Name: SHAIK SADIK
Reg. No: CL202409050182653
Batche NO: EMBVLRSR 01
Project Title: Flight Ticket Booking (S D A Aviation)
S/w Used: Online GDB Compiler
Data Structures USED: C Programming, Double Linked List
Trainer Name: S. Basavaraj

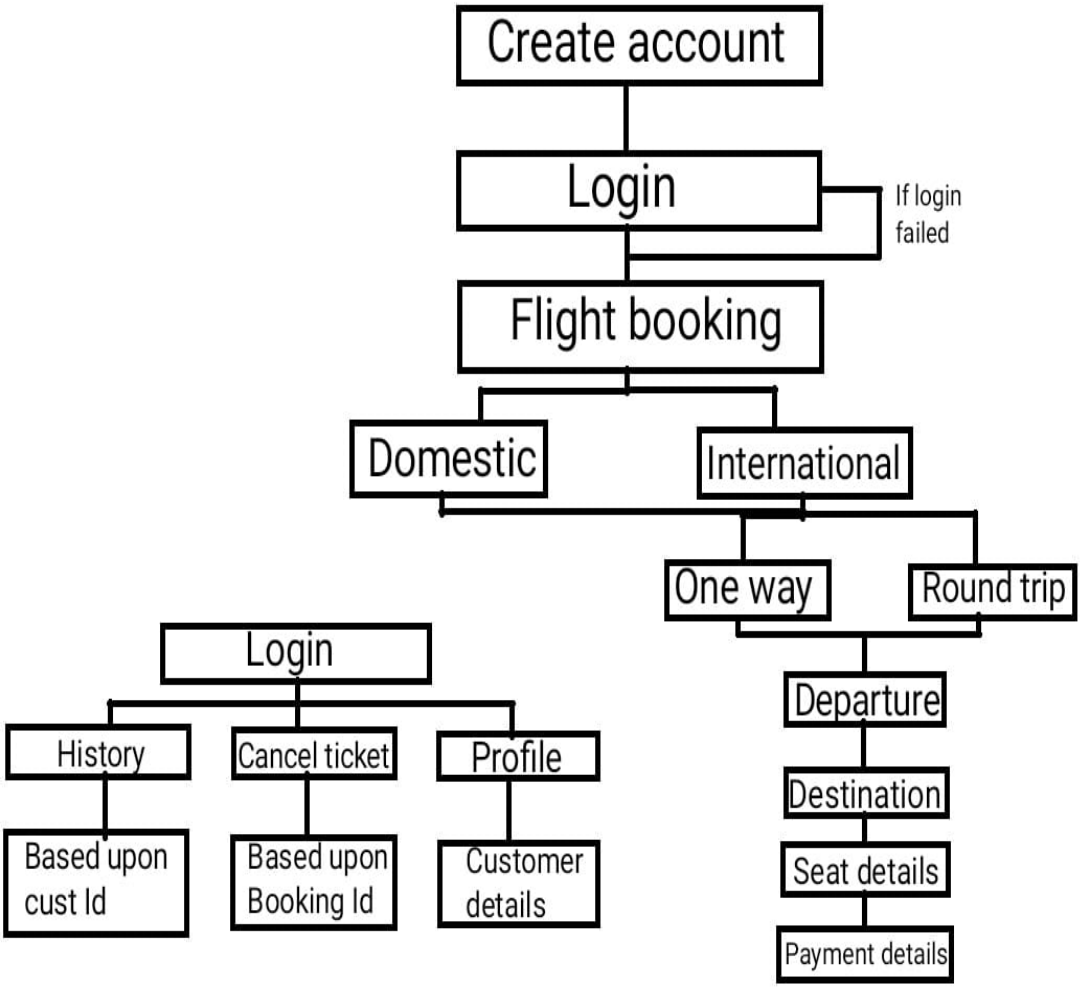
Abstract:

S D A Aviation helps the passenger’s to look up flights between two points which can be domestic or international. The passengers can find and book tickets for flights through this software. Developed in C, it is fairly easy to use software having a user-friendly interface. The users can access flight information and book tickets as per their convenience. The main objective is to reduce the mistakes which creep up in manual systems.it provides good level security so it takes care of the user’s safety concerns as well. passengers can access the whole list of all flights available on different routes with their timings and fare both for economy and business classes. One can compare the best deals for them and book a flight ticket accordingly. When the passengers enter all the details the software helps them to find all available flights and also information if there are seats available on that particular flight. The manual work is thus reduced and the chances of errors are reduced to minimum.

Introduction to Project:

SDA Aviation is a reputed airline offering a seamless and customer-friendly experience when it comes to booking flight tickets. Whether you are planning a domestic getaway or an international adventure, SDA Aviation provides a range of options designed to suit different budgets and preferences. With a user-centric approach, the airline's flight booking system is built to make the process as simple, efficient, and transparent as possible.

Customers can book tickets through multiple channels, including their official website, mobile apps, travel agencies, or directly via customer service. The online booking portal offers an intuitive interface where travelers can quickly search for available flights, compare prices, choose seats, and complete the payment process.


Create account
Login
If login failed
Flight booking
Domestic
International
One way
Round trip
Departure
Destination
Seat details
Payment details
Login
History
Cancel ticket
Profile
Based upon cust Id
Based upon Booking Id
Customer details

```
/*************************FLIGHT TICKET BOOKING
APPLICATION*******************************/


/* Creating Flight ticket booking application using c with functions to
perform operations like register ,login,booking view ticket,cancel ticket
,view history ,profile .
using double liked lit ,switch statements,if else conditions,loops etc..*/
//header files
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <time.h>
// structure for Customer details
typedef struct
{
      char cust_name[100];
      long int customer_id;
      char cust_mailid[200];
      char password[200];
      char address[100];
      char city[100];
      long int pin_code;
      char country[100];
      long int mno;
      char saved_card[20];
} Customer;
// structure for Booking details details
typedef struct
{
      int bookingID;
      char flight_name[100];
      char flight_code[10];
```

```
	char date[11];
	char dep_country[100];
	char dest_country[100];
	float cost;
	int adult;
	int child;
	int ages[20];
	char adultname[10][100];
	char childname[10][100];
	char classes_of_service[30];
	int seat_numbers[20];
	int seat_count;
	int is_round_trip;
	int num_adultss;
	int num_childrens;
	float ccost;
	float acost;
	int cc;
	int ac;

} Booking;

//Customer Node defining
typedef struct CustomerNode
{
	Customer customer;
	struct CustomerNode *next;
} CustomerNode;
//Booking Node defining
typedef struct BookingNode
{
	Booking booking;
```

```
    struct BookingNode *next;
} BookingNode;

CustomerNode *customer_head = NULL;
BookingNode *booking_head = NULL;
CustomerNode *current_customer = NULL;
int is_seat_available[141] = {0};

// Functions Declaration
void display_seat_selection(int seat_class, int *seats, int *seat_count);
void collect_payment_info(char *saved_card);
void book_flight_ticket();
void display_ticket();
void create_account();
void login();
void cancel_ticket();
void travel_history();
void display_customer_details();
int validate_pin(char *pin);
int validate_password(char *password);



// Array for destinations, airlines, codes, and costs
const char *domestic_locations[] = {"Delhi", "Mumbai", "Bengaluru",
"Kolkata", "Chennai", "Hyderabad", "Jaipur", "Goa", "Pune", "Lucknow"};
const char *international_locations[] = {"India","New York", "London",
"Dubai", "Singapore", "Paris", "Tokyo", "Sydney", "Toronto", "Bangkok",
"Hong Kong"};
const char *airlines[] = {"Air India", "Indigo", "SpiceJet", "Emirates",
"Lufthansa", "Qatar Airways", "British Airways", "Air France", "Singapore
Airlines", "Turkish Airlines"};
const char *airline_codes[] = {"AI", "IG", "SJ", "EK", "LH", "QR", "BA",
"AF", "SQ", "TK"};
```

```
// Separate costs for domestic and international flights

float domestic_adult_costs[] = {5000, 4500, 4000, 6000, 5500, 5000, 4500,
4800, 4200, 4600};

float domestic_child_costs[] = {2500, 2250, 2000, 3000, 2750, 2500, 2250,
2400, 2100, 2300};

float international_adult_costs[] = {10000, 9000, 8000, 15000, 14000,
13000, 12000, 12500, 11000, 11500};

float international_child_costs[] = {5000, 4500, 4000, 7500, 7000, 6500,
6000, 6250, 5500, 5750};

int fn=-1;

//main function that contain main interface of the application

int main()

{

      int mainchoice;

      srand(time(0));  // Seed random number generation only once

      //printing the logo and title


      printf("-----------------------------------------------------------------
---------------------------------------\n\n\n");

      printf("\t           *           \t\t");

      printf("  ******      ******            *\n");

      printf("\t            *         \t\t");

      printf(" ********    ********          ***\n");

      printf("\t*            *        \t\t");

      printf(" **    **    **     **        ** **\n");

      printf("\t *            *      \t\t");

      printf(" **          **      **      **   ** \n");

      printf("\t  ******************   \t\t");

      printf("  ******     **      **     *********\n");

      printf("\t *            *     \t\t");

      printf("       **    **      **    ***********\n");

      printf("\t*            *       \t\t");

      printf(" **    **    **     **    **         **\n");

      printf("\t            *       \t\t");
```

```
      printf(" ********    ********    **          **\n");
      printf("\t           *       \t\t");
      printf("  ******      ******    **            **\n\n\n");
      printf("\tM   A   K   E          Y   O   U   R       T   R   I   P
E   A   S   Y\t\t\t\n");
      printf("-------------------------------------------------------------
-------------------------------------\n\n");
//switch cases for Different operations
      while (1)
      {
            printf("\n1. Create Account\t 2. Login\t \t \t 3. Book Flight
Ticket\t  4. Display Ticket\n \n\
5. Cancel Ticket\t 6. View Booking History\t 7. Profile \t \t  8. Exit
\n\n");


      printf("_____________________________________________________________
_____________________________________\n\n\n");


            printf("Enter your option: ");
            scanf("%d", &mainchoice);
            getchar();


            switch (mainchoice)
            {
            case 1:
                  create_account();
                  break;
            case 2:
                  login();
                  break;
            case 3:
                  if (current_customer)
                  {
```

```
                        book_flight_ticket();
                }
                else
                {
                        printf("Please log in before booking a flight
ticket.\n");
                }
                break;
            case 4:
                display_ticket();
                break;
            case 5:
                cancel_ticket();
                break;
            case 6:
                travel_history();
                break;
            case 7:
                display_customer_details();
                break;
            case 8:
                exit(0);
            default:
                printf("Enter a valid option.\n");
            }
      }
}

// Function to create an account
void create_account()
{
      system("clear");
```

```
	//Allocation of memory
	CustomerNode *new_customer = (CustomerNode
*)malloc(sizeof(CustomerNode));
	if (!new_customer)
	{
		printf("Memory allocation failed\n");
		return;
	}

	printf("**************************CREATE NEW ACCOUNT
*****************************************************\n\n");
	printf("_____________________________________________________________________
_____________________________________\n\n\n");

	new_customer->customer.customer_id = rand() % 100000;

	printf("Enter your full name: \n");
	getchar();
	scanf("%[^\n]s", new_customer->customer.cust_name);

	printf("Enter your email ID: \n");
	scanf("%s", new_customer->customer.cust_mailid);

	char password[200], confirm_password[200];
	while (1)
	{
		printf("Enter your password (at least 8 characters,with one
uppercase,one lowercase,and one special character): \n");
		scanf("%s", password);

		if (validate_password(password))
		{
			printf("Confirm your password: \n");
```

```
                scanf("%s", confirm_password);

                if (strcmp(password, confirm_password) == 0)
                {
                      strcpy(new_customer->customer.password, password);
                      break;
                }
                else
                {
                      printf("Passwords do not match. Please try
again.\n");
                }
            }
            else
            {
                  printf("Password does not meet the requirements. Please
try again.\n");
            }
      }

      printf("Enter your address: \n");
      getchar();
      scanf("%[^\n]s", new_customer->customer.address);

      printf("Enter your city: \n");
      getchar();
      scanf("%[^\n]s", new_customer->customer.city);

      printf("Enter your PIN code: \n");
      scanf("%ld", &new_customer->customer.pin_code);

      printf("Enter your mobile number: \n");
      scanf("%ld", &new_customer->customer.mno);
```

```
	printf("Enter your country: \n");
	getchar();
	scanf("%[^\n]s", new_customer->customer.country);

	new_customer->next = customer_head;
	customer_head = new_customer;

	printf("Account created successfully!...Your Customer ID is: %ld\n\n\n", new_customer->customer.customer_id);
	printf("_____________________________________________________________________\n\n\n");

}
// Function definition for login page
void login()
{	system("clear");
	char email[200];
	char password[200];

	printf("************************************LOGIN**********************\n\n");
	printf("_____________________________________________________________________\n\n\n\n");

	printf("Enter your email ID: \n");
	scanf("%s", email);

	printf("Enter your password: \n");
	scanf("%s", password);

	CustomerNode *temp = customer_head;
	while (temp != NULL)
```

```
	{
		if (strcmp(temp->customer.cust_mailid, email) == 0 &&
strcmp(temp->customer.password, password) == 0)
		{
			current_customer = temp;
			printf("Login successful. Welcome....");
			return;
		}
		temp = temp->next;

	printf("_____________________________________________________________
_____________________________\n\n\n");


	}


	printf("Login failed. Please check your email and password.\n");
}
//function definition for booking operation
void book_flight_ticket()
{
	system("clear");
	//Allocation of memory
	BookingNode *new_booking = (BookingNode
*)malloc(sizeof(BookingNode));
	if (!new_booking)
	{
		printf("Memory allocation failed\n");
		return;
	}


	printf("************************************TICKET
BOOKING************************************\n\n");
	printf("_____________________________________________________________
_____________________________\n\n\n");
```

```
	printf("Select Flight Type:\n\n1. Domestic\t\t\t2. 
International\n\n");
	int flight_type;
	printf("Enter your choice:\n");
	scanf("%d", &flight_type);
	getchar();

	printf("Select Trip Type:\n\n1. One Way\t\t\t 2. Round Trip\n\n");
	int trip_type;
	printf("Enter your choice:\n");
	scanf("%d", &trip_type);
	getchar();
	new_booking->booking.is_round_trip = (trip_type == 2);


	const char **locations = flight_type == 1 ? domestic_locations : 
international_locations;

	int location_count = 10;
	// selection of departure details

	printf("\n\n\n**********************************************Departure
**********************************************\n\n");
	for (int i = 0; i < location_count; i++)
	{
		if(i==4||i==8||i==12)
		{
			printf("\n\n");
		}
		printf("%d. %s\t\t", i + 1, locations[i]);
	}
	int dep_code, dest_code;
```

```
	printf("\n\nEnter your choice:\n");
	scanf("%d", &dep_code);
	getchar();
		// selection of destination details

	printf("\n\n\n\n****************************************Destination
****************************************\n\n");
	for (int i = 0; i < location_count; i++)
	{
		if (i + 1 != dep_code)
		{
			if(i==4||i==8||i==12)
			{
				printf("\n\n");
			}
			printf("%d. %s\t\t", i + 1, locations[i]);
		}
	}
	printf("\n\nEnter your choice:\n");
	scanf("%d", &dest_code);
	getchar();

	strcpy(new_booking->booking.dep_country, locations[dep_code - 1]);
	strcpy(new_booking->booking.dest_country, locations[dest_code - 1]);
	printf("\nSelected Depatrure: %s\tAnd\t",new_booking-
>booking.dep_country);
	printf("Destination:%s\n\n",new_booking->booking.dest_country);

	// selection of Flights
	printf("\n\n\n****************************Available
Airlines****************************************\n\n");
	for (int i = 0; i < 10; i++)
	{
```

```
            float adult_cost = flight_type == 1 ? domestic_adult_costs[i]
: international_adult_costs[i];

            float child_cost = flight_type == 1 ? domestic_child_costs[i]
: international_child_costs[i];

            printf("%d. %s - %s (Adult: %.2f, Child: %.2f)\n", i + 1,
airlines[i], airline_codes[i], adult_cost, child_cost);

      }

      int airline_choice;

      printf("\n\nEnter your choice:\n");

      scanf("%d", &airline_choice);

      getchar();

      strcpy(new_booking->booking.flight_name, airlines[airline_choice -
1]);

      strcpy(new_booking->booking.flight_code,
airline_codes[airline_choice - 1]);

      fn=airline_choice;


      printf("\n\nSelect Seat Class:\n\n1. First Class (50%%
premium)\t\t\t2. Premium Economy (20%% premium)\t\t\t3. Economy\n\n");

      int seat_class;

      printf("Enter your choice:\n");

      scanf("%d", &seat_class);

      getchar();

      float class_multiplier = (seat_class == 1) ? 1.5 : (seat_class == 2)
? 1.2 : 1.0;

      strcpy(new_booking->booking.classes_of_service, seat_class == 1 ?
"First Class" : seat_class == 2 ? "Premium Economy" : "Economy");

      printf("\nSelected class is:  %s\n\n",new_booking-
>booking.classes_of_service);




      // Generating the dates of flights


      srand(time(NULL));
```

```
	struct tm start_date = {0};
	start_date.tm_year = 2024-1900;
	start_date.tm_mon = 10;
	start_date.tm_mday = 14;
	time_t current_time = mktime(&start_date);
	printf("\n\n\n*****************************DEPARTURE
DATE*********************************\n\n");
	printf("The Flight you chooseare Avaliable in these Dates\n\n");
	for (int i = 0; i < 5; ++i)
	{
		int random_gap = 1 + rand() % 3;
		current_time += random_gap * 24 * 60 * 60;
		struct tm *new_date = localtime(&current_time);
		printf("%02d-%02d-%d\t",new_date->tm_mday,new_date->tm_mon +
1,new_date->tm_year + 1900 );
	}
// operatin to take Departure date
	printf("\n\n\nEnter Departure Date (DD-MM-YYYY): ");
	fgets(new_booking->booking.date, sizeof(new_booking->booking.date),
stdin);
	new_booking->booking.date[strcspn(new_booking->booking.date, "\n")]
= 0;
	printf("\n\n\n");
	// Seat selection


	int seat_count;
	int seats[20];
	display_seat_selection(seat_class, seats, &seat_count);
	for (int i = 0; i < seat_count; i++)
	{
		new_booking->booking.seat_numbers[i] = seats[i];
	}
	new_booking->booking.seat_count = seat_count;
```

```
	int num_adults,num_children;
	printf("Enter number of adults: ");
	scanf("%d", &num_adults);
	printf("Enter number of children: ");
	scanf("%d", &num_children);
	new_booking->booking.num_adultss =num_adults;
	new_booking->booking.num_childrens =num_children;
	//for loop for taking the details of adults and children
	for (int i = 0; i < num_adults; i++)
	{
		printf("Enter adult name %d: ", i + 1);
		scanf("%s", new_booking->booking.adultname[i]);
		printf("Enter age for %s: ", new_booking-
>booking.adultname[i]);
		scanf("%d", &new_booking->booking.ages[i]);
	}

	for (int i = 0; i < num_children; i++)
	{
		printf("Enter child name %d: ", i + 1);
		scanf("%s", new_booking->booking.childname[i]);
		printf("Enter age for %s: ", new_booking-
>booking.childname[i]);
		scanf("%d", &new_booking->booking.ages[num_adults + i]);
	}
	// calculation of the cost

	float base_adult_cost = flight_type == 1 ?
domestic_adult_costs[airline_choice - 1] :
international_adult_costs[airline_choice - 1];
	float base_child_cost = flight_type == 1 ?
domestic_child_costs[airline_choice - 1] :
international_child_costs[airline_choice - 1];
```

```
      float adult_cost_total = base_adult_cost * num_adults *
class_multiplier;

      float child_cost_total = base_child_cost * num_children *
class_multiplier;

      float round_trip_multiplier = new_booking->booking.is_round_trip ?
2.0 : 1.0;

      float total_cost = (adult_cost_total + child_cost_total) *
round_trip_multiplier;

      new_booking->booking.ac=num_adults;

      new_booking->booking.cc=num_children;

      new_booking->booking.acost=adult_cost_total;

      new_booking->booking.ccost=child_cost_total;

      new_booking->booking.cost = total_cost;


//printing the cost breakdown

      printf("\n\n\n**********************************Cost
Breakdown*********************************************\n");

      printf("Base Adult Cost per person: %.2f\n", base_adult_cost);

      printf("Base Child Cost per person: %.2f\n", base_child_cost);

      printf("Seat Class Multiplier: %.2f\n", class_multiplier);

      printf("Adult Total Cost: %.2f\n", adult_cost_total);

      printf("Child Total Cost: %.2f\n", child_cost_total);

      if (new_booking->booking.is_round_trip)

      {

            printf("Round Trip Multiplier: %.2f\n",
round_trip_multiplier);

      }

      printf("Total Cost: %.2f\n", total_cost);


      // Generate random booking ID

      new_booking->booking.bookingID = rand() % 100000;

      printf("Booking ID: %d\n", new_booking->booking.bookingID);
```

```
      collect_payment_info(current_customer->customer.saved_card);
      printf("Ticket booked successfully. Details sent to registered
mobile number.\n");

      new_booking->next = booking_head;
      booking_head = new_booking;
}

//function definition for the seat  displaying selection
void display_seat_selection(int seat_class, int *seats, int *seat_count  )
{
    //switch cases and conditions to make the  seats quantity for
different flight
      int start_seat, end_seat;
      switch(seat_class)
      {
      case 1:
            if(fn==1)
            {
                  start_seat = 1;
                  end_seat = 8;
            }
            else if(fn==2)
            {
                  start_seat = 1;
                  end_seat = 8;
            }
            else if(fn==3)
            {
                  start_seat = 1;
                  end_seat = 20;
            }
            else if(fn==4)
```

```
            {
                  start_seat = 1;
                  end_seat = 30;
            }
            else if(fn==5)
            {
                  start_seat = 1;
                  end_seat = 40;
            }
            else if(fn==6)
            {
                  start_seat = 1;
                  end_seat = 25;
            }
            else if(fn==7)
            {
                  start_seat = 1;
                  end_seat = 35;
            }
            else if(fn==8)
            {
                  start_seat = 1;
                  end_seat = 25;
            }
            else
            {
                  start_seat = 1;
                  end_seat = 20;
            }
            break;
      case 2:
            if(fn==1)
```

```
{
      start_seat = 11;
      end_seat = 50;
}
else if(fn==2)
{
      start_seat = 11;
      end_seat = 40;
}
else if(fn==3)
{
      start_seat = 21;
      end_seat = 60;
}
else if(fn==4)
{
      start_seat = 31;
      end_seat = 70;
}
else if(fn==5)
{
      start_seat = 41;
      end_seat = 70;
}
else if(fn==6)
{
      start_seat = 26;
      end_seat = 70;
}
else if(fn==7)
{
      start_seat = 36;
```

```
                end_seat = 70;
            }
            else if(fn==8)
            {
                start_seat = 26;
                end_seat = 60;
            }
            else
            {
                start_seat = 21;
                end_seat = 50;
            }
            break;
        case 3:
            if(fn==1)
            {
                start_seat = 61;
                end_seat = 120;
            }
            else if(fn==2)
            {
                start_seat = 41;
                end_seat = 100;
            }
            else if(fn==3)
            {
                start_seat = 61;
                end_seat = 120;
            }
            else if(fn==4)
            {
                start_seat = 71;
```

```
                end_seat = 140;
        }
        else if(fn==5)
        {
                start_seat = 71;
                end_seat = 140;
        }
        else if(fn==6)
        {
                start_seat = 71;
                end_seat = 140;
        }
        else if(fn==7)
        {
                start_seat = 71;
                end_seat = 140;
        }
        else if(fn==8)
        {
                start_seat = 61;
                end_seat = 110;
        }
        else
        {
                start_seat = 51;
                end_seat = 110;
        }
        break;
default:
        return;
}
system("clear");
```

```
	// desplaying the avaliable seats
	printf("****************************************Available Seats*********************************************\n\n");
	printf("A\t\tB\t\tD\t\tE\t\tF\n\n");
	for (int i = start_seat; i <= end_seat; i++)
	{
		if (!is_seat_available[i])
		{
			printf("S-%d\t\t", i);


	if(i==5||i==10||i==15||i==20||i==25||i==30||i==35||i==40||i==45||i==50||i==55||i==60||i==65||i==70||i==75||i==80||i==85||i==90||i==95||i==100||i==105||i==110||i==115||i==120||i==125||i==130||i==135)
			{
				printf("\n\n");


			}
		}
	}


	printf("\nEnter the number of seats to book: ");
	scanf("%d", seat_count);


	for (int i = 0; i < *seat_count; i++)
	{
		int seat_no;
		printf("Select seat number %d: ", i + 1);
		scanf("%d", &seat_no);
		if (seat_no >= start_seat && seat_no <= end_seat && !is_seat_available[seat_no])
		{
			is_seat_available[seat_no] = 1;
			seats[i] = seat_no;
```

```
            }
            else
            {
                  printf("Invalid or already booked seat.\n");
                  i--;  // Prompt again for a valid seat number
            }
      }
}
//function definition for the payment process

void collect_payment_info(char *saved_card)
{
      printf("\n\n**************************************P A Y M E N T
S**************************************************\n\n");
      char card_number[20], card_holder[50], pin[5];
      printf("Enter Card Number (16 digits): ");
      scanf("%s", card_number);

      printf("Enter Card Holder Name: ");
      getchar();
      fgets(card_holder, 50, stdin);
      card_holder[strcspn(card_holder, "\n")] = 0;

      while (1)
      {
            printf("Enter 4-digit Card PIN: ");
            scanf("%s", pin);
            if (validate_pin(pin))
            {
                  printf("Payment confirmed.\n");
                  break;
            }
```

```
            else
            {
                  printf("Invalid PIN. Please try again.\n");
            }
      }
}
//function definition for displaying the ticket
void display_ticket()
{
      int booking_id;
      printf("Enter Booking ID to display ticket: ");
      scanf("%d", &booking_id);
      getchar();
      system("clear");
      BookingNode *temp = booking_head;
      int gate=temp->booking.bookingID;
      int acount=temp->booking.ac;
      int ccount=temp->booking.cc;
      if (gate < 0) {
            gate = -gate;
      }



      while (gate >= 10) {
            gate /= 10;
      }


      while (temp != NULL)
      {
            if (temp->booking.bookingID == booking_id)
            {
                  if(temp->booking.num_adultss!=0)
```

```
                {
                        for(int i=0; i<temp->booking.num_adultss; i++)

                        {
                                printf("ADULT TICKET NO:%d\n\n",i+1);

        printf("_____________________________________________________________
_____________________\n\n\n");
                                printf("\t      *   \t\t");
                                printf(" ****    ***          *\n");
                                printf("\t*       *  \t\t");
                                printf("*        *   *       *  *\n");
                                printf("\t **********\t\t");
                                printf(" ****    *    *     ******\n");
                                printf("\t*       * \t\t");
                                printf("     *   *   *     *      * \n");
                                printf("\t      *   \t\t");
                                printf(" ****    ***      *        *\n");

        printf("_____________________________________________________________
_____________________\n\n\n");
                                printf("\t\t\t\t B O A R D I N G   P A S S
\n\n\n");


                                printf("\tBOARDING PASS ID:%d\t\t\t\t",
temp->booking.bookingID);
                                printf("|| ||||| ||||||||  |||
|||||||\n\n");
                                printf("\tName of the
passenger\n\n\t%s\t\t\t\t\t\t",temp->booking.adultname[i]);
                                printf("ADULT PASS\n\n\n");
                                printf("\tFrom\t\t  Flight\t\t\tDate\n");
                                printf("\t%s\t\t", temp-
>booking.dep_country);
```

```
                              printf("   %s\t\t\t",temp-
>booking.flight_code);
                              printf(" %s\n\n", temp->booking.date);
                              printf("\tTo\n\t%s\n\n", temp-
>booking.dest_country);
                              printf("\tClass: %s\t\t\t\t", temp-
>booking.classes_of_service);
                              printf("\tTrip Type: %s\n\n", temp-
>booking.is_round_trip ? "Round Trip" : "One Way");
                              printf("\tSeat\t\tGATE\t");
                              printf("\t\t\tTotal Cost: RS%.2f\n", temp-
>booking.acost/acount);
                              printf("\tS-%d\t\t ", temp-
>booking.seat_numbers[i]);
                              printf("%d\n\n",gate);

      printf("\n_____
_____\n\n");
                              printf("\t\t\tPLEASE BE AT THE GATE AT
BOARDING TIME\n");

      printf("_____
_____\n\n\n");
                        }
                  }
                  printf("\n\n");
                  if(temp->booking.num_childrens!=0)
                  {
                        for(int i=0; i<temp->booking.num_childrens; i++)
                        {
                              printf("CHILD TICKET NO:%d\n\n",i+1);

      printf("_____
_____\n\n\n");
                              printf("\t      *   \t\t");
                              printf(" ****    ***          *\n");
                              printf("\t*       *  \t\t");
```

```
                                printf("*        *   *       *  *\n");
                                printf("\t **********\t\t");
                                printf(" ****    *    *     ******\n");
                                printf("\t*       * \t\t");
                                printf("     *   *   *     *      * \n");
                                printf("\t      *   \t\t");
                                printf(" ****    ***      *        *\n");

      printf("_____________________________________________________________
_____________________\n\n\n");
                                printf("\t\t\t\t B O A R D I N G   P A S S
\n\n\n");


                                printf("\tBOARDING PASS ID:%d\t\t\t\t",
temp->booking.bookingID);
                                printf("|| ||||| ||||||||  |||
|||||||\n\n");
                                printf("\tName of the
passenger\n\n\t%s\t\t\t\t\t\t",temp->booking.childname[i]);
                                printf("CHILD PASS\n\n\n");
                                printf("\tFrom\t\t  Flight\t\t\tDate\n");
                                printf("\t%s\t\t", temp-
>booking.dep_country);
                                printf("   %s\t\t\t",temp-
>booking.flight_code);
                                printf(" %s\n\n", temp->booking.date);
                                printf("\tTo\n\t%s\n\n", temp-
>booking.dest_country);
                                printf("\tClass: %s\t\t\t\t", temp-
>booking.classes_of_service);
                                printf("\tTrip Type: %s\n\n\n", temp-
>booking.is_round_trip ? "Round Trip" : "One Way");
                                printf("\tSeat\t\tGATE\t");
                                printf("\t\t\tTotal Cost: RS%.2f\n", temp-
>booking.ccost/ccount);
                                printf("\tS-%d\t\t ", temp-
>booking.seat_numbers[i]);
```

```
                              printf("%d\n\n",gate);

      printf("\n_____________________________________________________________
_____________________________\n\n");
                              printf("\t\t\tPLEASE BE AT THE GATE AT
BOARDING TIME\n");

      printf("_____________________________________________________________
_____________________________\n\n\n");
                        }
                  }

                  printf("\n\n");
                  return;
            }
            temp = temp->next;
      }
      printf("Booking ID not found.\n");
      printf("_____________________________________________________________
_____________________________\n\n\n");

}
// function definition for cancellation of ticket
void cancel_ticket()
{
      int booking_id;

      printf("************************************TICKET
CANCELLATION************************************\n\n");
      printf("_____________________________________________________________
_____________________________\n\n\n");

      printf("Enter Booking ID to cancel: ");
      scanf("%d", &booking_id);
      getchar();
```

```
	system("clear");

	BookingNode *temp = booking_head, *prev = NULL;
	while (temp != NULL && temp->booking.bookingID != booking_id)
	{
		prev = temp;
		temp = temp->next;
	}

	if (temp == NULL)
	{
		printf("Booking ID not found.\n");
		return;
	}

	printf("Confirm cancellation (1 for Yes, 0 for No): ");
	int confirm;
	scanf("%d", &confirm);
	getchar();

	if (confirm == 1)
	{
		if (prev == NULL)
			booking_head = temp->next;
		else
			prev->next = temp->next;

		for (int i = 0; i < temp->booking.seat_count; i++)
		{
			is_seat_available[temp->booking.seat_numbers[i]] = 0;
		}
		free(temp);
```

```
            printf("Ticket with Booking ID %d cancelled successfully.\n",
booking_id);
      }
      else
      {
            printf("Cancellation aborted.\n");
      }
}
// function definition for displaying the Customer details
void display_customer_details()
{
      CustomerNode *temp = customer_head;
      system("clear");
      while (temp != NULL)
      {

      printf("************************************PROFILE***************
********************\n\n");

      printf("_____________________________________________________________
_____________________\n\n\n");
            printf("Customer ID is: %ld\n", temp->customer.customer_id);
            printf("Name: %s\n", temp->customer.cust_name);
            printf("Email ID: %s\n", temp->customer.cust_mailid);
            printf("Address: %s\n", temp->customer.address);
            printf("City: %s\n", temp->customer.city);
            printf("PIN Code: %ld\n", temp->customer.pin_code);
            printf("Country: %s\n", temp->customer.country);
            printf("Mobile Number: %ld\n", temp->customer.mno);
            return;
            temp = temp->next;
      }
}
```

```
//function definition for viewin the travel history
void travel_history()
{
	if (!current_customer)
	{
		printf("Please log in to view travel history.\n");
		return;
	}

	long int entered_customer_id;
	printf("Enter Customer ID to view travel history: ");
	scanf("%ld", &entered_customer_id);
	getchar();
	system("clear");

	// Check if the entered customer ID matches the current logged-in customer's ID
	if (entered_customer_id != current_customer->customer.customer_id)
	{
		printf("Customer ID does not match the logged-in account.\n");
		return;
	}

	BookingNode *temp = booking_head;
	int found = 0;

	printf("Travel History for Customer ID: %ld\n", entered_customer_id);
	while (temp != NULL)
	{
		if (current_customer->customer.customer_id == entered_customer_id)
```

```
            {
                  found = 1;
                  printf("Booking ID: %d\n", temp->booking.bookingID);
                  printf("Flight Name: %s (%s)\n", temp-
>booking.flight_name, temp->booking.flight_code);
                  printf("Departure Date: %s\n", temp->booking.date);
                  printf("Class of Service: %s\n", temp-
>booking.classes_of_service);
                  printf("Total Cost: %.2f\n", temp->booking.cost);
                  printf("-------------------------------------\n");
            }
            temp = temp->next;
      }

      if (!found)
      {
            printf("No travel history found for Customer ID %ld.\n",
entered_customer_id);
      }
}



// function definition for validating the password
int validate_password(char *password)
{
      int has_upper = 0, has_lower = 0, has_special = 0, length =
strlen(password);

      if (length < 8) return 0;

      for (int i = 0; i < length; i++)
      {
            if (isupper(password[i])) has_upper = 1;
```

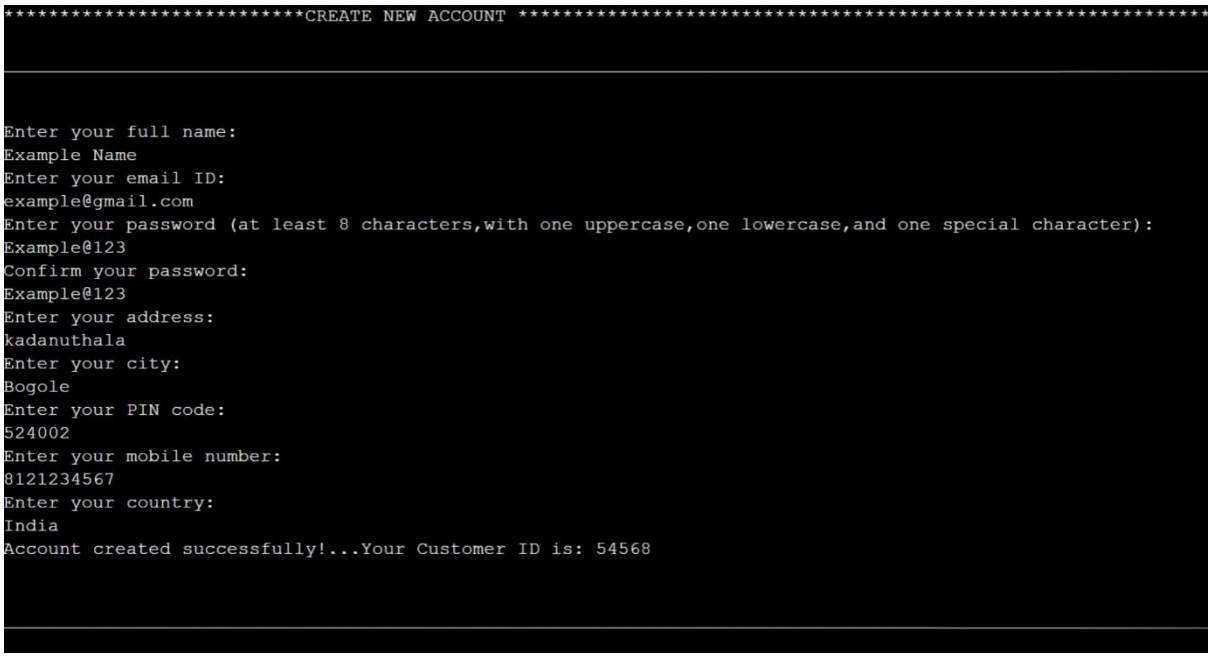
```
            else if (islower(password[i])) has_lower = 1;
            else if (ispunct(password[i])) has_special = 1;
      }

      return has_upper && has_lower && has_special;
}

// function definition for validating the PIN
int validate_pin(char *pin)
{
      if (strlen(pin) != 4) return 0;
      for (int i = 0; i < 4; i++)
      {
            if (pin[i] < '0' || pin[i] > '9')
                  return 0;
      }
      return 1;
}
```

Outputs:


M A K E Y O U R T R I P E A S Y
1. Create Account
2. Login
3. Book Flight Ticket
4. Display Ticket
5. Cancel Ticket
6. View Booking History
7. Profile
8. Exit
Enter your option:


**********CREATE NEW ACCOUNT **********
Enter your full name:
Example Name
Enter your email ID:
example@gmail.com
Enter your password (at least 8 characters,with one uppercase,one lowercase,and one special character):
Example@123
Confirm your password:
Example@123
Enter your address:
kadanuthala
Enter your city:
Bogole
Enter your PIN code:
524002
Enter your mobile number:
8121234567
Enter your country:
India
Account created successfully!...Your Customer ID is: 54568

```
****************************************TICKET BOOKING*****************************************


_____________________________________________________________________________________________



Select Flight Type:

1. Domestic                    2. International

Enter your choice:
1
Select Trip Type:

1. One Way                      2. Round Trip

Enter your choice:
1
```

```
***********************************Departure*****************************************

1. Delhi                2. Mumbai               3. Bengaluru            4. Kolkata

5. Chennai              6. Hyderabad            7. Jaipur               8. Goa

9. Pune         10. Lucknow

Enter your choice:
6




***************************************Destination*****************************************

1. Delhi                2. Mumbai               3. Bengaluru            4. Kolkata

5. Chennai              7. Jaipur               8. Goa

9. Pune         10. Lucknow

Enter your choice:
1

Selected Depatrure: Hyderabad   And     Destination:Delhi
```

```
**********************************DEPARTURE DATE*****************************************

The Flight you chooseare Avaliable in these Dates

16-11-2024      19-11-2024      22-11-2024      24-11-2024      25-11-2024


Enter Departure Date (DD-MM-YYYY): 24-11-2024
```

```
**************************************Available Seats*************************************

A               B               D               E               F

S-1             S-2             S-3             S-4             S-5

S-6             S-7             S-8
Enter the number of seats to book: 2
Select seat number 1: 1
Select seat number 2: 4
Enter number of adults: 1
Enter number of children: 1
Enter adult name 1: Example1
Enter age for Example1: 21
Enter child name 1: Example2
Enter age for Example2: 12
```

```
**********************************Cost Breakdown**************************************
Base Adult Cost per person: 5000.00
Base Child Cost per person: 2500.00
Seat Class Multiplier: 1.50
Adult Total Cost: 7500.00
Child Total Cost: 3750.00
Total Cost: 11250.00
Booking ID: 31957
```

```
************************************P A Y M E N T S*************************************

Enter Card Number (16 digits): 1234123412341234
Enter Card Holder Name: Example
Enter 4-digit Card PIN: 1234
Payment confirmed.
Ticket booked successfully. Details sent to registered mobile number.

1. Create Account       2. Login                    3. Book Flight Ticket    4. Display Ticket

5. Cancel Ticket        6. View Booking History     7. Profile               8. Exit

_____________________________________________________________________________________
```


ADULT TICKET NO:1
B O A R D I N G P A S S
BOARDING PASS ID:31957
Name of the passenger
Example1
ADULT PASS
From
Hyderabad
Flight
AI
Date
24-11-2024
To
Delhi
Class: First Class
Trip Type: One Way
Seat
S-1
GATE
3
Total Cost: RS7500.00
PLEASE BE AT THE GATE AT BOARDING TIME


CHILD TICKET NO:1
B O A R D I N G P A S S
BOARDING PASS ID:31957
Name of the passenger
Example2
CHILD PASS
From
Hyderabad
Flight
AI
Date
24-11-2024
To
Delhi
Class: First Class
Trip Type: One Way
Seat
S-1
GATE
3
Total Cost: RS3750.00
PLEASE BE AT THE GATE AT BOARDING TIME

```
**************************************PROFILE**************************************


Customer ID is: 51065
Name: xample Name
Email ID: example@gmail.com
Address: Kadanuthala kavali
City: Kavali
PIN Code: 524201
Country: India
Mobile Number: 8123456789
```

```
*********************************************TICKET CANCELLATION*********************************************


Enter Booking ID to cancel: 78204


Confirm cancellation (1 for Yes, 0 for No): 1
Ticket with Booking ID 78204 cancelled successfully.

1. Create Account       2. Login                    3. Book Flight Ticket    4. Display Ticket

5. Cancel Ticket        6. View Booking History     7. Profile               8. Exit
```

```
******************************************** Travel History ********************************************


  Enter Customer ID: 51065

  Travel History for Customer ID: 51065
  Booking ID: 31957
  Flight Name: Air India (AI)
  Departure Date: 24-11-2024
  Class of Service: First Class
  Total Cost: 11250.00
```


**********LOGIN**********
Enter your email ID:
example@gmail.com
Enter your password:
Example@123
Login successful. Welcome....
1. Create Account
2. Login
3. Book Flight Ticket
4. Display Ticket
5. Cancel Ticket
6. View Booking History
7. Profile
8. Exit
Enter your option:

Conclusion:

The flight ticket booking application is one of the greatest improvements ever made to make the traveling planning process easy. Using user-friendly interfaces, secure payment systems, and real-time flight information, the application helps in making a strong customer convenience experience. This digital form of air travel not only clears up the confusion surrounding the journey but also creates trust among its users based on transparency and efficiency. With the strength of its functionalities and ease of accessibility, the app has the potential to transform the way travel is contemplated and carried out to be more personalized, efficient, and hassle-free.

Future Upgrades:

1. AI-driven personalization:

Analysis of user preferences through AI algorithms for flight recommendations, offers, and travel opportunities

2. Voice Command Capability:

Introduce voice command operation that lets users search, book, and manage their tickets with voice assistants, ensuring a hands-free experience

3. Multimodal Integration of Travel:

Extend the platform for use to include other modes of transport, such as trains, buses, and car rentals, to give users end-to-end travel planning capabilities.

4. Sustainability Options:

Give features that allow users to filter flights based on carbon emissions, enabling contributions towards environmental initiatives through their booking.

5. Augmented Reality Navigation:

Implement AR for airport navigation in real-time to guide passengers to gates, lounges, and services.

6. Blockchain for Enhanced Security:

Adoptions of Blockchain to ensure safe ticketing and transaction management to maintain transparency with fraud prevention.

7. Loyalty and Reward Schemes:

Gamify loyalty schemes by giving rewards and a discount to users on frequent booking.

8. Multi-Currency and Crypto Payments:

Accept cryptocurrencies and multi-currency transactions for international customers.

9. Real-time Updates through Push:

Allow real-time updates such as flight status changes, gate assignments, weather conditions as push notifications.

10. Offline Capability:

To enable users to access their bookings and itineraries even without an internet connection, develop offline features.

The solutions mentioned above will not only enhance the user experience but also position the application as the cutting-edge solution in the competitive travel industry.