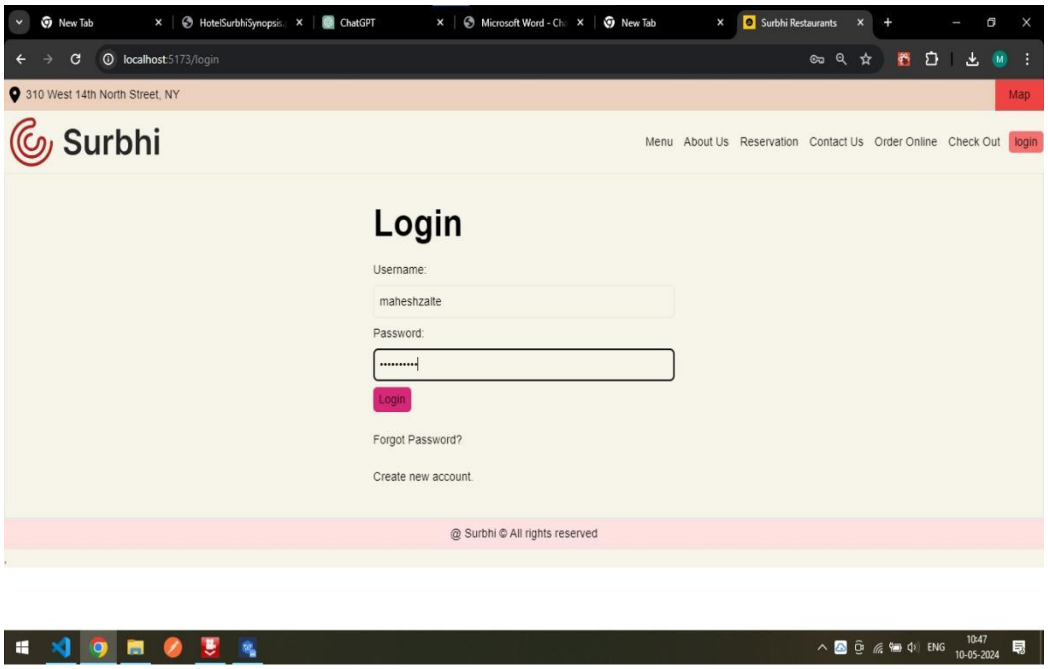
CHAPTER 4
CODING

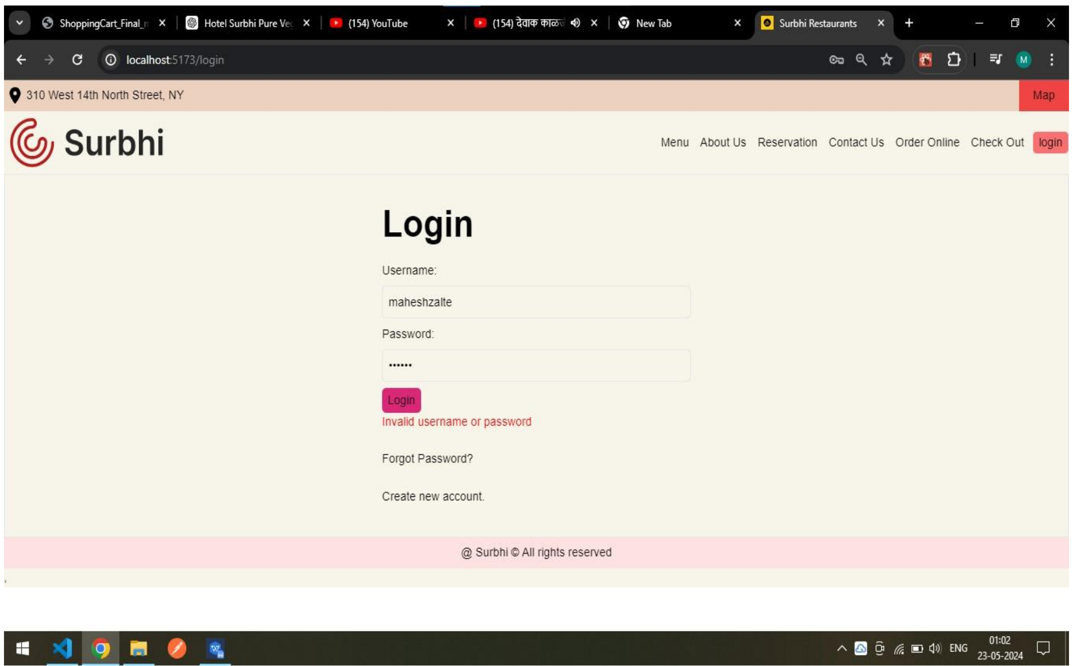
4.1 User Interface Design (Screens etc.)

4.1.1 Login



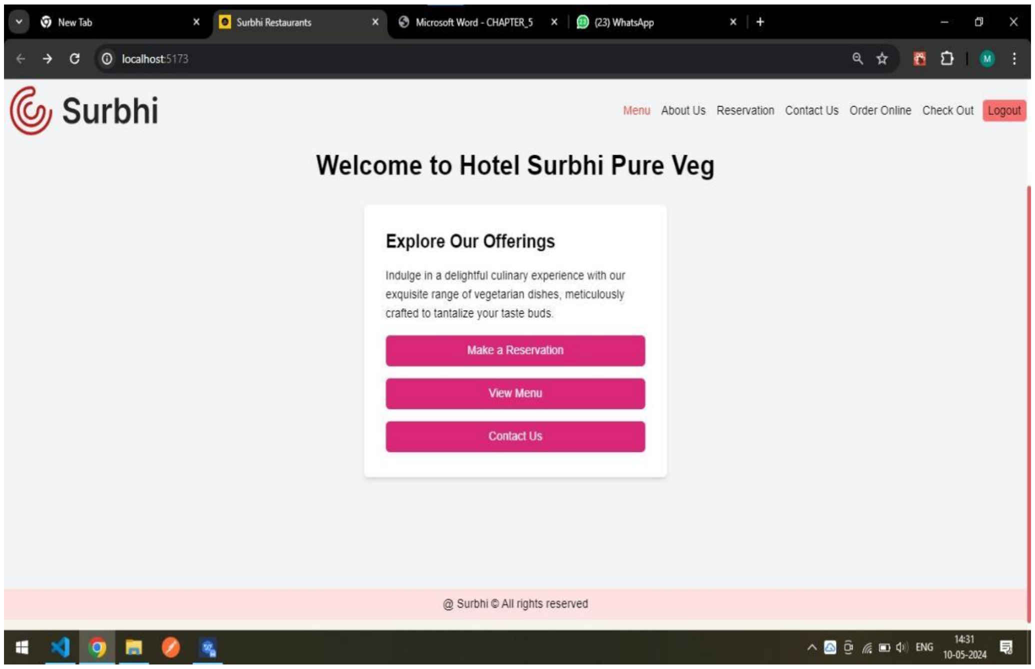
This is a Login Page where user is able to Login using their Login traditional.

Validation:



Invalid username or password

4.1.2 Home Page



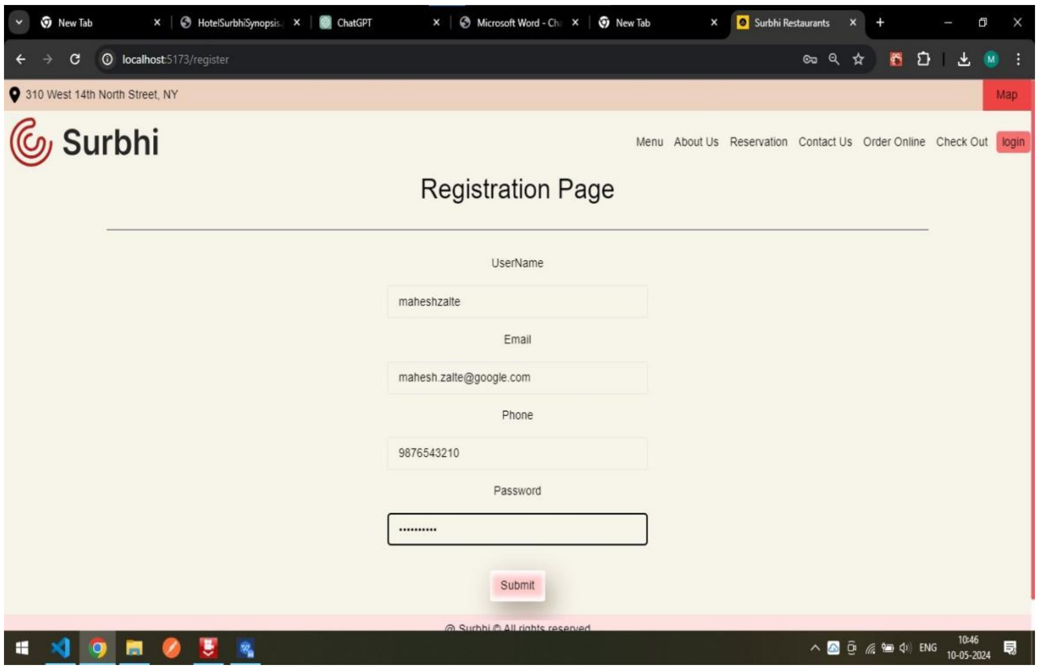
Welcome to Hotel Surbhi Pure Veg

Explore Our Offerings

Indulge in a delightful culinary experience with our exquisite range of vegetarian dishes, meticulously crafted to tantalize your taste buds.

This is a home page that can be shown to the user at initial stage.

4.1.3 Registration Page


localhost:5173/register
310 West 14th North Street, NY
Map
Surbhi
Menu About Us Reservation Contact Us Order Online Check Out login
Submit
© Surbhi © All rights reserved

Registration Page

UserName

maheshzalte

Email

mahesh.zalte@google.com

Phone

9876543210

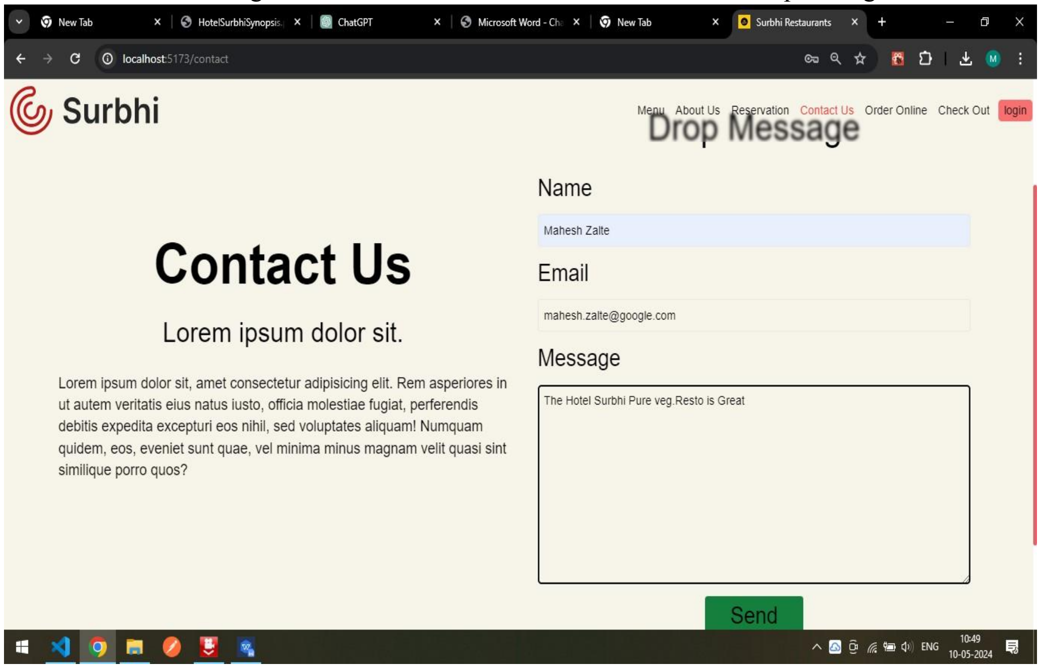
Password

••••••••••

This is a Registration page where new user is able to register, creates an new account.

4.1.4 Contact Us Page

User is able to give feedback and able to contact to Hotel Surbhi pure veg.


localhost:5173/contact
Surbhi
Menu About Us Reservation Contact Us Order Online Check Out login
Send

Drop Message

Contact Us

Lorem ipsum dolor sit.

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Rem asperiores in ut autem veritatis eius natus iusto, officia molestiae fugiat, perferendis debitis expedita excepturi eos nihil, sed voluptates aliquam! Numquam quidem, eos, eveniet sunt quae, vel minima minus magnam velit quasi sint similique porro quos?

Name

Mahesh Zalte

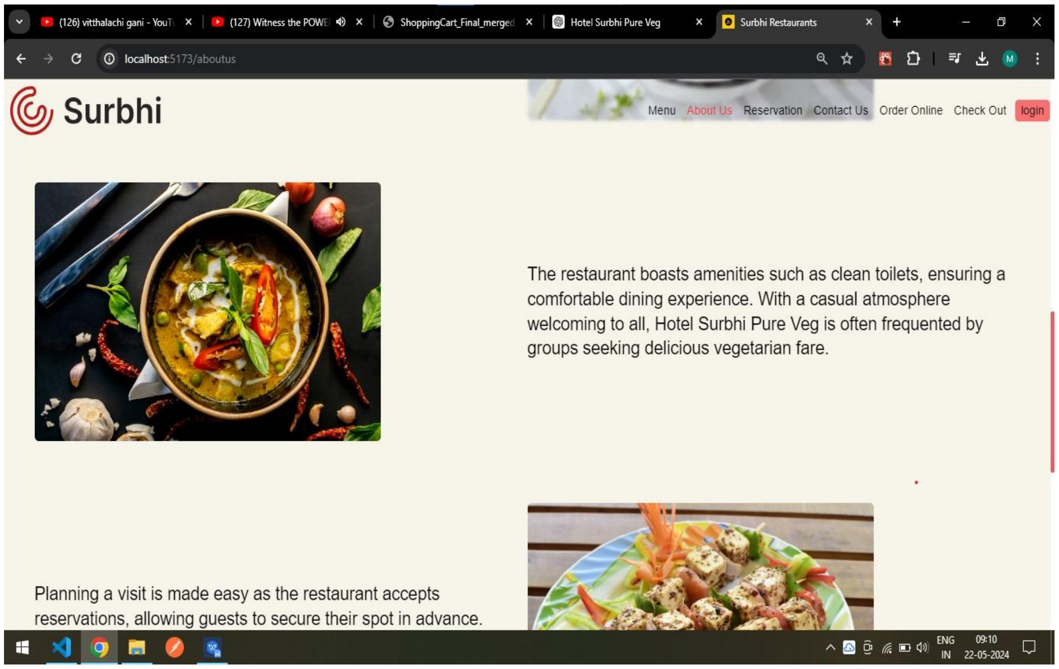
Email

mahesh.zalte@google.com

Message

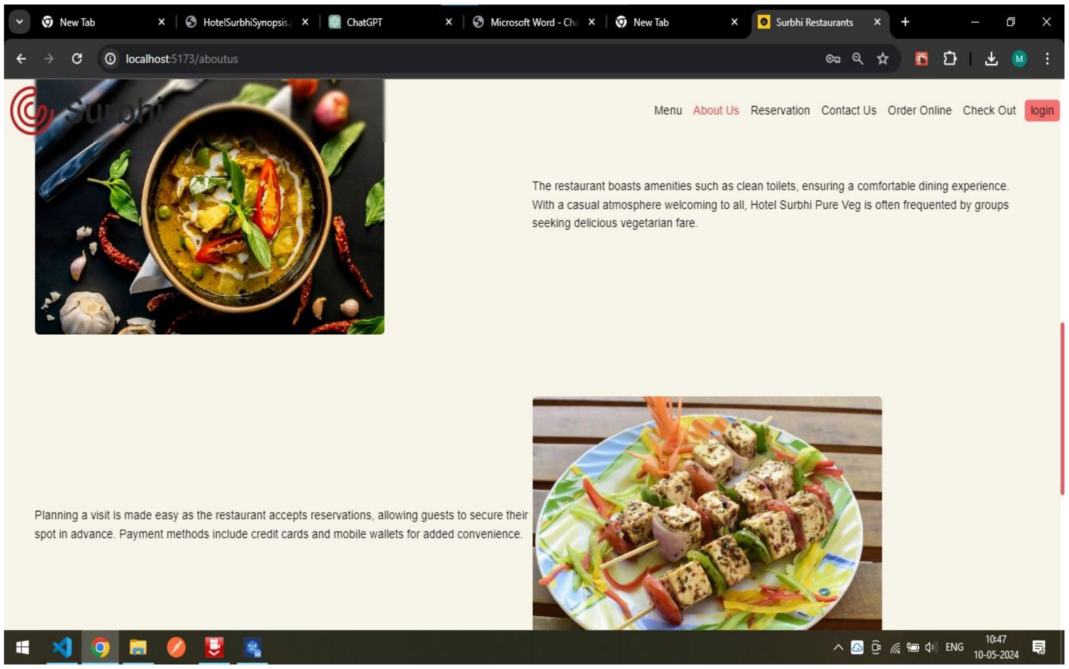
The Hotel Surbhi Pure veg.Resto is Great

4.1.5 About Us



The restaurant boasts amenities such as clean toilets, ensuring a comfortable dining experience. With a casual atmosphere welcoming to all, Hotel Surbhi Pure Veg is often frequented by groups seeking delicious vegetarian fare.

Planning a visit is made easy as the restaurant accepts reservations, allowing guests to secure their spot in advance.



The restaurant boasts amenities such as clean toilets, ensuring a comfortable dining experience. With a casual atmosphere welcoming to all, Hotel Surbhi Pure Veg is often frequented by groups seeking delicious vegetarian fare.

Planning a visit is made easy as the restaurant accepts reservations, allowing guests to secure their spot in advance. Payment methods include credit cards and mobile wallets for added convenience.

This page containing all the information of Hotel Surbhi Pure Veg.

4.2 Codding:

4.2.1 Registration Page:

```
import React, { useState } from 'react'
import { useNavigate } from 'react-router-dom';
import { useAuth } from '../Store/auth';

function Registration() {
    const [username, setUsername] = useState('');
    const [email, setEmail] = useState('');
    const [password, setPassword] = useState('');
    const [phone, setPhone] = useState('');
    const [error, setError] = useState('');

    // To Redirect to the login Page for login.
    const navigate = useNavigate();
    const { StoreToken } = useAuth()



    const handleSubmit = async (e) => {
        e.preventDefault();

        try {
            const response = await fetch('http://localhost:3000/register', {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json',
                },
                body: JSON.stringify({ username, phone, email, password }),
            });

            if (response.ok) {
                const data = await response.json();
                // alert('Registration success');
                if (data.msg !== 'User Exist') {
```

```
                console.log('Registration successful:', data);
                StoreToken(data.token)
                navigate('/Home')
            }
            else console.log({ msg: "Exist" });
            // Here you can handle successful login, e.g., redirect to dashboard
        }


    } catch (error) {
        console.error('Registration error:', error);
        setError('Invalid username or password');
        throw new Error('Invalid username or password');
    }
};


return (
    <div className='flex flex-col w-full gap-8 justify-center items-center '>
        <div >
            <h1 className='text-4xl'>Registration Page</h1>
        </div>
        <div className='w-[80%]'>
            <hr className='border-gray-600' />
        </div>
        <div className='flex flex-col w-full'>
            <form method='POST' onSubmit={handleSubmit} className='flex
                flex-col gap-4 w-full items-center justify-center'>
                <label htmlFor="username">UserName</label>
                <input className='border px-4 py-2 rounded m-0 w-96'
                value={username}
                    onChange={(e) => setUsername(e.target.value)}
                placeholder='Username' type="text" id='username'
                name='username' >
                <label htmlFor="email">Email</label>
```

```
                    <input className='border px-4 py-2 rounded m-0 w-96'
                        placeholder='Email' value={email}
                      onChange={(e) => setEmail(e.target.value)} type="email"
                        id='email' name='email' />
                    <label htmlFor="phone">Phone</label>
                    <input className='border px-4 py-2 rounded m-0 w-96'
                        placeholder='Phone' value={phone}
                      onChange={(e) => setPhone(e.target.value)} type="Number"
                        name='phone' id='phone' />

                    <label htmlFor="password">Password</label>
                    <input className='border px-4 py-2 rounded m-0 w-96'
                        value={password}
                      onChange={(e) => setPassword(e.target.value)}
                    placeholder='Password' type="password" name='password'
                    id='password' />

                    <button type="submit" className='box-shadow px-4 py-2 m-4
                    bg-red-200 rounded'>Submit</button>
                  </form>
                </div>
              </div>
  )
}

export default Registration;
```

4.2.2 Login Page:

```
import React, { useState } from 'react';
import { Link, useNavigate } from 'react-router-dom';
import { useAuth } from '../Store/auth';

const Login = () => {
  const [username, setUsername] = useState('');
  const [password, setPassword] = useState('');
  const [error, setError] = useState('');

  const navigate = useNavigate();
  const { StoreToken } = useAuth()

  const handleLogin = async (e) => {
    e.preventDefault();

    try {
      const response = await fetch(`http://localhost:3000/login`, {
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify({ username, password }),
      });

      if (response.ok) {
        const data = await response.json();

        // Set Token in to local Storage.
        StoreToken(data.token)
        console.log('Login successful:', data.token);

        // Here you can handle successful login, e.g., redirect to dashboard
        navigate('/')
      }
    } catch (error) {
        console.error('Login error:', error.message);
        setError('Invalid username or password');
    }
};

const handleForgotPassword = () => {
    console.log('Forgot password clicked');
    // Here you can implement the logic for forgot password functionality
};

return (
    <div className='flex flex-col border items-center justify-center bg-transparent py-
        10'>
        <div className='grid gap-6 sm:w-[440px]'>
            <h2 className=' text-5xl font-bold bg-transparent'>Login</h2>
            <form onSubmit={handleLogin} className='bg-transparent'>
                <div className='flex flex-col bg-transparent'>
                    <label htmlFor="username" className='bg-transparent'>Username:</label>
                    <input
                        className='p-2 border my-2 rounded-md bg-transparent'
                        type="text"
                        id="username"
                        placeholder='Username'
                        value={username}
                        onChange={(e) => setUsername(e.target.value)}
                        required
                    />
                </div>
                <div className='flex flex-col bg-transparent'>
                    <label htmlFor="password" className='bg-transparent'>Password:</label>
                    <input
                        className='p-2 border my-2 rounded-md bg-transparent'
                        type="password"
            id="password"
            placeholder='Password'
            value={password}
            onChange={(e) => setPassword(e.target.value)}
            required
          />
        </div>
        <div className='bg-transparent'>
          <button type="submit" className=' bg-pink-600 px-2 py-1 rounded-
            md'>Login</button>
        </div>
        {error && <div style={{ color: 'red' }}>{error}</div>}
      </form>
      <div className='bg-transparent'>
        <button onClick={handleForgotPassword}>Forgot Password?</button>
      </div>
      <div className='bg-transparent'>
        <Link to='/register'>Create new account.</Link>
      </div>
    </div>
  </div>
 );
};

export default Login;
```

CHAPTER 5 TESTING

5.1 Testing Strategy:

TESTING:

After finishing the development of any computer-based system the next complicated time-consuming work is system testing. During the time of testing, only the development company can know, how far the user requirements have been met, and so on. Following are some of the testing methods applied to this effective project:

SPECIFICATION TESTING:

It defines system requirements for login, profile management, subject management, session management, staff management, notification, attendance viewing, student leave, and feedback functionalities. It evaluates system performance against these predefined specifications to ensure compliance.

MODULE-LEVEL TESTING:

It test each module individually to identify and rectify errors without affecting other modules. Test login functionality, profile management, subject management, session management, staff management, notification system, attendance viewing, leave management, and feedback system separately.

UNIT TESTING:

It verifies each unit of source code for functionality and usability. Test individual functions within modules to ensure they perform as expected. Test boundary cases and data integrity within local data structures. Unit testing focuses on verifying the effort on the smallest unit of the software module.

INTEGRATION TESTING:

It combines unit-tested modules to form larger aggregates. Test interfaces between modules to uncover integration errors. Validate data flow between modules. In the

Integration testing (Modules) the data can be tested across an interface. Conducting tests to uncover errors associated with interring while integration testing is a technique for constructing a program structure.

VALIDATION TESTING:

It validates each module's functionality against user requirements. It performs black-box testing to verify system behaviour. Ensure the system functions as expected in various scenarios. In this, the majority of the validation is done during the data entry operation where there is a maximum possibility of entering wrong data other validation will be performed in all processes where correct details and data should be entered to get the required results.

RECOVERY TESTING:

It test the system's ability to recover from hardware failures and crashes. Verify that recovery mechanisms function as intended and provide confirmation of recovery. It detects where the failure of the software is occurred and provides many ways to recover the failure module and it sends the confirmation about the recovery. In the process of recovery testing do not be confused with reliability testing because it tries to discover the point at which failure occurs.

SECURITY TESTING:

Verify authentication, integrity, availability, confidentiality, non-repudiation, and authorization mechanisms. Test for vulnerabilities offensive penetrations. Ensure data protection mechanisms are in place and effective. It is a procedure to verify that the system maintains and protects data functionality as intended.

PERFORMANCE TESTING:

The usefulness of computer applications, networks, and devices can be decided by performance testing. In the context of an integrated testing system, it checks the performance of the software at runtime. Performance testing can be also used for calculating the characteristics like reliability, scalability, and interoperability. It is generally used together with stress testing.

BLACKBOX TESTING:

It focuses on the functional requirement of the software. It enables to derive of input states that will completely implement every one useful requirement for a program. Black box testing is used for finding errors in the categories shown below: It detects the wrong or misplaced functions. Poorly developed interfaces. It is also used for finding errors in data structures or external database access and performance errors.

5.2 Unit Test Plan

Unit Test Plan for Attendance Management System:

Test Plan Overview:

The purpose of this test plan is to ensure the functionality and accuracy of the Hotel Surbhi Pure Veg Restaurant Management System. The focus will be on testing individual units or components of the system, including login, profile management, menu management, reservation management, staff management, customer notification, reservation viewing, online ordering, payment processing, and customer feedback functionalit. The tests will cover both positive and negative scenarios to ensure the system handles various situations appropriately.

Reservation:

Test Case 1: Verify successful reservation with valid details.

Expected Outcome: Reservation confirmation message is displayed.

Test Case 2: Verify error message display for invalid reservation details (e.g., invalid date or time).

Expected Outcome: Appropriate error message is displayed.

Test Case 3: Verify error message display for blank reservation fields.

Expected Outcome: Error message prompts to fill all required fields.

Test Case 4: Verify reservation system handles duplicate reservations gracefully.

Expected Outcome: Error message or prompt to update the existing reservation.

Update Profile:

Test Case 1: Verify successful update of user profile information.

Expected Outcome: Profile updated confirmation message is displayed.

Test Case 2: Verify successful change of password.

Expected Outcome: Password change confirmation message is displayed.

Test Case 3: Verify error message display for incomplete profile update.

Expected Outcome: Error message prompts to fill all required fields.

Test Case 4: Verify error message display for incorrect current password during password change.

Expected Outcome: Error message indicating incorrect current password.

Add Menu Item:

Test Case 1: Verify successful addition of a new menu item.

Expected Outcome: New menu item is listed and available for customers.

Test Case 2: Verify error message display for adding a duplicate menu item.

Expected Outcome: Error message indicating item already exists.

Test Case 3: Verify error message display for incomplete menu item details during addition.

Expected Outcome: Error message prompts to fill all required fields.

Manage Menu Item:

Test Case 1: Verify successful update of menu item details.

Expected Outcome: Menu item details are updated and reflected in the menu.

Test Case 2: Verify successful deletion of a menu item.

Expected Outcome: Menu item is removed from the menu list.

Add Staff:

Test Case 1: Verify successful addition of a new staff member.

Expected Outcome: New staff member is listed and available for scheduling.

Test Case 2: Verify error message display for adding a staff member with incomplete details.

Expected Outcome: Error message prompts to fill all required fields.

Test Case 3: Verify error message display for adding a duplicate staff member.

Expected Outcome: Error message indicating staff member already exists.

Manage Staff:

Test Case 1: Verify successful update of staff details.

Expected Outcome: Staff details are updated and reflected in the system.

Test Case 2: Verify successful deletion of a staff member.

Expected Outcome: Staff member is removed from the staff list.

Notify Customers:

Test Case 1: Verify successful notification sent to all customers.

Expected Outcome: Notification is sent and confirmation message is displayed.

Test Case 2: Verify successful notification sent to individual customers.

Expected Outcome: Notification is sent and confirmation message is displayed.

View Reservations:

Test Case 1: Verify successful viewing of reservation records for a particular date.

Expected Outcome: Reservation records are displayed.

Test Case 2: Verify appropriate error message display for viewing reservations when no data is available.

Expected Outcome: Error message indicating no reservations found.

Customer Feedback:

Test Case 1: Verify successful submission of feedback by a customer.

Expected Outcome: Feedback submission confirmation message is displayed.

Test Case 2: Verify successful viewing of feedback by staff.

Expected Outcome: Feedback is displayed in the staff portal.

Test Case 3: Verify error message display for submitting feedback with empty content.

Expected Outcome: Error message prompts to fill feedback content.

Online Ordering:

Test Case 1: Verify successful submission of an online order.

Expected Outcome: Order confirmation message is displayed.

Test Case 2: Verify error message display for incomplete order details.

Expected Outcome: Error message prompts to fill all required fields.

Test Case 3: Verify error message display for ordering out-of-stock items.

Expected Outcome: Error message indicating item is out of stock.

Payment Processing:

Test Case 1: Verify successful payment processing for an order.

Expected Outcome: Payment confirmation message is displayed.

Test Case 2: Verify error message display for payment failure due to invalid card details.

Expected Outcome: Error message indicating invalid payment details.

Test Case 3: Verify error message display for payment failure due to insufficient funds.

Expected Outcome: Error message indicating payment failure.

Leave Request (for staff):

Test Case 1: Verify successful submission of leave request by staff.

Expected Outcome: Leave request submission confirmation message is displayed.

Test Case 2: Verify successful approval or rejection of leave request by administrator.

Expected Outcome: Leave request status is updated and notification is sent.

Test Case 3: Verify error message display for submitting leave request with incomplete details.

Expected Outcome: Error message prompts to fill all required fields.

Test Execution:

Each test case should be executed individually and independently. Test data should be prepared specifically for each test case to cover different scenarios. Any test environment setup or prerequisites should be clearly defined and met before executing the tests.

Test Reporting:

Each test case should be reported with a clear description of the test scenario, expected results, and actual results. Any deviations or issues encountered during testing should be documented along with steps to reproduce the problem, if applicable.

Acceptance Plan:

Purpose: The purpose of this acceptance plan is to outline the criteria and steps for validating the functionality of the Hotel Surbhi Pure Veg Restaurant Management System. The system should meet the requirements and expectations of the stakeholders, including efficient reservation management, accurate user management, reliable notification delivery, and comprehensive feedback handling.

Acceptance Criteria: The following acceptance criteria will be used to determine the successful acceptance of the Hotel Surbhi Pure Veg Restaurant Management System:

All specified features and functionalities are implemented and working as intended.

The system accurately handles reservations for various dates and times.

User accounts can be created, managed, and assigned appropriate roles (admin, staff, customer).

Notification delivery is reliable and timely, both for individual customers and group notifications.

Customer feedback submissions are handled correctly and accessible to staff for review.

The system provides accurate reports on reservations, payments, and feedback submissions.

Test Environment: The acceptance testing will be conducted in an environment that replicates the production setup of the Hotel Surbhi Pure Veg Restaurant Management System. This includes the necessary hardware, software, and network configurations.

Test Approach: The acceptance testing will be performed using a combination of manual testing and automated testing tools. The test cases will cover different scenarios to ensure comprehensive coverage of the system's functionality.

Test Scenarios and Test Cases:

Login:

Test Scenario: Successful login with valid credentials.

Test Case: Verify that users can log in with the correct username and password.

Reservation Management:

Test Scenario: Accurate handling of reservations.

Test Case: Verify that reservations are created, updated, and canceled correctly.

User Management:

Test Scenario: Creation and management of user accounts.

Test Case: Verify that admins can create, update, and delete user accounts.

Notification Delivery:

Test Scenario: Reliable delivery of notifications.

Test Case: Verify that notifications are delivered promptly to the intended recipients.

Feedback Handling:

Test Scenario: Submission and review of customer feedback.

Test Case: Verify that customers can submit feedback and staff can review and respond to it.

Payment Processing:

Test Scenario: Accurate handling of payment transactions.

Test Case: Verify that payments are processed correctly and receipts are generated.

Online Ordering:

Test Scenario: Accurate processing of online orders.

Test Case: Verify that customers can place orders online and track their status.

Test Execution and Reporting:

The acceptance tests will be executed according to the defined test scenarios and test cases. The results will be documented, and any deviations from the expected results will be reported. A test report will be generated, including the test execution status and any issues or defects encountered during testing.

Sign-off:

The acceptance plan will be considered complete when all defined test scenarios and test cases have been executed, and the system meets the acceptance criteria. The stakeholders will review the test results and provide their approval and sign-off for the Hotel Surbhi Pure Veg Restaurant Management System.

CHAPTER 6
LIMITATIONS

Limitations:

1. **People's Resistance**

One of the main limitations for Hotel Surbhi Pure Veg's online ordering system is data security and privacy concerns. Despite advanced data encryption security systems, customers are often hesitant to provide their personal and financial details. Moreover, some websites lack the necessary features to authenticate transactions, leading to instances of fraudulent activities. This fear of providing financial information like credit card details can hinder the growth of the online ordering system.

2. **Lack of Privacy**

Customer privacy is a significant concern in online transactions. Users need to provide personal details, such as addresses and telephone numbers, to complete orders. Many websites still lack advanced technology to protect sensitive information adequately. Additionally, some sites may collect consumer data without proper consent, leading to skepticism and distrust among potential customers.

3. **Tax Issues**

Taxation can become complicated when dealing with customers from different geographical locations. Calculating and applying the correct sales tax can be challenging, and inconsistencies may arise. Physical restaurants might also face a loss in business if online transactions are exempted from certain taxes, creating an uneven playing field.

4. **Product Suitability**

Online menus and pictures may not always accurately represent the actual dishes. Customers cannot physically examine or taste the food before ordering, leading to potential dissatisfaction if the delivered product does not meet their expectations. This absence of a 'touch and feel' experience can be a significant drawback.

5. **Cultural Obstacles**

   Serving a diverse customer base online means dealing with various cultural habits, traditions, and preferences. Language barriers and differing dietary restrictions can lead to misunderstandings and issues between the restaurant and its customers.

6. **Customer Hesitancy**

   Despite the growing popularity of online food ordering, some customers still prefer to dine in physical restaurants. The inability to see, smell, or taste the food before ordering online creates a degree of uncertainty and hesitancy among customers.

7. **Technical Limitations**

   Running an online ordering system requires a robust technological infrastructure. Issues such as inadequate domain management, network problems, and software glitches can disrupt the seamless operation of the eCommerce platform, affecting customer experience and satisfaction.

8. **High Technological Costs**

   Significant investment is required to develop and maintain the technical infrastructure for an online ordering system. Continuous upgrades and maintenance are necessary to keep up with evolving technology, which can be costly.

9. **Dependence on Internet Connectivity**

   Online ordering systems rely heavily on stable internet connections. Any disruption in internet service can prevent customers from placing orders and affect the restaurant's ability to manage online requests efficiently.

CHAPTER 7
Proposed Enhancements

Future / Proposed Enhancements

1. **Advanced Authentication Systems**

   Develop more robust authentication methods to ensure secure online transactions without requiring customers to expose sensitive financial details. This could involve biometric authentication or blockchain-based secure payment systems.

2. **Privacy-Centric Platforms**

   Create an online ordering platform that prioritizes user privacy by implementing end-to-end encryption for communication between customers and the restaurant, and by strictly regulating the collection and usage of customer data.

3. **Automated Tax Computation**

   Introduce automated tax calculation systems that can accurately compute sales tax based on the customer's location and the restaurant's tax obligations, reducing the burden on the restaurant and ensuring compliance with tax laws.

4. **Augmented Reality Menu Viewing**

   Implement augmented reality (AR) technology to allow customers to virtually examine dishes before ordering, providing a more immersive and confident dining experience similar to viewing the menu in the restaurant.

5. **Enhanced Menu Information**

   Improve menu descriptions and specifications with detailed multimedia content, user reviews, and interactive features to bridge the gap between physical and online ordering experiences, helping customers make informed decisions.

6. Cross-Cultural Adaptations

Develop an online ordering platform that is culturally sensitive and adaptable to diverse customer preferences and languages, providing localized experiences and addressing cultural barriers to enhance trust and satisfaction.

7. Next-Generation Technology Integration

Leverage emerging technologies such as artificial intelligence (AI), machine learning (ML), and Internet of Things (IoT) to overcome technical limitations and enhance the performance, reliability, and scalability of the online ordering platform.

8. Cost-Effective Technology Solutions

Explore cost-effective alternatives such as cloud-based infrastructure, open-source software, and modular architectures to reduce the initial investment and ongoing maintenance costs of running an online ordering system, making it more accessible to small and medium-sized restaurants.

9. Improved Delivery Systems

Develop an efficient and reliable delivery system to ensure timely and accurate delivery of orders, including real-time tracking of orders and optimizing delivery routes to enhance customer satisfaction.

10. Loyalty Programs and Personalized Offers

Implement loyalty programs and personalized offers based on customer preferences and ordering history to increase customer retention and encourage repeat business.

10. Real-Time Customer Support

Introduce real-time customer support through chatbots and live chat features to assist customers with their queries and issues promptly, improving the overall customer experience.

11. Sustainable Practices

Integrate sustainable practices into the online ordering system, such as eco-friendly packaging options and promoting plant-based menu items, to appeal to environmentally conscious customers.

CHAPTER 8
Conclusion

Conclusion

The main objective of this project was to develop an online ordering system for Hotel Surbhi Pure Veg, enabling the restaurant owner to manage products, customers, and orders efficiently while allowing customers to place orders and make payments online.

The system was developed with the aforementioned features. One of the biggest challenges faced during the development of this software project was integrating the various components of the MERN stack (MongoDB, Express.js, React.js, Node.js). A considerable amount of time and effort was invested in learning and implementing these technologies effectively. Another challenge was integrating secure payment methods into the system. Ensuring the security and privacy of customer financial details required extensive research and implementation of robust security measures.

Despite these challenges, a great deal of experience has been gained throughout the development process. Although all the requirements set out for the online ordering system have been met, there are still areas for improvement. Developing a mobile version of the application can provide users with better accessibility. Additionally, implementing multiple online payment methods, such as credit/debit cards and bank transfers, can enhance user convenience.

Throughout the project lifecycle, we have encountered and overcome numerous challenges, from technical limitations to user interface design issues. Yet, with each hurdle, we have emerged stronger and more resilient, armed with insights that have fueled continuous improvement and refinement.

Moreover, we recognize the importance of adaptability and scalability in the ever-evolving landscape of online services. Our platform is designed not just to meet current demands but to anticipate future needs, with a flexible architecture and robust infrastructure that can accommodate growth and expansion seamlessly.

As we reflect on the journey of building this online ordering system, we are filled with a sense of pride and accomplishment. Yet, we also acknowledge that our work is far from done. Continuous enhancements and updates will be necessary to keep up with technological advancements and customer expectations. We are committed to ongoing improvement and innovation to ensure Hotel Surbhi Pure Veg remains a leader in providing exceptional online dining experiences.

CHAPTER 9
Bibliography

References:

/1/ Miva 2011. The History of Ecommerce: How Did It All Begin? Accessed 15.07.2017.
http://www.miva.com/blog/the-history-of-ecommerce-how-did-it-all-begin/

/2/Arline, K. 2015. What is E-commerce?
Accessed 12.07.2017.
http://www.businessnewsdaily.com/4872-what-is-e-commerce.htm

/3/Tutorialspoint 2017. Java Tutorial. Accessed 17.07.2017.
https://www.tutorialspoint.com/java/index.html

/4/ Stackoverflow 2017. What is the difference between JDK and JRE? Accessed 17.07.2017
https://stackoverflow.com/questions/1906445/

/5/ Evans, I. 2012. Differences between Java EE and Java SE - Your First Cup: An
Introduction to the Java EE Platform. Accessed 17.07.2017.
http://docs.oracle.com/javaee/6/firstcup/doc/gkhoy.html

/6/ Tutorialspoint 2017. HTML Tutorial. Accessed 17.07.2017
https://www.tutorialspoint.com/html/

/7/ W3Schools 2017. CSS Tutorial. Accessed 17.07.2017
https://www.w3schools.com/css/

8/ Oracle. What is a Servlet – The Java EE 5 Tutorial. Accessed 18.07.2017.
http://docs.oracle.com/javaee/5/tutorial/doc/bnafe.html

/9/ Oracle. JavaServer Pages Technology. Accessed 19.07.2017.
http://www.oracle.com/technetwork/java/javaee/jsp/index.html

/10/ Tutorialspoint 2017. Eclipse Tutorial. Accessed 21.07.2017
https://www.tutorialspoint.com/eclipse/

/11/ The Apache Software Foundation 2017. Apache Tomcat. Accessed 22.07.2017.
https://tomcat.apache.org/

Video References:

- https://www.researchgate.net/publication/
- www.youtube.com/watch?v=pKMqJAc6wYw
- www.youtube.com/watch?v=kh4W7z2FoXs&t=971

Websites:

- Google:
  www.google.com
- Javacodegeeks:
  https://www.javacodegeeks.com/
- Stack Overflow:
  https://stackoverflow.com/
- GitHub:
  https://github.com/
- Tutorials Point:
  https://www.tutorialspoint.com/index.html

CHAPTER 10
USER MANUAL

User Manual: Surbhi Pure Veg Restaurant

Table of Contents

1. Introduction

Welcome to the Surbhi Pure Veg Restaurant user manual. This guide will help you navigate and use the online ordering system efficiently.

2. System Requirements

A device with internet access (PC, tablet, smartphone).

A modern web browser (Chrome, Firefox, Safari, Edge).

Active internet connection.

3. Installation Guide

No installation is required. The system is web-based and can be accessed via the internet.

4. Configuration

Ensure your browser is up-to-date and cookies are enabled for optimal performance.

5. Navigation Guide

Home Page

- Access the restaurant's main features and latest offers.
- View daily specials and new menu items.

Menu Page

- Browse through different categories of vegetarian dishes.
- View item details, including ingredients and prices.

Admin Dashboard

- Access restricted to administrators.
- Manage menu items, inventory, and orders.

Cart Page

- Review selected items before checkout.
- Update item quantities or remove items.

Profile Page

- View and update personal information.
- Check order history and track current orders.

6. User Operations

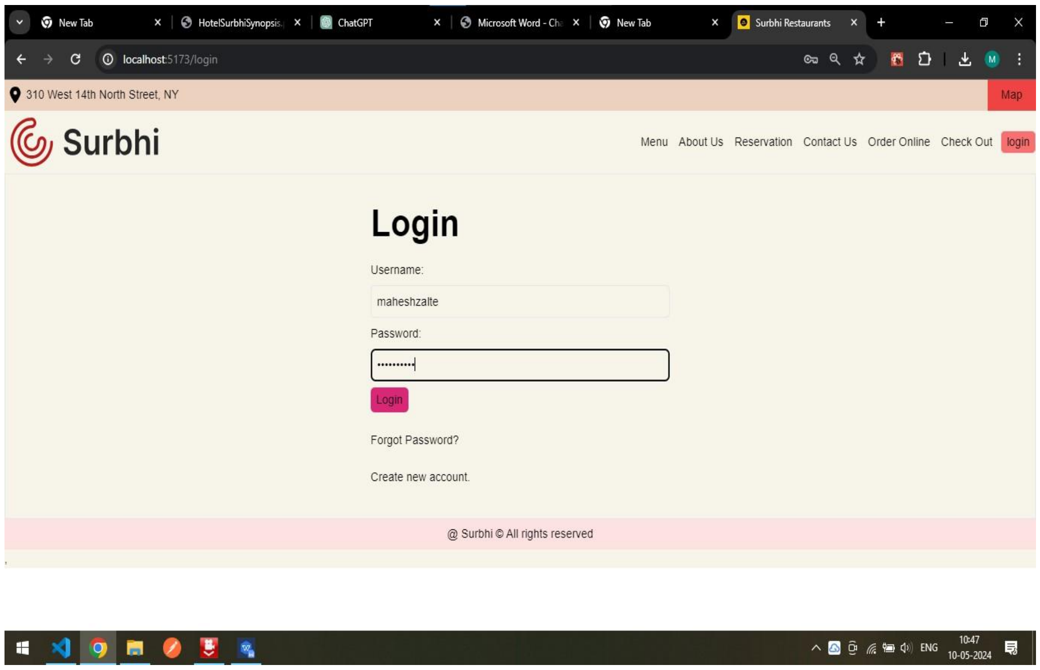
User Registration

- Click on the "Register" button on the home page.

- Fill in the required details and submit the form.
- Verify your email to complete the registration process.

User Login

- Click on the "Login" button.
- Enter your username and password to access your account.



Login

Username:

maheshzalte

Password:

Login

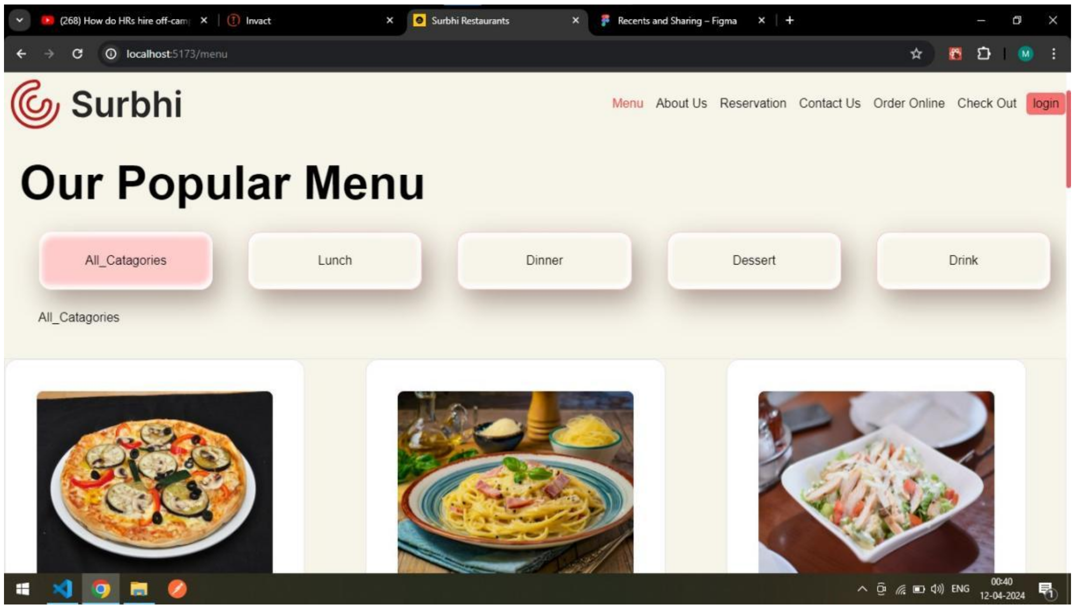
Forgot Password?

Create new account.

@ Surbhi © All rights reserved

Browsing Menu

- Navigate to the "Menu" page.
- Browse items by category or use the search bar to find specific dishes.



Adding Items to Cart

- Select the desired item and click "Add to Cart".
- Adjust the quantity as needed.

Checkout Process

- Go to the "Cart" page.
- Review your order and proceed to checkout.
- Choose your payment method and enter necessary details.
- Confirm your order.
- Order History

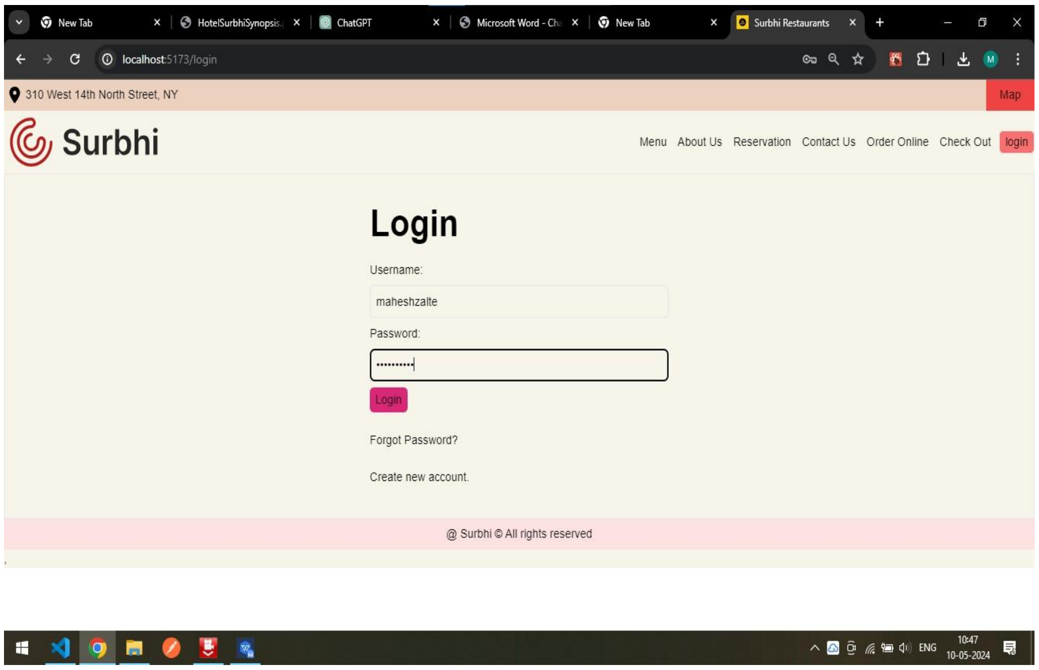
Go to the "Profile" page.

- View past orders and track the status of current orders.

7. Admin Operations

Admin Login

- Click on the "Admin Login" button.
- Enter admin credentials to access the dashboard.


localhost:5173/login
310 West 14th North Street, NY
Map
Surbhi
Menu About Us Reservation Contact Us Order Online Check Out login

Login

Username: maheshzalte

Password: ••••••••••

Login

Forgot Password?

Create new account.

@ Surbhi © All rights reserved

Managing Menu Items

- Add, update, or delete menu items.
- Ensure all items have accurate descriptions and prices.

Managing Inventory/Stock

- Update stock levels to reflect available quantities.
- Set alerts for low stock items.

Viewing Orders

- Access the "Orders" section to view all incoming orders.
- Filter orders by status (pending, processed, completed).

Processing Orders

- Update order status once it is being prepared or shipped.
- Ensure timely delivery and accurate order fulfillment.

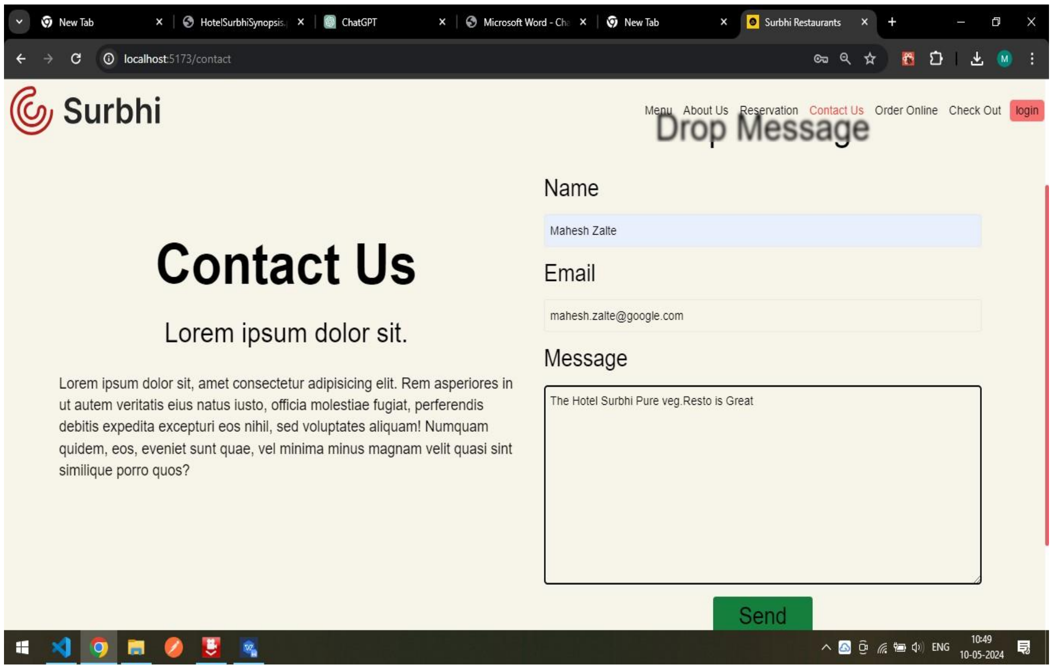
8. Troubleshooting

Login Issues: Ensure your credentials are correct. Reset your password if necessary.

Cart Problems: Check if your browser allows cookies. Clear cache if items are not updating.

Order Processing: Contact customer support if an order status does not update.

Technical Support: For any technical issues, contact our support team via the provided contact details on the website.



Contact Us

Lorem ipsum dolor sit.

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Rem asperiores in ut autem veritatis eius natus iusto, officia molestiae fugiat, perferendis debitis expedita excepturi eos nihil, sed voluptates aliquam! Numquam quidem, eos, eveniet sunt quae, vel minima minus magnam velit quasi sint similique porro quos?

Name: Mahesh Zalte

Email: mahesh.zalte@google.com

Message: The Hotel Surbhi Pure veg Resto is Great

Contact Us