Project Title: E-Commerce application on cloud foundry

Phase 5: PROJECT DOCUMENTATION & SUBMISSION

OBJECTIVE:

The objective of this project is to design and develop a personal travel blog website that is informative, engaging, and easy to use. The website should provide visitors with a valuable resource for planning and booking their own trips, as well as be a platform for the blogger to .share their own travel experiences and insights

Problem Definition:

- Developing an e-commerce application on IBM Cloud Foundry poses challenges such as ensuring seamless scalability, optimizing performance, and integrating secure payment gateways.
- Balancing these technical aspects while meeting user expectations for a responsive and user-friendly platform is a critical problem to address.
- Additionally, managing inventory, order processing, and customer data securely are essential considerations in delivering a reliable and efficient e-commerce solution.

Plan Your Content:

1. Homepage:

- Captivating introduction to the e-commerce platform.
- Quick highlights of key features.
- Call-to-action for visitors to explore the solution.

2. About Us:

- Background on the project and its significance.
- Team introduction and expertise.
- Mission and commitment to providing a seamless e-commerce experience.

3. Challenges in E-commerce Development:

- Engaging content on the complexities of e-commerce development.

- Visual representations of challenges.
- Link to detailed insights on the "Problem Definition" page.

4. Our Approach:

- Overview of the project's methodology.
- Emphasis on addressing scalability, performance, and security.
- Bullet points highlighting key strategies.

5. Project Scope:

- Clearly defined objectives and limitations.
- Interactive feature showcase.
- Link to detailed information on the "Project Scope" page.

6. Technical Architecture:

- Visual representation of the technical stack.
- User-friendly explanations of IBM Cloud Foundry's role.
- Infographics for easy comprehension.

7. Security Hub:

- Dedicated section on security measures.
- Infographics on encryption and authentication.
- Compliance badges and certifications.

8. Scalability and Performance:

- Interactive elements illustrating scalability strategies.
- Before-and-after scenarios for performance optimization.
- Direct links to related technical documentation.

9. Inventory Management:

- Showcase of the inventory system.
- User testimonials on improved inventory handling.
- Demo videos or interactive simulations.

10. Order Processing:

- Visual workflow of the order fulfillment process.
- User testimonials on efficient order handling.
- Links to case studies or success stories.

11. Payment Solutions:

- Section on secure payment gateway integration.
- Infographics on payment processing steps.
- Comparison charts for selected payment gateways.

12. Blog/Insights:

- Regularly updated content on e-commerce trends.
- Case studies, best practices, and industry insights.
- Call-to-action for newsletter subscriptions.

13. Contact Us:

- Clear contact information.
- Inquiry form for potential clients.
- Links to social media for engagement.

14. FAQs:

- Commonly asked questions and detailed answers.
- Search functionality for quick access.
- Link to customer support for further assistance.

15. Conclusion:

- Recap of the platform's benefits.

- Call-to-action for businesses seeking a robust e-commerce solution.

Choose A Domain:

Selecting a suitable domain is crucial for your website's identity and accessibility. Remember to check the availability of your chosen domain names and consider factors such as brand identity, memorability, and relevance to your e-commerce platform. Once you've chosen a domain, ensure to register it promptly to secure your online presence.

Hosting Provider:

Selecting a reliable hosting provider is crucial for the performance and availability of your e-commerce website. Selecting a reliable hosting provider is crucial for the performance and availability of your e-commerce website. Here we use IBM Cloud Foundry for Hosting our website.

IBM Cloud:

- Leverage IBM's own cloud services for seamless integration with Cloud Foundry.
- Benefit from the synergy of using the same provider for hosting and Cloud Foundry services.

Consider factors such as scalability, pricing, support, and ease of integration with IBM Cloud Foundry when choosing a hosting provider. Additionally, review each provider's documentation to ensure a smooth setup and deployment process.

Design Your PAGE

Create a simple and appealing design for your webpage. You can code it yourself using HTML, CSS. Certainly! Here's a concise outline for designing your e-commerce webpage:

- **Header:**
  - Include a clear and prominent title for your e-commerce solution.
  - Add a tagline or brief description emphasizing the use of IBM Cloud Foundry.

- **Navigation Bar:**
  - Design a user-friendly navigation menu with links to key sections (Home, About Us, Challenges, etc.).
  - Ensure easy accessibility and responsiveness for various devices.

- **Main Content Sections:**
  - Home:
  - Engage visitors with a welcoming message.
  - Use visuals and a call-to-action to encourage exploration.

  - About Us:
  - Provide information about your team, mission, and commitment.
  - Use visuals, team photos, or infographics to enhance the narrative.

  - Challenges:
  - Highlight the complexities overcome in e-commerce development.
  - Include visuals and links to detailed information.

  - Technical Details:
  - Explain the technical architecture briefly.
  - Use simple visuals or infographics to aid understanding.

  - Security Measures, Scalability, and Performance:
  - Dedicate sections to each aspect with concise explanations.
  - Use icons or visual cues to represent security and scalability.

- Inventory Management, Order Processing, Payment Solutions:
- Provide brief overviews with links to more detailed pages if needed.
- Include visuals to illustrate workflows and processes.

- Blog/Insights:
- Highlight recent blog posts or industry insights.
- Encourage visitors to subscribe for updates.
- Contact Us:
- Include a contact form and relevant contact information.
- Invite inquiries and questions.
- Footer:
- Display copyright information and links to privacy policies.
- Optionally, include social media icons for additional engagement.

- **Design Consistency:**
  - Maintain a consistent color scheme and typography throughout the website.
  - Ensure a clean and organized layout for easy navigation.

- **Mobile Responsiveness:**
  - Design the webpage to be responsive across various devices.
  - Prioritize a user-friendly experience on both desktop and mobile.

- **Visual Elements:**
  - Incorporate high-quality images, icons, and infographics.
  - Use visuals to complement textual content and enhance understanding.
  -

- **Call-to-Action:**
  - Strategically place buttons or links to encourage specific actions (Explore, Contact Us, Learn More).

By focusing on these elements, you can create a well-structured and visually appealing e-commerce webpage that effectively communicates your platform’s strengths and benefits.

Website Development:

Certainly! Here's a brief outline for the website development process:

1. Planning:

   - Define the website's purpose, target audience, and goals.

   - Create a sitemap outlining the structure and navigation.

2. Design:

   - Develop wireframes to visualize the layout and structure.

   - Design the user interface with a focus on user experience (UX) and branding.

3. Development:

   - Set up the development environment and choose a tech stack.

   - Write HTML, CSS, and JavaScript code based on the design.

   - Implement functionality, such as contact forms and interactive elements.

4. Backend Development (if needed):

   - Integrate backend systems for dynamic content or database interaction.

   - Ensure server-side logic supports the website's requirements.

5.Testing:

   - Conduct thorough testing for functionality, responsiveness, and compatibility.

   - Identify and fix bugs or issues for a seamless user experience.

6. Deployment:

   - Choose a hosting provider and deploy the website.

   - Configure domain settings and ensure proper security measures.

7. Optimization:

   - Optimize images and code for faster load times.

   - Ensure responsiveness across various devices and browsers.

8. Monitoring:

   - Implement analytics tools to monitor website traffic and user behavior.

   - Set up alerts for potential issues, ensuring continuous optimization.

9. Content Updates:

   - Establish a content management system (CMS) for easy updates.

   - Regularly update content, blog posts, and other relevant information.

10. Security Measures:

    - Implement SSL certificates for secure data transmission.

    - Regularly update software and plugins to address security vulnerabilities.

11. SEO Optimization:

    - Optimize meta tags, headings, and content for search engines.

    - Submit sitemap to search engines for indexing.

12. User Feedback and Iteration:

    - Gather user feedback and analyze website performance.

- Iterate based on feedback to improve usability and functionality.

By following these steps, you can ensure a systematic and effective website development process for your e-commerce platform on IBM Cloud Foundry.

<u>Domain Setup:</u>

Setting up your domain involves a few key steps:

1. Choose a Domain Name:

- Select a domain name that is relevant, easy to remember, and reflects your brand or business.

2. Check Domain Availability:

- Use domain registration platforms to check if your chosen domain name is available for purchase.

3. Select a Domain Registrar:

- Choose a reputable domain registrar to register and manage your domain. Popular registrars include GoDaddy, Namecheap, or Google Domains.

4. Register the Domain:

- Follow the registrar's instructions to register your chosen domain. Provide necessary information and complete the registration process.

5. Configure DNS Settings:

- Access your domain registrar's control panel to configure Domain Name System (DNS) settings.
- Set up DNS records, including A (IPv4 address), AAAA (IPv6 address), and CNAME (alias) records based on your hosting provider's requirements.

6. Link to Hosting Provider:

- Connect your domain to your hosting provider. This involves updating the nameservers or DNS records with the information provided by your hosting service.

7. SSL Certificate Installation:

- If your website uses HTTPS (recommended for security), install an SSL certificate. Some hosting providers offer free SSL certificates, while others may require purchase or configuration.

8. Verify Domain Ownership (Optional)

- Some registrars and hosting providers may require you to verify domain ownership. Follow any verification steps provided.

9. Test Domain Configuration:

- Ensure that your domain is correctly configured by accessing it through a web browser. Check for any issues with DNS resolution or SSL configuration.

10. Configure Email (Optional):

- If you plan to use custom email addresses associated with your domain (e.g., info@yourdomain.com), configure email settings with your hosting provider.

11. Renew Domain Registration:

- Keep track of your domain registration's expiration date and renew it before it expires to avoid service disruption.

Remember, the specific steps may vary slightly depending on your chosen domain registrar and hosting provider. Always refer to their documentation for accurate and up-to-date instructions.

Publish:

When publishing an ecommerce application on Cloud Foundry, follow these steps:

1. **Code Setup:** Ensure your ecommerce application code is ready and meets the necessary requirements for Cloud Foundry deployment.

2. **Cloud Foundry Setup:** Set up a Cloud Foundry environment. This involves creating an account, installing the Cloud Foundry Command Line Interface (CF CLI), and targeting your Cloud Foundry instance.

3. **Application Manifest:** Create a manifest.yml file for your application. This file includes details like the application name, memory allocation, and other configurations. This is crucial for Cloud Foundry to understand how to deploy your application.

4. **Update and Redeploy:** As you make updates to your ecommerce application, follow the same process to push the changes to Cloud Foundry. This ensures that your application is always running the latest version.

5. **Security Considerations:** Pay attention to security best practices, including securing communication, handling sensitive data properly, and staying informed about any security updates related to your tech stack.

Process

1. Start by creating an account on IBM Cloud if you haven't already.
2. Once you're logged in, navigate to the IBM Cloud Dashboard.
3. Click on "Create Resource" and search for "Static Web Apps" in the catalogue.
4. Select the Static Web Apps service and follow the prompts to create a new instance.
5. Choose a name for your app and select the region where you want it to be hosted.
6. Next, you'll need to upload your static website files. These can include HTML, CSS, JavaScript, and any other assets your blog requires.
7. Once your files are uploaded, configure the build settings and specify the entry point for your website.
8. Finally, deploy your app and wait for it to be deployed to the IBM Cloud.

<u>PROGRAM</u>:

Index.html

```
{% extends 'layout.html' %}

{% block body %}

<!-- Page Content -->

<div class="container">

  <div class="row">

    <!-- /.col-lg-3 -->

    <div class="col-lg-12">

      <div id="carouselExampleIndicators" class="carousel slide my-4" data-
ride="carousel">

        <ol class="carousel-indicators">

          <li data-target="#carouselExampleIndicators" data-slide-to="0"
class="active"></li>

          <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>

          <li data-target="#carouselExampleIndicators" data-slide-to="2"></li>

        </ol>

        <div class="carousel-inner" role="listbox" style="max-height: 300px">

          <div class="carousel-item active">

            <img class="d-block img-fluid" src="/static/image/other/watchwallpaper.jpg"
alt="New York"

              width="1100" height="300">

            <div class="carousel-caption">

              <h3>Los Angeles</h3>

              <p>We had such a great time in LA!</p>

            </div>

          </div>

          <div class="carousel-item">

            <img class="d-block img-fluid" src="/static/image/other/bodyspray.jpg"
alt="New York" width="1100"
```

```
                    height="300">
                  <div class="carousel-caption">
                    <h3>Los Angeles</h3>
                    <p>We had such a great time in LA!</p>
                  </div>
                </div>
                <div class="carousel-item">
                  <img class="d-block img-fluid" src="/static/image/other/t-shirt-wallpaper.jpg"
alt="New York"
                    width="1100" height="300">
                  <div class="carousel-caption">
                    <h3>Los Angeles</h3>
                    <p>We had such a great time in LA!</p>
                  </div>
                </div>
              </div>
              <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button"
data-slide="prev">
                <span class="carousel-control-prev-icon" aria-hidden="true"></span>
                <span class="sr-only">Previous</span>
              </a>
              <a class="carousel-control-next" href="#carouselExampleIndicators" role="button"
data-slide="next">
                <span class="carousel-control-next-icon" aria-hidden="true"></span>
                <span class="sr-only">Next</span>
              </a>
            </div>
            <h3><a href="/tshirt">T-Shirt</a></h3>
            <div class="row">
              {% for product in tshirt %}
              {% if product.category == 'tshirt' %}
              <div class="col-lg-3 col-md-4 col-sm-6">
                <div class="card mb-4">
                  <a href="/{{product.category}}?view={{product.id}}">
                    <div class="card-img-top id_{{product.id}}"
                       style="background:
url('static/image/product/{{product.category}}/{{product.picture}}');
background-repeat:no-repeat; background-size: cover; height:200px;width:100% "></div>
                  </a>
                  <div class="card-body">
                    <h5><a class="card-title" href="javascript:void();">{{
product.pName}}</a></h5>
                    <h4>৳{{product.price}}</h4>
                    <span class="text-warning"></span>
                  </div>
                </div>
                <!-- /.card -->
              </div>
              {% endif %}
               {% endfor %}
            </div>
            <!-- /.row -->
            <div class="border p-4 mb-4" style="height: 350px">
              <img class="w-100 h-100" src="/static/image/other/bodyspray.jpg" alt="New
              York">
            </div>
            <h3><a href="/wallet">Wallet</a></h3>
            <div class="row">
```

```
            {% for product in wallet %}
            <div class="col-lg-3 col-md-4 col-sm-6">
                <div class="card mb-4">
                    <a href="/{{product.category}}?view={{product.id}}">
                        <div class="card-img-top"
                            style="background:
url('static/image/product/{{product.category}}/{{product.picture}}');
background-repeat:no-repeat; background-size: cover; height:200px;width:100% "></div>
</a>
                    <div class="card-body">
                        <h5><a class="card-title" id=""
href="/{{product.category}}?view={{product.id}}">{{ product.pName}}</a></h5>
                        <h4>ъ{{product.price}}</h4>
                        <span class="text-warning"></span>
                    </div>
                </div>
                <!-- /.card -->
            </div>
            {% endfor %}
        </div>
        <!-- /.row -->
        <div class="border p-4 mb-4" style="height: 350px">
            <img class="w-100 h-100" src="/static/image/other/watchwallpaper.jpg" alt="New
York">
        </div>
        <h3><a href="/belt">Belt</a></h3>
        <div class="row">
            {% for product in belt %}
            <div class="col-lg-3 col-md-4 col-sm-6">
                <div class="card mb-4">
```

```
                    <a href="/{{product.category}}?view={{product.id}}">
                        <div class="card-img-top"
                            style="background:
url('static/image/product/{{product.category}}/{{product.picture}}');
background-repeat:no-repeat; background-size: cover; height:200px;width:100% "></div>
                    </a>
                    <div class="card-body">
                        <h5><a class="card-title"
                        href="/{{product.category}}?view={{product.id}}">{{
product.pName}}</a></h5>
                        <h4>ট{{product.price}}</h4>
                        <span class="text-warning"></span>
                    </div>
                </div>
                <!-- /.card -->
            </div>
            {% endfor %}
        </div>
        <div class="border p-4 mb-4" style="height: 350px">
            <img class="w-100 h-100" src="/static/image/other/bodyspray.jpg" alt="New
            York">
        </div>
        <h3><a href="/shoes">Shoes</a></h3>
        <div class="row">
            {% for product in shoes %}
            <div class="col-lg-3 col-md-4 col-sm-6">
                <div class="card mb-4">
                    <a href="/{{product.category}}?view={{product.id}}">
                        <div class="card-img-top"
                            style="background:
```

```
url('static/image/product/{{product.category}}/{{product.picture}}');
background-repeat:no-repeat; background-size: cover; height:200px;width:100% "></div>

                    </a>

                    <div class="card-body">

                      <h5><a class="card-title"
                      href="/{{product.category}}?view={{product.id}}">{{
product.pName}}</a></h5>

                      <h4>৳{{product.price}}</h4>

                      <span class="text-warning"></span>

                    </div>

                  </div>

                  <!-- /.card -->

                </div>

                {% endfor %}

              </div>

              <div class="border p-4 mb-4" style="height: 350px">

                <img class="w-100 h-100" src="/static/image/other/t-shirt-wallpaper.jpg" alt="New
York">

              </div>


        </div>

        <!-- /.col-lg-9 -->

    </div>

    <!-- /.row -->

</div>

<!-- /.container -->

{% endblock %}
```

Profile.html

```
{% extends 'layout.html' %}

{% block body %}
```

```
<!-- Page Content -->
<div class="container">
    <div class="row">
        <div class="col-lg-3">
            <h1 class="my-4">{{session.s_name}}</h1>
            <div class="list-group">
                <a href="/profile?user={{session.uid}}" class="list-group-item">Order List</a>
                <a href="/settings?user={{session.uid}}" class="list-group-item">Settings</a>
            </div>
        </div>
        <!-- /.col-lg-3 -->
        <div class="col-lg-9">
            <div class="card card-default my-4">
                <div class="card-header">
                    <i class="fa fa-bar-chart-o fa-fw"></i>My order list
                </div>
                <!-- /.panel-heading -->
                <div class="card-body">
                    <div class="row">
                        <div class="col-lg-12">
                            {% if result %}
                            <div class="table-responsive">
                                <table class="table table-bordered table-hover table-striped">
                                    <thead>
                                    <tr>
                                        <th>Order No</th>
                                        <th>Product Name</th>
                                        <th>Quantity</th>
                                        <th>Order Place</th>
                                        <th>Mobile</th>
                                        <th>Order Date</th>
                                        <th>Delivery Date</th>
                                    </tr>
                                    </thead>
                                    <tbody>
                                    {% for order in result %}
                                    <tr>
                                        <td>{{order.id}}</td>
                                        <td>{{order.ofname}}</td>
                                        <td>{{order.quantity}}</td>
                                        <td>{{order.oplace}}</td>
                                        <td>{{order.mobile}}</td>
                                        <td>{{order.odate}}</td>
                                        <td>{{order.ddate}}</td>
                                    </tr>
                                    {% endfor%}
                                    </tbody>
                                </table>
                            </div>
                            {% else %}
                            <h3>No orders found</h3>
                            {% endif %}
                            <!-- /.table-responsive -->
                        </div>
                    </div>
                    <!-- /.row -->
```

```
                </div>
                <!-- /.panel-body -->
            </div>
        </div>
        <!-- /.col-lg-9 -->

    </div>
    <!-- /.row -->

</div>
<!-- /.container -->
{% endblock .html
{% extends 'layout.html' %}
{% block body %}
<div class="jumbotron">
    <h2 style="text-align: center; color: #206aaa;">Sign Up Form</h2>
    {% from "includes/_formhelpers.html" import render_field %}
    <form method="POST" action="">
        <div class="form-group">
            {{render_field(form.name, class_="form-control")}}
        </div>
        <div class="form-group">
            {{render_field(form.email, class="form-control")}}
        </div>
        <div class="form-group">
            {{render_field(form.username, class="form-control")}}
        </div>
        <div class="form-group">
        {{render_field(form.password, class="form-control")}}
      </div>
      <div class="form-group">
        {{render_field(form.mobile, class="form-control")}}
      </div>
      <p>
        <input type="submit" class="btn btn-primary" value="Sign Up">
      </p>
   </form>
</div>
{% endblock %}
```

Modelviewproduct.html

```
<!—Modal

<div class="modal modal{{product.id}}" role="document">
   <div class="modal-dialog" style="max-width: 90%; margin: 56px auto;">

      <!—Modal content

      <div class="modal-content">
         <div class="modal-body">
            <div class="row">
               <div class="col-lg-6">


                  <div class="card mb-4">
                     <img class="card-img-top img-fluid"
                        Src="static/image/product/{{product.category}}/{{product.picture}}"
alt="">
                  </div>
            <!-- /.card

          </div>

          <div class=”col-lg-6”>

            <div class=”card card-outline-secondary”>

              <div class=”card-header”>

                Product Details

              </div>

              <div class=”card-body”>

                <div class=”card-body”>

                  <h3 class=”card-title”>{{product.pName}}</h3>

                  <h4>{{product.price}} Taka</h4>

                  <p class=”card-text”>{{product.description}}</p>

                  <span class=”text-warning”>&#9733; &#9733; &#9733; &#9733;
&#9734;</span>

                  4.0 stars

                </div>

                <hr>

                <button type=”button” class=”btn btn-success order_{{product.id}}”

                      Data-dismiss=”modal”>Order Now

                </button>

              </div>

            </div>

            <!-- /.card

          </div>

        </div>

        <h2 class=”mb-4”>Recommended for you:</h2>

        <div class=”row”>
```

```
          <div class=”col-lg-3”>
            <div class=”card mb-4”>
              <img class=”card-img-top img-fluid” src=http://placehold.it/900x400
alt=””>
              <div class=”card-body”>
                <h3 class=”card-title”>Product Name</h3>
                <h4>$24.99</h4>
                <p class=”card-text”>Lorem ipsum dolor sit amet, consectetur sint
aperiam
                  Sequi pariatur praesentium animi perspiciatis molestias iure,
ducimus!</p>
                <span class=”text-warning”>&#9733; &#9733; &#9733; &#9733;
&#9734;</span>
                4.0 stars
              </div>
            </div>
            <!-- /.card
          </div>
        </div>
        <div class=”modal-footer”>
          <button type=”button” class=”btn btn-danger” data-
dismiss=”modal”>Close</button>
        </div>
      </div>
    </div>

  </div>
</div>
```

```
<script>
$(document).ready(function(){
    $(“.id_{{product.id}}”).click(function(){
        $(“.modal{{product.id}}”).modal({backdrop: true});
    });
});

</script>
```

Sbadmin2.js

```
$(function() {
    $(‘#side-menu’).metisMenu();
});
```

Customize.js

```
//Loads the correct sidebar on window load,
//collapses the sidebar on window resize.
// Sets the min-height of #page-wrapper to window size
$(function() {
    $(window).bind(“load resize”, function() {
        Var topOffset = 50;
        Var width = (this.window.innerWidth > 0) ? this.window.innerWidth : this.screen.width;
        If (width < 768) {
            $(‘div.navbar-collapse’).addClass(‘collapse’);
            topOffset = 100; // 2-row-menu
        } else {
            $(‘div.navbar-collapse’).removeClass(‘collapse’);
        }

        Var height = ((this.window.innerHeight > 0) ? this.window.innerHeight :
this.screen.height) – 1;
        Height = height – topOffset;
        If (height < 1) height = 1;
        If (height > topOffset) {
            $(“#page-wrapper”).css(“min-height”, (height) + “px”);
        }
    });

    Var url = window.location;
    // var element = $(‘ul.nav a’).filter(function() {
    //     return this.href == url;
    // }).addClass(‘active’).parent().parent().addClass(‘in’).parent();
    Var element = $(‘ul.nav a’).filter(function() {
        Return this.href == url;
    }).addClass(‘active’).parent();

    While (true) {
        If (element.is(‘li’)) {
            Element = element.parent().addClass(‘in’).parent();
        } else {
            Break;
        }
    }
```

```
});
```

Menushut.sql

```
    -- phpMyAdmin SQL Dump
-- version 4.7.9
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1:3306
-- Generation Time: Oct 01, 2018 at 04:03 AM
-- Server version: 5.7.21
-- PHP Version: 5.6.35


SET SQL_MODE = “NO_AUTO_VALUE_ON_ZERO”;
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = “+00:00”;
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
*/;
/*!40101 SET NAMES utf8mb4 */;


--
-- Database: `shoptubedb`
--


            --------------------------------------------------------
```

```
--

-- Table structure for table `admin`

--


DROP TABLE IF EXISTS `admin`;

CREATE TABLE IF NOT EXISTS `admin` (

  `id` int(11) NOT NULL AUTO_INCREMENT,

  `firstName` varchar(125) NOT NULL,

  `lastName` varchar(125) NOT NULL,

  `email` varchar(100) NOT NULL,

  `mobile` varchar(25) NOT NULL,

  `address` text NOT NULL,

  `password` varchar(100) NOT NULL,

  `type` varchar(20) NOT NULL,

  `confirmCode` varchar(10) NOT NULL,

  PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;


--

-- Dumping data for table `admin`

--


INSERT INTO `admin` (`id`, `firstName`, `lastName`, `email`, `mobile`, `address`,
`password`, `type`, `confirmCode`) VALUES

(4, ‘Nur’, ‘Mohsin’, ‘mohsin@gmail.com’, ‘01677876551’, ‘Dhaka’,
‘$5$rounds=535000$WOAOMdgoK2JpZLY5$RFH9BZQCB3NEvG4R/FofxxJL/PUaeZm7T
6G9P3PRg05’, ‘manager’, ‘0’);
```

```
                    -----------------------------------------------------

--

-- Table structure for table `orders`

--


DROP TABLE IF EXISTS `orders`;

CREATE TABLE IF NOT EXISTS `orders` (

  `id` int(11) NOT NULL AUTO_INCREMENT,

  `uid` int(11) DEFAULT NULL,

  `ofname` text NOT NULL,

  `pid` int(11) NOT NULL,

  `quantity` int(11) NOT NULL,

  `oplace` text NOT NULL,

  `mobile` varchar(15) NOT NULL,

  `dstatus` varchar(10) NOT NULL DEFAULT ‘no’,

  `odate` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,

  `ddate` date DEFAULT NULL,

  PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=latin1;


--

-- Dumping data for table `orders`

--
```

```
INSERT INTO `orders` (`id`, `uid`, `ofname`, `pid`, `quantity`, `oplace`, `mobile`, `dstatus`,
`odate`, `ddate`) VALUES

(1, NULL, ‘Kashmiri Chador’, 1, 2, ‘Khilkhet, Dhaka’, ‘01609876543’, ‘no’, ‘2018-09-21
13:05:07’, NULL),

(2, NULL, ‘Nur Mohsin’, 1, 3, ‘Khilkhet, Dhaka’, ‘01609876543’, ‘no’, ‘2018-09-21
13:08:55’, NULL),

(3, NULL, ‘Nur Mohsin’, 2, 4, ‘Dhaka’, ‘09876543123’, ‘no’, ‘2018-09-21 13:13:04’,
NULL),

(4, NULL, ‘Nur Mohsin’, 4, 1, ‘Manikganj’, ‘798345’, ‘no’, ‘2018-09-21 13:18:47’, NULL),

(5, NULL, ‘Nur Mohsin’, 9, 4, ‘Dhaka, Bangladesh’, ‘01609876543’, ‘no’, ‘2018-09-22
02:01:02’, NULL),

(6, NULL, ‘Nur Mohsin’, 2, 1, ‘Manikganj’, ‘01609876543’, ‘no’, ‘2018-09-22 02:09:29’,
NULL),

(7, 9, ‘Nur Mohsin’, 2, 1, ‘Manikganj’, ‘01609876543’, ‘no’, ‘2018-09-22 02:10:46’, NULL),

(8, 9, ‘Nur Mohsin’, 1, 1, ‘Manikganj’, ‘0994’, ‘no’, ‘2018-09-22 03:04:13’, NULL),

(9, 9, ‘Kashmiri Chador’, 12, 4, ‘Dhaka’, ‘01609876543’, ‘no’, ‘2018-09-22 03:21:14’, ‘2018-
09-29’),

(10, 9, ‘Chador’, 13, 1, ‘Dhaka’, ‘01609876543’, ‘no’, ‘2018-09-22 03:22:05’, ‘2018-09-29’);


            --------------------------------------------------------

--

-- Table structure for table `products`

--


DROP TABLE IF EXISTS `products`;

CREATE TABLE IF NOT EXISTS `products` (

  `id` int(11) NOT NULL AUTO_INCREMENT,

  `pName` varchar(100) NOT NULL,

  `price` int(11) NOT NULL,
  `description` text NOT NULL,

  `available` int(11) NOT NULL,

  `category` varchar(100) NOT NULL,

  `item` varchar(100) NOT NULL,

  `pCode` varchar(20) NOT NULL,

  `picture` text NOT NULL,

  `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,

  PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO_INCREMENT=22 DEFAULT CHARSET=latin1;


--

-- Dumping data for table `products`

--


INSERT INTO `products` (`id`, `pName`, `price`, `description`, `available`, `category`, `item`,
`pCode`, `picture`, `date`) VALUES

(1, ‘T-Shirt’, 120, ‘T-Shirt’, 4, ‘tshirt’, ‘t-shirt’, ‘t-007’, ‘MSTS14738.jpg’, ‘2018-09-20
07:10:40’),

(2, ‘Baborry wallet’, 6000,
‘Baborry-Double-Zipper-Coin-Bag-RFID-Blocking-Men-Wallets-New-Brand-PU-Leather-
Wa llet-Money-Purses’, 3, ‘wallet’, ‘wallet’, ‘w-004’, ‘IMG_1212.jpg’, ‘2018-09-20
07:40:28’),

(3, ‘Loafer Shoes’, 2000, ‘Loafer black shoes’, 8, ‘shoes’, ‘shoes’, ‘s-001’, ‘8544789_5_.jpg’,
‘2018-09-20 08:33:57’),

(4, ‘Artificial Belt’, 1200, ‘Black artificial belt’, 9, ‘belt’, ‘belt’, ‘b-001’, ‘0283BLT.jpg’,
‘2018-09-20 08:35:44’),

(5, ‘Polo T-shirt’, 500, ‘Polo t-shirt’, 10, ‘tshirt’, ‘t-shirt’, ‘s-002’, ‘lp00-2.jpg’, ‘2018-09-20
08:40:06’),
(6, ‘T-shirt’, 300, ‘Polo colorful t-shirt’, 12, ‘tshirt’, ‘t-shirt’, ‘t-003’, ‘yellow_2_.jpg’, ‘2018-
09-20 08:41:18’),

(7, ‘Tshirt’, 200, ‘Design t-shirt’, 10, ‘tshirt’, ‘t-shirt’, ‘t-004’, ‘MSTSV14042.jpg’,
‘2018-09-20 08:42:11’),

(8, ‘T-shirt’, 200, ‘Color t-shirt’, 20, ‘tshirt’, ‘t-shirt’, ‘t-005’, ‘MSTS14759.jpg’, ‘2018-09-20
08:45:39’),

(9, ‘Men\’s Tshirt’, 500, ‘Colorful men\’s t-shirt’, 20, ‘tshirt’, ‘t-shirt’, ‘t-006’,
‘MSTSV14046.jpg’, ‘2018-09-20 08:57:07’),

(10, ‘Sports tshirt’, 1000, ‘Real madrid t-shirt’, 5, ‘tshirt’, ‘t-shirt’, ‘t-007’,
‘MSTSV14039.jpg’, ‘2018-09-20 08:58:38’),

(12, ‘T-shirt’, 300, ‘Design t-shirt’, 10, ‘tshirt’, ‘t-shirt’, ‘t-010’, ‘MSTSV14049.jpg’, ‘2018-
09-20 09:02:04’),

(13, ‘Leather Shoes’, 2000, ‘Best leather shoes’, 10, ‘shoes’, ‘shoes’, ‘s-002’,
‘8546789_5_.jpg’, ‘2018-09-21 10:39:32’),

(14, ‘Belt’, 2000, ‘Nice belt’, 20, ‘belt’, ‘belt’, ‘b-003’, ‘gbdl18_1.png’, ‘2018-10-01
03:47:08’),

(15, ‘Belt’, 300, ‘Nice one belt’, 20, ‘belt’, ‘belt’, ‘b-004’, ‘101010_1_.jpg’, ‘2018-10-01
03:48:09’),

(16, ‘Mens Belt’, 300, ‘Mens belt’, 15, ‘belt’, ‘belt’, ‘b-005’, ‘image4_2.jpg’, ‘2018-10-01
03:49:08’),

(17, ‘Leather Wallet’, 100, ‘Leather wallet’, 10, ‘wallet’, ‘wallet’, ‘w-005’,
‘Baborry-Double-Zipper-Coin-Bag-RFID-Blocking-Men-Wallets-New-Brand-PU-Leather-
Wa llet-Money-Purses.jpg_640x640.jpg’, ‘2018-10-01 03:51:52’),

(18, ‘Wallet’, 300, ‘Wallet’, 20, ‘wallet’, ‘wallet’, ‘w-007’, ‘1881_G.jpg’, ‘2018-10-01
03:52:43’),

(19, ‘Black walllet’, 300, ‘Black mens wallet’, 20, ‘wallet’, ‘wallet’, ‘w-009’,
‘image5_1_2.jpg’, ‘2018-10-01 03:53:37’),

(20, ‘Men\’s Shoes’, 1200, ‘Men\’s shoes’, 23, ‘shoes’, ‘shoes’, ‘s-003’, ‘IMG_2429.jpg’,
‘2018-10-01 03:56:41’),

(21, ‘Shoes’, 2000, ‘Formal Shoes’, 12, ‘shoes’, ‘shoes’, ‘s-004’, ‘G51A7054.jpg’,
‘2018-10-01 03:57:24’);
```

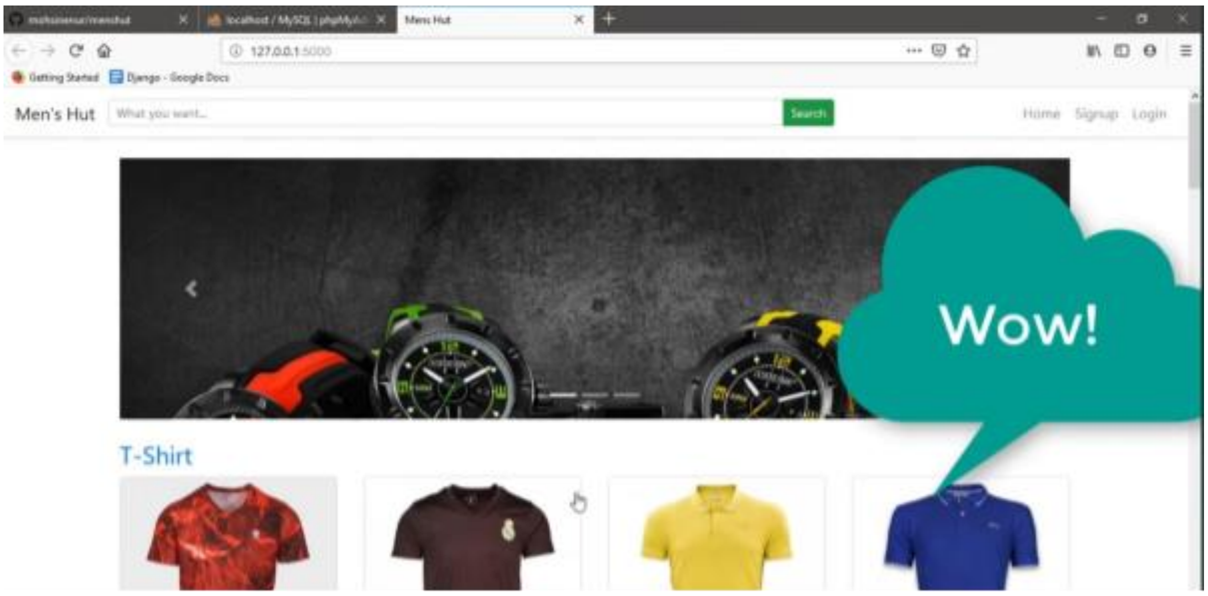
```
                    --------------------------------------------------------

--
-- Table structure for table `product_level`
--

DROP TABLE IF EXISTS `product_level`;
CREATE TABLE IF NOT EXISTS `product_level` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `product_id` int(11) NOT NULL,
  `v_shape` varchar(10) NOT NULL DEFAULT ‘no’,
  `polo` varchar(10) NOT NULL DEFAULT ‘no’,
  `clean_text` varchar(10) NOT NULL DEFAULT ‘no’,
  `design` varchar(10) NOT NULL DEFAULT ‘no’,
  `chain` varchar(10) NOT NULL DEFAULT ‘no’,
  `leather` varchar(10) NOT NULL DEFAULT ‘no’,
  `hook` varchar(10) NOT NULL DEFAULT ‘no’,
  `color` varchar(10) NOT NULL DEFAULT ‘no’,
  `formal` varchar(10) NOT NULL DEFAULT ‘no’,
  `converse` varchar(10) NOT NULL DEFAULT ‘no’,
  `loafer` varchar(10) NOT NULL DEFAULT ‘no’,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=22 DEFAULT CHARSET=latin1;

--
-- Dumping data for table `product_level`
```

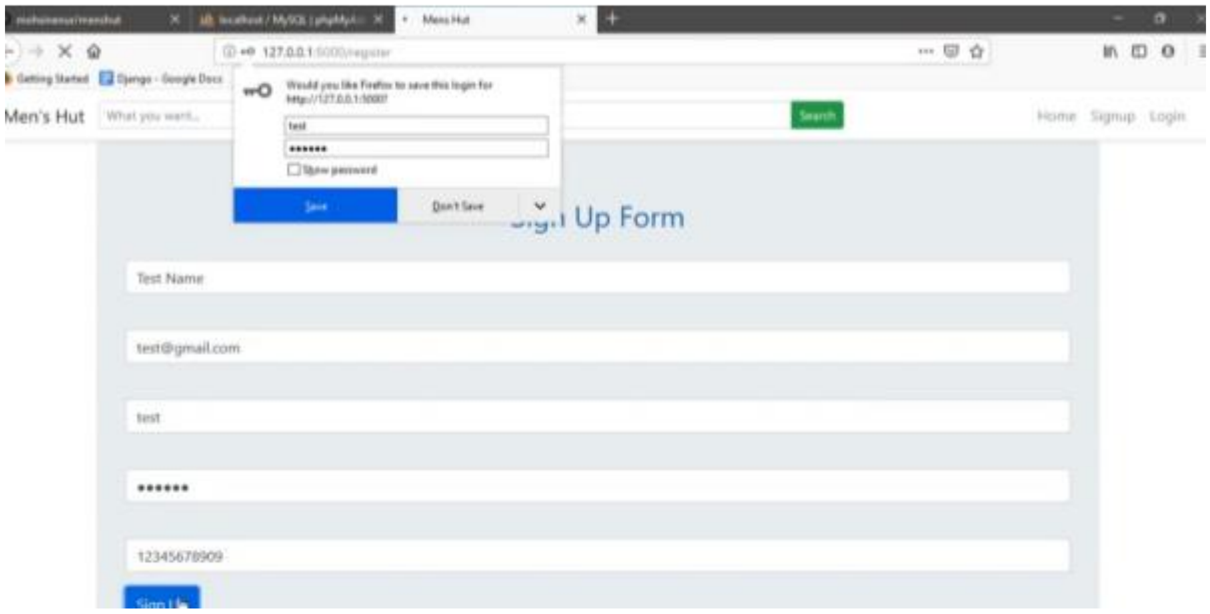
```
--

INSERT INTO `product_level` (`id`, `product_id`, `v_shape`, `polo`, `clean_text`, `design`,
`chain`, `leather`, `hook`, `color`, `formal`, `converse`, `loafer`) VALUES

(1, 1, ‘no’, ‘no’, ‘yes’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’),

(2, 2, ‘no’, ‘no’, ‘no’, ‘no’, ‘yes’, ‘yes’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’),

(3, 3, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘yes’, ‘no’, ‘no’, ‘no’, ‘no’, ‘yes’),

(4, 4, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘yes’, ‘yes’, ‘no’, ‘no’, ‘no’, ‘no’),

(5, 5, ‘no’, ‘yes’, ‘yes’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’),

(6, 6, ‘no’, ‘yes’, ‘yes’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’),

(7, 7, ‘yes’, ‘no’, ‘no’, ‘yes’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’),

(8, 8, ‘no’, ‘no’, ‘yes’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’),

(9, 9, ‘yes’, ‘no’, ‘no’, ‘yes’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’),

(10, 10, ‘yes’, ‘no’, ‘yes’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’),

(14, 14, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘yes’, ‘yes’, ‘no’, ‘no’, ‘no’, ‘no’),

(12, 12, ‘yes’, ‘no’, ‘no’, ‘yes’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’),

(13, 13, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘yes’, ‘no’, ‘no’, ‘no’, ‘no’, ‘yes’),

(15, 15, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘yes’, ‘no’, ‘yes’, ‘no’, ‘no’, ‘no’),

(16, 16, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘yes’, ‘yes’, ‘yes’, ‘no’, ‘no’, ‘no’),

(17, 17, ‘no’, ‘no’, ‘no’, ‘no’, ‘yes’, ‘yes’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’),

(18, 18, ‘no’, ‘no’, ‘no’, ‘no’, ‘yes’, ‘yes’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’),

(19, 19, ‘no’, ‘no’, ‘no’, ‘yes’, ‘yes’, ‘yes’, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’),

(20, 20, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘yes’, ‘no’, ‘no’, ‘no’, ‘yes’, ‘no’),

(21, 21, ‘no’, ‘no’, ‘no’, ‘no’, ‘no’, ‘yes’, ‘no’, ‘no’, ‘yes’, ‘no’, ‘no’);
```

```
--

-- Table structure for table `product_view`

--


DROP TABLE IF EXISTS `product_view`;

CREATE TABLE IF NOT EXISTS `product_view` (

  `id` int(11) NOT NULL AUTO_INCREMENT,

  `user_id` int(11) NOT NULL,

  `product_id` int(11) NOT NULL,

  `date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,

  PRIMARY KEY (`id`)

) ENGINE=MyISAM AUTO_INCREMENT=9 DEFAULT CHARSET=latin1;


--

-- Dumping data for table `product_view`

--


INSERT INTO `product_view` (`id`, `user_id`, `product_id`, `date`) VALUES

(1, 9, 9, ‘2018-09-22 02:19:30’),

(2, 9, 7, ‘2018-09-27 02:47:43’),

(3, 9, 12, ‘2018-09-22 03:20:59’),

(4, 9, 10, ‘2018-09-29 03:07:11’),

(5, 9, 5, ‘2018-09-22 03:19:19’),


(6, 9, 8, ‘2018-09-21 15:57:50’),

(7, 9, 6, ‘2018-09-22 02:12:54’),
(8, 9, 1, ‘2018-09-22 03:03:36’);

                -----------------------------------------------------

--
-- Table structure for table `users`
--

DROP TABLE IF EXISTS `users`;
CREATE TABLE IF NOT EXISTS `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(50) NOT NULL,
  `email` varchar(50) NOT NULL,
  `username` varchar(25) NOT NULL,
  `password` varchar(100) NOT NULL,
  `mobile` varchar(20) NOT NULL,
  `reg_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `online` varchar(1) NOT NULL DEFAULT ‘0’,
  `activation` varchar(3) NOT NULL DEFAULT ‘yes’,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=16 DEFAULT CHARSET=latin1;

-- Dumping data for table `users`
INSERT INTO `users` (`id`, `name`, `email`, `username`, `password`, `mobile`, `reg_time`,
`online`, `activation`) VALUES
(12, ‘Mukul’, ‘mukul@gmail.com’, ‘mukul’,
‘$5$rounds=535000$6PJhbzFlfJbcQbza$FbrPa3qqk1RJ5MSffRLO6LrQJXbgO8SudFuBpNf.
wR7’, ‘’, ‘2018-07-23 14:09:14’, ‘0’, ‘yes’),

(9, ‘Nur Mohsin’, ‘mohsin@gmail.com’, ‘mohsin’,
‘$5$rounds=535000$EnLkwqfGWGcWklRL$q9PbYw/TVXSzs.QpgUouZ3.6BzaPG2eLHkT
yv.Qx80D’, ‘123456789022’, ‘2018-07-21 06:47:57’, ‘1’, ‘yes’),

(14, ‘Nur Mohsin’, ‘khan@gmail.com’, ‘khan’,
‘$5$rounds=535000$wLKTQexvPQHueUsK$aFrFUXBHjrrAH61EFiYgj8cZECaaz8y6S5XS
/zkkHw9’, ‘’, ‘2018-09-07 09:02:35’, ‘0’, ‘yes’),

(13, ‘Robin’, ‘robin@gmail.com’, ‘robin’,
‘$5$rounds=535000$uiZc/VCwwa3XCTTe$Ec.JOjy4GkjpAXHtAvGt6pSc6KszajHgcyZy8v
6
Ivk1’, ‘’, ‘2018-07-26 12:36:57’, ‘0’, ‘yes’),

(15, ‘Sujon’, ‘sujon@yahoo.com’, ‘sujons’,
‘$5$rounds=535000$aGykDT1yrocgTaDt$p2dDAMDz9g3N6o/Jj7QJY9B6NnMlUot.DCq/L
OsCS13’, ‘89345793753’, ‘2018-09-08 13:58:36’, ‘0’, ‘yes’);

COMMIT;


/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

OUTPUT:

Home.html




Would you like Firefox to save this login for http://127.0.0.1:5000?

test

••••••

Show password

Save | Don't Save

Sign Up Form

Test Name

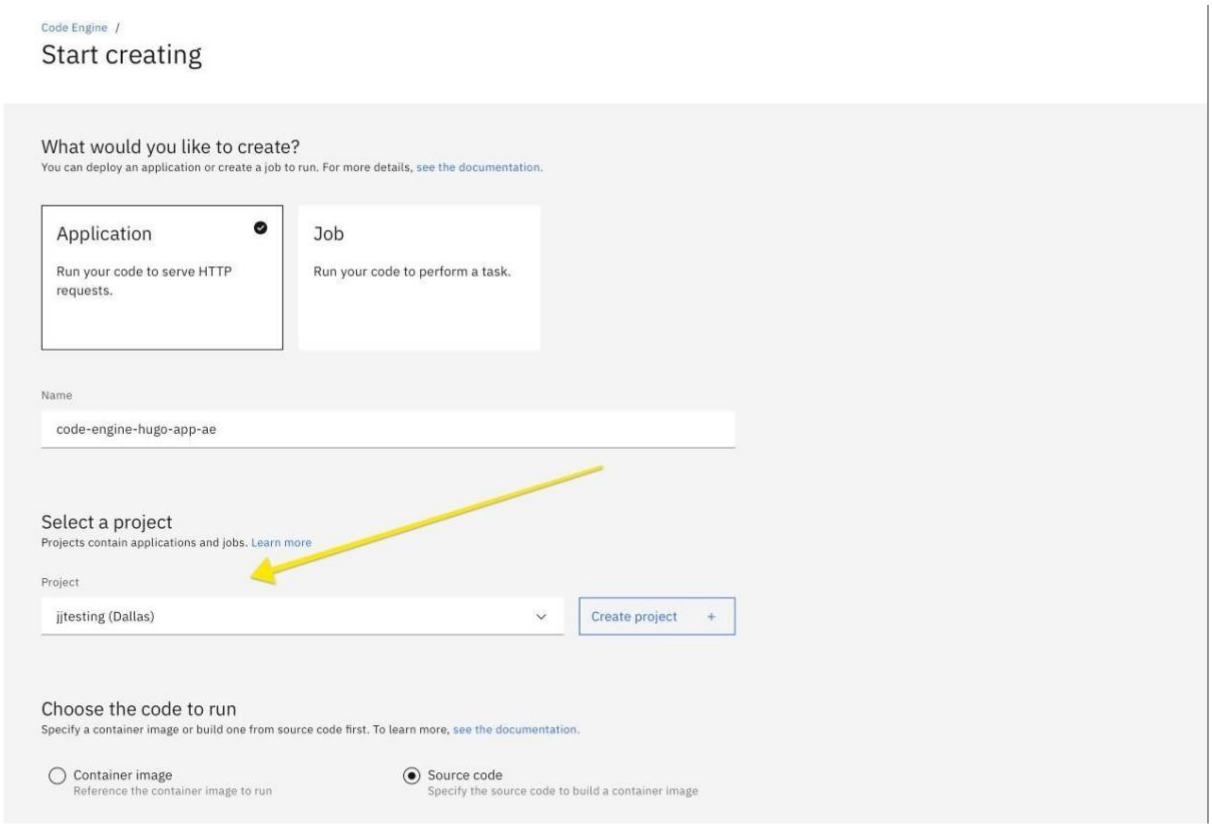
test@gmail.com

test

••••••

12345678909

Sign Up



Code Engine /

Start creating

What would you like to create?

You can deploy an application or create a job to run. For more details, see the documentation.

Application ✓
Run your code to serve HTTP requests.

Job
Run your code to perform a task.

Name: code-engine-hugo-app-ae

Select a project

Projects contain applications and jobs. Learn more

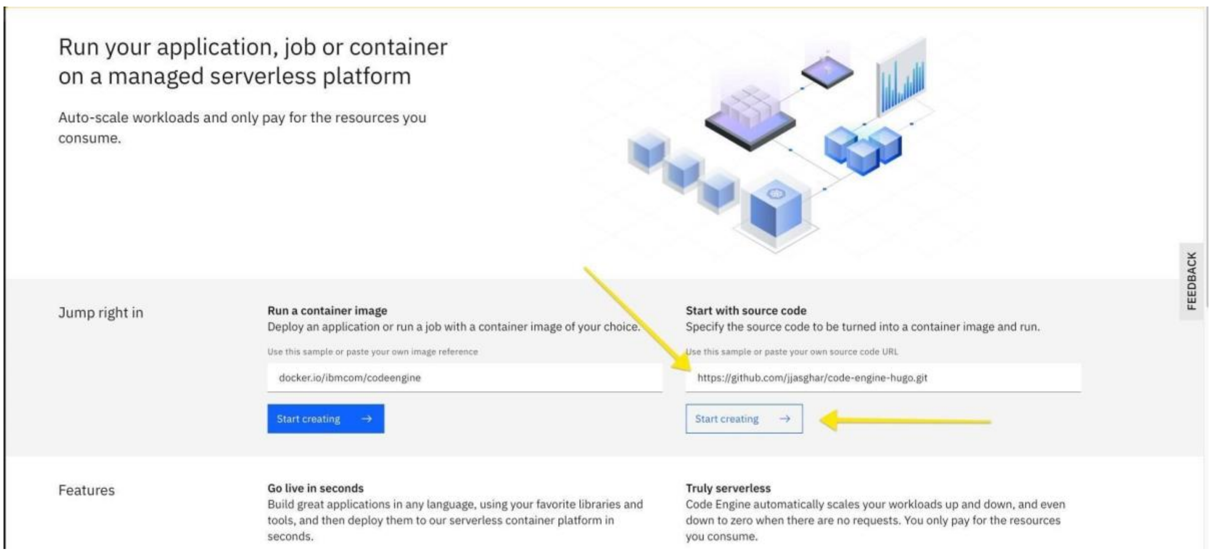
Project: jjtesting (Dallas)

Choose the code to run

Specify a container image or build one from source code first. To learn more, see the documentation.

Container image — Reference the container image to run

[✓ Source code] — Specify the source code to build a container image



Run your application, job or container on a managed serverless platform

Auto-scale workloads and only pay for the resources you consume.

Jump right in

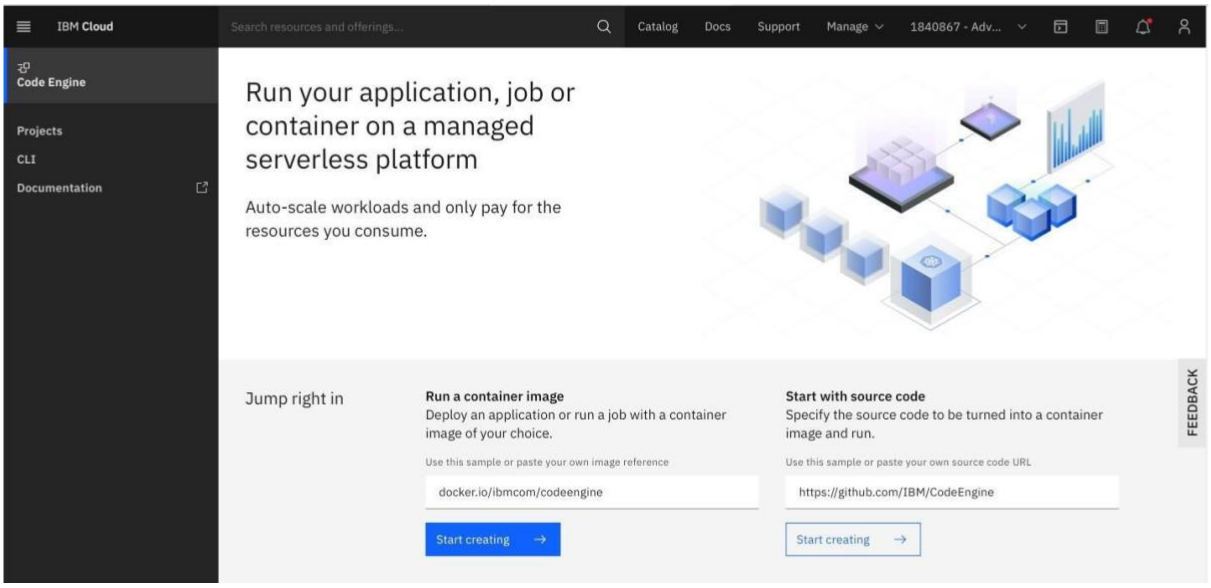
Run a container image
Deploy an application or run a job with a container image of your choice.
Use this sample or paste your own image reference
docker.io/ibmcom/codeengine

Start with source code
Specify the source code to be turned into a container image and run.
Use this sample or paste your own source code URL
https://github.com/jjasghar/code-engine-hugo.git

Features

Go live in seconds
Build great applications in any language, using your favorite libraries and tools, and then deploy them to our serverless container platform in seconds.

Truly serverless
Code Engine automatically scales your workloads up and down, and even down to zero when there are no requests. You only pay for the resources you consume.



Run your application, job or container on a managed serverless platform

Auto-scale workloads and only pay for the resources you consume.

Jump right in

Run a container image
Deploy an application or run a job with a container image of your choice.

Use this sample or paste your own image reference

docker.io/ibmcom/codeengine

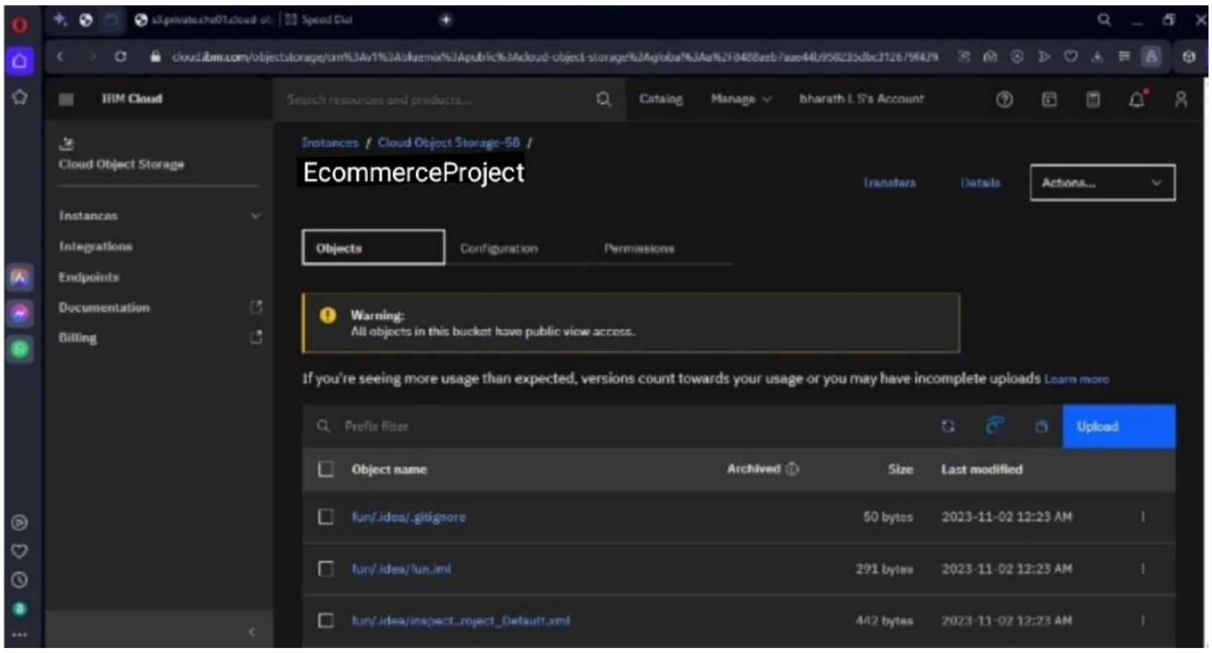
Start creating →

Start with source code
Specify the source code to be turned into a container image and run.

Use this sample or paste your own source code URL

https://github.com/IBM/CodeEngine

Start creating →



EcommerceProject

Warning:
All objects in this bucket have public view access.

If you're seeing more usage than expected, versions count towards your usage or you may have incomplete uploads Learn more

Object name	Archived	Size	Last modified
fun/.idea/.gitignore		50 bytes	2023-11-02 12:23 AM
fun/.idea/fun.iml		291 bytes	2023-11-02 12:23 AM
fun/.idea/inspect...roject_Default.xml		442 bytes	2023-11-02 12:23 AM

CONCLUSION:

In conclusion, successfully publishing an ecommerce application on Cloud Foundry involves thorough preparation, from code setup to security considerations. By following a structured process of code packaging, manifest creation, deployment, service binding, scaling, monitoring, and maintaining security, you can ensure a reliable and efficient deployment on the Cloud Foundry platform. Regular updates and adherence to best practices contribute to a robust and resilient ecommerce application in a Cloud Foundry environment. If you have further questions or need assistance with specific aspects, feel free to ask for more details.