Definition :

- This dataset provides a detailed view of the product catalog and pricing structure of Zepto, a fast-growing 10-minute grocery delivery platform. The data captures essential attributes for over 3,000+ SKUs (Stock Keeping Units) across various categories like Fruits & Vegetables, Dairy, Packaged Foods, Beverages, and more

The data is structured to support various types of retail analysis, including:

- Discount trends by category
- Inventory availability and stock-outs
- Price distribution and pricing strategy
- Product naming patterns (suitable for word cloud or NLP tasks)

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

from wordcloud import wordcloud
```

In [2]:
```
df=pd.read_csv('zepto_v2.csv',encoding="ISO-8859-1")
df.head()
```

Out[2]:

| | Category | name | mrp | discountPercent | availableQuantity | discountedSellingPrice |
|---|---|---|---|---|---|---|
| **0** | Fruits & Vegetables | Onion | 2500 | 16 | 3 | 2100 |
| **1** | Fruits & Vegetables | Tomato Hybrid | 4200 | 16 | 3 | 3500 |
| **2** | Fruits & Vegetables | Tender Coconut | 5100 | 15 | 3 | 4300 |
| **3** | Fruits & Vegetables | Coriander Leaves | 2000 | 15 | 3 | 1700 |
| **4** | Fruits & Vegetables | Ladies Finger | 1400 | 14 | 3 | 1200 |

Data preprocessing

In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3732 entries, 0 to 3731
Data columns (total 9 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   Category                3732 non-null   object
 1   name                    3732 non-null   object
 2   mrp                     3732 non-null   int64 
 3   discountPercent         3732 non-null   int64 
 4   availableQuantity       3732 non-null   int64 
 5   discountedSellingPrice  3732 non-null   int64 
 6   weightInGms             3732 non-null   int64 
 7   outOfStock              3732 non-null   bool  
 8   quantity                3732 non-null   int64 
dtypes: bool(1), int64(6), object(2)
memory usage: 237.0+ KB
```

In [4]:
```
df.shape
```

Out[4]:
```
(3732, 9)
```

In [5]:
```
df.duplicated().sum()
```

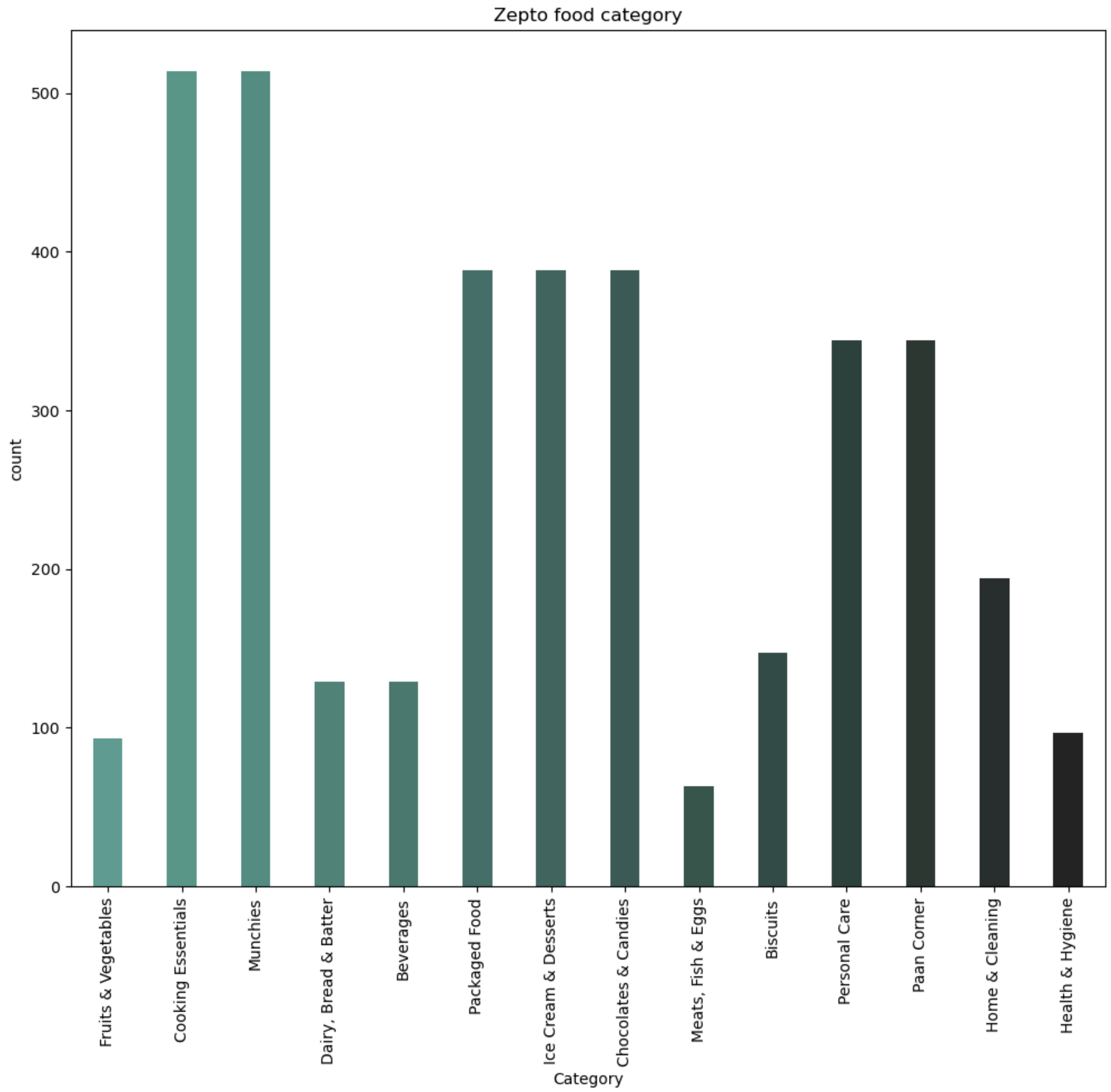
Out[5]:
```
2
```

In [6]:
```
df.isna().sum()
```

Out[6]:
```
Category                  0
name                      0
mrp                       0
discountPercent           0
availableQuantity         0
discountedSellingPrice    0
weightInGms               0
outOfStock                0
quantity                  0
dtype: int64
```

In [15]:
```
plt.figure(figsize=(12,10))
sns.countplot(x='Category',
             data=df,
             width=0.4,
             palette='dark:#5A9_r')
plt.title('Zepto food category')
plt.xticks(rotation='vertical');
```

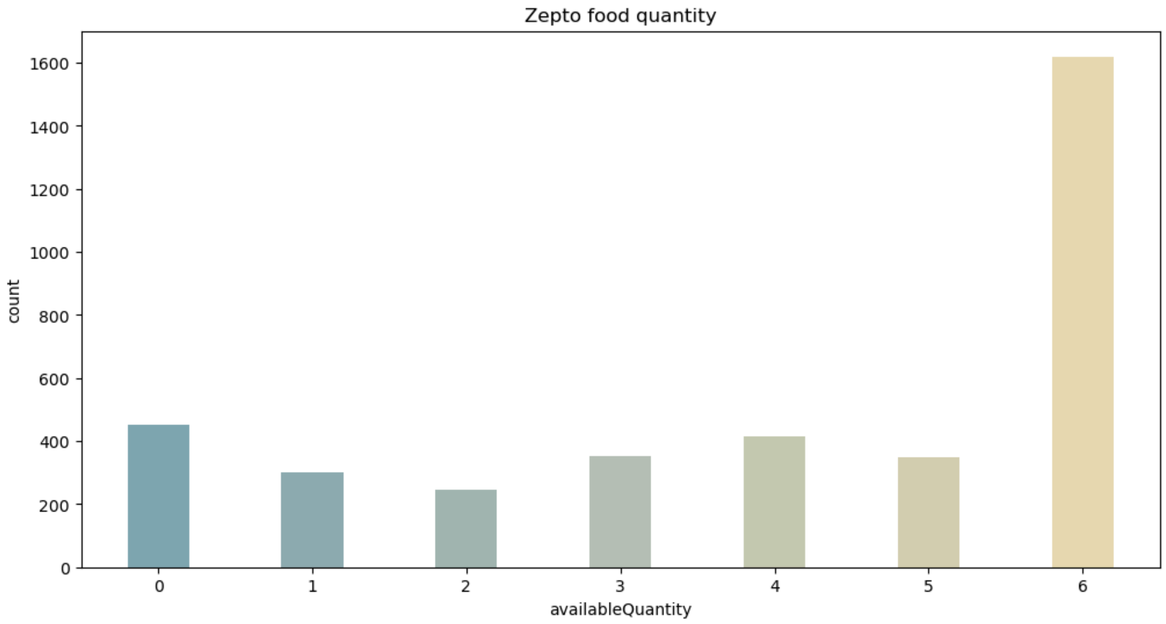
```
C:\Users\ashif\AppData\Local\Temp\ipykernel_13332\128187932.py:2: FutureWa
rning:

Passing `palette` without assigning `hue` is deprecated and will be remove
d in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for
the same effect.

  sns.countplot(x='Category',
```




In [8]:

```
plt.figure(figsize=(12,6))
sns.countplot(x='availableQuantity',
              data=df,
              width=0.4,
             palette="blend:#7AB,#EDA")
plt.title('Zepto food quantity')
plt.xticks(rotation='horizontal');
```

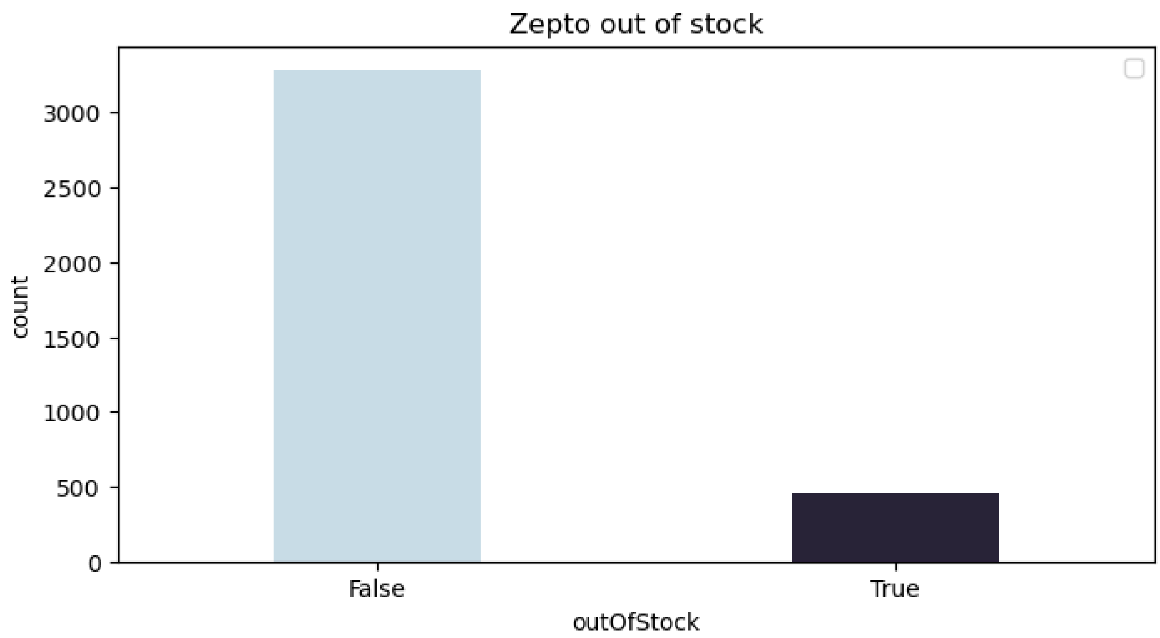
```
C:\Users\ashif\AppData\Local\Temp\ipykernel_13332\322488579.py:2: FutureWa
rning:

Passing `palette` without assigning `hue` is deprecated and will be remove
d in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for
the same effect.

  sns.countplot(x='availableQuantity',
```




In [17]:

```
plt.figure(figsize=(8,4))
sns.countplot(x='outOfStock',
              data=df,
              palette='ch:s=.25,rot=-.25',
             width=0.4)
plt.legend()
plt.title('Zepto out of stock');
```

```
C:\Users\ashif\AppData\Local\Temp\ipykernel_13332\1871124277.py:2: FutureW
arning:

Passing `palette` without assigning `hue` is deprecated and will be remove
d in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for
the same effect.

  sns.countplot(x='outOfStock',
C:\Users\ashif\AppData\Local\Temp\ipykernel_13332\1871124277.py:6: UserWar
ning: No artists with labels found to put in legend.  Note that artists wh
ose label start with an underscore are ignored when legend() is called wit
h no argument.
  plt.legend()
```




In [10]:
```
df.head()
```

Out[10]:

| | Category | name | mrp | discountPercent | availableQuantity | discountedSellingPrice |
|---|---|---|---|---|---|---|
| **0** | Fruits & Vegetables | Onion | 2500 | 16 | 3 | 2100 |
| **1** | Fruits & Vegetables | Tomato Hybrid | 4200 | 16 | 3 | 3500 |
| **2** | Fruits & Vegetables | Tender Coconut | 5100 | 15 | 3 | 4300 |
| **3** | Fruits & Vegetables | Coriander Leaves | 2000 | 15 | 3 | 1700 |
| **4** | Fruits & Vegetables | Ladies Finger | 1400 | 14 | 3 | 1200 |

In [18]:
```
Out_of_stock_items=df[df["outOfStock"]==True]
Out_of_stock_items[["name","quantity"]].sort_values(["quantity"],ascendin
```

Out[18]:

| | name | quantity |
|---|---|---|
| **606** | Everest Saffron Kesar | 0 |
| **1120** | Everest Saffron Kesar | 0 |
| **1110** | 24 Mantra Organic Whole Wheat Atta | 1 |
| **3625** | Gala No Dust Broom | 1 |
| **3623** | Chakaachak Super Steel Scrubber | 1 |
| **...** | ... | ... |
| **3076** | Dettol Original Liquid Handwash Refill | 750 |
| **3629** | Comfort After Wash Fabric Conditioner Lily Fresh | 860 |
| **2141** | Del Monte Original Blend Tomato Ketchup Pouch | 950 |
| **1753** | Del Monte Original Blend Tomato Ketchup Pouch | 950 |
| **2529** | Del Monte Original Blend Tomato Ketchup Pouch | 950 |

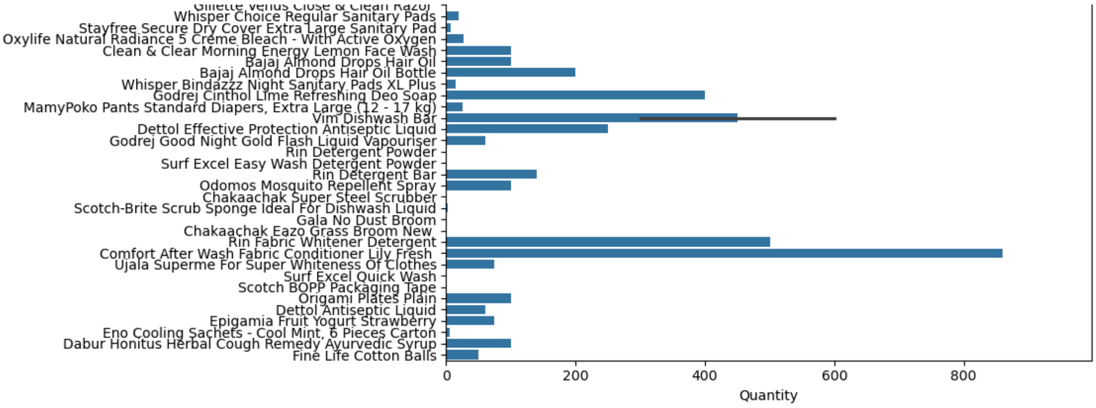
453 rows × 2 columns

In [21]:

```
plt.figure(figsize=(12, 25))  # Tall figure
sns.barplot(data=Out_of_stock_items, y="name", x="quantity", dodge=False)

plt.title("Out-of-Stock Items and Their Quantities")
plt.xlabel("Quantity")
plt.ylabel("Item Name")
plt.tight_layout()
plt.show()
```

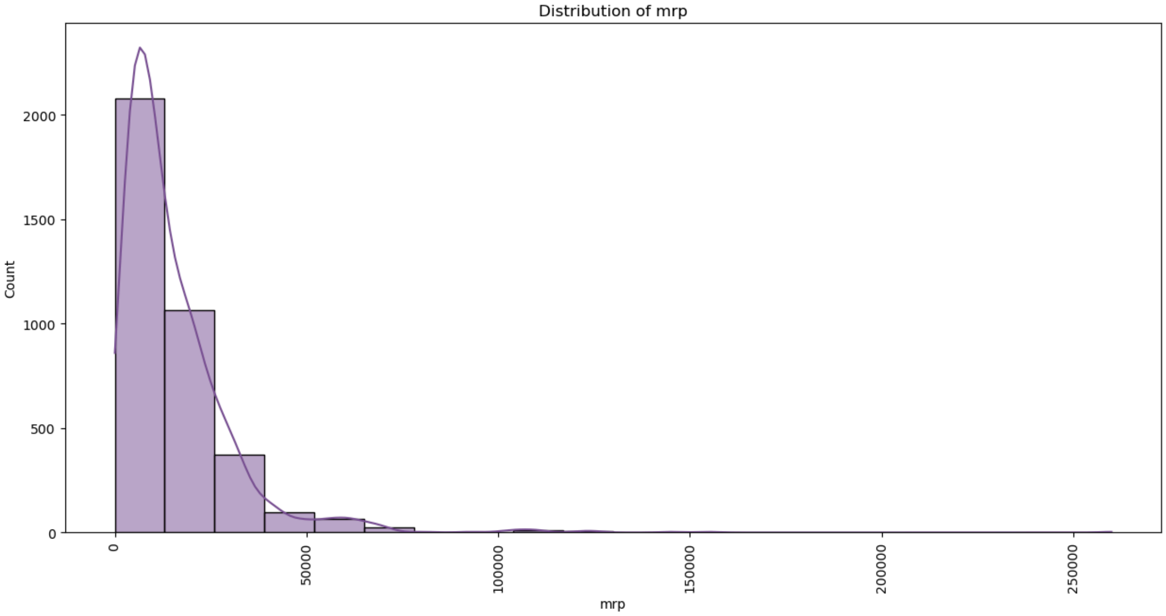
```
C:\Users\ashif\AppData\Local\Temp\ipykernel_13332\590317956.py:7: UserWarn
ing: Glyph 150 (\x96) missing from font(s) DejaVu Sans.
  plt.tight_layout()
C:\Users\ashif\anaconda3\Lib\site-packages\IPython\core\pylabtools.py:170:
UserWarning: Glyph 150 (\x96) missing from font(s) DejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
```

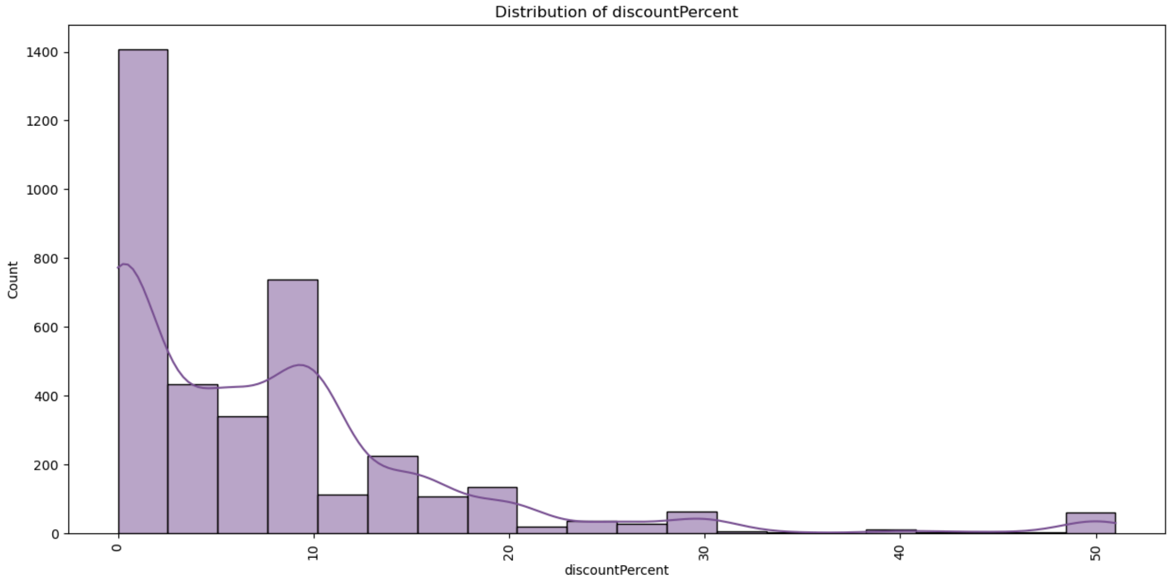

Out-of-Stock Items and Their Quantities
Item Name
Carrot
Beetroot
Moong Sprouts
Safal Frozen Mixed Vegetables
Safal Frozen Corn
Safal Frozen - Sweet Corn
Madhur Sugar
Amul Fresh Cream
Madhur Pure And Hygienic Sugar
Satyam Poha Jada
Aashirvaad Superior MP Whole Wheat Atta
Fortune Soyabean Oil
Maggi Masala Ae Magic
Aashirvaad Iodised Salt
Maggi 2-Minute Instant Noodles - Masala
Popular Essentials Phool Makhana (Jumbo Size)
Aashirvaad Atta With Mutigrains
Tata Sampann Poha
Fortune Besan Flour
Popular Essentials Toor Dal (Un-Polished)
Everest Tikhalal Hot & Red Chilly Powder
Popular Essentials Chia Seeds
Tata Sampann Unpolished Toor Dal/Arhar Dal
Everest Coriander Powder
Satyam Thin Poha Patla
Popular Essentials Sona Masouri Raw Rice
Tata Sampann High Protien Chana Dal - Split
Satyam Singdaha
Tata Sampann Moong Chilka
Tata Sampann High Protien Toor Dal - Split
Ching's Chow Mein Hakka Noodles Masala
Everest Chat Masala
Popular Essentials Black Brown Chana (Un-Polished)
Dabur Ginger Garlic Paste
Ching's Secret Veg Hakka Noodles
Saffola Oats 100 % Natural
Everest Kashmiri Lal Chilli Powder
Popular Essentials Black Raisins Seedless
Daawat Rozana Super Basmati Rice
Everest Cumin Powder (Jeera)
Everest Sambhar Masala
Maggi Chilly Chow Cuppa Noodles
Popular Essentials Mustard Seeds Rai Big
Patanjali Cow's Ghee
Everest Amchur Powder
Daawat Super Basmati Rice
Popular Essentials Sabudana
Popular Essentials Brown Sugar
Sunfeast Yippee! Magic Masala Noodles
Tata Tea Gold
Daawat Rozana Gold Basmati Rice
Top Ramen Masala
Everest Turmeric Powder
Popular Essentials Buckwheat Flour (Kuttu Atta)
Keya Italian Pizza Oregano
Chef's Basket Durum Wheat Penne Pasta
Tata Sampann Kala Chana
India Gate Basmati Rice
24 Mantra Organic Whole Wheat Atta
Zorabian Chicken Kheema Parathas
Sunfeast Pasta Treat Pack of Tomato Cheese
Sugarlite 50% Less Calories Sugar Pouch
Popular Essentials Sabja/Basil Seeds
Patanjali Pure Honey
Weikfield Falooda Mix - Mango
Pro Nature Organic Sonamasoori Brown Rice
Maggi Special Masala Spicy Yummy Noodles
Godrej Yummiez Chicken Breakfast Salami
Everest Saffron Kesar
Amul Malai Paneer Fresh
Amul Taaza Toned Homogenised Milk
Amul Lactose Free Milk
The Bake Shop Whole Wheat Bread
Amul Slim Trim Skimmed Milk
Amul Cheese Slices
Amul Pizza Cheese - Mozzarella
The Bake Shop Pizza Base Whole Wheat
Nestle A+ Slim Dahi - Low Fat
Britannia Cheese Garlic Bread
The Bake Shop Whole Wheat Ladi Pav
Amul Cheese Spread Yummy Plain
Britannia Atta Kulcha
Suguna Nourish Egg
The Bake Shop Multi Grain Bread
Nestle A+ Slim Milk
Go Cheese Pizza Blend Mozzarella + Cheddar
The Whole Truth Peanut Butter - Unsweetened & Crunchy
Amul Cheese Spread Spicy Garlic
The Bake Shop Burger Buns
D'lecta Shredded Mozzarella Cheese
Britannia Burger Bun with Sesame Seed
Suguna Vitamin D Egg
Pillsbury Pancake Mix - Funfetti
Prabhat Malai Paneer Pouch
Suguna Nourish Egg
RRO Mozzarella Block Cheese
Kurkure Green Chutney, Rajasthani Style
Kellogg's Chocos Combo Pack
Kissan Mixed Fruit Jam
Lijjat Urad Papad
Mother's Recipe Ginger And Garlic Paste
Maggi Magic Cubes Chicken
Pedigree Puppy Chicken Chunks in Gravy
Lijjat Moong Papad
Mothers Recipe Papad - Sabudana With Rock Salt
Veeba Favourites Eggless Burger Mayonnaise
Kelloggs Corn Flakes With Real Strawberry Pure
Mother's Recipe Potato Papad
Mothers Recipe Rice Papad Ajwain
Dabur Hommade - Imli Sauce Saunth
MTR Vegetable Upma
Del Monte Original Blend Tomato Ketchup Pouch
Del Monte Natural Vinegar
Del Monte Eggless Mayo
ITC Master Chef Cheesy Corn Triangles
Mother's Recipe Mixed Pickle
Hershey's Cocoa + Almond Spread
Del Monte Schezwan Sauce
Yummiez Green Peas
Yummiez American Sweet Corn
Dr Oetker Cucumber & Carrot Sandwich Spread Veg
Del Monte Pitted Green Olives
Fresh Chicken Curry Cut - Skinless (Halal)
Zorabian Chicken Breast Boneless
Fresh Chicken Mince - Kheema (Halal)
Godrej Real Good Chicken Boneless Cubes
Zorabian Chicken Mince
Zorabian Chicken Leg Boneless
Zorabian Chicken Cubes
Good Day Butter Cookies
Britannia Marie Gold Biscuits Packet
Britannia Good Day Cashew Cookies
Sunfeast Dark Fantasy Choco Fills
Britannia Nutri Choice Digestive Biscuits Packet
Britannia Classic Little Hearts
Britannia Good Day Cashew Cookies Packet
Britannia Nice Time Biscuit
Britannia Milk Bikis Cream Biscuit
Britannia Bourbon Chocolate Cream Biscuits
Dukes Waffy Chocolate Wafer Rolls
Britannia Milk Bikis Milky Sandwich
Britannia Marie Gold Biscuit
Britannia Marie Gold (Buy 4 Get 1 Free)
Britannia 50-50 Maska Chaska Biscuit
Unibic Choco Nut Cookies
Britannia Vita Marie Gold Biscuit
The Bake Shop Maska Khari
Parle Fab! Jam-In Cream Filled
Sunfeast Nice Biscuit
Britannia Nutrichoice 5 Grain Digestive Biscuits
Cadbury Oreo Strawberry Bicuits
Sunfeast Mom's Magic Cashew & Almond Biscuit
Britannia Roll Yo! Strawberry Swiss Roll
Bauli Moonfils Choco Cream (With Egg)
Britannia Nutri Choice Oats Choco Almond Biscuit
Britannia Nutri Choice Cracker Sugar Free Biscuit
Cadbury Oreo Chocolatey Sandwich Biscuits Family Pack
Bauli Moonfils Orange Puff Roll
Britannia Treat Jimjam Cream Biscuits
Britannia Toastea Premium Bake Rusk
The Bake Shop Jeera Khari
Cadbury Oreo Creme Biscuit - Vanilla, Original 120 g
Bauli Moonfils - Vanilla
Britannia Roll Yo! Choco Swiss Roll (With Egg)
Open Secret Choco Almond Story Box
Parle Monaco Sixer Salted Biscuits
Sunfeast Dark Fantasy Bourbon Bliss
Patanjali Atta Doodh Rusk
Dettol Original Liquid Handwash Refill
Parachute 100% Pure Coconut Oil
Stayfree Secure Cottony Soft Cover Sanitary Pad (Regular) with Wings
Dabur Gulabari Premium Rose Water
Parachute Advansed Gold Coconut Hair Oil
Closeup Everfresh+ Anti-Germ Gel Toothpaste Red Hot
Godrej No.1 Bathing Soap Lime & Aloe Vera
Sensodyne Sensitive Fresh Gel Toothpaste
Whisper Bindazzz Nights XL + Sanitary Pad
Whisper Choice Sanitary Pad
Godrej No.1 Bathing Soap, Jasmine

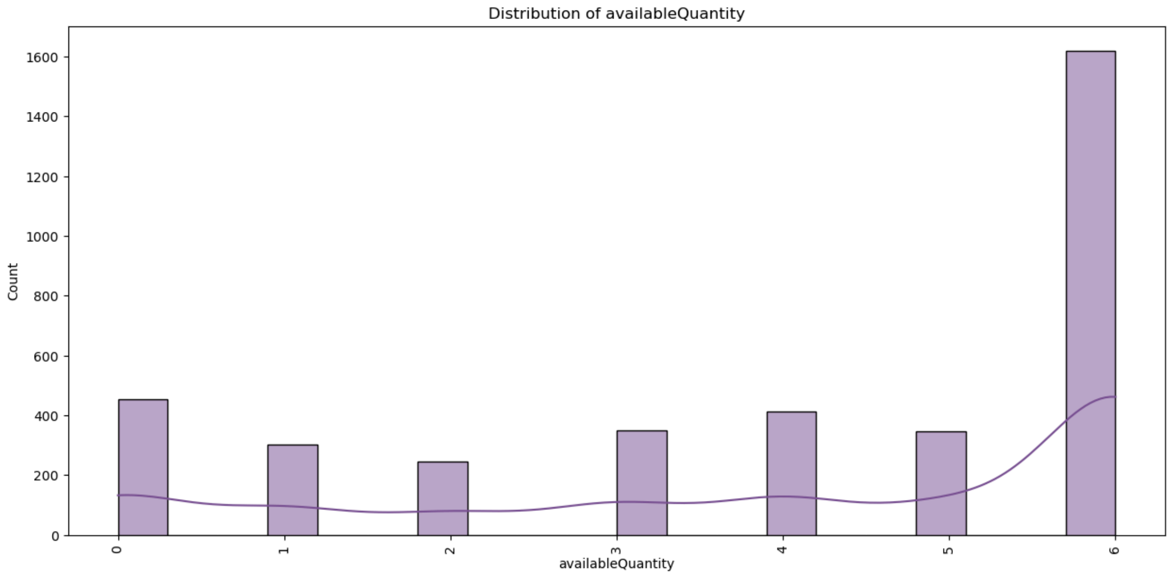


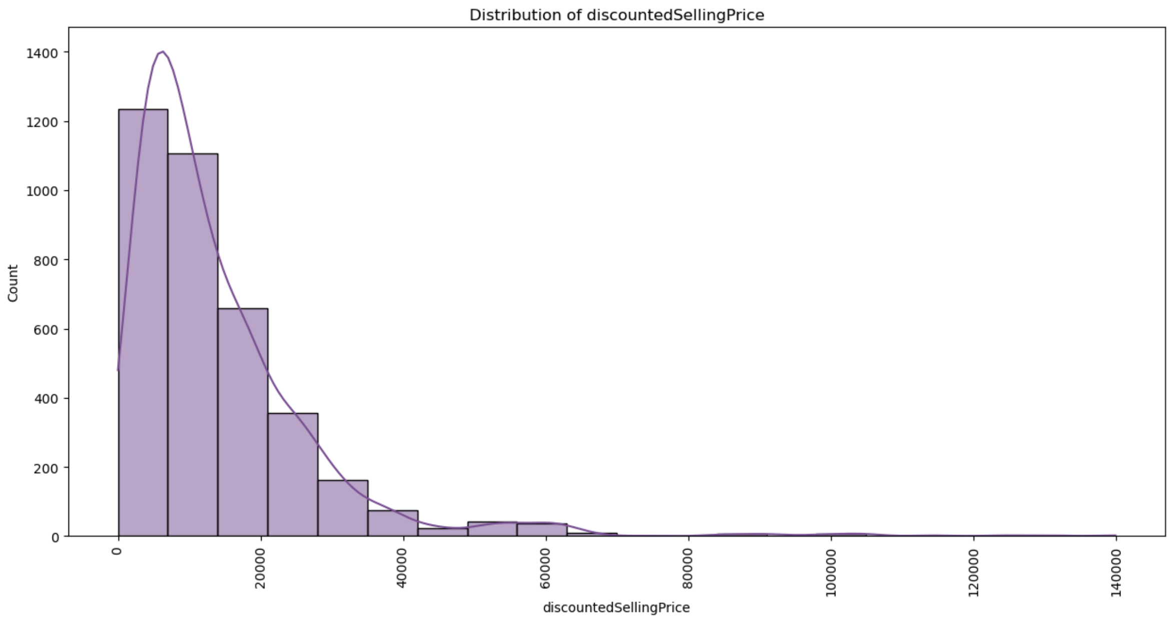

In [27]:

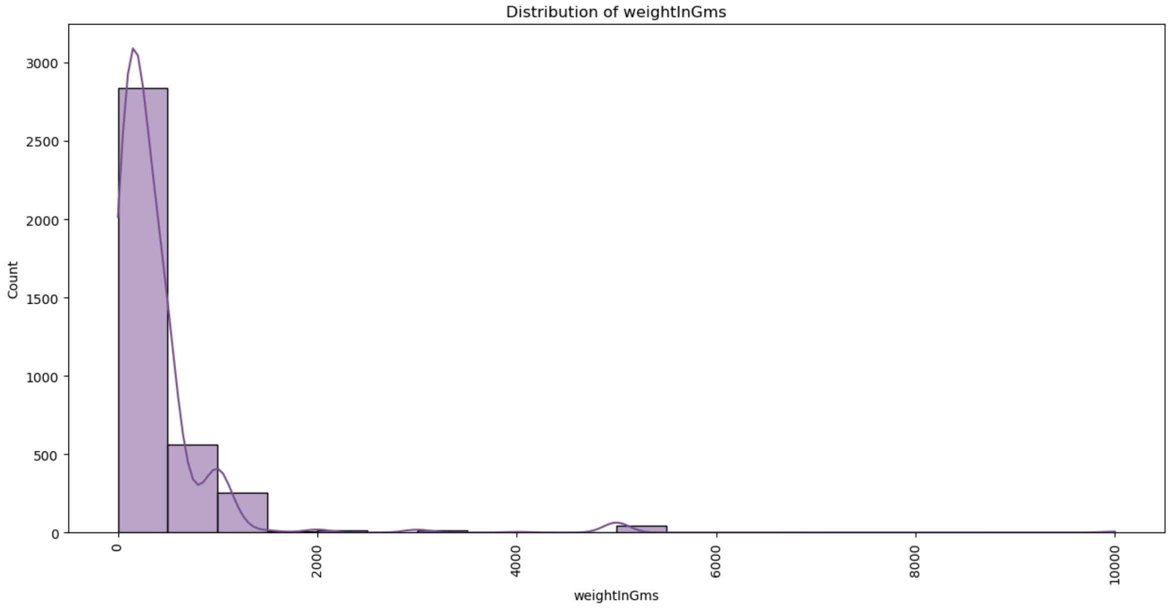
```python
d=df.select_dtypes(include='number')
for col in d:
    plt.figure(figsize=(15,7))
    sns.histplot(data=d,
    x=col,
    kde=True,
    bins=20,
    color='#7a5195')
    plt.title(f'Distribution of {col}')
    plt.xticks(rotation=90)
    plt.show()
```

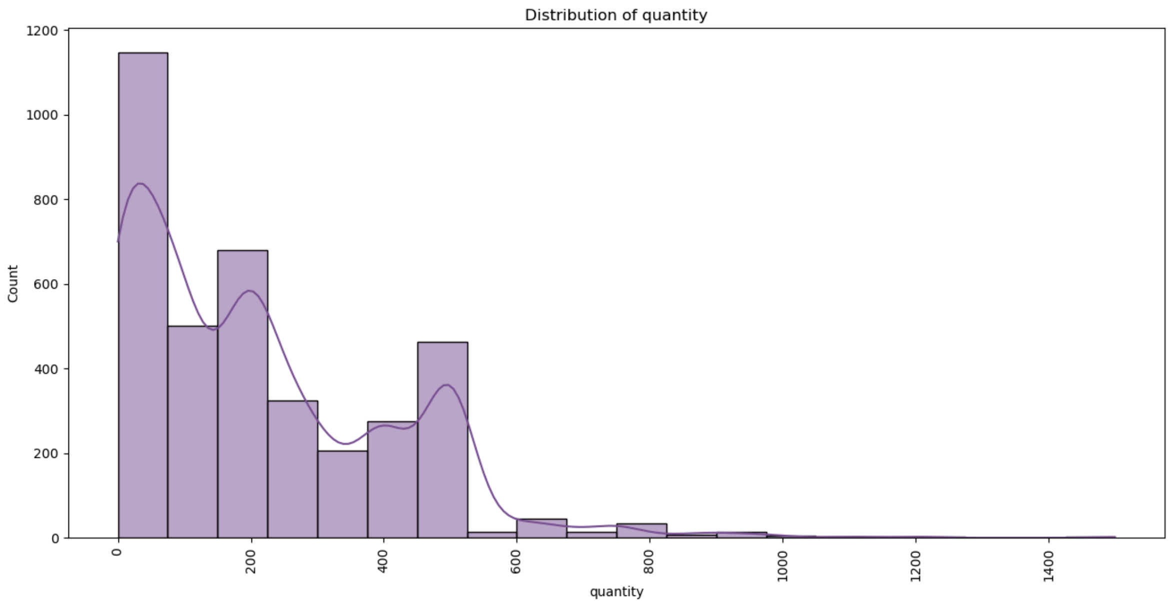



Distribution of discountPercent
Count
discountPercent


Distribution of availableQuantity
Count
availableQuantity


Distribution of discountedSellingPrice
Count
discountedSellingPrice




In [31]:
```
df.head()
```

Out[31]:

| | Category | name | mrp | discountPercent | availableQuantity | discountedSellingPrice |
|---|---|---|---|---|---|---|
| **0** | Fruits & Vegetables | Onion | 2500 | 16 | 3 | 2100 |
| **1** | Fruits & Vegetables | Tomato Hybrid | 4200 | 16 | 3 | 3500 |
| **2** | Fruits & Vegetables | Tender Coconut | 5100 | 15 | 3 | 4300 |
| **3** | Fruits & Vegetables | Coriander Leaves | 2000 | 15 | 3 | 1700 |
| **4** | Fruits & Vegetables | Ladies Finger | 1400 | 14 | 3 | 1200 |

In [33]:
```
Top_dicount_percentage=df.sort_values(["discountPercent"],ascending=False
Top_dicount_percentage[["Category","name","mrp","discountPercent","discou
```

Out[33]:

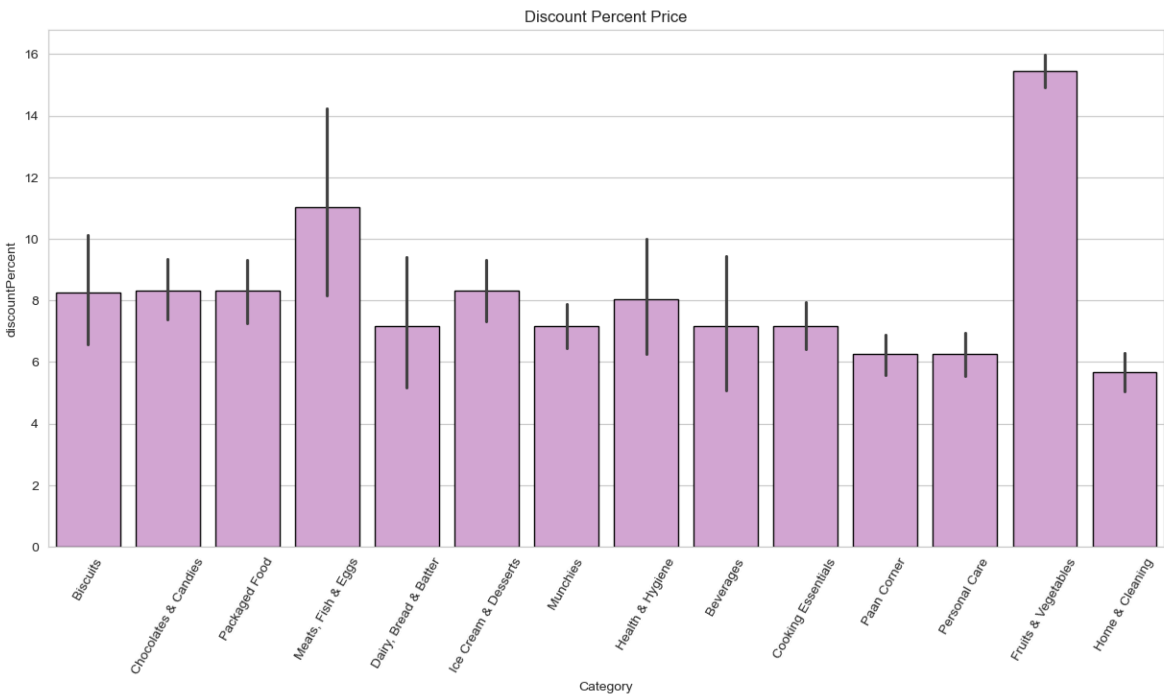
| | Category | name | mrp | discountPercent | discountedSellingPrice |
|---|---|---|---|---|---|
| **2608** | Biscuits | Dukes Waffy Chocolate Wafers | 4500 | 51 | 2200 |
| **2615** | Biscuits | Dukes Waffy Orange Wafers | 4500 | 51 | 2200 |
| **2619** | Biscuits | Dukes Waffy Strawberry Wafers | 4500 | 51 | 2200 |
| **2219** | Chocolates & Candies | Chef's Basket Durum Wheat Fusilli Pasta | 16000 | 50 | 8000 |
| **1431** | Packaged Food | Chef's Basket Durum Wheat Elbow Pasta | 16000 | 50 | 8000 |
| **...** | ... | ... | ... | ... | ... |
| **2723** | Biscuits | Britannia Marie Gold Biscuit | 2800 | 0 | 2800 |
| **2722** | Biscuits | Britannia Milk Bikis Milky Sandwich | 2500 | 0 | 2500 |
| **1349** | Beverages | Yoga Bar Peanut Butter Dark Chocolate Jar | 24900 | 0 | 24900 |
| **2719** | Biscuits | Britannia Milk Bikis Cream Biscuit | 4500 | 0 | 4500 |
| **3731** | Health & Hygiene | Dettol Antiseptic Liquid | 3000 | 0 | 3000 |

3732 rows × 5 columns

In [39]:

```
plt.figure(figsize=(15,7))
sns.set_style('whitegrid')
sns.barplot(data=Top_dicount_percentage,
            x='Category',
            y='discountPercent',
            color='plum',
            edgecolor='black')

plt.xticks(rotation=60)
plt.title('Discount Percent Price')
```

Out[39]: Text(0.5, 1.0, 'Discount Percent Price')




Qunatity selling

In [40]:
```
Top_quantity=df.sort_values(['quantity'])
Top_quantity[['Category','quantity','availableQuantity']]
```

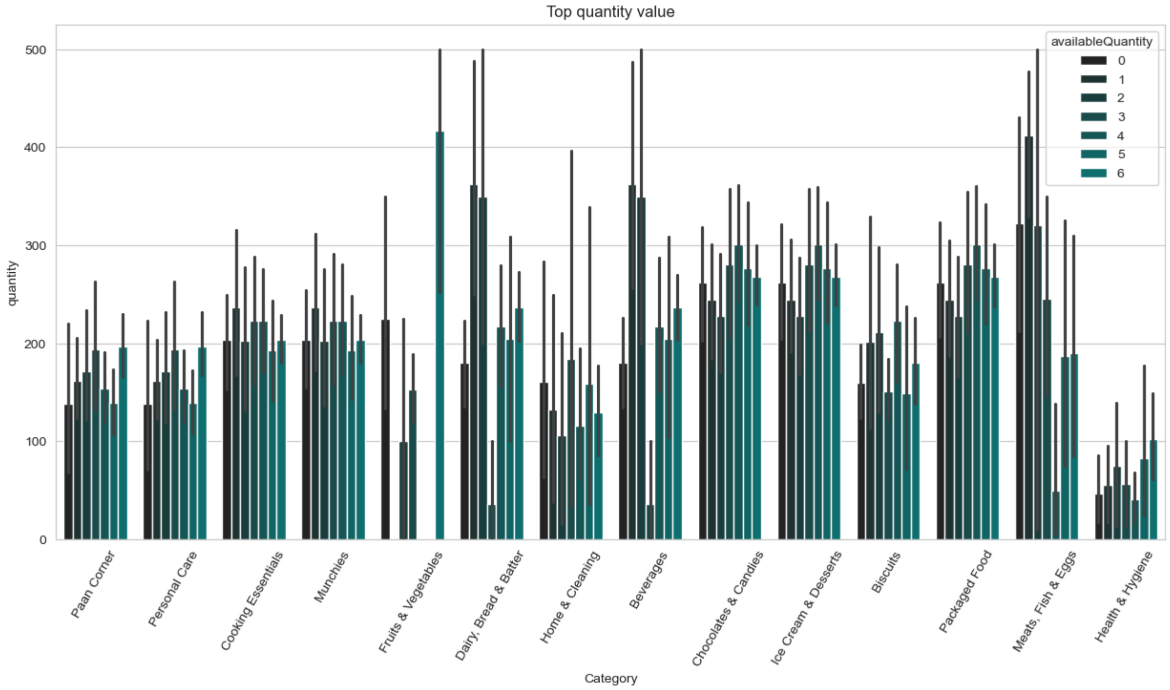
Out[40]:

| | Category | quantity | availableQuantity |
|---|---|---|---|
| **3273** | Paan Corner | 0 | 3 |
| **2929** | Personal Care | 0 | 3 |
| **606** | Cooking Essentials | 0 | 0 |
| **3184** | Paan Corner | 0 | 2 |
| **2840** | Personal Care | 0 | 2 |
| **...** | ... | ... | ... |
| **2320** | Chocolates & Candies | 1200 | 6 |
| **1544** | Packaged Food | 1200 | 6 |
| **1932** | Ice Cream & Desserts | 1200 | 6 |
| **3161** | Paan Corner | 1500 | 6 |
| **2817** | Personal Care | 1500 | 6 |

3732 rows × 3 columns

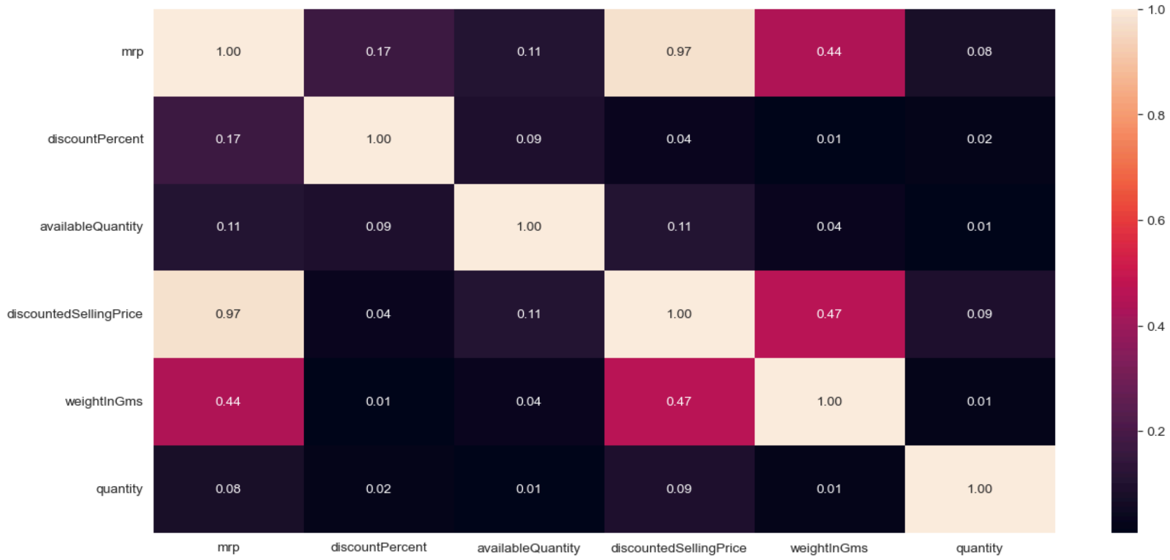
In [42]:
```
plt.figure(figsize=(15,7))
sns.set_style('whitegrid')
sns.barplot(data=Top_quantity,
            x='Category',
            y='quantity',
            hue='availableQuantity',
           color='teal')
plt.xticks(rotation=60)
plt.title('Top quantity value');
```

```
C:\Users\ashif\AppData\Local\Temp\ipykernel_13332\1466617710.py:3: FutureW
arning:

Setting a gradient palette using color= is deprecated and will be removed
in v0.14.0. Set `palette='dark:teal'` for the same effect.

  sns.barplot(data=Top_quantity,
```




Correlation Map

In [46]:
```python
plt.figure(figsize=(15,7))
corr=df.select_dtypes(include="number").corr()
sns.heatmap(data=corr,
            annot=True,
            fmt='.2f')
plt.show()
```




Applying in ANN model

Feature engineering

In [47]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3732 entries, 0 to 3731
Data columns (total 9 columns):
 #   Column                 Non-Null Count  Dtype 
---  ------                 --------------  ----- 
 0   Category               3732 non-null   object
 1   name                   3732 non-null   object
 2   mrp                    3732 non-null   int64 
 3   discountPercent        3732 non-null   int64 
 4   availableQuantity      3732 non-null   int64 
 5   discountedSellingPrice 3732 non-null   int64 
 6   weightInGms            3732 non-null   int64 
 7   outOfStock             3732 non-null   bool  
 8   quantity               3732 non-null   int64 
dtypes: bool(1), int64(6), object(2)
memory usage: 237.0+ KB
```

In [54]:
```
label=['Category','name','outOfStock']
from sklearn.preprocessing import LabelEncoder
```

In [55]:
```
le=LabelEncoder()
```

In [56]:
```
for feature in label:
    df[feature]=le.fit_transform(df[feature])
```

In [58]:
```
df.head()
```

Out[58]:

| | Category | name | mrp | discountPercent | availableQuantity | discountedSellingPrice | weig |
|---|---|---|---|---|---|---|---|
| **0** | 5 | 1046 | 2500 | 16 | 3 | 2100 | |
| **1** | 5 | 1551 | 4200 | 16 | 3 | 3500 | |
| **2** | 5 | 1500 | 5100 | 15 | 3 | 4300 | |
| **3** | 5 | 351 | 2000 | 15 | 3 | 1700 | |
| **4** | 5 | 835 | 1400 | 14 | 3 | 1200 | |

ANN Section ( Artificial intelligence network)

In [60]:
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,recall_score,precision_score,f
```

In [61]:
```
import tensorflow as tf
```

In [63]:
```
! pip install shap
```

```
Collecting shap
  Downloading shap-0.48.0-cp312-cp312-win_amd64.whl.metadata (25 kB)
Requirement already satisfied: numpy in c:\users\ashif\anaconda3\lib\site-packages (from shap) (1.26.4)
Requirement already satisfied: scipy in c:\users\ashif\anaconda3\lib\site-packages (from shap) (1.13.1)
Requirement already satisfied: scikit-learn in c:\users\ashif\anaconda3\lib\site-packages (from shap) (1.5.1)
Requirement already satisfied: pandas in c:\users\ashif\anaconda3\lib\site-packages (from shap) (2.2.3)
Requirement already satisfied: tqdm>=4.27.0 in c:\users\ashif\anaconda3\lib\site-packages (from shap) (4.66.5)
Requirement already satisfied: packaging>20.9 in c:\users\ashif\anaconda3\lib\site-packages (from shap) (24.1)
Collecting slicer==0.0.8 (from shap)
  Downloading slicer-0.0.8-py3-none-any.whl.metadata (4.0 kB)
Requirement already satisfied: numba>=0.54 in c:\users\ashif\anaconda3\lib\site-packages (from shap) (0.60.0)
Requirement already satisfied: cloudpickle in c:\users\ashif\anaconda3\lib\site-packages (from shap) (3.0.0)
Requirement already satisfied: typing-extensions in c:\users\ashif\anaconda3\lib\site-packages (from shap) (4.11.0)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in c:\users\ashif\anaconda3\lib\site-packages (from numba>=0.54->shap) (0.43.0)
Requirement already satisfied: colorama in c:\users\ashif\anaconda3\lib\site-packages (from tqdm>=4.27.0->shap) (0.4.6)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\ashif\anaconda3\lib\site-packages (from pandas->shap) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\ashif\anaconda3\lib\site-packages (from pandas->shap) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\ashif\anaconda3\lib\site-packages (from pandas->shap) (2023.3)
Requirement already satisfied: joblib>=1.2.0 in c:\users\ashif\anaconda3\lib\site-packages (from scikit-learn->shap) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\ashif\anaconda3\lib\site-packages (from scikit-learn->shap) (3.5.0)
Requirement already satisfied: six>=1.5 in c:\users\ashif\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas->shap) (1.16.0)
Downloading shap-0.48.0-cp312-cp312-win_amd64.whl (545 kB)
   ---------------------------------------- 0.0/545.3 kB ? eta -:--:--
   ---------------------------------------- 545.3/545.3 kB 6.0 MB/s eta 0:00:00
Downloading slicer-0.0.8-py3-none-any.whl (15 kB)
Installing collected packages: slicer, shap
Successfully installed shap-0.48.0 slicer-0.0.8
```

In [64]:

```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Input
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.optimizers import Adam
import shap
```

In [70]:

```python
X = df.drop(df.columns[-2],axis=1)
y = df.iloc[:,-2]
```

In [75]:

```python
X_train,X_test,y_train,y_test=train_test_split(X,
                                               y,
                                               test_size=0.2,
                                               random_state=42)
```

In [82]:
```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [83]:
```
model=Sequential([
    Input(shape=(X_train_scaled.shape[1],)),
    Dense(128,activation='relu'),
    Dense(64,activation='relu'),
    Dense(1)
])
```

In [86]:
```
model.compile(optimizer=Adam(),
              loss='mean_squared_error',
              metrics=['mse'])
model.fit(X_train_scaled,
          y_train,epochs=100,
          batch_size=32,
          validation_split=0.1)
```

```
Epoch 1/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0046 - mse: 0.0046 - val_loss: 0.0030 - val_mse: 0.0030
Epoch 2/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0021 - mse: 0.0021 - val_loss: 0.0021 - val_mse: 0.0021
Epoch 3/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0017 - mse: 0.0017 - val_loss: 0.0015 - val_mse: 0.0015
Epoch 4/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0018 - mse: 0.0018 - val_loss: 0.0021 - val_mse: 0.0021
Epoch 5/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0021 - mse: 0.0021 - val_loss: 0.0012 - val_mse: 0.0012
Epoch 6/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0010 - mse: 0.0010 - val_loss: 8.3732e-04 - val_mse: 8.3732e-04
Epoch 7/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 7.3045e-04 - mse: 7.3045e-04 - val_loss: 9.2200e-04 - val_mse: 9.2200e-04
Epoch 8/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 9.1120e-04 - mse: 9.1120e-04 - val_loss: 0.0010 - val_mse: 0.0010
Epoch 9/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.9457e-04 - mse: 5.9457e-04 - val_loss: 0.0010 - val_mse: 0.0010
Epoch 10/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 6.3691e-04 - mse: 6.3691e-04 - val_loss: 7.6235e-04 - val_mse: 7.6235e-04
Epoch 11/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.9928e-04 - mse: 4.9928e-04 - val_loss: 6.3912e-04 - val_mse: 6.3912e-04
Epoch 12/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.9446e-04 - mse: 3.9446e-04 - val_loss: 5.4683e-04 - val_mse: 5.4683e-04
Epoch 13/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 4.9065e-04 - mse: 4.9065e-04 - val_loss: 7.1757e-04 - val_mse: 7.1757e-04
Epoch 14/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.3326e-04 - mse: 5.3326e-04 - val_loss: 5.2001e-04 - val_mse: 5.2001e-04
Epoch 15/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.8861e-04 - mse: 2.8861e-04 - val_loss: 7.3625e-04 - val_mse: 7.3625e-04
Epoch 16/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0018 - mse: 0.0018 - val_loss: 0.0015 - val_mse: 0.0015
Epoch 17/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 8.8839e-04 - mse: 8.8839e-04 - val_loss: 4.6048e-04 - val_mse: 4.6048e-04
Epoch 18/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.9846e-04 - mse: 4.9846e-04 - val_loss: 8.3501e-04 - val_mse: 8.3501e-04
Epoch 19/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.7885e-04 - mse: 4.7885e-04 - val_loss: 6.3975e-04 - val_mse: 6.3975e-04
Epoch 20/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.3272e-04 - mse: 3.3272e-04 - val_loss: 3.8675e-04 - val_mse: 3.8675e-04
```

```
Epoch 21/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.9285e-04 - mse: 2.928
5e-04 - val_loss: 3.3543e-04 - val_mse: 3.3543e-04
Epoch 22/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.5958e-04 - mse: 2.595
8e-04 - val_loss: 5.6141e-04 - val_mse: 5.6141e-04
Epoch 23/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.2862e-04 - mse: 4.286
2e-04 - val_loss: 5.0637e-04 - val_mse: 5.0637e-04
Epoch 24/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 2.8760e-04 - mse: 2.876
0e-04 - val_loss: 0.0016 - val_mse: 0.0016
Epoch 25/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 8.8997e-04 - mse: 8.899
7e-04 - val_loss: 5.4797e-04 - val_mse: 5.4797e-04
Epoch 26/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.5122e-04 - mse: 5.512
2e-04 - val_loss: 4.2937e-04 - val_mse: 4.2937e-04
Epoch 27/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.7826e-04 - mse: 3.782
6e-04 - val_loss: 8.7348e-04 - val_mse: 8.7348e-04
Epoch 28/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.6969e-04 - mse: 4.696
9e-04 - val_loss: 8.8197e-04 - val_mse: 8.8197e-04
Epoch 29/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 7.3658e-04 - mse: 7.365
8e-04 - val_loss: 5.7461e-04 - val_mse: 5.7461e-04
Epoch 30/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.7704e-04 - mse: 5.770
4e-04 - val_loss: 6.5893e-04 - val_mse: 6.5893e-04
Epoch 31/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.9467e-04 - mse: 4.946
7e-04 - val_loss: 5.4343e-04 - val_mse: 5.4343e-04
Epoch 32/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.2725e-04 - mse: 3.272
5e-04 - val_loss: 3.8524e-04 - val_mse: 3.8524e-04
Epoch 33/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.8227e-04 - mse: 1.822
7e-04 - val_loss: 3.0552e-04 - val_mse: 3.0552e-04
Epoch 34/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.0664e-04 - mse: 2.066
4e-04 - val_loss: 2.7737e-04 - val_mse: 2.7737e-04
Epoch 35/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.8089e-04 - mse: 1.808
9e-04 - val_loss: 1.6250e-04 - val_mse: 1.6250e-04
Epoch 36/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.0679e-04 - mse: 1.067
9e-04 - val_loss: 3.6486e-04 - val_mse: 3.6486e-04
Epoch 37/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.5951e-04 - mse: 1.595
1e-04 - val_loss: 9.6381e-04 - val_mse: 9.6381e-04
Epoch 38/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 6.5370e-04 - mse: 6.537
0e-04 - val_loss: 3.0382e-04 - val_mse: 3.0382e-04
Epoch 39/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 7.5670e-04 - mse: 7.567
0e-04 - val_loss: 2.5439e-04 - val_mse: 2.5439e-04
Epoch 40/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 1.2392e-04 - mse: 1.239
2e-04 - val_loss: 3.7029e-04 - val_mse: 3.7029e-04
```

```
Epoch 41/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.7307e-04 - mse: 1.730
7e-04 - val_loss: 3.4336e-04 - val_mse: 3.4336e-04
Epoch 42/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.2646e-04 - mse: 3.264
6e-04 - val_loss: 2.2832e-04 - val_mse: 2.2832e-04
Epoch 43/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.0150e-04 - mse: 2.015
0e-04 - val_loss: 3.2426e-04 - val_mse: 3.2426e-04
Epoch 44/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.7754e-04 - mse: 2.775
4e-04 - val_loss: 1.6434e-04 - val_mse: 1.6434e-04
Epoch 45/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.9279e-04 - mse: 1.927
9e-04 - val_loss: 6.4458e-04 - val_mse: 6.4458e-04
Epoch 46/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.5137e-04 - mse: 4.513
7e-04 - val_loss: 6.6478e-04 - val_mse: 6.6478e-04
Epoch 47/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.9063e-04 - mse: 2.906
3e-04 - val_loss: 6.4317e-04 - val_mse: 6.4317e-04
Epoch 48/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 5.4536e-04 - mse: 5.453
6e-04 - val_loss: 9.8531e-04 - val_mse: 9.8531e-04
Epoch 49/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0011 - mse: 0.0011 -
val_loss: 2.5147e-04 - val_mse: 2.5147e-04
Epoch 50/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.8629e-04 - mse: 1.862
9e-04 - val_loss: 7.5062e-04 - val_mse: 7.5062e-04
Epoch 51/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.5871e-04 - mse: 3.587
1e-04 - val_loss: 1.3777e-04 - val_mse: 1.3777e-04
Epoch 52/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.7069e-04 - mse: 1.706
9e-04 - val_loss: 2.0620e-04 - val_mse: 2.0620e-04
Epoch 53/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.1897e-04 - mse: 1.189
7e-04 - val_loss: 1.8816e-04 - val_mse: 1.8816e-04
Epoch 54/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.8092e-04 - mse: 1.809
2e-04 - val_loss: 2.2791e-04 - val_mse: 2.2791e-04
Epoch 55/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.9251e-04 - mse: 2.925
1e-04 - val_loss: 2.3388e-04 - val_mse: 2.3388e-04
Epoch 56/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.2887e-04 - mse: 2.288
7e-04 - val_loss: 7.5892e-04 - val_mse: 7.5892e-04
Epoch 57/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 4.8976e-04 - mse: 4.897
6e-04 - val_loss: 1.8976e-04 - val_mse: 1.8976e-04
Epoch 58/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.1326e-04 - mse: 1.132
6e-04 - val_loss: 1.9471e-04 - val_mse: 1.9471e-04
Epoch 59/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.1549e-04 - mse: 1.154
9e-04 - val_loss: 3.2531e-04 - val_mse: 3.2531e-04
Epoch 60/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 9.4978e-05 - mse: 9.497
8e-05 - val_loss: 2.7650e-04 - val_mse: 2.7650e-04
```

```
Epoch 61/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.4203e-04 - mse: 2.420
3e-04 - val_loss: 2.2948e-04 - val_mse: 2.2948e-04
Epoch 62/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.4826e-04 - mse: 1.482
6e-04 - val_loss: 4.8371e-04 - val_mse: 4.8371e-04
Epoch 63/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.7365e-04 - mse: 2.736
5e-04 - val_loss: 1.4029e-04 - val_mse: 1.4029e-04
Epoch 64/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 7.7230e-05 - mse: 7.723
0e-05 - val_loss: 1.8268e-04 - val_mse: 1.8268e-04
Epoch 65/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 6.2853e-05 - mse: 6.285
3e-05 - val_loss: 2.0831e-04 - val_mse: 2.0831e-04
Epoch 66/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.3778e-04 - mse: 2.377
8e-04 - val_loss: 1.5975e-04 - val_mse: 1.5975e-04
Epoch 67/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.1240e-04 - mse: 1.124
0e-04 - val_loss: 1.3893e-04 - val_mse: 1.3893e-04
Epoch 68/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 8.3955e-05 - mse: 8.395
5e-05 - val_loss: 2.5568e-04 - val_mse: 2.5568e-04
Epoch 69/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.0388e-04 - mse: 1.038
8e-04 - val_loss: 3.3471e-04 - val_mse: 3.3471e-04
Epoch 70/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 3.5332e-04 - mse: 3.533
2e-04 - val_loss: 2.6968e-04 - val_mse: 2.6968e-04
Epoch 71/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.6177e-04 - mse: 2.617
7e-04 - val_loss: 1.6623e-04 - val_mse: 1.6623e-04
Epoch 72/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.1748e-04 - mse: 1.174
8e-04 - val_loss: 2.1731e-04 - val_mse: 2.1731e-04
Epoch 73/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 1.8857e-04 - mse: 1.885
7e-04 - val_loss: 6.7454e-04 - val_mse: 6.7454e-04
Epoch 74/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 2.6078e-04 - mse: 2.607
8e-04 - val_loss: 2.5877e-04 - val_mse: 2.5877e-04
Epoch 75/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 4.2363e-04 - mse: 4.236
3e-04 - val_loss: 3.9799e-04 - val_mse: 3.9799e-04
Epoch 76/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 4.5179e-04 - mse: 4.517
9e-04 - val_loss: 5.5317e-04 - val_mse: 5.5317e-04
Epoch 77/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 3.3746e-04 - mse: 3.374
6e-04 - val_loss: 4.2767e-04 - val_mse: 4.2767e-04
Epoch 78/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 1.7937e-04 - mse: 1.793
7e-04 - val_loss: 2.4753e-04 - val_mse: 2.4753e-04
Epoch 79/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 2.6018e-04 - mse: 2.601
8e-04 - val_loss: 1.8837e-04 - val_mse: 1.8837e-04
Epoch 80/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 9.4656e-05 - mse: 9.465
6e-05 - val_loss: 6.4708e-05 - val_mse: 6.4708e-05
```

```
Epoch 81/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 6.5077e-05 - mse: 6.507
7e-05 - val_loss: 4.8466e-05 - val_mse: 4.8466e-05
Epoch 82/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 4.5573e-05 - mse: 4.557
3e-05 - val_loss: 1.2191e-04 - val_mse: 1.2191e-04
Epoch 83/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 6.7483e-05 - mse: 6.748
3e-05 - val_loss: 4.5132e-05 - val_mse: 4.5132e-05
Epoch 84/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 4.3571e-05 - mse: 4.357
1e-05 - val_loss: 9.3072e-05 - val_mse: 9.3072e-05
Epoch 85/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 6.6859e-05 - mse: 6.685
9e-05 - val_loss: 1.4494e-04 - val_mse: 1.4494e-04
Epoch 86/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 9.7507e-05 - mse: 9.750
7e-05 - val_loss: 1.3516e-04 - val_mse: 1.3516e-04
Epoch 87/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.3500e-04 - mse: 1.350
0e-04 - val_loss: 5.9554e-05 - val_mse: 5.9554e-05
Epoch 88/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 8.1853e-05 - mse: 8.185
3e-05 - val_loss: 1.1984e-04 - val_mse: 1.1984e-04
Epoch 89/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 9.4043e-05 - mse: 9.404
3e-05 - val_loss: 1.3820e-04 - val_mse: 1.3820e-04
Epoch 90/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 1.2542e-04 - mse: 1.254
2e-04 - val_loss: 1.9397e-04 - val_mse: 1.9397e-04
Epoch 91/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 1.9652e-04 - mse: 1.965
2e-04 - val_loss: 1.8354e-04 - val_mse: 1.8354e-04
Epoch 92/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 8.2534e-05 - mse: 8.253
4e-05 - val_loss: 2.3352e-04 - val_mse: 2.3352e-04
Epoch 93/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.2416e-04 - mse: 2.241
6e-04 - val_loss: 0.0019 - val_mse: 0.0019
Epoch 94/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.0013 - mse: 0.0013 -
val_loss: 7.2335e-04 - val_mse: 7.2335e-04
Epoch 95/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 7.0522e-04 - mse: 7.052
2e-04 - val_loss: 4.7542e-04 - val_mse: 4.7542e-04
Epoch 96/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.7387e-04 - mse: 2.738
7e-04 - val_loss: 2.1938e-04 - val_mse: 2.1938e-04
Epoch 97/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.1216e-04 - mse: 1.121
6e-04 - val_loss: 3.8655e-04 - val_mse: 3.8655e-04
Epoch 98/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 2.0430e-04 - mse: 2.043
0e-04 - val_loss: 1.5194e-04 - val_mse: 1.5194e-04
Epoch 99/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 9.5175e-05 - mse: 9.517
5e-05 - val_loss: 1.6619e-04 - val_mse: 1.6619e-04
Epoch 100/100
84/84 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 1.0510e-04 - mse: 1.051
0e-04 - val_loss: 8.1086e-05 - val_mse: 8.1086e-05
```

Out[86]: <keras.src.callbacks.history.History at 0x1db75a78bf0>

In [87]:
```
y_pred = model.predict(X_test_scaled)
y_pred_labels = (y_pred > 0.5).astype(int).flatten()
```

```
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
```

In [88]:
```
print(classification_report(y_test,y_pred_labels))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       648
           1       1.00      1.00      1.00        99

    accuracy                           1.00       747
   macro avg       1.00      1.00      1.00       747
weighted avg       1.00      1.00      1.00       747
```

In [90]:
```
# Convert probabilities to class labels
y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)

# Convert probabilities to class labels
y_train_pred_labels = (y_train_pred > 0.5).astype(int)
y_test_pred_labels = (y_test_pred > 0.5).astype(int)

# Evaluate
from sklearn.metrics import classification_report

print("Train Report:")
print(classification_report(y_train, y_train_pred_labels))

print("Test Report:")
print(classification_report(y_test, y_test_pred_labels))
```

```
94/94 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step
Train Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      2631
           1       0.12      1.00      0.21       354

    accuracy                           0.12      2985
   macro avg       0.06      0.50      0.11      2985
weighted avg       0.01      0.12      0.03      2985

Test Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       648
           1       0.13      1.00      0.23        99

    accuracy                           0.13       747
   macro avg       0.07      0.50      0.12       747
weighted avg       0.02      0.13      0.03       747
```

```
C:\Users\ashif\anaconda3\Lib\site-packages\sklearn\metrics\_classificatio
n.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set
to 0.0 in labels with no predicted samples. Use `zero_division` parameter
to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\ashif\anaconda3\Lib\site-packages\sklearn\metrics\_classificatio
n.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set
to 0.0 in labels with no predicted samples. Use `zero_division` parameter
to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\ashif\anaconda3\Lib\site-packages\sklearn\metrics\_classificatio
n.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set
to 0.0 in labels with no predicted samples. Use `zero_division` parameter
to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\ashif\anaconda3\Lib\site-packages\sklearn\metrics\_classificatio
n.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set
to 0.0 in labels with no predicted samples. Use `zero_division` parameter
to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\ashif\anaconda3\Lib\site-packages\sklearn\metrics\_classificatio
n.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set
to 0.0 in labels with no predicted samples. Use `zero_division` parameter
to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
C:\Users\ashif\anaconda3\Lib\site-packages\sklearn\metrics\_classificatio
n.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set
to 0.0 in labels with no predicted samples. Use `zero_division` parameter
to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

Conclusion:

- Current ANN is severely underperforming and overfitting to the minority class.
- You should address class imbalance and retrain with class_weight, SMOTE, or both.
- After fixing, re-run the classification report to see improvements in both recall and precision across both classes.

In [ ]: