5.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_BOOKS 5
typedef struct {
 int id;
 char title[100];
 char author[100];
 int is_issued;
} Book;
Book books[MAX_BOOKS];
int book_count = 0;
void addBook() {
 if (book_count >= MAX_BOOKS) {
 printf("Library is full, cannot add more books.\n");
 return;
 }
 Book book;
 book.id = book_count + 1;
 printf("Enter book title: ");
 getchar(); // To consume the newline character left by previous input
 fgets(book.title, sizeof(book.title), stdin);
 book.title[strcspn(book.title, "\n")] = 0; // Remove trailing newline
 printf("Enter book author: ");
 fgets(book.author, sizeof(book.author), stdin);
 book.author[strcspn(book.author, "\n")] = 0; // Remove trailing newline
```



```
 book.is_issued = 0;
 books[book_count++] = book;
 printf("Book added successfully!\n");
}
void displayBooks() {
 if (book_count == 0) {
 printf("No books available in the library.\n");
 return;
 }
 printf("ID\tTitle\t\t\tAuthor\t\t\tStatus\n");
 for (int i = 0; i < book_count; i++) {
 printf("%d\t%-20s\t%-20s\t%s\n", books[i].id, books[i].title, books[i].author,
books[i].is_issued ? "Issued" : "Available");
 }
}
void issueBook() {
 int book_id;
 printf("Enter book ID to issue: ");
 scanf("%d", &book_id);
 if (book_id < 1 || book_id > book_count || books[book_id - 1].is_issued) {
 printf("Invalid book ID or book already issued.\n");
 return;
 }
 books[book_id - 1].is_issued = 1;
 printf("Book issued successfully!\n");
```

```
}
void returnBook() {
 int book_id;
 printf("Enter book ID to return: ");
 scanf("%d", &book_id);
 if (book_id < 1 || book_id > book_count || !books[book_id - 1].is_issued) {
 printf("Invalid book ID or book was not issued.\n");
 return;
 }
 books[book_id - 1].is_issued = 0;
 printf("Book returned successfully!\n");
}
```


```
int main() {
 int choice;
 while (1) {
 printf("\nLibrary Management System\n");
 printf("1. Add Book\n");
 printf("2. Display Books\n");
 printf("3. Issue Book\n");
 printf("4. Return Book\n");
 printf("5. Exit\n");
 printf("Enter your choice: ");
 scanf("%d", &choice);
 switch (choice) {
 case 1:
 addBook();
 break;
 case 2:
 displayBooks();
 break;
 case 3:
 issueBook();
 break;
 case 4:
 returnBook();
 break;
 case 5:
 exit(0);
 default:
 printf("Invalid choice! Please try again.\n");
 }
 }
 return 0;
}
```
Example Output:

```
Library Management System
1. Add Book
2. Display Books
3. Issue Book
4. Return Book
5. Exit
Enter your choice: 1
```

Enter book title: The C Programming Language
Enter book author: Brian W. Kernighan and Dennis M. Ritchie
Book added successfully!
Library Management System
1. Add Book
2. Display Books
3. Issue Book
4. Return Book
5. Exit
Enter your choice: 1
Enter book title: Clean Code
Enter book author: Robert C. Martin
Book added successfully!

6.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 100
// Structures to store user and booking information
typedef struct {
char name[50];
int age;
char gender[10];
char phone[15];
} Passenger;
typedef struct {
int ticketNumber;
Passenger passenger;
char from[50];
char to[50];
char date[15];
char trainNumber[10];
char seatClass[20];
float price;
} Booking;
Booking bookings[MAX];
```



```
int bookingCount = 0;
void registerPassenger(Passenger *passenger) {
printf("Enter Name: ");
scanf("%s", passenger->name);
printf("Enter Age: ");
scanf("%d", &passenger->age);
printf("Enter Gender: ");
scanf("%s", passenger->gender);
printf("Enter Phone Number: ");
scanf("%s", passenger->phone);
}
void bookTicket() {
```

```
if (bookingCount >= MAX) {
printf("No more bookings can be made.\n");
return;
}
Booking newBooking;
newBooking.ticketNumber = bookingCount + 1;
printf("Enter journey details:\n");
registerPassenger(&newBooking.passenger);
printf("From: ");
scanf("%s", newBooking.from);
printf("To: ");
scanf("%s", newBooking.to);
printf("Date of Journey (dd-mm-yyyy): ");
scanf("%s", newBooking.date);
printf("Train Number: ");
scanf("%s", newBooking.trainNumber);
printf("Class (Economy/Sleeper/First Class): ");
scanf("%s", newBooking.seatClass);
newBooking.price = 100.0; // Simple pricing for demonstration
bookings[bookingCount++] = newBooking;
printf("Ticket booked successfully! Your ticket number is %d.\n", newBooking.ticketNumber);
}
void cancelTicket() {
int ticketNumber;
printf("Enter ticket number to cancel: ");
scanf("%d", &ticketNumber);
int found = 0;
for (int i = 0; i < bookingCount; i++) {
if (bookings[i].ticketNumber == ticketNumber) {
found = 1;
for (int j = i; j < bookingCount - 1; j++) {
bookings[j] = bookings[j + 1];
Copyright by: Dr. Ramesh M. Kagalkar, Professor, ISE, Nagarjuna College of Engineering and
Technology, Bangaluru. 27 | Page}
bookingCount--;
printf("Ticket %d canceled successfully.\n", ticketNumber);
break;
}
}
if (!found) {
printf("Ticket number %d not found.\n", ticketNumber);
}
}
void displayTickets() {
if (bookingCount == 0) {
printf("No bookings available.\n");
return;
}
for (int i = 0; i < bookingCount; i++) {
Booking b = bookings[i];
printf("\nTicket Number: %d\n", b.ticketNumber);
printf("Name: %s\n", b.passenger.name);
printf("Age: %d\n", b.passenger.age);
printf("Gender: %s\n", b.passenger.gender);
printf("Phone: %s\n", b.passenger.phone);
```

```
printf("From: %s\n", b.from);
printf("To: %s\n", b.to);
printf("Date: %s\n", b.date);
printf("Train Number: %s\n", b.trainNumber);
printf("Class: %s\n", b.seatClass);
printf("Price: $%.2f\n", b.price);
}
}
void menu() {
int choice;
while (1) {
printf("\nRailway Reservation System\n");
printf("1. Book Ticket\n");
printf("2. Cancel Ticket\n");
printf("3. Display Booked Tickets\n");
printf("4. Exit\n");
printf("Enter your choice: ");
scanf("%d", &choice);
switch (choice) {
case 1:
bookTicket();
```

```
case 2:
cancelTicket();
break;
case 3:
displayTickets();
break;
case 4:
exit(0);
default:
printf("Invalid choice. Please try again.\n");
}
}
}
int main() {
menu();
return 0;
}
```

7.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAX_USERS 100
#define MAX_USERNAME 50
#define MAX_PASSWORD 50
typedef struct {
char username[MAX_USERNAME];
char password[MAX_PASSWORD];
} User;
User users[MAX_USERS];
int userCount = 0;
```

```
void registerUser() {
if (userCount >= MAX_USERS) {
printf("User limit reached. Cannot register more users.\n");
return;
}
User newUser;
printf("Enter username: ");
scanf("%s", newUser.username);
for (int i = 0; i < userCount; i++) {
if (strcmp(users[i].username, newUser.username) == 0) {
printf("Username already exists. Please try a different username.\n");
```

```
}
}
printf("Enter password: ");
scanf("%s", newUser.password);
users[userCount++] = newUser;
printf("User registered successfully.\n");
}
void loginUser() {
char username[MAX_USERNAME];
char password[MAX_PASSWORD];
printf("Enter username: ");
scanf("%s", username);
printf("Enter password: ");
scanf("%s", password);
for (int i = 0; i < userCount; i++) {
if (strcmp(users[i].username, username) == 0 && strcmp(users[i].password, password) == 0) {
printf("Login successful. Welcome, %s!\n", username);
return;
}
}
printf("Invalid username or password. Please try again.\n");
}
void resetPassword() {
char username[MAX_USERNAME];
printf("Enter username to reset password: ");
scanf("%s", username);
```

```
if (strcmp(users[i].username, username) == 0) {
printf("Enter new password: ");
scanf("%s", users[i].password);
printf("Password reset successfully.\n");
return;
}
}
printf("Username not found. Please register first.\n");
}
int main() {
int choice;
while (1) {
printf("\nE-commerce Platform Login System\n");
printf("1. Register\n");
```

```
printf("2. Login\n");
printf("3. Reset Password\n");
printf("4. Exit\n");
printf("Enter your choice: ");
scanf("%d", &choice);
switch (choice) {
case 1:
registerUser();
break;
case 2:
loginUser();
break;
case 3:
resetPassword();
```

```
case 4:
printf("Exiting...\n");
return 0;
default:
printf("Invalid choice. Please try again.\n");
}
}
}
```