# TRAIN TICKET BOOKING APPLICATION

# INTRODUCTION

The Train Ticket Booking System presents a streamlined solution for travelers to efficiently book train tickets and for administrators to manage train schedules and bookings. With its intuitive interface and comprehensive functionalities, this project consists of two modules aimed at simplifying the train booking process. Administrators play a pivotal role in the system, with the ability to register and log in, subsequently adding locations and scheduling trains with ease. By selecting source and destination locations and specifying dates, times, and fare prices, administrators ensure a seamless booking experience for customers.

For travelers, the Train Ticket Booking System offers a user-friendly platform to search for scheduled trains based on their preferred locations and view available tickets. With transparent access to train schedules and ticket availability, customers can make informed decisions and book tickets effortlessly, paying the fare price securely online. Meanwhile, administrators maintain oversight by monitoring all booked trains, ensuring efficient management of resources and providing timely support to customers as needed.

In conclusion, the Train Ticket Booking System represents a modern approach to train ticketing, empowering both administrators and customers with efficient tools for managing and booking train tickets. By centralizing train schedules, ticket bookings, and customer information, this system streamlines operations, enhances transparency, and ultimately improves the overall experience for travelers and administrators alike.

# PROBLEM STATEMENT

The Train Ticket Booking System addresses several key challenges encountered in traditional train ticketing processes. Firstly, manual booking systems often lead to inefficiencies and errors, causing delays and inconvenience for travelers. With the absence of a centralized platform, administrators face challenges in managing train schedules and ticket availability, resulting in potential overbooking or underutilization of resources.

Secondly, the lack of transparency in ticket availability and booking processes can frustrate travelers and lead to dissatisfaction with the service. Without access to real-time information on train schedules and ticket availability, customers may encounter difficulties in planning their journeys effectively, leading to missed opportunities or last-minute rush bookings.

Furthermore, the absence of a digital platform for train ticketing hinders the ability of administrators to analyze booking trends, track customer preferences, and optimize resource allocation. This lack of data-driven insights limits the potential for strategic decision-making and prevents administrators from efficiently meeting the evolving needs of travelers and ensuring a seamless booking experience.

# PROJECT SCOPE

The Train Ticket Booking System encompasses a wide-ranging scope aimed at revolutionizing the train ticketing process and enhancing the overall experience for both administrators and travelers. At its core, the system provides administrators with a robust platform to register, log in, and manage various aspects of train operations. This includes the addition of new locations, scheduling trains, and monitoring ticket bookings, ensuring comprehensive control over the entire booking process.

For travelers, the system offers an intuitive interface to search for available trains based on preferred locations, dates, and times. With real-time access to train schedules and ticket availability, travelers can make informed decisions and book tickets seamlessly, streamlining the booking process and minimizing the risk of overbooking or missed opportunities. Additionally, the system provides secure online payment options, enhancing convenience and accessibility for travelers.

Furthermore, the Train Ticket Booking System extends beyond basic ticketing functionalities to encompass advanced features such as reporting and analytics. Administrators have access to comprehensive dashboards and reports that offer insights into booking trends, customer preferences, and revenue generation. This enables administrators to make data-driven decisions, optimize resource allocation, and enhance the overall efficiency and profitability of train operations.

In summary, the scope of the Train Ticket Booking System is comprehensive, covering all aspects of the train ticketing process from administration to traveler convenience. By providing a user-friendly interface, robust functionalities, and advanced analytics capabilities, the system aims to revolutionize the train ticketing experience, ensuring a seamless and efficient booking process for both administrators and travelers.

# AIMS & OBJECTIVES

The Train Ticket Booking System (TTBS) is designed with specific aims and objectives aimed at revolutionizing the train ticketing process and enhancing the overall experience for both administrators and travelers.

* **Optimization of Train Ticketing Process:** The primary aim of the TTBS is to streamline and optimize the train ticketing process, providing travelers with a seamless and efficient booking experience. By offering a centralized platform for administrators to manage train schedules, ticket availability, and bookings, the TTBS simplifies complex processes and enhances operational efficiency within the railway system. Through intuitive interfaces and robust functionalities, the system aims to minimize the effort required for travelers to book tickets while ensuring accurate and up-to-date information on train schedules and ticket availability.

* **Promotion of Transparency and Accessibility:** Another aim of the TTBS is to promote transparency and accessibility in train ticketing operations. By centralizing train schedules, ticket bookings, and customer information, the system provides administrators, travelers, and railway staff with transparent access to critical data and processes. This fosters a culture of transparency and accountability within the railway system, ensuring that all stakeholders have access to accurate and up-to-date information, thereby enhancing trust and confidence in the booking process.

* **Empowerment through Data-Driven Insights:** The TTBS aims to empower stakeholders with timely and relevant insights to support informed decision-making. By consolidating data on train schedules, ticket bookings, and customer preferences, the system offers advanced reporting and analytics capabilities to administrators. These insights enable decision-makers to identify booking trends, optimize resource allocation, and enhance the overall efficiency and profitability of train operations. By leveraging data-driven insights, the TTBS empowers railway authorities to make informed choices that drive organizational growth and success in delivering superior train ticketing services.

# OVERALL DESCRIPTION

The Train Ticket Booking System (TTBS) revolutionizes the process of booking train tickets by providing a centralized platform for administrators and travelers. Administrators can efficiently manage train schedules, ticket bookings, and customer information, ensuring a seamless experience. With real-time access to train schedules and ticket availability, travelers can make informed decisions, enhancing the overall efficiency and convenience of the booking process.

# PRODUCT PERSPECTIVE

* The Train Ticket Booking System (TTBS) caters to administrators, railway authorities, and travelers looking for a convenient and centralized platform for managing train ticket bookings. By digitizing and centralizing train schedules, ticket availability, and customer information, the TTBS eliminates the need for manual processes and physical ticketing counters. With its intuitive interface and real-time access to train schedules, travelers can seamlessly book tickets online, reducing the hassle of physical presence at booking counters and streamlining the entire ticketing process.

**USER CHARACTERISTICS**

* User should be familiar with the terms like login, register etc.

**PRINCIPLE ACTORS**

* Admin, Customer.

**GENERAL CONSTRAINTS**

* A full internet connection is required.

**ASSUMPTIONS & DEPENDENCIES**

* Working of Train Ticket Booking System website needs Internet Connection

# BENEFITS OF TRAIN TICKET BOOKING SYSTEM

* This Train Ticketing System solution is fully functional and flexible.
* It is very easy to use.
* This online system helps in back-office administration by streamlining and standardizing the procedures.
* It saves a lot of time, because of remote access.
* The application acts as an office that is open 24/7.
* It increases the efficiency of the management at offering quality services to the customers.

**USERS AND CHARACTERISTICS**

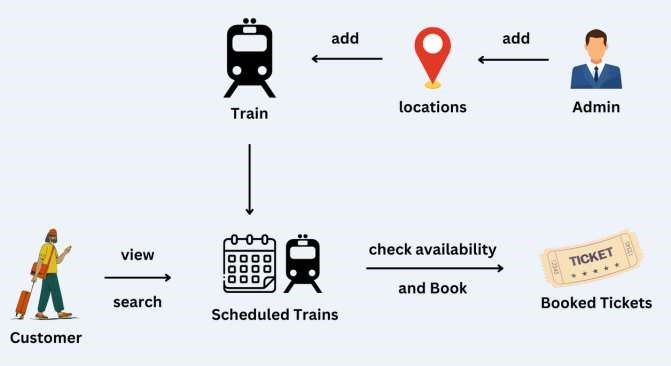
# Admin

* Admin can register and login.
* Admin can add the Locations, Train & Train Seats.
* Admin can update and delete the Trains.
* Admin can View all Trains.
* Admin can schedule the Trains.
* Admin can update the scheduled trains.
* Admin can view Train Tickets Bookings.

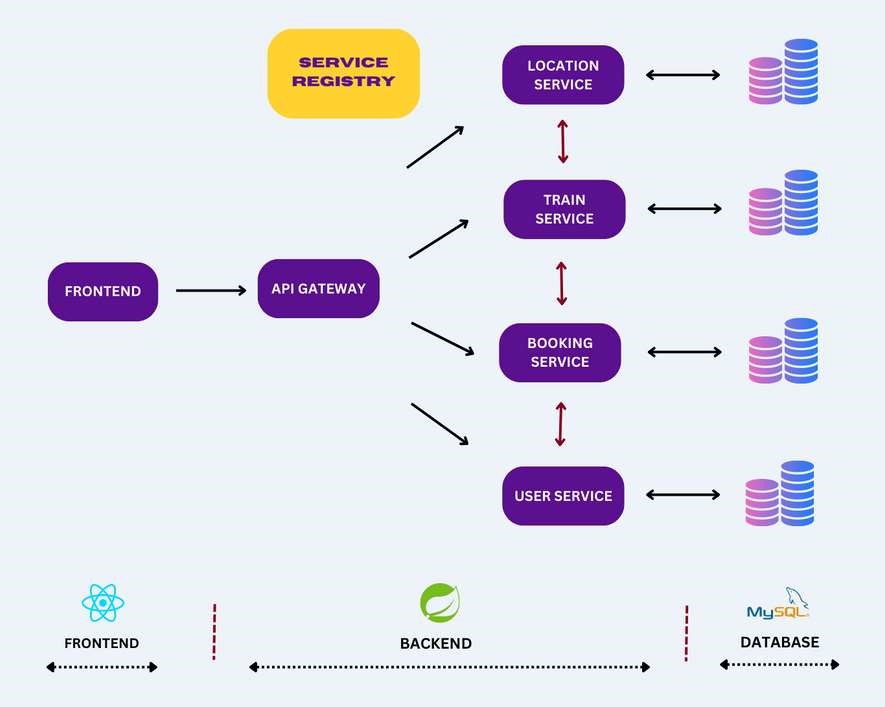
# Manager

* Customer can register & login.
* Customer can view the scheduled trains.
* Customer can search the scheduled Trains.
* Customer can view Train Seats availability.
* Customer can Book the Train by selecting the seats.
* Customer can cancel the Train Seat.

**PROJECT FLOW**



**MICRO SERVICES STRUCTURE:**



1. **API Gateway Service**

An API Gateway is a component that acts as a single entry point for client applications to interact with various micro services within the system.

1. **Service Registry Service**

A Service Registry is a component that facilitates service discovery and allows microservices to locate and communicate with each other dynamically.

1. **User Service**

The User Service will handle all the User-related Operations like user register, login, etc.

1. **Train Service**

The Train micro service will handle Trains, Train Seats & Train Scheduling related operations.

1. **Location Service**

The Location micro service will handle Train location-related operations.

1. **Train Booking Service**

The Train Booking Service will handle all the ticket booking-related operations.

So the request will come to the API Gateway Service first during the request initialization from the front end. After this Micro services can communicate between them get the proper data that is to be sent to the front end.

**MODULE SPECIFICATION**

# FUNCTIONAL REQUIREMENT

This section provides requirement overview of the system. Various functional modules that can be implemented by the system will be:

1. **User Authentication Module:**

User Authentication and Authorization with Spring Boot and React. The registration and Login system has been added so that only authenticated users (Admin, Train Manager, or Employee) can perform their functionalities.

1. **Train Module:**

Add Train, Update Train, Delete Train, Search Train, Fetch all trains, Add Train Seats, Fetch Train Seats, Schedule Trains, etc.

1. **Location Module:**

Add Location, Update Location, Delete Location, Search Location, Fetch Locations, etc.

1. **Ticket Booking Module:**

Book tickets, Update Ticket Status, Fetch Tickets, Fetch Ticket Status, Cancel Booking, View Available Tickets, etc.

**NON FUNCTIONAL REQUIREMENTS**

Following Non-Functional Requirements will be there in the insurance to the internet

* Secure access to member’s confidential data.
* 24X7 availability.
* Better component design to get better performance at peak time.
* Flexible service based architecture will be highly desirable for future extension Non-Functional Requirements define system properties and constraints.

# Various other Non-Functional Requirements are

* Security
* Reliability
* Maintainability
* Portability
* Extensibility
* Reusability
* Compatibility
* Resource Utilization

# REQUIREMENT ANALYSIS

Requirement Analysis means studying the existing system and collecting the details to find out what these requirements are. It is necessary to find out how the system works and where improvements are necessary.

# There are three activities involved in requirements analysis

* Requirement Anticipation
* Requirement Investigation
* Requirement Specification

# Requirement Anticipation

* Based on prior experience, which may have included specific issues or features and specifications for a new system, it forecasts the characteristics of the system.
* It can result in an analysis of regions that a less skilled analyst might otherwise overlook. But requirement Anticipation can be flawed if quick cuts are used and bias is incorporated throughout the inquiry.

# Requirement Investigation

* The system is being examined, and its attributes are being recorded for future research.
* At the core of system analysis, fact-finding methods, prototyping, and computer-aided tools are used to record and describe system features.

# Requirement Specifications

* It also comprises describing the features of the new system, stating the information requirements, and analyzing the data to determine the requirements.
* It involves factual data analysis, critical requirement identification, and requirement fulfillment strategy selection.

# Information Gathering Techniques

The primary goal of fact-finding methodologies is to identify the information needs of an organization so that analysts can provide a precise SRS that users can understand.

The perfect SRS document should

* Be thorough, clear, and jargon-free.
* Cite the informational needs that are operational, tactical, and strategic.
* Resolve any potential conflicts between users and analysts.
* Take advantage of visual tools that make understanding and design simpler.

There is various information gathering techniques.

# Interviewing

A systems analyst uses interviews to get data from people or groups. As the effectiveness of an interview depends on the analyst's abilities as an interviewer, the analyst can be formal, legalistic, play politics, or informal.

It can be done in two ways:

* **Unstructured Interview -** To gather fundamental system knowledge, the system analyst conducts a question and answer session.
* **Structured Interview -** It consists of predetermined questions to which the respondent must provide a close (objective) or open (descriptive) response.

# Advantage of Interviewing

* This approach is typically the most effective way to acquire qualitative data.
* It is helpful for those who struggle to express themselves clearly writing or who might not have the time to finish the questionnaire.
* It is simple and quick to cross-check and confirms information.
* It is capable of handling difficult issues.
* By asking for opinions, it is simple to identify the main issue.

# Questionnaires

The analyst employs this technique to collect data from a sizable population of people on a variety of systemic concerns. There are two types of questionnaires

* **Open-ended Questionnaires:** These contain open-ended, easily interpretable questions. They can go further into a problem and point in the right path for an answer.
* **Closed-ended Questionnaires:** These are questions that are employed after the systems analyst has effectively listed all alternate, mutually exclusive replies.

# Advantages of questionnaires

* It is excellent at gauging the interests, attitudes, sentiments, and beliefs of people who are not physically present.
* Knowing the percentage of a certain group that supports or opposes a specific system feature can be helpful in some circumstances.
* Prior to providing the system project with precise instructions, it is helpful to ascertain the general consensus.
* It offers higher levels of anonymity and more trustworthy, sincere responses.

# Review of Documents, Forms, and Procedures

In order to get insight into a system that specifies the current system capabilities, its operations, or its activities, it can be helpful to review existing records, processes, and forms.

* It enables the user to learn a little bit about the business or its processes before imposing it on others.
* As the procedure manuals and forms describe the structure and features of the current system, it aids in quickly recording current operations.
* It can offer a clear picture of the transactions handled within the company, identifying input for processing, and assessing performance.
* A system analyst may find it useful to comprehend the processes that the system must support.

# Observations

This technique involves noticing and observing people, events, and objects in order to gain information. The analyst pays a visit to the company to study the operation of the current system and comprehend its needs.

# Advantages

* In situations when the validity of the data collected is questioned or where the complexity of some system components makes it difficult for end users to understand something clearly, this direct approach to gathering information might be helpful.
* It generates data that is more accurate and dependable.
* It generates all the outdated and insufficient documentation aspects.

# HARDWARE AND SOFTWARE REQUIREMENTS SPECIFICATION Hardware Requirements

**Processor**: Any Processor above 1 GH & above.

**RAM**: Minimum 4GB.

**Hard Disk**: Minimum 500 GB.

**Processor**: Any processor above 1 GH & above.

# Software Requirements

**Front End**: React JS/JavaScript/Bootstrap.

**Backend (Business Logic)**: Java, Spring Boot Micro services.

**Database**: MySQL 8.0.

**Server**: Embedded Tomcat Server.

**Editor**: Spring Tool Suite (STS), VS Code.

**Operating System**: Windows.

# ABOUT THE TECHNOLOGIES USED

With the advent of the latest technology if we do not update our system then our business result in losses gradually with time.

Here, Java Spring Boot is used for logic and view purposes. For designing

React JS/ HTML/CSS/Bootstrap/JavaScript is used. For Database MySQL 8.0 version is used.

# Spring Tool Suite (STS)

Spring Tool Suite (STS) is an IDE designed for developing Spring Framework applications. It offers tools for coding, debugging, and deployment, including features like code completion and support for Maven and Gradle. Integration with Spring projects like Spring Boot simplifies setup and configuration, enhancing productivity. STS's comprehensive toolset streamlines the development process, making it a preferred choice for Spring developers.

# Java Spring Boot

Java Spring Boot simplifies the creation, configuration, and deployment of Java-based web applications by eliminating extensive XML configuration and boilerplate code. It uses convention over configuration principles, features an embedded HTTP server, and offers auto-configuration for quick setup of standalone, production-ready applications. Spring Boot's starter dependencies and built-in support for common tasks like security, database access, and RESTful APIs accelerate development. Its emphasis on opinionated defaults and streamlined workflows allows developers to focus on business logic rather than infrastructure, making it essential in modern Java application development.

# React JS

React JS, developed by Facebook, is a leading frontend web development framework known for its declarative, component-based architecture. It enables developers to build dynamic UIs with reusable components and optimizes performance with a virtual DOM. React's unidirectional data flow and state management tools like Redux ensure predictable and scalable applications. Its extensive ecosystem, including React Router and Redux Saga, supports feature-rich web application development. React's simplicity, performance, and strong community support contribute to its rising popularity, making it a preferred choice for modern web projects.

# HTML

# HTML, short for Hyper Text Mark-up Language, is used to format and layout text in documents, making them interactive and dynamic. It transforms text into images, tables, links, and more, enhancing presentation with styling. An HTML document comprises various HTML tags, each containing different content.

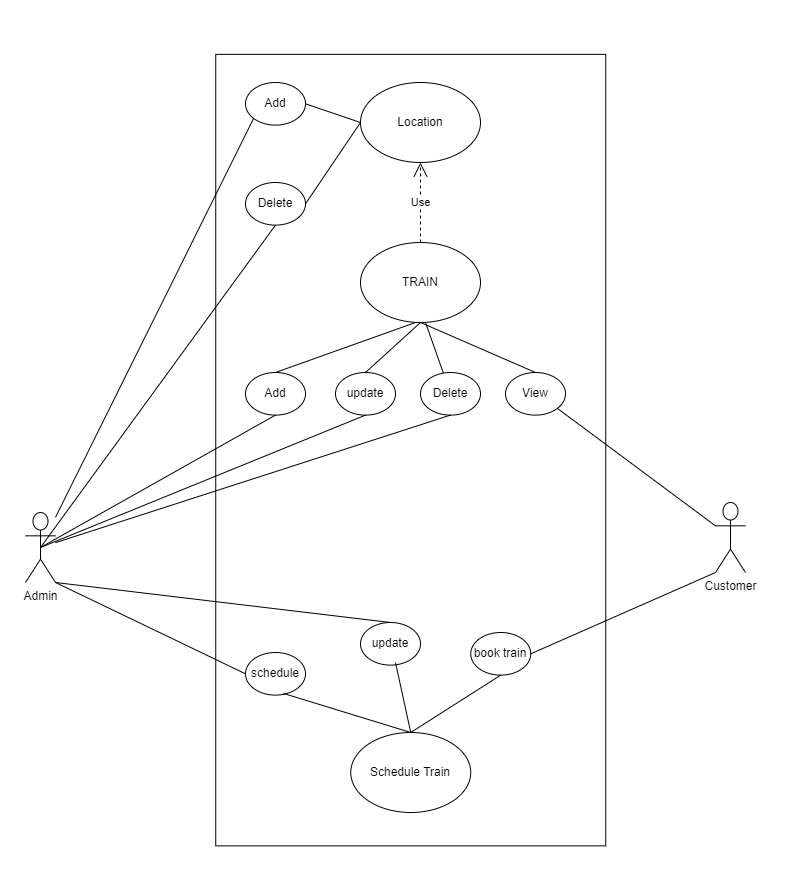
# CSS

CSS stands for Cascading Style Sheets. It is a style sheet language that is used to describe the look and formatting of a document written in a markup language. It provides an additional feature to HTML. It is generally used to change the style of the user interface. It can also be used with any kind of XML documents.

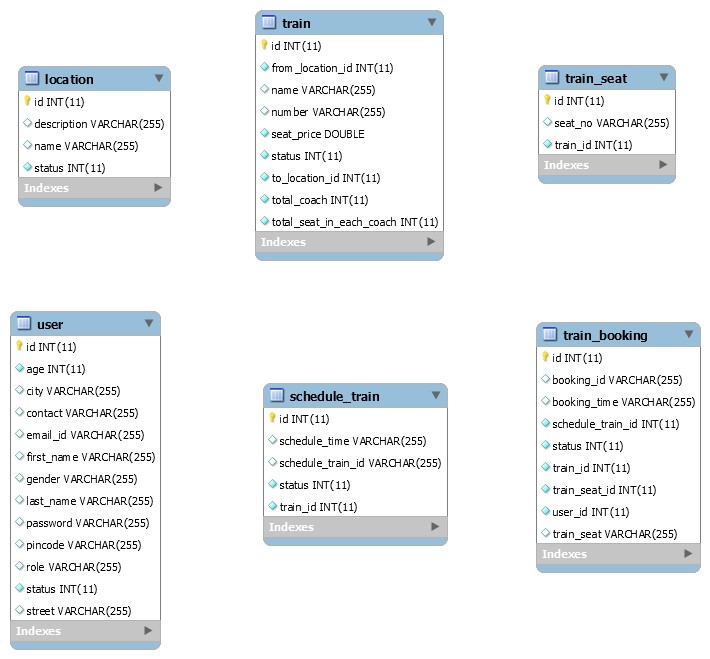
# Bootstrap

It includes HTML and CSS-based design templates for buttons, tables, images carousels, and many others. It facilitates the creation of a responsive design. Bootstrap contains a lot of reusable components and custom jQuery plugins. Bootstrap components are customizable and we can customize bootstrap’s components, less variable and jQuery plugins to get our own style.

# USE CASE DIAGRAM

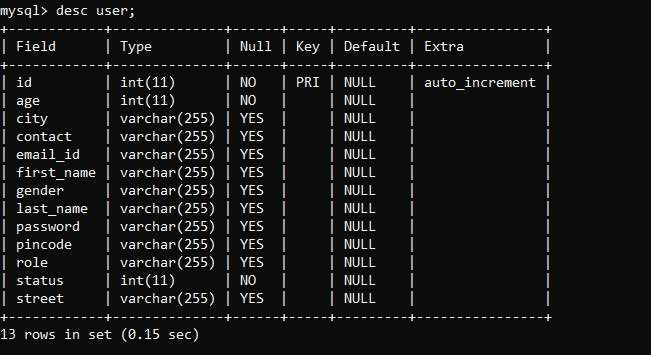


# ENTITY RELATIONSHIP DIAGRAM (ER Diagram)

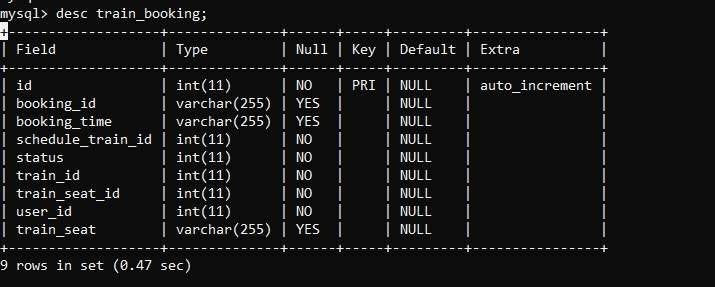


**DATA BASE STRUCTURE**

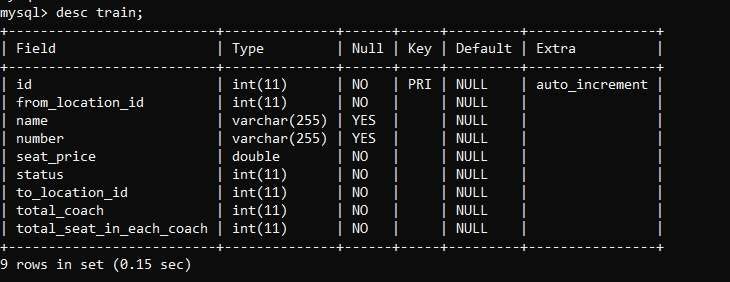
# USER TABLE



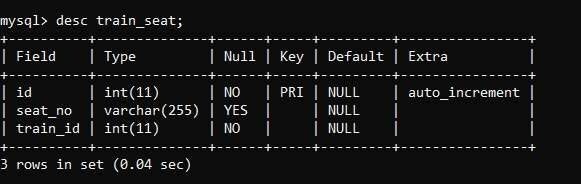
# TRAIN BOOKING TABLE



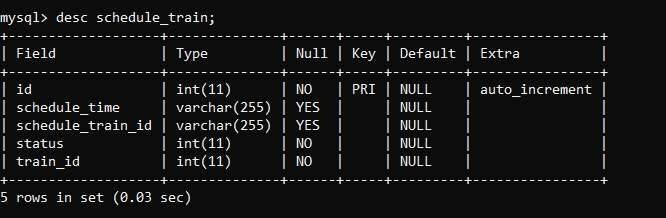
# TRAIN TABLE



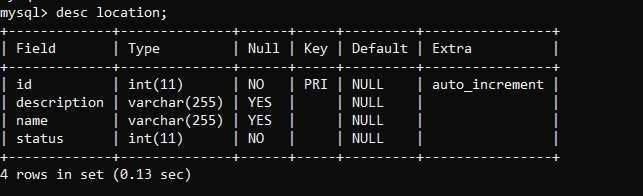
# TRAIN SEAT TABLE



# SCHEDULE TRAIN TABLE



# LOCATION TABLE



# TESTING

Testing is a crucial process in software development that identifies errors and measures quality. It involves executing the program with test cases to ensure it performs as expected. Testing verifies the program's correctness, completeness, and quality, aiding maintenance. Adhering to testing standards reduces costs and time. Extending throughout the coding phase, testing reviews configuration, design, and coding, determining the software's stability. Comprehensive testing of all components prevents bugs post-deployment.

# Black Box Testing

Black box testing also called behavioral testing, focuses on the functional requirements of software. This testing approach enables the software engineer to derive the input conditions that will fully exercise all requirements for a program. Black box testing attempts to find the errors like

* Incorrect or missing functions.
* Interface errors.
* Errors in data structures or external database access.
* Behavior or performance errors.
* Initialization and termination errors.
* In black box testing software is exercised over a full age of inputs and outputs are observed for correctness.

# White Box Testing

White box testing is also called Glass box testing is a test case design control, structure of the procedural design to derive test cases using white box testing method. The software engineer can derive the test cases that guarantee that all independent paths within the module have been exercised at least once. Exercise all logic decisions on their true or false sides. Execute all loops at their boundaries and within their operational bounds. Exercise internal data structure to ensure their validity.

# Types of Testing

* Unit Testing
* Integration Testing
* System Testing

# Unit Testing

Unit testing verifies individual software modules to uncover errors within their boundaries, using detailed design and process specifications. Each module must pass unit tests before integration testing begins. In this project, each service is treated as a module, tested with various inputs during and after development to ensure error-free functionality. Inputs are validated when received from users. Unit testing is performed on modules independently to detect errors, avoiding errors from module interactions initially. This ensures that software units, such as modules and routines, function correctly before integration.

# Integration Testing

After the unit testing we have to perform integration testing. The goal here is to see if modules can be integrated properly, the emphasis being on testing the design and hence the emphasis on testing module interactions.

In this project integrating all the modules forms the main system. When integrating all the modules we have checked whether the integration effects working of any of the services by giving different combination of inputs with which the two services are perfectly before integration.

# System Testing

Here the entire project system is tested. The reference document for this process is the requirements document, and the goal as to see if software meets its requirements.

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST**  **CASE**  **NUMB**  **ER** | **TESTING SCENARIO** | **TESTING**  **SCENARI**  **O** | **RESULT** |
|  | **Login Testing** | |  |
| TC – 1 | Clicking submit entering wrong  Username | Alert "Invalid User" | Pass |
| TC – 2 | Clicking submit entering wrong password | Alert "Invalid User" | Pass |
| TC – 3 | Clicking submit entering wrong  username and password | Alert "Invalid User" | Pass |
| TC – 4 | Clicking submit without entering name | Alert "Please fill  name" | Pass |
| TC – 5 | Clicking submit without entering phone no | Alert "Please fill phone no" | Pass |

# TEST CASES

Test cases are good in revealing the presence of faults. Successful in implementation of test cases implies that there are no error in program. Test cases should be minimum as they are expensive in cases of money and efforts. Primary Objectives of test cases are to ensure that if there is an error or fault in program it is exercise by the test cases. An ideal test cases set is one that succeeds only if there are no error in the program. One possible ideal set of set case is one that includes all possible input to the program and is called exhaustive testing. A test case is good if it detects in undiscovered error in program.

# PROJECT SCREENSHOTS CONCLUSIONS

In conclusion, the Train Ticket Booking System (TTBS) stands as a transformative solution that modernizes the train ticketing process and enhances the overall experience for both administrators and travelers. By providing a centralized platform for managing train schedules, ticket bookings, and customer information, the TTBS streamlines operations, reduces manual effort, and improves efficiency within the railway system. With its user friendly interface and real-time access to critical data, the TTBS empowers travelers to make informed decisions and book tickets seamlessly, ultimately enhancing the convenience and accessibility of train travel.

Furthermore, the TTBS fosters transparency and accountability within the railway system by promoting centralized communication and collaboration between administrators, railway authorities, and travelers. By centralizing train schedules and ticket availability, the system ensures that all stakeholders have access to accurate and up-to-date information, thereby enhancing trust and confidence in the booking process. Additionally, the TTBS enables administrators to monitor ticket bookings, optimize resource allocation, and make data-driven decisions that drive organizational growth and success in delivering superior train ticketing services.

Overall, the Train Ticket Booking System represents a significant step forward in revolutionizing train ticketing operations, empowering both administrators and travelers with efficient tools for managing and booking train tickets. Through its comprehensive functionalities and user-centric design, the TTBS enhances the overall efficiency, transparency, and convenience of the train ticketing process, ensuring a seamless experience for all stakeholders involved in railway travel.

# FUTURE ENHANCEMENT

Looking ahead, future enhancements for the Train Ticket Booking System (TTBS) could focus on leveraging emerging technologies to further improve efficiency and enhance the user experience. One potential enhancement could involve the integration of artificial intelligence (AI) and machine learning algorithms to provide personalized recommendations and predictive insights to travelers. By analyzing past booking patterns and user preferences, the TTBS could offer tailored suggestions for train routes, seating options, and travel packages, enhancing the overall booking experience and increasing customer satisfaction.

Additionally, the TTBS could benefit from the implementation of block chain technology to enhance security and transparency in ticket transactions. By utilizing block chain-based ticketing systems, the TTBS could ensure secure and tamper-proof ticket issuance, reducing the risk of fraud and unauthorized ticket duplication. Moreover, block chain technology could enable transparent tracking of ticket transactions, allowing travelers to verify the authenticity of their tickets and providing administrators with greater visibility into ticket sales and revenue.

Furthermore, future enhancements for the TTBS could involve the integration of Internet of Things (IoT) devices to provide real-time updates and notifications to travelers. By incorporating IoT sensors into trains and railway infrastructure, the TTBS could offer travelers timely information on train delays, platform changes, and other relevant updates, enabling them to better plan their journeys and minimize disruptions. Additionally, Io Tenable ticketing systems could facilitate seamless check-in and boarding processes, enhancing the overall travel experience for passengers.

In summary, future enhancements for the Train Ticket Booking System could leverage AI, block chain, and IoT technologies to further improve efficiency, security, and user experience. By embracing these innovations, the TTBS can continue to evolve and adapt to meet the changing needs and expectations of travelers in an increasingly digital and interconnected world.

**NAMES OF GROUP MEMBERS**

SHUYE ANITA BERINYU

ICTU20223188

AMADINE LENYONGA BAMBOT

ICTU20223006

NJIANGA AYOUBA

ICTU20222964

SEIGNOU MBI CHRISTIAN CYRIL

ICTU20223506

MVOGONKA CHRISTOPHE

ICTU20222936

MOHAMADOU AWAL

ICTU20223103

NOUMBISSI NGUEYO SAMUEL BRYAN

ICTU20223023

NJANZOU MBOUEMBOUE RAHMA

ICTU20223110

NGUTI KHYEIN

ICTU20233610