

INDEX

Sr. No.	Topic	Page no.
1.	Introduction	2
2.	Existing System	3
3.	Scope of Work	4
4.	Operating Environment i. Hardware Requirement ii. Software Requirement	5
5.	Design i. DFD ii. ER Diagram iii. Data Dictionary	6
6.	Input Screen	10
7.	Sample Code	12
8.	Future Enhancement	17
9.	Limitations	18
10.	Bibliography	19

1. Introduction

An Online Marketplace ShopNow is a digital platform. An Online Marketplace ShopNow connects buyers and sellers provides a platform where registered user can buy and sell products or items online. The system has four types of users: Administrator, Buyer, Seller and Visitor the Seller can upload any item on the site in order to sell it. The Administrator is responsible for the authentication of the user and the product both every component on the website is under the control of the Administrator. The administrator has full authority to insert, update or delete any user or product from the website. It contains online payment service which enables trade on local, national and international basis. It allows online platform where millions of items can be traded every day.

Online Marketplace ShopNow projects can take various forms, including horizontal marketplaces that cover a wide array of product categories (e.g., Amazon), vertical marketplaces specializing in specific niches (e.g., Airbnb for lodging), or peer-to-peer marketplaces that facilitate direct transactions between users (e.g., eBay).

An Online Marketplace ShopNow project serves as a digital hub that fosters commerce, enables entrepreneurship, and transforms the way people buy and sell products and services in the digital age.

2. Existing System

The existing system of an online marketplace project typically encompasses the technology and infrastructure that enables the platform to operate. An online marketplace is working manually. The current system is time consuming and also it is very costly, because it involves a lot of paperwork. To manually handle the system was very difficult task. But now-a-days computerization made easy to work.

The following are the reasons why the current system should be computerized:

- To increase efficiency with reduced cost.
- To reduce the burden of paper work.
- To save time management for recording details of each and every buyer and seller.
- To showcase the product systematically.

3. Scope of Work

Marketplace ShopNow is developed for complete selling and buying of any product uploaded on the site. Every type of user having authenticated username and password can access this system. Unlimited and any type of product can be uploaded for selling. Every user has his private control panel in which he can see and update his information and get it verified by the Administrator.

The scope of work for an Online Marketplace ShopNow System project involves defining the specific tasks, features, and deliverables required to develop and implement the system successfully.

1. Add new category related to products.
2. User registration.
3. Delete/Update Products.
4. Admin has rights to delete the product.
5. Post Advertisements of the product.

4. Operating Environment

i. Hardware Requirement

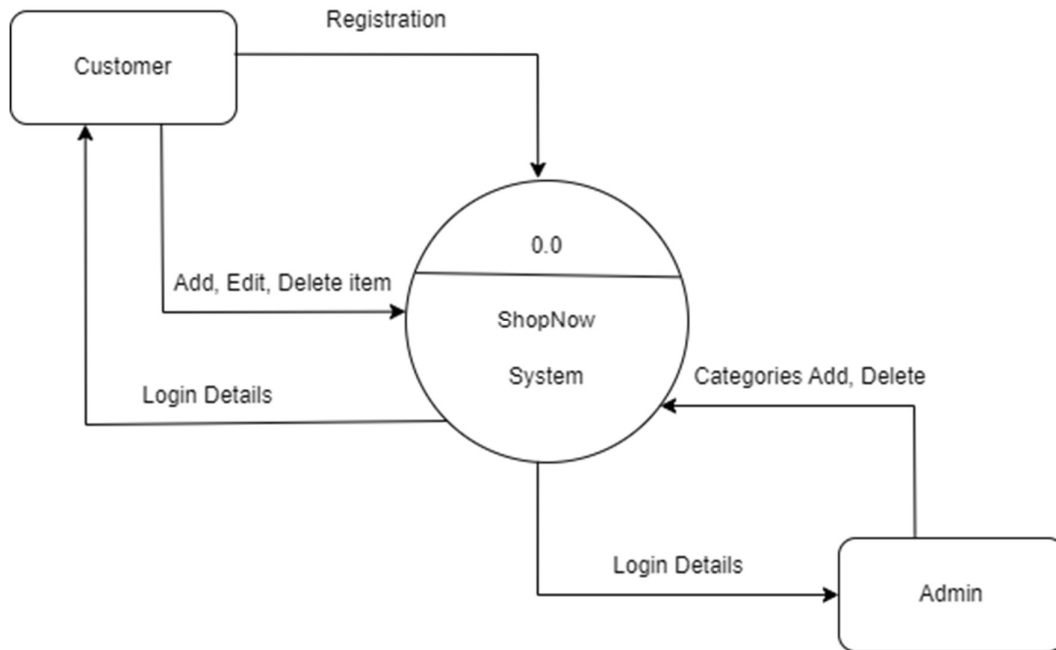
Processor:	Intel Core i3 / any updated
Disc Space:	20GB
RAM:	4GB

i. Software Requirement

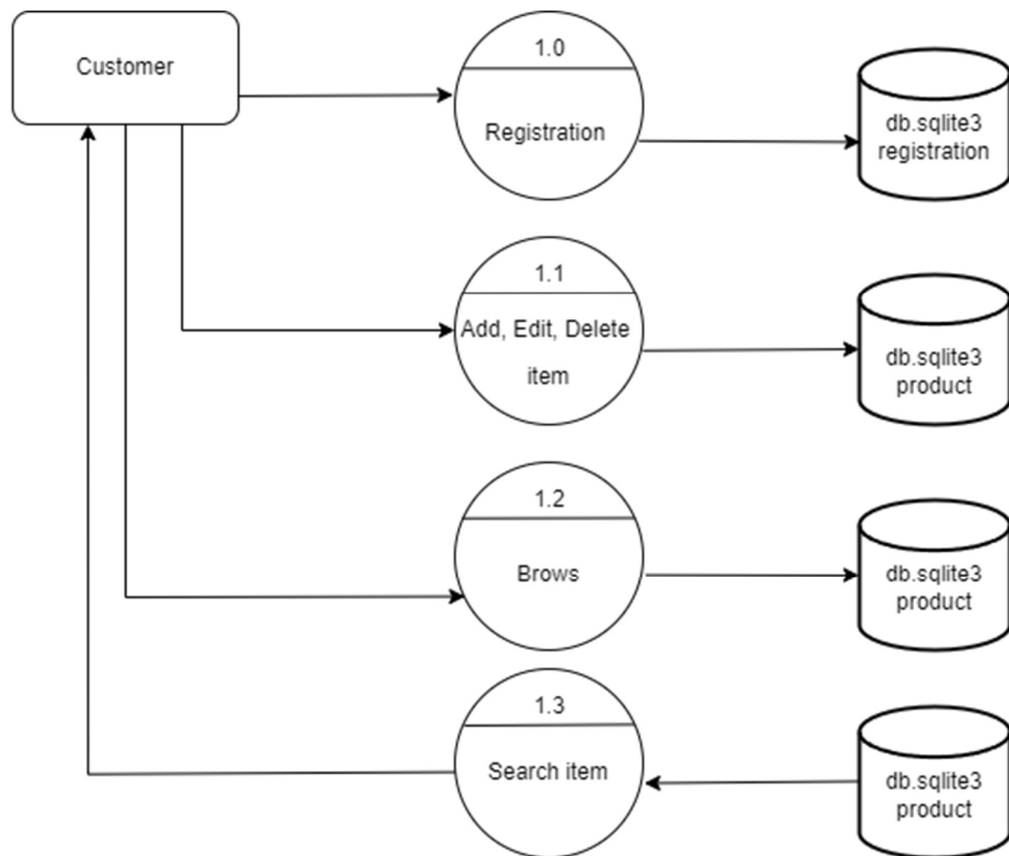
Operating System:	Windows
Development Tools:	VS Code
Development Language:	Python, HTML, CSS, Django

5. Design

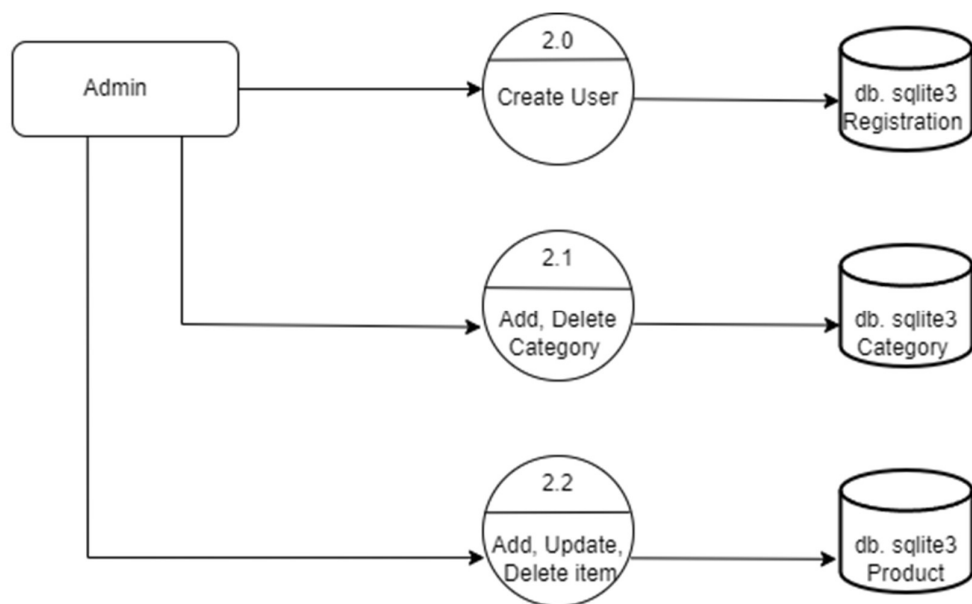
i. DFD



Level 0

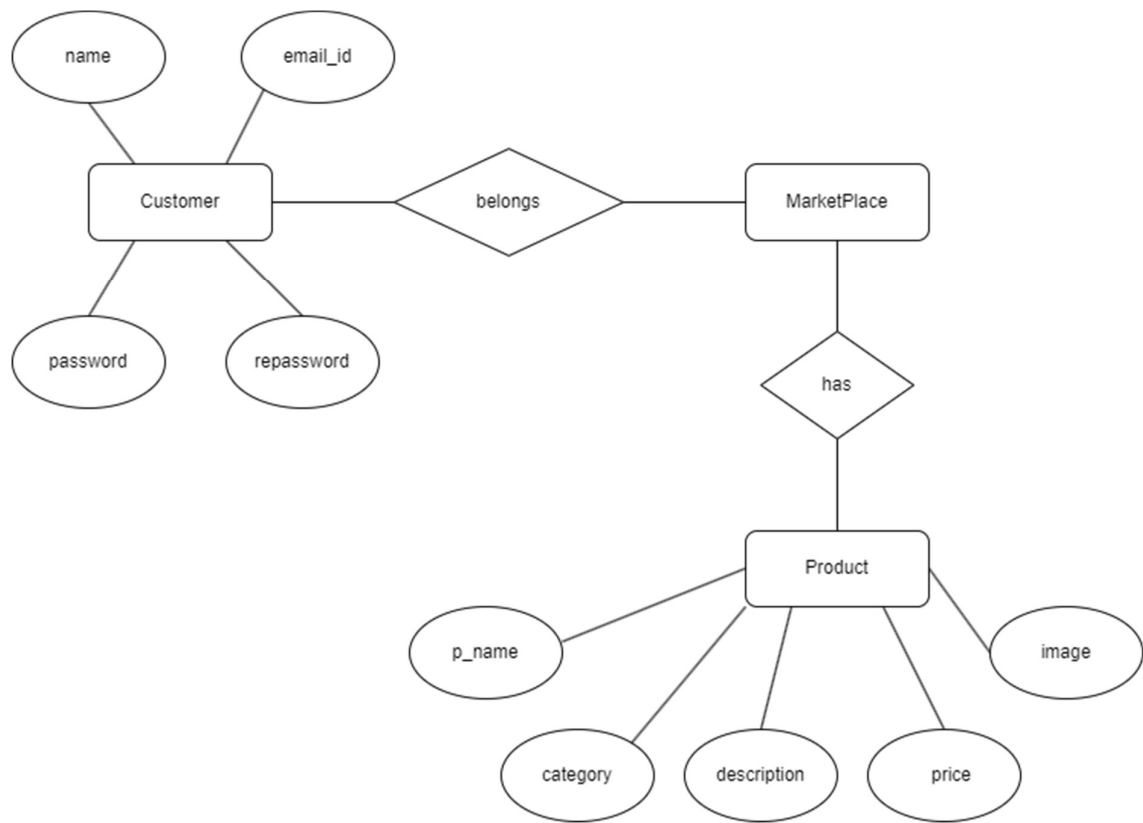


Level 1



Level 2

ii. ER Diagram



iii. Data Dictionary

To create account or login account.

Sr. No	Field Name	Size	Type	Description
1.	Username	225	Char	Username for login.
2.	Email ID	-	email	Email Id for account creation.
3.	Password	-	Password	Password for account security.
4.	Re-Password	-	Password	For two step verification.

To Add product on the website.

Sr. No	Field Name	Size	Type	Description
1.	Name	225	Char	Name of the product.
2.	Description	-	Text	Description about product.
3.	Price	-	Float	Price of product.
4.	Image	-	Image	Product image.

6. Input Screen

ShopNow

New itemBrowseDashboard

Sign up

Username

Enter Username

Email

Enter Email Address

Password

Enter New Password

Return Password

Enter Password Again

Submit

Sign up

ShopNow

New itemBrowseDashboard

Login

Username

Enter Username

Password

Enter New Password

Submit

Login

New Item

Category:

Name:

Description:

Add Product – 1

Price:

Image:

Choose File

No file chosen

Submit

About

This is a MaketPlace platform.

Menu

[About](#)[Contact](#)[Privacy Policy](#)[Term of use](#)

Add Product – 2

7. Sample Code

Base.html: -

```
<!doctype html>

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <script src="https://cdn.tailwindcss.com"></script>

    <title>{% block title %}{% endblock %} | Shopnow</title>

  </head>

  <body>
    <nav class="py-6 px-6 flex justify-between items-center border-b
border-gray-200">
      <a href="/" class="text-xl font-semibold">ShopNow</a>

      <div class="space-x-6">
        <a href="{% url 'item:new' %}" class="text-lg font-semibold
hover:text-grey-500">New item</a>
        <a href="{% url 'item:items' %}" class="text-lg font-semibold
hover:text-grey-500">Browse</a>

        {% if request.user.is_authenticated %}
          <!-- <a href="#" class="px-6 py-3 text-lg font-semibold bg-
teal-500 text-white rounded-xl hover:bg-teal-700">Inbox</a> -->
          <a href="{% url 'dashboard:index' %}" class="px-6 py-3
text-lg font-semibold bg-gray-500 text-white rounded-xl hover:bg-gray-
700">Dashboard</a>
        {% else %}
          <a href="{% url 'core:signup' %}" class="px-6 py-3 text-lg
font-semibold bg-teal-500 text-white rounded-xl hover:bg-teal-700">Sign up</a>
          <a href="{% url 'core:login' %}" class="px-6 py-3 text-lg
font-semibold bg-gray-500 text-white rounded-xl hover:bg-gray-700">Log in</a>
        {% endif %}

      </div>
    </nav>

    <div class="px-6 py-6">
      {% block content %}
      {% endblock %}
    </div>
```

```

    <footer class="py-6 px-6 flex justify-between bg-gray-800">
      <div class="w-2/3 pr-10">
        <h3 class="mb-5 font-semibold text-gray-400">About</h3>

        <p class="text-lg text-grey-500">This is a MaketPlace
platform.</p>
      </div>

      <div class="w-1/3">
        <h3 class="mb-5 font-semibold text-gray-400">Menu</h3>

        <ul class="space-y-2">
          <li><a href="#" class="text-lg text-teal-500 hover:text-
teal-700">About</a></li>
          <li><a href="{% url 'core:contact' %}" class="text-lg
text-teal-500 hover:text-teal-700">Contact</a></li>
          <li><a href="#" class="text-lg text-teal-500 hover:text-
teal-700">Privacy Policy</a></li>
          <li><a href="#" class="text-lg text-teal-500 hover:text-
teal-700">Term of use</a></li>

        </ul>
      </div>

    </footer>
  </body>
</html>

```

Signup.html: -

```

{% extends 'core/base.html' %}

{% block title %}Sign up{% endblock %}

{% block content %}
<div class="w-1/2 my-6 mx-auto p-6 bg-gray-100 rounded-xl">
  <h1 class="mb-6 text-3xl">Sign up</h1>

  <form method="post" action=".">
    {% csrf_token %}

    <div class="mb-3">
      <label class="inline-block mb-2">Username</label><br>
      {{ form.username }}
    </div>

```

```

    <div class="mb-3">
      <label class="inline-block mb-2">Email</label><br>
      {{ form.email }}
    </div>

    <div class="mb-3">
      <label class="inline-block mb-2">Password</label><br>
      {{ form.password1 }}
    </div>

    <div class="mb-3">
      <label class="inline-block mb-2">Return Password</label><br>
      {{ form.password2 }}
    </div>

    {% if form.errors or form.non_field_errors %}
      <div class="mb-3 p-6 bg-red-100 rounded-xl">
        {% for field in form %}
          {{ field.errors }}
        {% endfor %}

        {{ form.non_field_errors }}
      </div>
    {% endif %}

    <button class="py-4 px-8 text-lg bg-teal-500 hover:bg-teal-700
rounded-xl text-white">Submit</button>
  </form>
</div>
{% endblock %}

```

Index.html: -

```

{% extends 'core/base.html' %}

{% block title %}Welcome{% endblock %}

{% block content %}
  <div class="mt-6 px-6 py-12 bg-gray-100 rounded-xl">
    <h2 class="mb-12 text-2xl text-center">New Items</h2>

    <div class="grid grid-cols-3 gap-3">
      {% for item in items %}
        <div>
          <a href="{% url 'item:detail' item.id %}">
            <div>

```

```

        
    </div>

    <div class="p-6 bg-white rounded-b-x1">
        <h2 class="text-2xl">{{ item.name }}</h2>
        <p class="text-gray-500">Price: {{ item.price
    }}</p>
    </div>
</a>
</div>
{% endfor %}
</div>
</div>

<div class="mt-6 px-6 py-12 bg-gray-100 rounded-x1">
    <h2 class="mb-12 text-2xl text-center">Categories</h2>

    <div class="grid grid-cols-3 gap-3">
        {% for category in categories %}
            <div>
                <a href="#">
                    <div class="p-6 bg-white rounded-b-x1">
                        <h2 class="text-2xl">{{ category.name }}</h2>
                        <p class="text-gray-500">{{ category.items.count
    }} items </p>
                    </div>
                </a>
            </div>
        {% endfor %}
    </div>
</div>
{% endblock %}

```

Model.py: -

```

from django.contrib.auth.models import User
from django.db import models

# Create your models here.

class Category(models.Model):
    name = models.CharField(max_length=255)

    class Meta:
        ordering = ('name',)
        verbose_name_plural = 'Categories'

```

```

    def __str__(self):
        return self.name

class Item(models.Model):
    Category = models.ForeignKey(Category, related_name='items',
on_delete=models.CASCADE)
    name = models.CharField(max_length=255)
    description = models.TextField(blank=True, null=True)
    price = models.FloatField()
    image = models.ImageField(upload_to='item_images', blank=True, null=True)
    is_sold = models.BooleanField(default=False)
    created_by = models.ForeignKey(User, related_name='items',
on_delete=models.CASCADE)
    created_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.name

```


8. Future Enhancement

An Online Marketplace ShopNow System, like any software application, can benefit from continuous improvement and future enhancements to meet evolving business needs and technological advancements.

Future enhancements of an Online Marketplace ShopNow project should focus on improving user experience, expanding functionality, and increasing the platform's competitiveness. Regularly updating the system will help maintain its competitiveness in the rapidly evolving e-commerce industry.

Here are some potential future enhancements for an Online Marketplace ShopNow System project:

- **Mobile-First Approach:** As mobile usage continues to rise, consider developing a mobile app or optimizing the existing one to enhance accessibility and convenience for users.
- **AI-Powered Chatbots:** Integrate AI chatbots for improved customer support, handling routine inquiries, and assisting users in finding products or services.
- **AI-Driven Pricing and Inventory Management:** Employ AI for dynamic pricing, inventory management, and demand forecasting to optimize seller operations.

9. Limitations

While an Online Marketplace ShopNow System can provide numerous benefits to marketplace platform centers, it also comes with certain limitations and challenges that should be considered:

- **Competition:** High competition can make it challenging for new marketplaces to gain traction and establish a user base, especially in well-established niches.
- **Trust and Security:** Building trust among users and ensuring the security of transactions and data are ongoing challenges. Instances of fraud, scams, and data breaches can erode trust.
- **Customer Support:** Providing effective customer support for a large user base can be resource-intensive, and issues related to disputes or inquiries may arise frequently.

10.Bibliography

Websites

1. <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django>
2. <http://www.w3schools.com/css/default.asp>
3. <http://www.w3schools.com/html/default.asp>
4. <https://www.w3schools.com/python/>