

# E-Commerce Application on IBM cloud Foundry

## CAD\_Phase3

### INSTRUCTIONS

```
pip install virtualenv
pip install flask flask_sqlalchemy
```

```
virtualenv venv
```

```
////////////////////////////////
```

```
create a folder "templates"
```

```
copy & paste index.html
```

```
create a folder static
```

```
copy & paste styles.css
```

```
in the static folder
```

```
create a folder structure by following
```

```
inside of static "images/product_images/"place the img files here""
```

```
////////////////////////////////
```

```
.\venv\Scripts\activate
```

```
pip install Flask Flask-SQLAlchemy
```

```
python app.py
```

```
////////////////////////////////
```

```
in another cmd
```

```
.\venv\Scripts\activate
```

```
python init_db.py
```

```
////////////////////////////////////
```

```
in another cmd
```

```
.\venv\Scripts\activate
```

```
python add_products_data.py
```

```
////////////////////////////////////
```

FOLLOWING FILES ARE CREATED WITH THE FOLLOWING FILE NAMES

## **File Name: app.py**

```
from flask import Flask, render_template
```

```
from flask_sqlalchemy import SQLAlchemy
```

```
app = Flask(__name__)
```

```
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///artisanal_ecommerce.db'
```

```
db = SQLAlchemy(app)
```

```
class Product(db.Model):
```

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

```
    name = db.Column(db.String(150), nullable=False)
```

```
    description = db.Column(db.String(500), nullable=False)
```

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

```
    image = db.Column(db.String(300))
```

```
# @app.before_first_request
```

```
def setup_db():
```

```
    db.create_all()
```

```
@app.route('/')
```

```
def index():
```

```
    products = Product.query.all()
```

```
        return render_template('index.html', products=products)

if __name__ == '__main__':
    app.run(debug=True)
```

## File Name: index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Artisanal E-commerce</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">
</head>
<body>
    <h1>Welcome to Artisanal E-commerce</h1>
    <ul>
        {% for product in products %}
            <li>
                <h3>{{ product.name }}</h3>
                <p>{{ product.description }}</p>
                <p>Price: ${{ product.price }}</p>
                
            </li>
        {% endfor %}
    </ul>
</body>
</html>
```

## File Name: add\_products\_data.py

```
from app import db, app, Product

if __name__ == "__main__":
    with app.app_context():
        products_data = [
            Product(name="Handcrafted Wooden Vase", description="A beautifully carved vase made from oak.", price=25.50, image="images/img1.jpeg"),
            Product(name="Artistic Ceramic Plate", description="A decorative plate with intricate patterns.", price=15.00, image="images/img3.jpeg"),
            Product(name="Woven Basket", description="A sturdy basket made from high-quality reeds.", price=12.75, image="images/img4.jpeg")
        ]

        for product in products_data:
            db.session.add(product)

        db.session.commit()
        print("Dummy products added!")
```

## File Name: styles.css

```
body {
    font-family: Arial, sans-serif;
    margin: 40px;
}
```

```
h1 {
    color: #2c3e50;
}

li {
    border: 1px solid #ecf0f1;
    padding: 20px;
    margin-bottom: 10px;
    list-style-type: none;
}

img {
    max-width: 100%;
    height: auto;
}
```

## FOLDER STRUCTURE

### CREATE FOLDER&FILES AS PER THIS ORDER

```
ecommerce/
|
├── static/
|   ├── css/
|   |   └── (your CSS files here if any)
|   |
|   ├── images/
|   |   ├── product_images/
|   |   |   ├── img1.jpg
|   |   |   ├── product1.jpg
|   |   |   ├── product2.jpg
|   |   |   └── product3.jpg
|   |
|   └──
|
├── templates/
|   └── index.html
|
```

```
└─ venv/ (Your virtual environment)
|
└─ app.py (Main Flask app)
└─ init_db.py (Database initialization script)
└─ add_products_data.py (Script to add dummy products to the database)
└─ site.db (SQLite database file)
```