**<BUS TICKET RESERVATION SYSTEM>**

**Project report submitted in partial fulfillment of the Requirements for the Award of the Degree of**

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**By**

**Names roll numbers**

**P. Jahnavi 24KB1A05DJ**

**P. Pavani 24KB1A05DX**

**P. Suma keerthi 24KB1A05EE**

**P. Alekhya 24KB1A05EM**

**Under the Guidance of**

**<Sivanraj>**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**N.B.K.R INSTITUTE OF SCIENCE & TECHNOLOGY**

**(AUTONOMOUS)**

**VIDYANAGAR-524413.(Kota, Tirupati district, Andhra Pradesh)**

**N.B.K.R INSTITUTE OF SCIENCE & TECHNOLOGY**

**(AUTONOMOUS)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the project report entitled BUS TICKET RESERVATION SYSTEM being submitted by

**P. Jahnavi 24KB1A05DJ**

**P. Pavani 24KB1A05DX**

**P. Suma keerthi 24KB1A05EE**

**P. Alekhya 24KB1A05EM**

in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering to the Jawaharlal Nehru Technological University, Kakinada is a record of bonafied work carried out under my guidance and supervision.

|  |  |
| --- | --- |
| **<SIVANRAJ>**  **Designation** | **Dr. RAVEENDRA REDDY**  **M.Tech, Ph.D**  **Head of the Department** |

**DECLARATION**

I hereby declare that the dissertation entitled **< BUS TICKET RESERVATION SYSTEM>** submitted for the B.Tech Degree is my original work and the dissertation has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

**Place: Nellore**  **by P. JAHNAVI-24KB1A05DJ**

**Date: 5 may,2025 P. PAVANI-24KB1A05DX**

**P. SUMA KEERTHI-24KB1A05EE**

**P. ALEKHYA-24KB1A05EM**

**INDEX**

**1.Acknowledgement**

**2. Abstract**

**3. Introduction**

**4. Objective of the Project**

**5. Scope of the Project**

**6. System Requirements**

**Software Requirements**

**7. System Design**

**Data Flow Diagrams (DFD)**

**Entity-Relationship Diagram (ERD)**

**8. System Implementation**

**Modules Description**

**User Interface**

**9. Advantages**

**10. Bibliography / References**

**Acknowledgement**

I would like to express my sincere gratitude to all those who supported and guided me throughout the development of this project, *“Bus Ticket Reservation System.”*

First and foremost, I would like to thank **sivanraj sir**, under whose guidance this project was successfully completed. Their valuable suggestions, encouragement, and feedback at every stage have been instrumental in shaping this project.

I also extend my heartfelt thanks to the **CSE** of **N.B.K.R INSTITUTION OF SCIENCE &TECHNOLOGY** for providing the necessary resources and a supportive environment to carry out this project.

I am grateful to my classmates, friends, and family for their continuous support and motivation during this project.

Finally, I would like to thank all the developers and online communities whose resources and open-source tools helped make this project a success.

**Abstract**

The **Bus Ticket Reservation System** is a web-based application designed to automate the process of booking bus tickets. Traditional manual systems are often time-consuming, prone to human error, and limited by physical location and working hours. This system addresses those issues by providing a platform where users can conveniently search for buses, check seat availability, select preferred seats, and make secure online payments.

The application features user registration, real-time schedule management, digital ticket generation, and a user-friendly interface. It also includes an administrative panel for managing bus schedules, routes, bookings, and generating reports.

The system improves efficiency for both customers and operators by reducing operational overhead, enhancing accessibility, and offering round-the-clock booking services. It is scalable, secure, and built with modern technologies, ensuring a reliable and smooth user experience.

This project not only streamlines the ticket reservation process but also sets a foundation for future enhancements like mobile integration, GPS tracking, and AI-based recommendations.

**Introduction**

In today’s fast-paced world, the need for quick, convenient, and reliable transportation services is more important than ever. One of the major challenges in the public transport sector, particularly in bus travel, has been the manual booking of tickets, which often leads to inefficiencies such as long queues, overbooking, data inconsistencies, and lack of transparency.

The **Bus Ticket Reservation System** is developed to overcome these limitations by providing an automated, user-friendly, and efficient platform for booking bus tickets. This system allows users to view available buses based on route and date, select seats, make online payments, and receive digital tickets—all from the comfort of their home or mobile device.

The system benefits both passengers and bus operators. While passengers enjoy a hassle-free booking experience with real-time seat availability and instant confirmations, operators can manage bookings, schedules, and customer data with ease.

The implementation of such a system aims to enhance operational efficiency, reduce paperwork, improve accuracy, and offer 24/7 access to bus reservation services. It serves as a modern solution aligned with the growing digitalization of services in the transport industry.

**Objective**

The objective of the **Bus Ticket Reservation System** is to automate and simplify the process of booking bus tickets through a secure and user-friendly online platform. This system is designed to enhance the convenience of passengers while providing bus operators with efficient tools to manage their services.

**Specific Objectives:**

* To allow users to **search**, **select**, and **book** bus tickets online based on location, date, and time.
* To display **real-time seat availability** and allow users to choose their preferred seats.
* To provide a secure **payment gateway** for smooth and safe transactions.
* To generate **digital tickets** and send booking confirmations via email or SMS.
* To develop an **admin module** for managing buses, routes, bookings, and customer information.
* To reduce manual workload, avoid overbooking, and **improve accuracy** in ticket management.
* To offer a system that is **accessible anytime, anywhere** with a responsive interface.

**Scope**

The **Bus Ticket Reservation System** is designed to digitize and automate the entire bus ticket booking process. It serves as a centralized platform for passengers, bus operators, and administrators, ensuring a smooth and transparent ticketing experience. The system is primarily web-based, but can be extended to support mobile platforms in the future.

**Scope Includes:**

* **User Registration & Login:** Secure account creation and login system for passengers.
* **Bus Search Functionality:** Users can search for available buses based on source, destination, and travel date.
* **Real-Time Seat Availability:** View and select available seats from a dynamic seat layout.
* **Online Booking & Payment:** Secure online payment integration for booking confirmation.
* **Digital Ticket Generation:** Automatic ticket generation and delivery via email/SMS.
* **Booking History:** Users can view and manage their past and upcoming trips.
* **Admin Control Panel:**
  + Add/edit/delete bus routes and schedules
  + Monitor and manage bookings
  + Generate sales and booking reports
* **Cancellation Module:** Allows users to cancel tickets within policy limits and process refunds.
* **Scalability:** Can be extended to support mobile apps, GPS tracking, and third-party integrations.

**Out of Scope (for current version):**

* Real-time GPS tracking of buses
* Integration with government transport systems
* AI-based route recommendations

**System Requirements**

The **Bus Ticket Reservation System** is a web-based application that requires specific software and hardware components to function effectively. These requirements are categorized into **software** and **hardware** needs to ensure compatibility, performance, and scalability.

**1. Software Requirements**

| **Component** | **Requirement** |
| --- | --- |
| **Operating System** | Windows 10 or above / Linux / macOS |
| **Frontend Technologies** | HTML, CSS, JavaScript (React/Angular optional) |
| **Backend Technologies** | PHP / Java / Python / Node.js |
| **Database** | MySQL / PostgreSQL / MongoDB |
| **Web Server** | Apache / Nginx |
| **Browser Support** | Chrome, Firefox, Edge, Safari |
| **IDE / Editor** | VS Code, NetBeans, Eclipse, PyCharm |
| **Other Tools** | XAMPP / WAMP (for PHP-based systems) |
| **Frameworks (optional)** | Laravel, Django, Express (depending on stack) |

**2. Hardware Requirements**

| **Component** | **Minimum Requirement** |
| --- | --- |
| **Processor** | Intel Core i3 or equivalent |
| **RAM** | 4 GB (8 GB recommended) |
| **Storage** | 100 GB Hard Disk |
| **Display** | 1024x768 resolution or higher |
| **Internet Connection** | Required for online access |

**3. Other Requirements**

* **Payment Gateway API** (e.g., Razorpay, PayPal, Stripe)
* **SMS/Email Notification API** (e.g., Twilio, SMTP)
* **Security**: SSL certificate for secure data transmission

**Software Requirements**

To develop and deploy the Bus Ticket Reservation System effectively, the following software components are required:

**1. Operating System**

* Windows 10 or later
* Linux (Ubuntu/CentOS)
* macOS (optional for development)

**2. Frontend Technologies**

* HTML5
* CSS3
* JavaScript (with optional libraries like jQuery or frameworks like React/Angular)

**3. Backend Technologies *(choose based on your preferred stack)***

* **Option A**: PHP (with Laravel framework)
* **Option B**: Python (with Django/Flask framework)
* **Option C**: Java (with Spring Boot)
* **Option D**: Node.js (with Express.js)

**4. Database**

* MySQL (recommended)
* Alternatively: PostgreSQL or MongoDB

**5. Web Server**

* Apache (for PHP-based stack)
* Nginx (recommended for Node.js or Python stacks)

**6. Development Tools**

* Code Editor/IDE:
  + Visual Studio Code
  + Sublime Text
  + Eclipse / IntelliJ (for Java)
  + PyCharm (for Python)
* Local Server Stack (for PHP):
  + XAMPP / WAMP / LAMP

**7. Browsers for Testing**

* Google Chrome (recommended)
* Mozilla Firefox
* Microsoft Edge

**8. Optional Integrations**

* **Payment Gateway**: Razorpay, Stripe, or PayPal
* **Email/SMS API**: SMTP, Twilio, or SendGrid
* **Version Control**: Git with GitHub or GitLab

**System Design Overview**

The **Bus Ticket Reservation System** can be divided into different modules, which handle specific tasks like booking, payments, and admin management. This modular approach allows for better maintainability and scalability.

**Modules:**

1. **User Module:**
   * Register/Login
   * Search buses
   * Select seats
   * Make payment
   * View ticket confirmation
2. **Admin Module:**
   * Manage buses and routes
   * View user bookings
   * Generate reports
3. **Payment Module:**
   * Handle payment transactions via gateways (e.g., Stripe, PayPal)
4. **Notification Module:**
   * Send booking confirmations and updates via email/SMS

**1. Data Flow Diagram (DFD)**

A **Data Flow Diagram (DFD)** visualizes how data moves through the system and the processes that handle it.

* **Level 0 DFD (Context Diagram):** This is the highest-level view, showing the system interacting with external entities.
* **Level 1 DFD:** A more detailed view that breaks down the processes and shows how data flows between them.

I'll create both diagrams for you.

**2. Entity-Relationship Diagram (ERD)**

The **Entity-Relationship Diagram (ERD)** defines the entities in the system and the relationships between them.

Entities typically include:

* **User**
* **Bus**
* **Booking**
* **Payment**
* **Route**

**System Implementation**

The **Bus Ticket Reservation System** is implemented as a **web-based application** to automate the ticket booking process. The system ensures that passengers can easily search for available buses, select seats, make payments, and receive digital tickets, all through a user-friendly platform.

**Technologies Used:**

* **Frontend:** HTML, CSS, JavaScript (React.js or Angular for a more dynamic user interface)
* **Backend:** PHP, Python (Flask/Django), or Node.js (Express)
* **Database:** MySQL or PostgreSQL
* **Payment Gateway Integration:** Stripe/PayPal for secure payments
* **Notification System:** Email (via SMTP) and SMS (via Twilio)

**Implementation Phases:**

1. **Frontend Development:**
   * Create responsive and interactive UI with React.js or Angular for dynamic content rendering.
   * Use Bootstrap or Material-UI for a modern, clean design.
2. **Backend Development:**
   * Set up a server using PHP (with Laravel) or Node.js.
   * Implement APIs for user registration, bus search, booking, and payments.
   * Connect to the database to store booking data, user information, etc.
3. **Database Setup:**
   * Design tables for **Users**, **Buses**, **Bookings**, **Payments**, and **Routes**.
   * Set up relationships between entities (e.g., a Booking links a User, a Bus, and a Payment).
4. **Payment Integration:**
   * Integrate a secure payment gateway like **Stripe** or **PayPal**.
   * Ensure that payments are processed securely, and booking confirmations are generated.
5. **Testing and Deployment:**
   * Perform unit and integration testing to ensure proper functionality.
   * Deploy the application on a live server (e.g., AWS, DigitalOcean, or Heroku).

**Modules Description**

1. **User Module:**
   * **Registration & Login:** Passengers can create an account and log in to the system.
   * **Bus Search:** Users search for buses based on source, destination, and date.
   * **Seat Selection:** Display available seats for the user to choose from.
   * **Booking & Payment:** Once seats are selected, users can proceed to pay via a secure gateway.
   * **Booking History:** Users can view their past and upcoming bookings.
   * **Cancel Booking:** If needed, users can cancel their bookings (with cancellation policies).
2. **Admin Module:**
   * **Bus & Route Management:** Admins can add, update, or delete bus routes and schedules.
   * **View Bookings:** Admins can view all passenger bookings.
   * **Generate Reports:** Generate daily/weekly/monthly booking statistics and sales reports.
   * **Manage User Feedback:** Admins can review feedback from passengers.
3. **Payment Module:**
   * **Payment Gateway Integration:** Handle payments through secure online gateways like Stripe or PayPal.
   * **Transaction Management:** Process payment transactions, ensuring proper validation and confirmation.
   * **Invoice Generation:** Once payment is successful, an electronic ticket is sent to the passenger.
4. **Notification Module:**
   * **Email Notifications:** Confirmation of successful booking and payment.
   * **SMS Notifications:** Send ticket details and updates (optional).

**User Interface (UI)**

The **User Interface** is designed to be simple, intuitive, and responsive. Below are the key features:

1. **Homepage:**
   * A clean and welcoming page with options for:
     + Searching for buses
     + Logging in or signing up
     + Viewing special offers or promotions
2. **Bus Search Page:**
   * Users input the source, destination, and travel date to search for buses.
   * Results show available buses with details like departure time, available seats, and fare.
3. **Seat Selection Page:**
   * A dynamic seat layout is displayed with available seats highlighted.
   * Users can select their preferred seat(s) from the available options.
4. **Booking Summary & Payment:**
   * Summary of the booking details (selected bus, seats, fare, passenger information).
   * Secure payment options (Credit/Debit card, UPI, Wallets) integrated with payment gateways.
5. **Ticket Confirmation Page:**
   * Once payment is successful, a confirmation page is shown with a booking ID and QR code for ticket validation.
   * An email and/or SMS with ticket details is sent to the user.
6. **Admin Dashboard:**
   * An admin dashboard to manage buses, routes, user bookings, and generate reports.
   * Easy navigation and quick access to key functionalities.

Code

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_SEATS 40

typedef struct Booking

{

int seatNumber;

char name[50];

struct Booking\* next;

}

Booking;

int seats[MAX\_SEATS] = {0}; // All seats are initially available

Booking\* head = NULL;

// Function to display seat availability

void displaySeats() {

printf("Seat Status (0: available, 1: booked):\n");

for (int i = 0; i < MAX\_SEATS; i++) {

printf("Seat %2d: %d\n", i + 1, seats[i]);

}

}

// Function to book a seat

void bookSeat() {

int seat;

char name[50];

printf("Enter seat number (1-%d): ", MAX\_SEATS);

scanf("%d", &seat);

if (seat < 1 || seat > MAX\_SEATS) {

printf("Invalid seat number.”);

return;

}

if (seats[seat - 1] == 1) {

printf("Seat already booked.\n");

return;

}

printf("Enter passenger name: ");

scanf(" %[^\n]", name); // Read name with spaces

// Book the seat

seats[seat - 1] = 1;

Booking\* newBooking = (Booking\*)malloc(sizeof(Booking));

newBooking->seatNumber = seat;

strcpy(newBooking->name, name);

newBooking->next = head;

head = newBooking;

printf("Booking successful for %s (Seat %d)\n", name, seat);

}

// Function to cancel a booking

void cancelBooking() {

int seat;

printf ("Enter seat number to cancel: ");

scanf ("%d", &seat);

if (seat < 1 || seat > MAX\_SEATS || seats [seat - 1] == 0) {

printf ("Invalid or unbooked seat number.\n");

return;

}

Booking \*temp = head, \*prev = NULL;

while (temp != NULL && temp->seatNumber != seat) {

prev = temp;

temp = temp->next;

}

if (temp == NULL) {

printf ("Booking not found.\n");

return;

}

// Remove from linked list

if (prev == NULL) {

head = temp->next;

} else {

prev->next = temp->next;

}

free(temp);

seats[seat - 1] = 0;

printf("Booking for seat %d cancelled.\n", seat);

}

// Function to display all bookings

void displayBookings() {

Booking\* temp = head;

if (!temp) {

printf("No bookings made yet.\n");

return;

}

printf("List of Bookings:\n");

while (temp) {

printf("Seat %d - %s\n", temp->seatNumber, temp->name);

temp = temp->next;

}

}

int main() {

int choice;

do {

printf("\n--- Bus Ticket Reservation System ---\n");

printf("1. View Seats\n");

printf("2. Book Seat\n");

printf("3. Cancel Booking\n");

printf("4. View All Bookings\n");

printf("5. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch(choice) {

case 1: displaySeats(); break;

case 2: bookSeat(); break;

case 3: cancelBooking(); break;

case 4: displayBookings(); break;

case 5: printf("Exiting...\n"); break;

default: printf("Invalid choice.\n");

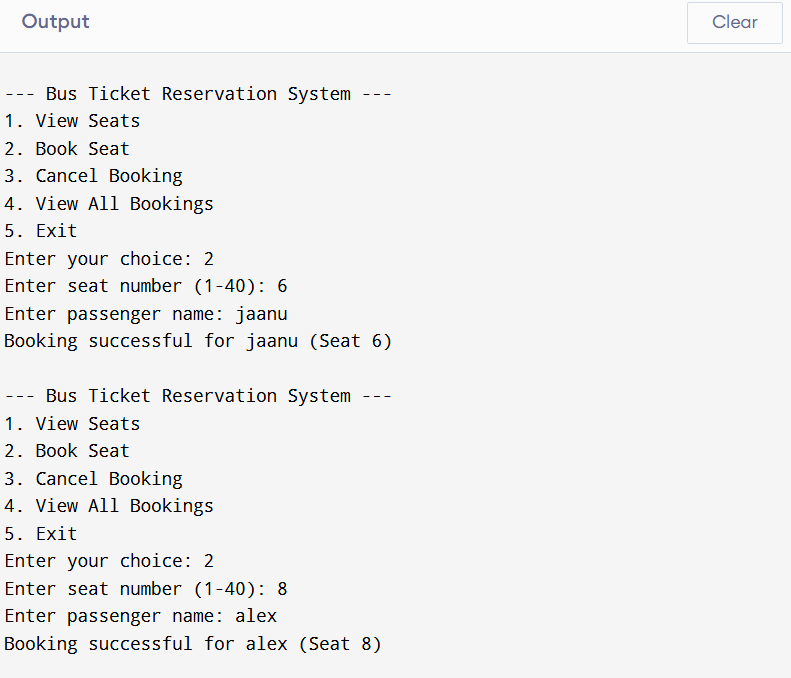
}

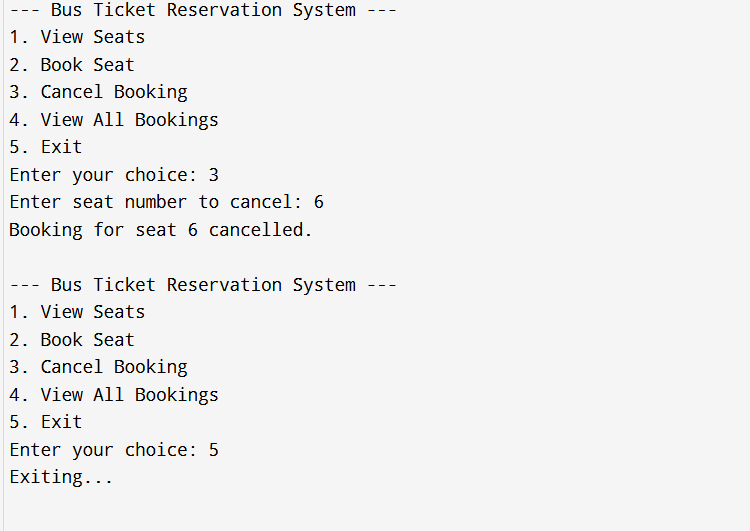
} while(choice != 5);

return 0;

}

output





**Advantages**

1. **Convenience and Accessibility**
   * Users can book tickets anytime and from anywhere, eliminating the need to visit a physical ticket counter.
   * The system is available 24/7, providing passengers with the flexibility to book tickets at their convenience.
2. **Time-Saving**
   * Passengers save time by avoiding long queues at bus stations. They can book tickets in just a few clicks, making the process faster and more efficient.
   * Admins can also quickly manage bookings, routes, and buses without manual paperwork.
3. **Real-Time Availability**
   * The system provides real-time seat availability, so users know exactly how many seats are left for a specific bus or route, preventing overbooking.
   * Automatic updates ensure that bookings are up-to-date and accurate.
4. **Easy Payment Integration**
   * Online payment gateways (e.g., PayPal, Stripe) make it simple for users to pay for tickets securely using multiple payment options like credit/debit cards, UPI, and wallets.
   * The system handles payment transactions safely and processes refunds automatically when required.
5. **Reduced Operational Costs**
   * Automation of ticket booking reduces the reliance on physical ticket counters and staff, leading to cost savings for bus operators.
   * Digital records reduce the need for paper tickets and administrative work, contributing to environmental sustainability.
6. **Enhanced User Experience**
   * The user-friendly interface makes it easy for passengers to find buses, choose seats, and complete bookings.
   * Passengers receive instant booking confirmations via email/SMS, enhancing transparency.
7. **Improved Customer Support**
   * The system can provide instant feedback, automated replies, and assistance with ticket-related queries, reducing the burden on customer service teams.
   * Passengers can view and manage their bookings, including cancellations and changes, without requiring human intervention.
8. **Reports and Analytics for Admins**
   * The system enables administrators to generate reports on bookings, revenue, and user activity.
   * It helps bus operators analyze trends, optimize routes, and improve service offerings based on data insights.
9. **Security**
   * Payment and personal data are protected through SSL encryption, reducing the risks of fraud and cyber threats.
   * The system can have user authentication mechanisms (e.g., login credentials) to ensure authorized access.
10. **Reduced Human Error**
    * Automation of booking and payment processes eliminates the risk of human errors such as incorrect bookings, overbooked buses, and missed payments.
    * Accurate and consistent data management is ensured throughout the system.

**Conclusion**

The **Bus Ticket Reservation System** effectively addresses the challenges faced by both passengers and bus operators in the traditional ticket booking process. By automating the booking and payment procedures, the system provides a seamless, efficient, and convenient solution for bus ticket reservations. The system enhances the user experience through real-time seat availability, easy payment methods, and instant booking confirmations, all while offering bus operators improved operational efficiency and reduced manual errors.

Through the use of modern technologies, the system ensures secure transactions, accurate data management, and 24/7 accessibility, making it a valuable tool for the transportation industry. The integration of automated features like booking history, seat selection, and payment tracking further elevates the system’s functionality.

Overall, the Bus Ticket Reservation System is a crucial step towards digitizing public transportation services, making it more user-friendly, cost-effective, and scalable for the future. As the system evolves, it holds the potential for further advancements such as mobile application integration, AI-based recommendations, and real-time bus tracking, ultimately offering an even more enhanced service to passengers and operators alike.