
CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

The objective of this project is to design and develop a responsive website that enables users to search and book hotels online. The website should allow users to browse through various sections, including hotel services, room types, amenities, rates and ratings. The website should also provide a secure and user-friendly online booking, with the ability to make reservations. This project will be able to give the option to check if there is a Parking availability for the specific rooms.

1.2 OBJECTIVES

- 1) Parking and No Parking: The ability to check if there is a availability for Vehicle Parking or not available for the each rooms available in the hotel.
- 2) Easy access to the hotel services: It gives an easy access to the details of all the services provided by the hotel.
- 3) Details of all the rooms: It gives the details about the specific room with few images of the room. It also gives a brief description of the room itself and also provide the rates of the room per night.
- 4) Able to filter the rooms: We can be able to filter the rooms according to the room type, no of guests, room price, room size and also the Parking Availability.
- 5) Reservation Option: The customers will be able to provide their details of them so that the hotel receptionist can contact them later.

CHAPTER 2

FUNDAMENTALS OF THE LANGUAGES USED

2.1 INTRODUCTION TO REACT. JS

React is a UI development library based on JavaScript. It is run by Facebook and an open-source developer community. Despite the fact that React is more of a library than a language, it is frequently utilized in web development. The library debuted in May 2013 and has since grown to become one of the most widely used front-end libraries for web development. Beyond only UI, React provides several extensions for full application architecture support, such as Flux and React Native.

Today, React's popularity has surpassed all other front-end development frameworks. This is why:

- Easy production of dynamic online apps: React makes it easier to create dynamic web applications by requiring less coding and providing more functionality, as opposed to JavaScript, which can quickly become difficult.
- React leverages Virtual DOM, which speeds up the development of online apps. Instead of updating all of the components again, as traditional web applications do, Virtual DOM examines the components' prior states and updates only the objects in the Real DOM that have changed.
- Components are the building parts of any React application, and a single app is typically made up of numerous components. These components have their own logic and controls, and they may be reused across the program, reducing the development time significantly.
- As a result, developers frequently nest child components within parent components when developing a Reactapp.

Because data flows in a single path, debugging faults and determining where a problem arises in an application at any given time becomes easier.

- React has a short learning curve since it mixes core HTML and Java Script concepts with some useful enhancements. Still, like with other technologies and frameworks, you'll need to devote some effort to fully comprehending React'slibrary.
- It may be used to create both web and mobile applications: React is well-known for its usage in the creation of web applications, but that's not all it can do. React Native is a popular framework that is derived from React itself and is used to create stunning mobile applications. React may therefore be used to create both web and mobile applications.
- Debug React applications with dedicated tools: Facebook has created a Chrome extension for debugging React apps. Debugging React web applications becomes much easier and faster as a result of this.

The following arguments are more than enough to explain why the React library is so popular and why so many organizations and enterprises are using it. Let's become acquainted with the features of React now.



FIG NO.2.1.1

2.2 EXPRESS.JS

Express.js is a fast and minimalist web application framework for Node.js. It simplifies the process of building web applications by providing a set of flexible and powerful features. Express.js follows the middleware pattern, allowing developers to easily define routes, handle HTTP requests and responses, and incorporate additional functionality through middleware.

At the core of Express.js is the `express` module, which needs to be installed and required in your Node.js project. You can create an Express application by calling the `express()` function, which returns an instance of the Express application.

Express.js provides a routing system that allows you to handle different HTTP methods and URLs. You can define routes using the `app.METHOD()` functions, such as `app.get()`, `app.post()`, `app.put()`, and so on. These functions take a URL pattern and a callback function as parameters.

In addition to routing and middleware, Express.js provides various features like view rendering, static file serving, and template engines. You can use template engines such as EJS, Pug, or Handlebars to generate dynamic HTML pages with data from the server.

Overall, Express.js is a lightweight and flexible web application framework that simplifies the development of Node.js web applications. It provides an intuitive routing system, middleware support, and additional features that make building web applications in Node.js efficient and scalable.

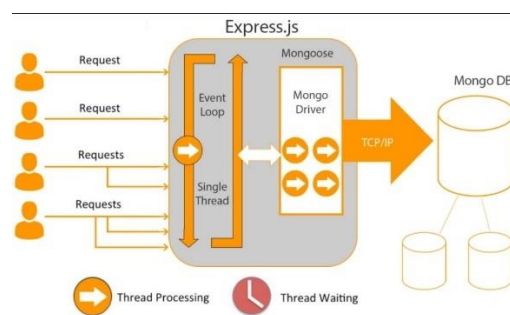


FIG NO.2.2.2

2.1 HTML

HTML (Hypertext Markup Language) is a standard markup language used for creating the structure and presentation of web pages. It provides a set of tags and attributes that define the elements and content of a webpage. HTML uses a hierarchical structure of tags to organize content. Tags are enclosed in angle brackets (`<` `>`) and are typically paired with an opening tag and a closing tag.

The opening tag denotes the start of an element, while the closing tag denotes the end. Some tags, like `
` and ``, are self-closing and do not require a closing tag. HTML tags are used to define different types of elements on a webpage.

For example, the `<h1>` to `<h6>` tags are used for headings of different levels, with `<h1>` being the highest and `<h6>` the lowest. Paragraphs are defined using the `<p>` tag, while links are created with the `<a>` tag and images with the `` tag. Attributes are used to provide additional information about an HTML element.

HTML also supports the use of CSS (Cascading Style Sheets) to control the appearance and layout of elements. CSS rules can be included within `<style>` tags in the `<head>` section of an HTML document, or in an external CSS file that is linked to the HTML document. HTML documents have a basic structure consisting of an opening `<html>` tag, followed by `<head>` and `<body>` tags. The `<head>` section contains metadata about the webpage, such as the title displayed in the browser tab, while the `<body>` section contains the visible content of the webpage. Within the `<body>` section, you can add headings, paragraphs, lists, tables, forms, and other HTML elements to structure and present the content of the webpage. CSS can be used to style these elements, specifying properties like colors, fonts, margins, and layouts. With the use of CSS and JavaScript, HTML forms the foundation for building modern websites and web applications.

2.4 HTML TAGS

HTML tags are similar to keywords in that they determine how the text will be formatted and displayed by a web browser. Tags allow a web browser to distinguish between HTML content and plain text. The three essential sections of an HTML tag are the opening tag, the content tag, and the closing tag. However, some HTML tags aren't-closing tags.

Tag	Description
<code><!--...--></code>	Defines a comment
<code><!DOCTYPE></code>	Defines the document type
<code><html></code>	Root of the HTML document
<code><head></code>	Contains metadata/information for document
<code>
</code>	Single Line Break
<code><body></code>	Document's body
<code><button></code>	Defines a clickable button
<code><div></code>	Defines a section of a document
<code><frameset></code>	Defines a set of frames
<code><frame></code>	Defines a window in frameset
<code><h1>to <h6></code>	Defines headings
<code><p></code>	Defines a paragraph
<code></code>	Defines an image
<code><input></code>	Defines an input control
<code><pre></code>	Defines preformatted text
<code></code>	Defines a section of document
<code><blockquote></code>	Defines a section that is quoted from another source

FIG NO.2.4.1

<title>	Defines a title for the document
	Defines an unordered list
	Defines an ordered list
<table>	Defines a table
<td>	Defines a cell in table
<th>	Defines a heading in table
<tr>	Defines a row in table
<link>	Defines the relationship between a document and an external source

FIG NO.2.4.2

2.5 XHTML

The abbreviation for Extensible HyperTextMarkupLanguage(XHTML) is ExtensibleHyperTextMarkupLanguage. It's a hybrid language that incorporates HTML AND XML. The only difference between XHTML and HTML is that it is more strict. XHTML is HTML that is defined as an XML application. It is compatible with all major browsers.

Despite the fact that XHTML is essentially identical to HTML, writing effective code is more important since XHTML has stricter syntax and case sensitivity than HTML. Unlike HTML, Which necessitates the use of a merciful HTML-specific parser, XHTMLtexts are well-formed and may be parsed by any XML parser. XHTML was intended to help HTML become more extensible and interoperable with other data formats.

2.6 CSS

CSS is a simple technique for incorporating style(such as fonts, colors, and spacing) into web pages. These sections contain tutorials on how to learn and use CSS as well as details on the tools available.

The latest news from the CSS working group is also given. CSS, or CascadingStyleSheets is a simple design language that makes it easy to make web pages seem nice. CSS is a stylesheet language for separating display from content, such as layout, colors, and fonts. This separation can improve content accessibility, providing more flexibility and control in the specification of presentation characteristics, allow multiple web pages to share formatting by specifying the relevant CSS in a separate.css file, which reduces complexity and repetition in the structural content, and allow the.css file to Be cached to improve page load speed between the pages that share the file and its formatting. The ability to offer the same markup page in different styles for distinct rendering techniques, such as on-screen, in print, by voice (through speech-based browser or screen reader), and on Braille-based tactile devices, is also made possible By the separation of formatting and content if the material is accessible on a mobile device, CSS contains rules for different formatting. Cascading gets its name from the priority mechanism used to select which style rule applies when many rules match the same element. This priority cascading strategy is predictable. The WorldWideWeb consortium maintains the CSS specification (W3C).

RFC 2318 specifies the text/css|Internet media type (MIME type) for use with CSS (March 1998). For CSS documents, the W3C offers a free CSS validation service. Other markup languages, such as XHTML, plain XML, SVG, and XUL, permit the usage of CSS in addition to HTML.

2.7 JAVA SCRIPT

JavaScript, abbreviated as JS, is a computer language that, along with HTML and CSS, is one of the most important Web technologies. On the client side, over 97 percent of websites use JavaScript for web page functionality, with third-party libraries frequently used. All major web browsers have a dedicated Java Script engine for running the code on users' devices. JavaScript is a high-level compiled language that is frequently just-in-time and compliant with ECMAScript. First-class functions, Dynamic Typing, and prototype-based object orientation are all included. It's multi-paradigm, allowing for event-driven, functional, or imperative programming.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 HARDWARE REQUIREMENTS

- Windows operating system: Windows 8 or later
- Processor: intel i3 or later
- Memory: 4GB minimum
- Internet connection: not required

3.2 SOFTWARE REQUIREMENTS

- Application: Visual Studio Code
- Language: HTML, CSS, JavaScript, NodeJS, ReactJS

CHAPTER 4

DESIGN

4.1 DESIGN GOALS

Our design mechanism includes few main things:

- 1) The Home where we can be able to interact and see the services provided by the hotel
- 2) The Room where we can be able to see all the rooms available by the hotel with its features like details, info and extra.
- 3) The Booking where we can be able to give the customers details to the hotel so that the hotel receptionist will contact the customer later.
- 4) The Room system consists of
 - Room Type: The customer can select the desired room according to their willing.
 - Guests: The customer can select how many guests will be coming for accommodation
 - Room Price: The customer can filter the rooms according to their price point.
 - Room Size: The customer can filter based on the room sqft
 - Parking Available: The customer can view all the rooms with vehicle parking available by enabling this option.
 - No Parking: The customer can view all the rooms which do not have any vehicle parking by enabling this option.

CHAPTER 5

IMPLEMENTATION

5.1 MODULE 1 FUNCTIONALITY

HOME

```
import React from 'react'
import Hero from '../components/Hero'
import Banner from '../components/Banner'
import {Link} from 'react-router-dom'
import Services from '../components/Services'
import FeaturedRooms from '../components/FeaturedRooms'

const Home = () => {
  return (
    <>
      <Hero>
        <Banner title="Luxurious Rooms" subtitle="Deluxe rooms starting at the most affordable price">
          <Link to="/rooms" className='btn-primary'>
            our rooms
          </Link>
        </Banner>
      </Hero>
      <Services/>
      <FeaturedRooms/>
    </>
  )
}

export default Home;
```

The above code is used to design the home page.

5.2 MODULE 2 FUNCTIONALITY

ROOMS

```
import React from 'react'
import { Link } from 'react-router-dom'
import defaultImg from '../images/room-1.jpeg'
import PropTypes from 'prop-types'

export default function Room({ room }) {
  const { name, slug, images, price } = room;
  return (
    <article className='room'>
      <div className='img-container'>
        <img src={images[0] || defaultImg} alt="single room"></img>
        <div className='price-top'>
          <h6>Rs{price}</h6>
          <p>per night</p>
        </div>
        <Link to={`~/rooms/${slug}`} className="btn-primary room-link">
          Features
        </Link>
      </div>
      <p className='room-info'>{name}</p>
    </article>
  );
}

Room.propTypes={
  room:PropTypes.shape({
    name:PropTypes.string.isRequired,
    slug:PropTypes.string.isRequired,
    images:PropTypes.arrayOf(PropTypes.string).isRequired,
    price:PropTypes.number.isRequired,
  })
}
```

The above code is used to design the room page.

5.3 MODULE 3 FUNCTIONALITY

ROOM FILTER

```
<section className='filter-container'>
  <Title title="search rooms" />
  <form className='filter-form'>
    { /* select type */ }
    <div className='form-group'>
      <label htmlFor='type'>room type</label>
      <select name='type' id='type' value={type} className="form-control" onChange={handleChange}>
        {types}
      </select>
    </div>
    { /* end select type */ }
    { /* guests type */ }
    <div className='form-group'>
      <label htmlFor='capacity'>Guests</label>
      <select name='capacity' id='capacity' value={capacity} className="form-control" onChange={handleChange}>
        {people}
      </select>
    </div>
    { /* guests type */ }
    { /* room price */ }
    <div className='form-group'>
      <label htmlFor='price'>
        room price Rs{price}
      </label>
      <input type="range" name="price" min={minPrice} max={maxPrice} id="price" value={price} onChange={handleChange} cla:
    </div>
    { /* end room price */ }
    { /* size */ }
    <div className='form-group'>
      <label htmlFor='size'>room size</label>
      <div className='size-inputs'>
        <input type="number" name="minSize" id='size' value={minSize} onChange={handleChange} className="size-input" />
        <input type="number" name="maxSize" id='size' value={maxSize} onChange={handleChange} className="size-input" />
      </div>
    </div>
    { /* end size */ }
    { /* extras */ }
    <div className='form-group'>
      <div className='single-extra' >
```

The above code is used to design the filter options

5.4 MODULE 4 FUNCTIONALITY

HOTEL SERVICE

```
import React, { Component } from 'react'
import Title from './Title'
import { FaCocktail, FaHiking, FaBeer, FaShuttleVan } from "react-icons/fa";
export default class Services extends Component {
  state={
    services :[
      {
        icon:<FaCocktail/>,
        title:"free cocktails",
        info:' Indulge in the ultimate luxury at our hotel, where you can

      },
      {
        icon:<FaHiking/>,
        title:"Endless Hiking",
        info:' Nestled amidst breathtaking natural beauty, our hotel offer

      },
      {
        icon:<FaShuttleVan/>,
        title:"Free shuttle",
        info:' Experience the convenience and ease of exploring the city v

      },
      {
        icon:<FaBeer/>,
        title:"Strongest Beer",
        info:' Unwind and savor the robust flavors of our hotels selectionior

      }
    ]
  }
  render() {
    return (
      <section className="services">
        <Title title="services"/>
        <div className='services-center'>
          {this.state.services.map((item,index)=>{
            return<article key={index} className="service">
              <span>{item.icon}</span>
              <h6>{item.title}</h6>
              <p>{item.info}</p>
            }
          )}
        </div>
      </section>
    )
  }
}
```

The above code is used to design the Hotel Services

5.5 MODULE 5 FUNCTIONALITY

BOOKING

```
const Booking = () => {  
  const [userData, setUserData] = useState(  
    {  
      customerName: ""  
      , email: ""  
      , PhoneNo: ""  
      , StartDate: ""  
      , EndDate: ""  
      , People: ""  
    }  
  );  
  let name, value;  
  const postuserData = (event) => {  
    name = event.target.name;  
    value = event.target.value;  
    setUserData({ ...userData, [name]: value });  
  };  
  
  const submitData = async (event) => {  
    event.preventDefault();  
    const {customerName  
      , email  
      , PhoneNo  
      , StartDate  
      , EndDate  
      , People} = userData;  
  };
```

The above code is used to design the Booking page.

CHAPTER 6

RESULTS

6.1 HOME PAGE



Fig no.6.1.1

6.2 HOTEL SERVICES

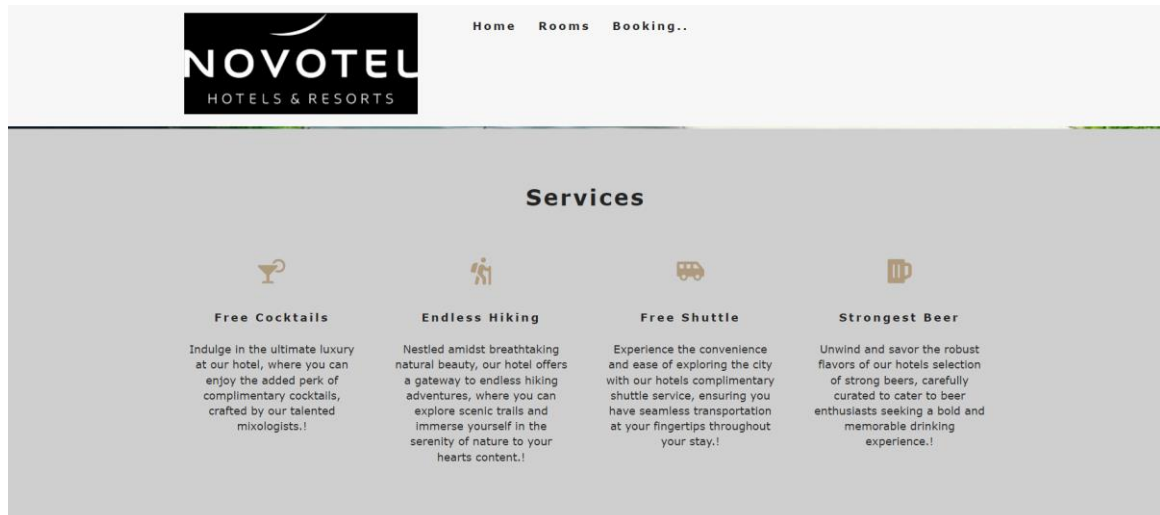



Fig no:6.2.1

6.3 ROOMS AVAILABLE




HomeRoomsBooking..

Search Rooms

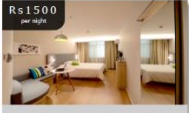
Room Type
all

Guests
1

room price Rs7000



Room Size
01000

☐ Parking Available
☐ No Parking




Rs1500
per night

Single Economy




Rs1750
per night

Single Basic




Rs2000
per night

Single Standard

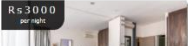


Rs2500
per night


Single Deluxe




Rs2750
per night



Rs3000
per night



Rs3500
per night




Rs4000
per night




Fig no:6.3.1

18

6.4 ROOM FEATURES



[Home](#) [Rooms](#) [Booking..](#)



Details

A luxury hotel is a sanctuary of opulence and refinement, offering meticulously designed rooms, world-class dining, indulgent spas, and exceptional service. With attention to detail, prime locations, and a focus on exceeding expectations, it provides an unforgettable experience for those seeking the pinnacle of hospitality and luxury.

Info

Price : Rs4500

Size : 500 SQFT

Max Capacity : 3 People

No Pets Allowed

Extras

Plush pillows and breathable bed linens

Complimentary refreshments

Comfortable beds

Soft, oversized bath towels

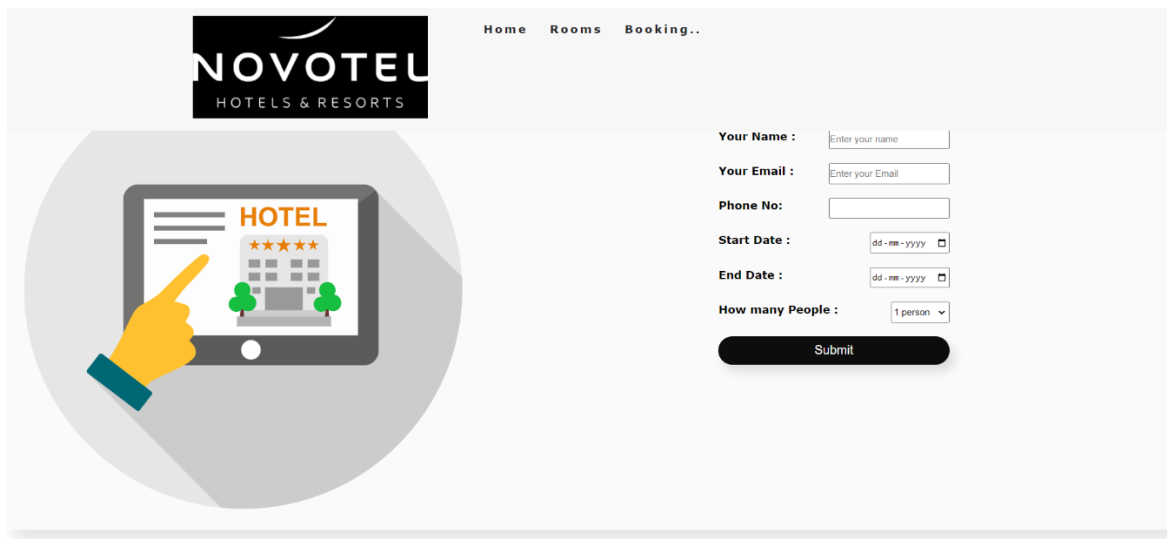
Adequate safety/security

Full-sized, pH-balanced toiletries

Internet

Fig no 6.4.1

6.5 BOOKING



The image shows a web form for booking a hotel room. At the top left is the Novotel Hotels & Resorts logo. To its right are navigation links: Home, Rooms, and Booking.. Below the logo is a large circular graphic containing an illustration of a hand pointing at a tablet screen that displays a hotel building and the word 'HOTEL'. To the right of this graphic is a form with the following fields: 'Your Name' with a placeholder 'Enter your name', 'Your Email' with a placeholder 'Enter your Email', 'Phone No:' with an empty text box, 'Start Date' with a date picker showing 'dd - mm - yyyy', 'End Date' with a date picker showing 'dd - mm - yyyy', and 'How many People' with a dropdown menu showing '1 person'. A black 'Submit' button is located at the bottom of the form.

NOVOTEL
HOTELS & RESORTS

Home Rooms Booking..

Your Name :

Your Email :

Phone No:

Start Date :

End Date :

How many People :

Submit

Fig no 6.5.1

CHAPTER 7

CONCLUSION

In conclusion, the development of a hotel booking website using HTML, CSS, Node.js, and React.js has proven to be a highly effective and efficient solution for providing an intuitive and user-friendly platform for booking accommodations.

Through the implementation of HTML and CSS, the website's structure and design have been crafted to deliver a visually appealing and seamless user experience. The use of responsive design techniques ensures that the website adapts to different devices, enabling users to make bookings from various platforms such as desktops, tablets, and mobile phones.

Overall, the combination of HTML, CSS, Node.js, and React.js has resulted in a feature-rich hotel booking website that offers a seamless user experience, efficient server-side operations, and a visually appealing design. The use of these technologies has greatly enhanced the functionality, performance, and user satisfaction of the website, making it a compelling choice for both hotel owners and guests seeking a reliable and user-friendly platform for booking accommodations.

REFERENCES

[1]<https://www.w3schools.com/html/>

[2]<https://www.w3.org/Style/CSS/Overview.en.html>