ADVANCE RAILWAY TICKETING SYSTEM

INDEX

SYNOPSIS OF TRAIN RESERVATION PROJECT

The project aims at creating a railway reservation module where the user can do online advance booking of tickets for any particular train for a particular day and cancellation of tickets by a GUI based menu driven program. Based on the availability and demand, the person will be given confirmed berth or RAC or Waiting list.

The module is being developed with menu driven options for various activities user can perform while accessing railway reservation system. The project uses Python programming language with Links to connect with MYSQL databases for various tables for storing and retrieving information as well as train information database. Also, various file operation functionality modules are also incorporated for passenger booking status, cancellation, E-ticket preparation as well chart preparation and berth allocation.

Various modules and functionality have been developed to bring modularity to the program code as follows

1) **Welcome module**

This is the starting screen setup of the train reservation module. This module has a pull-down menu driven options for user registration for new users, login option for existing users, checking for availability in a particular train for a given date of journey to plan the travel and exit option to close the module.

2) **User Registration module**

This module is developed with tkinter interface with python to accept user's name, user id, password and other details of the user for the successful creation of user account. Code is developed to check and alert empty entries for username, user id. Also, it checks for valid password entry with 8 alphanumeric characters with minimum one capital alphabet. Checks are also

incorporated to alert the user if the confirm password is not same as the password. Username, user id, password, confirm password are mandatory fields. If any of these are entries are empty the registration will be deemed unsuccessful. The registration details of the user is stored as record in SQL table named 'users'.

3) **Login module**

Successful User login is mandatory for new ticket booking and cancellation of ticket. This module is developed with tkinter interface with python. It queries the users table for authentication. Empty or Invalid user id or password will flash an alert and empty the entry boxes for re-entry of details. This module expects valid user id and password combination to allow the passenger to

i) Book a new ticket
ii) Cancel the ticket
iii) Print a E-ticket
iv) To view train availability at the time of booking.

Also, there is a facility for administrator to login with special user id and password to prepare chart for a particular train or reallocate seats of cancelled seats to senior citizens in a particular train for a particular date of Journey.

4) **View train availability**

This module allows the user to select the train and date of travel from a pull-down menu and check for availability of trains in various classes for a maximum period of one month. The train table is queried using chosen parameters.

5) **Booking Module**

The booking module allows the user to reserve tickets. The origin, destination and date of journey are taken as user input and are checked for validity. The date of booking is restricted to within a month prior to the date of journey and before 5 hours on the day of journey.

The trains are displayed along with the number of seats which are available in 3 classes: SL, 2AC, 3AC. The user can choose the class from the pull-down menu and is directed to the screen where he/she has to submit the appropriate details like age, name, food preference and sex of the passengers for whom the tickets are booked. The seats are allocated with special preference for senior citizens. There is also appropriate concession in ticket cost for children and senior citizens. The maximum number of tickets that can be booked per booking session is restricted to 5. An 8-digit PNR number is generated which the user can use in the future. The total cost of the ticket(s) is displayed upon successful booking.

6) **E-Ticket creation-**

In this module the passenger can print E-ticket by entering the 8-digit PNR number. For an invalid PNR number alert is flashed. For a valid PNR, the ticket details of the passengers are captured as a text file and converted into pdf format using a FPDF module and a prompt for successful creation of e-ticket will also be flashed.

7) **Cancellation module**

This module focuses on cancellation of tickets after logging in as an authenticated user. It asks for an 8-digit PNR and displays all the passengers scheduled to travel with that PNR number, Partial cancellation is done, meaning the user can cancel as many tickets

under the same PNR as they desire. The ticket status of the passenger whose ticket is cancelled will then be assigned ‘CAN’ in the passenger table and their details are updated in the Cancellation table.

8) **Chart preparation**

To prepare the list of passenger’s details (chart) for particular train and date of journey. In this module the administrator can select the particular train and date of journey from a pull-down menu. It queries the users table for the records. The details are displayed in a text box.

9) **Reallocation**

The reallocation module is linked with the booking and cancellation modules. It helps reallocate seats to a senior citizen. Reallocation is done only if cancellation is done on one of the lower births and a senior citizen occupies an upper birth.

SALIENT FEATURES

1. Modularity of program structure for the ease of navigation
2. GUI based Menu driven program
3. Integration of SQL database , Tkinter with Python code
4. Authentication Based login
5. Generation of e-ticket in pdf format
6. Partial cancellation of tickets.
7. Dynamic seat reallocation.

PROJECT ANALYSIS

This project is an online train reservation system developed using python language, using Tkinter for user-friendly interaction and MySQL database to store various tables required. The entire project is divided into various modules dedicated to specific tasks.

It also uses successful establishment of sql-python connection for accessing SQL tables from python. This project focuses on advance ticket booking, cancellation and viewing train availability based on user choice to facilitate booking of ticket. Also, there is a provision for reallocating cancelled lower berths to senior citizens who were initially allocated middle or upper berth of the same date of journey and class of travel.

Modules are developed for chart preparation as well as E-Ticket creation in pdf mode. This project simplifies online advance booking for any time and date up to 30 days before date of travel. Checks and balances were incorporated to ensure that only authorised users can book and cancel the ticket.

All the modules are developed as individual functions for easy coding and reusability.

Following python library modules were imported to use its various methods for successful code development:

a) Tkinter
b) MySQL connector
c) Datetime module
d) Csv
e) Tkcalendar
f) FPDF

Various sql tables used:

a) Admin table – to store user id and password of administrators
b) Users table – to store user registration details
c) Train table – to store various trains and its details with start and destination.
d) Passengers table – to store passenger booking details.
e) Cancellation table- to store the details of cancelled tickets.

Various files used:

a) Fare table file as a csv file – to store fare details of trains based on distance and class of travel.
b) E-ticket file as text file – to store booking information for a given pnr for ticket creation in a pdf form.

Explanation of various functions:

1) **ulogin()**
   This function opens a tkinter window with labels and empty text entry boxes for the user to enter their userid and password. It uses the authenticate() function to validate the entered details.
2) **authenticate()**
   This function authenticates the entered details with the already registered details in the users table in SQL database. A successful login prompt is shown for valid userid and password. For invalid details it clears up the entry text boxes and gives an alert prompt for the user to retry.
3) **createacc()**
   This function opens a tkinter window with labels and empty text entry boxes for the user to register their Name, Userid, Password, Confirm password,Phone number and Email id . It uses the submit() function to register their details.
4) **Submit()**
   This function registers the details of the person with the railway system. If Name, Userid or password is left empty then it gives an alert prompt. It requires the password to be 8 character, with minimum 1 capital alphabet and numbers. It also checks if the password and confirmpassword are same, if yes then account is created successfully. If not , then the entry text boxes are emptied and alert prompt is given.

5) **View()**

This function opens a tkinter window with pull down menu for the user to select the train and travel date for which they want to see availability.

6) **Getdates()** and **adgetdates()**

This function creates a pull down menu of the 30 dates from the current day . this function uses datetime module of python to get the date format .

7) **Textprint()**

This function opens a scrollable text box in which the required details are displayed.

8) **Print_answers()**

This function takes the selected values and does the required action. In case of view train availability, it calculates the seats available in the selected train and date dynamically. if no train or date is not selected it gives an alert prompt.

Similarly in case of eticket preparation it compares the entered values with already existing passengers table in SQL database. If no such pnr exists it will give an alert prompt. If it is valid pnr entry then it forwards the information to ticpdf() function.

9) **adminviewtr ()**

This function opens a tkinter window with options for the admin to select between chart preparation and reallocate the seats.

10) **adprint_answers()**

This function is used to prepare chart by the admin. The date of travel and the trains are given in a pull-down menu. If no train or date is not selected it gives an alert prompt.

11) **eticket()**

This function opens a tkinter window with labels and empty text entry boxes for the user to enter the pnr number.

This uses print_answers() and ticpdf () to create the ticket in pdf form. This function also shows a successful eticket created

prompt message on eticket creation. It store the generated eticket as a pdf file in the current working directory with 8 digit pnr code so that various etickets are distinguishable.

12) ticpdf()

This function uses FPDF module of python to create a empty Pdf page, set the font and copy all the details of all the passengers travelling with that PNR number.

13) book()

This is the main function of the booking module. It is called when the user selects the 'Book A Ticket' option.

14) ticketcost((trainno., class))

This function takes a tuple consisting of train number and class as its parameter. It calculates the ticket cost keeping in mind the base fare, distance, food preference and age of the passenger. The base fare and distance are obtained from the sql table 'trains'. The cost based on distance is calculated using data from the csv table 'fare.csv'. The ticket cost of each passenger is updated in the sql table 'passengers'.

Cost of food:

Food preference	Cost
Vegetarian	100
Non-Vegetarian	120
None	0

15) status ((trainno., class))

This function takes a tuple consisting of train number and class as a single parameter. It is used to determine the status of the ticket

booked based on the number of seats already occupied. This status is updated in the sql table ‘passengers’.

How status is decided:

Seats occupied	Status
<72	CNF
>=72 and <=79	RAC
>79 and <=83	WL

16) seatallocate((trainno.,class),age)

This function takes two arguments: a tuple consisting of the train number and class and the age of the passenger. Seat allocation is done based on these parameters and also keeping in mind the seats which are cancelled and already allocated. The lower berth is mostly allocated for senior citizens. The seat allocated is updated in the record of the corresponding passenger in the sql table ‘passengers’

17) allocateberth(class,seatno.)

This function takes two parameters: The class and the seat number of the passenger. The berth is assigned based on these parameters. The berth assigned to the given seat is updated in the record of the corresponding passenger in the sql table ‘passengers’.

18) pnrgenerate()

This function is used to generate a unique 8-digit PNR number per booking session. The user can use the PNR number for future reference.

19) bookprint([passenger details])

This function has a single parameter which consists of passenger details like PNR number, seat number, name of passenger, etc. This function is used to print the passenger booking details in a consolidated form at the end of the booking session.

20) passengerdet((trainno., class))

This function takes a tuple consisting of train number and class as its parameter. This function uses tkinter as its graphic user interface to obtain details of the passengers from the user. The number of passengers is restricted to 5 per booking session(by calling the function addmore()) and upon submitting the details, a message box is displayed acknowledging the successful booking of the ticket(s)(by calling the function nomore()).

21) submit_details()

The details of the passengers which the user has entered are inserted into the table ‘passengers’ along with the uniquely generated PNR number, seat number, seat type and the ticket cost.

22) availability()

This function is used to show the availability of seats in a particular coach/class in a particular train on a particular day. The user is then given the option to proceed or exit.

23) userinput()

This function is used to obtain the origin and destination of the travel from the user.

24) advancebooking()

This function is used to check if the booking can be done on the date of travel selected by the user using the tkcalender widget. The date of booking is restricted to within a month prior to the

date of journey and before 5 hours on the day of journey. If the user input does not meet these criteria, the user is asked to enter a valid date.

25) searchtrain()

This function is used to fetch the details of the trains for the origin, destination and date given by the user using appropriate query from the sql table 'trains'. If there are no trains satisfying the criteria given by the user, the user is prompted to re-enter the origin, destination and date of journey.

26) display_trains([details of the trains])

This function is called by the searchtrain() function to display the details of the trains which were queried.

27) Cancelatic()

This function is used to cancel tickets. Pnr number is used as the parameter, and when the pnr number is entered in the tkinter interface, all the primary details of all passengers booked under that pnr number is shown from the passengers table and the user can select one or more of them and cancel. This is called the partial cancellation system. Once the cancellation is done, the details of the passenger(s) will be updated in the cancellation table, and the ticket status of those passenger(s) in the passenger table will be updated to "CAN". Checks are placed to ensure the right pnr number is entered.

28) Reallocate()

This function reallocates a lower berth to a senior citizen who is originally assigned an upper birth in case one of the lower berths are cancelled.

For this purpose it initially checks for the validity of the cancellation by assuring if the name and pnr no. of the person cancelling is same

in both passenger and cancellation tables. If this is true, his/her status is set to “nb” in the cancellation table.

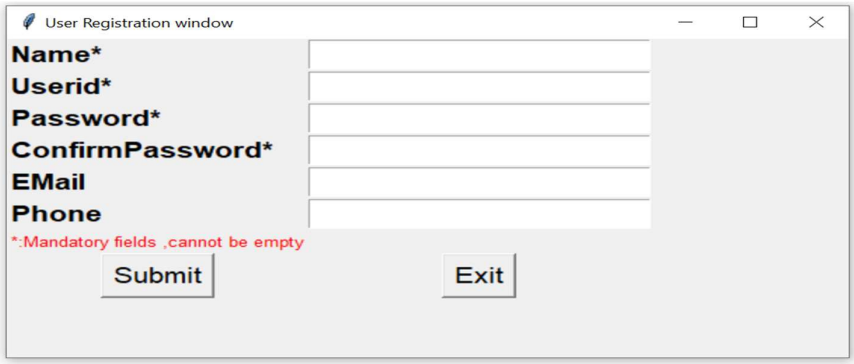
The module then checks if the following conditions are satisfied:

1. Person to whom reallocation is applied is a senior citizen i.e., above 60 years of age
2. Cancelled status is “nb”
3. Cancelled berth is a lower berth
4. The senior citizen is not already occupying a lower berth
5. Date of journey and train name are same for both the person who is cancelling as well as the senior citizen

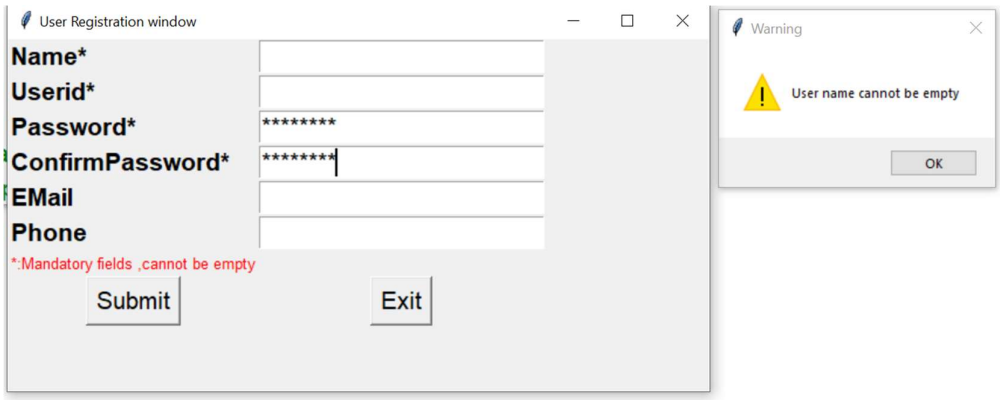
To avoid confusion, the module checks if the record of the person who is cancelling is available in both ‘passenger table’ and ‘cancellation table’. If the record is only in the cancellation table it implies that that particular seat number has already been allocated to another passenger (in the booking module).

OUTPUTS

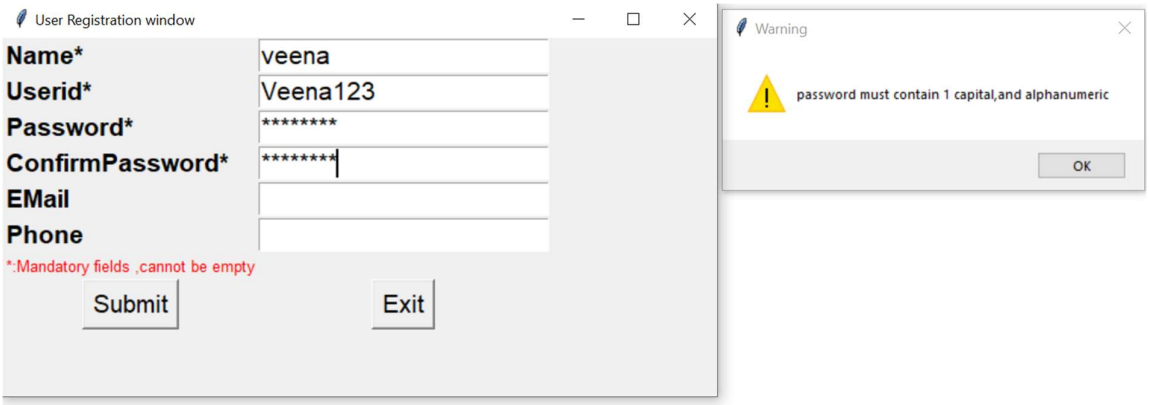
User registration screen



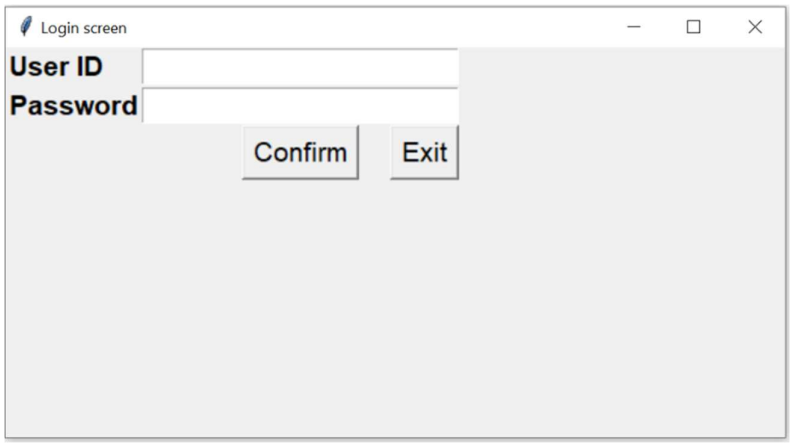
On entering empty values for name or userid



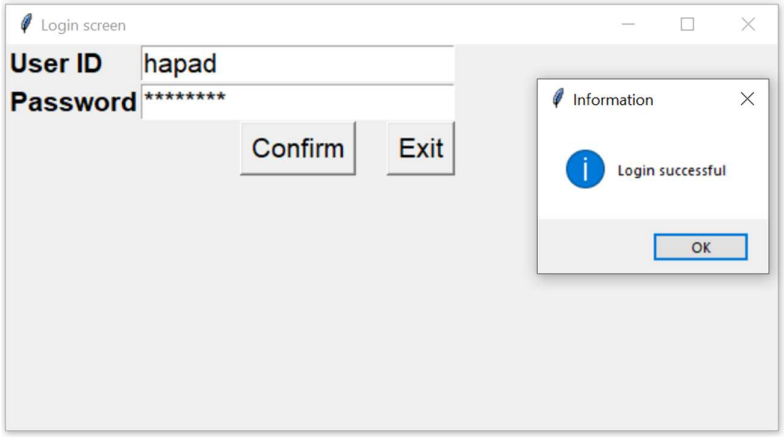
On entering password with wrong specifications



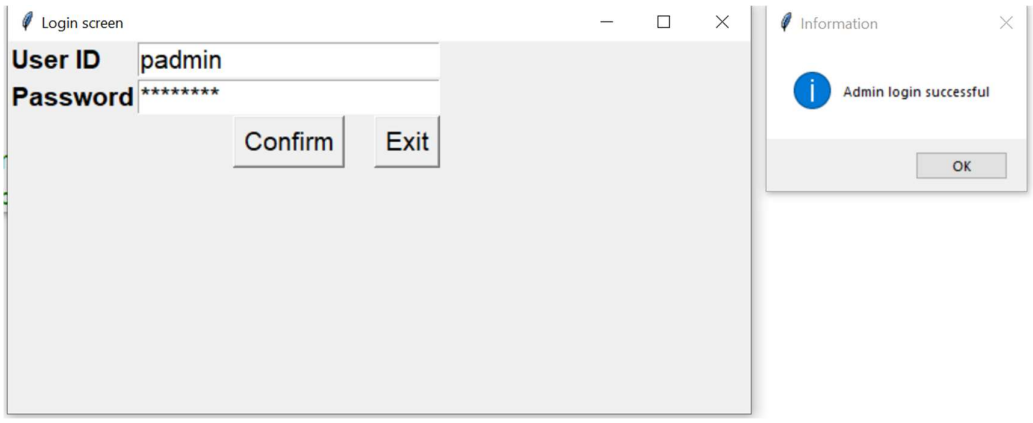
Login module



Successful login prompt:



Admin login screen



After logging in as admin

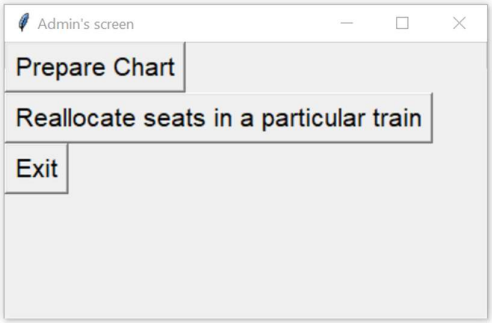
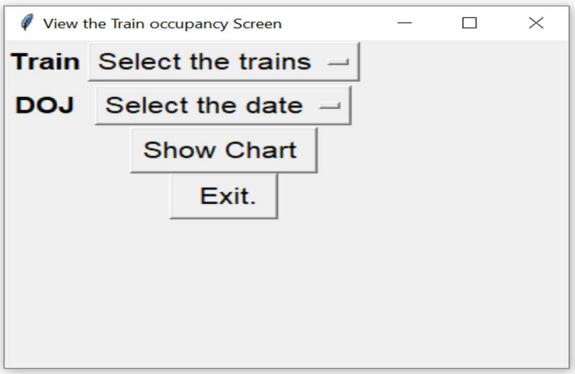
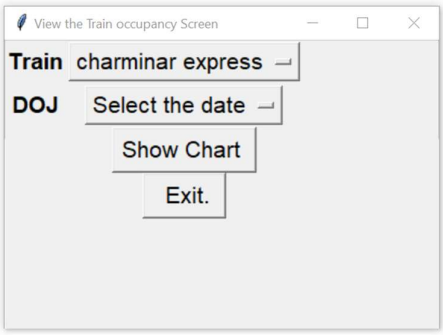
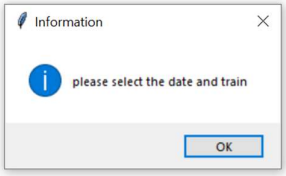


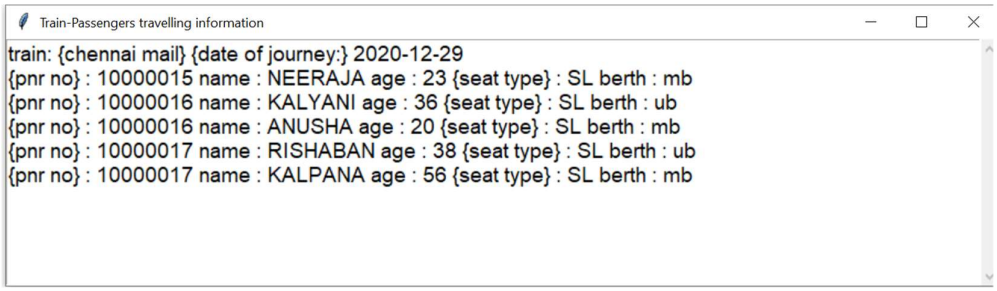
Chart prepared by admin



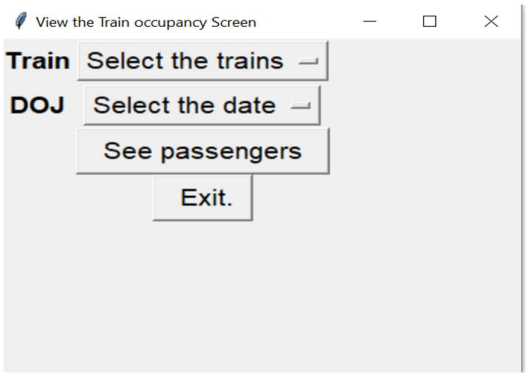
On not selecting date and train



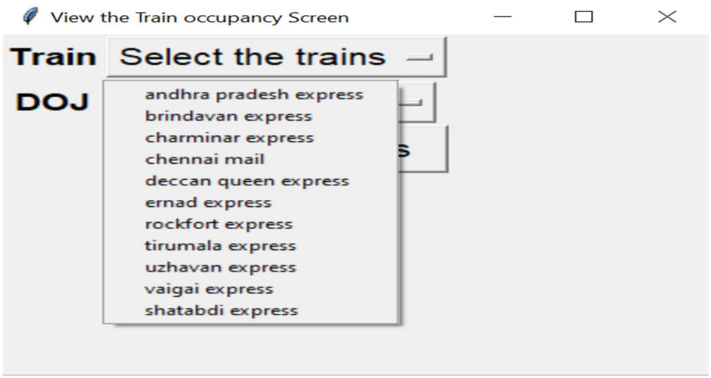




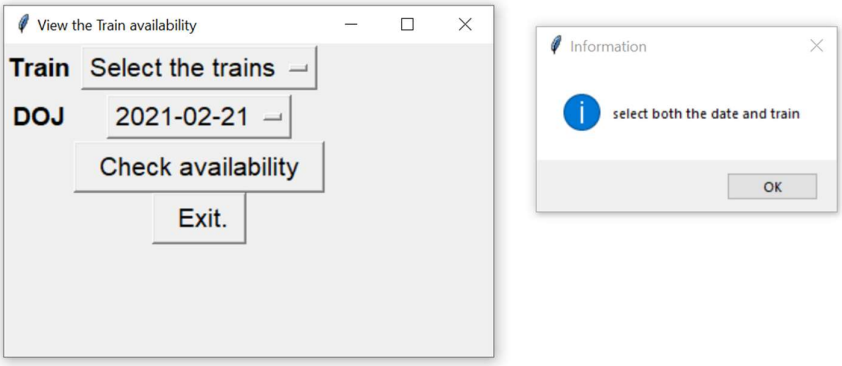
View train availability



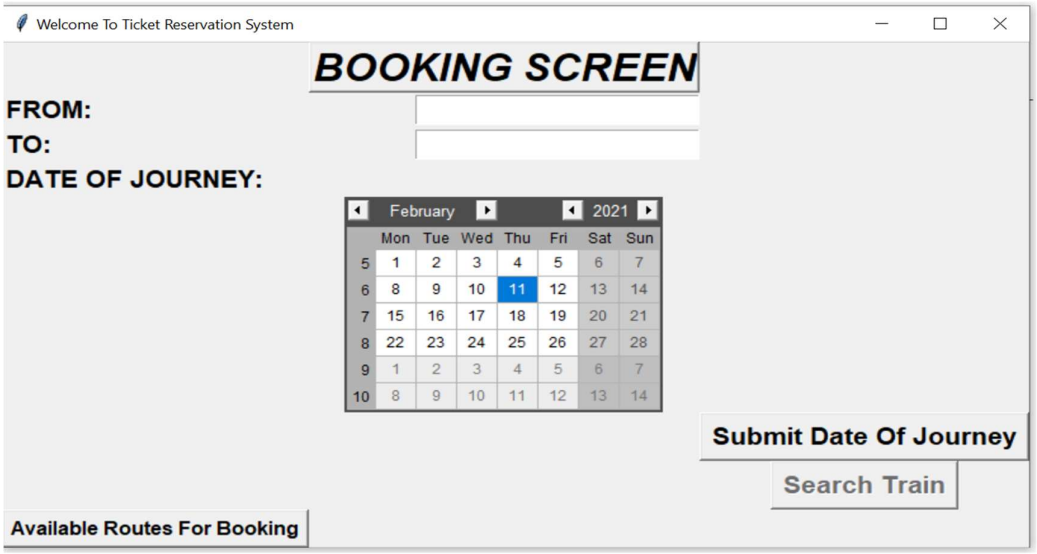
Pull down menu to select trains as well as date of travel



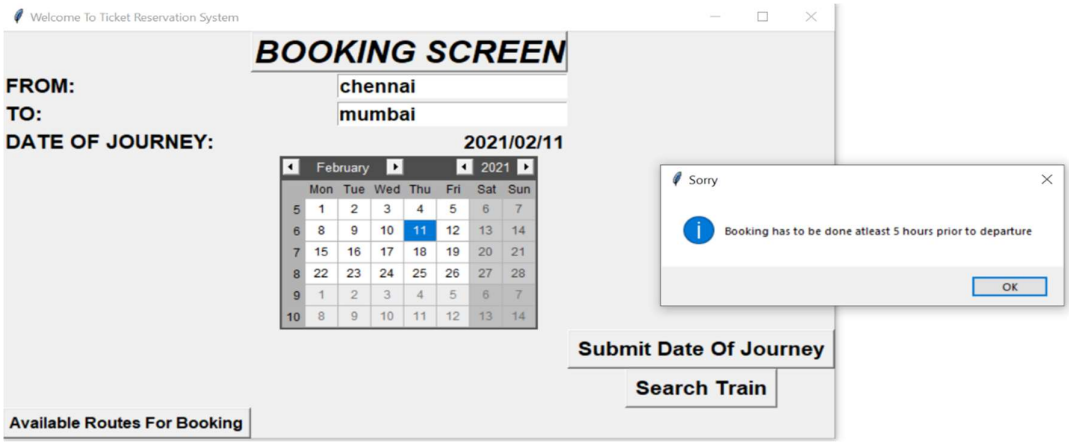
On not selecting date or train



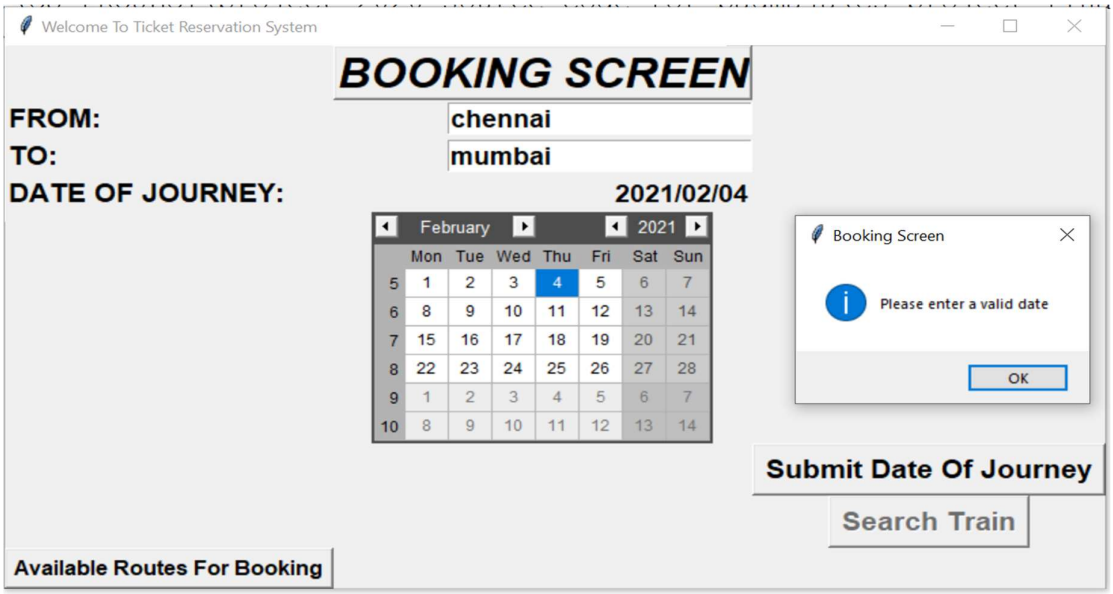
The screen visible to the user on selecting 'Book a ticket' option



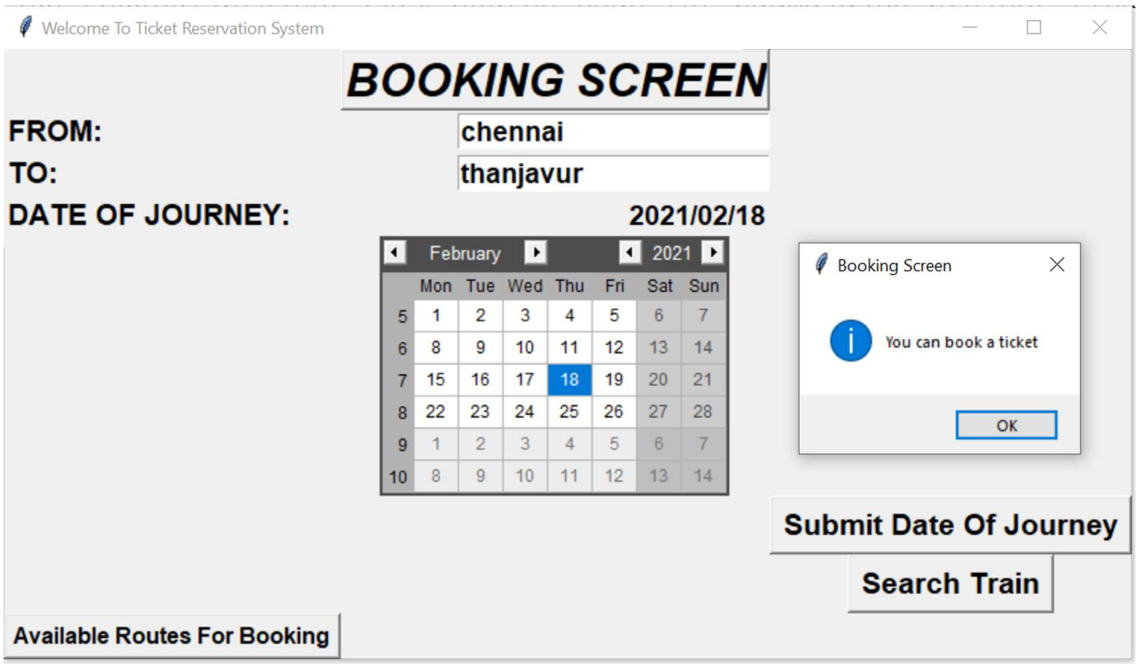
Message box displayed when user tries to book a ticket within 5 hours prior to the journey



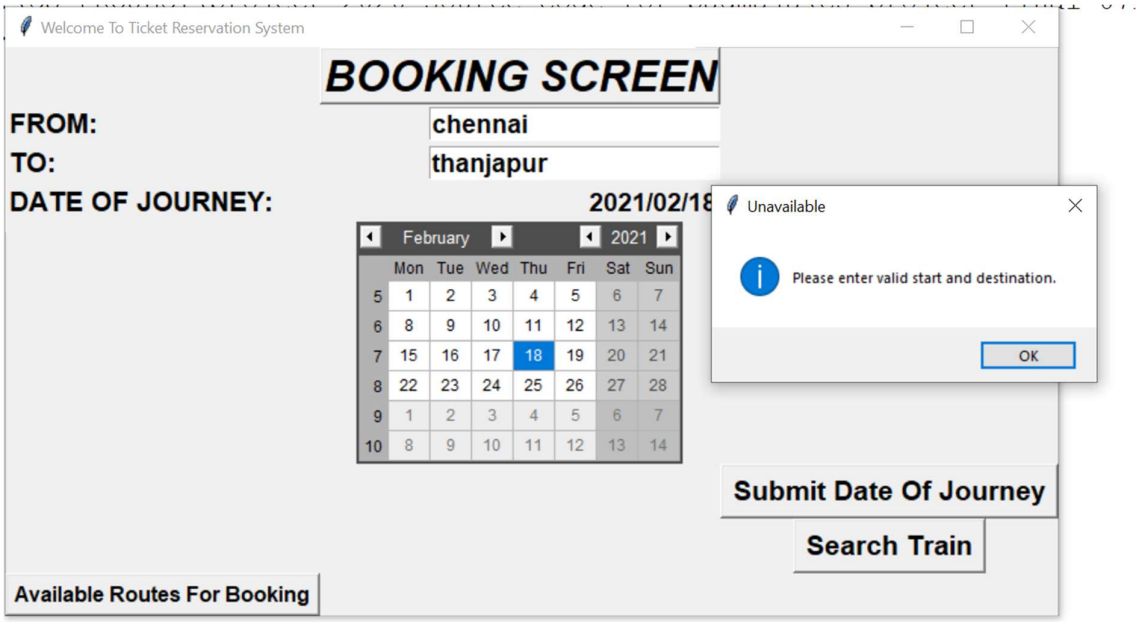
Message box displayed when user tries to input an invalid date or when he tries to book a ticket for a day more than a month from the day of booking



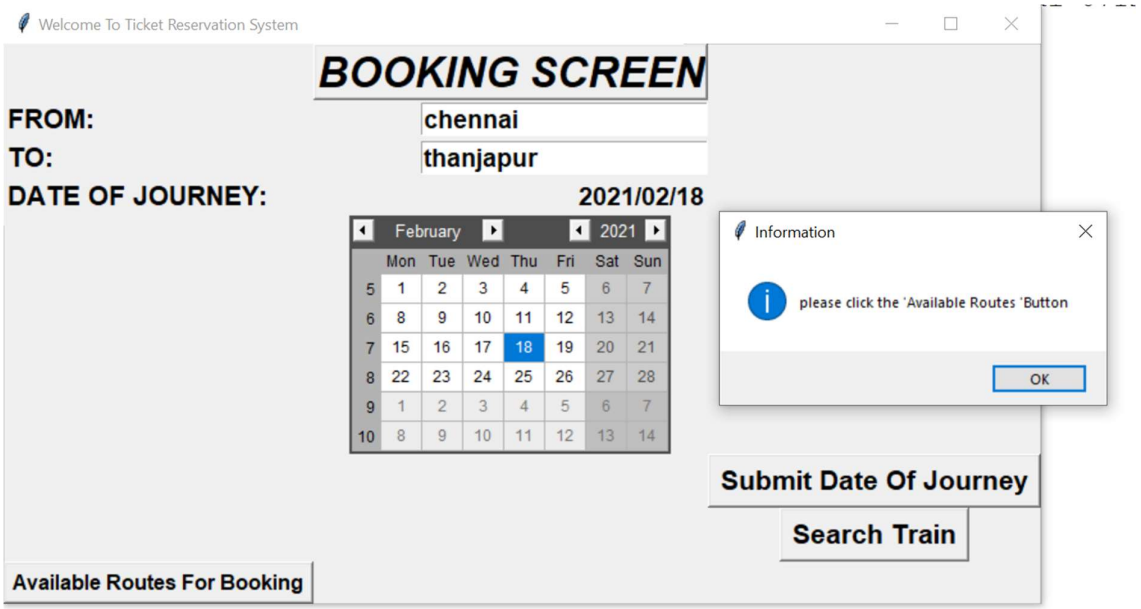
Message box displayed when the date selected by the user is valid



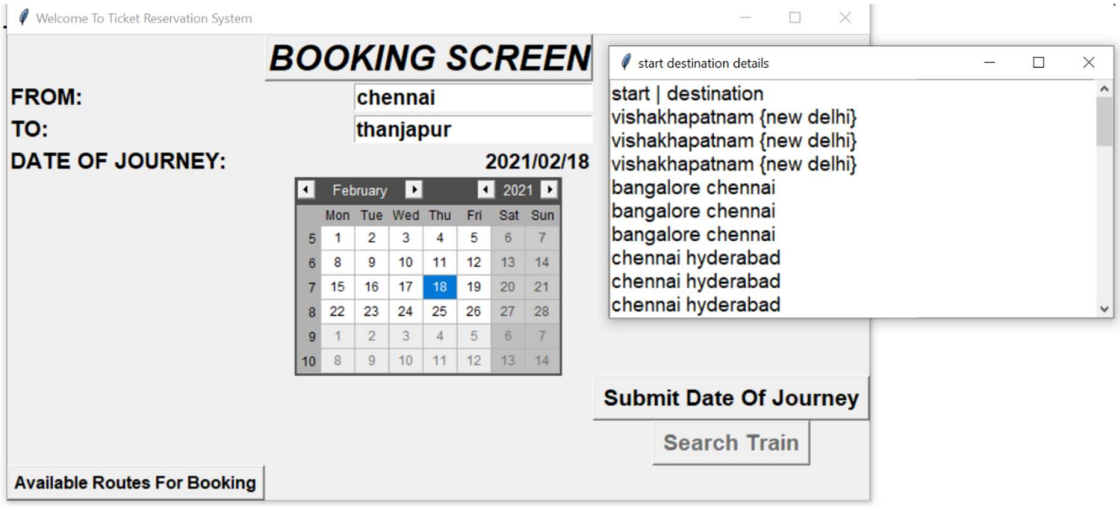
The message box which is displayed when the user enters invalid origin or destination



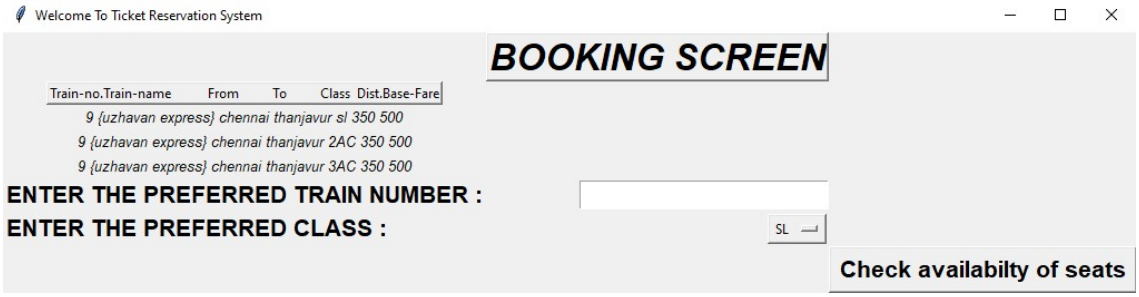
On clicking the 'OK ' button, new information box opens for further navigation



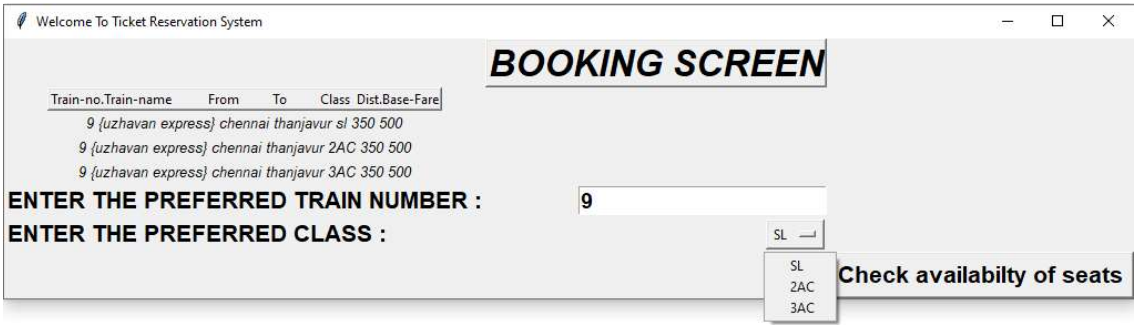
Text box displaying all the start and destination stations available on clicking available routes for booking button



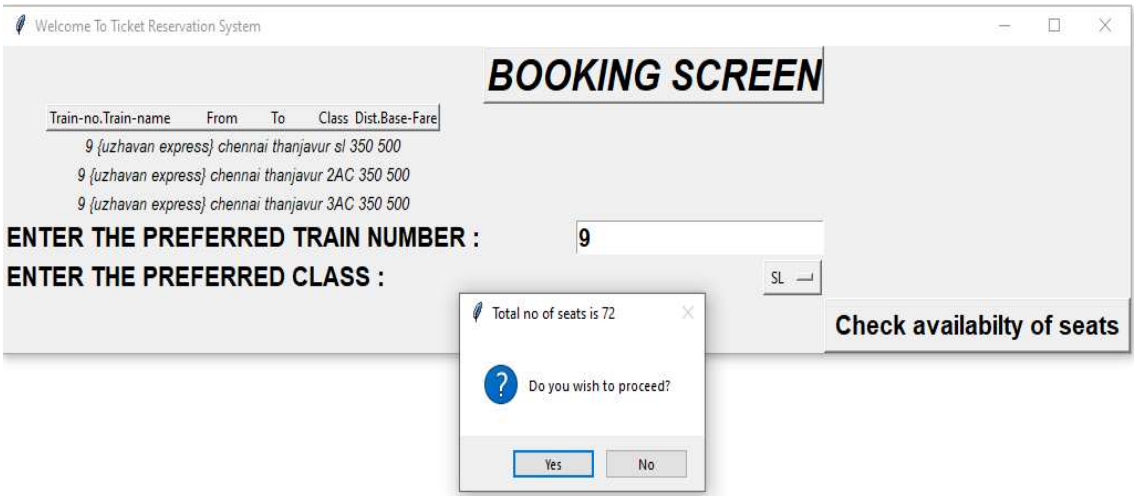
The screen which is visible on clicking the 'Search Train' button on the previous screen



The user is required to select the preferred class using the pull-down menu

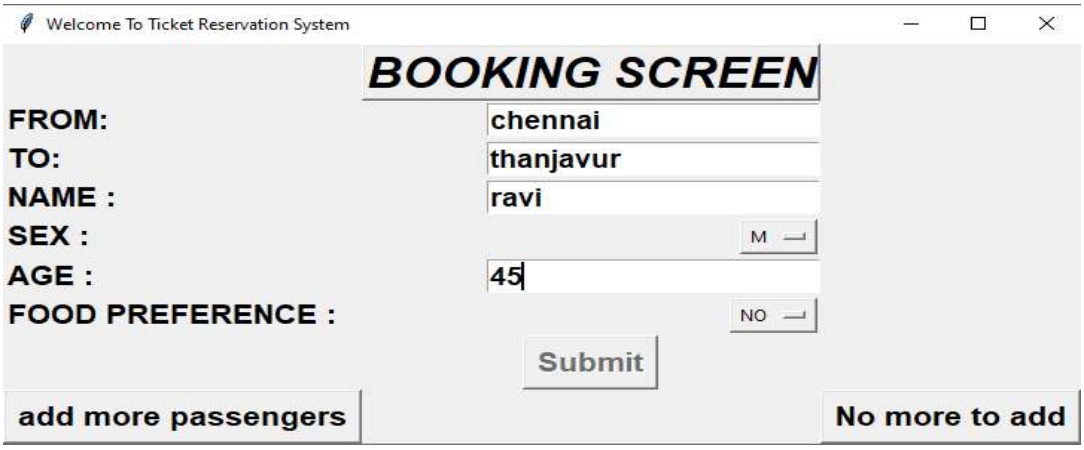


A message box shows the number of tickets available on the selected train of the selected class and also gives the user a choice as to whether he wants to proceed with the booking or not

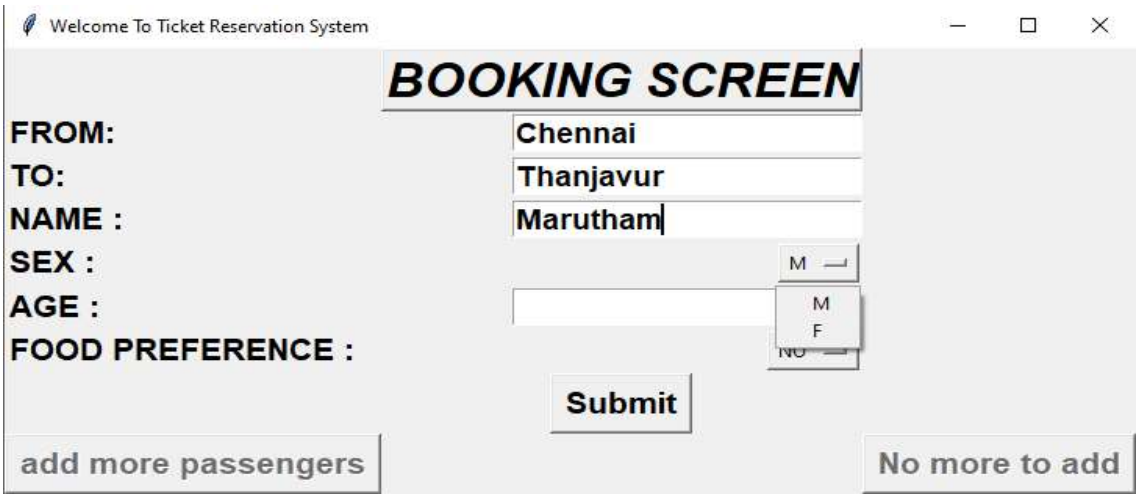


The message box displayed upon clicking 'No' in the previous screen

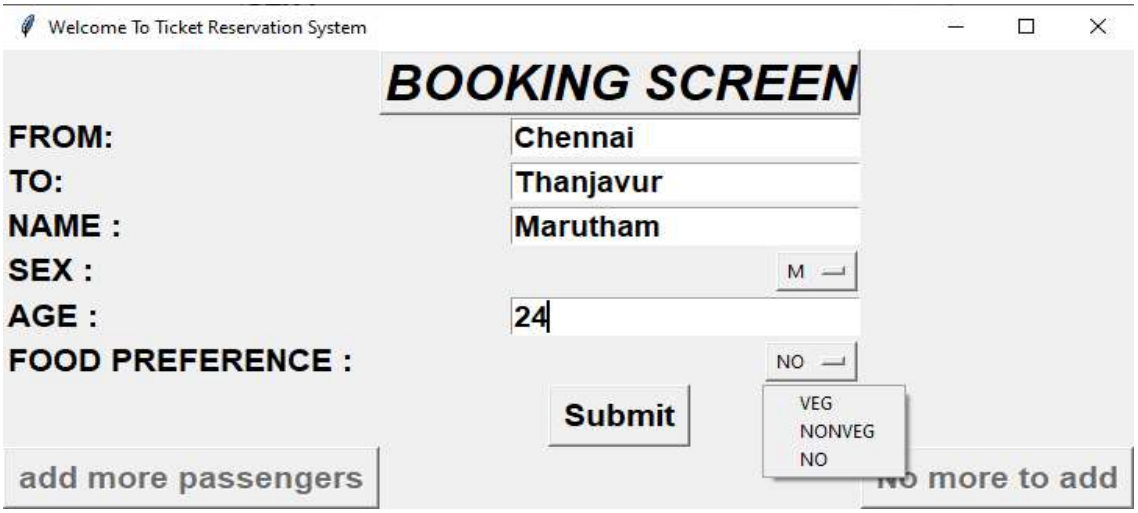
The screen which is visible on clicking 'Yes' on the previous page. The user should fill the details of the passenger



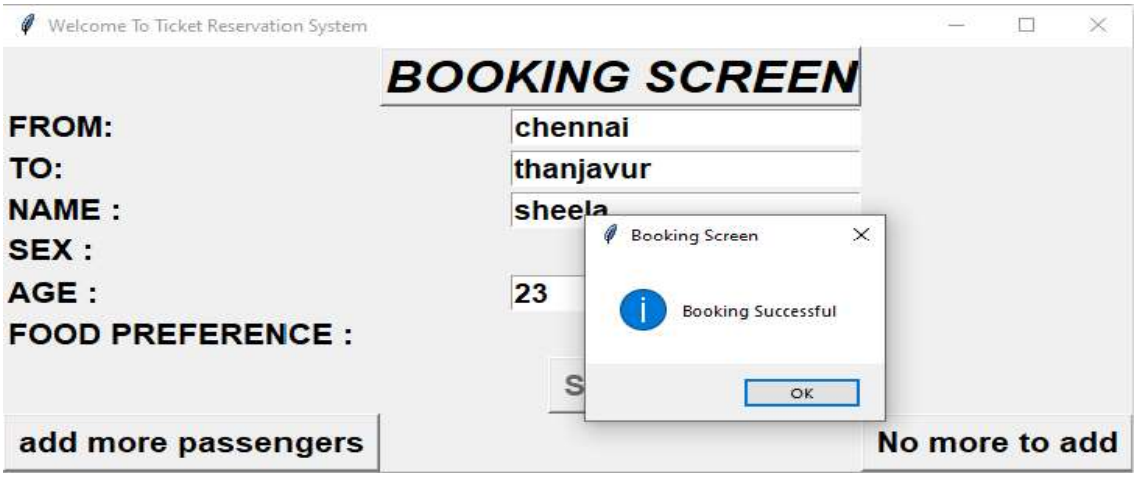
A pull-down menu is provided for the user to select the sex of the passenger



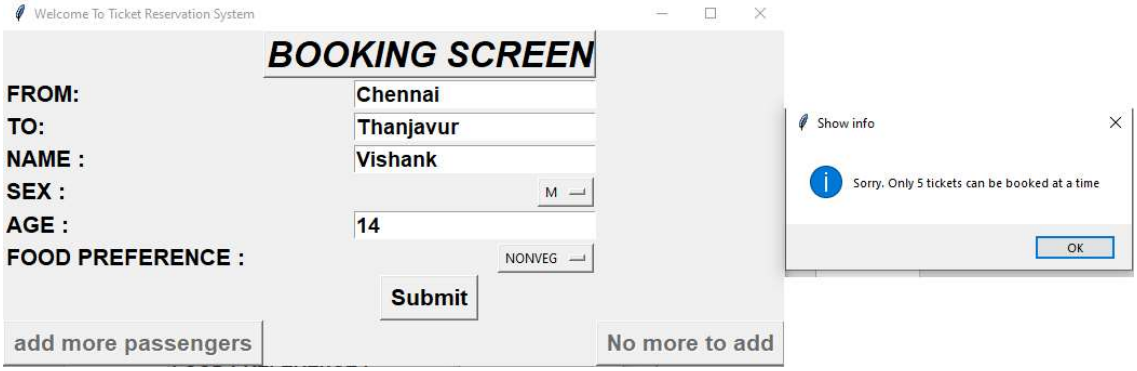
A pull-down menu is provided for the user to select the food preference of the passenger



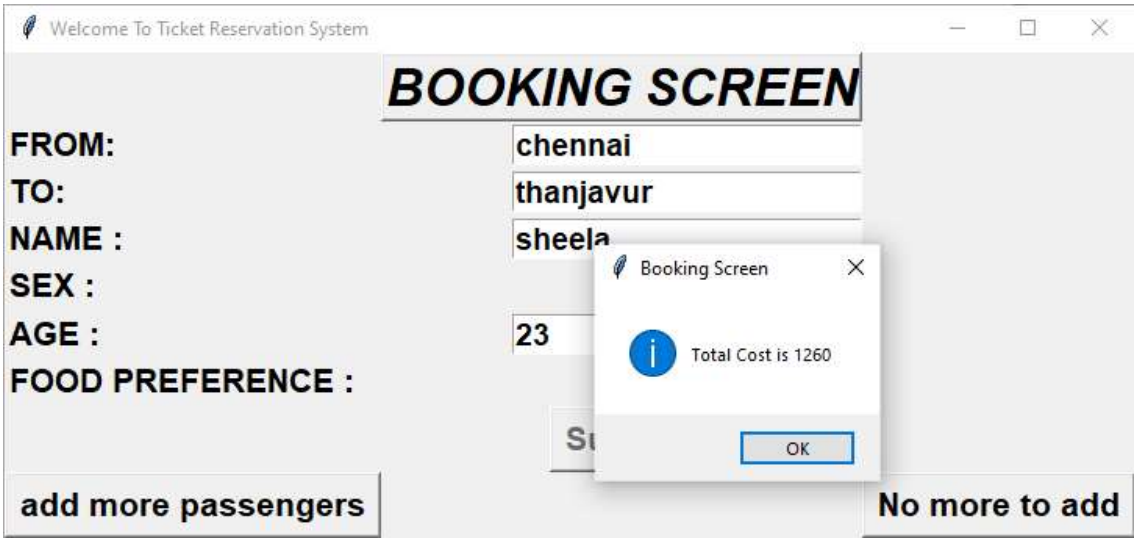
The message box acknowledging the success of booking the ticket(s)



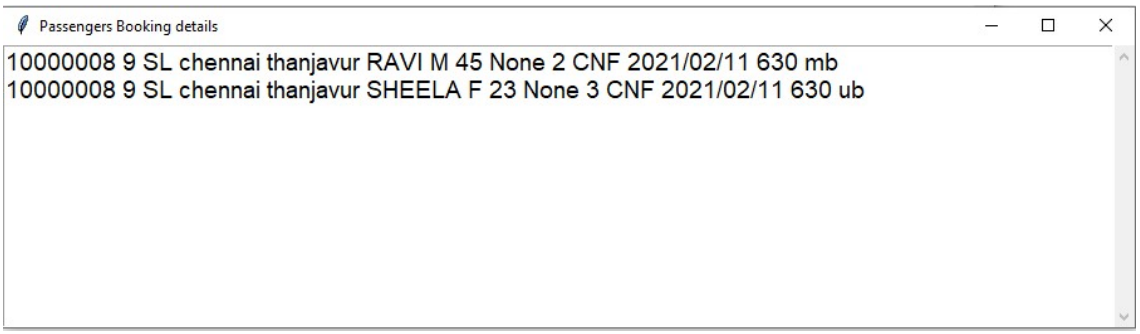
A message box is displayed when the user tries to book more than 5 tickets at a time



The total cost is printed on screen as a message box after successful booking

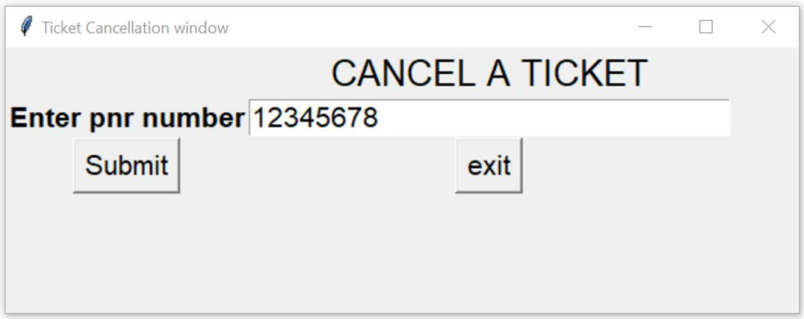


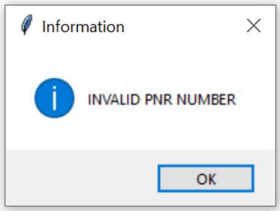
The passenger booking details are printed on screen upon successful booking



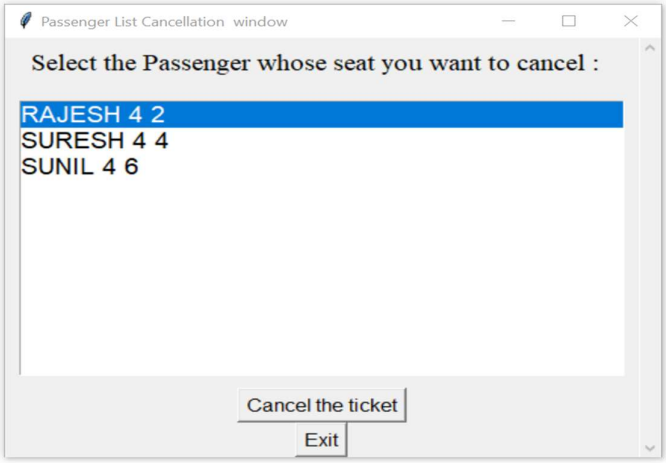
Cancellation window-

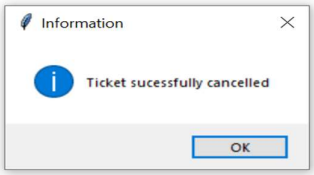
On entering invalid PNR





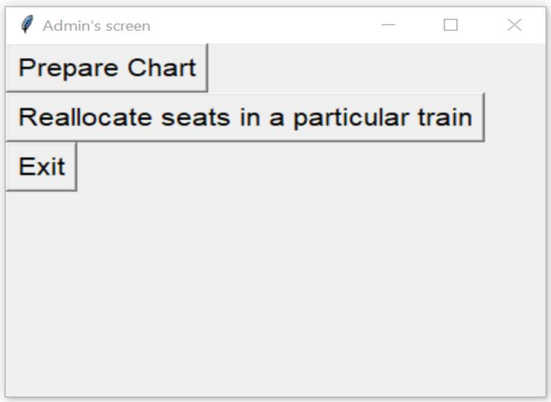
On entering valid PNR and partial cancellation of tickets

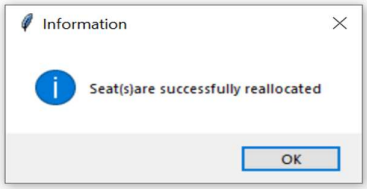




Reallocation:

On successful reallocation of seats





VARIOUS TABLES FROM SQL DATABASE

1) Users table

Structure of the table

Field	Type	Null	Key	Default	Extra
name	char(20)	YES		NULL	
userid	char(10)	YES		NULL	
password	char(8)	YES		NULL	
confirmpwd	char(8)	YES		NULL	
email	char(40)	YES		NULL	
phone	char(15)	YES		NULL	

2) Admin Table

Structure of the table

Field	Type	Null	Key	Default	Extra
name	char(20)	YES		NULL	
adminuser	char(20)	YES		NULL	
password	char(8)	YES		NULL	

3) Passengers Table

Structure of the table

Field	Type	Null	Key	Default	Extra
PNR_NO	int(11)	YES		NULL	
TRAIN_NO	int(11)	YES		NULL	
SEAT_TYPE	char(5)	YES		NULL	
START	char(20)	YES		NULL	
DESTINATION	char(20)	YES		NULL	
NAME	char(30)	YES		NULL	
SEX	char(1)	YES		NULL	
AGE	int(11)	YES		NULL	
FOOD	char(10)	YES		NULL	
SEATNO	int(11)	YES		NULL	
STATUS	varchar(20)	YES		NULL	
DATE_OF_JOURNEY	date	YES		NULL	
cost	float	YES		NULL	
BERTH	char(3)	YES		NULL	

4) Cancelled table

Structure of the table

Field	Type	Null	Key	Default	Extra
PNR_NO	int(11)	YES		NULL	
TRAIN_NO	int(11)	YES		NULL	
SEAT_TYPE	char(5)	YES		NULL	
START	char(20)	YES		NULL	
DESTINATION	char(20)	YES		NULL	
NAME	char(30)	YES		NULL	
SEX	char(1)	YES		NULL	
AGE	int(11)	YES		NULL	
FOOD	char(10)	YES		NULL	
SEATNO	int(11)	YES		NULL	
STATUS	varchar(20)	YES		NULL	
DATE_OF_JOURNEY	date	YES		NULL	
cost	float	YES		NULL	
BERTH	char(3)	YES		NULL	

5) Train Table

Structure of the table

Field	Type	Null	Key	Default	Extra
train_no	int(5)	YES		NULL	
train_name	char(50)	YES		NULL	
start	varchar(30)	YES		NULL	
destination	char(30)	YES		NULL	
seat_type	char(30)	YES		NULL	
no_of_seats	int(3)	YES		NULL	
distance	int(4)	YES		NULL	
base_fare	int(5)	YES		NULL	
departure	time	YES		NULL	
arrival	time	YES		NULL	

Contents of the table

1	andhra pradesh express	vishakhapatnam	new delhi	sl	72	1800	650	06:40:00	18:45:00
1	andhra pradesh express	vishakhapatnam	new delhi	2AC	72	1800	650	06:40:00	18:45:00
1	andhra pradesh express	vishakhapatnam	new delhi	3AC	72	1800	650	06:40:00	18:45:00
2	brindavan express	bangalore	chennai	sl	72	350	500	07:50:00	14:00:00
2	brindavan express	bangalore	chennai	2AC	72	350	500	07:50:00	14:00:00
2	brindavan express	bangalore	chennai	3AC	72	350	500	07:50:00	14:00:00
3	charminar express	chennai	hyderabad	sl	72	600	650	05:00:00	19:00:00
3	charminar express	chennai	hyderabad	2AC	72	600	650	05:00:00	19:00:00
3	charminar express	chennai	hyderabad	3AC	72	600	650	05:00:00	19:00:00
4	chennai mail	chennai	mumbai	sl	72	1350	700	04:00:00	21:00:00
4	chennai mail	chennai	mumbai	2AC	72	1350	700	04:00:00	21:00:00
4	chennai mail	chennai	mumbai	3AC	72	1350	700	04:00:00	21:00:00
5	deccan queen express	mumbai	pune	sl	72	150	500	14:00:00	17:00:00
5	deccan queen express	mumbai	pune	2AC	72	150	500	14:00:00	17:00:00
5	deccan queen express	mumbai	pune	3AC	72	150	500	14:00:00	17:00:00
6	ernad express	mangalore	nagercoil	sl	72	700	600	06:00:00	21:00:00
6	ernad express	mangalore	nagercoil	2AC	72	700	600	06:00:00	21:00:00
6	ernad express	mangalore	nagercoil	3AC	72	700	600	06:00:00	21:00:00
7	rockfort express	chennai	trichy	sl	72	300	550	07:00:00	14:00:00
7	rockfort express	chennai	trichy	2AC	72	300	550	07:00:00	14:00:00
7	rockfort express	chennai	trichy	3AC	72	300	550	07:00:00	14:00:00
8	tirumala express	vishakhapatnam	tirupati	sl	72	750	650	04:00:00	19:00:00
8	tirumala express	vishakhapatnam	tirupati	2AC	72	750	650	04:00:00	19:00:00
8	tirumala express	vishakhapatnam	tirupati	3AC	72	750	650	04:00:00	19:00:00
9	uzhavan express	chennai	thanjavur	sl	72	350	500	09:00:00	17:00:00
9	uzhavan express	chennai	thanjavur	2AC	72	350	500	09:00:00	17:00:00
9	uzhavan express	chennai	thanjavur	3AC	72	350	500	09:00:00	17:00:00
10	vaigai express	chennai	madurai	sl	72	450	550	10:00:00	18:00:00
10	vaigai express	chennai	madurai	2AC	72	450	550	10:00:00	18:00:00
10	vaigai express	chennai	madurai	3AC	72	450	550	10:00:00	18:00:00
11	shatabdi express	chennai	bangalore	sl	72	350	650	06:00:00	11:00:00
11	shatabdi express	chennai	bangalore	2AC	72	350	650	06:00:00	11:00:00
11	shatabdi express	chennai	bangalore	3AC	72	350	650	06:00:00	11:00:00
11	shatabdi express	bangalore	mysore	sl	72	150	650	11:15:00	13:15:00
11	shatabdi express	bangalore	mysore	2AC	72	150	650	11:15:00	13:15:00
11	shatabdi express	bangalore	mysore	3AC	72	150	650	11:15:00	13:15:00
11	shatabdi express	chennai	mysore	sl	72	500	650	06:00:00	13:15:00
11	shatabdi express	chennai	mysore	2AC	72	500	650	06:00:00	13:15:00
11	shatabdi express	chennai	mysore	3AC	72	500	650	06:00:00	13:15:00