

server.js:

```
const express = require('express');
const session = require('express-session');
const path = require('path');
const mysql = require('mysql2');
const app = express();
```

// Configurar middleware para sesiones

```
app.use(session({
  secret: 'asd8jkl09jhhk',
  resave: false,
  saveUninitialized: false,
  cookie: { secure: false }
}));
```

// Configurar middleware para manejar archivos estáticos

```
app.use(express.static(path.join(__dirname, 'public')));
```

// Middleware para manejar el cuerpo de las solicitudes POST

```
app.use(express.urlencoded({ extended: true }));
```

// Configurar la conexión a la base de datos

```
const pool = mysql.createPool({
  host: 'localhost',
  user: 'root',
  database: 'olimpiadas',
  password: ''
}).promise();
```

```
// Middleware para verificar la sesión
function verificarSesion(req, res, next) {
  if (req.session.usuario) {
    next(); // El usuario está autenticado, continuar
  } else {
    res.redirect('/login'); // Redirigir al login si no está autenticado
  }
}

// Ruta para servir el index.html, protegido por sesión
app.get('/', (req, res) => {
  res.sendFile(path.join(__dirname, 'public', 'index.html'));
});

// Ruta para servir el formulario de login
app.get('/login', (req, res) => {
  res.sendFile(path.join(__dirname, 'public', 'login.html'));
});

// Ruta para manejar el inicio de sesión
app.post('/login', async (req, res) => {
  const { mail, password } = req.body;

  // Verificar las credenciales con la base de datos
  const [rows] = await pool.query('SELECT * FROM usuario WHERE Gmail = ? AND Contraseña = ?',
    [mail, password]);

  if (rows.length > 0) {
    req.session.usuario = mail; // Guardar el usuario en la sesión
  }
});
```

```
    res.redirect('/'); // Redirigir al index después de iniciar sesión
  } else {
    res.send('Usuario o contraseña incorrectos');
  }
});
```

```
app.get('/check-session', (req, res) => {
  if (req.session.usuario) {
    res.json({ usuario: req.session.usuario });
  } else {
    res.json({ usuario: 'guest' });
  }
});
```

```
// Ruta para cerrar sesión
app.get('/logout', (req, res) => {
  req.session.destroy(err => {
    if (err) {
      return res.send('Error al cerrar sesión');
    }
    res.redirect('/login'); // Redirigir al login después de cerrar sesión
  });
});
```

```
// Ruta para servir el formulario de registro
app.get('/registro', (req, res) => {
  res.sendFile(path.join(__dirname, 'public', 'registro.html'));
});
```

```

// Ruta para manejar el registro de nuevos usuarios
app.post('/registro', async (req, res) => {
  const { username, mail, password } = req.body;

  // Insertar el nuevo usuario en la base de datos
  try {
    await pool.query('INSERT INTO usuario (Nombre, Gmail, Contraseña) VALUES (?, ?, ?)',
[username, mail, password]);
    res.redirect('/login'); // Redirigir al login después de registrarse
  } catch (error) {
    console.error(error);
    res.send('Error al registrarse');
  }
});

// Ruta para obtener productos
app.get('/api/productos', async (req, res) => {
  try {
    // Obtener todos los productos de la base de datos
    const [productos] = await pool.query('SELECT nombre_producto, precio, cantidad FROM
productos');

    // Enviar los productos como respuesta JSON
    res.json(productos);
  } catch (error) {
    console.error('Error al obtener productos:', error);
    res.status(500).send('Error al obtener productos');
  }
});

```

```

// Ruta para servir carrito.html
app.get('/carrito.html', verificarSesion, (req, res) => {
  res.sendFile(path.join(__dirname, 'public', 'carrito.html'));
});

// Ruta para crear un nuevo pedido
app.post('/api/pedidos', async (req, res) => {
  try {
    const usuarioid = req.session.usuarioid;

    if (!usuarioid) {
      return res.status(401).send('No estás autenticado');
    }

    // Crear un nuevo pedido para el usuario
    const [result] = await pool.query('INSERT INTO pedidos (id_usuario, is_carrito) VALUES (?, true)', [usuarioid]);

    res.status(201).json({ id_pedido: result.insertId });
  } catch (error) {
    console.error('Error al crear pedido:', error);
    res.status(500).send('Error al crear el pedido');
  }
});

// Ruta para obtener o crear el carrito del usuario
app.get('/api/carrito', async (req, res) => {
  try {

```

```

const usuariold = req.session.usuariold;

if (!usuariold) {
  return res.status(401).send('No estás autenticado');
}

// Verificar si el usuario ya tiene un carrito activo
const [carrito] = await pool.query('SELECT * FROM pedidos WHERE id_usuario = ? AND is_carrito = true', [usuariold]);

let id_pedido;
if (carrito.length === 0) {
  // Si no existe, crear un nuevo pedido como carrito
  const [result] = await pool.query('INSERT INTO pedidos (id_usuario, is_carrito) VALUES (?, true)', [usuariold]);
  id_pedido = result.insertId;
} else {
  // Si ya existe, obtener el ID del carrito
  id_pedido = carrito[0].id_pedido;
}

res.json({ id_pedido });
} catch (error) {
  console.error('Error al obtener carrito:', error);
  res.status(500).send('Error al obtener el carrito');
}
});

const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {

```

```
console.log(`Servidor ejecutándose en http://localhost:${PORT}`);  
});
```

carrito.html:

```
<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Carrito</title>

  <link rel="stylesheet" href="css/style.css">

  <script defer src="js/carrito.js"></script>

</head>

<body>

  <header>

    <nav class="Navegation">

      <div class="Navegation-left">

        <a href="/">Principal</a>

        <a href="productos.html">Todos los Productos</a>

        <a href="carrito.html">Carrito</a>

      </div>

      <div class="Navegation-right" id="DERECHA">

        <!-- Menú de navegación dinámico aquí -->

      </div>

    </nav>

  </header>

  <main>

    <h1>Carrito de Compras</h1>

    <div id="carrito-container">

      <!-- Los productos del carrito se llenarán aquí -->

    </div>

  </main>
```



</body>

</html>

index.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Olimpiadas</title>

  <link rel="stylesheet" type="text/css" href="css/style.css">

</head>

<body>

  <header>

    <nav class="Navegation">

      <div class="Navegation-left">

        <a href="/">Principal</a>

        <a href="productos.html">Todos los Productos</a>

      </div>

      <div class="Navegation-right" id="DERECHA">

        <script>

          fetch('/check-session')

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

            .then(data => {

              const nav = document.getElementById('DERECHA');

              if (data.usuario === 'guest') {

                nav.innerHTML = '<a href="login.html">Iniciar sesión</a>';

              } else {

                nav.innerHTML = `<a href="carrito.html">Carrito</a><a href="/logout">Cerrar
sesión (${data.usuario})</a>`;

              }

            })

        </script>

      </div>

    </nav>

  </header>

</body>

</html>
```

```
    })

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

</script>

</div>

</nav>

</header>

<div class="container">

  <h2 id="Titulo">Bienvenido a Olimpiadas</h2>

  <div class="card-container">

    <div class="card">

      

      <div class="card-body">

        <h3 class="card-title">Nombre del Producto 1</h3>

        <p class="card-price">$100</p>

        <p class="card-description">Descripción del producto 1.</p>

      </div>

    </div>

    <div class="card">

      

      <div class="card-body">

        <h3 class="card-title">Nombre del Producto 2</h3>

        <p class="card-price">$150</p>

        <p class="card-description">Descripción del producto 2.</p>

      </div>

    </div>

    <div class="card">

      

      <div class="card-body">
```

<h3 class="card-title">Nombre del Producto 3</h3>

<p class="card-price">\$200</p>

<p class="card-description">Descripción del producto 3.</p>

</div>

</div>

</div>

</div>

</body>

</html>

**login.html:**

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">


<meta name="viewport" content="width=device-width, initial-scale=1.0">


<title>Iniciar sesión</title>

<link rel="stylesheet" type="text/css" href="css/cosocss.css">

</head>

<body>

<div class="container">

<form method="post" action="/login">

  <h2>Iniciar sesión</h2>

  <div class="form-group">

    <label for="mail">correo:</label>

    <input type="text" id="mail" name="mail" required>

  </div>

  <div class="form-group">

    <label for="password">Contraseña:</label>

    <input type="password" id="password" name="password" required>

  </div>

  <input type="submit" value="Iniciar sesión">

  <a href="registro.html">Registrarse</a>

</form>

</div>

</body>
```

</html>

<form action="insertar\_login.php" method="POST"></form>

## productos.html:

```
<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Productos</title>

  <link rel="stylesheet" href="css/style.css"> <!-- Asegúrate de que la ruta sea correcta -->

  <link rel="stylesheet" href="css/products.css"> <!-- Asegúrate de que la ruta sea correcta -->

  <script defer src="js/productos.js"></script> <!-- Asegúrate de que la ruta sea correcta -->

</head>

<body>

  <header>

    <nav class="Navegation">

      <div class="Navegation-left">

        <a href="/">Principal</a>

        <a href="productos.html">Todos los Productos</a>

      </div>

      <div class="Navegation-right" id="DERECHA">

        <script>

          fetch('/check-session')

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

            .then(data => {

              const nav = document.getElementById('DERECHA');

              if (data.usuario === 'guest') {

                nav.innerHTML = '<a href="login.html">Iniciar sesión</a>';

              } else {
```

```
        nav.innerHTML = `Carrito</a><a href="/logout">Cerrar
sesión (${data.usuario})</a>`;
    }
  })
  .catch(error => console.error('Error:', error));
</script>
</div>
</nav>
</header>
<main>
  <h1>Lista de Productos</h1>
  <ul id="productos-lista">

    </ul>
</main>
</body>
</html>
```



**registro.html:**

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Registro</title>

<link rel="stylesheet" type="text/css" href="css/cosocss.css">

</head>

<body>

<div class="container">

  <form method="post" action="/registro">

    <h2>Registrarse</h2>

    <div class="form-group">

      <label for="username">Nombre:</label>

      <input type="text" id="username" name="username" required>

    </div>

    <div class="form-group">

      <label for="mail">Correo:</label>

      <input type="email" id="mail" name="mail" required>

    </div>

    <div class="form-group">

      <label for="password">Contraseña:</label>

      <input type="password" id="password" name="password" required>

    </div>

    <div class="form-group">

      <input type="submit" value="Registrarse">

    </div>

  </form>

</div>

</body>

</html>
```

[Iniciar sesión](login.html)

## **COSOCSS.CSS:**

```
.container {  
    width: 300px;  
    margin: 0 auto;  
    padding: 20px;  
    border: 1px solid #ccc;  
    border-radius: 5px;  
    margin-top: 10%;  
}  
  
.form-group {  
    margin-bottom: 10px;  
}  
  
label {  
    display: block;  
    font-weight: bold;  
}  
  
input[type="text"],  
input[type="password"] {  
    width: 100%;  
    padding: 5px;  
    border-radius: 3px;  
    border: 1px solid #ccc;  
}  
  
input[type="submit"] {  
    width: 100%;  
    padding: 8px;  
    background-color: #3e583f;  
    color: white;
```

```
border: none;  
border-radius: 3px;  
cursor: pointer;  
}
```

```
body{  
background-color: #6c996e;  
}
```

## Products.css:

```
/* Estilos para la lista de productos */

#productos-lista {
    list-style-type: none; /* Elimina los puntos de la lista */
    padding: 0; /* Elimina el relleno de la lista */
}

#productos-lista li {
    border: 1px solid #ddd; /* Borde alrededor de cada producto */
    border-radius: 5px; /* Bordes redondeados */
    padding: 15px; /* Espaciado interno */
    margin-bottom: 10px; /* Espacio entre productos */
    display: flex; /* Para alinear el contenido */
    justify-content: space-between; /* Espacio entre el nombre y el botón */
    align-items: center; /* Alinea verticalmente el contenido */
}

#productos-lista button {
    background-color: #4CAF50; /* Color de fondo del botón */
    color: white; /* Color del texto del botón */
    border: none; /* Elimina el borde del botón */
    padding: 10px 20px; /* Espaciado interno del botón */
    border-radius: 5px; /* Bordes redondeados del botón */
    cursor: pointer; /* Cambia el cursor al pasar sobre el botón */
}

#productos-lista button:hover {
    background-color: #45a049; /* Color de fondo al pasar el ratón */
}
```



style.css:

```
/* Estilo general del header */
```

```
header {  
    position: fixed;  
    top: 0;  
    left: 0;  
    display: flex;  
    height: 60px;  
    width: 100%;  
    justify-content: space-between;  
    align-items: center;  
    padding: 10px 50px;  
    z-index: 1;  
    background-color: rgb(94, 131, 102, 0.9);  
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);  
}
```

```
.Navegation {  
    display: flex;  
    justify-content: space-between;  
    width: 100%;  
}
```

```
.Navegation-left {  
    display: flex;  
    gap: 20px; /* Espacio entre los botones */  
}
```

```
.Navegation-right {  
    display: flex;  
    gap: 20px; /* Espacio entre los botones */  
    margin-right: 60px;  
}
```

```
.Navegation a {  
    font-size: 1.1rem;  
    color: aliceblue;  
    text-decoration: none;  
    padding: 10px 15px;  
    border-radius: 5px;  
    transition: background 0.3s, color 0.3s;  
}
```

```
.Navegation a:hover {  
    background: rgba(255, 255, 255, 0.2);  
    color: #fff;  
}
```

```
.Navegation .logout {  
    background: #d9534f; /* Color rojo para logout */  
    color: #fff;  
}
```

/\* Estilo de los enlaces en la navegación \*/

```
.Navegation a {  
    font-size: 1.1rem;
```



```
color: aliceblue;

text-decoration: none;

margin-left: 20px; /* Ajuste de margen */

position: relative;

}
```

```
/* Efecto de subrayado en hover */
```

```
.Navegation a::after {

    content: "";

    position: absolute;

    width: 100%;

    height: 3px;

    background: #fff;

    border-radius: 5px;

    left: 0;

    bottom: -6px;

    transform: scaleX(0);

    transition: .5s;

    transform-origin: right;

}
```

```
.Navegation a:hover::after {

    transform: scaleX(1);

}
```

```
/* Estilo para el header cuando está en estado sticky */
```

```
header.sticky {

    padding: 5px 50px;

    background-color: #fff;
```

```
    opacity: 0.9;
}
```

```
header.sticky a {
    color: black;
}
```

```
/* Estilo general del body */
```

```
body {
    background-color: rgb(94, 131, 102);
    color: rgb(240, 234, 234);
    margin: 0; /* Asegurarse de que no haya margen en el body */
    padding-top: 60px; /* Espacio para el header fijo */
}
```

```
/* Estilo del título */
```

```
#Titulo {
    font-size: 50px;
    text-align: center; /* Centrar el título */
    margin-top: 100px; /* Ajuste del margen superior para espacio entre título y contenido */
}
```

```
/* Estilo del contenedor principal */
```

```
.container {
    max-width: 1000px;
    margin: 50px auto; /* Centrar el contenedor y añadir margen */
    padding: 30px;
}
```

```
/* Estilo de los botones del header */
```

```
.header-buttons {  
    display: flex;  
    align-items: center;  
    gap: 10px; /* Espacio entre botones */  
}
```

```
/* Estilo de los botones */
```

```
#login-button {  
    background-color: #007bff; /* Color de fondo */  
    color: #fff; /* Color del texto */  
    border: none; /* Sin borde */  
    border-radius: 5px; /* Bordes redondeados */  
    padding: 10px 20px; /* Espaciado interno */  
    text-align: center; /* Centra el texto */  
    text-decoration: none; /* Sin subrayado */  
    display: inline-block; /* Mostrar en línea */  
    font-size: 1rem; /* Tamaño de fuente */  
    transition: background-color 0.3s; /* Transición para el color de fondo */  
}
```

```
#login-button:hover {  
    background-color: #0056b3; /* Color de fondo en hover */  
}
```

```
/* Estilo del contenedor de las tarjetas */
```

```
.card-container {  
    display: flex;  
    flex-wrap: wrap;
```

```
gap: 20px;
justify-content: center; /* Centrar las tarjetas */
}
```

```
/* Estilo de las tarjetas */
```

```
.card {
  border: 1px solid #ddd;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0,0,0,0.1);
  width: 240px;
  overflow: hidden;
  transition: transform 0.3s;
}
```

```
.card img {
  width: 100%;
  height: 200px;
  object-fit: cover;
}
```

```
.card:hover {
  transform: translateY(-10px);
}
```

```
.card-body {
  padding: 20px;
  text-align: left;
  color: rgb(240, 234, 234);
}
```

```
.card-title {  
  font-size: 1.5em;  
  margin-bottom: 10px;  
}
```

```
.card-price {  
  color: #050505;  
  font-size: 1.2em;  
  margin-bottom: 20px;  
}
```

**carrito.js:**

```
// carrito.js
```

```
document.addEventListener('DOMContentLoaded', () => {  
  const carritoContainer = document.getElementById('carrito-container');  
  
  // Obtener carrito actual  
  const carrito = JSON.parse(localStorage.getItem('carrito')) || [];  
  
  if (carrito.length > 0) {  
    carrito.forEach((producto, index) => {  
      const productoDiv = document.createElement('div');  
      productoDiv.className = 'producto';  
  
      productoDiv.innerHTML = `  
        <h2>${producto.nombre_producto}</h2>  
        <p>Precio: $$${producto.precio}</p>  
        <button data-index="${index}" class="btn-remove">Eliminar</button>  
      `;  
  
      carritoContainer.appendChild(productoDiv);  
    });  
  
    // Agregar evento de clic para el botón de eliminar  
    document.querySelectorAll('.btn-remove').forEach(boton => {  
      boton.addEventListener('click', (event) => {  
        const index = event.target.getAttribute('data-index');  
  
        // Obtener carrito actual
```

```
const carrito = JSON.parse(localStorage.getItem('carrito')) || [];  
  
// Eliminar el producto del carrito  
carrito.splice(index, 1);  
  
// Guardar el carrito actualizado en localStorage  
localStorage.setItem('carrito', JSON.stringify(carrito));  
  
// Recargar la página para actualizar la vista  
location.reload();  
});  
});  
} else {  
    carritoContainer.innerHTML = '<p>No hay productos en tu carrito.</p>';  
}  
});
```

**productos.js:**

// productos.js

```
document.addEventListener('DOMContentLoaded', () => {  
  // Datos de ejemplo de productos  
  const productos = [  
    { Id_Productos: 1, nombre_producto: 'Zapatillas Adidas', precio: 345 },  
    { Id_Productos: 2, nombre_producto: 'Camisa Nike', precio: 250 }  
  ];  
  
  const productosLista = document.getElementById('productos-lista');  
  
  productos.forEach(producto => {  
    const productItem = document.createElement('li');  
    productItem.className = 'producto-item';  
  
    productItem.innerHTML = `  
      <h2>${producto.nombre_producto}</h2>  
      <p>Precio: $${producto.precio}</p>  
      <button data-id="${producto.Id_Productos}" class="btn-add">Añadir al carrito</button>  
    `;  
  
    productosLista.appendChild(productItem);  
  });  
  
  document.querySelectorAll('.btn-add').forEach(boton => {  
    boton.addEventListener('click', (event) => {  
      const productid = event.target.getAttribute('data-id');  
      const producto = productos.find(p => p.Id_Productos == productid);
```



```
// Obtener carrito actual o inicializarlo
let carrito = JSON.parse(localStorage.getItem('carrito')) || [];

// Añadir el producto al carrito
carrito.push(producto);

// Guardar el carrito en localStorage
localStorage.setItem('carrito', JSON.stringify(carrito));

alert('Producto añadido al carrito');
});
});
});
```