

# Phase 4 - E-commerce Application Development in IBM Cloud Foundry

## Introduction:

Welcome to the second phase of our journey in building a comprehensive E-commerce application within the IBM Cloud Foundry ecosystem. In Part 1, we laid the essential foundation, focusing on database integration and front-end development. Now, in Part 2, we enter the critical realm of backend development, where we will explore the intricate workings that power your E-commerce platform.

In this part of our series, we will take a deep dive into the backend of your E-commerce application, completing the core components necessary to deliver a seamless and feature-rich online shopping experience. The backend serves as the engine that drives your E-commerce platform, handling critical functions such as data processing, user authentication, order management, and more.

Here's what you can expect to delve into in this phase of our project:

**Server-Side Logic:** We will guide you through the development of server-side logic, explaining how to handle data processing, user authentication, and other essential functionalities. You'll learn how to create efficient and reliable backend services that support your E-commerce operations.

**APIs and Microservices:** In the modern E-commerce landscape, flexibility and scalability are key. We will explore the use of APIs and microservices to create a modular and agile backend architecture that can adapt to changing demands and evolving business needs.

**Integrating with the Frontend:** Your backend and frontend must work in harmony to deliver a seamless user experience. We'll show you how to integrate your backend services with the frontend components, ensuring smooth communication and data flow.

**User Management and Authentication:** Security and user management are paramount in E-commerce. We will delve into the implementation of robust user authentication and authorization mechanisms to protect customer data and ensure a secure shopping environment.

**Order Processing and Inventory Management:** Efficient order processing and inventory management are critical to the success of your E-commerce business. We will cover how to handle orders, update inventory, and keep track of product availability.

### CSS CODE:

```
/*-----*\
```

```
#SIDEBAR
```

```
\*-----*/
```

```
.sidebar {  
  background: var(--white);  
  position: fixed;  
  top: 0;  
  left: -100%;  
  bottom: 0;  
  width: 100%;  
  max-width: 320px;  
  padding: 30px;  
  overflow-y: scroll;  
  overscroll-behavior: contain;  
  visibility: hidden;
```

```
    transition: 0.5s ease;
    z-index: 20;
}
```

```
.sidebar.active {
    left: 0;
    visibility: visible;
}
```

```
.sidebar-category {
    margin-bottom: 15px;
    padding-bottom: 15px;
    border-bottom: 1px solid var(--cultured);
}
```

```
.sidebar-top {
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 10px;
}
```

```
.sidebar-title {
    color: var(--onyx);
    font-size: var(--fs-5);
    text-transform: uppercase;
    letter-spacing: 0.8px;
```

```
font-weight: var(--weight-600);  
}
```

```
.sidebar-close-btn {  
  color: var(--eerie-black);  
  font-size: 22px;  
  font-weight: var(--weight-600);  
}
```

```
.sidebar-close-btn ion-icon { --icon-stroke-width: 50px; }
```

```
.sidebar-accordion-menu {  
  width: 100%;  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  padding: 7px 0;  
}
```

```
.sidebar .menu-title-flex {  
  display: flex;  
  align-items: center;  
  gap: 10px;  
}
```

```
.sidebar .menu-title {  
  font-size: var(--fs-5);
```

```
color: var(--sonic-silver);  
font-weight: var(--weight-500);  
}
```

```
.sidebar-accordion-menu ion-icon {  
  color: var(--sonic-silver);  
  font-size: 14px;  
  --icon-stroke-width: 70px;  
}
```

```
.sidebar-submenu-category-list {  
  border-top: 1px solid var(--cultured);  
  max-height: 0;  
  overflow: hidden;  
  visibility: hidden;  
  transition: 0.5s ease-in-out;  
}
```

```
.sidebar-submenu-category-list.active {  
  padding: 13px 0 8px;  
  max-height: 122px;  
  visibility: visible;  
}
```

```
.sidebar-submenu-title {  
  display: flex;  
  justify-content: space-between;
```

```
align-items: center;
color: var(--sonic-silver);
font-size: var(--fs-7);
padding: 2px 0;
}
```

```
.sidebar-submenu-title:hover { color: var(--eerie-black); }
```

```
.sidebar .product-name { text-transform: capitalize; }
```

```
.sidebar-accordion-menu.active .add-icon,
.sidebar-accordion-menu .remove-icon { display: none; }
```

```
.sidebar-accordion-menu .add-icon,
.sidebar-accordion-menu.active .remove-icon { display: block; }
```

```
.sidebar .showcase-heading {
  font-size: var(--fs-5);
  font-weight: var(--weight-600);
  color: var(--onyx);
  text-transform: uppercase;
  letter-spacing: 0.8px;
  margin-bottom: 15px;
}
```

```
.sidebar .showcase {
  display: flex;
```

```
align-items: center;
gap: 15px;
}
```

```
.sidebar .showcase:not(:last-child) { margin-bottom: 15px; }
```

```
.sidebar .showcase-img { border-radius: var(--border-radius-sm); }
```

```
.sidebar .showcase-content { width: calc(100% - 90px); }
```

```
.sidebar .showcase-title {
  color: var(--onyx);
  font-size: var(--fs-7);
  font-weight: var(--weight-400);
  text-transform: capitalize;
  overflow: hidden;
  white-space: nowrap;
  text-overflow: ellipsis;
  letter-spacing: 0.5px;
}
```

```
.sidebar .showcase-rating {
  display: flex;
  align-items: center;
  color: var(--sandy-brown);
  font-size: 13px;
  padding: 4px 0;
```

```
}
```

```
.sidebar .price-box {  
  display: flex;  
  align-items: center;  
  gap: 15px;  
}
```

```
.sidebar .price-box del {  
  color: var(--sonic-silver);  
  font-size: 13px;  
}
```

```
.sidebar .price-box .price {  
  font-size: var(--fs-7);  
  font-weight: var(--weight-600);  
  color: var(--davys-gray);  
}
```

```
/*-----*\n#PRODUCT MINIMAL\n/*-----*/
```

```
.product-minimal { margin-bottom: 30px; }
```



```
.product-minimal .product-showcase { margin-bottom: 10px; }
```

```
.product-minimal .showcase-wrapper {  
  display: flex;  
  align-items: center;  
  overflow-x: auto;  
  overscroll-behavior-inline: contain;  
  scroll-snap-type: inline mandatory;  
}
```

```
.product-minimal .showcase-container {  
  min-width: 100%;  
  padding: 0 5px;  
  scroll-snap-align: start;  
}
```

```
.product-minimal .showcase {  
  display: flex;  
  justify-content: flex-start;  
  align-items: center;  
  gap: 15px;  
  border: 1px solid var(--cultured);  
  padding: 15px;  
  border-radius: var(--border-radius-md);  
}
```

```
.product-minimal .showcase:not(:last-child) { margin-bottom: 15px; }
```

```
.product-minimal .showcase-content { width: calc(100% - 85px); }
```

```
.product-minimal .showcase-title {  
  color: var(--eerie-black);  
  font-size: var(--fs-7);  
  font-weight: var(--weight-600);  
  text-transform: capitalize;  
  white-space: nowrap;  
  overflow: hidden;  
  text-overflow: ellipsis;  
  margin-bottom: 2px;  
}
```

```
.product-minimal .showcase-category {  
  width: max-content;  
  color: var(--davys-gray);  
  font-size: var(--fs-8);  
  text-transform: capitalize;  
  margin-bottom: 3px;  
}
```

```
.product-minimal .showcase-category:hover { color: var(--salmon-pink); }
```

```
.product-minimal .price-box {  
  display: flex;  
  align-items: center;
```

```
gap: 10px;
}
```

```
.product-minimal .price {
  font-size: var(--fs-7);
  font-weight: var(--weight-700);
  color: var(--salmon-pink);
}
```

```
.product-minimal .price-box del {
  font-size: var(--fs-9);
  color: var(--sonic-silver);
}
```

**A small part of the css code is only given due to space constraints full source code is available in the [GitHub](#). After the frontend designing we move on to the backend development of the project.**

### **Backend Code:**

```
'use strict';
```

```
// modal variables
```

```
const modal = document.querySelector('[data-modal]');
```

```
const modalCloseBtn = document.querySelector('[data-modal-close]');
```

```
const modalCloseOverlay = document.querySelector('[data-modal-overlay]');
```

```
// modal function
```

```
const modalCloseFunc = function () { modal.classList.add('closed') }
```

```
// modal eventListener
modalCloseOverlay.addEventListener('click', modalCloseFunc);
modalCloseBtn.addEventListener('click', modalCloseFunc);

// notification toast variables
const notificationToast = document.querySelector('[data-toast]');
const toastCloseBtn = document.querySelector('[data-toast-close]');

// notification toast eventListener
toastCloseBtn.addEventListener('click', function () {
  notificationToast.classList.add('closed');
});

// mobile menu variables
const mobileMenuOpenBtn = document.querySelectorAll('[data-mobile-menu-open-btn]');
const mobileMenu = document.querySelectorAll('[data-mobile-menu]');
const mobileMenuCloseBtn = document.querySelectorAll('[data-mobile-menu-close-btn]');
const overlay = document.querySelector('[data-overlay]');

for (let i = 0; i < mobileMenuOpenBtn.length; i++) {

  // mobile menu function
  const mobileMenuCloseFunc = function () {
    mobileMenu[i].classList.remove('active');
```

```
overlay.classList.remove('active');  
}
```

```
mobileMenuOpenBtn[i].addEventListener('click', function () {  
  mobileMenu[i].classList.add('active');  
  overlay.classList.add('active');  
});
```

```
mobileMenuCloseBtn[i].addEventListener('click', mobileMenuCloseFunc);  
overlay.addEventListener('click', mobileMenuCloseFunc);  
  
}
```

```
// accordion variables
```

```
const accordionBtn = document.querySelectorAll('[data-accordion-btn]');  
const accordion = document.querySelectorAll('[data-accordion]');
```

```
for (let i = 0; i < accordionBtn.length; i++) {
```

```
  accordionBtn[i].addEventListener('click', function () {
```

```
    const clickedBtn = this.nextElementSibling.classList.contains('active');
```

```
    for (let i = 0; i < accordion.length; i++) {
```

```
      if (clickedBtn) break;
```

```
      if (accordion[i].classList.contains('active')) {
```

```
    accordion[i].classList.remove('active');  
    accordionBtn[i].classList.remove('active');  
  }  
}  
this.nextElementSibling.classList.toggle('active');  
this.classList.toggle('active');  
});  
}
```

Integrating your E-commerce backend with IBM Cloud Foundry involves setting up and connecting the backend services to the platform. Here's a step-by-step guide on how to integrate your backend with IBM Cloud Foundry:

1. **Create Backend Services:** First, ensure that your backend services are developed and ready for deployment. This may include services for user management, order processing, inventory management, and other core E-commerce functions.
2. **Choose a Cloud Foundry Service:** IBM Cloud Foundry provides a range of services that can help you integrate your backend. For E-commerce, you might need services like databases (e.g., IBM Db2), message queues (e.g., IBM Message Hub), or caching (e.g., Redis). Depending on your needs, select the appropriate service(s) from the IBM Cloud catalog.
3. **Deployment:** Deploy your backend services to IBM Cloud Foundry. You can use the **cf push** command-line tool or the IBM Cloud Console to deploy your applications and services. Make sure to specify the necessary runtime environment and resource allocation.
4. **Bind Services:** Once your backend services are deployed, bind them to the selected Cloud Foundry service(s). This step establishes a connection between your services and the platform's offerings, enabling them to communicate and share data. Use the **cf bind-service** command or the Cloud Foundry Console to complete this binding process.

5. **Environment Variables:** Store sensitive information and configuration settings, such as API keys and database connection strings, as environment variables within your Cloud Foundry service. This ensures that your backend services can securely access these resources without exposing sensitive data in your codebase.
6. **Configuration and Secrets Management:** IBM Cloud Foundry offers options for managing configuration and secrets securely. You can utilize the built-in capabilities or integrate with external secrets management services for added security.
7. **Testing and Monitoring:** Thoroughly test your integrated backend services to ensure they are functioning correctly within the IBM Cloud Foundry environment. Use monitoring tools and logs to keep track of service performance, detect errors, and troubleshoot issues.
8. **Access Control and Security Policies:** Implement access control and security policies to protect your backend services. Utilize Cloud Foundry's security features to restrict access to authorized users and protect against unauthorized access or data breaches.
9. **API Gateway:** If your E-commerce backend includes multiple microservices, consider using an API gateway to manage and secure your APIs. IBM Cloud Foundry can work in conjunction with API gateways to provide centralized management and security for your API endpoints.

IBM Cloud Foundry is a powerful platform for hosting and managing the backend of your E-commerce application. Here are some key advantages and use cases for utilizing IBM Cloud Foundry in the backend development of your E-commerce platform:

**Scalability and Flexibility:** IBM Cloud Foundry offers the ability to easily scale your backend services up or down to handle varying levels of traffic and demand. This flexibility is crucial for an E-commerce platform, where traffic can fluctuate significantly based on various factors such as promotions, holidays, and product launches. By leveraging Cloud Foundry, you can ensure that your backend can adapt to these changes seamlessly.

**Microservices Architecture:** IBM Cloud Foundry supports the microservices architecture, which is well-suited for E-commerce applications. You can break down your backend into smaller, independent services, each responsible for a specific function, such as user authentication, inventory management, or order processing. This modular approach makes it easier to develop, maintain, and scale individual components of your application.

**Developer Productivity:** Cloud Foundry simplifies the deployment and management of backend services. Developers can focus on coding without worrying about server provisioning or infrastructure management. This can significantly increase developer productivity and allow them to concentrate on creating features and improving user experiences.

**High Availability and Reliability:** IBM Cloud Foundry is designed to provide high availability and reliability. It offers features such as load balancing, auto-recovery, and redundancy to ensure that your backend services are accessible and performant at all times. This is critical for E-commerce platforms where downtime can lead to lost sales and customer frustration.

**Security and Compliance:** Security is of utmost importance in E-commerce. IBM Cloud Foundry provides robust security features and allows you to implement security best practices for your backend services. You can also ensure compliance with industry regulations and standards, such as GDPR or PCI DSS, to protect customer data and transactions.

**Integration with Other IBM Services:** IBM Cloud Foundry can seamlessly integrate with other IBM Cloud services, such as databases, analytics, and AI services. This enables you to enhance your E-commerce backend with data analytics, personalization, and other advanced capabilities to improve customer experiences and drive business growth.