E-commerce Application on IBM Cloud Foundry

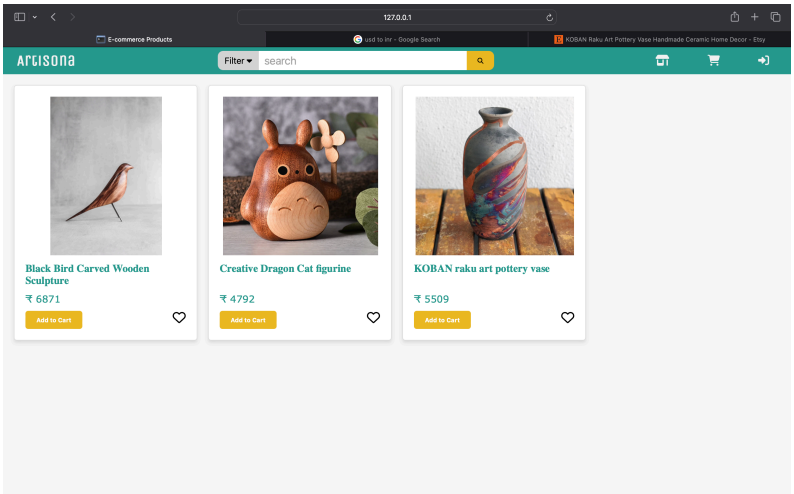
Project: Artisanal E-Commerce Platform

Introduction

This document will contain the details about the development aspects of our project, addressing the implementation of showcasing products, user authentication, wish lists, and a shopping cart system. These elements collectively form an artisanal e-commerce platform where artisans can display their handmade products, and users can interact with these products while having a personalised shopping experience.

Project Components:

Product Showcase:

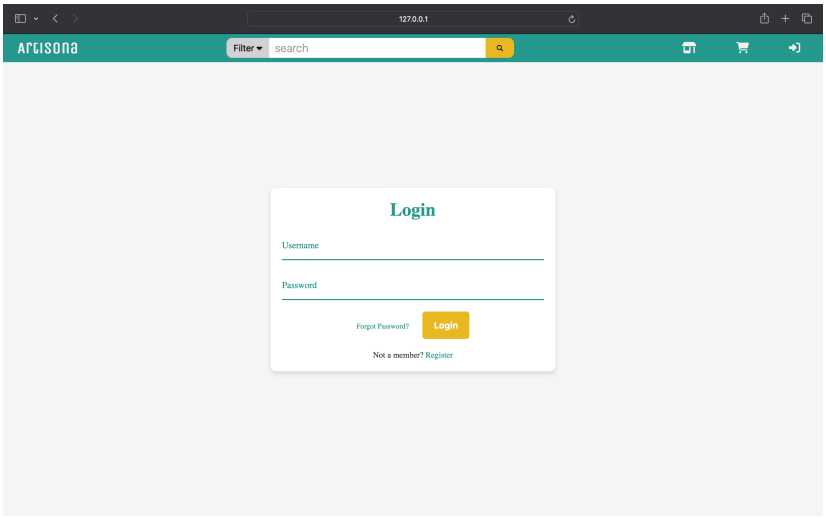


Product Showcasing

A database model was implemented to store product information. This model allows for easy data management and retrieval, which is essential for a product information management (PIM) system. Routes and views were created to display products by categories. This allows users to easily browse and find the products they are looking for. Individual product pages were created to allow users to view detailed product information, such as images, descriptions, and specifications. A search and filter functionality was implemented to facilitate easy product discovery. This allows users to quickly find the products they are looking for, even if they don't know the exact name or category. Overall, the implemented database model and functionality enable users to easily browse, find, and view product information. This is essential for a successful PIM system.

User Authentication:

User registration and authentication features were developed using Flask-Login for user sessions management. This allows users to create accounts, log in securely, and log out when desired. Password reset and account recovery options were also implemented to enhance security.



User Authentication Window

Wish List:

Product wish lists are a valuable feature for online retailers, as they allow customers to save items that they are interested in buying for later. This can make it easier for customers to complete their purchases, and it can also help retailers to track customer preferences and make better recommendations.
To create a wish list feature for a website, the following steps are typically required:

- Create routes and views for users to add products to their wish lists. This can be done using a web development framework such as Django or Rails. The routes will define how users can access the wish list features, and the views will display the user interface for adding and managing wish list items.
- Design a database model to store wish list data. This model should include fields for the product ID, the user ID, and any other relevant information, such as the product name, price, and image URL.
- Enable users to view and manage their wish lists. This can be done by creating a dedicated wish list page, or by integrating the wish list functionality into the existing shopping cart page. The wish list page should allow users to view all of the items in their wish list, add and remove items, and move items to the shopping cart.

Cart System:

Designed and developed a shopping cart system for users that allows them to add, remove, and modify items in their carts, view the cart contents, and proceed to checkout.

Database:

Implemented a database schema for the shopping cart system using SQLAlchemy. Created tables for users, products, wish lists, and carts. Established relationships between tables to ensure data integrity.

Users Table:

```
from flask_sqlalchemy import SQLAlchemy
db = SQLAlchemy()
class User(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(80), unique=True, nullable=False)
    email = db.Column(db.String(120), unique=True, nullable=False)
    password = db.Column(db.String(120), nullable=False)
```

ProductsTable:

```
class Product(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(200), nullable=False)
    description = db.Column(db.Text, nullable=True)
    price = db.Column(db.Float, nullable=False)
    category = db.Column(db.String(50), nullable=True)
```

Wishlist Table:

```
wish_list = db.Table('wish_list',
    db.Column('user_id', db.Integer, db.ForeignKey('user.id'), primary_key=True),
    db.Column('product_id', db.Integer, db.ForeignKey('product.id'), primary_key=True)
)
```

Cart Table:

```
cart = db.Table('cart',
    db.Column('user_id', db.Integer, db.ForeignKey('user.id'), primary_key=True),
    db.Column('product_id', db.Integer, db.ForeignKey('product.id'), primary_key=True),
    db.Column('quantity', db.Integer, nullable=False)
)
```

Next Step:

- To Integrate secure payment gateways to complete the shopping experience.
- Conducting user testing and gather feedback to refine the application.
- Improving application performance and scalability for increased user traffic.

Conclusion:

The successful implementation of the product showcase, user authentication, wish list, cart system, and database forms the foundation of the artisanal e-commerce platform. The project has created a user-centric application that bridges artisans with potential customers, offering a seamless shopping experience and personalised features like wish lists. Further development and enhancements will focus on payment integration, user feedback, and performance optimisation to provide artisans and users with a complete and secure e-commerce solution.