# INSTANT RESERVATION SYSTEM

***A Report submitted to the Rajiv Gandhi University of Knowledge and Technologies in partial fulfillment of the degree of***

**Bachelor of technology in**

**Computer Science and Engineering**

**Submitted by**

**D. Sandhya (S180678)**

**G. Manibabu (S180169)**

**P. Rajeswari (s180978)**

**3rd year BTech 2ndSemester**

**Under the supervision of**

**Sri. Sateesh kumar Sankarapu**

*Asst. Professor-*

*Department of CSE RGUKT-SRIKAKULAM*



#### Department of Computer Science and Engineering

**Rajiv Gandhi University of Knowledge and Technologies,Srikakulam**

#### S.M. Puram (V), Etcherla (M), Srikakulam (Dt) – 532410



# CERTIFICATE

This is to certify that the report entitled “**INSTANT RESERVATION SYSTEM**” was submitted by Sandhya Dubba, bearing ID. No. S180678, Gokeda Manibabu, bearing ID. No. S180169,Pachipenta Rajeswari,bearing ID. No. s180978 in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science is a bonafide work carried out by them under my supervision and guidance.

The report has not been submitted previously in part or in full to this or any other University or Institution to award any degree or diploma.

#### Sateesh Kumar sankarapu , Sesha Kumar Nalluri,

**Project Guide, Head of the Department,**

#### Department of CSE, Department of CSE,

**RGUKT,SRIKAKULAM. RGUKT,SRIKAKULAM.**

# DECLARATION

We **Sandhya Dubba and Gokeda Manibabu and Pachipenta Rajeswari** hereby declare that this report entitled **“INSTANT RESERVATION SYSTEM”** submitted by us under the guidance and supervision of **Sateesh Kumar Sankarapu** is a bonafide work.

We also declare that it has not been submitted previously in part or in full to this University or other University or Institution to award any degree or diploma.

(Sandhya . Dubba)

ID NO . S180678

(Gokeda. Manibabu)

ID NO . S180169

(Pachipenta.Rajeswari)

ID NO . S180978

# ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to, my project Guide **Sateesh Kumar Sankarapu**, for valuable suggestions and keen interest throughout the progress of my course of research.

We are grateful to **SeshaKumar Nalluri**, HOD CSE, for providing excellent computing facilities and a congenial atmosphere for progressing with my project.

At the outset, we would like to thank **Rajiv Gandhi University of Knowledge and Technologies, Srikakulam** for providing all the necessary resources for the successful completion of our course work. At last, but not least we thank our classmates and other students for their physical and moral support.

With Sincere Regards **Sandhya Dubba, Manibabu Gokeda,**

**Pachipenta Rajeswari.**

# ABSTRACT

The IRCTC Online Current Reservation System is an innovative solution that addresses the challenges of immediate train reservations in India. With the traditional ticket booking process, passengers had to plan their travel well in advance, as booking seats after the chart preparation process was difficult, if not impossible. This often led to missed opportunities and inconvenience for travelers. The IRCTC Online Current Reservation System aims to eliminate these issues by allowing passengers to book unreserved seats even after the chart preparation. The system provides users with a list of un booked seats for their complete and partial journeys according to the vacancies. The system also suggests the best options for changing berths to ensure passenger.

Contents

[INSTANT RESERVATION SYSTEM 1](#_Toc140417041)

[Bachelor of technology in 1](#_Toc140417042)

[Computer Science and Engineering 1](#_Toc140417043)

[D.Sandhya (S180678) 1](#_Toc140417044)

[G.Manibabu (S180169) 1](#_Toc140417045)

[P.Rajeswari (s180978) 1](#_Toc140417046)

[Sri. Sateesh kumar Sankarapu 1](#_Toc140417047)

[CERTIFICATE i](#_Toc140417048)

[DECLARATION ii](#_Toc140417049)

[ACKNOWLEDGEMENTS iii](#_Toc140417050)

[ABSTRACT iv](#_Toc140417051)

[CHAPTER 1 1](#_Toc140417053)

[1INTRODUCTION 1](#_Toc140417054)

[1.1 Introduction 1](#_Toc140417052)

[1.2 Motivation 1](#_Toc140417055)

[1.3 Problem Statement 1](#_Toc140417056)

[1.4 Objectives 2](#_Toc140417057)

[1.5 Goals: 3](#_Toc140417058)

[1.6 Scope 3](#_Toc140417064)

[1.7 Applications 3](#_Toc140417065)

[1.8 Limitations 3](#_Toc140417066)

[CHAPTER2 5](#_Toc140417067)

[LITERATURE SURVEY 5](#_Toc140417068)

[2.1 Collect Information 5](#_Toc140417069)

[2.2 Summary 7](#_Toc140417070)

[CHAPTER 3 8](#_Toc140417071)

[3.1 Existing System 8](#_Toc140417072)

[3.2 Proposed System 9](#_Toc140417073)

[3.3 Advantages 10](#_Toc140417074)

[3.4 Software System Requirements 10](#_Toc140417075)

[Hardware Requirements 10](#_Toc140417076)

[CHAPTER 4 11](#_Toc140417077)

[4.1 Implementation Tools 11](#_Toc140417080)

[4.2 Frameworks 11](#_Toc140417080)

[CHAPTER 5 14](#_Toc140417082)

[5.1 SOURCE CODE 11](#_Toc140417080)

[CHAPTER 6 40](#_Toc140417083)

[6.1 Conclusion 41](#_Toc140417080)

[CHAPTER 7 41](#_Toc140417085)

[7.1 Future Enchancement 41](#_Toc140417080)

[CHAPTER 8 42](#_Toc140417086)

[References: 42](#_Toc140417087)

## 

## 

## 

### Introduction

## CHAPTER 1

## 1INTRODUCTION

The Indian Railway Catering and Tourism Corporation (IRCTC) is an online platform widely used for booking train tickets in India. One of the critical features of the IRCTC application is seat booking which allows users to reserve seats for their train journeys. However, developing a reliable and efficient seat booking system posses a significant challenge for IRCTC due to the vast number of passengers using the service and the complexity of the Indian railway network. This study intends to propose a suggested system that helps users to book their berth even in the last minute, if berths are available.

### Motivation

To make it possible for those who are unable to book their tickets at a certain time before chart preparation. The objective of this project is to develop a ticket booking system that will allow users to book their tickets independently without any human intervention, like TCs in the last moment.

### Problem Statement

The Indian rail ticket reservation system lacks alternative options for users when there are no vacancies available for a specific source to destination journey. It does not provide information on vacancies at prior or subsequent stations, nor does it offer berth shift facilities when direct vacancies are unavailable. The absence of suggestions for partial journeys and berth shifts leads to inefficiencies and inflexibility in the booking process, resulting in a suboptimal user experience. Enhancements are required to address these limitations, provide comprehensive journey options, and improve user satisfaction while maximizing seat utilization.

### Objectives

1. Ticket booking for direct journey from required source to destination.

2. Enhance the Indian rail ticket reservation system to incorporate alternative options for users when there are no vacancies for a specific source to destination journey.

3. Provide information on vacancies at prior or subsequent stations along the route, allowing users to consider alternative boarding points or destinations.

4. Introduce the functionality to suggest partial journeys with berth shifts, enabling users to combine multiple segments to create a complete journey.

5. Optimize seat utilization on Indian trains by providing comprehensive journey options and maximizing the utilization of vacant seats.

### Goals:

### Goal 1: Enable ticket booking for a direct journey from a specified source to destination.

### Goal 2: Enhance the Indian rail ticket reservation system to offer alternative options when there are no vacancies for a specific source to destination journey.

### Goal 3: Provide information on vacancies at prior or subsequent stations along the route, allowing users to consider alternative boarding points or destinations.

### Goal 4: Optimize seat utilization on Indian trains by providing comprehensive journey options and maximizing the utilization of vacant seats.

### Goal 5: Introduce functionality to suggest partial journeys with berth shifts, enabling users to combine multiple segments for a complete journey.

### Scope

IRCTC is a government-owned company that handles the online booking and catering services for Indian Railways. IRCTC enables passengers to book tickets, check availability, and track their trains' status online, making it easier and more convenient for individuals to plan their train journeys. If you encounter an emergency situation where reservations are not feasible and general tickets are not available, you have the option to book tickets while the train is running. This is a facility we offer to assist travelers in such circumstances

### Applications

* Online Ticket Booking
* Train Ticket Availability
* Emergency Ticket Booking

### Limitations

1. Limited availability of real-time data for vacancies at prior or subsequent stations.

2. Challenges in coordinating partial journeys and berth shifts within the existing train schedule.

3. Communication and passenger awareness of available choices may be challenging.

4. Operational complexities in managing catering, baggage transfer, and accommodating specific passenger needs.

5. Modifications required for ticketing and fare calculation process

## CHAPTER2

## LITERATURE SURVEY

### Collect Information

This study was done by considering the above mentioned views.

**1. Railway Reservation Systems:**

- We reviewed existing literature on railway reservation systems, focusing on their functionalities, challenges, and optimization strategies.

- Several studies provided valuable insights into user experience, system efficiency, and seat utilization in reservation systems.

**2. Alternative Journey Options:**

- Our literature review included research on providing alternative journey options in transportation systems, such as air travel or bus reservations.

- We explored studies that offered methods for suggesting alternative routes, boarding points, or destinations in reservation systems to accommodate passenger preferences.

**3. Partial Journeys and Berth Shifts:**

- In our literature survey, we examined the concept of partial journeys and berth shifts in railway systems.

- Various research papers provided insights into algorithms and approaches for suggesting and managing partial journeys and berth shifts in transportation systems.

**4. Passenger Preferences and Decision Making:**

- We delved into literature on passenger preferences and decision-making processes in transportation systems.

- Our review encompassed studies exploring factors influencing passenger choices, including travel time, convenience, amenities, and pricing.

**5. Case Studies and Implementations:**

- Additionally, we studied relevant case studies and implementations of similar enhancements in reservation systems.

- These case studies and research reports provided valuable insights into successful implementations, challenges encountered, and lessons learned.

By conducting a thorough literature review encompassing these key areas, we acquired a comprehensive understanding of existing research, best practices, and potential strategies. The insights gained from our literature survey have informed the development and implementation of our enhanced reservation system, ensuring that our project is grounded in established knowledge and addresses the identified limitation in the existing system.

### Summary

The literature review conducted for this project encompassed various aspects, including railway reservation systems, alternative journey options, partial journeys and berth shifts, passenger preferences, and case studies. The insights gained from the literature survey have informed the development and implementation of the enhanced reservation system, ensuring a well-informed approach to addressing the identified limitations.

## CHAPTER 3

EXISTING AND PROPOSED SYSTEMS

### Existing System

The IRCTC system allows users to search for available trains and seats based on various criteria, such as travel dates, source and destination stations, and class of travel. The system also provides users with real-time updates on seat availability, train schedules, and wait listed tickets. People can book their seats from desired source to destination.

#### Limitations:

When we want to make a train reservation, we need to do it directly one or two months in advance or else we have to take Tatkal 24 hours before the train starts. Apart from these two cases, we can't do the train reservation.

### Proposed System

If we have an emergency, we can use this system in a situation where we do not have the opportunity to make a reservation in the two cases mentioned above. That's how once the train is ready and the chart is prepared, then the Bhogi wise fully and partially booked seats details will come in that chart. In that we will also have the un booked partial vacancy seats details. we can provide the available vacancy seats details to user. So that user can book that seat instantly.

### Advantages

1. Emergency Booking Option

2. Real-Time Seat Availability

3. Instant Booking Facility

4. Optimized Seat Utilization:

### Software System Requirements

1. Operating system

2.Database

### Hardware Requirements

1. RAM :4gb

2.Harddisc

### 

## CHAPTER 4 SYSTEM IMPLEMENTATION

**4.1Implementation Tools**

### Python:

Python is a powerful, high-level programming language that is renowed for being easy to learn and understand. With libraries and frameworks for numerous applications, including web development, scientific computing, and artificial intelligence, it has a sizable and vibrant community.

### Visual Studio Code:

Visual studio code is the platform for wring the source code

**JAVASCRIPT:**

JavaScript has a wide range of uses and is primarily used for client-side web development. Here are some common uses of JavaScript allows developers to add interactivity, dynamic elements, and behavior to web pages. It enables features such as form validation, image sliders, interactive maps, drop-down menus, and more, enhancing the user experience.

### 4.2 Frameworks

### Flask:

The initial version of the well-known Python web framework Flask appeared in 2010. Its lightweight and modular design makes it simple to construct web apps fast and effectively.

Flask offers a straightforward and adaptable method of handling online requests and responses because it is built on the Werkzeug toolkit and the Jinja2 template engine. Additionally, it supports a wide variety of extensions for jobs like testing, database integration, and user authentication.

Built-in development server, support for secure cookies and sessions, and support for RESTful request handling are some of the features of Flask. Additionally, Flask has a sizable and vibrant developer community, which has sparked the development of numerous tools and extensions.

In general, Flask is a well-liked and adaptable Python web framework that works well for creating small to medium-sized online applications. It is simple tounderstand and use, whichmakes

it is a wonderful option for new users, and it is a favorite among seasoned developers thanks to its versatility and adaptability.

## CHAPTER 5 SOURCE CODE

PAGE -1

<!DOCTYPE html>

<html>

<head>

<title>IRCTC Form</title>

<style>

body {

background-image: url('static/4.jpg');

background-size: cover;

background-position: center;

background-repeat: no-repeat;

font-family: Arial, sans-serif;

}

.container {

margin: 150px auto;

max-width: 400px;

padding: 20px;

background-color: #fff;

border-radius: 10px;

margin-left: auto;

margin-right: 10%;

}

.form-group {

margin-bottom: 20px;

}

.form-group label {

display: block;

font-weight: bold;

margin-bottom: 5px;

}

.form-group select,

.form-group input[type="text"] {

width: 100%;

padding: 10px;

.form-group select,

.form-group input[type="text"] {

width: 100%;

padding: 10px;

border-radius: 5px;

border: 1px solid #ccc;

}

.submit-button {

width: 100%;

padding: 10px;

background-color: #4CAF50;

color: #fff;

border: none;

border-radius: 5px;

cursor: pointer;

}

.submit-button:hover {

background-color: #45a049;

}

</style>

</head>

<body>

<div class="container">

<h2>Book Ticket</h2>

<form id="myForm" action="/vacancylist2" method="post">

<div class="form-group">

<label for="station1">Source:</label>

<select id="station1" name="station1">

<option value="VISAKHAPATNAM (VSKP)">VISAKHAPATNAM (VSKP)</option>

<option value="DUVVADA (DVD)">DUVVADA (DVD)</option>

<option value="ANAKAPALLE (AKP)">ANAKAPALL E (AKP)</option>

<option value="TUNI (TUNI)">TUNI (TUNI)</option>

<option value="ANNAVARAM (ANV)">ANNAVARAM (ANV)</option>

<option value="SAMALKOT JN (SLO)">SAMALKOT JN (SLO)</option>

<option value="ANAPARTI (APT)">ANAPARTI (APT)</option>

<option value="RAJAHMUNDRY (RJY)">RAJAHMUNDRY (RJY)</option>

<option value="NIDADAVOLU JN (NDD)">NIDADAVOLU JN (NDD)</option>

<option value="TADEPALLIGUDEM (TDD)">TADEPALLIGUDEM (TDD)</option>

<option value="ELURU (EE)">ELURU (EE)</option>

<option value="NUZVID (NZD)">NUZVID (NZD)</option>

<option value="VIJAYAWADA JN (BZA)">VIJAYAWADA JN (BZA)</option>

</select>

</div>

<div class="form-group">

<label for="station2">Destination:</label>

<select id="station2" name="station2">

<option value="VISAKHAPATNAM (VSKP)">VISAKHAPATNAM (VSKP)</option>

<option value="DUVVADA (DVD)">DUVVADA (DVD)</option>

<option value="ANAKAPALLE (AKP)">ANAKAPALLE (AKP)</option>

<option value="TUNI (TUNI)">TUNI (TUNI)</option>

<option value="ANNAVARAM (ANV)">ANNAVARAM (ANV)</option>

<option value="SAMALKOT JN (SLO)">SAMALKOT JN (SLO)</option>

<option value="ANAPARTI (APT)">ANAPARTI (APT)</option>

<option value="RAJAHMUNDRY (RJY)">RAJAHMUNDRY (RJY)</option>

<option value="NIDADAVOLU JN (NDD)">NIDADAVOLU JN (NDD)</option>

<option value="TADEPALLIGUDEM (TDD)">TADEPALLIGUDEM (TDD)</option>

<option value="ELURU (EE)">ELURU (EE)</option>

<option value="NUZVID (NZD)">NUZVID (NZD)</option>

<option value="VIJAYAWADA JN (BZA)">VIJAYAWADA JN (BZA)</option>

</select>

</div>

<button class="submit-button" type="submit">Search</button>

</form>

</div>

</body>

</html>

OUTPUT:



fig no .1 Home page

**PAGE -2**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>IRCTC Reservation Chart</title>**

**<style>**

**/\*table, th, td {**

**border: 1px solid black;**

**}**

**/\* Apply basic styling to all tables \*/**

**table {**

**display: none;**

**}**

**/\* Display the active table \*/**

**.active-table {**

**display: table;**

**}**

**table {**

**width: 100%;**

**border-collapse: collapse;**

**}**

**/\* Style table headers \*/**

**th {**

**background-color: #1c3c5e;**

**color: white;**

**padding: 10px;**

**}**

**/\* Style table cells \*/**

**td {**

**padding: 10px;**

**border: 1px solid #ccc;**

**}**

**/\* Style the "Book Now" links \*/**

**a {**

**display: inline-block;**

**background-color: #1c3c5e;**

**color: white;**

**padding: 5px 10px;**

**text-decoration: none;**

**border-radius: 3px;**

**}**

**a:hover {**

**background-color: #154169;**

**}**

**body {**

**font-family: Arial, sans-serif;**

**margin: 0;**

**padding: 0;**

**justify-content: center;**

**align-items: center;**

**}**

**.disabled-label {**

**opacity: 0;**

**cursor: not-allowed;**

**}**

**/\* .button {**

**background-color: #1c3c5e;**

**color: white;**

**padding: 10px;**

**border: 20px solid #fff;**

**cursor: pointer;**

**font-size: 1em;**

**margin: 10px;**

**border-radius: 3px;**

**justify-content: center;**

**align-items: center;**

**} \*/**

**.button {**

**display: inline-block;**

**background-color: #003152; /\* Change this color to your preferred shade \*/**

**color: #fff;**

**padding: 10px 20px;**

**border: none;**

**border-radius: 3px;**

**font-size: 1em;**

**margin: 10px;**

**cursor: pointer;**

**transition: background-color 0.3s ease;**

**}**

**.button:hover {**

**background-color: red; /\* Change this color to your preferred hover shade \*/**

**}**

**.head {**

**text-align: center;**

**background-color: white;**

**}**

**#header {**

**background-color:#003152;**

**padding: 10px;**

**color: white;**

**text-align: center;**

**}**

**#nav {**

**background-color: #003152;**

**overflow: hidden;**

**display: flex;**

**justify-content: center;**

**align-items: center;**

**flex-wrap: wrap;**

**margin-top: 20px;**

**width: 100%;**

**}**

**#nav a {**

**float: left;**

**color: yellow;**

**text-align: center;**

**padding: 14px 16px;**

**text-decoration: none;**

**}**

**#nav a:hover {**

**background-color: #002d47;**

**}**

**#logo {**

**margin-top: 10px;**

**text-align: left;**

**height:100px;**

**width:90px;**

**position:absolute;**

**top:0;**

**left:0;**

**}**

**#irctc {**

**margin-top: 10px;**

**text-align: right;**

**height:100px;**

**width:90px;**

**position:absolute;**

**top:0;**

**right:0;**

**}**

**</style>**

**<script>**

**function toggleTable(tableId) {**

**var tables = document.getElementsByTagName("table");**

**// Hide all tables**

**for (var i = 0; i < tables.length; i++) {**

**tables[i].style.display = "none";**

**}**

**// Display the clicked table**

**var clickedTable = document.getElementById(tableId);**

**if (clickedTable) {**

**clickedTable.style.display = "table";**

**}**

**}**

**function gettable1() {**

**// Get the input values**

**//var station1 = document.getElementById("station1").value;**

**//var station2 = document.getElementById("station2").value;**

**// Validate input**

**var station1=document.getElementById("1").innerText;**

**var station2=document.getElementById("2").innerText;**

**if (station1 === "" || station2 === "") {**

**alert("Please enter both stations.");**

**return;**

**}**

**// Find intermediate stations**

**var stations=["VISAKHAPATNAM (VSKP)","DUVVADA (DVD)","ANAKAPALLE (AKP)","TUNI (TUNI)","ANNAVARAM (ANV)","SAMALKOT JN (SLO)","ANAPARTI (APT)","RAJAHMUNDRY (RJY)","NIDADAVOLU JN (NDD)","TADEPALLIGUDEM (TDD)","ELURU (EE)","NUZVID (NZD)","VIJAYAWADA JN (BZA)"] ;**

**var startIndex = stations.indexOf(station1);**

**var endIndex = stations.indexOf(station2);**

**console.log(station1);**

**console.log(startIndex);**

**// Display intermediate stations**

**if (startIndex === -1 || endIndex === -1) {**

**alert("Invalid station names. Please enter valid station names.");**

**} else if (startIndex >= endIndex) {**

**alert("The second station should be ahead of the first station.");**

**} else {**

**var intermediateStations = stations.slice(startIndex , endIndex);**

**var intermediateStationsText = intermediateStations.join(", ");**

**}**

**var table1 = document.getElementById("participantTable");**

**var table2 = document.getElementById("participantTable2");**

**var rows1 = table1.getElementsByTagName("tr");**

**var rows2 = table2.getElementsByTagName("tr");**

**for (var i = 1; i < rows1.length; i++) {**

**var sourceCell = rows1[i].getElementsByTagName("td")[0].textContent;**

**var destinationCell = rows1[i].getElementsByTagName("td")[1].textContent;**

**if (sourceCell === station1 && destinationCell === station2) {**

**rows1[i].style.display = "";**

**} else {**

**rows1[i].style.display = "none";**

**}**

**}**

**document.getElementById("participantTable").style.display = "table";**

**}**

**function gettable2()**

**{**

**var table2 = document.getElementById("participantTable2");**

**var rows2 = table2.getElementsByTagName("tr");**

**for (var i = 1; i < rows2.length; i++) {**

**var station1=document.getElementById("1").innerText;**

**var station2=document.getElementById("2").innerText;**

**var sourceCell = rows2[i].getElementsByTagName("td")[0].textContent;**

**var destinationCell = rows2[i].getElementsByTagName("td")[1].textContent;**

**var stations=["VISAKHAPATNAM (VSKP)","DUVVADA (DVD)","ANAKAPALLE (AKP)","TUNI (TUNI)","ANNAVARAM (ANV)","SAMALKOT JN (SLO)","ANAPARTI (APT)","RAJAHMUNDRY (RJY)","NIDADAVOLU JN (NDD)","TADEPALLIGUDEM (TDD)","ELURU (EE)","NUZVID (NZD)","VIJAYAWADA JN (BZA)"] ;**

**var startIndex = stations.indexOf(station1);**

**var endIndex = stations.indexOf(station2);**

**var intermediateStations = stations.slice(startIndex , endIndex);**

**console.log(intermediateStations.includes(sourceCell));**

**console.log(sourceCell,intermediateStations);**

**console.log(rows2[i]);**

**if (intermediateStations.includes(sourceCell)) {**

**rows2[i].style.display = "";**

**} else {**

**rows2[i].style.display = "none";**

**}**

**}**

**document.getElementById("participantTable").style.display = "table";**

**}**

**function gettable3()**

**{**

**var table3 = document.getElementById("participantTable3");**

**var rows3 = table3.getElementsByTagName("tr");**

**for (var i = 1; i < rows3.length; i++) {**

**var station1=document.getElementById("1").innerText;**

**console.log(station1);**

**var station2=document.getElementById("2").innerText;**

**var sourceCell = rows3[i].getElementsByTagName("td")[0].textContent;**

**var destinationCell = rows3[i].getElementsByTagName("td")[1].textContent;**

**var stations=["VISAKHAPATNAM (VSKP)","DUVVADA (DVD)","ANAKAPALLE (AKP)","TUNI (TUNI)","ANNAVARAM (ANV)","SAMALKOT JN (SLO)","ANAPARTI (APT)","RAJAHMUNDRY (RJY)","NIDADAVOLU JN (NDD)","TADEPALLIGUDEM (TDD)","ELURU (EE)","NUZVID (NZD)","VIJAYAWADA JN (BZA)"] ;**

**var startIndex = stations.indexOf(station1);**

**var endIndex = stations.indexOf(station2);**

**var intermediateStations = stations.slice( 0, startIndex);**

**console.log(intermediateStations,sourceCell);**

**console.log(intermediateStations.includes(sourceCell));**

**console.log(sourceCell,intermediateStations);**

**console.log(rows3[i]);**

**if (intermediateStations.includes(sourceCell)) {**

**rows3[i].style.display = "";**

**} else {**

**rows3[i].style.display = "none";**

**}**

**}**

**document.getElementById("participantTable").style.display = "table";**

**}**

**function gettable4()**

**{ var stations=["VISAKHAPATNAM (VSKP)","DUVVADA (DVD)","ANAKAPALLE (AKP)","TUNI (TUNI)","ANNAVARAM (ANV)","SAMALKOT JN (SLO)","ANAPARTI (APT)","RAJAHMUNDRY (RJY)","NIDADAVOLU JN (NDD)","TADEPALLIGUDEM (TDD)","ELURU (EE)","NUZVID (NZD)","VIJAYAWADA JN (BZA)"] ;**

**var table4 = document.getElementById("participantTable4");**

**var rows4 = table4.getElementsByTagName("tr");**

**for (var i = 1; i < rows4.length; i++) {**

**//var station1= rows4[i].getElementsByTagName("td")[0].textContent;**

**//var station2 = rows4[i].getElementsByTagName("td")[1].textContent;**

**var station1=document.getElementById("1").innerText;**

**var station2=document.getElementById("2").innerText;**

**var sourceCell = rows4[i].getElementsByTagName("td")[0].textContent;**

**var destinationCell = rows4[i].getElementsByTagName("td")[1].textContent;**

**var startIndex = stations.indexOf(station1);**

**var endIndex = stations.indexOf(station2);**

**var intermediateStations = stations.slice(startIndex , endIndex);**

**console.log(intermediateStations.includes(sourceCell));**

**console.log(sourceCell,intermediateStations);**

**var intermediateStations = stations.slice(startIndex+1 , endIndex);**

**if (intermediateStations.includes(destinationCell)) {**

**rows4[i].style.display = "";**

**} else {**

**rows4[i].style.display = "none";**

**}**

**}**

**}**

**function gettable5()**

**{**

**var stations=["VISAKHAPATNAM (VSKP)","DUVVADA (DVD)","ANAKAPALLE (AKP)","TUNI (TUNI)","ANNAVARAM (ANV)","SAMALKOT JN (SLO)","ANAPARTI (APT)","RAJAHMUNDRY (RJY)","NIDADAVOLU JN (NDD)","TADEPALLIGUDEM (TDD)","ELURU (EE)","NUZVID (NZD)","VIJAYAWADA JN (BZA)"] ;**

**//var source = document.getElementById("station1").value.toLowerCase();**

**//var destination = document.getElementById("station2").value.toLowerCase();**

**var station1=document.getElementById("1").innerText;**

**var station2=document.getElementById("2").innerText;**

**var startIndex = stations.indexOf(station1);**

**var endIndex = stations.indexOf(station2);**

**var intermediateStations = stations.slice(startIndex , endIndex);**

**var table5 = document.getElementById("participantTable5");**

**var rows5 = table5.getElementsByTagName("tr");**

**for (var i = 1; i < rows5.length; i++) {**

**var sourceCell = rows5[i].getElementsByTagName("td")[0].textContent;**

**var destinationCell = rows5[i].getElementsByTagName("td")[1].textContent;**

**var display = false;**

**console.log(station1,station2);**

**console.log(sourceCell,destinationCell);**

**if (station1 && station2) {**

**if ((sourceCell === station1 && destinationCell !== station2) || (sourceCell !== station1 && destinationCell === station2)) {**

**display = true;**

**}**

**}**

**if (display) {**

**rows5[i].style.display = "";**

**} else {**

**rows5[i].style.display = "none";**

**}**

**}**

**document.getElementById("participantTable").style.display = "table";**

**}**

**function gettable6() {**

**var fromStation=document.getElementById("1").innerText;**

**var toStation=document.getElementById("2").innerText;**

**if (fromStation === "" || toStation === "") {**

**alert("Please enter both stations.");**

**return;**

**}**

**// Find intermediate stations**

**var stations=["VISAKHAPATNAM (VSKP)","DUVVADA (DVD)","ANAKAPALLE (AKP)","TUNI (TUNI)","ANNAVARAM (ANV)","SAMALKOT JN (SLO)","ANAPARTI (APT)","RAJAHMUNDRY (RJY)","NIDADAVOLU JN (NDD)","TADEPALLIGUDEM (TDD)","ELURU (EE)","NUZVID (NZD)","VIJAYAWADA JN (BZA)"] ;**

**var startIndex = stations.indexOf(fromStation);**

**var endIndex = stations.indexOf(toStation);**

**console.log(startIndex);**

**console.log(endIndex);**

**var table = document.getElementById("participantTable6");**

**var rows = table.getElementsByTagName("tr");**

**var max = 0;**

**var row=null;**

**var result=[];**

**var x=0;**

**console.log(result);**

**for (var j =startIndex ; j <= endIndex; j++)**

**{**

**for (var i = 1; i < rows.length; i++)**

**{**

**var sourceCell = rows[i].getElementsByTagName("td")[0].textContent;**

**var destinationCell = rows[i].getElementsByTagName("td")[1].textContent;**

**//console.log(sourceCell,destinationCell)**

**//case 1**

**//if (sourceCell === fromStation && destinationCell === toStation){**

**//rows[i].style.display = "";**

**//console.log("direct match found" +rows[i]);**

**//return;**

**//}**

**if ( sourceCell === fromStation && stations.indexOf(destinationCell) >= max && stations.indexOf(destinationCell) <= endIndex+2 ){**

**//console.log("else");**

**max = stations.indexOf(destinationCell);**

**row = rows[i];**

**rows[i].style.display="none";**

**}**

**else{**

**//console.log("else2")**

**//console.log(sourceCell,destinationCell);**

**//console.log(fromStation,toStation);**

**//console.log(stations.indexOf(destinationCell),max);**

**//console.log(rows[i]);**

**rows[i].style.display="none";**

**}**

**}**

**console.log("row is:");**

**console.log(row);**

**if (row && !result.includes(row)) {**

**//console.log(row);**

**result.push(row);**

**//console.log(result);**

**//row.style.display = "";**

**startIndex=stations.indexOf(row.getElementsByTagName("td")[1].textContent);**

**fromStation=row.getElementsByTagName("td")[1].textContent;**

**row=null;**

**//document.getElementById("dest").textContent = row.getElementsByTagName("td")[1].textContent;**

**}**

**}**

**console.log(result.length);**

**for (var n=0; n<result.length; n++){**

**console.log("result");**

**console.log(result[n]);**

**console.log(n);**

**result[n].style.display="";**

**}**

**console.log(result);**

**document.getElementById("participantTable").style.display = "table";**

**}**

**</script>**

**</head>**

**<body>**

**<div id="header">**

**<h1 >IRCTC Reservation Chart</h1>**

**</div>**

**<!--<label id="1" disabled >{{ from\_address }}</label>**

**<label id="2" >{{ to\_address }}</label>-->**

**<label id="1" class="disabled-label">{{ from\_address }}</label>**

**<label id="2" class="disabled-label">{{ to\_address }}</label>**

**<div id="nav">**

**<!--<a href="#participantTable1">-->**

**<button class="button" onclick="gettable1();toggleTable('participantTable')">Direct source-destination</button>**

**</a>**

**<!--<a href="#participantTable2">-->**

**<button class="button" onclick="gettable2();toggleTable('participantTable2')">After source</button>**

**</a>**

**<!--<a href="#participantTable3">-->**

**<button class="button" onclick="gettable3();toggleTable('participantTable3')">Before Source-destination</button>**

**</a>**

**<!--<a href="#participantTable4">**

**<button class="button" onclick="gettable4()">Find table4</button>**

**</a>-->**

**<!--<a href="#participantTable5">-->**

**<button class="button" onclick="gettable5();toggleTable('participantTable5')">Start\_souce or ends\_destination</button>**

**</a>**

**<!--<a href="#participantTable6">-->**

**<button class="button" onclick="gettable6();toggleTable('participantTable6')">Berth shift</button>**

**</a>**

**</div>**

**<div id="logo">**

**<img src="static\logo1.jpg" height="80" width="90" alt="logo">**

**</div>**

**<div id="irctc">**

**<img src="static\irctc.jpg" height="80" width="90" alt="irctc">**

**</div>**

**<br>**

**<table id="participantTable" style="width: 100%">**

**<tr>**

**<th>Source</th>**

**<th>Destination</th>**

**<th>Coach</th>**

**<th>Berth</th>**

**<th>Seat</th>**

**<th>Book Now</th>**

**</tr>**

**{% for vacancy in data %}**

**<tr>**

**<td>{{ vacancy[0] }}</td>**

**<td>{{ vacancy[1] }}</td>**

**<td>{{ vacancy[2] }}</td>**

**<td>{{ vacancy[3] }}</td>**

**<td>{{ vacancy[4] }}</td>**

**<td>**

**<!--<a href="#" onclick="navigateToPageAndRemoveRow('index.html', this); return false;">Book Now</a> -->**

**<!--<a href="{{ url\_for('file') }}">Go to Page 2</a>-->**

**<a href="{{ url\_for('file', Berth=vacancy[3],Coach=vacancy[2]) }}">Book Now</a>**

**<!--<a href="templates/file.html">Go to Page 2</a>-->**

**<!--<a href="{{ url\_for('file', data=vacancy[3]) }}">Go to Page 2</a> -->**

**</td>**

**</tr>**

**{% endfor %}**

**</table>**

**<br>**

**<table id="participantTable2" style="width: 100%">**

**<tr>**

**<th>Source</th>**

**<th>Destination</th>**

**<th>Coach</th>**

**<th>Berth</th>**

**<th>Seat</th>**

**<th>Book Now</th>**

**</tr>**

**{% for vacancy in data %}**

**<tr>**

**<td>{{ vacancy[0] }}</td>**

**<td>{{ vacancy[1] }}</td>**

**<td>{{ vacancy[2] }}</td>**

**<td>{{ vacancy[3] }}</td>**

**<td>{{ vacancy[4] }}</td>**

**<td>**

**<!--<a href="#" onclick="navigateToPageAndRemoveRow('index.html', this); return false;">Book Now</a> -->**

**<!--<a href="{{ url\_for('file') }}">Go to Page 2</a>-->**

**<a href="{{ url\_for('file', Berth=vacancy[3],Coach=vacancy[2]) }}">Book Now</a>**

**<!--<a href="templates/file.html">Go to Page 2</a>-->**

**<!--<a href="{{ url\_for('file', data=vacancy[3]) }}">Go to Page 2</a> -->**

**</td>**

**</tr>**

**{% endfor %}**

**</table>**

**<br>**

**<table id="participantTable3" style="width: 100%">**

**<tr>**

**<th>Source</th>**

**<th>Destination</th>**

**<th>Coach</th>**

**<th>Berth</th>**

**<th>Seat</th>**

**<th>Book Now</th>**

**</tr>**

**{% for vacancy in data %}**

**<tr>**

**<td>{{ vacancy[0] }}</td>**

**<td>{{ vacancy[1] }}</td>**

**<td>{{ vacancy[2] }}</td>**

**<td>{{ vacancy[3] }}</td>**

**<td>{{ vacancy[4] }}</td>**

**<td>**

**<!--<a href="#" onclick="navigateToPageAndRemoveRow('index.html', this); return false;">Book Now</a> -->**

**<!--<a href="{{ url\_for('file') }}">Go to Page 2</a>-->**

**<a href="{{ url\_for('file', Berth=vacancy[3],Coach=vacancy[2]) }}">Book Now</a>**

**<!--<a href="templates/file.html">Go to Page 2</a>-->**

**<!--<a href="{{ url\_for('file', data=vacancy[3]) }}">Go to Page 2</a> -->**

**</td>**

**</tr>**

**{% endfor %}**

**</table>**

**<br>**

**<table id="participantTable4" style="width: 100%">**

**<tr>**

**<th>Source</th>**

**<th>Destination</th>**

**<th>Coach</th>**

**<th>Berth</th>**

**<th>Seat</th>**

**<th>Book Now</th>**

**</tr>**

**{% for vacancy in data %}**

**<tr>**

**<td>{{ vacancy[0] }}</td>**

**<td>{{ vacancy[1] }}</td>**

**<td>{{ vacancy[2] }}</td>**

**<td>{{ vacancy[3] }}</td>**

**<td>{{ vacancy[4] }}</td>**

**<td>**

**<!--<a href="#" onclick="navigateToPageAndRemoveRow('index.html', this); return false;">Book Now</a> -->**

**<!--<a href="{{ url\_for('file') }}">Go to Page 2</a>-->**

**<a href="{{ url\_for('file', Berth=vacancy[3],Coach=vacancy[2]) }}">Book Now</a>**

**<!--<a href="templates/file.html">Go to Page 2</a>-->**

**<!--<a href="{{ url\_for('file', data=vacancy[3]) }}">Go to Page 2</a> -->**

**</td>**

**</tr>**

**{% endfor %}**

**</table>**

**<br>**

**<table id="participantTable5" style="width: 100%">**

**<tr>**

**<th>Source</th>**

**<th>Destination</th>**

**<th>Coach</th>**

**<th>Berth</th>**

**<th>Seat</th>**

**<th>Book Now</th>**

**</tr>**

**{% for vacancy in data %}**

**<tr>**

**<td>{{ vacancy[0] }}</td>**

**<td>{{ vacancy[1] }}</td>**

**<td>{{ vacancy[2] }}</td>**

**<td>{{ vacancy[3] }}</td>**

**<td>{{ vacancy[4] }}</td>**

**<td>**

**<!--<a href="#" onclick="navigateToPageAndRemoveRow('index.html', this); return false;">Book Now</a> -->**

**<!--<a href="{{ url\_for('file') }}">Go to Page 2</a>-->**

**<a href="{{ url\_for('file', Berth=vacancy[3],Coach=vacancy[2]) }}">Book Now</a>**

**<!--<a href="templates/file.html">Go to Page 2</a>-->**

**<!--<a href="{{ url\_for('file', data=vacancy[3]) }}">Go to Page 2</a> -->**

**</td>**

**</tr>**

**{% endfor %}**

**</table>**

**<br>**

**<table id="participantTable6" style="width: 100%">**

**<tr>**

**<th>Source</th>**

**<th>Destination</th>**

**<th>Coach</th>**

**<th>Berth</th>**

**<th>Seat</th>**

**<th>Book Now</th>**

**</tr>**

**{% for vacancy in data %}**

**<tr>**

**<td>{{ vacancy[0] }}</td>**

**<td>{{ vacancy[1] }}</td>**

**<td>{{ vacancy[2] }}</td>**

**<td>{{ vacancy[3] }}</td>**

**<td>{{ vacancy[4] }}</td>**

**<td>**

**<!--<a href="#" onclick="navigateToPageAndRemoveRow('index.html', this); return false;">Book Now</a> -->**

**<!--<a href="{{ url\_for('file') }}">Go to Page 2</a>-->**

**<a href="{{ url\_for('file', Berth=vacancy[3],Coach=vacancy[2]) }}">Book Now</a>**

**<!--<a href="templates/file.html">Go to Page 2</a>-->**

**<!--<a href="{{ url\_for('file', data=vacancy[3]) }}">Go to Page 2</a> -->**

**</td>**

**</tr>**

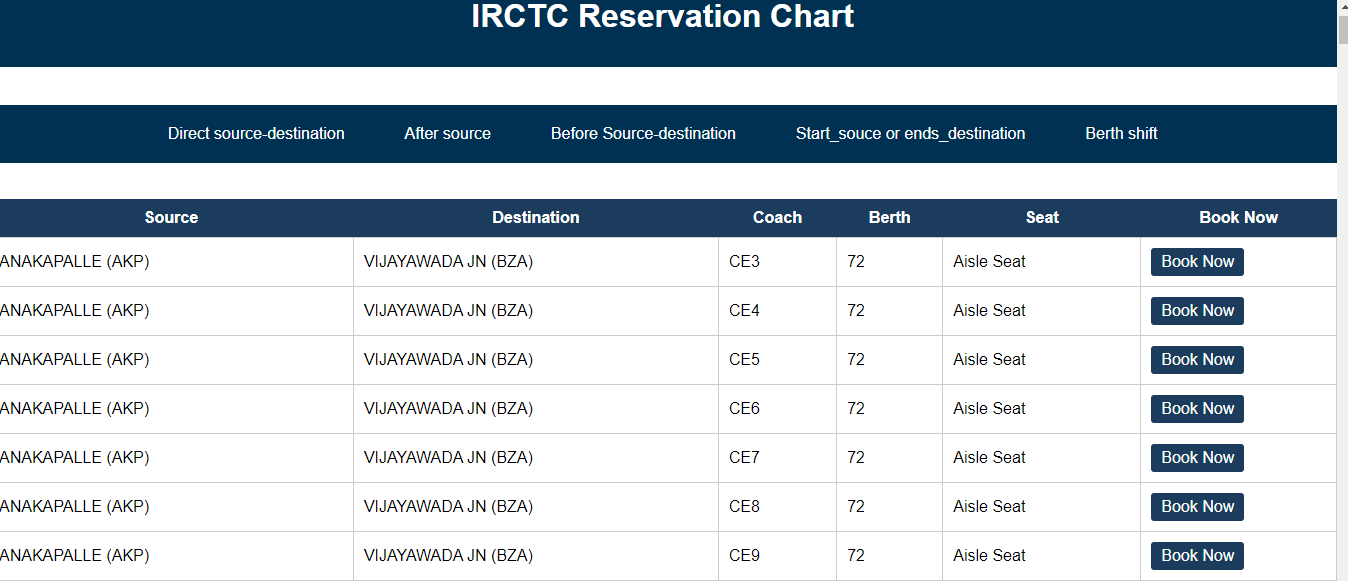
**{% endfor %}**

**</table>**

**</body>**

**</html>**

**OUTPUT:**

****

**Fig no .2 Check Availability page**

**PAGE -3**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Booking Page</title>**

**<style>**

**body {**

**font-family: Arial, sans-serif;**

**background-image: url('static/train.jpg');**

**background-size: cover;**

**background-position: center;**

**background-repeat: no-repeat;**

**}**

**.container {**

**margin: 50px auto;**

**max-width: 400px;**

**padding: 20px;**

**background-color: #f0f0f0;**

**border-radius: 10px;**

**}**

**.passenger-details {**

**margin-bottom: 20px;**

**}**

**.passenger-details label {**

**display: block;**

**font-weight: bold;**

**margin-bottom: 5px;**

**}**

**.passenger-details input[type="text"],**

**.passenger-details input[type="number"] {**

**width: 100%;**

**padding: 10px;**

**margin-bottom: 10px;**

**border-radius: 5px;**

**border: 1px solid #ccc;**

**box-sizing: border-box; /\* Ensures padding and border are included in the width \*/**

**}**

**#passenger-details-container {**

**margin-bottom: 20px;**

**}**

**#passenger-details-container .passenger-details {**

**border: 1px solid #ccc;**

**padding: 10px;**

**border-radius: 5px;**

**background-color: #fff;**

**}**

**.submit-button {**

**width: 100%;**

**padding: 15px;**

**margin-top: 20px;**

**background-color: #4CAF50;**

**color: #fff;**

**border: none;**

**border-radius: 5px;**

**cursor: pointer;**

**}**

**.submit-button:hover {**

**background-color: #45a049;**

**}**

**</style>**

**</head>**

**<body>**

**<div class="container">**

**<h2>Booking Page</h2>**

**<form id="booking-form" action="/last" method="POST" onsubmit="return validateForm(event)">**

**<div id="passenger-details-container">**

**<div class="passenger-details">**

**<label for="age">Age:</label>**

**<input type="number" id="age" name="age" placeholder="Enter Age">**

**<span id="age-error" class="error"></span>**

**</div>**

**<div class="passenger-details">**

**<label for="name">Name:</label>**

**<input type="text" id="name" name="name" placeholder="Enter Name">**

**<span id="name-error" class="error"></span>**

**</div>**

**<div class="passenger-details">**

**<label for="email">Email:</label>**

**<input type="text" id="email" name="email" placeholder="Enter Email">**

**<span id="email-error" class="error"></span>**

**</div>**

**<div class="passenger-details">**

**<label for="mobile">Mobile:</label>**

**<input type="text" id="mobile" name="mobile" placeholder="Enter Mobile Number">**

**<span id="mobile-error" class="error"></span>**

**</div>**

**</div>**

**<button class="submit-button" type="submit">Submit</button>**

**</form>**

**</div>**

**<script>**

**function validateForm(event) {**

**event.preventDefault(); // Prevent form submission**

**// Clear previous error messages**

**clearErrors();**

**// Get form inputs**

**var age = document.getElementById("age").value;**

**var name = document.getElementById("name").value;**

**var email = document.getElementById("email").value;**

**var mobile = document.getElementById("mobile").value;**

**// Validate age**

**if (age === "") {**

**displayError("age-error", "Age is required");**

**return false; // Stop form submission**

**} else if (isNaN(age) || age < 1) {**

**displayError("age-error", "Age must be a positive number");**

**return false; // Stop form submission**

**}**

**// Validate name**

**if (name === "") {**

**displayError("name-error", "Name is required");**

**return false; // Stop form submission**

**}**

**// Validate email**

**if (email === "") {**

**displayError("email-error", "Email is required");**

**return false; // Stop form submission**

**} else if (!isValidEmail(email)) {**

**displayError("email-error", "Invalid email format");**

**return false; // Stop form submission**

**}**

**// Validate mobile**

**if (mobile === "") {**

**displayError("mobile-error", "Mobile number is required");**

**return false; // Stop form submission**

**} else if (!isValidMobile(mobile)) {**

**displayError("mobile-error", "Invalid mobile number format");**

**return false; // Stop form submission**

**}**

**// If form is valid, proceed with form submission**

**document.getElementById("booking-form").submit();**

**}**

**function isValidEmail(email) {**

**// Basic email format validation using regular expression**

**var emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;**

**return emailPattern.test(email);**

**}**

**function isValidMobile(mobile) {**

**// Basic mobile number format validation using regular expression**

**var mobilePattern = /^[0-9]{10}$/;**

**return mobilePattern.test(mobile);**

**}**

**function displayError(elementId, message) {**

**var errorElement = document.getElementById(elementId);**

**errorElement.textContent = message;**

**}**

**function clearErrors() {**

**var errorElements = document.getElementsByClassName("error");**

**for (var i = 0; i < errorElements.length; i++) {**

**errorElements[i].textContent = "";**

**}**

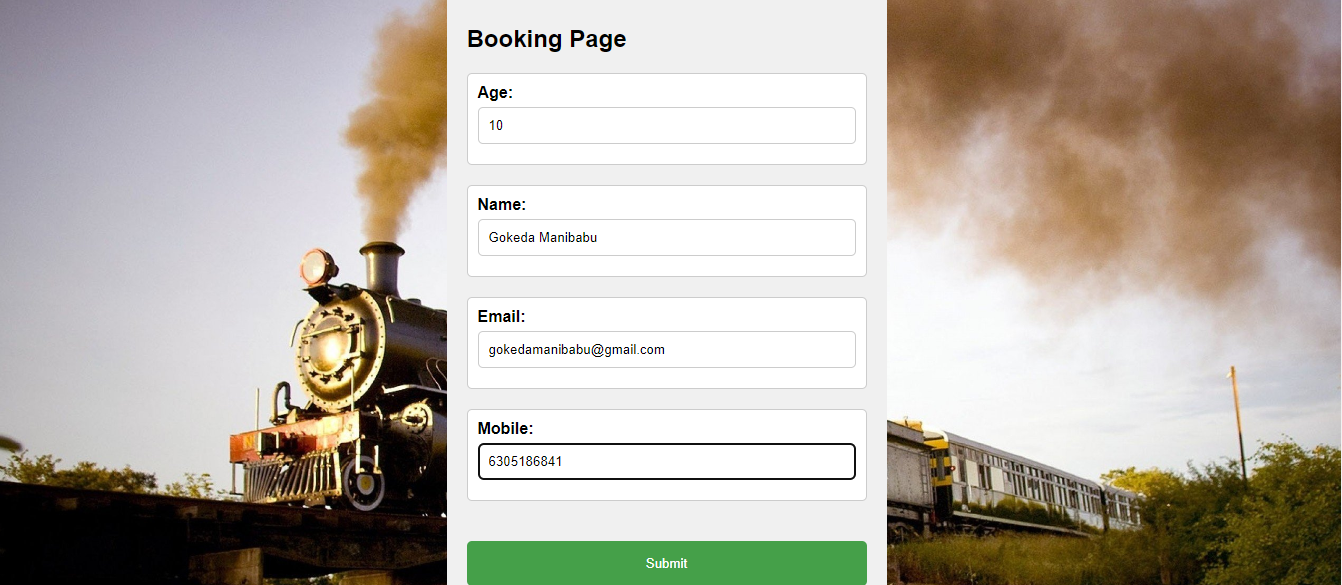
**}**

**</script>**

**</body>**

**</html>**

**OUTPUT:**

****

**Fig no. 3 Taking Details from user**

**API GENERATION AND BACKEND CONNECTION**

from flask import Flask, render\_template, request,redirect

from flask import url\_for

import pandas as pd

import sqlite3

app = Flask(\_\_name\_\_)

@app.route('/', methods=['GET', 'POST'])

def index():

if request.method == 'POST':

# Get the form data

from\_address = request.form.get('station1')

to\_address = request.form.get('station2')

train\_name = request.form.get('train-name')

# Perform any necessary processing with the form data

# ...

# Redirect to the "join.html" page with the form data

return redirect('/vacancylist2?from={}&to={}&train\_name={}'.format(from\_address, to\_address, train\_name))

return render\_template('irctc2.html')

stations=["VISHAKAPARNAM(VSKP)","DUVVADA(DVD)","ANAKAPALLE(AKP)","TUNI(TUNI)","ANNAVARAM(ANV)","SAMARLAKOT JN(SLO)","ANAPARTI(APT)","RAJAMUNDRY(RJY)","NIDADAVOLU JN(NDD)","TADEPALLIGUDEM(TDD)","ELURU(EE)","NUZVID(NZD)","VIJAYAWADA JN(BZA)"]

connect = sqlite3.connect('database2.db')

df = pd.read\_excel('IRCTC 4.xlsx')

#df1 = pd.read\_excel('IRCTC 4.xlsx')

df.to\_sql('vacancy', connect, if\_exists='replace', index=False)

#df1.to\_sql('list', connect, if\_exists='replace', index=False)

connect.execute(

'CREATE TABLE IF NOT EXISTS VACANCY1 (source TEXT, \

destination TEXT, coach TEXT, berth TEXT)')

##participants chart

@app.route('/irctc2', methods=['GET', 'POST'])

def irctc2():

return render\_template('irctc2.html')

@app.route('/vacancylist2', methods=['GET', 'POST'])

def vacancylis2():

if request.method == 'POST':

# Retrieve the form data from the request

from\_address = request.form.get('station1')

to\_address = request.form.get('station2')

train\_name = request.form.get('train-name')

else:

# Retrieve the form data from the query parameters

from\_address = request.args.get('station1')

to\_address = request.args.get('station2')

train\_name = request.args.get('train\_name')

connect = sqlite3.connect('database2.db')

cursor = connect.cursor()

cursor.execute('SELECT \* FROM vacancy')

data = cursor.fetchall()

# Pass the form data to the "join.html" template

return render\_template('vacancylist2.html', from\_address=from\_address, to\_address=to\_address, train\_name=train\_name,data=data)

@app.route('/last', methods=['GET', 'POST'])

def last():

if request.method == 'POST':

# Retrieve the form data from the request

name = request.form.get('name')

else:

# Retrieve the form data from the query parameters

name = request.args.get('name')

return render\_template("last.html",name=name)

@app.route('/vacancylist')

def vacancylist():

connect = sqlite3.connect('database2.db')

cursor = connect.cursor()

cursor.execute('SELECT \* FROM vacancy')

data = cursor.fetchall()

return render\_template("vacancylist.html", data=data)

@app.route('/file')

def file():

Berth = request.args.get('Berth')

Coach = request.args.get('Coach')

connect = sqlite3.connect('database2.db')

cursor = connect.cursor()

cursor.execute("SELECT \* FROM vacancy")

columns = [description[0] for description in cursor.description] # Get column names

data = cursor.fetchall() # Fetch all rows from the table

with sqlite3.connect('database2.db') as connect:

cursor = connect.cursor()

cursor.execute("DELETE FROM list WHERE Berth = ? AND Coach = ?", (Berth, Coach))

connect.commit()

cursor.close()

connect.close()

berth\_values = [row[columns.index('Berth')] for row in data]

return render\_template("book1.html", columns=columns,berth\_values=berth\_values)

Page 4

if \_\_name\_\_ == '\_\_main\_\_':

#app.run(debug=False)

app.run(debug=True)

<!DOCTYPE html>

<html>

<head>

<title>Thanks for Booking</title>

<style>

body {

margin: 0;

padding: 0;

background-image: url("static/3.jpg");

background-size: cover;

background-position: center;

font-family: Arial, sans-serif;

display: flex;

align-items: center;

justify-content: center;

min-height: 100vh;

}

.container {

display: grid;

grid-template-columns: 1fr;

grid-template-rows: auto auto;

gap: 20px;

background-color: rgba(255, 255, 255, 0.8);

padding: 30px;

border-radius: 10px;

text-align: center;

}

h1 {

font-size: 32px;

font-weight: bold;

color: #4CAF50;

margin: 0;

}

p {

font-size: 18px;

line-height: 1.5;

margin: 0;

}

</style>

</head>

<body>

<div class="container">

<h1> Dear {{name}}</h1>

<h1>Thanks for Booking!</h1>

<p>Your train ticket has been successfully booked.</p>

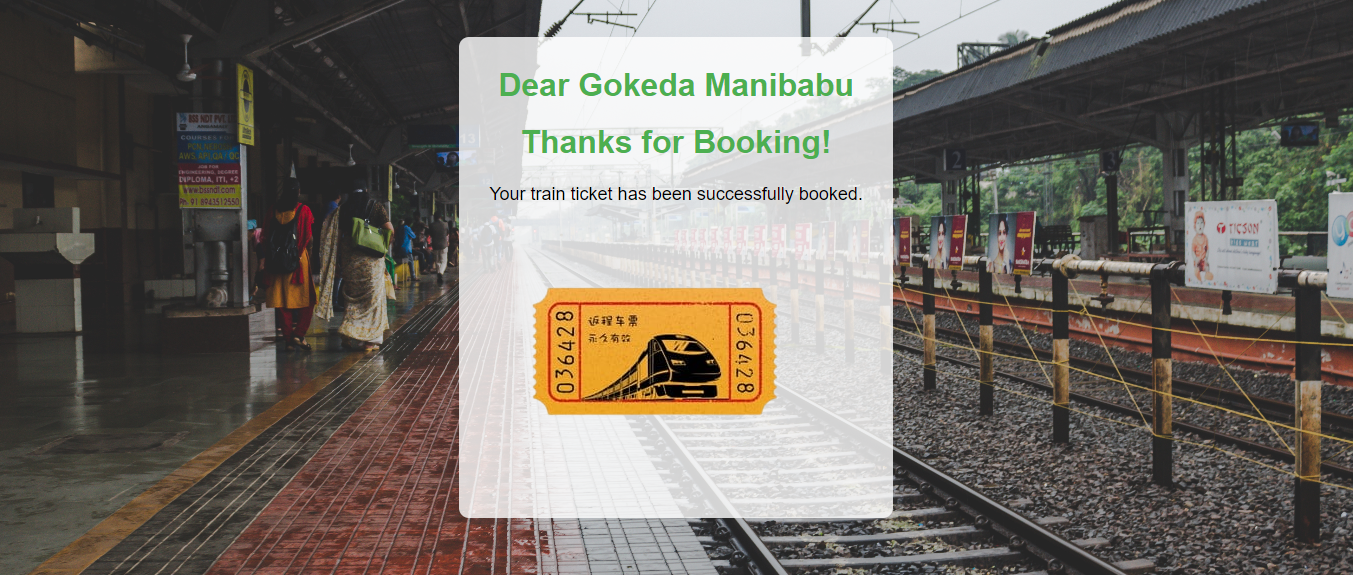
<img class="ticket-image" src="static/ticket1.png" alt="Ticket Image">

</div>

</body>

</html>

OUTPUT:



**Fig no.4 Confirmation**

CHAPTER 6

## CONCLUSION

## Finally,

The problem statement will be resolved with ticket reservation system that enables passengers to book and manage train tickets efficiently. Overall, the IRCTC online current reservation system project is a critical initiative that supports the Indian Railways' efforts to improve passenger experience and streamline ticketing processes. The successful implementation of the project will have significant benefits for the railway industry, passengers, and the country as a whole.

## 

## CHAPTER 7

**FUTURE ENHANCEMENTS**

Develop a dedicated mobile application for the ticket reservation system. This would allow passengers to easily book and manage their train tickets using their smart phones, providing greater convenience and accessibility.

Implement real-time updates for train schedules, availability, and ticket status. Passengers will be able to receive instant notifications about any changes or updates related to their booked tickets, such as delays, cancellations, or platform changes. Enhance the system to allow passengers to choose their preferred seats during the ticket booking process. This feature could include seat maps, indicating available seats and allowing passengers to select seats according to their preferences, such as window seats, aisle seats, or seats in specific coach classes.

## CHAPTER 8

### References:

We have referenced from many a sites to get information/ for knowledge gathering to understand the current scenario of the reservation system, below are the references we have got help from, and we acknowledge the same.

1) <https://www.scribd.com/document/424451030/Railway-Reservation-System>

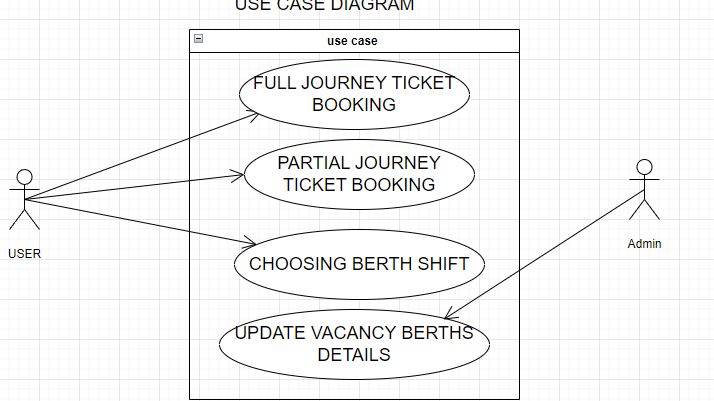
2)<https://en.wikipedia.org/wiki/Indian_Railway_Catering_and_Tourism_Corporation>

3) <https://www.slideshare.net/KOYELMAJUMDAR1/railway-reservation-system-249646777>

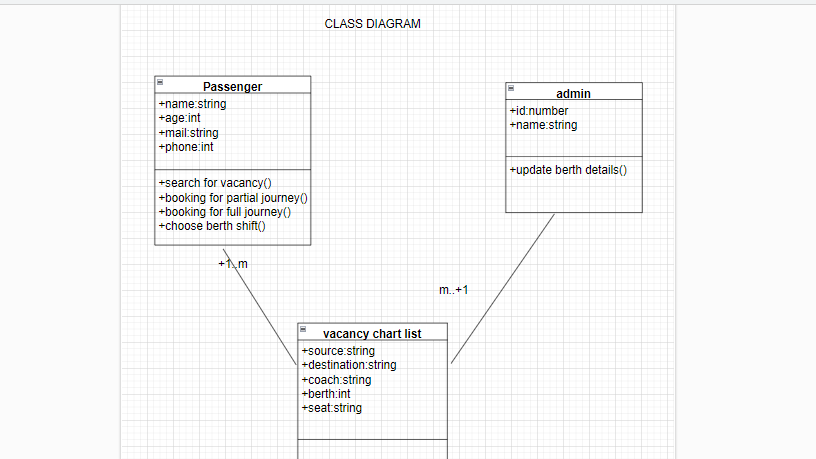
**CHAPTER 9**

**LIST OF FIGURES**

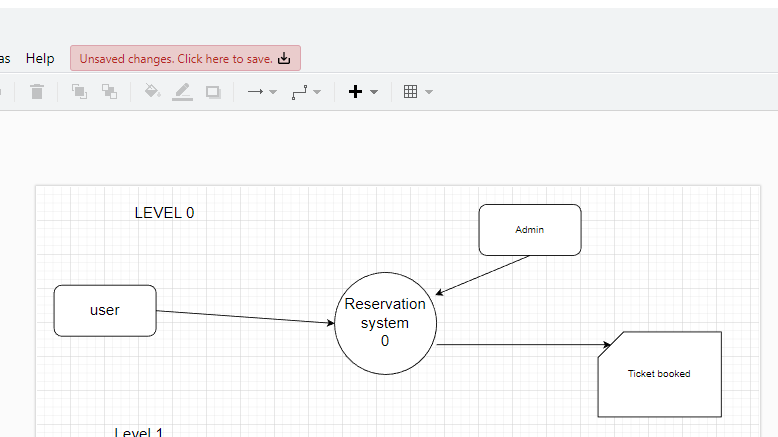
Uml diagrams



class diagram



DFD DIAGRAMS



level 1

