iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** XI **Month of publication:** November 2023

DOI: https://doi.org/10.22214/ijraset.2023.56851

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue XI Nov 2023- Available at www.ijraset.com

Parking Slot Booking System

Pavithran M[1], Jothish B[2], Rajesh R[3], Varunkumar B[4]
Sri Shakthi Institute of Engineering and Technology, India

Abstract: ***The Parking Slot Booking System is a comprehensive and user-friendly digital platform designed to simplify and enhance the process of reserving and managing parking slots. In today's fast-paced world, efficient parking solutionsare crucial, and this system aims to bridge the gap between drivers and parkingfacilities by providing a convenient and accessible solution.***
This project assists parking facilities, such as parking garages or lots, in allowing clients or users to reserve parking slots online. The system enables users to easily browse through available parking slots, view their details, including location, availability, and pricing, and book slots for their preferreddate and time. This parking slot booking system efficiently organizes and manages the allocation of parking slots, ensuring a seamless and hassle-free experience for both users and parking facility administrators.
The system operates with three main roles: the administrator, the parking facility, and the user. The administrator populates the system with informationabout available parking slots, their locations, and other relevant details.
Users visit the parking slot booking system website or app to find a suitable parking slot, check the availability for their desired time, and make a reservation. The parking facility can view and manage all reservations, ensuring that the parking slots are efficiently allocated.
This system offers parking facilities a robust tool for overseeing their parking slot reservations, reducing administrative workload, and enhancing the overalluser experience for those seeking convenient and reliable parking solutions.

I. INTRODUCTION

In our fast-paced and digitally driven world, the need for efficient and convenient parking solutions has become increasingly critical. The Parking Slot Booking System is a modern and innovative solution designed to streamline thereservation and management of parking slots. This system represents a departure from traditional parking methods, offering significant advantages for both parking facilities and users. The Parking Slot Booking System leverages technology to simplify the process of reserving parking slots, aiming to address common challenges associated with parking, such as finding available slots, time-consuming searches, and potential mismanagement of parking facilities.

By providing real-time information about parking slot availability, this system empowers users to take control of their parking needs, giving them access to a user-friendly platform where they can easily find information about available parking slots, their locations, and pricing. This real-time availability information enables users to select parking slots that best suit their schedules, minimizing the time spent searching for parking and optimizing the utilization of parking resources. The Parking Slot Booking System not only simplifies the parking reservation process but also enhances the overall parking experience, improving the efficiency of parking facilities. As a result, it makes parking more accessible, convenient, and effective for both parking facility operators and users, aligningwith the evolving demands

A. Ruby Programming Language

1) About Ruby

Ruby is a dynamic, open-source programming language known for its versatility and user-friendly nature. It is well-suited for a wide range of applications, including web development, automation, data processing, and more. Ruby's ease of use and elegant syntax make it a popular choice amongdevelopers, and its open-source nature ensures accessibility to all.

a) Web Development

Ruby is often employed in web development, particularly with the Ruby on Rails framework. This framework simplifies the creation of robust and scalableweb applications. Ruby's dynamic nature allows for rapid development, makingit an ideal choice for web developers looking to build feature-rich and interactive websites.

b) Ruby's Versatility

Beyond web development, Ruby can be used for various applications.

Its dynamic typing, object-oriented design, and extensive libraries make it a valuable tool for automation, scripting, and data analysis. Ruby's flexibility allows developers to write clean and concise code, which can be a time-saving advantage in various programming tasks.

c) Community and Resources

Ruby has a dedicated community of developers and a wealth of resources. This includes extensive documentation, open-source libraries, and the RubyGems package manager, which simplifies the integration of third-party libraries and tools. Ruby's popularity and strong community support ensure that developers have access to the tools and knowledge needed to succeed in their projects.

d) Command Line Scripting

Ruby's command line scripting capabilities enable developers to create efficient scripts for automating tasks, data processing, and system administration. The command line interface (CLI) in Ruby allows for deep interaction with theunderlying system, making it a versatile choice for a wide range of programming needs.

The versatility and simplicity of Ruby, along with its strong community and resources, make it a valuable programming language for a broad spectrum ofapplications, not just web development.

2) Features of Ruby

a) Open Source

Ruby is an open-source programming language, which means it is freely available for anyone to use, modify, and distribute. This open nature encouragesa large and collaborative community of developers and contributors, which leads to continuous improvement and innovation.

b) Cross-Platform

Ruby is a cross-platform language, compatible with various operating systemslike Windows, Linux, macOS, and more. This cross-platform compatibility makes it a convenient choice for developers working on diverse environments.

c) Simplicity

Ruby is celebrated for its elegant and readable syntax, which is relatively easyto learn and understand. It shares similarities with other popular programminglanguages, making it accessible to developers with varying skill levels.

d) Versatility

Ruby is a versatile language that can be used for a wide range of applications, including web development, scripting, automation, and data analysis. Its flexibility allows developers to write concise and efficient code.

e) Community and Support

Ruby boasts a robust community of developers who provide extensive documentation, tutorials, and online forums. This community-driven aspect ensures continuous support and the availability of a wealth of resources, making it a welcoming environment for both beginners and experienced programmers.

f) Object-Oriented

Ruby is a pure object-oriented language, where everything is an object. This consistency simplifies programming and fosters good coding practices.

g) Command Line Scripting

Ruby excels in command line scripting, enabling developers to create efficient scripts for automation, data processing, and system administration. The command line interface (CLI) in Ruby provides a powerful tool for various scripting needs.

h) Security

While Ruby itself is a secure language, security issues can still arise due to poorcoding practices.

Developers are responsible for following security best practices to prevent vulnerabilities like SQL injection and cross-site scripting (XSS).

In summary, Ruby is a versatile, open-source programming language with a strong community, simple syntax, cross-platform compatibility, and the ability to excel in various applications, from web development to command line scripting. Its elegant and readable code makes it a favored language among developers, and its object-oriented nature fosters good coding practices. The open-source nature of Ruby ensures continuous development and improvement, while its security depends on the practices and vigilance of its users.

3) Features of MySQL

a) Client/Server Architecture

MySQL follows a client/server architecture. It comprises a central database server (MySQL) that manages data and communicates with multiple clients or application programs. These clients can be situated on the same server or distributed across various machines, enabling efficient and distributed data access.

b) Relational Database

MySQL is a robust relational database management system (RDBMS) that organizes data into structured tables with rows and columns. This relationalmodel allows for the efficient storage, retrieval, and management of data.

c) SQL Compatibility

MySQL supports SQL (Structured Query Language), a standardized language for querying and updating data and managing databases. With configuration settings, MySQL can be configured to behave compatibly with various other database systems, making it versatile and compatible.

d) Platform Independence

MySQL can be deployed on several operating systems, including Apple Macintosh OS X, Linux, Microsoft Windows, and various UNIX-like systems.This platform independence ensures that MySQL can adapt to a wide range ofcomputing environments.

e) High Performance

MySQL is well-known for its high performance and efficient data handling. It excels at managing large datasets and executing complex queries, making it suitable for demanding and data-intensive applications. The database system includes advanced optimization techniques for query execution.

f) Scalability

MySQL is highly scalable, accommodating applications ranging from small personal projects to large-scale enterprise systems. It provides features for replication, clustering, and partitioning to ensure that it can grow with your application's requirements.

g) Data Types

MySQL supports a wide array of data types, including integers, floating-pointnumbers, strings, dates, and more. This comprehensive support for data typesallows developers to work with various data formats efficiently.

h) Data Security

MySQL offers robust security features, including user authentication, authorization, and encryption, to protect data from unauthorized access and maintain compliance with security standards. It ensures that your data is safeand secure.

i) Open Source and Community Support

MySQL is an open-source database system, benefitting from a large and activecommunity of developers and users. This community-driven development ensures that MySQL remains updated, secure, and well-supported. Extensive documentation, resources, and user forums are readily available for assistance and learning.

II. LITERATURE REVIEW

A. Implementation of Parking Slot Booking SystemUsing Ruby

The implementation of a Parking Slot Booking System using Ruby has gained recognition in the domain of parking management. This literature review explores the existing body of knowledge to shed light on various facets of these systems. Ruby, with its versatility and ease of use, has emerged as a suitable choice for developing the backend of parking slot booking systems.

Research has indicated that Parking Slot Booking Systems implemented using Ruby offer several advantages. These systems streamline the process of reserving and managing parking spaces, improving the overall parking experience for users (Khurana et al., 2020). Users can access real-time information about available parking slots, make reservations, and receive information about location, pricing, and availability, all of which contribute toenhanced user satisfaction.

Ruby, known for its flexibility and simplicity, is well-suited for managing the backend of parking slot booking systems. It excels in tasks such as database integration, data processing, and interaction with the front-end, making it an ideal choice for building user-friendly and efficient systems. The Ruby community provides valuable resources, documentation, and libraries, ensuring that developers have access to the tools and knowledge required for successful system implementation.

The effectiveness of Parking Slot Booking Systems is dependent on continuous research and innovation. Future research should continue to explore the impact of Ruby-based Parking Slot Booking Systems on parking facility management and investigate the potential of emerging technologies in enhancing parking service. As parking challenges continue to evolve in urban environments, technology-driven solution like those built with Ruby are poised to play a significant role in providing efficient and convenient parking options.

B. Database Connection To System Using MYSQL

Several studies have demonstrated the positive impact of database integration using MySQL in various applications, including educational institutions. For instance, Osman et al. (2020) found that a well-integrated database system can enhance student attendance tracking and subsequently improve academic performance. In our system, a review of the literature suggests that database integration, particularly with MySQL, is crucial for effective management andretrieval of data in educational settings.

In our implementation, we utilized various online resources to ensure the effective connection of our system to the MySQL database. Here are some of the most commonly used references and resources that played a pivotal role inour project:

1) *MySQL Official Documentation:* The official MySQL documentation, accessible through the MySQL website, serves as an exhaustive and authoritative resource. It encompasses everything from installation and configuration guides to advanced query techniques and performance optimization. It provides essential guidance for developers in establishing a secure and efficient database connection.
2) *W3Schools MySQL Tutorial:* W3Schools, a widely recognized online learning platform, offers an extensiveMySQL tutorial that covers fundamental concepts of the language as well as more advanced topics. This resource is valuable for developers seeking a comprehensive understanding of MySQL and its practical applications.
3) *MySQL Cheat Sheet by SQLtutorial.org:* SQLtutorial.org provides a MySQL cheat sheet that serves as a quick reference for common MySQL commands and syntax. This resource proves handy for developers who need to swiftly access essential commands or functions while working on their projects.
4) *MySQL Forums:* The MySQL Forums offer a valuable community-driven resource for developers encountering specific MySQL-related issues. These forums are an excellent platform for seeking assistance, sharing knowledge, and resolving technical challenges related to database integration.

The literature underscores the importance of a well-integrated and efficient database system in educational institutions and various other contexts. Adequate database connection, as enabled by MySQL, is fundamental for data management, retrieval, and performance optimization. By utilizing these references and resources, we ensured a robust and reliable database connection for our system, facilitating seamless data management and retrieval processes.

III. SYSTEM STUDY

A. Existing System

The existing parking slot booking system predominantly relies on manual procedures and on-site interactions for reserving and managing parking slots.

Users typically need to physically visit parking facilities, often engaging with parking attendants or administrative staff who manually check for available slots and facilitate the reservation process. Parking information is usually recorded using paper records

or ledgers, and users are often provided with physical parking permits or tickets as evidence of their reservations.

In the current system, users are required to be present at the parking facility,look for available slots, and interact with attendants to secure their parking space. As parking facilities expand or urban areas become more congested, traditional parking systems may face scalability challenges and efficiency concerns. Manual processes can become increasingly burdensome, and maintaining paper records may become impractical.

Moreover, traditional parking systems can present limitations in terms of flexibility for rescheduling or canceling parking reservations. Users may need tovisit the facility in person or make additional phone calls to adjust their reservations, and these modifications are subject to staff availability. Access to parking information may also be limited, and making last-minute changes can be inconvenient.

The traditional parking booking approach may lead to issues such as inefficient space utilization, long waiting times at parking facilities, manual errors, and limitations in accommodating the evolving demands of urban mobility.

B. Disadvantages Of Existing System

The existing parking slot booking systems, whether manual or digital, come with several disadvantages that affect both users and parking facility operators.Here are some common drawbacks associated with these systems:

1) Limited Availability: Traditional systems may only allow parking reservations during specific hours, inconveniencing users who require parkingoutside of those times.
2) Inefficient Rescheduling: Users might encounter difficulties in reschedulingor canceling parking reservations without in-person visits or additional phone calls to the facility.
3) Time-Consuming: Manual parking reservation processes can be time-consuming, causing delays for users and parking attendants.
4) Resource Allocation: Parking facilities may struggle to allocate resources effectively, leading to overallocation for reservations that result in no-shows orunderallocation during peak demand.
5) Communication Inefficiencies: Communication primarily occurs throughface-to-face interactions or phone calls, which can lead to communication breakdowns, missed calls, or long wait times.
6) Inadequate Record-Keeping: Traditional systems rely on paper records, which can be prone to errors, damage, or loss, and necessitate physical storage.
7) Data Privacy Concerns: Concerns may arise regarding the security and privacy of user data, especially if robust security measures are not in place.
8) Inconvenience: Users may find traditional systems inconvenient, as theyrequire on-site visits or phone calls for booking, leading to potential time wastage and reduced user satisfaction.

These disadvantages underscore the need for a more efficient and user-friendlyparking slot booking system that can leverage modern technology to address these shortcomings.

IV. PROPOSED METHODOLOGY

A. Proposed System

The proposed parking slot booking system represents a modern and user- friendly solution designed to address the limitations of traditional parking reservation systems and improve the overall user experience. The main objective of this project is to develop an application that minimizes manualprocesses in parking slot management, providing a more convenient and efficient approach.

This advanced system harnesses digital technology to offer online parking slot reservations, allowing users to book slots 24/7, thereby increasing convenience and reducing the need for physical visits to parking facilities. The system includes user interfaces and mobile applications for easy access and management of parking reservations. Advanced algorithms are employed for optimizing resource allocation and enhancing parking facility management. Thesystem aims to reduce inefficiencies, improve user satisfaction, and streamline parking operations.

B. Advantages Of Proposed System

The proposed parking slot booking system offers several advantages, including:

1) *Efficient Parking Slot Management:* The system streamlines the process of reserving parking slots, making it efficient for both users and parking facility operators. This efficiency reducesmanual workload and improves parking facility management.
2) *Enhanced User Experience:* Users benefit from the convenience of booking parking slots online and managing their reservations through user-friendly interfaces and mobile applications. They can create accounts, book slots, and access their

bookinghistory with ease.

3) *Improved Resource Allocation:* The system employs advanced algorithms to optimize parking slot allocation, reducing the likelihood of over-allocated or under-allocated slots. This results in better resource management and an improved user experience.
4) *Reduced Waiting Times:* Real-time availability information and intelligent slot allocation can significantly reduce waiting times for users, making the booking process moreuser-centric and efficient.
5) *Global Accessibility:* The system can be adapted and scaled for use in various parking facilities worldwide, improving access to parking services and addressing the growingdemands of urban mobility.
6) *Security and Privacy:* The implementation of strong security measures ensures that user data and payment information are protected, enhancing user trust and compliance withdata privacy regulations.

The proposed parking slot booking system represents a modern and user-centric approach to parking facility management, addressing the inefficiencies and inconveniences associated with traditional parking reservation systems. This solution leverages technology to offer a more efficient, convenient, and secure parking experience for users.

C. System Speicfication

1) Software Requirements

This section gives the details of the software that are used for the development.

- Operating System : Windows 10 /Linux
- Web Browser : Google Chrome
- Coding Language : Ruby
- Database : MySQL

2) Hardware specification

This section gives the details and specification of hardware on whichthe system is expected to work.

- Processor : Intel (R) core(TM) i3-6100U
- Ram : 4 GB
- Hard Disk : 256 GB

D. Software Descripition

1) Ruby

Ruby is a versatile, dynamic, and general-purpose programming language thathas found popularity in various domains, including web development. It is known for its elegant and readable syntax, which prioritizes developer happiness. Ruby is open-source and freely available, making it accessible to developers.

Ruby is particularly well-suited for web development and can be embedded directly into HTML code, similar to PHP. When Ruby code is executed on a web server, the result can be any type of data, including dynamically generated HTML or binary image data. This output can be used as a whole or in part to form an HTTP response, making it ideal for building dynamic web applications.

Ruby's syntax is designed for clarity and expressiveness, and it draws inspiration from several programming languages, including Perl and C, similarto PHP. Its clean and human-readable syntax makes it a popular choice amongdevelopers.

Furthermore, Ruby excels in interacting with databases, with support for variousdatabase systems, including popular choices like MySQL. This capability allows developers to easily store and retrieve data, making Ruby a preferred solution for creating content management systems, e-commerce platforms, and a wide range of web-based services that require robust database interactions.

In summary, Ruby is a dynamic and expressive programming language that is well-suited for web development. Its readability and elegant syntax, coupled with its strong database interaction capabilities, have made it a go-to choice fora wide array of web-based applications and services.

2) MySQL

A database is a structured collection of data, ranging from simple shopping liststo extensive picture galleries or vast amounts of information within a corporate network. MySQL is a prominent open-source Relational Database ManagementSystem (RDBMS) utilized for the storage and management of structured data. Itplays a pivotal role in the development of dynamic web applications, content management systems, e-commerce platforms, and various other software systems.

MySQL's relational database structure is designed for efficient data storage, retrieval, and manipulation. It employs the universally recognized Structured Query Language (SQL) for managing and interacting with data, making it a reliable choice for developers and businesses seeking to organize and work withstructured information effectively.

V. RESULT AND DISCUSSION

The Parking Slot Booking System has transformed the way parking services areaccessed and managed. This system offers users the convenience of reserving parking slots from the comfort of their own devices. Moreover, it ensures secure and confidential communication between users and parking facility operators, safeguarding their personal information. The system also plays a crucial role in streamlining the parking process by maintaining accurate and real-time parking slot records, simplifying the management of available slots.

One of the significant benefits of this system is its contribution to efficient parking facility management. By optimizing parking slot reservations and reducing no-shows, it minimizes administrative overhead. This, in turn, enhances the overall efficiency of parking facilities, resulting in a smoother andmore user-friendly parking experience for both facility operators and users.

VI. CONCLUSION AND FUTURE SCOPE

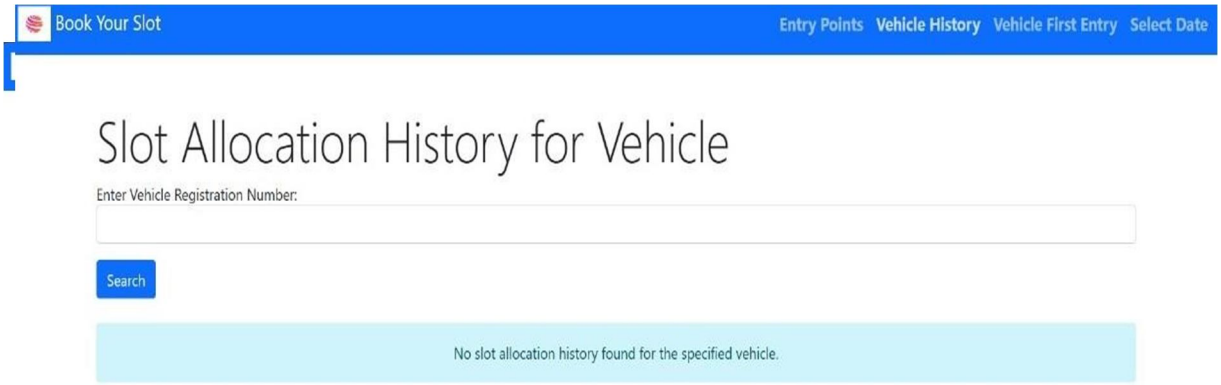
A. *Conclusion*

1) Integration with Smart Cities: As cities continue to develop smart infrastructure, the system can integrate with city-wide traffic managementsystems to provide real-time data on parking availability, further reducingcongestion.
2) Sustainability Features: The system can incorporate eco-friendly options such as reserving electric vehicle charging stations and promotingcarpooling.
3) Parking Guidance: Future developments can include GPS-based parking guidance, directing drivers to their reserved slots efficiently.

B. *Future Scope*

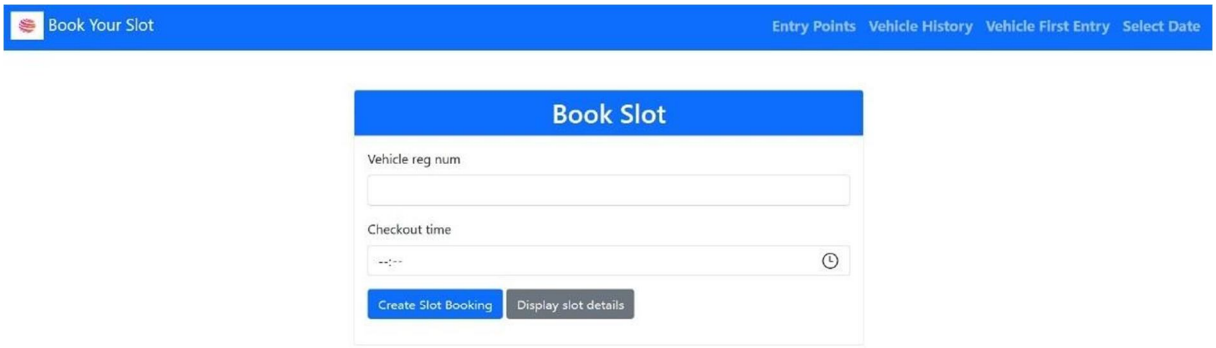
1) The integration of artificial intelligence (AI) will play a critical role in optimizing the parking slot booking system. AI-driven algorithms will analyze historical booking patterns and user preferences to offer personalized parking slot recommendations, ensuring efficient space allocation and enhancing the user experience. These algorithms will alsoenable predictive analytics for better resource management and demand forecasting.
2) These enhancements in the future scope of the parking slot booking system aim to improve communication, streamline the booking process,and make parking management more efficient and user-centric.

SLOT HISTROY

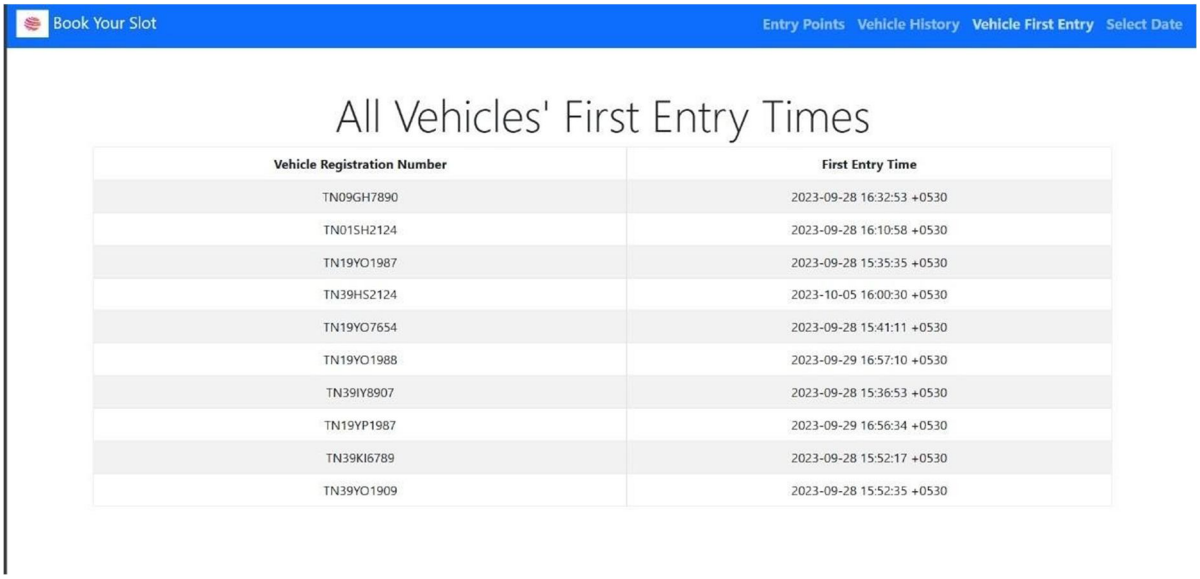


HOME PAGE

SLOT BOOKING MODULE



ENTRY HISTROY



APPENDIX

Source Code:

```
class SlotBookingsController < ApplicationControllerbefore_action :set_entry, only: [:new,:create]
  def new
    @booking = @entry_point.slot_bookings.newend
  def create
  @booking = @entry_point.slot_bookings.new(slot_params)@booking.checkin_date = Date.today

  @booking.entry_time = Time.now
  slot = ParkingSlot.where(occupied: false
  ).order("distance_between_entry_point_#{@booking.entry_point_id}
```

```
ASC").firstif slot
    @booking.parking_slot_id = slot.id

    if @booking.save slot.update(occupied: true )
          ChangeOccupationJob.set(wait_until: @booking.checkout_time).perform_later(slot)
    flash[:success] = "Slot booked successfully. Your slot number is #{slot.id}."redirect_to
    new_entry_point_slot_booking_path(@entry_point)
    else
    flash[:error_1] = @booking.errors
      redirect_to new_entry_point_slot_booking_path(@entry_point)end
    else
    flash[:error] = "Sorry, no slots are available at the selected entrance."redirect_to new_entry_point_slot_booking_path(@entry_point)
      endend

    def select_date
    if params[:selected_date].present?
    @selected_date = Date.parse(params[:selected_date])@slot_bookings = SlotBooking.where(entry_time:
@selected_date.beginning_of_day..@selected_date.end_of_day)else
      @slot_bookings = []end
    end
    def vehicle_history
    @vehicle_registration_number = params[:vehicle_reg_num]@vehicle_history = SlotBooking.where(vehicle_reg_num:
    @vehicle_registration_number)
    end
  def first_entry_times @first_entry_times =
    SlotBooking.group(:vehicle_reg_num).minimum(:entry_time)
    end
    private
    def set_entry
      @entry_point = EntryPoint.find(params[:entry_point_id])end
    def slot_params
    params.require(:slot_booking).permit(:entry_point_id,:vehicle_reg_num,:entry_ time,:checkout_time)
    end
    end
    class EntryPointsController < ApplicationController
    def index
      @entry_point = EntryPoint.allend
    end
    <div class="container mt-5">
    <h1 class="display-4 text-center">All Vehicles' First Entry Times</h1>
    <div class="table-responsive">
    <table class="table table-bordered table-striped">
    <thead class="thead-dark">
    <tr>
    <th class="text-center">Vehicle Registration Number</th>
    <th class="text-center">First Entry Time</th>
    </tr>
    </thead>
    <tbody>
          <% @first_entry_times.each do |vehicle_registration_number,first_entry_time| %>
```

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue XI Nov 2023- Available at www.ijraset.com

```
<tr>
<td class="text-center"><%= vehicle_registration_number %></td>
<td class="text-center"><%= first_entry_time %></td>
</tr>
<% end %>
</tbody>
</table>
</div>
</div>
<div class="container mt-5">
<div class="row justify-content-center">
<div class="col-md-6">
<div class="card">
<div class="card-header bg-primary text-center text-white">
<h2 class="card-title mb-0">Book Slot</h2>
</div>
<div class="card-body">
<%= form_with(model: [@entry_point, @booking], class: "my-form") do
|form| %>
<% if flash[:success] %>
<div class="alert alert-success">
<%= flash[:success] %>
</div>
<% end %>
<% if flash[:error] %>
<div class="alert alert-success">
<%= flash[:error] %>
</div>
<% end %>
<% if flash[:error_1] %>
<div class="alert alert-danger">
<ul>
<% flash[:error_1].each do |attribute, message| %>
<li><%= "#{attribute.to_s.humanize} #{message}" %></li>
<% end %>

</ul>
</div>
<% end %>

<div class="mb-3">
<%= form.label :vehicle_reg_num, class: "form-label" %>
<%= form.text_field :vehicle_reg_num, class: "form-control" %>
</div>
<div class="mb-3">
<%= form.label :checkout_time, class: "form-label" %>
<%= form.time_field :checkout_time, class: "form-control" %>
</div>

<div class="mb-3">
```

```
<%= form.submit "Create Slot Booking", class: "btn btn-primary" %>
            <%= link_to "Display slot details", slot_bookings_select_date_path,class: "btn btn-secondary" %>
</div>

<% end %>
</div>
</div>
</div>
</div>
</div>
<div class="container mt-5">
<div class="row justify-content-center">

<div class="col-md-6">
<div class="card">
<div class="card-header bg-primary text-center text-white">
<h2 class="card-title mb-0">Book Slot</h2>
</div>
<div class="card-body">
<%= form_with(model: [@entry_point, @booking], class: "my-form") do
|form| %>

<% if flash[:success] %>
<div class="alert alert-success">
<%= flash[:success] %>
</div>
<% end %>
<% if flash[:error] %>
<div class="alert alert-success">
<%= flash[:error] %>
</div>
<% end %>

<% if flash[:error_1] %>
<div class="alert alert-danger">
<ul>
<% flash[:error_1].each do |attribute, message| %>
<li><%= "#{attribute.to_s.humanize} #{message}" %></li>
<% end %>
</ul>
</div>

<% end %>

<div class="mb-3">
<%= form.label :vehicle_reg_num, class: "form-label" %>
<%= form.text_field :vehicle_reg_num, class: "form-control" %>
</div>

<div class="mb-3">
```

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue XI Nov 2023- Available at www.ijraset.com

```
<%= form.label :checkout_time, class: "form-label" %>
<%= form.time_field :checkout_time, class: "form-control" %>
</div>

<div class="mb-3">
<%= form.submit "Create Slot Booking", class: "btn btn-primary" %>
            <%= link_to "Display slot details", slot_bookings_select_date_path,class: "btn btn-secondary" %>
</div>

<% end %>
</div>
</div>
</div>
</div>
</div>
[2:36 PM, 10/26/2023] Pavithran: <div class="container mt-5">
<div class="row">
<div class="col-md-8 mx-auto">
<div class="card">

<div class="card-header bg-primary text-white">
<h2 class="card-title text-center">Displaying Slot Details</h2>
</div>
<div class="card-body">
        <%= form_for(slot_bookings_select_date_path, method: :get, class: "mb-4") do %>
<div class="form-group">
<label for="selected_date">Select Date:</label>
<%= date_field_tag :selected_date, nil, class: "form-control" %>
</div>
<%= submit_tag "Search", class: "btn btn-primary btn-block mt-3" %>
<% end %>

<% if @slot_bookings.any? %>
<div class="table-responsive">
<table class="table table-bordered table-striped">
<thead class="thead-dark">
<tr>
<th class="text-center">Slot Number</th>
<th class="text-center">Vehicle Registration Number</th>
<th class="text-center">Entry Time</th>
</tr>
</thead>
<tbody>
<% @slot_bookings.each do |booking| %>
<tr>
<td class="text-center"><%= booking.parking_slot_id %></td>
<td class="text-center"><%= booking.vehicle_reg_num %></td>

                <td class="text-center"><%= booking.entry_time.strftime("%H:%M:%S") %></td>
</tr>
```

```
<% end %>
</tbody>
</table>
</div>
<% else %>
<div class="alert alert-info text-center">No slots used on the selected date
</div>
<% end %>
</div>
</div>
</div>
</div>
</div>
[2:36 PM, 10/26/2023] Pavithran: <div class="container mt-5">
<h1 class="display-4">Slot Allocation History for Vehicle</h1>
  <%= form_for(slot_bookings_vehicle_history_path, method: :get, class: "mb-4") do %>
<div class="form-group">
      <label for="vehicle_registration_number">Enter Vehicle RegistrationNumber:</label>
<%= text_field_tag :vehicle_reg_num, nil, class: "form-control" %>
</div>
<%= submit_tag "Search", class: "btn btn-primary mt-3" %>

<% end %>

<% if @vehicle_history.any? %>
<div class="table-responsive">
<table class="table table-bordered table-striped">
<thead class="thead-dark">
<tr>
<th class="text-center">Entry Date</th>
<th class="text-center">Entry Time</th>
<th class="text-center">Check Out Time</th>
<th class="text-center">Slot Number</th>
</tr>
</thead>
<tbody>
<% @vehicle_history.each do |booking| %>
<tr>
<td class="text-center"><%= booking.entry_time.strftime("%Y-%m-
%d") %></td>
          <td class="text-center"><%= booking.entry_time.strftime("%H:%M:%S") %></td>
<td class="text-center"><%= booking.checkout_time.strftime("
%H:%M:%S") %></td>
<td class="text-center"><%= booking.parking_slot_id %></td>

</tr>
<% end %>
</tbody>
</table>
```

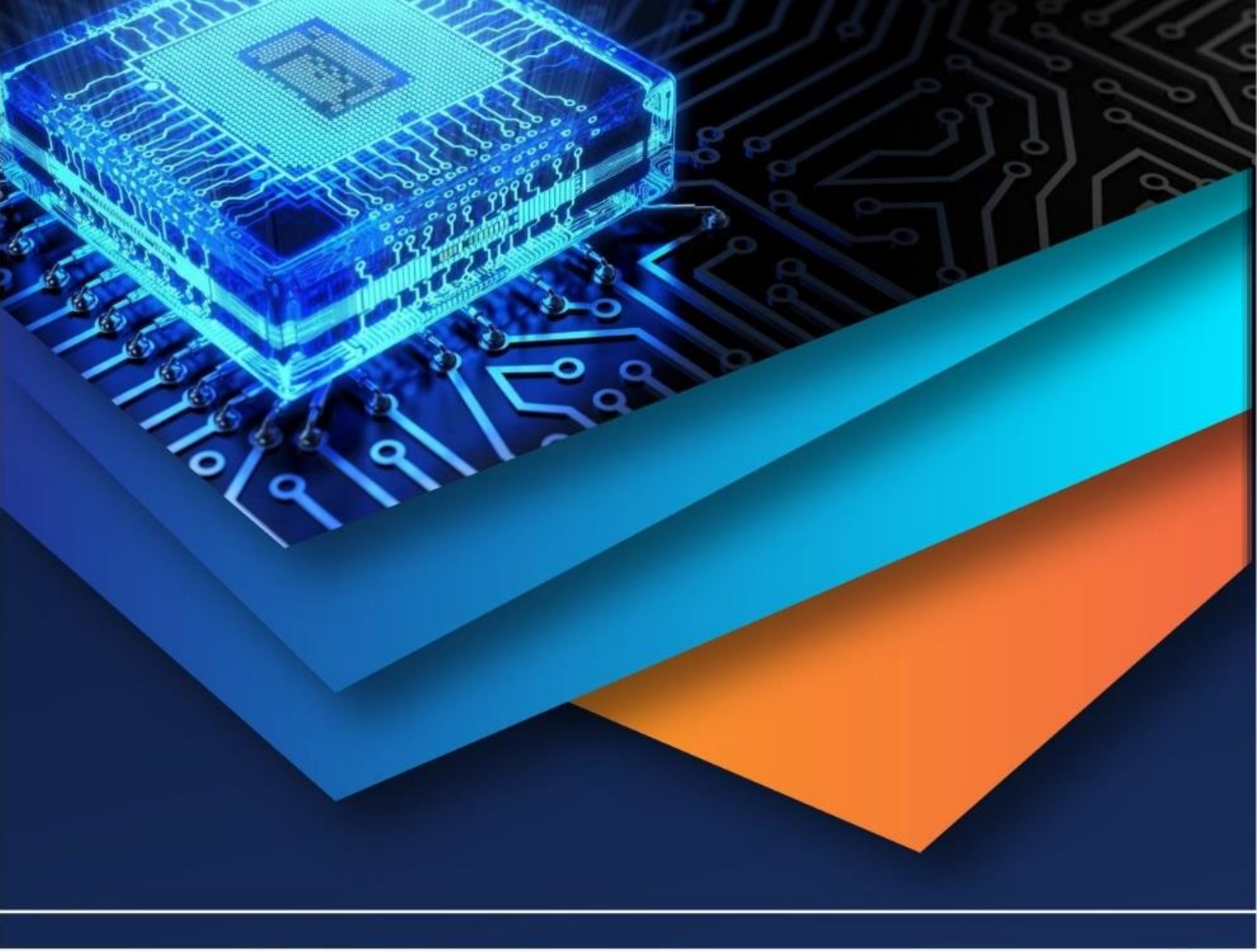
```
</div>
<% else %>
<div class="alert alert-info text-center">
No slot allocation history found for the specified vehicle.
</div>
<% end %>
</div>
[2:37 PM, 10/26/2023] Pavithran: <div class="container d-flex justify-content-center align-items-center  vh-100">
<div class="card text-center">
<div class="card-body">
      <h1 class="card-title display-4 mb-4">Welcome to Parking Slot BookingSystem</h1>
      <a href="<%= entry_points_path %>" class="btn btn-primary btn-lg">Select Entry Point</a>
</div>
</div>
</div>
[2:37 PM, 10/26/2023] Pavithran: class SlotBooking < ApplicationRecordbelongs_to :entry_point
belongs_to :parking_slot
  validates :vehicle_reg_num, presence: true, format: { with: /\A[A-Z]{2}\d{2}[A-Z]{2}\d{4}\z/,
message: "should be in the format
XX99XX9999" }
validates :entry_time, presence: true validates :checkout_time, presence: true
validate :checkout_time_must_be_in_future
def checkout_time_must_be_in_future
if checkout_time.present? && checkout_time <= Time.nowerrors.add(:checkout_time, "must be in the future")
    endend end

[2:36 PM, 10/26/2023] Pavithran: <div class="container mt-5">
<div class="row">
<div class="col-md-8 mx-auto">
<div class="card">
<div class="card-header bg-primary text-white">
<h2 class="card-title text-center">Displaying Slot Details</h2>
</div>
<div class="card-body">
        <%= form_for(slot_bookings_select_date_path, method: :get, class: "mb-4") do %>
<div class="form-group">
<label for="selected_date">Select Date:</label>
<%= date_field_tag :selected_date, nil, class: "form-control" %>
</div>
<%= submit_tag "Search", class: "btn btn-primary btn-block mt-3" %>
<% end %>

<% if @slot_bookings.any? %>
<div class="table-responsive">
<table class="table table-bordered table-striped">

<thead class="thead-dark">
<tr>
<th class="text-center">Slot Number</th>
<th class="text-center">Vehicle Registration Number</th>
```

```
<th class="text-center">Entry Time</th>
</tr>
</thead>
<tbody>
<% @slot_bookings.each do |booking| %>
<tr>
<td class="text-center"><%= booking.parking_slot_id %></td>
<td class="text-center"><%= booking.vehicle_reg_num %></td>
                    <td class="text-center"><%= booking.entry_time.strftime("%H:%M:%S") %></td>
</tr>
<% end %>
</tbody>
</table>
</div>
<% else %>
<div class="alert alert-info text-center">No slots used on the selected date
</div>
<% end %>
</div>
</div>
</div>
</div>

</div>
[2:36 PM, 10/26/2023] Pavithran: <div class="container mt-5">
<h1 class="display-4">Slot Allocation History for Vehicle</h1>


  <%= form_for(slot_bookings_vehicle_history_path, method: :get, class: "mb-4") do %>
<div class="form-group">
      <label for="vehicle_registration_number">Enter Vehicle RegistrationNumber:</label>
<%= text_field_tag :vehicle_reg_num, nil, class: "form-control" %>
</div>
<%= submit_tag "Search", class: "btn btn-primary mt-3" %>
<% end %>

<% if @vehicle_history.any? %>
<div class="table-responsive">
<table class="table table-bordered table-striped">
<thead class="thead-dark">
<tr>
<th class="text-center">Entry Date</th>
<th class="text-center">Entry Time</th>
<th class="text-center">Check Out Time</th>
<th class="text-center">Slot Number</th>
</tr>
</thead>
<tbody>
<% @vehicle_history.each do |booking| %>
<tr>
```

```
<td class="text-center"><%= booking.entry_time.strftime("%Y-%m-
%d") %></td>
            <td class="text-center"><%= booking.entry_time.strftime("%H:%M:%S") %></td>
<td class="text-center"><%= booking.checkout_time.strftime("
%H:%M:%S") %></td>
<td class="text-center"><%= booking.parking_slot_id %></td>

</tr>
<% end %>
</tbody>
</table>
</div>
<% else %>
<div class="alert alert-info text-center">
No slot allocation history found for the specified vehicle.
</div>
<% end %>
</div>
[2:37 PM, 10/26/2023] Pavithran: <div class="container d-flex justify-content-center align-items-center  vh-100">
<div class="card text-center">
<div class="card-body">
      <h1 class="card-title display-4 mb-4">Welcome to Parking Slot BookingSystem</h1>
      <a href="<%= entry_points_path %>" class="btn btn-primary btn-lg">Select Entry Point</a>
</div>
</div>
def checkout_time_must_be_in_future
if checkout_time.present? && checkout_time <= Time.nowerrors.add(:checkout_time, "must be in the future")
end
<div class="container mt-5">
<h1 class="display-4 text-center">All Vehicles' First Entry Times</h1>
<div class="table-responsive">
<table class="table table-bordered table-striped">
<thead class="thead-dark">
<tr>
<th class="text-center">Vehicle Registration Number</th>
<th class="text-center">First Entry Time</th>
</tr>
</thead>
<tbody>
      <% @first_entry_times.each do |vehicle_registration_number,first_entry_time| %>
<tr>
<td class="text-center"><%= vehicle_registration_number %></td>
<td class="text-center"><%= first_entry_time %></td>
</tr>
<% end %>
</tbody>
</table>
</div>
</div>
```

REFERENCES

[1] Arthur Hylton III and Suresh Sankaran arayanan "Application of Intelligent Agents in Hospital Appointment Scheduling System", International Journal of Computer Theory and Engineering, Vol. 4, August 2012, pp. 625-630.
[2] Yeo Symey, Suresh Sankaran arayanan, Siti Nurafifah binti Sait "Application of Smart Technologies for Mobile Patient Appointment System", International Journal of Advanced Trends in Computer Science and Engineering, august 2013
[3] Cayirli, T, E. Veral, and H. Rosen. (2006). Designing appointmentscheduling systems for ambulatory care services. Health Care Management Science 9, 47–58.
[4] A. Peter Idowu, O. Olusegun Adeosun and K. Oladipo Williams,"Dependable Online Appointment Booking System for Nhis Outpatient in Nigerian Teaching Hospitals", International Journal of Computer Science and Information Technology, vol. 6, no. 4, pp. 59-73, 2014.
[5] https://www.researchgate.net/publication/312946008_Mr_Doc_A_Doctor_Appointment_Application_System



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY