



# **MIT Art, Design and Technology University**

## **MIT School of Computing, Pune**

**Department of Information Technology**

### **Lab Manual**

**Practical - CPAD**

**Class - T.Y. (SEM-VI), SMAD**

**Batch - SMAD-TY**

**Mr.**

**Swarit Omi Upadhyay**

**A.Y. 2024 – 2025 (SEM-II)**

## Experiment No.1

### Problem Statement:

To design and develop a mobile application that combines an educational blog on operating systems with an integrated e-commerce platform, enabling users to explore informative OS-related content and purchase digital tech products in a unified, interactive environment.

### Objective:

- To build an engaging blog interface where users can read about major operating systems such as Windows, macOS, and Linux, including their features, history, and usage.
- To implement a shop module offering digital products and services related to operating systems, such as OS licenses, backup tools, support plans, and bootable media.
- To provide a shopping cart system that allows users to browse, add, and manage items, with a smooth checkout experience including payment and shipping form handling.
- To incorporate user authentication for login and registration, supporting persistent user sessions and personalized experiences.
- To ensure responsive and intuitive navigation across all sections of the app, offering a seamless user experience.

### Theory:

The application serves as a dual-purpose mobile platform that merges educational content with a commerce experience. The **blog component** functions as a knowledge base, offering comprehensive information on various operating systems. This educates users, builds credibility, and fosters engagement from a tech-curious audience.

The **e-commerce component** complements the blog by offering curated digital products directly related to the content. For instance, a user reading about Linux can seamlessly purchase a Linux administration course or bootable USB drive within the same app. The **shopping cart** and **checkout flow** support a convenient transaction experience, including quantity control, pricing summaries, and validation for secure purchases.

By combining content and commerce into a single app, users are empowered to learn and shop in one place, enhancing both educational and transactional value. This hybrid approach also opens up monetization opportunities while maintaining a strong focus on delivering meaningful content.

## Code:

```
import React, { useState, useEffect } from 'react';

import { BrowserRouter as Router, Routes, Route, Link, Navigate, useNavigate } from 'react-router-dom';

import './App.css';

// Helper functions

const getFromStorage = (key) => {

  try {

    const value = localStorage.getItem(key);

    return value ? JSON.parse(value) : null;

  } catch (e) {

    console.error("Error getting data from localStorage:", e);

    return null;

  }

};

const saveToStorage = (key, value) => {

  try {

    localStorage.setItem(key, JSON.stringify(value));

    return true;

  } catch (e) {

    console.error("Error saving data to localStorage:", e);

    return false;

  }

};

// Initialize data if not exists

const initializeData = () => {
```

```
    if (!getFromStorage('users')) {  
      saveToStorage('users', []);  
    }  
  
    if (!getFromStorage('cart')) {  
      saveToStorage('cart', []);  
    }  
  };  
  
  // Main App Component  
  
  function App() {  
  
    const [user, setUser] = useState(getFromStorage('currentUser') || null);  
  
    useEffect(() => {  
      initializeData();  
    }, []);  
  
    const login = (userData) => {  
      setUser(userData);  
      saveToStorage('currentUser', userData);  
    };  
  
    const logout = () => {  
      setUser(null);  
      localStorage.removeItem('currentUser');  
    };  
  
    return (  
      <Router>  
        <div className="App">  
          <Navbar user={user} onLogout={logout} />  
        </div>  
      </Router>  
    );  
  }  
}
```

```

    <div className="container">

      <Routes>

        <Route path="/" element={<Home />} />

        <Route path="/login" element={user ? <Navigate to="/" /> : <Login onLogin={login} /
>} />

        <Route path="/register" element={user ? <Navigate to="/" /> : <Register
onLogin={login} />} />

        <Route path="/shop" element={<Shop />} />

        <Route path="/cart" element={<Cart />} />

        <Route path="/checkout" element={<Checkout />} />

        <Route path="/os/:name" element={<OSDetail />} />

        <Route path="*" element={<NotFound />} />

      </Routes>

    </div>

  </div>

</Router>

);
}

```

```
// Navbar Component
```

```

function Navbar({ user, onLogout }) {

  return (

    <nav className="navbar">

      <div className="nav-brand">

        <Link to="/">OS Blog & Shop</Link>

      </div>

      <ul className="nav-links">

        <li><Link to="/">Home</Link></li>

        <li><Link to="/shop">Shop</Link></li>

        {user ? (

```

```

    </>

    <li><Link to="/cart">Cart</Link></li>

    <li><span className="welcome-text">Welcome, {user.username}</span></li>

    <li><button className="nav-btn" onClick={onLogout}>Logout</button></li>

  </>

) : (

  < >

    <li><Link to="/login">Login</Link></li>

    <li><Link to="/register">Register</Link></li>

  </>

  )}

</ul>

</nav>

);

}

```

```
// Login Component
```

```

function Login({ onLogin }) {

  const navigate = useNavigate();

  const [formData, setFormData] = useState({

    username: '',

    password: ''

  });

  const [error, setError] = useState('');

  const handleChange = (e) => {

    setFormData({

      ...formData,

      [e.target.name]: e.target.value

    });

```

```

};

const handleSubmit = (e) => {

  e.preventDefault();

  const users = getFromStorage('users') || [];

  const user = users.find(u =>

    u.username === formData.username &&

    u.password === formData.password);

  if (user) {

    onLogin(user);

    navigate('/');

  } else {

    setError('Invalid username or password');

  }

};

return (

  <div className="auth-container">

    <h2>Login</h2>

    {error && <div className="error">{error}</div>}

    <form onSubmit={handleSubmit}>

      <div className="form-group">

        <label>Username</label>

        <input

          type="text"

          name="username"

          value={formData.username}

          onChange={handleChange}

          required

```

```

        minLength="3"

    />

</div>

<div className="form-group">

    <label>Password</label>

    <input

        type="password"

        name="password"

        value={formData.password}

        onChange={handleChange}

        required

        minLength="6"

    />

</div>

<button type="submit" className="btn btn-primary">Login</button>

</form>

<p>

    Don't have an account? <Link to="/register">Register here</Link>

</p>

</div>

);

}

// Register Component

function Register({ onLogin }) {

    const navigate = useNavigate();

    const [formData, setFormData] = useState({

        username: '',

        email: '',

        password: '',

```



```
      confirmPassword: ''
    });

    const [error, setError] = useState('');

    const handleChange = (e) => {

      setFormData({

        ...formData,

        [e.target.name]: e.target.value

      });
    };

    const handleSubmit = (e) => {

      e.preventDefault();

      if (formData.password !== formData.confirmPassword) {

        setError('Passwords do not match');

        return;

      }

      const users = getFromStorage('users') || [];

      // Check if username or email already exists

      if (users.some(u => u.username === formData.username)) {

        setError('Username already exists');

        return;

      }

      if (users.some(u => u.email === formData.email)) {

        setError('Email already exists');

        return;

      }

    }
  }
}
```

```
}

// Create new user

const newUser = {

  id: Date.now(),

  username: formData.username,

  email: formData.email,

  password: formData.password

};

users.push(newUser);

saveToStorage('users', users);

// Auto login after registration

onLogin(newUser);

navigate('/');

};

return (

  <div className="auth-container">

    <h2>Register</h2>

    {error && <div className="error">{error}</div>}

    <form onSubmit={handleSubmit}>

      <div className="form-group">

        <label>Username</label>

        <input

          type="text"

          name="username"

          value={formData.username}

          onChange={handleChange}
```

```
        required

        minLength="3"

    />
</div>

<div className="form-group">

    <label>Email</label>

    <input

        type="email"

        name="email"

        value={formData.email}

        onChange={handleChange}

        required

    />

</div>

<div className="form-group">

    <label>Password</label>

    <input

        type="password"

        name="password"

        value={formData.password}

        onChange={handleChange}

        required

        minLength="6"

    />

</div>

<div className="form-group">

    <label>Confirm Password</label>

    <input

        type="password"

        name="confirmPassword"
```

```

        value={formData.confirmPassword}

        onChange={handleChange}

        required

        minLength="6"

      />

    </div>

    <button type="submit" className="btn btn-primary">Register</button>

  </form>

  <p>

    Already have an account? <Link to="/login">Login here</Link>

  </p>

</div>

);

}

// Home Component (OS Blog)

function Home() {

  const operatingSystems = [

    {

      id: 1,

      name: 'Windows',

      slug: 'windows',

      shortDesc: 'Microsoft\'s flagship operating system for personal computers.',

      image: 'https://upload.wikimedia.org/wikipedia/commons/5/5f/Windows_logo_-_2012.svg'

    },

    {

      id: 2,

      name: 'macOS',

      slug: 'macos',

```

```

        shortDesc: 'Apple\'s operating system for Mac computers, known for sleek design and
integration.',

        image: 'https://upload.wikimedia.org/wikipedia/commons/2/21/
MacOS_wordmark_%282017%29.svg'

    },

    {

        id: 3,

        name: 'Linux',

        slug: 'linux',

        shortDesc: 'Open-source operating system based on Unix, known for flexibility and
stability.',

        image: 'https://upload.wikimedia.org/wikipedia/commons/3/35/Tux.svg'

    }

];

return (

<div className="home-container">

    <h1>Operating Systems Blog</h1>

    <p className="intro-text">

        Welcome to our blog about different operating systems. Explore the features,

        history, and advantages of various operating systems used around the world.

    </p>

    <div className="os-grid">

        {operatingSystems.map(os => (

            <div key={os.id} className="os-card">

                <img src={os.image} alt={os.name} className={os.name.toLowerCase() === "macos" ?
"os-logo-macOS" : "os-logo"}

                />

                <h2>{os.name}</h2>

                <p>{os.shortDesc}</p>

                <Link to={`/os/${os.slug}`} className="read-more">Read More</Link>

```

```

        </div>

    )})

</div>

</div>

);
}

// OS Detail Component

function OSDetail() {

    const navigate = useNavigate();

    const pathname = window.location.pathname;

    const osSlug = pathname.split('/').pop();

    const osDetails = {

        windows: {

            name: 'Windows',

            fullDesc: `Windows is a series of operating systems developed by Microsoft. First
released in 1985,

            Windows has become the most widely used desktop operating system worldwide. It features a
graphical

            user interface (GUI), virtual memory management, multitasking, and support for numerous
peripherals.

            Notable versions include Windows 95, which introduced the Start menu and taskbar; Windows
XP, known for

            its stability; and Windows 10, which aimed to unify the user experience across different
device types.

            Windows 11, the latest major release, launched in 2021 with a centered Start menu and
improved virtual

            desktop support.` ,

            history: `Windows began as a graphical shell for MS-DOS. Windows 1.0 was released in
1985, followed by

            Windows 2.0 in 1987. Windows 3.0 and 3.1 gained significant popularity in the early
1990s. The release

```

of Windows 95 marked a significant advancement with its integrated GUI and improved multitasking.

Windows 98, Me, 2000, and XP followed, with XP becoming one of the most successful versions. Windows Vista

faced criticism, but Windows 7 was well-received. Windows 8 introduced a touch-friendly interface, while

Windows 10 returned to a more traditional desktop experience with additional features. Windows 11 was

released in 2021 with a refreshed design and enhanced productivity features.`,

features: `Modern Windows features include:

- The Windows Shell, featuring the Start menu, taskbar, and file explorer
- Windows Security (formerly Windows Defender) for protection against malware
- DirectX for gaming and multimedia
- Microsoft Store for downloading apps
- Virtual desktops for organization
- Cortana virtual assistant
- Microsoft Edge web browser
- Integration with Microsoft 365 services
- Windows Subsystem for Linux (WSL) for running Linux applications
- Support for a wide range of hardware and peripherals`

},

macos: {

name: 'macOS',

fullDesc: `macOS (formerly OS X) is Apple's operating system for Macintosh computers.

Known for its intuitive

interface, stability, and seamless integration with other Apple devices, macOS is built on a Unix foundation,

providing advanced security features and robust performance. The system is designed around a philosophy of

simplicity and user experience, featuring the Dock for application access, Spotlight for searching, and Mission

Control for window management. Apple releases annual updates to macOS, each named after California landmarks

until 2016, and now using version numbers alongside the macOS name.`,

history: `macOS evolved from NeXTSTEP, an operating system developed by NeXT, a company founded by Steve Jobs

after he left Apple in 1985. When Apple acquired NeXT in 1997, they began developing Mac OS X based on NeXTSTEP.

The first public beta was released in 2000, and Mac OS X 10.0 (Cheetah) officially launched in 2001. Subsequent

versions were named after big cats until OS X 10.9 (Mavericks), when Apple switched to California landmarks. In

2016, Apple rebranded OS X as macOS to align with their other operating systems (iOS, watchOS, tvOS). Recent

versions include macOS Monterey, macOS Ventura, and macOS Sonoma.`,

features: `Key macOS features include:

- Unix-based foundation, providing stability and security
- Intuitive user interface with the Dock, Menu Bar, and Mission Control
- Time Machine for automated backups
- Spotlight for system-wide searching
- Continuity features for integration with iOS devices
- iCloud integration for file sharing and syncing
- Terminal for command-line operations
- Gatekeeper for app security
- Built-in apps: Safari, Mail, Photos, Messages, Maps, etc.
- Apple Silicon support, allowing for running iOS/iPadOS apps`

},

linux: {

name: 'Linux',

fullDesc: `Linux is a family of open-source Unix-like operating systems based on the Linux kernel, first released by

Linus Torvalds in 1991. Unlike proprietary operating systems, Linux is developed collaboratively worldwide, with

its source code freely available for modification and distribution. Linux powers a vast range of devices, from



```

    embedded systems and smartphones (Android) to supercomputers and web servers. It's known
    for its stability,

    security, flexibility, and efficiency. Linux is distributed in various "distributions" or
    "distros" that package

    the kernel with different software selections, default configurations, and
    philosophies.`

    history: `Linux began in 1991 when Finnish student Linus Torvalds started developing a
    free kernel for his 386 PC.

    He posted about his project on a newsgroup, which attracted developers worldwide who
    began contributing to the code.

    The kernel was released under the GNU General Public License, making it free software.
    Over time, the kernel was

    combined with GNU tools and other software to create complete operating systems
    (distributions). Early distributions

    included Slackware and Debian in 1993, followed by Red Hat, SUSE, and others. Ubuntu,
    launched in 2004, helped make

    Linux more accessible to average users. Today, Linux powers most of the internet's
    infrastructure, Android devices,

    and is increasingly adopted in enterprise environments.`

    features: `Key Linux features include:

    - Open-source code that anyone can view, modify, and distribute

    - High stability and security with frequent updates

    - Extensive hardware support and efficiency on older hardware

    - Highly customizable interface through various desktop environments (GNOME, KDE, Xfce,
    etc.)

    - Package management systems for easy software installation and updates

    - Command-line interface providing powerful system control

    - No mandatory licensing costs

    - Vibrant community support

    - Strong networking capabilities

    - Multi-user design with robust permission systems`

}

};

```

```
const osData = osDetails[osSlug];

if (!osData) {

  navigate('/');

  return null;

}

return (

  <div className="os-detail-container">

    <h1>{osData.name}</h1>

    <div className="os-section">

      <h2>Overview</h2>

      <p>{osData.fullDesc}</p>

    </div>

    <div className="os-section">

      <h2>History</h2>

      <p>{osData.history}</p>

    </div>

    <div className="os-section">

      <h2>Key Features</h2>

      <p>{osData.features}</p>

    </div>

    <div className="actions">

      <Link to="/" className="btn btn-secondary">Back to All Operating Systems</Link>

    </div>

  </div>

);

}
```

// Shop Component

```
function Shop() {  
  
  const products = [  
  
    {  
  
      id: 1,  
  
      name: 'Windows 11 Pro License',  
  
      price: 199.99,  
  
      image: 'https://upload.wikimedia.org/wikipedia/commons/5/5f/Windows_logo_-_2012.svg',  
  
      description: 'Official Windows 11 Pro license key for one PC.'  
  
    },  
  
    {  
  
      id: 2,  
  
      name: 'macOS Extended Support',  
  
      price: 99.99,  
  
      image: 'https://upload.wikimedia.org/wikipedia/commons/2/21/MacOS_wordmark_%282017%29.svg',  
  
      description: 'Extended support and services for your Mac.'  
  
    },  
  
    {  
  
      id: 3,  
  
      name: 'Linux Administration Course',  
  
      price: 149.99,  
  
      image: 'https://upload.wikimedia.org/wikipedia/commons/3/35/Tux.svg',  
  
      description: 'Comprehensive Linux administration course with certification.'  
  
    },  
  
    {  
  
      id: 4,  
  
      name: 'Multi-OS USB Boot Drive',  
  
      price: 39.99,  
  
      image: 'https://upload.wikimedia.org/wikipedia/commons/thumb/5/5a/USB_Icon.svg/1024px-USB_Icon.svg.png',  
  
    }  
  
  ]  
  
}
```

```

        description: '32GB USB drive pre-configured for booting multiple operating systems.'
    },
    {
        id: 5,
        name: 'OS Backup Software',
        price: 59.99,
        image: 'https://upload.wikimedia.org/wikipedia/commons/thumb/d/da/Database-icon.svg/1024px-Database-icon.svg.png',
        description: 'Reliable backup software compatible with all major operating systems.'
    }
];

```

```

const addToCart = (product) => {

    const cart = getFromStorage('cart') || [];

    // Check if product is already in cart

    const existingProduct = cart.find(item => item.id === product.id);

    if (existingProduct) {

        existingProduct.quantity += 1;

    } else {

        cart.push({

            ...product,

            quantity: 1

        });

    }

    saveToStorage('cart', cart);

    alert(`${product.name} added to cart!`);

};

```

```

return (

  <div className="shop-container">

    <h1>Shop</h1>

    <div className="products-grid">

      {products.map(product => (

        <div key={product.id} className="product-card">

          <img src={product.image} alt={product.name} />

          <h3>{product.name}</h3>

          <p className="product-description">{product.description}</p>

          <p className="product-price">${product.price.toFixed(2)}</p>

          <button

            className="btn btn-primary"

            onClick={() => addToCart(product)}

          >

            Add to Cart

          </button>

        </div>

      ) )}

    </div>

  </div>

);

}

// Cart Component

function Cart() {

  const navigate = useNavigate();

  const [cartItems, setCartItems] = useState(getFromStorage('cart') || []);

  const updateQuantity = (productId, change) => {

```

```
const updatedCart = cartItems.map(item => {

  if (item.id === productId) {

    const newQuantity = item.quantity + change;

    return newQuantity > 0 ? { ...item, quantity: newQuantity } : null;

  }

  return item;

}).filter(Boolean); // Remove null items (quantity reduced to 0)

setCartItems(updatedCart);

saveToStorage('cart', updatedCart);

};

const removeItem = (productId) => {

  const updatedCart = cartItems.filter(item => item.id !== productId);

  setCartItems(updatedCart);

  saveToStorage('cart', updatedCart);

};

const calculateTotal = () => {

  return cartItems.reduce((total, item) => total + (item.price * item.quantity), 0);

};

const proceedToCheckout = () => {

  if (cartItems.length === 0) {

    alert('Your cart is empty');

    return;

  }

  navigate('/checkout');

};
```

```

return (

  <div className="cart-container">

    <h1>Shopping Cart</h1>

    {cartItems.length === 0 ? (

      <div className="empty-cart">

        <p>Your cart is empty</p>

        <Link to="/shop" className="btn btn-primary">Go Shopping</Link>

      </div>

    ) : (

      <>

        <div className="cart-items">

          {cartItems.map(item => (

            <div key={item.id} className="cart-item">

              <img src={item.image} alt={item.name} />

              <div className="item-details">

                <h3>{item.name}</h3>

                <p>${item.price.toFixed(2)}</p>

              </div>

              <div className="quantity-controls">

                <button onClick={() => updateQuantity(item.id, -1)}>-</button>

                <span>{item.quantity}</span>

                <button onClick={() => updateQuantity(item.id, 1)}>+</button>

              </div>

              <p className="item-total">${(item.price * item.quantity).toFixed(2)}</p>

              <button

                className="remove-btn"

                onClick={() => removeItem(item.id)}

              >

                Remove

```

```

        </button>

    </div>

    )}

</div>

<div className="cart-summary">

    <h3>Summary</h3>

    <div className="summary-row">

        <span>Subtotal</span>

        <span>${calculateTotal().toFixed(2)}</span>

    </div>

    <div className="summary-row">

        <span>Tax</span>

        <span>${(calculateTotal() * 0.07).toFixed(2)}</span>

    </div>

    <div className="summary-row total">

        <span>Total</span>

        <span>${(calculateTotal() * 1.07).toFixed(2)}</span>

    </div>

    <button

        className="btn btn-primary checkout-btn"

        onClick={proceedToCheckout}

    >

        Proceed to Checkout

    </button>

</div>

</>

    )}

</div>

);

```



```
}

// Checkout Component

function Checkout() {

  const navigate = useNavigate();

  const [cartItems] = useState(getFromStorage('cart') || []);

  const [formData, setFormData] = useState({

    firstName: '',

    lastName: '',

    email: '',

    address: '',

    city: '',

    zipCode: '',

    cardName: '',

    cardNumber: '',

    expDate: '',

    cvv: ''

  });

  useEffect(() => {

    if (cartItems.length === 0) {

      navigate('/cart');

    }

  }, [cartItems, navigate]);

  const handleChange = (e) => {

    setFormData({

      ...formData,

      [e.target.name]: e.target.value

    });

  });
```

```
};

const calculateTotal = () => {

  return cartItems.reduce((total, item) => total + (item.price * item.quantity), 0);

};

const handleSubmit = (e) => {

  e.preventDefault();

  // Clear cart after successful checkout

  saveToStorage('cart', []);

  // Show order confirmation

  alert('Order placed successfully! Thank you for your purchase.');
```

navigate('/');

```
};

return (

  <div className="checkout-container">

    <h1>Checkout</h1>

    <div className="checkout-grid">

      <div className="checkout-form">

        <form onSubmit={handleSubmit}>

          <h2>Shipping Information</h2>

          <div className="form-row">

            <div className="form-group">

              <label>First Name</label>

              <input

                type="text"
```

```
        name="firstName"

        value={formData.firstName}

        onChange={handleChange}

        required

    />
</div>

<div className="form-group">

    <label>Last Name</label>

    <input

        type="text"

        name="lastName"

        value={formData.lastName}

        onChange={handleChange}

        required

    />

</div>

</div>


<div className="form-group">

    <label>Email</label>

    <input

        type="email"

        name="email"

        value={formData.email}

        onChange={handleChange}

        required

    />

</div>


<div className="form-group">
```

```
<label>Address</label>

<input

  type="text"

  name="address"

  value={formData.address}

  onChange={handleChange}

  required

/>

</div>


<div className="form-row">

  <div className="form-group">

    <label>City</label>

    <input

      type="text"

      name="city"

      value={formData.city}

      onChange={handleChange}

      required

    />

  </div>

  <div className="form-group">

    <label>Zip Code</label>

    <input

      type="text"

      name="zipCode"

      value={formData.zipCode}

      onChange={handleChange}

      required

      pattern="[0-9]{5}"

    />

  </div>

</div>
```

```
        title="Five digit zip code"

    />

</div>

</div>
```

```
<h2>Payment Information</h2>

<div className="form-group">

    <label>Name on Card</label>

    <input

        type="text"

        name="cardName"

        value={formData.cardName}

        onChange={handleChange}

        required

    />

</div>
```

```
<div className="form-group">

    <label>Card Number</label>

    <input

        type="text"

        name="cardNumber"

        value={formData.cardNumber}

        onChange={handleChange}

        required

        pattern="[0-9]{16}"

        title="16 digit card number"

    />

</div>
```

```
<div className="form-row">

  <div className="form-group">

    <label>Expiration Date</label>

    <input

      type="text"

      name="expDate"

      value={formData.expDate}

      onChange={handleChange}

      required

      placeholder="MM/YY"

      pattern="(0[1-9]|1[0-2])\[0-9]{2}"

      title="MM/YY format"

    />

  </div>

  <div className="form-group">

    <label>CVV</label>

    <input

      type="text"

      name="cvv"

      value={formData.cvv}

      onChange={handleChange}

      required

      pattern="[0-9]{3}"

      title="3 digit security code"

    />

  </div>

</div>

<button type="submit" className="btn btn-primary">Place Order</button>

</form>
```

```
</div>
```

```
<div className="order-summary">
```

```
  <h2>Order Summary</h2>
```

```
  <div className="summary-items">
```

```
    {cartItems.map(item => (
```

```
      <div key={item.id} className="summary-item">
```

```
        <div>
```

```
          <span className="item-name">{item.name}</span>
```

```
          <span className="item-quantity">x{item.quantity}</span>
```

```
        </div>
```

```
        <span className="item-price">${(item.price * item.quantity).toFixed(2)}</span>
```

```
      </div>
```

```
    ) ) }
```

```
</div>
```

```
<div className="summary-totals">
```

```
  <div className="summary-row">
```

```
    <span>Subtotal</span>
```

```
    <span>${calculateTotal().toFixed(2)}</span>
```

```
  </div>
```

```
  <div className="summary-row">
```

```
    <span>Tax (7%)</span>
```

```
    <span>${(calculateTotal() * 0.07).toFixed(2)}</span>
```

```
  </div>
```

```
  <div className="summary-row">
```

```
    <span>Shipping</span>
```

```
    <span>FREE</span>
```

```
  </div>
```

```
  <div className="summary-row total">
```

```
    <span>Total</span>
```

```
        <span>${ (calculateTotal() * 1.07).toFixed(2) }</span>

    </div>

</div>

</div>

</div>

</div>

);

}

// 404 Not Found Component

function NotFound() {

    return (

        <div className="not-found">

            <h1>404 - Page Not Found</h1>

            <p>The page you are looking for does not exist.</p>

            <Link to="/" className="btn btn-primary">Back to Home</Link>

        </div>

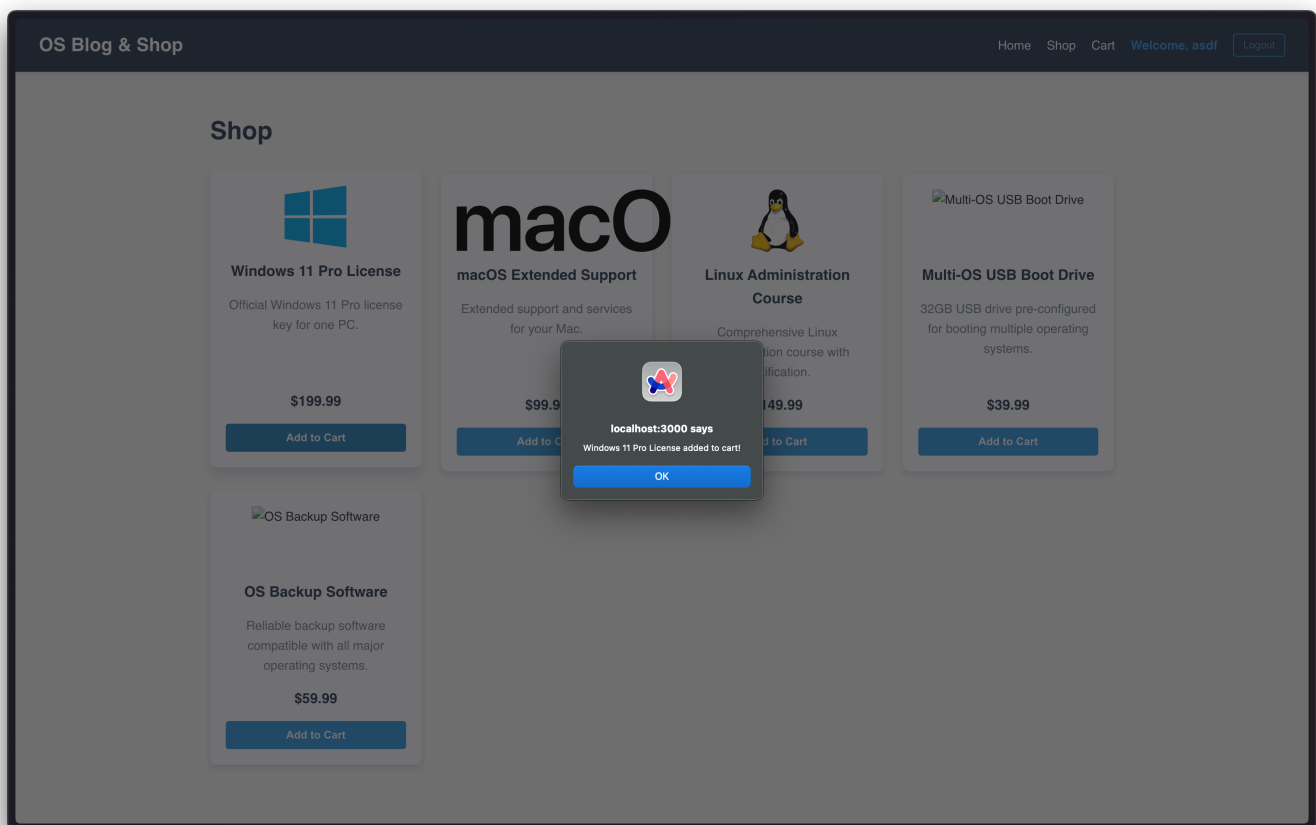
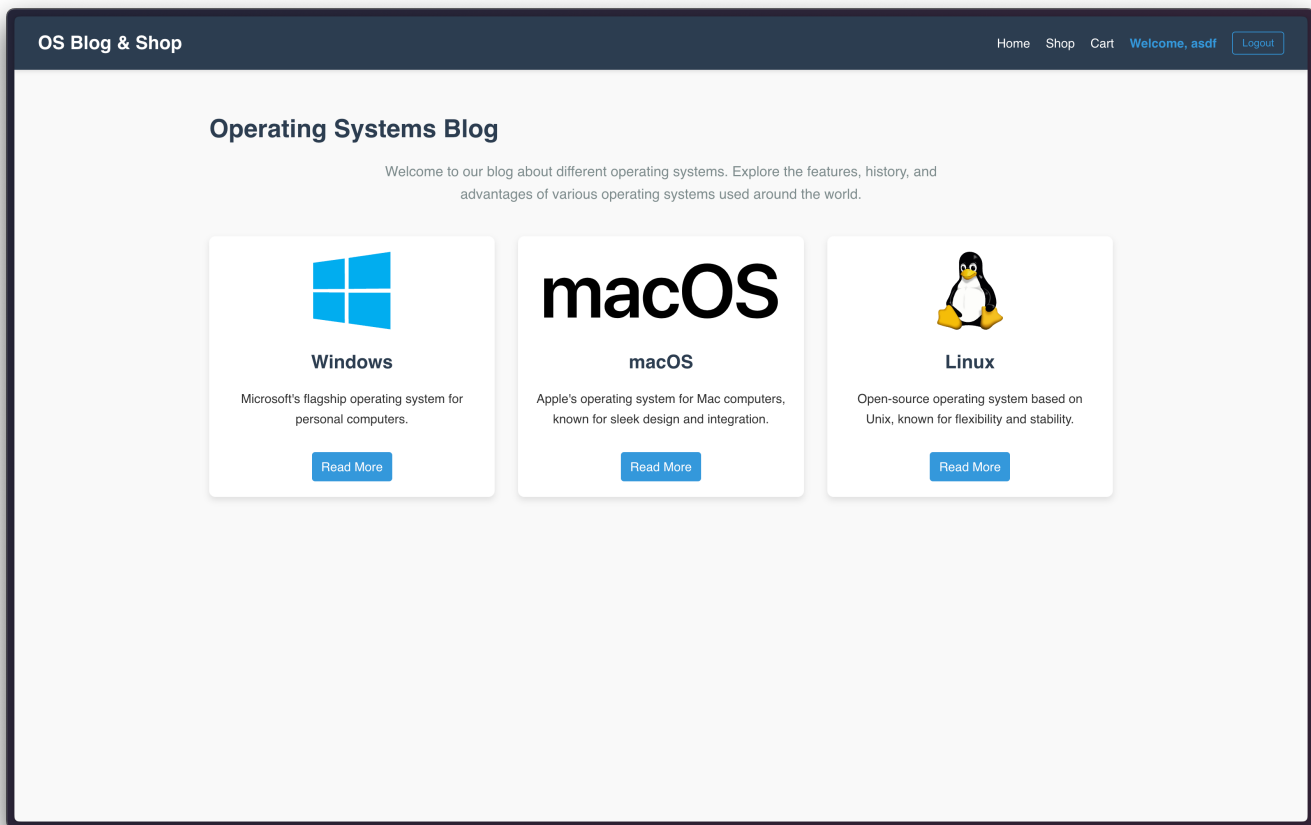
    );

}

export default App;
```



## Output:



## Shopping Cart



Windows 11 Pro License  
\$199.99

- 2 +

\$399.98 [Remove](#)

### Summary

Subtotal	\$399.98
Tax	\$28.00
<b>Total</b>	<b>\$427.98</b>

[Proceed to Checkout](#)

## Checkout

### Shipping Information

First Name	Last Name
<input type="text"/>	<input type="text"/>
Email	
<input type="text"/>	
Address	
<input type="text"/>	
City	Zip Code
<input type="text"/>	<input type="text"/>

### Payment Information

Name on Card	
<input type="text"/>	
Card Number	
<input type="text"/>	
Expiration Date	CVV
<input type="text"/>	<input type="text"/>

[Place Order](#)

### Order Summary

Windows 11 Pro License x2	\$399.98
Subtotal	\$399.98
Tax (7%)	\$28.00
Shipping	FREE
<b>Total</b>	<b>\$427.98</b>

### Login

Username

Password

Login

Don't have an account? [Register here](#)

### Register

Username

Email

Password

Confirm Password

Register

Already have an account? [Login here](#)