

**A Project Report on**  
**Hotel Management System**

**Submitted to**

**Edubridge Learning Private Limited**

**In partial fulfillment of the requirements for the award of**

**Course in**

**SOFTWARE TESTING**

**By**

**Ms. Diksha Ganpatrao Kunjarkar**

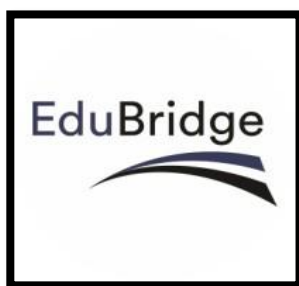
**(Center–Thane, Batch code –7670)**

**Under Guidance of**

**Mrs. Mohana Priya B.E M.E (Ph.D.)**

**(Center Trainer)**

**Course for Software Testing**



**Edubridge Learning Private Limited**

**2022-2023**

# **EDUBRIDGE LEARNING COURSE**

## **BONAFIDE CERTIFICATE**

Certified that this project report “.....**Hotel Management System.....**”is the Bonafide work of “.....**Diksha G. Kunjarkar.....**” who carried out the miniproject work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other these or dissertation on these basis of which a degree or award was conferred on an earlier occasion on any other candidate.

SIGNATURE OF THE SUPERVISOR

**Mrs. MOHANA PRIYA B.E M.E (Ph.D.)**

TRAINER OF SOFTWARE TEST ENGINEER

EDUBRIDGE LEARNING COURSE

# **TABLE OF CONTENT**

## **1. SYSTEM ANALYSIS**

- 1.1 Introduction
- 1.2 Objective of the system
- 1.3 Scope of project
- 1.4 Hardware & Software Requirements

## **2. SYSTEM DESIGN**

- 2.1 Use Case Diagram
- 2.2 Activity Diagram
- 2.3 Class Diagram
- 2.4 ER Diagram
- 2.5 Data Level Flow Diagram

## **3. SYSTEM CODING**

- 3.1 Implementation
- 3.2 Coding
- 3.3 Database Connectivity
- 3.4 Screenshot

## **4. SYSTEM TESTING**

- 4.1 Types of Testing
- 4.2 Testing Methodologies
- 4.3 Equivalence Class Partitioning
- 4.4 Test Reports
- 4.5 Done the Automated selenium of Amazon
- 4.6 Done the Automated selenium of Cucumber
- 4.7 Conclusion

## LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
JS	JavaScript
SQL	Structured Query Language

## 1. SYSTEM ANALYSIS

### 1.1 Introduction:

The **Hotel Management System**, referred to as **HMS**, is an application that will help users better utilize rooms used by UP employees and other guests. HMS helps users manage guest flows by affording them the ability to easily check UP guests in, check them out, and generate stay reports, among other things. This help text will outline some of the most common processes you will perform in HMS.

### 1.2 Objective:

This website is developed using HTML & to add designs and make it more attractive we used CSS and JavaScript to make it as best. we can modify or drop our room type or our booking.

HTML is used to prepare the main web page for the hotel booking system. CSS is used to make it more attractive and elegant. Customers are always attracted to the attraction we made for them. different types of loading are used for different font styling in it.

And also it includes the address and contact number of the officials and members of the hotel if there is any issue is there for any customer they can easily contact and resolve their issue.

Nowadays we all know very well that people are shifted to the world of the internet. If any random person goes to some unknown place and needs to take the hotel what he can do in an unknown place. That is why the internet is the best thing around us. Our HTML, CSS, and Java-Script based web application project is to develop a hotel management system.

## **Interface (UI) of Online Hotel Management System:**

The user interface is a very important part of the system. This helps every member of the system to interact with each detail properly. As we have shown several use cases in this system. We have developed these interfaces to interact with the system.

### **Login Page:**

Need to log in using login id and password

### **Home Page:**

Can view their details as well as Booking details.

### **Booking Table:**

Customer can book table.

### **Payment Page:**

This interface helps in paying the money via several modes.

### **Contact Page:**

Customer can contact to the manger.

### **Menu Page:**

Detailed lists of food items.

## **1.3 Scope of Project:**

The introducing software, Hotel Management System which is going to be implemented for Hotel will automate the major operations of the hotel. The Reservation System is to keep track in room and hall reservation and check availability. Customer come for an enquiry our system will give the update status of room, and it will help the customer to choose the room.

## **1.4 Hardware & Software Requirements:**

### **Software Requirements:**

- Operating system : Windows 10
- Web browser : Google chrome
- Program code : Html, CSS, JavaScript

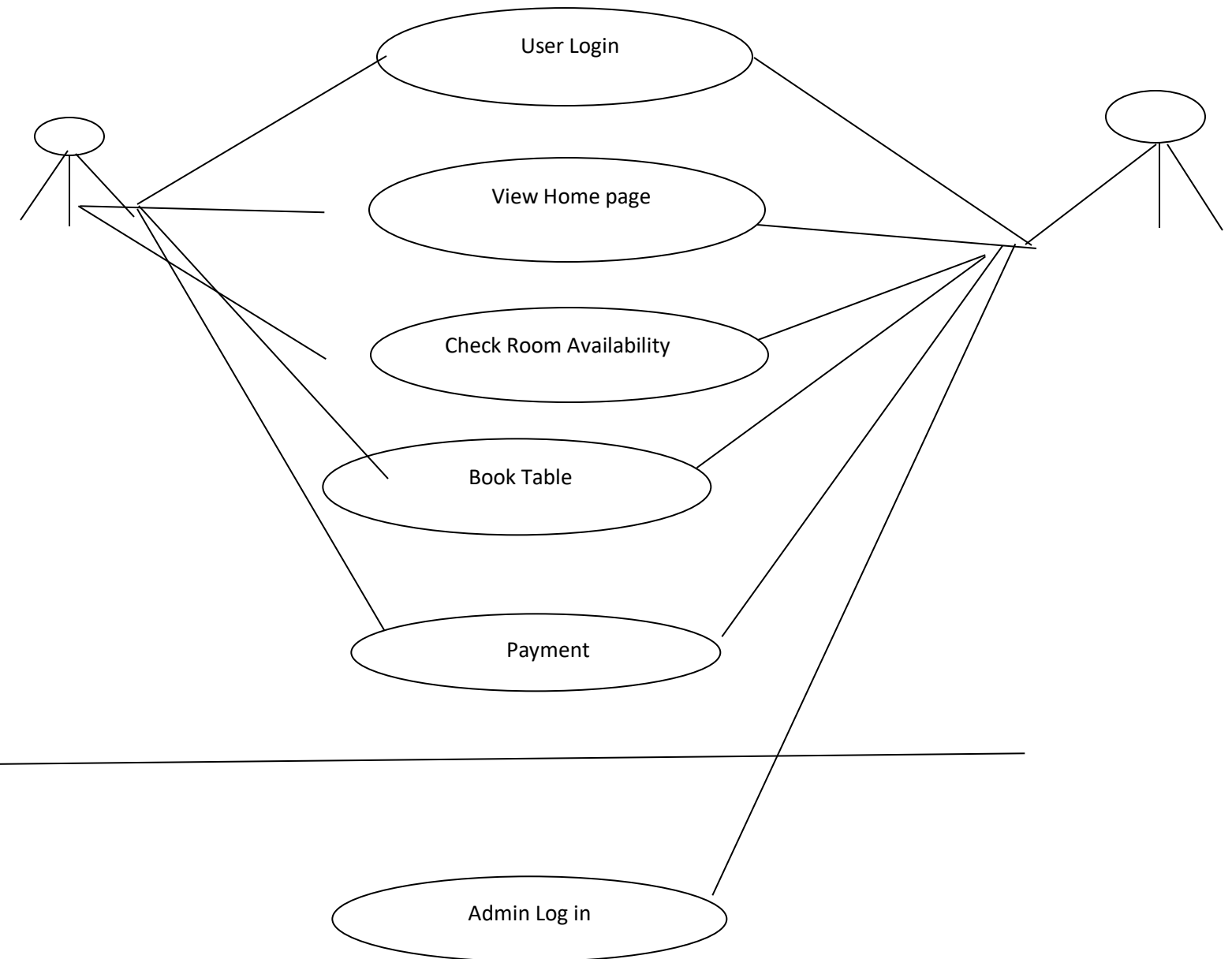
### **Hardware Requirements:**

- Processor : AMD E2-9000 RADEON R2, 4 COMPUTE CORES 2C+2G
- System type : 64-bit operating system, x64-based processor
- Pen and touch : Touch support with 10 touch points
- Version : 22H2
- CD-RO : Required

## 2. System Design:

### 2.1 Use Case diagram

#### Hotel Booking Management System



## Use case Description:

### Use case 001: User Login

#### 1.Introduction

This use case outlines the steps that need to be followed in order to login into the system.

#### 2. Actors

Admin, User

#### 3.Pre-Condition

The user/admin must have valid credentials

Scenario

	ACTION	Software Reaction
1.	Customer fills out their <b>name</b>	The system verify that all the required fields have been <b>registered</b> .
2.	Customer fills out their Password	Check the correct or <b>incorrect password</b> .

#### 4. Post-Condition

If use case is successfully executed, the user/admin should be logged into the system,

#### 5. Basic Flow

- Valid Login
- The page will request the user/actor to provide valid credentials



## 6. Alternate Flow

- Invalid Credentials
- If user/actor provides invalid credentials in the basic flow, a validation message or error message should appear. Hence, returning the user to the basic flow.

## 7. Special Requirements

None

## 8. Associated Use Case(s)

None

# Use case 002: Book a Table

## 1. Introduction

This use case outlines the steps that need to be followed in order to login into the system

## 2. Actors

- Customer

## 3. Pre-Condition

The customer must have valid credentials.

## 4. Post-Condition

If use case is successfully executed, the user/actor should be logged into the system, otherwise, the state remains unchanged.

## 5. Basic Flow

- Login
  - ✓ The page will request Customer have to give valid credentials.
  - ✓ Passenger have to enter valid credentials.
  - ✓ Passenger will log into system.

## 6. Alternate Flow

- Invalid Credentials
  - ✓ If user/actor provides invalid credentials in the basic flow, a validation message or error message should appear. Hence, returning the user to the basic flow..

## 7.Special Requirements : None

## 8. Associated Use Case(s) :None

## 9.Scenario:

	Action (stimulus)	Software Reaction
1.	The Customer indicates that they would like to booking there onlinetable.	System request the following data from the customer : <ul style="list-style-type: none"><li>• Full Name</li><li>• Day</li><li>• Phone number</li><li>• Time</li><li>• Payment</li></ul>
2.	Customer fill out data	<ul style="list-style-type: none"><li>• Full Name</li><li>• Day</li><li>• Phone number</li><li>• Time</li><li>• Payment</li></ul>

## **Use case 003: Payments**

### **1. Introduction**

This use case outlines the steps that need to be followed in order to login into the system

### **2. Actors**

- Admin
- Customer

### **3. Pre-Condition**

The customer must have valid credentials.

### **4. Post-Condition**

If use case is successfully executed, the user/actor should be logged into the system, otherwise, the state remains unchanged.

### **5. Basic Flow**

- Registration
  - ✓ The page will request Customer have to give valid credentials.
  - ✓ Customer have to entre valid credentials.
  - ✓ Customer will log into system.

### **6. Alternate Flow**

- Invalid Credentials

- ✓ If user/actor provides invalid credentials in the basic flow, a validation message or error message should appear. Hence, returning the user to the basic flow..

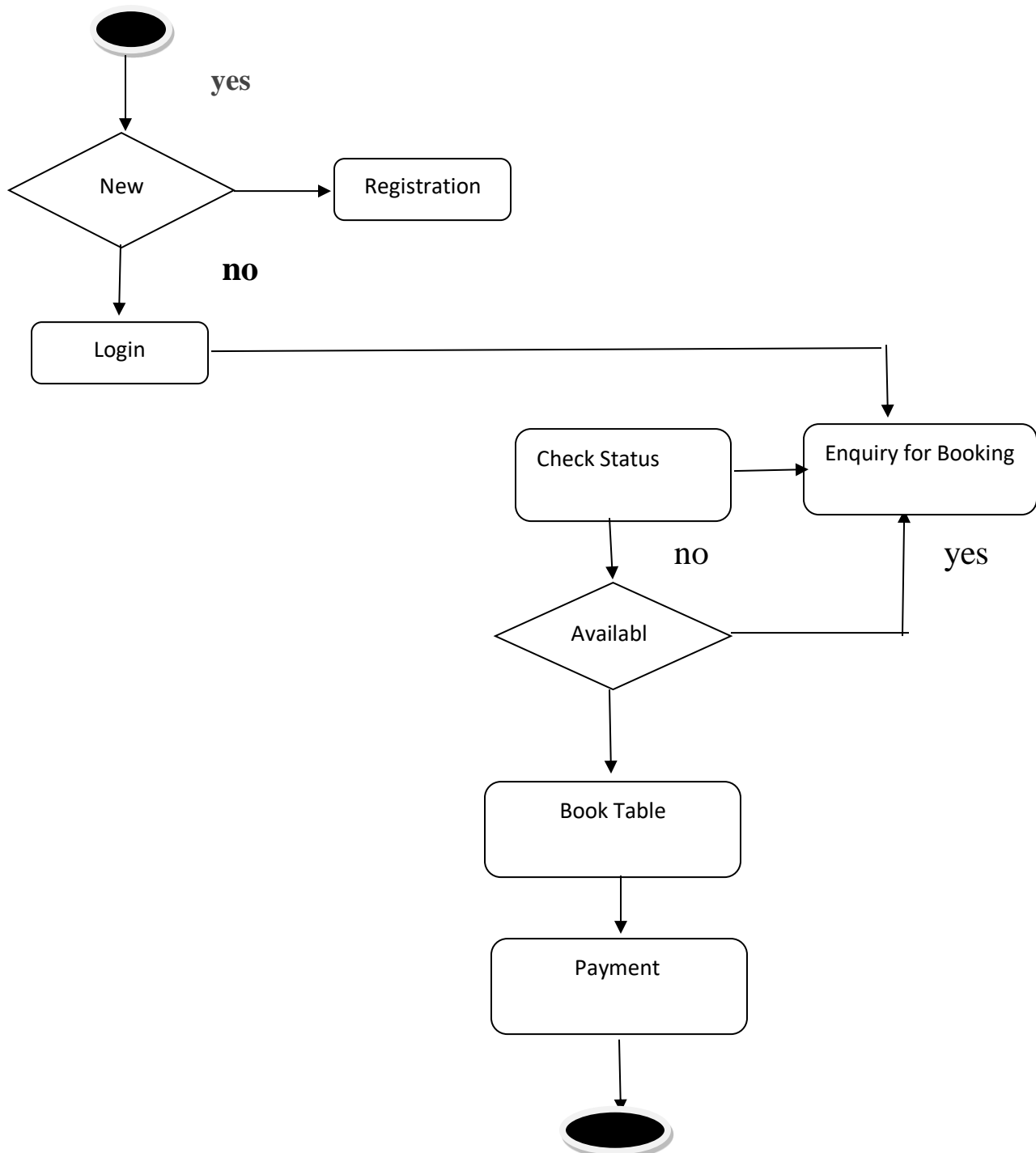
**7.Special Requirements :** None

**8. Associated Use Case(s) :** None

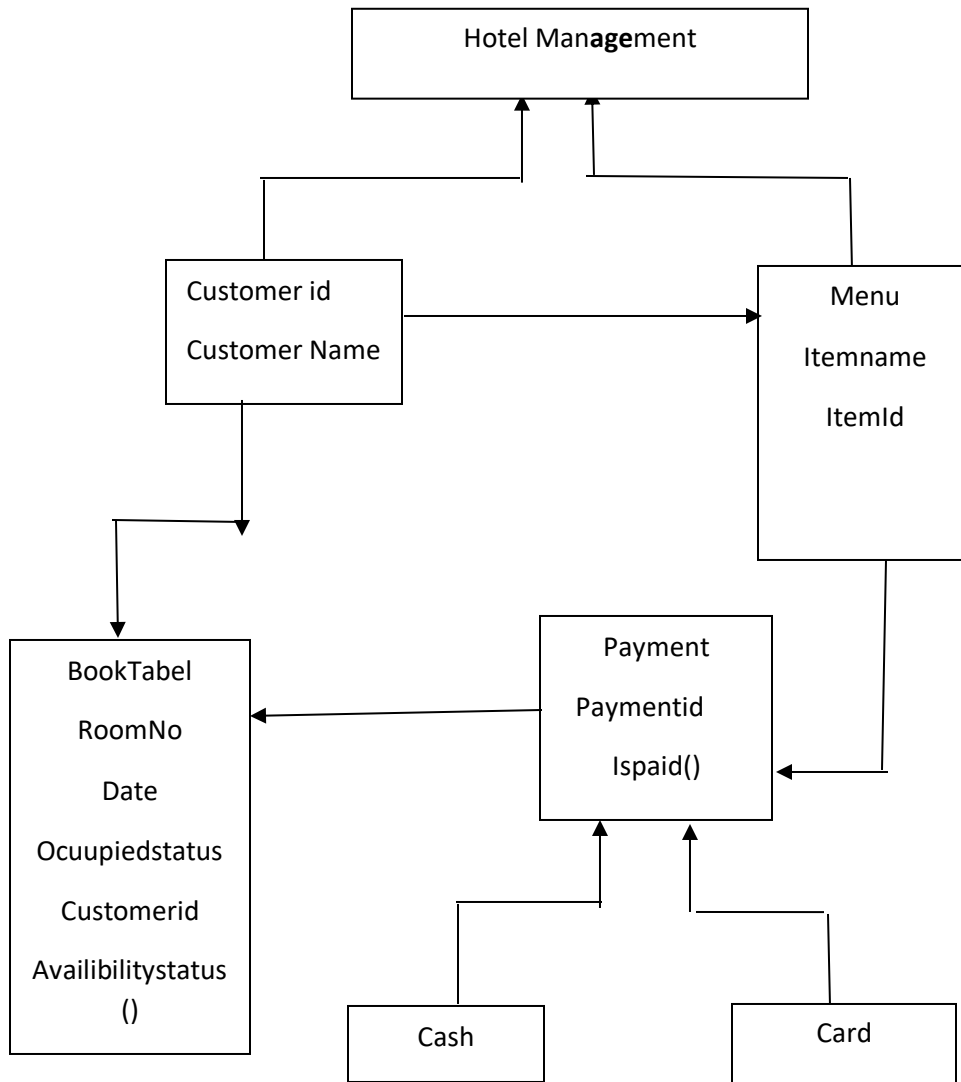
**9.Scenario:**

	Action (stimulus)	Software Reaction
<i>1.</i>	Customer fill out data	<ul style="list-style-type: none"><li>• Payment</li><li>• Card Holder name</li><li>• Card number expiry date</li><li>• CSV</li></ul>

## 2.2 Activity Diagram:



## 2.3 Class Diagram



## **2.4 Er-Diagram of Online Hotel Management System**

### **Hotel:**

Every hotel registered with the system has the information in this entity. Hotel is the main concern of system.

### **Name:**

For customer point of view, every hotel does have its name. The name is very important.

### **No of Rooms:**

Every hotel has many rooms. To make it easy for the customer. And check availability no. of rooms should be in the database.

### **Address:**

This attribute would hold the address of the hotel. Address of Hotel would help the client to find it easy and visit the location. This would provide better services.

### **Book:**

Before going to book any room. The client should check the availability. If the desired room is available. They can book it.

### **Room no:**

With reference to hotel booking must have room no.

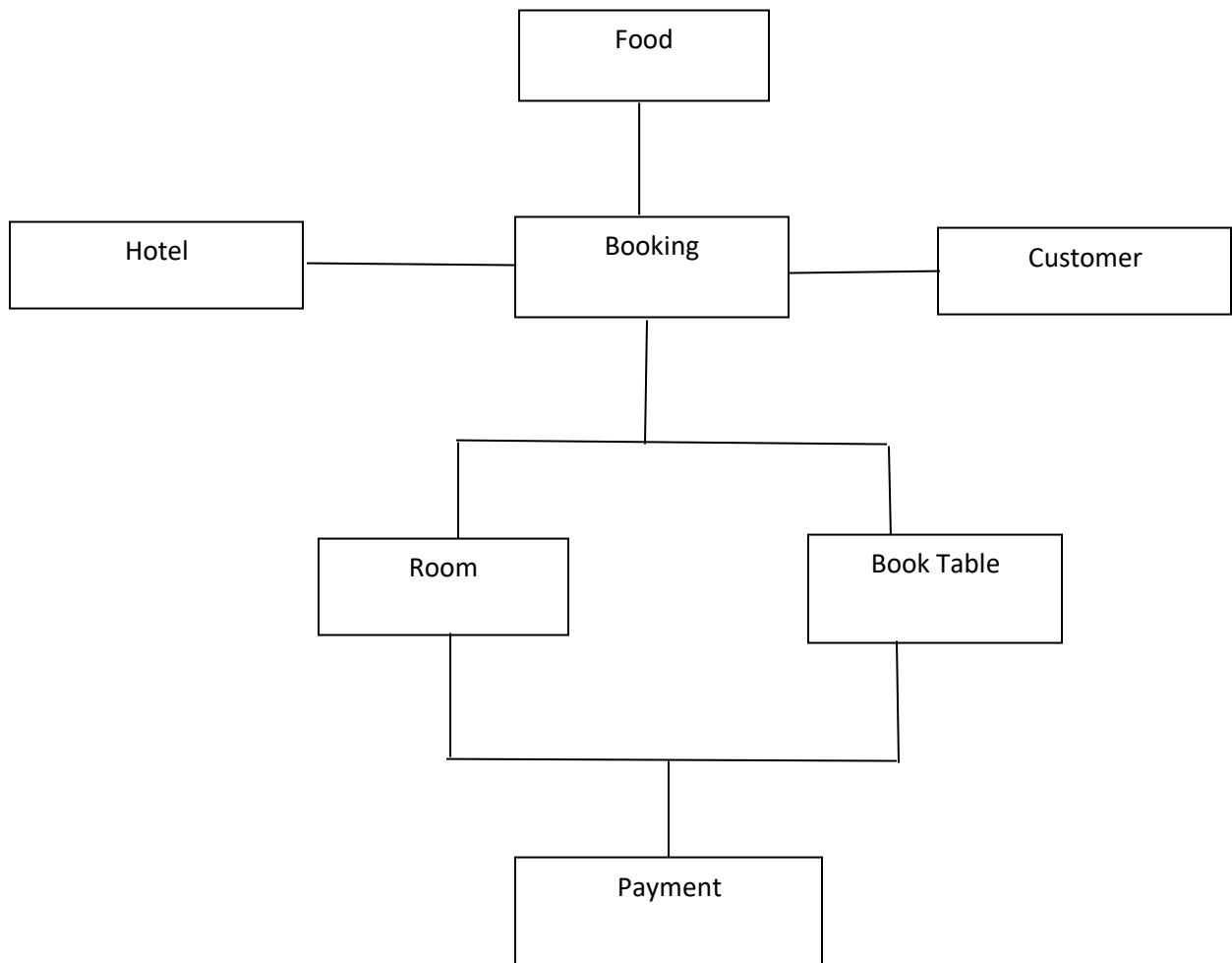
### **Date:**

This would hold the date of the transaction.

### **Payment:**

Payment Entity stores the data about payment. Every payment data is here. Any conflict can be resolved by looking at this.

## 2.5 Data Level Flow Diagram





## 3. System Coding

### 3.1 Implementation :

#### HTML :

- HTML Stands for **Hyper Text Markup Language**
- HTML is the standard markup language for documents designed to be displayed in a web browser.

#### CSS :

- CSS stands for **Cascading Style Sheet**
- CSS is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML.

#### JS :

- JS stands for JavaScript
- JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS.

#### SQL :

- SQL stands for **structured Query Language**
- SQL is a language to define database objects and manipulate the data.

#### PHP :

- PHP stands for **Hypertext Preprocessor**
- PHP is the most widely used open source and general purpose server side scripting language used mainly in web development to create dynamic websites and applications.

## 3.2 Coding

### Login:

```
<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

body { font-family: Arial, Helvetica, sans-serif;}

* {box-sizing: border-box;

}

* {background:url(back.jpg)no-repeat;

background-size:cover;}

}.form-popup {

display: none;

position: fixed;

bottom: 0;

right: 15px;

border: 3px solid #f1f1f1;

z-index: 9;

}

.form-container {

max-width: 300px;

padding: 10px;

background-color: white;

}

.form-container input[type=text], .form-container input[type=password] {

width: 100%;

padding: 15px;

margin: 5px 0 22px 0;

border: none;background: #f1f1f1;

}.form-container input[type=text]:focus, .form-container input[type=password]:focus {
```

```

background-color: #ddd;
outline: none;
}.form-container .btn {
background-color: #0000FF;
color: white;
padding: 16px 20px;
border: none;
cursor: pointer;
width: 100%;
margin-bottom: 10px;
opacity: 0.8;
}.form-container .cancel {
background-color: red;
}.form-container .btn:hover, .open-button:hover {
opacity: 1;}h2 {text-align: center;}
</style></head><body>
<h2>Heritage Hotel</h2><div class="form-popup" id="myForm">
  <form
    action="/action_page.php"
    class="form-container"><h1>Login</h1><label
for="email"><b>Email</b></label><input type="text" placeholder="Enter Email" name="email"
required><label for="psw"><b>Password</b></label><input type="password" placeholder="Enter
Password" name="psw" required></form><form
  action=""
  method="post"><a
href="hotelwebsite.html"><input type="button" value="Login">
</a>
</form></div>
<script>
function openForm() {
  document.getElementById("myForm").style.display = "block";
}
</script>
</body>
</html>

```

## Homepage:

```
<!DOCTYPE html>

<html><head>

<style>

body{ font-family:latha;

    color:white;

    background:url(image.jpg)no-repeat;

    background-size:cover;

    }

.box{

width: 600px;

float:right;

border:1px solid none;}

.box ul li{ width: 100px;

float:left;

margin:10px auto;

left-align: center;}

.box ul li a{

text-decoration:none;color:darkblue;}

.box ul li:hover{background-color: green;}

.box ul li a:hover{ color: white;}

.wd{

width: 220px;

height:539px;

background-color:

    black;

opacity:0.9;padding:55px;}.wd h1 {text-align:center;text-transform:uppercase;font-weight:100px;}.wd

h4{

text-align:justify;
```

```

color:darkgray;
font-weight:normal;}
.wd h2{
    text-align:center;
text-transform:uppercase;
font-weight:normal;
margin:30px auto;}
.opt form, input[type="button"]{background-color: black;
color: white;
padding:7px;
margin:14px
auto;
padding-left:40px;
padding-right:40px;
text-align:center;
}
    form, input[type="button"]:hover{ background-color: green}</style>
</head><body>
<div class="box">
<ul type="none">
<li><a href="">Home</a>
</li>
<li><a href="all menu.html">Menu</a>
</li><li><a href="team.html">Team</a></li>
<li>
<a href="contactform.html">Contact</a></li></ul></div><div
class="wd"><h1>WELCOME!</h1><h4>The Hotel offers an array of services for a comfortable stay.
The fundamental amenities include front desk, air conditioning, internet, elevators, travel desk, non-
smoking rooms, security, and business center. Social gatherings and business events can be well-arranged
with the help of banquet facilities</h4>
<h2>Reservation</h2>

```

```
<div class="opt">
<form action="" method="post">
<a href="Booktable.html">
<input type="button" value="Start now">
</a></div>
</div></body></html>
```

## Booktable:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body
```

```
{
```

```
font-family:'Poppins',sans-serif;
```

```
}
```

```
.banner{
```

```
min-height:100vh;
```

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

```
display:flex;  
flex-direction:column;
```

```
justify-content:center;
```

```
align-items:center;  
color:black;
```

```
padding-bottom:30px;
```

```
}  
</style>
```

```
</head>
```

```
<body>
```

```
<section class="banner">
```

```
<h2>Book your Table Now</h2>
```

```
<div class="class container">
```

```
<div class="image.jpg">
```

</div>

<div class="class container">

<h1>Reservation</h1>

<form>

<div class="form.row">

<select name="days">

<option value="day.select">Select Day</option>

<option value="sunday">Sunday</option>

<option value="sunday">Monday</option>

<option value="sunday">Tuesday</option>

<option value="sunday">Wednesday</option>

<option value="sunday">Thursday</option>

<option value="sunday">Friday</option>

<option value="sunday">Saturday</option>

</select>



```
<select name="hours">
  <option value="hours.select">
    Select Hours</option>
  <option value="10">10:00</option>

  <option value="10">12:00</option>
  <option value="10">14:00</option>

  <option value="10">16:00</option>

  <option value="10">20:00</option>

  <option value="10">22:00</option>
</select>

</div>
```

```
<div class="form-row">
  <input type="text"
    placeholder="FullName">
```

```
  <input type="text"
    placeholder="PhoneNumber">
</div>
```

```
<div class="form-row">
```

```
<input type="number"
placeholder="How Many Person?" min="1">
```

```
<form action="" method="post"> <a href="Paymentpage.html"> <input type="button" value="BOOK
TABLE"> </a>
```

```
</div>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
</section>
```

```
</body></body></html>
```

## **Payment:**

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Page Title</title>
```

```
<style>
```

```
*{  
  
margin:0;  
padding:0;  
box-sizing: border-box;  
font-family: 'Ubuntu', sans-serif;  
}
```

```
body{  
  
background: #2196f3;  
margin: 0 10px;  
  
}
```

```
.payment{  
background: #f8f8f8;  
max-width: 360px;  
margin: 80px auto;  
height auto;  
padding: 35px;  
padding-top:70px;  
border-radius: 5px;  
position: relative;  
}
```

```
.payment h2{  
text-align:center;
```

```
Letter-spacing: 2px;  
margin-bottom: 40px;  
color:black;  
  
}
```

```
.form .label{  
display: block;  
color: #555555;  
margin-bottom: 6px;  
  
}
```

```
.form .input{  
  
padding: 13px 0 13px 25px;  
width: 100%;  
text-align: center;  
border: 2px solid #dddddd;  
border-radius: 5px;  
Letter-spacing: 1px;  
word-wrap: 3px;  
  
}
```

```
.card-grp{  
display: flex;
```

```
justify-content: space-between;  
}
```

```
.card-item{  
width: 48%;  
  
}
```

```
.space{  
margin-bottom: 20px;  
}
```

```
.btn{  
margin-top: 40px;  
background: #2196f3;  
padding: 12px;  
text-align: center;  
color: #f8f8f8;  
border-radius: 5px;  
  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="wrapper">
```

```
<div class="payment">
```

```
<h2>Payment</h2>
```

```
<div class="form">
```

```
<div class="card space">
```

```
<label class="label">Card holder:</label>
```

```
<input type="text" class="input" name="card_holder" placeholder="Card Holder Name">
```

```
</div>
```

```
<div class="card space">
```

```
<label class="label">Card Number:</label>
```

```
<input type="text" class="input" name="card_number" placeholder="Card Number">
```

```
</div>
```

```
<div class="card-grp space">
```

```
<div class="card-item">
```

```
<label class="label">Expiry Date:</label>
```

```
<input type="text" class="input" name="expiry_date" placeholder="00 / 00">
```

```
</div>
```

```
<div class="card-item">
```

```
<label class="label">CVC:</label>
```

```
<input type="text" class="input" name="cvc" placeholder="000">
```

```
</div>
```

```
</div>
```

```
<div class="btn">
```

```
Pay
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

### 3.3 Database Connectivity

```
$conn=mysqli_connect($server_name,$username,$password,$database_name);
```

```
//now check the connection
```

```
if(!$conn)
```

```
{
```

```
    die("Connection Failed:" . mysqli_connect_error());
```

```
}
```

```
if(isset($_POST['save']))
```

```
{
```

```
    $first_name = $_POST['first_name'];
```

```
    $last_name = $_POST['last_name'];
```

```
$gender = $_POST['gender'];

$email = $_POST['email'];

$phone = $_POST['phone'];


$sql_query = "INSERT INTO entry_details
(first_name,last_name,gender,email,mobile)

VALUES ('$first_name','$last_name','$gender','$email','$phone')";

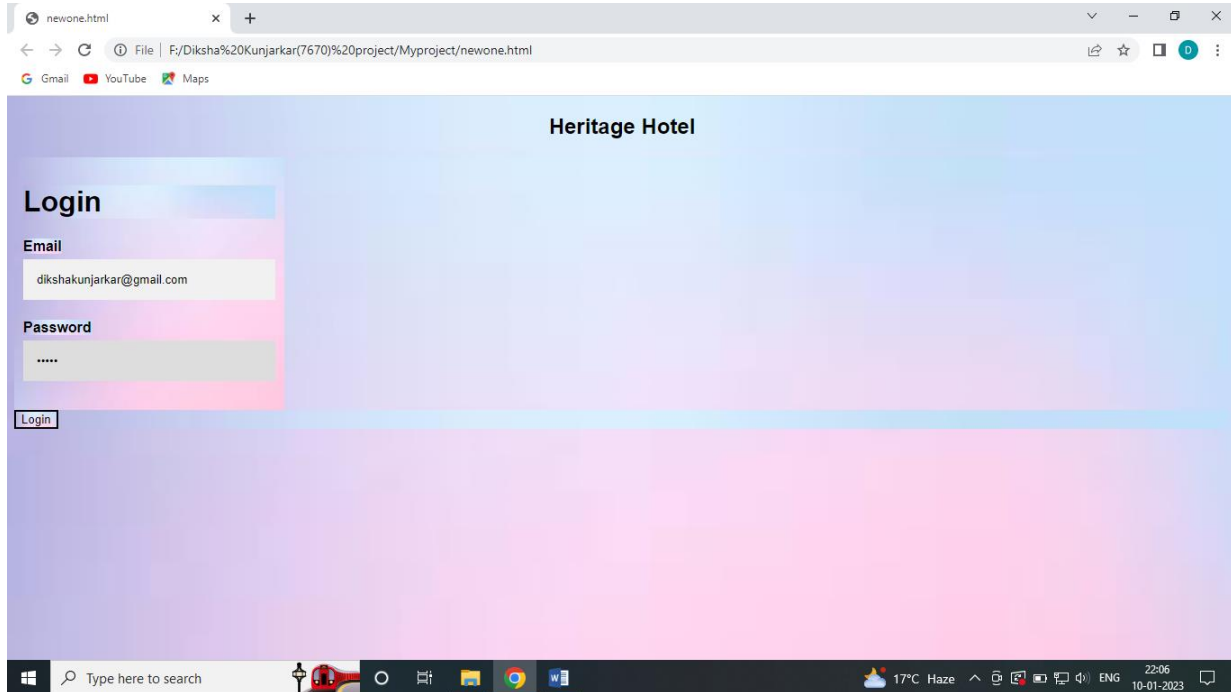

if (mysqli_query($conn, $sql_query))
{
    echo "New Details Entry inserted successfully !";
}
else
{
    echo "Error: " . $sql . " " . mysqli_error($conn);
}

mysqli_close($conn);
}

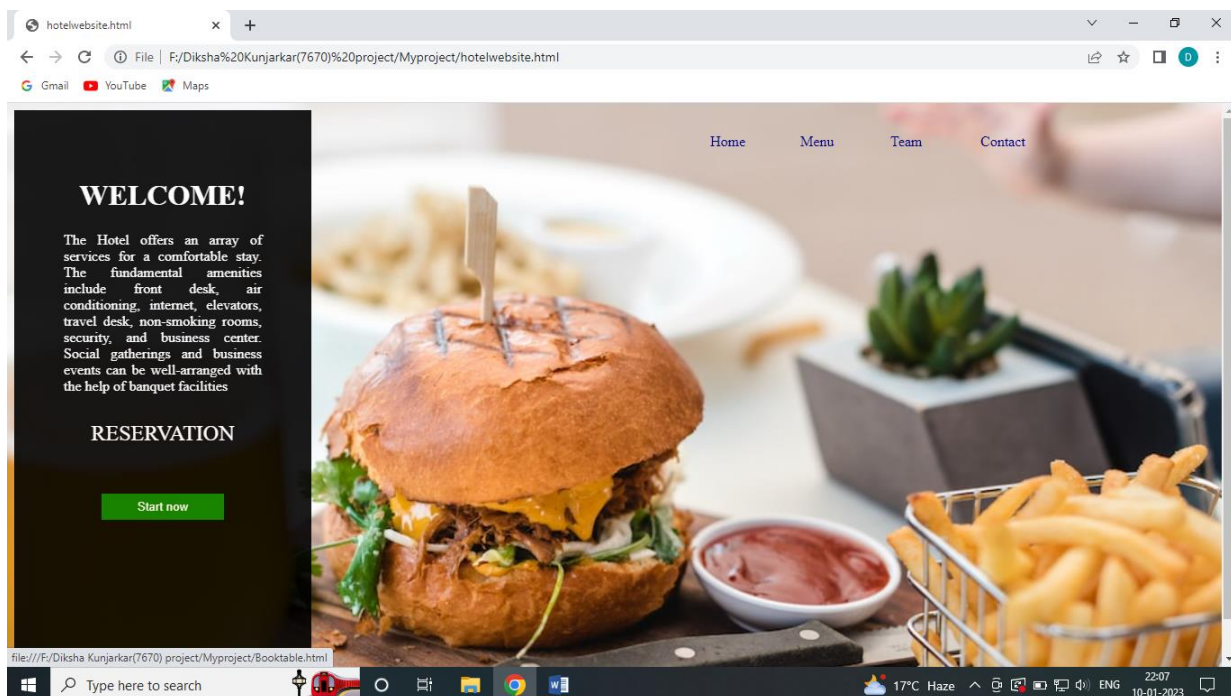
?>
```



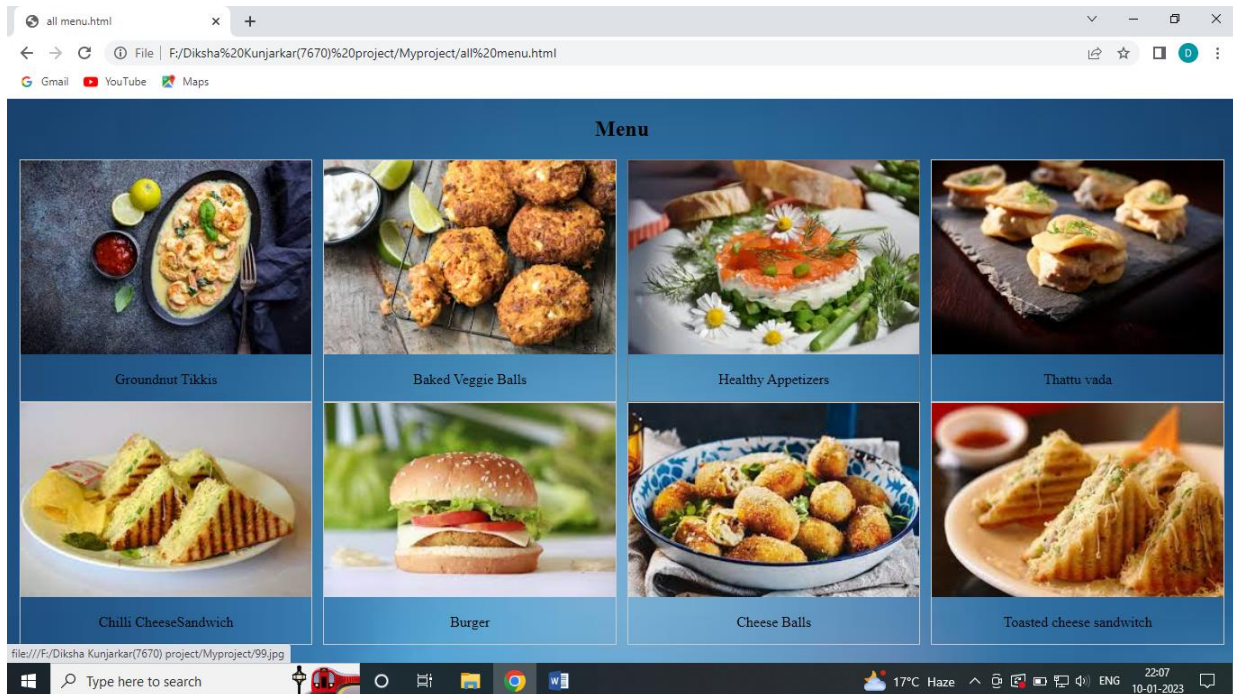
### 3.4 Screenshots: Login page



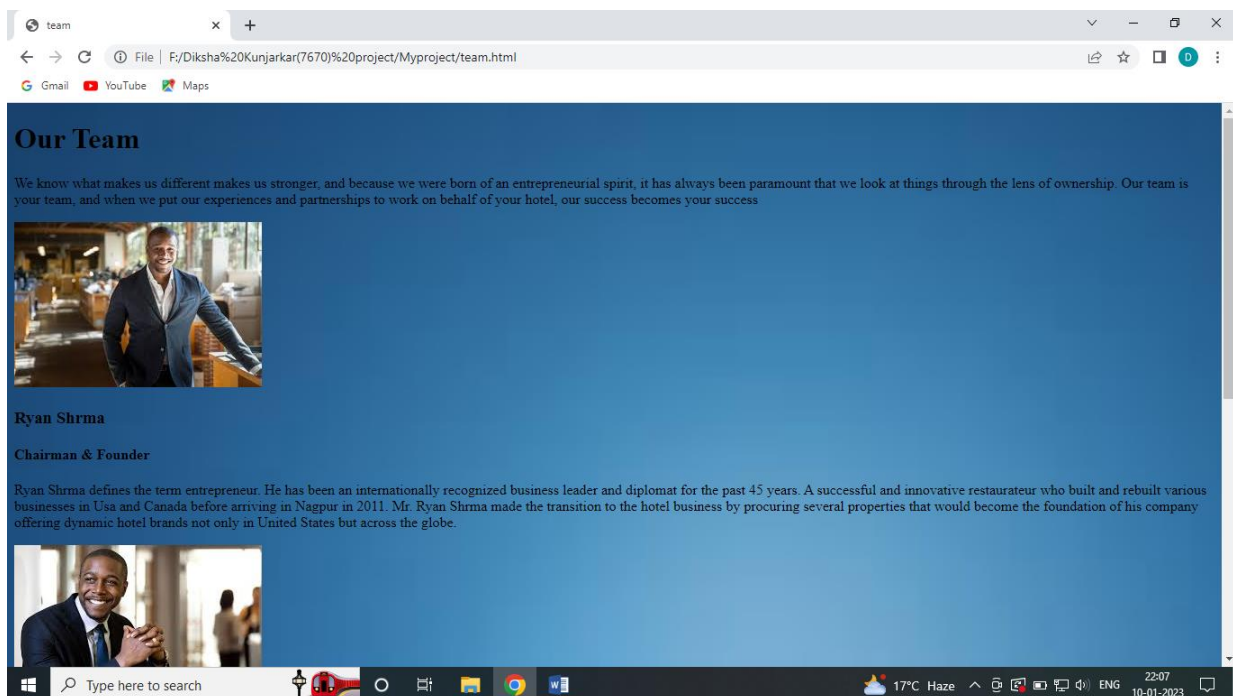
### Homepage:



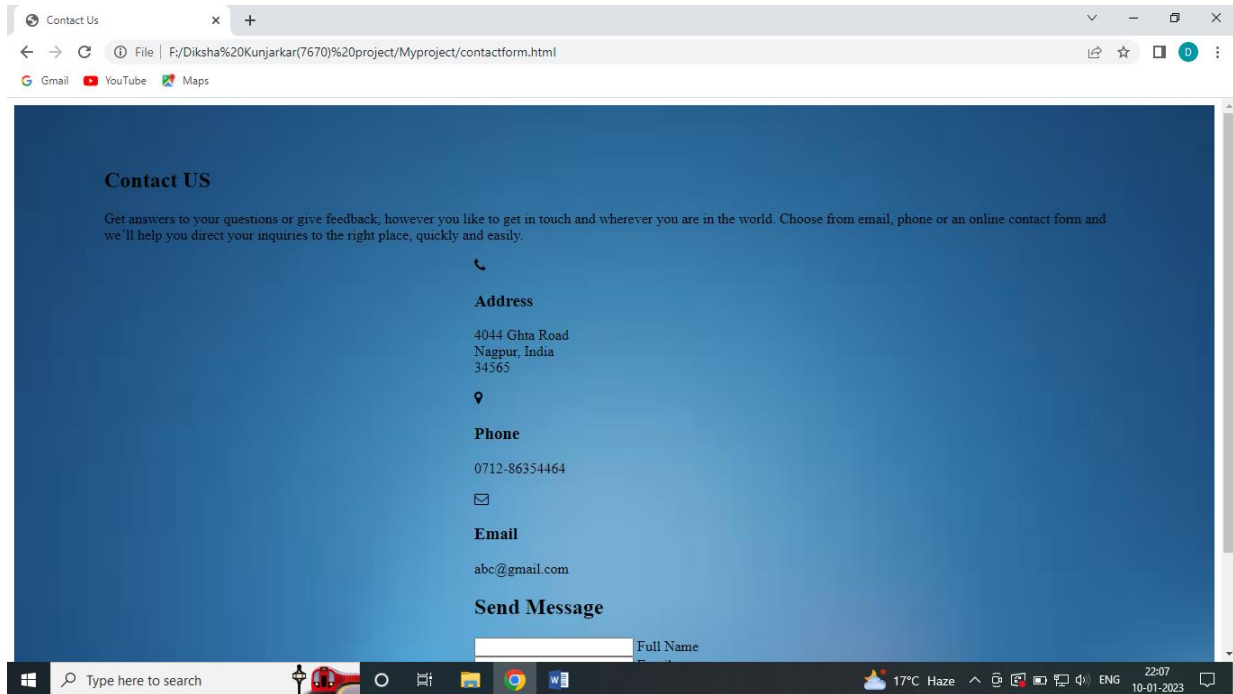
## Menu:



## Team:



## Contact us:



Contact Us

Get answers to your questions or give feedback, however you like to get in touch and wherever you are in the world. Choose from email, phone or an online contact form and we'll help you direct your inquiries to the right place, quickly and easily.

**Address**

4044 Ghta Road  
Nagpur, India  
34565

**Phone**

0712-86354464

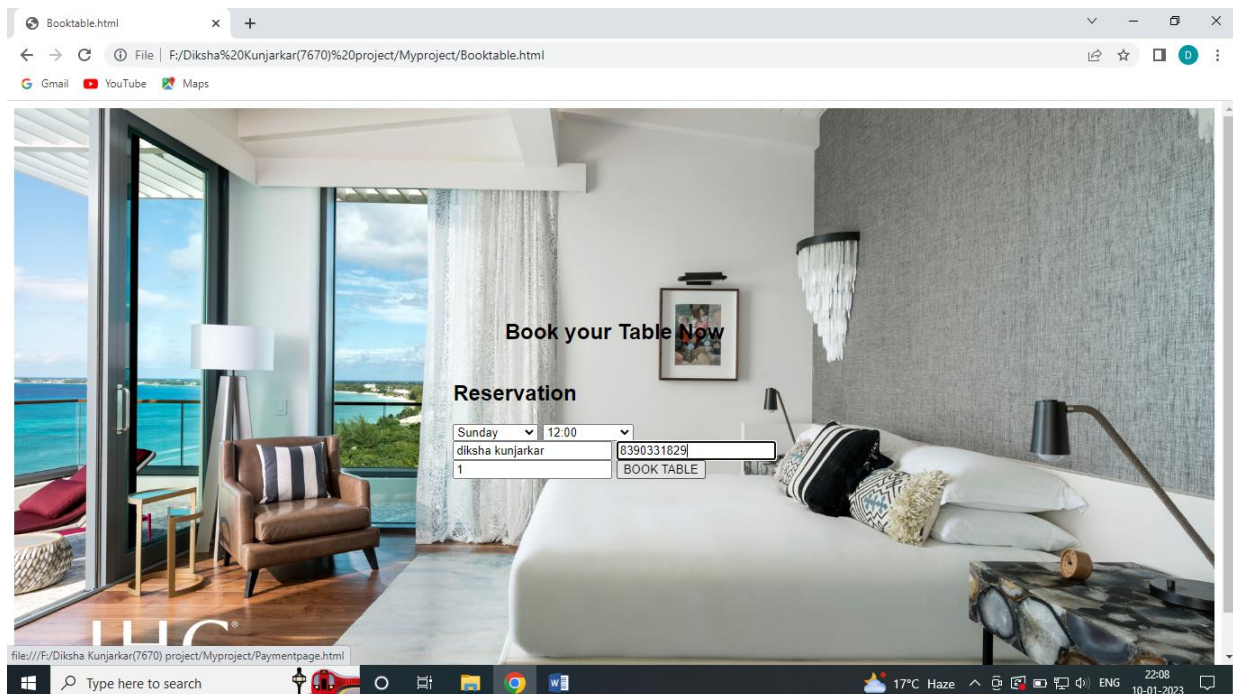
**Email**

abc@gmail.com

**Send Message**

Full Name

## Book table:



Book your Table Now

**Reservation**

Sunday 12:00

diksha kunjarkar 8390331829

1 BOOK TABLE

## Payment:

Payment

Card holder:  
diksha kujarkar

Card Number:  
636-378-33333

Expiry Date: 12/26 CVC: 3098

Pay

## Data base Connectivity

Server: 127.0.0.1 » Database: database123 » Table: entry\_details

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 4 (5 total. Query took 0.0004 seconds)

SELECT \* FROM `entry\_details`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table

first_name	last_name	gender	email	mobile
Kiyaan	Kokane	male	kiyaan@gmail.com	1111111111
teju	kokane	female	teju@gmail.com	2222222222
teju	kokane	female	teju@gmail.com	2222222222
Diksha	Kunjarkar	female	Diksha@gmail.com	8390331821

Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations



# SYSTEM TESTING

## 4.1 Types Of Testing :

Testing is the process of detecting errors. Testing performs a very critical role for quality

Testing Objectives: The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Testing is a process of executing a program with the intent of finding an error. A successful test is one that uncovers an as yet undiscovered error. A good test case is one that has a high probability of finding error, if it exists. The tests are inadequate to detect possibly present errors. The software more or less confirms to the quality and reliable standards.

### Levels Of Testing :

1. Unit/Component Testing : Checking the separate level of testing.
2. Integration Testing : Checking the top – down & down – top.
3. System Testing : Checking the whole product.
4. Acceptance Testing : Checking the product is working in the right way or not.

There are two ways of Acceptance Testing:

1. Alpha testing Organization.
2. Beta testing Customer place.

### Testing Concept:

1. Black Box Testing : Checking the External behaviour of the software.
2. White Box Testing : Checking the Internal code of the software.
3. Gray Box Testing : Combination of both white box testing & black box testing.

### Functional Testings :

1. Assertion testing – It allows testing the correctness of any assumptions that have been made in the program.
2. Gorilla testing – It is a type of software testing which is performed on a module based on some random inputs repeatedly and checks the module's functionalities and confirms no bugs in that module.

3. Sanity testing – It is done to check the bugs have been fixed after the build.
4. Monkey testing – It is a type of software testing in which the tester tests the application or software by providing some random inputs and checking the behavior of the application or the software.
5. Smoke testing – It is a type of testing which is done to assure that the acute functionalities of the program is working fine.
6. Exploratory testing - It is a type of software testing in which the tester is free to select any possible methodology to test the software.
7. Mutation testing – It is a type of Software Testing that is performed to design new software tests and also evaluate the quality of already existing software tests
8. Benchmark testing – It is performed against a system to determine current performance and can be used to improve application performance.

### **Non-Functional Testings:**

1. Load testing – It is used to perform the maximum quantity of software applications without important performance breakdown.
2. Stress testing – It is a type of software testing that verifies the stability and reliability of the system.
3. Usability testing – It is a type of testing, that is done from an end user's perspective to determine if the system is easily usable.
4. – It is a type of testing, that is done from an end user's perspective to determine if the system is easily usable
5. Volume testing – It is a type of software testing which is carried out to test a software application with a certain amount of data.
6. Scalability testing - To determine the user limit for the web application and ensure end user experience, under a high load, is not compromised
7. Security testing – It is a type of Software Testing that uncovers vulnerabilities of the system and determines that the data and resources of the system are protected from possible intruders.

## 4.2 Testing Methodologies:

Some well-accepted testing methodologies comprise –

### **1. Waterfall model :**

It is a linear, sequential approach to the software development lifecycle (SDLC) that is popular in software engineering and product development.

### **2. Agile methodology :**

It aligns with iterative development methodology in which requirements develop gradually from customers and testing teams.

### **3. Iterative model :**

A subset of the final product under development, which grows from iteration to iteration to become the final product or software.

### **4. V-Model :**

It is an SDLC model where execution of processes happens in a sequential manner in a V-shape. It is also known as Verification and Validation model.

### **5. Spiral Model :**

It is a systems development lifecycle (SDLC) method used for risk management that combines the iterative development process model with elements of the Waterfall model.

### **6. Extreme Programming Model :**

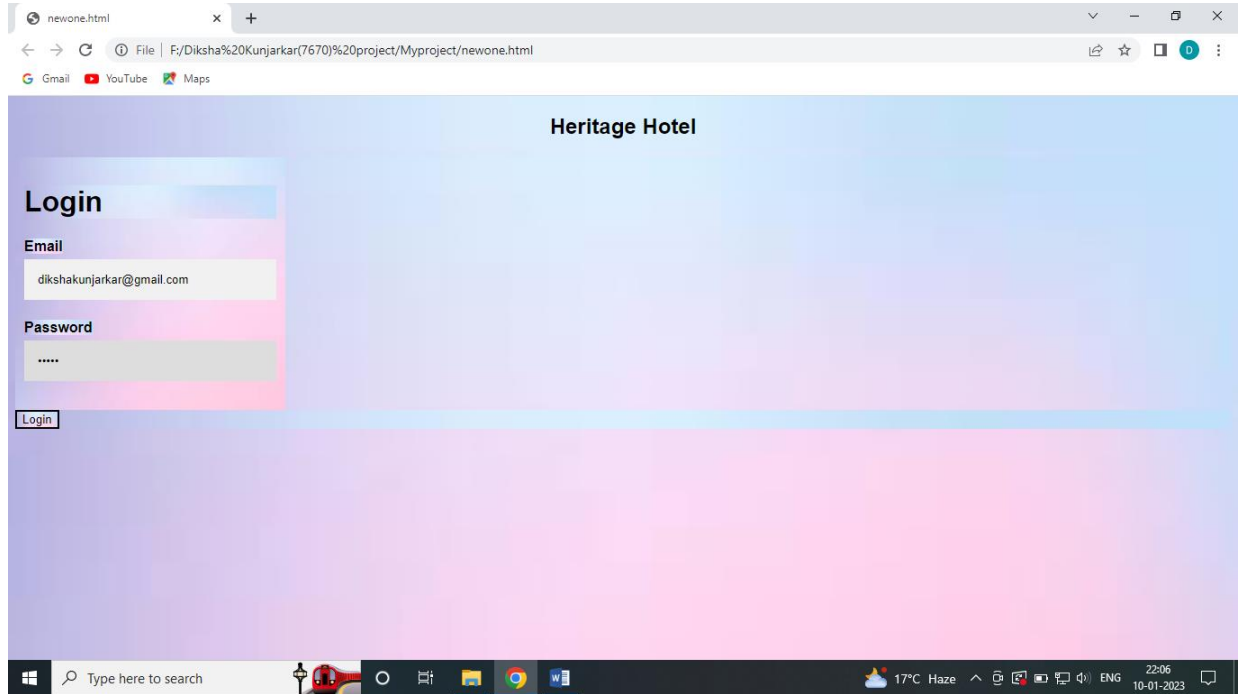
It is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team.

### **7. RAD (Rapid Action Development) Model :**

It is a software development process based on prototyping without any specific planning. In model, there is less attention paid to the planning and more priority is given to the development tasks. It targets at developing software in a short span of time.

## 4.3 Equivalence and Boundary Value

Let's create a decision table for a login screen



The condition is simple if the user provides the correct username and password the user will be redirected to the homepage.

Conditions	Rule 1	Rule 2	Rule 3	Rule 4
Username (T/F)	F	T	F	T
Password (T/F)	F	F	T	T
Output (E/H)	E	E	E	H

While converting this to a test case, we can create 2 scenarios,

Enter the correct username and correct password and click on login, and the expected result will be the user should be navigated to the homepage.



#### 4.4 Test Reports:

##### Test case Report :

Project name	Hotel Management System
Reference	D:\Users\project\project.html
Created By	Diksha kunajrkar
On Date	12-01-2023
Review Date	

##### Test Scenario : 01 Login

Test Case ID	Test Case name	Steps	Pre Condition	Input	Actual	Expected	Post Condition	Result
TC – 001	Login Valid	1.Internet Connectivity	1.Enter valid Username 2.Enter valid Password 3.Click on submit button	Username – diksha Password - *****	User should able to see Home page	As Expected	Home page displayed	Pass
TC – 002	Login Invalid	2.User must have account	1.Enter Invalid Username 2.Enter Invalid Password 3.Click on submit button	Username – diksha Password - *****	User should not able to see Home page	As Expected	Home page not displayed	Fail

**Test Scenario : 02 Book Table**

Test Case ID	Test Case name	Steps	Pre Condition	Input	Actual	Expected	Post Condition	Result
TC – 003	Checking Valid detail	1.Internet Connectivity	1.User select date, time 2. User select name, phoneno ,person 3. Click on book button	Check valid detail	User should able to add detail	As Expected	Book page displayed	Pass
TC – 004	Checking Invalid detail	2.User must have account 3.Photo should be open	1.User select date, time 2. User select name, phoneno ,person 3. Click on book button	Check Invalid detail	User should not able to add detail	As Expected	Book page not displayed	Fail

### Test Scenario : 03 Payment

Test Case ID	Test Case name	Steps	Pre Condition	Input	Actual	Expected	Post Condition	Result
TC – 006	Checking Valid detail	1.Internet Connectivity 2.User must have account 3.User details should properly check	User details able to add properly	Check the valid details	N number of details should added	As Expected	Details page displayed	Pass
TC – 007	Checking Invalid detail		User details not able to add properly	Check the Invalid details	Number of details should not added	As Expected	Details page not displayed	Fail

## 4.5 Done Automation Selenium of Amazon:

```
package TestExercise;
import java.util.Iterator;
import java.util.Set;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Then;

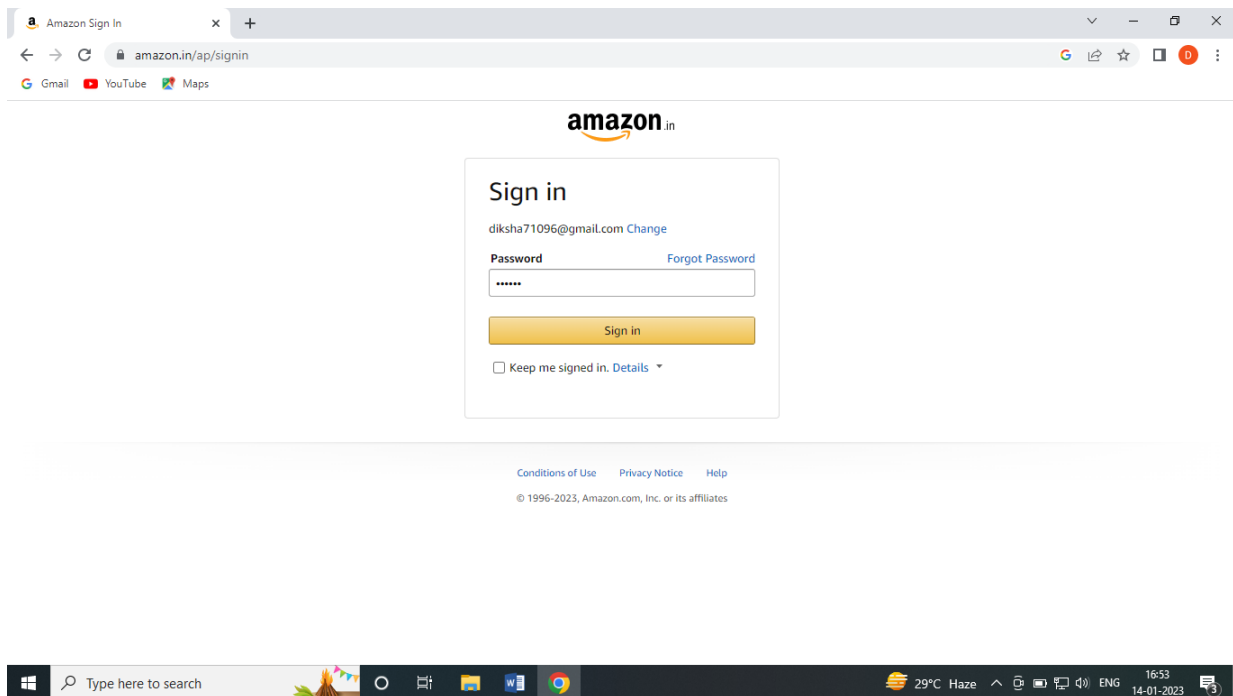
public class SeleniumMiniProject{
    public static void main(String[] args) throws InterruptedException{
        System.setProperty("Webdriver.chrome.driver","C:\\\\SeleniumWebDriver\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        //Launch Amazon
        browser
        driver.get("http://www.amazon.in/");String
        Exception1 =
        driver.getTitle();
        System.out.println(driver.getTitle());
        //Login to Amazon page
        driver.findElement(By.id("nav-link-accountList")).click();
        driver.findElement(By.id("ap_email")).sendKeys("dharinisri99@gmail.com"); driver.findElement(By.className("a-button-input")).click();
        driver.findElement(By.id("ap_password")).sendKeys("srilatha");
        driver.findElement(By.id("signInSubmit")).click();
        System.out.println("Successfully Login");
        //Search bar
        driver.findElement(By.id("twotabsearchtextbox")).sendKeys("neck band");driver.findElement(By.id("nav-search-submit-button")).click(); System.out.println("Done the searchbar");
        driver.findElement(By.linkText("boAt Rockerz 255 Pro+ in-Ear Bluetooth Neckband with Upto40 Hours Playback, ASAP™Charge, IPX7, Dual Pairing, BT v5.0, with Mic (Active Black)")).click();
        //Add cart
        Set<String> ids =
        driver.getWindowHandles();
```

```

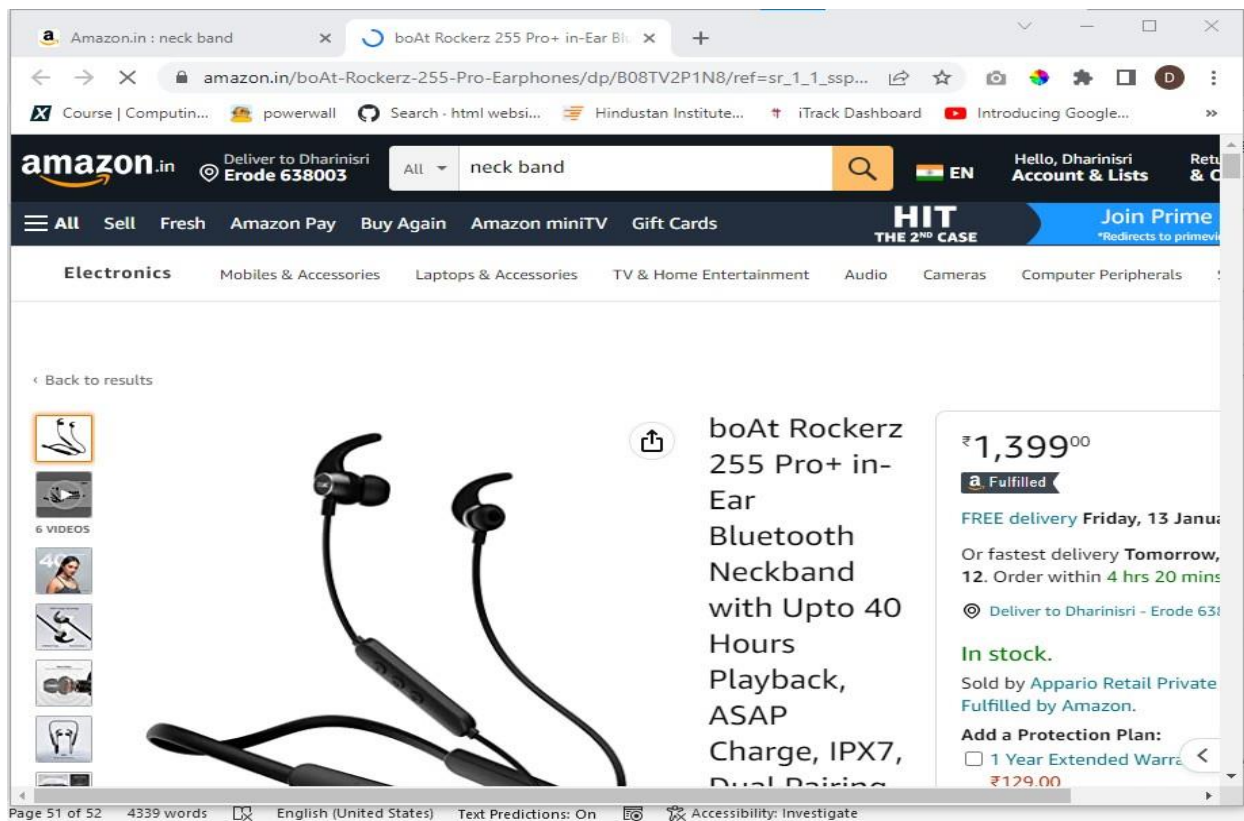
Iterator<String> it = ids.iterator();
String
parentId =
it.next();
String childId
= it.next();
driver.switchTo().window(childId);
driver.findElement(By.id("add-to-cart-
button")).click();
driver.findElement(By.xpath("/html/body/div[2]/header/div/div[1]/div[3]/div/a[5]/div[1]/span[1]
.click();
System.out.println("successfully
Add the cart");Thread.sleep(5000);
driver.quit();} }

```

**Fig:1.1 Login page**



**Fig : 1.2 Search bar**



**Fig : 1.3 Added cart**

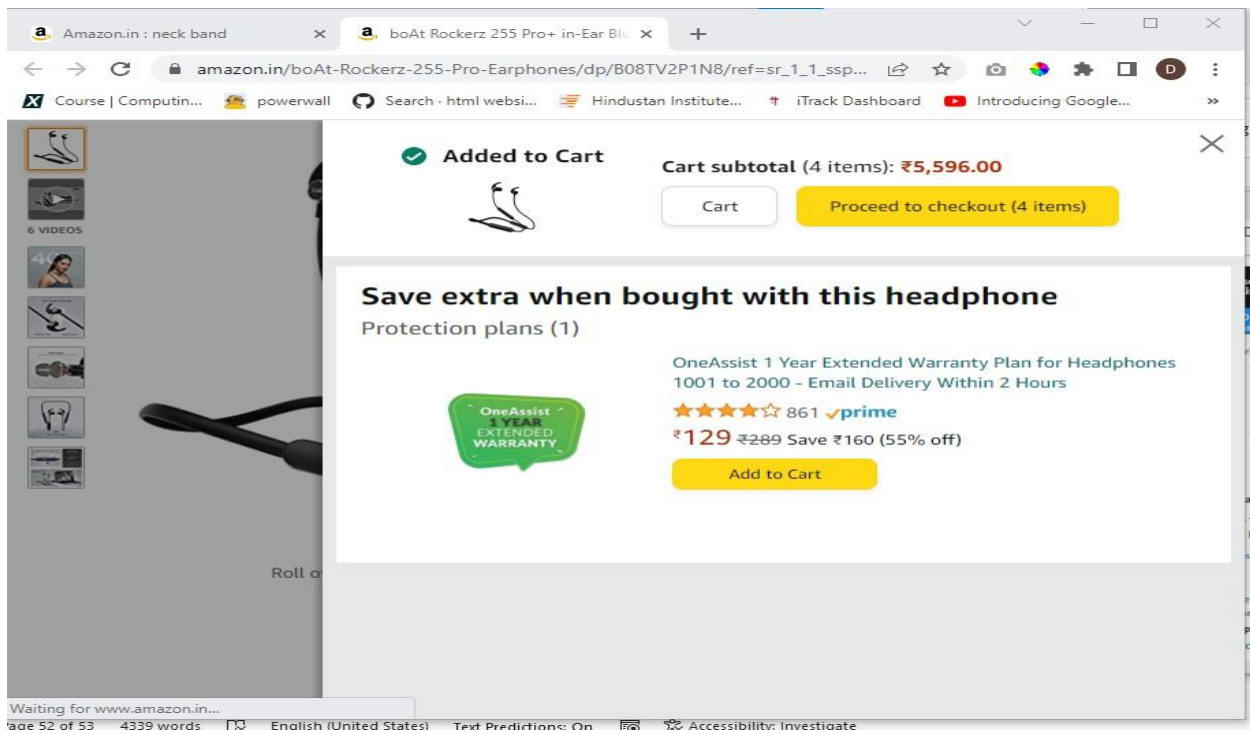
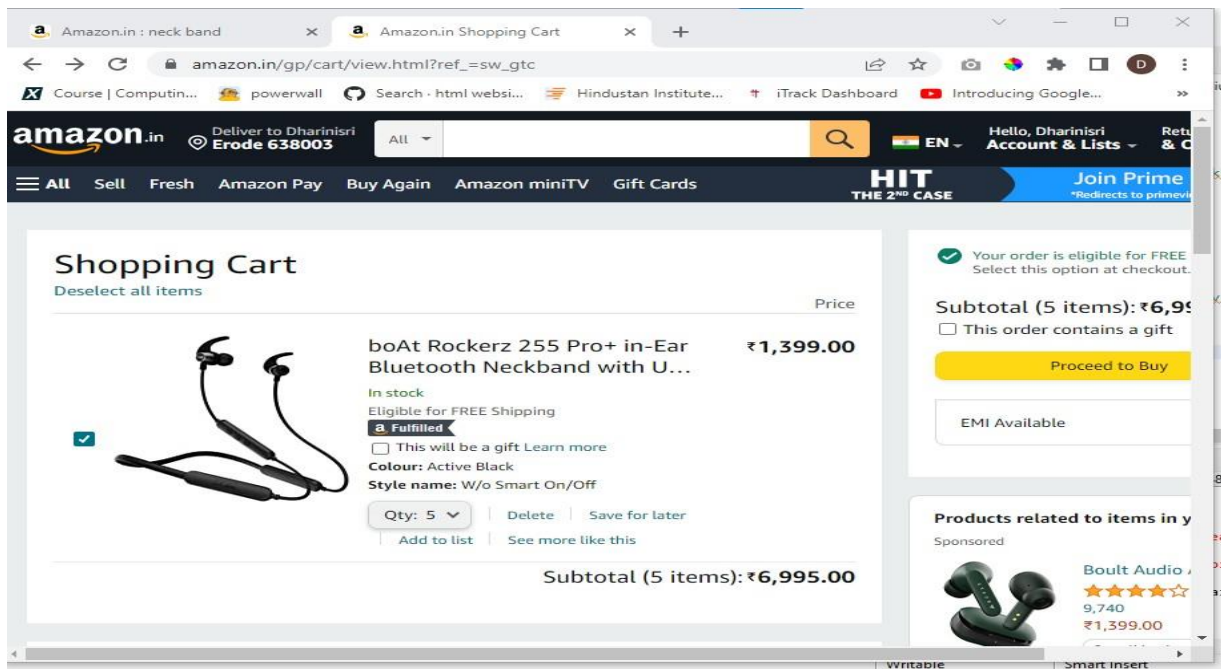


Fig : 1.4 Go to cart page



## 4.6 Done the Automation testing of Cucumber :

```
package StepsUser;

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 LoginUser {
    @Given("user is on login page")
    public void user_is_on_login_page() {
        System.out.println("Inside Steps - user is on login page");
    }

    @When("user enters username and password")
    public void user_enters_username_and_password() {
        System.out.println("Inside Steps - user enters username and password");
    }

    @And("clicks on login button")
    public void clicks_on_login_button() {
        System.out.println("Inside Steps - 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 Steps - user is navigated to the home page");
    }
}
```



## Features of Cucumber :

Feature: features to Amazon

functionality Scenario: Check

Amazon with selenium credentials

Given user is on visit page

When user enters

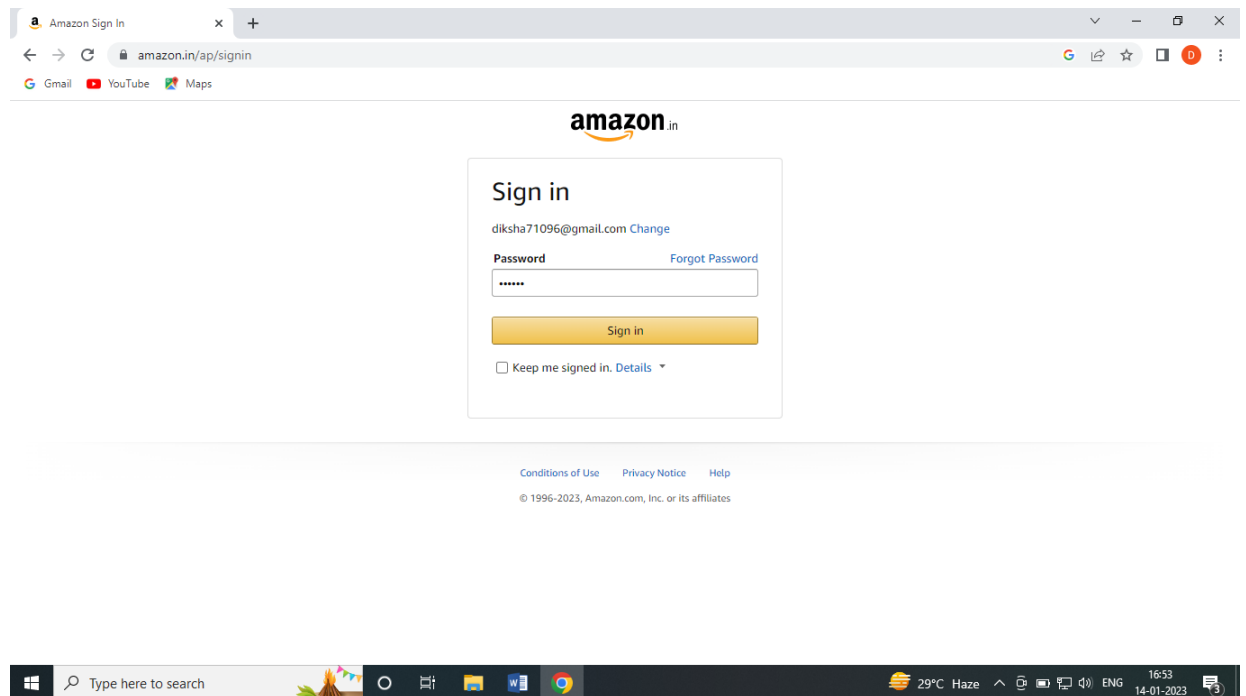
correct valuesAnd

clicks on button

Then user is directed to the homepage

## Screenshots of Cucumber :

**Fig : 1.1 Login Page**



## **4.7 CONCLUSION**

After successfully completing the major project, a simple conclusion can be drawn that a “Hotel Management System” is a well-organized management system. The proposed system magnificently maintains the records of this organization. In this report; the issues related to either customer’s or hotel manager’s discomfort, while booking using a manual reservation system by creating an online reservation system for the customers to have a reservation according to their welfare was addressed ably. The debate has also been done on the former studies of an online booking system and the combination of the internet connection by the hotels enhancing their ability to connect better with their users.

Discussion on the different types of methods has been done such as; the collection of the needed requirements, and the type of developed technique was selected for the particular research. A very detailed description of the website pages was discussed and the types of examination used to examine the website.