A

Theme Based Project Report

on

**PACK UR BAGS**

Submitted for partial fulfilment of the requirements for the award of the degree of

**BACHELOR OF ENGINEERING**

in

**COMPUTER SCIENCE AND ENGINEERING**

By

**J.Avinash 2451-21-733-015**

**S.Lakshmi prasanna 2451-21-733-017**

**D.Bhagya 2451-21-733-027**

Under the guidance of

**Mr.B.VENKATA RAMANA**

Assistant Professor

Department of CSE



**MATURI VENKATA SUBBA RAO(MVSR) ENGINEERING COLLEGE**

**(An Autonomous Institution)**

Department of Computer Science and Engineering

(Affiliated to Osmania University & Recognized by AICTE)

Nadergul, Balapur Mandal, Hyderabad – 501 510

Academic Year: 2022-2023

**MATURI VENKATA SUBBA RAO(MVSR) ENGINEERING COLLEGE**

**(An Autonomous Institution)**

Department of Computer Science and Engineering

(Affiliated to Osmania University & Recognized by AICTE)

Nadergul, Balapur Mandal, Hyderabad – 501 510



**CERTIFICATE**

*This is to certify that the Theme Based project work entitled “****PACK UR BAGS****” is a bonafide work carried out by* ***J.AVINASH (2451-21-733-015), S.LAKSHMI PRASANNA (2451-21-733-017), D.BHAGYA (2451-21-733-027)*** *in partial fulfilment of the requirements for the award of degree of* ***Bachelor of Engineering*** *in* ***Computer Science and Engineering*** *from* ***Maturi Venkata Subba Rao(MVSR) Engineering College,*** *affiliated to OSMANIA UNIVERSITY, Hyderabad, during the Academic Year 2022-2023. under our guidance and supervision.*

*The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.*

**Internal Guide Head of the Department**

Mr B.Venkata Ramana Prof J Prasanna Kumar

Assistant Professor Professor & Head

Department of CSE Department of CSE

MVSREC. MVSREC.

i

**DECLARATION**

This is to certify that the work reported in the present Theme Based project entitled **“PACK UR BAGS”** is a record of bonafide work done by us in the Department of Computer Science and Engineering, M.V.S.R. Engineering College, Osmania University. The reports are based on the work done entirely by us and not copied from any other source.

The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

**J.AVINASH S.LAKSHMI PRASANNA D.BHAGYA**

**2451-21-733-015 2451-21-733-017 2451-21-733-027**

ii

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude and indebtedness to our project guide **Mr. B.VENKATA RAMANA, Assistant Professor** for his valuable suggestions and interest throughout the course of this project.

We are also thankful to our principal **Dr. G. Kanaka Durga** and **Prof. J Prasanna Kumar,** Professor and Head, Department of Computer Science and Engineering, MVSR Engineering College, Hyderabad for providing excellent infrastructure and a nice atmosphere for completing this project successfully as a part of our B.E. Degree (CSE).

We convey our heartfelt thanks to the lab staff for allowing us to use the required equipment whenever needed.

Finally, we would like to take this opportunity to thank our family for their support through the work. We sincerely acknowledge and thank all those who gave directly or indirectly their support in completion of this work.

**J.Avinash 2451-21-733-015**

**S.Lakshmi prasanna 2451-21-733-017**

**D.Bhagya 2451-21-733-027**

iii

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**VISION**

To impart technical education of the highest standards, producing competent and confident engineers with an ability to use computer science knowledge to solve societal problems.

**MISSION**

To make the learning process exciting, stimulating and interesting.

To impart adequate fundamental knowledge and soft skills to students.

To expose students to advanced computer technologies in order to excel in engineering practices by bringing out the creativity in students.

To develop economically feasible and socially acceptable software.

**PROGRAM EDUCATIONAL OBJECTIVES (PEOs)**

The Bachelor’s program in Computer Science and Engineering is aimed at preparing graduates who will:-

PEO-1: Achieve recognition through demonstration of technical competence for successful execution of software projects to meet customer business objectives.

PEO-2: Practice life-long learning by pursuing professional certifications, higher education or research in the emerging areas of information processing and intelligent systems at a global level.

PEO-3: Contribute to society by understanding the impact of computing using a multidisciplinary and ethical approach.

**Program Specific Outcomes(PSOs)**

PSO1: Demonstrate competence to build effective solutions for computational real-world problems using software and hardware across multi-disciplinary domains.

PSO2: Adapt to current computing trends for meeting the industrial and societal needs through a holistic professional development leading to pioneering careers or entrepreneurship.

iv

**COURSE OBJECTIVES AND OUTCOMES**

Course Objectives:

* + - To enhance practical and professional skills.
    - To familiarize tools and techniques of systematic literature survey and documentation.
    - To expose the students to industry practices and teamwork.
    - To encourage students to work with innovative and entrepreneurial ideas.

Course Outcomes:

Student will be able to

* + - Demonstrate the ability to synthesize and apply the knowledge and skills acquired in the academic program to the real-world problems.
    - Evaluate different solutions based on economic and technical feasibility.
    - Effectively plan a project and confidently perform all aspects of project management.
    - Demonstrate effective coding, written, presentation and oral communication skills.

v

**ABSTRACT**

Our project PackUrBags is a Bus Management system which provides a glance view of our Hyderabad RTC and the details regarding all buses present in our Hyderabad. The interface is very user-friendly it is coded in python.

Bus Management System i.e., PackUrBags is flexible and easy to use and is designed and developed to deliver real conceivable benefits from Hyderabad RTC to the user. It is designed to cover a wide range of bus administration and management processes. It is an integrated end-to-end Bus Management System that provides relevant information about the buses to support effective decision-making for travelling unknown distances, bus administration, and knowing the buses and their bus stops, in a seamless flow.

This project has been implemented using AWS (Amazon Web Services). It is a free and open source website and is a public cloud so the resources are shared among all the users. It offers a broad set of global cloud based products including, compute, storage, database, analytics networking, IoT, security, management tools, development tools and enterprise applications. Huge amount of data can be experienced by the customers according to their needs.

The purpose of this project is to develop a computerized bus management system that will upgrade the quality of information management and efficiency of the Hyderabad RTC and also for the user to get genuine opinion on the bus and their bus stops by going through the website.

**1. INTRODUCTION**

* 1. PROBLEM STATEMENT

Our project PackUrBags is a website that provides a glance view of buses of Hyderabad RTC and the details regarding buses present in our Hyderabad.

A good bus shall always prioritize the passengers' comfort and convenience, no matter the stage of their journey. They focus on ensuring the passengers receiving comfort and safe journey through roads.

Elements on our website are:

->It enables 24/7 online presence

->Cost effective, user friendly

->Info about the services we provide

->Specialised bus details like bus number, source, destination and type of bus

* 1. OBJECTIVES

The objectives of our website PackUrBags are:

* To provide an online website
* To enable the user to get a glance view Hyderabad bus services
* To enable the user to know the details regarding the all buses in Hyderabad
* To provide systematic way of creating a bus management website
* It available 24/7 and there is no registration charges
* This web application can be developed by using object oriented programming languages for font end like python which

will provide the latest technology in developing quite user friendly

user interface so it is very easy for all users to understand the system.

* 1. MOTIVATION

The real motivation for our bus management system project is to make a easy process of all management process like getting information of specialised buses, like bus numbers, bus stops, type of a bus, fare of a journey.

We always see that to get an bus struggle of a passenger a lot for getting information of the bus, knowing the bus stops when bus may be available sometimes the bus may not be available. This results in wastage of huge amount of time and you gonna lose your mettings. So by this website it will become easy to manage all this process.

So now by taking the motivation of this scenario which was regularly done in bus stops we are designing this website which can be benefited for the passengers. So, we’d like to process this system which help passengers to reach fast and effective.

**2. System Requirement Specifications**

2.1 Software Requirements

Software requirement is the most important things to be installed.

Table 2.1.1 shows the software requirement to develop PackUrBags.

**Table 2.1.1 Software Requirements**

|  |  |  |
| --- | --- | --- |
| SOFTWARE | DESCRIPTION | COST |
| Windows Notepad 11.2210.5.0 | Website coding editing tool | Free |
| Python | Website Interface | Free |
| Ms Office Version: 18.2210.1203.0 | Documentation Tools | Free |
| AWS lab 2 | Platform for deploying the website | Free |
| Google Chrome, Internet Explorer | Web Browser | Free |
| Git Hub | Platform for dumping the code | Free |

2.2 Hardware Requirements

To develop this system, hardware requirement is important things to be required. Table 2.1.2 shows the software requirement to develop PackUrBags.

**Table 2.1.2 Hardware Requirements**

|  |  |
| --- | --- |
| HARDWARE | DESCRIPTION |
| Laptop with 8GB RAM | Workstation for the project |
| Hard disk space (227GB) | To save data |

**3. TOOL USED**

The tool we are using is AWS (Amazon Web Services)

**The salient features of this tool are:**

1. Most secure, Easy to use, Flexible, Cost effective
2. **Communication**: the remainders will be received in the form of emails, messages
3. **Big data analytics:** huge amount of Data can experienced by the passengers in according to their needs
4. **Disaster recovery**: with cloud computing it is fast and relatively simple to launch your applications and data services in the cloud and get your business backup running in no time.
5. **Data security**: it is easy to implement and provides maximum data security
6. **Storage:** With cloud storage we can get as much as space as you require and store as much as data you need

4. **System Design and Implementation**

Here we will use Amazon Virtual Private Cloud (VPC) to create own VPC and add additional components to produce a customized network. We will also create a security group. We will then configure and customize an EC2 instance to run a web server and will launch the EC2 instance to run in a subnet in the VPC.

**Amazon Virtual Private Cloud (Amazon VPC)** enables to launch Amazon Web Services (AWS) resources into a virtual network that is defined. This virtual network closely resembles a traditional network that would operate in our own data centre, with the benefits of using the scalable infrastructure of AWS. We can create a VPC that spans multiple Availability Zones.

We should be able to do the following:

* Create a VPC.
* Create subnets.
* Configure a security group.
* Launch an EC2 instance into a VPC.

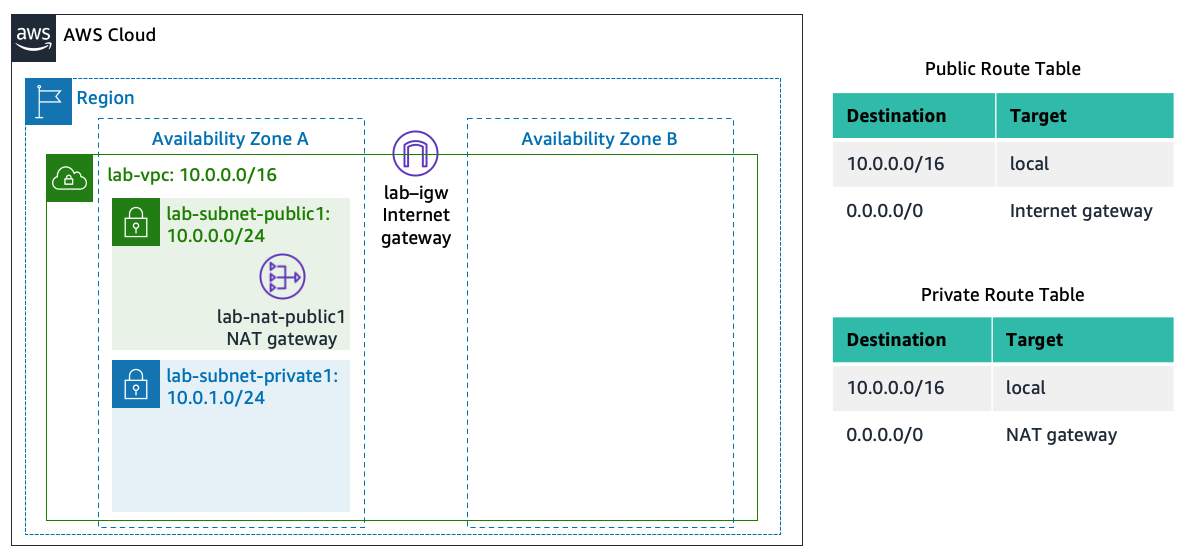
Steps to create a VPC:

Use the VPC and more option in the VPC console to create multiple resources, including a VPC, an Internet Gateway, a public subnet and a private subnet in a single Availability Zone, two route tables, and a NAT Gateway.

Configure the VPC details in the *VPC settings* panel:

* + Choose **VPC and more**.
  + Under **Name tag auto-generation**, keep *Auto-generate* selected, however change the value from project to lab.
  + Keep the **IPv4 CIDR block** set to 10.0.0.0/16
  + For **Number of Availability Zones**, choose **1**.
  + For **Number of *public* subnets**, keep the **1** setting.
  + For **Number of *private* subnets**, keep the **1** setting.
  + Expand the **Customize subnets CIDR blocks** section
    - Change **Public subnet CIDR block in us-east-1a** to 10.0.0.0/24
    - Change **Private subnet CIDR block in us-east-1a** to 10.0.1.0/24
  + Set **NAT gateways** to **In 1 AZ**.
  + Set **VPC endpoints** to **None**.
  + Keep both **DNS hostnames** and **DNS resolution** *enabled*.

The diagram 4.1 summarizes the VPC resources created and how they are configured.



**Image 4.1**

**Create Additional Subnets**

**Create subnet** then configure:

* + **VPC ID:** **lab-vpc**
  + **Subnet name:** lab-subnet-public2

**Availability Zone:** Select the *second* Availability Zone (for example, us-east-1b)

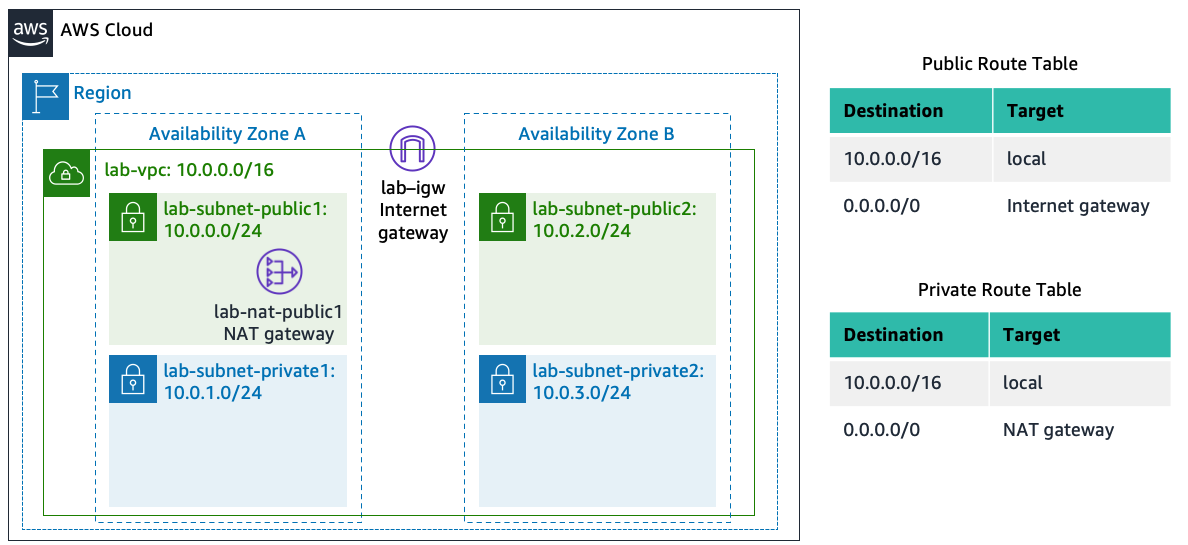
* + **IPv4 CIDR block:** 10.0.2.0/24

The subnet will have all IP addresses starting with **10.0.2.x**.

**Create subnet** then configure:

* + **VPC ID:** lab-vpc
  + **Subnet name:** lab-subnet-private2
  + **Availability Zone:** Select the *second* Availability Zone (for example, us-east-1b)
  + **IPv4 CIDR block:** 10.0.3.0/24

The subnet will have all IP addresses starting with **10.0.3.x**.



**Image 4.2**

**Create a VPC Security Group**

create a VPC security group, which acts as a virtual firewall. launch an instance, associate one or more security groups with the instance. can add rules to each security group that allow traffic to or from its associated instances.

**Security group name:** Web Security Group

* + **Description:** Enable HTTP access
  + **VPC:** choose the X to remove the currently selected VPC, then from the drop down list choose **lab-VPC**

Configure the following settings:

* + **Type:** *HTTP*
  + **Source:** *Anywhere-IPv4*
  + **Description:** Permit web requests

**Launch a Web Server Instance**

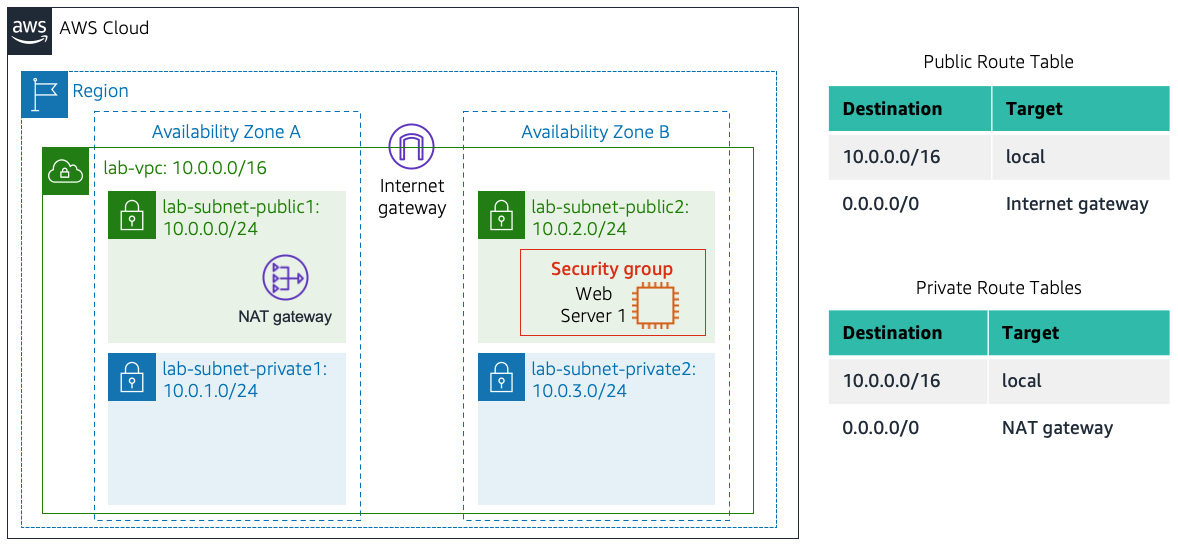
launch an Amazon EC2 instance into the new VPC configure the instance to act as a web server.

 Name the instance->Choose an AMI from which to create the instance->Choose an Instance type ->Select the key pair to associate with the instance

Configure the Network settings:

* + Next to Network settings, choose **Edit**, then configure:
    - **Network:** *lab-VPC*
    - **Subnet:** *lab-subnet-public2* (*not* Private!)
    - **Auto-assign public IP:** *Enable*
  + Next, you will configure the instance to use the *Web Security Group* that you created earlier.
    - Under Firewall (security groups), choose **Select an existing security group**.
    - For **Common security groups**, select **Web Security Group**.

This security group will permit HTTP access to the instance.

The complete architecture you deployed is given below in figure 4.3 **Figure 4.3**

5.**Conclusion**

The aim of this PackUrBags website project is to construct an online system by use of which a user can know the details of Hyderabad RTC buses. This system can be beneficial for the bus passengers. As the website will be based on the internet, the user can get information about the buses and also solutions to their problems and also deals with the collection of bus’s information, bus details, etc. Having a website for Hyderabad RTC buses means, passengers are always able to find their bus routes anytime and anywhere. This means that they can get all the information every time at anywhere. website will continue to serve passengers and secure new ones.

6. References

Hyderabad city bus: http://www.hyderabadcitybus.com/

Hyderabad bus routes: http://www.hyderabadbusroutes.in/show-buses/all

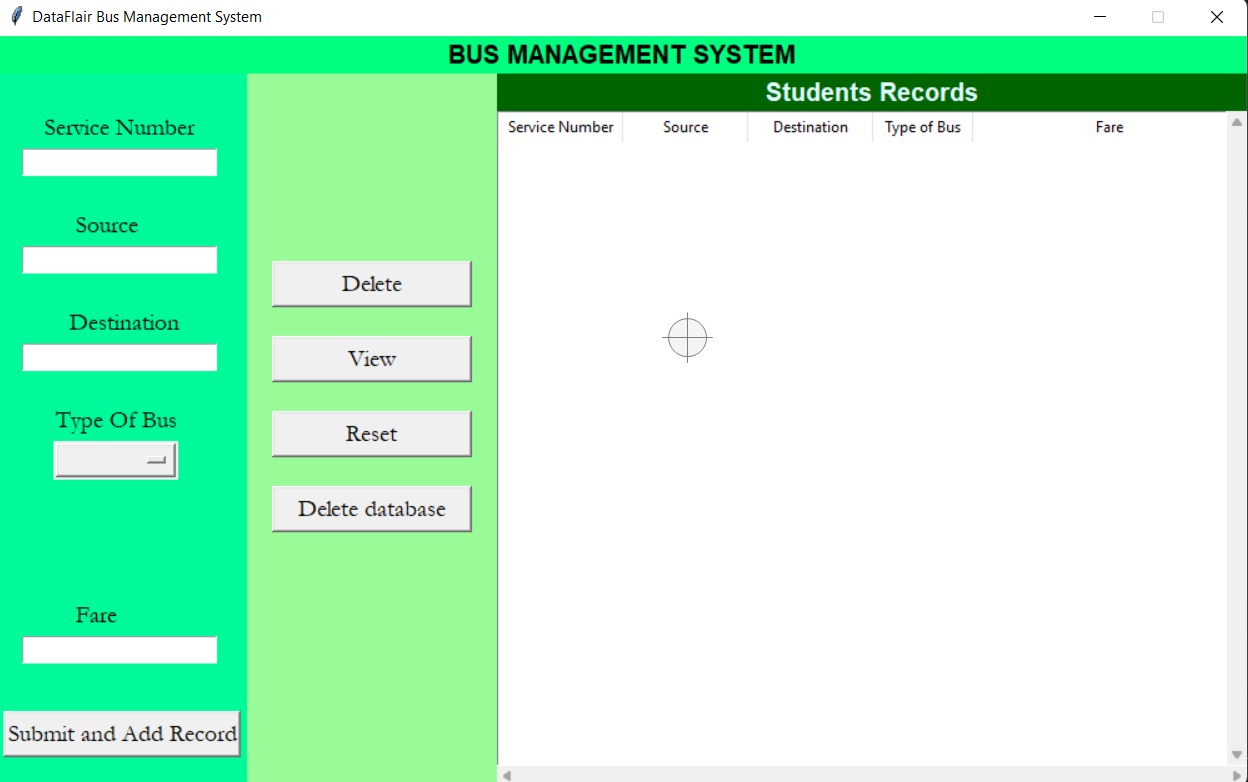
PythonCode:<https://data-flair.training/blogs/python-school-students-management-system/>

Deployment in AWS: 1) https://youtu.be/3sQhVKO5xAA

2) https://youtu.be/foCG\_mH8bxk

3) https://youtu.be/Iv\_dECet\_oM

7. Results:



APPENDIX: INDEX PAGE CODE

import datetime

from tkinter import \*

import tkinter.messagebox as mb

from tkinter import ttk

from tkcalendar import DateEntry # pip install tkcalendar

import sqlite3

import streamlit as st

import sys

import os

# Creating the universal font variables

headlabelfont = ("Noto Sans CJK TC", 15, 'bold')

labelfont = ('Garamond', 14)

entryfont = ('Garamond', 12)

# Connecting to the Database where all information will be stored

connector = sqlite3.connect(r'BusManagement.db')

cursor = connector.cursor()

connector.execute(

"CREATE TABLE IF NOT EXISTS BUS\_MANAGEMENT (SERVICE\_NO INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, SOURCE TEXT, DESTINATION TEXT, TYPE\_OF\_BUS TEXT, FARE INTEGER)"

)

# Creating the functions

def reset\_fields():

global service\_no\_strvar, source\_strvar, destination\_strvar, type\_of\_bus\_strvar, fare\_strvar

for i in ['service\_no\_strvar', 'source\_strvar', 'destination\_strvar', 'type\_of\_bus\_strvar', 'fare\_strvar']:

exec(f"{i}.set('')")

def reset\_form():

global tree

tree.delete(\*tree.get\_children())

reset\_fields()

def display\_records():

tree.delete(\*tree.get\_children())

curr = connector.execute('SELECT \* FROM BUS\_MANAGEMENT')

data = curr.fetchall()

for records in data:

tree.insert('', END, values=records)

def add\_record():

global service\_no\_strvar, source\_strvar, type\_of\_bus\_strvar, destination\_strvar,fare\_strvar

serviceno= service\_no\_strvar.get()

source = source\_strvar.get()

destination = type\_of\_bus\_strvar.get()

typeofbus = destination\_strvar.get()

fare = fare\_strvar.get()

if not serviceno or not source or not destination or not typeofbus or not fare:

mb.showerror('Error!', "Please fill all the missing fields!!")

else:

try:

connector.execute(

'INSERT INTO BUS\_MANAGEMENT (SERVICE\_NO,SOURCE,DESTINATION,TYPE\_OF\_BUS,FARE) VALUES (?,?,?,?,?)', (serviceno,source,destination,typeofbus,fare)

)

connector.commit()

mb.showinfo('Record added', f"Record of {serviceno} was successfully added")

reset\_fields()

display\_records()

except:

mb.showerror('Wrong type', 'The type of the values entered is not accurate. Pls note that the contact field can only contain numbers')

def remove\_record():

if not tree.selection():

mb.showerror('Error!', 'Please select an item from the database')

else:

current\_item = tree.focus()

values = tree.item(current\_item)

selection = values["values"]

tree.delete(current\_item)

connector.execute('DELETE FROM BUS\_MANAGEMENT WHERE SERVICE\_NO=%d' % selection[0])

connector.commit()

mb.showinfo('Done', 'The record you wanted deleted was successfully deleted.')

display\_records()

def view\_record():

global service\_no\_strvar, source\_strvar, type\_of\_bus\_strvar, destination\_strvar, fare\_strvar

current\_item = tree.focus()

values = tree.item(current\_item)

selection = values["values"]

service\_no\_strvar.set(selection[1]); source\_strvar.set(selection[2])

type\_of\_bus\_strvar.set(selection[3]); destination\_strvar.set(selection[4])

fare\_strvar.set(selection[5])

# Initializing the GUI window

main = Tk()

main.title('DataFlair School Management System')

main.geometry('1000x600')

main.resizable(0, 0)

# Creating the background and foreground color variables

lf\_bg = 'MediumSpringGreen' # bg color for the left\_frame

cf\_bg = 'PaleGreen' # bg color for the center\_frame

# Creating the StringVar or IntVar variables

service\_no\_strvar = StringVar()

source\_strvar = StringVar()

type\_of\_bus\_strvar = StringVar()

destination\_strvar = StringVar()

fare\_strvar = StringVar()

# Placing the components in the main window

Label(main, text="BUS MANAGEMENT SYSTEM", font=headlabelfont, bg='SpringGreen').pack(side=TOP, fill=X)

left\_frame = Frame(main, bg=lf\_bg)

left\_frame.place(x=0, y=30, relheight=1, relwidth=0.2)

center\_frame = Frame(main, bg=cf\_bg)

center\_frame.place(relx=0.2, y=30, relheight=1, relwidth=0.2)

right\_frame = Frame(main, bg="Gray35")

right\_frame.place(relx=0.4, y=30, relheight=1, relwidth=0.6)

# Placing components in the left frame

Label(left\_frame, text="Service Number", font=labelfont, bg=lf\_bg).place(relx=0.175, rely=0.05)

Label(left\_frame, text="Source", font=labelfont, bg=lf\_bg).place(relx=0.3, rely=0.18)

Label(left\_frame, text="Destination", font=labelfont, bg=lf\_bg).place(relx=0.275, rely=0.31)

Label(left\_frame, text="Type Of Bus", font=labelfont, bg=lf\_bg).place(relx=0.22, rely=0.44)

Label(left\_frame, text="Fare", font=labelfont, bg=lf\_bg).place(relx=0.3, rely=0.7)

Entry(left\_frame, width=19, textvariable=service\_no\_strvar, font=entryfont).place(x=20, rely=0.1)

Entry(left\_frame, width=19, textvariable=type\_of\_bus\_strvar, font=entryfont).place(x=20, rely=0.23)

Entry(left\_frame, width=19, textvariable=source\_strvar, font=entryfont).place(x=20, rely=0.36)

Entry(left\_frame, width=19, textvariable=fare\_strvar, font=entryfont).place(x=20, rely=0.75)

OptionMenu(left\_frame, destination\_strvar, 'Luxury', "Ordinary").place(x=45, rely=0.49, relwidth=0.5)

Button(left\_frame, text='Submit and Add Record', font=labelfont, command=add\_record, width=18).place(relx=0.025, rely=0.85)

# Placing components in the center frame

Button(center\_frame, text='Delete', font=labelfont, command=remove\_record, width=15).place(relx=0.1, rely=0.25)

Button(center\_frame, text='View', font=labelfont, command=view\_record, width=15).place(relx=0.1, rely=0.35)

Button(center\_frame, text='Reset', font=labelfont, command=reset\_fields, width=15).place(relx=0.1, rely=0.45)

Button(center\_frame, text='Delete database', font=labelfont, command=reset\_form, width=15).place(relx=0.1, rely=0.55)

# Placing components in the right frame

Label(right\_frame, text='Students Records', font=headlabelfont, bg='DarkGreen', fg='LightCyan').pack(side=TOP, fill=X)

tree = ttk.Treeview(right\_frame, height=100, selectmode=BROWSE,

columns=('Service Number', "Source", "Destination", "Type of Bus", "Fare"))

X\_scroller = Scrollbar(tree, orient=HORIZONTAL, command=tree.xview)

Y\_scroller = Scrollbar(tree, orient=VERTICAL, command=tree.yview)

X\_scroller.pack(side=BOTTOM, fill=X)

Y\_scroller.pack(side=RIGHT, fill=Y)

tree.config(yscrollcommand=Y\_scroller.set, xscrollcommand=X\_scroller.set)

tree.heading('Service Number', text='Service Number', anchor=CENTER)

tree.heading('Source', text='Source', anchor=CENTER)

tree.heading('Destination', text='Destination', anchor=CENTER)

tree.heading('Type of Bus', text='Type of Bus', anchor=CENTER)

tree.heading('Fare', text='Fare', anchor=CENTER)

tree.column('#0', width=0, stretch=NO)

tree.column('#1', width=100, stretch=NO)

tree.column('#2', width=100, stretch=NO)

tree.column('#3', width=100, stretch=NO)

tree.column('#4', width=80, stretch=NO)

tree.place(y=30, relwidth=1, relheight=0.9, relx=0)

#if (st.button('Go')):

display\_records()

# Finalizing the GUI window

main.update()

main.mainloop