# **# Objectives:**

* initializing the product\_module.
* making different modules suitable as per needs and registering to the database.

# **# Introduction:**

A model is the single, definitive source of information about data. It contains the essential fields and behaviors of the data that is being stored. Generally, each model maps to a single database table.

The basics:

* Each model is a Python class that subclasses django.db.models.Model.
* Each attribute of the model represents a database field.
* With all of this, Django gives an automatically generated database access API

# **# Procedure:**

1. in 'models.py' create a model for brands

code:

class Brand(models.Model):

name = models.CharField(max\_length=200)

is\_active = models.BooleanField()

1. add the brand table to the database by

python manage.py makemigrations

python manage.py migrate

1. in the 'admin.py' which adds the content of the admin panel adds the following code :

from .models import Brand

admin.site.register(Brand)

1. run the server and verify the table by performing the CRUDE operation.

python manage.py runserver

1. in the 'models.py' edit the code for the brand model with the following code:

class Brand(models.Model):

name = models.CharField(max\_length=200)

is\_active = models.BooleanField()

1. in the same edit the code for the category model

**code:**

class Category(models.Model):

name = models.CharField(max\_length=200)

is\_active = models.BooleanField()

class Meta:

verbose\_name\_plural = "Categories"

1. add the necessary fields to the product model

**code:**

class Product(models.Model):

name = models.CharField(max\_length=200)

price = models.FloatField()

quantity = models.IntegerField()

image\_url = models.CharField(max\_length=500)

color\_code = models.CharField(max\_length=20)

brand = models.ForeignKey(Brand, on\_delete=models.CASCADE)

category = models.ForeignKey(Category, on\_delete=models.CASCADE)

registered\_on = models.DateTimeField()

is\_active = models.BooleanField()

1. save the changes to the database.
2. to enable the category and product models in the admin interface, add the following code in 'admin.py'

from .models import Brand, Category, Product

admin.site.register(Brand)

admin.site.register(Category)

admin.site.register(Product)

1. run the project server and verify the CRUD operations for the brand, category, and product respectively

python manage.py runserver

# **# Conclusion:**

Here in this lab session, we got to know about how to create a model, edit the model in an appropriate manner, and enter and evaluate the entered data in the Django server database.