**ONLINE FOOD ORDER SYSTEM**

**School Of Coding - Certified Software Test Engineer**

Submitted By

Chaitali Yuvraj Wadghule

Batch:2022-7669

Enrollment Number: EBEON0722618054

Under the Guidance of Trainer

Mohana Priya



**Online food ordering system process**



**INDEX**

|  |  |
| --- | --- |
| **SR.NO** | **CONTENT** |
| **1** | **INTRODUCTION** |
|  | **1.1 PROBLEM STATEMENT** |
|  | **1.2 MOTIVATION** |
|  | **1.3 SCOPE** |
| **2** | **METHODOLOGY** |
|  | **2.1 SYSTEM ANALYSIS** |
|  | **2.2 HARDWARE REQUIREMENTS** |
|  | **2.3 SOFTWARE REQUIREMENTS** |
| **3** | **SYSTEM DESIGN** |
|  | **3.1 USE CASE** |
|  | **3.2 CLASS** |
|  | **3.3 ACTIVITY** |
|  | **3.4 ENTITY RELATIONSHIP DIAGRAM(ERD)** |
| **4** | **IMPLEMENTATION** |
| **5** | **MODULES** |
| **6** | **SYSTEM TESTING** |
| **7** | **ADVANTAGES** |
| **8** | **LIMITATIONS** |
| **9** | **FUTURE ENHANCEMENT** |
| **10** | **CONCLUSIONS** |

**1.INTRODUCTION**

Restaurants and takeaways benefit hugely from allowing customers to order food online. Web-based ordering creates more sales opportunities while making existing delivery or takeout processes more accessible. It’s also a great way to set up contactless table service at your restaurant.

An online food ordering system is software that lets restaurants, coffee shops, or bars accept orders online. It typically allows customers to choose and pay for food, then alerts the kitchen when an order is made. This happens without contact between staff and customers.

Despite the lack of personal interaction, the software provides online customers with the same features someone visiting the restaurant or ordering on the phone would have. This includes access to the entire menu with full customization and varied payment methods such as card, PayPal, or cash on pickup or delivery.

Online ordering platforms should include both a browser-based system so customers can order from their homes or offices, and an app that lets them buy on the move. This gives customers complete ordering flexibility.

**1.1 PROBLEM STATEMENT**

The online food ordering system sets up a food menu online and customers can easily place the order as per they like. Also, the online customers can easily track their orders. The management maintains customer’s database, and improve food delivery service. This system also provides a feedback system in which user can rate the food items. Also, the proposed system can recommend hotels, food, based on the ratings given by the user, the hotel staff will be informed for the improvements along with the quality. The payment can be made online or cash or pay-on-delivery system. For more secured ordering

separate accounts are maintained for each user by providing them an ID and a password.

**1.2 MOTIVATION**

Appealing online is the best way to make your customers order food online! To put the right foot forth, restaurant owners have to post quality content consistently. Tailored ads and placing digital banners on third-party apps or websites is something the restaurant owners can do or simply hire professionals who know how to do it properly. Note here: the restaurant owners might have to pay a small fee to place these banners or ads promoted on other domains and social media channels like YouTube.

Newsletters are also a promising advertising resource. Sending out a regular newsletter that is visually and contextually engaging has a higher chance to build an audience for you. Newsletters can help your restaurant with:

* Building a community.
* Spreading brand awareness.
* Speak volumes about your objectives.
* And, bring noteworthy improvements to your business in the long run.

**1.3 SCOPE**

Food Ordering app can sale Food product, preferred brands, kitchen needs, essential restaurant supplies and more, through this online, one stop Food store. It provides you with a convenient way to sale from your Food shopping app. You can use this app as one big super market app to sale product of your store. This app make easy for user to buy product from store with easy steps and store can get easy order.

Our scope approvement:-

• Content and image to be added in the application.

• Client side testing devices

• Payment Gateway Credentials

• More elaborate requirements with explaining the setup and current system and complete scope of the solution, For this please conduct a meeting at client.

**2.REQUIREMENT SPECIFICATION**

**2.1 SYSTEM ANALYSIS**

The study indicated that the online food delivery business model is highly demandable, potential and

money efficient. This space is increasing in leaps and bounds because of the size of market. Every human needs to eat multiple times and variety in a day .So it ensures repeat in order and growing business. Due to repeat customers, Profit margins are high. Ordering online is nowadays is fashion or a way of life. Ordering online is much comfortable and less expensive than dine out. The Pros to the

customers are:

1. Ordering online is quick and human error and intervention free

2. Due to technology driven portal there are minimalistic chances of error.

3. The customer has the variety of menus and restaurants to choose from.

4. The consumers have the advantage of promotional deals, loyalty benefits and discounts offered.

5. It gives restaurant like pleasure at their own places.

Different questionnaire was given to the consumers and the restaurant owners / chefs to find out the pros and cons that are experienced by them while using the online food ordering application process.

**2.2 HARDWARE REQUIREMENT**

Processors will continue to get faster, smaller and cheaper, whereas memory will continue to get faster, larger and cheaper. The trend is to have a reasonable memory to a powerful processor.

Processor: Intel core i3 Processor

RAM: 8GB

System type: 64 bit OS

**2.3 SOFTWARE REQUIREMENT**

When an application project is considered the three basic software requirements are the platform in which the project is developed, the front end tool that provides the interaction with the uses and the backend tool that stores the data.

**3.SYSTEM DESIGN**

**3.1 Use Case Diagram**

Admin Customer

**3.2 Class Diagram**

Payment

+id: int

+cardnumber: String

+amount: String

+add()

+update()

Delivery

#id:int

+name: String

+date: String

+address: String

+update()

Admin

#id: int

+name: string

+age: string

+contactnum: string

-username: string

-password: string

+create()

+update()

Customer

+address: string

#cardnum: int

#create()

Order

#id: int

+details: int

+ ordertype: string

+processDebit()

Menu

#id: int

+details: string

+listofdish: string

+add()

+update()

Dish

#id: int

+names: string

+price: string

+add()

+update()

**3.3 ACTIVITY DIAGRAM**

start

Login to the food ordering system

Check user level and permissions

Check permission

Check permission

Check permission

Check permission

Check permission

Manage food item

Manage Customer

Manage Payment

Manage category

Manage order

End

Logout from the system

**3.4 ENTITY RELATIONSHIP DIAGRAM (ERD)**

Orders

order\_ID(PK)

customer\_ID(FK)

date

Deliveries

Delivery\_ID(PK)

Order\_ID(FK)

Customer\_ID(FK)

Product\_ID(FK)

date

Product

Product\_ID(PK)

Category\_ID(FK)

Product\_name

Customer

Customer\_ID(PK)

Name

Contact\_add

address

Categories

Category\_ID (PK)

Category\_name

Category\_type

Sales Reports

Report\_ID (PK)

Customer\_ID(FK)

Order\_ID(FK)

Product\_ID(FK)

Payment\_ID(FK)

date

Seller

seller\_ID (PK)

product\_ID (FK)

name

contac\_add

Payment

Payment\_ID(PK)

Customer\_ID(PK)

date

**4.IMPLEMENTATION**

**4.1 Style.CSS**

/\*CSS Reset\*/

\*{

margin:0px;

padding:0px;

}

/\*Header Styling\*/

#top{

color:rgb(245, 10, 10);

text-align:center;

font-size:46px;

font-family: 'Ubuntu Mono', monospace;

}

#top1{

text-align:center;

color:black;

font-size:21px;

font-family: 'Ubuntu Mono', monospace;

}

/\* Navigation bar styling\*/

/\*Navbar image styling\*/

#navbar img{

display:block;

width: 50px;

height:50px;

margin:auto;

margin-bottom: 3px;

}

/\*Navbar Functionality\*/

#navbar{

display:flex;

flex-direction: column;

background-color: gray;

height: 14vh;

width: 100vw;

font-family: 'Ubuntu Mono', monospace;

margin-top:10px;

border: 2px solid black;

border-radius:15px;

}

/\*Navbar content functionality\*/

#navcontent{

display: flex;

justify-content: center;

}

/\*Navbar content styling\*/

ul li{

list-style: none;

margin: 15px;

border-radius: 20px;

}

ul li a{

padding: 1px;

color:white;

text-decoration: none;

border-radius:10px;

}

ul li a:hover{

background-color: red;

border-radius: 10px;

}

/\* website background image designing \*/

#container1{

position: relative;

display: flex;

flex-direction: column;

align-items: center;

width: 100vw;

height:63vh;

}

/\*Setting the background image using before pseudo selector\*/

#container1::before{

content:'';

background: url("./1.jpg") no-repeat center center/cover;

width: 100vw;

height:59vh;

position: absolute;

top:0px;

left: 0px;

font-family: 'Ubuntu Mono', monospace;

margin-right:45px;

font-weight: bold;

z-index: -1;

opacity: 0.89;

border: 2px solid black;

border-bottom-left-radius: 100px;

}

#row1{

color: black;

font-weight:bold;

font-size: 2rem;

text-align: center;

margin-top: 35px;

}

/\*Button Styling\*/

.btn{

margin-top:15px;

border: 3px solid white;

border-radius:15px;

background-color:yellow;

font-size:20px;

font-weight: bold;

font-family: 'Ubuntu Mono', monospace;

}

.btn:hover{

cursor: pointer;

background-color: red;

}

#container3{

display: flex;

justify-content: space-evenly;

}

#row2{

width:24vw;

height:21vh;

box-shadow: 2px 7px 16px 19px;

margin-top:51px;

margin-bottom: 51px;

display: flex;

justify-content: center;

align-items: flex-end;

position: relative;

margin-right: 38px;

border-radius: 40px;

}

#row2::before{

content: '';

background: url('2.jpeg') no-repeat center center/cover;

position: absolute;

top: 0px;

left:0px;

width: 24vw;

height:21vh;

z-index: -1;

border-radius: 40px;

}

#row3{

position: relative;

width: 24vw;

height:21vh;

display: flex;

justify-content: center;

align-items: flex-end;

box-shadow: 2px 7px 16px 19px;

margin-top:51px;

margin-bottom: 51px;

margin-left: 38px;

border-radius: 40px;

}

#row3::before{

content: '';

background: url('3.jpeg') no-repeat center center/cover;

position: absolute;

top:0px;

left:0px;

width: 24vw;

height:21vh;

z-index: -1;

border-radius: 40px;

}

#top3{

text-align: center;

color:red;

font-family: 'Ubuntu Mono', monospace;

}

#container4{

display: grid;

grid-template-columns: repeat(auto-fit, minmax(350px, 1fr));

margin: 41px;

}

#row4{

width: 29vw;

height: 360px;

border: 2px solid black;

background:url('a1.jpeg') no-repeat center center/cover;

display: flex;

justify-content:center;

align-items:flex-end;

border-radius: 15px;

}

#row5{

width: 29vw;

height: 360px;

border: 2px solid black;

background:url('a2.jpeg') no-repeat center center/cover;

display: flex;

justify-content:center;

align-items:flex-end;

border-radius: 15px;

}

#row6{

width: 30vw;

height: 360px;

border: 2px solid black;

background:url('a3.jpeg') no-repeat center center/cover;

display: flex;

justify-content:center;

align-items:flex-end;

border-radius: 15px;

}

/\*Designing the footer\*/

footer{

text-align: center;

}

/\*Designing of Contact Us\*/

#ContactUs{

width:100vw;

height:100vh;

display:flex;

flex-direction: column;

align-items: center;

background-color: gray;

}

.form-shape input, .form-shape select, .form-shape textarea{

width: 92%;

padding: 0.5rem;

}

/\*Changing the color of placeholder\*/

::placeholder{

color:gray;

}

/\*Making the webpage responsive using media queries\*/

@media only screen and (max-width:1131px){

#row4{

width: 42vw;

}

#row5{

width:45vw;

}

#row6{

margin-top: 20px;

width:90vw;

}

}

@media only screen and (min-width:600px) and (max-width:781px)

{

#row4{

width:84vw;

margin: auto;

margin-left: 22px;

}

#row5{

width:84vw;

margin: auto;

margin-top: 20px;

margin-left: 22px;

}

#row6{

width:84vw;

margin: auto;

margin-top: 20px;

margin-left: 22px;

}

}

@media only screen and (max-width:600px){

#row4{

width:78vw;

margin: auto;

margin-left: 3px;

}

#row5{

width:78vw;

margin: auto;

margin-top: 20px;

margin-left: 3px;

}

#row6{

width:78vw;

margin: auto;

margin-top: 20px;

margin-left: 3px;

}

}

@media only screen and (min-height:1000px){

#navbar{

height:10vh;

}

#container1::before{

height:39vh;

}

#container1{

height:44vh

}

}

**4.2 Index.html**

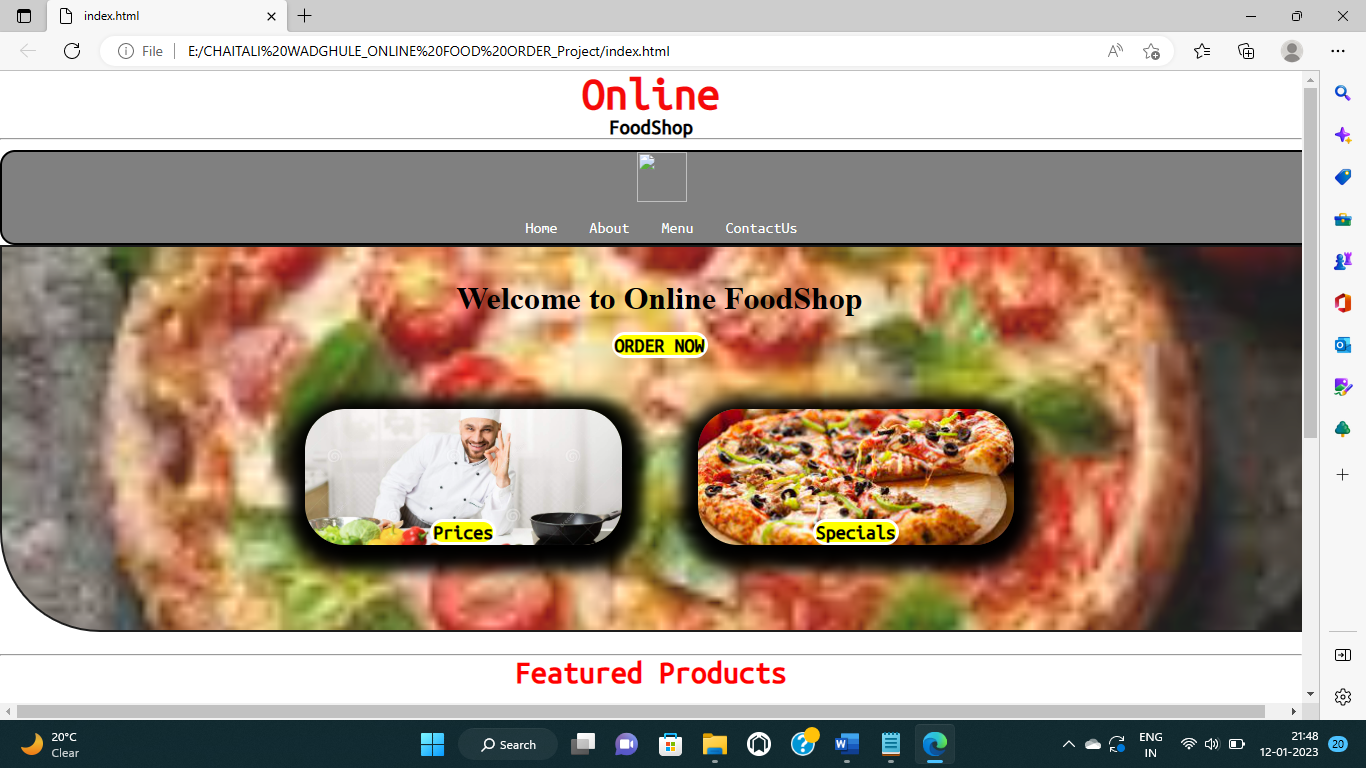
|  |  |
| --- | --- |
|  | <!DOCTYPE html> |
|  | <html lang="en"> |
|  |  |
|  | <head> |
|  | <meta charset="UTF-8"> |
|  | <meta name="viewport" |
|  | content="width=device-width, initial-scale=1.0"> |
|  | <link rel="stylesheet" href="[style.css](file:///E:\CHAITALI%20WADGHULE_ONLINE%20FOOD%20ORDER_Project\style.css)"> |
|  | <link |
|  | href="[https://fonts.googleapis.com/css2?](https://fonts.googleapis.com/css2?%0a%09family=Ubuntu+Mono&display=swap) |
|  | family=Ubuntu+Mono&display=swap" |
|  | rel="stylesheet"> |
|  | </head> |
|  |  |
|  | <body> |
|  | <header> |
|  | <h1 id="top">Online</h1> |
|  | <h1 id="top1">FoodShop</h1> |
|  | </header> |
|  | <hr> |
|  | <nav id="navbar"> |
|  | <img src="[5.jpeg](file:///E:\CHAITALI%20WADGHULE_ONLINE%20FOOD%20ORDER_Project\5.jpeg)"> |
|  | <ul id="navcontent"> |
|  | <li><a href="[#](file:///E:\CHAITALI%20WADGHULE_ONLINE%20FOOD%20ORDER_Project\index.html)">Home</a></li> |
|  | <li><a href="[#](file:///E:\CHAITALI%20WADGHULE_ONLINE%20FOOD%20ORDER_Project\index.html)">About</a></li> |
|  | <li><a href="[#](file:///E:\CHAITALI%20WADGHULE_ONLINE%20FOOD%20ORDER_Project\index.html)">Menu</a></li> |
|  | <li><a href="[contact.html](file:///E:\CHAITALI%20WADGHULE_ONLINE%20FOOD%20ORDER_Project\contact.html)">ContactUs</a></li> |
|  | </ul> |
|  | </nav> |
|  | <div id="container1"> |
|  | <div id="row1"> |
|  | Welcome to Online FoodShop |
|  | </div> |
|  | <button class="btn">ORDER NOW</button> |
|  | <div id="container3"> |
|  | <div id="row2"> |
|  | <button class="btn">Prices</button> |
|  | </div> |
|  | <div id="row3"> |
|  | <button class="btn">Specials</button> |
|  | </div> |
|  | </div> |
|  | </div> |
|  | <hr> |
|  | <h1 id="top3">Featured Products</h1> |
|  | <div id="container4"> |
|  | <div id="row4"> |
|  | <button class="btn">ORDER NOW</button> |
|  | </div> |
|  | <div id="row5"> |
|  | <button class="btn">ORDER NOW</button> |
|  | </div> |
|  | <div id="row6"> |
|  | <button class="btn">ORDER NOW</button> |
|  | </div> |
|  | </div> |
|  | <footer>Copyright © 2020-2021 OnlineFoodShop. |
|  | All Rights are reserved</footer> |
|  | </body> |
|  | </html> |
|  |  |

**4.3 Contact.html**

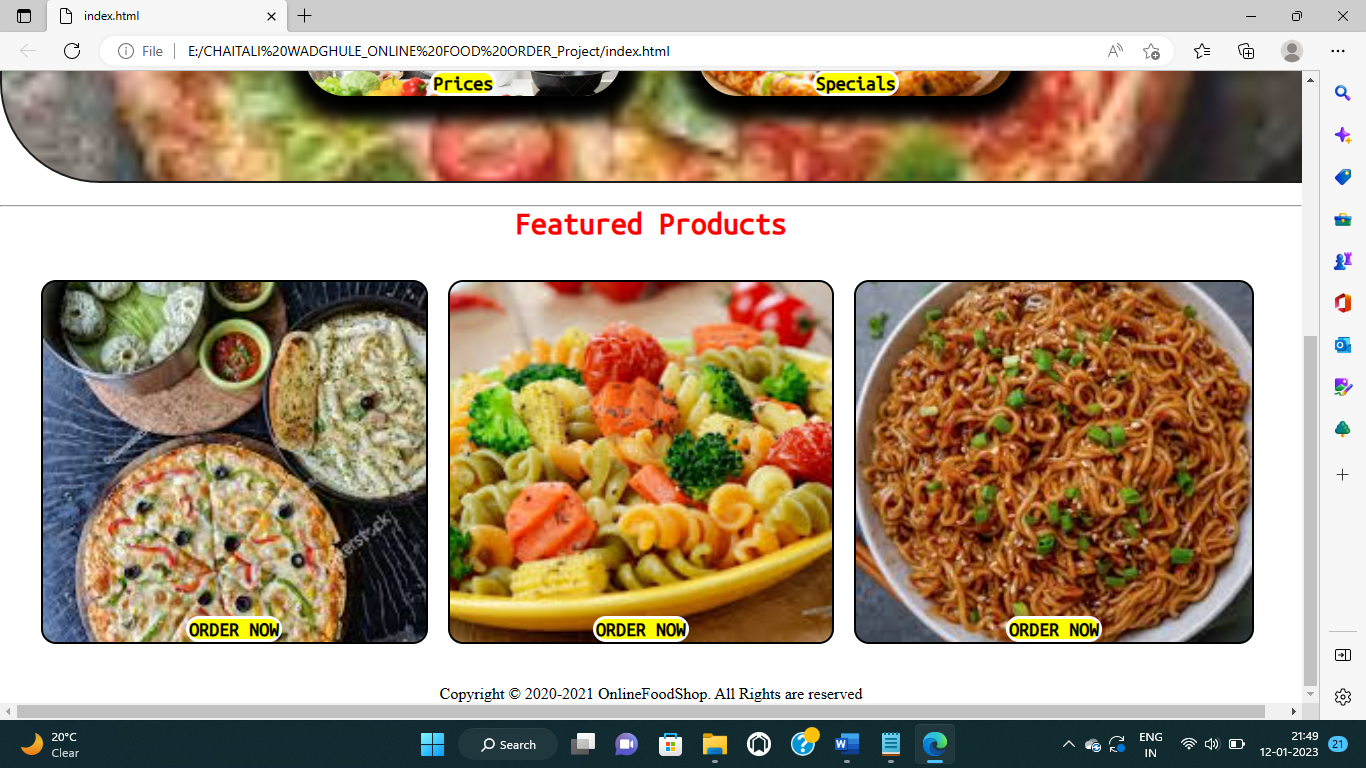
|  |  |
| --- | --- |
|  | <!DOCTYPE html> |
|  | <html lang="en"> |
|  |  |
|  | <head> |
|  | <meta charset="UTF-8"> |
|  | <meta name="viewport" content= |
|  | "width=device-width, initial-scale=1.0"> |
|  |  |
|  | <link rel="stylesheet" href="[style.css](file:///E:\CHAITALI%20WADGHULE_ONLINE%20FOOD%20ORDER_Project\style.css)"> |
|  | <title>ContactUs</title> |
|  | </head> |
|  |  |
|  | <body> |
|  | <div id="ContactUs"> |
|  | <h1>Contact Us</h1> |
|  | <form action="#"> |
|  | <div class="form-shape"> |
|  | <label for="query"> |
|  | Type of Query |
|  | </label> |
|  | <select name="myQuery" id="query"> |
|  | <option value="sel" selected> |
|  | Select |
|  | </option> |
|  | <option value="ord"> |
|  | Order related Issues |
|  | </option> |
|  | <option value="Site"> |
|  | Site related Issues |
|  | </option> |
|  | <option value="fed"> |
|  | Complaint related Issues |
|  | </option> |
|  | <option value="others"> |
|  | Others |
|  | </option> |
|  | </select> |
|  | </div> |
|  | <div class="form-shape"> |
|  | <label for="name">Name</label> |
|  | <input type="text" name="myName" |
|  | id="name" |
|  | placeholder="Enter your Name"> |
|  | </div> |
|  | <div class="form-shape"> |
|  | <label for="email">Email-Id</label> |
|  | <input type="email" name="myEmail" |
|  | id="email" |
|  | placeholder="Enter your email"> |
|  | </div> |
|  | <div class="form-shape"> |
|  | <label for="pho">Phone Number</label> |
|  | <input type="phone" name="myPhone" |
|  | id="pho" |
|  | placeholder="Enter your Phone no"> |
|  | </div> |
|  | <div id="radio"> |
|  | Are you a member of OnlneFoodShop: |
|  | Yes <input type="radio" name="eligible"> |
|  | No <input type="radio" name="eligible"> |
|  | </div> |
|  | <div class="form-shape"> |
|  | <label for="message"> |
|  | Ellaborate your query |
|  | </label> |
|  | <textarea name="mesg" id="message" |
|  | cols="30" rows="10"> |
|  | </textarea> |
|  | </div> |
|  | <input type="submit" value="Submit"> |
|  | <input type="reset" value="Reset"> |
|  | </form> |
|  | </div> |
|  | </body> |
|  |  |
|  | </html> |
|  |  |

**5.MODULES**

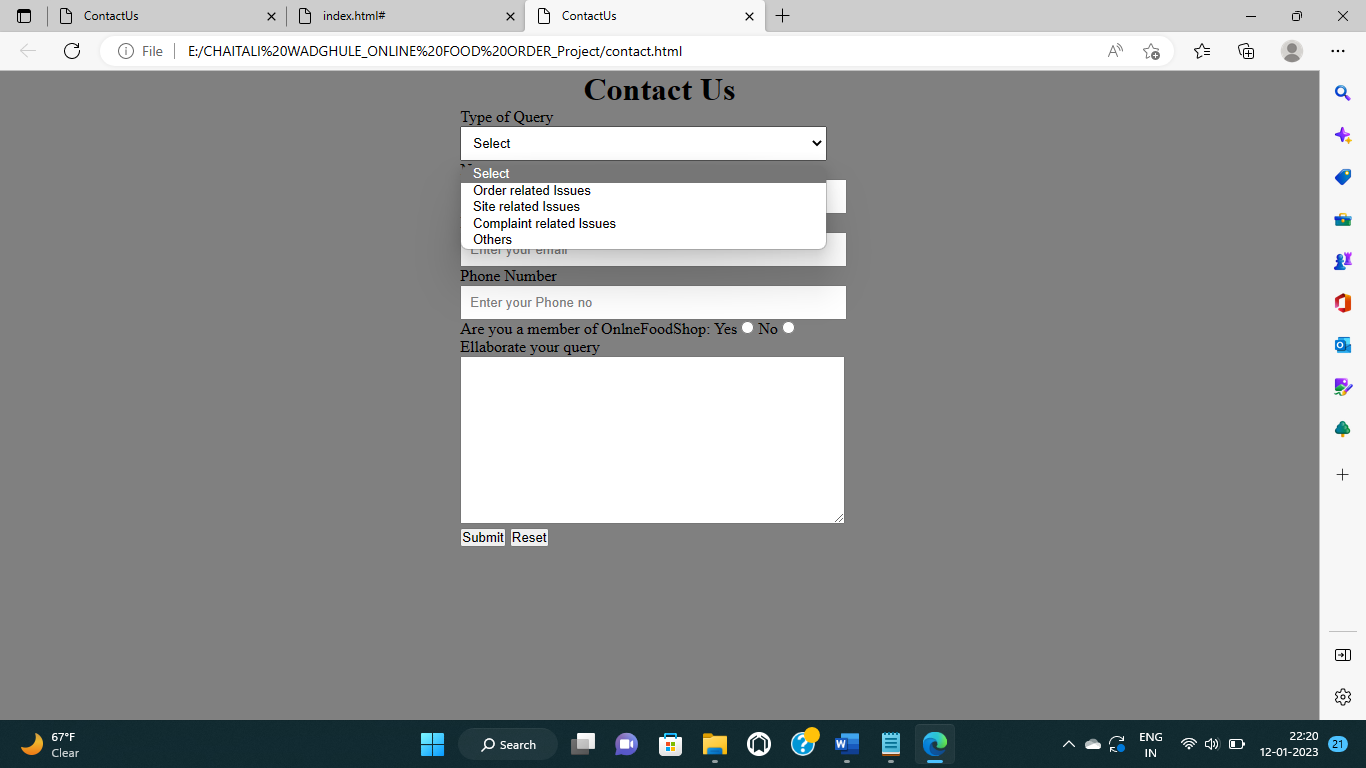
**5.1 Home page**



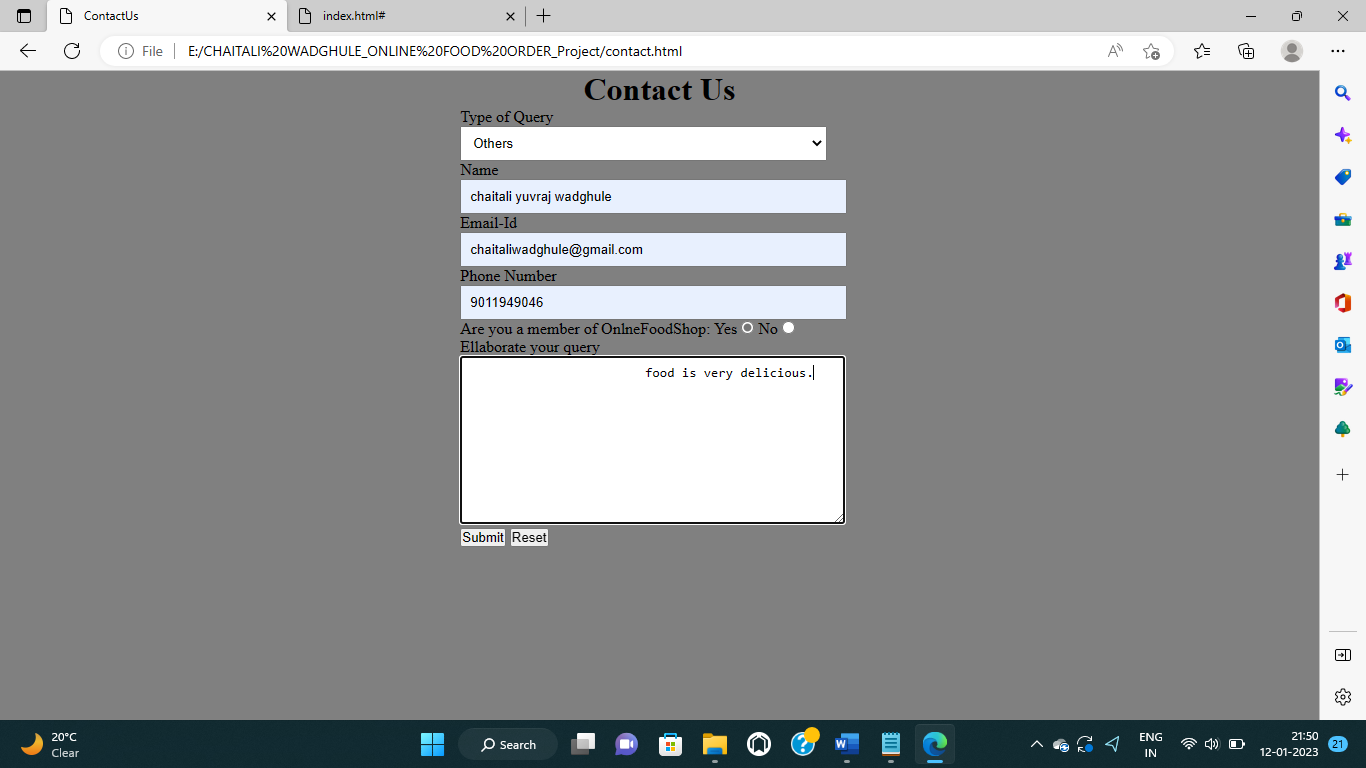
**5.2 About page**



**5.3 Contact Us**



**5.4 Order Now**



**6.SYSTEM TESTING**

**Testing Type**

**Selenium**

**6.1 SELEMIUM TESTING**

**Online.java**

**package** selenium1p;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**public** **class** online{

**public** **static** **void** main(String[] args) {

System.*setProperty*("webdriver.gecko.driver",

"C:\\Users\\chait\\Downloads\\chromedriver\_win32\\chromedriver.exe");

WebDriver driver = **new** ChromeDriver();

//URL launch

driver.get("https://www.dominos.co.in/");

//identify username

WebElement l = driver.findElement(By.*name*("username"));

l.sendKeys("Pass@gmail.com");

//identify password

WebElement p = driver.findElement(By.*name*("password"));

p.sendKeys("Pass@123");

WebElement b = driver.findElement(By.*className*("Igw0E"));

b.click();

//obtain value entered for username

String s = l.getAttribute("value");

System.***out***.println("Value entered for username: " + s);

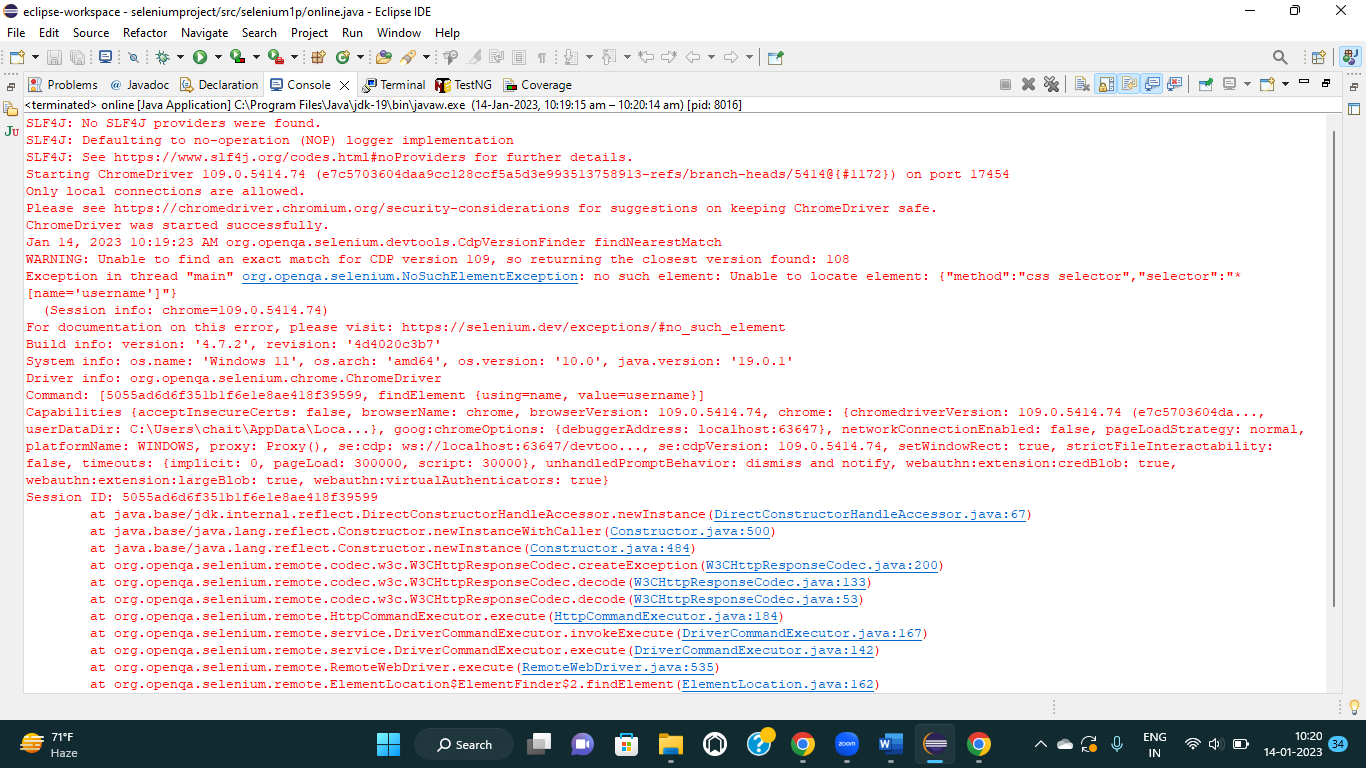
//quit browser

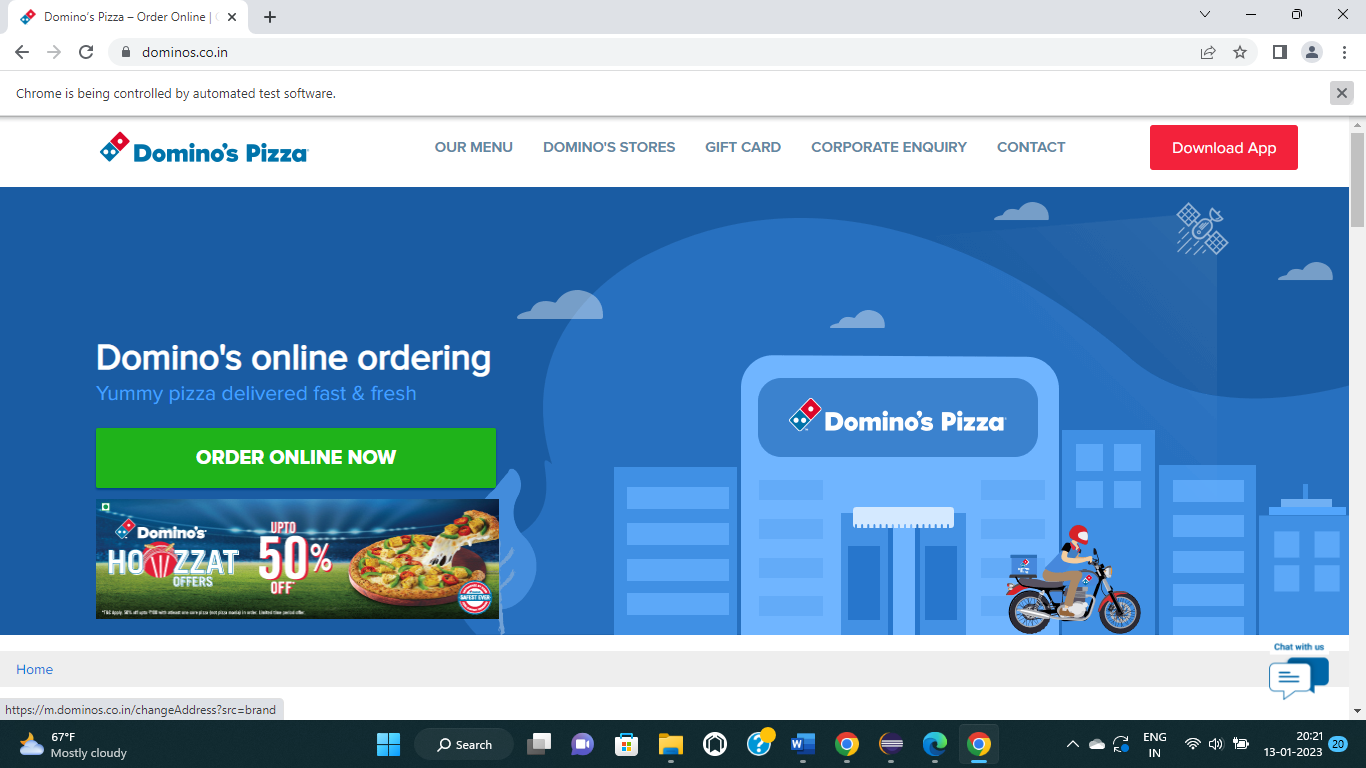
driver.quit();

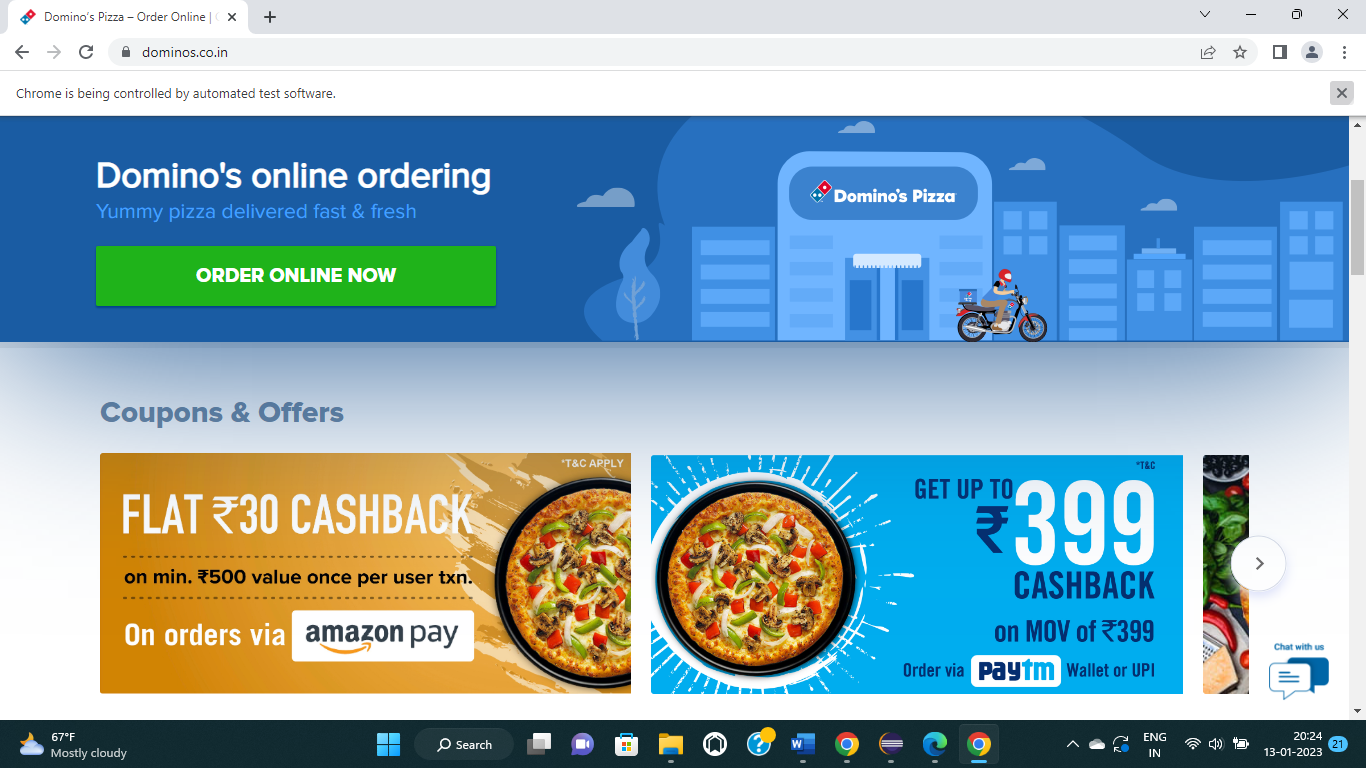
}

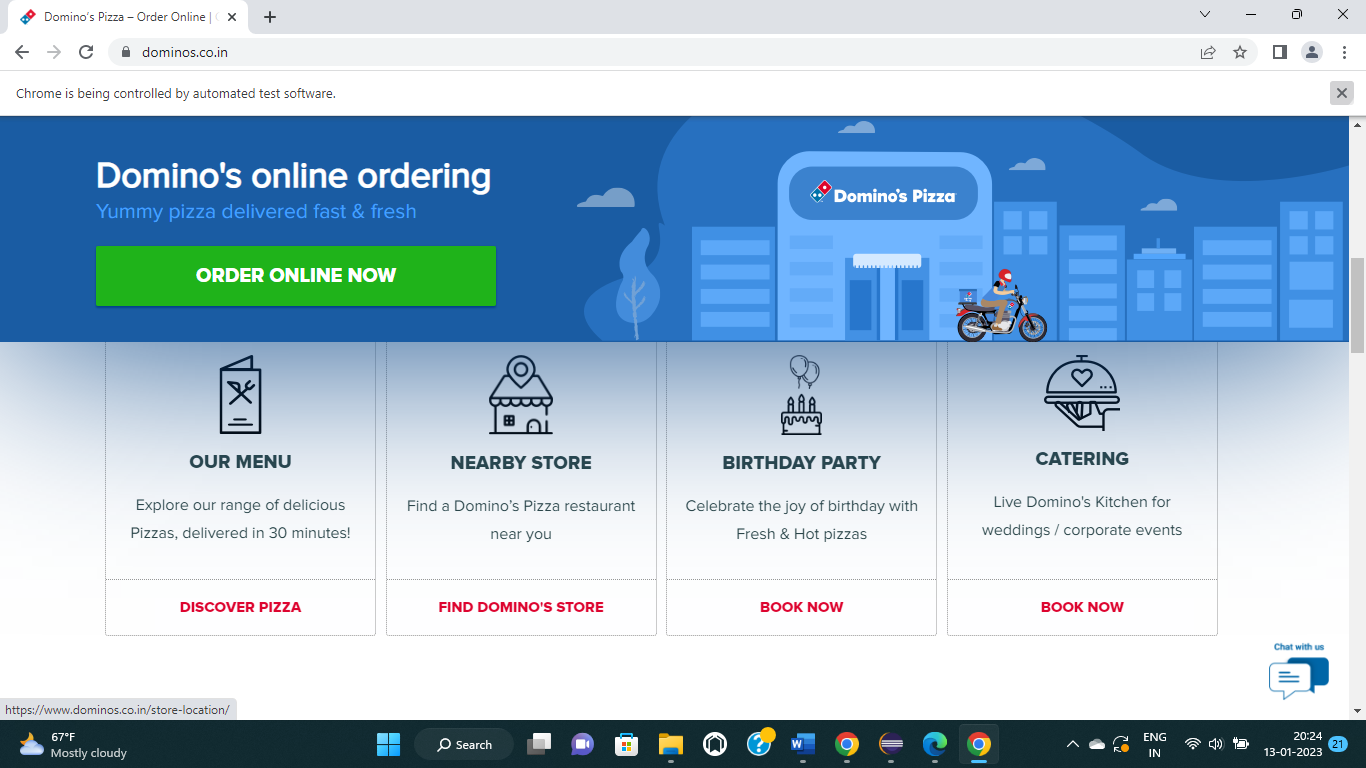
}

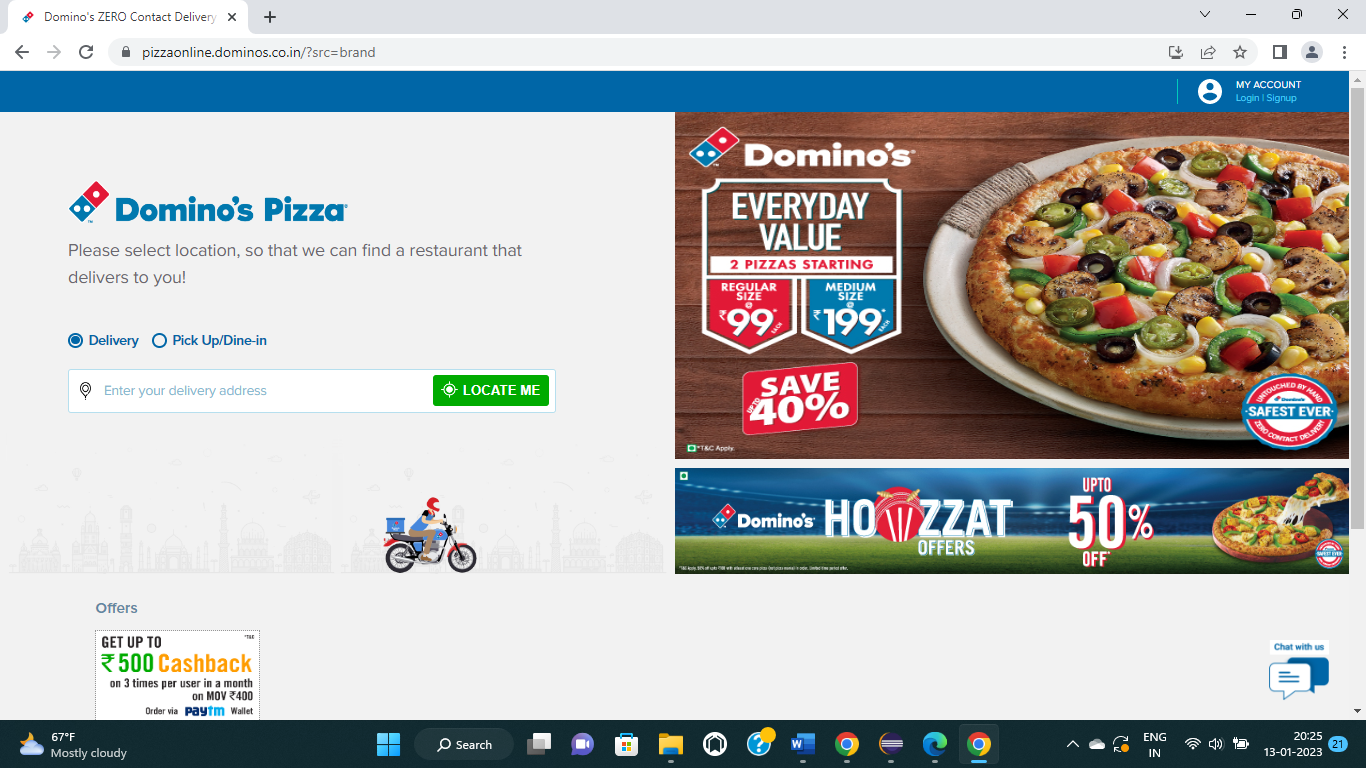
**6.2 Screen Designing**

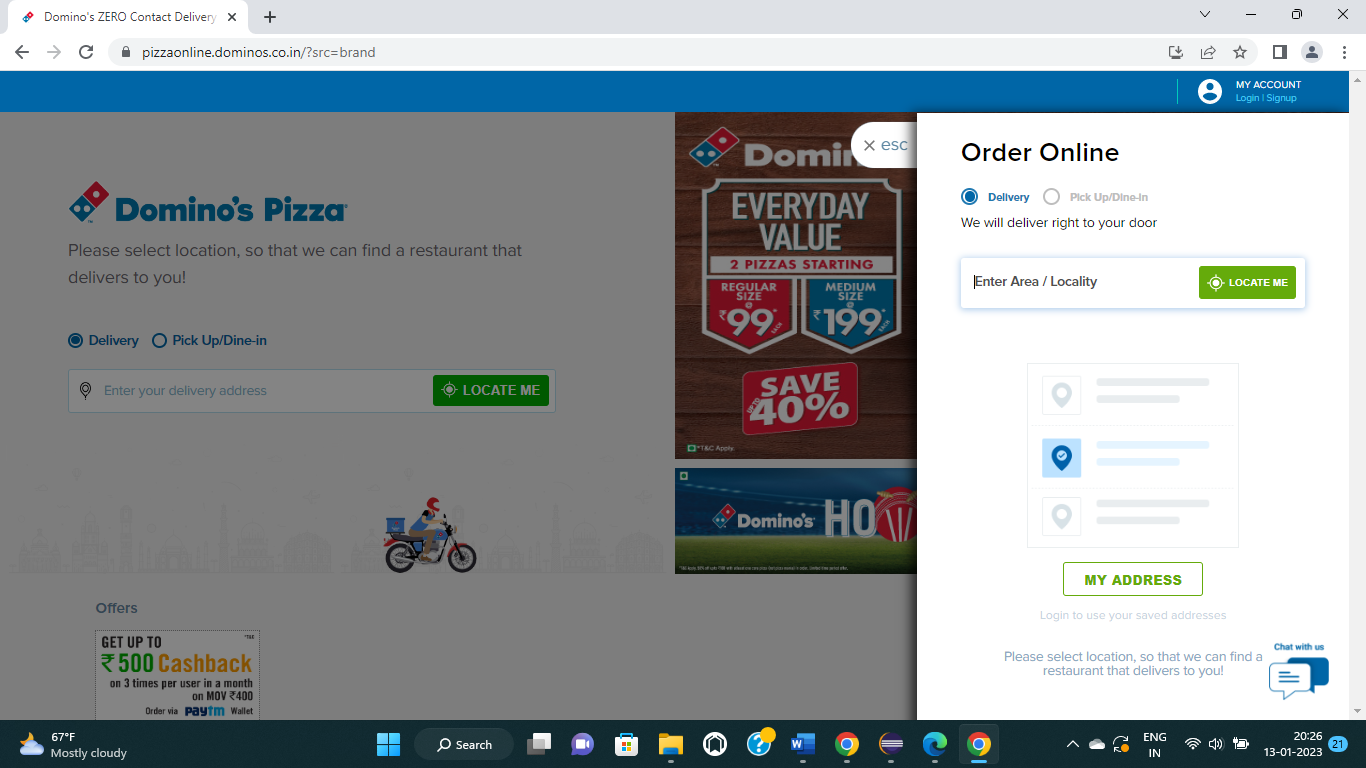


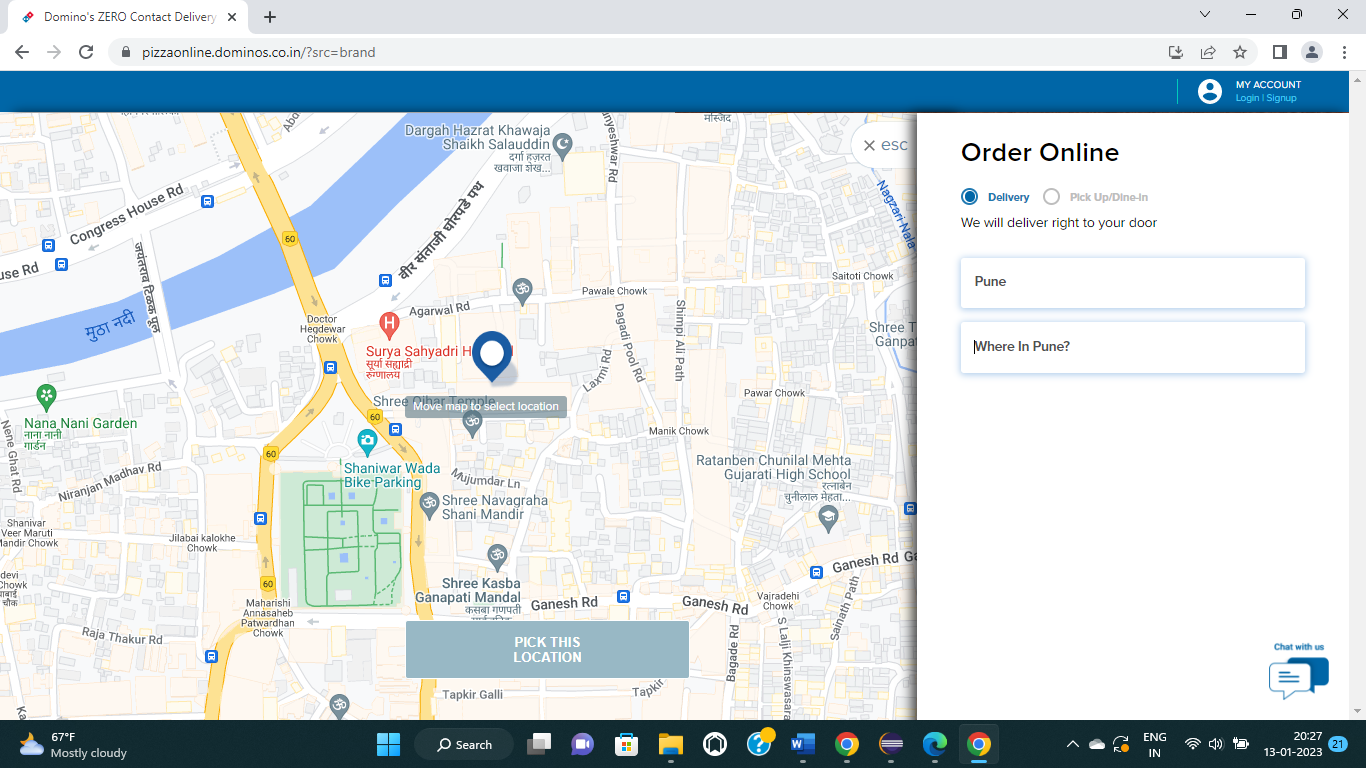




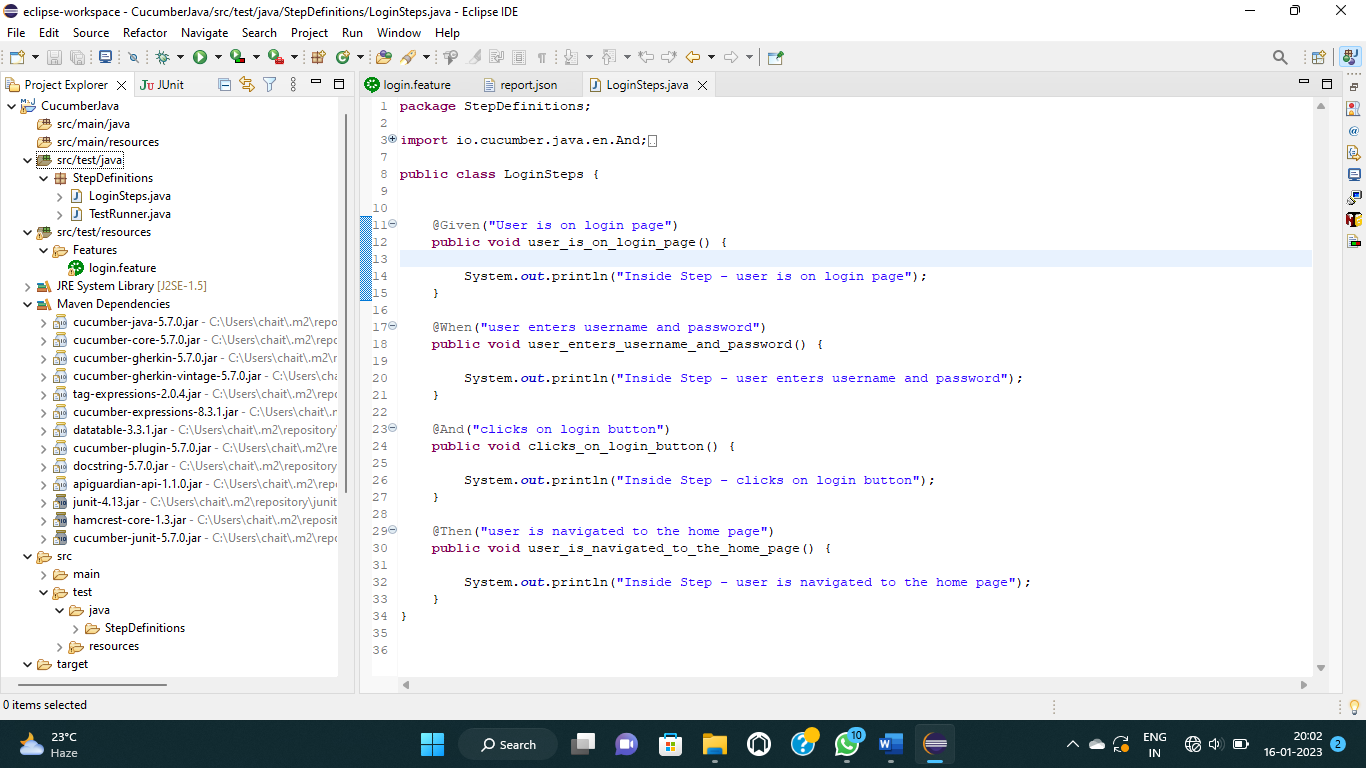


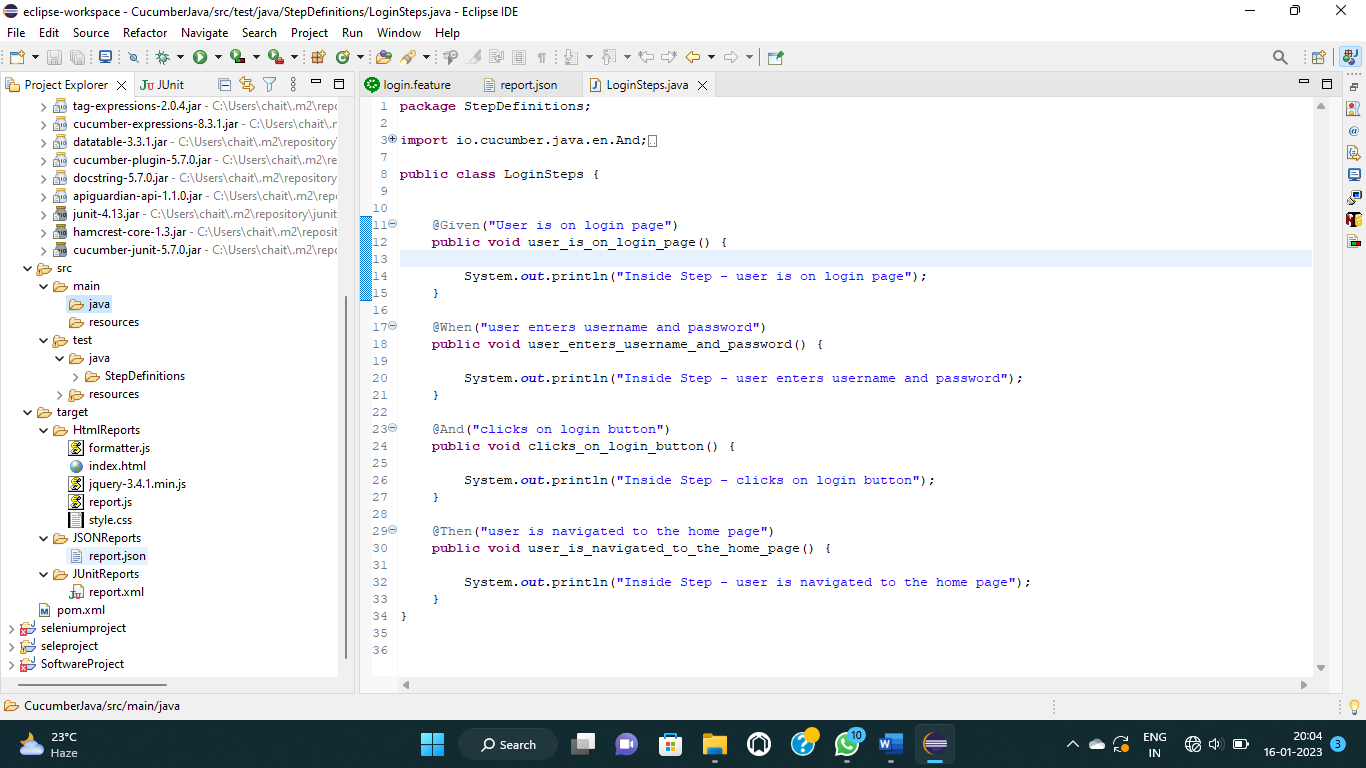






**CUCUMBER**





**LoginSteps.java**

package StepDefinitions;

import io.cucumber.java.en.And;

import io.cucumber.java.en.Given;

import io.cucumber.java.en.Then;

import io.cucumber.java.en.When;

public class LoginSteps {

@Given("User is on login page")

public void user\_is\_on\_login\_page() {

System.out.println("Inside Step - user is on login page");

}

@When("user enters username and password")

public void user\_enters\_username\_and\_password() {

System.out.println("Inside Step - user enters username and password");

}

@And("clicks on login button")

public void clicks\_on\_login\_button() {

System.out.println("Inside Step - clicks on login button");

}

@Then("user is navigated to the home page")

public void user\_is\_navigated\_to\_the\_home\_page() {

System.out.println("Inside Step - user is navigated to the home page");

}

}

**TestRunner.java**

package StepDefinitions;

import io.cucumber.junit.CucumberOptions;

import org.junit.runner.RunWith;

import io.cucumber.junit.Cucumber;

@RunWith(Cucumber.class)

@CucumberOptions(features="src/test/resources/Features",glue={"StepDefinitions"},

monochrome = true,

plugin = {"pretty","junit:target/JUnitReports/report.xml",

"json:target/JSONReports/report.json",

"html:target/HtmlReports"},

tags="@smoketest"

)

public class TestRunner {

}

**login.feature**

#Author

#Date

#Description

@SmokeFeature

Feature: feature to test login functionality

@smoketest

Scenario: Check login is successful with valid credentials

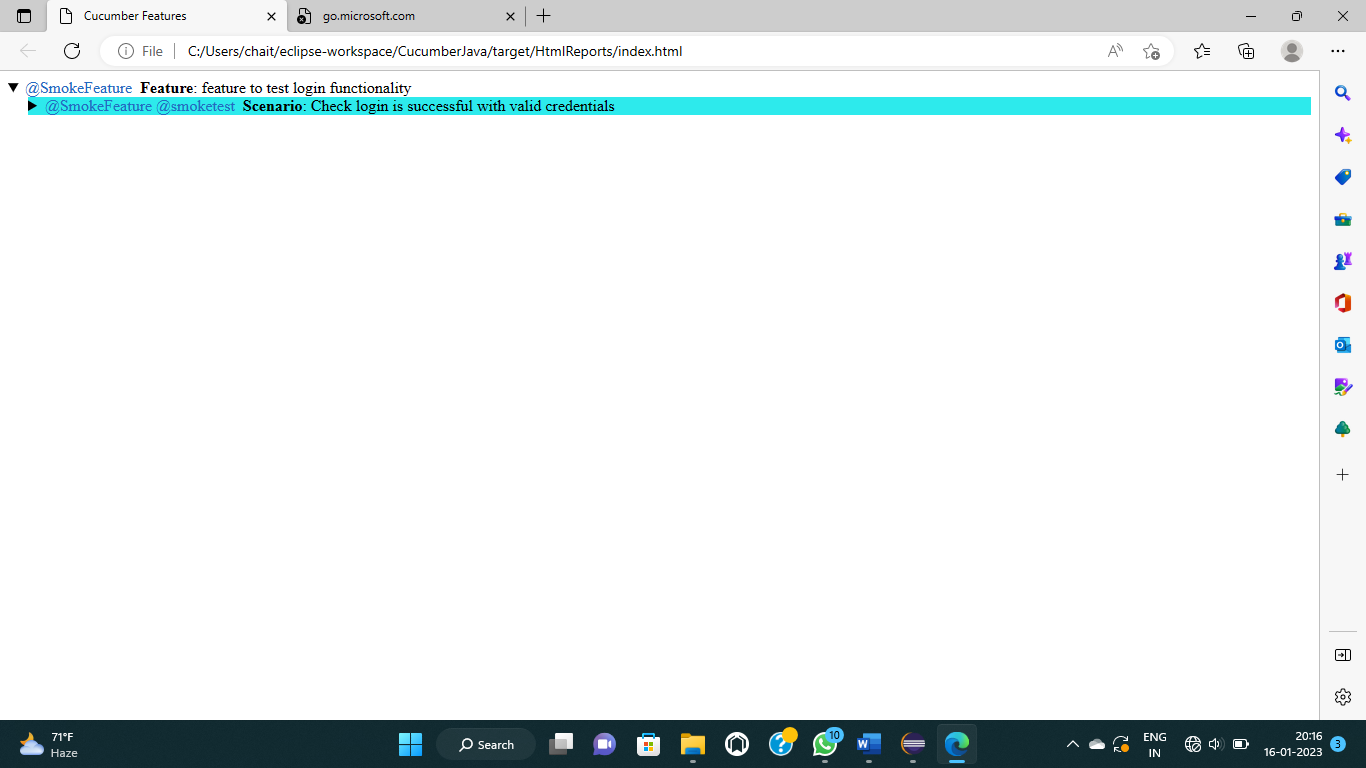
Given user is on login page

When user enters username and password

And clicks on login button

Then user is navigated to the home page

**Index.html**



**Style.css**

.cucumber-report .body {

font-family: Helvetica,Arial,sans-serif;

}

.cucumber-report .keyword {

font-weight: bold;

}

.cucumber-report .description {

font-style: italic;

margin-left: 20px;

white-space: pre;

}

.cucumber-report details > section {

margin-left: 20px;

}

.cucumber-report ol.steps {

list-style-type: none;

margin-top: 0;

margin-bottom: 0;

}

.cucumber-report .step .embedded-text {

background: #dddddd;

}

.cucumber-report .doc\_string {

margin: 0 0 0 20px;

}

.cucumber-report table {

border-collapse: collapse;

border: 1px;

border-style: solid;

}

.cucumber-report td, .cucumber-report th {

border: 1px;

border-style: solid;

padding-left: 4px;

padding-right: 4px;

}

.cucumber-report table {

margin-left: 20px;

}

.cucumber-report thead {

background-color: #C0C0C0;

}

.cucumber-report .passed {

background-color: #C5D88A;

}

.cucumber-report .undefined, .cucumber-report .pending {

background-color: #EAEC2D;

}

.cucumber-report .skipped {

background-color: #2DEAEC;

}

.cucumber-report .failed {

background-color: #D88A8A;

}

.cucumber-report .tags {

display: inline;

}

.cucumber-report .tag {

margin-right: 0.25em;

color: #246ac1;

}

.cucumber-report .comments {

display: inline;

}

.cucumber-report .comment {

margin: 0;

padding: 0;

}

.cucumber-report .error {

margin: .2em .75em;

padding: .2em;

border: 1px solid #900;

background-color: #EDBBBB;

}

#cucumber-templates {

display: none;

}

**Report.json**

[

{

"line": 6,

"elements": [

{

"start\_timestamp": "2023-01-06T18:47:01.353Z",

"line": 9,

"name": "Check login is successful with valid credentials",

"description": "",

"id": "feature-to-test-login-functionality;check-login-is-successful-with-valid-credentials",

"type": "scenario",

"keyword": "Scenario",

"steps": [

{

"result": {

"status": "undefined"

},

"line": 11,

"name": "user is on login page",

"match": {},

"keyword": "Given "

},

{

"result": {

"duration": 1003200,

"status": "skipped"

},

"line": 12,

"name": "user enters username and password",

"match": {

"location": "StepDefinitions.LoginSteps.user\_enters\_username\_and\_password()"

},

"keyword": "When "

},

{

"result": {

"duration": 989300,

"status": "skipped"

},

"line": 13,

"name": "clicks on login button",

"match": {

"location": "StepDefinitions.LoginSteps.clicks\_on\_login\_button()"

},

"keyword": "And "

},

{

"result": {

"status": "skipped"

},

"line": 14,

"name": "user is navigated to the home page",

"match": {

"location": "StepDefinitions.LoginSteps.user\_is\_navigated\_to\_the\_home\_page()"

},

"keyword": "Then "

}

],

"tags": [

{

"name": "@SmokeFeature"

},

{

"name": "@smoketest"

}

]

}

],

"name": "feature to test login functionality",

"description": "",

"id": "feature-to-test-login-functionality",

"keyword": "Feature",

"uri": "file:src/test/resources/Features/login.feature",

"tags": [

{

"name": "@SmokeFeature",

"type": "Tag",

"location": {

"line": 5,

"column": 1

}

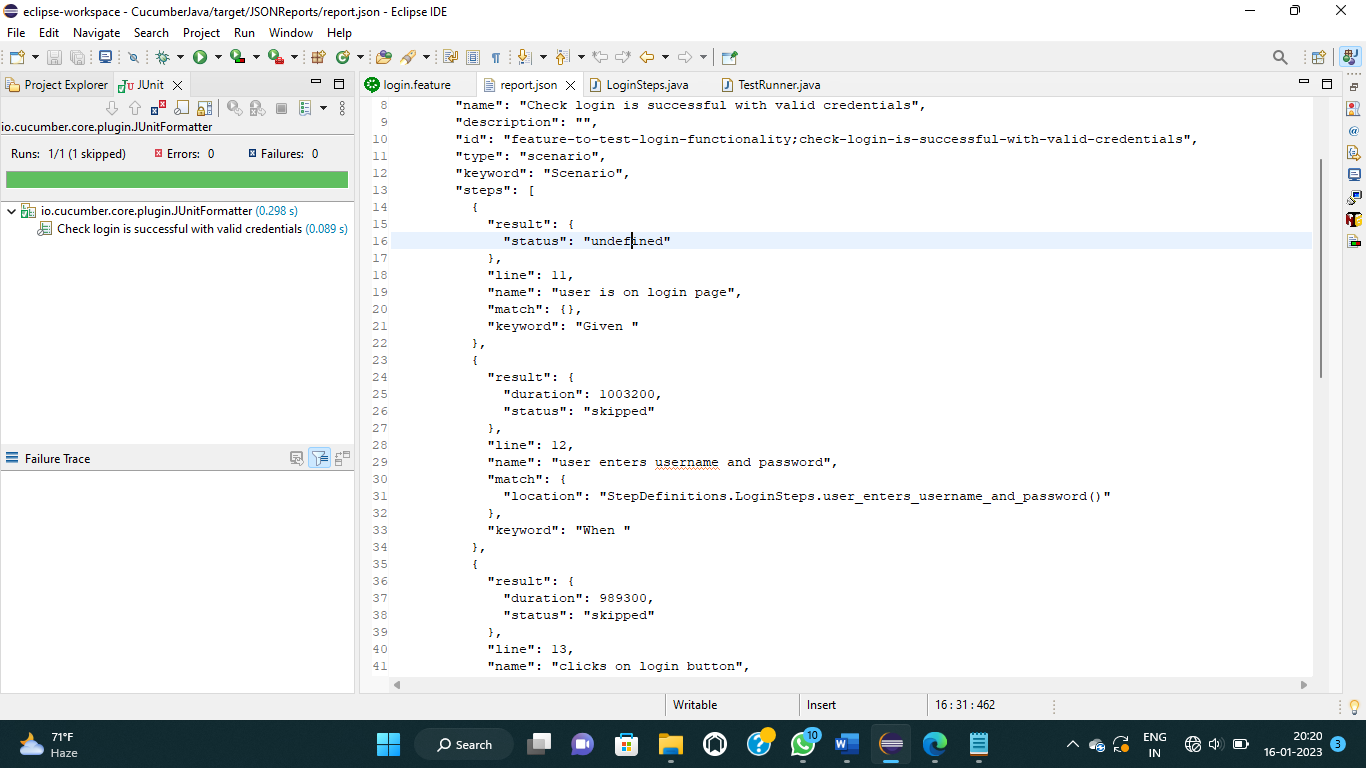
}

]

}

]

**report.xml**



**Pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>cucumberJava</groupId>

<artifactId>cucumberJava</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-java -->

<dependency>

<groupId>io.cucumber</groupId>

<artifactId>cucumber-java</artifactId>

<version>5.7.0</version>

</dependency>

<!-- https://mvnrepository.com/artifact/junit/junit -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13</version>

<scope>test</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/io.cucumber/cucumber-junit -->

<dependency>

<groupId>io.cucumber</groupId>

<artifactId>cucumber-junit</artifactId>

<version>5.7.0</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

**6.3 White Box Testing**

In this technique, the close examination of the logical parts through the software are tested by cases that exercise species sets of conditions or loops. all logical parts of the software checked once. errors that can be corrected using this technique are typographical errors, logical expressions which should be executed once may be getting executed more than once and error resulting by using wrong controls and loops. When the box testing tests all the independent part within a module a logical decisions on their true and the false side are exercised , all loops and bounds within their operational bounds were exercised and internal data structure to ensure their validity were exercised once.

**6.4 Black Box Testing**

This method enables the software engineer to device sets of input techniques that fully exercise all functional requirements for a program. black box testing tests the input, the output and the external data. it checks whether the input data is correct and whether we are getting the desired output.

White Box Testing:

In this technique, the close examination of the logical parts through the software are tested by

cases that exercise species sets of conditions or loops. all logical parts of the software checked

once. errors that can be corrected using this technique are typographical errors, logical

expressions which should be executed once may be getting executed more than once and error

resulting by using wrong controls and loops. When the box testing tests all the independent part

within a module a logical decisions o

White Box Testing:

In this technique, the close examination of the logical parts through the software are tested by

cases that exercise species sets of conditions or loops. all logical parts of the software checked

once. errors that can be corrected using this technique are typographical errors, logical

expressions which should be executed once may be getting executed more than once and error

resulting by using wrong controls and loops. When the box testing tests all the independent part

within a module a logical decisions on their true and the false side are exercised , all loops and

bounds within their operation

**6.5 Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sr.  No | Test  Cases  ID | Feature | Description | Steps to Execute | Test Data/Input | Expected Result |
| 1 | TC\_01 | First name | 1. Verify that the name field design should be have as per the customer’s specification | 1.Enter the name in start with Capital letter | 1.Chaitali | It should not show any validation message |
| 2 | TC\_02 | Last name | 1. Verify that the name field design should be have as per the customer’s specification | 1.Enter the name in start with Capital letter | 1.Wadghule | It should not show any validation message |
| 3 | TC\_03 | Email validation | 1. Check the Email text field with an Email address without @ symbol  2. Check the Email text field with a random string instead of a real email   3. Check the Email text field that has @ symbol written in words 4. Check the Email text field with a missing dot in the email address | 1. Enter Invalid Emails 2. Click on the Register Button | 1. testAtgmail.com 2. test@gmailcom 3. test@gmail 4. @gmail | It should show the validation message for valid email |
| 4 | TC\_04 | Email validation | Check all the valid emails | 1. Enter valid Emails 2. Click on the Register Button. | 1. [test.22@gmail.com](mailto:test.22@gmail.com) 2. [test@gmail.com](mailto:test@gmail.com) | It should not show any validation message |
| 5 | TC\_05 | Phone Number validation | Check the phone number when passing alphanumeric data | 1. Enter alphanumeric data in the Phone field 2. Click on Register button | 1. dada5$7567#7 | It should show the validation message for Phone Number |
| 6 | TC\_06 | Phone Number validation | Check the phone number when not pass country code | 1. Enter a valid phone number without country code 2. Click on the Register button | 1. 9012078654 | It should show the validation message for country code is required |
| 7 | TC\_07 | Phone Number validation | Check the phone number when passing country code | 1. Enter a valid phone number with country code 2. Click on Register button | 1. +9190112244 | It should not show any validation error message |
| 8 | TC\_08 | Phone Number validation | Verify if the phone number length is incorrect, i.e., more than ten characters | 1. Enter a phone number with less than ten digits. 2. Enter all the required fields. 3. Click on Register Button | 91901122445566 | It should show the validation error message for phone number length. |

**7.ADVANTAGES**

* **Easy, fast, and comfortable:**

In short, your customers choose to order food online because it is really at their fingertips.

* **Health benefits:**

One of the important benefits of food ordering systems is health benefits. Because the meal is planned, it is easy to determine the exact number of calories consumed in each meal.

* **Safer and healthier:**

To reopen, food businesses will have to set up shop to meet the health and safety regulations of the Indian government.

* **Less chance for errors:**

One of the best advantages of an online food ordering system for customers is that it ensures prices are accurate and there is less room for error when it comes time to settle the bill.

* **More customers:**

As the new life progresses with technologies, online orders and payments are expected to be accepted. If your payment and menu method is hassle free, your regular customers will recommend you to their friends and will share on social media about your restaurant.

.

* **Increased customer loyalty:**

If you give customers a reason to come back, they will choose your store over your competitor. You can promote their loyalty through the loyalty program.

* **Higher customer spend:**

We all know that more and more customers are now engaging in digital products and services than ever before.

They also spend more when ordering online. Because reading the online menu is different from standing in line.

.

* **Highly customizable:**

Food ordering apps are highly customizable so you can easily advertise your logo, brand colors, or other features that make your business unique. Additionally, if you want to delete or add an item to the menu, you must sign in, make your changes, and it’s done.

**8.LIMITATIONS:**

* **Price:**

One of the major drawbacks of online food ordering systems is price. When food is ordered for more than one person, the cost is usually equal to eating at a good restaurant every night.

* **Limited menu:**

Another disadvantage for food ordering systems is menu choices. Most food ordering systems have a limited number of meals. The menu changes every few weeks or months.

* **Preparation:**

The preparation factor may be a disadvantage to food ordering systems. Most food ordering systems give frozen food. They are usually easy to prepare.

* **Quality of food may be suffer:**

  One problem with the food ordering system is that the quality of the food served is often worse than eating at a restaurant.

* **Food may get cold:**

Due to the long ordering distances, your food may also be cold once it is finally delivered to your home. You need to reheat it or eat it cold.

* **The vibe of the restaurant is missing:**

In some restaurants, there is also a good circumstance which you will miss if you order your food at home.

For example, if you spend your evening in a good Chinese restaurant, you will often feel like you are actually in China because the decoration and the whole atmosphere are in line with the Chinese way of life.If you order food at your home, you will lose all of these.

Also, from time to time, it would be great if you could take your partner or family to a nice restaurant for dinner to spend a good evening.

* **Wrapping up:**

Despite some drawbacks, online food ordering systems generally benefit customers and restaurants. By trying out new restaurants (or ordering convenient meals) and helping restaurants attract new customers, customers can taste the delicacies foods as if they were in their bed. We understand both sides of this mess. brilliant technologies are here, Contact us to provide growth to your online food ordering system through mobile apps.

**9.FUTURE ENHANCEMENT**

Ordering food online gains popularity among the restaurant customers in recent decades due to its convenience. Earlier, it was a troublesome task for restaurant owners to receive orders over phone calls, initiate the process manually, and deliver it in a chaotic way.

* **Drive sales online**

The arrival of smartphones and the usage of social media have brought the world closer. Even the small-scale businesses started using online platforms such as website, food-ordering applications, or even social media to receive orders from their restaurant customers and deliver food with ease.

* **Ease the ordering and delivering process**

Making calls and ordering food would be a waste of time where there is a possibility of missing orders. Even the customers will have no idea whether they will receive the food or not.

* **Outstrip conventional ordering method**

The food ordering system has provided a comfort that has outlawed the traditional way of ordering food. The restaurant business owners have started understanding the reality that the conventional way of receiving orders from customers will not work anymore.

* **Outlaw aggregation model**

Besides, the restaurant owners have to spend a lump-sum amount for the aggregators (third-party) to reach their customers. There are a few businesses that connect the restaurants and the customers and allow customers to order food through their ordering portals by getting commission from the restaurant owners.

**10.CONCLUSIONS**

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

* A description of the background and context of the project and its relation to work already done in the area
* Made statement of the aims and objectives of the project.
* The description of Purpose. Scope, and applicability
* We define the problem on which we are working in the project.
* We describe the requirement Specifications of the system and the actions.