

# DELHI PUBLIC SCHOOL BHILAI



## A COMPUTER SCIENCE PROJECT ON **FLIGHT RESERVATION SYSTEM** SESSION - 2018-'19

Guided by  
**Mr. A. H. Zunjani**

Submitted by  
**Aamin Dev**  
*(Team inclusive of  
Mrinal Bhawe and Charvy Koshta)*

# CERTIFICATE

This is to certify that the project work for

## **COMPUTER SCIENCE**

entitled

## **FLIGHT RESERVATION SYSTEM**

Based on the Curriculum of C.B.S.E has been  
Submitted by

**AAMIN DEV**

**STUDENT OF CLASS - XII C**  
**OF**

**DELHI PUBLIC SCHOOL, BHILAI**

The report has been submitted for the academic session  
2018-'19 towards the partial fulfillment of requirements of  
C.B.S.E. Board

It is certified that the project embodies his original work and  
has been carried out under my guidance.

Signature of Internal

Signature of External

# PREFACE

This project has been carried out and presented as per the requirements of CBSE for class 12<sup>th</sup>. As every project involves time and efforts and requires some innovative ideas to present the project in most simple and creative manner as possible, providing beneficial to the readers. I too have tried to make the project fascinating and absorbing.

Though every care has been taken to keep the presentation descriptive, simple and catchy, measuring up all needs of topics, any suggestions for further improvement and enhancement of the project may thankfully be acknowledged.

# ACKNOWLEDGEMENT

Every successful task has some support behind it. I express my deepest gratitude to our Computer Science teacher, MR. A. H. ZUNJANI, whose guidance and inspiration have been the greatest help to me in bringing out this work in its present shape.

His help, support and constructive feedback helped me in improving the code and the UI.

I am grateful to our Principal, MR. PARSHANT VASHISHTA, who made us available of all facilities throughout the years, without which the project would not have been possible.

- Aamin Dev

# INDEX

## PAGE CONTENT

1	Certificate
2	Preface
3	Acknowledgement
4	Index
5	Code
6	Structures, Global Variables, Function Prototypes
7	Function Prototypes, main(), +3 <i>functions</i>
8	init(), login()
9	line(), userMenu(), checkPNR()
10	mainMenu(), adminMenu()
11	addFlight(), createUser()
12	createUser() <i>continued</i>
13	viewFlights(), viewAllPNRs()
14	viewBookings()
15	viewAllFlights, continueAsGuest(), bookFlight()
16	displayStruct() +1 <i>overload</i>
17	swapFlightStruct(), sortFlight(), +2 <i>functions</i>
18	loadingScreen() <i>continued</i>
19	Output
20	Output <i>login as user</i>
21	Output <i>continue as guest</i>
22	Output <i>login as admin</i>
23	Output <i>login as admin + create user</i>
24	Bibliography

CODE



```

1  //A PROGRAM TO MANAGE ALL THE FLIGHT BOOKINGS
2  //INCLUDING LOGGING IN AND ALL SORTS OF STUFF
3  //MADE IN 32 HOURS, I guess
4
5
6  #include <fstream.h>           //For file management as well is basic IO
7  #include <string.h>           //For strlen and strcmp
8  #include <stdlib.h>           //For random functions
9  #include <stdio.h>            //For gets()
10 #include <conio.h>             //For getch() and getche()
11 #include <time.h>             //For time()
12 #include <dos.h>              //For delay()
13
14
15 #define true 1                //Much better than typing return 1
16 #define false 0              //Much better than typing return 0
17
18
19 char fileUsername[15] = "guest"; //global username who is logged in
20 const char FILEPNR[8] = "pnr.txt"; //file for storing PNR
21 const char FILEDB[13] = "database.txt"; //file for storing username and password
22 const char FILENAME[12] = "flights.txt"; //file for storing flight details
23 static const char ALPHANUM[37] = "0123456789" //using these characters only to generate PNR
24 "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
25
26 int loggedAsGuest = true;      //Booleans to
27 int loggedAsAdmin = false;    //check who
28 int loggedAsUser = false;     //is logged in
29
30
31 struct PNR{                   //Structure for storing in FILEPNR
32     char paxName[24];
33     char FlightNumber[7];
34     int paxCount;
35     char PNR[7];
36     char username[15];
37 };
38
39 struct Time{                  //Structure to help with arrival and departure times
40     char hours[3];
41     char minutes[3];
42 };
43
44 struct Flight{                //Structure for storing in FILENAME
45     char FlightNumber[7];
46     char source[4];
47     char destination[4];
48     Time departure;
49     Time arrival;
50 };
51
52 struct Database{              //Structure for storing in FILEDB
53     char username[15];
54     char password[24];
55 };
56
57
58 int PNRExists(char PNRNumber[7]); //returns true if PNR already exists in file, else false
59 int flightExists(char fNumber[6]); //returns true if flight already exists in file, else false
60 int usernameAvailable(char* username); //returns false if username already exists in file, else true
61
62
63 void init();                  //Initializes files if first run
64 void login();                 //Logs in the user
65 void line(int);               //prints out i times lines
66 void userMenu();              //menu for regular user
67 void checkPNR();              //checks the validity of PNR and prints it
68 void mainMenu();              //master menu
69 void adminMenu();             //menu for admin ONLY
70 void addFlight();              //function available ONLY for ADMIN to add flight to file
71 void createUser();            //create a new user if not exists
72 void viewFlights();           //function to view available flights on routes
73 void viewAllPNRs();           //function available ONLY for ADMIN to see all PNRs in file

```

```

74 void viewBookings(); //function to see bookings in file
75 void viewAllFlights(); //function available ONLY for ADMIN to see all flights in file
76 void continueAsGuest(); //to continue without login or signing up
77 void bookFlight(Flight flight[], int); //function to book a flight selected
78 void displayStruct(Flight flight[], int j); //function to display the flight structure in presentable format
79 void displayStruct(Flight flight, PNR pnr); //same as above but referenced with PNR
80 void swapFlightStruct(Flight &x, Flight &y); //just like swapping integers, but this time its structures
81 void displayPNRStruct(Flight flight, PNR pnr, int i); //function to display the PNR in a presentable format
82 void sortFlight(Flight flight[], int); //function to sort flights by departure > arrival > name in asc order
83 void loadingScreen(char *string, int times, int); //best part of this program(animatesFakeLoadingScreen)
84
85
86 int main(){
87     init(); //Initializes files if first run
88     mainMenu(); //master menu
89     getch();
90     return 0;
91 }
92
93 int PNRExists(char PNRNumber[7]){ //returns true if PNR already exists in file, else false
94     PNR pnr;
95     ifstream file;
96
97     file.open(FILEPNR, ios::in);
98
99     if (!file) //file does not exist, therefore no PNR exists
100         return false;
101
102     while (!file.eof()){
103         file.read((char*) &pnr, sizeof(PNR));
104         if (!strcmp(PNRNumber, pnr.PNR)){ //pnr of such kind exists
105             file.close();
106             return true;
107         }
108     }
109
110     file.close();
111
112     return false; //No PNR was found
113 }
114
115 int flightExists(char fNumber[6]){ //returns true if flight already exists in file, else false
116     Flight flight;
117     ifstream file;
118
119     file.open(FILENAME, ios::in);
120
121     if (!file) //file does not exist, therefore no flight exists
122         return false;
123
124     while (!file.eof()){
125         file.read((char*) &flight, sizeof(Flight));
126         if (!strcmp(fNumber, flight.FlightNumber)){ //flight of such kind exists
127             file.close();
128             return true;
129         }
130     }
131
132     file.close();
133
134     return false; //No Flights were found
135 }
136
137 int usernameAvailable(char* username){ //returns false if username already exists in file, else true
138     Database database;
139     ifstream file;
140
141     file.open(FILEEDB, ios::in);
142
143     if (!file) //file does not exist, therefore no username exists
144         return true;
145
146     while (!file.eof()){

```



```

147     file.read((char *) &database, sizeof(database));
148     if (!strcmp(database.username, username)){ //username of such kind exists
149         file.close();
150         return false;
151     }
152 }
153
154 file.close();
155 return true; //No username was found
156 }
157
158
159 void init(){ //Initializes files if first run
160     if (usernameAvailable((char *) "admin")){ //Sees if "admin" is available or not
161         ofstream file;
162         file.open(FILEDB, ios::out);
163         Database database;
164         strcpy(database.username, "admin");
165         strcpy(database.password, "admin"); //DO NOT EVER DO THIS IN REAL LIFE EVER!!!!
166         file.write((char*) &database, sizeof(Database));
167         file.close();
168     }
169 }
170
171 void login(){ //Logs in the user
172     int loginSuccess = false; //boolean to see if login successful or not
173     ifstream file;
174     Database database;
175     file.open(FILEDB, ios::in);
176
177     cout << "Login\n";
178     line(28);
179     char username[15];
180     char password[25];
181     char ch;
182     int i;
183     cout << "username : ";
184     cin >> username;
185     cout << "password : ";
186     for (i = 0;;){ //loop to enter password
187         ch = getch();
188
189         if (ch >= 33 && ch <= 126){ //logical thing entered
190             password[i++] = ch;
191             cout << "*"; //displaying password is not a good idea
192         }
193
194         if (ch == 127 && i >= 1){ //mimicking backspace
195             cout<<"\b\b"; //removing the last index
196             --i;
197         }
198
199         if (ch == 10 && i >= 1){ //checks if enter pressed
200             password[i] = '\0'; //adds null character and ends loop
201             break;
202         }
203     }
204     loadingScreen((char *) "Logging in", 8, false); //fake loading screen to show logging in
205
206     while (!file.eof()){
207         file.read((char *) &database, sizeof(Database));
208         if (!strcmp(username, database.username)){ //correct username
209             if (!strcmp(password, database.password)){ //correct password
210                 if (!strcmp(username, "admin")){ //checks if admin is logged in
211                     loggedAsAdmin = true; //yes indeed
212                     loggedAsUser = false;
213                     loggedAsGuest = false;
214                 }
215                 loginSuccess = true; //confirms login success
216             }
217         }
218     }
219 }

```

```

220 if (!loginSuccess){                                //if login failed
221     char ch;
222     cout << "Incorrect username or password!\n";
223     cout << "Try logging in again? (y/n)";
224     ch = getch();
225     cout << '\n';
226     if (ch == 'y' || ch == 'Y'){                    //try logging in again
227         loadingScreen((char *) "Returning to login page", 3, false);
228         login();                                    //try again
229     }
230 } else {
231     cout << "Logged in successfully!\n";
232     if (loggedAsAdmin){                             //opens admin page then
233         strcpy(fileUsername, "admin");
234         adminMenu();                                //menu for ADMIN ONLY
235     } else {                                         //opens regular user page
236         strcpy(fileUsername, username);
237         loggedAsUser = true;
238         loggedAsAdmin = false;
239         loggedAsGuest = false;
240         userMenu();                                //menu for regular user
241     }
242 }
243 }
244
245 void line(int k){                                    //prints out i times lines
246     for (int i = 0; i < k; ++i)
247         cout << '-';
248     cout << '\n';
249 }
250
251 void userMenu(){                                    //menu for regular user
252     char ch;
253     clrscr();
254     cout << "Choose an option to continue" << '\n';
255     line(30);
256     cout << "1 : Book flight \n";
257     cout << "2 : View bookings \n";
258     cout << "3 : PNR Enquiry \n";
259     cout << "0 : Logout and Quit" << '\n';
260     ch = getch();
261     switch (ch){
262         case '1':
263             loadingScreen((char *) "Opening book flights page", 6, true);
264             viewFlights();                          //to book a flight
265             break;
266         case '2':
267             loadingScreen((char *) "Opening view bookings page", 4, true);
268             viewBookings();                         //to view all bookings
269             break;
270         case '3':
271             loadingScreen((char *) "Opening PNR Enquiry page", 3, true);
272             checkPNR();                             //to check a PNR
273         case '0':
274             break;
275         default:
276             userMenu();                             //refreshes the whole thing
277             break;
278     }
279 }
280
281 void checkPNR(){                                    //checks the validity of PNR and prints it
282     PNR pnr;
283     Flight flight;
284     ifstream filePNR;
285     ifstream fileFlight;
286     int PNRExists = false;                          //boolean to see if PNR exists in file
287     char PNRNumber[7];
288     char flightNumber[7];
289
290     cout << "Enter PNR : ";
291     cin.getline(PNRNumber, sizeof(PNRNumber));
292

```

```

293 filePNR.open(FILEPNR, ios::in);
294
295 while(!filePNR.eof()){
296     filePNR.read((char*) &pnr, sizeof(PNR));
297     if (!strcmpi(PNRNumber, pnr.PNR)){ //PNR of such type exists
298         PNRExists = true; //PNR Existence confirmed
299         strcpy(flightNumber, pnr.FlightNumber);
300         break;
301     }
302 }
303
304 filePNR.close();
305
306 if (!PNRExists)
307     cout << "Sorry, the entered PNR is not valid!\n";
308 else {
309     fileFlight.open(FILENAME, ios::in);
310     while (!fileFlight.eof()){ //To get flight info
311         fileFlight.read((char*) &flight, sizeof(Flight)); //associated with
312         if (!strcmp(flightNumber, flight.FlightNumber)){ //this PNR
313             displayStruct(flight, pnr); //To display the data
314             break; //in a presentable format
315         }
316     }
317 }
318
319 fileFlight.close();
320
321 }
322
323 void mainMenu(){ //master menu
324     char ch;
325     clrscr();
326     cout << "Welcome to Generic Airline Booking! \n";
327     line(35);
328     cout << "Choose an option to continue" << '\n';
329     line(35);
330     cout << "1 : Login \n";
331     cout << "2 : Create account \n";
332     cout << "3 : Continue as guest \n";
333     cout << "0 : Quit" << '\n';
334     ch = getch();
335     switch (ch){
336         case '1':
337             loadingScreen((char *) "Opening login page", 6, true);
338             login(); //to open login screen
339             break;
340         case '2':
341             loadingScreen((char *) "Creating new account", 8, true);
342             createUser(); //create new user
343             break;
344         case '3':
345             loadingScreen((char *) "Continuing as guest", 8, false);
346             continueAsGuest(); //to continue as guest
347         case '0':
348             break;
349         default:
350             mainMenu(); //refreshes the whole thing
351             break;
352     }
353 }
354
355 void adminMenu(){ //menu for ADMIN ONLY
356     char ch;
357     clrscr();
358     cout << "Choose an option to continue" << '\n';
359     line(30);
360     cout << "1 : Add Flights \n";
361     cout << "2 : View all PNRs \n";
362     cout << "3 : View all Flights \n";
363     cout << "0 : Logout and Quit" << '\n';
364     ch = getch();
365     switch (ch){

```

```

366     case '1':
367         loadingScreen((char *) "Opening add flights page", 6, true);
368         addFlight();           //adds another flight
369         break;
370     case '2':
371         loadingScreen((char *) "Opening view all PNRs page", 8, true);
372         viewAllPNRs();         //views all the PNRs
373         break;
374     case '3':
375         loadingScreen((char *) "Opening view all flights page", 8, true);
376         viewAllFlights();      //views all the flight
377         break;
378     case '0':
379         break;
380     default:
381         adminMenu();           //refreshes the whole thing
382     }
383 }
384
385 void addFlight(){             //function available for ADMIN ONLY to add a flight to file
386     ofstream file;
387     Flight flight;
388
389     file.open(FILENAME, ios::app);
390
391     cout << "Enter FlightNumber\t: ";
392     cin >> flight.FlightNumber;
393
394     if (flightExists(flight.FlightNumber)){           //checks if flight number in use or not
395         char ch;
396         cout << "ERROR! Flight already exists!\n";
397         file.close();
398         cout << "Still add another flight? (y/n) : ";   //to continue with other flight number
399         ch = getch();
400         cout << '\n';
401         if (ch == 'y' || ch == 'Y'){
402             loadingScreen((char *) "Refreshing", 2, false);
403             addFlight();           //refresh
404         } else
405             loadingScreen((char *) "Returning to menu", 4, false);
406             adminMenu();           //go back to admin menu
407     } else {
408         cout << "Enter Source\t\t: ";
409         cin >> flight.source;
410         cout << "Enter destination\t: ";
411         cin >> flight.destination;
412         cout << "Enter Departure hours\t: ";
413         cin >> flight.departure.hours;
414         cout << "Enter Departure minutes\t: ";
415         cin >> flight.departure.minutes;
416         cout << "Enter Arrival hours\t: ";
417         cin >> flight.arrival.hours;
418         cout << "Enter Arrival minutes\t: ";
419         cin >> flight.arrival.minutes;
420
421         file.write((char*) &flight, sizeof(Flight));
422         cout << flight.FlightNumber << " successfully added!\n";
423         file.close();
424         char ch;
425         cout << "Add another flight? (y/n) : ";           //to add more flights
426         ch = getch();
427         cout << '\n';
428         if (ch == 'y' || ch == 'Y'){
429             loadingScreen((char *) "Opening add flights page", 2, true);
430             addFlight();           //to add more flights
431         } else {
432             loadingScreen((char *) "Returning to menu", 2, false);
433             adminMenu();           //back to admin menu
434         }
435     }
436 }
437
438 void createUser(){           //create new user if not exists

```

```

439 ofstream file;
440 Database database;
441
442 clrscr();
443 cout << "Create New User\n";
444 line(20);
445 char username[15];
446 char password[25];
447 char passwordRepeat[25];
448 char ch;
449 int i;
450 cout << "Select username : ";
451 cin >> username;
452 if (!usernameAvailable(username) || !strcmp(username, "guest")) { //to see if username available
453     cout << "Username not available! Press 1 to try again!\n";
454     char ch;
455     ch = getch();
456     if (ch == '1')
457         createUser();
458     return;
459 } else
460     cout << "Username available!\n";
461 cout << "Password : ";
462 cin >> password;
463 for (i = 0; ;){ //complex algorithm to mimic password entry
464     ch = getch();
465
466     if (ch >= 33 && ch <= 126){
467         password[i++] = ch;
468         cout << "*";
469     }
470
471     if (ch == 127 && i >= 1){
472         cout << "\b\b";
473         --i;
474     }
475
476     if (ch == 10 && i >= 1){
477         password[i] = '\0';
478         break;
479     }
480 }
481 cout << "Reenter password : "; //to prevent password mismatch
482 for (i = 0; ;){
483     ch = getch();
484     for (i = 0; ;){
485         ch = getch();
486
487         if (ch >= 33 && ch <= 126){
488             passwordRepeat[i++] = ch;
489             cout << "*";
490         }
491
492         if (ch == 127 && i >= 1){
493             cout << "\b\b";
494             --i;
495         }
496
497         if (ch == 10 && i >= 1){
498             passwordRepeat[i] = '\0';
499             break;
500         }
501     }
502 }
503 if (strcmp(password, passwordRepeat)){ //password mismatch
504     cout << "\nPasswords do not match!!!\n";
505     cout << "Please try again!\n";
506     cout << "Press 1 to try again!\n";
507     char ch;
508     ch = getch();
509     if (ch == '1')
510         createUser();
511     else

```

```

512     return;
513 } else { //everything good, create user
514     loadingScreen((char *) "creating user", 10, true);
515     strcpy(database.username, username);
516     strcpy(database.password, password);
517     file.open(FILEDB, ios::app);
518     file.write((char *) &database, sizeof(Database));
519     file.close();
520     loadingScreen((char *) "User created, heading to login page", 10, true);
521     login(); //continue to login page
522 }
523 }
524
525 void viewFlights(){ //function to view flights on a particular route
526     Flight flight;
527     Flight send[10];
528     ifstream file;
529     int i = 0;
530     int j = 0;
531     int exist = false; //to see if at least one flight exists
532     char source[4];
533     char destination[4];
534
535     cout << "Book a flight!\n";
536     line(24);
537     cout << "Enter source    : ";
538     cin >> source;
539     cout << "Enter destination : ";
540     cin >> destination;
541
542     file.open(FILENAME, ios::in);
543
544     while(1){
545         file.read((char*) &flight, sizeof(Flight));
546         if (file.eof())
547             break;
548         if (!strcmpi(source, flight.source)){
549             if (!strcmpi(destination, flight.destination)){
550                 send[j++] = flight; //stores the flights in an on that route to display
551                 exist = true;
552             }
553         }
554     }
555
556     if (exist){ //if at least one flight exists
557         int choice;
558         displayStruct(send, j); //to display structure in a presentable format
559         tryAgain:
560             cout << "Select S.no to book flight (0 to quit) : ";
561             cin >> choice;
562             if (!choice) //quit
563                 return;
564             else if (choice > j || choice < 0){
565                 cout << "Out of bounds!\n";
566                 goto tryAgain;
567             } else
568                 bookFlight(send, choice); //finalizes bookings
569     }
570     else //no flights available
571         cout << "\nOh oh, no flights on this route available :\n\n";
572
573     file.close();
574
575 }
576
577 void viewAllPNRs(){ //function available ONLY for ADMIN to see all the PNRs in file
578     PNR pnr;
579     int i = 0;
580     Flight flight;
581     ifstream filePNR;
582     ifstream fileFlight;
583     char flightNumber[7];
584

```



```

585 filePNR.open(FILEPNR, ios::in);
586 fileFlight.open(FILENAME, ios::in);
587
588 while (1){
589     filePNR.read((char*) &pnr, sizeof(PNR));
590     fileFlight.seekg(0, ios::beg);
591     if (filePNR.eof())
592         break;
593     while (!fileFlight.eof()){
594         fileFlight.read((char*) &flight, sizeof(Flight));
595         if (!strcmp(pnr.FlightNumber, flight.FlightNumber)){//sees if FlightNo. of pnr matches with flights.txt
596             displayPNRStruct(flight, pnr, i++);           //displays the structure in a..you know it
597             break;
598         }
599     }
600 }
601
602 line(112);
603
604 filePNR.close();
605 fileFlight.close();
606
607 cout << "Press 1 to go back" << '\n';
608 char ch;
609 ch = getch();
610 if (ch == '1'){
611     loadingScreen((char *) "Returning to menu", 2, false);
612     adminMenu();           //goes back to admin menu
613 }
614 }
615
616 void viewBookings(){           //function to see all the bookings made
617     PNR pnr;
618     int i = 0;
619     Flight flight;
620     ifstream filePNR;
621     ifstream fileFlight;
622     char flightNumber[7];
623
624     filePNR.open(FILEPNR, ios::in);
625     fileFlight.open(FILENAME, ios::in);
626
627     while (!filePNR.eof()){
628         filePNR.read((char*) &pnr, sizeof(PNR));
629         fileFlight.seekg(0, ios::beg);
630         if (filePNR.eof())
631             break;
632         if (!strcmp(pnr.username, fileUsername)){ //checks username, to only display bookings of logged in user
633             if (strcmp(pnr.username, "guest")){
634                 while (!fileFlight.eof()){
635                     fileFlight.read((char*) &flight, sizeof(Flight));
636                     if (!strcmp(pnr.FlightNumber, flight.FlightNumber)){ //cross references the flight number
637                         displayPNRStruct(flight, pnr, i++);
638                         break;
639                     }
640                 }
641             }
642         }
643     }
644
645     line(112);
646
647     filePNR.close();
648     fileFlight.close();
649
650     cout << "Press 1 to go back" << '\n';
651     char ch;
652     ch = getch();
653     if (ch == '1'){
654         loadingScreen((char *) "Returning to menu", 2, false);
655         userMenu();           //back to the user menu
656     }
657 }

```

```

658 void viewAllFlights(){           //function available for ADMIN ONLY to see all the flights in file
659     int i = 1;
660     Flight flight;
661     ifstream file;
662
663     file.open(FILENAME, ios::in);
664
665     if (file){           //if file is in good health
666         line(71);
667         cout << "S.No\tFlightNumber\tSource\tDestination\tDeparture\tArrival\n";
668         line(71);
669     }
670
671     while (1){
672         file.read((char*) &flight, sizeof(Flight));
673         if (file.eof())
674             break;
675         cout << i++ << '\t' << flight.FlightNumber << "\t\t" << flight.source           //displays the
676             << '\t' << flight.destination << "\t\t" << flight.departure.hours           //data in
677             << ':' << flight.departure.minutes << "\t\t"                               //a presentable
678             << flight.arrival.hours << ':' << flight.arrival.minutes                   //format
679             << '\n';
680     }
681
682     line(71);
683
684     file.close();
685
686     cout << "Press 1 to go back\n";
687     char ch;
688     ch = getch();
689     if (ch == '1'){
690         loadingScreen((char *) "Returning to menu", 4, false);
691         adminMenu();           //back to the admin menu
692     }
693 }
694
695 void continueAsGuest(){           //function to continue without login or signing up
696     char ch;
697     clrscr();
698     cout << "Choose an option to continue" << '\n';
699     line(30);
700     cout << "1 : Book flight \n";
701     cout << "2 : PNR Enquiry \n";
702     cout << "0 : Quit" << '\n';
703     ch = getch();
704     switch (ch){
705         case '1':
706             loadingScreen((char *) "Opening book flights page", 6, true);
707             viewFlights();           //to book a flight
708             break;
709         case '2':
710             loadingScreen((char *) "Opening PNR Enquiry page", 3, true);
711             checkPNR();           //to see PNR details
712             cout << "Press 1 to go back\n";
713             char choice;
714             choice = getch();
715             if (choice == '1'){
716                 loadingScreen((char *) "Returning to menu", 4, false);
717                 continueAsGuest();           //refresh
718             }
719             break;
720         case '0':
721             break;
722         default:
723             continueAsGuest();           //refresh
724             break;
725     }
726 }
727
728 void bookFlight(Flight flight[], int c){           //function to book a flight selected
729     ofstream file;
730

```

```

731     PNR pnr;
732     char passengersChar;
733     int paxCount, reference;
734     char paxName[24];
735     char PNRNumber[7];
736
737     file.open(FILEPNR, ios::app);
738     --c;
739
740     cout << "Booking on " << flight[c].FlightNumber << '\n';
741     cout << "Enter number of passengers (Maximum 10) : ";
742     passengersChar = getche();
743     cout << '\n';
744     paxCount = passengersChar - '0';
745
746     cout << "Enter name of any 1 passenger : ";
747     gets(paxName);
748
749     srand(time(NULL));
750
751     do {
752         for (int i = 0; i < 7; ++i)
753             PNRNumber[i] = ALPHANUM[rand() % 37];    //generates random PNR
754             PNRNumber[6] = 0;
755     } while(PNRExists(PNRNumber));    //to see if the PNR does not already exists
756
757     pnr.paxCount = paxCount;
758     strcpy(pnr.PNR, PNRNumber);
759     strcpy(pnr.paxName, paxName);
760     strcpy(pnr.username, fileUsername);
761     strcpy(pnr.FlightNumber, flight[c].FlightNumber);
762
763     file.write((char*) &pnr, sizeof(pnr));
764
765     cout << "Your PNR : " << PNRNumber << '\n';
766     cout << "Please go to your nearest airport and pay against this PNR\n";
767
768     file.close();
769 }
770
771 void displayStruct(Flight flight[], int j){    //function to display the flight structure you know how
772     if (j != 1)    //if more than 1 flights
773         sortFlight(flight, j);    //sort it without asking questions how the algorithm works
774
775     line(71);
776     cout << "S.No\tFlightNumber\tSource\tDestination\tDeparture\tArrival\n";
777     line(71);
778
779     for (int i = 0; i < j; ++i){
780         cout << i+1 << '\t' << flight[i].FlightNumber << "\t\t" << flight[i].source
781             << '\t' << flight[i].destination << "\t\t" << flight[i].departure.hours
782             << ':' << flight[i].departure.minutes << "\t\t"
783             << flight[i].arrival.hours << ':' << flight[i].arrival.minutes
784             << '\n';
785     }
786
787     line(71);
788 }
789
790 void displayStruct(Flight flight, PNR pnr){    //overloaded function to display one structure with PNR number
791     line(98);
792     cout << "FlightNumber\tSource\tDestination\tDeparture\tArrival\tPassengers\tName\n";
793     line(98);
794
795     cout << flight.FlightNumber << "\t\t" << flight.source
796         << '\t' << flight.destination << "\t\t" << flight.departure.hours
797         << ':' << flight.departure.minutes << "\t\t"
798         << flight.arrival.hours << ':' << flight.arrival.minutes
799         << '\t' << pnr.paxCount << "\t\t" << pnr.paxName << '\n';
800
801     line(98);
802 }
803

```

```

804 void swapFlightStruct(Flight &x, Flight &y){ //function to swap 2 flight identifiers
805     Flight temp = x;
806     x = y;
807     y = temp;
808 }
809
810 void displayPNRStruct(Flight flight, PNR pnr, int i){ //display pnr with flight details
811     if (i == 0){
812         line(112);
813         cout << "PNR\tFlightNumber\tSource\tDestination\tDeparture\tArrival\tPassengers\tName\t\tUsername\n";
814         line(112);
815     }
816
817     cout << pnr.PNR << '\t' << flight.FlightNumber << "\t\t" << flight.source
818         << '\t' << flight.destination << "\t\t" << flight.departure.hours
819         << ':' << flight.departure.minutes << "\t\t"
820         << flight.arrival.hours << ':' << flight.arrival.minutes
821         << '\t' << pnr.paxCount << "\t\t" << pnr.paxName << '\t' << pnr.username << '\n';
822 }
823
824 void sortFlight(Flight flight[], int n){ //function to sort flights by departure > arrival > name in asc order
825     for (int i = 0; i < n - 1; ++i){ //don't ever bother on how this works
826         for (int j = 0; j < n - i - 1; ++j){ //take it for granted that it works and your life will be
827             if (flight[j].departure.hours[0] > flight[j + 1].departure.hours[0]) //a whole lot simpler
828                 swapFlightStruct(flight[j], flight[j + 1]); //use this function
829             else if (flight[j].departure.hours[0] == flight[j + 1].departure.hours[0]){ //without asking
830                 if (flight[j].departure.hours[1] > flight[j + 1].departure.hours[1]) //any questions
831                     swapFlightStruct(flight[j], flight[j + 1]);
832                 else if (flight[j].departure.hours[1] == flight[j + 1].departure.hours[1]){
833                     if (flight[j].departure.minutes[0] > flight[j + 1].departure.minutes[0])
834                         swapFlightStruct(flight[j], flight[j + 1]);
835                     else if (flight[j].departure.minutes[0] == flight[j + 1].departure.minutes[0]){
836                         if (flight[j].departure.minutes[1] > flight[j + 1].departure.minutes[1])
837                             swapFlightStruct(flight[j], flight[j + 1]);
838                     else if (flight[j].departure.minutes[1] == flight[j + 1].departure.minutes[1]){
839                         if (flight[j].arrival.hours[0] > flight[j + 1].arrival.hours[0])
840                             swapFlightStruct(flight[j], flight[j + 1]);
841                         else if (flight[j].arrival.hours[0] == flight[j + 1].arrival.hours[0]){
842                             if (flight[j].arrival.hours[1] > flight[j + 1].arrival.hours[1])
843                                 swapFlightStruct(flight[j], flight[j + 1]);
844                             else if (flight[j].arrival.hours[1] == flight[j + 1].arrival.hours[1]){
845                                 if (flight[j].arrival.minutes[0] > flight[j + 1].arrival.minutes[0])
846                                     swapFlightStruct(flight[j], flight[j + 1]);
847                             else if (flight[j].arrival.minutes[0] == flight[j + 1].arrival.minutes[0]){
848                                 if (flight[j].arrival.minutes[1] > flight[j + 1].arrival.minutes[1])
849                                     swapFlightStruct(flight[j], flight[j + 1]);
850                             else if (flight[j].arrival.minutes[1] == flight[j + 1].arrival.minutes[1]){
851                                 for (int k = 0; k < strlen(flight[j].FlightNumber); ++k){
852                                     if ((int) flight[j].FlightNumber[k] - (int) flight[j + 1].FlightNumber[k] > 0){
853                                         swapFlightStruct(flight[j], flight[j + 1]);
854                                     }
855                                 }
856                             }
857                         }
858                     }
859                 }
860             }
861         } //take a moment to appreciate how beautiful it looks
862     }
863 }
864
865 }
866 }
867 }
868
869 void loadingScreen(char *string, int times, int showLoading){ //the dopest part of the program
870     clrscr();
871     delay(250); //delay taken in milliseconds (as good as 0.25s)
872     if (showLoading){ //to show a loading text or continue with user text
873         cout << string << '\n';
874         cout << "Loading...\n";
875     } else {
876         cout << string << "... \n";

```

```

877 }
878 delay(250);           //add a delay of a quarter of a second (4FPS)
879 int loadingAnim = 0;
880 int iterations = 0;
881 while (iterations < times){ //as long as we have time to run
882     clrscr();
883     if (showLoading)
884         cout << string << '\n';
885     switch (loadingAnim){
886     case 0:             //stage 1 of loading symbol
887         if (showLoading)
888             cout << "Loading...\\n";
889         else
890             cout << string << "...\\n";
891         break;
892     case 1:             //stage 2 of loading symbol
893         if (showLoading)
894             cout << "Loading...|n";
895         else
896             cout << string << "...|n";
897         break;
898     case 2:             //stage 3 of loading symbol
899         if (showLoading)
900             cout << "Loading...^n";
901         else
902             cout << string << "...^n";
903         break;
904     case 3:             //stage 4 of loading symbol
905         if (showLoading)
906             cout << "Loading...-n";
907         else
908             cout << string << "...-n";
909         loadingAnim = -1;
910         break;
911     }
912     ++loadingAnim;
913     ++iterations;
914     delay(250);         //delay of a quarter of a second to mimic 4FPS animation
915 }
916 clrscr();
917 }
918
919

```

# OUTPUT



Welcome to Generic Airline Booking!

Choose an option to continue

1 : Login  
2 : Create account  
3 : Continue as guest  
0 : Quit

Opening login page  
Loading.../

Login

username : aamin  
password : \*\*\*\*\*

Logging in.../

Logged in successfully!

Choose an option to continue

1 : Book flight  
2 : View bookings  
3 : PNR Enquiry  
0 : Logout and Quit

Opening book flights page  
Loading...\

Book a flight!

Enter source : RPR  
Enter destination : DEL

S.No	FlightNumber	Source	Destination	Departure	Arrival
1	9W910	RPR	DEL	08:45	10:05
2	AI469	RPR	DEL	08:45	10:05
3	6E341	RPR	DEL	12:30	14:25
4	6E342	RPR	DEL	12:30	14:25
5	6E343	RPR	DEL	13:30	15:25
6	UK420	RPR	DEL	20:40	22:20
7	9W677	RPR	DEL	20:40	22:25

Select S.no to book flight (0 to quit) : 2

Booking on AI469

Enter number of passengers (Maximum 10) : 2

Enter name of any 1 passenger : Em Schmidt

Your PNR : N4TVKM

Please go to your nearest airport and pay against this PNR

Choose an option to continue

1 : Book flight  
2 : View bookings  
3 : PNR Enquiry  
0 : Logout and Quit

Opening view bookings page  
Loading.../

PNR	FlightNumber	Source	Destination	Departure	Arrival	Passengers	Name	Username
J808E7	6E341	RPR	DEL	12:30	14:25	4	Aamin Gem	aamin
R86UYV	UK420	RPR	DEL	20:40	22:20	5	Kumar	aamin
N4TVKM	AI469	RPR	DEL	08:45	10:05	2	Em Schmidt	aamin

Press 1 to go back

Returning to menu...|

Choose an option to continue

1 : Book flight  
2 : View bookings  
3 : PNR Enquiry  
0 : Logout and Quit

Welcome to Generic Airline Booking!

Choose an option to continue

-----  
 1 : Login  
 2 : Create account  
 3 : Continue as guest  
 0 : Quit

Continuing as guest.../

Choose an option to continue

-----  
 1 : Book flight  
 2 : PNR Enquiry  
 0 : Quit

Opening book flights page  
 Loading...\

Book a flight!

-----  
 Enter source : FRA  
 Enter destination : DEL

S.No	FlightNumber	Source	Destination	Departure	Arrival
1	LH760	FRA	DEL	13:20	01:30

-----  
 Select S.no to book flight (0 to quit) : 1  
 Booking on LH760  
 Enter number of passengers (Maximum 10) : 4  
 Enter name of any 1 passenger : Kim Jong Un  
 Your PNR : C6XUBK  
 Please go to your nearest airport and pay against this PNR

Choose an option to continue

-----  
 1 : Book flight  
 2 : PNR Enquiry  
 0 : Quit

Opening PNR Enquiry page  
 Loading.../

Enter PNR : C6XUBK

FlightNumber	Source	Destination	Departure	Arrival	Passengers	Name
LH760	FRA	DEL	13:20	01:30	4	Kim Jong Un

Press 1 to go back

Returning to menu.../

Choose an option to continue

-----  
 1 : Book flight  
 2 : PNR Enquiry  
 0 : Quit

Opening PNR Enquiry page  
 Loading...-

Enter PNR : N4TVKM

FlightNumber	Source	Destination	Departure	Arrival	Passengers	Name
AI469	RPR	DEL	08:45	10:05	2	Em Schmidt

Press 1 to go back

Returning to menu.../

Choose an option to continue

-----  
 1 : Book flight  
 2 : PNR Enquiry  
 0 : Quit

Welcome to Generic Airline Booking!

Choose an option to continue

1 : Login  
2 : Create account  
3 : Continue as guest  
0 : Quit

Opening login page  
Loading.../

Login

username : admin  
password : \*\*\*\*\*

Logging in...-

Logged in successfully!

Choose an option to continue

1 : Add Flights  
2 : View all PNRs  
3 : View all Flights  
0 : Logout and Quit

Opening view all PNRs page  
Loading.../

PNR	FlightNumber	Source	Destination	Departure	Arrival	Passengers	Name	Username
J808E7	6E341	RPR	DEL	12:30	14:25	4	Aamin Gem	aamin
Y0DH0Q	LH756	FRA	BOM	12:30	01:00	2	Emma Stone	guest
R86UYV	UK420	RPR	DEL	20:40	22:20	5	Kumar	aamin
N4TVKM	AI469	RPR	DEL	08:45	10:05	2	Em Schmidt	aamin
C6XUBK	LH760	FRA	DEL	13:20	01:30	4	Kim Jong Un	guest

Press 1 to go back

Returning to menu...|

Choose an option to continue

1 : Add Flights  
2 : View all PNRs  
3 : View all Flights  
0 : Logout and Quit

Opening add flights page  
Loading.../

Enter FlightNumber : SQ306  
Enter Source : SIN  
Enter destination : LHR  
Enter Departure hours : 01  
Enter Departure minutes : 00  
Enter Arrival hours : 07  
Enter Arrival minutes : 20  
SQ306 successfully added!  
Add another flight? (y/n) :

Returning to menu...\

Choose an option to continue

1 : Add Flights  
2 : View all PNRs  
3 : View all Flights  
0 : Logout and Quit

Opening view all flights page  
Loading...\

S.No	FlightNumber	Source	Destination	Departure	Arrival
1	6E343	RPR	DEL	13:30	15:25
2	9W677	RPR	DEL	20:40	22:25
3	AI469	RPR	DEL	08:45	10:05
4	6E342	RPR	DEL	12:30	14:25
5	UK420	RPR	DEL	20:40	22:20
6	9W910	RPR	DEL	08:45	10:05
7	6E341	RPR	DEL	12:30	14:25
8	LH761	DEL	FRA	03:35	07:35
9	LH760	FRA	DEL	13:20	01:30
10	UK989	DEL	PNQ	03:00	05:00
11	LH757	BOM	FRA	02:55	07:20
12	LH756	FRA	BOM	12:30	01:00
13	SQ306	SIN	LHR	01:00	07:20

Press 1 to go back

Returning to menu...\

Choose an option to continue

-----  
 1 : Add Flights  
 2 : View all PNRs  
 3 : View all Flights  
 0 : Logout and Quit

Welcome to Generic Airline Booking!

Choose an option to continue

-----  
 1 : Login  
 2 : Create account  
 3 : Continue as guest  
 0 : Quit

Creating new account  
 Loading...\

Create New User

-----  
 Select username : newuser  
 Username available!  
 Password : \*\*\*\*\*  
 Reenter password : \*\*\*\*\*

creating user  
 Loading...\

User created, heading to login page  
 Loading...!

Login

-----  
 username : newuser  
 password : \*\*\*\*\*

Logging in...-

Choose an option to continue

-----  
 1 : Book flight  
 2 : View bookings  
 3 : PNR Enquiry  
 0 : Logout and Quit

# BIBLIOGRAPHY

<https://uditkarode.github.io/ucc/>

<https://github.com/uditkarode/ucc>

<https://www.gnu.org/software/gcc/>

<https://github.com/babyygemperor/flight-booking>

<http://www.stroustrup.com/C++.html>

<https://stackoverflow.com>

<https://de.wikibooks.org/wiki/Hauptseite>

<https://code.visualstudio.com/>

<https://www.jetbrains.com/clion/>