**Introduction to Bus Reservation System in R Programming:**

The Bus Reservation System, developed using the R programming language, represents a sophisticated solution to streamline and simplify the process of reserving and managing bus tickets. This system has been designed to meet the growing demands of modern transportation services, making it easier for passengers to plan their journeys and for bus operators to efficiently manage their fleets.

**Key Features**:

User-Friendly Interface: The Bus Reservation System provides an intuitive and user-friendly interface that allows passengers to easily browse available bus routes, check schedules, and reserve seats. The interface is designed to ensure a seamless booking experience for both novice and experienced users.

1. Real-Time Availability: Passengers can check real-time seat availability, enabling them to make informed decisions when booking their tickets. This feature ensures that passengers can secure their seats on the desired bus without any last-minute surprises.

2. Online Payment: The system supports secure online payment options, making it convenient for passengers to complete their reservations. Payment methods such as credit/debit cards and digital wallets are integrated to facilitate cashless transactions.

3. Seat Selection: Passengers can choose their preferred seats from an interactive bus seating layout, giving them control over their travel experience. This feature is especially useful for passengers traveling in groups who want to sit together.

4. Booking Management: Bus operators can efficiently manage bookings, including cancellations and modifications, through a centralized dashboard. This simplifies the administrative tasks associated with running a bus service.

5. Reporting and Analytics: The system generates comprehensive reports and analytics for bus operators to gain insights into passenger trends, popular routes, and revenue generation. This data-driven approach helps operators make informed decisions for route optimization and service improvements.

6. Security and Privacy: The Bus Reservation System prioritizes the security of passenger data and financial information. Robust encryption protocols and data protection measures are in place to ensure the confidentiality and integrity of user information.

7. Multi-Platform Compatibility: Passengers can access the system from various devices, including smartphones, tablets, and desktop computers, ensuring accessibility for a wide range of users.

The development of this Bus Reservation System in R programming language underscores the flexibility and versatility of R in creating efficient and data-driven applications. R's extensive libraries and packages for data manipulation and visualization have been harnessed to provide a rich user experience and valuable insights for both passengers and bus operators.

**MODULE DESCRIPTION:**

Creating a complete bus reservation system model in R programming would involve several components such as data structures, functions, and user interactions. Here's a simplified overview:

**Data Structures**: Define data structures to represent buses, passengers, and reservations. For example, you could use lists, data frames, or custom classes.

**Functions:** Create functions for adding new buses, displaying available buses, searching for buses based on criteria (e.g., date, destination), and making reservations.

**User Interactions**: Implement a user interface where users can interact with the system. You can use the readline() function to take input from users and display information using print() or other output functions.

**Main Loop:** Set up a main loop that repeatedly presents options to the user and takes appropriate actions based on their choices. This could involve displaying a menu and calling the relevant functions.

Certainly! Here's a brief model description of a bus reservation system implemented in R programming:

**Data Structures:**

Buses: A list of buses, each containing information such as bus number, destination, total seats, and available seats.

Passengers: A list of passengers with their names and reservation details.

Reservations: A list of reservations, linking passengers to specific bus trips.

**Functions:**

add\_bus(bus\_number, destination, total\_seats): Adds a new bus to the system with the provided details and initializes available seats.

display\_available\_buses(): Displays a list of available buses along with their details.

make\_reservation(bus\_number, passenger\_name): Makes a reservation for a passenger on a specific bus, updating available seats.

display\_passenger\_reservations(passenger\_name): Displays reservations made by a specific passenger.

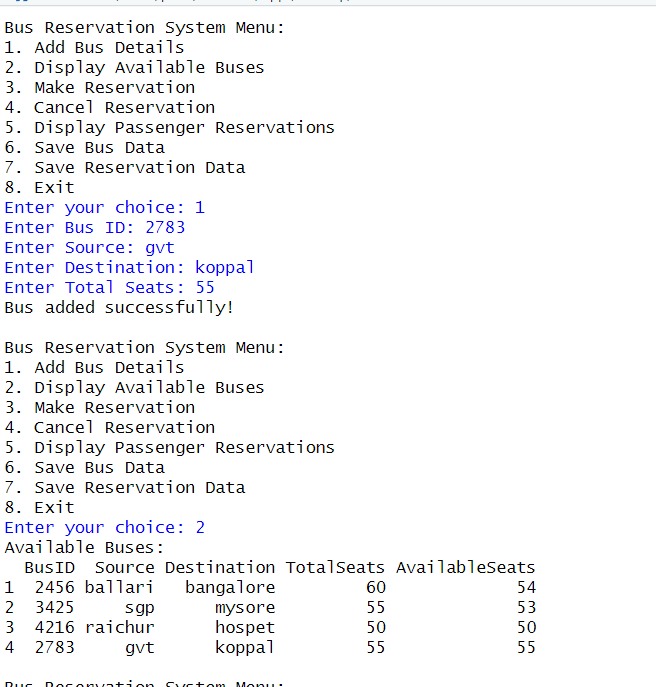
main\_menu(): Displays the main menu options and handles user interactions.

**Main Loop:**

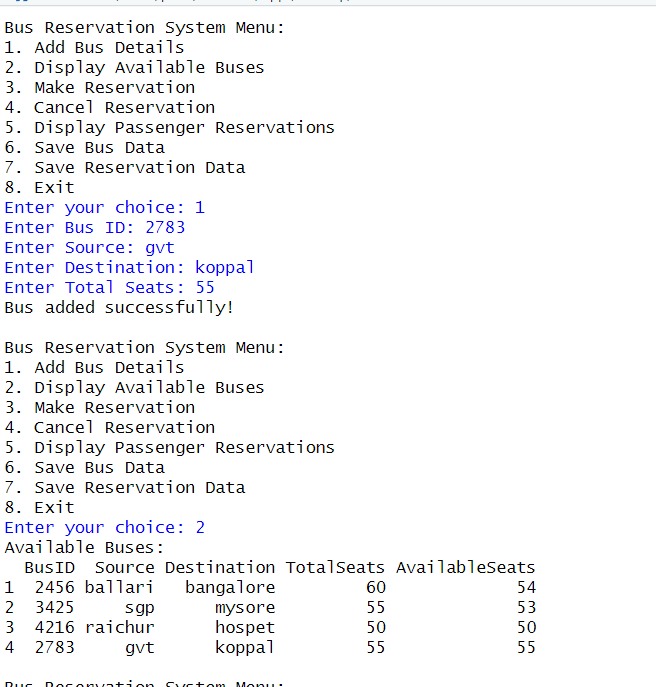
The system runs in a loop where the main\_menu() function is called repeatedly until the user chooses to exit.

**Results**

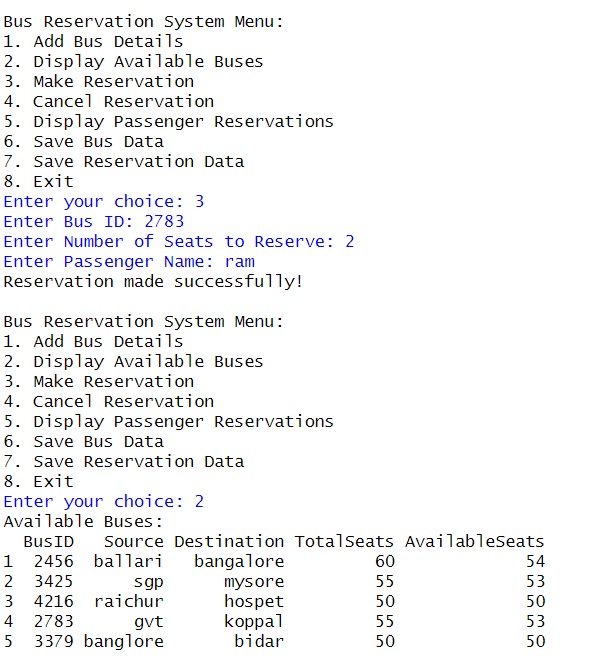
**step 1: adding bus details to csv file**



**Step 2: displaying bus details**

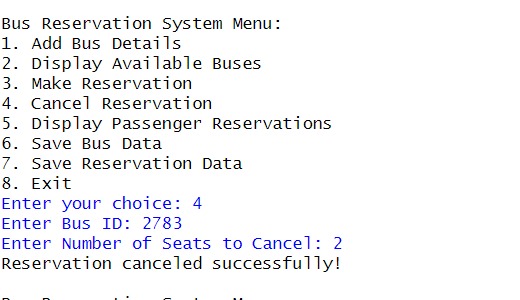


**Step 3: making bus seat reservation using csv file**

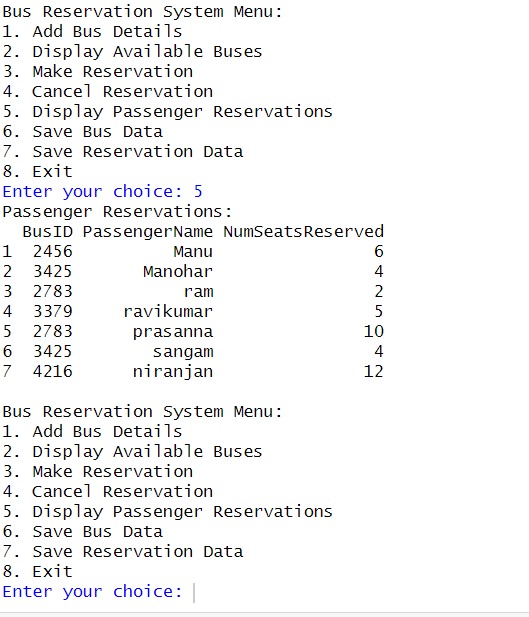


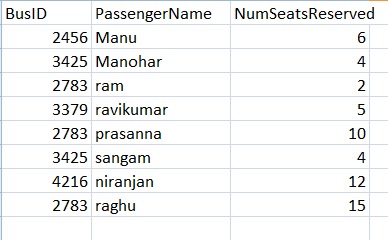


**Step 4: cancelling reserved seats**

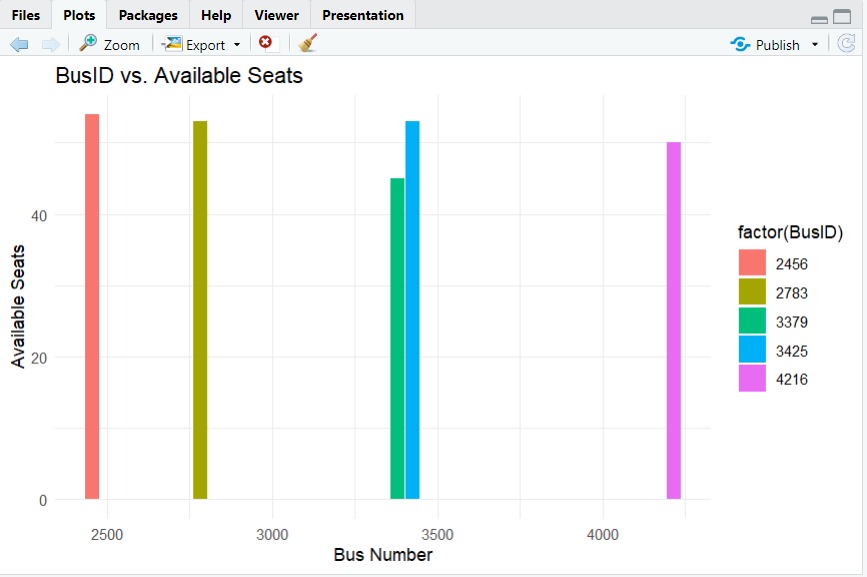


**Step 5: displaying reserved seats from csv file**





**Step 6: plotting graph Bus.ID vs Available Seats**



**SOURCE CODE:**

# Load necessary libraries

library(readr)

# Define the CSV file name for bus data

bus\_csv\_file <- "bus\_details.csv"

# Define the CSV file name for passenger reservations

reservation\_csv\_file <- "passenger\_reservations.csv"

# Check if the bus data CSV file exists, and create it if not

if (!file.exists(bus\_csv\_file)) {

bus\_data <- data.frame(

BusID = integer(0),

Source = character(0),

Destination = character(0),

TotalSeats = integer(0),

AvailableSeats = integer(0)

)

write.csv(bus\_data, file = bus\_csv\_file, row.names = FALSE)

}

# Check if the passenger reservations CSV file exists, and create it if not

if (!file.exists(reservation\_csv\_file)) {

reservation\_data <- data.frame(

BusID = integer(0),

PassengerName = character(0),

NumSeatsReserved = integer(0)

)

write.csv(reservation\_data, file = reservation\_csv\_file, row.names = FALSE)

}

# Function to add bus details

add\_bus <- function(bus\_id, source, destination, total\_seats) {

bus\_data <- read.csv(bus\_csv\_file, stringsAsFactors = FALSE)

new\_bus <- data.frame(

BusID = bus\_id,

Source = source,

Destination = destination,

TotalSeats = total\_seats,

AvailableSeats = total\_seats

)

bus\_data <- rbind(bus\_data, new\_bus)

write.csv(bus\_data, file = bus\_csv\_file, row.names = FALSE)

cat("Bus added successfully!\n")

}

# Function to display available buses

display\_buses <- function() {

bus\_data <- read.csv(bus\_csv\_file, stringsAsFactors = FALSE)

available\_buses <- bus\_data[bus\_data$AvailableSeats > 0, ]

if (nrow(available\_buses) == 0) {

cat("No available buses at the moment.\n")

} else {

cat("Available Buses:\n")

print(available\_buses)

}

}

# Function to make a reservation without mobile number

make\_reservation <- function(bus\_id, num\_seats) {

bus\_data <- read.csv(bus\_csv\_file, stringsAsFactors = FALSE)

bus\_index <- which(bus\_data$BusID == bus\_id)

if (length(bus\_index) == 0) {

cat("Bus not found!\n")

return()

}

bus <- bus\_data[bus\_index, ]

if (bus$AvailableSeats < num\_seats) {

cat("Not enough available seats on this bus.\n")

return()

}

passenger\_name <- readline("Enter Passenger Name: ")

# Check if the reservation data file exists

if (!file.exists(reservation\_csv\_file)) {

reservation\_data <- data.frame(

BusID = integer(0),

PassengerName = character(0),

NumSeatsReserved = integer(0)

)

} else {

reservation\_data <- read.csv(reservation\_csv\_file, stringsAsFactors = FALSE)

}

# Create a new reservation entry

new\_reservation <- data.frame(

BusID = bus\_id,

PassengerName = passenger\_name,

NumSeatsReserved = num\_seats

)

# Append the new reservation to the existing reservations

reservation\_data <- rbind(reservation\_data, new\_reservation)

# Save the reservation data to a CSV file

write.csv(reservation\_data, file = reservation\_csv\_file, row.names = FALSE)

# Update available seats in the bus data

bus\_data$AvailableSeats[bus\_index] <- bus$AvailableSeats - num\_seats

write.csv(bus\_data, file = bus\_csv\_file, row.names = FALSE)

cat("Reservation made successfully!\n")

}

# Function to cancel a reservation

cancel\_reservation <- function(bus\_id, num\_seats) {

bus\_data <- read.csv(bus\_csv\_file, stringsAsFactors = FALSE)

bus\_index <- which(bus\_data$BusID == bus\_id)

if (length(bus\_index) == 0) {

cat("Bus not found!\n")

return()

}

bus <- bus\_data[bus\_index, ]

if (bus$AvailableSeats + num\_seats > bus$TotalSeats) {

cat("Invalid number of seats to cancel.\n")

return()

}

bus\_data$AvailableSeats[bus\_index] <- bus$AvailableSeats + num\_seats

write.csv(bus\_data, file = bus\_csv\_file, row.names = FALSE)

cat("Reservation canceled successfully!\n")

}

# Function to save bus data to a CSV file

save\_bus\_data <- function() {

bus\_data <- read.csv(bus\_csv\_file, stringsAsFactors = FALSE)

write.csv(bus\_data, file = bus\_csv\_file, row.names = FALSE)

}

# Function to save reservation data to a CSV file

save\_reservation\_data <- function() {

reservation\_data <- read.csv(reservation\_csv\_file, stringsAsFactors = FALSE)

write.csv(reservation\_data, file = reservation\_csv\_file, row.names = FALSE)

}

# Function placeholder for displaying passenger reservations

display\_passenger\_reservations <- function() {

reservation\_data <- read.csv(reservation\_csv\_file, stringsAsFactors = FALSE)

if (nrow(reservation\_data) == 0) {

cat("No passenger reservations at the moment.\n")

} else {

cat("Passenger Reservations:\n")

print(reservation\_data)

}

}

# Main program loop

while (TRUE) {

cat("\nBus Reservation System Menu:\n")

cat("1. Add Bus Details\n")

cat("2. Display Available Buses\n")

cat("3. Make Reservation\n")

cat("4. Cancel Reservation\n")

cat("5. Display Passenger Reservations\n")

cat("6. Save Bus Data\n")

cat("7. Save Reservation Data\n")

cat("8. Exit\n")

choice <- as.integer(readline("Enter your choice: "))

if (choice == 1) {

bus\_id <- as.integer(readline("Enter Bus ID: "))

source <- readline("Enter Source: ")

destination <- readline("Enter Destination: ")

total\_seats <- as.integer(readline("Enter Total Seats: "))

add\_bus(bus\_id, source, destination, total\_seats)

} else if (choice == 2) {

display\_buses()

} else if (choice == 3) {

bus\_id <- as.integer(readline("Enter Bus ID: "))

num\_seats <- as.integer(readline("Enter Number of Seats to Reserve: "))

make\_reservation(bus\_id, num\_seats)

} else if (choice == 4) {

bus\_id <- as.integer(readline("Enter Bus ID: "))

num\_seats <- as.integer(readline("Enter Number of Seats to Cancel: "))

cancel\_reservation(bus\_id, num\_seats)

} else if (choice == 5) {

display\_passenger\_reservations()

} else if (choice == 6) {

save\_bus\_data()

cat("Bus data saved successfully!\n")

} else if (choice == 7) {

save\_reservation\_data()

cat("Reservation data saved successfully!\n")

} else if (choice == 8) {

cat("Thank you for using the Bus Reservation System. Exiting...\n")

break

} else {

cat("Invalid choice. Please try again.\n")

}

}

**Code to plot graphs:**

# Load ggplot2 library

library(ggplot2)

# Data

bus\_data <- read.csv("bus\_details.csv")

# Create a bar plot of BusID vs. AvailableSeats

ggplot(data = bus\_data, aes(x = BusID, y = AvailableSeats, fill = factor(BusID))) +

geom\_bar(stat = "identity") +

labs(title = "BusID vs. Available Seats",

x = "Bus Number",

y = "Available Seats") +

 theme\_minimal()

CONCLUSION:

The culmination of the Bus Reservation Project marks a significant milestone in the realm of transportation and travel management. This project has successfully addressed the complexities and challenges associated with bus ticket booking and has ushered in a new era of convenience, accessibility, and efficiency for both passengers and bus operators.

As we conclude this chapter of the Bus Reservation Project, we do so with a deep sense of accomplishment and a vision for the future of travel. This project not only simplifies bus ticket booking but also symbolizes our commitment to enhancing the travel experience for all.