

# **FLEXPACK**

**A Project Report**

*Submitted by:*

**PARTH MANIYAR (AU1741068)**

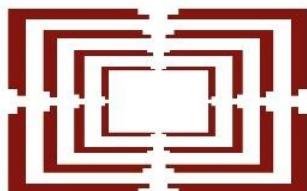
*partial fulfillment for the award of the degree  
of*

**BACHELOR OF TECHNOLOGY**

**IN**

**INFORMATION AND COMMUNICATION TECHNOLOGY (ICT)**

at



**Ahmedabad  
University**

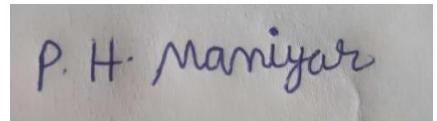
School of Engineering and Applied Sciences (SEAS)

Ahmedabad, Gujarat

May, 2021

## **DECLARATION**

I hereby declare that the project entitled “Flexpack” submitted for the B. Tech. (ICT) degree is my original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.



**Signature of the Student**

**Place: Rajkot**

**Date: 03/05/2021**

## **Simform Solutions Pvt Ltd.**

501 – Binori Bsquare 2,  
Nr. Doubletree Hotel  
Iscon-Bopal Road, Ahmedabad-380058  
Gujarat, India  
CIN: U72900GJ1988PTC011352

+91 (79) 40070170  
[hello@simform.com](mailto:hello@simform.com)  
[www.simform.com](http://www.simform.com)

**Date: 19<sup>th</sup> April,2021**

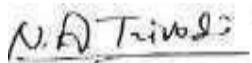
## **CERTIFICATE OF INTERNSHIP**

This is to certify that **Mr. Parth Maniyar** a student of B.Tech., School of Engineering and Applied Science- Ahmedabad University has successfully completed his internship in **Python technology** from **15<sup>th</sup> December 2020 to 15<sup>th</sup> April 2021**.

During his internship, he has demonstrated his skills with self-motivation to learn new skills and his performance exceeded the company's expectations.

We at Simform wish him the best and success in his life and career.

### **Authorized Signatory**



**Nikita Trivedi**

**Senior Manager- Human Resource**



## **CERTIFICATE**

This is to certify that the project titled “Flexpack” is the bona fide work carried out by Parth Maniyar (AU1741068), a student of B Tech (ICT) of School of Engineering and Applied Sciences at Ahmedabad University during the academic year 2020-2021, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology (Information and Communication Technology) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

*Auraag*

**Signature of the Guide**

**Place: Ahmedabad**

**Date: 12/05/2021**

# TABLE OF CONTENTS

Title Page	i
Declaration of the Student	ii
Internship Certificate	iii
Certificate of the Guide	iv
Abstract	x
Acknowledgement	xi
List of Figures	xii
List of Tables	xiv
Gantt Chart	xv
1 INTRODUCTION	1
1.1 PROJECT DETAILS	1
1.2 PURPOSE	1
1.3 SCOPE	1
1.4 OBJECTIVE	1
1.5 TECHNOLOGY AND LITERATURE REVIEW	1
2 PROJECT MANAGEMENT	2
2.1 FEASIBILITY STUDY	2
2.1.1 Technical Feasibility	2
2.1.2 Time Schedule Feasibility	2
2.1.3 Operational Feasibility	2
2.1.4 Implementation Feasibility	2
2.2 PROJECT PLANNING	3
2.2.1 Project Development Approach and Justification	3
2.2.2 Milestones and Deliverables	3
2.2.3 Roles and Responsibilities	4
2.2.4 Group Dependencies	4
2.2.5 Project Scheduling Chart	4
3 SYSTEM REQUIREMENTS STUDY	5
3.1 STUDY OF CURRENT SYSTEM	5

3.2 PROBLEMS AND WEAKNESS OF CURRENT SYSTEM	5
3.3 USER CHARACTERISTICS	5
3.4 HARDWARE AND SOFTWARE REQUIREMENTS	5
3.4.1 Hardware Requirements	5
3.4.2 Software Requirements	5
3.5 CONSTRAINTS	6
3.5.1 Hardware Limitations	6
3.5.2 Interface to other applications	6
3.5.3 Reliability Requirements	6
3.5.4 Criticality of the Application	6
3.6 ASSUMPTIONS AND DEPENDENCIES	6
<b>4 SYSTEM ANALYSIS</b>	<b>7</b>
4.1 REQUIREMENTS OF NEW SYSTEM	7
4.1.1 Use Case Diagram	7
4.1.1.1 System Functional Requirements	10
4.2 CLASS DIAGRAM	11
4.3 ACTIVITY DIAGRAM	12
4.4 SEQUENCE DIAGRAM	13
4.5 DEPLOYMENT DIAGRAM	14
4.6 SYSTEM DIAGRAM	15
<b>5 SYSTEM DESIGN</b>	<b>16</b>
5.1 SYSTEM DESIGN ARCHITECTURE	16
5.1.1 Architectural Design	16
5.1.2 Logical Design	16
5.1.3 Physical Design	16
<b>6 IMPLEMENTATION AND PLANNING</b>	<b>17</b>
6.1 IMPLEMENTATION ENVIRONMENT	17
6.2 PROGRAM / MODULES SPECIFICATION	17
6.3 CODING STANDARDS	18
<b>7 TESTING</b>	<b>19</b>
7.1 TESTING PLAN	19

<b>7.2 TESTING STRATEGY</b>	19
7.2.1    Unit Testing	19
7.2.2    Sub System Testing	20
7.2.3    System Testing	20
7.2.4    Acceptance Testing	20
<b>7.3 TESTING METHODS</b>	21
<b>7.4 TEST CASES</b>	22
7.4.1 SIGNUP	22
7.4.2 LOGIN	25
7.4.3 DESIGN BOX	27
<b>8 USER MANUAL</b>	32
<b>8.1 SNAPSHOTS OF CUSTOMER SIDE OF THE WEBSITE</b>	32
8.1.1 Home Page(customer)	32
8.1.1.1 Home Page Create custom box	32
8.1.1.2 Choose Style (customer)	33
8.1.1.3 Get Inspired	34
8.1.1.4 Feature	34
8.1.2 Design A Box(customer)	35
8.1.2.1 Size of the Box	35
8.1.2.2 Materials	36
8.1.2.3 Add Image: Choose	37
8.1.2.3.1 Choose File validation	37
8.1.2.3.2 Fit and Cover the image	37
8.1.2.3.2 Remove Image	38
8.1.3 Add Text	39
8.1.3.2 Text Format	39
8.1.4.3 Text Color	40
8.1.3.4 Text Align	40
8.1.3.4.1 Fit Align	40
8.1.3.4.2 Center Align	41
8.1.3.5 Stroke Color	41
8.1.3.6 Stroke Width	42

8.1.3.7 Font Size	42
8.1.3.8 Background Color	43
8.1.3.9 Bold Align	43
8.1.3.10 Italic Align	44
8.1.3.11 Underline and Line through Align	44
8.1.3.12 Overline Align	45
8.1.3.13 Remove Text	45
8.1.4 Quantity	46
8.1.5 Add Background	46
8.1.5.1 Add Color of Background	46
8.1.5.2 Reset Background	47
8.1.6 Box Size Validation	47
8.1.7 Checkout Page	48
8.1.7.1 Address Detail	48
8.1.7.2 Transaction Complete Message	48
8.1.8.3 Transaction mail to customer	49
8.1.8 About Us	49
8.1.8.1 About Us: Our history	50
8.1.9 Contact Us –Before email is send	50
8.1.9.1 Send email to Customer	51
8.1.9.2 Email to Customer	52
8.1.9.3 Email to Admin	52
8.1.10 Saved Design	53
8.1.10.1 Ordered Box Cannot be Deleted	54
8.1.11 Signup	54
8.1.12 Login	54
8.2 Snapshot of Admin Side of the Website	55
8.2.10 Login	55
8.2.11 Dashboard	55
8.2.12 Detail view of the user	56
8.2.13 Update the status of Delivery	56
8.3 Snapshot of Database of the Website	57
<b>9 LIMITATIONS AND FUTURE ENHANCEMENT</b>	60
9.1 LIMITATIONS	60

9.2 FUTURE ENHANCEMENT	60
10 CONCLUSION AND DISCUSSION	61
10.1 CONCLUSION	61
DISCUSSION	61
REFERNCES	62
Appendix A (Comparison of Django and Nodejs)	63

## **ABSTRACT**

Today, packaging boxes are being used in almost all products, such as electronics, toys, cosmetics, food and bakery products and a lot more. From small gift to large product for shipping boxes are used.

Most of the customer has to go to box making company to choose design the boxes for their different products. Also, in that they have the limited choices for them.

The project this team is working on highlights of the problem that currently people are facing while starting their new business. While starting new business people need boxes to export things and if the pattern and design of box is not good then it might affect your product sales.

The exact problem which is being faced is that people have things in mind that what they want but they won't be able to get exact match of their imagination or it might be possible that if they get the exact match of the box but it won't fit in the budget. Therefore, we have made this website for the people who are facing this issue and we have developed it user friendly, so people don't get trouble while using it. We have made a 3D design lab for designing the boxes at minimum cost.

# Acknowledgment

I would like to thank the Department of Information & Communication Technology, School of Engineering and Applied Science, Ahmedabad University, and Simform Solutions, Ahmedabad for giving me an opportunity to work practically in the industry under a major project.

Moreover, I would like to thank my Internal Guide **Prof. Anurag Lakhiani** for her consistent support and expert insights.

I would likewise thank my mentors **Mr. Mayur Chhapra, Mr. Varis Bhalala** and my Team Manager **Mr. Bhumin Patel** at Simform Solutions. I can confidently say I wouldn't have grown as much as I have these past five months without your detailed feedback and unwavering support.

My heartfelt gratitude also goes to all the people who helped me in this project via their inputs and suggestions. The atmosphere at the university and the company played a major role in the development of this project.

Finally, I would like to thank my parents and my colleagues for their untiring support throughout the course of this project.

School of Engineering and Applied Science



## LIST OF FIGURES

<b>NAME</b>	<b>PAGE</b>
Fig. 4.1.1 Use Case Diagram	7
Fig. 4.2 Class Diagram	11
Fig. 4.3 Activity Diagram	12
Fig. 4.4 Sequence Diagram	13
Fig. 4.5 Deployment Diagram	14
Fig. 4.6 System Diagram	15
Fig 8.1.1.1 Home Page – Create Custom Box	32
Fig 8.1.1.2 Choose Style	33
Fig. 8.1.1.3 Get Inspired	34
Fig 8.1.1.4 Features	34
Fig. 8.1.2 Design A Box	35
Fig. 8.1.2.1 Size of the Box	35
Fig. 8.1.2.2.1 White Board	36
Fig. 8.1.2.2.2 Dream Coat Board	36
Fig. 8.1.2.2.3 Kraft Board	36
Fig. 8.1.2.3.1 Choose File Validation	37
Fig. 8.1.2.3.2 Fit and Cover the Image	38
Fig. 8.1.2.3.3 Remove Image	38
Fig. 8.1.3.1 Add Text	39

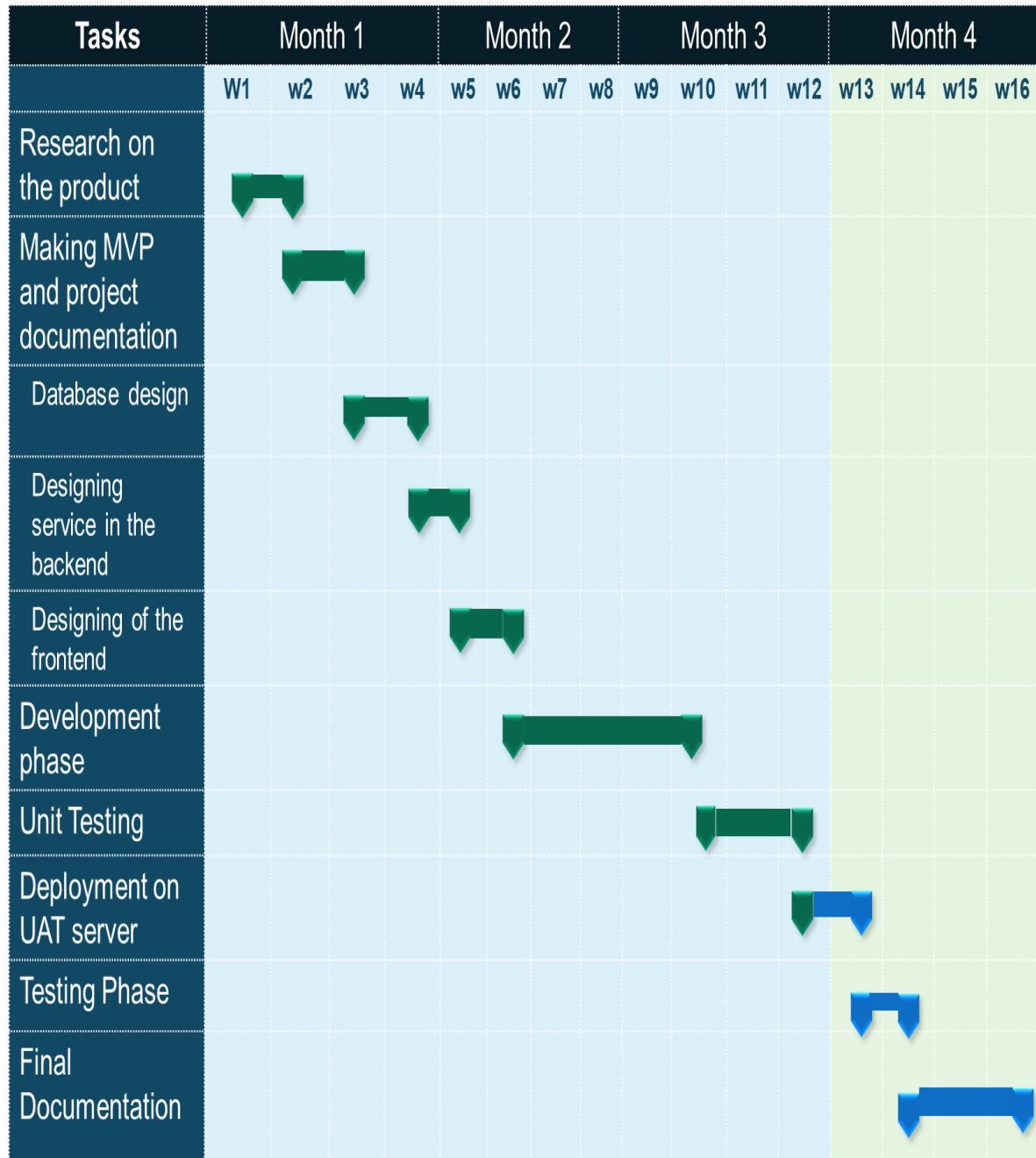
Fig. 8.1.3.2 Text Format	39
Fig. 8.1.3.3 Text Color	40
Fig 8.1.3.4.1 Fit Align	40
Fig. 8.1.3.4.2 Center Align	41
Fig. 8.1.3.5 Stroke Color	41
Fig. 8.1.3.6 Stroke Width	42
Fig. 8.1.3.7 Font Size	42
Fig. 8.1.3.8 Background Color	43
Fig. 8.1.3.9 Bold Align	43
Fig. 8.1.3.10 Italic Align	44
Fig. 8.1.3.11 Underline and line through align	44
Fig. 8.1.3.12 Overline Align	45
Fig. 8.1.3.13 Remove Text	45
Fig. 8.1.4 Quantity	46
Fig. 8.1.5.1 Add Color Background	46
Fig. 8.1.5.2 Reset Background	47
Fig. 8.1.6 Box Size Validation	47
Fig. 8.1.7.1 Address Detail	48
Fig. 8.1.7.2 Transaction Complete Message	48
Fig. 8.1.7.3 Transaction email to customer	49
Fig. 8.1.8 About Us	49
Fig. 8.1.8.1 About Us – Our History	50
Fig. 8.1.9 Contact Us – Before Email is sent	51
Fig. 8.1.9.1 Sent Email to Customer	51

Fig. 8.1.9.2 Email to Customer	52
Fig. 8.1.9.3 Email to Admin	52
Fig. 8.1.10 Saved Design	53
Fig. 8.1.10.1 Ordered Box Cannot be Deleted	53
Fig. 8.1.11 Sign Up	54
Fig. 8.1.12 Login	54
Fig. 8.2.1 Login - Admin	55
Fig. 8.2.2 Dashboard	55
Fig. 8.2.3 Detail View of the user	56
Fig. 8.2.4 Update Status of Delivery	57
Fig. 8.3 Snapshot of Database of the Website	59

## **LIST OF TABLES**

<b>NAME</b>	<b>PAGE</b>
Table 2.2.3 Roles and Responsibilities	4
Table 2.2.5 Project Scheduling Chart	4
Table 7.4.1 Sign Up	24
Table 7.4.2 Login In	26
Table 7.4.3 Design Box	31

## Gantt Chart



# **1. INTRODUCTION**

## **1.1 PROJECT DETAILS**

Our project Flexpack is an online shopping website for custom boxes. Flexpack is custom design lab with the live preview and it helps customer to design their box for their product accordingly. Our boxes can be customized with different functionality of color, shape, materials, and size. In a market where more and more companies are shipping goods to customers all over the India and the world, custom shipping boxes have suddenly become a vital element of commerce. Our main feature is custom packages, whether they're small cartons, large printed shipping box, or anything in between.

## **1.2 PURPOSE**

It is difficult for many people to find the box in customize way. So, the purpose of the project is to provide customize box. Our boxes are engineered to be the best. That means the finest materials and the most precise layout process - so putting your box together is a smooth and easy task that can be done in seconds.

## **1.3 SCOPE**

Online shopping is trending so many companies has need of boxes. But as people don't get much options for their design or imagination of their boxes. So, we tried to make design lab for customizing the box.

## **1.4 OBJECTIVE**

Our main objective is to redevelop the imagination of the box which user has imagined. It's quite user friendly and we tried to redefine the user experience of designing the boxes.

## **1.5 TECHNOLOGY AND LITERATURE REVIEW**

This system is totally web-based and created on open-source applications and software. This system is created in Python-Django with database is Postgres and it supports and is compatible with every web browser and operating system. We have also used jQuery framework for dynamically changing the content of the web page without reloading the pa

## **2. PROJECT MANAGEMENT**

### **2.1 FEASIBILITY STUDY**

#### **2.1.1 Technical Feasibility**

Python-Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. The free availability of all these technologies that we are going to use makes this project technically feasible. PostgreSQL is a powerful, open-source object-relational database system. PostgreSQL, also known as Postgres, is a free and open-source relational database management system emphasizing extensibility and SQL compliance.

#### **2.1.2 Time Schedule Feasibility**

This project mainly has four steps. The first one is extensive research about the requirements of small and medium-scale businesses. Understanding their workflow and generalizing them to be used in any industry. Then the second step is designing the prototype and workflow of the software. The third step is implementation i.e., coding. This phase includes developing the module and carefully integrating them. The last step includes testing and maintenance. This is also a very important phase of software development. In this phase, we would ensure that we provide bug-free software to the customer. We will ensure that we deliver the software in the given time duration by following the project scheduling chart. So, this project is time scheduling feasible.

#### **2.1.3 Operational Feasibility**

The main goal of this project is to reduce the hassle of designing the box for customers. As the only company that provides a custom design lab, live preview, and instant quote, we not only give you the freedom to unpack your creativity and develop your own design, but we also help you do it efficiently. And if you're having trouble, we're here to assist with anything from design to technical support.

#### **2.1.4 Implementation Feasibility**

This Flexpack web-application is mainly for Box manufacturing company to provide ease to their customer. By implementing our web-application customer don't have to visit the company frequently

## **2.2 PROJECT PLANNING**

### **2.2.1 Project Development Approach and Justification**

#### *Project Development Approach*

We plan on taking a time-based approach to delivering this website on time. To begin with, we will create a set of deadlines to complete certain tasks. This will give us a general idea of how to meet the final deadline on time. Furthermore, these tasks contain various independent divisions of the software, which can be tested individually. Thus, this will help us report our project in parts and test it.

#### *Agile model:*

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks.

As illustrated in Fig, this model divides the cycle into the phases mentioned below:

1. Planning.
2. Requirement's analysis.
3. Design.
4. Coding
5. Unit Testing.
6. Acceptance Testing.

#### *Justification*

After the feasibility study as the functional requirements are almost clear, but in some cases like implementing, designing as well as in testing time errors can occur. Here we have to decompose the system into modules. That is why we decide to use the Agile model which is the most suitable model here i.e., if we find any difficulty in coding and testing a modification in design can be done easily.

### **2.2.2 Milestones and Deliverables**

The milestone in this project would be to develop the admin dashboard and its functionalities. Other milestones would be the customer interface for designing the boxes i.e., adding image, add different colors, different kind of materials, sizes, different text formatting, price deciding and payment module. Other milestones is sending email for customer queries. The delivery of the software would be as a whole at the end of three months, but once the admin module is done, we would take suggestions from the client.

### **2.2.3 Roles and Responsibilities**

Following table describes roles and responsibilities of each team member

Name	Analysis	Design	Prototype	Coding	Testing	Documentation
Parth Maniyar	Yes	Yes	Yes	Yes	Yes	Yes

Table 2.2.3 Roles and Responsibilities

### **2.2.4 Group Dependencies**

The dependencies among the tasks include the following:

- Analysis or System Requirement Study (SRS) is independent of all, yet will be started after completion of the feasibility study and project planning.
- Designing the prototype can be done simultaneously with system analysis.
- Development of the project is preceded by the designing of prototype and system analysis.
- Testing can be only done once the development of some major functionalities is completed and are ready to be tested.
- Documentation is independent of all the tasks and can be done as the other tasks proceed.

### **2.2.5 Project Scheduling Chart**

Following table describes project schedule.

Sr. No.	Task	Start Date	End Date	Duration
1	Analysis	03/01/21	15/01/21	14 days
2	Design	17/01/21	29/01/21	14 days
3	Coding	01/02/21	28/02/21	28 days
4	Testing	01/03/21	13/03/21	14 days
5	Maintenance	15/03/21	25/03/21	10 days

Table 2.2.5 Project Scheduling Chart

### **3. SYSTEM REQUIREMENTS STUDY**

#### **3.1 STUDY OF CURRENT SYSTEM**

In the current system the customer has to go to box manufacturing company for designing and placing the order of the boxes. To get customize box they have to elaborate their needs and they are been short listed for their customizations. As of now people don't get their customized box as they demanded of. They have less choice for designing of their boxes.

#### **3.2 PROBLEMS AND WEAKNESS OF CURRENT SYSTEM**

The weakness of the system available in the market is that it is very costly and unsatisfied by the user. And also the process of designing the box is too time consuming as people have to visit the company and search for their design.

#### **3.3 USER CHARACTERISTICS**

The user of this system will be owners of medium and small-scale businesses. This system will be user friendly, so that people can easily design and can place the order of the box.

#### **3.4 HARDWARE AND SOFTWARE REQUIREMENTS**

##### **3.4.1 Hardware Requirements**

The device should be enabled with the internet and with at least 4 GB RAM and at least 1GB of free space on hard disk is required.

##### **3.4.2 Software Requirements**

Operating system: Any OS

IDE: Any IDE

Programming Language: Python

Browser: Any browser

Database: PostgreSQL

Software: RabbitMQ

## **3.5 CONSTRAINTS**

### **3.5.1 Hardware Limitations**

There are no hardware limitations for this system because once the complete system is developed care would be taken while deploying system so necessary prerequisites are met.

### **3.5.2 Interface to other applications**

We have made API to perform curd operations on the box and integrated a PayPal interface for payment service

### **3.5.3 Reliability Requirements**

The web application does demand much reliability and it is fully assured that the particular information about the user should be secured and flow is maintained and accessed according to the rights.

### **3.5.4 Criticality of the Application**

The web application deals with the user's personal tasks so the task and respective details should be highly confidential and in proper flow.

## **3.6 ASSUMPTIONS AND DEPENDENCIES**

Assumptions:

- User have sufficient privileges to access internet.
- Server is running smoothly.
- Database updates are giving expected and accurate results.

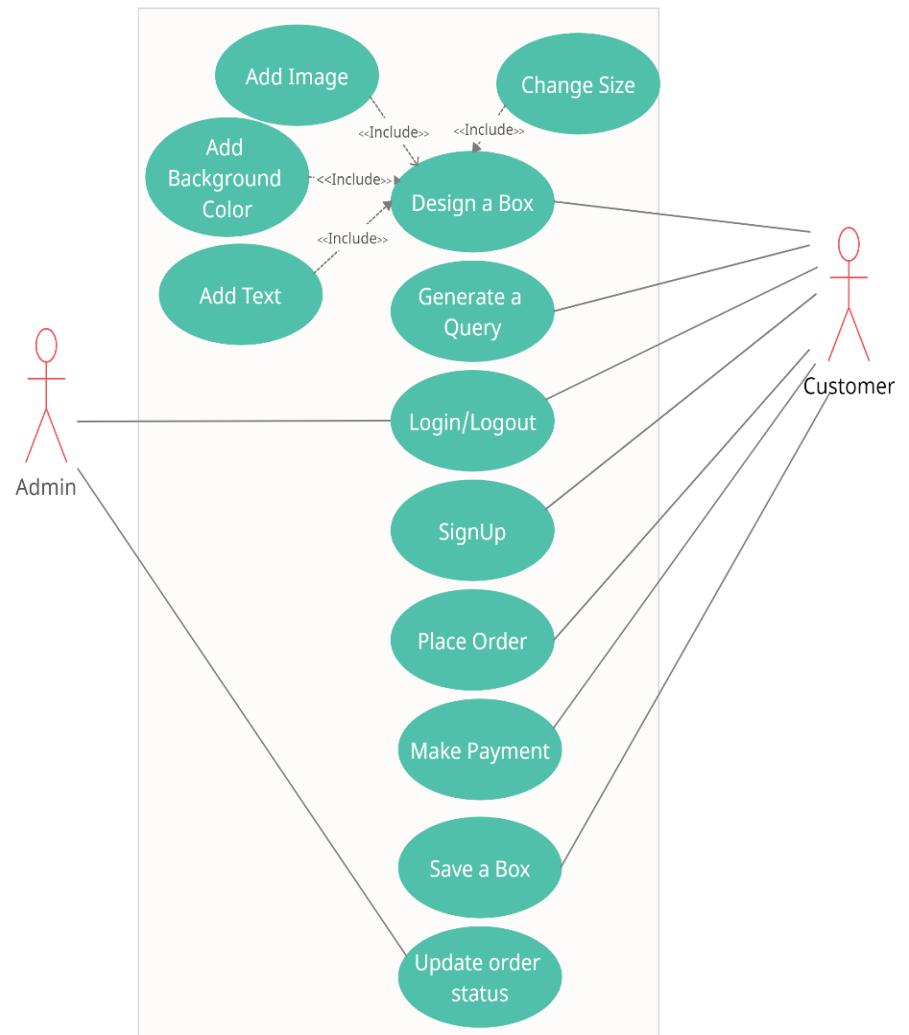
Dependencies:

- Browser version should be latest.
- Python version should be 3.8

## 4. SYSTEM ANALYSIS

### 4.1 REQUIREMENTS OF NEW SYSTEM

#### 4.1.1 Use Case Diagram



**Fig. 4.1.1 Use Case Diagram**

The SRS is organized with the functional requirements such that major functionalities are described from signing up, log in, stress score calculation

## 4.1.1.1 System Functional Requirements

- **SOFTWARE REQUIREMENT SPECIFICATIONS (SRS):**

### Functional

#### **Requirements: Role**

##### **R1: Super Admin:**

**R1.1 Authentication:** Super Admin can log into their account.

**R1.1.1 Login:** Super Admin can login by using credentials details.

- 1) **Pre-condition:** Super Admin have their credentials.
- 2) **Input:** Super Admin has to enter his/her email and password.
- 3) **Processing:** System will validate the email and password pair and check if user is registered or not and after that system will check that this user is Super Admin or not.
- 4) **Output:**
  - a) If username and password is correct then login to admin dashboard or
  - b) If authentication is incorrect then error message will generate.
- 5) **Postcondition :**
  - a) If authentication fails then user has to retry.

**R1.2 Delivery status:** Only Super Admin can set the status of the Delivery.**R1.2.1 Status:** Super Admin has to enter status of the delivery.

- 1) **Pre-condition:** User is super admin.
- 2) **Input:** Super Admin has to edit delivery status of box.
- 3) **Processing:** All status will store in database.
- 4) **Output:** Status Updated.
- 5) **Post-condition:** After Updating status will be updated to user profile.

#### **Role R2:**

**R2.1 Authentication:** Customer can sign up or log into their account and also can retrieve their password using forget password.

**R2.1.1 Registration:** Customer can register into system.

- 1) **Pre-condition:** The user is not registered yet.
- 2) **Input:** The user has to input his/her username, first name, last name, email and password.

- 3) **Processing:** The username, email and password will be checked.
- 4) **Output:**
  - a) New user is created in database.
- 5) **Postcondition :**
  - a) If registration fails then user has to retry, or
  - b) If registration succeed then user will redirect to login page.

**R2.1.2 Login:** Customer can login by using credentials details.

- 1) **Pre-condition:** Customer have their credentials.
- 2) **Input:** Customer has to enter his/her email and password.
- 3) **Processing:** System will validate the email and password pair and check if user is registered or not and after that system will check that this user is Customer or not.
- 4) **Output:**
  - A) If authentication failed then error message will be shown.
  - B) If authentication succeed then user is redirect to home page
- 5) **Postcondition :**
  - a) If authentication fails then user has to retry.

**R2.2 Design box:** customer can design new customize box.

**R2.2.1 mailer box:** customer can design mailer box.

- 1) **Input:** Customer can enter property details like dimension, material, design, quantity etc.
- 2) **Processing:** All property data will store in database.
- 3) **Output:** Property added successfully.

**R2.2.2 product box:** customer can design product box.

- 1) **Input:** Customer has to enter property details like dimension, material, design, quantity etc.
- 2) **Processing:** All property data will store in database.
- 3) **Output:** Property added successfully.

**R2.2.3 shipping box:** customer can design product shipping box

**Input:** Customer has to enter property details like dimension, material, design, quantity etc.

- 2) **Processing:** All property data will store in database.
- 3) **Output:** Property added successfully.

**R2.2.4 Econoflex shipping box:** customer can design **Econoflex shipping** box.

- 1) **Input:** Customer has to enter property details like dimension, material, design, quantity etc.

- 2) **Processing:** All property data will store in database.
- 3) **Output:** Property added successfully.

**R4.1 Payment:** User can start payment process from this page

- 1)**Input:** After customize a box, customer clicks the payment button, selects the payment option and then gives the credentials.
- 2)**Output:** Redirect to payment gateway and then display if the transaction is successful or not.

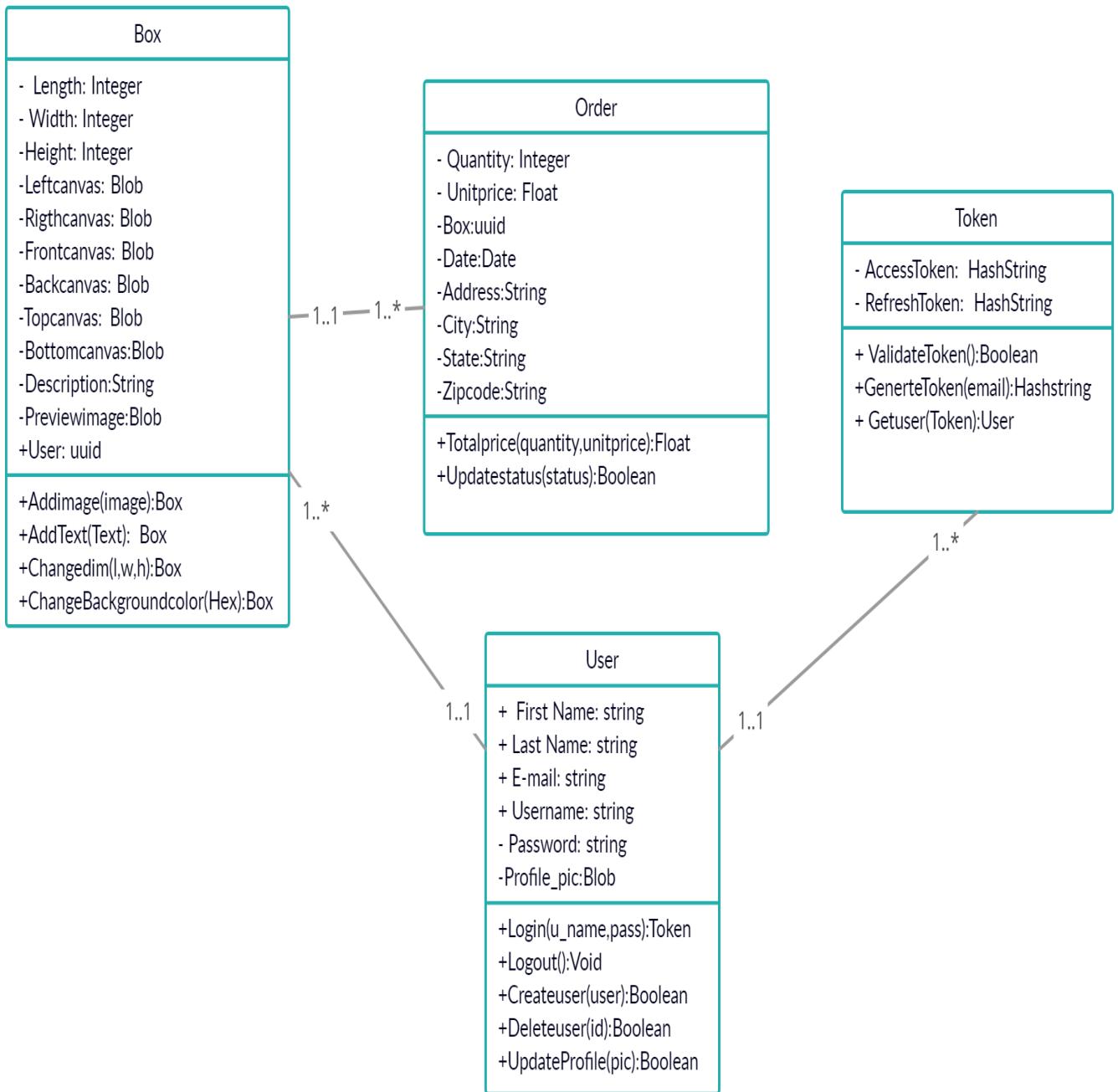
**R5.1 Customer profile: Information should be display, deleted & updated**

- 1)**Pre-condition:** User must be logged in.
- 3) **Input:** Details to be updated.
- 4) **Output:** Updated user information.

**R6.1 Contact Us:** User can send us any query related to the box design and order.

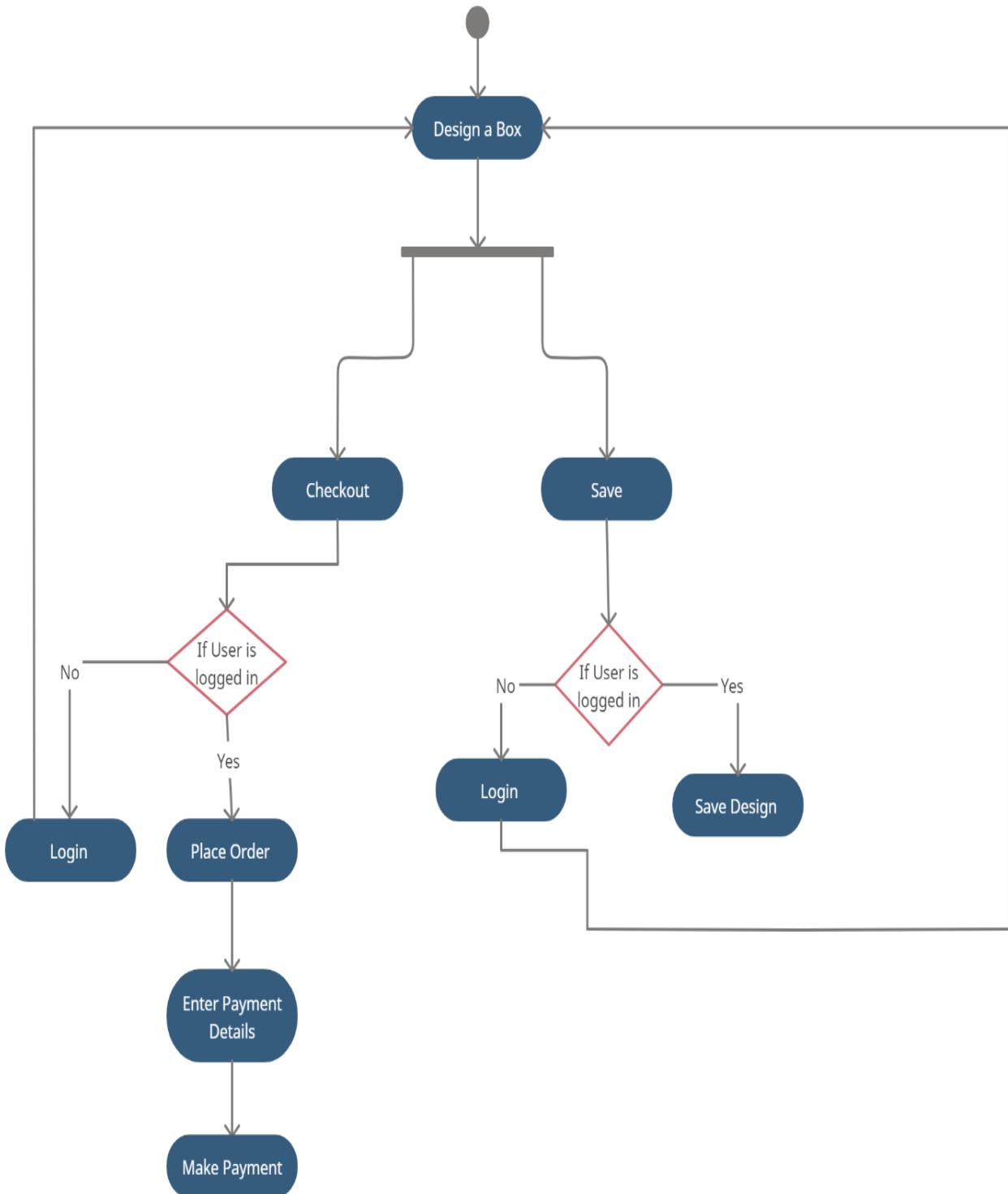
1. **Input:** Enter name, e-mail address, message, subject and clicks on send message.
2. **Processing:** Message is pushed into the message queue.
3. **Output:** Email is sent to admin.

## 4.2 CLASS DIAGRAM



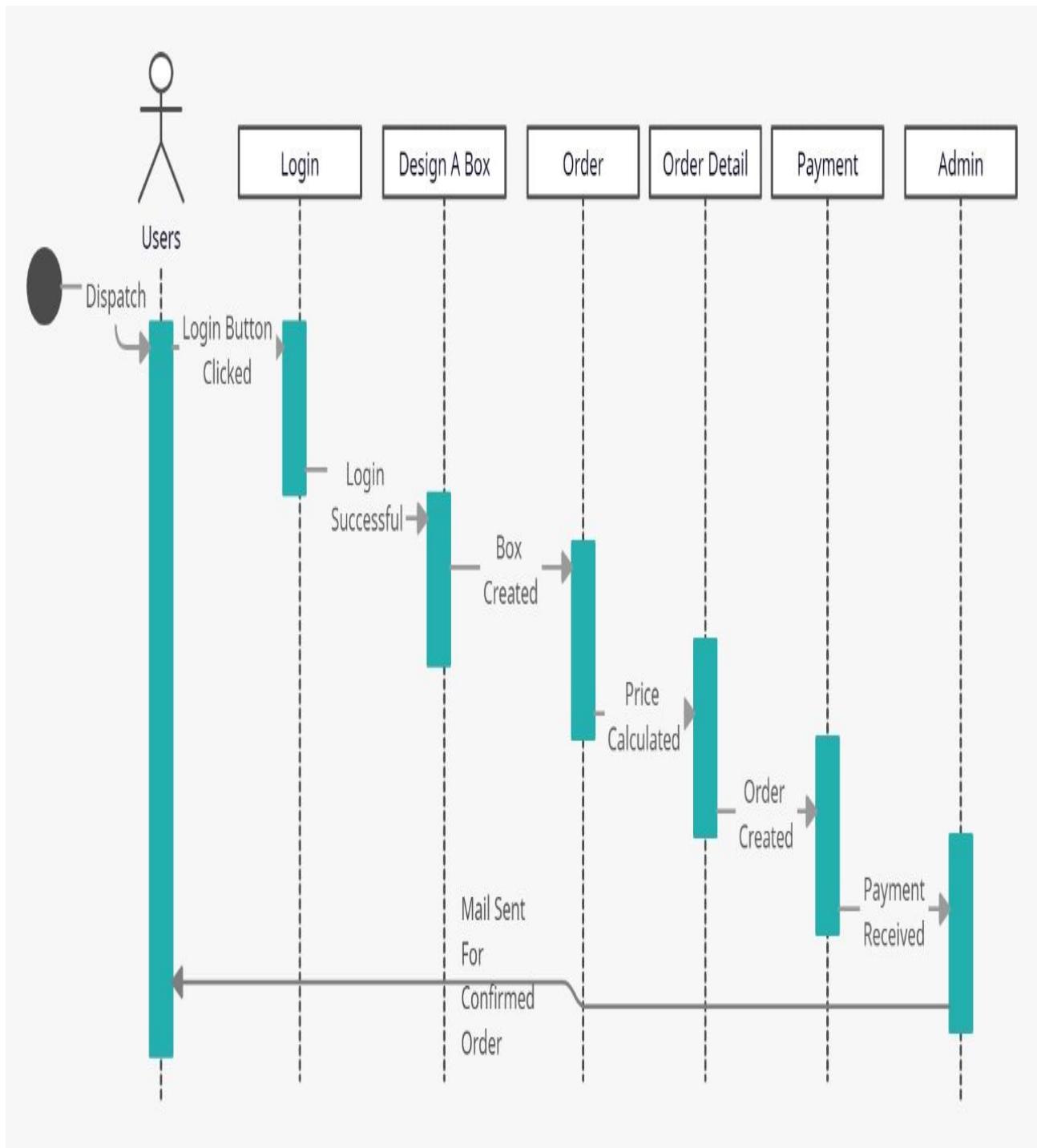
**Fig. 4.2 Class Diagram**

### 4.3 ACTIVITY DIAGRAM



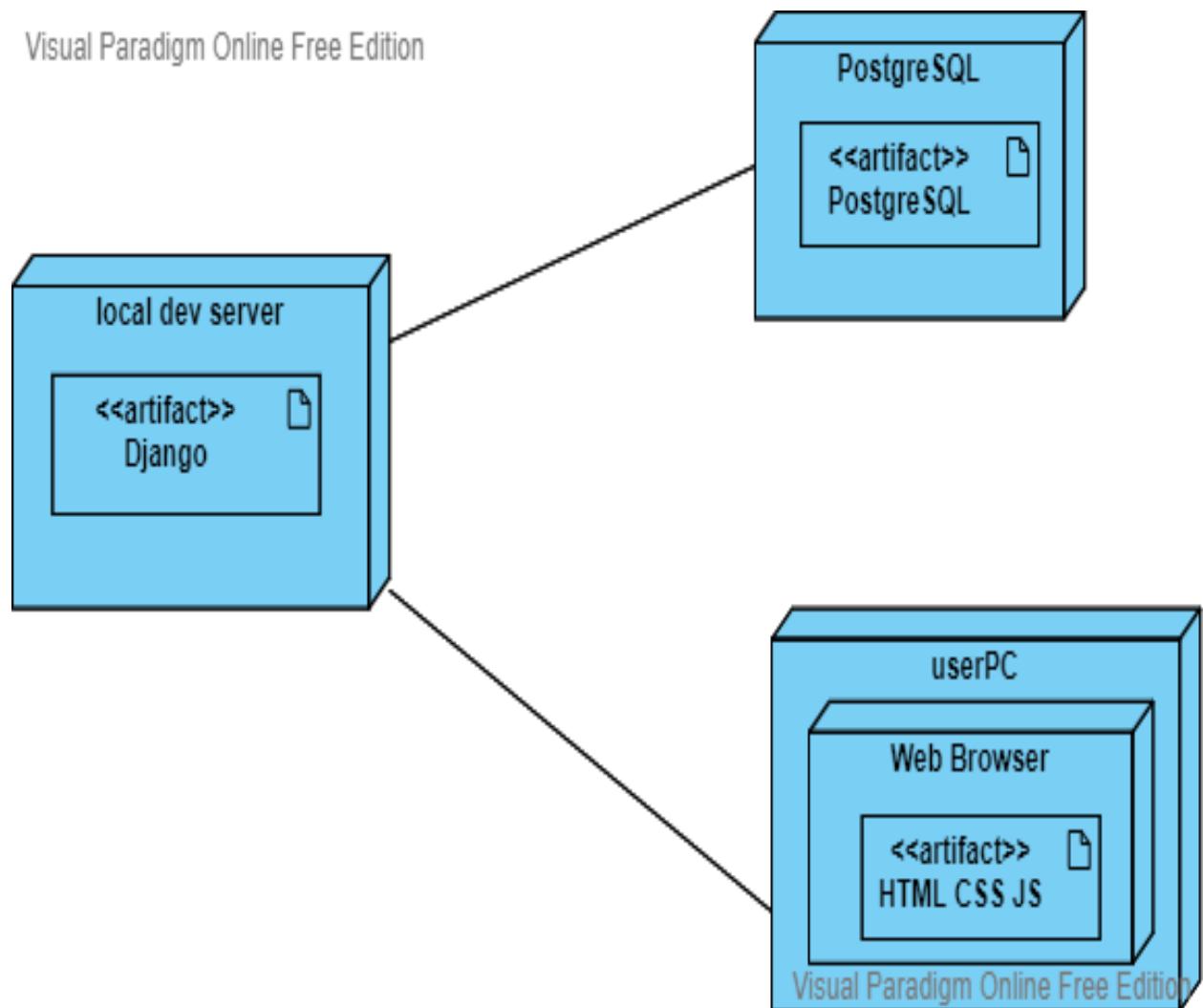
**Fig. 4.3 Activity Diagram**

#### 4.4 SEQUENCE DIAGRAM



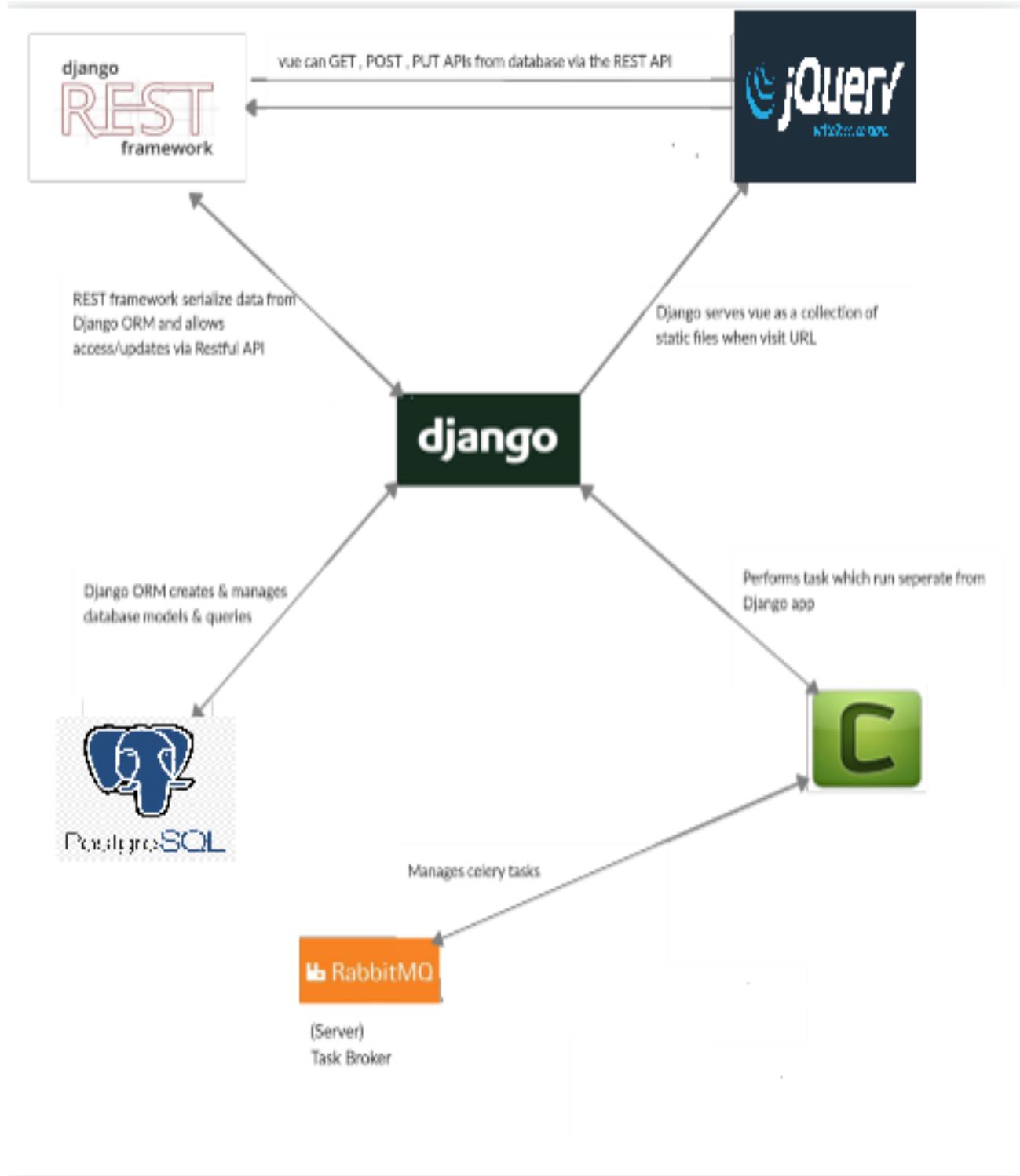
**Fig. 4.4 sequence diagram**

#### 4.5 DEPLOYMENT DIAGRAM



**Fig. 4.5 Deployment Diagram**

## 4.6 SYSTEM DIAGRAM



**Fig. 4.6 System Diagram**

## **5. SYSTEM DESIGN**

### **5.1 SYSTEM ARCHITECTURE DESIGN**

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.

#### **5.1.1 Architectural Design**

The architectural design of a system emphasizes the design of the system's architecture that describes the structure, behavior and more views of that system and analysis.

#### **5.1.2 Logical Design**

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modelling, using an over-abstract (and sometimes graphical) model of the actual system. In the context of systems, designs are included.

#### **5.1.3 Physical Design**

The physical design relates to the actual input and output processes of the system. This is explained in terms of how data is input into a system, how it is verified/authenticated, how it is processed, and how it is displayed. In physical design, the following requirements about the system are decided.

- Input requirement
- Output requirements
- Storage requirements
- Processing requirements
- System control and backup or recovery

User Interface Design is concerned with how users add information to the system and with how the system presents information and responses back to them. Data Design is concerned with how the data is represented and stored within the system. Finally, Process Design is concerned with how data moves through the system, and with how and where it is validated, secured and/or transformed as it flows into, through and out of the system. At The end of the system design phase, documentation describing the three sub- tasks is produced and made available for use in the next phase

## **6. IMPLEMENTATION PLANNING**

### **6.1 IMPLEMENTATION ENVIRONMENT**

For Implementation I have User:

1. Django Framework
2. Python As Programming Language

During the complete implementation we have worked on Visual Studio Code (VSCode). Visual Studio Code is a free source-code GUI (Graphical User Interface) editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. It also provides extension to work in a team with a multi user experience. Even compiling and deploying of the application was done easily with the help of VSCode.

Django web-framework has been used for development of the site. It provides various features for integrating python-based applications and also provides a database to store, update, create and delete records.

### **6.2 PROGRAM/MODULE SPECIFICATION**

Design a Box (Customer):

- Create box: adding new box.
- Edit box: edit any box.
  - Add Image,
  - Add Text,
  - Add Background Color,
  - Change Material,
  - Change Dimension,
  - Change Quantity.
- Delete box: delete any box.

Admin User:

- Update Order Status.
- Delete Order.
- Stock handling.

Customer Order:

- Make Order: To place an order.
- Make Payment: To make payment.

Customer Profile:

- Edit Profile Picture.

Contact Us:

- Generate a query.

### 6.3 CODING STANDARDS

We have followed PEP: 8 standards which is defined by developers of Python programming. Coding Standards contribute to an improved comprehension of source code. Perhaps one of the most influential aids to understanding the logical flow of an application is how the various elements of the application are named. A name should tell "what" rather than "how". By avoiding names that expose the underlying implementation, which can change, you preserve a layer of abstraction that simplifies the complexity. Naming Conventions make programs more understandable by making them easier to read. They can also give information about the function of the identifier – for example, whether it's a constant, class, etc. which can be helpful in understanding the code. All code should be well commented. All procedures and functions should begin with a comment to explain what the function/procedure performs.

- Good and meaningful comments make code more maintainable.
- Do not write comments for every line of code and every variable defined.

Write comments wherever required. But good readable code will require very less comments. If all the variables and methods names are meaningful, that would make the code very readable and will not need more comments. To make the intents identify correctly the precise and correct language should be used. Reasons for using the coding standards are –

- Uniform distribution
- Sound understanding
- Encourages Good programming skills

We have followed four spaces for indentation which is followed by Django developers. In Django template code, put one (and only one) space between the curly brackets and the tag contents. In Django views, the first parameter in a view function should be called **request**. Field names should be all lowercase, using underscores instead of camelCase. The class Meta should appear after the fields are defined, with a single blank line separating the fields and the class definition. Remove import statements that are no longer used when you change code [4].

## **7. TESTING**

### **7.1 TESTING PLAN**

Once code has been generated, software must be tested to uncover as many errors as possible before delivery to the customer. Our goal is to design a series of test cases that have a high likelihood of finding errors. Software testing techniques provide systematic guidance for designing tests that (1) exercise the internal logic of software components, and (2) exercise the inputs and outputs domains of the program to uncover errors in program function, behavior and performance. We will list out all possible cases to test our modules and whole software. We will first perform the unit testing and then system testing.

### **7.2 TESTING STRATEGY**

During early stages of testing, a software engineer performs all tests. However, as the testing process progresses, testing specialists may become involved. Reviews and other activities can and do uncover errors, but they are not sufficient. Every time the program is executed, the customer tests it! Therefore, you have to execute the program before it gets to the customer with the specific intent of finding and removing all errors. In order to find the highest possible number of errors, tests must be conducted systematically and test cases must be designed using disciplined techniques.

Testing Objective:

- 1- Testing is a process of executing a program with the intention of finding an error.
- 2- A good test case is one that has a high probability of finding an as-yet undiscovered error.
- 3- A successful test is one that uncover an as-yet undiscovered error.

#### **7.2.1 Unit Testing**

Unit testing is a software development process in which the smallest testable part of an application, called units, are individually scrutinized for proper operation. Unit testing is often automated but it can also be done manually. This testing mode is a component of Extreme Programming (XP), a pragmatic method of software development that takes a meticulous approach to building a product by means of continual testing and revision.

Unit testing involves only those characteristics that are vital to the performance of the unit under test. This encourages developer to modify the source code without immediate concerns about how such changes might affect the functioning of the units or the program as a whole. Once of whole of the units in a program have been found to be working in the most efficient and error free manner possible, larger components of the program can be evaluated by mean

## **TESTING**

of integration testing. I tested each single part of the entire application. I tested each and every module individually.

### **7.2.2 Sub System Testing**

After testing each unit, we move on to larger units called sub system. In subsystem testing I tested the whole user side as one system. On the user side all the modules like dashboard, API, etc. were tested together to see if there was any error or bug found.

### **7.2.3 System Testing**

After testing all the sub-system, it is time to test the whole system. System testing of software is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. While testing the whole system I found many errors like the data mining delays leading to difficulties in inputs. I solved it by making appropriate changes in the duration of data mining as well as changed its properties. I worked on each error and exception that I got while testing and most of them are removed or made such correction that it will not happen again.

- 1- Recovery Testing: It is a system test that forces the software to fail in a variety of ways and verifies that recovery is properly performed.
- 2- Security Testing: It attempts to verify that protection mechanisms built into a system will, in fact, protect it from improper penetration.
- 3- Performance Testing: It is designed to test the run-time performance of software within the context of an integrated system performance testing occurs throughout all steps in the testing process.

### **7.2.4 Acceptance Testing**

Acceptance testing can be conducted by the end user, customer, or client to validate whether or not to accept the product. Acceptance testing may be performed as part of the hand-off process between any two phases of development. The acceptance test suite is run again the supplied input data or using an acceptance test script to direct the tester. Then the results obtained are compared with the expected results. If there is a correct match for every case, the test suite is said to pass.

### 7.3 TESTING METHODS

The verification activities fall into the category of static testing. During static testing, you have a checklist to check whether the work you are doing is going as per the set standards of the organization. These standards can be for coding, integrating and deployment. Reviews, Inspections and Walkthroughs are static testing methodology. Dynamic testing involves working with the software giving input values and checking if the output is as expected. These are the validation activities. Unit test, integration test, System and acceptance tests are few of the dynamic testing methodologies.

#### Alpha & Beta Testing

The alpha test is conducted at the developer's site by a customer. The software is used in a natural setting with the developer "looking over shoulder" of the user and recording errors and usage problems. Alpha test is conducted in a controlled environment. The beta testing is conducted at one or more customer site by the end-user of the software. Unlike alpha testing, the developer is generally not present. Therefore, the beta test is a "live" application of the software in an environment that cannot be controlled by the developer.

#### Black Box Testing

Also known as functional testing. A software testing technique where by the internal working of the item being tested are not known by the tester. For example, in a black box test on software design the tester only knows the inputs and what the expected outcomes should be and not how the program arrives at those outputs. The tester does not ever examine the programming code and does not need any further knowledge of the program other than its specification.

#### White Box Testing

Also known as glass box, structural, clear box and open box testing. A software testing technique whereby explicit knowledge of the internal workings of the item being tested are used to select the test data. Unlike black box testing, white box testing uses specific knowledge of programming code to examine outputs. The test is accurate only if the tester knows what the program is supposed to do. He or she can then see if the program diverges from its intended goal.

#### Design of Test Cases

To minimize the number of errors in software, a rich variety of test design methods have evolved for software. These methods provide the developer with a systematic approach to testing. More important, methods provide a mechanism that can help to ensure the completeness of test and provide the highest likelihood for uncovering errors in software.

## TESTING

An engineering product can be tested in one of the two ways:

(1) Knowing the specified function that product has been designed to perform, tests can be conducted that demonstrate each function is fully operational while at the same time searching for errors in each function.

(2) Knowing the internal workings of a product, tests can be conducted to ensure that “all gear mesh”, that is, internal oppression are performed according to specifications and all internal components have been adequately exercised. Here are the test cases that we had made for our application.

### 7.4 TEST CASES

#### 7.4.1 SIGNUP

Testcase	Test cases description	steps	Input	expected outcome	Pass/Fail
Flexpack_1	Verify that signup page contains Text "Flexpack", Username ,firstname ,lastname , email,passwo rd & Button of signup.	1). Open the browser & goto flexpack signup 2).Check if sign up page contains Text "Flexpack", Username ,firstname ,lastname , email,password & Button of signup.		Sign up page should contain contains Text "Flexpack", Username ,firstname ,lastname , email,password & Button of signup.	Pass
Flexpack_2	Verify that all the fields such as Username ,firstname ,lastname , email,passwo rd have a valid placeholder.	1).Open the browser & goto Flexpack signup 2) Check if all the filed have valid placeholder.		All the fields such as1Username ,firstname ,lastname , email,password have valid placeholder.	Pass
Flexpack_3	Verify that clicking on next button after entering	1).Open the browser & goto Flexpack signup 2. Enter data in	UserName=jiya email- jiyaver34@gmail.com	If user click on "Signup "button after entering all the required field,	Pass

## TESTING

	all the mandatory fields, redirect on other page.	all the field such as Username ,firstname ,lastname , email,password. 3. Click on "Signup" button and check if it redirect or not.	password=jiya123 45@	it should redirect the data successfully on other page.	
Flexpack_4	Verify that entering blank spaces on any field lead to validation error.	1.Open the browser & goto Flexpack signup 2. Enter blank space on any field and click on "Signup" button.		There should be validation error if user enters blank spaces on any field	Fail
Flexpack_5	Verify the case sensitivty of user name. (Usually name field should not follow case sensitivty)	1).Open the browser & goto Flexpack signup 2)enter capital alphabet	Name=jiya	Name field should not follow the case sensitivty.	fail
Flexpack_6	verify email field contain "@gmail.com" syntax	1).Open the browser & goto Flexpack signup 2)enter email address without "@gmail.com"	email="gugsgyug dahsgdgjhgmail"	There should be validation error if user enters email without "@gmail.com"	fail
Flexpack_7	verify email field not contain special symbol such as "&","@", "\$", "#" etc	1).Open the browser & goto Flexpack signup 3)Enter email address with contain special symbol such as "&","@", "\$", "#" etc	email="gugsgyu# \$122@gmail.com "	There should be validation error if user enters email which contain with contain special symbol such as "&","@", "\$", "#" etc	Pass
Flexpack_8	Verify that after enter all fields such as Username ,firstname	1).Open the browser & goto Flexpack signup 2)click on signup button		There should page redirect on login page	Pass

## TESTING

	,lastname , email,passwo rd than clicking the signup button which redirect on login page				
Flexpack_9	Verify that the password is in encrypted form when entered.	1).Open the browser & goto Flexpack signup 2) Verify that password is in encrypted form.		Password should be in encrypted form.	Pass
Flexpack_10	Verify the validation of email field by entering incorrect email id.	1)Open the browser & goto Flexpack signup 2) Verify the validation of email field by entering incorrect email id.		User should not be able to submit when enter invalid email id.	fail

**Table 7.4.1 Sign Up**

## TESTING

### 7.4.2 LOGIN

Testcase	Test cases description	steps	Input	expected outcome	Pass/Fail
Flexpack_1	Verify that Login page contains Text "Flexpack", Username , password ,goggle signup button, signup link & Button of Login.	1).Open the browser & goto flexpack Login 2).Check if sign up page contains Text "Flexpack", Username ,password , password ,goggle signup button, signup link & Button of Login.		Sign up page should contain contains Text "Flexpack", Username ,password, password ,goggle signup button, signup link & Button of Login.	Pass
Flexpack_2	Verify that all the fields such as Username ,password have a valid placeholder.	1).Open the browser & goto Flexpack Login 2) Check if all the filed have valid placeholder.		All the fields such as Username „password have valid placeholder.	Pass
Flexpack_3	Verify that clicking on next button after entering all the mandatory fields, redirect on other page.	1).Open the browser & goto Flexpack Login 2). Enter data in all the field such as Username ,password . 3). Click on "Login" button and check if it rediect or not.	UserName=jiya email-jiyaver34@gmail.com password=jiya12345@	If user click on "Login "button after entering all the required field, it should redirect the data successfully on other page.	Pass
Flexpack_4	Verify that entering blank spaces on any field lead to validation error.	1.Open the browser & goto Flexpack Login 2. Enter blank space on any field and click on "Login" button.		There should be validation error if user enters blank spaces on any field	Fail
Flexpack_5	Verify that the password is in encrypted	1).Open the browser & goto Flexpack Login 2) Verify that		Password should be in encrypted form.	Pass

## TESTING

	form when entered.	password is in encrypted form.			
Flexpack_6	Enter some different username & password than in registration	1)Open the browser & goto Flexpack Login 2)Check if register data can only login		Only register user can only login	fail
Flexpack_7	Check by entering wrong password so it matches with username	1)Open the browser & goto Flexpack Login 2)Check username & password both matches		There should correct password	fail
Flexpack_8	Check by entering wrong Username so it matches with password	1)Open the browser & goto Flexpack Login 2)Check username & password both matches		There shoult correct username	fail
Flexpack_8	Verify If Forgot password link is working.	1)Open the browser & goto Flexpack Login 2) Check whether forget link redirect on Forget password		There should be link of forget password which redirect on forget password page	Pass
Flexpack_9	Verify that goggle signup button signup directly	1)Open the browser & goto Flexpack Login 2) Check whether goggle signup button works correctly		There should be signup by goggle id	Pass
Flexpack_10	Verify signup link redirect on signup page	1)Open the browser & goto Flexpack Login 2) Check whether signup link works correctly		There should redirect on signup page	Pass

**Table 7.4.2 Login In**

## TESTING

### 7.4.3 DESIGNBOX

Testcases	Test cases description	steps	Input	Result output	Pass/Fail
Flexpack_1	Verify that DesignBox page contains size, material,design,quantity,reset,select side ,createCustomize size	1).Open the browser & goto flexpack DesignBox 2).Check if Home pasge contains size, material,design,quantity,reset,select side ,createCustomize size		Home page should contain contains size, material,design ,quantity,reset, select side ,createCustomize size	Pass
Flexpack_2	verify image of box is visible	1).Open the browser & goto Flexpack DesignBox 2) Check Box is visible		Box should be visible	Pass
Flexpack_3	verify that in "size" module there is number of size option is given	1).Open the browser & goto Flexpack DesignBox 2).check that by clicking size module list is open of size of box	Select stock size 1)4" x 4" x 2" 2)6" x 6" x 2" 3)7" x 6" x 1" 4)8" x 5" x 3" 5)9" x 6" x 4" 6)9" x 8" x 2" 7)10" x 8" x 4" 8)10" x 9"x1.5" 9)12" x 9" x 2" 10)13"x10"x 5" 11)14"x10"x 4"	there should list is open of size of box	Pass
Flexpack_4	Verify that by click particular size it is applicable on box size	1.Open the browser & goto Flexpack DesignBox 2. check the box size by selecting particular size from given list of size	Select: 7" x 6"x 1"	There should be box of size "7" x 6" x 1" visisble	Pass

## TESTING

Flexpack_5	verify that by "select side" there list is open of different side of box	1).Open the browser & goto Flexpack DesignBox 2)click on list of "select side"	Select side: 1)Interior RightFront 2)Left 3)Back 4)Right 5)Top 6)Bottom 7)Interior Left 8)Interior Right	There should list is open of "select side"	Pass
Flexpack_6	Verify that by clicking particular "side" it is applicable on box side	1).Open the browser & goto Flexpack DesignBox 2) check the box side by selecting particular side from given list of side	select side=back	There should be "back " of box side is visisble	Pass
Flexpack_8	verify that in "material" module there is list of material option is given	1).Open the browser & goto Flexpack DesignBox 2).check that by clicking material module list is open of materiials of box	select your material 1)White Classic matte white cardboard printed with a satin finish.  2)Dreamcoat Premium bright white cardboard. 3)Kraft Natural matte brown cardboard printed with a satin finish.	There should be list is open of materials of box	Pass
Flexpack_9	Verify that by clicking particular "material" it is applicable on box	1).Open the browser & goto Flexpack DesignBox 2)check the box material by selecting particular material from given list of side	Dreamcoat Premium bright white cardboard.	There should be "Dreamcoat material " of box is visisble	Pass

## TESTING

Flexpack_10	verify that in "design" module there is number of design option is given	1)Open the browser & goto Flexpack DesignBox 2).check that by clicking "design" module list is open of different designs of box	Selec design= 1)Select color 2)Add Images 3)Add textRequest 4)dieline	There should be list is open of design of box	Pass
Flexpack_11	Verify that by clicking particular "design" it is applicable on box	1). Open the browser & goto Flexpack DesignBox 2) check the box design by selecting particular design from given list of side	select=Add Images	There should be "ADDIMAGE DESIGN" of box is visisble	Pass
Flexpack_12	Verify that by clicking particular "Quantity" list of quantities are open	1). Open the browser & goto Flexpack DesignBox 2). check that by clicking "Quantity" module list is open of quantity number of box	select quantity Unit price 1) $\geq 10,000$ Request 2) $> 2,000$ Request 3)2,000 4)1,000 5)500 etc	There should be list is open of quantity of box	Pass
Flexpack_13	Verify that by clicking particular "Quantity" it is applicable on box	1).Open the browser & goto Flexpack DesignBox	select Quantity: 4) 1,000	There should be "100 quantity" of box is added	Pass
Flexpack_14	Verify that by clicking "RESET" button all applied features sould be remove	1).Open the browser & goto Flexpack DesignBox 2)check the box by selecting reset is it remove all features		There should remove all property which applied before	Pass
Flexpack_15	Verify that by clicking Checkout button it redirect on correct page	1).Open the browser & goto Flexpack DesignBox 2)check it redirect on payment page		There should redirecting on payment page	Pass
Flexpack_16	verify that there is one summary display is given where all	1).Open the browser & goto Flexpack DesignBox 2)check there is one	Size 4" x 4" x 2" Material	There should one summary display is given where	Pass

## TESTING

	selected property is store there such as size,material,quantity ,unit price, subtotal.	summary display is given where all selected property is store there such as size,material,quantity ,unit price, subtotal.	White Quantity 250 Unit price \$1.13 Subtotal \$282.50	all selected property is store there such as size,material,quantity ,unit price, subtotal.	
Flexpack_17	verify that user can not add value more than 900 and less than 400 in box height text field	1).Open the browser & goto Flexpack DesignBox 2) add details of boxes	size Box height = 1000	Enter height less than 900	Fail
Flexpack_18	verify that user can not add value more than 900 and less than 400 in box height text field	1).Open the browser & goto Flexpack DesignBox 2) add details of boxes	size Box height = 300	Enter height more than 400	Fail
Flexpack_19	verify that user can not add value more than 900 and less than 400 in box width text field	1).Open the browser & goto Flexpack DesignBox 2) add details of boxes	size Box width = 300	Enter width more than 400	Fail
Flexpack_20	verify that user can not add value more than 900 and less than 400 in box width text field	1).Open the browser & goto Flexpack DesignBox 2) add details of boxes	size Box width = 1000	Enter width less than 900	Fail
Flexpack_21	verify user can not be able to checkout without entering each and every details	1).Open the browser & goto Flexpack DesignBox 2) add details of boxes	Add sizes of boxes	There should be validation that enter box size	Fail
Flexpack_22	verify the user can not add quantity .	1).Open the browser & goto Flexpack DesignBox 2) add details of boxes	Not Add quantity of boxes	There should be validation that enter box quantity	Fail

## TESTING

Flexpack_23	verify all the buttons are working in the size page	1).Open the browser & goto Flexpack DesignBox 2) add details of boxes		There should be buttons are working	Pass
Flexpack_24	verify the unit price and subtotal price are rightly calculated	1).Open the browser & goto Flexpack DesignBox 2) add details of boxes		verify unit & subtotal price displayed correctly	Pass

**Table 7.4.3 DESIGNBOX**

## 8. USER MANUAL

### 8.1 SNAPSHOTS OF CUSTOMER SIDE OF THE WEBSITE:

#### 8.1.1 Home Page (customer)

##### 8.1.1.1 Home Page Create custom box

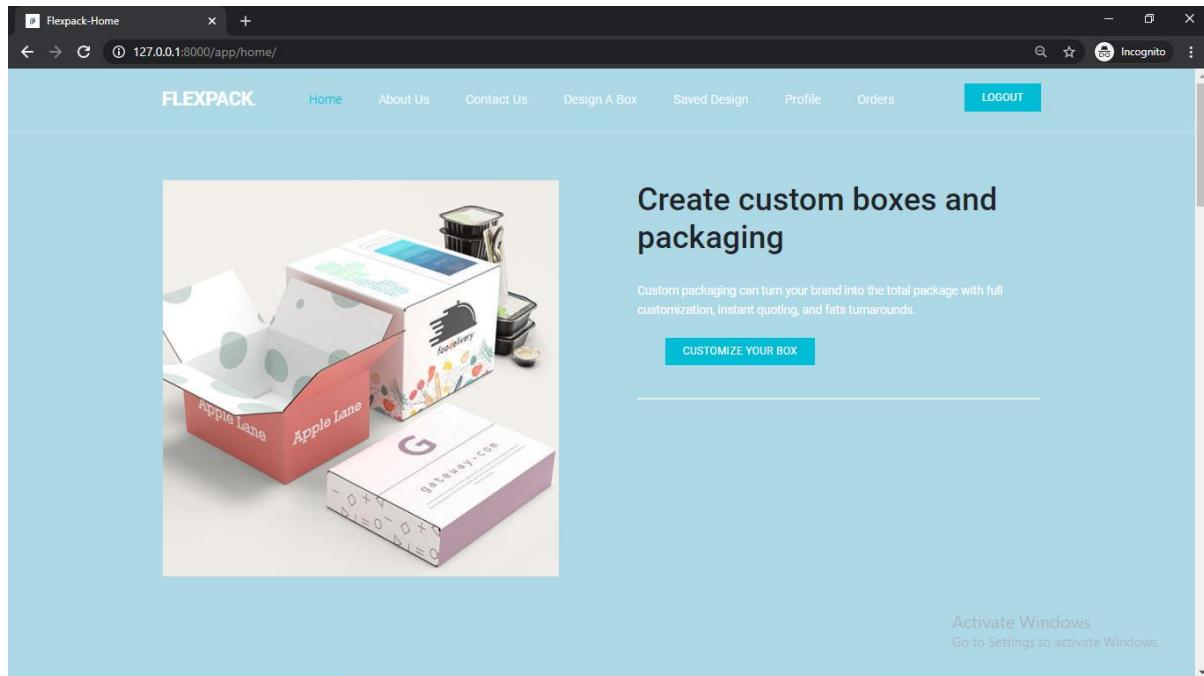
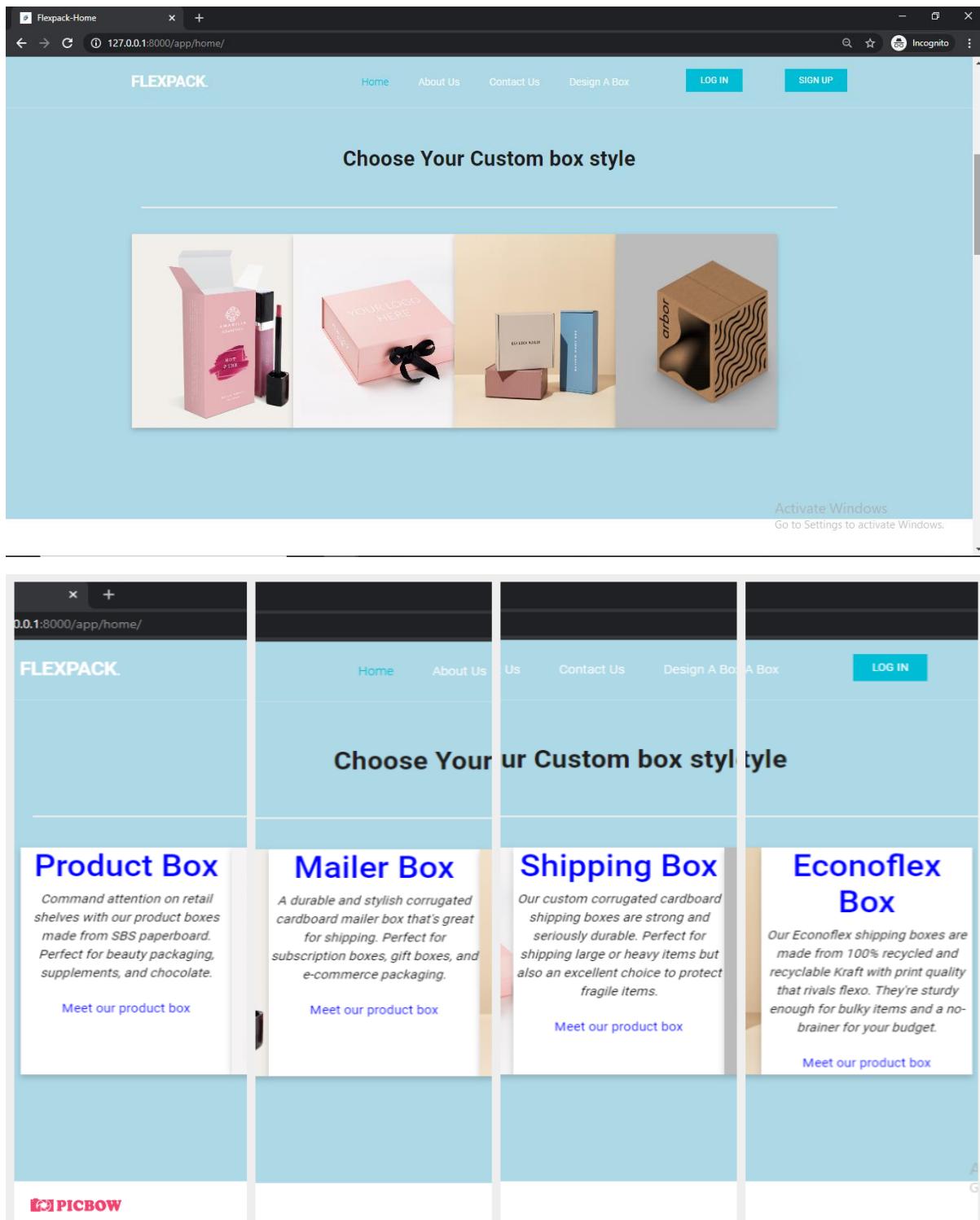


Fig. 8.1.1.1 Home page

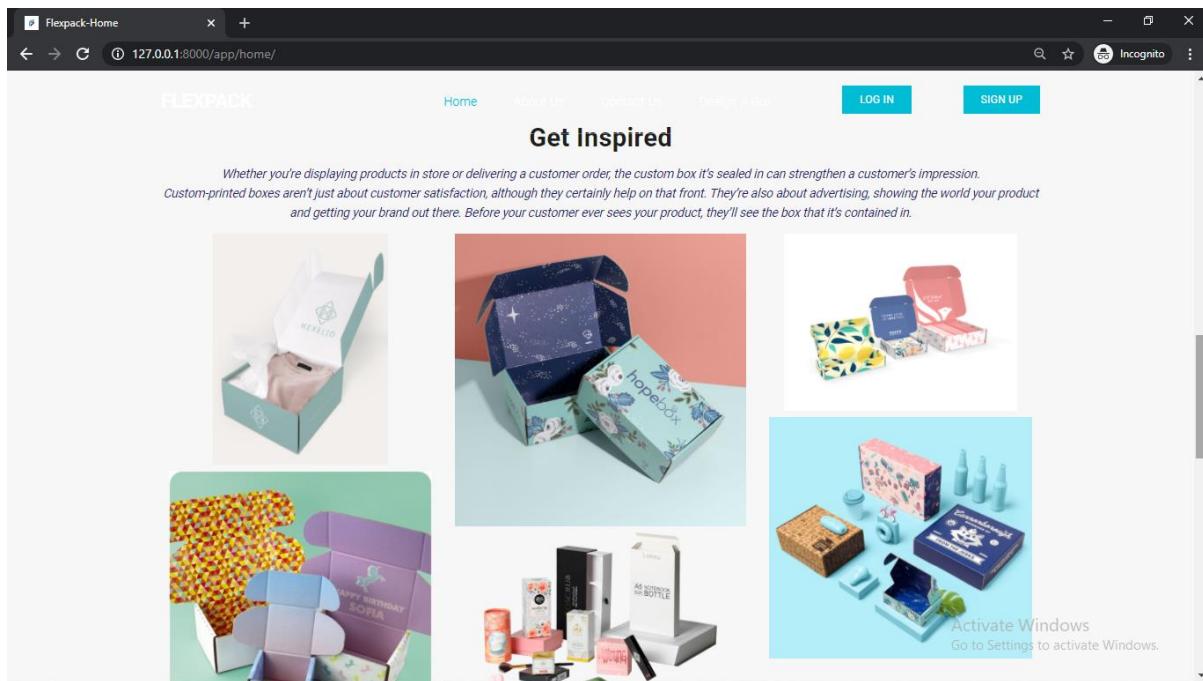
## 8.1.1.2 Choose Style (customer)



**Fig. 8.1.2.1 Choose Style (Customer)**

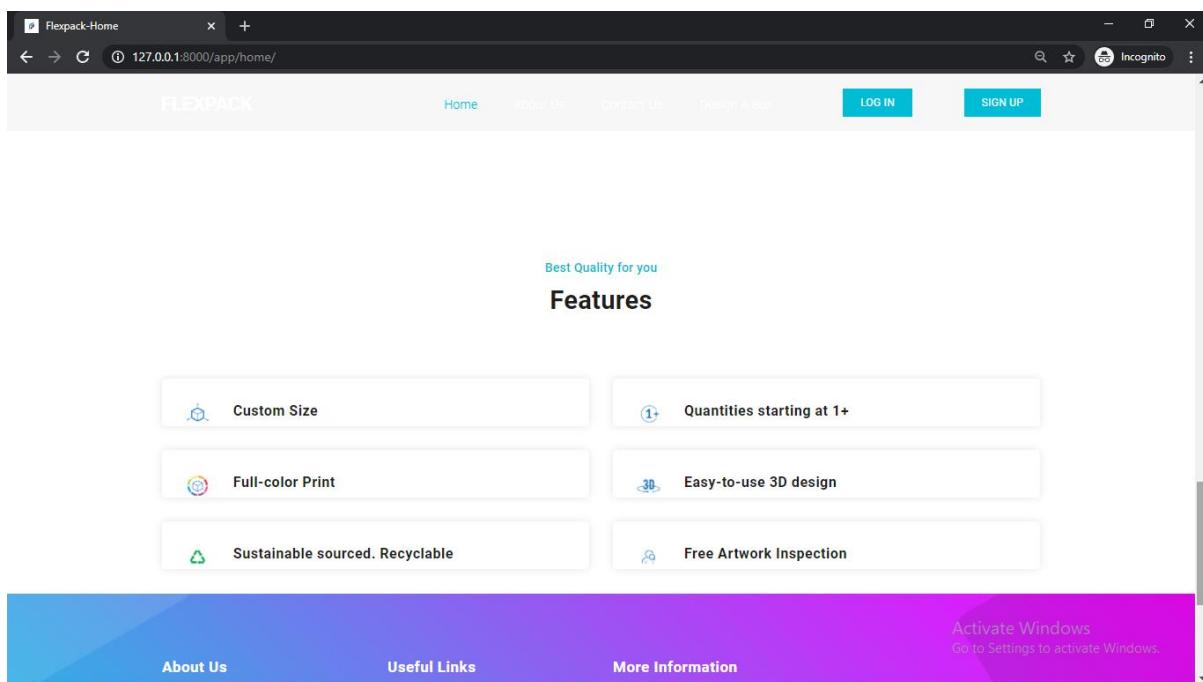
## USER MANUAL

### 8.1.1.3 Get Inspired



**Fig. 8.1.1.3 Get Inspired**

### 8.1.1.4 Feature



**Fig. 8.1.1.4 Feature**

## USER MANUAL

### 8.1.2 Design A Box(customer)

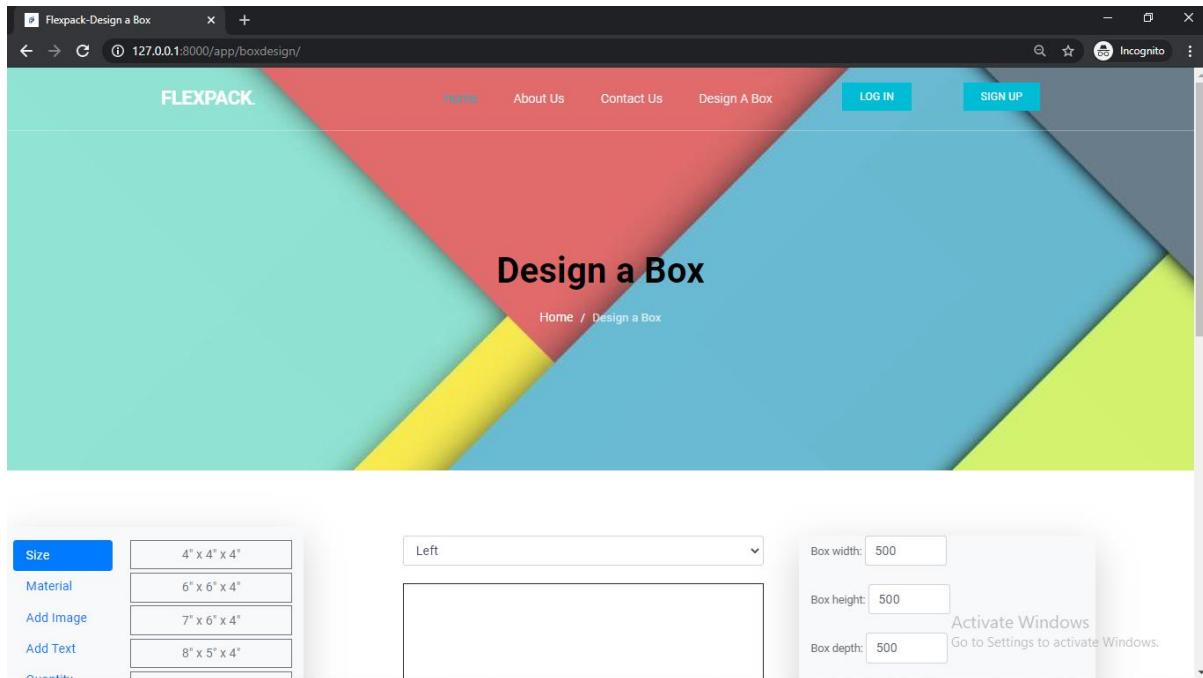


Fig. 8.1.2 Design A box

### 8.1.2.1 Size of the Box

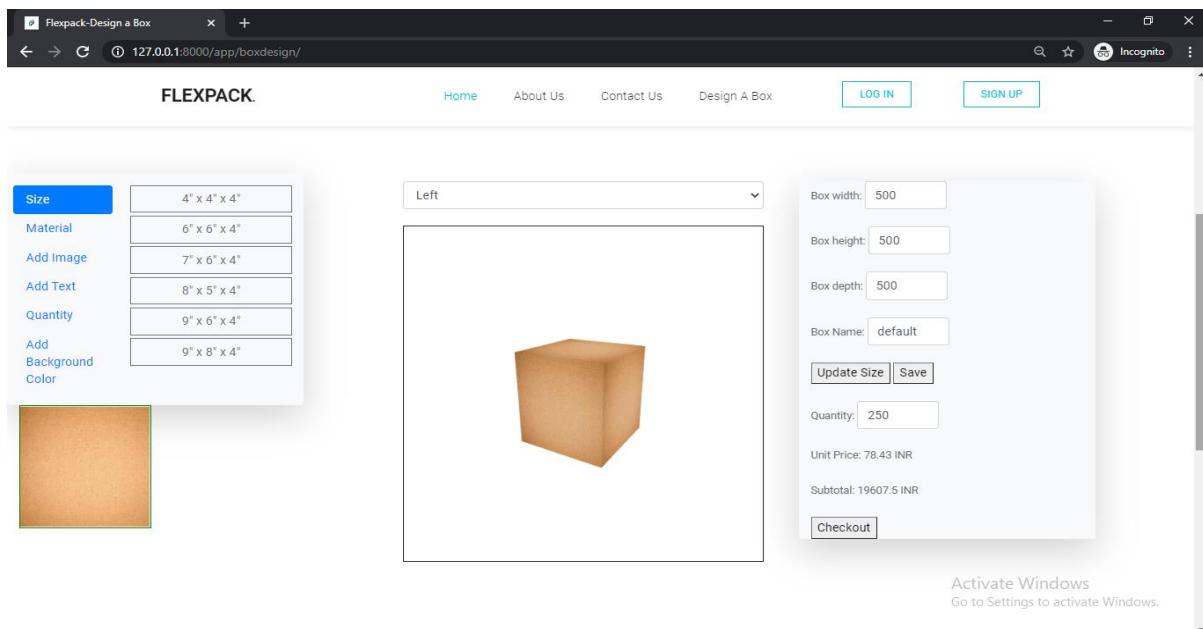


Fig. 8.1.2.1 Size of the Box

## USER MANUAL

### 8.1.2.2 Materials

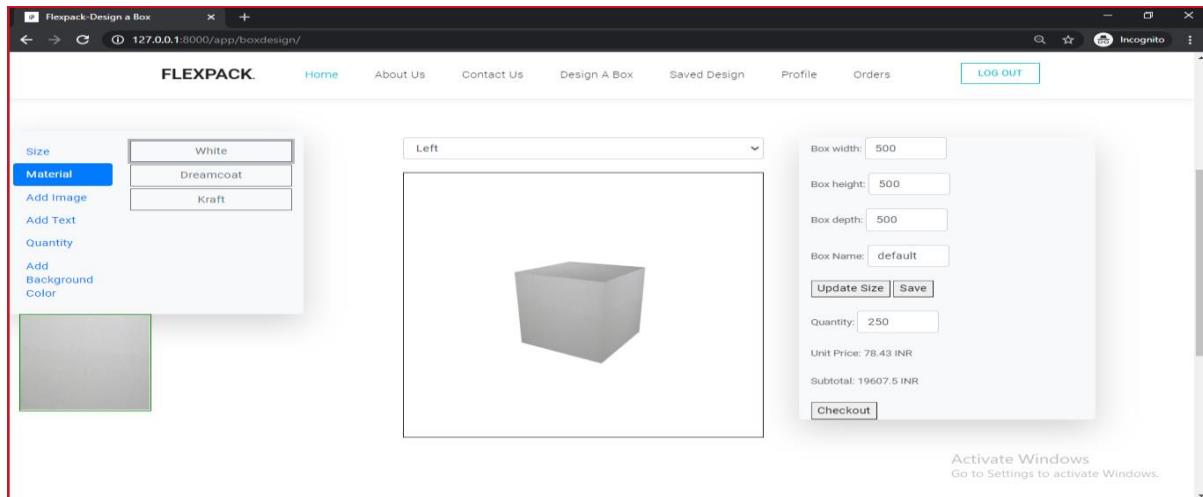


Fig. 8.1.2.2.1 White Board

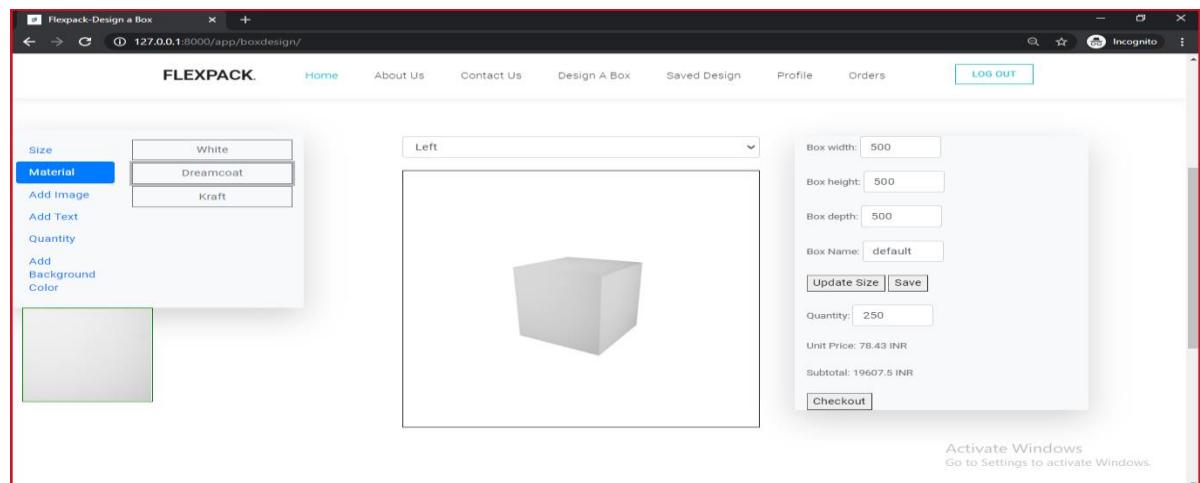


Fig. 8.1.2.2.2 Dreamcoat Board

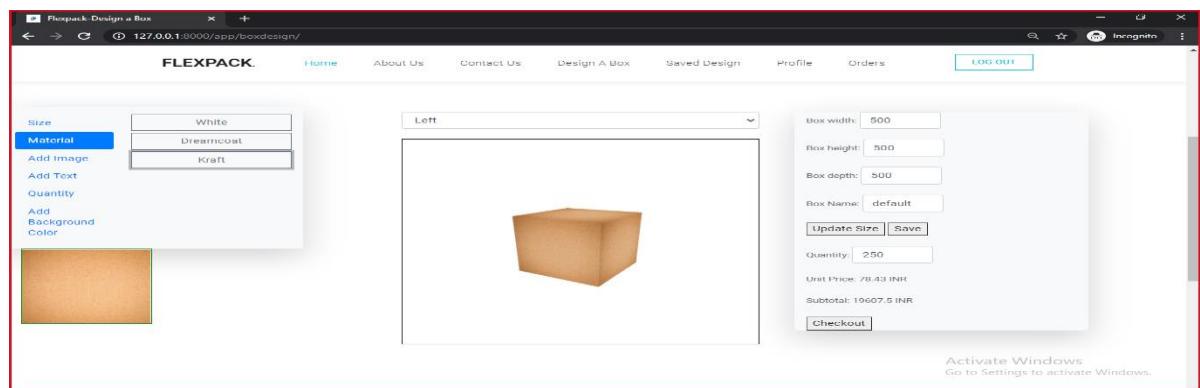
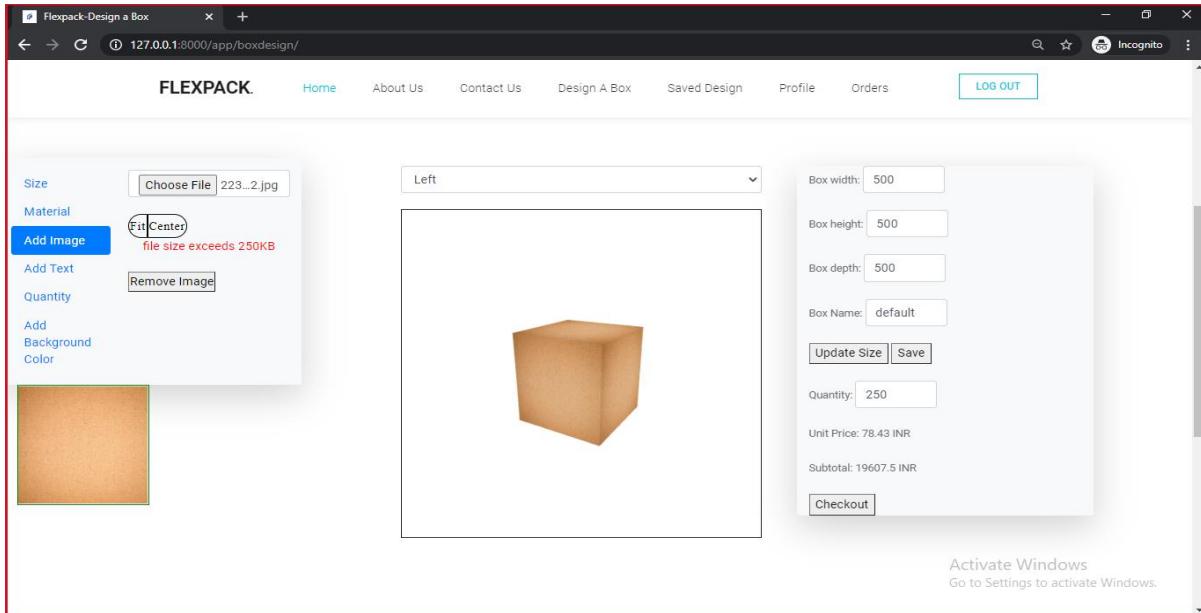


Fig. 8.1.2.2.3 Kraft Board

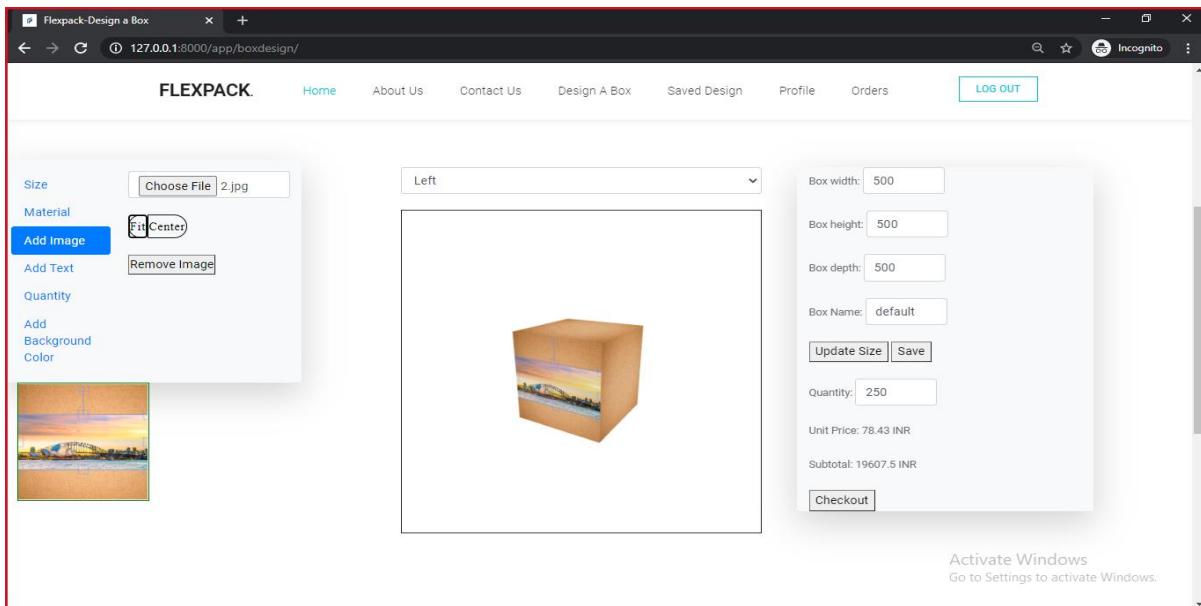
## 8.1.2.3 Add Image: Choose

### 8.1.2.3.1 Choose File validation

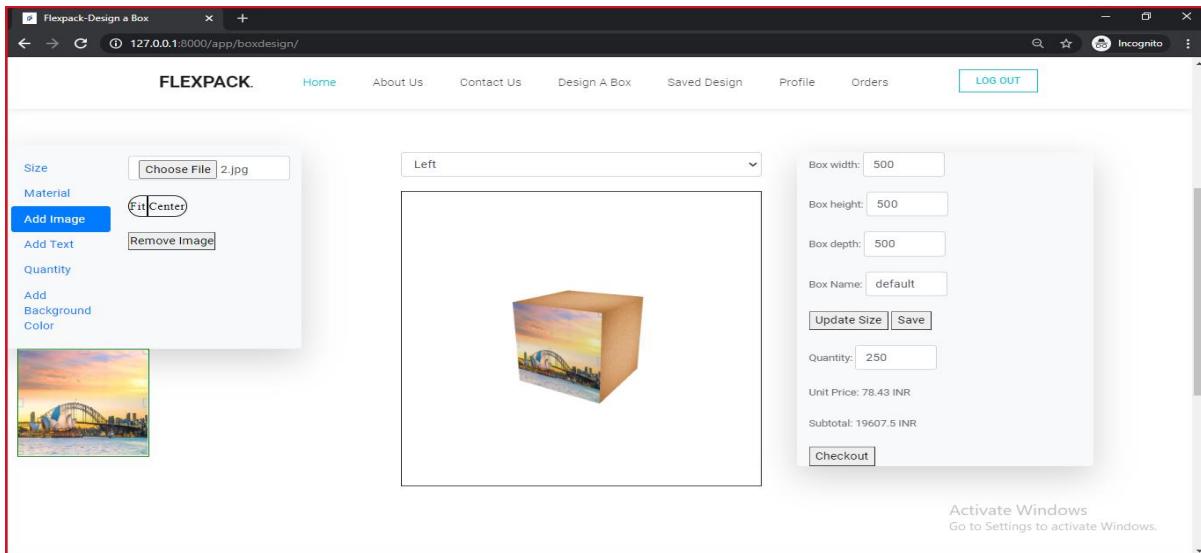


**Fig. 8.1.2.3.1 Choose File validation**

### 8.1.2.3.2 Fit and Cover the image

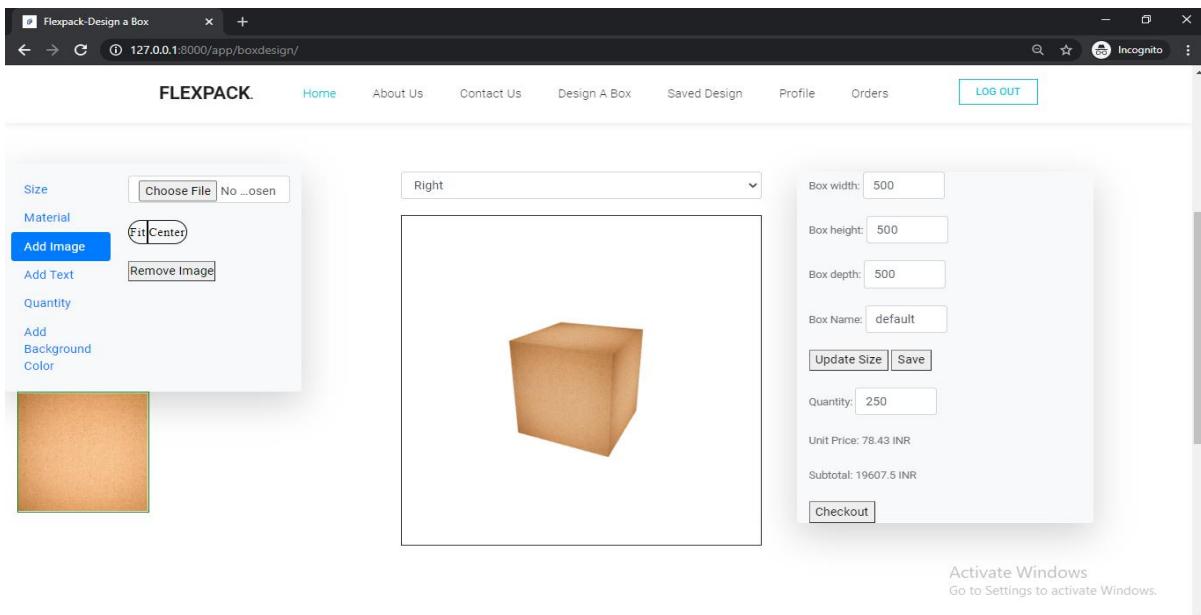


## USER MANUAL



**Fig. 8.1.2.3.2 Fit and Cover the image**

### 8.1.2.3.3 Remove Image



**Fig. 8.1.3.3 Remove image**

## USER MANUAL

### 8.1.3 Add Text

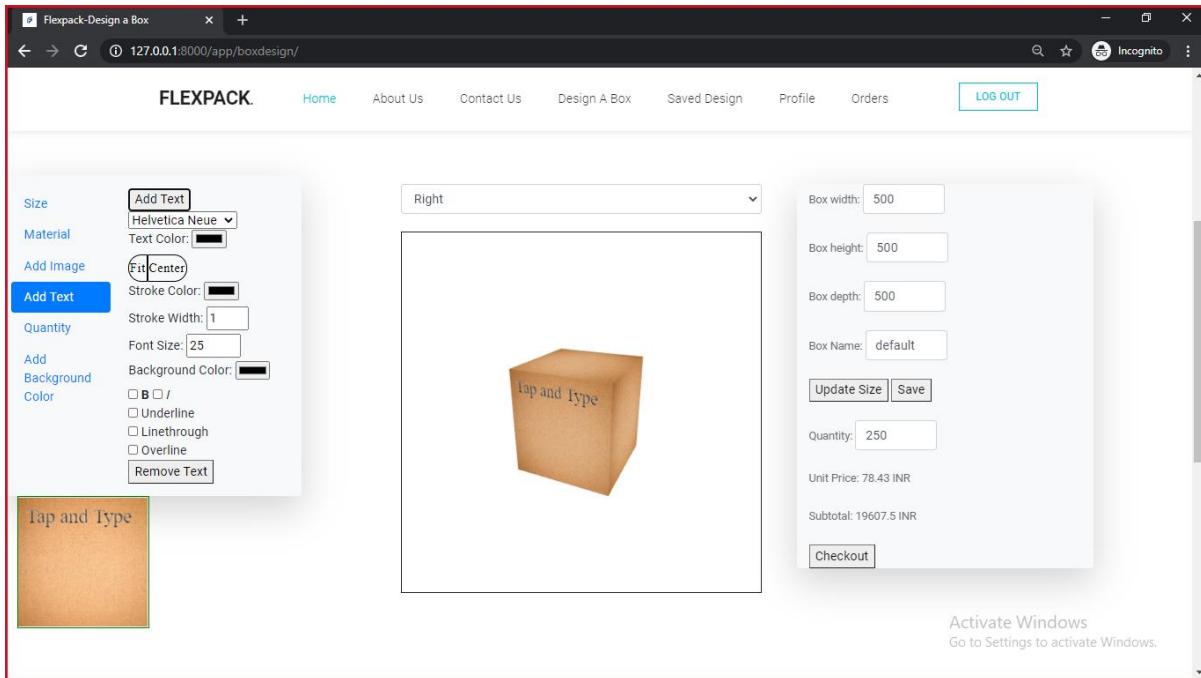


Fig. 8.1.3.1 Add Text

### 8.1.3.2 Text Format

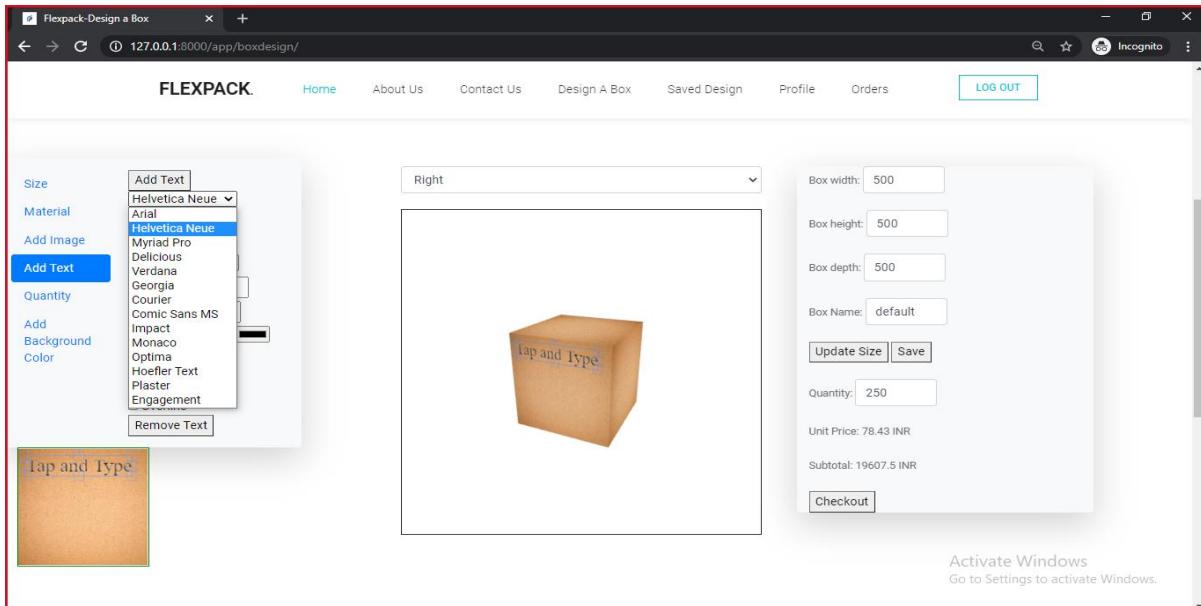


Fig. 8.1.3.2 Text Format

# USER MANUAL

## 8.1.3.3 Text Color

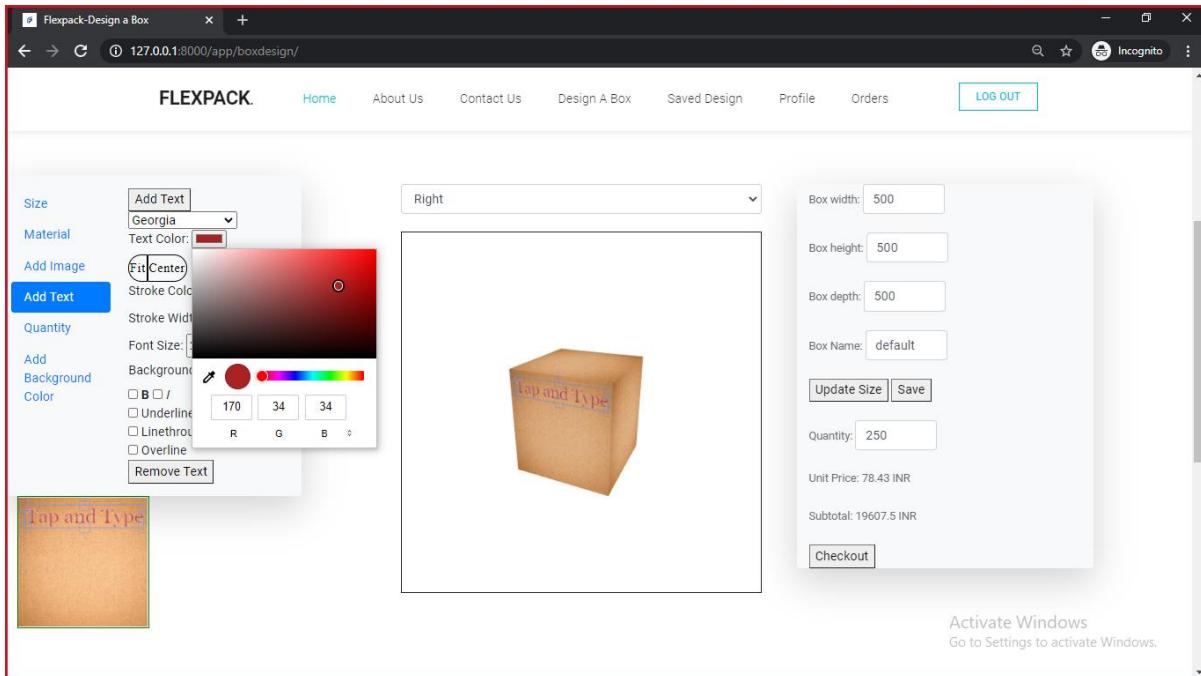


Fig. 8.1.3.3 Text Color

## 8.1.3.4 Text Align

### 8.1.3.4.1 Fit Align

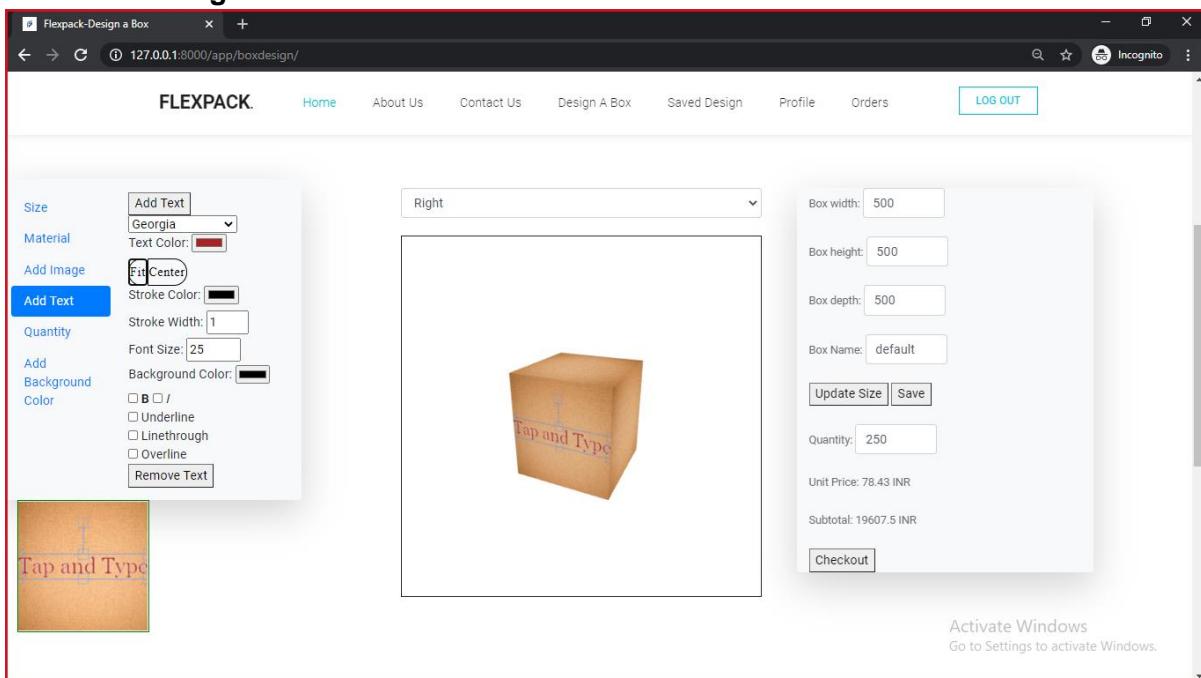


Fig. 8.1.3.4.1 Fit Align

## USER MANUAL

### 8.1.3.4.2 Center Align

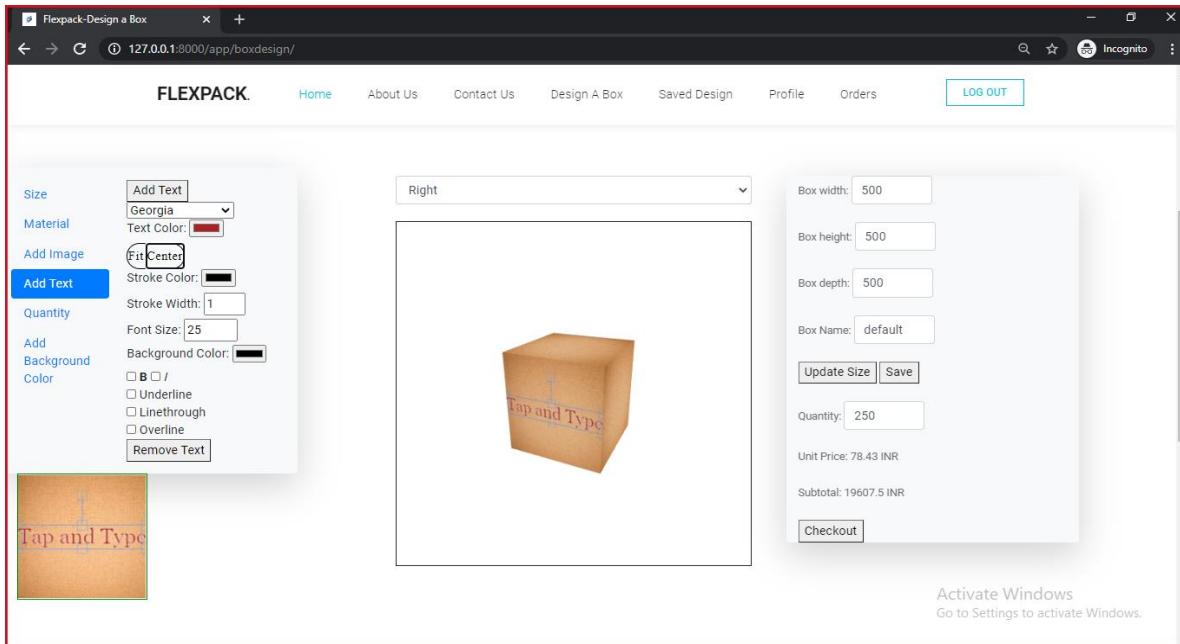


Fig. 8.134.2 Center Align

### 8.1.3.5 Stroke Color

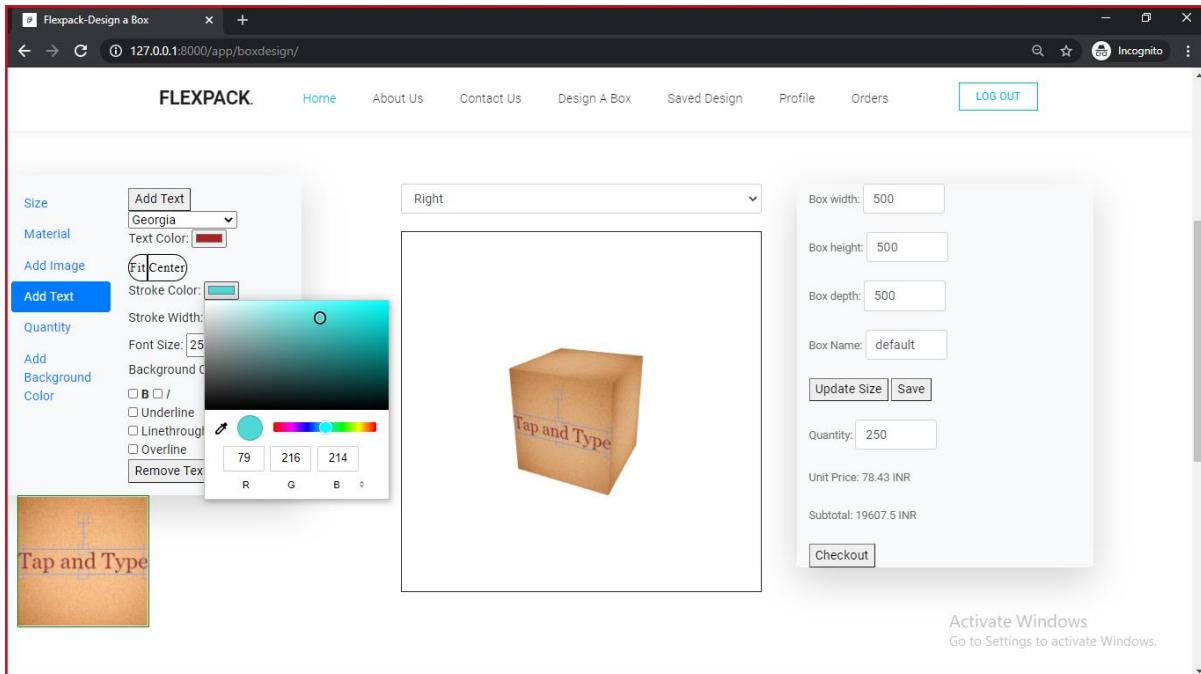
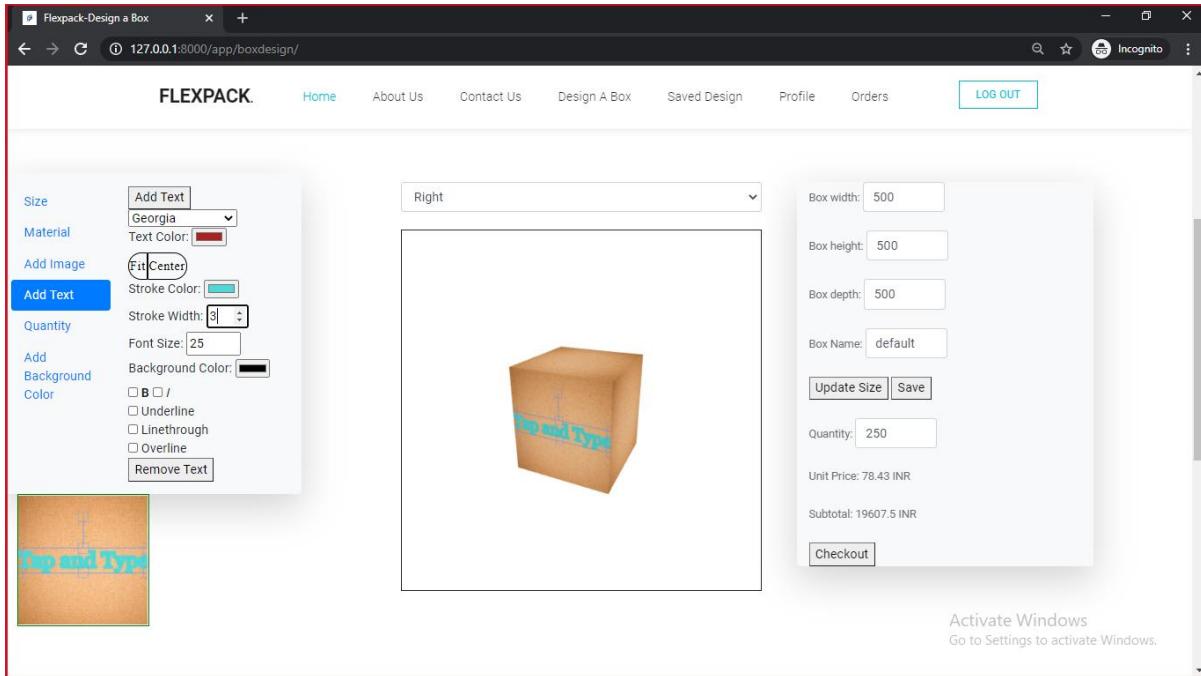


Fig. 8.1.3.5 Stroke Color

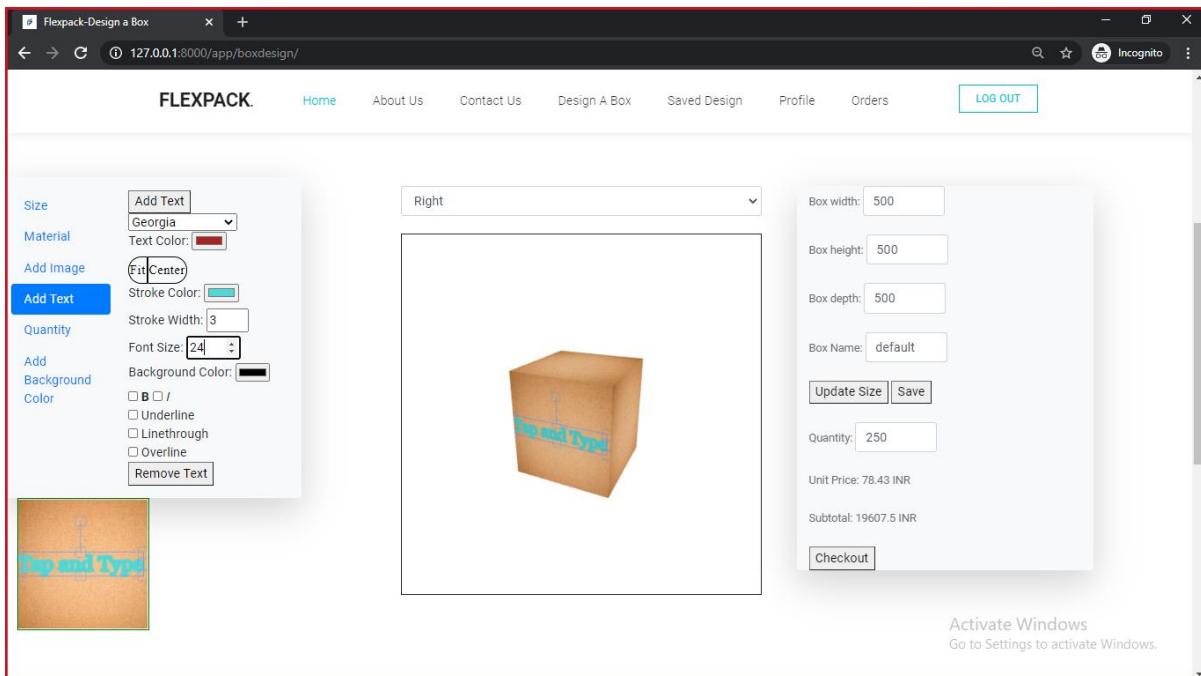
## USER MANUAL

### 8.1.3.6 Stroke Width



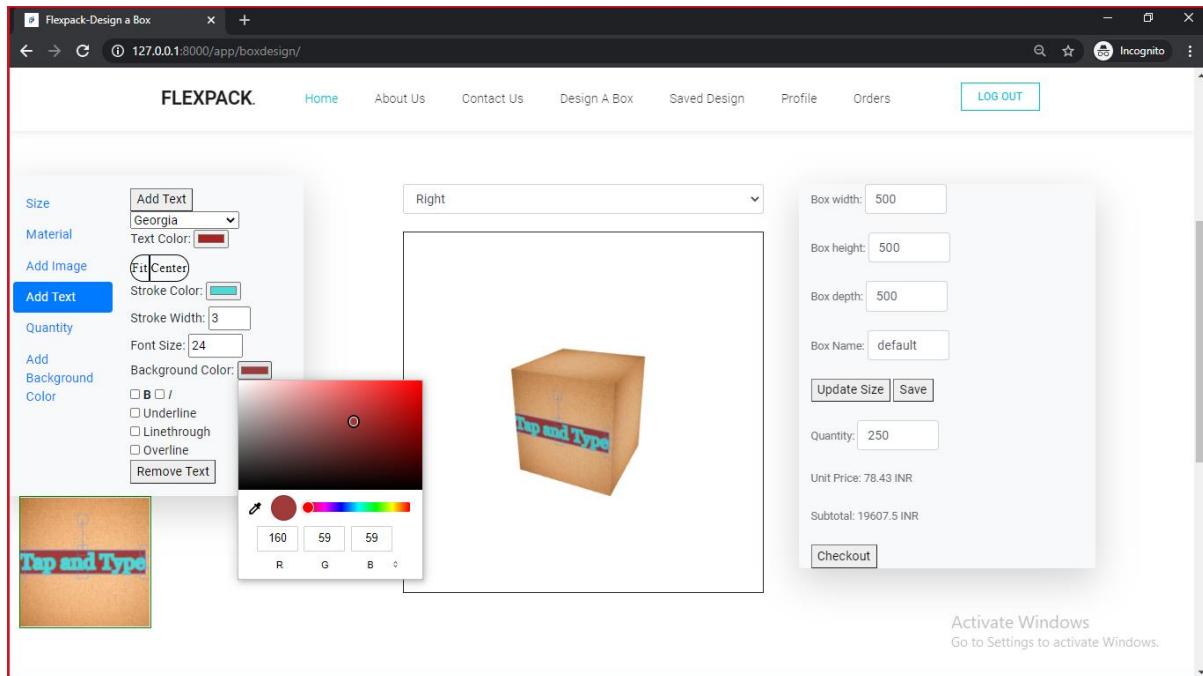
**Fig. 8.1.3.6 Stroke Width**

### 8.1.3.7 Font Size



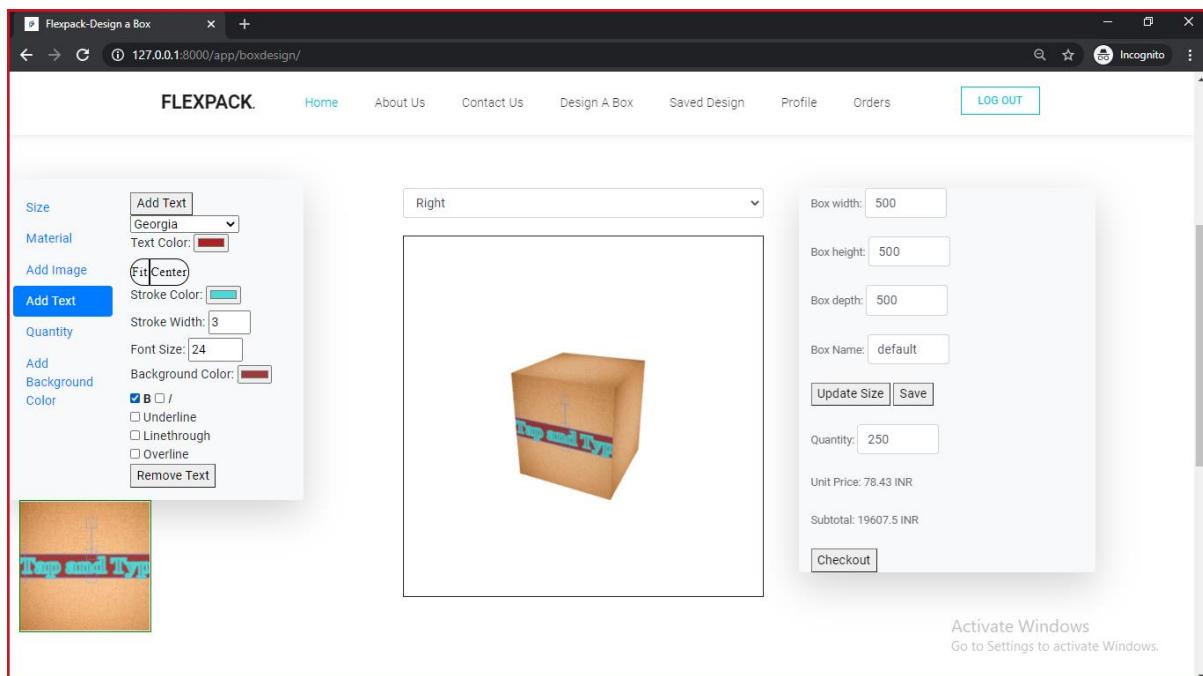
**Fig. 8.1.3.7 Font Size**

### 8.1.3.8 Background Color



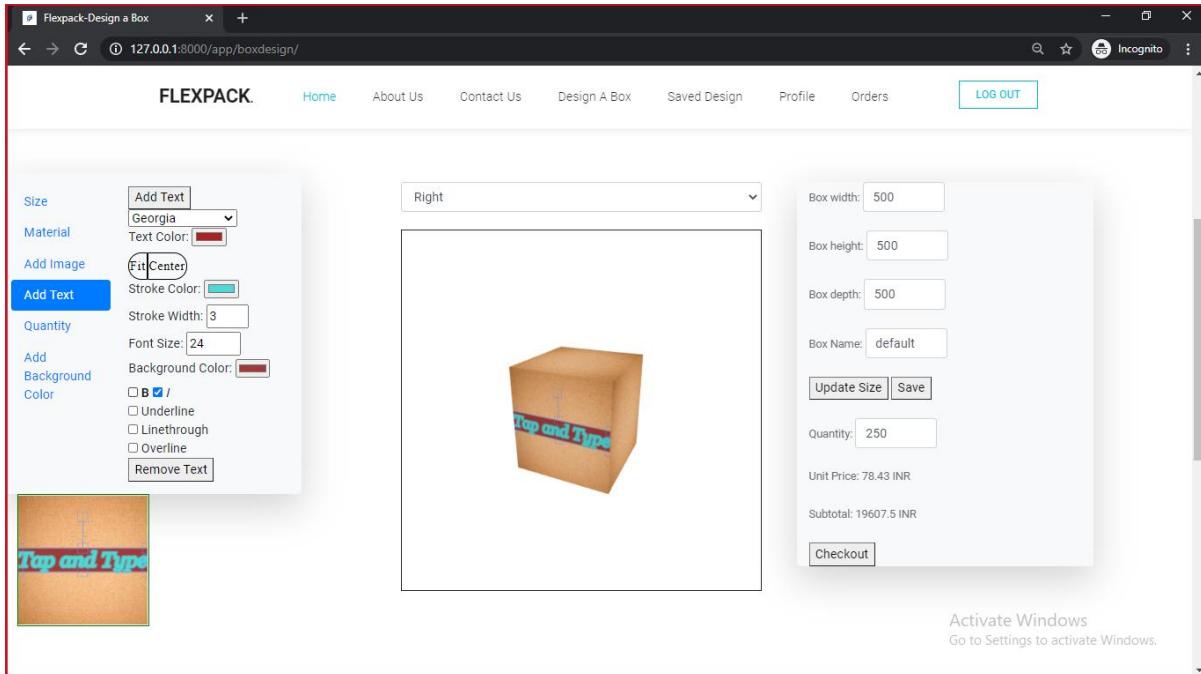
**Fig. 8.1.3.8 Background Color**

### 8.1.3.9 Bold Align



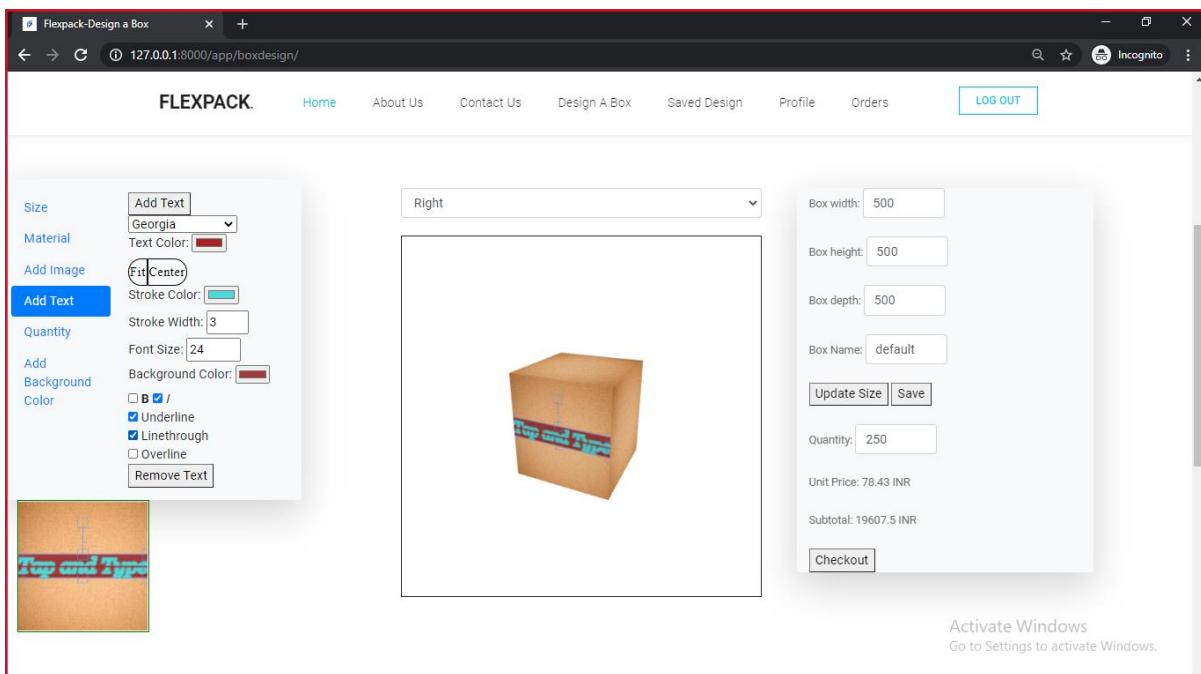
**Fig. 8.1.3.9 Bold Align**

### 8.1.3.10 Italic Align



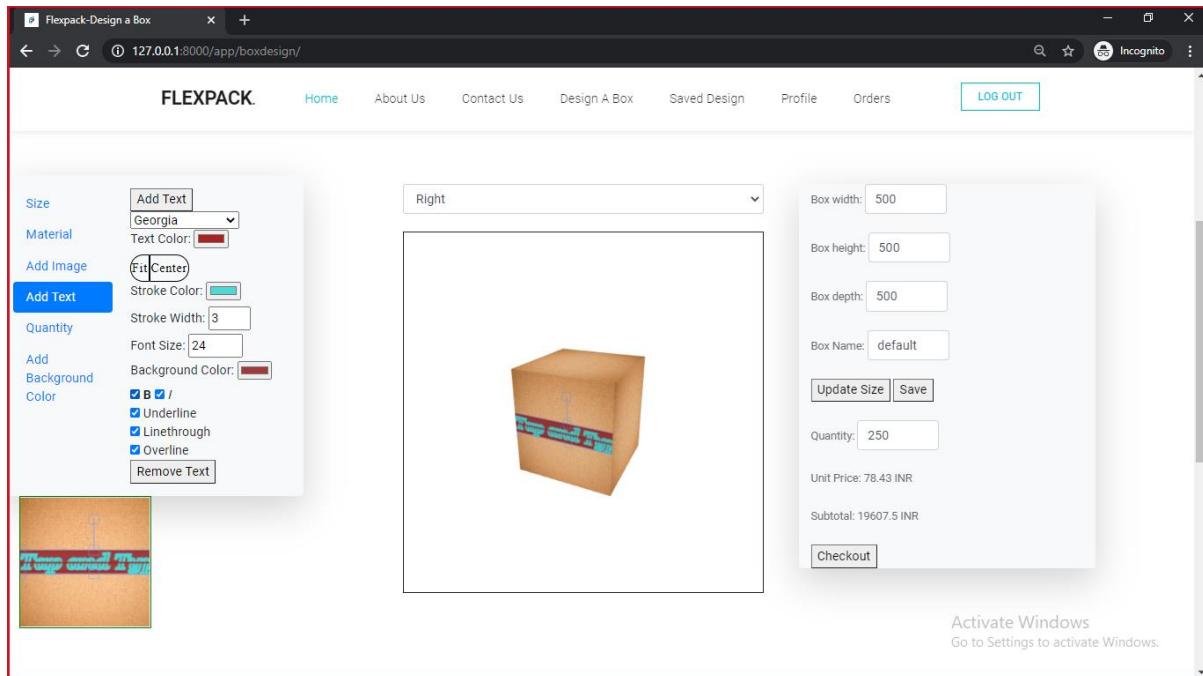
**Fig. 8.1.3.10 Italic Align**

### 8.1.3.11 Underline and Line through Align



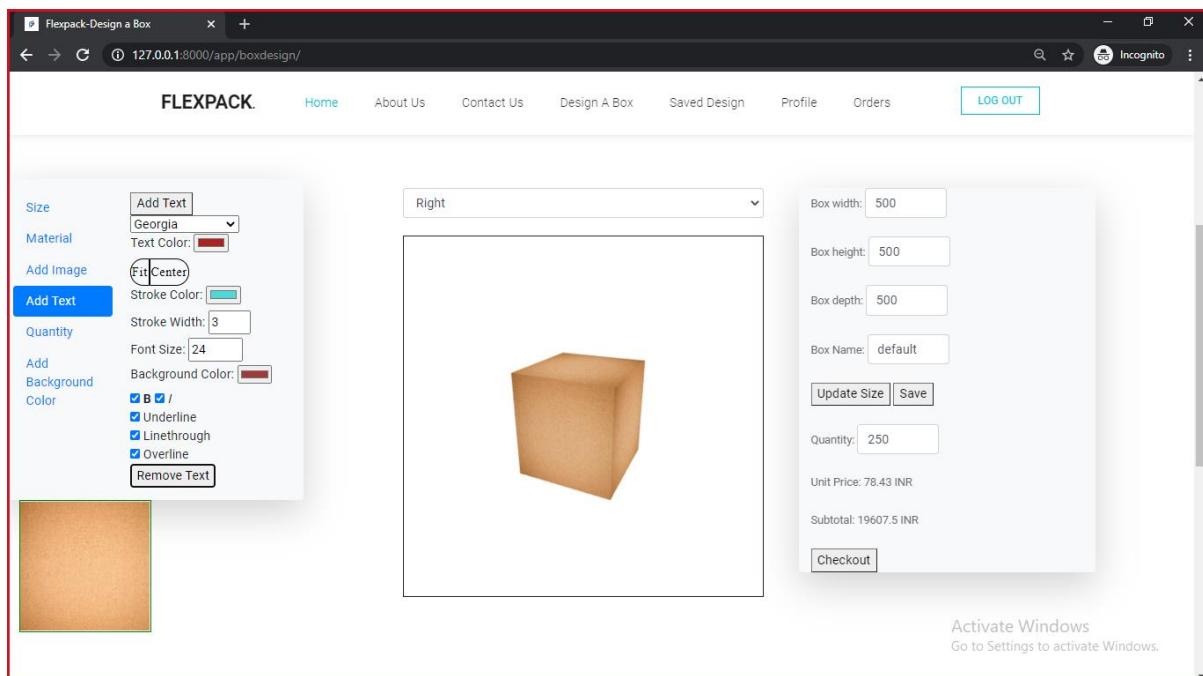
**Fig. 8.1.3.11 Underline and Line through Align**

### 8.1.3.12 Overline Align



**Fig. 8.1.3.12 Overline Align**

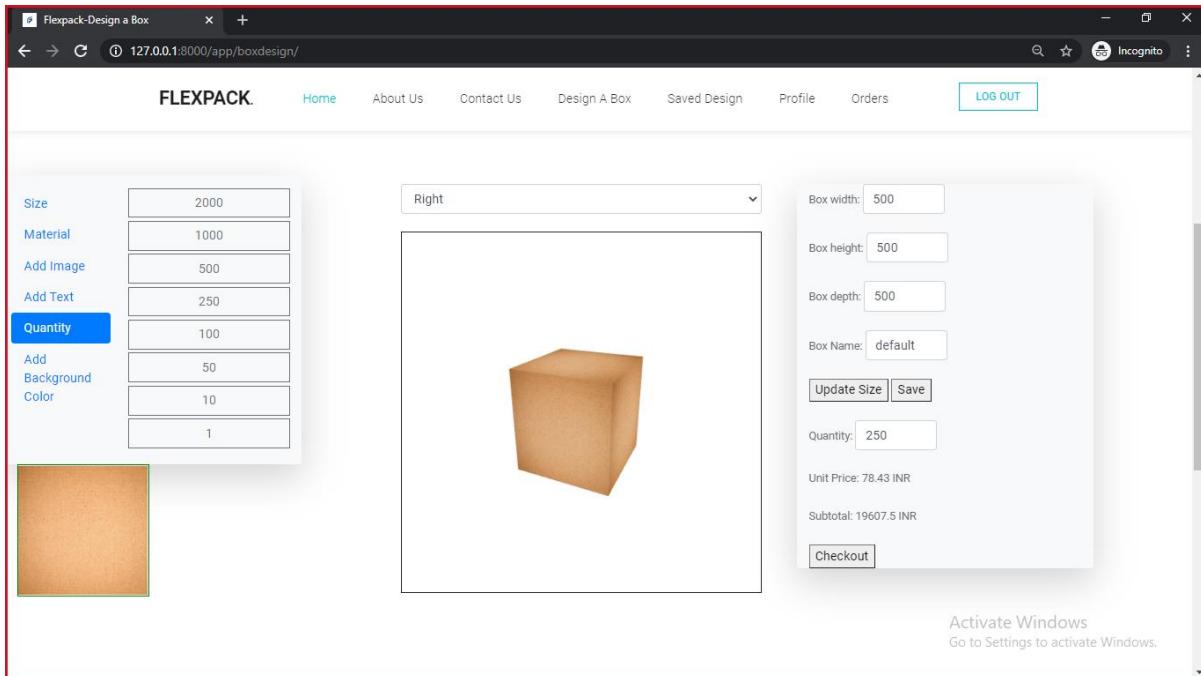
### 8.1.3.13 Remove Text



**Fig. 8.1.3.13 Remove Text**

## USER MANUAL

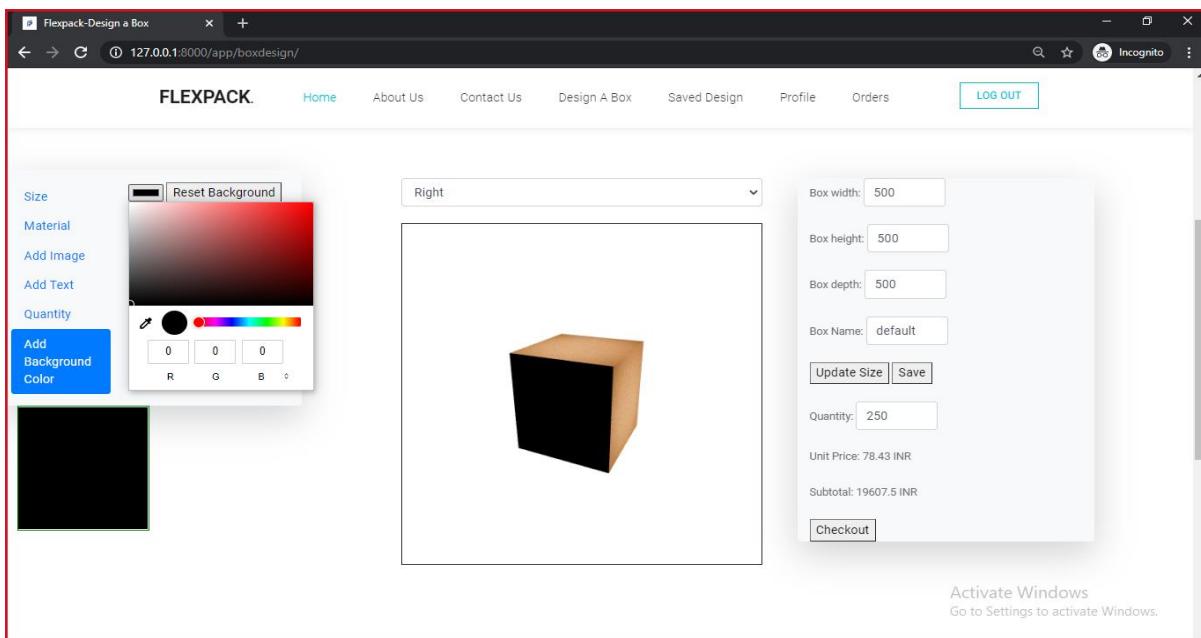
### 8.1.4 Quantity



**Fig. 8.1.4 Quantity**

### 8.1.5 Add Background

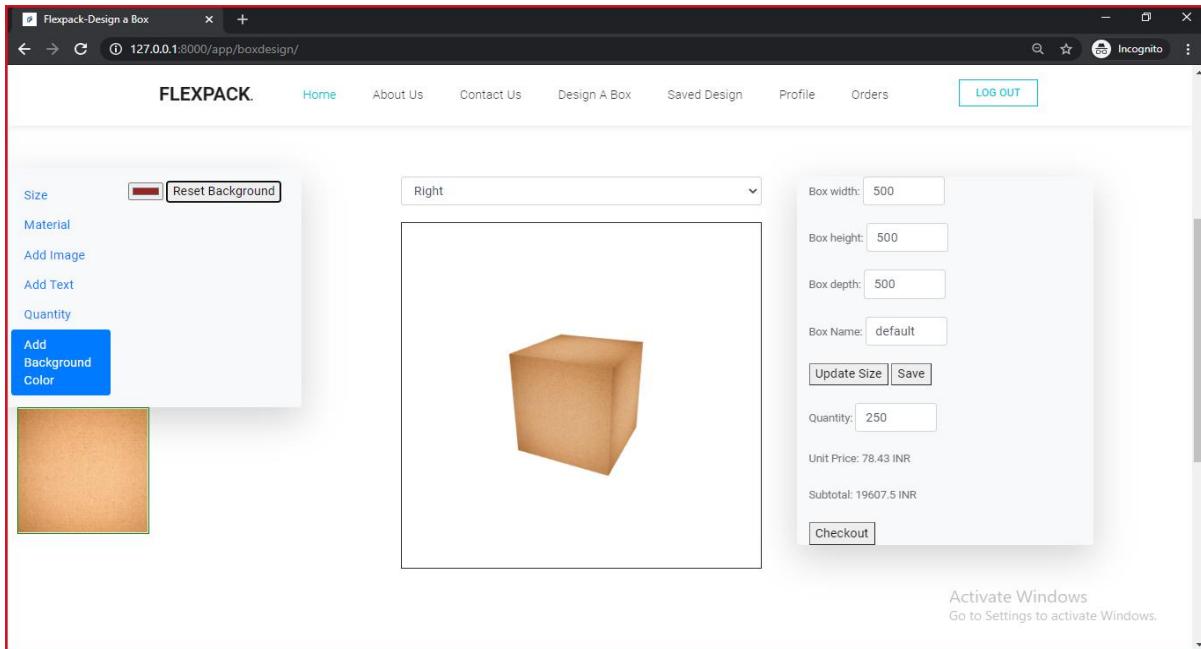
#### 8.1.5.1 Add Color of Background



**Fig. 8.1.5.1 Add Color of Background**

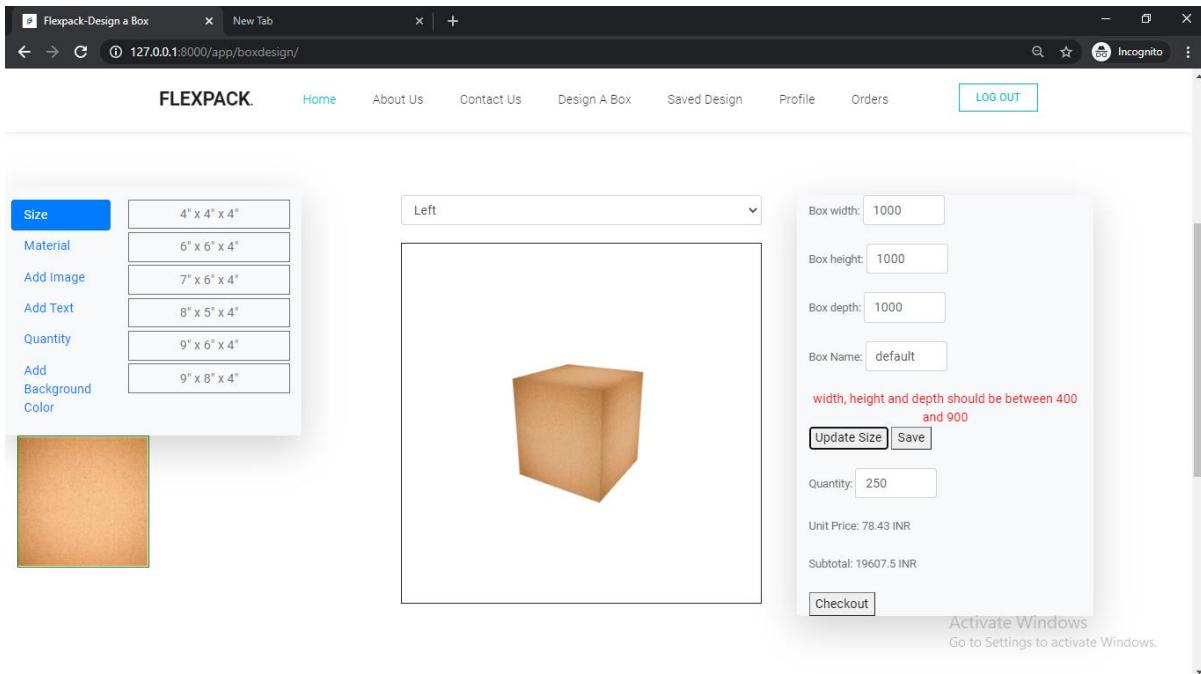
## USER MANUAL

### 8.1.5.2 Reset Background



**Fig. 8.1.5.2 Reset Background**

### 8.1.6 Box Size Validation



**Fig. 8.1.6 Box Size Validation**

## USER MANUAL

### 8.1.7 Checkout Page

#### 8.1.7.1 Address Detail

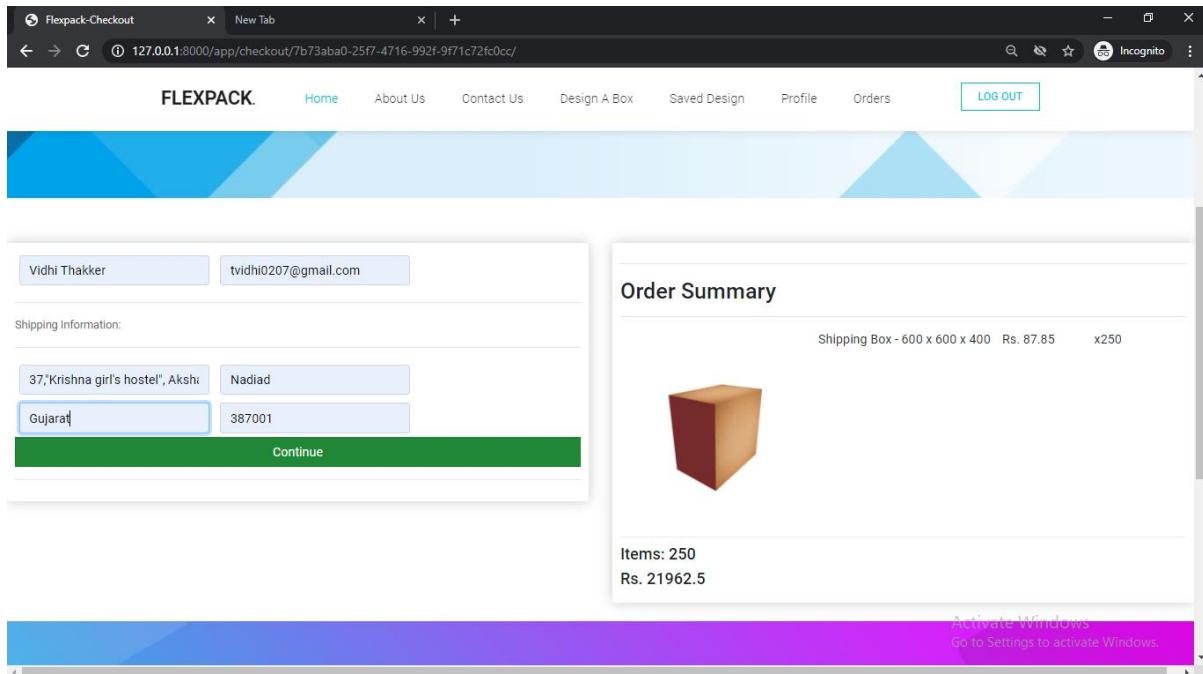


Fig. 8.1.7.1 Address Detail

#### 8.1.7.2 Transaction Complete Message

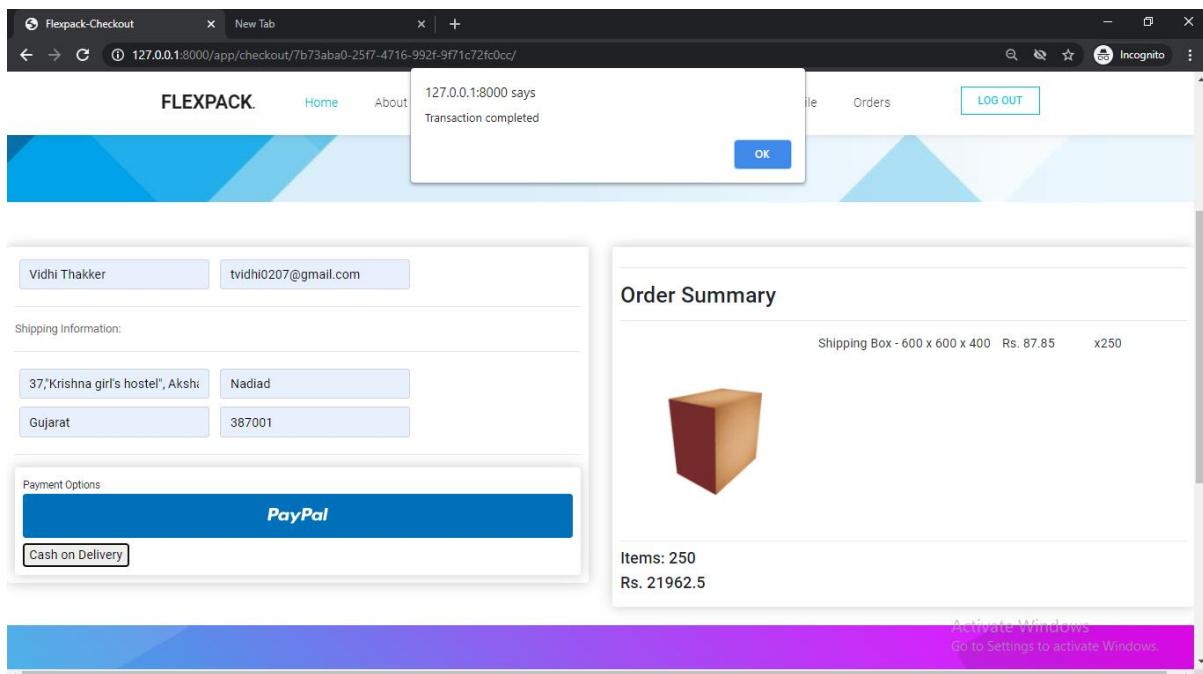


Fig. 8.1.7.2 Transaction Complete Message

## USER MANUAL

### 8.1.7.3 Transaction mail to customer

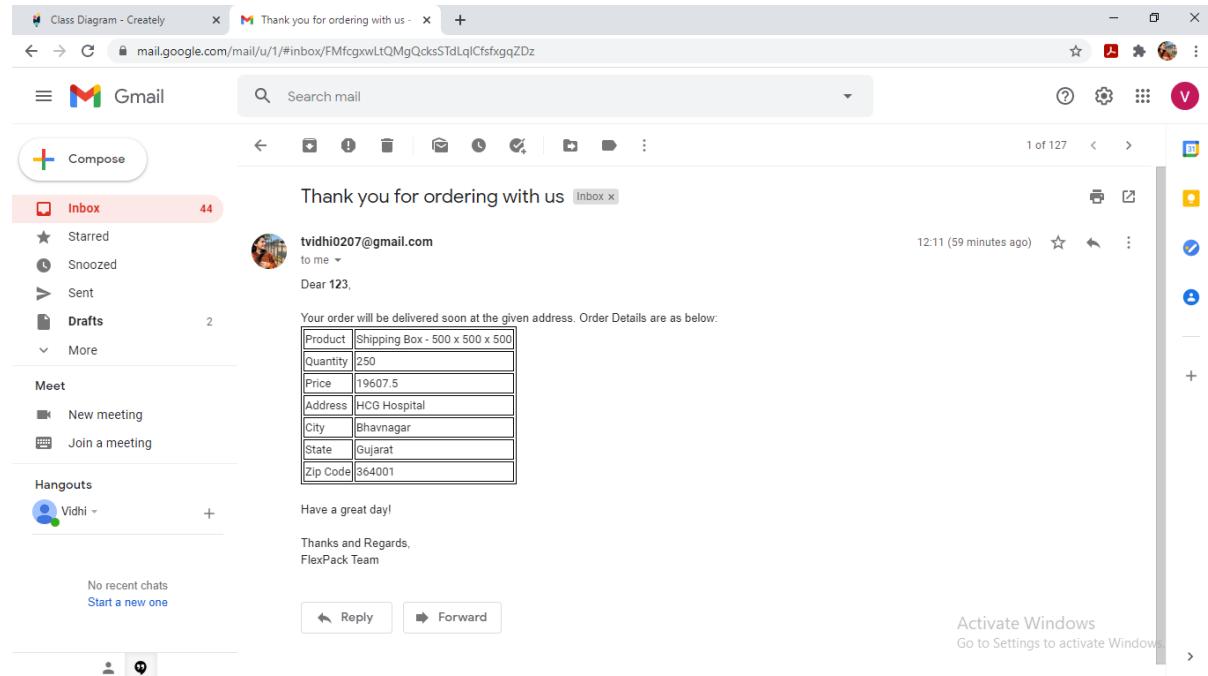


Fig. 8.1.7.3 Transaction mail to customer

### 8.1.8 About Us

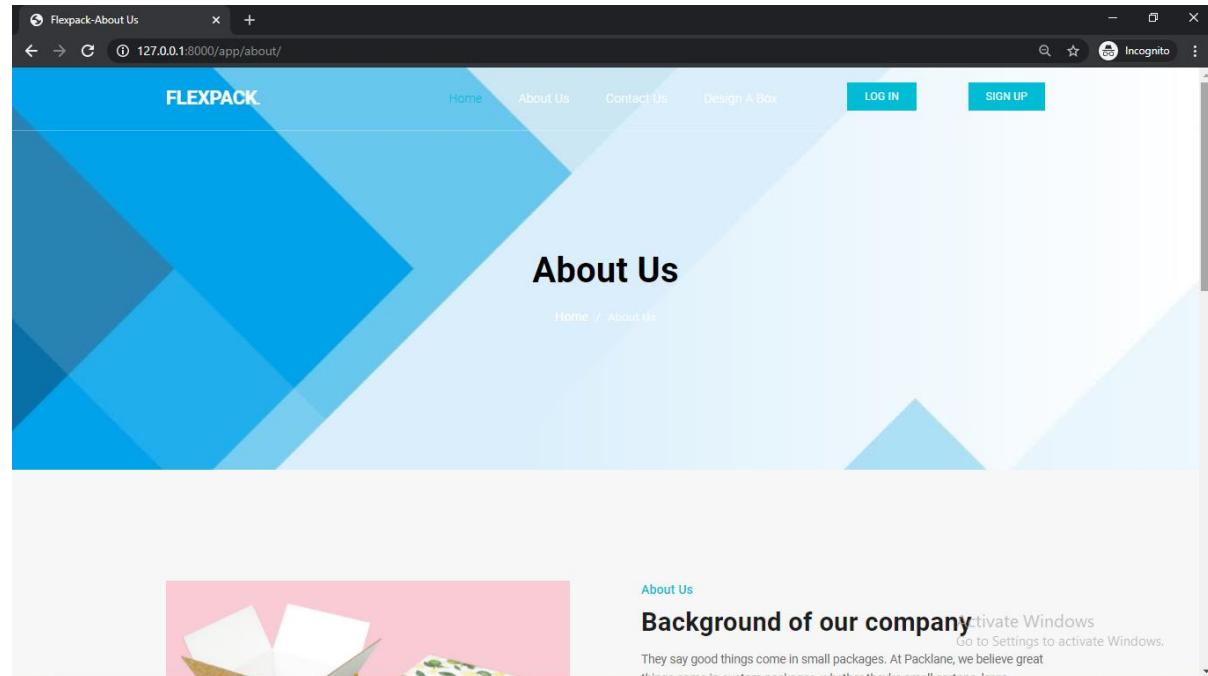
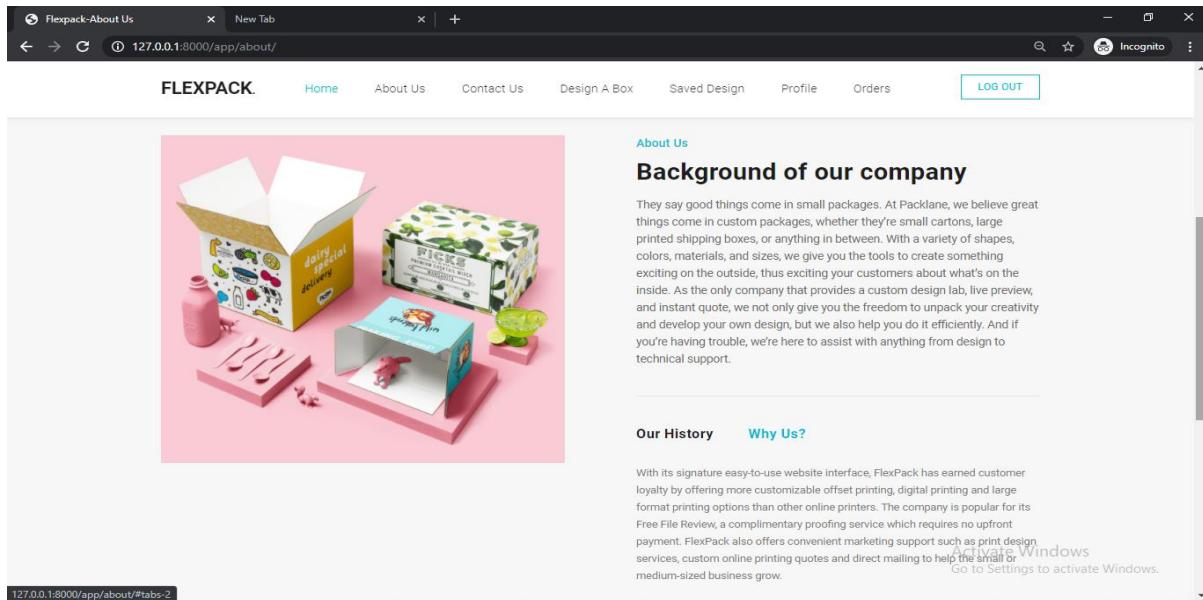


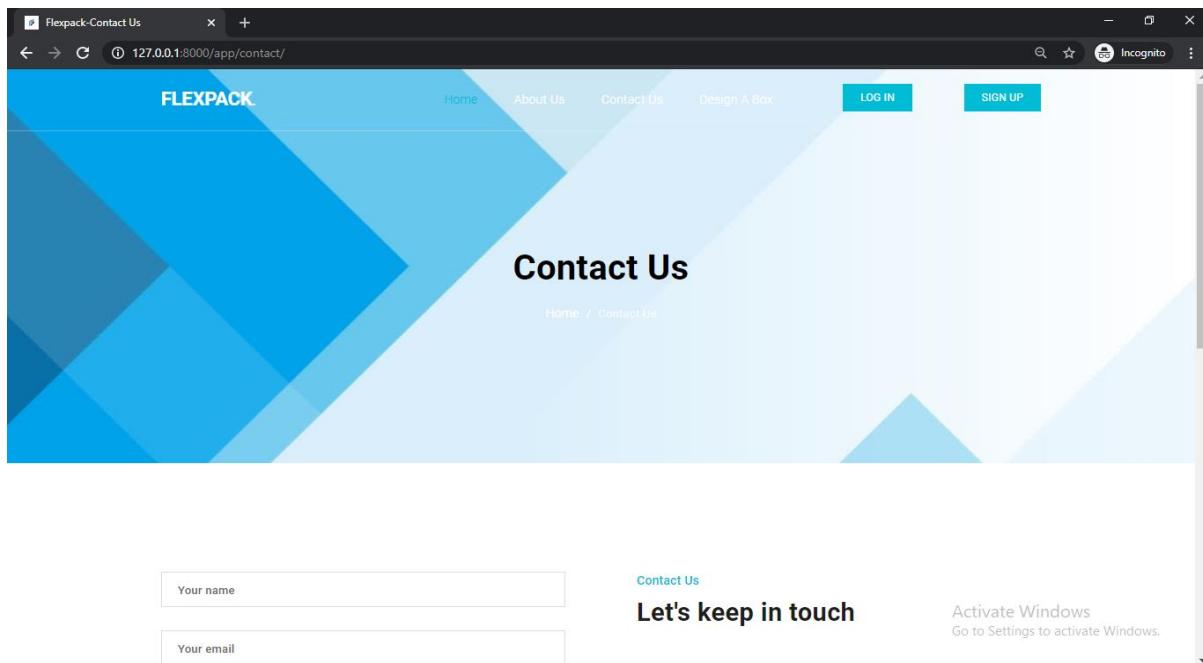
Fig. 8.1.8 About Us

## 8.1.8.1 About Us: Our history

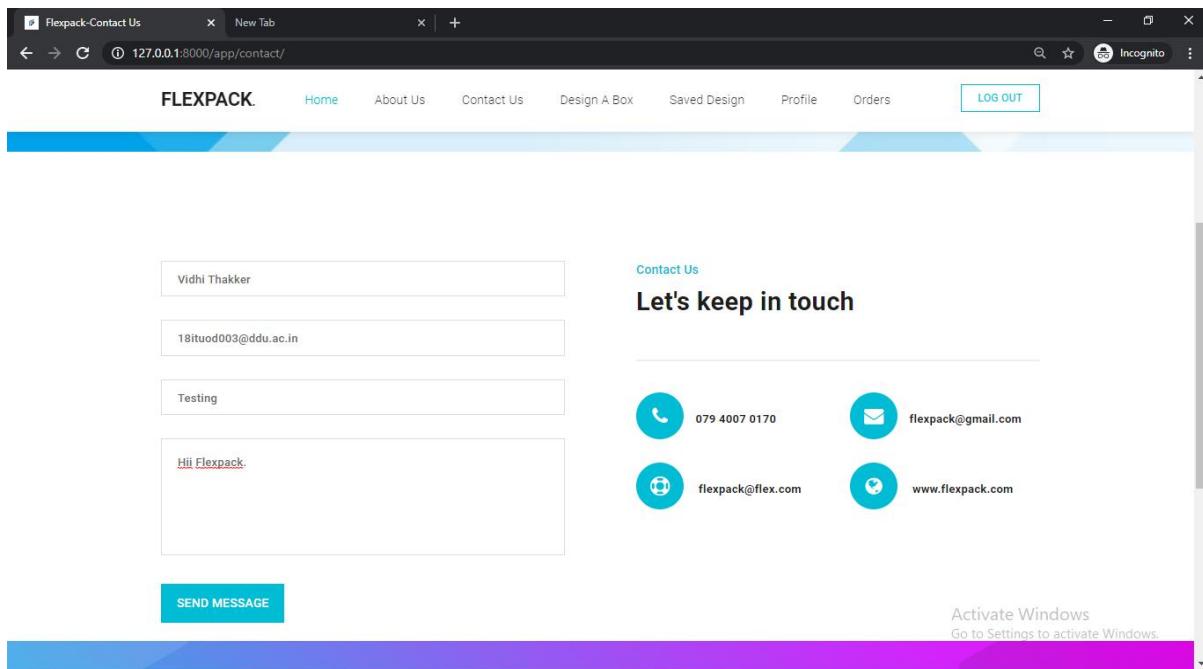


**Fig. 8.1.8.1 About Us: Our history**

## 8.1.9 Contact Us –Before email is send

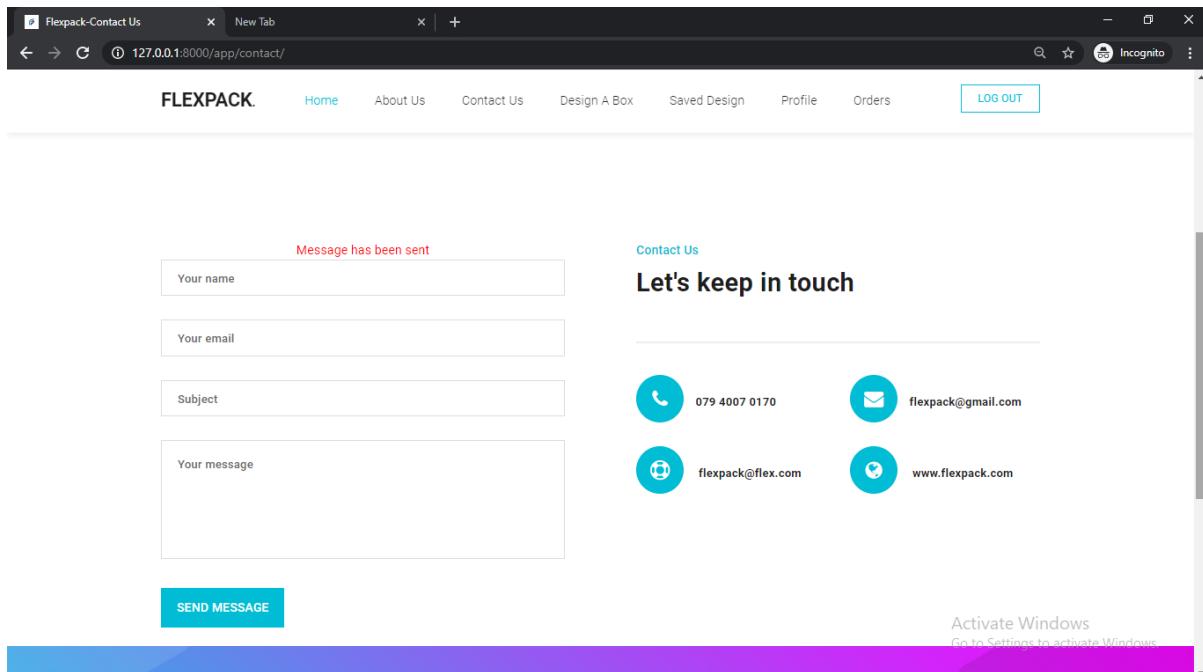


## USER MANUAL



**Fig. 8.1.9 Contact Us –Before email is send**

### 8.1.9.1 Send email to Customer



**Fig. 8.1.9.1 Send email to Customer**

## USER MANUAL

### 8.1.9.2 Email to Customer

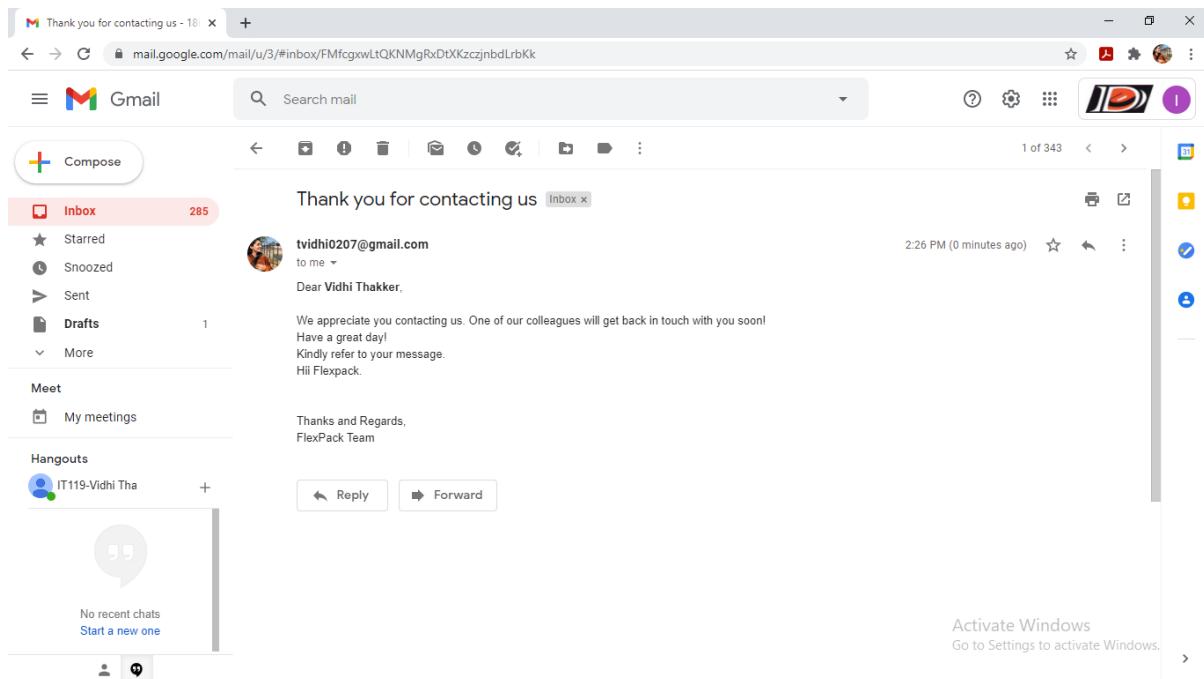


Fig. 8.1.9.2 Email to Customer

### 8.1.9.3 Email to Admin

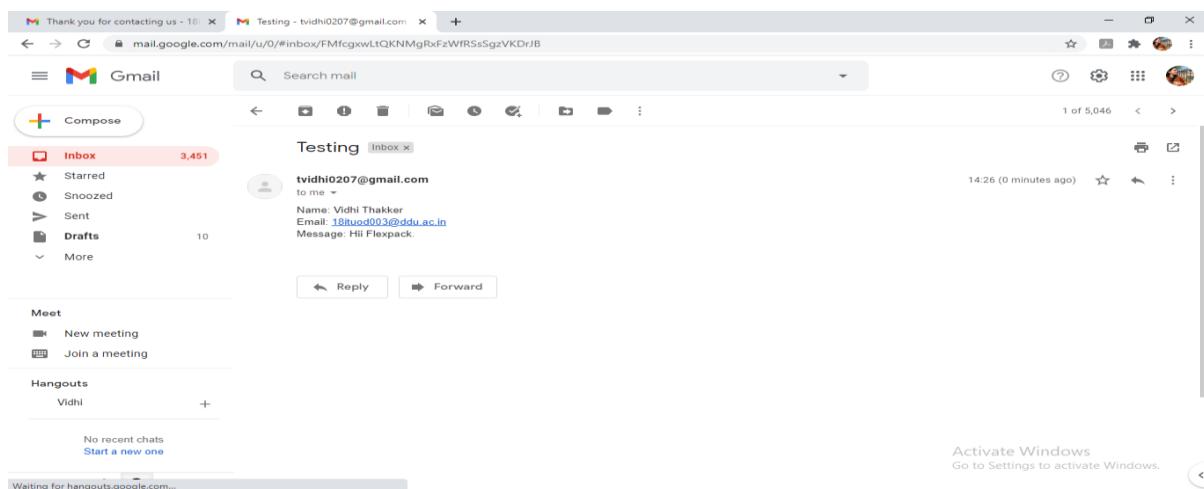
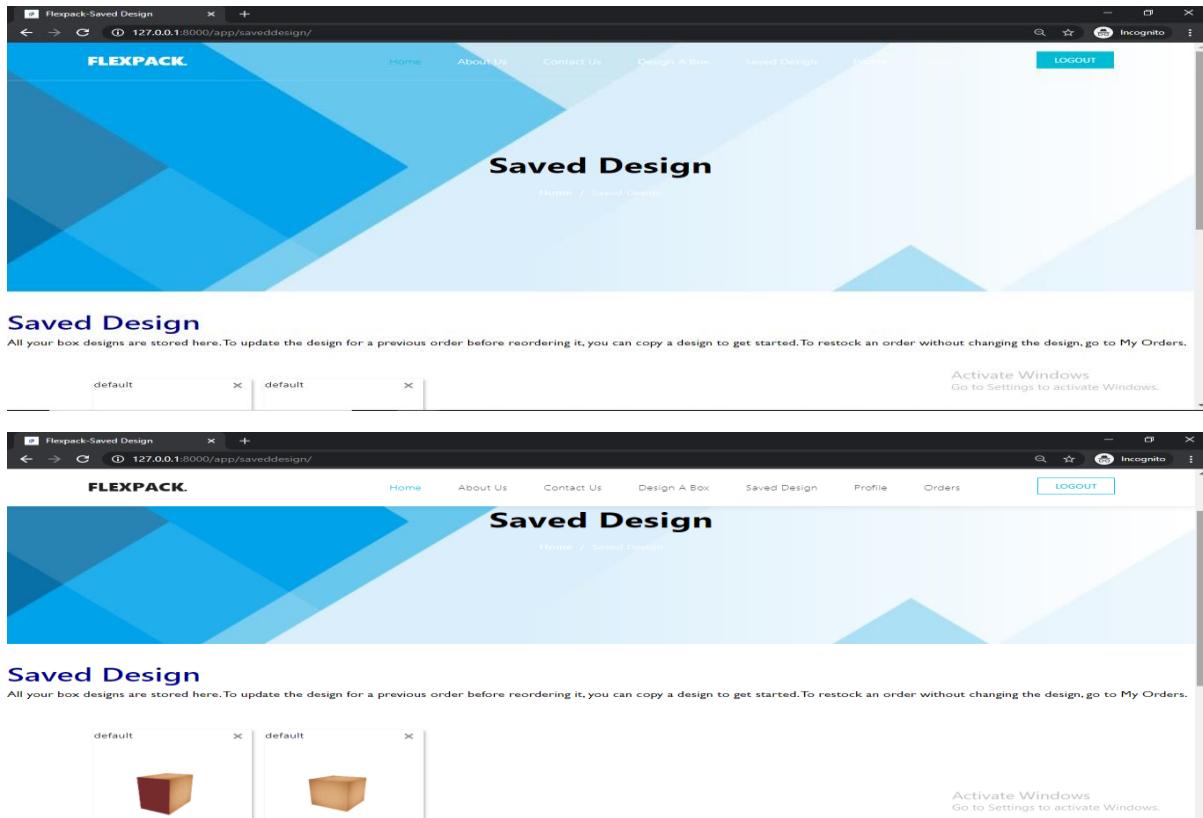


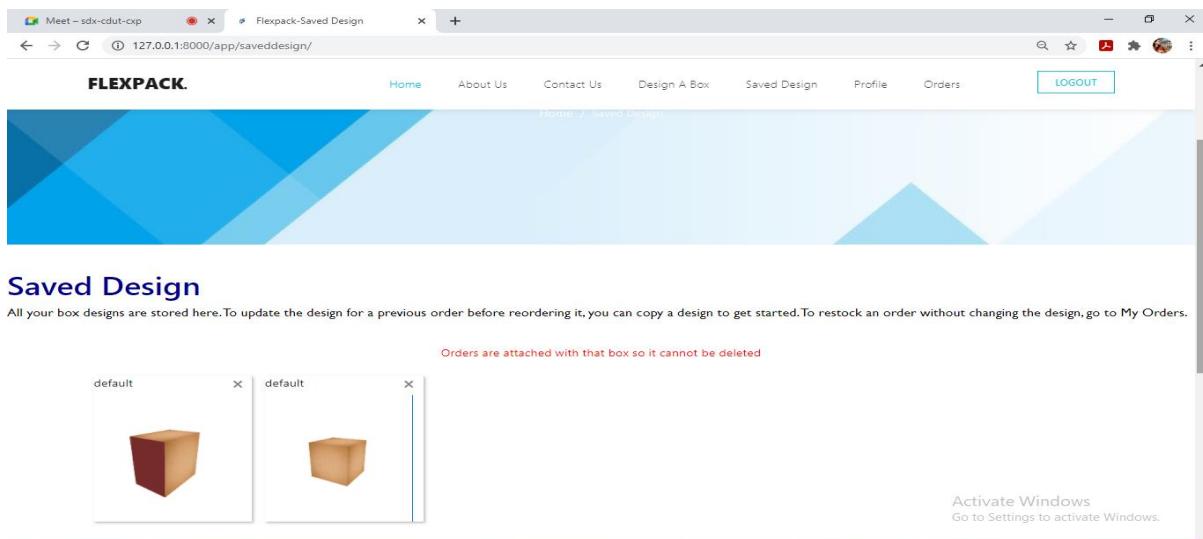
Fig. 8.1.9.3 Email to Admin

## 8.1.10 Saved Design



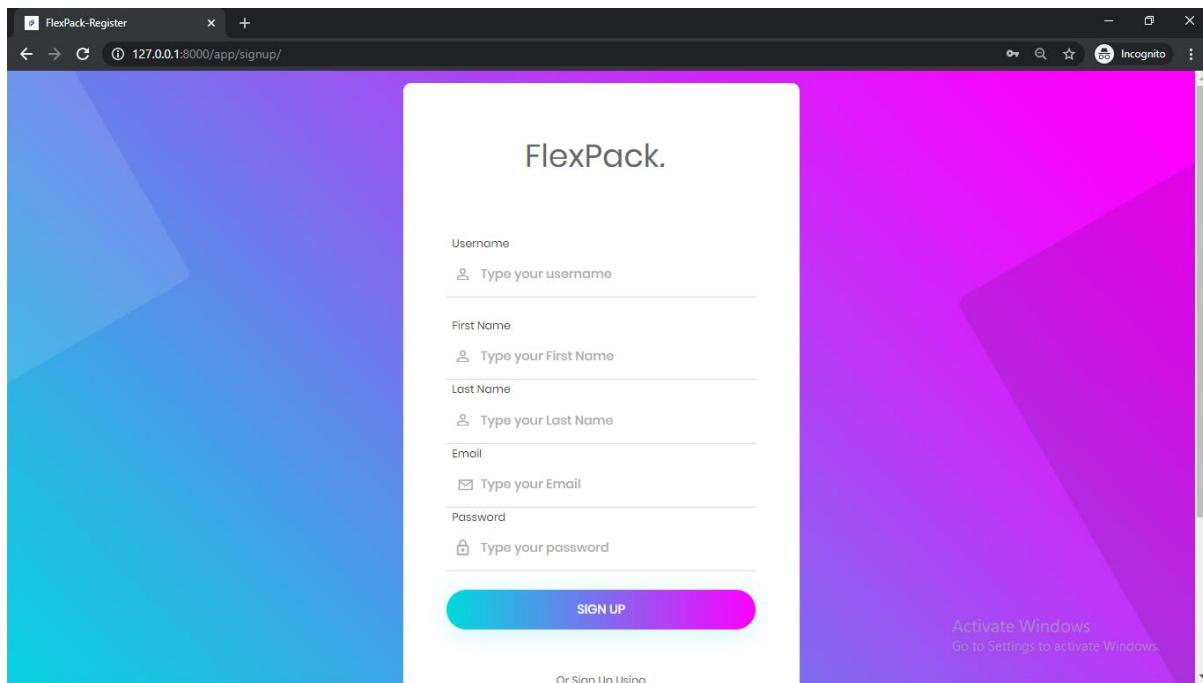
**Fig. 8.1.10 Saved Design**

## 8.1.10.1 Ordered box cannot be deleted



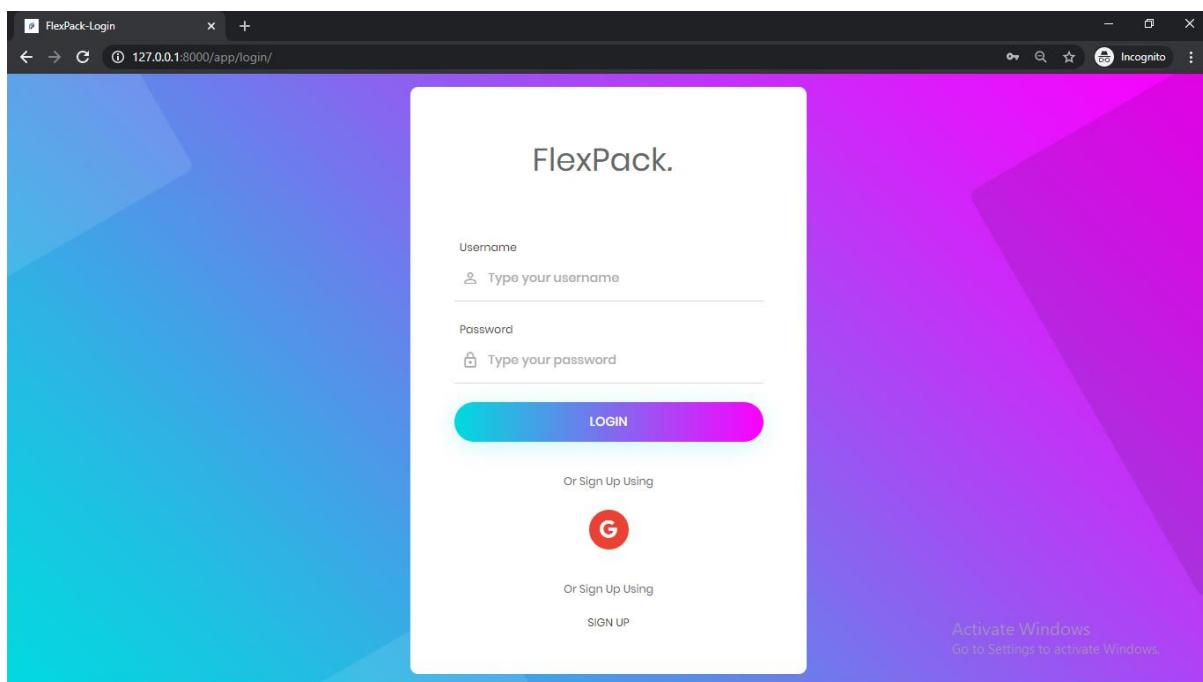
**Fig. 8.1.10.1 Ordered box cannot be deleted**

## 8.1.11 Sign Up



**Fig. 8.1.11 Sign Up**

## 8.1.12 Login



**Fig. 8.1.12 Login**

## USER MANUAL

### 8.2 SNAPSHOT OF ADMIN SIDE OF THE WEBSITE:

#### 8.2.1 Login

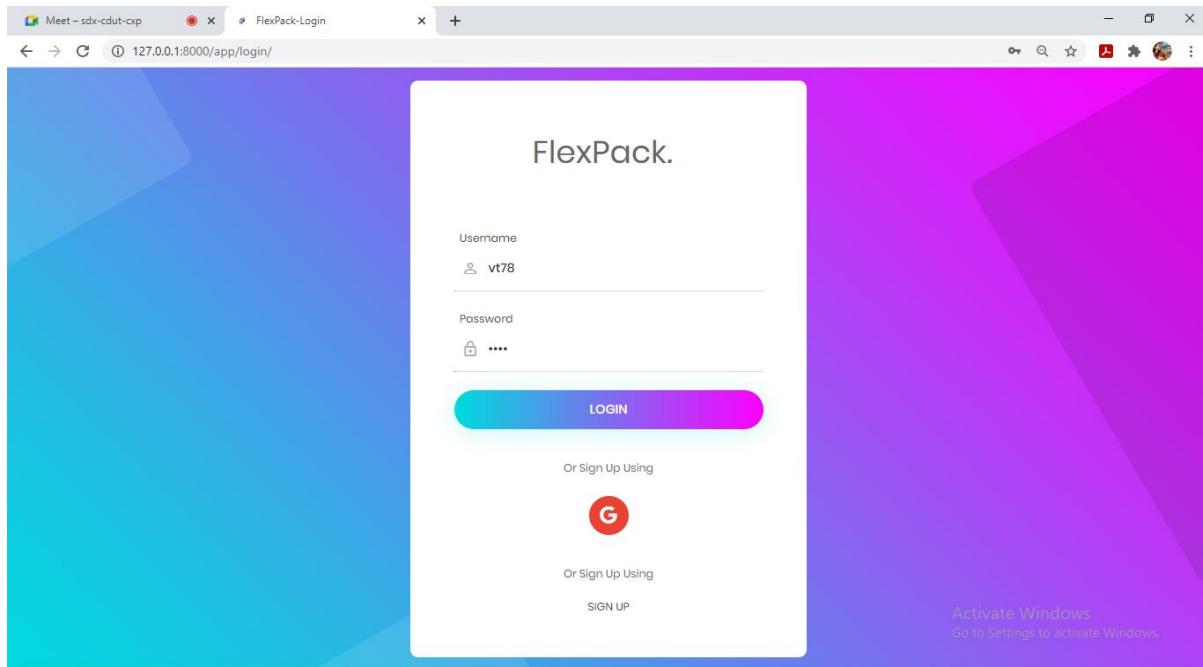


Fig. 8.2.1 Login

#### 8.2.2 Dashboard

A screenshot of the FlexPack Admin dashboard. The title bar says "Flexpack-Admin". The dashboard has a dark header with "Dashboard" and "Logout" buttons. Below the header, there are three teal-colored summary boxes: "Total Orders" (2), "Orders Delivered" (0), and "Orders Pending" (2). Under "CUSTOMERS:2", there is a table with two rows, each showing a customer name (Vidhi Thakker) and email (vt78@gmail.com or vidhi@gmail.com), with "View" buttons. Under "ORDERS", there is a table with columns: Product, Date Ordered, Address, Status, Update, and Remove. It lists two shipping boxes: one from 03-25 and another from 03-26, both pending. Each row has "Update" and "Delete" buttons. At the bottom right, there's an "Activate Windows" message.

Fig. 8.2.2 Dashboard

## USER MANUAL

### 8.2.3 Detail view of the user

The screenshot shows the FlexPack Admin Dashboard. At the top, there's a header bar with the title 'FlexPack.' and a 'Logout' link. Below the header, there are three main sections: 'Customer:' with a 'Delete Customer' button, 'Contact Information' with an email address 'Email: vidhi@gmail.com', and 'Total Orders' showing a count of '2'. Below these sections is a table titled 'Order History' with columns: Product, Date Ordered, Address, Status, Update, and Remove. The table contains two entries:

Product	Date Ordered	Address	Status	Update	Remove
Shipping Box - 500 x 500 x 500	2021-03-25	13/B, 'Vraj bhumi', senetorium, Meghani circle, Bhavnagar,364001, Bhavnagar, Gujarat- 364001	pending	<a href="#">Update</a>	<a href="#">Delete</a>
Shipping Box - 600 x 600 x 400	2021-03-26	13/B, 'Vraj bhumi', senetorium, Meghani circle, Bhavnagar,364001, Bhavnagar, Gujarat- 364001	pending	<a href="#">Update</a>	<a href="#">Delete</a>

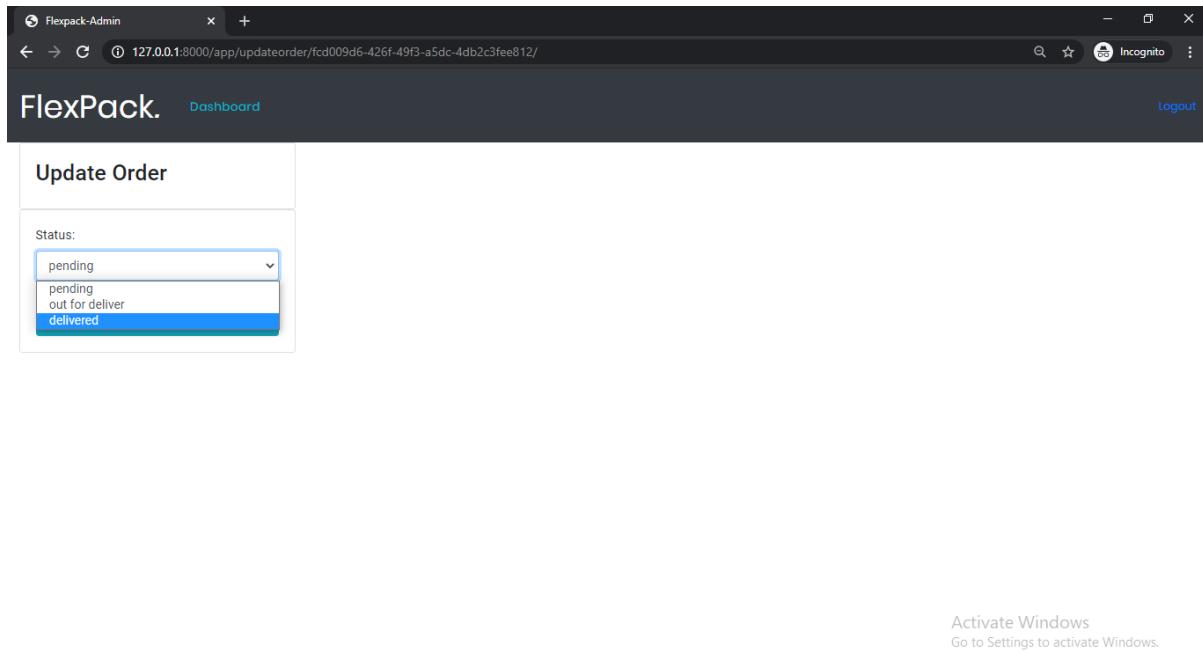
At the bottom right of the dashboard, there's a message: 'Activate Windows Go to Settings to activate Windows.'

**Fig. 8.2.3 Detail view of the user**

### 8.2.4 Update the status of Delivery

The screenshot shows the FlexPack Admin Dashboard. At the top, there's a header bar with the title 'FlexPack.' and a 'Logout' link. Below the header, there's a section titled 'Update Order' containing a 'Status:' dropdown menu set to 'pending' and a 'Submit' button. At the bottom right of the dashboard, there's a message: 'Activate Windows Go to Settings to activate Windows.'

## USER MANUAL



**Fig. 8.2.4 Update the status of Delivery**

### 8.3 SNAPSHOT OF DATABASE OF THE WEBSITE:

The screenshot shows the Django administration interface at the URL 127.0.0.1:8000/admin/jwt\_auth\_login\_register/user/. The left sidebar shows various Django models: AUTHENTICATION AND AUTHORIZATION (Groups, Users), DJCELERY (Crontabs, Intervals, Periodic tasks, Tasks, Workers), and JWT\_AUTH\_LOGIN\_REGISTER (Box prices, Boxes, Roles, Users). The 'Users' section under 'JWT\_AUTH\_LOGIN\_REGISTER' is highlighted with a yellow background. The main area displays a table titled 'Select user to change' with the following data:

Action:	USERNAME	FIRST NAME	LAST NAME	EMAIL	ROLE	PROVIDER
<input type="checkbox"/>	123	Parth	Maniyar	parthmaniyar@gmail.com	admin	-
<input type="checkbox"/>	admin	admin	admin	admin@admin.com	end_user	-
<input type="checkbox"/>	234	234	234	234@234.com	end_user	-
<input type="checkbox"/>	parthmaniyar90_ju2x-k1xn-5wqt-dhws	PARTH	MANIYAR	parthmaniyar90@gmail.com	end_user	google
<input type="checkbox"/>	parth.m_ju7z-jmgz-yjjk-c3fk	PARTH	MANIYAR	parth.m@ahuni.edu.in	end_user	google

At the bottom, there are 5 users listed.

# USER MANUAL

The screenshot shows the Django administration interface at the URL [127.0.0.1:8000/admin/jwt\\_auth\\_login\\_register/role/](http://127.0.0.1:8000/admin/jwt_auth_login_register/role/). The left sidebar lists various Django models: Groups, Users, DJCELERY (Crontabs, Intervals, Periodic tasks, Tasks, Workers), and JWT\_AUTH\_LOGIN\_REGISTER (Box prices, Boxes, Roles, Users). The 'Roles' model is selected and highlighted in yellow. The main content area is titled 'Select role to change' and displays a table with two rows:

Action:	NAME
<input type="checkbox"/>	end_user
<input type="checkbox"/>	admin

Below the table, it says '2 roles'.

The screenshot shows the Django administration interface at the URL [127.0.0.1:8000/admin/jwt\\_auth\\_login\\_register/box/](http://127.0.0.1:8000/admin/jwt_auth_login_register/box/). The left sidebar lists the same models as the previous screenshot. The 'Boxes' model is selected and highlighted in yellow. The main content area is titled 'Select box to change' and displays a table with 21 selected rows:

Action:	NAME	USER
<input type="checkbox"/>	default	parth.m.ju7z-jmgz-yjjk-c3fk
<input type="checkbox"/>	default	234
<input type="checkbox"/>	default	234
<input type="checkbox"/>	default	parth.m.ju7z-jmgz-yjjk-c3fk
<input type="checkbox"/>	Color Demo	123
<input type="checkbox"/>	default	parth.m.ju7z-jmgz-yjjk-c3fk
<input type="checkbox"/>	default	234
<input type="checkbox"/>	default	parth.m.ju7z-jmgz-yjjk-c3fk
<input type="checkbox"/>	First	123
<input type="checkbox"/>	default	parth.m.ju7z-jmgz-yjjk-c3fk
<input type="checkbox"/>	default	parth.m.ju7z-jmgz-yjjk-c3fk

## USER MANUAL

The screenshot shows the Django administration interface at the URL [127.0.0.1:8000/admin/jwt\\_auth\\_login\\_register/boxprice/](http://127.0.0.1:8000/admin/jwt_auth_login_register/boxprice/). The left sidebar has a 'JWT\_AUTH\_LOGIN\_REGISTER' section with 'Box prices' highlighted in yellow. The main content area is titled 'Select box price to change' and displays a table of three items:

MATERIAL	MATERIAL PRICE PER KG IN INR	DECORATION PRICE PER M <sup>2</sup> IN INR
dreamcoat	35.0	12.0
white	45.0	12.0
kraft	40.66	12.0

At the bottom of the table, it says '3 box prices'. There is a 'Go' button and a dropdown menu labeled 'Action:'. A red box highlights the 'ADD BOX PRICE' button in the top right corner.

**Fig. 8.3 SNAPSHOT OF DATABASE OF THE WEBSITE**

## **9. LIMITATION AND FUTURE ENHANCEMENTS**

### **9.1 LIMITATIONS**

User cannot enter box size less than 400m and greater than 900m. User cannot enter quantity greater than 2000. There can be some difference between real box and virtual designed box.

### **9.2 FUTURE ENHANCEMENTS**

For the future enhancement we will be focusing on increasing the efficiency of software. More of the Business wants and needs will be achieved seamlessly. We will work in detail for the security patches in our system and if there are any, we will try to remove every loophole in our system.

## **10. CONCLUSION AND DISCUSSION**

### **10.1 CONCLUSION**

The project gave me the opportunity to get to know new trending technology which has a bright future. This project has been implemented from what I learned in the given time period. We can enhance this project by keeping more intents and extending it to a college level in which it can solve many general queries. The scope of this project is subjective to the domain that was selected, however, it was developed by keeping in mind the goal to keep it as generic as possible.

### **10.2 DISCUSSION**

According to me, this project is absolutely a good start for gaining hands-on expertise on trending technology. Besides, there are numerous other improvements and enhancements possible to this software. The main problem encountered while working on designing the box module, as it is 3D model and while deciding the price of the box and other expenditures.

## **REFERENCES**

- [1]. Website of three js - <https://threejs.org/docs/>
- [2]. Website of Django REST framework - <https://www.django-rest-framework.org/>
- [3]. Website of Django Girls Tutorial – <https://tutorial.djangogirls.org/en/>
- [4]. Website of Django Documentation – “The web framework for perfectionists with deadlines.” - <https://docs.djangoproject.com/en/3.2/>
- [5]. Comparison Between Django and NodeJs

## **Appendix A (Comparison of Django and Nodejs) [5]**

Following table describes comparison between different frameworks

<b>Characteristic</b>	<b>Django</b>	<b>Node JS</b>
<b>Definition</b>	It is an open-source <u>Python-based framework</u> for expert developers whose aim is to build computer applications in a stipulated time frame.	It is an open-source framework that runs on JavaScript. It is meant for developers whose motive is to build strong APIs (client-side).
<b>Architecture</b>	It follows the MTV model template view. It helps with handling data for validating and interacting.	It is a runtime environment and thus, works on an event-driven model. It runs on an operating system, maintaining a small stack of requests.
<b>Security</b>	Django is more secured and comes with a built-in system, preventing any security deficiency.	NodeJS is not as secured as Django and requires manual operations in the system to manage security flaws.
<b>Performance</b>	It offers better performance, as there is a built-in house template system facilitating the execution of a required task quickly.	The performance of NodeJS is also good, as it allows the web professionals with more freedom when it comes to implementations. But again, this increases the overall time required to build the application.
<b>Complexity</b>	Django is more complex, as a developer has to follow a specified path for solving problems.	This system is less complicated. Here, the developer has a free hand to solve the problems as they like.
<b>Efficiency</b>	This framework is more efficient and offers fast speed. And so, it is more cost-effective.	This framework is easy to learn but consumes more operating time. Thus, it is a less cost-effective option.
<b>Reputation</b>	Django has a more solid reputation.	Growing steadily in popularity all over the world, NodeJS might soon become the more preferred framework.
<b>Community</b>	Django has a reasonably active yet small community.	NodeJS has quite an active community with experienced users to assist you on up-dates and customization.