- **TITLE :** Mini project on “Automation Testing on Online Restaurant Feedback Website:”
- **Problem Statement :** This Projects test plan consist of features to be tested and bug taxonomy. Narrate scripts in order to perform regression tests. Identify the bugs using Selenium WebDriver and IDE and generate test reports encompassing exploratory testing
- **Prepared By :**

  Pratik Chavan (BC113)
  Ajay Dangade (BC117)
  Prathamesh Dubbewar (BC122)

- **Theory :**

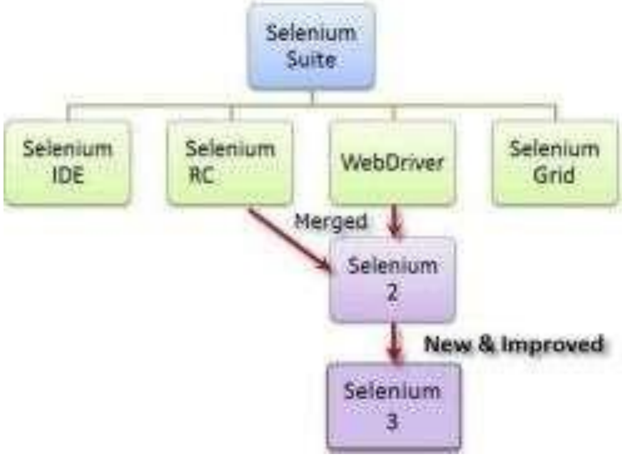
Application:

The restaurant feedback website prioritizes user-friendliness, offering visitors an effortless experience. With a comprehensive list of restaurants, users can explore and provide feedback with ease. The intuitive interface streamlines the process, making it convenient for all visitors. This user-centric approach enhances engagement and ensures a rich database of restaurant reviews, ultimately benefiting both diners and restaurant owners.

Testing Tool: Selenium

Selenium is a free (open-source) automated testing suite for web applications across different browsers and platforms. It is quite similar to HP Quick Test Pro (QTP now UFT) only that Selenium focuses on automating web-based applications. Testing done using the Selenium tool is usually referred as Selenium Testing. Selenium is not just a single tool but a suite of software, each catering to different testing needs of an organization. It has four components.

- Selenium Integrated Development Environment (IDE)
- Selenium Remote Control (RC)
- Web Driver
- Selenium Grid



Selenium IDE (Integrated Development Environment) is the simplest tool in the Selenium Suite. It is a Firefox add-on that creates tests very quickly through its record- and-playback functionality. This feature is similar to that of QTP. It is effortless to install and easy to learn. Because of its simplicity, Selenium IDE should only be used as a prototyping tool, not an overall solution for developing and maintaining complex test suites.

Though you will be able to use Selenium IDE without prior knowledge in programming, you should at least be familiar with HTML, JavaScript, and the DOM (Document Object Model) to utilize this tool to its full potential. Knowledge of JavaScript will be required when we get to the section about the Selenese command "run Script."

Selenium IDE supports auto-complete mode when creating tests. This feature serves two purposes: It helps the tester to enter commands more quickly.

It restricts the user from entering invalid commands.

The way to set up a Selenium 2.0 Java project is to use Maven. Maven will download the java bindings (the Selenium 2.0 java client library) and all its dependencies, and will create the project for you, using a maven pom.xml (project configuration) file. Once done, you can import the maven project into your preferred IDE, IntelliJ IDEA or Eclipse.

Test Plan consisting of features to be tested and bug taxonomy.

Test ID:ID1

Level: Basic Level Testing

Test: Unit Testing, Integration Testing

Version: Not Revised

Test Plan ID1 involves basic level testing of a restaurant feedback website's user-friendliness and user experience. This encompasses both unit and integration testing, ensuring intuitive navigation, feedback submission, and a user-centric interface. Additionally, the test will verify the comprehensiveness of the restaurant review database. Common bug categories to consider include navigation issues, feedback submission problems, interface glitches, and database inaccuracies. A thorough evaluation will be carried out through test cases covering various scenarios, helping identify and rectify potential issues.

References

This test plan is developed on some supporting documents. Refer to the actual version of the document. Here is the list for those reference documents.

- ➢ Project Plan
- ➢ Software Requirements Specification.
- ➢ Detail design document.
- ➢ Test plan template IEEE.
- ➢ International Standard Testing.

For this test plan, we followed IEEE format and the Software Requirements Specification document.

Introduction

Introduction this document is intended to give a complete planning of a systematic strategy for software testing of Online restaurant Feedback Website. This test plan is actually designed to ensure those features work up to the mark. Both directly and indirectly affected elements will be addressed here. In the following paragraphs we will discuss the purpose of the plan, its scope and define the acronyms used in this document.

➢ Objectives

This document supports the following objectives:

- ● Identify existing project information.
- ● Identify the approach that should be followed.
- ● Identify the features that should be tested.
- ● List the recommended test requirements.
- ● Recommend and describe the testing strategies to be employed.
- ● Identify the required resources and provide an estimation of the test efforts.
- ● Fix the schedule of intended testing activities.
- ● Identify the risks associated with the test strategy.
- ● List the deliverable elements of the test activities.

➢ Scope

Testing will begin at the component level and work toward the unit of the entire system. This document will mainly provide the blueprint of high-level testing approaches of Online restaurant Feedback Website. It will validate major system functions of the Online restaurant Feedback Website against the user requirements.

Test Items (Functions)

Test Items In this section we will provide a list of all those components that has been identified as test items. It is assumed that unit testing will be done thorough testing of all module interfaces will be ensured. We will perform testing on use case specification, functional requirement and non-functional requirement. The interfaces between the following subsystems will be tested:

- User Login and Logout
- Login/Logout
- Visit Restaurant
- Blog
- About
- Add Restaurant
- Add new Blog
- Accountant Login and Logout

Software Risk Issues

There are several parts of the project that are not within the control of the application but have direct impacts on the process and must be checked as well. Our main goal is to design a test strategy that utilizes a balance of testing techniques to cover a representative sample of the system in order to minimize risk.

Features to be Tested

We will perform testing on use case specification functional requirement and non- functional requirement of all use cases and functions. They include

- User Login and Logout
- Login/Logout
- Visit Restaurant
- Blog
- About
- Add Restaurant
- Add new Blog
- Accountant Login and Logout

Features not to be Tested

The following list of areas that will not be specially addressed.

● Password Recovery:-

It can be done manually by the admin. As there is a limited schedule we will skip this feature.

Approach (Strategy)

The tests below base on the use case specification functional requirements, and non- functional requirements which have been identified as targets for testing. The test strategy presents the recommended approach to the testing of software applications.

1. Unit Testing:

Definition: Test smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation.

Participants/ Tested by : Developers

Methodology : Used for the Database test, records in each table, Basic function test, add a student, add accountant, email address format, empty field.

2. Regression Testing :

Definition : Regression testing for a project involves re-running previously executed test cases to ensure that new code changes have not adversely affected existing features. Here's an example of regression testing scenarios for your project based on the provided test cases

- Home Page Functionality
- User Authentication
- Blog Management

3. User Acceptance Testing -:

Definition : Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies.

Participants/ Tested by : Users / End Users

Methodology : It is used for Whole System Test

Item Pass/Fail Criteria

Item Pass/Fail Criteria The entrance criteria for each phase of testing must be met before the next phase can commence. Now the criteria for pass and fail are given below. These criteria are set with the help of software requirements specification version 1.0 1.

According to the given scenario the expected result need to take place then the scenario will be considered as pass otherwise that criteria should be failed.

▪ If an item tested 10 times, 9 times perfectly worked and single time do not work properly then it will be considered a fail case.

▪ System crash will be considered as a fail case.

▪ After submitting a query in the system, if expected data entry in database won't
appear then it will be considered as a fail case.

Suspension Criteria and Resumption Requirements

Suspension Criteria testing work is performed by the test team to ensure that all the functional activities are working well in the system properly. If some of these functions are not working properly in the system the further test procedure should not be continued to the next level. So here are some criterias for which we will be paused the test work for the System.

- If a sanity check failed.
- If the smoke test failed.
- If the number or type of defects reaches a point where the follow-on testing has no value, it makes no sense to continue the test; it just wasting resources.
- Certain individual test cases may be suspended, skipped or reduced if prerequisite tests have previously failed e.g. usability testing may be skipped if a significant number of system navigational tests fail.

These criteria are to be used to suspend the testing activity and resolve these criteria specifying testing activities which must be redone when testing is resumed.

Test Deliverables

The following documents will be generated as a result of these testing activities:

- test plan (this document)
- Test Design Specifications
- Acceptance Test plan.
- Test cases
- Unit test plan.
- Screen Prototypes.
- Test report.
- Test scenario and expected result in an excel sheet.
- System manual.

Environmental Needs

Software requirement:-

1. System Language: HTML, CSS, JavaScript, Expree js.
2. Operating system: windows10 and above.
3. Software: VS Code, Web Browser, Node js

Hardware requirement:-

1. Hard Disk : 50 GB and above
2. RAM : minimum 1GB
3. Processor: Intel i3 and above

- **Test Report for Test Cases inclusive of Test Procedures for identified Test Scenarios.**

Sr. No.	**Test Objective**	**Steps**	**Expected Result**	**Actual Result**	**Test Status**
TC 01	Home page display	- Open the website	"Resto-HUB" is in the driver title	"Resto-HUB" is in the driver title	Passed
TC 02	Reload home page	- Click on the logo button	Home page is reloaded	Home page is reloaded	Passed
TC 03	Display Restaurants page	- Click on "Visit Restaurants" button	Navigate to the Restaurants page	Navigate to the Restaurants page	Passed
TC 04	Display Restaurant detailed page	- Click on a restaurant	Navigate to the detailed restaurant page	Navigate to the detailed restaurant page	Passed
TC 05	Display page after login	- Open the Owners page	Should not navigate to the Owners page	Navigated to the Owners page	Failed
TC 06	Display Blogs page	- Open the Blogs page	"All Blogs" is displayed in an H1 element	"All Blogs" is displayed in an H1 element	Passed

TC 07	Display Blog page	- Click on a blog	Navigate to the blog page	Navigate to the blog page	Passed
TC 08	Display About page	- Open the About page	"/about" is in the current URL	"/about" is in the current URL	Passed
TC 09	Display Login page	- Open the Login page	"Login" is in the driver title	"Login" is in the driver title	Passed
TC 10	Display Signup page	- Open the Signup page	"Signup" is in the driver title	"Signup" is in the driver title	Passed
TC 11	Display Home page for owners	- Open the Owners page	"Welcome" is in the driver title	"Welcome" is in the driver title	Passed
TC 12	Display Signup page	- Click on the Signup button	"Signup" is in the driver title	"Signup" is in the driver title	Passed
TC 14	Login success	- Enter valid email and password	Navigate to admin-blogs page	Navigate to admin-blogs page	Passed
TC 15	Display Blog page	- Click on a blog	Navigate to the blog page	Navigate to the blog page	Passed

TC 16	Display Edit Blog page	- Click on "Edit Blog"	Navigate to the edit blog page	Navigate to the edit blog page	Passed
TC 17	Update the blog page	-Fill the required inputs	Blog data is updated	Blog data is updated	Passed
TC 18	Display Add Post page	- Click on "Add New Post"	Navigate to the add post page	Navigate to the add post page	Passed
TC 19	Display Add Restaurant page	- Open the Recommend page	"Add new post" is in the title	"Add new post" is in the title	Passed
TC 20	Display Blog detailed page	- Open a blog detailed page	"Blog title" is in the driver title	"Blog title" is in the driver title	Passed
TC 21	Display Facebook page	- Click on the Facebook button	Should navigate to Facebook	Navigated to Facebook	Passed
TC 22	Display Twitter page	- Click on the Twitter button	Should navigate to Twitter	Navigated to Twitter	Passed
TC 23	Display LinkedIn page	- Click on the LinkedIn button	Should navigate to LinkedIn	Navigated to LinkedIn	Passed

TC 24	Display Logged-in email	- Open the Owners page	Logged-in email is displayed	Logged-in email is displayed	Passed
TC 25	Display Blogs detailed page	- Open a blog detailed page	"Resto-HUB" is in the driver title	"Resto-HUB" is in the driver title	Passed
TC 26	Display Blogs from home page	- Click on "Blogs Here!!!" link	Navigate to admin-blogs page	Navigate to admin-blogs page	Passed
TC 27	Visit Website	- Click on "View Website" button	Should navigate to a website	Navigated to a website	Passed
TC 28	Visit Signup page	- Click on the Signup button from the Login page	Should navigate to the Signup page	Navigated to the Signup page	Passed
TC 29	Visit Login page	- Click on the Login button from the Signup page	Should navigate to the Login page	Navigated to the Login page	Passed
TC 30	Visit About page	- Click on "About Us" link	Should navigate to the About page	Navigated to the About page	Passed

- **Code:**

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.alert import Alert
from webdriver_manager.firefox import GeckoDriverManager
from selenium.webdriver.firefox.service import Service
from selenium.webdriver.support.ui import Select
import time
from webdriver_manager.chrome import ChromeDriverManager
webdriver_path=r"C:\Users\Dell\Downloads\chromedriver-win32\chromedriver-win32\chromedriver.exe"
chrome_options = webdriver.ChromeOptions()
service = webdriver.chrome.service.Service(executable_path=webdriver_path)
driver = webdriver.Chrome(service=service, options=chrome_options)
try:
    driver.get("http://localhost:3000/index")
    if "Resto-HUB" in driver.title:
        print("TC01 Passed: Home page is displayed")
    else:
        print("TC01 Failed: Home page is not displayed")
    time.sleep(1)
    logo_button = driver.find_element(By.CSS_SELECTOR, "#logo a")
    logo_button.click()
    driver.get('http://localhost:3000/index')
    if 'Resto-HUB' in driver.title :
         print("TC02 Passed:reloaded the home page")
```

```
  else:
    print("TC02 Failed: Home page is not reloaded")
  time.sleep(1)
  visit_restaurants_button=driver.find_element(By.CSS_SELECTOR,
"a.card-hover[href='restaurants']")
  visit_restaurants_button.click()
  driver.get("http://localhost:3000/restaurants")
  if 'http://localhost:3000/restaurants' in driver.current_url:
    print("TC03 Passed:Restaurants should be displayed")
  else:
    print("TC03 Failed: Restaurants page is not displayed")
  time.sleep(1)
  element = driver.find_element(By.CSS_SELECTOR, "div.restaurant-actions a")
  href_value = element.get_attribute("href")
  element.click()
  driver.get(href_value)
  if href_value in driver.current_url:
    print("TC04 Passed:Restaurant detailed page should be displayed")
  else:
    print("TC04 Failed: Restaurants deatiled page is not displayed")
  time.sleep(2)
  driver.get('http://localhost:3000/owners')
  if 'http://localhost:3000/owners' not in driver.current_url:
    print("TC05 Passed:Page should be displayed after login only")
  else:
    print("TC05 Failed: page is displayed")
  time.sleep(1)
  driver.get("http://localhost:3000/blogs")
```

```
main_element = driver.find_element(By.ID, "all-posts")

h1_element = main_element.find_element(By.TAG_NAME, "h1")

h1_text = h1_element.text

if h1_text=='All Blogs':

   print("TC06 Passed:Blogs page should be displayed")

else:

   print("TC06 Failed: Blogs page is not displayed")

time.sleep(1)

element = driver.find_element(By.CSS_SELECTOR, "div.post-actions a")

href_value = element.get_attribute("href")

driver.get(href_value)

if href_value in driver.current_url:

   print("TC07 Passed:Restaurants should be displayed")

else:

   print("TC07 Failed: Restaurants page is not displayed")

time.sleep(1)

driver.get("http://localhost:3000/about")

if '/about' in driver.current_url :

   print("TC08 Passed:About page should be displayed")

else:

   print("TC08 Failed: About page is not displayed")

time.sleep(1)

driver.get("http://localhost:3000/login")

if "Login" in driver.title:

   print("TC09 Passed:Login page should be displayed")

else:

   print("TC09 Failed: Login page is not displayed")

time.sleep(1)
```

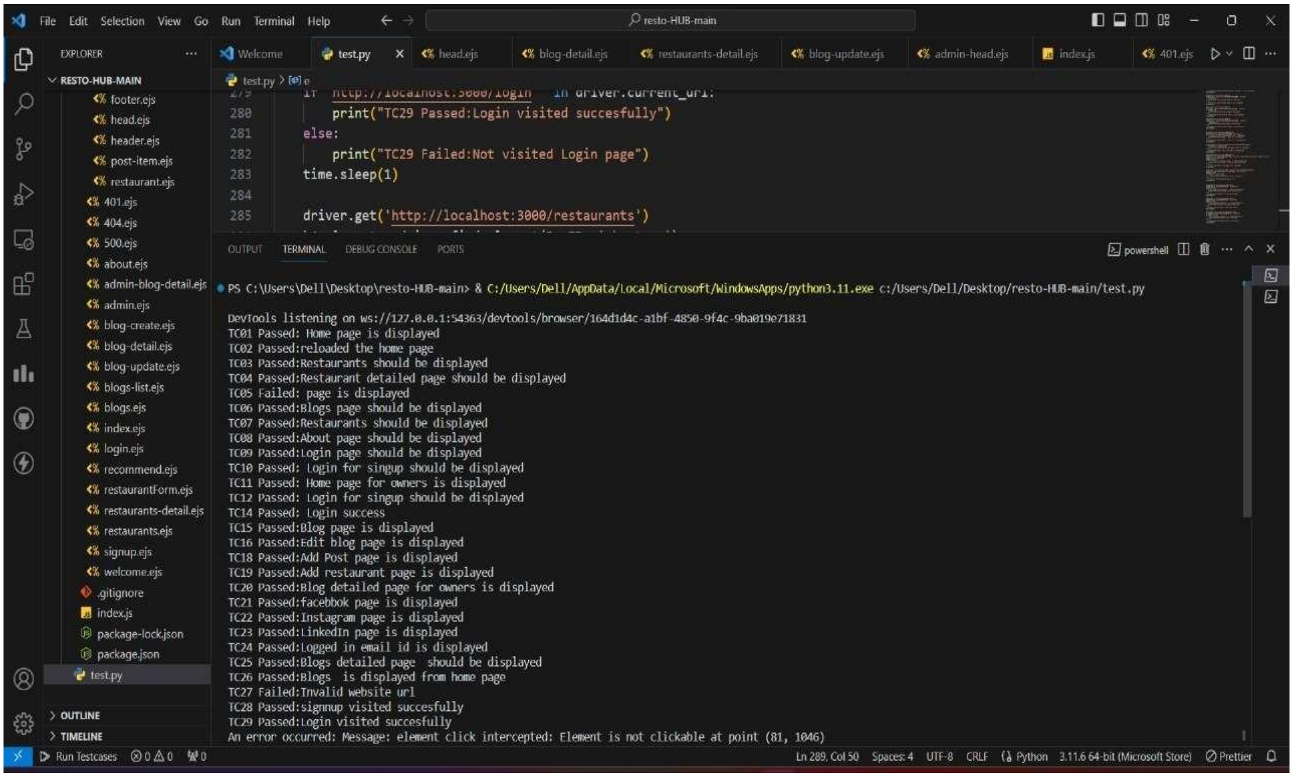
```
driver.get("http://localhost:3000/signup")
if "Signup" in driver.title:
    print("TC10 Passed: Login for singup should be displayed")
else:
    print("TC10 Failed: singup page not displayed")
time.sleep(1)
driver.get("http://localhost:3000/owners")
if "Welcome" in driver.title:
    print("TC11 Passed: Home page for owners is displayed")
else:
    print("TC11 Failed: Home page for owners is not displayed")
time.sleep(1)
signup_button = driver.find_element(By.CSS_SELECTOR, "a.btn-sl[href='/signup']")
signup_button.click()
driver.get("http://localhost:3000/signup")
if "Signup" in driver.title:
    print("TC12 Passed: Login for singup should be displayed")
else:
    print("TC12 Failed: singup page not displayed")
time.sleep(1)
# email_field=driver.find_element(By.ID,"email")
#  confirmEmail_field=driver.find_element(By.ID,"confirm-email")
# password_field=driver.find_element(By.ID,"password")
# email_field.send_keys('pravinchavan3002@gmail.com')
# confirmEmail_field.send_keys('pravinchavan3002@gmail.com')
# password_field.send_keys('pravin@123')
# submit_button=driver.find_element(By.CLASS_NAME,'btn')
# submit_button.click()
```

```
    #if 'http://localhost:3000/login' in driver.current_url:
    #     print("TC13 Passed: singup success")
    # else:
    #     print("TC13 Failed: singup fails")
    # time.sleep(2)
    driver.get("http://localhost:3000/login")
    email_field=driver.find_element(By.ID,"email")
    password_field=driver.find_element(By.ID,"password"
    email_field.send_keys('pratikchavan470@gmail.com')
    password_field.send_keys('pratik')
    submit_button=driver.find_element(By.CLASS_NAME,'btn')
    submit_button.click()
    if 'http://localhost:3000/admin-blogs' in driver.current_url:
        print("TC14 Passed: Login success")
    else:
        print("TC14 Failed: Login fails")
    time.sleep(1)
    element = driver.find_element(By.ID, "view-blog")
    href_value = element.get_attribute("href")
    driver.get(href_value)
    if href_value in driver.current_url:
        print("TC15 Passed:Blog page is displayed")
    else:
        print("TC15 Failed: Blog page is not displayed")
    time.sleep(1)
   driver.get('http://localhost:3000/admin-blogs')
    element = driver.find_element(By.ID, "edit-blog")
    href_value = element.get_attribute("href")
    driver.get(href_value)
    if href_value in driver.current_url:
        print("TC16 Passed:Edit blog page is displayed")
    else:
        print("TC16 Failed: Edit page is not displayed")
    time.sleep(1)
    # edit_title=driver.find_element(By.ID,"title")
    # edit_title.send_keys(' updated title')
    #  update_blog_button=driver.find_element(By.CLASS_NAME,'btn')
    # update_blog_button.click()
    #if 'http://localhost:3000/admin-blogs' in driver.current_url:
    # print("TC17 Passed:Blog is updated")
    # else:
    #     print("TC17 Failed: Blog are updated")
    driver.get('http://localhost:3000/admin-blogs')
    add_blog_button=driver.find_element(By.ID,'add-new-blog')
    add_blog_button.click()
   if 'http://localhost:3000/new-post' in driver.current_url:
        print("TC18 Passed:Add Post page is displayed")
    else:
        print("TC18 Failed: add post page is not displayed")
    time.sleep(1)
    driver.get('http://localhost:3000/recommend')
    if 'Add new post' in driver.title:
        print("TC19 Passed:Add restaurant page is displayed")
    else:
        print("TC19 Failed: add restaurant page is not displayed")
    time.sleep(1)
```

```
driver.get('http://localhost:3000/blog-detail/653bb839f7c711be2010e495')
if 'Blog title' in driver.title:
    print("TC20 Passed:Blog detailed page for owners is displayed")
else:
    print("TC20 Failed:Blog deatiled page for ownder is not displayed")
time.sleep(1)
driver.get('http://localhost:3000/')
facebook_btn = driver.find_element(By.CLASS_NAME ,'facebook')
if facebook_btn :
    print("TC21 Passed:facebbok page is displayed")
else:
    print("TC21 Failed:Facebook page is not displayed")
time.sleep(1)
driver.get('http://localhost:3000/')
twitter_btn = driver.find_element(By.CLASS_NAME ,'twitter')
if  twitter_btn:
    print("TC22 Passed:Instagram page is displayed")
else:
    print("TC22 Failed:Instagram page is not displayed")
time.sleep(1)
driver.get('http://localhost:3000/')
linkedin_btn = driver.find_element(By.CLASS_NAME ,'linkedin')
if linkedin_btn :
    print("TC23 Passed:LinkedIn page is displayed")
else:
    print("TC23 Failed:LinkedIn page is not displayed")
time.sleep(1)
driver.get('http://localhost:3000/owners')
```

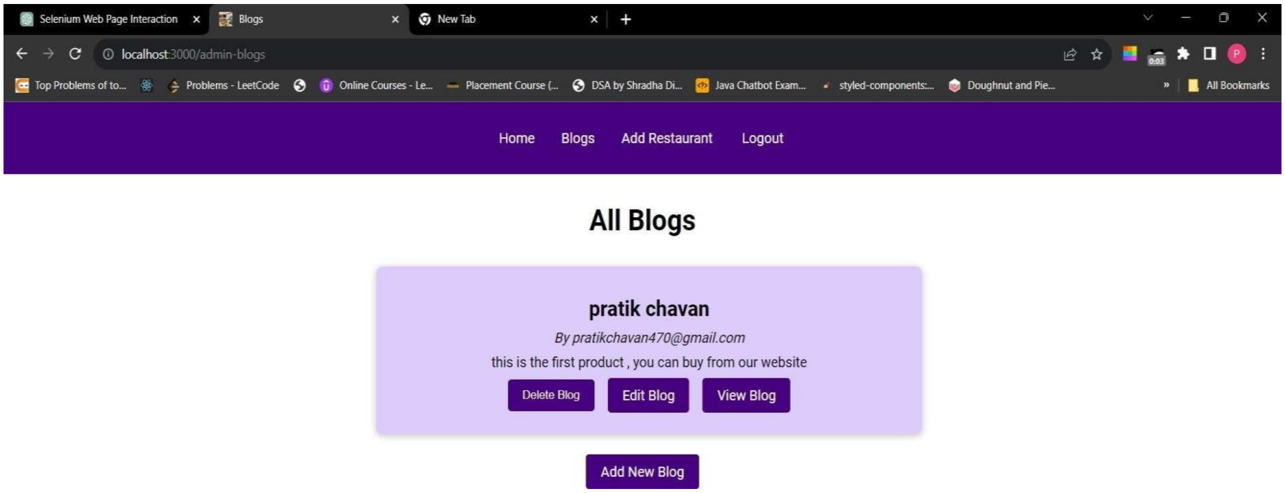
```
    email=driver.find_element(By.CLASS_NAME,'login-email')

    if 'You logged as pratikchavan470@gmail.com' in email.text:

        print("TC24 Passed:Logged in email id is displayed")

    else:

        print("TC24 Failed:Logged in email id is not displayed")

    time.sleep(1)

    driver.get("http://localhost:3000/blogs/6457f916919155091d448200")

    if 'Resto-HUB' in driver.title:

        print("TC25 Passed:Blogs detailed page should be displayed")

    else:

        print("TC25 Failed: Blogs detiailed is not displayed")

    time.sleep(1)

    driver.get('http://localhost:3000/owners')

    element = driver.find_element(By.XPATH, "//a[@href='/admin-blogs' and text()='Blogs
Here!!!']")

    element_text = element.text

    element.click()

    if 'http://localhost:3000/admin-blogs' in driver.current_url:

         print("TC26 Passed:Blogs is displayed from home page")

    else:

        print("TC26 Failed: Blogs page is not displayed")

    time.sleep(1)

except Exception as e:

    print(f"An error occurred: {str(e)}")

driver.quit()
```

- **Output:**



```
    if "http://localhost:3000/login" in driver.current_url:
        print("TC29 Passed:Login visited succesfully")
    else:
        print("TC29 Failed:Not visited Login page")
    time.sleep(1)

    driver.get('http://localhost:3000/restaurants')
```

```
PS C:\Users\Dell\Desktop\resto-HUB-main> & C:/Users/Dell/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Dell/Desktop/resto-HUB-main/test.py

DevTools listening on ws://127.0.0.1:54363/devtools/browser/164d1d4c-a1bf-4850-9f4c-9ba019e71831
TC01 Passed: Home page is displayed
TC02 Passed:reloaded the home page
TC03 Passed:Restaurants should be displayed
TC04 Passed:Restaurant detailed page should be displayed
TC05 Failed: page is displayed
TC06 Passed:Blogs page should be displayed
TC07 Passed:Restaurants should be displayed
TC08 Passed:About page should be displayed
TC09 Passed:Login page should be displayed
TC10 Passed: Login for singup should be displayed
TC11 Passed: Home page for owners is displayed
TC12 Passed: Login for singup should be displayed
TC14 Passed: Login success
TC15 Passed:Blog page is displayed
TC16 Passed:Edit blog page is displayed
TC18 Passed:Add Post page is displayed
TC19 Passed:Add restaurant page is displayed
TC20 Passed:Blog detailed page for owners is displayed
TC21 Passed:facebbok page is displayed
TC22 Passed:Instagram page is displayed
TC23 Passed:LinkedIn page is displayed
TC24 Passed:Logged in email id is displayed
TC25 Passed:Blogs detailed page  should be displayed
TC26 Passed:Blogs  is displayed from home page
TC27 Failed:Invalid website url
TC28 Passed:signnup visited succesfully
TC29 Passed:Login visited succesfully
An error occurred: Message: element click intercepted: Element is not clickable at point (81, 1046)
```

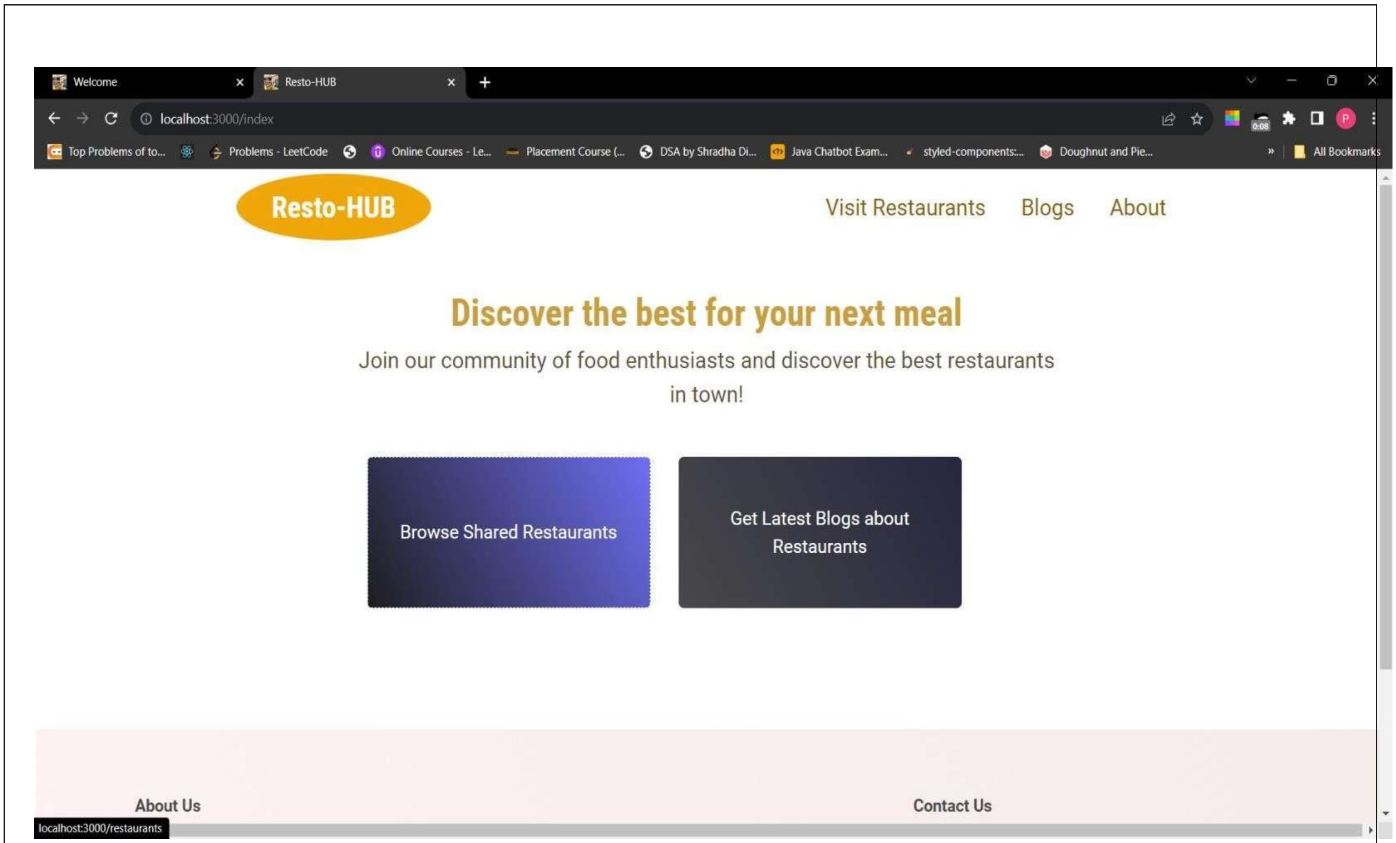


All Blogs

pratik chavan

By pratikchavan470@gmail.com

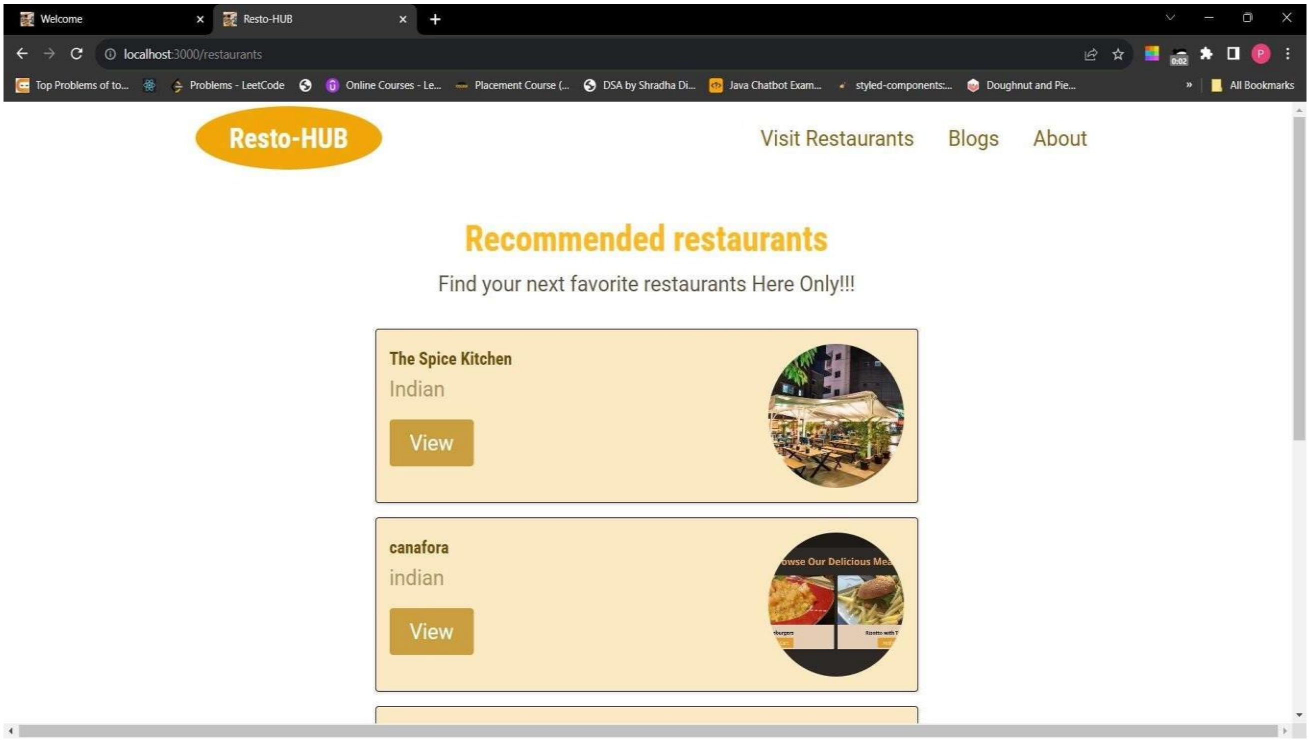
this is the first product , you can buy from our website



Discover the best for your next meal

Join our community of food enthusiasts and discover the best restaurants in town!

Browse Shared Restaurants
Get Latest Blogs about Restaurants
About Us
Contact Us
localhost:3000/restaurants



Recommended restaurants

Find your next favorite restaurants Here Only!!!

The Spice Kitchen
Indian

canafora
indian