

Instituto Tecnológico Nacional de México.
Campus Mexicali.



Materia: Programación WEB.

Proyecto Unidad 4.

Docente: López Castellanos Carlos Alberto

Sergio Ulises Lizárraga Pérez 18490949

Mexicali Baja California a 30 de mayo del 2024.

Introducción:

Este reporte presenta el desarrollo de una tienda en línea utilizando una combinación de tecnologías modernas y eficientes como FLASK, MySQL, HTML, CSS, Bootstrap y JavaScript.

El proyecto se enfoca en crear una plataforma de comercio electrónico responsiva, asegurando que los usuarios tengan una experiencia óptima sin importar el dispositivo que utilicen. Además, se ha integrado un sistema de autenticación con diferentes roles de usuario, permitiendo una gestión diferenciada de los recursos y operaciones dentro del sistema.

Requisitos adicionales del proyecto:

Agregar un **LOGIN**. debe contener los campos de **Usuario** y **Password**. Habrá dos tipos de usuario, el usuario **ADMINISTRADOR** y el usuario **CLIENTE** En este momento no importa que las credenciales sean fijas, lo importante es que en una prueba errónea, el proceso **SERVIDOR** responda con un mensaje de **ERROR** cuando las credenciales no funcionen.

Almacenes de colchones

Iniciar sesion

Nombre de usuario

Contraseña

Ingresar



Código

Se utilizo un ModelUsers

```
@classmethod
def login(cls, db, user):
    try:
        cursor = db.connection.cursor()
        cursor.execute("CALL sp_verifyIdentity(%s, %s)", (user.username, user.password))
        row = cursor.fetchone()
        if row is not None and len(row) >= 5:
            user = User(row[0], row[1], row[2], row[4], row[3])
            return user
        else:
            return None
    except Exception as ex:
        raise Exception("Error during login: " + str(ex))
```

Ruta para Login

```
@app.route("/login", methods=["GET", "POST"])
def login():
    if request.method == "POST":
        user = User(0, request.form['username'], request.form['password'], 0)
        logged_user = ModelUsers.login(db, user)
        print("Logged User:", logged_user) # Debug print
        if logged_user is not None:
            login_user(logged_user)
            return redirect(url_for("admin"))
        else:
            flash('Acceso rechazado.', 'danger')
            return render_template("auth/login.html")
    else:
        return render_template("auth/login.html")
```

Una vez loggeado debe desplegarse en las páginas el usuario activo.



Código

Despliegue del usuario actual mediante jinja.

```
<div class="nav-item dropdown">
  <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-bs-toggle="dropdown" aria-expanded="false">
    {{ current_user.fullname }}
    <i class="bi bi-person-circle" style="color: white; margin-left: 5px;"></i>
  </a>
  <ul class="dropdown-menu dropdown-menu-end" aria-labelledby="navbarDropdownMenuLink">
    <li><a class="dropdown-item" href="/logout">Cerrar sesión</a></li>
  </ul>
</div>
```

Debe haber un encabezado con menú en todas las páginas que incluya

- logo de la tienda
- catálogo de productos (que cumpla al menos las especificaciones de la tarea 3)
- tienda (que cumpla al menos las especificaciones de la tarea 4)

Inicio



ALMACENES DE COLCHONES

La experiencia de descanso que te mereces.

NUEVAS OFERTAS

Conocelas dentro de nuestra tienda.

[Ir a la Tienda →](#)



Catalogo



Catalogo de productos

[Registrar nuevo producto](#)



Colchon 1
Queen Size
\$6700



Colchon 2
King Size
\$5000



Colchon 3
Queen Size
\$3991

Usuarios

Usuarios

Crear usuario

ID	Usuario	Nombre completo	Tipo de usuario	Acciones	
3	admin	Sergio L.	Admin	Editar	Eliminar
13	user	user	Usuario	Editar	Eliminar
14	Roberto	Roberto Cerda	Usuario	Editar	Eliminar
16	Alex20	Alejandro Lizarraga	Usuario	Editar	Eliminar

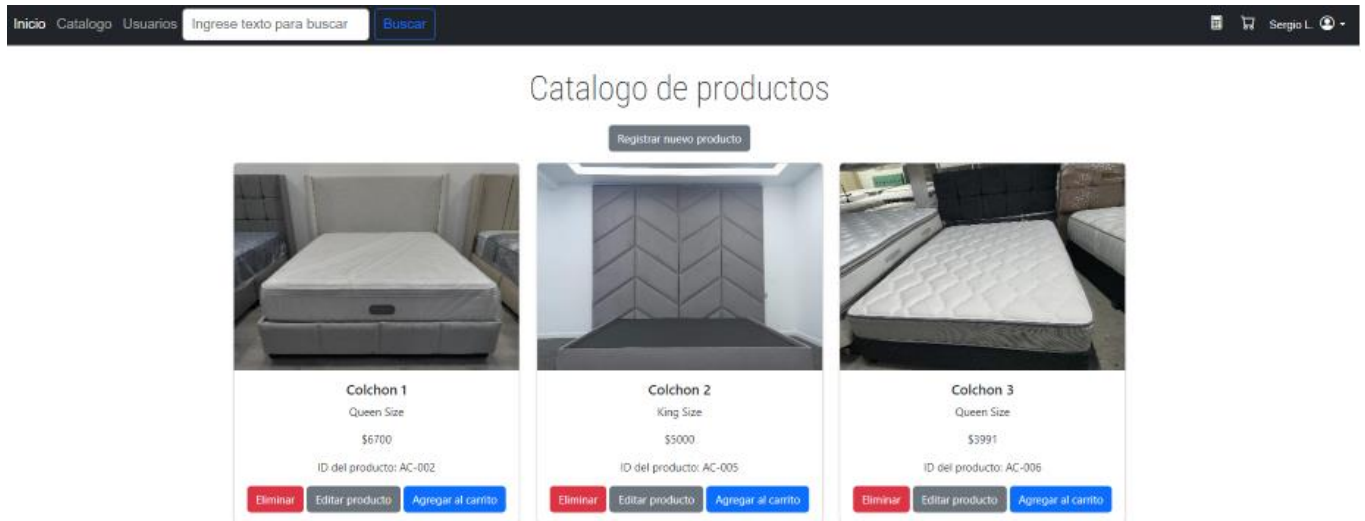
```

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container-fluid">
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav"
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav me-auto">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="/admin">Inicio</a>
        </li>
        <li class="nav-item" id="catalogueLI">
          <a class="nav-link" href="/catalogue">Catalogo</a>
        </li>
        {% if current_user.usertype == 1 %}
          <li class="nav-item">
            <a class="nav-link" href="/users">Usuarios</a>
          </li>
        {% endif %}
        <form class="d-flex" action="{{ url_for('search') }}" method="POST">
          <input type="hidden" name="csrf_token" value="{{ csrf_token() }}" />
          <input class="form-control me-2" type="search" placeholder="Ingrese texto para buscar" aria-label="Search" name="query">
          <button class="btn btn-outline-primary" type="submit">Buscar</button>
        </form>
      </ul>
      <div class="navbar-nav">
        <a class="nav-item btn" id="cartIcon" data-bs-toggle="modal" data-bs-target="#calculatorModal">
          <i class="bi bi-calculator" style="color: white;"></i>
        </a>
        <a class="nav-item btn" id="cartIcon" href="{{ url_for('view_cart') }}">
          <i class="bi bi-cart" style="color: white;"></i>
        </a>
        <!-- Dropdown para el logout -->
        <div class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-bs-toggle="dropdown" aria-expanded="false">
            {{ current_user.fullname }}
          </a>
          <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-bs-toggle="dropdown" aria-expanded="false">
            {{ current_user.fullname }}
            <i class="bi bi-person-circle" style="color: white; margin-left: 5px;"></i>
          </a>
          <ul class="dropdown-menu dropdown-menu-end" aria-labelledby="navbarDropdownMenuLink">
            <li><a class="dropdown-item" href="/logout">Cerrar sesión</a></li>
          </ul>
        </div>
      </div>
    </div>
  </div>
</nav>

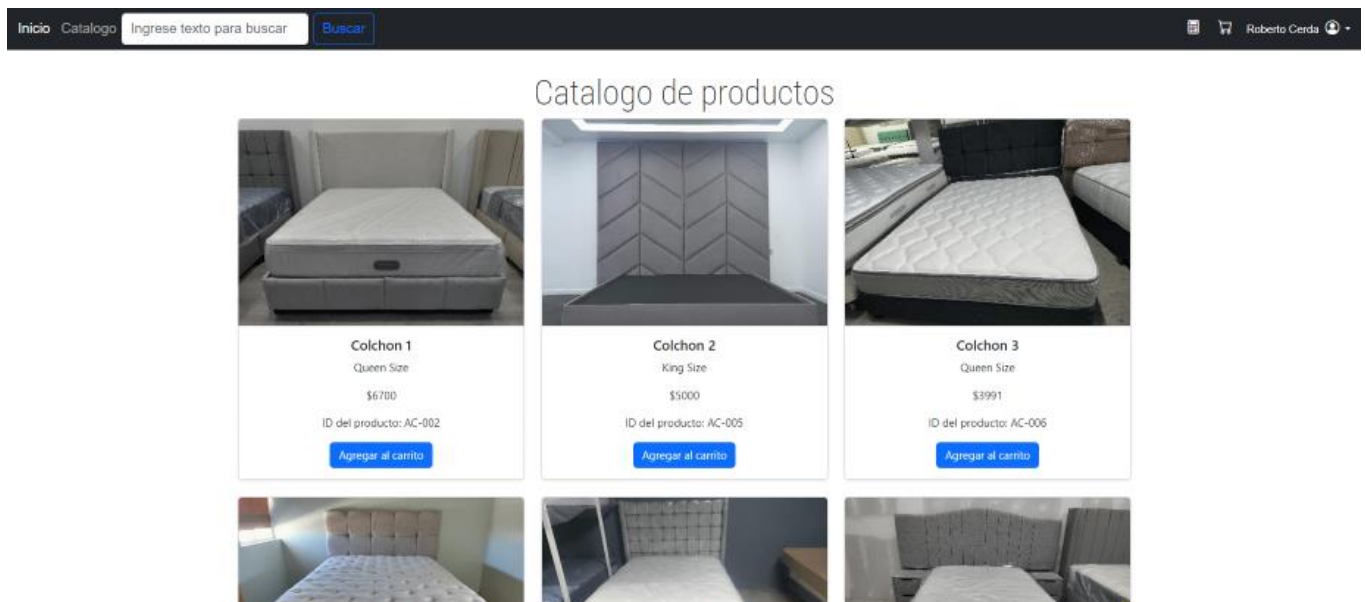
```

El usuario **ADMINISTRADOR** tiene acceso a todas las opciones, el usuario **CLIENTE** no tiene acceso al catálogo de productos

Vista desde Administrador:



Vista usuario común:

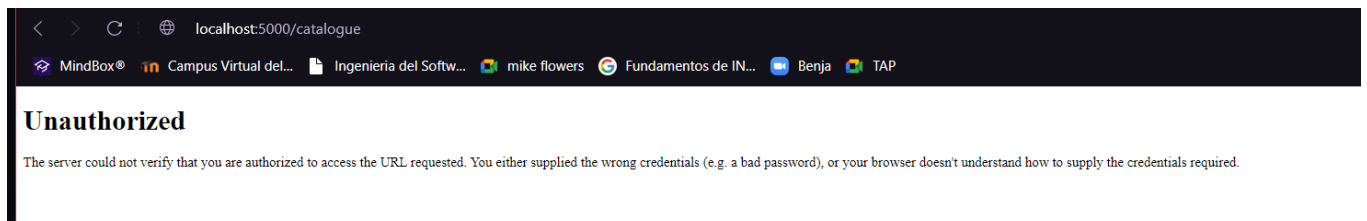


Los botones que controla las acciones del administrador se despliegan o se ocultan utilizando jinja.

Código

```
{% if current_user.usertype == 1 %}
    <button class="btn btn-danger admin-content" onclick="openDeleteProductModal({{ product.id }})">Eliminar</button>
    <button class="btn btn-secondary admin-content" onclick="openEditProductModal({{ product.id }})">Editar producto</button>
{% endif %}
```


Si no se accedió con el usuario no deben poder abrirse las páginas



Se usa el decorador Login_required del flask para estar verificando que exista una sesión válida

Código

```
@app.route('/admin')
@login_required
def admin():
    return render_template("admin.html")

@app.route("/logout")
@login_required
def logout():
    logout_user()
    return redirect(url_for("login"))

@app.route('/catalogue')
@login_required
def catalogue():
    products = ModelProducts.get_all_products(db)
    return render_template('catalogue.html', products=products)
```

Debe implementarse el **CRUD** para los usuarios. Para cada usuario debe incluirse un **id**, el **nombre de usuario**, el **password**, el **nombre completo** y el **tipo de usuario** (Administrador/Cliente). Solo los usuarios administradores pueden usar el CRUD

Usuarios

Crear usuario				
ID	Usuario	Nombre completo	Tipo de usuario	Acciones
3	admin	Sergio L.	Admin	<button>Editar</button> <button>Eliminar</button>
13	user	user	Usuario	<button>Editar</button> <button>Eliminar</button>
14	Roberto	Roberto Cerda	Usuario	<button>Editar</button> <button>Eliminar</button>
16	Alex20	Alejandro Lizarraga	Usuario	<button>Editar</button> <button>Eliminar</button>

Código

Funciones para el manejo de usuarios dentro de model users

```
@classmethod
def get_by_id(cls, db, id):
    try:
        cursor = db.connection.cursor()
        cursor.execute("SELECT id, username, usertype, fullname FROM users WHERE id = %s", (id,))
        row = cursor.fetchone()
        if row is not None:
            return User(row[0], row[1], None, row[2], row[3])
        else:
            return None
    except Exception as ex:
        raise Exception("Error fetching user by ID: " + str(ex))

@classmethod
def get_all_users(self, db):
    try:
        cursor = db.connection.cursor()
        cursor.execute("SELECT id, username, usertype, fullname FROM users")
        rows = cursor.fetchall()
        users = []
        for row in rows:
            users.append(User(row[0], row[1], None, row[2], row[3]))
        return users
    except Exception as ex:
        raise Exception(ex)
```

```

@classmethod
def add_user(self, db, user):
    try:
        cursor = db.connection.cursor()
        cursor.execute("CALL sp_AddUser(%s, %s, %s, %s)", (user.username, user.password, user.fullname, user.usertype))
        db.connection.commit()
    except Exception as ex:
        raise Exception(ex)

@classmethod
def update_user(self, db, user):
    try:
        cursor = db.connection.cursor()
        cursor.execute("UPDATE users SET username = %s, fullname = %s, usertype = %s WHERE id = %s",
            (user.username, user.fullname, user.usertype, user.id))
        db.connection.commit()
    except Exception as ex:
        raise Exception(ex)


@classmethod
def delete_user(self, db, id):
    try:
        cursor = db.connection.cursor()
        cursor.execute("DELETE FROM users WHERE id = %s", (id,))
        db.connection.commit()
    except Exception as ex:
        raise Exception(ex)

```

El catálogo de productos debe implementar el **CRUD**, la tabla de productos debe desplegar la imagen


Catalogo de productos

[Registrar nuevo producto](#)




Colchon 1
Queen Size
\$6700
ID del producto: AC-002

[Eliminar](#)
[Editar producto](#)
[Agregar al carrito](#)



Colchon 2
King Size
\$5000
ID del producto: AC-005

[Eliminar](#)
[Editar producto](#)
[Agregar al carrito](#)



Colchon 3
Queen Size
\$3991
ID del producto: AC-006

[Eliminar](#)
[Editar producto](#)
[Agregar al carrito](#)

Código:

Funciones para el manejo de usuarios dentro de model products

```
class ModelProducts:
    @classmethod
    def get_all_products(self, db):
        try:
            cursor = db.connection.cursor()
            cursor.execute("SELECT * FROM products")
            rows = cursor.fetchall()
            products = []
            for row in rows:
                product = Product(row[0], row[1], row[2], row[3], row[4])
                products.append(product)
            return products
        except Exception as ex:
            raise Exception(ex)

    @classmethod
    def get_product_by_id(self, db, id):
        try:
            cursor = db.connection.cursor()
            cursor.execute("SELECT * FROM products WHERE id = %s", (id,))
            row = cursor.fetchone()
            if row:
                return Product(row[0], row[1], row[2], row[3], row[4])
            else:
                return None
        except Exception as ex:
            raise Exception(ex)

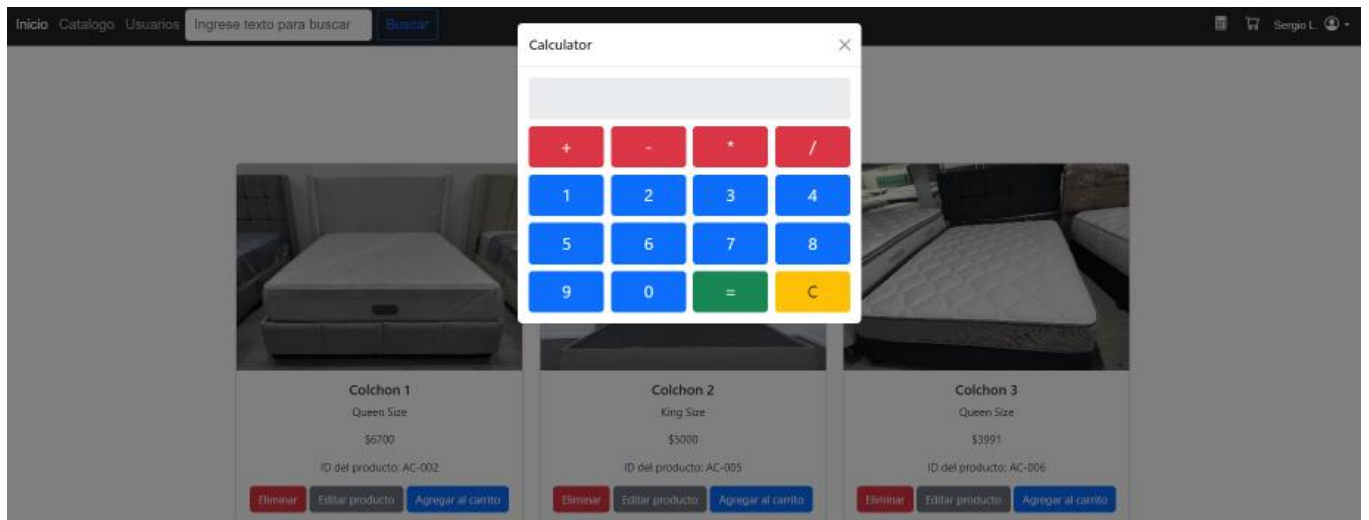
    @classmethod
    def add_product(self, db, product):
        try:
            cursor = db.connection.cursor()
            cursor.execute("INSERT INTO products (prodName, descript, price, image) VALUES (%s, %s, %s, %s)",
                           (product.name, product.description, product.price, product.image))
            db.connection.commit()
        except Exception as ex:
            raise Exception(ex)

    @classmethod
    def update_product(self, db, product):
        try:
            cursor = db.connection.cursor()
            cursor.execute("UPDATE products SET prodName=%s, descript=%s, price=%s, image=%s WHERE id=%s",
                           (product.name, product.description, product.price, product.image, product.id))
            db.connection.commit()
        except Exception as ex:
            raise Exception(ex)

    @classmethod
    def delete_product(self, db, id):
        try:
            cursor = db.connection.cursor()
            cursor.execute("DELETE FROM products WHERE id=%s", (id,))
            db.connection.commit()
        except Exception as ex:
            raise Exception(ex)
```

```
@classmethod
def search_products(cls, db, query):
    try:
        cursor = db.connection.cursor()
        search_query = f"%{query}%"
        cursor.execute("SELECT * FROM products WHERE prodName LIKE %s OR descript LIKE %s", (search_query, search_query))
        rows = cursor.fetchall()
        products = []
        for row in rows:
            products.append(Product(row[0], row[1], row[2], row[3], row[4]))
        return products
    except Exception as ex:
        raise Exception(ex)
```

La tienda, debe permitir habilitar y deshabilitar una calculadora en la misma página



Código:

Se permite desplegarse encima de las ventanas utilizando un modal.

```
<!-- Calculator Modal -->
<div class="modal fade" id="calculatorModal" tabindex="-1" aria-labelledby="calculatorModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="calculatorModalLabel">Calculator</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <div class="calculator">
          <input type="text" class="display form-control" id="display" readonly>
          <button type="button" class="btn btn-danger" data-value="+">+</button>
          <button type="button" class="btn btn-danger" data-value="-">-</button>
          <button type="button" class="btn btn-danger" data-value="*">*</button>
          <button type="button" class="btn btn-danger" data-value="/">/</button>
          <button type="button" class="btn btn-primary" data-value="1">1</button>
          <button type="button" class="btn btn-primary" data-value="2">2</button>
          <button type="button" class="btn btn-primary" data-value="3">3</button>
          <button type="button" class="btn btn-primary" data-value="4">4</button>
          <button type="button" class="btn btn-primary" data-value="5">5</button>
          <button type="button" class="btn btn-primary" data-value="6">6</button>
          <button type="button" class="btn btn-primary" data-value="7">7</button>
          <button type="button" class="btn btn-primary" data-value="8">8</button>
          <button type="button" class="btn btn-primary" data-value="9">9</button>
          <button type="button" class="btn btn-primary" data-value="0">0</button>
          <button type="button" class="btn btn-success" data-value="=">=</button>
          <button type="button" class="btn btn-warning" data-value="C">C</button>
        </div>
      </div>
    </div>
  </div>
</div>
```

Todas las páginas deben ser homogéneas

Inicio

[Inicio](#) [Catalogo](#) [Usuarios](#) [Buscar](#)

  [Sergio L.](#) 

Bienvenido/a, Sergio L.

ALMACENES DE COLCHONES

La experiencia de descanso que te mereces.

NUEVAS OFERTAS

Conocelas dentro de nuestra tienda.

[Ir a la Tienda →](#)






Catalogo

[Inicio](#) [Catalogo](#) [Usuarios](#) [Buscar](#)

  [Sergio L.](#) 

Catalogo de productos

[Registrar nuevo producto](#)

		
Colchon 1	Colchon 2	Colchon 3
Queen Size	King Size	Queen Size
\$6700	\$5000	\$3991
ID del producto: AC-002	ID del producto: AC-005	ID del producto: AC-006

Usuarios

Usuarios

[Crear usuario](#)

ID	Usuario	Nombre completo	Tipo de usuario	Acciones
3	admin	Sergio L.	Admin	<button>Editar</button> <button>Eliminar</button>
13	user	user	Usuario	<button>Editar</button> <button>Eliminar</button>
14	Roberto	Roberto Cerda	Usuario	<button>Editar</button> <button>Eliminar</button>
16	Alex20	Alejandro Lizarraga	Usuario	<button>Editar</button> <button>Eliminar</button>

Código:

Header

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container-fluid">
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav"
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav me-auto">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="/admin">Inicio</a>
        </li>
        <li class="nav-item" id="catalogueLI">
          <a class="nav-link" href="/catalogue">Catalogo</a>
        </li>
        {% if current_user.usertype == 1 %}
          <li class="nav-item">
            <a class="nav-link" href="/users">Usuarios</a>
          </li>
        {% endif %}
        <form class="d-flex" action="{{ url_for('search') }}" method="POST">
          <input type="hidden" name="csrf_token" value="{{ csrf_token() }}" />
          <input class="form-control me-2" type="search" placeholder="Ingrese texto para buscar" aria-label="Search" name="query">
          <button class="btn btn-outline-primary" type="submit">Buscar</button>
        </form>
      </ul>
      <div class="navbar-nav">
        <a class="nav-item btn" id="cartIcon" data-bs-toggle="modal" data-bs-target="#calculatorModal">
          <i class="bi bi-calculator" style="color: white;"></i>
        </a>
        <a class="nav-item btn" id="cartIcon" href="{{ url_for('view_cart') }}">
          <i class="bi bi-cart" style="color: white;"></i>
        </a>
        <!-- Dropdown para el logout -->
        <div class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-bs-toggle="dropdown" aria-label="Dropdown menu" data-cs="2" data-kind="parent">
            </a>
            <div class="dropdown-menu">
              <a href="#">Logout</a>
            </div>
          </div>
      </div>
    </div>
  </div>
</nav>
```

Footer

```
<!-- Footer -->
<footer class="footer mt-auto bg-dark text-white pt-4 pb-2">
  <div class="container text-center">
    <p>Almacenes de colchones © 2024</p>
    <ul class="list-unstyled">
      <li><a href="https://www.facebook.com/profile.php?id=100064155750266" class="text-light">Contactanos</a></li>
    </ul>
  </div>
</footer>
```

Conclusión.

El desarrollo de nuestra tienda en línea con FLASK, MySQL, HTML, CSS, Bootstrap y JavaScript ha resultado en una plataforma moderna y funcional. Este proyecto cumplió con todos los requisitos y logró ofrecer una experiencia de usuario fácil y segura. En resumen, hemos creado una tienda en línea sólida y adaptable que no solo es técnicamente robusta, sino que también proporciona una experiencia de usuario positiva y segura. Este proyecto es una excelente base para futuras mejoras y crecimiento. De esta manera poniendo en practica los conocimientos obtenidos durante las diferentes unidades del semestre.