

# E-COMMERCE APPLICATION ON IBM CLOUD FOUNDRY

## PHASE-3: DEVELOPMENT PART-1

Begin building the artisanal e-commerce platform on IBM Cloud Foundry.

To begin building the artisanal e-commerce platform, we'll start by setting up the basic structure, creating a product catalog, and implementing a simple shopping cart. We'll use **FLASK**, **IBM-DB2** for the **backend** and **HTML**, **CSS**, **JAVASCRIPT** for the **frontend**, and a basic in-memory data structure for the product catalog with **IBM CLOUD OBJECT STORAGE** using **DOCKER** and **KUBERNETES** in containers.

Here's a step-by-step guide:

### 1. TO EXECUTE THE FRONTEND PART – HTML IS USED

[https://github.com/SUDH003/CLOUD\\_SKILL-UP/blob/main/index.html](https://github.com/SUDH003/CLOUD_SKILL-UP/blob/main/index.html)

- Save the file with **.html** extension

#### CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Artisanal E-Commerce</title>
<link rel="stylesheet" href="styles.css">
</head>
HTML
<body>
<header>
<h1>Welcome to Artisanal Marketplace</h1>
<nav>
<ul>
<li><a href="#home">Home</a></li>
<li><a href="#products">Products</a></li>
<li><a href="#cart">Cart</a></li>
</ul>
</nav>
</header>

<section id="home">
<h2>Discover Unique Handcrafted Items</h2>
<p>Explore our collection of artisan-made products.</p>
</section>

<section id="products">
<h2>Featured Products</h2>
<div class="product-gallery">
<!-- Display product images, names, and prices -->
```

```

<div class="product">

<h3>Artistic Jewelry</h3>
<p>$50.00</p>
<button>Add to Cart</button>
</div>
<!-- Add more products -->
</div>
</section>

<section id="cart">
<h2>Your Shopping Cart</h2>
<!-- Display cart items and total -->
<div class="cart-items">
<!-- Display cart items -->
</div>
<p>Total: ${Total Amount}</p>
<button>Proceed to Checkout</button>
</section>

<footer>
<p>&copy; 2023 Artisanal Marketplace. All rights reserved.</p>
</footer>
</body>
</html>

```

## 2. INSTALLING FLASK TO PERFORM BACKEND

*pip install FLASK (at command prompt)*

```

Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\velus>pip install Flask
Collecting Flask
  Downloading flask-3.0.0-py3-none-any.whl (99 kB)
    99.7/99.7 kB 1.9 MB/s eta 0:00:00
Collecting Jinja2>=3.1.2
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
    133.1/133.1 kB 2.0 MB/s eta 0:00:00
Collecting click>=8.1.3
  Downloading click-8.1.7-py3-none-any.whl (97 kB)
    97.9/97.9 kB 1.9 MB/s eta 0:00:00
Collecting blinker>=1.6.2
  Downloading blinker-1.6.3-py3-none-any.whl (13 kB)
Collecting Werkzeug>=3.0.0
  Downloading werkzeug-3.0.0-py3-none-any.whl (226 kB)
    226.6/226.6 kB 3.4 MB/s eta 0:00:00
Collecting itsdangerous>=2.1.2
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting colorama
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.3-cp310-cp310-win_amd64.whl (17 kB)
Installing collected packages: MarkupSafe, itsdangerous, colorama, blinker, Werkzeug, Jinja2, click, Flask
WARNING: The script flask.exe is installed in 'C:\Users\velus\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\LocalCache\local-p
ackages\Python310\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed Flask-3.0.0 Jinja2-3.1.2 MarkupSafe-2.1.3 Werkzeug-3.0.0 blinker-1.6.3 click-8.1.7 colorama-0.4.6 itsdangerous-2.1.2

[notice] A new release of pip is available: 23.0.1 -> 23.3.1
[notice] To update, run: C:\Users\velus\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\python.exe -m pip install --u
pgrade pip

C:\Users\velus>

```

### 3. RENDERING THE REQUIRED HTML FILE IN FLASK

- Save the file with **.py** extension

```
from flask import Flask, render_template
```

```
app = Flask(__name__)
```

```
@app.route('/index')
```

```
def home():
```

```
    return render_template('index.html')
```

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```

### 4. TO CREATE A LOGIN PAGE

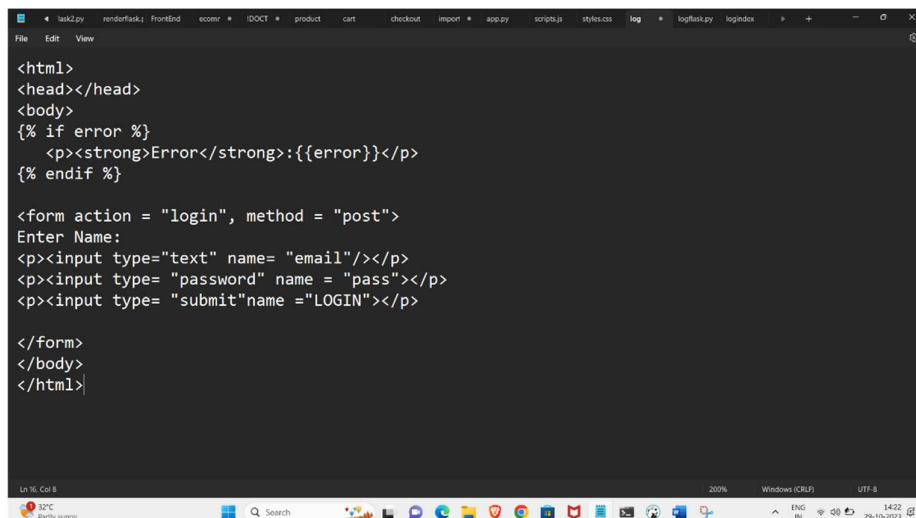
#### 4.1 . HTML code for Login Page

[https://github.com/SUDH003/CLOUD\\_SKILL-UP/blob/main/log.html](https://github.com/SUDH003/CLOUD_SKILL-UP/blob/main/log.html)

```
<html>
<head></head>
<body>
{% if error %}
    <p><strong>Error</strong>:{{error}}</p>
{% endif %}

<form action = "login", method = "post">
Enter Name:
<p><input type="text" name= "email"/></p>
<p><input type= "password" name = "pass"></p>
<p><input type= "submit"name ="LOGIN"></p>

</form>
</body>
</html>
```



## 4.2. HTML code to connect Login Page with Main Ecommerce Page

[https://github.com/SUDH003/CLOUD\\_SKILL-UP/blob/main/logindex.html](https://github.com/SUDH003/CLOUD_SKILL-UP/blob/main/logindex.html)

```
<html>

<body>

    {% with message = mget_flashed_message() %}

        {% if message %}

            {%for message in messages %}

                <p>{{message}}</p>

            {%endfor%}

        {%endif%}

    {%endwidth%}


    <h1>welcome people<h1>

    <a href = "{{url_for{"/login}}}">LOGIN</a>

</body>

</html>
```

## 4.3. Rendering the LOGIN PAGE code in FLASK using python

[https://github.com/SUDH003/CLOUD\\_SKILL-UP/blob/main/logflask.py](https://github.com/SUDH003/CLOUD_SKILL-UP/blob/main/logflask.py)

```
from flask import*

app = Flask(__name__)


@app.route("/index")

def home():

    return render_template(logindex.html)


@app.route("/login")

def login():

    error = None;

    if request.method == "POST":

        if request.form["pass"] != "AAA":
```

```
        error = "Invalid user"

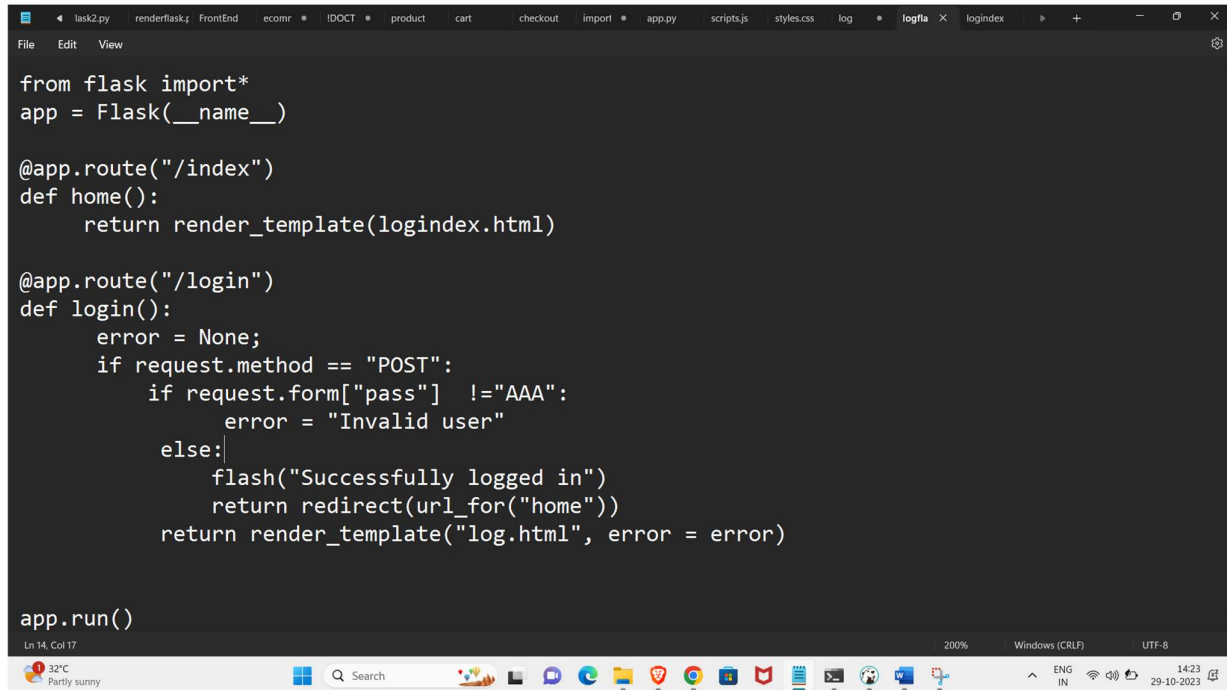
    else:

        flash("Successfully logged in")

        return redirect(url_for("home"))

    return render_template("log.html", error = error)

app.run()
```

A screenshot of a code editor window with a dark theme. The editor shows Python code for a Flask application. The code includes imports for Flask, a route for the home page, and a route for the login page. The login route checks the password and either shows an error or redirects to the home page. The code is saved as 'lask2.py'. The editor's status bar at the bottom shows 'Ln 14, Col 17', '200%', 'Windows (CRLF)', and 'UTF-8'. The Windows taskbar is visible at the bottom of the screen.

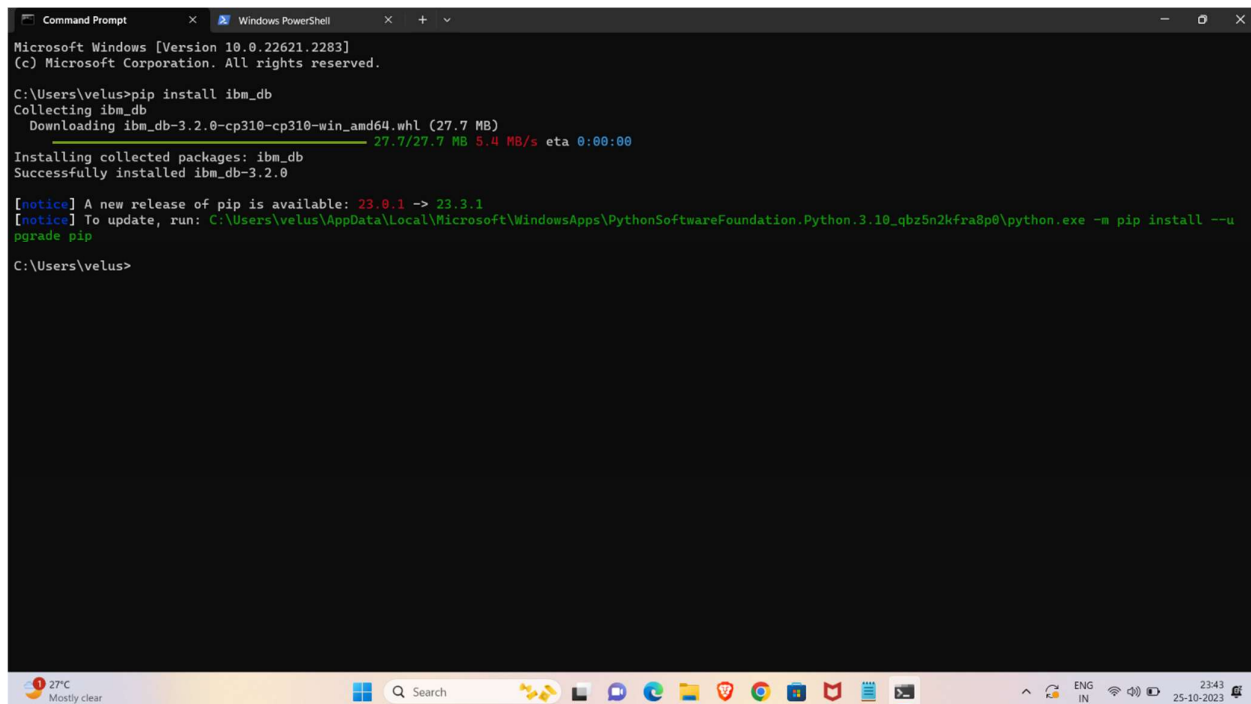
```
from flask import*
app = Flask(__name__)

@app.route("/index")
def home():
    return render_template(logindex.html)

@app.route("/login")
def login():
    error = None;
    if request.method == "POST":
        if request.form["pass"] != "AAA":
            error = "Invalid user"
        else:
            flash("Successfully logged in")
            return redirect(url_for("home"))
    return render_template("log.html", error = error)

app.run()
```

#### 4.4. To make Collected to be stored and sorted in table IBM DB2 in Imported

A screenshot of a Windows Command Prompt window. The window title is 'Command Prompt'. The text shows the command 'pip install ibm\_db' being executed. It displays the progress of downloading the package and the successful installation of 'ibm\_db-3.2.0'. There are also notices about a new release of pip and instructions to update it. The command prompt is running in the directory 'C:\Users\velus>'. The Windows taskbar is visible at the bottom of the screen.

```
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\velus>pip install ibm_db
Collecting ibm_db
  Downloading ibm_db-3.2.0-cp310-cp310-win_amd64.whl (27.7 MB)
    Installing collected packages: ibm_db
    Successfully installed ibm_db-3.2.0

[notice] A new release of pip is available: 23.0.1 -> 23.3.1
[notice] To update, run: C:\Users\velus\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.10_qbz5n2kfra8p0\python.exe -m pip install --u
pgrade pip

C:\Users\velus>
```

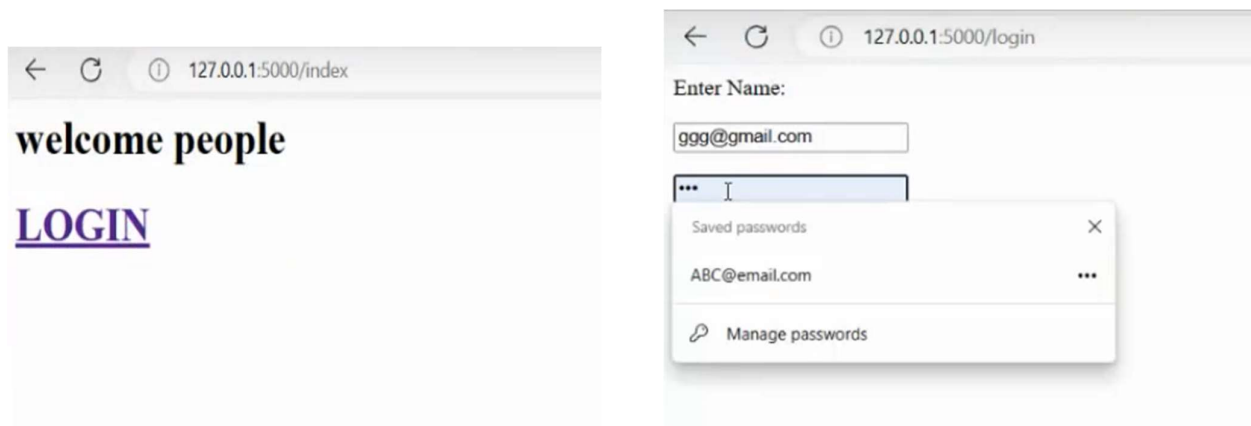
## 4.5. Now Changing Directory executed to deploy the LOGIN PAGE

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\velus> cd C:\Users\velus\Desktop\FLASK
PS C:\Users\velus\Desktop\FLASK> python "renderflask.py"
* Serving Flask app 'renderflask'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 119-636-866
* Detected change in 'C:\Users\velus\Desktop\FLASK\logflask.py', reloading
* Restarting with stat
* Debugger is active!
* Debugger PIN: 119-636-866
* Detected change in 'C:\Users\velus\Desktop\FLASK\ecomm.py', reloading
* Restarting with stat
* Debugger is active!
* Debugger PIN: 119-636-866
```

## 4.6. Executed LOGIN PAGE for the E-Commerce Application



THEREFORE, LOGIN PAGE HAS BEEN DEPLOYED SUCCESSFULLY