12.69-booking_clinet 5
Post booking	255.255.255.240	0.0.0.15	121.37.12.80	121.37.12.96	14	121.37.12.95	121.37.12.81-cancel_clinet 1 121.37.12.85-cancel_clinet 2 121.37.12.82-cancel_clinet 3 121.37.12.83-view_clinet 1 121.37.12.84-view_clinet 2 121.37.12.86-view_clinet 3
Home	255.255.255.240	0.0.0.15	121.37.12.97	121.37.12.111	14	121.37.12.110	121.37.12.97-home_pc1 121.37.12.98-home_laptop 1 121.37.12.99-home_pc 2 121.37.12.100-home_laptop 2 121.37.12.101-home_pc 3 121.37.12.102-home_pc 4
Travel agency	255.255.255.240	0.0.0.15	121.37.12.112	121.37.12.127	14	121.37.12.126	121.37.12.113-travel_clinet 1 121.37.12.114-travel_clinet 2 121.37.12.115-travel_clinet 3 121.37.12.116-travel_clinet 4 121.37.12.117-travel_client 5 121.37.12.118-travel_clinet 6

## Servers used:

web server, DNS server, Mail server and FTP server

Packet Tracer model using VLAN and OSPF protocol:



## Ip protocol used:

Router2

Physical Config **CLI** Attributes

IOS Command Line Interface

```

00:00:10: %OSPF-5-ADJCHG: Process 10, Nbr 121.37.12.126 on Serial0/0/1 from
LOADING to FULL, Loading Done

00:00:10: %OSPF-5-ADJCHG: Process 10, Nbr 121.37.12.94 on Serial0/0/0 from
LOADING to FULL, Loading Done

Router>
Router>
Router>
Router>show ip protocols

Routing Protocol is "ospf 10"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 70.0.0.2
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    60.0.0.0 0.0.0.3 area 0
    70.0.0.0 0.0.0.3 area 0
    40.0.0.0 0.255.255.255 area 0
  Routing Information Sources:
    Gateway        Distance      Last Update
    70.0.0.2          110      00:00:30
    121.37.12.94      110      00:00:30
    121.37.12.126     110      00:00:30
  Distance: (default is 110)

Router>

```

Ctrl+F6 to exit CLI focus      **Copy**      **Paste**

Top

## VLAN configuration

Switch0

Physical Config **CLI** Attributes

IOS Command Line Interface

```
%LINK-5-CHANGED: Interface FastEthernet0/5, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/5, changed
state to up

Switch>
Switch>
Switch>show vlan brief

VLAN Name          Status    Ports
-----            -----
1    default        active    Fa0/21, Fa0/22, Fa0/23,
Fa0/24
                                         Gig0/2
10   ***DEPT.01***  active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                         Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                         Fa0/9, Fa0/10
20   ***DEPT.20***  active    Fa0/11, Fa0/12, Fa0/13,
Fa0/14
                                         Fa0/15, Fa0/16, Fa0/17,
Fa0/18
                                         Fa0/19, Fa0/20
1002 fddi-default  active
1003 token-ring-default  active
1004 fddinet-default  active
1005 trnet-default    active
Switch>
```

Ctrl+F6 to exit CLI focus      **Copy**      **Paste**

Switch1

Physical Config **CLI** Attributes

IOS Command Line Interface

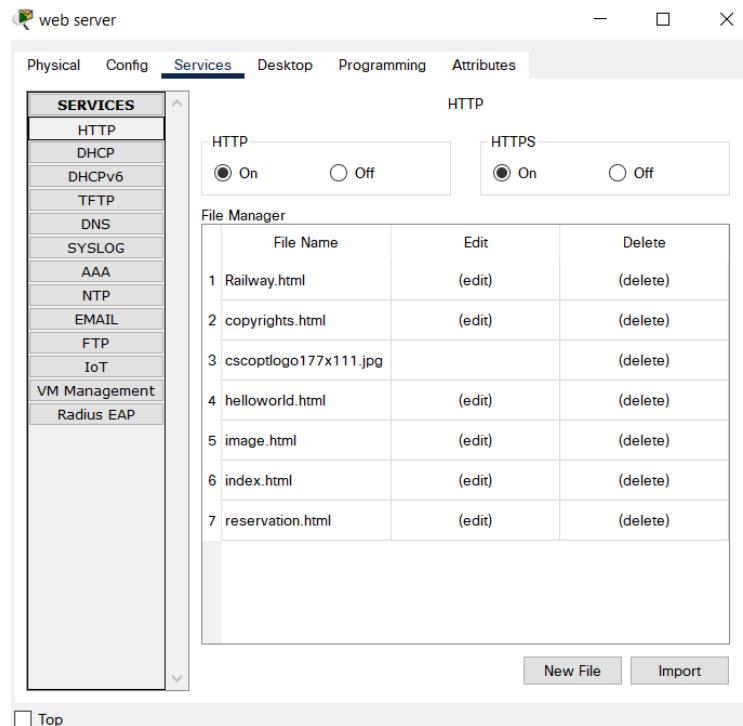
```
%LINK-5-CHANGED: Interface FastEthernet0/16, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/16, changed
state to up

Switch>
Switch>
Switch>show vlan brief

VLAN Name          Status    Ports
-----            -----
1    default        active    Fa0/21, Fa0/22, Fa0/23,
Fa0/24
                                         Gig0/2
30   ***DEPT.03***  active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                         Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                         Fa0/9, Fa0/10
40   ***DEPT.04***  active    Fa0/11, Fa0/12, Fa0/13,
Fa0/14
                                         Fa0/15, Fa0/16, Fa0/17,
Fa0/18
                                         Fa0/19, Fa0/20
1002 fddi-default  active
1003 token-ring-default  active
1004 fddinet-default  active
1005 trnet-default    active
Switch>
```

Ctrl+F6 to exit CLI focus      **Copy**      **Paste**

## Web server configuration

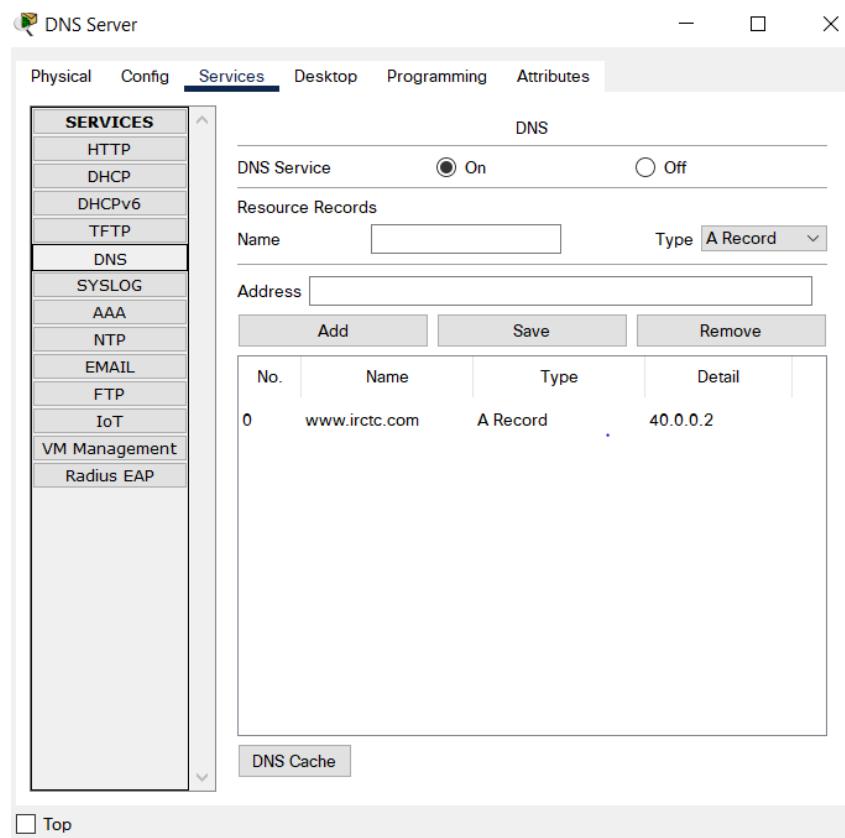


Top

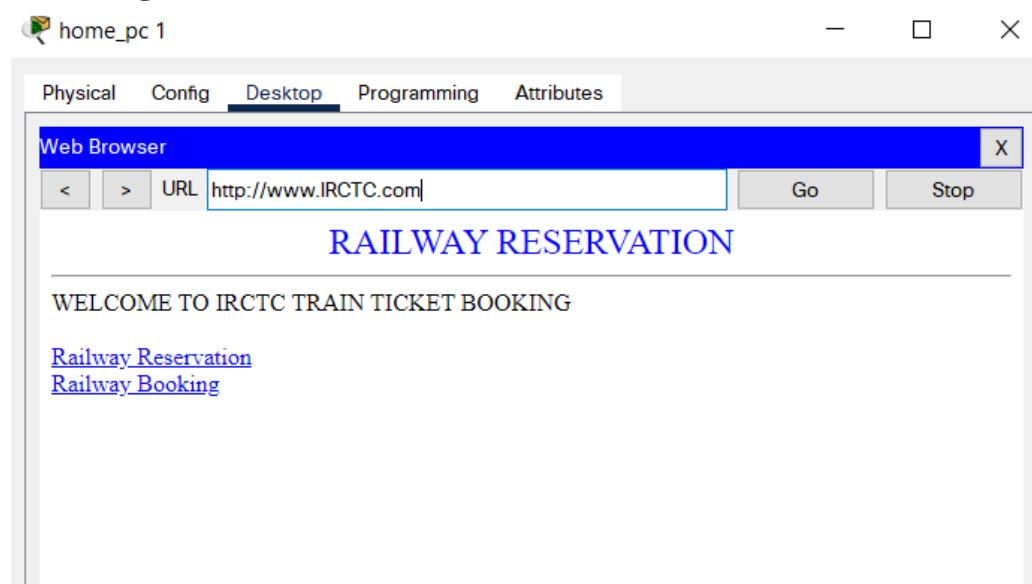
## Working of Web server:

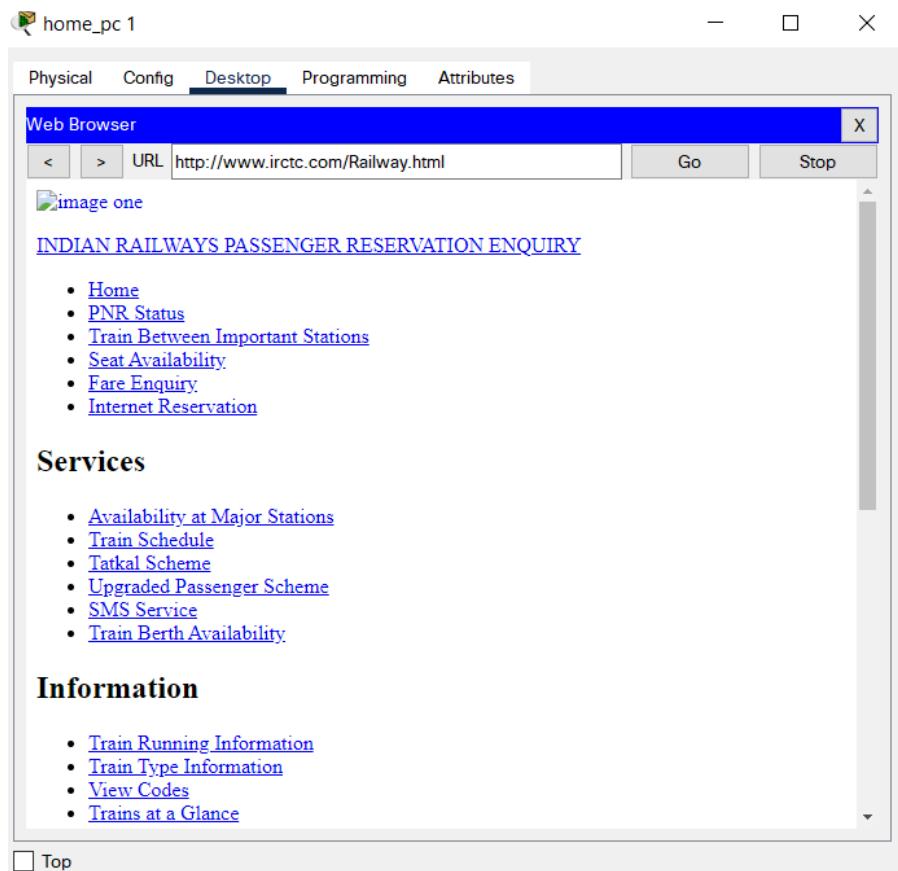
The screenshot shows a travel client application window. At the top, there are tabs for Physical, Config, Desktop, Programming, and Attributes. The Desktop tab is selected, showing a 'Web Browser' window. The URL bar in the browser shows 'http://40.0.0.2'. The page content displays the 'RAILWAY RESERVATION' logo and a welcome message: 'WELCOME TO IRCTC TRAIN TICKET BOOKING'. It also contains links for 'Railway Reservation' and 'Railway Booking'. Below this, there is a search form with fields for 'From' (NCL - NEW DELHI), 'To' (MMCT - Mumbai Central), and 'Departure Date' (Thu, 07 Oct). A large green 'SEARCH' button is prominently displayed. The bottom of the screen features a green banner with the text 'Highest rated train booking app'.

## Configuration of DNS server:

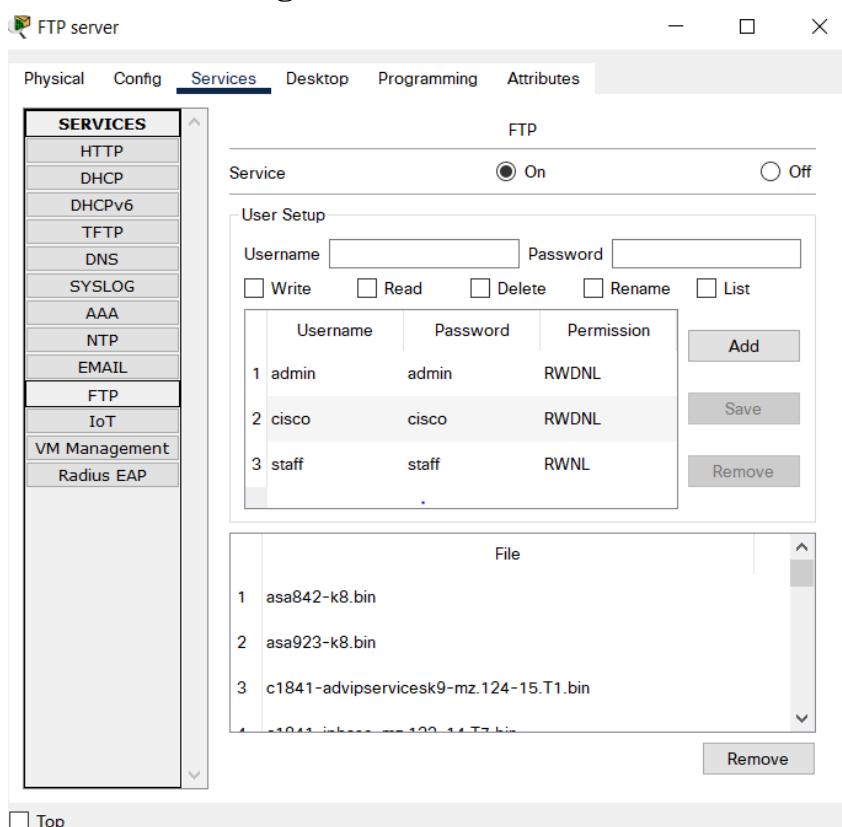


## Working of DNS server:

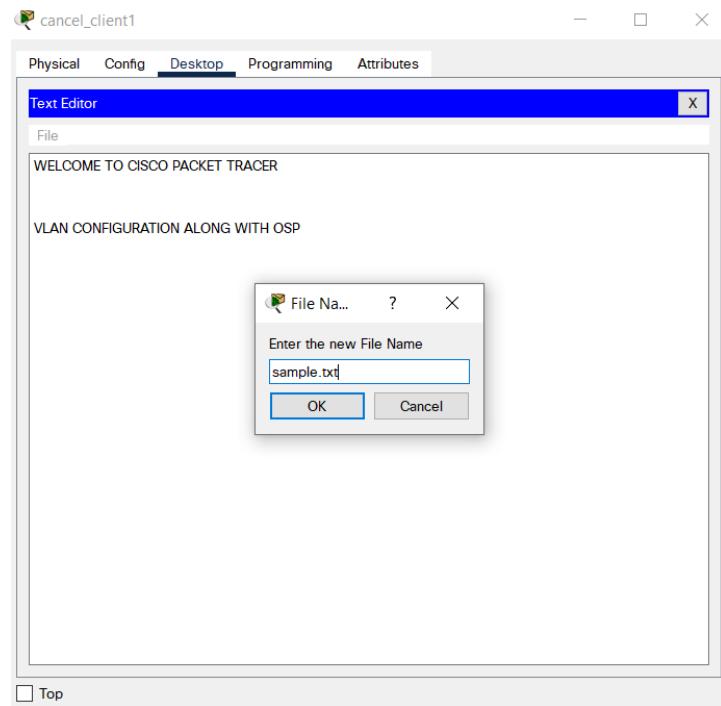




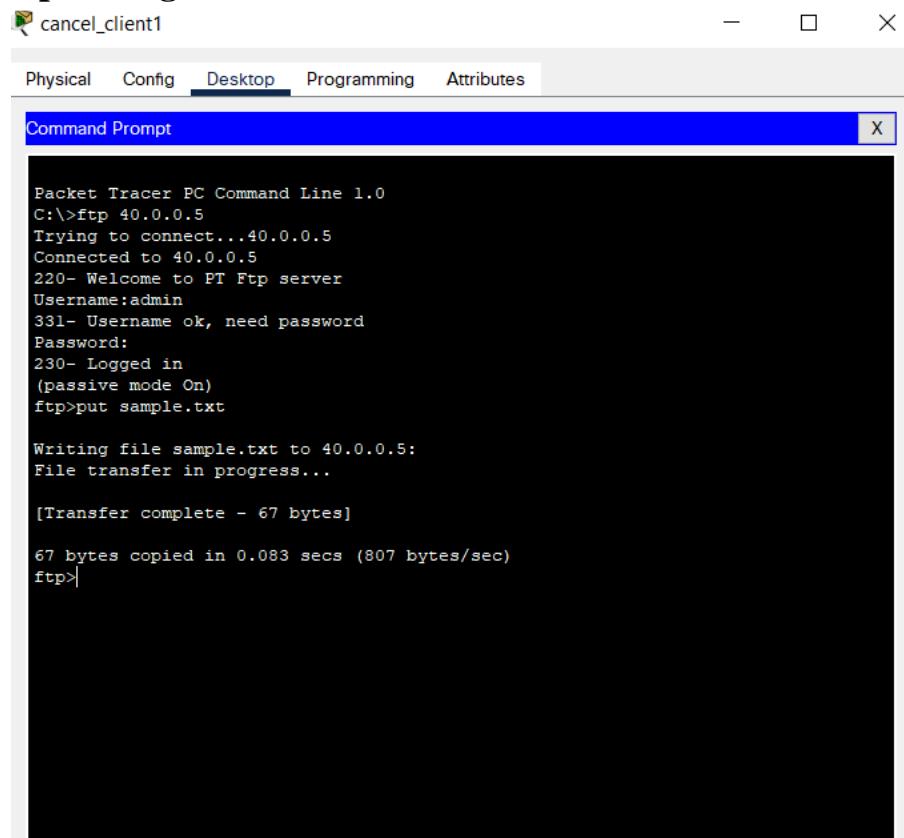
## FTP server configuration:



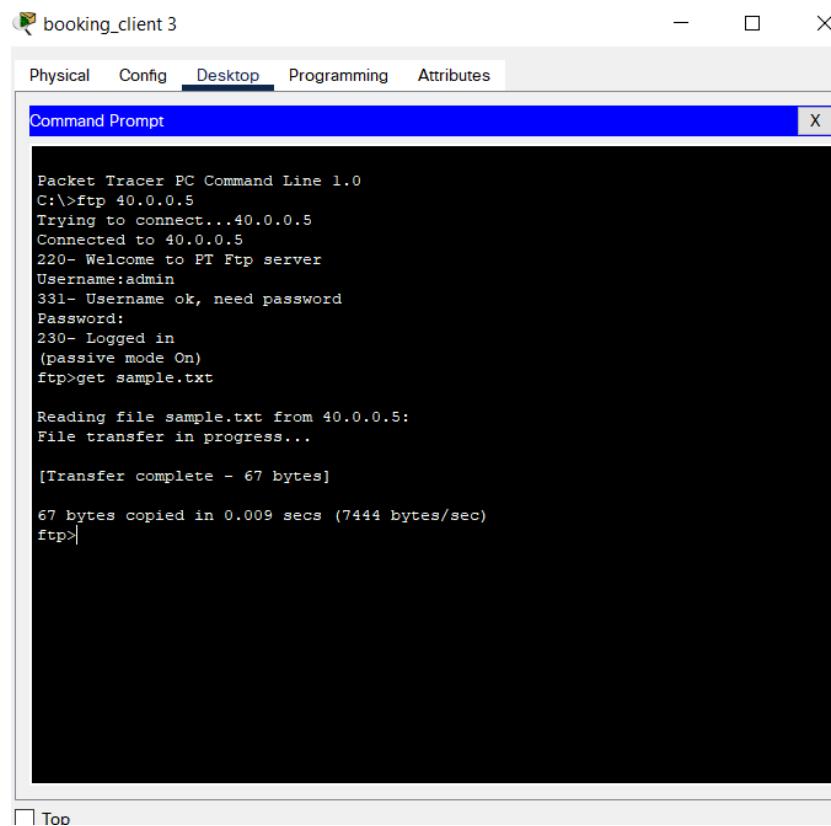
## Working of FTP server:



## Uploading the file



## Getting the file



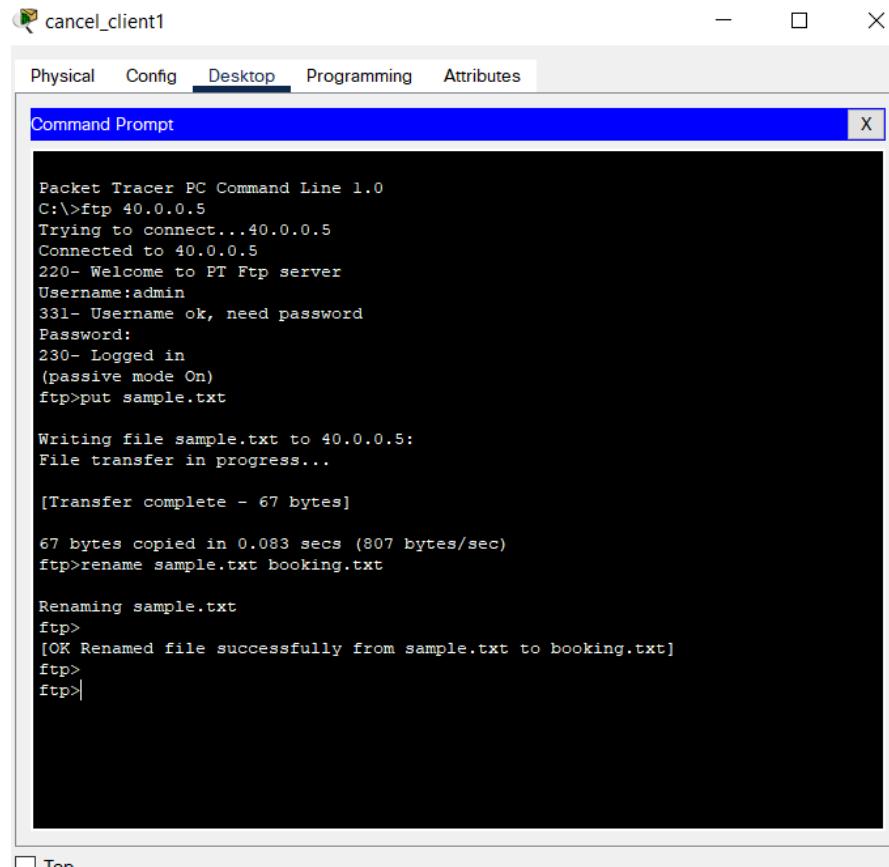
```
Packet Tracer PC Command Line 1.0
C:\>ftp 40.0.0.5
Trying to connect...40.0.0.5
Connected to 40.0.0.5
220- Welcome to PT Ftp server
Username:admin
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>get sample.txt

Reading file sample.txt from 40.0.0.5:
File transfer in progress...

[Transfer complete - 67 bytes]

67 bytes copied in 0.009 secs (7444 bytes/sec)
ftp>
```

## Renaming the file



```
Packet Tracer PC Command Line 1.0
C:\>ftp 40.0.0.5
Trying to connect...40.0.0.5
Connected to 40.0.0.5
220- Welcome to PT Ftp server
Username:admin
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>put sample.txt

Writing file sample.txt to 40.0.0.5:
File transfer in progress...

[Transfer complete - 67 bytes]

67 bytes copied in 0.083 secs (807 bytes/sec)
ftp>rename sample.txt booking.txt

Renaming sample.txt
ftp>
[OK Renamed file successfully from sample.txt to booking.txt]
ftp>
ftp>
```

## Deleting the file

The screenshot shows a terminal window titled "cancel\_client1" with the "Desktop" tab selected. The window contains a "Command Prompt" section with the following text:

```
Packet Tracer PC Command Line 1.0
C:\>ftp 40.0.0.5
Trying to connect...40.0.0.5
Connected to 40.0.0.5
220- Welcome to PT Ftp server
Username:admin
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>put sample.txt

Writing file sample.txt to 40.0.0.5:
File transfer in progress...

[Transfer complete - 67 bytes]

67 bytes copied in 0.083 secs (807 bytes/sec)
ftp>rename sample.txt booking.txt

Renaming sample.txt
ftp>
[OK Renamed file successfully from sample.txt to booking.txt]
ftp>
ftp>delete booking.txt

Deleting file booking.txt from 40.0.0.5: ftp>
[Deleted file booking.txt successfully ]
ftp>
```

Top

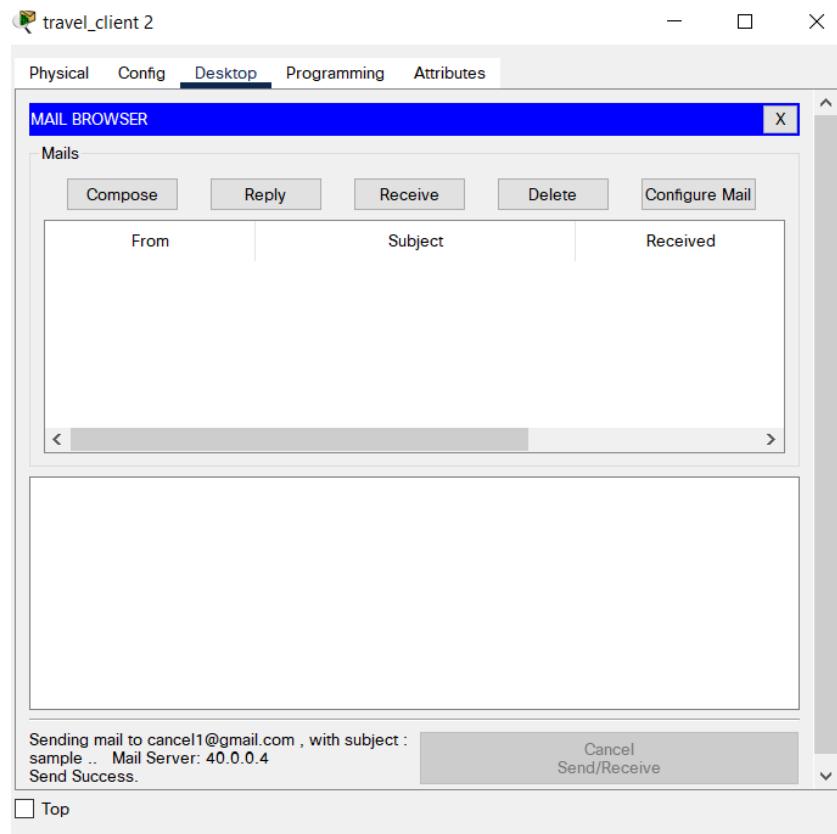
## Mail server Configuration:

The screenshot shows the "Mail Server" configuration interface with the "Services" tab selected. The left sidebar lists various services: HTTP, DHCP, DHCPv6, TFTP, DNS, SYSLOG, AAA, NTP, EMAIL, FTP, IoT, VM Management, and Radius EAP. The EMAIL section is expanded, showing the "SMTP Service" and "POP3 Service" status as "ON". The "Domain Name" is set to "gmail.com". The "User Setup" section lists users: booking1, booking2, booking3, booking4, cancel1, cancel2, cancel3, view1, view2, view3, home1, home2, home3, home4, travel1, travel2, travel3, travel4, and railway. Buttons for adding (+), removing (-), changing, and password management are available.

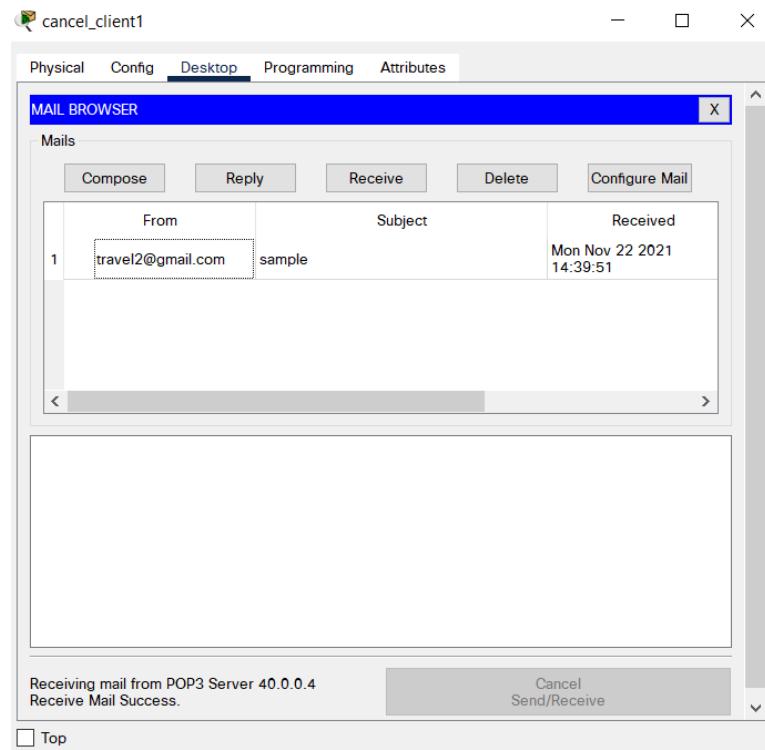
Top

## Working of Mail server:

### Sending mail

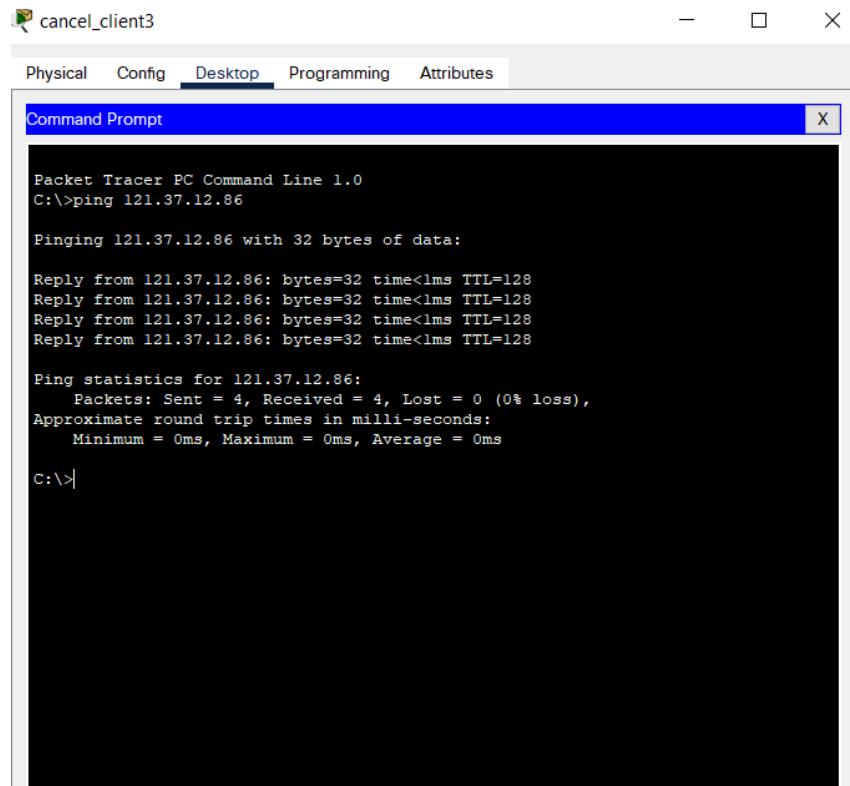


### Receiving mail



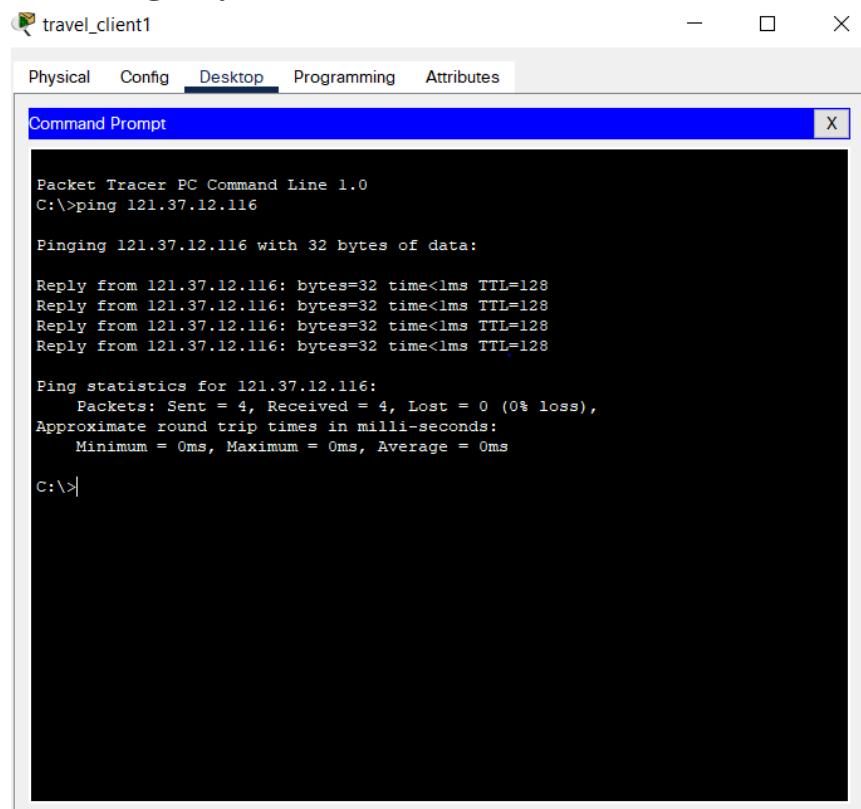
## Connectivity between clients of the same department

### Post booking



Packet Tracer PC Command Line 1.0  
C:\>ping 121.37.12.86  
  
Pinging 121.37.12.86 with 32 bytes of data:  
  
Reply from 121.37.12.86: bytes=32 time<1ms TTL=128  
  
Ping statistics for 121.37.12.86:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 0ms, Average = 0ms  
  
C:\>

### Travel agency

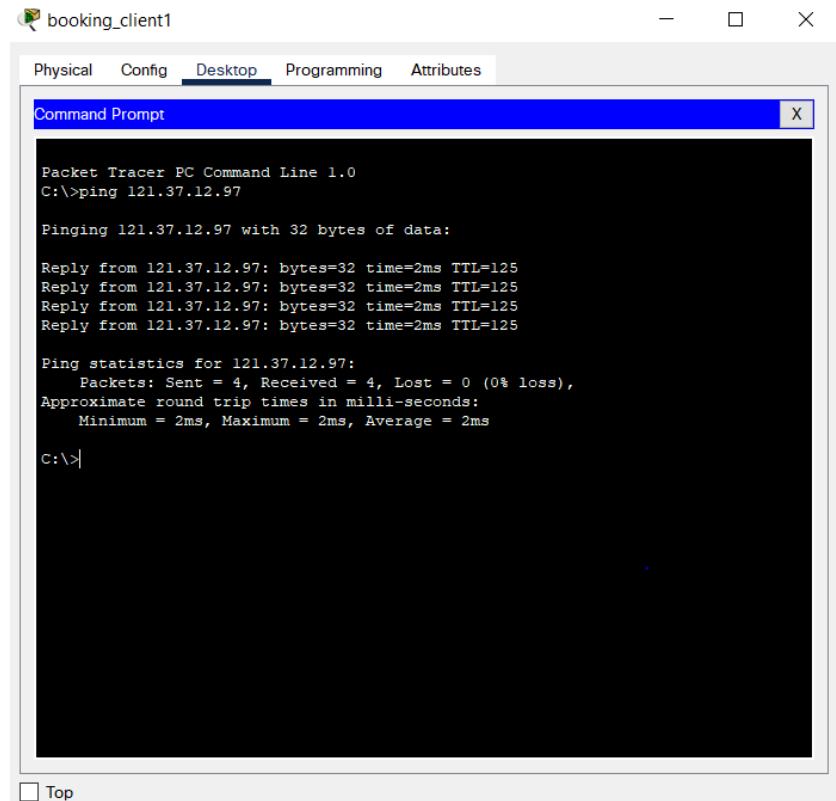


Packet Tracer PC Command Line 1.0  
C:\>ping 121.37.12.116  
  
Pinging 121.37.12.116 with 32 bytes of data:  
  
Reply from 121.37.12.116: bytes=32 time<1ms TTL=128  
  
Ping statistics for 121.37.12.116:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 0ms, Average = 0ms  
  
C:\>

Top

## Connectivity between clients of different department

### Reservation and Travel agency



```
Packet Tracer PC Command Line 1.0
C:\>ping 121.37.12.97

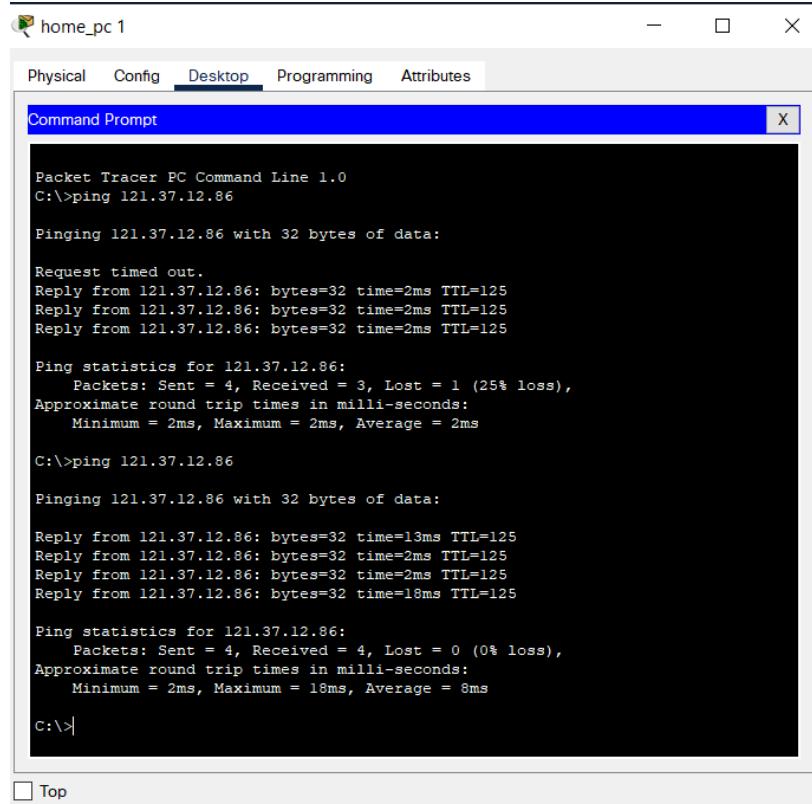
Pinging 121.37.12.97 with 32 bytes of data:

Reply from 121.37.12.97: bytes=32 time=2ms TTL=125

Ping statistics for 121.37.12.97:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 2ms, Average = 2ms

C:\>|
```

### Home and Post booking



```
Packet Tracer PC Command Line 1.0
C:\>ping 121.37.12.86

Pinging 121.37.12.86 with 32 bytes of data:

Request timed out.
Reply from 121.37.12.86: bytes=32 time=2ms TTL=125
Reply from 121.37.12.86: bytes=32 time=2ms TTL=125
Reply from 121.37.12.86: bytes=32 time=2ms TTL=125

Ping statistics for 121.37.12.86:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 2ms, Average = 2ms

C:\>ping 121.37.12.86

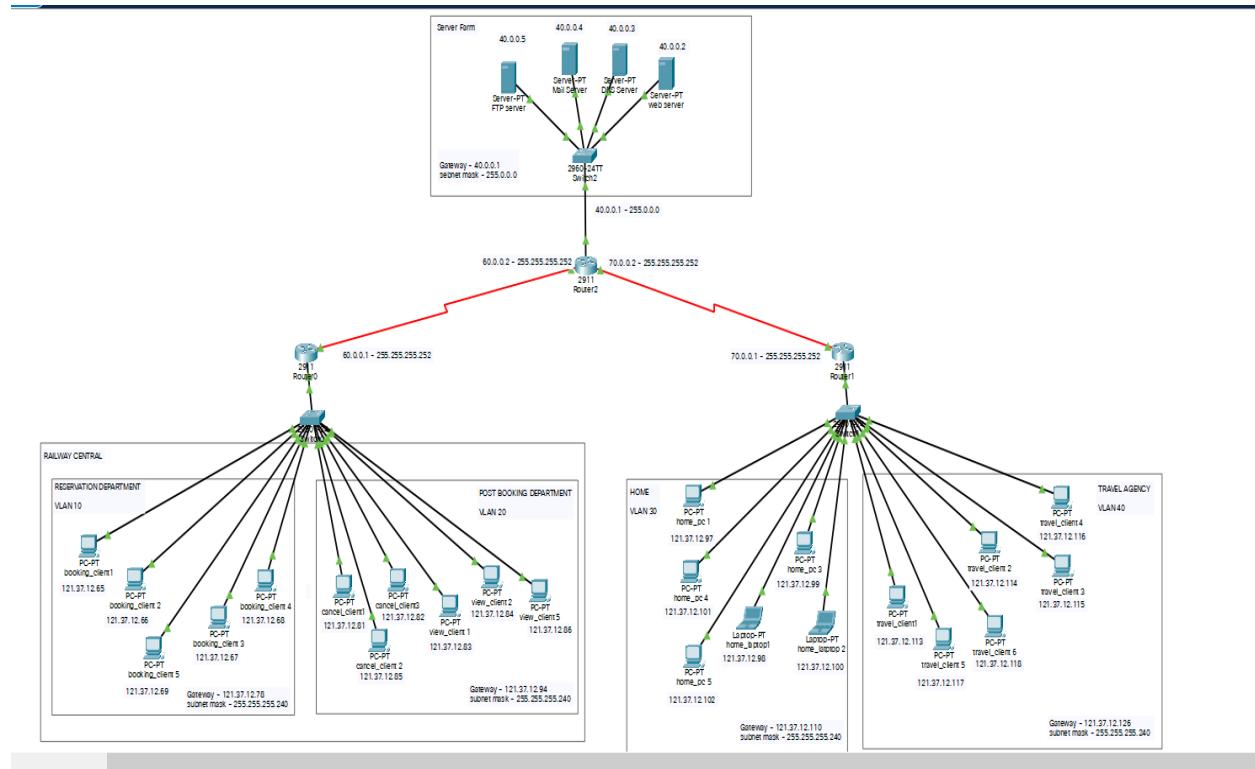
Pinging 121.37.12.86 with 32 bytes of data:

Reply from 121.37.12.86: bytes=32 time=13ms TTL=125
Reply from 121.37.12.86: bytes=32 time=2ms TTL=125
Reply from 121.37.12.86: bytes=32 time=2ms TTL=125
Reply from 121.37.12.86: bytes=32 time=18ms TTL=125

Ping statistics for 121.37.12.86:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 2ms, Maximum = 18ms, Average = 8ms

C:\>|
```

## Packet Tracer model using VLAN and RIP protocol



### Ip protocol used:

Router2

Physical Config **CLI** Attributes

IOS Command Line Interface

```
up

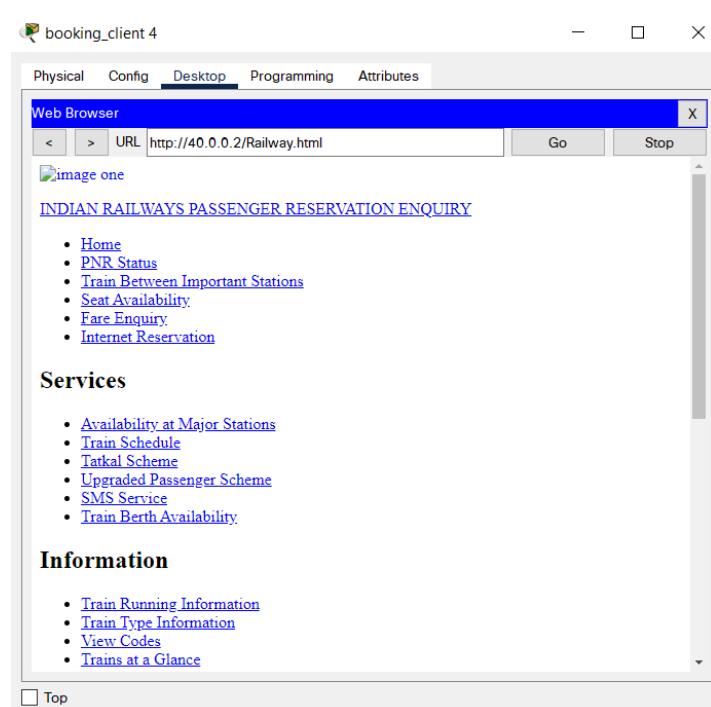
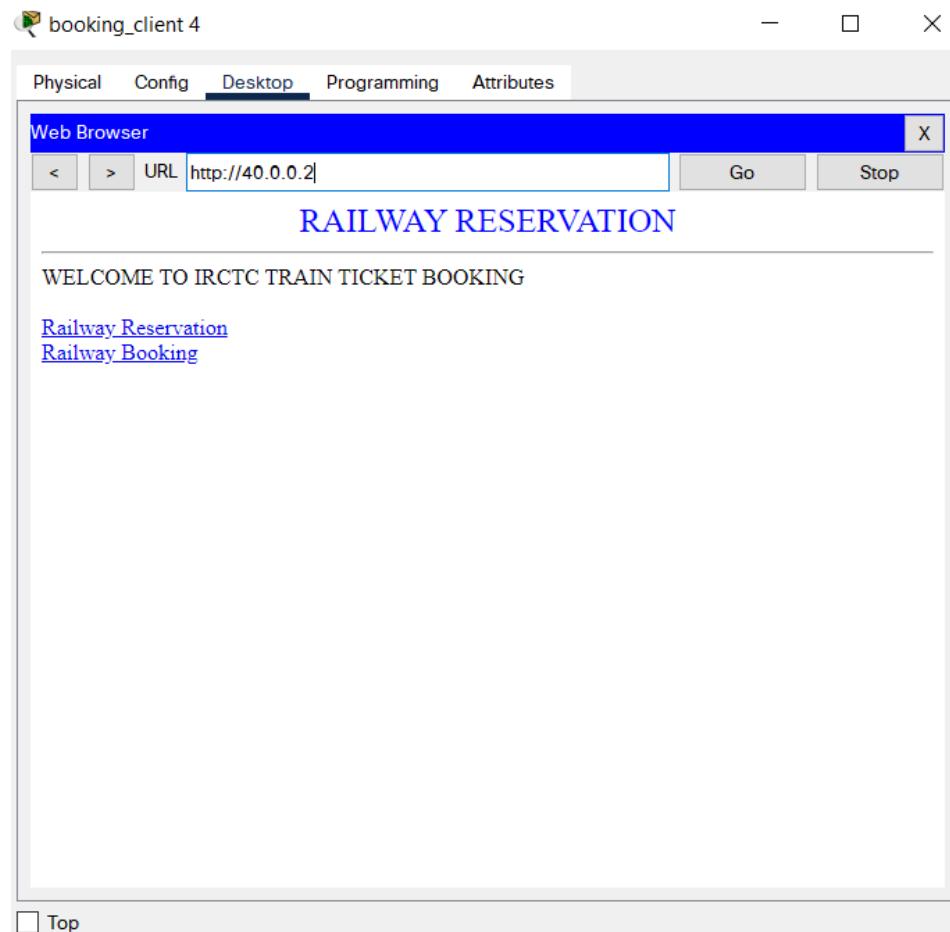
Router>
Router>show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 13 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: rip
  Default version control: send version 2, receive 2
  Interface          Send   Rcv   Triggered RIP  Key-chain
  GigabitEthernet0/0    2      2
  Serial0/0/0           2      2
  Serial0/0/1           2      2
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    40.0.0.0
    60.0.0.0
    70.0.0.0
  Passive Interface(s):
  Routing Information Sources:
    Gateway          Distance      Last Update
    60.0.0.1           120          00:00:14
    70.0.0.1           120          00:00:14
  Distance: (default is 120)
  Router>
```

Ctrl+F6 to exit CLI focus     

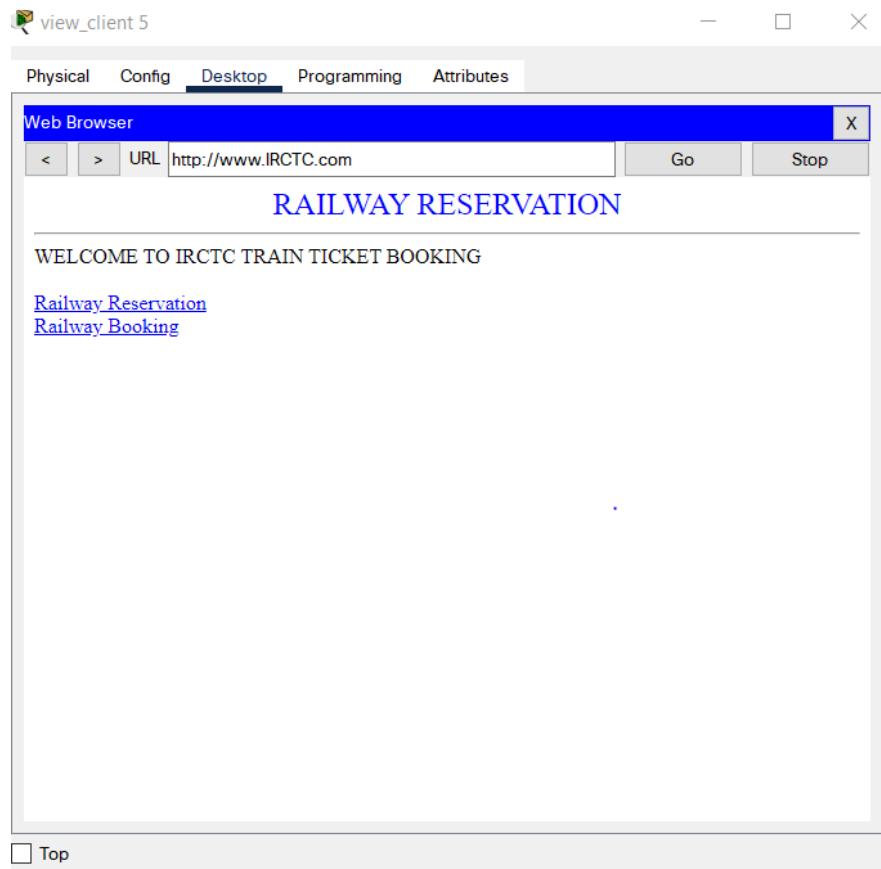
Top

Configurations of Mail server, FTP server, DNS server and Web server are same as the above model

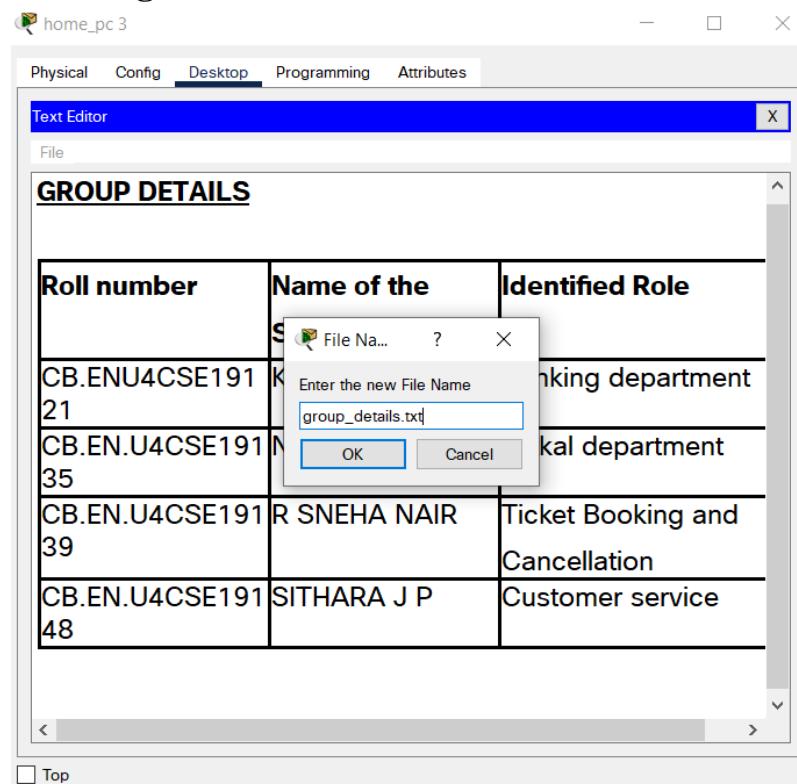
## Working of Web server:



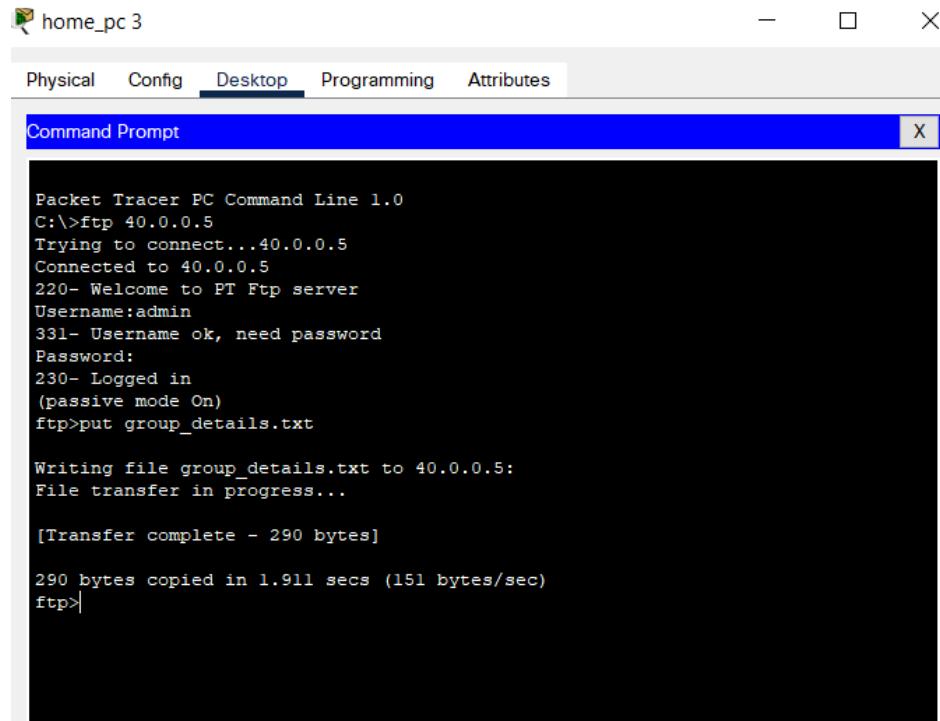
## Working of DNS server:



## Working of FTP server:



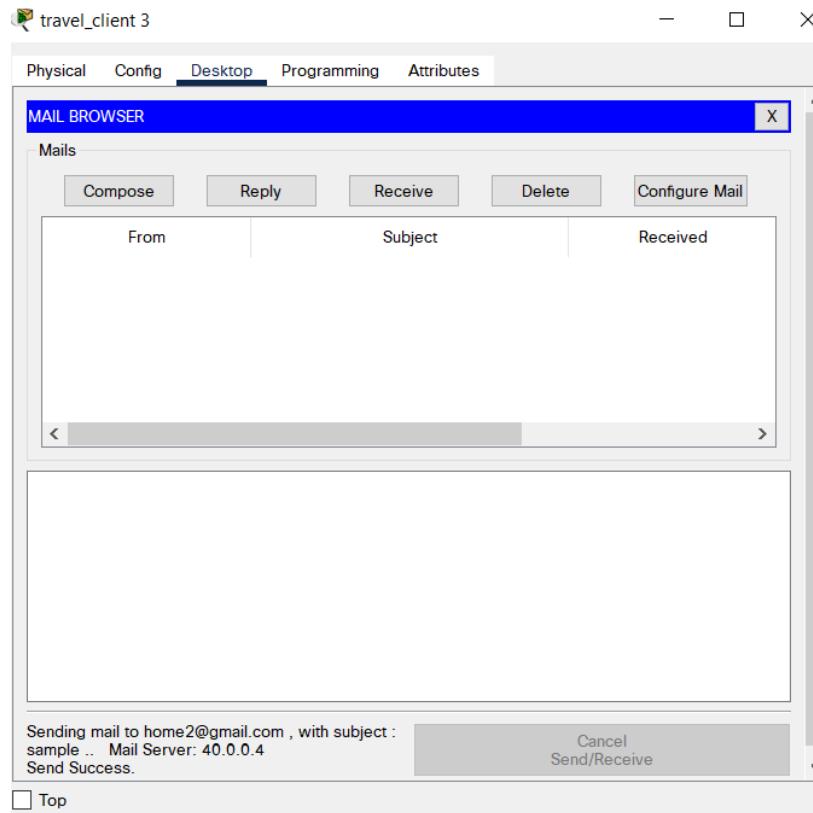
## Upload file



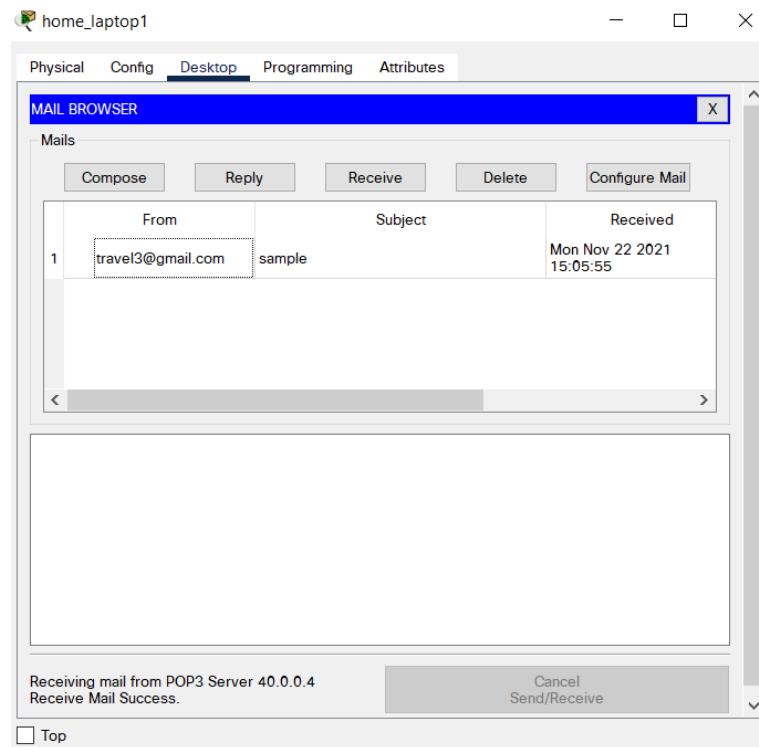
Packet Tracer PC Command Line 1.0  
C:\>ftp 40.0.0.5  
Trying to connect...40.0.0.5  
Connected to 40.0.0.5  
220- Welcome to PT Ftp server  
Username:admin  
331- Username ok, need password  
Password:  
230- Logged in  
(passive mode On)  
ftp>put group\_details.txt  
  
Writing file group\_details.txt to 40.0.0.5:  
File transfer in progress...  
  
[Transfer complete - 290 bytes]  
  
290 bytes copied in 1.911 secs (151 bytes/sec)  
ftp>|

## Working of Mail server:

### Sending mail

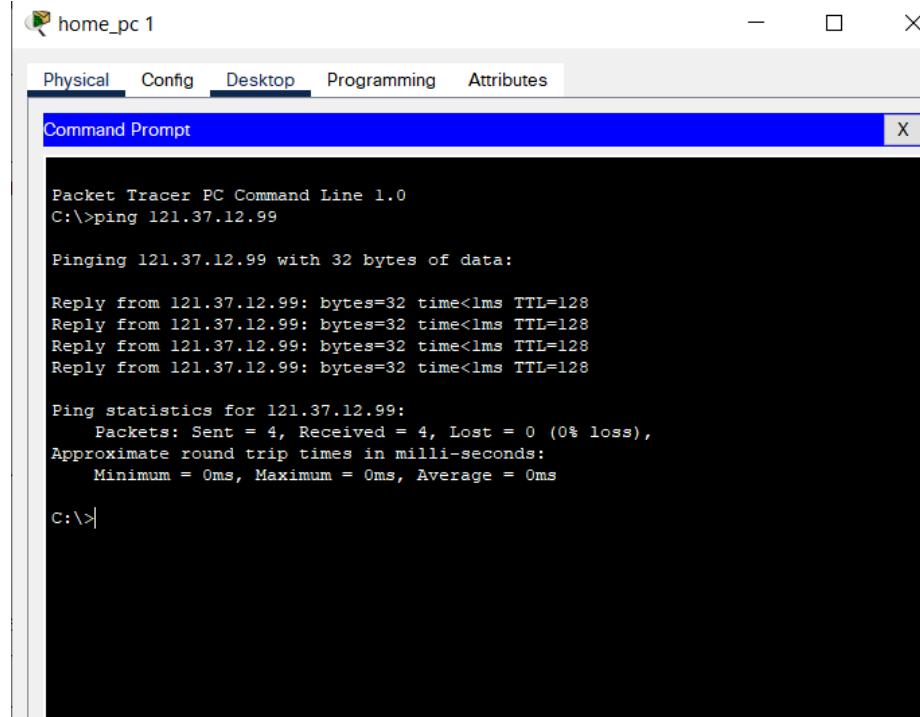


## Receiving mail

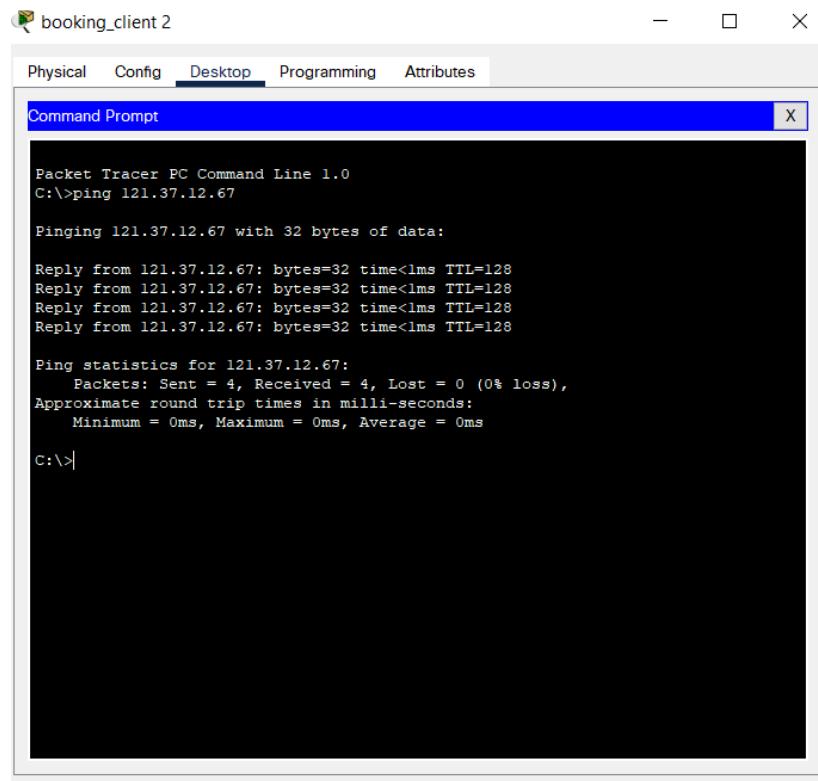


## Connectivity between clients of the same department:

### home



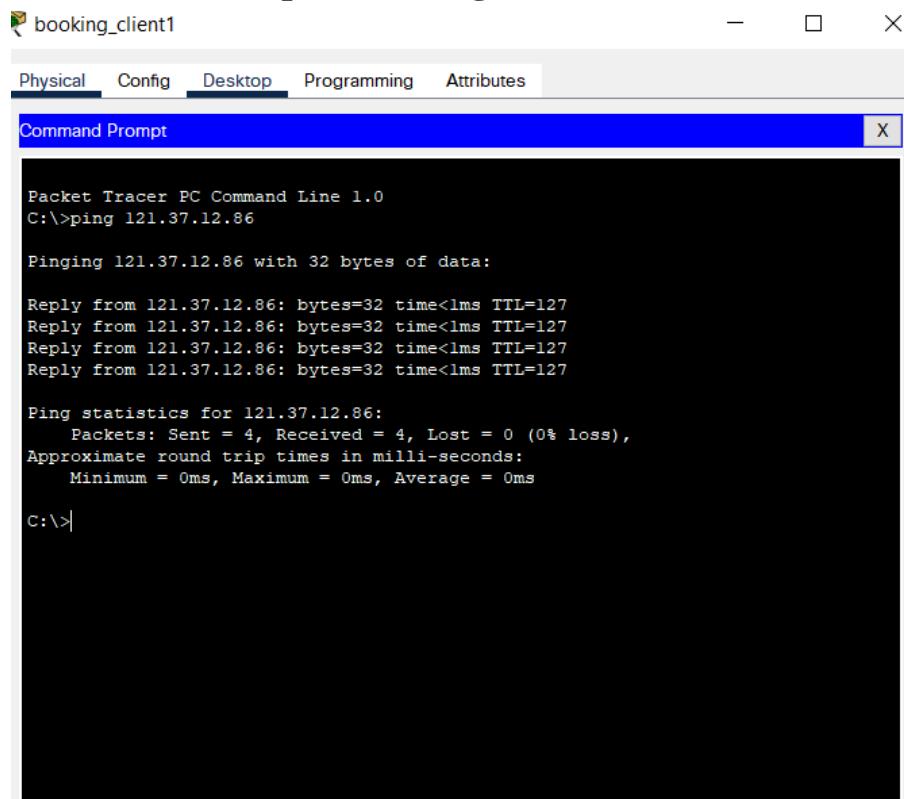
## Reservation



Packet Tracer PC Command Line 1.0  
C:\>ping 121.37.12.67  
  
Pinging 121.37.12.67 with 32 bytes of data:  
  
Reply from 121.37.12.67: bytes=32 time<1ms TTL=128  
  
Ping statistics for 121.37.12.67:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 0ms, Average = 0ms  
  
C:\>

## Connectivity between department

### Reservation and post-booking



Packet Tracer PC Command Line 1.0  
C:\>ping 121.37.12.86  
  
Pinging 121.37.12.86 with 32 bytes of data:  
  
Reply from 121.37.12.86: bytes=32 time<1ms TTL=127  
  
Ping statistics for 121.37.12.86:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 0ms, Average = 0ms  
  
C:\>

## Travel agency and post-booking

travel\_client 5

Physical Config Desktop Programming Attributes

Command Prompt

```
Packet Tracer PC Command Line 1.0
C:\>ping 121.37.12.84

Pinging 121.37.12.84 with 32 bytes of data:

Request timed out.
Reply from 121.37.12.84: bytes=32 time=4ms TTL=125
Reply from 121.37.12.84: bytes=32 time=3ms TTL=125
Reply from 121.37.12.84: bytes=32 time=5ms TTL=125

Ping statistics for 121.37.12.84:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 5ms, Average = 4ms

C:\>ping 121.37.12.84

Pinging 121.37.12.84 with 32 bytes of data:

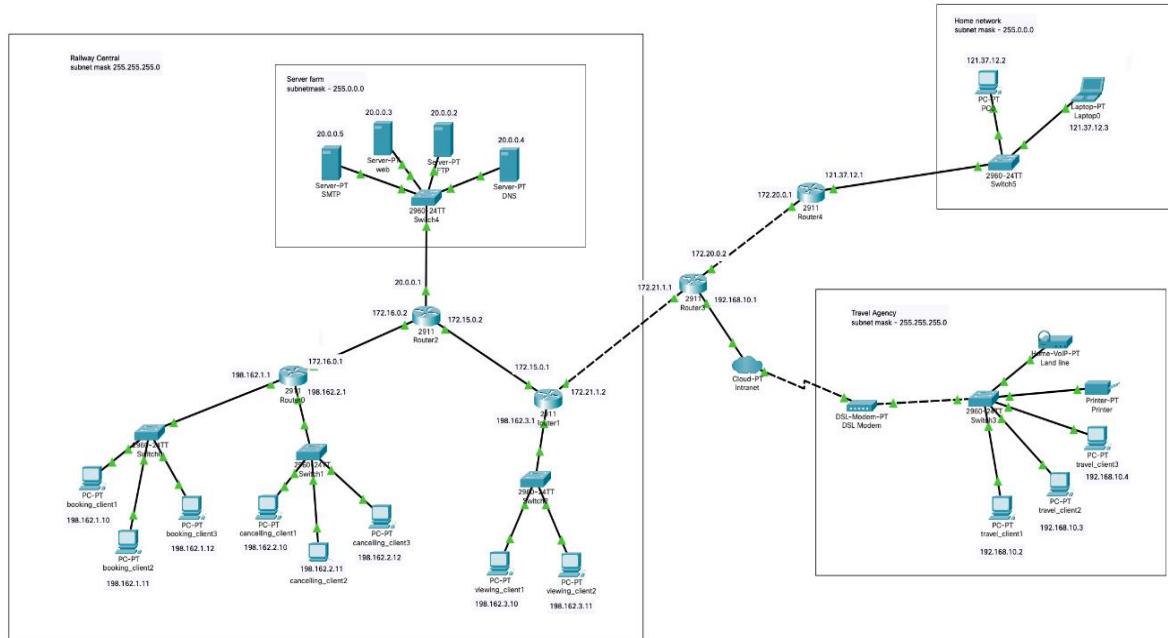
Reply from 121.37.12.84: bytes=32 time=21ms TTL=125
Reply from 121.37.12.84: bytes=32 time=23ms TTL=125
Reply from 121.37.12.84: bytes=32 time=2ms TTL=125
Reply from 121.37.12.84: bytes=32 time=2ms TTL=125

Ping statistics for 121.37.12.84:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 23ms, Average = 12ms

C:\>
```

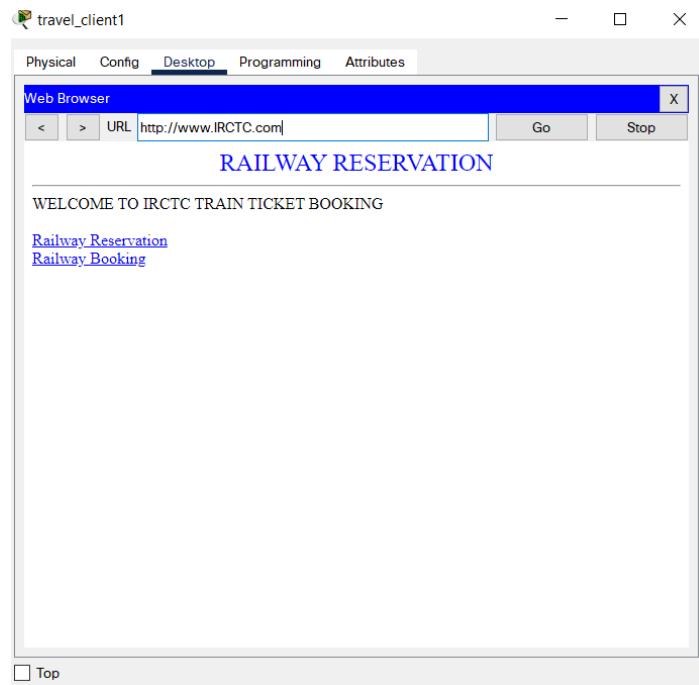
Top

## Implementation of Cloud for a network which is similar to the above network



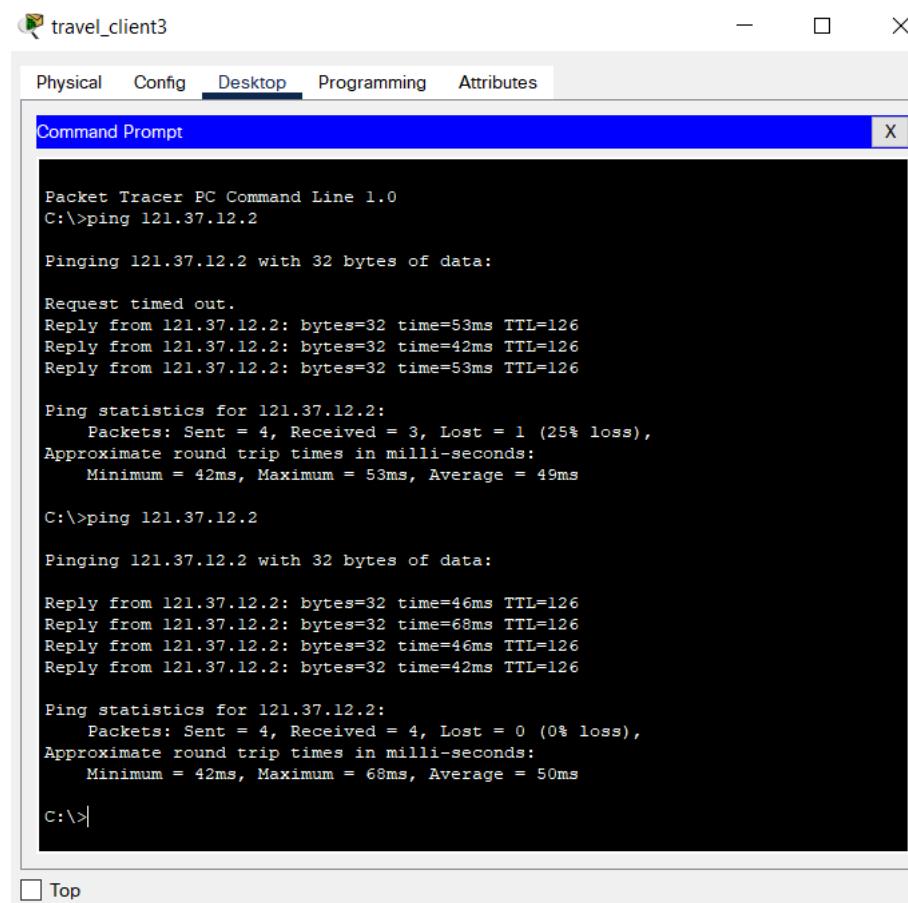
Configurations of Mail server, FTP server, DNS server and Web server are same as above model

## Accessing webserver from user1 via cloud

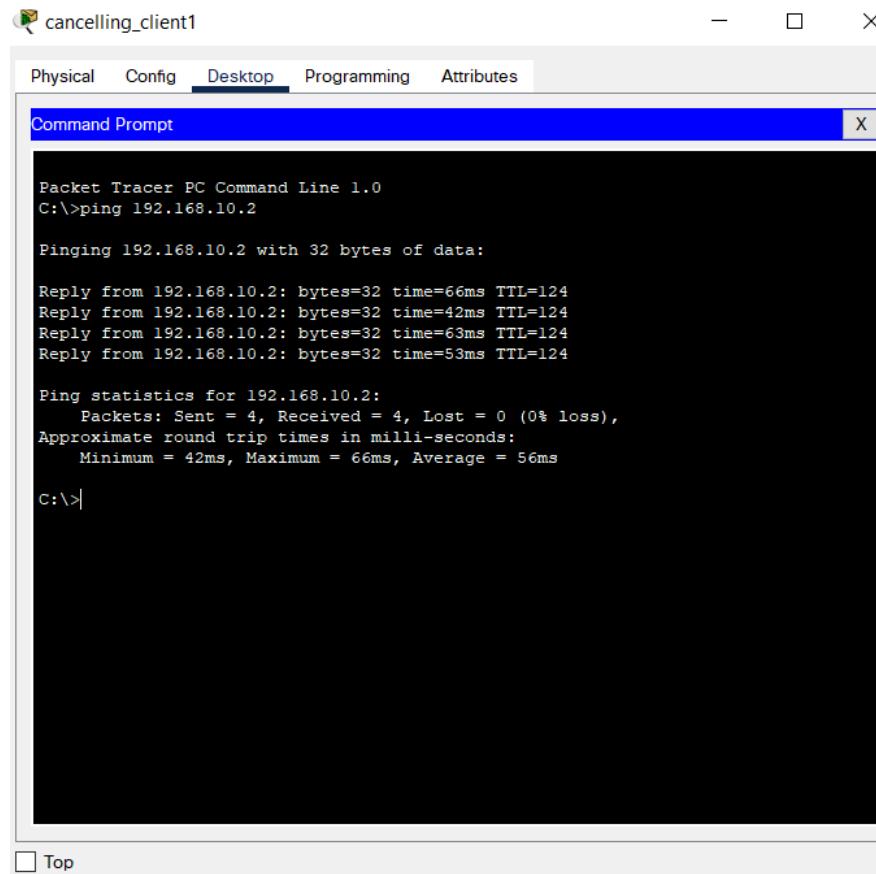


We can access the web server and all other servers from out of the network via cloud

## Similarly, we can ping computers outside the network via cloud



## Vice versa is also possible



The screenshot shows a Windows Command Prompt window titled "cancelling\_client1". The window has tabs for Physical, Config, Desktop, Programming, and Attributes, with "Desktop" selected. The main area displays the output of a "ping" command:

```
Packet Tracer PC Command Line 1.0
C:\>ping 192.168.10.2

Pinging 192.168.10.2 with 32 bytes of data:
Reply from 192.168.10.2: bytes=32 time=66ms TTL=124
Reply from 192.168.10.2: bytes=32 time=42ms TTL=124
Reply from 192.168.10.2: bytes=32 time=63ms TTL=124
Reply from 192.168.10.2: bytes=32 time=53ms TTL=124

Ping statistics for 192.168.10.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 42ms, Maximum = 66ms, Average = 56ms

C:\>
```

## Cloud computing in Railways

Indian Railways is the largest rail network in Asia and the world's second largest rail network which is governed by the Central Railway Ministry of India. It connects over from these 17 million passengers, only 1 million passengers' travels with confirmed tickets and 16 million passengers don't bother about confirmed tickets for travel, which causes a heavy monetary loss to Central Railways. Therefore, we will discuss how cloud computing may help Central Railways to avoid this loss by implementing some techniques and tools.

Transforming the Railway system to I-Railway System is difficult but not impossible. IRCTC (Indian Railway Catering and Tourism Corporation) is already using a next-generation cloud-based e-ticketing system which brings a phenomenal shift in internet rail ticketing by significantly improving the end-user experience in terms of ease of use, flexibility, security, cost, and convenience. The system uses advanced fraud control and security management tools. IBM assisted China Railway to migrate towards cloud computing-based technology using its cloud Infrastructure-as-a-Service (IaaS) software. An Indian IT firm TCS has already engaged in large-scale partnerships with Indian Railways by

developing an ERP (Enterprise Resource Planning) system for the IRCTC in the year 2006-2007.

## **WHY CLOUD COMPUTING?**

In cloud computing, the word cloud is used as a metaphor for "the Internet," so the phrase cloud computing means "a type of Internet-based computing," where different services—such as servers, storage, and applications are delivered to an organization's computers and devices through the internet. Therefore, Cloud computing may help I-RS in the following ways:

### **A. Cost Reduction**

Cloud computing technology can greatly help in reducing the costs as resources are acquired on a demand basis and charged accordingly.

### **B. Better usage of Personnel**

Since most of the interaction and work happens through the cloud itself and once the cloud is set up, manual interaction requirement is very minimal. Hence, employed personnel can focus on other things that will deliver value in spite of spending time on the maintenance of hardware and software.

### **C. Scalability**

Cloud computing technology can scale itself up or down in terms of resources. It can scale up, when there is more resource requirement and scale down when the resources are no longer required. This way resources can be effectively managed across the cloud and also, the user is charged based on the used resources.

## **CLOUD SERVICE MODELS**

### **A. Infrastructure-as-a-Service (IaaS)**

Infrastructure-as-a-Service is the first layer and foundation for cloud computing. Using this service model, we can manage our applications, data, operating system, middleware, and runtime environment. This service provider manages virtualization, servers, networking, and storage. This allows to avoid expenditure on hardware and human capital; reduce ROI (Return of Investment) risk, streamline, automate scaling and users only have to pay for the used resources. This means that the extra data processing space is available to us

whenever we need it, and when we don't need it, we are not paying for it which saves money and in turn that will fulfill Central Railways' existing needs.

## B. Platform-as-a-Service (PaaS)

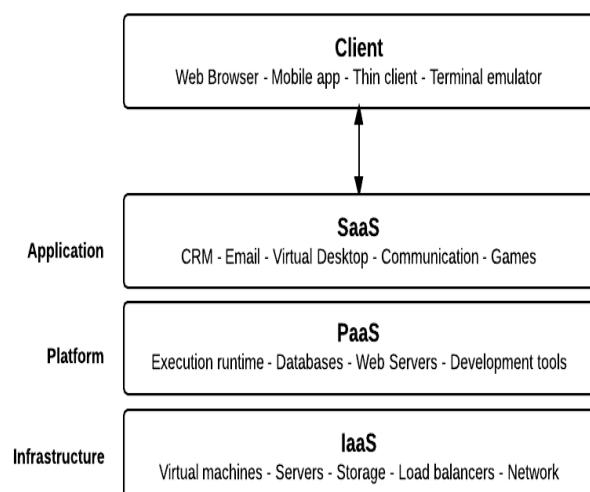
This cloud service model could be considered as the second layer. Applications and data are to be managed by the user and the cloud vendor manages everything else. The benefits of using PaaS include a streamlined version of deployment and ability to change or upgrade with minimized expenses.

The services of Central Railways in application testing or development might find PaaS as beneficial to eliminate the cost for upkeep of hardware. In this model, Central Railways may also be benefited as there is no need to hire people for maintenance of these systems. A scalable processing center is available at your disposal and that can be used as per needs (here also, we have to pay only for what we use).

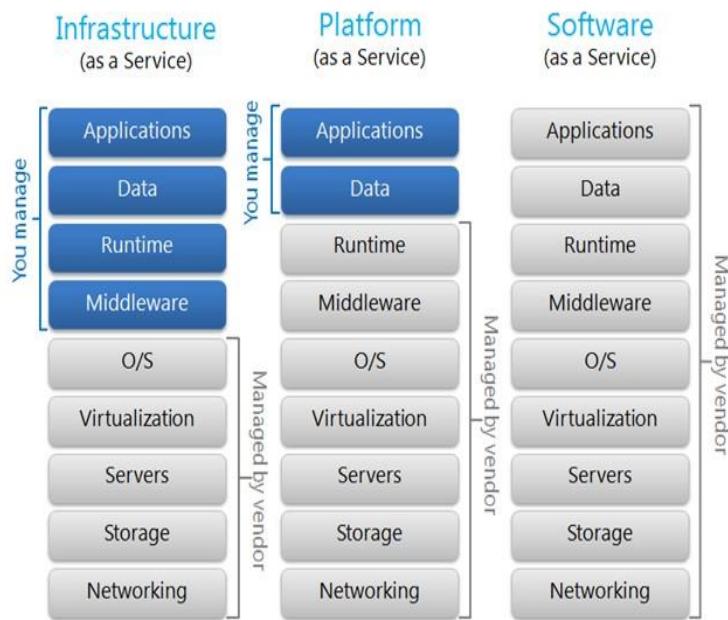
## C. Software-as-a-Service (SaaS)

This is the final layer of the cloud services model. This can be used by Central Railways to run programs in the cloud where all portions are managed by the cloud vendor. The users will be assured of compatibility and ease of collaboration as the software will be used by everyone. Central Railways won't have to pay for extra licensing fees and new users can be added easily. By using SaaS users will be able to access the software from a variety of devices, in their office or on go, which in turn develops easy collaboration among all users.

## Basic Architecture of Cloud Service Models



## Working of Cloud Service Models



## CLOUD DEPLOYMENT MODELS

### A. Public Cloud

A public cloud is a cloud computing model in which services, such as applications and storage, are available for general use over the internet. Public cloud services may be offered on a pay-per-usage model or other purchasing models.

### B. Private Cloud

A private cloud is virtualized as a data center that operates within a firewall. Private clouds are joined together by mass quantities of IT infrastructure into resource pools, and are privately owned and managed.

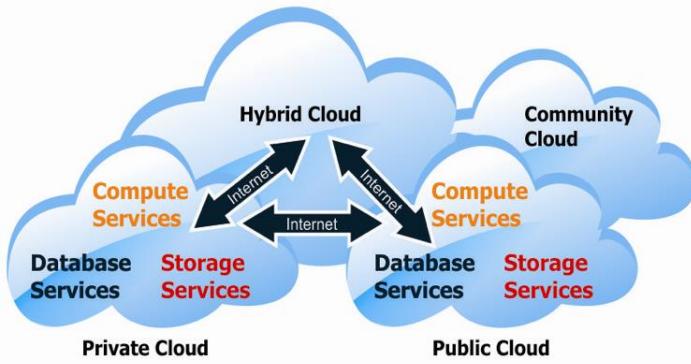
### C. Hybrid Cloud

A hybrid cloud is a mix of public and private clouds.

### D. Community Cloud

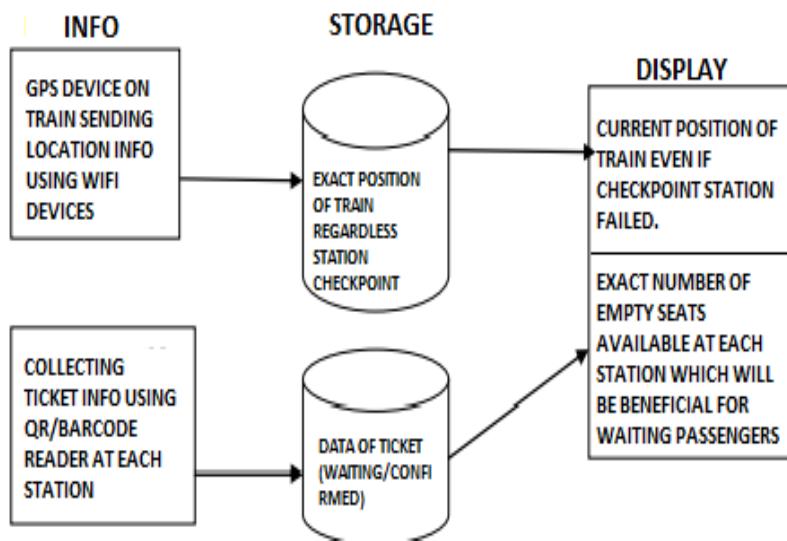
A community cloud is an infrastructure shared by several organizations which supports a specific community.

## Types of Cloud Deployment Models

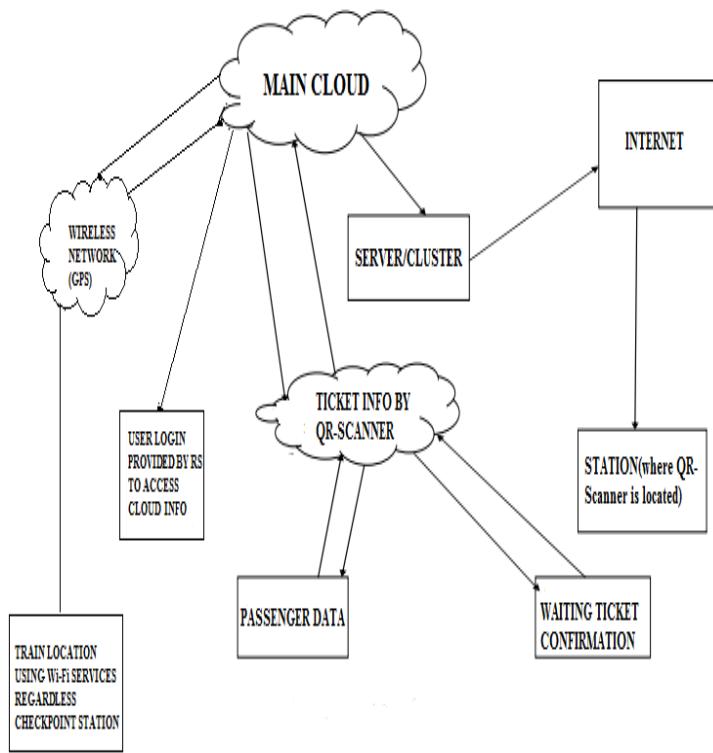


## IMPROVEMENT IN RAILWAYS USING CLOUD COMPUTING

The IT technology needs a vast and immediate improvement according to the new scenario. Some new IT equipment and techniques should be introduced like Tickets with unique QR (Quick Response) - Code and ticket checking with QR Scanner. Also, Wi-Fi enabled GPS (Global Positioning System) devices should be set up on each train, so that the exact location can be tracked without worrying about station checkpoints. So, everything is dependent on two main tasks viz. checking of tickets and allotting confirmed seats to waiting for holder according to priority and this can be done using cloud storage with cloud computing concepts as shown in the following figure.



## WORKING MODEL OF I-RS CLOUD



## CHALLENGES FOR USING CLOUD COMPUTING IN I-RS

**A. Possible downtime** - Cloud computing makes businesses dependent on the reliability of available internet connections. Here, in our proposed system, it depends on the quality of Wi-Fi services to be used by Central Railway. If they use services of quality service providers then this issue won't harm our proposed cloud computing technology on I - RS.

**B. Security issues** – As discussed earlier, cloud computing means internet-based computing. So, you are not supposed to use cloud computing applications that involve using or storing data that may harm the proposed system or affect its security. So, by doing filtering data using different type's security filters can resolve this issue.

## **CONCLUSION**

While planning the use of cloud computing initiatives in the future, Central Railways should choose service and delivery models that will match with requirements for operational flexibility, cost savings, and efficiency. Central Railways should adopt a gradual evolutionary approach towards cloud computing services. We can envision that I-RS will have an application portfolio with a mix of cloud-based services delivered across a combination of private, hybrid, and public cloud-based deployment models with the share of cloud services gradually increasing in this services mix. Private clouds are expected to become an increasing deployment model for cloud services among Railways, giving Central Railways full control through ownership and operations of their cloud systems. Hence, using cloud computing technology in Railway systems is an efficient, cost deterrent, and serializable technique for waiting for ticket holders. Therefore, implementing this service will help the government as well as passengers in terms of fiscal aspects.