JAVA PROJECT

HOTEL MANAGEMENT SYSTEM

SUBMITTED TO :

PUNEET KUMAR

SUBMITTED BY

Name	Reg. No	Roll No
Karna Kumar	12110436	RK21PBB48
Md Sahil Imam	12110431	RK21PBB47
Saksham Gupta	12110629	RK21PBB49

Lovely Professional University

Phagwara, Punjab

Introduction :

This is the final report document for developed hotel management system. It consists of the milestones in development of finalized hotel management system. As previously mentioned current manual system used by hotel, caused for decrement in growth of success and efficiency of the hotel, caused for decrement in growth of success and efficiency of the hotel. Iterative waterfall method was used as the software development life cycle. Coding was handled through an Object-oriented approach. Above mentioned methodologies made project work load light and provided the ease of developing. The system was evaluated by several people regarding user levels of the developed system. Results of the evaluation helped for further maintenance of the product. Fully functional Hotel Management System will fulfil the main objectives and all the events of the hotel.

Purpose :

The software Requirements specification (SRS) will provide a detailed description of the requirements for the Hotel Management System (HMS). This srs will allow for a complete understanding of what is to be expected from the newly introduced system which is to be constructed. The clear understanding of the system and it's functionality will allow for the correct software to be developed for the end user and will be used for the development of the future stages of the project.

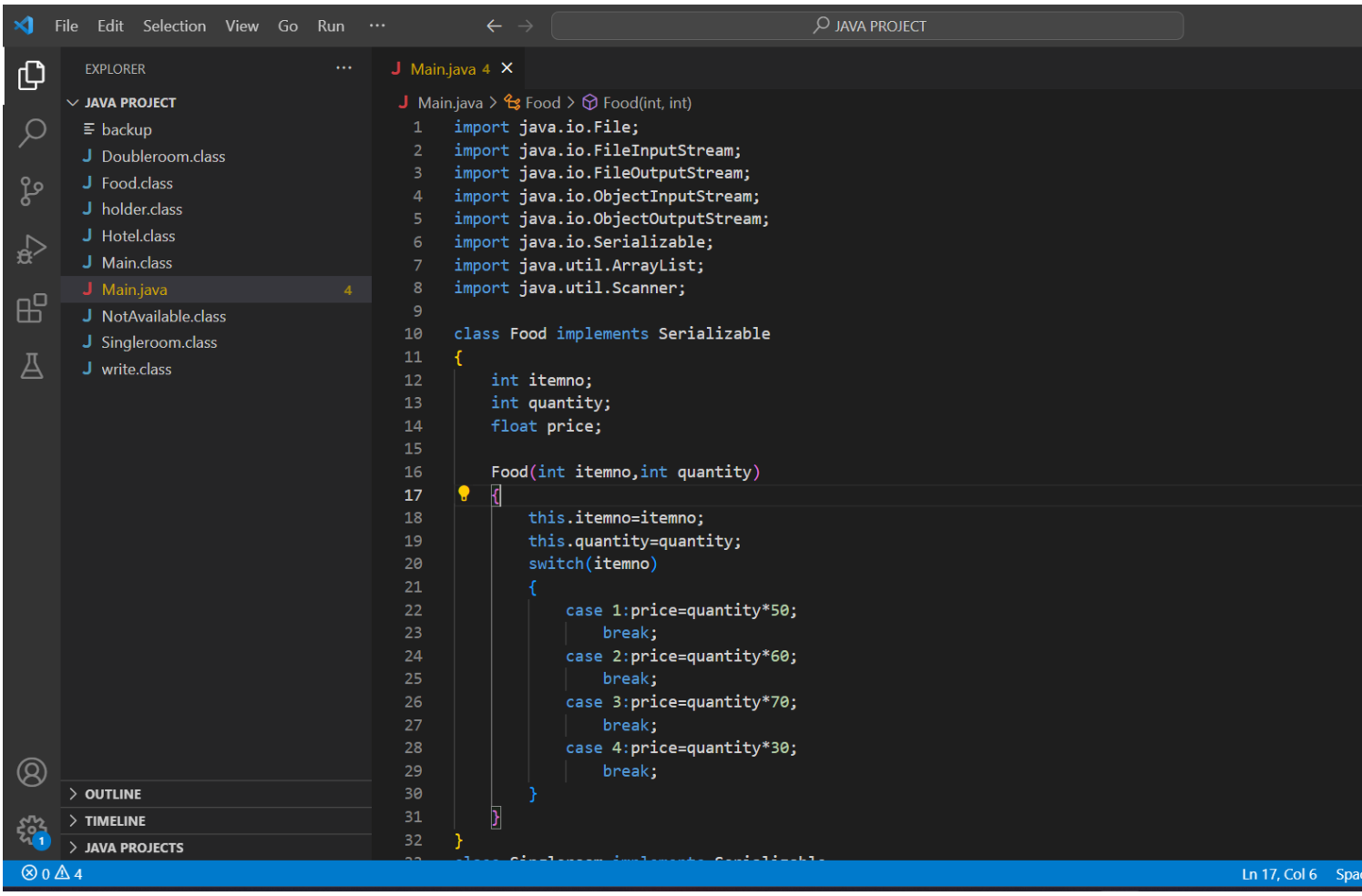
Scope of Project :

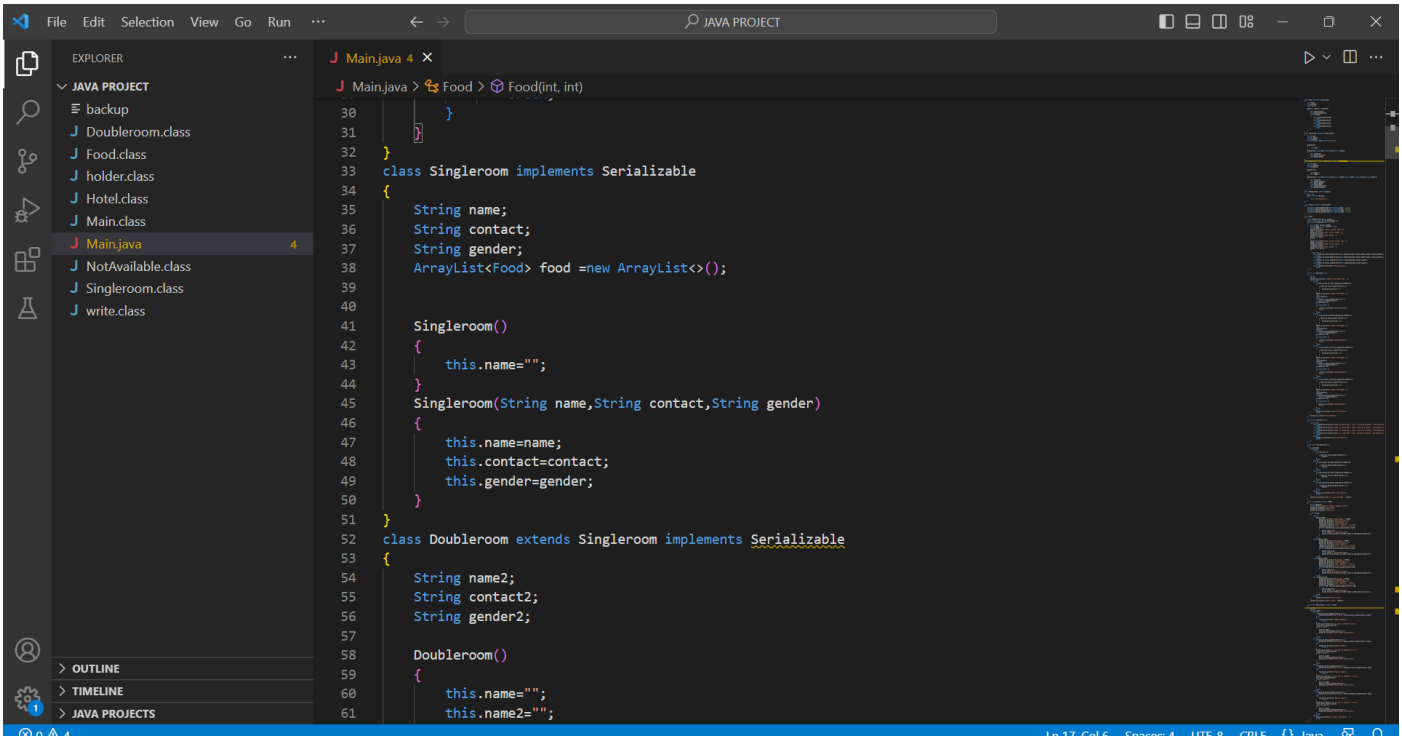
The introducing software, Hotel Management System which is going to be implemented for Hotel Dayal will automate the major operations of the hotel. The Reservation System is to keep track in room and hall reservation and check availability. The Room Management System is for manage all room types room services. The Inventory Control System will keep track in all inventories of the hotel and guest details will handled by guest management. Administration department will monitor the all. There is three End Users for HMS. The End Users Are Owner, Manager and Receptionist. Owner can access to all system functionalities without any restrictions. Manager can access to all system functionalities with limited restrictions. Receptionist can only access to the

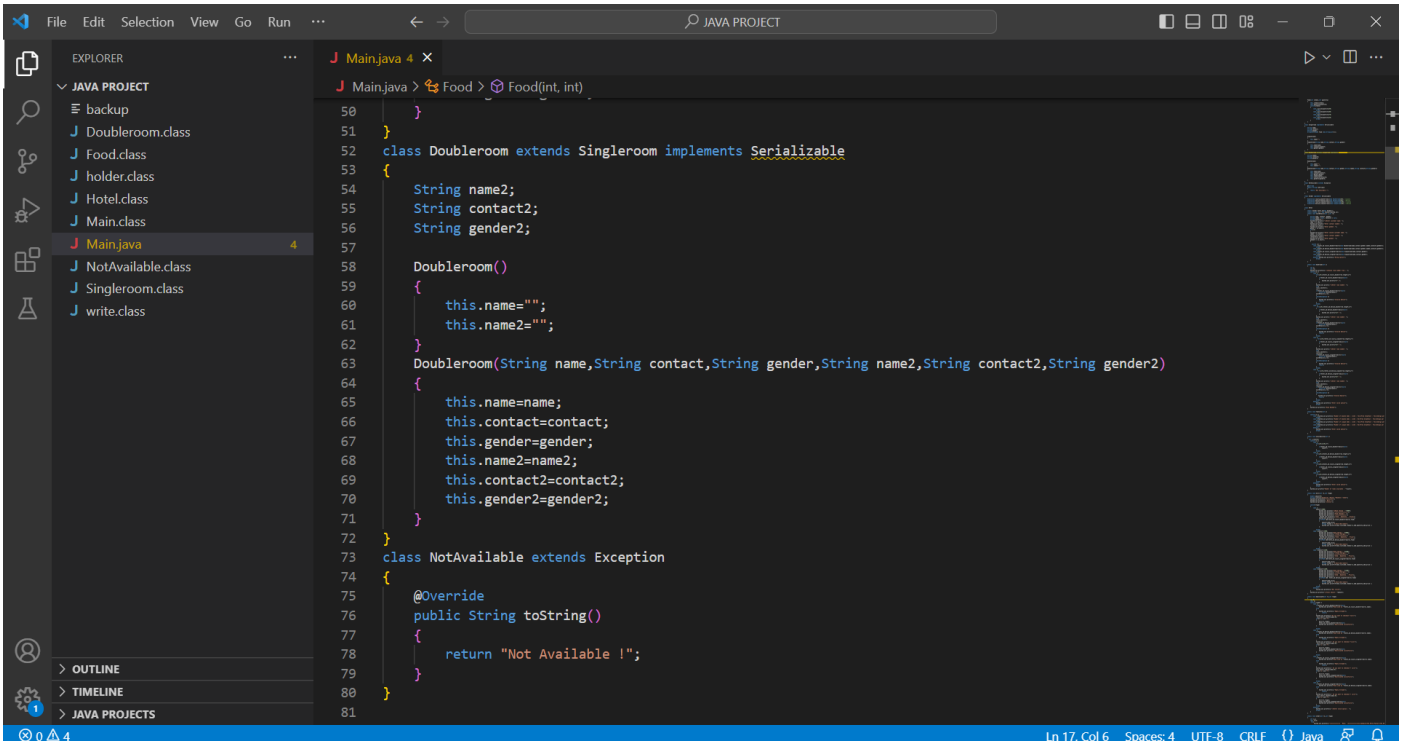
Reservation management section. To keep restrictions for each End User levels HMS can create different Login functions. The objectives of the automated Hotel Management System is to simplify the day to day processes of the hotel. The system will be able to handle many services to take care of all customers in a quick manner. As a solution to the large amount of file handling happening at the hotel,

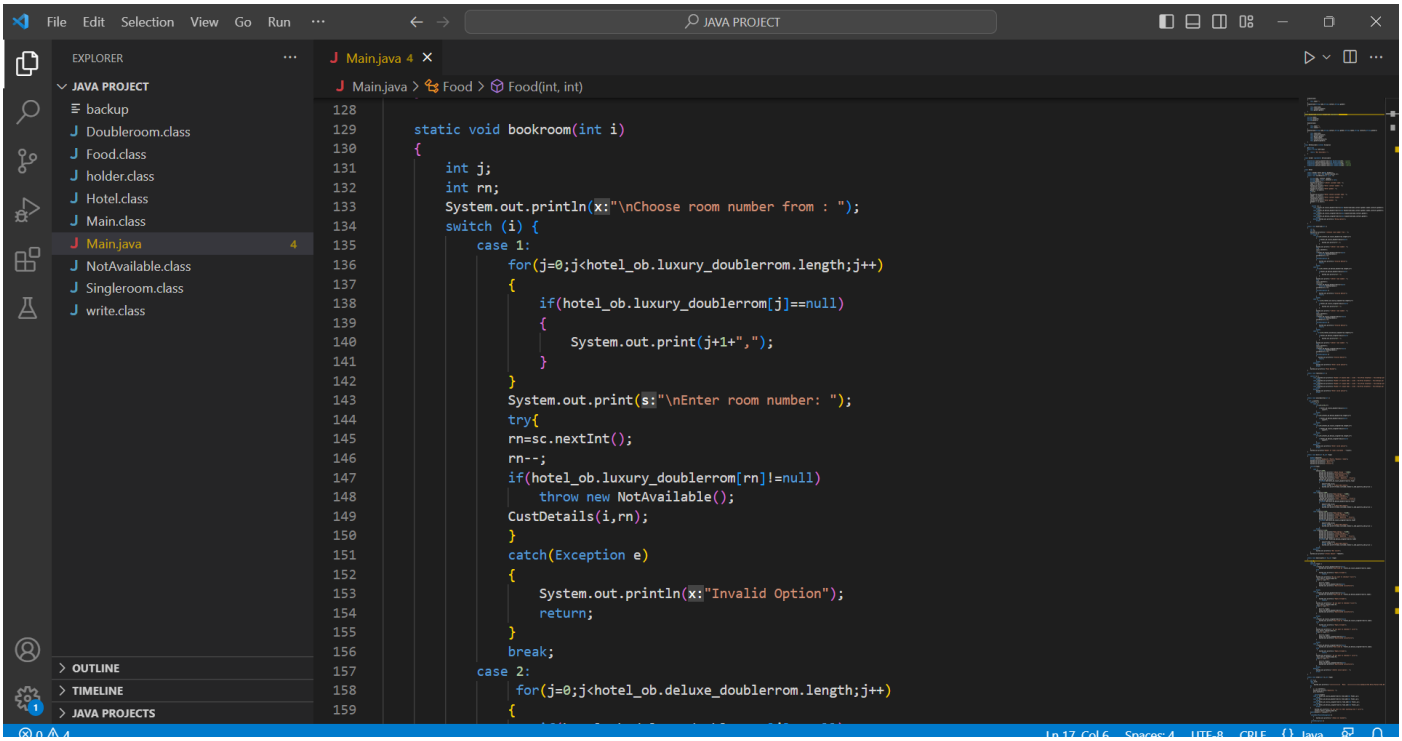
this software will be used to overcome those drawbacks. Safety, easiness of using and most importantly the efficiency of information retrieval are some benefits the development team going to present with this system. The system should be user appropriate, easy to use, provide easy recovery of errors and have an overall end user high subjective satisfaction.

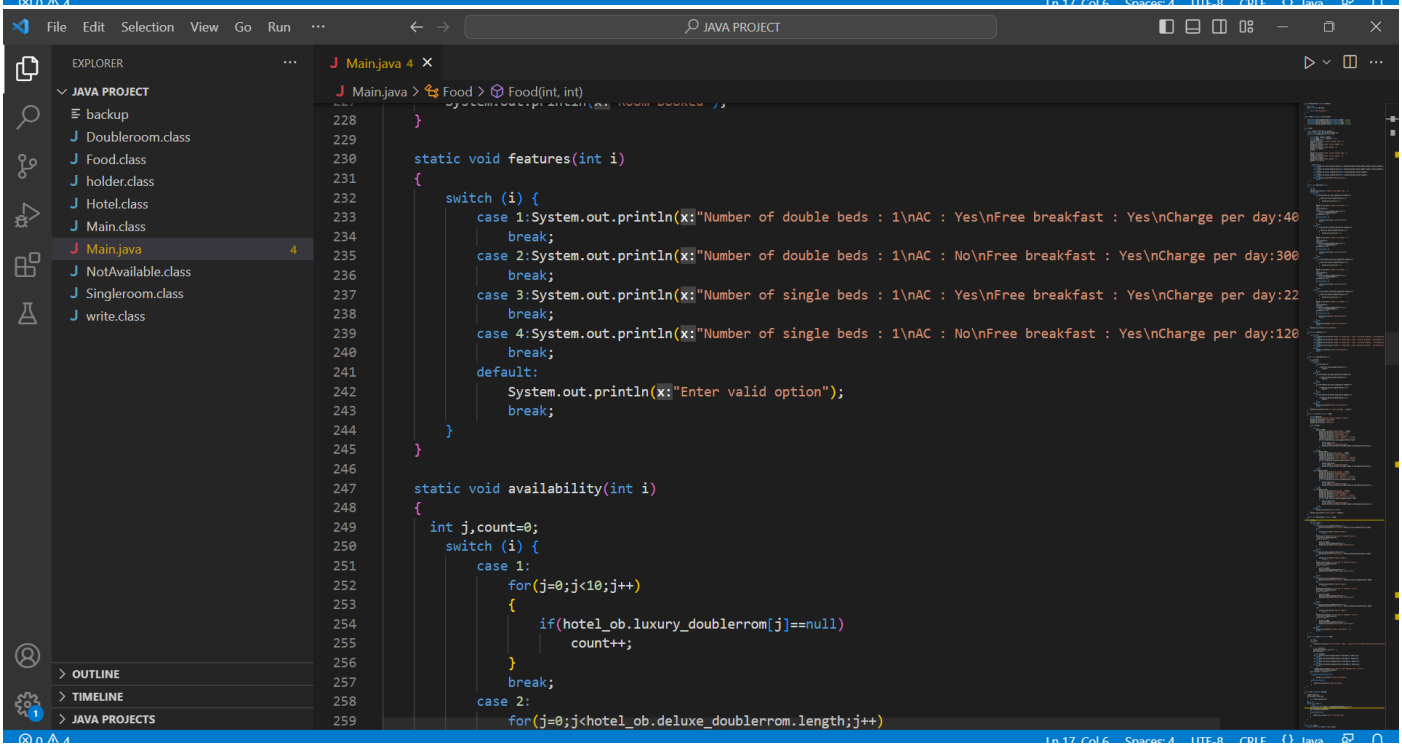
Screenshot of Code :

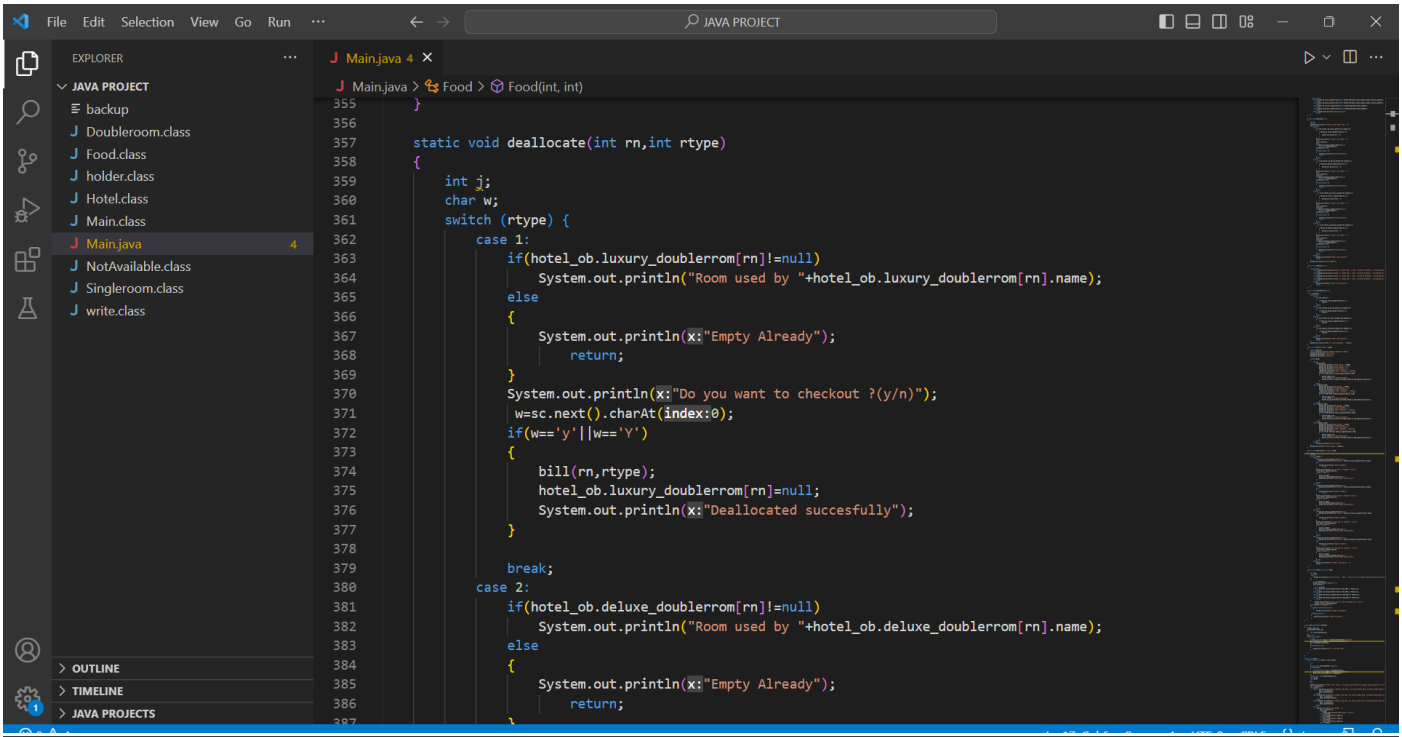


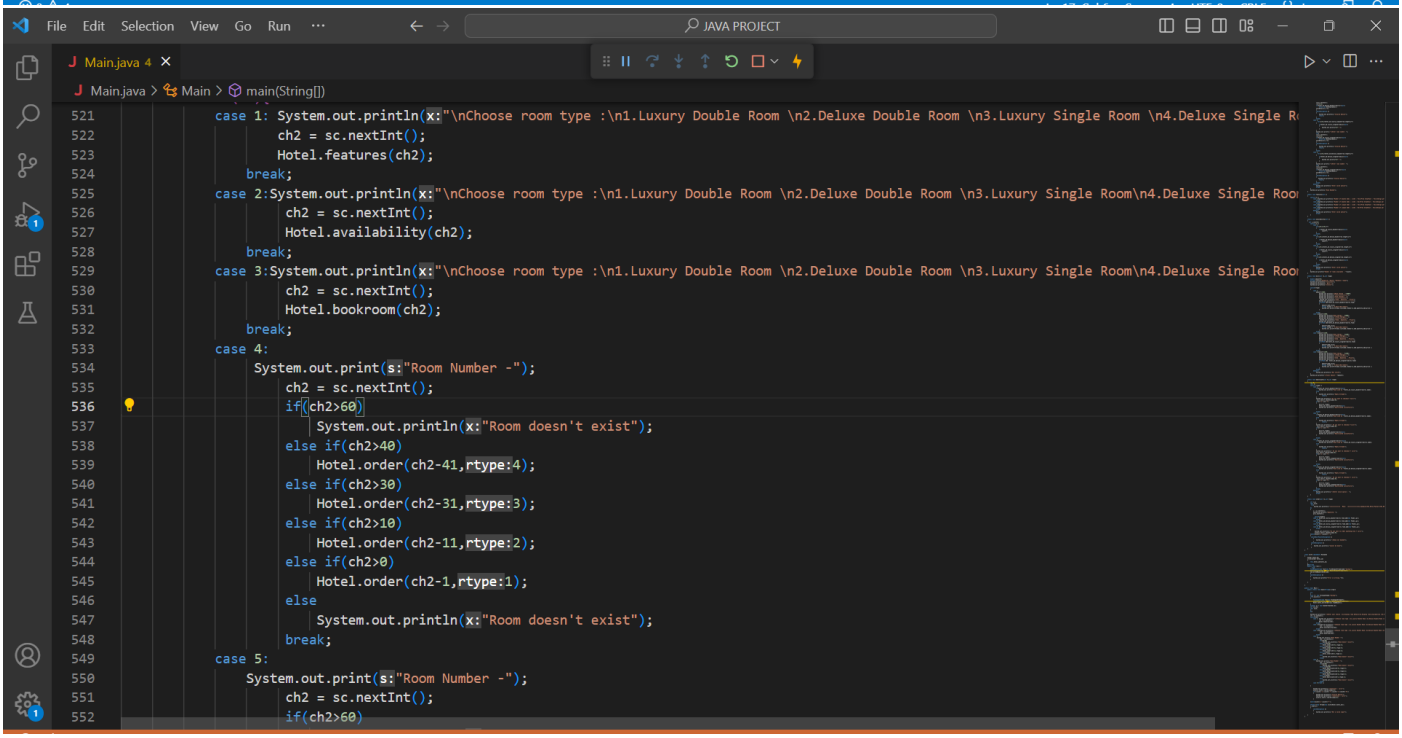

File Edit Selection View Go Run
JAVA PROJECT
EXPLORER
JAVA PROJECT
backup
Doubleroom.class
Food.class
holder.class
Hotel.class
Main.class
Main.java 4
NotAvailable.class
Singleroom.class
write.class
OUTLINE
TIMELINE
JAVA PROJECTS
Main.java 4
Main.java > Food > Food(int, int)
30 }
31 }
32 }
33 class Singleroom implements Serializable
34 {
35 String name;
36 String contact;
37 String gender;
38 ArrayList<Food> food =new ArrayList<>();
39
40
41 Singleroom()
42 {
43 this.name="";
44 }
45 Singleroom(String name,String contact,String gender)
46 {
47 this.name=name;
48 this.contact=contact;
49 this.gender=gender;
50 }
51 }
52 class Doubleroom extends Singleroom implements Serializable
53 {
54 String name2;
55 String contact2;
56 String gender2;
57
58 Doubleroom()
59 {
60 this.name="";
61 this.name2="";


File Edit Selection View Go Run
JAVA PROJECT
EXPLORER
JAVA PROJECT
backup
Doubleroom.class
Food.class
holder.class
Hotel.class
Main.class
Main.java 4
NotAvailable.class
Singleroom.class
write.class
OUTLINE
TIMELINE
JAVA PROJECTS
Main.java 4
Main.java > Food > Food(int, int)
50 }
51 }
52 class Doubleroom extends Singleroom implements Serializable
53 {
54 String name2;
55 String contact2;
56 String gender2;
57
58 Doubleroom()
59 {
60 this.name="";
61 this.name2="";
62 }
63 Doubleroom(String name,String contact,String gender,String name2,String contact2,String gender2)
64 {
65 this.name=name;
66 this.contact=contact;
67 this.gender=gender;
68 this.name2=name2;
69 this.contact2=contact2;
70 this.gender2=gender2;
71 }
72 }
73 class NotAvailable extends Exception
74 {
75 @Override
76 public String toString()
77 {
78 return "Not Available !";
79 }
80 }
81
Ln 17, Col 6 Spaces: 4 UTF-8 CRLF Java

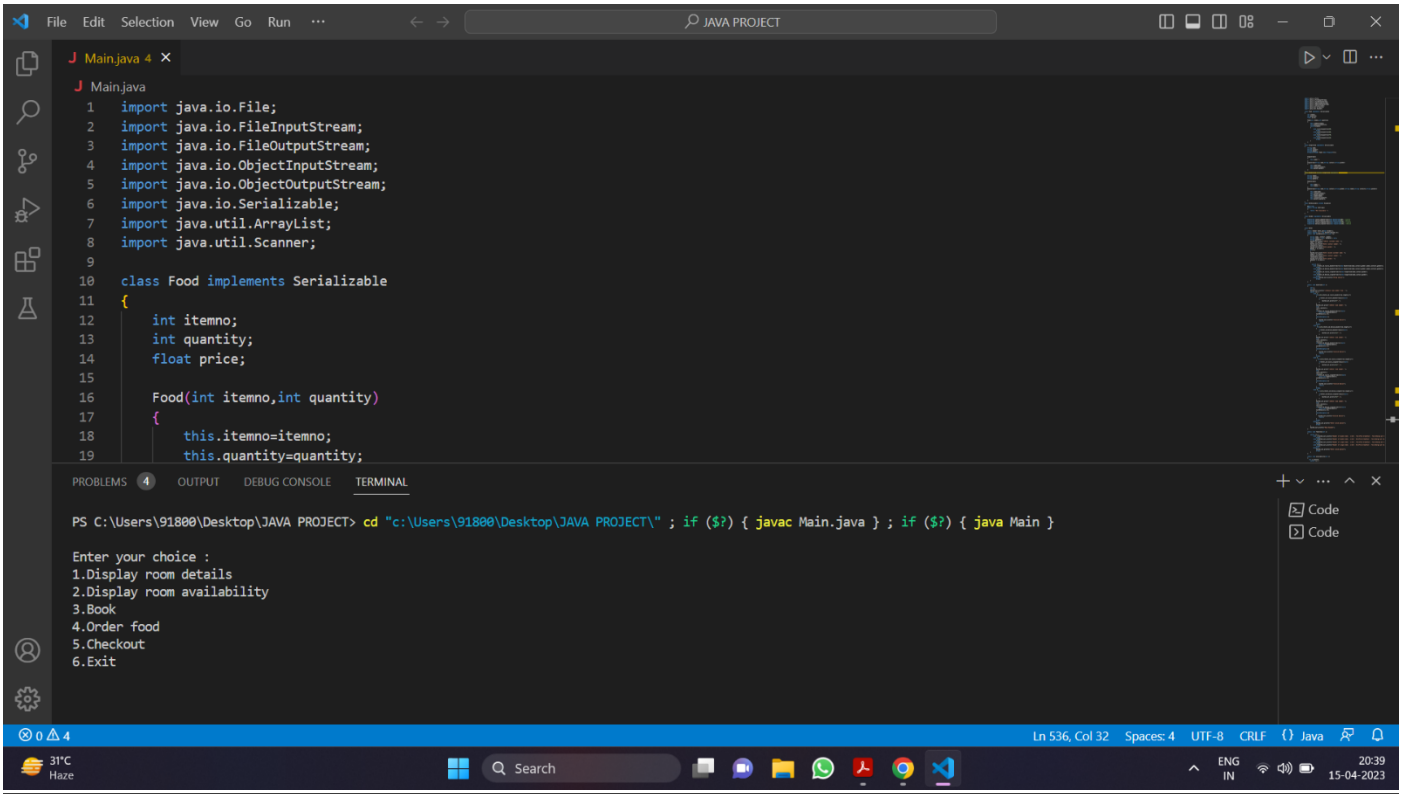

File Edit Selection View Go Run
JAVA PROJECT
EXPLORER
JAVA PROJECT
backup
Doubleroom.class
Food.class
holder.class
Hotel.class
Main.class
Main.java 4
NotAvailable.class
Singleroom.class
write.class
OUTLINE
TIMELINE
JAVA PROJECTS
Main.java 4
Main.java > Food > Food(int, int)
128 static void bookroom(int i)
129 {
130 int j;
131 int rn;
132 System.out.println(x:"\nChoose room number from : ");
133 switch (i) {
134 case 1:
135 for(j=0;j<hotel_ob.luxury_doublerrom.length;j++)
136 {
137 if(hotel_ob.luxury_doublerrom[j]==null)
138 {
139 System.out.print(j+1+",");
140 }
141 }
142 System.out.print(s:"\nEnter room number: ");
143 try{
144 rn=sc.nextInt();
145 rn--;
146 if(hotel_ob.luxury_doublerrom[rn]!=null)
147 throw new NotAvailable();
148 CustDetails(i,rn);
149 }
150 catch(Exception e)
151 {
152 System.out.println(x:"Invalid Option");
153 return;
154 }
155 break;
156 case 2:
157 for(j=0;j<hotel_ob.deluxe_doublerrom.length;j++)
158 {
159

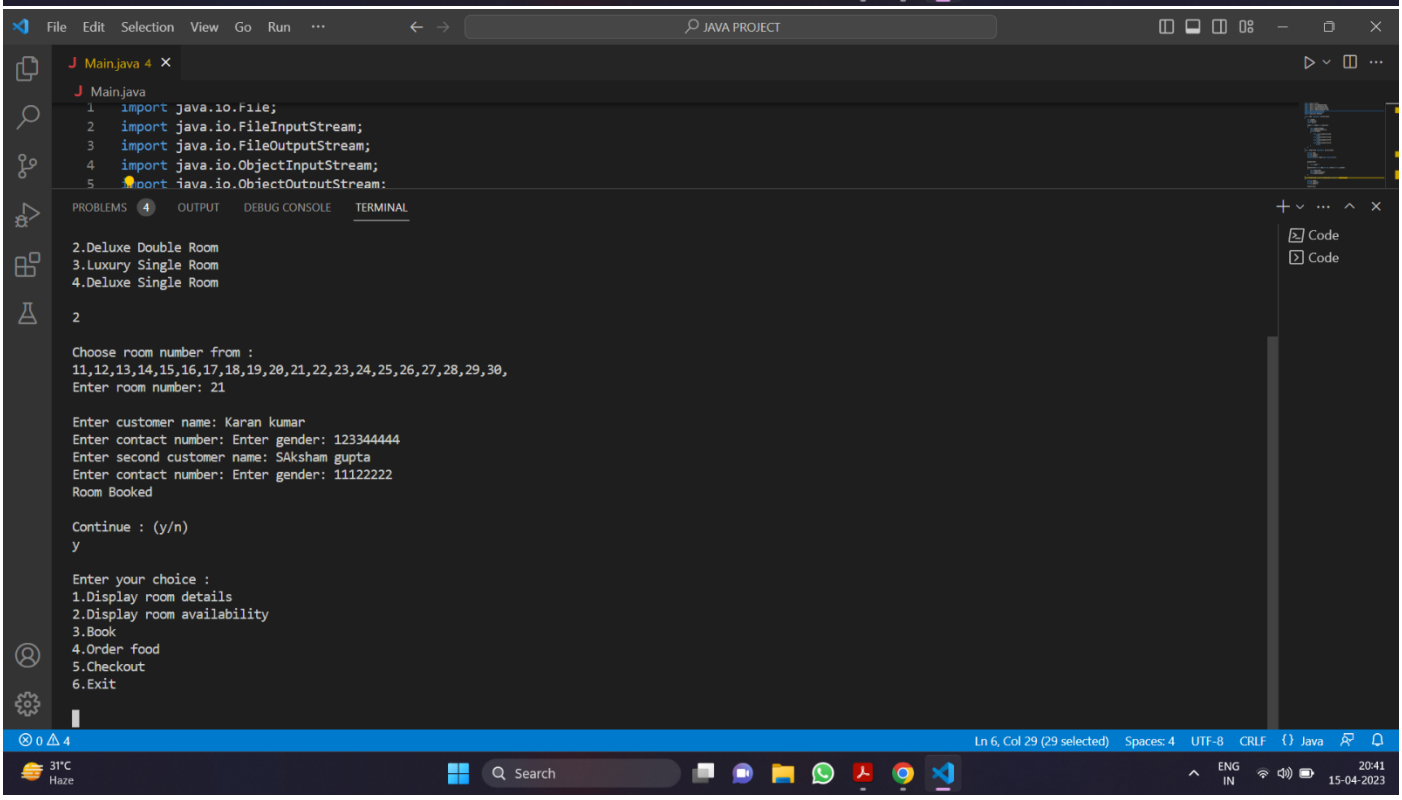

File Edit Selection View Go Run
JAVA PROJECT
EXPLORER
JAVA PROJECT
backup
Doubleroom.class
Food.class
holder.class
Hotel.class
Main.class
Main.java 4
NotAvailable.class
Singleroom.class
write.class
OUTLINE
TIMELINE
JAVA PROJECTS
Main.java 4
Main.java > Food > Food(int, int)
228 }
229
230 static void features(int i)
231 {
232 switch (i) {
233 case 1:System.out.println(x:"Number of double beds : 1\nAC : Yes\nFree breakfast : Yes\nCharge per day:40
234 break;
235 case 2:System.out.println(x:"Number of double beds : 1\nAC : No\nFree breakfast : Yes\nCharge per day:300
236 break;
237 case 3:System.out.println(x:"Number of single beds : 1\nAC : Yes\nFree breakfast : Yes\nCharge per day:22
238 break;
239 case 4:System.out.println(x:"Number of single beds : 1\nAC : No\nFree breakfast : Yes\nCharge per day:120
240 break;
241 default:
242 System.out.println(x:"Enter valid option");
243 break;
244 }
245 }
246
247 static void availability(int i)
248 {
249 int j,count=0;
250 switch (i) {
251 case 1:
252 for(j=0;j<10;j++)
253 {
254 if(hotel_ob.luxury_doublerrom[j]==null)
255 count++;
256 }
257 break;
258 case 2:
259 for(j=0;j<hotel_ob.deluxe_doublerrom.length;j++)

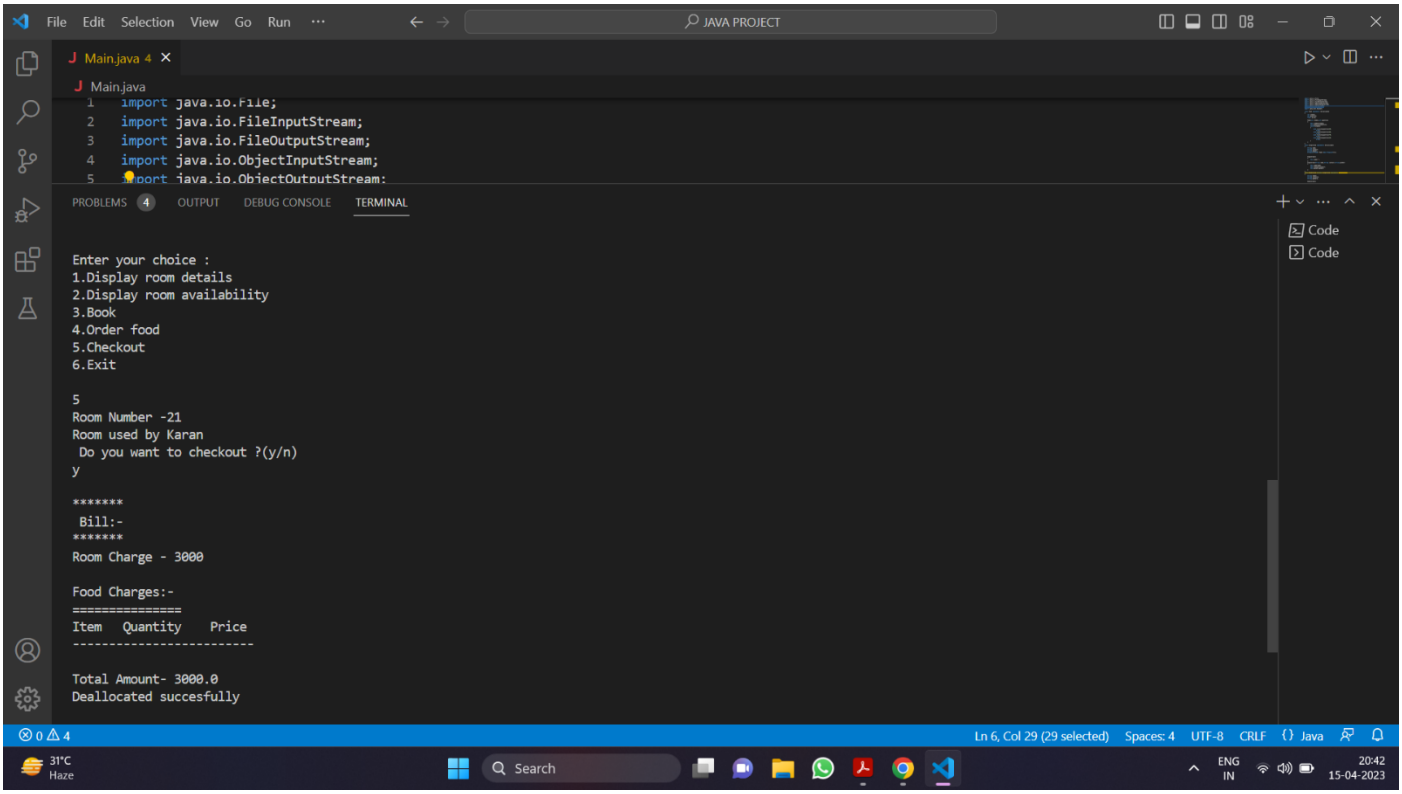

File Edit Selection View Go Run
JAVA PROJECT
EXPLORER
JAVA PROJECT
backup
Doubleroom.class
Food.class
holder.class
Hotel.class
Main.class
Main.java 4
NotAvailable.class
Singleroom.class
write.class
OUTLINE
TIMELINE
JAVA PROJECTS
Main.java 4
Main.java > Food > Food(int, int)
355 }
356
357 static void deallocate(int rn,int rtype)
358 {
359 int j;
360 char w;
361 switch (rtype) {
362 case 1:
363 if(hotel_ob.luxury_doublerrom[rn]!=null)
364 System.out.println("Room used by "+hotel_ob.luxury_doublerrom[rn].name);
365 else
366 {
367 System.out.println(x:"Empty Already");
368 return;
369 }
370 System.out.println(x:"Do you want to checkout ?(y/n)");
371 w=sc.next().charAt(index:0);
372 if(w=='y'||w=='Y')
373 {
374 bill(rn,rtype);
375 hotel_ob.luxury_doublerrom[rn]=null;
376 System.out.println(x:"Deallocated succesfully");
377 }
378
379 break;
380 case 2:
381 if(hotel_ob.deluxe_doublerrom[rn]!=null)
382 System.out.println("Room used by "+hotel_ob.deluxe_doublerrom[rn].name);
383 else
384 {
385 System.out.println(x:"Empty Already");
386 return;


File Edit Selection View Go Run
JAVA PROJECT
Main.java 4
Main.java > Main > main(String[])
521 case 1: System.out.println(x:"\nChoose room type :\n1.Luxury Double Room \n2.Deluxe Double Room \n3.Luxury Single Room \n4.Deluxe Single R
522 ch2 = sc.nextInt();
523 Hotel.features(ch2);
524 break;
525 case 2:System.out.println(x:"\nChoose room type :\n1.Luxury Double Room \n2.Deluxe Double Room \n3.Luxury Single Room\n4.Deluxe Single Roo
526 ch2 = sc.nextInt();
527 Hotel.availability(ch2);
528 break;
529 case 3:System.out.println(x:"\nChoose room type :\n1.Luxury Double Room \n2.Deluxe Double Room \n3.Luxury Single Room\n4.Deluxe Single Roo
530 ch2 = sc.nextInt();
531 Hotel.bookroom(ch2);
532 break;
533 case 4:
534 System.out.print(s:"Room Number -");
535 ch2 = sc.nextInt();
536 if(ch2>60)
537 System.out.println(x:"Room doesn't exist");
538 else if(ch2>40)
539 Hotel.order(ch2-41,rtype:4);
540 else if(ch2>30)
541 Hotel.order(ch2-31,rtype:3);
542 else if(ch2>10)
543 Hotel.order(ch2-11,rtype:2);
544 else if(ch2>0)
545 Hotel.order(ch2-1,rtype:1);
546 else
547 System.out.println(x:"Room doesn't exist");
548 break;
549 case 5:
550 System.out.print(s:"Room Number -");
551 ch2 = sc.nextInt();
552 if(ch2>60)

Output :


File Edit Selection View Go Run
JAVA PROJECT
Main.java 4
Main.java
1 import java.io.File;
2 import java.io.FileInputStream;
3 import java.io.FileOutputStream;
4 import java.io.ObjectInputStream;
5 import java.io.ObjectOutputStream;
6 import java.io.Serializable;
7 import java.util.ArrayList;
8 import java.util.Scanner;
9
10 class Food implements Serializable
11 {
12 int itemno;
13 int quantity;
14 float price;
15
16 Food(int itemno,int quantity)
17 {
18 this.itemno=itemno;
19 this.quantity=quantity;
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\91800\Desktop\JAVA PROJECT> cd "c:\Users\91800\Desktop\JAVA PROJECT\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
Enter your choice :
1.Display room details
2.Display room availability
3.Book
4.Order food
5.Checkout
6.Exit
Code
Code
Ln 536, Col 32 Spaces: 4 UTF-8 CRLF Java
31°C Haze
Search
ENG IN
20:39 15-04-2023


File Edit Selection View Go Run
JAVA PROJECT
Main.java 4
Main.java
1 import java.io.File;
2 import java.io.FileInputStream;
3 import java.io.FileOutputStream;
4 import java.io.ObjectInputStream;
5 import java.io.ObjectOutputStream;
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL
2.Deluxe Double Room
3.Luxury Single Room
4.Deluxe Single Room
2
Choose room number from :
11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,
Enter room number: 21
Enter customer name: Karan kumar
Enter contact number: Enter gender: 123344444
Enter second customer name: SAksham gupta
Enter contact number: Enter gender: 11122222
Room Booked
Continue : (y/n)
y
Enter your choice :
1.Display room details
2.Display room availability
3.Book
4.Order food
5.Checkout
6.Exit
Code
Code
Ln 6, Col 29 (29 selected) Spaces: 4 UTF-8 CRLF Java
31°C Haze
Search
ENG IN
20:41 15-04-2023


File Edit Selection View Go Run ···
JAVA PROJECT
Main.java 4
Main.java
1 import java.io.File;
2 import java.io.FileInputStream;
3 import java.io.FileOutputStream;
4 import java.io.ObjectInputStream;
5 import java.io.ObjectOutputStream;
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL
Code
Code
Enter your choice :
1.Display room details
2.Display room availability
3.Book
4.Order food
5.Checkout
6.Exit
5
Room Number -21
Room used by Karan
Do you want to checkout ?(y/n)
y
****
Bill:-
****
Room Charge - 3000
Food Charges:-
==========
Item Quantity Price

Total Amount- 3000.0
Deallocated succesfully
Ln 6, Col 29 (29 selected) Spaces: 4 UTF-8 CRLF Java
31°C Haze
Search
ENG IN
20:42 15-04-2023