**NANMUDHALVAN**

**-** **CAD101 Cloud Application Development - Group 1**

**COLLEGE NAME: JEPPIAAR INSTITUTE OF TECHNOLOGY**

**TITLE: 2106-E-commerece Application on IBM Cloud Foundry**

**TEAM NAME: Proj\_228508\_Team\_2**

**TEAM MEMBERS:**

1. **S.Jagadesh**
2. **S.Ihsan Ahamed**
3. **R.Aravindhan**
4. **S.Guhan**
5. **P.Manoj**

**PHASE - 3: Development Part 1**

In this part you will begin building your project. Begin building the artisanal E-commerce platform on IBM Cloud Foundry.

Design the platform layout and create a database to store product information.

**OBJECTIVIES:**

Before creating the database, design the layout of your e-commerce platform. Consider aspects like the user interface, navigation, product listing, product pages, shopping cart, and checkout process. You can use wireframing or prototyping tools to create a visual representation of your platform.

**ALGORITHIMS:**

Designing the layout of your artisanal e-commerce platform is a critical step in creating a user-friendly and visually appealing website. Here's a basic wireframe of the key components of the platform, including the user interface, navigation, product listing, product pages, shopping cart, and checkout process:

**Loading the Dataset**

import pandas as pd

# Load the dataset

df = pd.read\_csv('ecommerce\_data.csv')

# Print the first few rows to verify the data has been loaded correctly

print(df.head())

**Dataset**

**A table with numbers and numbers

Description automatically generated**

**Preprocessing**

import pandas as pd

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

# Assuming 'df' is your DataFrame

# Step 1: Handling Missing Values

df.fillna(df.mean(), inplace=True)

# Step 2: Handling Categorical Data (One-Hot Encoding)

df = pd.get\_dummies(df, columns=['categorical\_column'])

# Step 3: Feature Scaling

scaler = StandardScaler()

df[['numerical\_column1', 'numerical\_column2']] = scaler.fit\_transform(df[['numerical\_column1', 'numerical\_column2']])

# Step 8: Splitting Data

X = df.drop('target\_variable', axis=1)

y = df['target\_variable']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

**Challenges faced by an IBM E-Commerce Platform:**

**1. Platform Complexity:** Understanding the intricacies of the cloud platform, including deployment, services, and configurations, can pose challenges for newcomers to this environment.

**2. Service Integration:** The complex task of seamlessly integrating various services (e.g., databases, caching, authentication) with the application can be daunting.

**3. Scalability and Performance:** Designing the application to efficiently handle high traffic loads, particularly during peak times, can be quite demanding.

**4. Security Concerns:** Ensuring that the application is safeguarded against common web vulnerabilities like SQL injection and cross-site scripting is of paramount importance.

**5. Data Management:** Managing data effectively, including storage, retrieval, and the organization of user information, products, and orders, can be a significant challenge.

**6. User Authentication and Authorization:** The implementation of secure and user-friendly authentication and authorization mechanisms is essential to protect user accounts and sensitive data.

**7. Payment Processing**: Integrating a secure and compliant payment gateway can be a complex endeavor.

**8. User Experience (UX) and Interface Design:** Designing an intuitive and visually pleasing user interface that delivers a seamless user experience is a significant undertaking.

**9. Cross-Browser Compatibility:** Ensuring that the application functions correctly across various web browsers and devices can be challenging due to varying compatibility levels.

**10. Regulatory Compliance**: Adhering to legal and regulatory requirements, especially those related to data privacy and security, can be a complex and time-consuming process.

**11. Testing and Quality Assurance:** Conducting rigorous testing for functionality, security, and performance is indispensable, and setting up automated testing suites can be challenging.

**12. Documentation and Knowledge Transfer:** Maintaining comprehensive documentation for the application's architecture, deployment process, and codebase can often be overlooked but is crucial for future development and maintenance.

**User Interface:**

**Header:**

- Logo: Incorporate your brand logo.

- Navigation Menu: Include options like Home, Shop, Categories, About, Contact, and Account (for user profiles), and Cart.

- Search Bar: A tool for searching for products.

- User Account: Include options for Login, Register, and User Profile.

**Hero Banner:**

- Utilize a high-quality image to showcase featured or seasonal products.

- Add a Call-to-Action (CTA) button, encouraging users to Shop Now.

**Product Categories:**

- Present a grid or list of product categories (e.g., jewelry, home decor, clothing) with accompanying images and category names.

**Featured Products:**

- Highlight a selection of best-selling or new arrival products, displaying images, names, and prices.

- Include "Shop Now" buttons for each featured product.

**Navigation:**

- Maintain a clear and user-friendly navigation menu with dropdowns for product categories.

- Implement breadcrumbs to show users their location within the site.

**Product Listing:**

- Display products in a grid or list view within a selected category.

- For each product, showcase the product image, name, price, and an "Add to Cart" button.

- Consider adding a "Quick View" button for more details without leaving the page.

- Implement pagination for multiple pages of products.

**Product Pages:**

- Each product page should include a large product image.

- Present the product title, description, price, and rating.

- Allow users to select quantities and include an "Add to Cart" button.

- Provide additional product details such as materials and dimensions.

- Display customer reviews and ratings.

- Suggest related products or include a "Customers also bought" section.

- Add social sharing buttons and an option to "Add to Wishlist" or "Save for Later."

**Shopping Cart:**

- Present a list of items in the cart, complete with images, names, quantities, and prices.

- Include sections for the subtotal, taxes, shipping, and total cost.

- Offer the option to apply promo codes or discounts and enable users to update the cart (e.g., change quantities or remove items).

- Include a "Proceed to checkout" button.

**Checkout Process:**

- Step 1: Shipping Information - This step should encompass a shipping address form, shipping method options, and associated shipping costs.

- Step 2: Payment Information - Include payment method options (e.g., credit card, PayPal) and billing address information if it differs from the shipping address. Present an order summary.

- Step 3: Review and Confirm - Provide a summary of the order, including products, shipping, and payment details. Request user agreement to terms and conditions and place the order with a relevant button.

**Confirmation:**

- After the order is placed, display an order confirmation message, order number, order details, and estimated delivery time. Optionally, provide users with the opportunity to create an account for easy order tracking.

**Additional Features:**

- Include a user account section for managing profiles, order history, and saved addresses.

- Implement a wishlist functionality, allowing users to save products for future consideration.

- Offer a newsletter signup option for updates and promotions.

- Create a contact page with a form for user inquiries.

- Display trust symbols, such as secure payment icons and customer reviews.

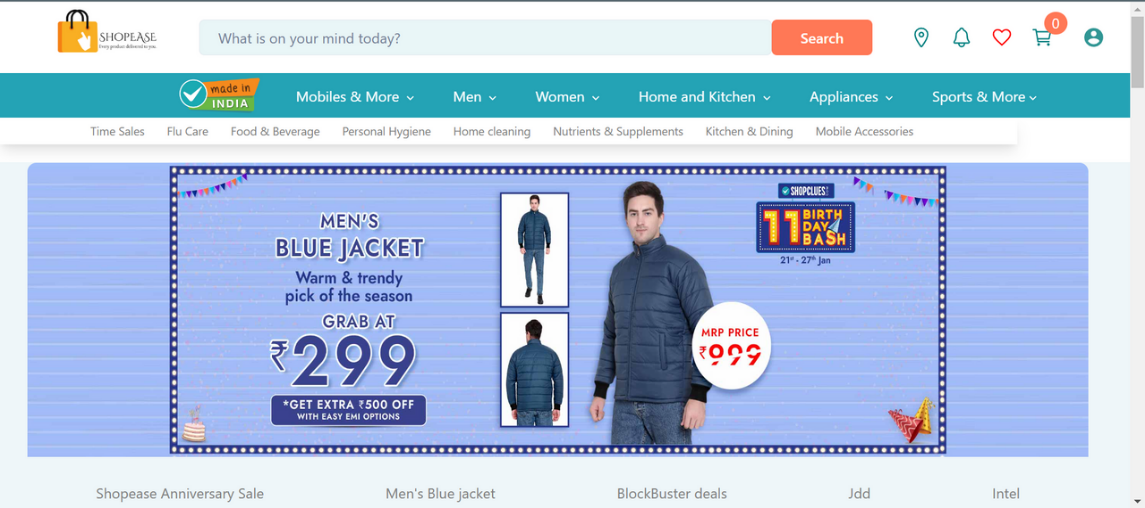
- Ensure a responsive design to cater to mobile and tablet users.

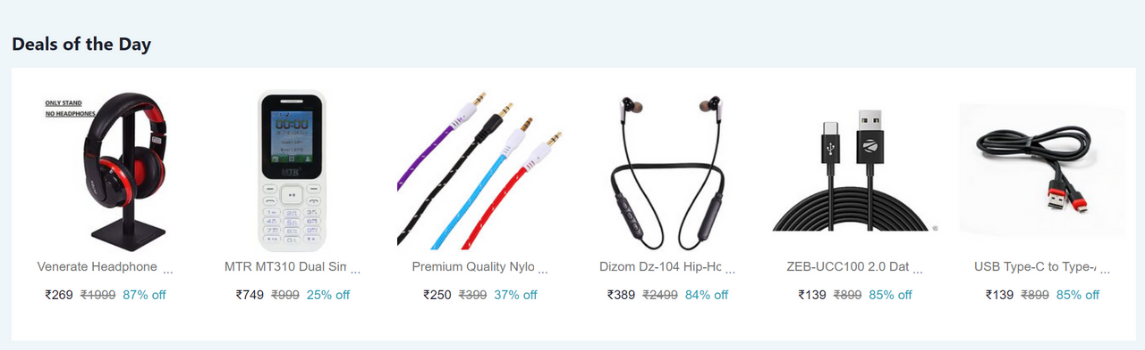
- Utilize high-quality product images and professional photography.

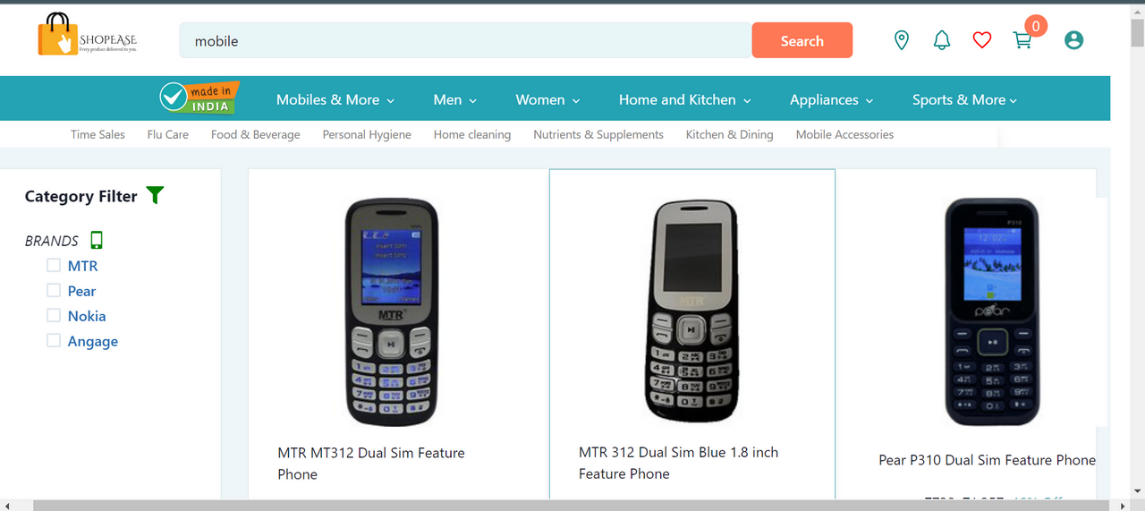
**Tools Used:**

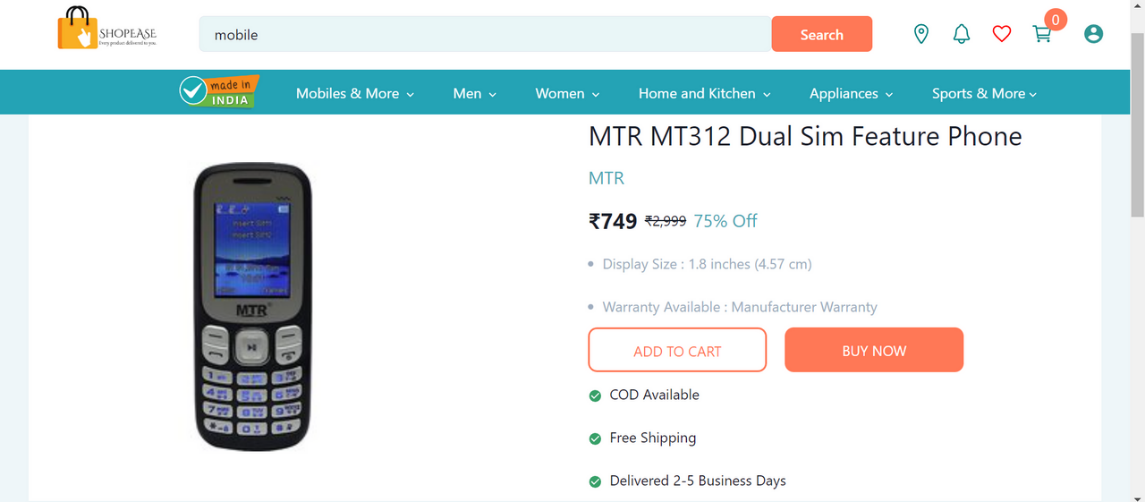
The layout of the E-commerce app was implemented using HTML, CSS, and JavaScript,Mern Stack.

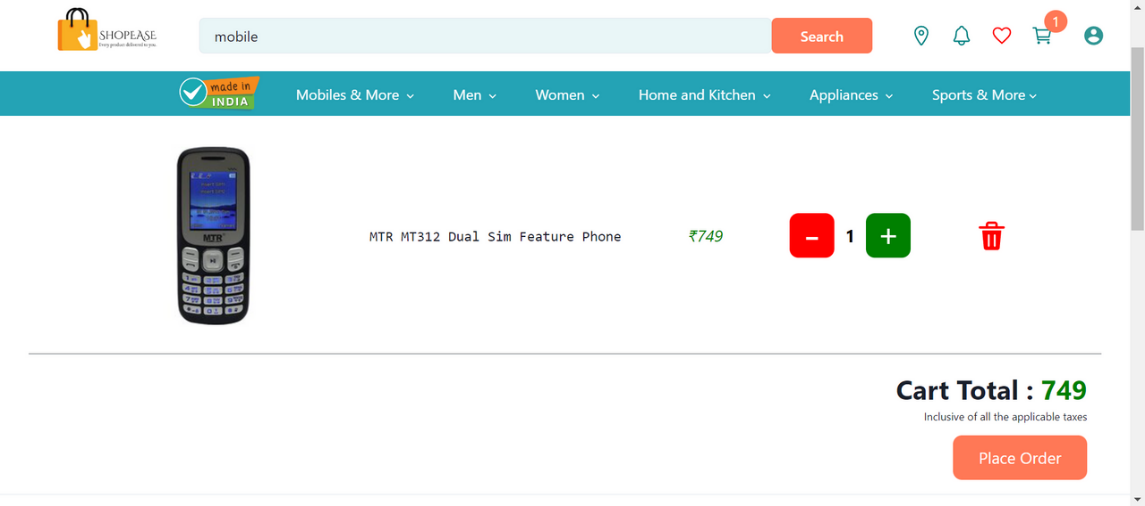
**APP SCREENSHOT:**

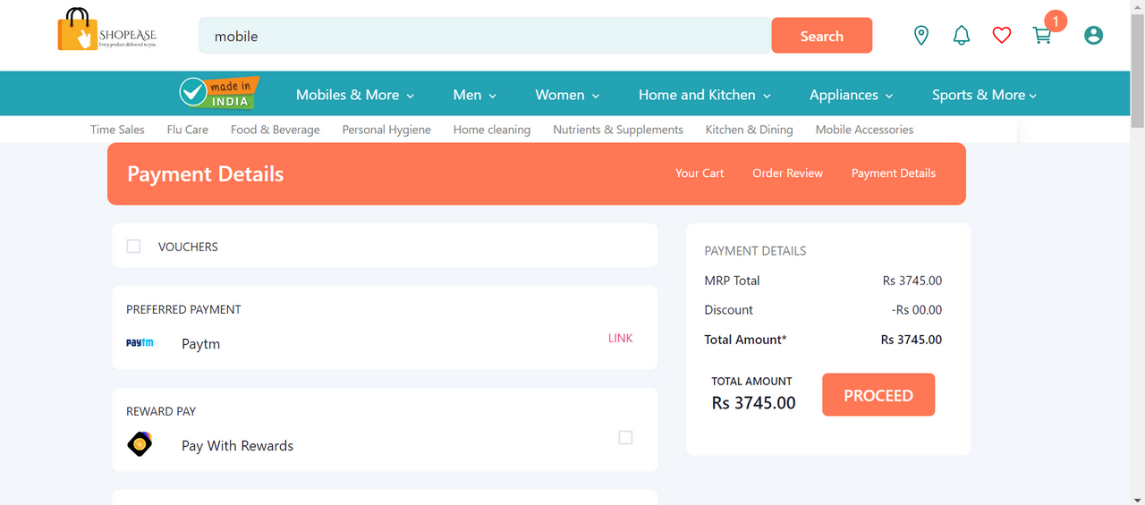
[](https://camo.githubusercontent.com/cc61d50c0fe905f3b4cb0f83003acdfb3b2bdbb21b7ecf52a037925316294e53/68747470733a2f2f692e706f7374696d672e63632f6d7257327a7a5a4d2f53637265656e73686f742d32303233303132322d3039313732312e706e67)

[](https://camo.githubusercontent.com/2ddef8cddc42998b4d8e8d1244ca57a6e04d0c0aeff6c552cbe2781ebe8f3d8b/68747470733a2f2f692e706f7374696d672e63632f59327639507750762f53637265656e73686f742d32303233303132322d3039313735382e706e67)

[](https://camo.githubusercontent.com/450e7306daed7b0b96a8dce8f7732e2656f72798df387b0ac75f4f86787eca24/68747470733a2f2f692e706f7374696d672e63632f6e4c77784b5870302f53637265656e73686f742d32303233303132322d3039313832322e706e67)

[](https://camo.githubusercontent.com/950115dfc84d4770b8135787e19c26d96afb8dbe43b2bb1a77b49a41eda78ad2/68747470733a2f2f692e706f7374696d672e63632f6e633848773937572f53637265656e73686f742d32303233303132322d3039313930312e706e67)

[](https://camo.githubusercontent.com/c41e15317041a4e1f1413af72fd83ad659cd13f8eea63df372a89e797aa2d4ce/68747470733a2f2f692e706f7374696d672e63632f62763679363648302f53637265656e73686f742d32303233303132322d3039313931392e706e67)

[](https://camo.githubusercontent.com/9a3957f3d0254da5ee1e2c67a2c5ba981583bc6fafde372226d720b17fbc0624/68747470733a2f2f692e706f7374696d672e63632f42765a5a425744512f53637265656e73686f742d32303233303132322d3039313935312e706e67)