Project Title: Railway Booking and Tracking System

Category: Web Application

PURPOSE

The purpose of developing this app is to allow users to reserve tickets, cancel reservation, viewing train information, viewing reservation details, updating train information and update reservation details easily and allow them to keep track of their PNR status and location of the train.

SCOPE

Railway Booking System is an attempt to simulate basic online ticketing system. The project will allow booking of train tickets anywhere anytime. It will integrate convenience of an online platform with the reliability of physical booking. The app will work

INTRODUCTION

Introduction has following subsections

Existing System

The present systems are ether manual ticket booking system or a semi-automated system.

Existing system has the following disadvantages :-

➢ Present online reservation websites are not very user friendly.

➢ The data, which is stored on the paper only, may be lost, stolen or destroyed.

➢ The websites are too cluttered with a lot of ads.

➢ Authentication is not very user friendly.

➢ Manual System is very inconvenient.

➢ Waste too much of time.

➢ Are not very intuitive.

➢ User needs to be technically sound to use the existing systems.

Proposed System

The aim of this project is to create a system that is more polished, modernized and gives the user an overall smoother and experience by making the system more user friendly.

The Proposed System will have following functionalities:

➢ **Search**: This function allows the booking agent to search for train that are available between the two travel cities, namely the "Departure city" and "Arrival city" as desired by the traveller. The system initially prompts the agent for the departure and arrival city, the date of departure, preferred time slot and the number of passengers. It then displays a list of train available with different airlines between the designated cities on the specified date and time.

➢ **Selection**: This function allows a particular train to be selected from the displayed list.

All the details of the train are shown :-

✓ Train Number

✓ Date, time and place of departure

✓ Date, time and place of arrival

✓ TRAIN Duration

✓ Fare per head

➢ Review: If the seats are available, then the software prompts for the booking of train. The train information is shown. The total fare including taxes is shown and flight details are reviewed

➢ **Traveller Information**: It asks for the details of all the passengers supposed to travel including name, address, telephone number and e-mail id.

➢ **Payment**: It asks the agent to enter the various credit card details of the person making the reservation.

✓ Credit card type

✓ Credit card number

✓ CVC number of the card

✓ Expiration date of the card

✓ The name on the card

➢ **Cancellation**: The system also allows the passenger to cancel an existing reservation. This function registers the information regarding a passenger who has requested for a cancellation of his/her ticket. It includes entries pertaining to the train No., Confirmation No., Name, Date of Journey, Fare deducted

Advantages for Users:

➢ **Time and Cost Saving**: A lot time and overhead cost is saved as Users can book ticker in a matter of minutes.

➢ **No technical Experience needed**: The webapp would not much technical experience to use.

➢ **Tracking Trains**: Trains location can be tracked so the user can make decision accordingly.

Functional Requirements

Functional requirements are explained below.

➢ **Search**:- User should be able to search for a train easily.

➢ **Booking and cancelling**: Users should be able to book and cancel train tickets.

➢ **Status**: Users should be able to check the status of their ticket.

➢ **Tracking**: Users should be able to track trains

➢ **PNR Exchange** : Users should be able to change their seats by changing their PNR No.

Non-Functional Requirements

➢ **Security**: The system should be secure.

➢ **Reliability**: The system should be able to handle sporadic traffic.

➢ **Maintainability:** The system should be easy to maintain.

➢ **Supportability:** The code should be well documented so that it is easy to fix errors.

➢ **Error Handling**: - Response to user errors and undesired situations has been takencare of to ensure that the system operates without halting

➢ **Response Time**: -The response of all the operation is good. This has been madepossible by careful programming
➢ **User friendliness**: - The system is easy to learn and understand. A native user can

also use the system effectively, without any difficulties

➢ **Portable**: - The software should not be architecture specific. It should be easily

transferable to other platforms if needed

➢ **Safety and Robustness**: - The system is able to avoid or tackle disastrous action. In

other words, it should be foul proof. The system safeguards against undesired events,

without human intervention.

Software Tools

Database Server: MongoDB

Client: Web Browser

Development Tools: VS Code

Programming Language: Nodejs, React, ExpressJS

Deployment

Operating System Server: Window 10, Linux

Hardware Specification

Processor: Intel Core i5

Ram: 4GB

Hard Disk: 256 GB