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 { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';
import AsyncStorage from '@react-native-async-storage/async-storage';
import {
  View,
  Text,
  StyleSheet,
  TouchableOpacity,
  Image,
  ScrollView,
  FlatList,
  TextInput,
  Alert,
  StatusBar
} from 'react-native';
import { useFocusEffect } from '@react-navigation/native';


// Helper functions
const getFromStorage = async (key) => {
  try {
    const value = await AsyncStorage.getItem(key);
    return value ? JSON.parse(value) : null;
  } catch (e) {
    console.error("Error getting data from AsyncStorage:", e);
    return null;
  }
};
```

```
const saveToStorage = async (key, value) => {
  try {
    await AsyncStorage.setItem(key, JSON.stringify(value));
    return true;
  } catch (e) {
    console.error("Error saving data to AsyncStorage:", e);
    return false;
  }
};

// Initialize data if not exists
const initializeData = async () => {
  const users = await getFromStorage('users');
  if (!users) {
    await saveToStorage('users', []);
  }

  const cart = await getFromStorage('cart');
  if (!cart) {
    await saveToStorage('cart', []);
  }
};

const Stack = createNativeStackNavigator();

// Navbar Component (will be implemented in each screen as a header)
function Navbar({ navigation, user, onLogout }) {
  return (
    <View style={styles.navbar}>
```

```
<TouchableOpacity onPress={() => navigation.navigate('Home')}>
  <Text style={styles.navBrand}>OS Blog & Shop</Text>
</TouchableOpacity>

<View style={styles.navLinks}>
  <TouchableOpacity onPress={() => navigation.navigate('Home')}>
    <Text style={styles.navLink}>Home</Text>
  </TouchableOpacity>

  <TouchableOpacity onPress={() => navigation.navigate('Shop')}>
    <Text style={styles.navLink}>Shop</Text>
  </TouchableOpacity>

  {user ? (
    <>
      <TouchableOpacity onPress={() => navigation.navigate('Cart')}>
        <Text style={styles.navLink}>Cart</Text>
      </TouchableOpacity>

      <Text style={styles.welcomeText}>Welcome, {user.username}</Text>

      <TouchableOpacity style={styles.navBtn} onPress={onLogout}>
        <Text style={styles.navBtnText}>Logout</Text>
      </TouchableOpacity>
    </>
  ) : (
    <>
      <TouchableOpacity onPress={() => navigation.navigate('Login')}>
        <Text style={styles.navLink}>Login</Text>
      </TouchableOpacity>
```

```
            <TouchableOpacity onPress={() => navigation.navigate('Register')}>
              <Text style={styles.navLink}>Register</Text>
            </TouchableOpacity>
          </>
        )}
      </View>
    </View>
  );
}

// Home Screen
function HomeScreen({ navigation, user, onLogout }) {
  const operatingSystems = [
    {
      id: 1,
      name: 'Windows',
      slug: 'windows',
      shortDesc: 'Microsoft\'s flagship operating system for personal computers.',
      image: require('./assets/windows-image.jpg') // You'll need to add these images to your
project
    },
    {
      id: 2,
      name: 'macOS',
      slug: 'macos',
      shortDesc: 'Apple\'s operating system for Mac computers, known for sleek design and
integration.',
      image: require('./assets/macos-image.png')
    },
```

```
    {
      id: 3,
      name: 'Linux',
      slug: 'linux',
      shortDesc: 'Open-source operating system based on Unix, known for flexibility and stability.',
      image: require('./assets/linux.png')
    }
  ];

  return (
    <View style={styles.container}>
      <Navbar navigation={navigation} user={user} onLogout={onLogout} />

      <ScrollView style={styles.content}>
        <View style={styles.homeContainer}>
          <Text style={styles.header}>Operating Systems Blog</Text>

          <Text style={styles.introText}>
            Welcome to our blog about different operating systems. Explore the features,
            history, and advantages of various operating systems used around the world.
          </Text>

          <View style={styles.osGrid}>
            {operatingSystems.map(os => (
              <TouchableOpacity
                key={os.id}
                style={styles.osCard}
                onPress={() => navigation.navigate('OSDetail', { slug: os.slug })}
              >
```

```
                <Image source={os.image} style={styles.osLogo} resizeMode="contain" />
                <Text style={styles.osTitle}>{os.name}</Text>
                <Text style={styles.osDesc}>{os.shortDesc}</Text>
                <Text style={styles.readMore}>Read More</Text>
              </TouchableOpacity>
            ))}
          </View>
        </View>
      </ScrollView>
    </View>
  );
}

// OS Detail Screen
function OSDetailScreen({ route, navigation, user, onLogout }) {
  const { slug } = route.params;

  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[slug];

if (!osData) {
  navigation.navigate('Home');
  return null;
}

return (
  <View style={styles.container}>
    <Navbar navigation={navigation} user={user} onLogout={onLogout} />

    <ScrollView style={styles.content}>
      <View style={styles.osDetailContainer}>
        <Text style={styles.header}>{osData.name}</Text>

        <View style={styles.osSection}>
          <Text style={styles.sectionHeader}>Overview</Text>
          <Text style={styles.sectionText}>{osData.fullDesc}</Text>
        </View>

        <View style={styles.osSection}>
          <Text style={styles.sectionHeader}>History</Text>
          <Text style={styles.sectionText}>{osData.history}</Text>
        </View>
```

```
          <View style={styles.osSection}>
            <Text style={styles.sectionHeader}>Key Features</Text>
            <Text style={styles.sectionText}>{osData.features}</Text>
          </View>

          <View style={styles.actions}>
            <TouchableOpacity
              style={styles.btnSecondary}
              onPress={() => navigation.navigate('Home')}
            >
              <Text style={styles.btnText}>Back to All Operating Systems</Text>
            </TouchableOpacity>
          </View>
        </View>
      </ScrollView>
    </View>
  );
}

// Shop Screen
function ShopScreen({ navigation, user, onLogout }) {
  const products = [
    {
      id: 1,
      name: 'Windows 11 Pro License',
      price: 199.99,
      image: require('./assets/windows-image.jpg'),
      description: 'Official Windows 11 Pro license key for one PC.'
    },
    {
```

```
    id: 2,
    name: 'macOS Extended Support',
    price: 99.99,
    image: require('./assets/macos-image.png'),
    description: 'Extended support and services for your Mac.'
  },
  {
    id: 3,
    name: 'Linux Administration Course',
    price: 149.99,
    image: require('./assets/linux.png'),
    description: 'Comprehensive Linux administration course with certification.'
  },
  {
    id: 4,
    name: 'Multi-OS USB Boot Drive',
    price: 39.99,
    image: require('./assets/usb.png'),
    description: '32GB USB drive pre-configured for booting multiple operating systems.'
  },
  {
    id: 5,
    name: 'OS Backup Software',
    price: 59.99,
    image: require('./assets/backup.png'),
    description: 'Reliable backup software compatible with all major operating systems.'
  }
];

const addToCart = async (product) => {
```

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

  // Check if product is already in cart
  const existingProductIndex = cart.findIndex(item => item.id === product.id);

  if (existingProductIndex !== -1) {
    cart[existingProductIndex].quantity += 1;
  } else {
    cart.push({
      ...product,
      quantity: 1
    });
  }

  await saveToStorage('cart', cart);
  Alert.alert('Success', `${product.name} added to cart!`);
};

return (
  <View style={styles.container}>
    <Navbar navigation={navigation} user={user} onLogout={onLogout} />

    <ScrollView style={styles.content}>
      <View style={styles.shopContainer}>
        <Text style={styles.header}>Shop</Text>

        <View style={styles.productsGrid}>
          {products.map(product => (
            <View key={product.id} style={styles.productCard}>
```

```
                <Image source={product.image} style={styles.productImage}
resizeMode="contain" />
                <Text style={styles.productTitle}>{product.name}</Text>
                <Text style={styles.productDescription}>{product.description}</Text>
                <Text style={styles.productPrice}>${product.price.toFixed(2)}</Text>
                <TouchableOpacity
                  style={styles.btnPrimary}
                  onPress={() => addToCart(product)}
                >
                  <Text style={styles.btnText}>Add to Cart</Text>
                </TouchableOpacity>
              </View>
            ))}
          </View>
        </View>
      </ScrollView>
    </View>
  );
}

// Cart Screen
function CartScreen({ navigation, user, onLogout }) {
  const [cartItems, setCartItems] = useState([]);

  useFocusEffect(
    React.useCallback(() => {
      const loadCart = async () => {
        const cart = await getFromStorage('cart') || [];
        setCartItems(cart);
      };
```

```
    loadCart();
  }, [])
);

const updateQuantity = async (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);
  await saveToStorage('cart', updatedCart);
};

const removeItem = async (productId) => {
  const updatedCart = cartItems.filter(item => item.id !== productId);
  setCartItems(updatedCart);
  await saveToStorage('cart', updatedCart);
};

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

const proceedToCheckout = () => {
  if (cartItems.length === 0) {
```

```
    Alert.alert('Error', 'Your cart is empty');
    return;
  }
  navigation.navigate('Checkout');
};

return (
  <View style={styles.container}>
    <Navbar navigation={navigation} user={user} onLogout={onLogout} />

    <ScrollView style={styles.content}>
      <View style={styles.cartContainer}>
        <Text style={styles.header}>Shopping Cart</Text>

        {cartItems.length === 0 ? (
          <View style={styles.emptyCart}>
            <Text style={styles.emptyCartText}>Your cart is empty</Text>
            <TouchableOpacity
              style={styles.btnPrimary}
              onPress={() => navigation.navigate('Shop')}
            >
              <Text style={styles.btnText}>Go Shopping</Text>
            </TouchableOpacity>
          </View>
        ) : (
          <>
            <View style={styles.cartItems}>
              {cartItems.map(item => (
                <View key={item.id} style={styles.cartItem}>
```

```
                    <Image source={item.image} style={styles.cartItemImage}
resizeMode="contain" />

                    <View style={styles.itemDetails}>
                      <Text style={styles.itemTitle}>{item.name}</Text>
                      <Text style={styles.itemPrice}>${item.price.toFixed(2)}</Text>
                    </View>

                    <View style={styles.quantityControls}>
                      <TouchableOpacity
                        style={styles.quantityBtn}
                        onPress={() => updateQuantity(item.id, -1)}
                      >
                        <Text style={styles.quantityBtnText}>-</Text>
                      </TouchableOpacity>

                      <Text style={styles.quantity}>{item.quantity}</Text>

                      <TouchableOpacity
                        style={styles.quantityBtn}
                        onPress={() => updateQuantity(item.id, 1)}
                      >
                        <Text style={styles.quantityBtnText}>+</Text>
                      </TouchableOpacity>
                    </View>

                    <Text style={styles.itemTotal}>${(item.price * item.quantity).toFixed(2)}</
Text>

                    <TouchableOpacity
```

```
            style={styles.removeBtn}
            onPress={() => removeItem(item.id)}
          >
            <Text style={styles.removeBtnText}>Remove</Text>
          </TouchableOpacity>
        </View>
      ))}
    </View>

    <View style={styles.cartSummary}>
      <Text style={styles.summaryHeader}>Summary</Text>

      <View style={styles.summaryRow}>
        <Text>Subtotal</Text>
        <Text>${calculateTotal().toFixed(2)}</Text>
      </View>

      <View style={styles.summaryRow}>
        <Text>Tax</Text>
        <Text>${(calculateTotal() * 0.07).toFixed(2)}</Text>
      </View>

      <View style={[styles.summaryRow, styles.summaryTotal]}>
        <Text style={styles.totalText}>Total</Text>
        <Text style={styles.totalText}>${(calculateTotal() * 1.07).toFixed(2)}</Text>
      </View>

      <TouchableOpacity
        style={[styles.btnPrimary, styles.checkoutBtn]}
        onPress={proceedToCheckout}
```

```
                >
                  <Text style={styles.btnText}>Proceed to Checkout</Text>
                </TouchableOpacity>
              </View>
            </>
          )}
        </View>
      </ScrollView>
    </View>
  );
}

// Checkout Screen
function CheckoutScreen({ navigation, user, onLogout }) {
  const [cartItems, setCartItems] = useState([]);
  const [formData, setFormData] = useState({
    firstName: '',
    lastName: '',
    email: '',
    address: '',
    city: '',
    zipCode: '',
    cardName: '',
    cardNumber: '',
    expDate: '',
    cvv: ''
  });

  useEffect(() => {
    const loadCart = async () => {
```

```
    const cart = await getFromStorage('cart') || [];
    setCartItems(cart);

    if (cart.length === 0) {
      navigation.navigate('Cart');
    }
  };

  loadCart();
}, [navigation]);

const handleChange = (name, value) => {
  setFormData({
    ...formData,
    [name]: value
  });
};

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

const handleSubmit = async () => {
  // Validate form fields
  for (const key in formData) {
    if (!formData[key]) {
      Alert.alert('Error', 'All fields are required');
      return;
    }
  }
```

```
// Validate zip code format
if (!/^\d{5}$/.test(formData.zipCode)) {
  Alert.alert('Error', 'Please enter a valid 5-digit zip code');
  return;
}

// Validate card number format
if (!/^\d{16}$/.test(formData.cardNumber)) {
  Alert.alert('Error', 'Please enter a valid 16-digit card number');
  return;
}

// Validate expiration date format
if (!/^(0[1-9]|1[0-2])\/\d{2}$/.test(formData.expDate)) {
  Alert.alert('Error', 'Please enter a valid expiration date (MM/YY)');
  return;
}

// Validate CVV format
if (!/^\d{3}$/.test(formData.cvv)) {
  Alert.alert('Error', 'Please enter a valid 3-digit CVV');
  return;
}

// Clear cart after successful checkout
await saveToStorage('cart', []);

// Show order confirmation
Alert.alert(
```

```
    'Success',
    'Order placed successfully! Thank you for your purchase.',
    [{ text: 'OK', onPress: () => navigation.navigate('Home') }]
  );
};

return (
  <View style={styles.container}>
    <Navbar navigation={navigation} user={user} onLogout={onLogout} />

    <ScrollView style={styles.content}>
      <View style={styles.checkoutContainer}>
        <Text style={styles.header}>Checkout</Text>

        <View style={styles.checkoutContent}>
          <View style={styles.checkoutForm}>
            <Text style={styles.formHeader}>Shipping Information</Text>

            <View style={styles.formRow}>
              <View style={[styles.formGroup, styles.formGroupHalf]}>
                <Text style={styles.label}>First Name</Text>
                <TextInput
                  style={styles.input}
                  value={formData.firstName}
                  onChangeText={(text) => handleChange('firstName', text)}
                  placeholder="First Name"
                />
              </View>

              <View style={[styles.formGroup, styles.formGroupHalf]}>
```

```
      <Text style={styles.label}>Last Name</Text>
      <TextInput
        style={styles.input}
        value={formData.lastName}
        onChangeText={(text) => handleChange('lastName', text)}
        placeholder="Last Name"
      />
    </View>
  </View>

  <View style={styles.formGroup}>
    <Text style={styles.label}>Email</Text>
    <TextInput
      style={styles.input}
      value={formData.email}
      onChangeText={(text) => handleChange('email', text)}
      placeholder="Email"
      keyboardType="email-address"
    />
  </View>

  <View style={styles.formGroup}>
    <Text style={styles.label}>Address</Text>
    <TextInput
      style={styles.input}
      value={formData.address}
      onChangeText={(text) => handleChange('address', text)}
      placeholder="Address"
    />
  </View>
```

```
<View style={styles.formRow}>
  <View style={[styles.formGroup, styles.formGroupHalf]}>
    <Text style={styles.label}>City</Text>
    <TextInput
      style={styles.input}
      value={formData.city}
      onChangeText={(text) => handleChange('city', text)}
      placeholder="City"
    />
  </View>

  <View style={[styles.formGroup, styles.formGroupHalf]}>
    <Text style={styles.label}>Zip Code</Text>
    <TextInput
      style={styles.input}
      value={formData.zipCode}
      onChangeText={(text) => handleChange('zipCode', text)}
      placeholder="Zip Code"
      keyboardType="numeric"
      maxLength={5}
    />
  </View>
</View>

<Text style={styles.formHeader}>Payment Information</Text>

<View style={styles.formGroup}>
  <Text style={styles.label}>Name on Card</Text>
  <TextInput
```

```
    style={styles.input}
    value={formData.cardName}
    onChangeText={(text) => handleChange('cardName', text)}
    placeholder="Name on Card"
  />
</View>

<View style={styles.formGroup}>
  <Text style={styles.label}>Card Number</Text>
  <TextInput
    style={styles.input}
    value={formData.cardNumber}
    onChangeText={(text) => handleChange('cardNumber', text)}
    placeholder="Card Number"
    keyboardType="numeric"
    maxLength={16}
  />
</View>

<View style={styles.formRow}>
  <View style={[styles.formGroup, styles.formGroupHalf]}>
    <Text style={styles.label}>Expiration Date</Text>
    <TextInput
      style={styles.input}
      value={formData.expDate}
      onChangeText={(text) => handleChange('expDate', text)}
      placeholder="MM/YY"
      maxLength={5}
    />
  </View>
```

```
    <View style={[styles.formGroup, styles.formGroupHalf]}>
      <Text style={styles.label}>CVV</Text>
      <TextInput
        style={styles.input}
        value={formData.cvv}
        onChangeText={(text) => handleChange('cvv', text)}
        placeholder="CVV"
        keyboardType="numeric"
        maxLength={3}
      />
    </View>
  </View>

  <TouchableOpacity
    style={[styles.btnPrimary, styles.placeOrderBtn]}
    onPress={handleSubmit}
  >
    <Text style={styles.btnText}>Place Order</Text>
  </TouchableOpacity>
</View>

<View style={styles.orderSummary}>
  <Text style={styles.summaryHeader}>Order Summary</Text>

  <FlatList
    data={cartItems}
    keyExtractor={item => item.id.toString()}
    renderItem={({ item }) => (
      <View style={styles.summaryItem}>
```

```
                    <View style={styles.summaryItemInfo}>
                      <Text style={styles.summaryItemName}>{item.name}</Text>
                      <Text style={styles.summaryItemQuantity}>x{item.quantity}</Text>
                    </View>
                    <Text style={styles.summaryItemPrice}>${(item.price *
item.quantity).toFixed(2)}</Text>
                  </View>
                )}
                ListFooterComponent={() => (
                  <View style={styles.summaryTotals}>
                    <View style={styles.summaryRow}>
                      <Text>Subtotal</Text>
                      <Text>${calculateTotal().toFixed(2)}</Text>
                    </View>

                    <View style={styles.summaryRow}>
                      <Text>Tax (7%)</Text>
                      <Text>${(calculateTotal() * 0.07).toFixed(2)}</Text>
                    </View>

                    <View style={styles.summaryRow}>
                      <Text>Shipping</Text>
                      <Text>FREE</Text>
                    </View>

                    <View style={[styles.summaryRow, styles.summaryTotal]}>
                      <Text style={styles.totalText}>Total</Text>
                      <Text style={styles.totalText}>${(calculateTotal() * 1.07).toFixed(2)}</
Text>
                    </View>
```

```
                  </View>
                )}
              />
            </View>
          </View>
        </View>
      </ScrollView>
    </View>
  );
}

// Login Screen
function LoginScreen({ navigation, onLogin }) {
  const [formData, setFormData] = useState({
    username: '',
    password: ''
  });
  const [error, setError] = useState('');

  const handleChange = (name, value) => {
    setFormData({
      ...formData,
      [name]: value
    });
  };

  const handleSubmit = async () => {
    // Validate form
    if (!formData.username || !formData.password) {
      setError('All fields are required');
```

```
    return;
  }

  if (formData.username.length < 3) {
    setError('Username must be at least 3 characters');
    return;
  }

  if (formData.password.length < 6) {
    setError('Password must be at least 6 characters');
    return;
  }

  // Authentication logic
  try {
    const users = await AsyncStorage.getItem('users');
    const parsedUsers = users ? JSON.parse(users) : [];

    const user = parsedUsers.find(u =>
      u.username === formData.username &&
      u.password === formData.password);

    if (user) {
      onLogin(user);
      navigation.navigate('Home');
    } else {
      setError('Invalid username or password');
    }
  } catch (error) {
    setError('An error occurred. Please try again.');
```

```
    console.error(error);
  }
};

return (
  <View style={styles.container}>
    <Text style={styles.title}>Login</Text>

    {error ? <Text style={styles.errorText}>{error}</Text> : null}

    <View style={styles.formGroup}>
      <Text style={styles.label}>Username</Text>
      <TextInput
        style={styles.input}
        value={formData.username}
        onChangeText={(value) => handleChange('username', value)}
        placeholder="Enter your username"
        autoCapitalize="none"
      />
    </View>

    <View style={styles.formGroup}>
      <Text style={styles.label}>Password</Text>
      <TextInput
        style={styles.input}
        value={formData.password}
        onChangeText={(value) => handleChange('password', value)}
        placeholder="Enter your password"
        secureTextEntry
      />
```

```
      </View>

      <TouchableOpacity
        style={styles.button}
        onPress={handleSubmit}
      >
        <Text style={styles.buttonText}>Login</Text>
      </TouchableOpacity>

      <View style={styles.linkContainer}>
        <Text style={styles.linkText}>
          Don't have an account?
        </Text>
        <TouchableOpacity onPress={() => navigation.navigate('Register')}>
          <Text style={styles.link}>Register here</Text>
        </TouchableOpacity>
      </View>
    </View>
  );
}

// Register Screen
function RegisterScreen({ navigation, onLogin }) {
  const [formData, setFormData] = useState({
    username: '',
    email: '',
    password: '',
    confirmPassword: ''
  });
  const [error, setError] = useState('');
```

```
  const handleChange = (name, value) => {
    setFormData({
      ...formData,
      [name]: value
    });
  };

  const handleSubmit = async () => {
    // Validate form
    if (!formData.username || !formData.email || !formData.password || !
formData.confirmPassword) {
      setError('All fields are required');
      return;
    }

    if (formData.username.length < 3) {
      setError('Username must be at least 3 characters');
      return;
    }

    if (formData.password.length < 6) {
      setError('Password must be at least 6 characters');
      return;
    }

    if (formData.password !== formData.confirmPassword) {
      setError('Passwords do not match');
      return;
    }
```

```
// Email validation
const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
if (!emailRegex.test(formData.email)) {
  setError('Please enter a valid email address');
  return;
}

try {
  // Get existing users
  const usersJSON = await AsyncStorage.getItem('users');
  const users = usersJSON ? JSON.parse(usersJSON) : [];

  // 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().toString(),
    username: formData.username,
    email: formData.email,
    password: formData.password
```

```
    };

    // Save updated users array
    users.push(newUser);
    await AsyncStorage.setItem('users', JSON.stringify(users));

    // Auto login after registration
    onLogin(newUser);
    navigation.navigate('Home');
  } catch (error) {
    setError('An error occurred. Please try again.');
    console.error(error);
  }
};

return (
  <ScrollView contentContainerStyle={styles.scrollContainer}>
    <View style={styles.container}>
      <Text style={styles.title}>Register</Text>

      {error ? <Text style={styles.errorText}>{error}</Text> : null}

      <View style={styles.formGroup}>
        <Text style={styles.label}>Username</Text>
        <TextInput
          style={styles.input}
          value={formData.username}
          onChangeText={(value) => handleChange('username', value)}
          placeholder="Enter username"
          autoCapitalize="none"
```

```
    />
  </View>

  <View style={styles.formGroup}>
    <Text style={styles.label}>Email</Text>
    <TextInput
      style={styles.input}
      value={formData.email}
      onChangeText={(value) => handleChange('email', value)}
      placeholder="Enter email"
      keyboardType="email-address"
      autoCapitalize="none"
    />
  </View>

  <View style={styles.formGroup}>
    <Text style={styles.label}>Password</Text>
    <TextInput
      style={styles.input}
      value={formData.password}
      onChangeText={(value) => handleChange('password', value)}
      placeholder="Enter password"
      secureTextEntry
    />
  </View>

  <View style={styles.formGroup}>
    <Text style={styles.label}>Confirm Password</Text>
    <TextInput
      style={styles.input}
```

```
            value={formData.confirmPassword}
            onChangeText={(value) => handleChange('confirmPassword', value)}
            placeholder="Confirm password"
            secureTextEntry
          />
        </View>

        <TouchableOpacity
          style={styles.button}
          onPress={handleSubmit}
        >
          <Text style={styles.buttonText}>Register</Text>
        </TouchableOpacity>

        <View style={styles.linkContainer}>
          <Text style={styles.linkText}>
            Already have an account?
          </Text>
          <TouchableOpacity onPress={() => navigation.navigate('Login')}>
            <Text style={styles.link}>Login here</Text>
          </TouchableOpacity>
        </View>
      </View>
    </ScrollView>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
```

```
    backgroundColor: '#f9f9f9',
    paddingTop: StatusBar.currentHeight || 40,
  },
  content: {
    padding: 16,
  },
  navbar: {
    flexDirection: 'row',
    justifyContent: 'space-between',
    alignItems: 'center',
    backgroundColor: '#1e1e2f',
    paddingVertical: 12,
    paddingHorizontal: 16,
  },
  navBrand: {
    fontSize: 20,
    fontWeight: '700',
    color: '#ffffff',
  },
  navLinks: {
    flexDirection: 'row',
    alignItems: 'center',
  },
  navLink: {
    color: '#ffffff',
    marginHorizontal: 8,
    fontSize: 16,
  },
  navBtn: {
    backgroundColor: '#ff5c5c',
```

```
  borderRadius: 6,
  paddingHorizontal: 10,
  paddingVertical: 6,
  marginLeft: 10,
},
navBtnText: {
  color: '#fff',
  fontWeight: '600',
},
welcomeText: {
  color: '#ffffff',
  fontStyle: 'italic',
  marginLeft: 12,
},
header: {
  fontSize: 28,
  fontWeight: 'bold',
  marginVertical: 16,
  color: '#333',
},
introText: {
  fontSize: 16,
  color: '#555',
  marginBottom: 24,
},
osGrid: {
  flexDirection: 'row',
  flexWrap: 'wrap',
  justifyContent: 'space-between',
},
```

```
osCard: {
  backgroundColor: '#fff',
  borderRadius: 10,
  padding: 16,
  marginBottom: 16,
  width: '48%',
  shadowColor: '#000',
  shadowOffset: { width: 0, height: 2 },
  shadowOpacity: 0.1,
  shadowRadius: 4,
  elevation: 3,
},
osLogo: {
  width: '100%',
  height: 100,
  marginBottom: 12,
},
osTitle: {
  fontSize: 18,
  fontWeight: 'bold',
  marginBottom: 8,
  color: '#222',
},
osDesc: {
  fontSize: 14,
  color: '#666',
  marginBottom: 8,
},
readMore: {
  fontSize: 14,
```

```
  color: '#007bff',
  fontWeight: '600',
},
btnPrimary: {
  backgroundColor: '#007bff',
  borderRadius: 8,
  paddingVertical: 10,
  paddingHorizontal: 16,
  alignItems: 'center',
  marginTop: 10,
},
btnSecondary: {
  backgroundColor: '#6c757d',
  borderRadius: 8,
  paddingVertical: 10,
  paddingHorizontal: 16,
  alignItems: 'center',
  marginTop: 20,
},
btnText: {
  color: '#fff',
  fontWeight: '600',
  fontSize: 16,
},
formGroup: {
  marginBottom: 16,
},
formRow: {
  flexDirection: 'row',
  justifyContent: 'space-between',
```

```
},
formGroupHalf: {
  flex: 0.48,
},
label: {
  marginBottom: 6,
  color: '#444',
  fontWeight: '600',
},
input: {
  backgroundColor: '#fff',
  borderColor: '#ccc',
  borderWidth: 1,
  borderRadius: 6,
  paddingHorizontal: 12,
  paddingVertical: 10,
  fontSize: 16,
},
errorText: {
  color: '#ff4444',
  marginBottom: 12,
  fontWeight: '600',
},
title: {
  fontSize: 26,
  fontWeight: 'bold',
  marginBottom: 20,
  color: '#222',
  textAlign: 'center',
},
```

```
button: {
  backgroundColor: '#28a745',
  paddingVertical: 12,
  borderRadius: 6,
  alignItems: 'center',
  marginTop: 10,
},
buttonText: {
  color: '#fff',
  fontWeight: '600',
  fontSize: 16,
},
linkContainer: {
  flexDirection: 'row',
  justifyContent: 'center',
  marginTop: 16,
},
linkText: {
  fontSize: 14,
  marginRight: 4,
},
link: {
  color: '#007bff',
  fontWeight: '600',
},
productCard: {
  width: '48%',
  backgroundColor: '#fff',
  borderRadius: 10,
  padding: 12,
```

```
    marginBottom: 16,
    shadowColor: '#000',
    shadowOpacity: 0.1,
    shadowOffset: { width: 0, height: 2 },
    elevation: 2,
  },
  productImage: {
    width: '100%',
    height: 100,
    marginBottom: 8,
  },
  productTitle: {
    fontWeight: '700',
    fontSize: 16,
    marginBottom: 4,
    color: '#222',
  },
  productDescription: {
    fontSize: 13,
    color: '#555',
    marginBottom: 6,
  },
  productPrice: {
    fontSize: 15,
    fontWeight: '600',
    color: '#000',
    marginBottom: 8,
  },
  cartItem: {
    flexDirection: 'row',
```

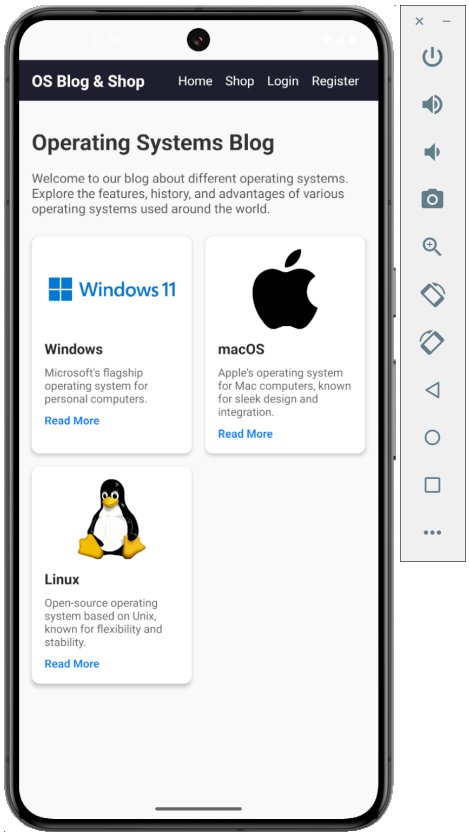
```
    alignItems: 'center',
    marginBottom: 16,
    backgroundColor: '#fff',
    padding: 12,
    borderRadius: 8,
    shadowColor: '#000',
    shadowOpacity: 0.05,
    shadowOffset: { width: 0, height: 1 },
    elevation: 1,
  },
  cartItemImage: {
    width: 60,
    height: 60,
    marginRight: 12,
  },
  itemDetails: {
    flex: 2,
  },
  itemTitle: {
    fontSize: 16,
    fontWeight: '600',
    marginBottom: 4,
  },
  itemPrice: {
    fontSize: 14,
    color: '#666',
  },
  quantityControls: {
    flexDirection: 'row',
    alignItems: 'center',
```

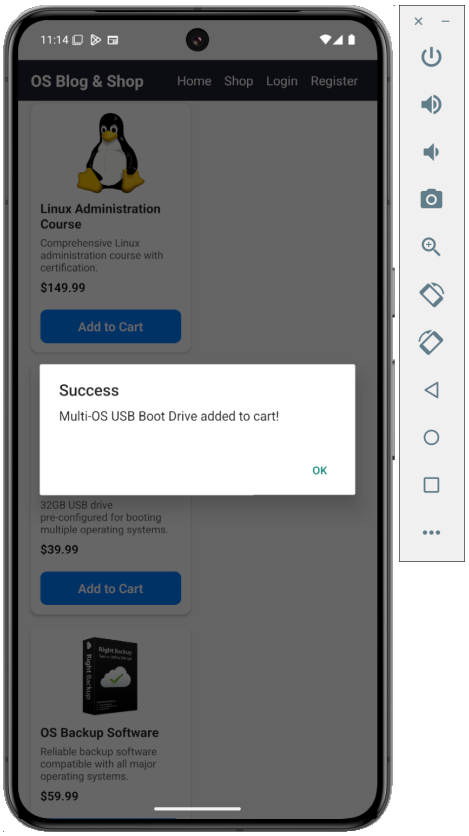
```
},
quantityBtn: {
  backgroundColor: '#e0e0e0',
  paddingHorizontal: 10,
  paddingVertical: 4,
  borderRadius: 4,
},
quantityBtnText: {
  fontSize: 18,
  fontWeight: 'bold',
},
quantity: {
  marginHorizontal: 8,
  fontSize: 16,
},
itemTotal: {
  fontSize: 14,
  fontWeight: '600',
  marginLeft: 8,
},
removeBtn: {
  marginLeft: 8,
},
removeBtnText: {
  color: '#dc3545',
  fontWeight: '600',
},
cartSummary: {
  marginTop: 20,
  backgroundColor: '#fff',
```

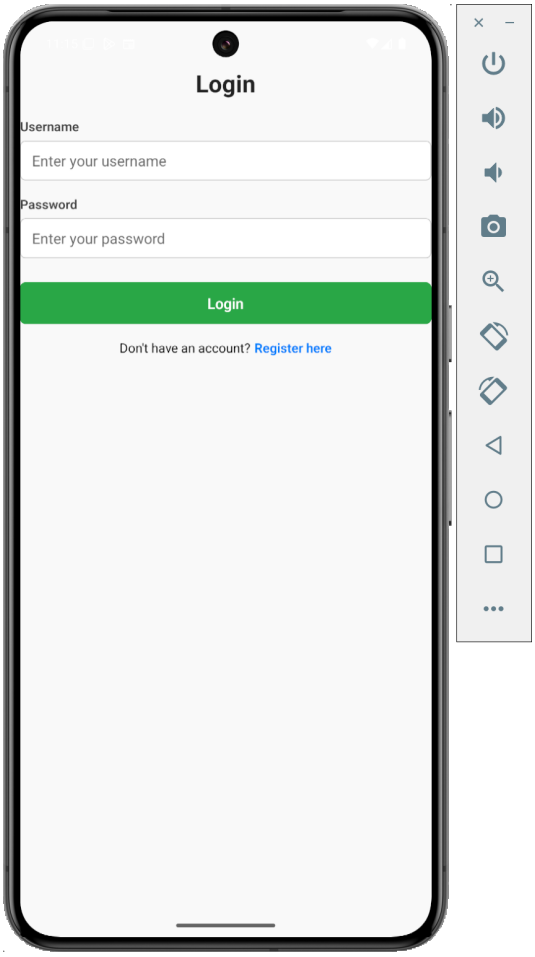
```
  padding: 16,
  borderRadius: 8,
},
summaryRow: {
  flexDirection: 'row',
  justifyContent: 'space-between',
  marginBottom: 8,
},
summaryHeader: {
  fontSize: 18,
  fontWeight: '700',
  marginBottom: 12,
},
summaryTotal: {
  marginTop: 8,
  borderTopWidth: 1,
  borderTopColor: '#ccc',
  paddingTop: 8,
},
totalText: {
  fontSize: 16,
  fontWeight: '700',
},
placeOrderBtn: {
  marginTop: 20,
},
checkoutContainer: {
  paddingBottom: 20,
},
scrollContainer: {
```

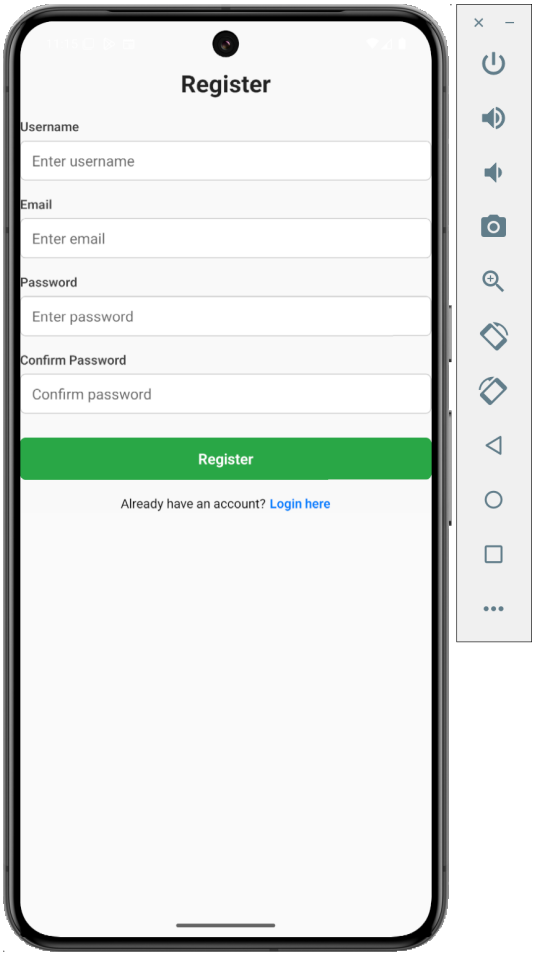
```
    paddingBottom: 40,
  },
});
export default HomeScreen;
```

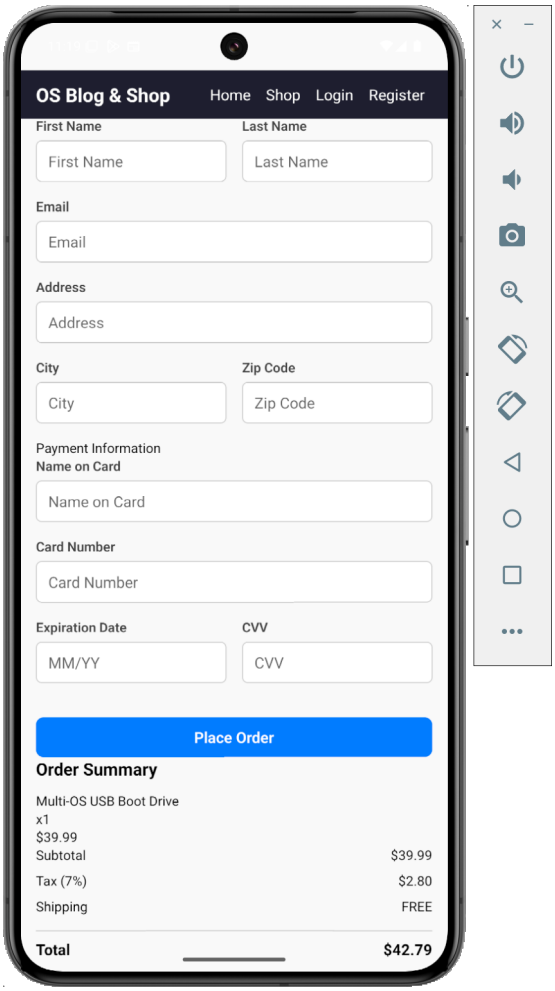
Output:






Login
Username
Enter your username
Password
Enter your password
Login
Don't have an account? Register here


Register
Username
Enter username
Email
Enter email
Password
Enter password
Confirm Password
Confirm password
Register
Already have an account? Login here


OS Blog & Shop
Home Shop Login Register
First Name
Last Name
Email
Address
City
Zip Code
Payment Information
Name on Card
Card Number
Expiration Date
CVV
Place Order
Order Summary
Multi-OS USB Boot Drive
x1
$39.99
Subtotal $39.99
Tax (7%) $2.80
Shipping FREE
Total $42.79