INTRODUCTION

What is BARTER-MARKET ON WHEELS?

Barter-market on wheels is a unique start-up idea we are working on.
It is a platform for the small scale farmers to exchange theirs produces with the goods from the supermarket which they will require for their daily usage.
According to the census, around 68.84% (833.1 million) of Indians live in villages. They are an integral part of the developing country India. They are not exposed to supermarkets at all. They purchase their daily requireme
nts goods from the local Kisan shops which cannot offer such a wide range of products and discounts. Also, the small scale farmers are not getting a fair price for their produces due to various factors such as high transportation cost, middle-men paying a very less amount than the market price. We propose this platform keeping this social-concern in mind.

How is the application developed?

We have developed a web application interface to achieve the software requirements for the idea. It can be operated by various people simultaneously, which mainly consists of the Warehouse Manager and the Truck Driver. The interfaces are simple so that anyone can operate it without the complete the internal knowledge about it.
It is developed such that it operates on a TOUCH-SCREEN environment and it can be deployed on the server and can be accessed on the Point-of-Sale machine.

What does the Application contain?

The Warehouse interface will consist of different options to view the trucks associated with the particular warehouse, to view the customers associated with the trucks that belong to the particular warehouse, to add the items of the supermarket that will be sold on the truck and can

be updated from the warehouse, the list of the items that can be procured from the farmers can be updated along with the price update.

The truck interface will consist of three major options, to load the supermarket items into the truck which will also make the changes dynamically in the warehouse interface, an interface to procure the goods from the framers which will generate an amount bill, to sell the items to the farmers- they will have to purchase the items for a minimum of 80% of the total amount and the rest of the amount can be taken as case.

What tools have we used?

We are making use of Django as a major backend platform to develop the project. Django is a Python-based free and open-source web framework, which follows the model-view-template (MVT) architectural pattern. Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used for the complete development of the project. Some of the well-known websites that have used Django are Public Broadcasting Service, Instagram, Mozilla, The Washington Times, Disqus, Bitbucket, and Nextdoor. We are using HTML, CSS, and JavaScript for front-end development.

How does our project solve the social issue?

We are basically providing a platform of ‘market at the doorstep’, where they can sell their produces at a higher rate eliminating the middle-men and transportation issues they are facing now. They will be able to get a fair price for their crops grown.

Also, we are providing an exposure of supermarkets to rural areas, which will have a wide range of products and will offer huge discounts. They will be able to purchase their daily needs from the supermarkets at a discounted rate.

The products received from the farmers will be sold to the consumers in the urban areas at a lower price making it profitable for farmers, urban population and the company as well.

LITERATURE SURVEY

This a unique model we are developing. Nowhere in India, there is this system of barter with money equivalence for exchange of goods and supermarkets on wheels.

We are making an attempt to provide a profitable platform of ‘market at the doorstep’ for the small scale farmers and also providing an exposure towards the supermarket with a wide range of products with huge discounts.

The development of the rural areas and the farmers will lead to the development of the country.

Barter System

Bartering is trading services or goods with another person when there is no money involved. This type of exchange was relied upon by early civilizations. A barter system is an old method of exchange. The system has been used for centuries and long before money was invented. People exchanged services and goods for other services and goods in return. The history of bartering dates all the way back to 6000 BC. Introduced by Mesopotamia tribes, bartering was adopted by Phoenicians. Due to lack of money, bartering became popular in the 1930s during the Great Depression. It was used to obtain food and various other services. It was done through groups or between people who acted similar to banks. If any items were sold, the owner would receive credit and the buyer's account would be debited.

A complication of bartering is determining how trustworthy the person you are trading with is. The other person does not have any proof or certification that they are legitimate, and there is no consumer protection or warranties involved. This means that services and goods you are exchanging may be exchanged for poor or defective items. Also, the actual value of the product cannot be valued using the barter system. One sack of rice cannot be equated to one sack of cement.

These are some of the issues with the barter system and the reasons for which the barter was not successful.

On the other hand, there are great advantages to bartering. As mentioned earlier, you do not need money to barter. Another advantage is that there is flexibility in bartering. The Indian market has seen a precipitous acceptance, with hotel giants, newspapers, advertising agencies, and different sectors of the corporate industry accepting and adapting to this system for their mutual benefit.

In this world of demonetization and emerging digital currencies, barter trade is playing a crucial role in corporate agencies. With the help of trade currency or credit, companies can exchange and trade with any obligation of buying or selling the product or service instantly. This cashless exchange is leading the Indian market towards increased profit, improved purchasing power, better inventory management, incremental revenue, and better productivity.

With the rising popularity of the barter system, commercial, as well as corporate barter companies are emerging at a rapid rate. Thousands of companies are getting involved with these smart barter solutions, with a high rate of almost 40%. While India is steadily paving its way towards a barter-oriented corporate world, the percentage is still quite low. Only 15% to 20% of businesses are availing the benefits of barter trade.

The barter system can be made more trusted by equating the money value of the goods and exchanging the goods which are of the same value.

Farmers and the Middle-Man

Farm direct marketing is a long felt need of the farmers and consumers of the country as it goes a long way in ensuring higher remuneration to the farmers and meeting the satisfaction level of the consumers through the direct sale of the agricultural commodity by the farmers to the consumer at affordable prices. Direct marketing of agricultural produce helps in complete elimination of middlemen and commission agents who charge high level of commission fee from the agriculturists/farmers coming to the market yards for selling their produce and then artificially inflate the retail prices.

The Centre's attempt to make a difference in farmers' livelihood with e-NAM has been resisted by the states due to vested interests. A large number of middlemen or arhtiyas are exploiting the farmers for years.

How our model solves these issues?

The transportation cost is a major factor for a small scale farmer. Since we are providing a market at their doorstep, they will not be having an issue of transportation and their money will be saved.

Our model doesn't involve any kind of middle-men who will be interacting with the farmer, we will be in direct communication and the goods from the farmers will be directly sold to the consumers in the urban area at a lower rate than the actual market price making it profitable for both farmer as well as the urban population.

We are also providing insight and exposure to the supermarkets to the rural area. The supermarkets usually purchase the products directly from the companies, and will get it at a lower price, hence will be able to sell the items at a discounted price. They will be getting the benefits of these discounts and also the wide range of products with a lot of options to choose between will be available to them.

REQUIREMENT SPECIFICATION

Functional Requirements

- Procure items from farmers with the equivalent price to the entered quantity.
- Sell supermarket items to the farmers (minimum 80% of the procured price).
- Add supermarket items to the truck.
- View the customers and trucks associated to the particular warehouse.
- Update the list and quantity of supermarket items in the warehouse.
- Update the price of the procuring products from the warehouse.

Hardware Requirements

- Computer or a laptop with a browser (preferably Mozilla Firefox).
- Minimum RAM of 4-Gigabytes.
- Screen-touch system(preferably)

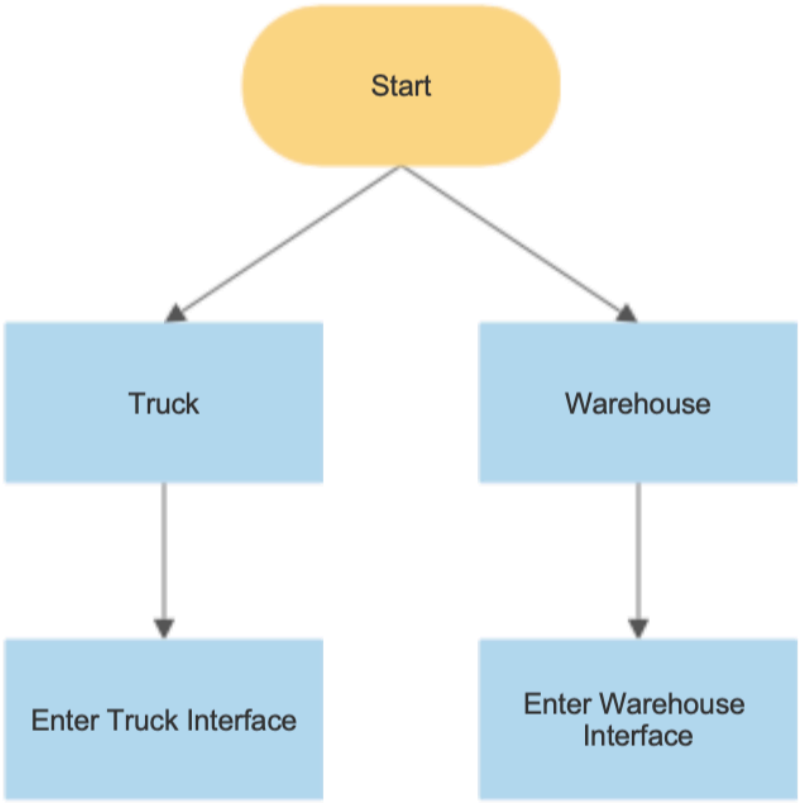
Software Requirements

- Django version 2.0 and above.
- Python 3.5 and above.
- Operating System: Windows 7 and above or MacOs El Capitan and above.
- Command line tools.
- HTML.
- CSS.
- JavaScript.

DESIGN AND IMPLEMENTATION

We have basically developed 2 interfaces- Truck and Warehouse, all the operations are linked to these two interfaces.

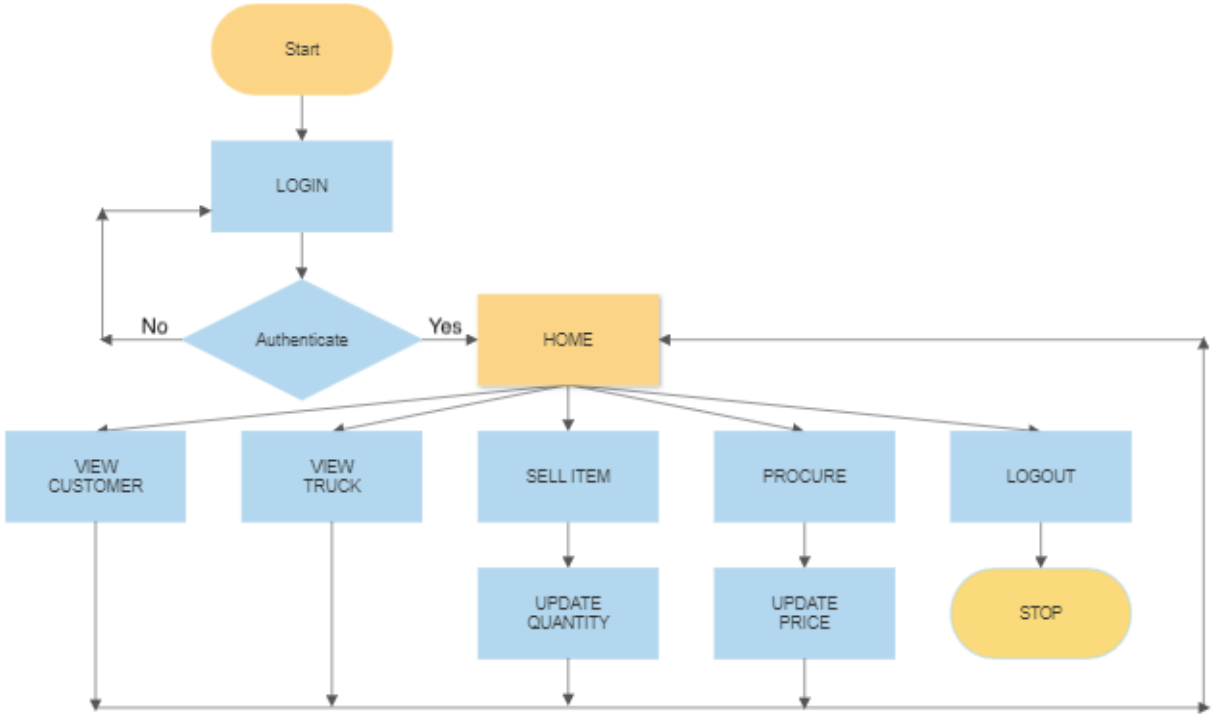
The home page consists of two buttons to navigate to truck and warehouse interface.



The truck interface consists of two basic operation - procure of goods from the farmers and sell the supermarket items. Once the login is authenticated, either procure or sell or items can be done. In the procure items, items can be procured from the farmer and the corresponding amount bill will be generated. Then it will be navigated to the sell page where the items have to be purchased for a minimum of 80% of the amount bill.


Start
Login
NO
Authenticated
Yes
HOME
Procure
Sell
Select Item
Select Item
Enter quantity
Enter quantity
Generate bill
No
Check if bill amount is 80% of procured amount
Yes
Generate bill

The warehouse interface will consists of four buttons to view the trucks associated with the particular warehouse, to view the customers associated with the trucks that belong to the particular warehouse, to add the items of the supermarket that will be sold on the truck and can be updated from the warehouse, the list of the items that can be procured from the farmers can be updated along with the price update.



Various libraries were imported from Django for the implementation of the Views

views.py

```
from django.http import HttpResponse
from .models import item,truck,ware_house,procure_item,customer,driver,item_qtty_ware,item_qtty_truck
from .models import procitem_qtty_ware,procitem_qtty_truck
from django.shortcuts import render,redirect,get_object_or_404
from django.contrib.auth.models import User
from django.db.models import Sum
from django.db.models import Q
```

Views will contain the function that will be used for the authentication and also the data from the HTML pages will be able to move from one page to another.

views.py

```
def ware_house_login(request):
    if request.method == 'POST':
        w_place = request.POST['w_place']
        w_password=request.POST['w_password']

        single_ware = ware_house.objects.filter(ware_place=w_place).first()
        if single_ware == None:
            msg="The Ware-house location is invalid..!"
            back='ware_house_login'
            return render(request,'error.html',{'msg':msg,'back':back})

        else:
            if single_ware.ware_password == w_password:
                return redirect('ware_home',pk=single_ware.id)
            else:
                return render(request,'ware_login.html')


    return render(request,'ware_login.html')

def truck_login(request):
    if request.method == 'POST':
        t_number = request.POST['t_number']
        t_password=request.POST['t_password']

        single_truck = truck.objects.filter(truck_number=t_number).first()
        if single_truck == None:
            msg="The truck number is invalid..!"
            back='truck_login'
            return render(request,'error.html',{'msg':msg,'back':back})

        else:
            if single_truck.truck_password == t_password:
                return redirect('truck_home',pk=single_truck.id)
            else:
                return render(request,'truck_login.html')
```

The models are basically a representation of your application's database layout. All models are subclass of the django.db.models.Model class. Each class will be transformed into database tables. Each field is represented by instances of django.db.models.Field subclasses (built-in Django core) and will be translated into database columns.In the Board model definition, more

specifically in the name field, we are also setting the parameter unique=True, as the name suggests, it will enforce the uniqueness of the field at the database level.

models.py

```
class ware_house(models.Model):
    ware_place=models.CharField(max_length=30,unique=True)
    ware_password=models.CharField(max_length=30)

    def __str__(self):
        return (self.ware_place)
#-----------------------------------------------------------------------------------------------------------------------

class zone(models.Model):
    zone_name=models.CharField(max_length=30)
    z_ware_id=models.ForeignKey(ware_house,related_name='zware_id',on_delete=models.SET(0),)

class village(models.Model):
    village_name=models.CharField(max_length=30)
    v_zone_id=models.ForeignKey(zone,related_name='vzone_id',on_delete=models.SET(0),)

class driver(models.Model):
    driver_name=models.CharField(max_length=30)
    driver_age=models.IntegerField()
    driver_mobile=models.CharField(max_length=12)
    #d_truck_id=models.ForeignKey(truck,related_name='dtruck_id',)
    def __str__(self):
        return (self.driver_name)

class truck(models.Model):
    truck_number=models.CharField(max_length=6,unique=True)
    truck_password=models.CharField(max_length=30,default=None)
    #d_truck_id=models.ForeignKey(truck,related_name='dtruck_id',)
    t_ware_id=models.ForeignKey(ware_house,related_name='tware_id',on_delete=models.CASCADE,)
    t_zone_id=models.ForeignKey(zone,related_name='tzone_id',on_delete=models.SET(0),null=True,default=None)
    t_driver_id=models.ForeignKey(driver,related_name='tdriver_id',on_delete=models.SET(0),null=True,default=None)

    def __str__(self):
        return (self.truck_number)
```

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

selling_item_list.html

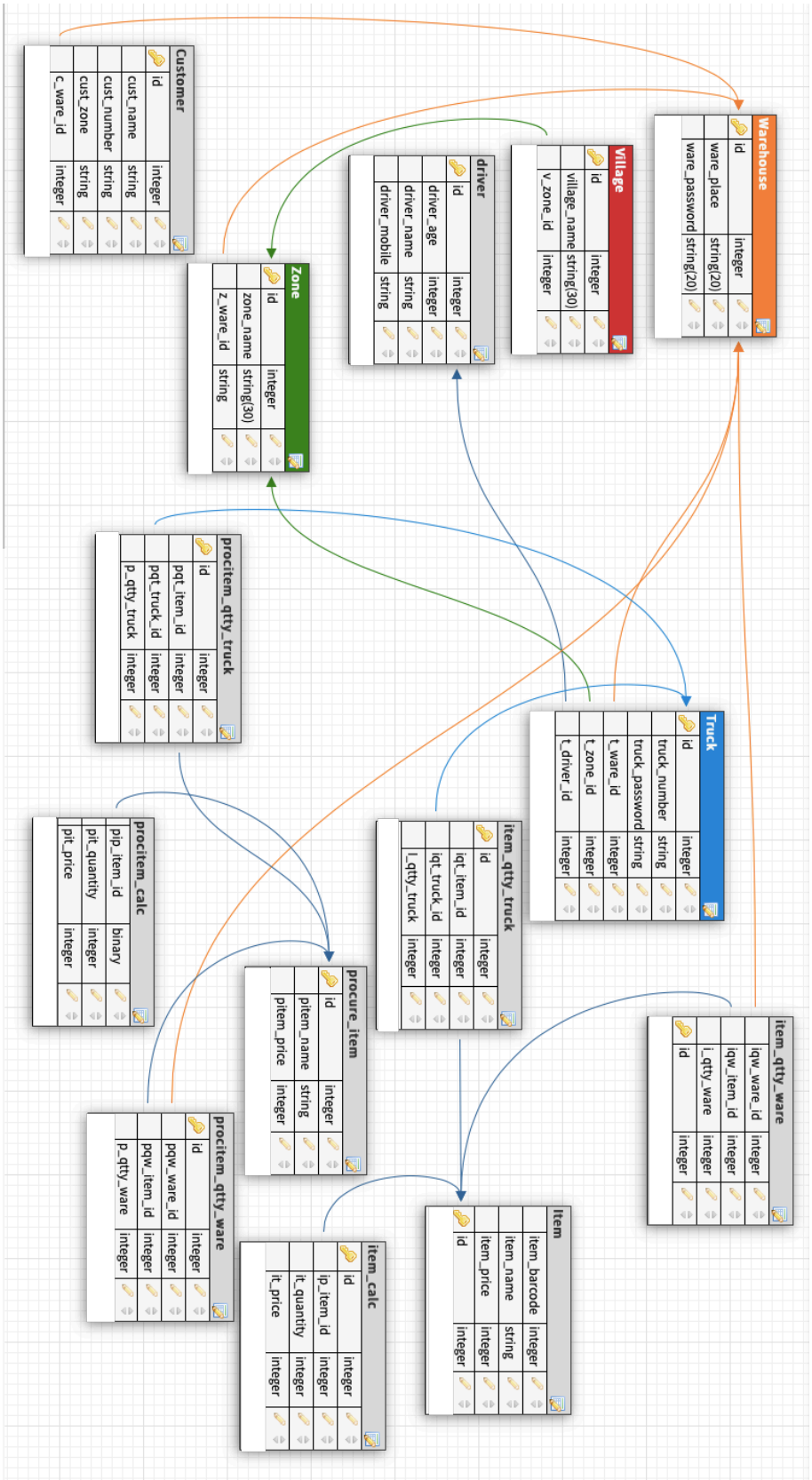
```
{% extends 'base.html' %}
<!DOCTYPE html>
<html>
<head>
    <title>SMW</title>
</head>
<body>
{% block content %}
<center>
    <form method='POST'>
    {% csrf_token %}
    <br>
    <centre><h1>Item Update</h1></centre>
    <div class="row justify-content-center" >
        <div class="col-lg-4 col-md-10 col-sm-10">
            <div class="card">
                <div class="card-body">
    <b>Customer Name:</b> {{cust.cust_name}}<br><br>
    <div class='form-group'>

        <label for='id_i_Id'>Item</label>

        <select id='id_i_id' name='i_id'>
            {% for i in i_list %}
                <option value= "{{ i.id }}">{{ i.item_name }} Rs.{{ i.item_price }}</option>
            {% endfor %}
    </div>
    <div class='form-group'>
        <label for='id_i_unique'>Enter the quantity</label>
        <input type='number' class='form-control' id='id_i_unique' name='i_unique'>
    </div>

    <center><button type="submit" class="btn btn-success">Add item</button>
    </form><br><br>
    <b>Total Price :</b> {{ final_total }}
    <br>
</div>
</div>
</div>
</div>
<center><a href="{%url 'sell_billing' pk_t cust.id final_total %}"><button type="submit" class="btn btn-success">Generate bill</button></a
><br><br><br>
{% endblock %}
</body>
```

This is a code-snippet of the HTML page of the selling item page.

```
{% block content %}
    <head>
        <h4 style="color:#BEE307;"><center><b>Market to your doorstep</b></center></h4>

<style type="text/css">

</style>
</head><br>
    <center>

        <div>
            <marquee direction = "right"><img src="static\img\cart.jpg"  style="height: 100px; width: 100px;"/></marquee><br
            ><br>
    </div>
    <body><center>
        <a href="{%url 'ware_house_login' %}"  style="width: 300px; font-size:20px;background-color:blue;
    "><button class="button"style="width: 500px"><span style='font-size:100px;'>&#127978;</span><br>
            Warehouse Login</h1></button></a>           
        <a href="{%url 'truck_login' %}" style="width: 300px; font-size:20px;background-color:blue"><button class="button"
        style="width: 500px">
            <span style='font-size:100px;'>&#128666;</span><br>Truck Login</h1></button></a><br>

    <!--<div id="watermark5">
            <img src="static\img\gk.jpg"  style=" height: 200px; width: 300px;opacity: 1;"/>
    </div>-->
</center>
    </div>
</body>
{% endblock %}
</html>
```

This is a code-snippet of HOME PAGE

Database Schema



RESULTS

We have successfully developed the interfaces of both the warehouse and trucks with full basic functionalities. We used Django as a backend framework development. Django is a Python-based free and open-source web framework, which follows the MVT architectural pattern. It eases the creation of complex, database-driven websites. Models was used to create classes and declare all the database layout and views was used to have communication between the HTML pages.

We have implemented the model for a touch screen environment with an onscreen Keyboard.

The following are the few snapshots of the actual model we have implemented.

Fig: HOME PAGE

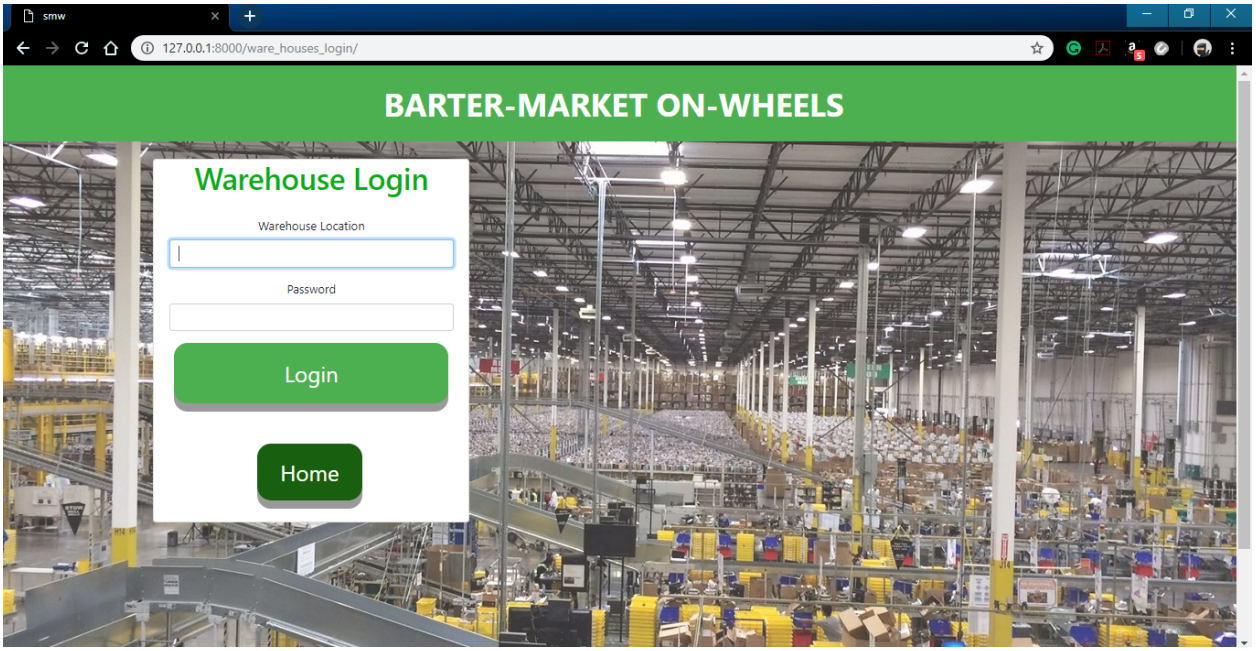


Fig: Warehouse Login Page

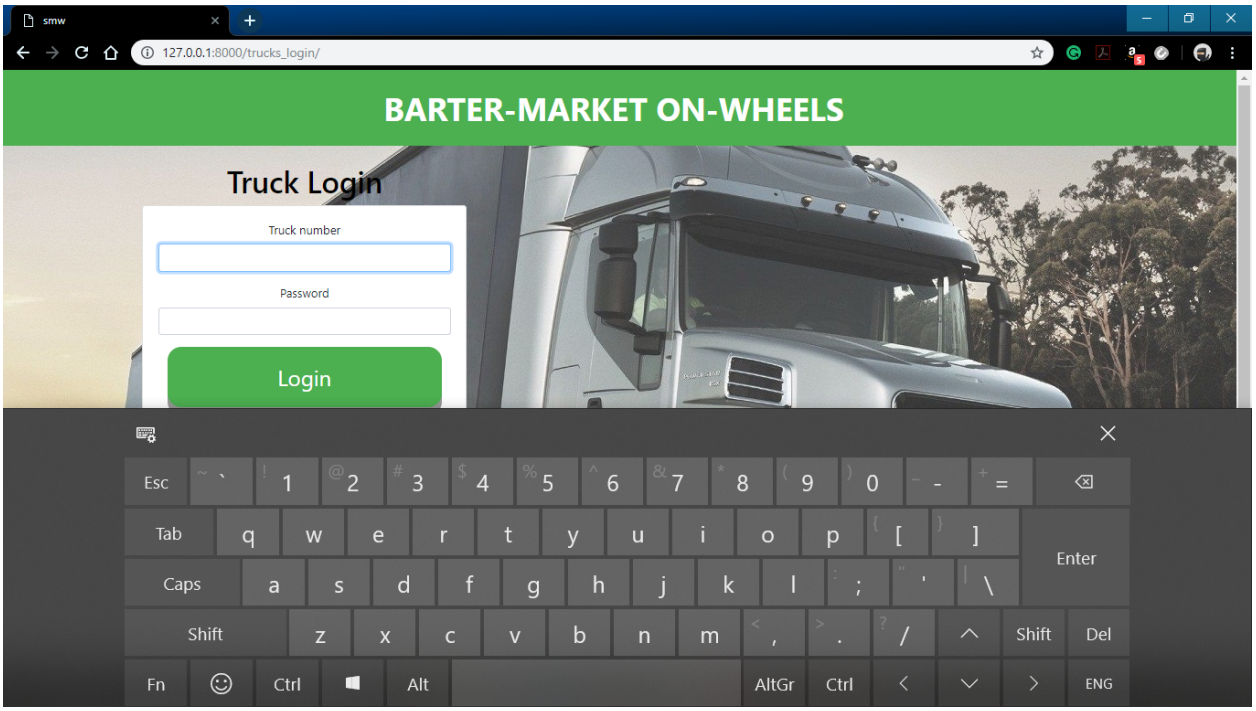


Fig: Truck Login Page with On Screen Touch Keyboard

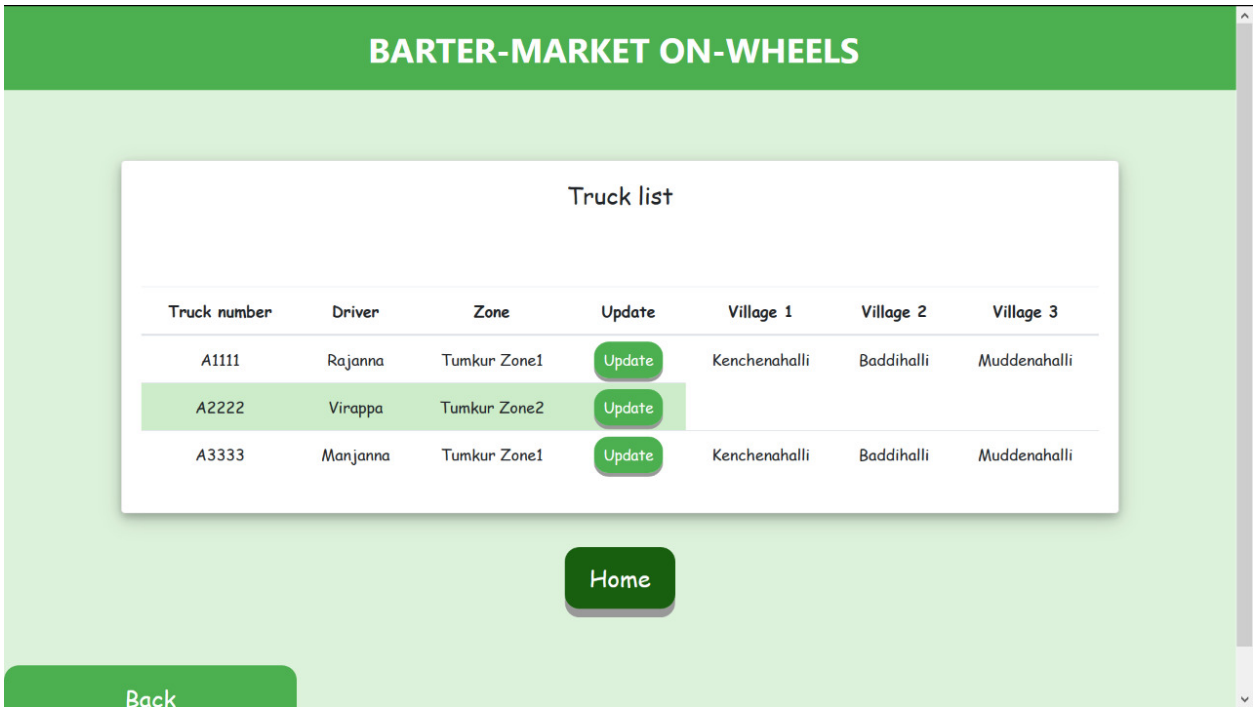


Fig: Truck list to the corresponding Villages and Zones

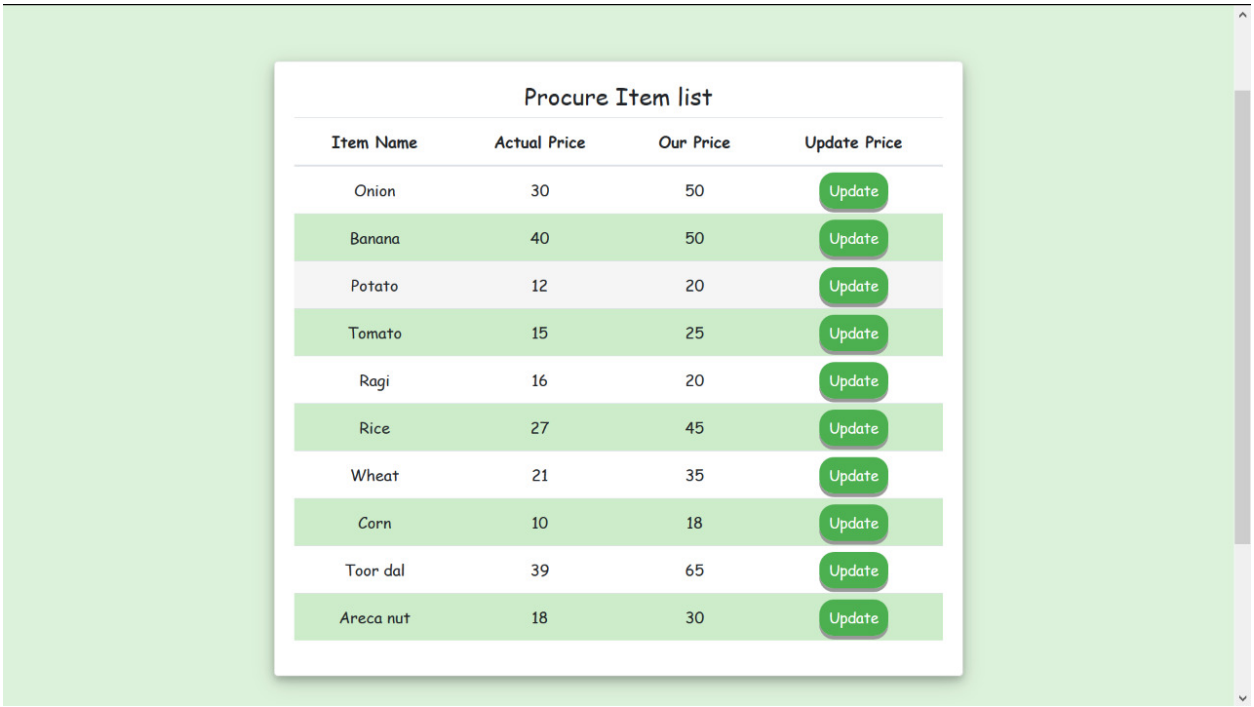


Fig: List of items to be procured with our increased price

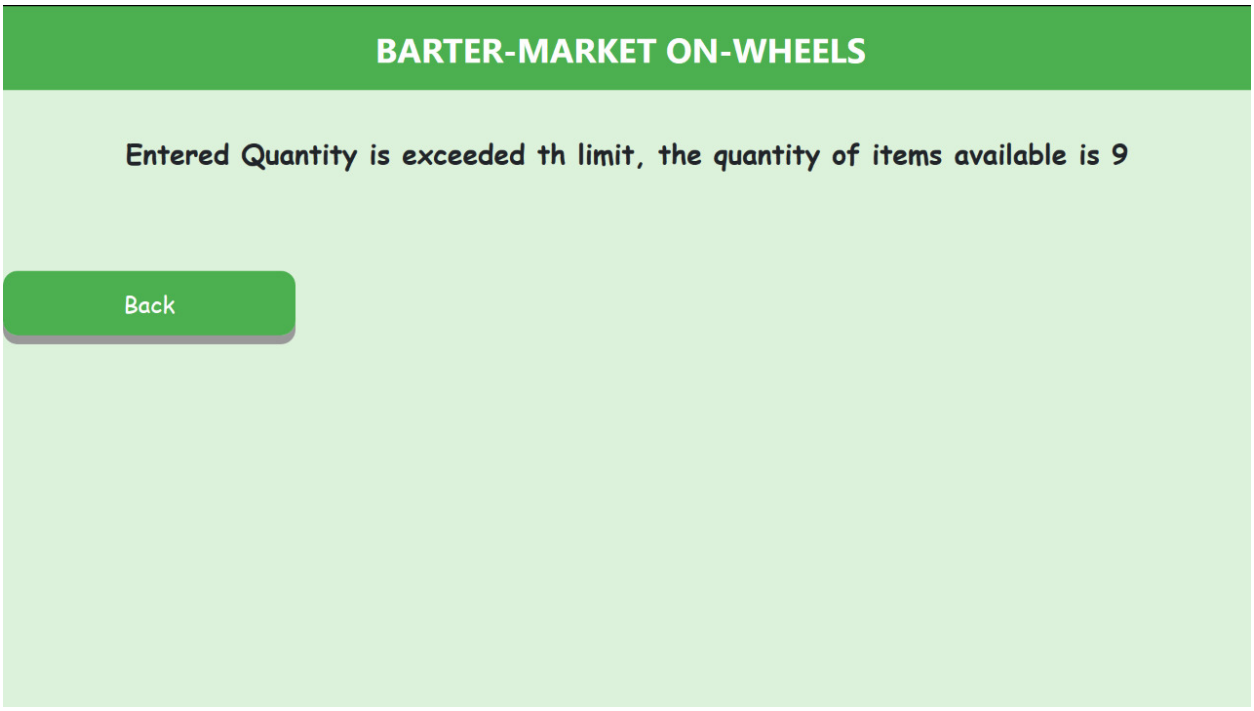


Fig: Error handling if the selling items exceed the availability in warehouse

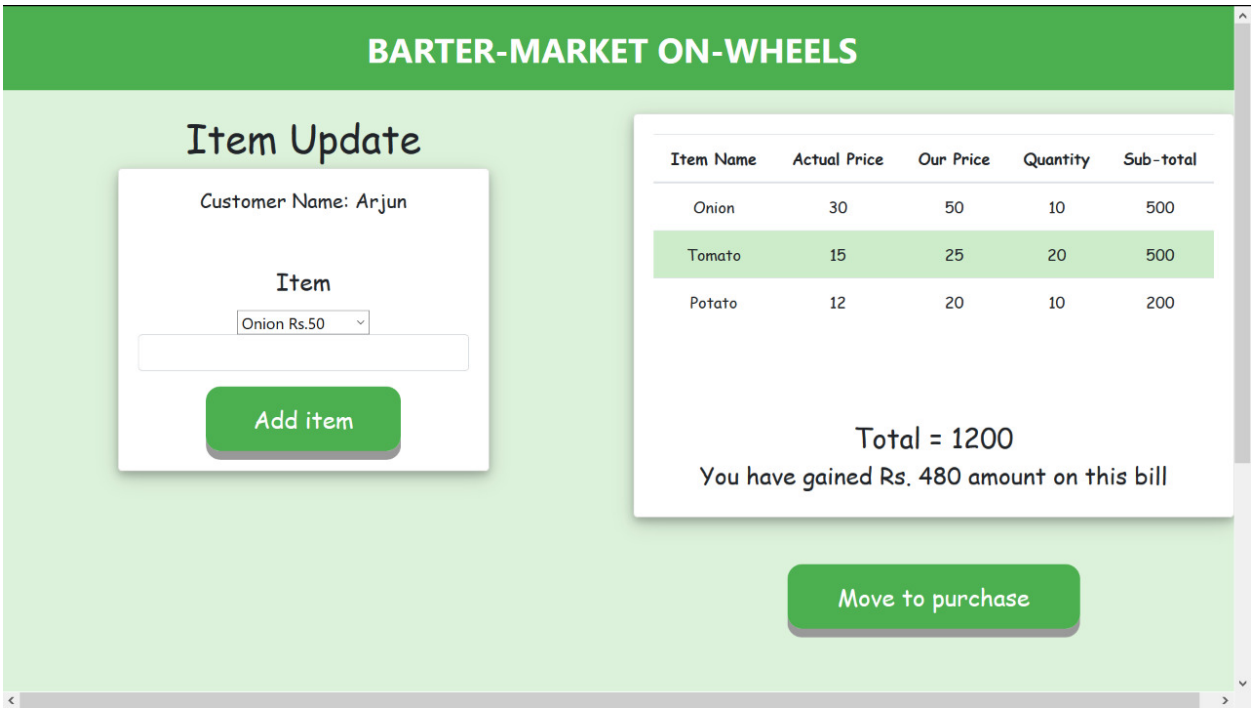


Fig: Procuring the goods from the farmers

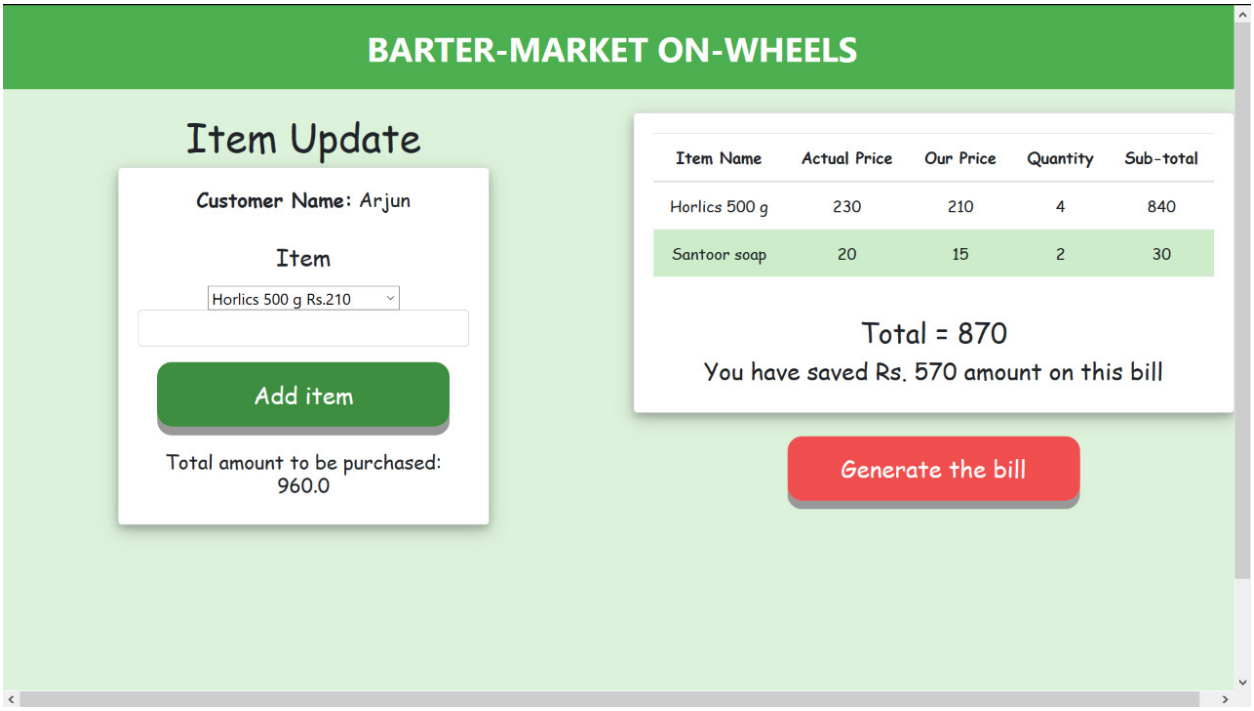


Fig: Sell item where the button remains red and inactive until the amount to be purchased has reached

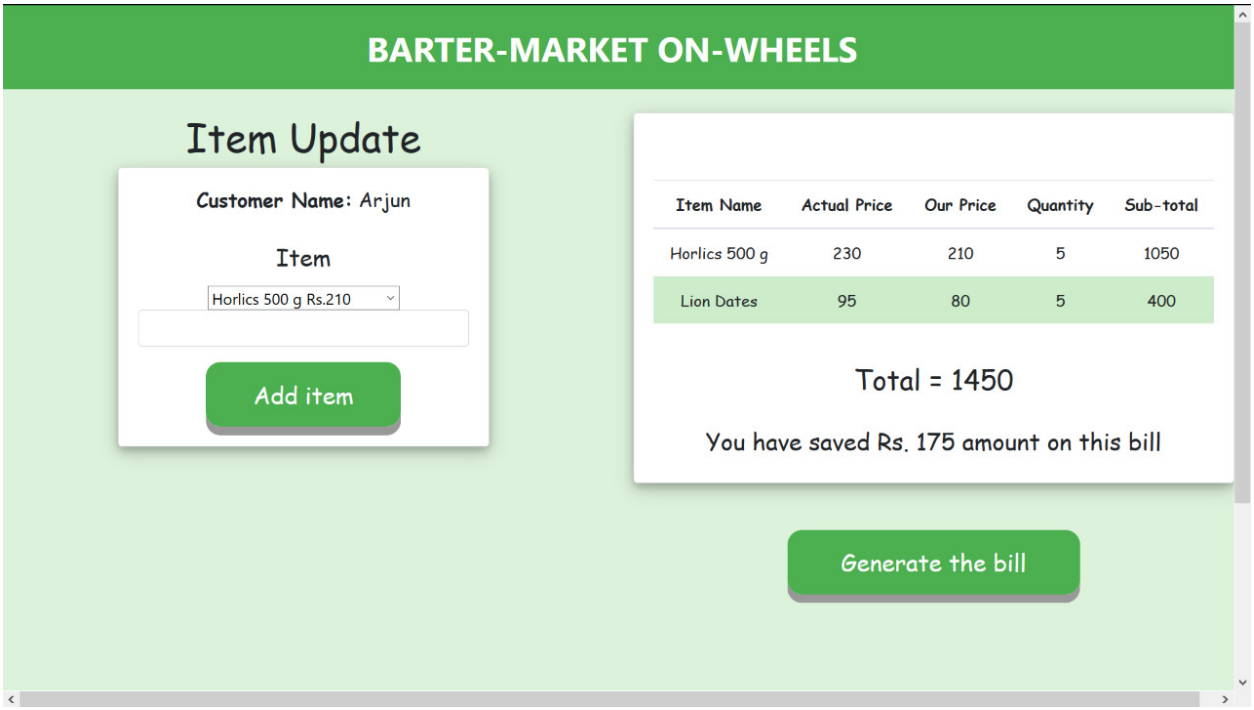


Fig: Sell item where the button turns green and active after the amount to be purchased has reached

CONCLUSION

It can be seen that this idea is a unique, one of a kind project. We are solving major issues related to the farmers of our country. We are providing a platform for them to sell their produces at a higher and profitable rate and also buy items from the supermarket at discounted rates.

It has several advantages over the current system of middle-men and having no exposure to huge markets. Farmers will be getting a higher rates for their crops, they will be able to buy huge variety of items with big discounts, the urban population will be benefited by directly purchasing the procured items from the farmers at a lower price than the market, also the company will be making a profit by selling the items procured as well as selling the goods from the supermarket.

In the future this application will be improved with various technologies, such as Track-the-Truck using GPS, automating the daily update of the crops prices, storing of all the data on the cloud, prediction of the items that will be sold in a village using data analytics and artificial intelligence.

REFERENCES

https://en.wikipedia.org/wiki/Django_(web_framework)

http://www.onebarterindia.com/barter-system-in-india.html

https://www.mint.com/barter-system-history-the-past-and-present

https://www.quora.com/Does-barter-system-exist-in-India

https://www.linkedin.com/pulse/f2c-farmer-consumer-farm-direct-marketing-mandeep-pujara

https://www.tribuneindia.com/news/in-focus/free-farmers-from-middlemen/535083.html

https://cloud.smartdraw.com

https://simpleisbetterthancomplex.com/series/2017/09/11/a-complete-beginners-guide-to-django-part-2.html#introduction

https://en.wikipedia.org/wiki/HTML