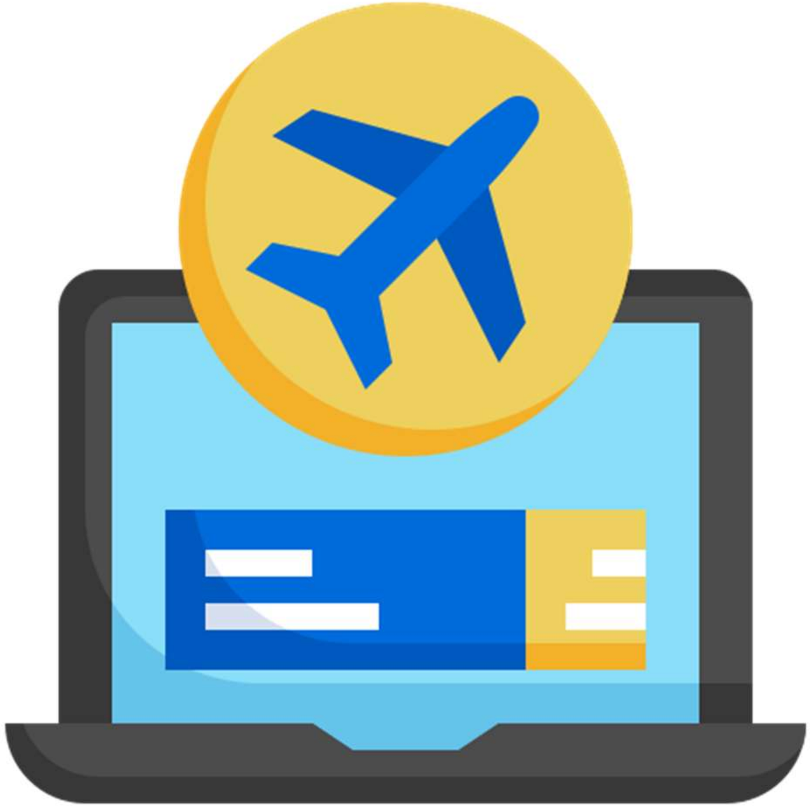


BOOKING MANAGEMNT SYSTEM

BY- Anoop Kumar Yadav

ABSTRACT

The Laravel Booking Management System is a comprehensive web application designed to streamline the process of managing bookings, user accounts, and website content. This document provides an overview of the system's architecture, components, and functionalities. It delves into the implementation details of controllers, models, routes, and migrations, highlighting their roles and interactions within the system. Additionally, it discusses the objectives, methodology, and outcomes of the system, along with concluding remarks on its effectiveness and potential for future enhancements.

OBJECTIVE

The primary objective of the Laravel Booking Management System is to provide businesses with a robust platform for efficiently managing bookings, user accounts, and website content. By leveraging Laravel's powerful features and conventions, the system aims to streamline booking processes, enhance user experience, and improve overall operational efficiency. Key objectives include:

•Facilitating seamless authentication and registration processes for users.
•Enabling administrators to manage bookings, users, and web pages effectively.
•Ensuring data integrity, security, and scalability throughout the system.
•Enhancing user satisfaction through intuitive interfaces and responsive design.
•Promoting best practices and standards for code quality, maintainability, and scalability.

INTRODUCTION

The Laravel Booking Management System is built on the Laravel framework, a popular PHP framework known for its elegance, simplicity, and expressive syntax. The system follows a modular architecture, with distinct components for handling different aspects of functionality. These components include controllers, models, routes, and migrations, each serving a specific purpose in the system's operation.

METHODOLOGY

CONTROLLER :

AuthController :

- Manages authentication-related functionalities such as user login, logout, and registration.
- Utilizes Laravel's built-in authentication features for secure user authentication and session management.
- Implements form validation to ensure data integrity and security during authentication and registration processes.
- Routes user requests to appropriate views and performs necessary actions based on user input

BookingController :

- Handles all aspects of booking management including creation, retrieval, updating, and deletion of bookings.
- Utilizes Laravel's ORM (Object-Relational Mapping) to interact with the database and perform CRUD (Create, Read, Update, Delete) operations on booking entities.
- Implements role-based access control to restrict certain booking functionalities to authorized users.
- Validates user input to ensure the integrity and validity of booking data before processing.

METHODOLOGY

UserController :

- Manages user-related operations such as listing users, adding new users, editing existing users, and deleting users.
- Implements user role management functionalities to assign appropriate roles (e.g., administrator, regular user) to users.
- Validates user input for user creation and updating processes to maintain data consistency and security.
- Implements authentication checks to restrict user management functionalities to authorized administrators.

WebpageController :

- Facilitates the management of web pages within the system, including creation, editing, and deletion of pages.
- Implements access control mechanisms to restrict page management functionalities to authenticated users with appropriate permissions.
- Utilizes Laravel's ORM for database interactions, enabling seamless retrieval and manipulation of webpage data.
- Validates user input to ensure the integrity and validity of webpage content before persisting it to the database.

METHODOLOGY

MODELS :

User Model (User.php) :

- Represents the user entity in the system, encapsulating attributes such as name, email, password, user type, and phone number.

- Defines relationships with other models such as bookings and web pages to establish associations and retrieve related data.

- Implements authentication-related functionalities such as password hashing and token generation for secure user authentication.

- Utilizes Laravel's built-in features like attribute casting and accessors/mutators for efficient attribute manipulation and formatting.

Booking Model (Bookings.php):

- Represents booking entities within the system, encapsulating attributes such as booking name, datetime, status, and user ID.

METHODOLOGY

- Defines relationships with the user model to establish ownership and retrieve user-related information associated with bookings.

- Implements business logic related to booking statuses (e.g., booked, canceled, fulfilled) and booking operations (e.g., updating, canceling).

WebPage Model (WebPage.php):

- Represents webpage entities in the system, encapsulating attributes such as name, slug, HTML content, status, and creator/updater IDs.

- Facilitates the management of web pages and their content, allowing users to create, edit, and delete web pages dynamically.

- Implements relationships with user models to track creator/updater information and establish ownership of web pages.

METHODOLOGY

ROUTES :

Routes (web.php):

- Defines the routing configuration for various HTTP request methods and URIs, mapping them to corresponding controller actions.

- Utilizes Laravel's middleware functionality for implementing cross-cutting concerns such as authentication and role-based access control.

- Organizes routes into logical groups based on functionality (e.g., authentication, booking management, user management) for clarity and maintainability.

- Implements route parameters and resourceful routing conventions for RESTful API design, promoting consistency and adherence to best practices

METHODOLOGY

MIGRATIONS :

create_users_table.php :

- Defines the database schema for the users table, specifying attributes such as name, email, password, user type, and timestamps.

- Establishes constraints such as uniqueness for email addresses and default values for user types to ensure data integrity and consistency.

- Implements indexes and foreign key constraints where necessary to optimize query performance and maintain referential integrity.

create_bookings_table.php:

- Specifies the schema for the bookings table, encompassing attributes such as booking name, datetime, status, and user ID.

- Defines appropriate data types and constraints for each attribute to enforce data integrity and facilitate efficient data storage and retrieval.

- Establishes foreign key relationships with the users table to associate bookings with corresponding users and enable cascading operations.

METHODOLOGY

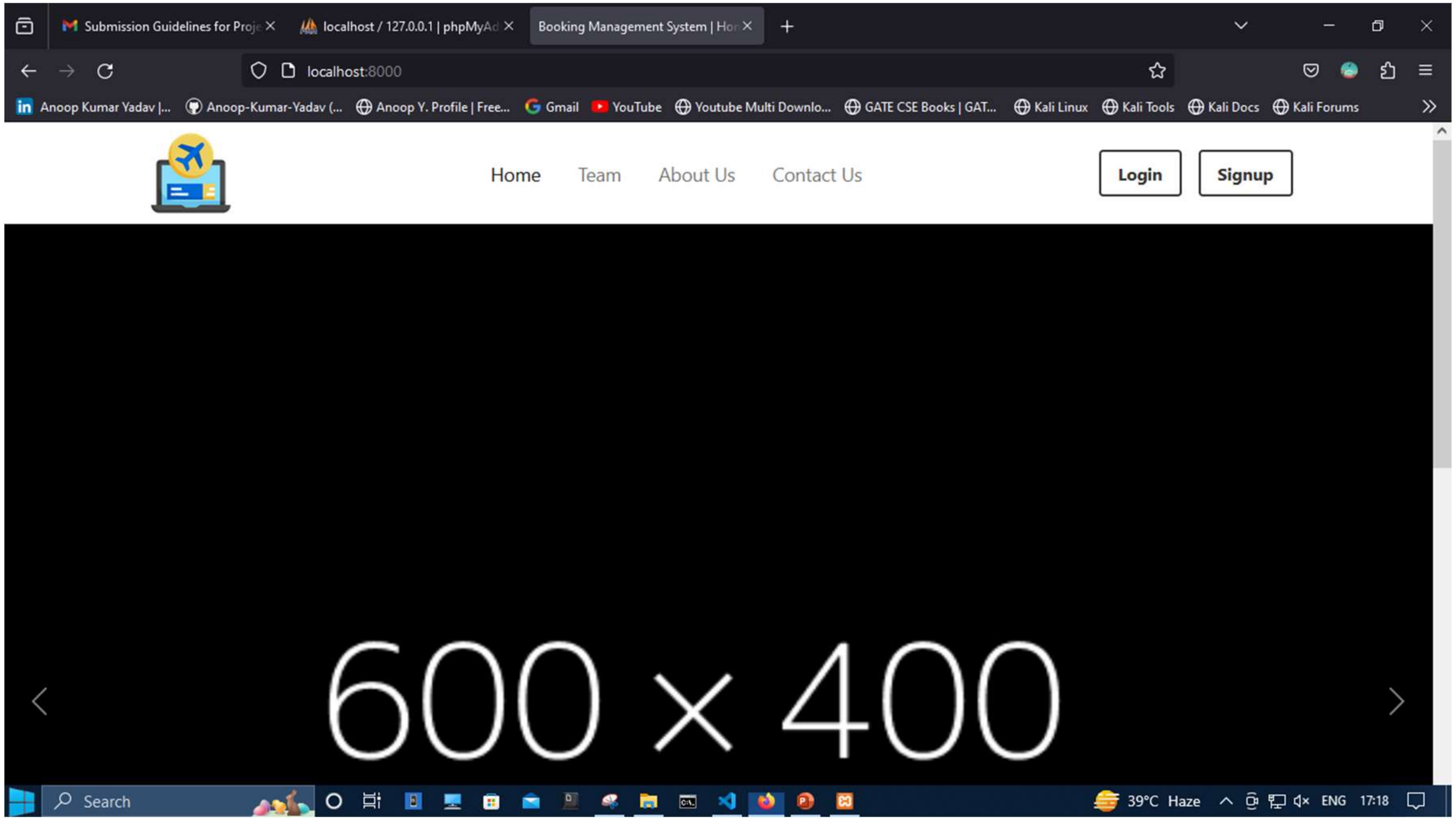
create_webpage_table.php:

- Defines the database structure for the webpage table, including attributes such as name, slug, HTML content, status, and creator/updater IDs.

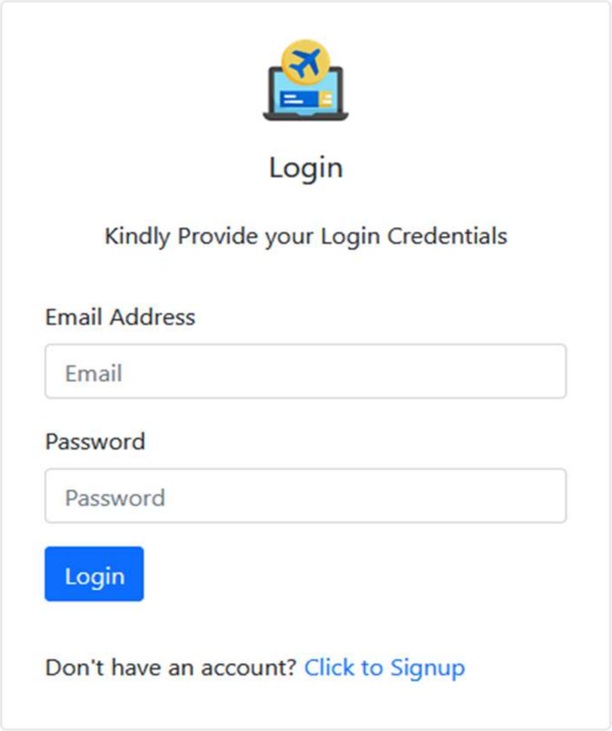
- Implements nullable constraints for optional attributes and default values for status attributes to streamline webpage creation and management.

- Utilizes indexes and foreign keys to optimize query performance and maintain data consistency across related entities.

SCREENSHOTS



Booking Manger System
(Home Page)



Login

Kindly Provide your Login Credentials

Email Address

Email

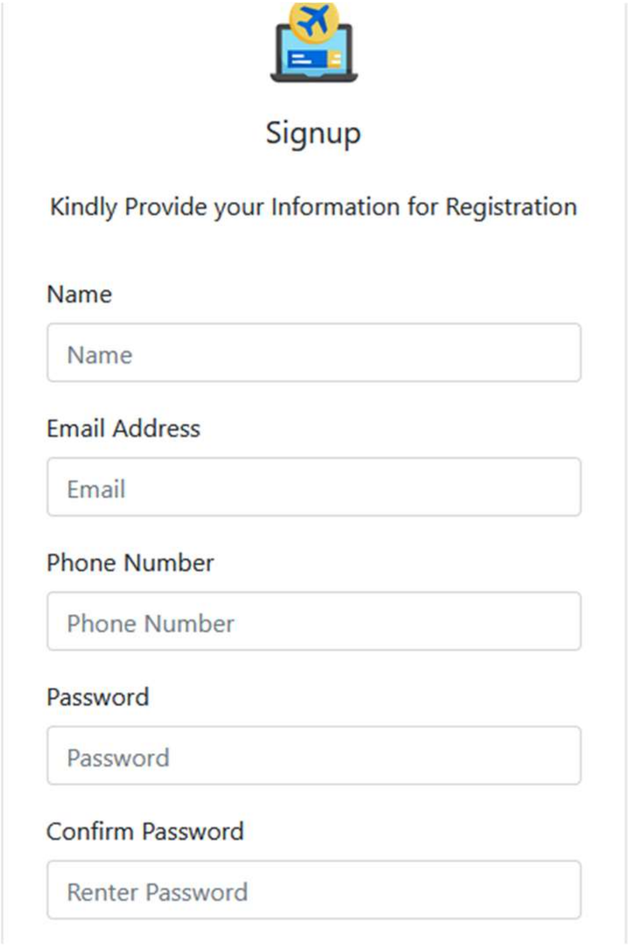
Password

Password

Login

Don't have an account? Click to Signup

Login Page



Signup

Kindly Provide your Information for Registration

Name

Name

Email Address

Email

Phone Number

Phone Number

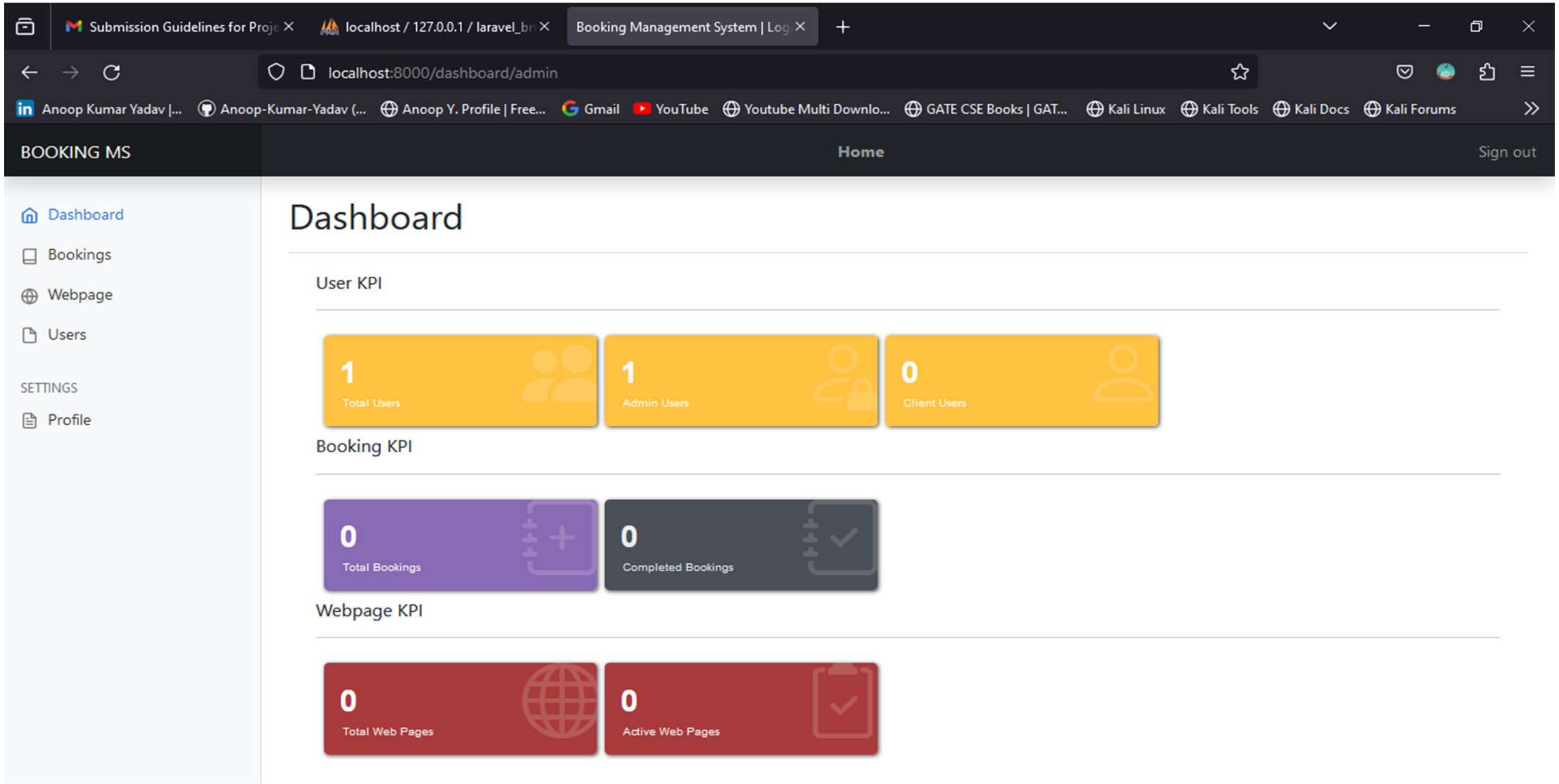
Password

Password

Confirm Password

Renter Password

Sign-Up



Dashboard

User KPI

1 Total Users

1 Admin Users

0 Client Users

Booking KPI

0 Total Bookings

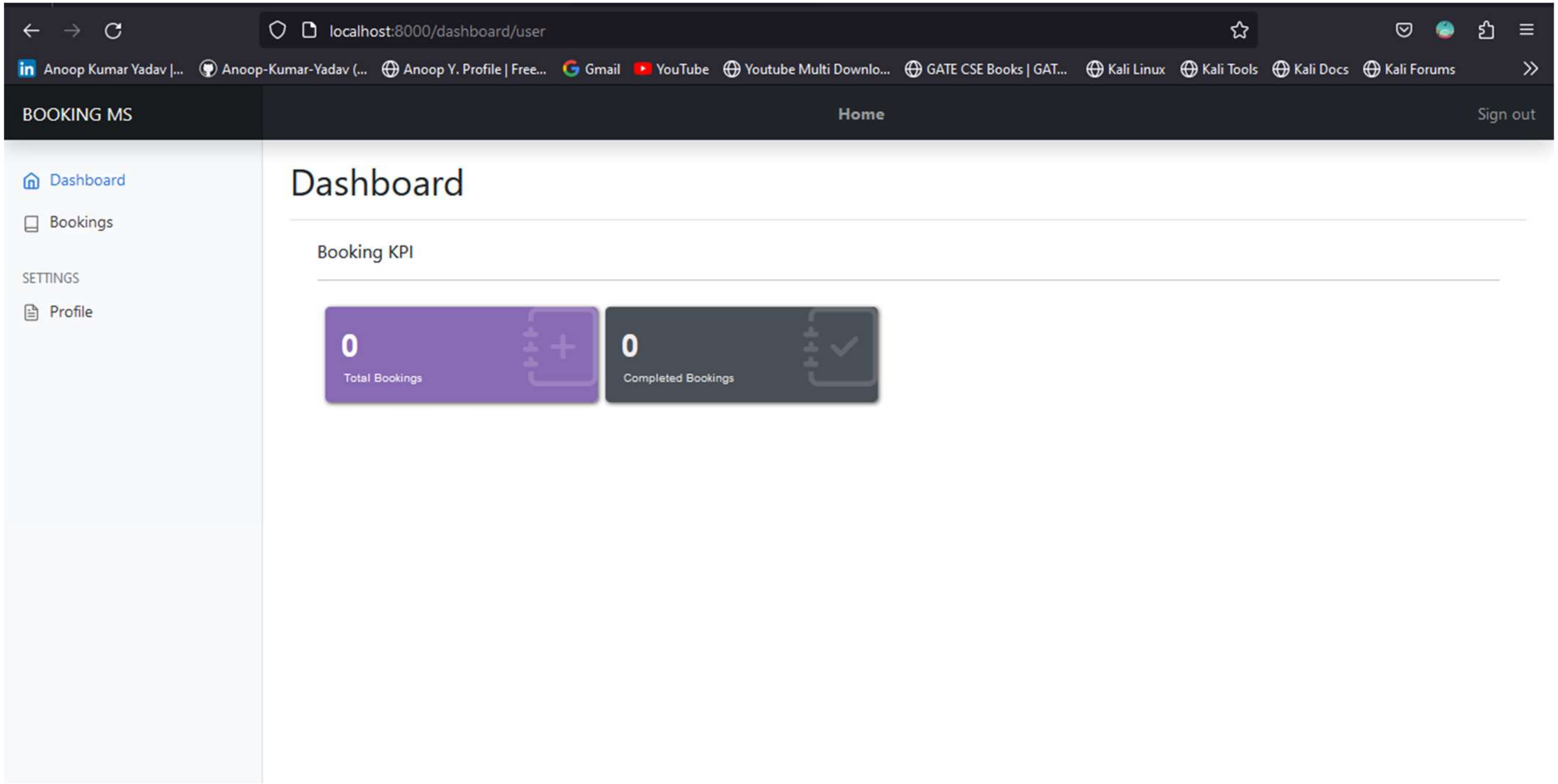
0 Completed Bookings

Webpage KPI

0 Total Web Pages

0 Active Web Pages

Admin-Dashboard



localhost:8000/dashboard/user

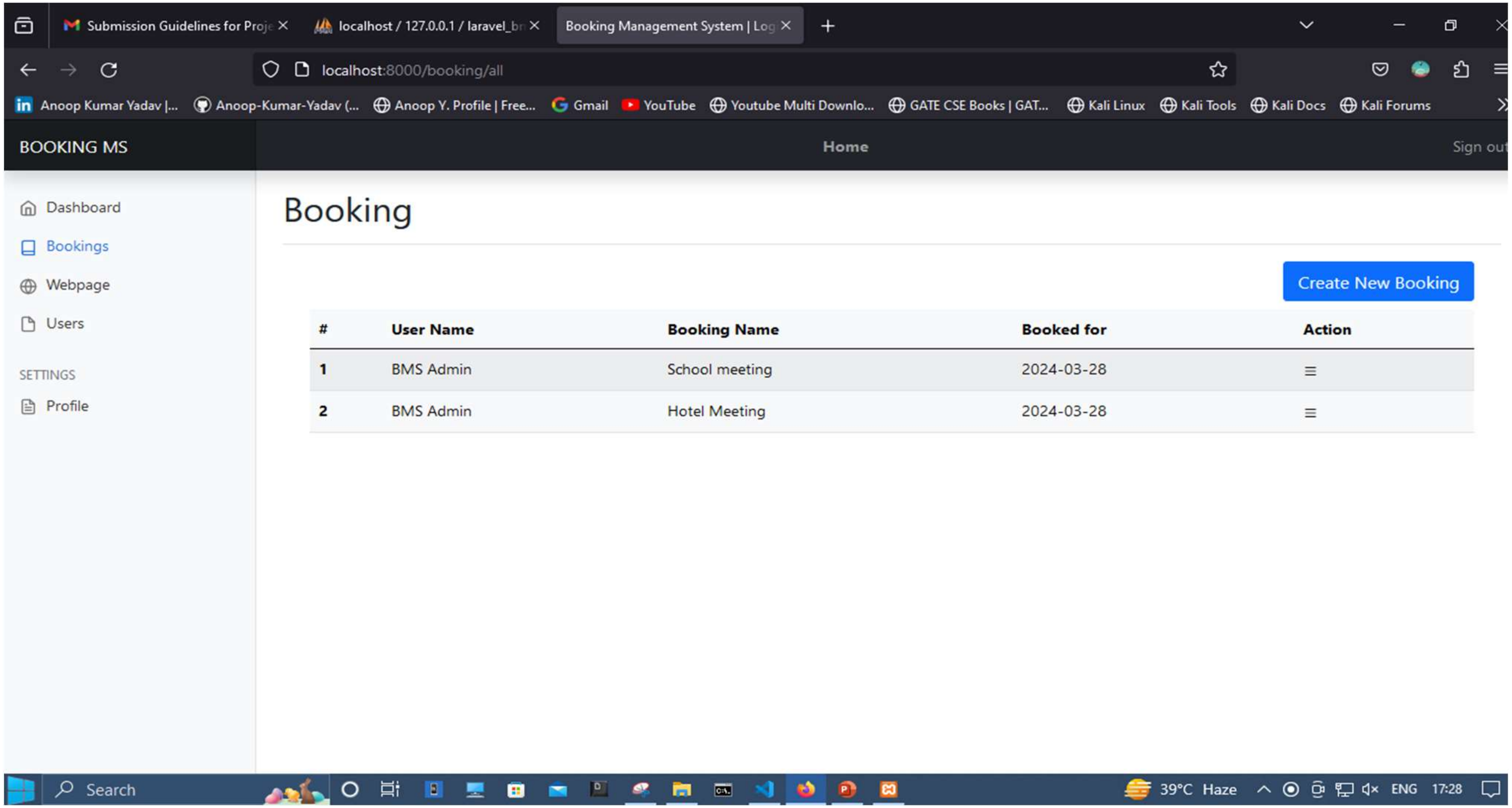
Dashboard

Booking KPI

0 Total Bookings

0 Completed Bookings

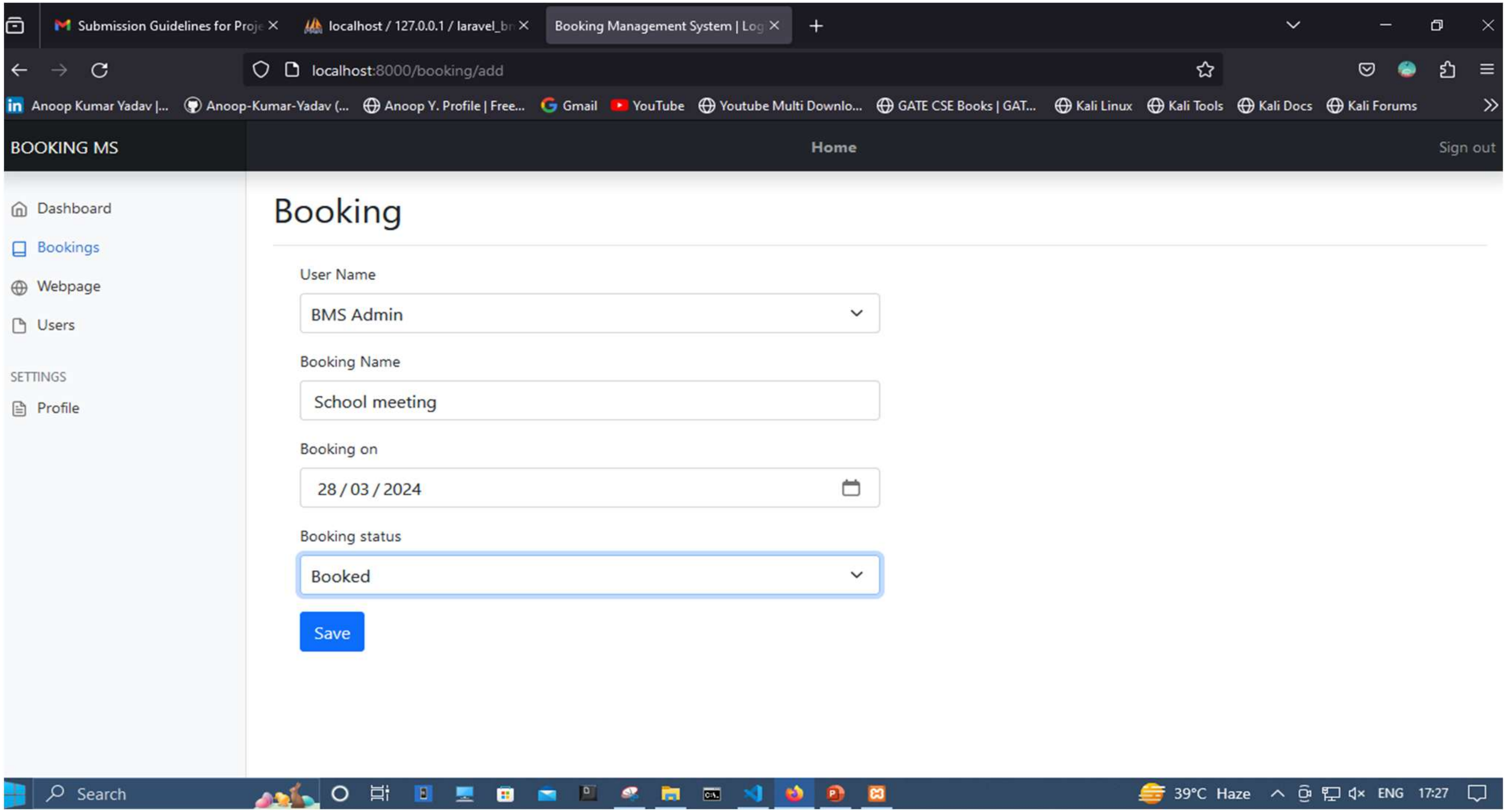
User-Dashboard



Booking

#	User Name	Booking Name	Booked for	Action
1	BMS Admin	School meeting	2024-03-28	≡
2	BMS Admin	Hotel Meeting	2024-03-28	≡

Booking-Table-Page
(Same for Admin & user)


Submission Guidelines for Proje
localhost / 127.0.0.1 / laravel_b
Booking Management System | Log
localhost:8000/booking/add
BOOKING MS
Home
Sign out
Dashboard
Bookings
Webpage
Users
SETTINGS
Profile
Search
39°C Haze
ENG
17:27

Booking

User Name: BMS Admin

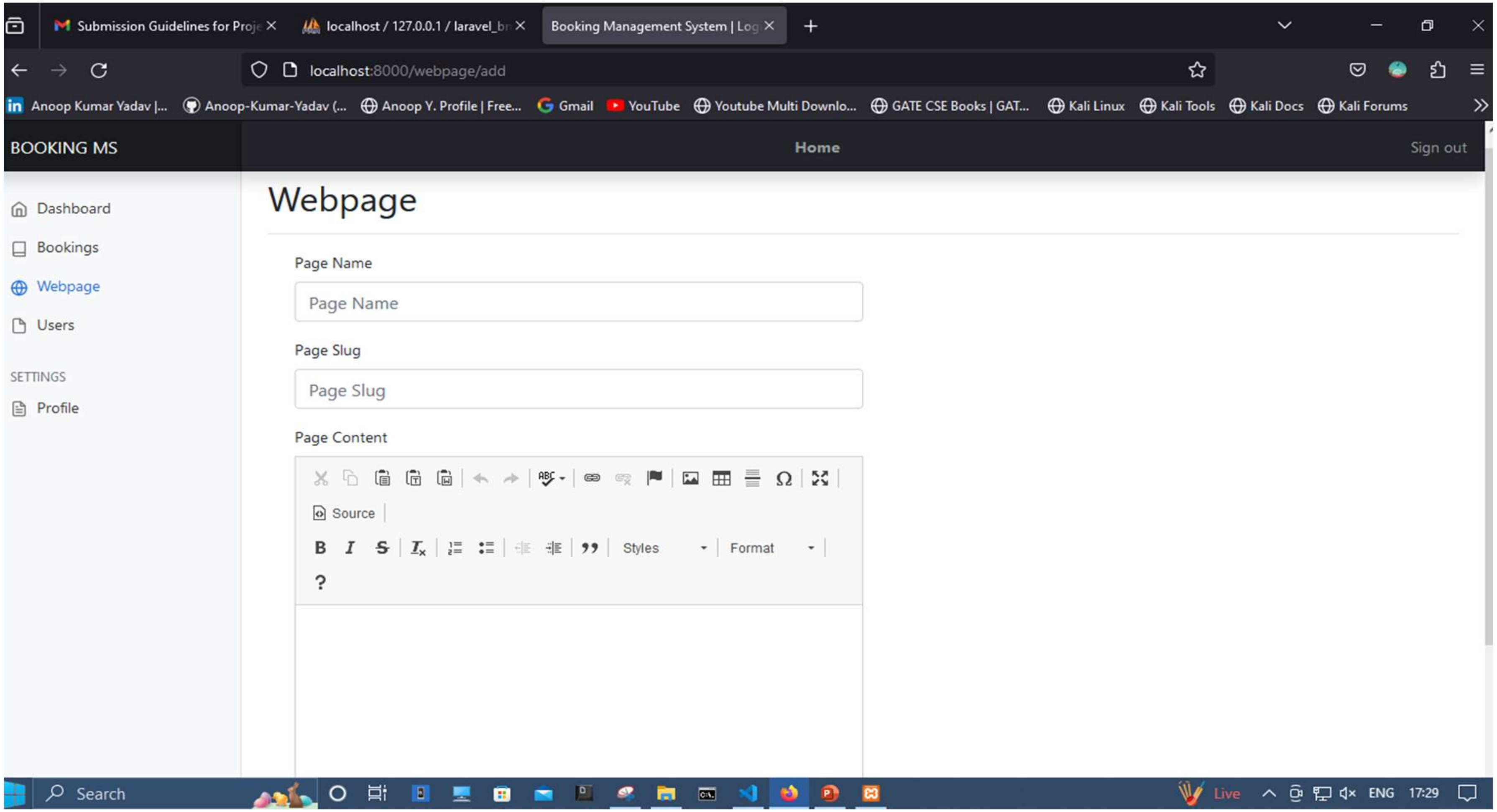
Booking Name: School meeting

Booking on: 28 / 03 / 2024

Booking status: Booked

Save

Booking-Create-Page
(Same for Admin & user)



Webpage

Page Name

Page Name

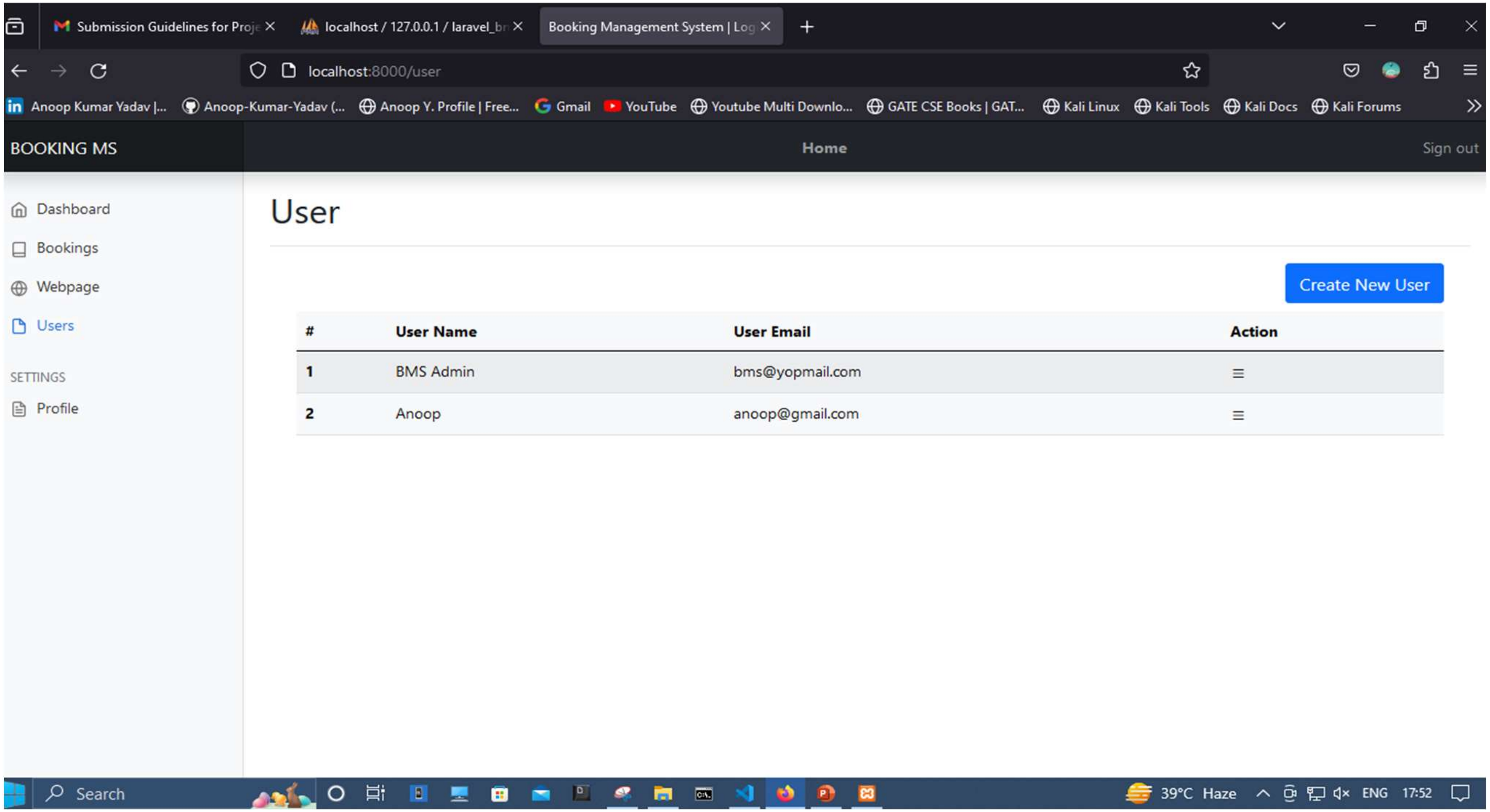
Page Slug

Page Slug

Page Content

WebPage-Create-Page

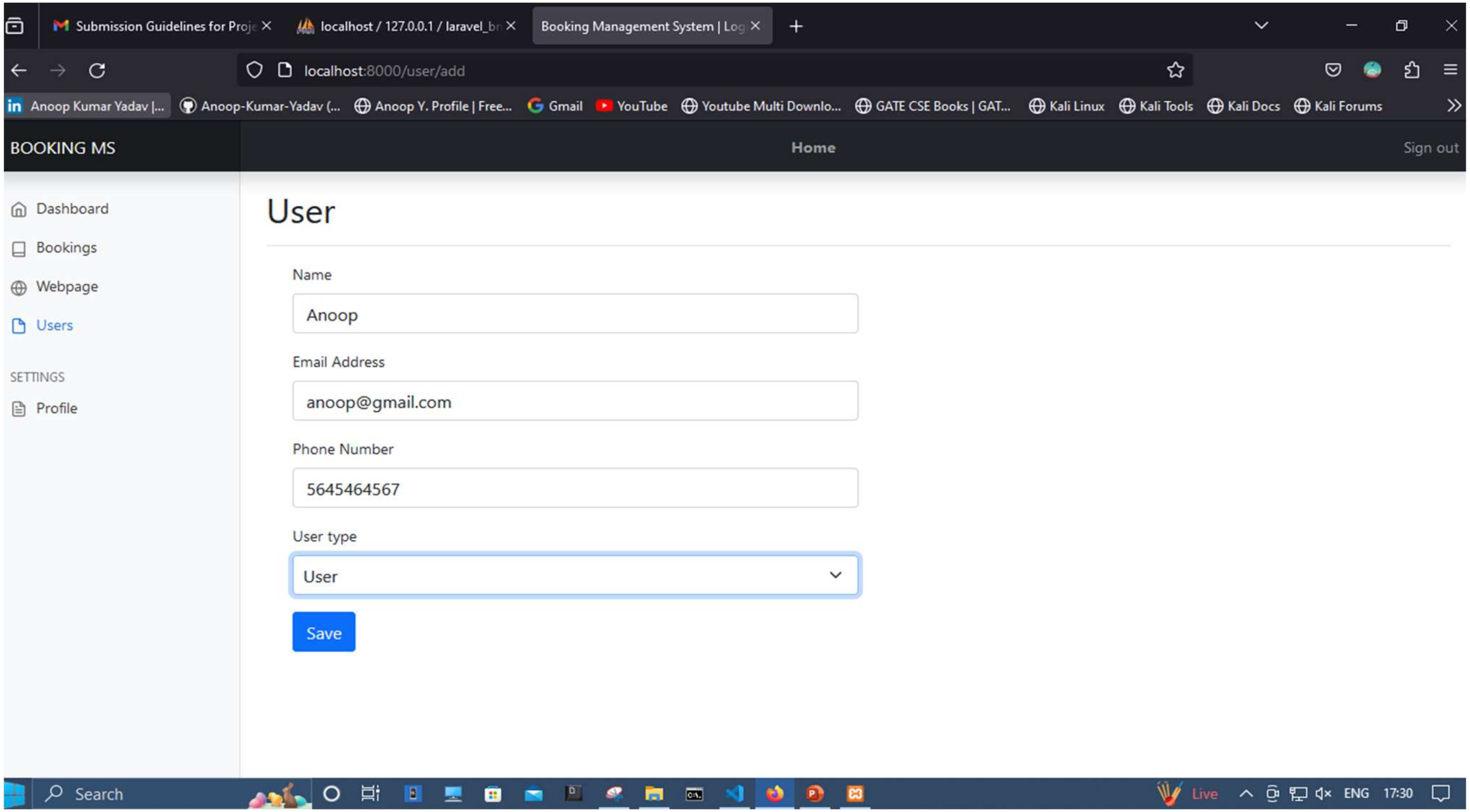
(Only For Admin)



User

#	User Name	User Email	Action
1	BMS Admin	bms@yopmail.com	≡
2	Anoop	anoop@gmail.com	≡

User-Table-Page
(Only For Admin)



User

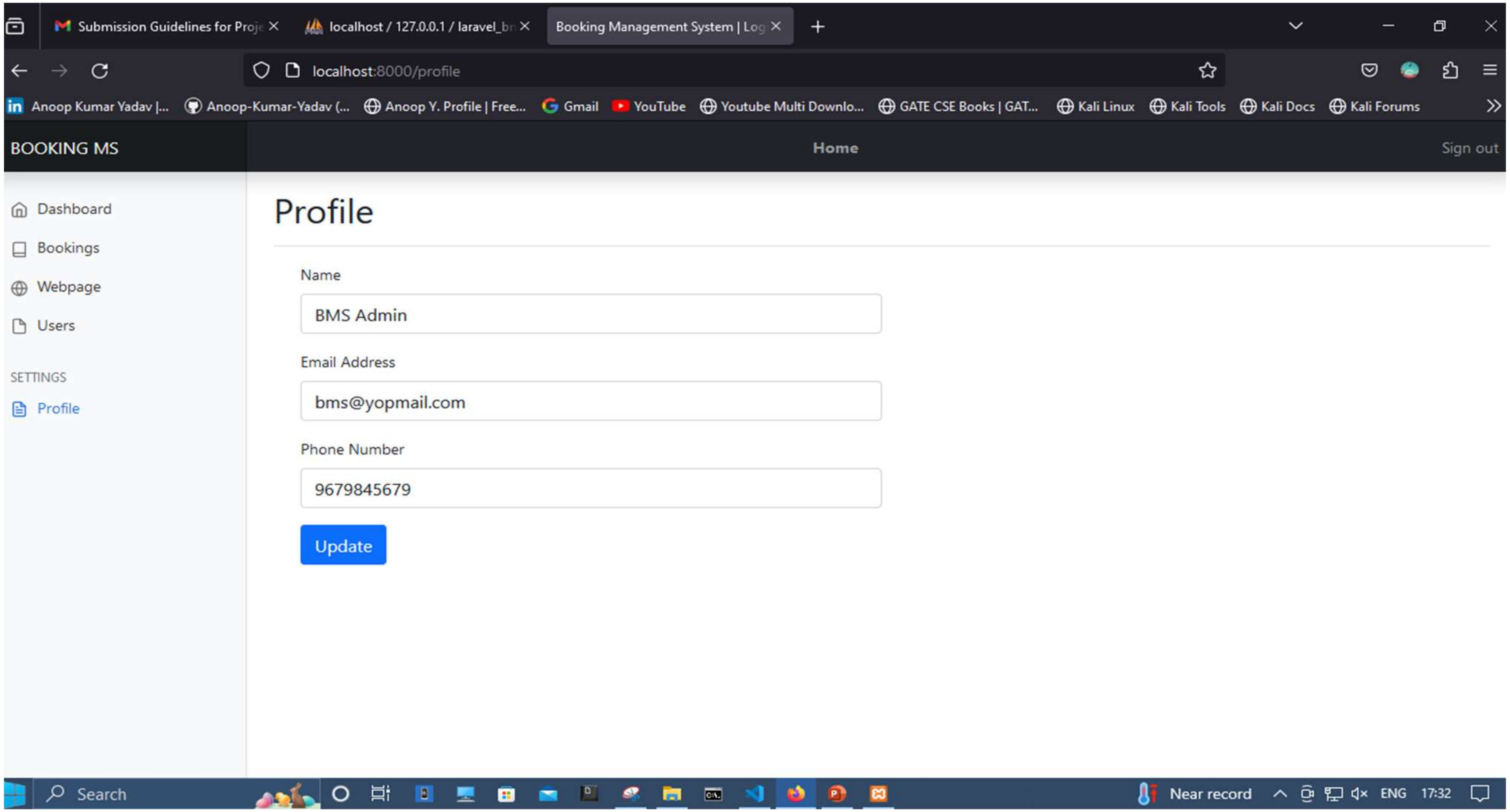
Name: Anoop

Email Address: anoop@gmail.com

Phone Number: 5645464567

User type: User

User-Create-Page
(Only For Admin)



Profile

Name

BMS Admin

Email Address

bms@yopmail.com

Phone Number

9679845679

Profile-Page
(Same For User & Admin)

CONCLUSION:

The Laravel Booking Management System embodies a comprehensive approach to managing bookings and user interactions, leveraging Laravel's rich ecosystem of features and conventions. Through meticulous design and implementation of controllers, models, routes, and migrations, the system provides a robust and scalable solution for businesses to streamline booking processes and enhance user satisfaction. By adhering to best practices and standards, the system ensures data integrity, security, and maintainability, thus empowering businesses to effectively manage their booking operations with confidence.