

Railway Reservation System - Code Explanation

Admin Login Function

```
bool login() {...}
```

Handles the admin login process:

- Prompts for username and password.
- Compares input to 'hammad' and '1111'.
- Grants or denies access accordingly.

Ticket Structs

```
struct Ticket {...} - Stores passenger name and seat number.
```

```
struct WaitingNode {...} - Represents a node in the waiting list linked list.
```

```
struct Booking {...} - Stores passenger name and class type for future use.
```

Global Variables

- Arrays economyTickets and businessTickets store booked tickets.
- Boolean arrays mark which seats are booked.
- booking counters keep track of how many are booked.
- waitingListHead points to the head of the linked list for the waiting list.

Book Ticket Function

```
void booktickets() {...}
```

Allows booking a seat in Business or Economy class:

- Validates input.
- Checks if the seat is already booked.
- If full, calls addtowaitinglist().
- Else, books the seat and stores passenger info.

Cancel Ticket Function

```
void canceltickets() {...}
```

Cancels a ticket by seat number:

- Validates if seat is booked.
- Removes booking and shifts tickets.
- Removes first person from waiting list if available.

Display Booked Tickets

Railway Reservation System - Code Explanation

```
void displaybooked() {...}
```

Displays all booked seats in both classes.

- If no tickets, informs the user.

Display Waiting List

```
void displaywaiting() {...}
```

Traverses the linked list and displays each waiting passenger with their class type.

Timings Class

```
class timings {...}
```

- Contains Monday time for Business class.
- Can update Monday's time using getline.
- Displays fixed times for both classes on Monday and Tuesday.

Main Function

```
int main() {...}
```

- Calls login(), and if successful, runs beginsystem().
- Shows a menu with 6 options using a do-while loop.
- Calls appropriate functions based on user choice.

System Initialization

```
void beginsystem() {...}
```

Initializes seat status and resets waiting list pointers.

Add to Waiting List

```
void addtowaitinglist() {...}
```

Prompts for name and class type.

- Adds new node to the end of the waiting list (linked list).