



**PM
SHRI**
Creating holistic and well-rounded individuals
equipped with key 21st Century skills



COMPUTER SCIENCE PROJECT

2023-24

† **PROJECT TOPIC :**

RAILWAY RESERVATION SYSTEM

Submitted By:

Name: A.VINIT KUMAR

Class: XII-A

Board roll no:

**Submitted To: SAROJ KANTA MISRA,
PGTS(CS)**

CERTIFICATE

This is to certify that **A.VINIT KUMAR** of class: **XII A** of **KENDRIYA VIDYALAYA BERHAMPUR** has done his project on **RAILWAY RESERVATION SYSTEM** under my supervision. He has taken interest and has shown at most sincerity in completion of this project.

I certify this project up to my expectation & as per guidelines issued by **CBSE, NEW DELHI**.

INTERNAL EXAMINER

EXTERNAL EXAMINER

PRINCIPAL

ACKNOWLEDGMENT

It is with pleasure that I acknowledge my sincere gratitude to our teacher, **MR. SAROJ KANTA MISRA** who taught and undertook the responsibility of teaching the subject computer science. I have been greatly benefited from his classes.

I am especially indebted to our Principal **MR. SHIVAPRIYA DASH** who has always been a source of encouragement and support and without whose inspiration this project would not have been a successful I would like to place on record heartfelt thanks to him.

Finally, I would like to express my sincere appreciation for all the other students in my batch their friend & the fine time that we all shared together.

STUDENT'S SIGN

PRINCIPAL SIGN

HARDWARES AND SOFTWARES REQUIRED

HARDWARES

1. Desktop / Laptop
2. Mobile Phone

SOFTWARES

1. Python (latest version)
2. MySQL
3. Python Connector Module



CONTENTS

<u>S.No.</u>	<u>Topic</u>	<u>Page No.</u>
1	Certificate	2
2	Acknowledgement	3
3	Hardware and Softwares Required	4
4	Introduction	6
5	Python Source Code	10
6	MySQL Database	22

7	Outputs	25
8	References	30

INTRODUCTION

The project **RAILWAY RESERVATION SYSTEM**

Objectives of the Project

1. **Administrative Empowerment** The core objectives of our project are multi-faceted, addressing the intricate needs of both administrators and passengers within the railway ecosystem.

User Registration & Authentication

The project introduces a streamlined User Registration process, requiring users to provide essential details such as user ID, name, phone number, email ID, and password. This information is securely stored in a MySQL database. Additionally, robust User Authentication mechanisms ensure the security and integrity of user accounts.

Administrative Functions

The Admin Panel, a focal point of the system, equips administrators with indispensable tools:

Add Train: Administrators can add new trains to the system, specifying crucial details such as train number,

name, origin, destination, journey distance, journey time, seat availability, and fares.

Update Train Details: The system allows administrators to modify existing train details, adapting to dynamic demands and operational changes. This includes updating seat availability, journey details, and fare information.

Cancel Train: In response to operational constraints or changing demands, administrators have the authority to cancel trains, ensuring flexibility and adaptability in the system.

2. Enhancing Customer Experience

The Customer Panel is designed with the end-user in mind, focusing on providing an intuitive and user-friendly interface for seamless ticket booking and management.

User Interaction

User Login: The system provides a secure login interface for registered users, requiring user ID and password for access.

Forgot User ID: Users can retrieve forgotten user IDs through their registered email addresses, enhancing user experience and reducing barriers to access.

Passenger Panel Functionalities

The Passenger Panel, an extension of the Customer Panel, offers a range of functionalities:

Train Search: Users can search for trains based on origin and destination, providing comprehensive details about available trains, schedules, seat availability, and fares.

Book Tickets: The system facilitates the ticket booking process, allowing users to select trains, classes, and passengers for reservation.

Cancel Tickets: A streamlined process enables users to cancel booked tickets, providing flexibility and convenience in managing travel plans.

Key Code Functionalities Integrated

User Management Functions

New User Registration

The `new_user()` function employs a systematic approach to register new users, generating a unique user ID and storing essential details in the MySQL database.

Forgot User ID

The `forgot_user_id()` function provides a mechanism for users to retrieve their registered user IDs via email verification, ensuring a smooth user experience.

Old User Authentication

The `old_user()` function verifies user credentials, allowing access to registered users and guiding them through the system functionalities.

Admin Panel Operations

Admin Password Verification

The ``adminpassword()`` function implements a secure password system for accessing the admin panel, ensuring restricted access and system security.

Train Management Operations

Functions such as ``add_train()``, ``update_details()``, and ``cancel_train()`` constitute essential operations within the Admin Panel. These functionalities empower administrators to add new trains, update train details, and cancel trains when necessary.

Passenger Panel Operations

Train Search

The ``Train_Search()`` function allows users to search for trains based on origin and destination, providing detailed information about available trains.

Book Tickets

The ``Book_Ticket(uid)`` function simulates the ticket booking process, allowing users to select trains, classes, and passengers for reservation.

Cancel Tickets

The `Cancel_Ticket()` function facilitates the cancellation of booked tickets, showcasing the system's ability to handle modifications to user bookings.

Significance of the Project

The significance of our project extends beyond its technical complexity; it lies in its real-world applicability. In a world where efficient transportation is vital, an effective railway reservation system is crucial. The project addresses the pressing need for systems that can adapt to the evolving demands of the railway industry while prioritizing user satisfaction, security, and transparency.

PYTHON

SOURCE CODE:

1. import random
2. from tabulate import tabulate
3. import mysql.connector as con
4. dbo=con.connect(host="localhost",user="root",password="admin",database="train_reservation")
5. co=dbo.cursor()

i. #NEW USER REGISTRATION SECTION

6. def new_user():
7. dbo=con.connect(host="localhost",user="root",password="admin",database="train_reservation")
8. co=dbo.cursor()
9. import random
10. pid=random.randint(0,1000)*10
11. print("-----")
12. print("\n Welcome to our reservation system \n Register Yourself here to use our system")
13. uid=input("Enter your user id: ")
14. name=input("Enter your name: ")
15. pno=input("Enter your phone no: ")
16. eid=input("Enter your email_id: ")
17. pwd=input("Enter your password: ")
18. co.execute("insert into user values ('{}',{},{},{},{},{})".format(uid,pid,name,pno,eid,pwd))
19. print("***** Congratulations!!! Your id is successfully created *****")
20. print("-----")
21. dbo.commit()

i. #FORGET USER ID

22. def forgot_user_id():
23. dbo=con.connect(host="localhost",user="root",password="admin",database="train_reservation")
24. co=dbo.cursor()
25. email=input("Enter your registered email: ")
26. co.execute("select user_id from user where email_id like '{}".format(email))
27. emel=co.fetchall()
28. for i in emel:
29. print("Your user_id is: ",(i[0]))

i. # old user id

30. def old_user():
31. dbo=con.connect(host="localhost",user="root",password="admin",database="train_reservation")
32. co=dbo.cursor()

```

33. print("\n-----\n")
34. uid=input("Enter your user id: ")
35. co.execute("select user_id from user where user_id like '{}".format(uid))
36. b=co.fetchall()
37. c=len(b)
38. if c==0:
39. print("----- Your given id is not registered -----")
40. print("\n-----\n")
41. print("1. Try again")
42. print("2. Forgot user id")
43. print("3. Register as a new user")
44. choose=int(input("Choose an option from above: "))
45. if choose==1:
46. old_user()
47. elif choose==2:
48. forgot_user_id()
49. elif choose==3:
50. new_user()
51. else:
52. pas=input("Enter your password: ")
53. co.execute("select password from user where password like '{}".format(pas))
54. n=co.fetchall()
55. for i in n:
56. if pas==(i[0]):
    a. print("\n-----\n")
    b. print("-----Welcome back sir/ma'am what's your plan Today??-----\n")
    c. passenger_panel(uid)

```

i. #PASSWORD FUNCTION FOR ACCESS TO USER PANEL

```

57. def user_panel():
58. print(" 1. Register")
59. print(" 2. Login")
60. print(" 3. Back")
61. out=int(input("Enter your choice: "))
62. if out==1:
63. new_user()
64. elif out==2:
65. old_user()
66. elif out==3:
67. main_menu()

```

i. #PASSWORD FUNCTION FOR ACCESS TO ADMIN FUNCTION

```

68. def adminpassword():

```

```

69.password=(input("Enter your password: "))
70.if password=="CLASS12CS":
71.print("*****Access Granted*****")
72.print("-----")
73.admin_panel()
74.else:
75.print("*****ACCESS NOT GRANTED ENTER CORRECT
    PASSWORD*****")
76.print("-----")
77.adminpassword()

```

i. #ADD TRAIN

```

78.def add_train():
79.dbo=con.connect(host="localhost",user="root",password="admin",database="train_
    reservation")
80.co=dbo.cursor()
81.print("-----")
82.a=int(input("Enter train no: "))
83.b=input("Enter train name: ")
84.c=input("Enter train origin: ")
85.d=input("Enter train destination: ")
86.e=int(input("Enter train journey distance: "))
87.g=input("Enter train journey time: ")
88.h=int(input("Enter no of seats in AC: "))
89.i=int(input("Enter no of seats in SL: "))
90.j=int(input("Enter no of seats in GEN: "))
91.k=int(input("Enter price of AC: "))
92.l=int(input("Enter price of SL: "))
93.m=int(input("Enter price of GEN: "))
94.n=input("Enter days available: ")
95.print("-----")
96.co.execute("insert into train_schedule values
    ({},{},{},{},{},{},{},{},{},{},{},{},{})".format(a,b,c,d,e,g,h,i,j,k,l,m,n))
97.print("*****You have added a new train details successfully*****")
98.dbo.commit()

```

i. #UPDATE TRAIN TABLE

```

99.def update_details():
100.    dbo=con.connect(host="localhost",user="root",password="admin",database=
    "train_reservation")
101.    co=dbo.cursor()
102.    print("-----")
103.    print("*****Welcome to update train system*****")

```

```

104.     print("1. Update train no")
105.     print("2. Update train name")
106.     print("3. Update train origin")
107.     print("4. Update train destination")
108.     print("5. Update journey dist")
109.     print("6. Update available days")
110.     print("7. Update journey time")
111.     print("8. Update no of seats in AC")
112.     print("9. Update no of seats in SL")
113.     print("10. Update no of seats in GEN")
114.     print("11. Update price of AC")
115.     print("12. Update price of SL")
116.     print("13. Update price of GEN")
117.     print("14. Exit")
118.     x=int(input("Enter your choice to use: "))

119.     while True:
120.         i. # To Update train no
121.         if x==1:
122.             print("-----")
123.             print("*****YOU ARE GOING TO UPDATE TRAIN NO*****")
124.             tname=input("Enter train name whose no you want to update: ")
125.             tno=int(input("Enter updated train no: "))
126.             co.execute("update train_schedule set train_no={} where train_name='{}'".
127.                 format(tno,tname))
128.             print("*****UPDATED SUCCESSFULLY*****")
129.             print("-----")
130.             dbo.commit()
131.             return

132.         i. # To Update train name
133.         elif x==2:
134.             print("-----")
135.             print("*****YOU ARE GOING TO UPDATE TRAIN NAME*****")
136.             tno=int(input("Enter train no whose name you want to update: "))
137.             tname=input("Enter updated train name: ")
138.             co.execute("update train_schedule set train_name='{}' where train_no={}".
139.                 format(tname,tno))
140.             print("*****UPDATED SUCCESSFULLY*****")
141.             print("-----")
142.             dbo.commit()
143.             return

144.         i. # To Update train origin
145.         elif x==3:

```

```

141.     print("-----")
142.     print("*****YOU ARE GOING TO UPDATE TRAIN ORIGIN*****")
143.     tno=int(input("Enter train no whose origin you want to update: "))
144.     orgn=input("Enter updated train origin: ")
145.     co.execute("update train_schedule set origin='{}' where train_no={}".
format(orgn,tno))
146.     print("*****UPDATED SUCCESSFULLY*****")
147.     print("-----")
148.     dbo.commit()
149.     return
        i. # To Update train destination
150.     elif x==4:
151.     print("-----")
152.     print("*****YOU ARE GOING TO UPDATE TRAIN DESTINATION*****")
153.     tno=int(input("Enter train no whose destination you want to update: "))
154.     td=input("Enter updated train destination: ")
155.     co.execute("update train_schedule set destination='{}' where train_no={}".
format(td,tno))
156.     print("*****UPDATED SUCCESSFULLY*****")
157.     print("-----")
158.     dbo.commit()
159.     return

        i. # To Update journey distance
160.     elif x==5:
161.     print("-----")
162.     print("*****YOU ARE GOING TO UPDATE JOURNEY DISTANCE *****")
163.     tno=int(input("Enter train no whose journey dist you want to update: "))
164.     tjd=input("Enter updated journey dist: ")
165.     co.execute("update train_schedule set journey_distance='{}' where
train_no={}". format(tjd,tno))
166.     print("*****UPDATED SUCCESSFULLY*****")
167.     print("-----")
168.     dbo.commit()
169.     return

        i. # To Update the days train is available
170.     elif x==6:
171.     print("-----")
172.     print("*****YOU ARE GOING TO UPDATE AVAILABLE DAYS*****")
173.     tno=int(input("Enter train no whose avl_days you want to update: "))
174.     tad=input("Enter updated available days: ")
175.     co.execute("update train_schedule set avl_days ={}' where train_no={}".
format(tad,tno))
176.     print("*****UPDATED SUCCESSFULLY*****")
177.     print("-----")
178.     dbo.commit()

```



```

179.     return

        i. # To Update journey time
180.     elif x==7:
181.         print("-----")
182.         print("*****YOU ARE GOING TO UPDATE JOURNEY TIME*****")
183.         tno=int(input("Enter train no whose journey_time you want to update: "))
184.         tj=input("Enter updated journey time: ")
185.         co.execute("update train_schedule set total_time='{}' where train_no={}".
            format(tj,tno))
186.         print("*****UPDATED SUCCESSFULLY*****")
187.         print("-----")
188.         dbo.commit()
189.         return

        i. # To Update no of seats in ac coach of that train
190.     elif x==8:
191.         print("-----")
192.         print("*****YOU ARE GOING TO UPDATE SEATS IN AC*****")
193.         tno=int(input("Enter train no whose no of seats in AC you want to update: "))
194.         tsa=input("Enter updated no of seats in AC: ")
195.         co.execute("update train_schedule set ac1='{}' where train_no={}".
            format(tsa,tno))
196.         print("*****UPDATED SUCCESSFULLY*****")
197.         print("-----")
198.         dbo.commit()
199.         return

        i. # To Update no of seats in SL coach of that train
200.     elif x==9:
201.         print("-----")
202.         print("*****YOU ARE GOING TO UPDATE SEATS IN SL*****")
203.         tno=int(input("Enter train no whose no of seats in SI you want to update: "))
204.         tss=input("Enter updated no of seats in SI: ")
205.         co.execute("update train_schedule set sl='{}' where train_no={}".
            format(tss,tno))
206.         print("*****UPDATED SUCCESSFULLY*****")
207.         print("-----")
208.         dbo.commit()
209.         return

        i. # To Update no of seats in GEN coach of that train
210.     elif x==10:
211.         print("-----")
212.         print("*****YOU ARE GOING TO UPDATE SEATS IN GEN*****")
213.         tno=int(input("Enter train no whose no of seats in GEN you want to update: "))

```

```

214.     tsg=input("Enter updated no of seats in GEN: ")
215.     co.execute("update train_schedule set gen='{}' where train_no={}".
format(tsg,tno))
216.     print("*****UPDATED SUCCESSFULLY*****")
217.     print("-----")
218.     dbo.commit()
219.     return

        i. # To Update fare price of AC of that train
220.     elif x==11:
221.     print("-----")
222.     print("*****YOU ARE GOING TO UPDATE FARE PRICE OF AC*****")
223.     tno=int(input("Enter train no whose fare price of ac you want to update: "))
224.     tfa=input("Enter updated fare price of ac: ")
225.     co.execute("update train_schedule set ac_fare={} where train_no={}".
format(tfa,tno))
226.     print("*****UPDATED SUCCESSFULLY*****")
227.     print("-----")
228.     dbo.commit()
229.     return

        i. # To Update fare price of SL of that train
230.     elif x==12:
231.     print("-----")
232.     print("*****YOU ARE GOING TO UPDATE FARE PRICE OF SL*****")
233.     tno=int(input("Enter train no whose fare price of SL you want to update: "))
234.     tfs=input("Enter updated fare price of SL: ")
235.     co.execute("update train_schedule set sl_fare={} where train_no={}".
format(tfs,tno))
236.     print("*****UPDATED SUCCESSFULLY*****")
237.     print("-----")
238.     dbo.commit()
239.     return

        i. # To Update fare price of GEN of that train

240.     elif x==13:
241.     print("-----")
242.     print("*****YOU ARE GOING TO UPDATE FARE PRICE OF GEN*****")
243.     tno=int(input("Enter train no whose fare price of GEN you want to update: "))
244.     tfg=input("Enter updated fare price of GEN: ")
245.     co.execute("update train_schedule set gen_fare={} where train_no={}".
format(tfg,tno))
246.     print("*****UPDATED SUCCESSFULLY*****")
247.     print("-----")
248.     dbo.commit()
249.     return

```

```

250.     elif x==14:
251.     print("*****YOU ARE NOW OUT OF UPDATE DETAILS SECTION*****")
252.     break
253.     return

```

i. #CANCEL TRAIN

```

254.     def cancel_train():
255.     dbo=con.connect(host="localhost",user="root",password="admin",database=
"train_reservation")
256.     co=dbo.cursor()
257.     ct=int(input("enter train no which you want to cancel: "))
258.     co.execute("delete from train_schedule where train_no={}".format(ct))
259.     dbo.commit()
260.     print("***** Train cancelled Successfully *****")

```

i. #ADMIN PANEL OPTIONS

```

261.     def admin_panel():
262.     while True:
263.     try:
264.     print("-----")
265.     print("*****Welcome to admin panel*****")
266.     print("1. Add train")
267.     print("2. Update details")
268.     print("3. Cancel Train")
269.     print("4. Log out")
270.     opt=int(input("Choose your option: "))
271.     if opt==1:
a.     add_train()
272.     elif opt==2:
a.     update_details()
273.     elif opt==3:
a.     cancel_train()
274.     elif opt==4:
a.     print("*****You are out of admin panel*****")
b.     print("-----")
c.     main_menu()
275.     except InvalidOptionError as e:
276.     print(f"Error: {e}")
277.     except:
278.     print("*****Choose a correct option*****")
279.     print("-----")

```

i. #PASSENGER PANEL FUNCTIONALITIES

```
280. def Train_Search():
281.     dbo=con.connect(host="localhost",user="root",password="admin",database=
        "train_reservation")
282.     co=dbo.cursor()
283.     o=input("Enter your origin: ")
284.     d=input("Enter your destination: ")
285.     co.execute("select * from train_schedule where origin like '%{}%' and
        destination like '%{}%'".format(o,d))
286.     a=co.fetchall()
287.     for i in a:
288.         print("Train no.: ",a[0][0])
289.         print("Train name: ",a[0][1])
290.         print("Origin: ",a[0][2])
291.         print("Destination: ",a[0][3])
292.         print("Journey distance: ",a[0][4])
293.         print("Available days: ",a[0][12])
294.         print("total time: ",a[0][5])
295.         print("Seats in ac1: ",a[0][6])
296.         print("Seats in sl: ",a[0][7])
297.         print("Seats in GEN: ",a[0][8])
298.         print("Fare of ac1: ",a[0][9])
299.         print("Fare of sl: ",a[0][10])
300.         print("Fare of gen: ",a[0][11])

301.     dbo.commit()
```

i. #BOOK TICKETS

```
302. def Book_Ticket(uid):
303.     dbo=con.connect(host="localhost",user="root",password="admin",database=
        "train_reservation")
304.     co=dbo.cursor()
305.     print("THIS IS OUR ALL TRAINS AVAILABLE \n -----
        -----")
306.     op="select train_no,train_name,origin,destination from train_schedule"
307.     co.execute(op)
308.     r=co.fetchall()
309.     columns=[i[0] for i in co.description]
310.     print(tabulate(r,headers=columns,tablefmt="fancy_grid"))
311.     print("-----\n\n")
312.     trainno=int(input("Enter your Train no: "))
313.     tcktno=int(input("Enter no of seats you want to book: "))
314.     print("1. AC TICKET")
315.     print("2. SL TICKET")
316.     print("3. GEN TICKET")
317.     typ=int(input("Enter your choice of class: "))
318.     if typ==1:
```

```

319.     a=co.execute("select ac1_fare from train_schedule where
        train_no={}".format(trainno))
320.     b=co.fetchall()
321.     print(b[0][0])
322.     elif typ==2:
323.     a=co.execute("select sl_fare from train_schedule where
        train_no={}".format(trainno))
324.     b=co.fetchall()
325.     print(b[0][0])
326.     elif typ==3:
327.     a=co.execute("select gen_fare from train_schedule where
        train_no={}".format(trainno))
328.     b=co.fetchall()
329.     print(b[0][0])
330.     for i in range (tcktno):
331.     cus1=input("Enter customer name: ")
332.     age=int(input("Enter your age: "))
333.     print("----- For gender M=Male, F=Female, O=Other -----")
334.     gender=input("Enter your gender: ")
335.     j=random.randint(100000,999999)
336.     print("Your PNR no is: ",j)
337.     cnf="Confirmed"
338.     p=b[0][0]
339.     co.execute("insert into booked_tickets values
        ('{}',{},{},'{}',{},{},'{}',{})".format(uid,j,trainno,cus1,age,gender,p,cnf))
340.     p=b[0][0]
341.     amt=tcktno*p
342.     print("Your total ticket price is: ",amt)
343.     dbo.commit()

344.     def Cancel_Ticket():
345.     dbo=con.connect(host="localhost",user="root",password="admin",database=
        "train_reservation")
346.     co=dbo.cursor()
347.     pnr=int(input("enter pnr no. you want to cancel: "))
348.     co.execute("delete from booked_tickets where pnr_no={}".format(pnr))
349.     dbo.commit()
350.     print("***** Your ticket cancelled Successfully *****")

        i. #PASSENGER PANEL
351.     def passenger_panel(uid):
352.     dbo=con.connect(host="localhost",user="root",password="admin",database=
        "train_reservation")
353.     co=dbo.cursor()
354.     while True:
355.     print("-----")

```

```

356.     print("*****Welcome to passenger panel*****")
357.     print("1. Train Search")
358.     print("2. Book Ticket")
359.     print("3. Cancel Ticket")
360.     print("4. Log out")
361.     choic=int(input("Enter your choice to use: "))
362.     if choic==1:
363.         Train_Search()
364.     elif choic==2:
365.         Book_Ticket(uid)
366.     elif choic==3:
367.         Cancel_Ticket()
368.     elif choic==4:
369.         main_menu()
370.     print("*****You are successfully logged out of Passenger panel*****")
371.     print("-----")

```

i. #Panel functions

```

372.     def main_menu():
373.         print("-----")
374.         print("*****WELCOME TO TRAIN RESERVATION SYSTEM*****")
375.         print("1. Admin panel")
376.         print("2. Passenger panel")
377.         print("3. Exit")
378.         ch=int(input("Enter your choice: "))
379.         while True:
380.             if ch==1:
381.                 adminpassword()

382.             elif ch==2:
383.                 user_panel()

384.             elif ch==3:
385.                 print("***** Thank You for using reservation system*****")
386.                 print("-----")
387.                 break

388.         main_menu()

```

MYSQL DATABASE:

TRAIN TABLE:

```
mysql> select * from train_schedule;
```

train_no	train_name	origin	destination	journey_distance	total_time	ac1	sl	gen	ac1_fare	sl_fare	gen_fare	avl_days
11020	KONARK EXPRESS	Bhubaneswar	CSMT	1930	36HR 40MIN	120	250	100	2400	950	200	S M T W T H F S
11209	HIRAKUD EXP	BALASORE	BHADRAK	200	5HR 55MIN	85	65	90	1900	500	200	S M T W T H F S
12321	MUMBAI MAIL	HOWRAH JN	CSMT	2159	37HR 40MIN	128	150	50	2850	850	350	S M T W T H F S
12457	HOWRAH EXP	CSMT	DELHI	4500	55HR 54MIN	58	56	54	2600	300	365	S M T H W S
12472	SWARAJ EXPRESS	JAMMU	BANDRA TERMINUS	2028	30HR 18MIN	256	182	90	2650	650	450	S M T W T H F S
12809	HOWRAH EXPRESS	CSMT	HOWRAH	1968	32HR 55MIN	90	350	80	2300	800	300	S M T W T H F S
12841	COROMONDAL EXPRESS	SHALIMAR	CHENNAI	1659	25HR 35MIN	125	100	90	2260	760	360	S M T W T H F S
20807	HIRAKUD EXPRESS	VISAKHAPATNAM	AMRITSAR	2570	38HR 45MIN	156	220	68	3400	700	100	S M T W T H F S
20861	PURI ADI SF	PURI	SURAT	1977	31HR 05MIN	128	240	200	2900	900	250	S M T W T H F S
123432	PRASANTI EXPRESS	Puri	Chennai	1980	40HR 30 MIN	123	234	345	3000	2000	1000	S M T W T H F S

```
10 rows in set (0.02 sec)
```


USER DETAILS :

user_id	pass_id	pass_name	ph_no	email_id	password
asish	7830	Asish Kumar	814427568	asish@gmail.com	asish@123
john@doe	6990	John Doe	5551234	john.doe@gmail.com	john@doe
kalia	8810	chintu	2514584	roshni133@gmail.com	789
LP	8230	lokashree	789956784	lokashree18@gmail.com	panipuri
mohit	8910	n.mohit	36533	mohit@123	mohit123
om	6600	Omm panda	78942665	omm@gmail.com	omm@panda
Sheetal	2420	Sheetal sahu	456321456	sheetal19@gmail.com	sheetal1906
swati	7510	ssp	451024526	swati09@gmail.com	topper
vinit	3000	avk18	68546894	avk18@gmail.com	chal

BOOKED_TICKETS:

user_id	pnr_no	train_no	customer_name	age	gender	price	ticket_status
asish	794470	11020	hgdfhgt	46	m	200	Confirmed
vinit	926193	12321	LIZA	22	F	850	Confirmed
vinit	312519	12321	PRAMILA	48	F	850	Confirmed
vinit	203328	12321	MIRA	45	F	850	Confirmed
swati	964758	20861	swatimita pradhan	17	f	900	Confirmed
vinit	662927	11020	sameer	17	m	2400	Confirmed
vinit	161090	11020	basundhara	16	f	2400	Confirmed
Sheetal	141050	12472	Sheetal Sahu	18	F	2650	Confirmed

OUTPUTS:

TRAIN RESERVATION MAIN PAGE:

*****WELCOME TO TRAIN RESERVATION SYSTEM*****

1. Admin panel
2. Passenger panel
3. Exit

Enter your choice: 1

Enter your password: CLASS12CS

*****Access Granted*****

*****Welcome to admin panel*****

1. Add train
2. Update details
3. Cancel Train
4. Log out

Choose your option: 1

Enter train no: 12347

Enter train name: RAJDHANI EXP

Enter train origin: DELHI

Enter train destination: JAMMU KASHMIR

Enter train journey distance: 1200

Enter train journey time: 9 HR 30MIN

Enter no of seats in AC: 290

Enter no of seats in SL: 200

Enter no of seats in GEN: 300

Enter price of AC: 3900

Enter price of SL: 900

Enter price of GEN: 500

Enter days available: S M T W T H F S

*****You have added a new train details successfully*****

THE ADMIN LOGIN PANEL:

*****Welcome to admin panel*****

1. Add train
2. Update details
3. Cancel Train
4. Log out

Choose your option: 2

THIS IS OUR ALL TRAINS AVAILABLE

train_no	train_name	origin	destination
11020	KONARK EXPRESS	Bhubaneswar	CSMT
11209	HIRAKUD EXP	BALASORE	BHADRAK
12321	MUMBAI MAIL	HOWRAH JN	CSMT
12347	RAJDHANI EXP	DELHI	JAMMU KASHMIR
12457	HOWRAH EXP	CSMT	DELHI
12472	SWARAJ EXPRESS	JAMMU	BANDRA TERMINUS
12809	HOWRAH EXPRESS	CSMT	HOWRAH
12841	COROMONDAL EXPRESS	SHALIMAR	CHENNAI
20807	HIRAKUD EXPRESS	VISAKHAPATNAM	AMRITSAR
20861	PURI ADI SF	PURI	SURAT
123432	PRASANTI EXPRESS	Puri	Chennai

CANCEL TICKET:

*****Welcome to passenger panel*****

1. Train Search
2. Book Ticket
3. Cancel Ticket
4. Log out

Enter your choice to use: 3

enter pnr no. you want to cancel: 545868

***** Your ticket cancelled Successfully *****

BOOKING TICKETS :

Enter your Train no: 12321

Enter no of seats you want to book: 2

1. AC TICKET
2. SL TICKET
3. GEN TICKET

Enter your choice of class: 1

2850

Enter customer name: snehashish

Enter your age: 18

----- For gender M=Male, F=Female, O=Other -----

Enter your gender: m

Your PNR no is: 216440

Enter customer name: bansika

Enter your age: 17

----- For gender M=Male, F=Female, O=Other -----

Enter your gender: f

Your PNR no is: 545868

Your total ticket price is: 5700

SEARCH TRAIN:

Enter your origin: csmt
Enter your destination: delhi
Train no.: 12457
Train name: HOWRAH EXP
Origin: CSMT
Destination: DELHI
Journey distance: 4500
Available days: S M TH W S
total time: 55HR 54MIN
Seats in ac1: 58
Seats in sl: 56
Seats in GEN: 54
Fare of ac1: 2600
Fare of sl: 300
Fare of gen: 365

REFERENCES:

**1. CLASS 11TH & 12TH COMPUTER SCIENCE BOOK
(SUMITA ARORA)**

2. PYTHON

[HTTPS://WWW.PYTHON.ORG/](https://www.python.org/)

3. MYSQL

[HTTPS://WWW.MYSQL.COM/](https://www.mysql.com/)

4. KV CODERS

[HTTPS://KVCODERS.IN](https://kvcoders.in)

79874: 5. IRCTC:

**[HTTPS://WWW.IRCTC.CO.IN/NGET/
TRAIN-SEARCH](https://www.irctc.co.in/nget/train-search)**