

## Author

Utkarsh Singh

21f2001496

21f2001496@ds.study.iitm.ac.in

I am a 21-year-old student, who hails from Kashipur, Uttarakhand. This is my 1st term in the Diploma level. I am also pursuing B.Arch. from Indian Institute of technology, (BHU) Varanasi.

## Description

For MAD-1 project we needed to build an e-commerce website.

It should have login and register portal for user, admin where user can add, modify and delete products belonging to a category along with adding a new one and a user can search and buy the products, belonging to same or different categories.

## Technologies used

- Flask - Flask, flash, render template, url\_for, redirect, request, session, current\_app,g
- Flask-Login - UserMixin, LoginManager,
- Flask-SQLAlchemy -SQLAlchemy
- flask\_wtf- FlaskForm
- wtforms -StringField, PasswordField, SubmitField
- flask\_msearch- Search
- flask\_bcrypt-Bcrypt

## DB Schema Design

```
class User(db.Model, UserMixin):
```

```
    id = db.Column(db.Integer, primary_key=True)
```

```
    username = db.Column(db.String(20), nullable=False, unique=True)
```

```
    password = db.Column(db.String(80), nullable=False)
```

```
class Merchant(db.Model, UserMixin):
```

```
    id = db.Column(db.Integer, primary_key=True)
```

```
    username = db.Column(db.String(20), nullable=False, unique=True)
```

```
    password = db.Column(db.String(80), nullable=False)
```

```
class Cart(db.Model, UserMixin):
```

```
    id = db.Column(db.Integer, primary_key=True)
```

```
    pname = db.Column(db.String(20), nullable=False)
```

```
    price = db.Column(db.Integer, nullable=False)
```

```
    quantity=db.Column(db.Integer,nullable=False)
```

```
    product_id=db.Column(db.Integer, db.ForeignKey('products.id'))
```

```

product=db.relationship('Products',backref=db.backref('products',lazy=True))
user_id=db.Column(db.Integer, db.ForeignKey('user.id'))
user=db.relationship('User',backref=db.backref('users',lazy=True))
class Products(db.Model, UserMixin):
    __searchtable__=['name','unit']
    id = db.Column(db.Integer, primary_key=True)
    name=db.Column(db.String(20), nullable=False, unique=True)
    unit=db.Column(db.String(20), nullable=False)
    rate=db.Column(db.Integer,nullable=False)
    quantity=db.Column(db.Integer,nullable=False)
    category_id=db.Column(db.Integer, db.ForeignKey('category.id'),nullable=False)
    category=db.relationship('Category',backref=db.backref('categories',lazy=True))
    admin_id=db.Column(db.Integer, db.ForeignKey('merchant.id'),nullable=False)
    merchant=db.relationship('Merchant',backref=db.backref('merchants',lazy=True))

class Category(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key=True)
    name=db.Column(db.String(20), nullable=False, unique=True)
    admin_id=db.Column(db.Integer, db.ForeignKey('merchant.id'),nullable=False)
    merchant=db.relationship('Merchant',backref=db.backref('admin',lazy=True))

```

## API Design

API is created for:

User\_login,User\_logout,Admin\_login,Admin\_logout,User\_dashboard,Admin\_dashboard,Display Cart, Checkout, Delete from cart, Adding product to cart, Get category, Update product, Delete product, Delete category, Edit category.

## Architecture and Features

- Inside the zip folder, there is project directory which include 3 folders (instance, static and templates) , one python file name app.py and one project file report.
- The instance folder contains the database file named as database.sqlite3, the static folder contains static.css and templates folder contains all the html files.
- App.py is the main controller which loads the html files from templates and modifies it using database from database.sqlite3 in the instance folder. Style.css in the static folder passes the necessary .css commands to app.py which then passes it to html files.
- Database.sqlite3 ultimately stores all the login and register inputs along with all the other necessary data.

## Video

[https://drive.google.com/file/d/188O\\_XxMx9D49mEldv1zX6wrqazhz4HUx/view?usp=drive\\_link](https://drive.google.com/file/d/188O_XxMx9D49mEldv1zX6wrqazhz4HUx/view?usp=drive_link)