# E-commerce Application on IBM Cloud Foundry

## Phase 4: Development Part 2

In this part you will continue building your project.

- Continue building the e-commerce platform by implementing user authentication, shopping cart, and checkout functionality.
- Implement user registration and authentication features using a backend server (e.g. Node.js, Python).
- Implement shopping cart functionality, calculate the total, and enable a smooth checkout process

**Continue building the e-commerce platform by implementing user authentication, shopping cart, and checkout functionality.**

## USER AUTHENTICATION:

### LOGIN.HTML

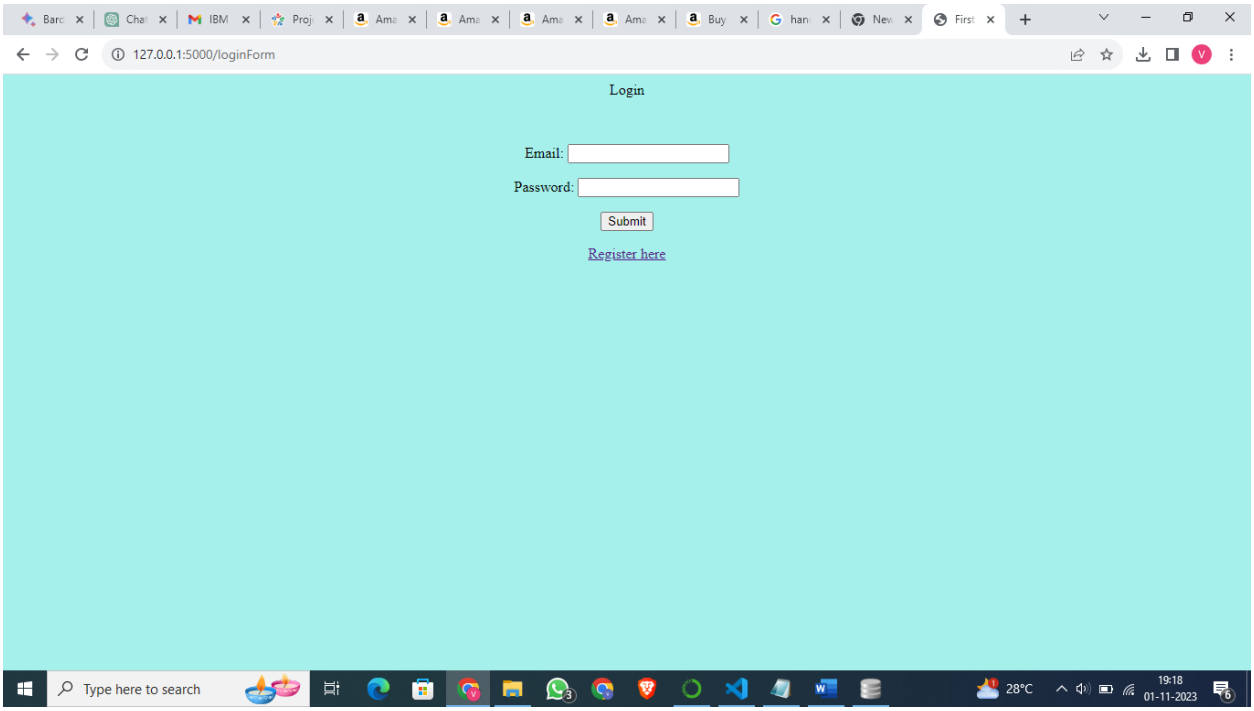
```
<html>
<head>
<title> First flask app </title>
</head>
<body style="background-color: #a6f0eb;">
    <center><p> Login</p><br>
<p> {{error}} </p>
<form action="/login" method="POST">
    <p>Email: <input type="text" name="email"></p>
    <p>Password: <input type="password" name="password"></p>
    <p><input type="submit"></p>
    <a href="/registerationForm">Register here</a>
</form>
</center>
</body>
</html>
```

### MAIN.PY

```
@app.route("/loginForm")
def loginForm():
    if 'email' in session:
        return redirect(url_for('root'))
```

```
    else:
        return render_template('login.html', error='')

@app.route("/login", methods = ['POST', 'GET'])
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        if is_valid(email, password):
            session['email'] = email
            return redirect(url_for('root'))
        else:
            error = 'Invalid UserId / Password'
            return render_template('login.html', error=error)
```



**SHOP CART:**

**CART.HTML**

```
<!DOCTYPE HTML>
<html>
<head>
<title>Category: {{categoryName}}</title>
<link rel="stylesheet" href={{ url_for('static', filename='css/home.css') }} />
<link rel="stylesheet" href={{ url_for('static', filename='css/topStyle.css') }}
/>
</head>
<body>
<div id="title">
    <a href="/">
        <img id="logo" src= {{ url_for('static', filename='images/logo.png') }}
/>
    </a>
    <form>
        <input id="searchBox" type="text" name="searchQuery">
        <input id="searchButton" type="submit" value="Search">
    </form>

    {% if not loggedIn %}
    <div id="signInButton">
        <a class="link" href="/loginForm">Sign In</a>
    </div>
    {% else %}
    <div class="dropdown">
        <button class="dropbtn">Hello, <br>{{firstName}}</button>
        <div class="dropdown-content">
            <a href="/account/orders">Your orders</a>
            <a href="/account/profile">Your profile</a>
            <hr>
            <a href="/logout">Sign Out</a>
        </div>
    </div>
    {% endif %}
    <div id="kart">
```

```
        <a class="link" href="/cart">
            <img src={{url_for('static', filename='images/shoppingCart.png')}}
id="cartIcon" />
            CART {{noOfItems}}
        </a>
    </div>
</div>

<div>
    <h2>Showing all products of Category {{categoryName}}:</h2>
    {% for itemData in data %}
    <table>
        <tr id="productName">
            {% for row in itemData %}
            <td>
                {{row[1]}}
            </td>
            {% endfor %}
        </tr>
        <tr id="productImage">
            {% for row in itemData %}
            <td>
                <a href="/productDescription?productId={{row[0]}}">
                    <img src={{ url_for('static', filename='uploads/' + row[3])
}} id="itemImage" />
                </a>
            </td>
            {% endfor %}
        </tr>
        <tr id="productPrice">
            {% for row in itemData %}
            <td>
                ${{row[2]}}
            </td>
            {% endfor %}
        </tr>
    </table>
    {% endfor %}
</div>
</body>
</html>

</body>
</html>
```

**CART.CSS**

```
#tableItems {
    margin-left: 20px;
    margin-right: 20px;
    margin-top: 20px;
    margin-bottom: 20px;
}

#itemImage {
    margin-left: 5px;
    margin-right: 5px;
    margin-top: 5px;
    margin-bottom: 5px;
    height: 100px;
    float: left;
}

#itemName {
    margin-left: 5px;
    margin-right: 5px;
    margin-top: 5px;
    margin-bottom: 5px;
    height: 100px;
    width: 200px;
    float: left;
}

#titleName {
    width: 200px;
    float: left;
}

#titlePrice {
```

```
    float: left;
}

#itemPrice {
    margin-left: 5px;
    margin-right: 5px;
    margin-top: 5px;
    margin-bottom: 5px;
    height: 100px;
    display: inline-block;
}

#image {
    height: 100px;
    width: 80px;
}

#seperator {
    margin: 0px;
    max-width: 400px;
}

#total {
    padding-left: 280px;
}

#itemNameTag {
    font-weight: bold;
}

#subtotal {
    font-weight: bold;
    font-size: 20px;
}
```

**MAIN.PY**

```
@app.route("/")
def root():
```

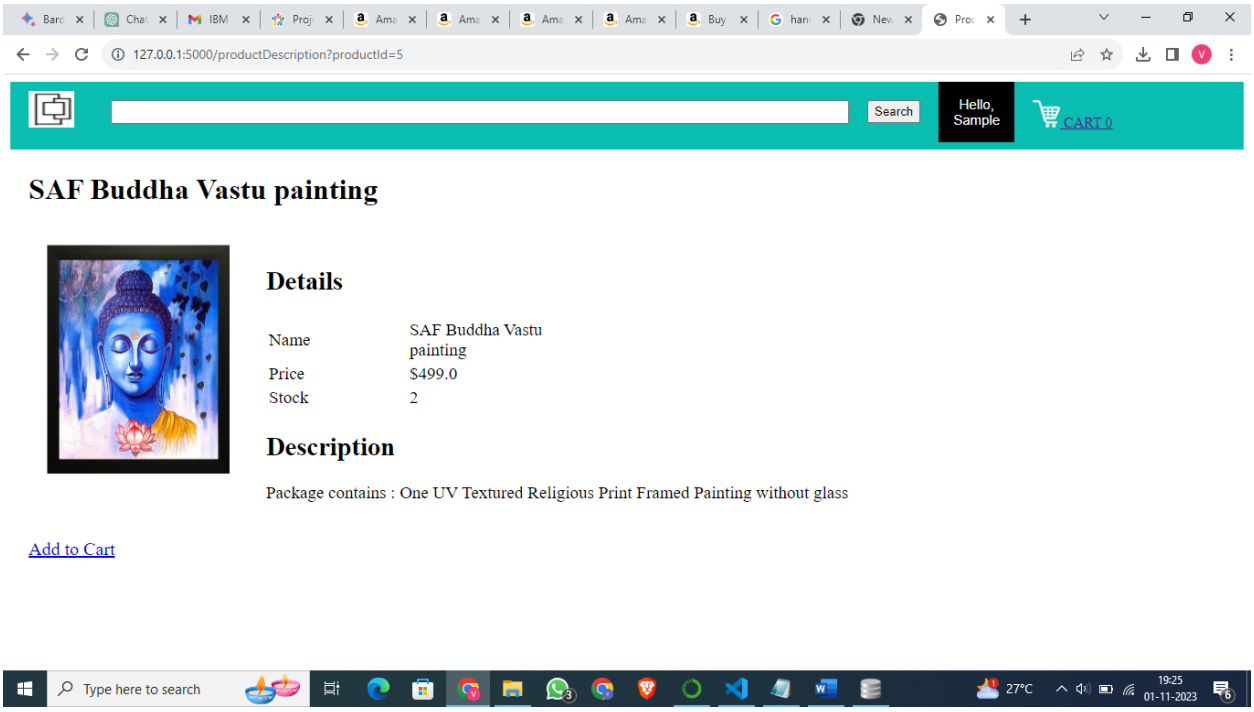
```
    loggedIn, firstName, noOfItems = getLoginDetails()
    with sqlite3.connect('database.db') as conn:
        cur = conn.cursor()
        cur.execute('SELECT productId, name, price, description, image, stock
FROM products')
        itemData = cur.fetchall()
        cur.execute('SELECT categoryId, name FROM categories')
        categoryData = cur.fetchall()
    itemData = parse(itemData)
    return render_template('home.html', itemData=itemData, loggedIn=loggedIn,
firstName=firstName, noOfItems=noOfItems, categoryData=categoryData)

@app.route("/add")
def admin():
    with sqlite3.connect('database.db') as conn:
        cur = conn.cursor()
        cur.execute("SELECT categoryId, name FROM categories")
        categories = cur.fetchall()
    conn.close()
    return render_template('add.html', categories=categories)

@app.route("/addItem", methods=["GET", "POST"])
def addItem():
    if request.method == "POST":
        name = request.form['name']
        price = float(request.form['price'])
        description = request.form['description']
        stock = int(request.form['stock'])
        categoryId = int(request.form['category'])

        #Uploading image procedure
        image = request.files['image']
        if image and allowed_file(image.filename):
            filename = secure_filename(image.filename)
            image.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
        imagename = filename
        with sqlite3.connect('database.db') as conn:
            try:
                cur = conn.cursor()
                cur.execute('''INSERT INTO products (name, price, description,
image, stock, categoryId) VALUES (?, ?, ?, ?, ?, ?)''', (name, price,
description, imagename, stock, categoryId))
                conn.commit()
                msg="added successfully"
            except:
```

```
            msg="error occured"
            conn.rollback()
    conn.close()
    print(msg)
    return redirect(url_for('root'))
```



**CHECKOUT :**

**MYCART.HTML**

```
<!DOCTYPE HTML>
<html>
<head>
<title>Your Cart</title>
<link rel="stylesheet" href={{url_for('static', filename='css/cart.css')}} />
<link rel="stylesheet" href={{url_for('static', filename='css/topStyle.css') }}
/>
</head>
<body>
<div id="title">
    <a href="/">
    <img id="logo" src= {{ url_for('static', filename='images/logo.png') }} />
    </a>
    <form>
        <input id="searchBox" type="text" name="searchQuery">
        <input id="searchButton" type="submit" value="Search">
    </form>

    {% if not loggedIn %}
        <div id="signInButton">
            <a class="link" href="/loginForm">Sign In</a>
        </div>
    {% else %}
    <div class="dropdown">
        <button class="dropbtn">Hello, <br>{{firstName}}</button>
        <div class="dropdown-content">
            <a href="/account/orders">Your orders</a>
            <a href="/account/profile">Your profile</a>
            <hr>
            <a href="/logout">Sign Out</a>
        </div>
    </div>
    {% endif %}
    <div id="kart">
        <a class="link" href="/cart">
        <img src={{url_for('static', filename='images/shoppingCart.png')}}
id="cartIcon" />
```

```
        CART {{noOfItems}}
        </a>
    </div>
</div>
<div id="cartItems">
    <h2>Shopping Cart</h2>
    <div id="tableItems">
        {% for row in products %}
        <div>
            <hr id="seperator">
            <div id="itemImage">
                <img src={{url_for('static', filename='uploads/'+row[3])}}
id="image"/>
            </div>
            <div id="itemName">
                <span id="itemNameTag">{{row[1]}}</span><br>
                In stock<br>
                <a href="/removeFromCart?productId={{row[0]}}">Remove</a>
            </div>
            <div id="itemPrice">
                ${{row[2]}}
            </div>
        </div>
        {% endfor %}
        <hr id="seperator">
        <div id="total">
            <span id="subtotal">Subtotal</span> : ${{totalPrice}}
        </div>
    </div>
</div>
<a href="/checkout">Proceed to checkout</a>
</body>
</html>
```

CHECKOUT.CSS

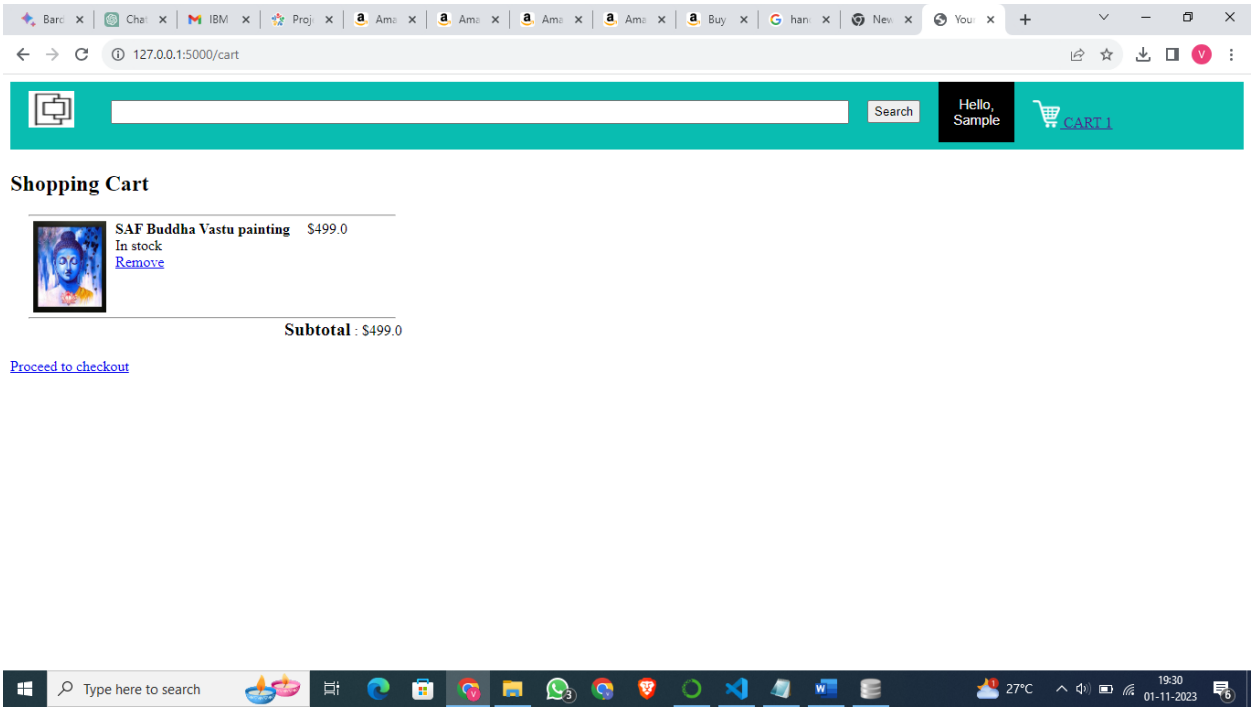
```
#display {
    margin-top: 20px;
    margin-left: 20px;
    margin-right: 20px;
    margin-bottom: 20px;
}
```

```css
#productImage {
    height: 250px;
    width: 200px;
    margin-left: 20px;
    margin-right: 20px;
    margin-top: 20px;
    margin-bottom: 20px;
    display: inline-block;
    float: left;
}

#productDescription {
    margin-left: 20px;
    margin-right: 20px;
    margin-top: 20px;
    margin-bottom: 20px;
    display: inline-block;
    font-size: 19px;
}

#descriptionTable td {
    width: 150px;
}

#addToCart {
    font-size: 20px;
}
```







## REGISTRATION :

## REGISTRATION.HTML:

```
<html>
<head>
<title>Registration</title>
<link rel="stylesheet" href={{ url_for('static', filename='css/remove.css') }} />
<script type="text/javascript" src="{{ url_for('static', filename =
'js/validateForm.js') }}">
</script>
</head>
<body style="background-color: #a6f0eb;">
    <center>
    <p style="font-size: 60 px;">SIGNIN</p><br>
    <form action="/register" method="POST" onsubmit="return validate()">
    <p>Email: <input type="email" name="email"></p>
```

```
    <P>Password: <input type="password" name="password" id="password"
required></p>

    <p>Confirm Password: <input type="password" name="cpassword"
id="cpassword"></p>

    <p>First Name: <input type="text" name="firstName"></p>
    <p>Last Name: <input type="text" name="lastName"></p>
    <p>Address Line 1: <input type="text" name="address1"></p>
    <p>Address Line 2: <input type="text" name="address2"></p>
    <p>Zipcode: <input type="text" name="zipcode"></p>
    <p>City: <input type="text" name="city"></p>
    <p>State: <input type="text" name="state"></p>
    <p>Country: <input type="text" name="country"></p>
    <p>Phone Number: <input type="text" name="phone"></p>

    <p><input type="submit" value="Register"></p>

    <a href="/loginForm">Login here</a>
</form>
</center>
</body>
</html>
```

**FORMVALIDATION.JS**

```
function validate() {
    var pass = document.getElementById("password").value;
    var cpass = document.getElementById("cpassword").value;
    if (pass == cpass) {
        return true;
    } else {
        alert("Passwords do not match");
        return false;
    }
}
```

**MAIN.PY**

```
@app.route("/register", methods = ['GET', 'POST'])
def register():
    if request.method == 'POST':
        #Parse form data
        password = request.form['password']
        email = request.form['email']
        firstName = request.form['firstName']
        lastName = request.form['lastName']
        address1 = request.form['address1']
        address2 = request.form['address2']
        zipcode = request.form['zipcode']
        city = request.form['city']
        state = request.form['state']
        country = request.form['country']
        phone = request.form['phone']

        with sqlite3.connect('database.db') as con:
            try:
                cur = con.cursor()
                cur.execute('INSERT INTO users (password, email, firstName,
lastName, address1, address2, zipcode, city, state, country, phone) VALUES (?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?)', (hashlib.md5(password.encode()).hexdigest(), email,
firstName, lastName, address1, address2, zipcode, city, state, country, phone))

                con.commit()

                msg = "Registered Successfully"
            except:
                con.rollback()
                msg = "Error occured"
        con.close()
        return render_template("login.html", error=msg)

@app.route("/registerationForm")
def registrationForm():
    return render_template("register.html")

def allowed_file(filename):
    return '.' in filename and \
            filename.rsplit('.', 1)[1] in ALLOWED_EXTENSIONS

def parse(data):
    ans = []
```

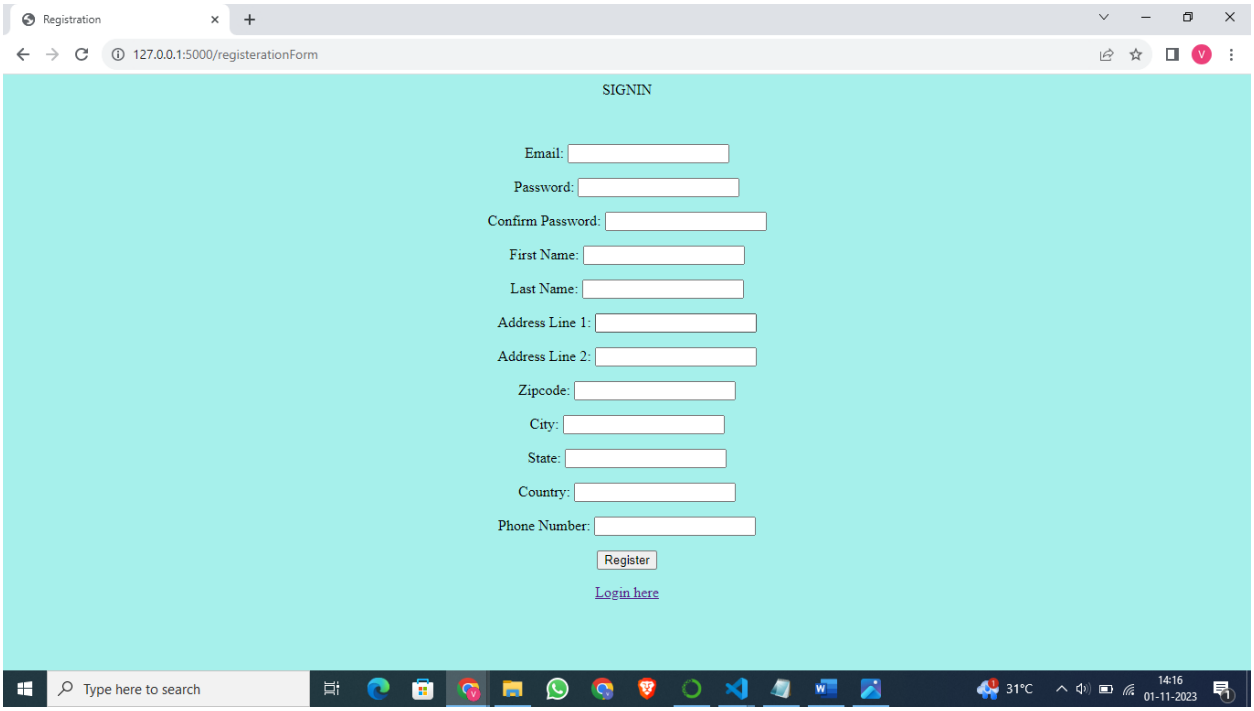
```
    i = 0
    while i < len(data):
        curr = []
        for j in range(7):
            if i >= len(data):
                break
            curr.append(data[i])
            i += 1
        ans.append(curr)
    return ans




@app.route("/account/profile/changePassword", methods=["GET", "POST"])
def changePassword():
    if 'email' not in session:
        return redirect(url_for('loginForm'))
    if request.method == "POST":
        oldPassword = request.form['oldpassword']
        oldPassword = hashlib.md5(oldPassword.encode()).hexdigest()
        newPassword = request.form['newpassword']
        newPassword = hashlib.md5(newPassword.encode()).hexdigest()
        with sqlite3.connect('database.db') as conn:
            cur = conn.cursor()
            cur.execute("SELECT userId, password FROM users WHERE email = ?",
(session['email'], ))
            userId, password = cur.fetchone()
            if (password == oldPassword):
                try:
                    cur.execute("UPDATE users SET password = ? WHERE userId = ?",
(newPassword, userId))
                    conn.commit()
                    msg="Changed successfully"
                except:
                    conn.rollback()
                    msg = "Failed"
                return render_template("changePassword.html", msg=msg)
            else:
                msg = "Wrong password"
        conn.close()
        return render_template("changePassword.html", msg=msg)
    else:
        return render_template("changePassword.html")
```

```
@app.route("/updateProfile", methods=["GET", "POST"])
def updateProfile():
    if request.method == 'POST':
        email = request.form['email']
        firstName = request.form['firstName']
        lastName = request.form['lastName']
        address1 = request.form['address1']
        address2 = request.form['address2']
        zipcode = request.form['zipcode']
        city = request.form['city']
        state = request.form['state']
        country = request.form['country']
        phone = request.form['phone']
        with sqlite3.connect('database.db') as con:
                try:
                    cur = con.cursor()
                    cur.execute('UPDATE users SET firstName = ?, lastName = ?,
address1 = ?, address2 = ?, zipcode = ?, city = ?, state = ?, country = ?, phone
= ? WHERE email = ?', (firstName, lastName, address1, address2, zipcode, city,
state, country, phone, email))

                    con.commit()
                    msg = "Saved Successfully"
                except:
                    con.rollback()
                    msg = "Error occured"
        con.close()
        return redirect(url_for('editProfile'))
```


Registration
127.0.0.1:5000/registerationForm
SIGNIN
Email:
Password:
Confirm Password:
First Name:
Last Name:
Address Line 1:
Address Line 2:
Zipcode:
City:
State:
Country:
Phone Number:
Register
Login here
Type here to search
31°C
14:16
01-11-2023