IBM-CLOUD APPLICATION DEVELOPMENT

GROUP-3

PHASE 5 **Project Documentation & Submission**

PROJECT: E-commerce web Application Creating using IBM cloud Database.



INTRODUTION:

Design thinking and innovation are essential to solving problems in IBM Cloud application development. Here's a framework to integrate design and innovation into the process:

1. Problem Identification and Understanding:

Design Thinking: Start by empathizing with the users and understanding their pain points and needs in the context of IBM Cloud application development. Conduct user research and gather feedback.

Innovation: Encourage a culture of innovation within the organization. Create forums or mechanisms for employees to share their ideas and feedback. Promote an open and collaborative environment for brainstorming.

2. Ideation and Solution Generation:

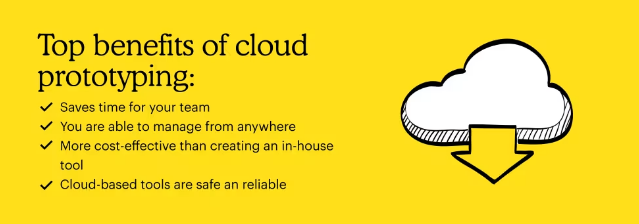
Design Thinking: Use ideation techniques to generate creative solutions that address the identified problems. Collaborate with cross-functional teams and involve designers, developers, and business stakeholders in the ideation process.

Innovation: Implement structured innovation methodologies, such as the Innovation Funnel, to filter, evaluate, and prioritize ideas. Experiment with new technologies and development approaches.

3. Prototyping and Testing:

Design Thinking: Develop prototypes of your cloud applications to quickly test and validate concepts with users. This can involve wireframes, mockups, or clickable prototypes.

Innovation: Invest in innovation labs or sandboxes where teams can experiment with emerging technologies, test new development methodologies, and explore alternative cloud services.



4. User-Centered Design:

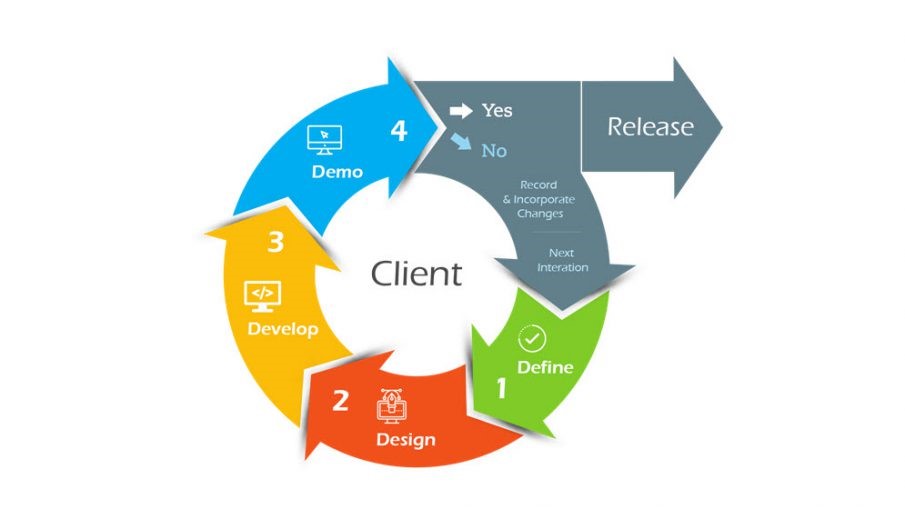
Design Thinking: Ensure that the design of your IBM Cloud applications is user-centered. Conduct usability testing and gather feedback from users at every stage of development.

Innovation: Challenge the status quo and explore disruptive technologies that can fundamentally change the way applications are developed and deployed in the cloud.

5. Agile Development and Continuous Improvement:

Design Thinking: Implement agile development practices that allow for iterative improvements based on user feedback. Continuously refine the design and user experience.

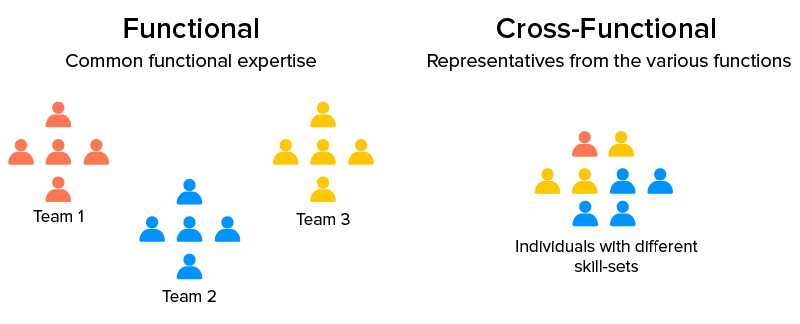
Innovation: Foster a culture of continuous improvement and experimentation. Encourage teams to regularly review and optimize their development processes and toolchains.



6. Collaboration and Cross-Functional Teams:

Design Thinking: Promote collaboration between designers, developers, and other stakeholders. Ensure that all team members have a shared understanding of the user's needs and the project's goals.

Innovation: Encourage multidisciplinary teams to work together, bringing diverse perspectives to the development process. This can lead to innovative solutions that address complex challenges.



7. User Feedback and Data-Driven Decision Making:

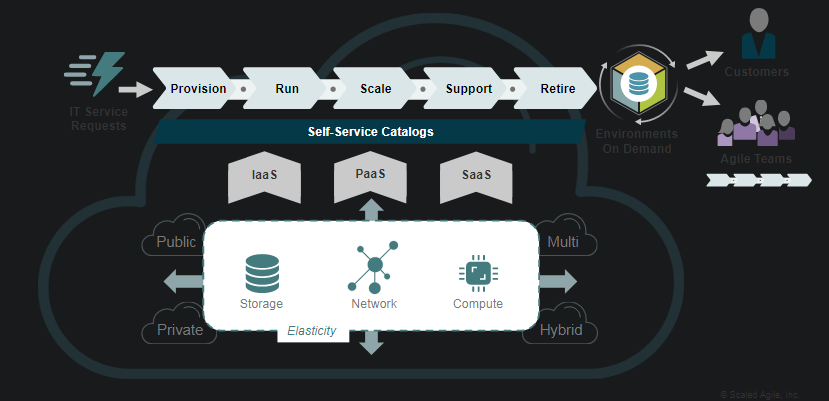
Design Thinking: Continuously gather user feedback, both qualitative and quantitative, to inform design decisions and prioritize improvements.

Innovation: Leverage data analytics and insights to make informed decisions about which innovative ideas have the most potential for improving IBM Cloud application development.

8. Scalable and Sustainable Solutions:

Design Thinking: Design for scalability and sustainability, considering long-term user needs and system requirements.

Innovation: Explore new cloud technologies and architectures that can provide scalable and cost-effective solutions, such as serverless computing or container orchestration.



PROGRAM:

HTML CODING:

### **HTML (index.html)**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>E-commerce Web App</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<header>

<h1>Our E-commerce Store</h1>

</header>

<div id="productsList">

<!-- Product list will be populated here -->

</div>

<script src="scripts.js"></script>

</body>

</html>

CSS CODING:

### **CSS (styles.css)**

/\* Your CSS styles for layout, colors, fonts, etc. \*/

header {

background-color: #333;

color: white;

text-align: center;

padding: 20px;

}

/\* Styles for product display \*/

.product {

border: 1px solid #ccc;

margin-bottom: 20px;

padding: 10px;

}

JAVASCRIPT CODING:

### **JavaScript (scripts.js)**

// Fetch data from the backend API

fetch('get\_products.php')

.then(response => response.json())

.then(products => {

const productsList = document.getElementById('productsList');

products.forEach(product => {

// Display product information

const productDiv = document.createElement('div');

productDiv.classList.add('product');

productDiv.innerHTML = `

<h2>${product.name}</h2>

<p>Price: $${product.price}</p>

<p>Description: ${product.description}</p>

`;

productsList.appendChild(productDiv);

});

})

.catch(error => console.error('Error fetching products:', error));

PHP CODING:

### **PHP (get\_products.php)**

<?php

// Code to connect to your IBM Cloud database (using PDO or IBM DB2 functions)

// Fetch products from the database and return as JSON

$products = [

['name' => 'Product 1', 'price' => 19.99, 'description' => 'Description for Product 1'],

['name' => 'Product 2', 'price' => 29.99, 'description' => 'Description for Product 2'],

// ... (retrieve products from the database)

];

header('Content-Type: application/json');

echo json\_encode($products);

?>

Feature Engineering

Feature engineering in the context of an e-commerce web application using an IBM Cloud database involves creating or enhancing the data features that can be used for various data analysis and machine learning tasks. Here are some steps and ideas for feature engineering in an e-commerce context using an IBM Cloud database:



1. Data Retrieval:

- Connect to your IBM Cloud database and retrieve the relevant data for your e-commerce application. This might include product information, customer data, transaction history, and more.

2. Categorical Features:

- Convert categorical features into numerical representations. For example, you can use one-hot encoding or label encoding to transform product categories, customer types, or payment methods into numeric values.

3. Temporal Features:

- Extract meaningful information from timestamps. For example, you can create features like day of the week, month, or time of day for orders or customer activity.

4. Session Features:

- If your e-commerce platform tracks user sessions, you can engineer features related to session duration, pages viewed, and actions taken within a session.

5. Geospatial Features:

- If your application involves physical locations, consider creating features related to customer locations, shipping distances, or regional sales trends.



Remember that the choice of features should align with the specific objectives of your e-commerce project, whether it's for customer segmentation, recommendation systems, sales prediction, or any other application. Additionally, as the data evolves over time, continuously update and re-engineer features to adapt to changing patterns and user behavior.

MODEL TRAINING

Training a model for an e-commerce web application using data stored in an IBM Cloud database involves the process of using machine learning techniques to make predictions or gain insights from the data. Below are the general steps to train a model for an e-commerce application using an IBM Cloud database:



1. Model Selection:

- Choose an appropriate machine learning or deep learning model based on the nature of your problem. For example, for sales prediction, regression models like linear regression or decision trees may be suitable, while for recommendation systems, collaborative filtering or deep learning models could be used.

2. Model Training:

- Train the selected model on the training data. This involves feeding the input features and the target variable to the model and adjusting its parameters to minimize the error or loss function.

3. Model Evaluation:

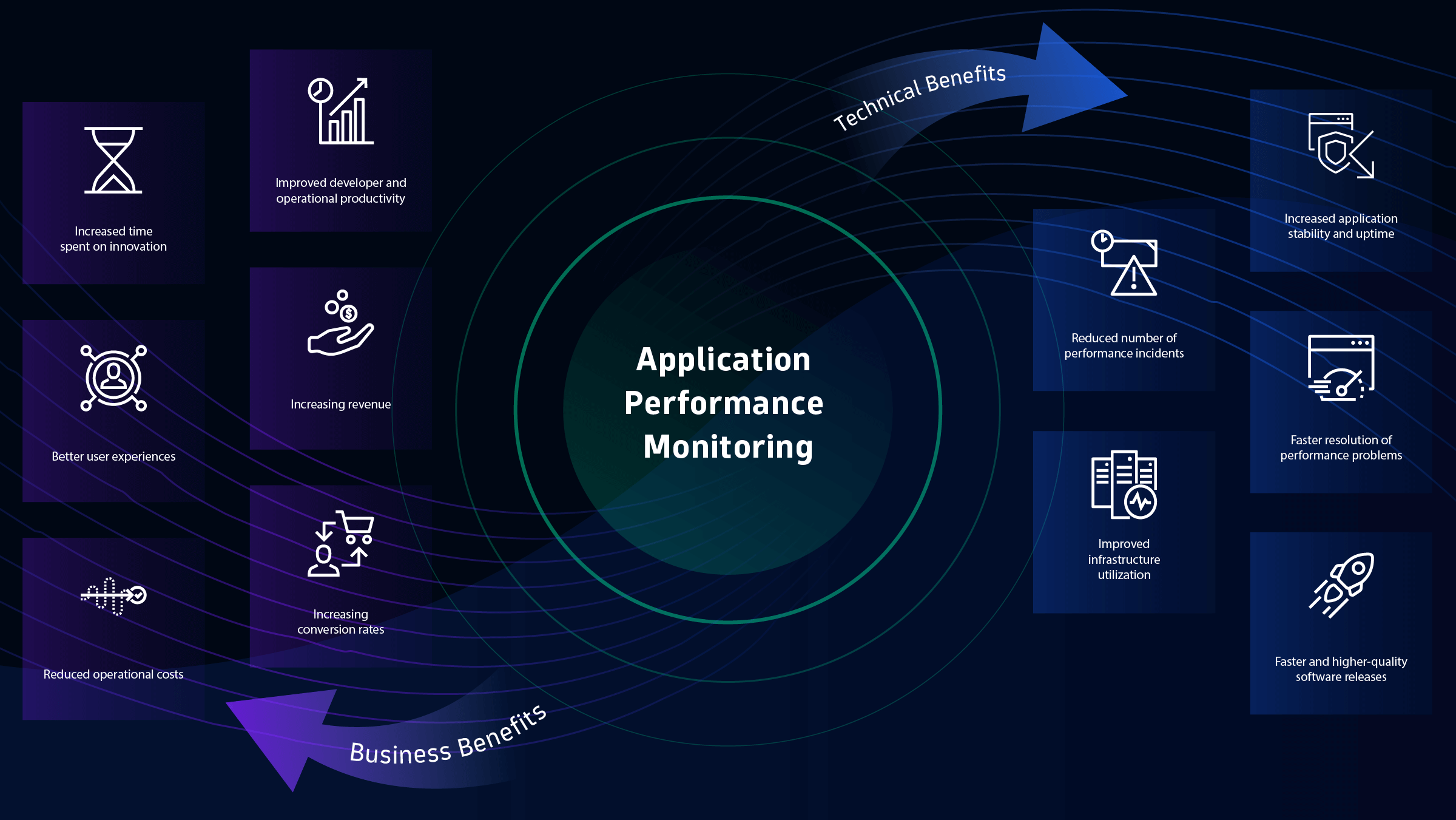
- Evaluate the model's performance on the validation set using appropriate metrics. For example, for regression tasks, you might use metrics like mean squared error (MSE) or root mean squared error (RMSE), while classification tasks may use metrics like accuracy, F1-score, or ROC AUC.

4. Model Testing:

- After finalizing your model, evaluate its performance on the testing set to assess its generalization capability. This gives you a sense of how well the model will perform on new, unseen data.

5. Model Deployment:

- If the model meets your performance requirements, deploy it to your e-commerce web application. This may involve creating APIs or integrating the model into the application to make real-time predictions or recommendations.



Keep in mind that model training is an iterative process, and you may need to fine-tune your model or update it as new data becomes available or as the e-commerce platform evolves. The choice of model and the specific techniques used will depend on your application's objectives, whether it's sales prediction, customer segmentation, or recommendation systems.

EVALUATING

Evaluating an e-commerce web application using an IBM Cloud database involves assessing its performance, user experience, and the effectiveness of various features and functionalities. Evaluation can be conducted through a combination of automated testing, user testing, and analytical assessments. Here are some key aspects to consider when evaluating an e-commerce web application using an IBM Cloud database:



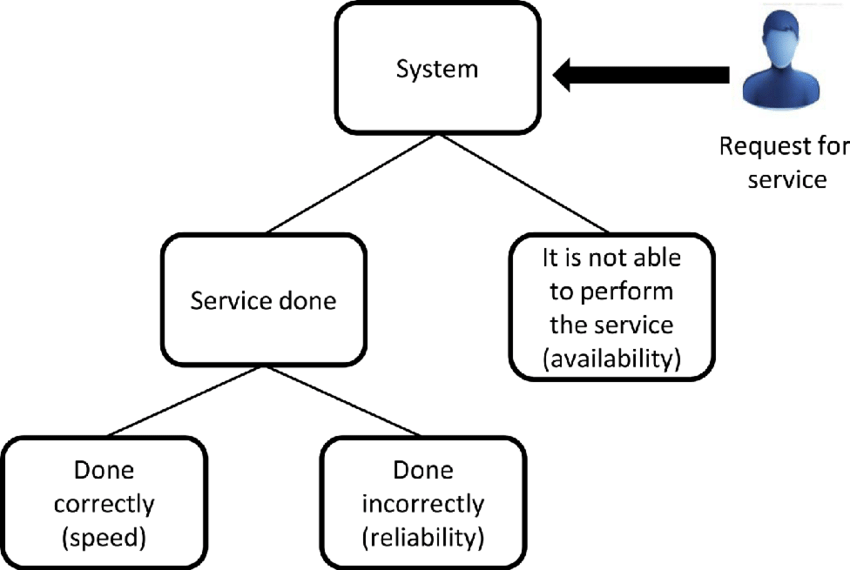
1. Performance Evaluation:

**Page Load Times**: Measure the time it takes for web pages to load, including the homepage, product pages, and checkout pages. Optimize for fast load times to enhance user experience.

**Scalability**: Assess the application's ability to handle increasing traffic and transactions without degrading performance. Test it under load to identify potential bottlenecks.

**Server Response Times**: Monitor server response times and minimize delays to ensure a responsive user experience.

**Error Handling**: Evaluate how the application handles errors, such as 404 errors, and ensure that users are provided with clear error messages.



2. User Experience Evaluation:

**User Interface (UI) Design**: Assess the design, layout, and visual appeal of the website. Ensure that it is user-friendly and visually appealing.

**Navigation**: Evaluate the ease of navigation within the website. Ensure that users can easily find products, access their shopping cart, and check out.

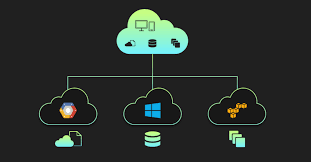
**Mobile Responsiveness**: Test the website's compatibility with different devices, including mobile phones and tablets.

**Search Functionality**: Evaluate the search feature's accuracy and efficiency in helping users find products.

**Checkout Process**: Review the checkout process to ensure it is streamlined and straightforward. Minimize the number of steps required for a successful purchase.

**Payment Processing**: Verify that payment processing is secure, reliable, and supports various payment methods.

**User Registration**: Evaluate the registration process, and consider offering guest checkout options to reduce friction for first-time customers.



Regularly gather and analyze data to track key performance indicators (KPIs) related to user engagement, conversion rates, and revenue. This ongoing evaluation will help identify areas for improvement and guide decisions for optimizing the e-commerce web application.

conclusion:

By integrating design thinking and innovation into IBM Cloud application development, you can create solutions that not only address existing problems but also have the potential to transform the way applications are built, deployed, and managed on the cloud. This approach fosters a culture of continuous improvement and adaptability, positioning your organization for long-term success in the cloud application development landscape.

It's important to note that the specific services and tools available on IBM Cloud may have evolved since my last update, so it's advisable to check the latest IBM Cloud offerings and documentation for the most up-to-date information and guidance on loading and preprocessing data for an e-commerce application.