NANDHA ENGINEERING COLLEGE

ERODE–638052 (Autonomous)

(Affiliated to Anna University, Chennai)

PROJECT BASED LEARNING REPORT

Submitted by

SANTHOSH RAJ P (22AI047)

in partial fulfilment for the award of the degree of

B.Tech (ARTIFICIAL INTELLIGENCE AND DATA SCIENCE)

22AIC15 – FULL STACK DEVELOPMENT

PROJECT BASED LEARNING ON
RESTAURANT TABLE BOOKING WEBSITE

NANDHA ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

This is to certify that the project work entitled **“RESTAURANT TABLE BOOKING WEBSITE”** is the Bonafide work of **SANTHOSH RAJ P(22AI047)** who carried out the work under my supervision.

Signature of the HOD

Dr. P. Karunakaran,

Head of the Department,

Department of AI &DS,

Nandha Engineering College,

Erode – 638052.

Signature of the Supervisor

Ms. M. Senthamarai,

Assistant Professor,

Department of AI & DS,

Nandha Engineering College,

Erode – 638052.

Submitted for the project Viva Voce examination held on ______________

ACKNOWLEDGEMENT

I express my thanks to our beloved Chairman of Sri Nandha Educational Trust **Thiru.V. Shanmugan** and our beloved Secretaries, **Thiru**. **S. Nandhakumar Pradeep** of Sri Nandha Educational Trust and **Thiru**. **S**. **Thirumoorthi** of Nandha Educational Institutions for providing me all the basic amenities to complete the course successfully.

I specially thank **Dr. S. Arumugam**, Chief Executive Officer of Nandha Educational Institutions for his affection and support in all aspects that have made me complete the course successfully.

I wish to convey my earnest gratefulness to our cherished Principal of Nandha Engineering College, **Dr. U. S. Ragupathy**, M.E., Ph.D., for his constant support in my successful completion of my project work.

I articulate my genuine and sincere thanks to our dear hearted Head of the Department of Artificial Intelligence and Data Science **Dr. P. Karunakaran**, M.E., Ph.D., and **Ms. T. Indhumathi**,M.E., Assistant Professor,Department of Artificial Intelligence and Data Science who has been the key spring of motivation to me throughout the completion of my course and my project work.

I wish to convey my heartfelt thanks to my beloved Project Supervisor **Ms. M. Senthamarai**, M.E., Assistant Professor, Department of Artificial Intelligence and Data Science for her continuous monitoring of the project work.

I am very much gratified to all teaching and non-teaching staff of our department who were direct and indirect strokes throughout my progress. I would like to acknowledge my heartfelt thanks to my parents and my friends who have supported me with their conditional love and encouragement. Finally, I would like to thank the Almighty for his blessings.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
1	ABSTRACT	5
2	SOFTWARE AND HARDWARE REQUIREMENTS	6
3	SOFTWARE DESCRIPTION	7
4	PROJECT DESCRIPTION	9
5	APPENDIX	13
6	SCREENSHOTS	18
7	CONCLUSION	22
8	SCOPE FOR FUTURE ENHANCEMENT	23
9	REFERENCES	24

CHAPTER 1

ABSTRACT

With the rapid increase in digital reservations in the hospitality industry, modern diners now expect seamless, secure, and visually engaging platforms to book tables at their preferred restaurants. However, many existing booking websites fall short due to outdated user interfaces, lack of interactivity, non-personalized options, and poor mobile responsiveness. These shortcomings can lead to a poor user experience, reduced trust, and lower customer retention.

In this project, we propose a full stack web application for restaurant table booking, built using modern web development technologies—HTML5, CSS3, and Bootstrap for the frontend, with optional backend integration via Node.js and package management using npm. The platform simulates a fully functional restaurant booking system, offering key modules such as restaurant selection, date/time scheduling, dish pre-selection, user login, and confirmation page.

The user interface follows a component-based design and employs Bootstrap’s responsive grid system to ensure optimal performance across devices. Interactive functionalities—such as selecting a restaurant, choosing a time slot, and viewing dish summaries—are implemented using JavaScript, resulting in a fluid and intuitive experience.

Unlike static restaurant brochures, this system offers real-time features that replicate the convenience of modern online booking platforms. Through this application, we aim to provide a user-friendly, aesthetically pleasing, and efficient reservation experience tailored for restaurant customers. This project also serves as a practical example of how students and developers can create scalable, customer-focused web solutions using standard front-end technologies without requiring complex backend systems or third-party plugins.

CHAPTER 2

SOFTWARE AND HARDWARE REQUIREMENTS

HARDWARE REQUIREMENTS

System	**:**HP
Processor	**:**12th Gen Intel(R) Core(TM) i5
RAM	**:**16 GB
Storage	**:**1 TB HDD/SSD

SOFTWARE REQUIREMENTS

Operating System	**:** Windows 11
Frontend	**:** HTML5, CSS3, JavaScript, Bootstrap 5
Backend	**:** Node.js (basic dependency setup)
IDE Used	**:** Visual Studio Code
Browser	**:** Chrome / Microsoft Edge

CHAPTER 3

SOFTWARE DESCRIPTION

HTML5, CSS3 & Bootstrap 5 (Frontend Technologies):

HTML5 provides the structural foundation of the web application, using semantic tags like <header>, <section>, and <footer> to enhance readability and accessibility. It also supports native features such as multimedia and form validation. CSS3 is used to style the layout, colors, fonts, and animations, bringing visual appeal to the interface. Together with Bootstrap 5—a responsive CSS framework—developers can quickly build mobile-friendly designs using its grid system and ready-made components like buttons, modals, and navbars. This combination ensures a consistent, attractive, and responsive user experience across all devices in the restaurant booking platform.

JavaScript (Scripting Language):

JavaScript is used to add interactivity and dynamic behaviour to the website. In this project, it powers key features such as table booking validation, time slot updates, cart management, and user feedback through modals. It allows real-time updates without reloading the page, enhancing user experience. JavaScript also enables event handling and DOM manipulation, making the interface more responsive and engaging. Its modular nature supports easy future integration with backend services or APIs.

Node.js &npm (Runtime & Dependency Management):

Node.js is a server-side JavaScript runtime used in this project primarily for managing development tools. While the project is frontend-focused, Node.js enables the use of npm (Node Package Manager) to install and manage packages like Bootstrap and jQuery. This setup lays the groundwork for future backend integrations, such as APIs or authentication

MongoDB(Database):

MongoDB is a NoSQL, document-based database that stores data in flexible JSON-like formats. It is ideal for modern web applications due to its high scalability, performance, and ability to handle large, complex data. In this project, MongoDB is used to store user details, table bookings, and reservation records. It works seamlessly with Node.js through tools like Mongoose, which help define data models and manage database operations efficiently without needing a fixed schema.

Visual Studio Code (Integrated Development Environment):

Visual Studio Code (VS Code) is a powerful and lightweight source-code editor developed by Microsoft. It supports HTML, CSS, JavaScript, and Node.js with intelligent features like IntelliSense (code suggestions), integrated terminal, realtime error detection, and Git integration. In this project, VS Code provides a seamless development environment for writing, debugging, and organizing the project's frontend and dependency setup. Its wide range of extensions and themes further enhances productivity for full stack developers.

CHAPTER 4

PROJECT DESCRIPTION

1. Contents of the Project

The key files and folders included in the project are:

Frontend (HTML, CSS, JavaScript):

- **HTML Pages**
  - index_main.html – Homepage with navigation and call to action.
  - menu.html – Interactive menu with dish selection and price filters.
  - location.html – Restaurant selection with filter options (cuisine, type, timing).
  - booking.html – Main booking form with real-time time slot validation and summary display.
  - signup.html / login.html – User registration and login pages.
  - contact.html – Contact form for customer inquiries.
- **CSS Files**
  - style.css – Base styles and shared components (e.g., navbar, layout).
  - booking.css – Custom styles for the booking page.
  - contact.css – Layout and responsiveness for the contact page.
- **JavaScript**
  - Embedded in booking.html, menu.html, and location.html
  - Handles DOM manipulation, session/local storage, booking validation, modal confirmations, and form interactivity.

Backend (Node.js)

- server.js – Entry point to run the Express server.
- /routes/bookingRoutes.js, /routes/userRoutes.js – Defines API endpoints for booking and authentication.
- /controllers/bookingController.js – Contains business logic for booking creation and validation.
- /models/Booking.js, /models/User.js – Mongoose schemas for user and booking data.
- /config/db.js – MongoDB connection setup.

Database (MongoDB)

- Users – Stores user credentials and contact details.
- Bookings – Stores table reservation information with date/time, guests, and selected dishes.
- Contacts – Stores submitted user messages from the contact form.

2. Project Initialization and Setup

- HTML files were created manually for all core frontend pages.
- Bootstrap 5 was added via npm and linked via package.json.
- CSS and JS files were structured into modular folders for maintainability.
- Visual Studio Code was used as the development environment.
- Future Node.js server support is partially enabled via npm initialization.

3.User Authentication (Simulated)

- Login and registration forms are created using HTML and styled with Bootstrap.

- User authentication is simulated on the client side—no actual database or backend validation is implemented at this stage.
- Form validations (e.g., matching passwords, required fields) are handled using JavaScript.
- User credentials are not stored or validated—future implementation may include JWT-based auth.

4.Table Booking and Menu Management (Frontend CRUD Simulation) Users can:

- Browse available restaurants
- View restaurant details(Address, Timings,)
- Select dishes from the menu
- Book a table (date, time slot, number of guests, and optional special requests.)

Cart data is managed dynamically using JavaScript and session Storage, and is not persisted once the session ends. Backend cart persistence can be added in future updates.

5. Booking and Checkout Simulation :

- Users can enter personal details (name, email, phone, date, time, number of guests, and special requests) through a structured HTML form on the booking page.
- A **cart summary** of selected dishes, restaurant info, and total price is displayed on the cart page.
- There is currently no payment gateway or database integration—intended for future enhancement

6.Frontend Features

- Responsive and modern UI designed using Bootstrap 5

- Navigation bar, footer, carousel, product grids, and cards are styled using Bootstrap classes
- Pages included:
    - Home (index.html) – Introduction to the platform and navigation hub
    - Menu (menu.html) – Grid view of dishes with search, filters, and item selection
    - Restaurants (location.html) – Display of restaurant cards with timing, type, and cuisine
    - Booking Page (booking.html) – Collects user information, shows selected items, and confirms reservations
    - About Page (about.html) – Overview of the website's purpose and features
    - Contact Page (contact.html) – Form for user feedback and inquiries

7.API Routes and Integration

While the current version is static and frontend-only, the project is designed to support future backend development:

- **GET /restaurants** – Fetch all restaurants with details such as name, address, cuisine, and operating hours
- **GET /menu/:restaurantId** – Retrieve the full menu for a selected restaurant
- **POST /bookings** – Submit a new table booking including user info, selected dishes, date, time, and guest count
- **GET /user/bookings** – Retrieve all bookings associated with a logged-in user
- **POST /auth/register** – Register a new user with name, email, password, and phone number
- **POST /auth/login**– Authenticate existing users and return a JWT token for session management

CHAPTER 5

APPENDIX

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Restaurant Website </title>
    <link rel="stylesheet" href="style.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css" integrity="sha512-Fo3rlrZj/k7ujTnHg4CGR2D7kSs0v4LLanw2qksYuRlEzO+tcaEPQogQ0KaoGN26/zrn20ImR1DfuLWnOo7aBA==" crossorigin="anonymous" referrerpolicy="no-referrer" />
</head>
<body>
    <div class="top-cart-section">
        <button id="cartButton" class="cart-button" onclick="openCart()">
            🛒 Cart <span id="cartCount" class="cart-count">0</span>
        </button>
    </div>

    <div class="top">
        <div class="logo">FOODIES</div>
        <div class="nav">
            <a href="index.html">Home</a>
            <a href="menu.html">Menu</a>
            <a href="about.html">About</a>
            <a href="location.html">Location</a>
        <a href="contact.html">Contact</a>
    </div>
    <div class="auth-section">
        <div class="auth-buttons" id="authButtons">
            <a href="login.html" class="login-btn">Login</a>
            <a href="signup.html" class="signup-btn">Sign Up</a>
        </div>
        <div class="user-profile" id="userProfile">
            <span class="user-name" id="userName"></span>
            <span class="logout-btn" onclick="handleLogout()">Logout</span>
        </div>
    </div>
    <div class="search">
        <i class="fa-solid fa-search"></i>
        <i class="fa-solid fa-bars"></i>
    </div>
</div>

<div class="heading">
    <div class="left">
        <p>Are You Hungry?</p>
        <h1>Don't Wait</h1>
        <p>Let start to Book the Table now</p>

       <a href="menu.html"><button >Check out Menu</button></a>
    </div>
    <div class="right" id="food">

    </div>
    <div>
    </div>

    <div class="SocialMedia">
        <i class="fa-brands fa-facebook-f"></i>
        <i class="fa-brands fa-twitter"></i>
        <i class="fa-brands fa-instagram"></i>
        <i class="fa-brands fa-whatsapp"></i>
    </div>

</div>

<div class="bottom">
    <div class="menu">
        <div id="food1"></div>
        <div id="food2"></div>
    </div>
</div>

<script>
    // Check authentication state on page load
    window.onload = function() {
```

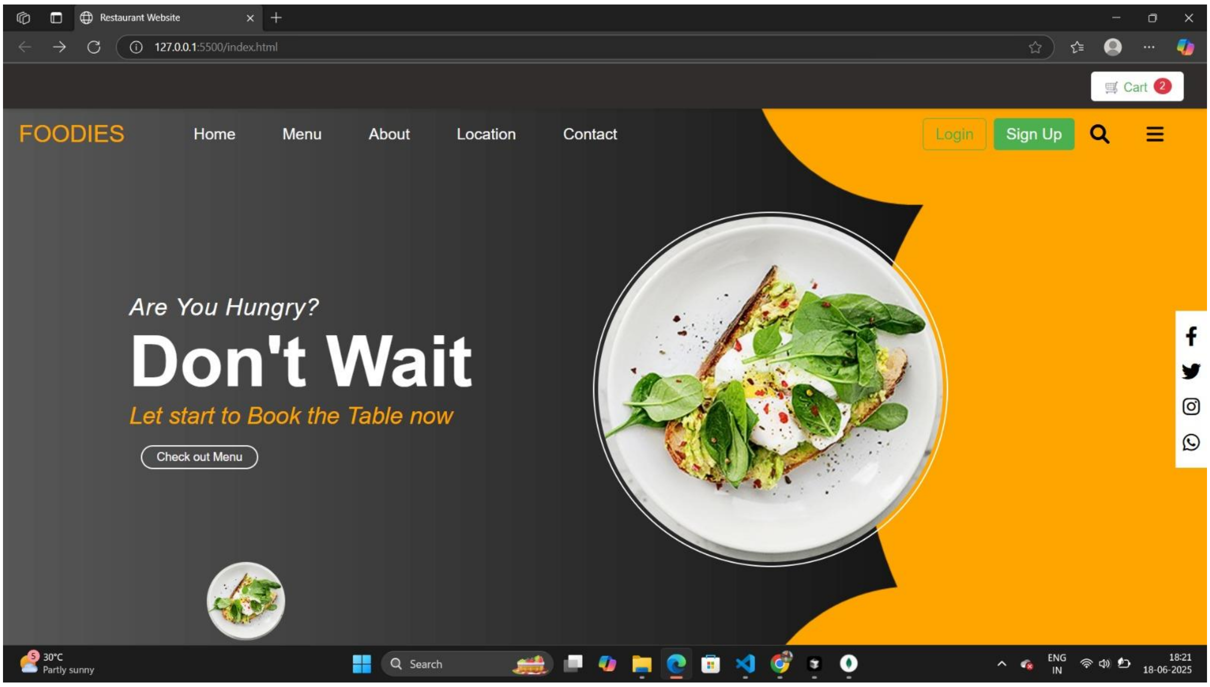
```
    constisLoggedIn = sessionStorage.getItem('loggedIn') === 'true';
    constauthButtons = document.getElementById('authButtons');
    constuserProfile = document.getElementById('userProfile');
    constuserName = document.getElementById('userName');

    if (isLoggedIn) {
        const user = JSON.parse(sessionStorage.getItem('currentUser'));
        userName.textContent = user.name;
        authButtons.style.display = 'none';
        userProfile.style.display = 'flex';
    } else {
        authButtons.style.display = 'flex';
        userProfile.style.display = 'none';
    }
};

function handleLogout() {
    sessionStorage.removeItem('loggedIn');
    sessionStorage.removeItem('currentUser');
    window.location.reload();
}

// Initialize cart data in localStorage if it doesn't exist
if (!localStorage.getItem('cartItems')) {
    localStorage.setItem('cartItems', JSON.stringify([]));
}
```

```
        // Function to update cart count
        function updateCartCount() {
            constcartItems = JSON.parse(localStorage.getItem('cartItems'));
            document.getElementById('cartCount').textContent = cartItems.length;
        }

        // Function to open cart
        function openCart() {
            window.location.href = 'cart.html';
        }

        // Function to remove item from cart
        function removeFromCart(index) {
            constcartItems = JSON.parse(localStorage.getItem('cartItems')) || [];
            cartItems.splice(index, 1);
            localStorage.setItem('cartItems', JSON.stringify(cartItems));
            updateCartCount();
        }
        // Update cart count on page load
        updateCartCount();
    </script>
    <script src="javascript.js"></script>
</body>
</html>
```

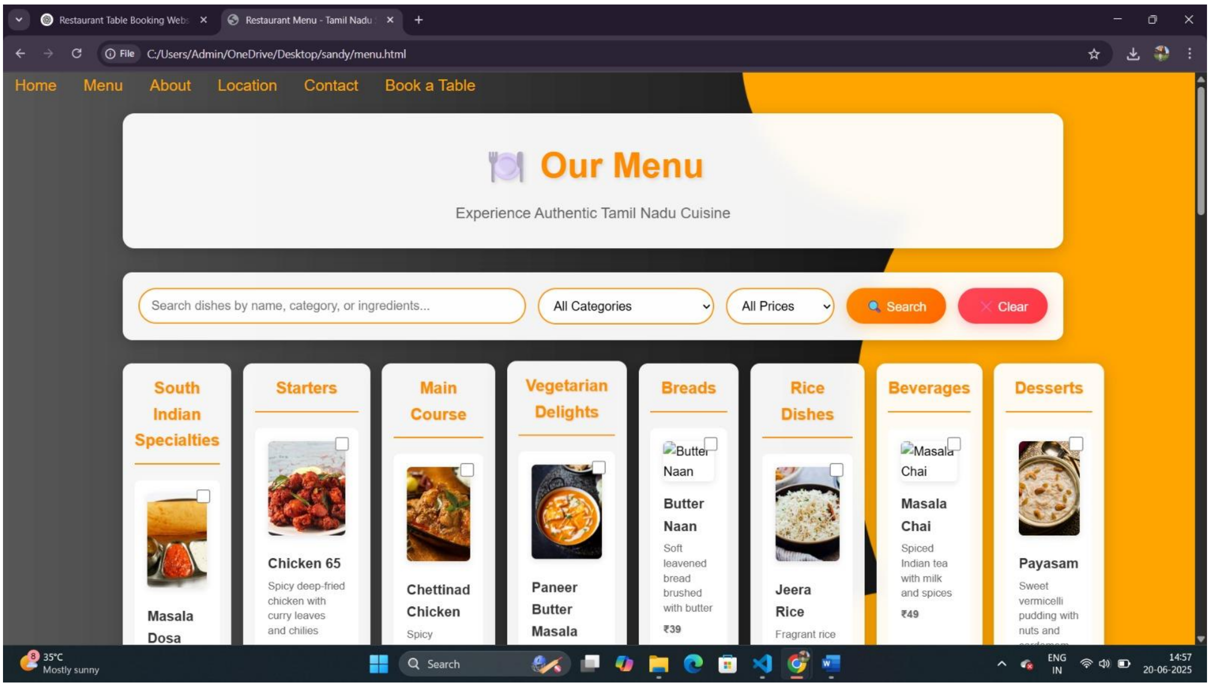
CHAPTER 6

SCREENSHOTS

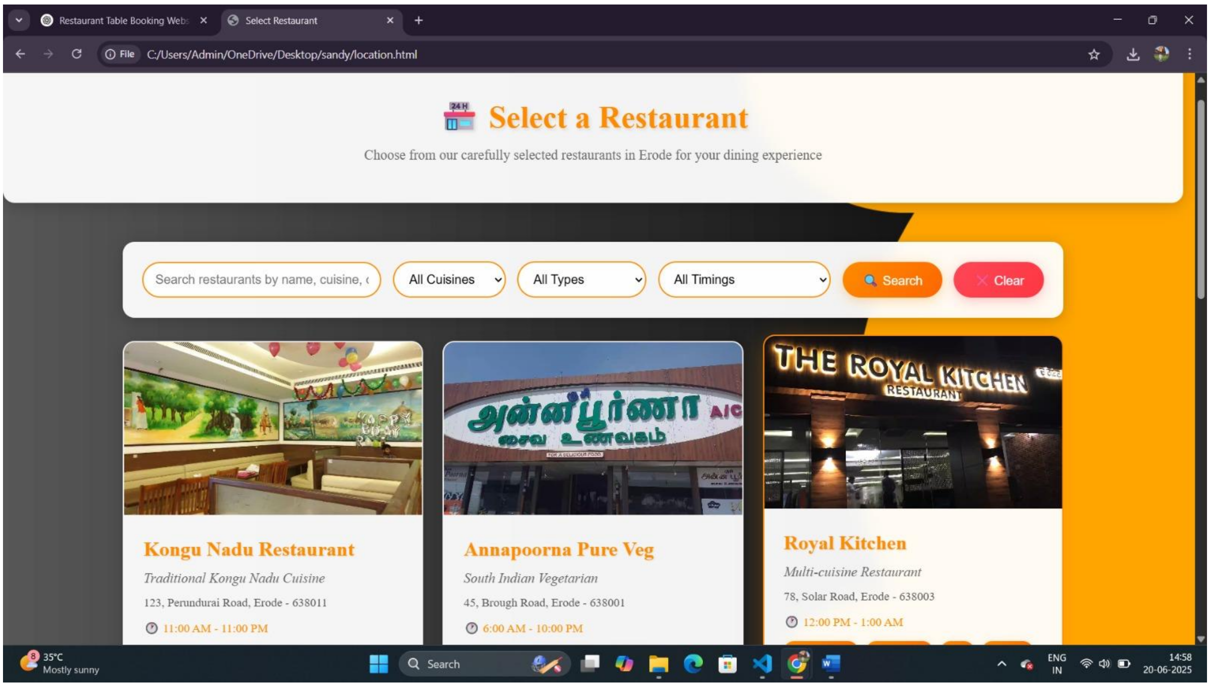
1. Home page:



2. Menu Page:



3. Restaurant Page:


Restaurant Table Booking Web
Select Restaurant
File
C:/Users/Admin/OneDrive/Desktop/sandy/location.html

Select a Restaurant

Choose from our carefully selected restaurants in Erode for your dining experience

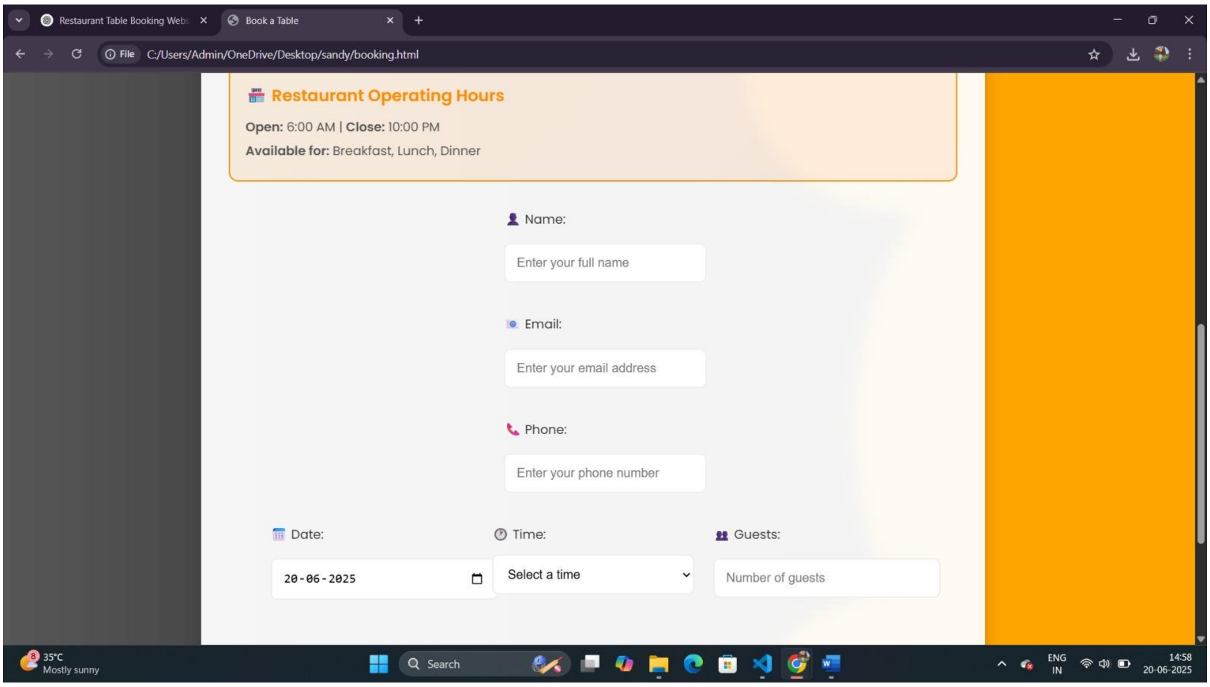
Search restaurants by name, cuisine, c
All Cuisines
All Types
All Timings
Search
Clear
THE ROYAL KITCHEN
RESTAURANT

Kongu Nadu Restaurant
Traditional Kongu Nadu Cuisine
123, Perundurai Road, Erode - 638011
11:00 AM - 11:00 PM

Annapoorna Pure Veg
South Indian Vegetarian
45, Brough Road, Erode - 638001
6:00 AM - 10:00 PM

Royal Kitchen
Multi-cuisine Restaurant
78, Solar Road, Erode - 638003
12:00 PM - 1:00 AM

4. Booking Page:


Restaurant Table Booking Web
Book a Table
File
C:/Users/Admin/OneDrive/Desktop/sandy/booking.html

Restaurant Operating Hours

Open: 6:00 AM | **Close:** 10:00 PM
Available for: Breakfast, Lunch, Dinner

Name:
Enter your full name

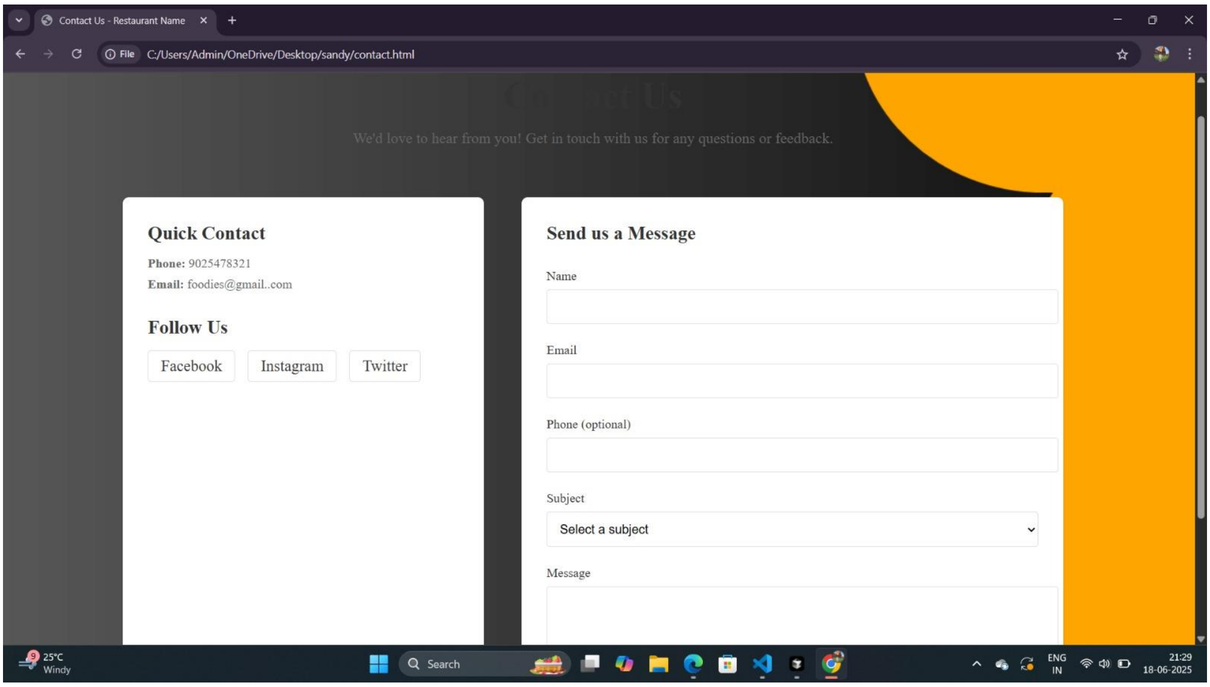
Email:
Enter your email address

Phone:
Enter your phone number

Date: 20-06-2025
Time: Select a time
Guests: Number of guests

35°C
Mostly sunny
Search
ENG
IN
14:58
20-06-2025

5. Contact Page:



Contact Us - Restaurant Name
File
C:/Users/Admin/OneDrive/Desktop/sandy/contact.html
25°C
Windy
Search
ENG
IN
21:29
18-06-2025

Contact Us

We'd love to hear from you! Get in touch with us for any questions or feedback.

Quick Contact

Phone: 9025478321

Email: foodies@gmail..com

Follow Us

Facebook Instagram Twitter

Send us a Message

Name

Email

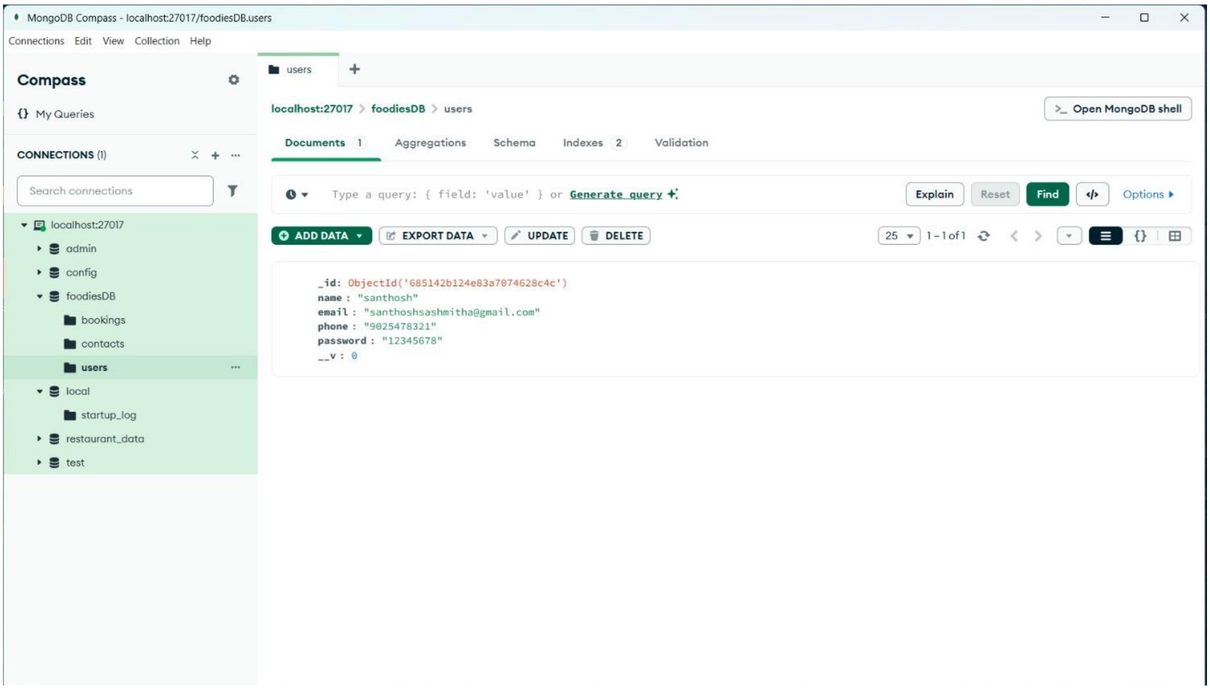
Phone (optional)

Subject

Select a subject

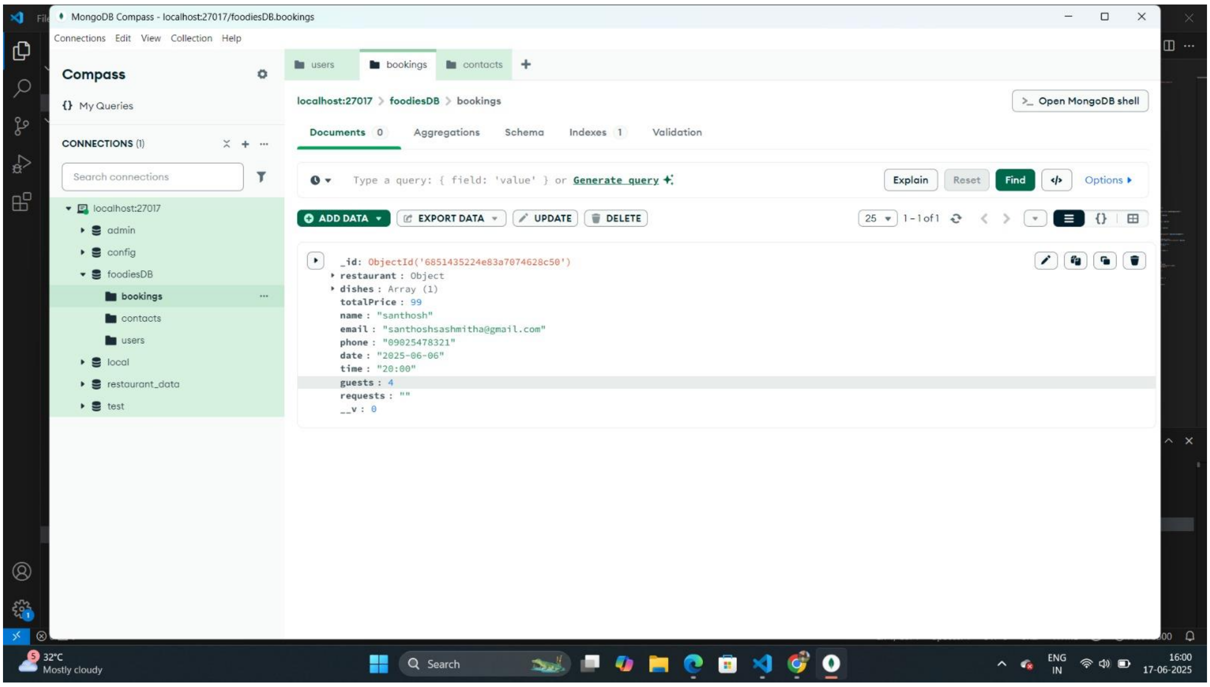
Message

6. User database collection:



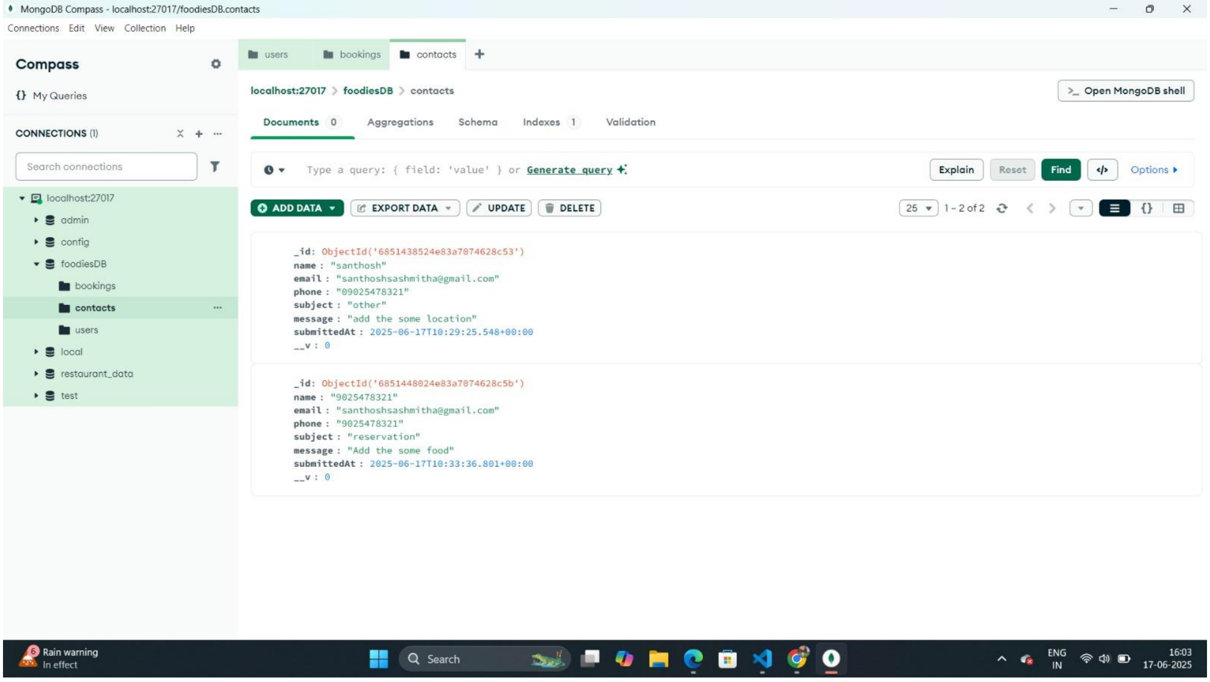
```
_id: ObjectId('685142b124e83a7074628c4c')
name : "santhosh"
email : "santhoshsashmitha@gmail.com"
phone : "9025478321"
password : "12345678"
__v : 0
```

7. Bookings database collection:



```
_id: ObjectId('6851435224e83a7074628c50')
▸ restaurant : Object
▸ dishes : Array (1)
totalPrice : 99
name : "santhosh"
email : "santhoshsashmitha@gmail.com"
phone : "09025478321"
date : "2025-06-06"
time : "20:00"
guests : 4
requests : ""
__v : 0
```

8. Contacts database collection:



```
_id: ObjectId('6851438524e83a7074628c53')
name : "santhosh"
email : "santhoshsashmitha@gmail.com"
phone : "09025478321"
subject : "other"
message : "add the some location"
submittedAt : 2025-06-17T10:29:25.548+00:00
__v : 0
```

```
_id: ObjectId('6851448024e83a7074628c5b')
name : "9025478321"
email : "santhoshsashmitha@gmail.com"
phone : "9025478321"
subject : "reservation"
message : "Add the some food"
submittedAt : 2025-06-17T10:33:36.801+00:00
__v : 0
```

CHAPTER 7

CONCLUSION

This project showcases how modern web development techniques can be applied to build a streamlined and interactive restaurant table booking platform. Using core technologies such as HTML5, CSS3, Bootstrap 5, and JavaScript, the application delivers a responsive and user-friendly frontend experience. Features like dynamic dish selection, time-slot validation, and interactive booking forms simulate a real-world reservation system.

Though currently frontend-focused, the project is structured for scalability and future backend integration using Node.js, Express.js, and MongoDB. It supports the potential for features such as user authentication, booking history, and data persistence via RESTful APIs.

Overall, this project provides a practical foundation for building real-time, customer-focused web solutions in the food and hospitality industry. It also serves as a learning model for students and developers aiming to understand full stack development in a real-world context.

CHAPTER 8

SCOPE FOR FUTURE ENHANCEMENT

Admin Dashboard: Develop a secure backend panel for restaurant managers to manage bookings, update menu items, monitor reservation trends, and handle customer feedback.

Online Payment Integration: Integrate payment gateways like PayPal to allow users to securely pay for reservations and pre-ordered dishes during the booking process.

Personalized Restaurant & Dish Recommendations: Implement a recommendation system using user preferences, past orders, and dining history to suggest relevant restaurants or menu items through AI or rule-based filtering.

Mobile App Development: Build a cross-platform mobile application (using React Native or Flutter) to enable users to easily search, book, and manage their reservations on the go.

Real-time Table Availability & Notifications: Add real-time table status updates and send booking confirmations or reminders via email/SMS notifications.

Loyalty and Rewards System: Introduce a point-based rewards system to improve user engagement and retention.

CHAPTER 9

REFERENCES

[1] M. Haverbeke, *Eloquent JavaScript: A Modern Introduction to Programming*, 3rd ed., No Starch Press, 2018.

[2] J. Robbins, *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics*, 5th ed., O'Reilly Media, 2018.

[3] M. Bowers, *Pro HTML5 and CSS3 Design Patterns*, Apress, 2012.

[4] V. Subramanian, *Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node.js*, 2nd ed., Apress, 2019.

[5] A. Banks and E. Porcello, *Learning React: Functional Web Development with React and Redux*, 2nd ed., O'Reilly Media, 2020.

[6] Node.js Official Documentation – https://nodejs.org/en/docs

[7] Bootstrap 5 Documentation – https://getbootstrap.com/docs/5.3

[8] Mozilla Developer Network (MDN) – https://developer.mozilla.org

[9] S. Bradshaw and E. Brazil, *MongoDB: The Definitive Guide*, 3rd ed., O'Reilly Media, 2019.