

# **Compliance Report**

## **for**

# **Food Delivery Service Management Software**

**Prepared by-**

**Gangaram Arvind Sudewad (20CS30017)**

**Ragala Abhishek (20CS10048)**

**Sailada Vishnu Vardhan(20CS10051)**

**March 22, 2022**

**Team: Brogrammers**

**Group No.:22**

# Contents:

<b>1. Introduction with main points.....</b>	<b>3</b>
<b>2. Compliance report for frontend.....</b>	<b>3</b>
<b>2.1 Home page.....</b>	<b>3</b>
<b>2.2 Login Page.....</b>	<b>3</b>
<b>2.3 Signup Page.....</b>	<b>3</b>
<b>2.4 Restaurant selection for customer.....</b>	<b>4</b>
<b>2.5 Choose the number of food items.....</b>	<b>4</b>
<b>2.6 Edit menu for restaurants.....</b>	<b>4</b>
<b>2.7 Manger buttons.....</b>	<b>3</b>
<b>3. Backend Database Management:.....</b>	<b>5</b>
<b>3.1. Testing 'class Menu(models.Model):' .....</b>	<b>5</b>
<b>3.2. Testing functions.....</b>	<b>3</b>

## 1.Introduction with main points

This document comprises of the Compliance Report document of Food Delivery Service Management Software. In this we have listed down the test cases that were mentioned in the test suite also denoting whether we obtained the results as PASS/FAIL that were obtained after testing the software. The exact test case and outputs are taken from the test case document and some of the extra things are also added.

**Important:** We are denoting the results as **PASSED** when the positive test cases are passed and there are no errors observed. But, when we are entering any negative cases it will not pass then we will denote its expected output as **FAILED** and also showing some errors for the users.

## 2. Compliance report for frontend

### 2.1 Home Page

Working Properly

### 2.2 Login Page

1. Both Username and Password are correct  
**PASSED**

2. Username is correct but Password is incorrect  
**FAILED** (*error message: Invalid credentials*)

3. Username is incorrect but Password is correct  
**FAILED** (*error message: Invalid credentials*)

4. Both Username and Password are incorrect  
**FAILED** (*error message: Invalid credentials*)

### 2.3. Signup Page

1. Not entered anything or any blank is empty  
**FAILED** (*error message: please fill out this field*)

2. Not entered correct email address

Eg. a@ej, abc123

**FAILED** (*error message: Invalid Email*)

3. Entered an email which already exists

**FAILED** (*error message: Email already exists*)

4. Entered valid email and filled all the fields

**PASSED**

## 2.4 Restaurant selection for customer

**PASSED** all the cases

## 2.5 Choose the number of food items

1. Not entered any value any submitted the order

**FAILED** (*error message: please select at least one item*)

2. Submit button

a. Entered ratings 0 or above 5

**FAILED** (*error message: rate between 1 and 5*)

b. else

**PASSED**

3. Back Buttons

**PASSED**

## 2.6 Edit menu for restaurants

All the buttons edit, stats, back, previous records

**PASSED**

## 2.7 Manger buttons

All the buttons stats, edit, details of all users, logout, back

**PASSED**

### 3. Backend Database Management:

#### 3.1. Testing 'class Menu(models.Model):'

1.testing edited prices

**PASSED** (*price edited successful*)

2.testing average ratings

**PASSED**

3.testing customers names

**PASSED**

#### 3.2. Testing functions

1.def edit(request):

Edit function for all the restaurants

**PASSED**

2.def back(request): and def logout(request):

Back options and logout buttons

**PASSED**

3.def reviews(request):

Review will take ratings and average

**PASSED**

4.def BillH(request):

Take all the requests given by customer and return the net amount

**PASSED**

5.def stats1():

Will plot the bar graph of required restaurants items

**PASSED**

6.def user(request):

Take the username and email and return the required page

a. Wrong email or password

**FAILED** (*error message: Invalid credentials*)

b. Else **PASSED**

**7.**def signup(request):

For new users

**PASSED**

**8.**def create(request):

Store all the details of newly joined users

**PASSED**

**9.**def prevorders(request):

Storing all the previous orders of customers and restaurants

**PASSED**

**10.**def requests(request):

Print all the history of restaurant

**PASSED**