VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE ENGINEERING AND INFORMATION SYSTEMS

INTEGRATED MTECH SOFTWARE ENGINEERING

FALL SEMESTER 2024~2025

TECHNICAL ANSWERS FOR REALWORLD PROBLEMS

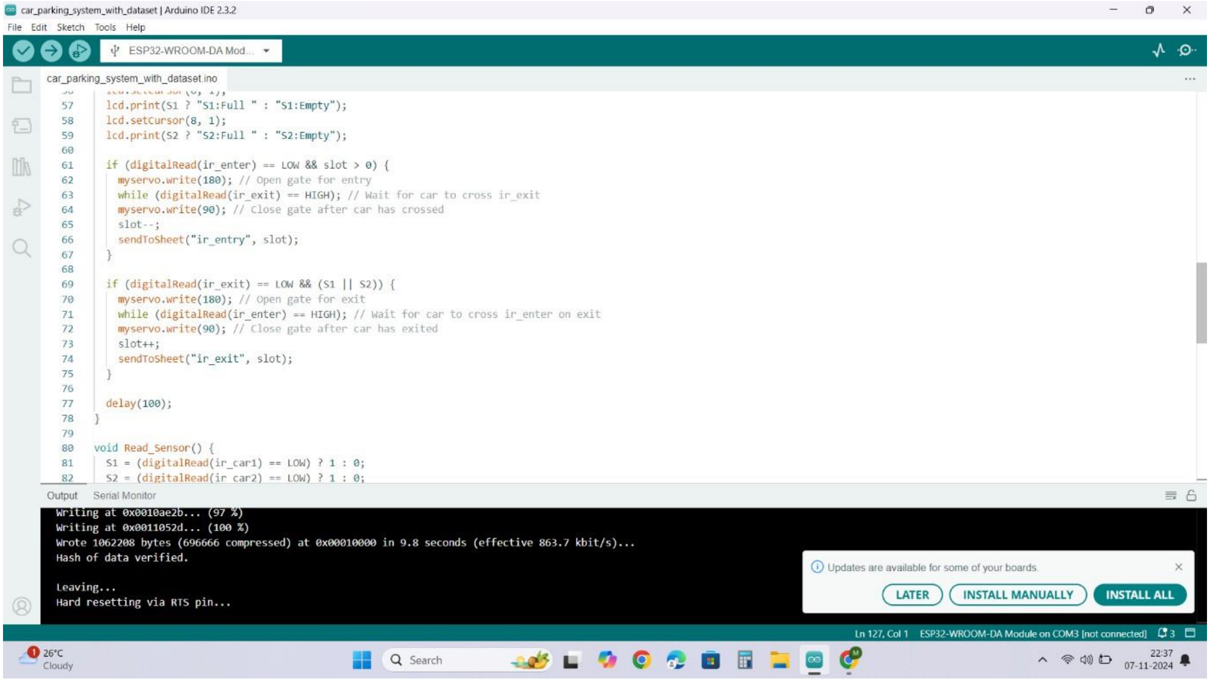
SWE1901

FACULTY: DR. PRABADEVI B

TITLE: IOT BASED CAR PARKING SYSTEM USING HYDRAULIC LIFT

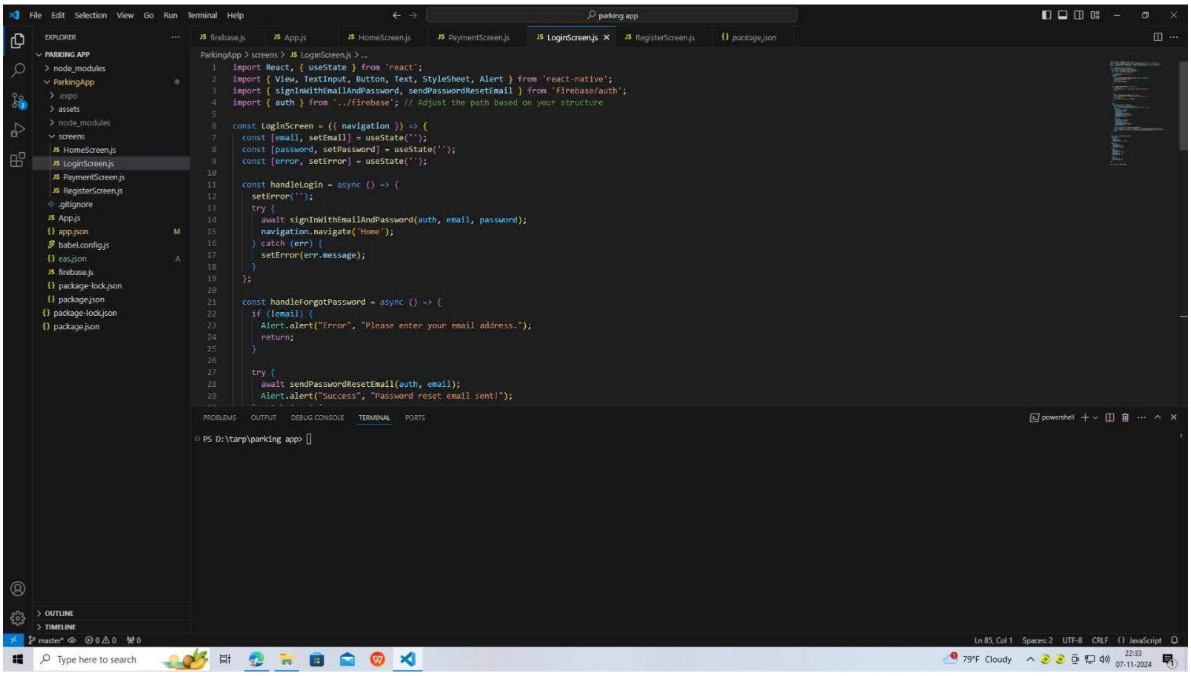
1) Implementation screenshots:

ESP32 CODE:


car_parking_system_with_dataset | Arduino IDE 2.3.2
File Edit Sketch Tools Help
ESP32-WROOM-DA Mod...
car_parking_system_with_dataset.ino
Output Serial Monitor
Updates are available for some of your boards.
LATER
INSTALL MANUALLY
INSTALL ALL
Ln 127, Col 1 ESP32-WROOM-DA Module on COM3 [not connected]
26°C
Cloudy
Search
22:37
07-11-2024

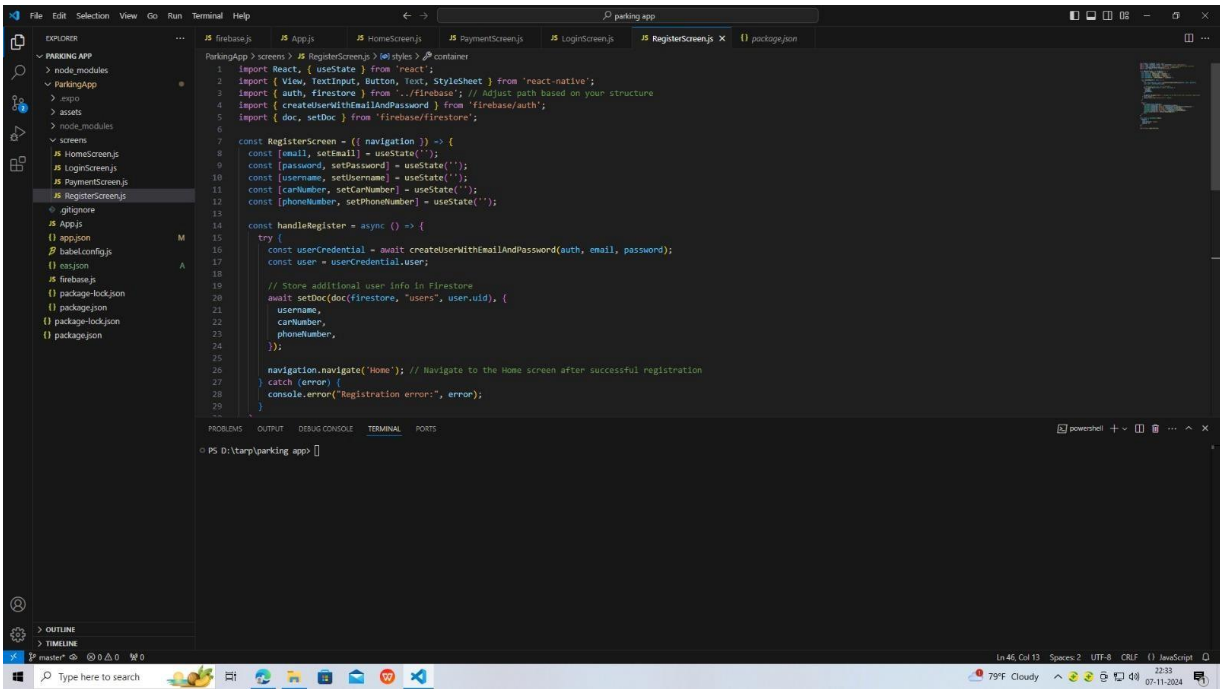
```
  lcd.print(S1 ? "S1:Full " : "S1:Empty");
  lcd.setCursor(8, 1);
  lcd.print(S2 ? "S2:Full " : "S2:Empty");

  if (digitalRead(ir_enter) == LOW && slot > 0) {
    myservo.write(180); // Open gate for entry
    while (digitalRead(ir_exit) == HIGH); // Wait for car to cross ir_exit
    myservo.write(90); // Close gate after car has crossed
    slot--;
    sendToSheet("ir_entry", slot);
  }

  if (digitalRead(ir_exit) == LOW && (S1 || S2)) {
    myservo.write(180); // Open gate for exit
    while (digitalRead(ir_enter) == HIGH); // Wait for car to cross ir_enter on exit
    myservo.write(90); // Close gate after car has exited
    slot++;
    sendToSheet("ir_exit", slot);
  }

  delay(100);
}

void Read_Sensor() {
  S1 = (digitalRead(ir_car1) == LOW) ? 1 : 0;
  S2 = (digitalRead(ir_car2) == LOW) ? 1 : 0;
```

```
Writing at 0x0010ae2b... (97 %)
Writing at 0x0011052d... (100 %)
Wrote 1062208 bytes (696666 compressed) at 0x00010000 in 9.8 seconds (effective 863.7 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```



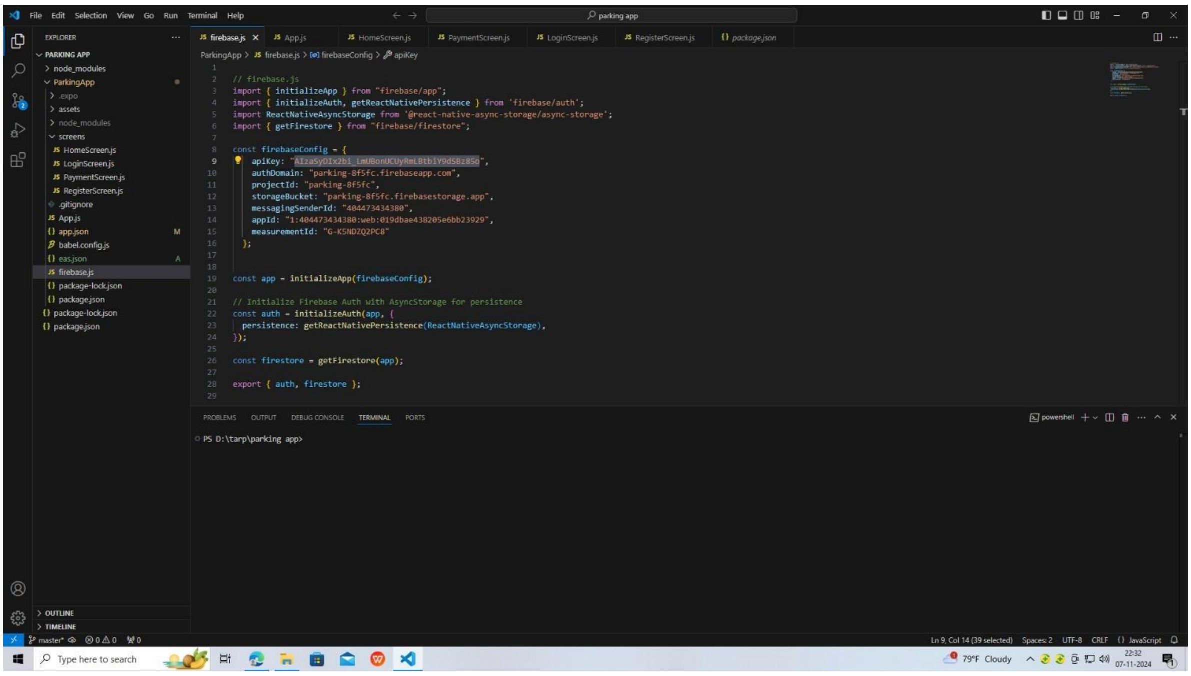
```
import React, { useState } from 'react';
import { View, TextInput, Button, Text, StyleSheet, Alert } from 'react-native';
import { signInWithEmailAndPassword, sendPasswordResetEmail } from 'firebase/auth';
import { auth } from '../firebase'; // Adjust the path based on your structure

const LoginScreen = ({ navigation }) => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [error, setError] = useState('');

  const handleLogin = async () => {
    setError('');
    try {
      await signInWithEmailAndPassword(auth, email, password);
      navigation.navigate('Home');
    } catch (err) {
      setError(err.message);
    }
  };

  const handleForgotPassword = async () => {
    if (!email) {
      Alert.alert("Error", "Please enter your email address.");
      return;
    }

    try {
      await sendPasswordResetEmail(auth, email);
      Alert.alert("Success", "Password reset email sent!");
```

```
PS D:\tarp\parking app>
```



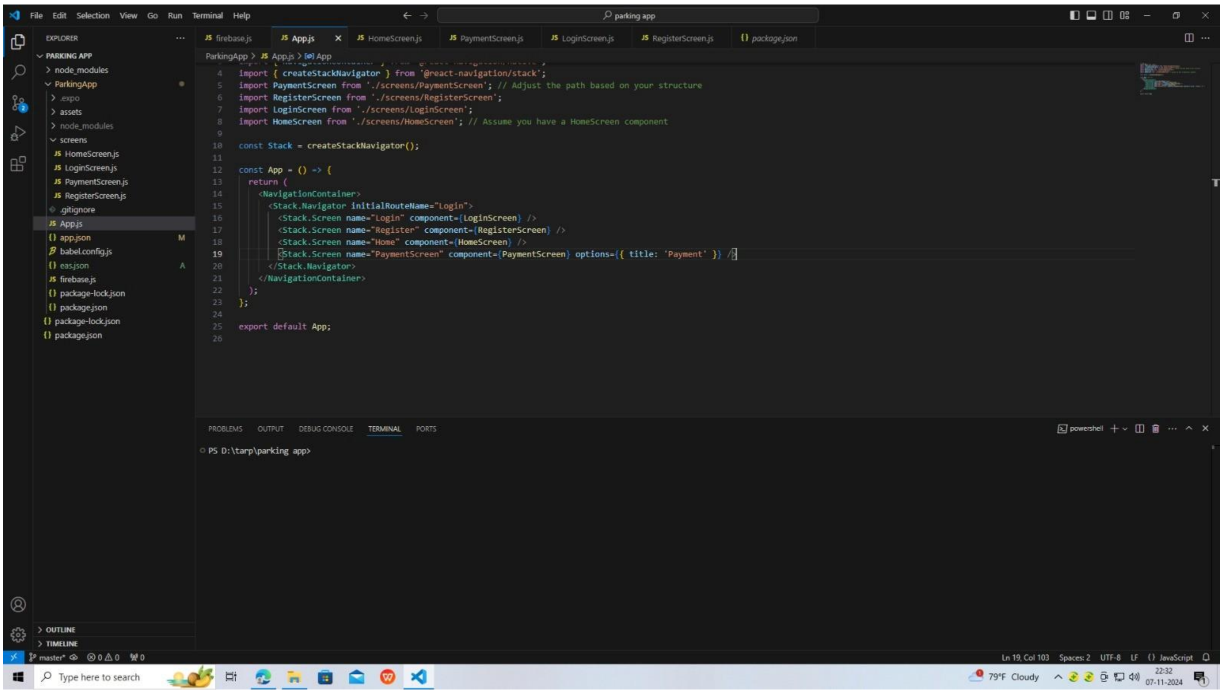
```javascript
import React, { useState } from 'react';
import { View, TextInput, Button, Text, StyleSheet } from 'react-native';
import { auth, firestore } from '../firebase'; // Adjust path based on your structure
import { createUserWithEmailAndPassword } from 'firebase/auth';
import { doc, setDoc } from 'firebase/firestore';

const RegisterScreen = ({ navigation }) => {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [username, setUsername] = useState('');
  const [carNumber, setCarNumber] = useState('');
  const [phoneNumber, setPhoneNumber] = useState('');

  const handleRegister = async () => {
    try {
      const userCredential = await createUserWithEmailAndPassword(auth, email, password);
      const user = userCredential.user;

      // Store additional user info in Firestore
      await setDoc(doc(firestore, "users", user.uid), {
        username,
        carNumber,
        phoneNumber,
      });

      navigation.navigate('Home'); // Navigate to the Home screen after successful registration
    } catch (error) {
      console.error("Registration error:", error);
    }
```

```
PS D:\tarp\parking app> 
```



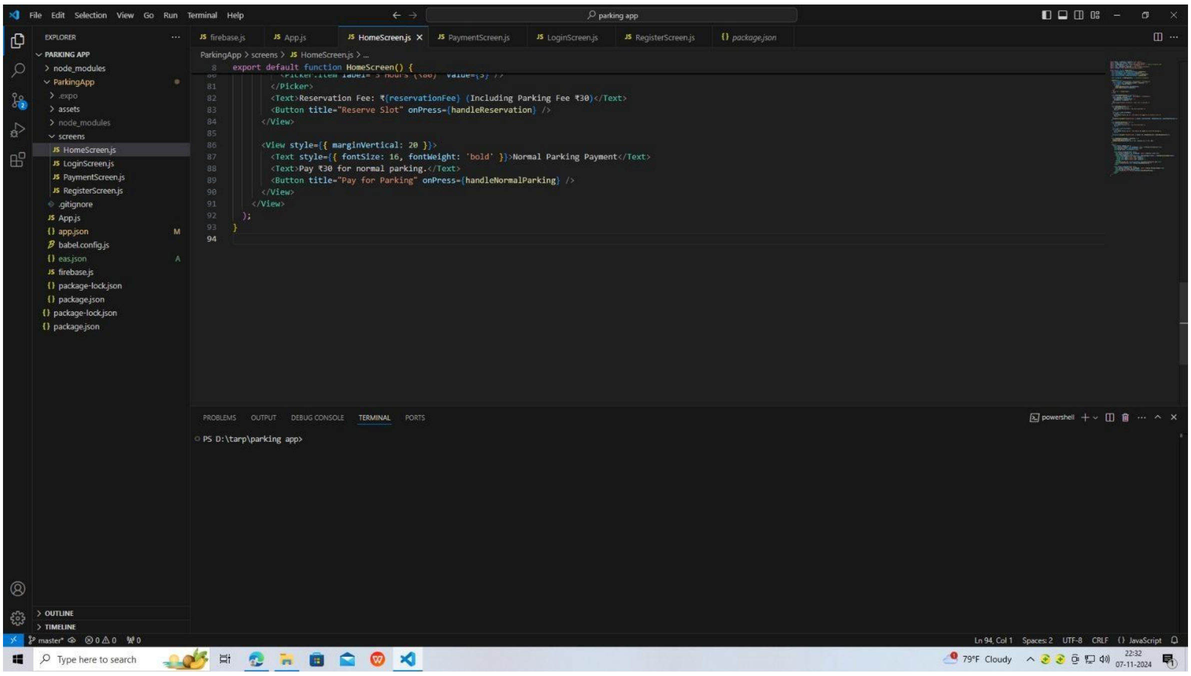
```javascript

// firebase.js
import { initializeApp } from "firebase/app";
import { initializeAuth, getReactNativePersistence } from 'firebase/auth';
import ReactNativeAsyncStorage from '@react-native-async-storage/async-storage';
import { getFirestore } from "firebase/firestore";

const firebaseConfig = {
    apiKey: "AIzaSyDIx2bi_LmUBonUCUyRmLBtbiY9dSBz8So",
    authDomain: "parking-8f5fc.firebaseapp.com",
    projectId: "parking-8f5fc",
    storageBucket: "parking-8f5fc.firebasestorage.app",
    messagingSenderId: "404473434380",
    appId: "1:404473434380:web:019dbae438205e6bb23929",
    measurementId: "G-K5NDZQ2PC8"
  };


const app = initializeApp(firebaseConfig);

// Initialize Firebase Auth with AsyncStorage for persistence
const auth = initializeAuth(app, {
  persistence: getReactNativePersistence(ReactNativeAsyncStorage),
});

const firestore = getFirestore(app);

export { auth, firestore };
```

```
PS D:\tarp\parking app>
```



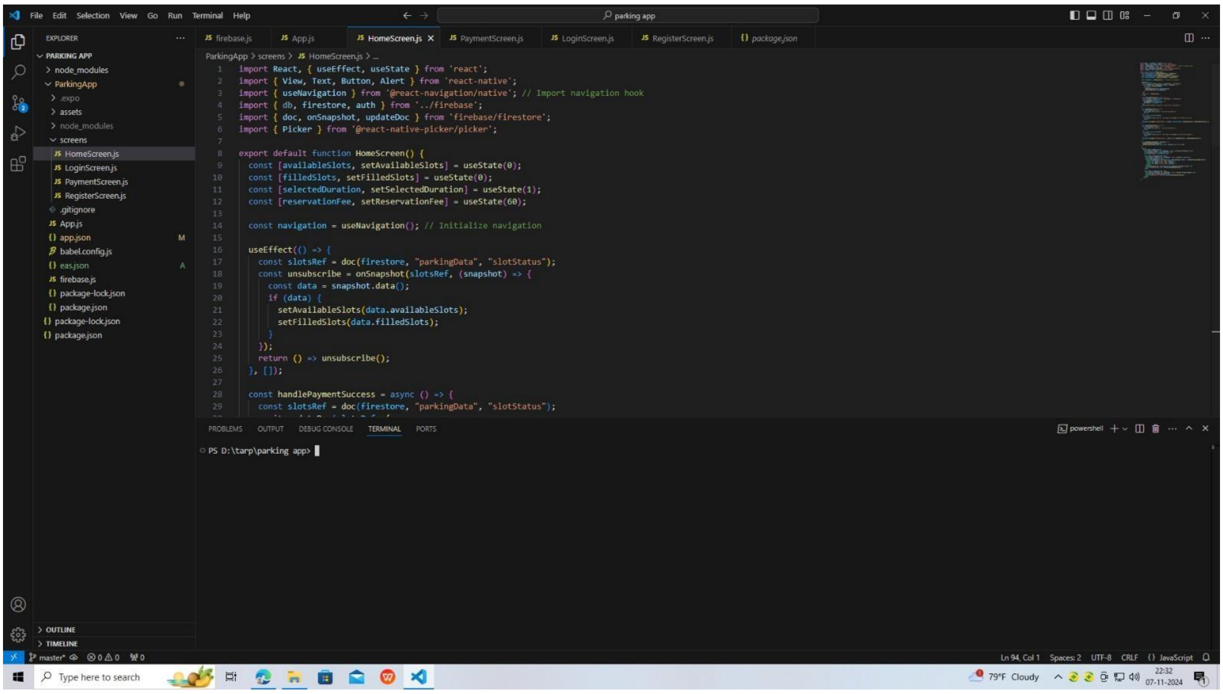
```
import { createStackNavigator } from '@react-navigation/stack';
import PaymentScreen from './screens/PaymentScreen'; // Adjust the path based on your structure
import RegisterScreen from './screens/RegisterScreen';
import LoginScreen from './screens/LoginScreen';
import HomeScreen from './screens/HomeScreen'; // Assume you have a HomeScreen component

const Stack = createStackNavigator();

const App = () => {
  return (
    <NavigationContainer>
      <Stack.Navigator initialRouteName="Login">
        <Stack.Screen name="Login" component={LoginScreen} />
        <Stack.Screen name="Register" component={RegisterScreen} />
        <Stack.Screen name="Home" component={HomeScreen} />
        <Stack.Screen name="PaymentScreen" component={PaymentScreen} options={{ title: 'Payment' }} />
      </Stack.Navigator>
    </NavigationContainer>
  );
};

export default App;
```

```
PS D:\tarp\parking app>
```



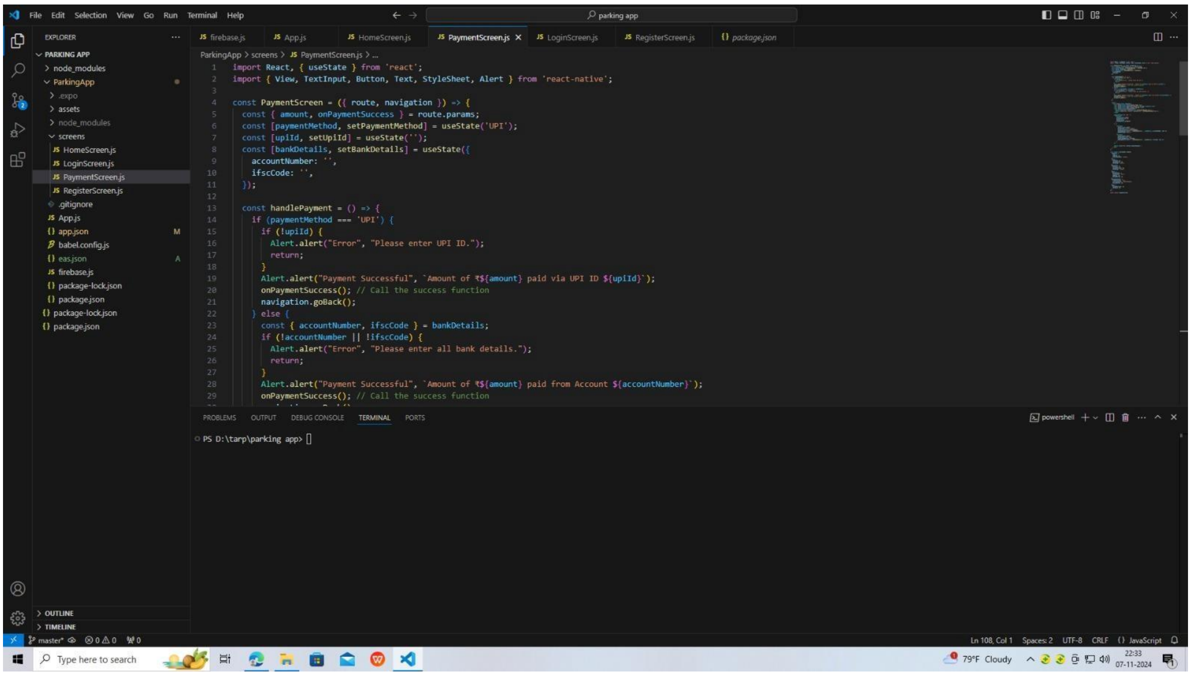
```
export default function HomeScreen() {
        </Picker>
        <Text>Reservation Fee: ₹{reservationFee} (Including Parking Fee ₹30)</Text>
        <Button title="Reserve Slot" onPress={handleReservation} />
      </View>

      <View style={{ marginVertical: 20 }}>
        <Text style={{ fontSize: 16, fontWeight: 'bold' }}>Normal Parking Payment</Text>
        <Text>Pay ₹30 for normal parking.</Text>
        <Button title="Pay for Parking" onPress={handleNormalParking} />
      </View>
    </View>
  );
}
```

```
PS D:\tarp\parking app>
```



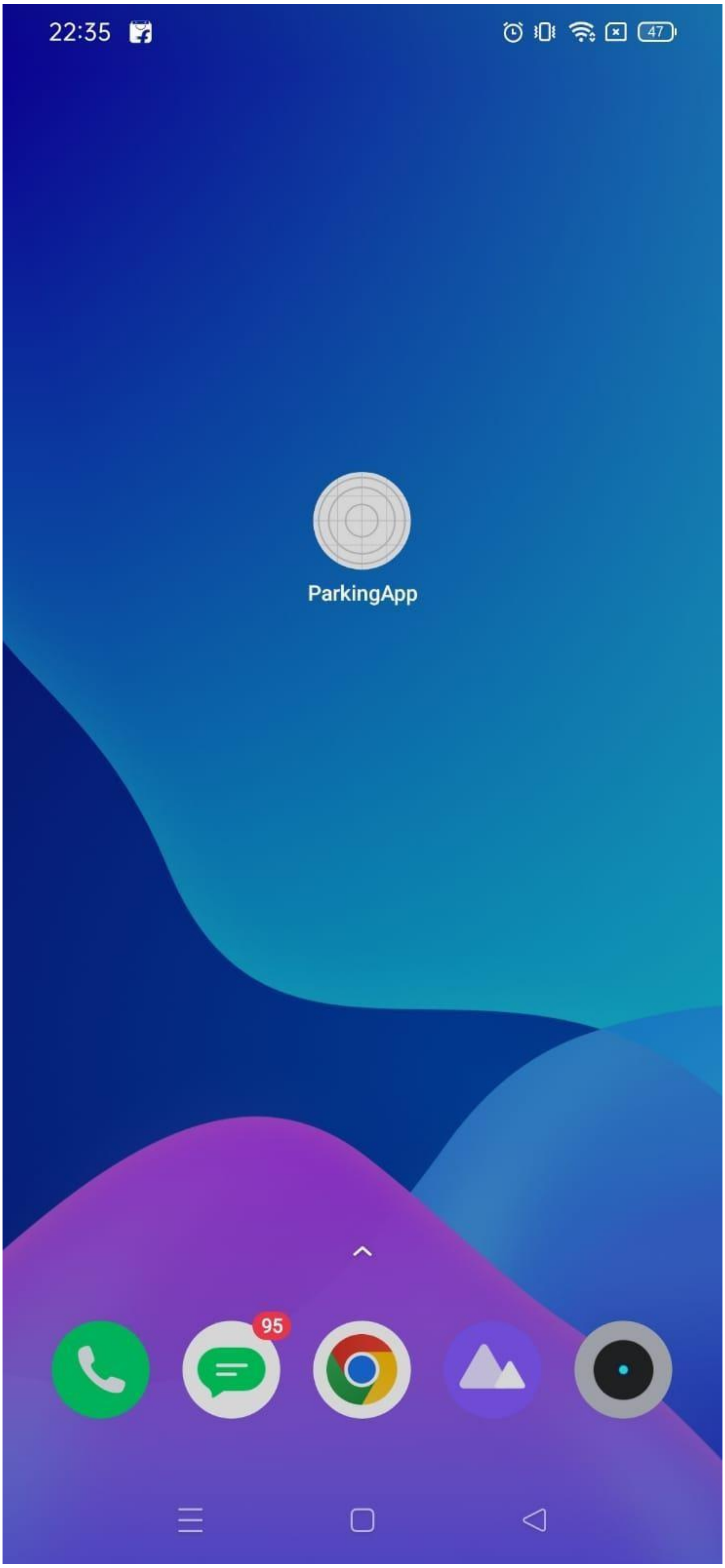
```javascript
import React, { useEffect, useState } from 'react';
import { View, Text, Button, Alert } from 'react-native';
import { useNavigation } from '@react-navigation/native'; // Import navigation hook
import { db, firestore, auth } from '../firebase';
import { doc, onSnapshot, updateDoc } from 'firebase/firestore';
import { Picker } from '@react-native-picker/picker';

export default function HomeScreen() {
  const [availableSlots, setAvailableSlots] = useState(0);
  const [filledSlots, setFilledSlots] = useState(0);
  const [selectedDuration, setSelectedDuration] = useState(1);
  const [reservationFee, setReservationFee] = useState(60);

  const navigation = useNavigation(); // Initialize navigation

  useEffect(() => {
    const slotsRef = doc(firestore, "parkingData", "slotStatus");
    const unsubscribe = onSnapshot(slotsRef, (snapshot) => {
      const data = snapshot.data();
      if (data) {
        setAvailableSlots(data.availableSlots);
        setFilledSlots(data.filledSlots);
      }
    });
    return () => unsubscribe();
  }, []);

  const handlePaymentSuccess = async () => {
    const slotsRef = doc(firestore, "parkingData", "slotStatus");
```

```
PS D:\tarp\parking app>
```

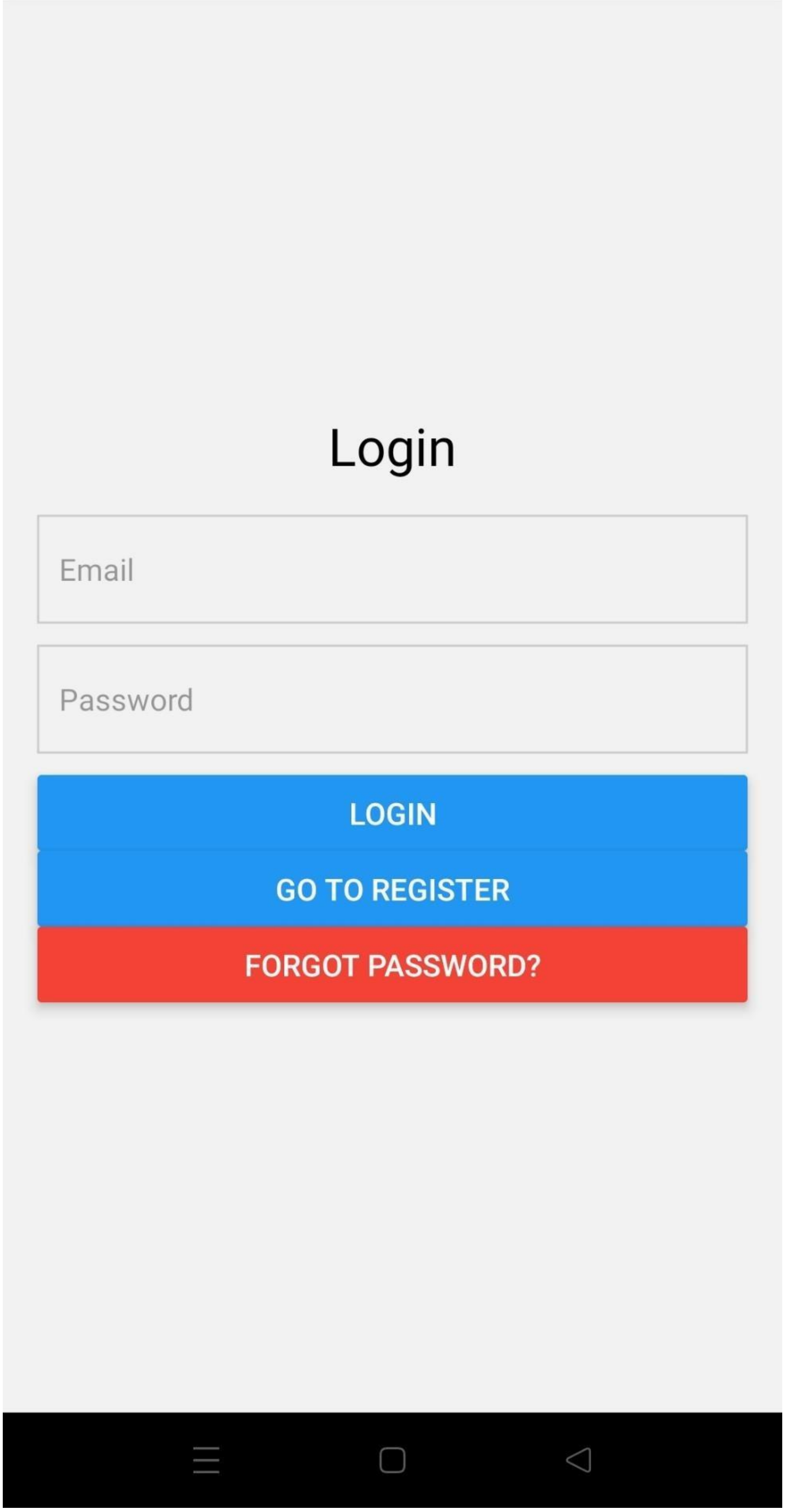


```javascript
import React, { useState } from 'react';
import { View, TextInput, Button, Text, StyleSheet, Alert } from 'react-native';

const PaymentScreen = ({ route, navigation }) => {
  const { amount, onPaymentSuccess } = route.params;
  const [paymentMethod, setPaymentMethod] = useState('UPI');
  const [upiId, setUpiId] = useState('');
  const [bankDetails, setBankDetails] = useState({
    accountNumber: '',
    ifscCode: '',
  });

  const handlePayment = () => {
    if (paymentMethod === 'UPI') {
      if (!upiId) {
        Alert.alert("Error", "Please enter UPI ID.");
        return;
      }
      Alert.alert("Payment Successful", `Amount of ₹${amount} paid via UPI ID ${upiId}`);
      onPaymentSuccess(); // Call the success function
      navigation.goBack();
    } else {
      const { accountNumber, ifscCode } = bankDetails;
      if (!accountNumber || !ifscCode) {
        Alert.alert("Error", "Please enter all bank details.");
        return;
      }
      Alert.alert("Payment Successful", `Amount of ₹${amount} paid from Account ${accountNumber}`);
      onPaymentSuccess(); // Call the success function
```

```
PS D:\tarp\parking app>
```


22:35
47
ParkingApp
95

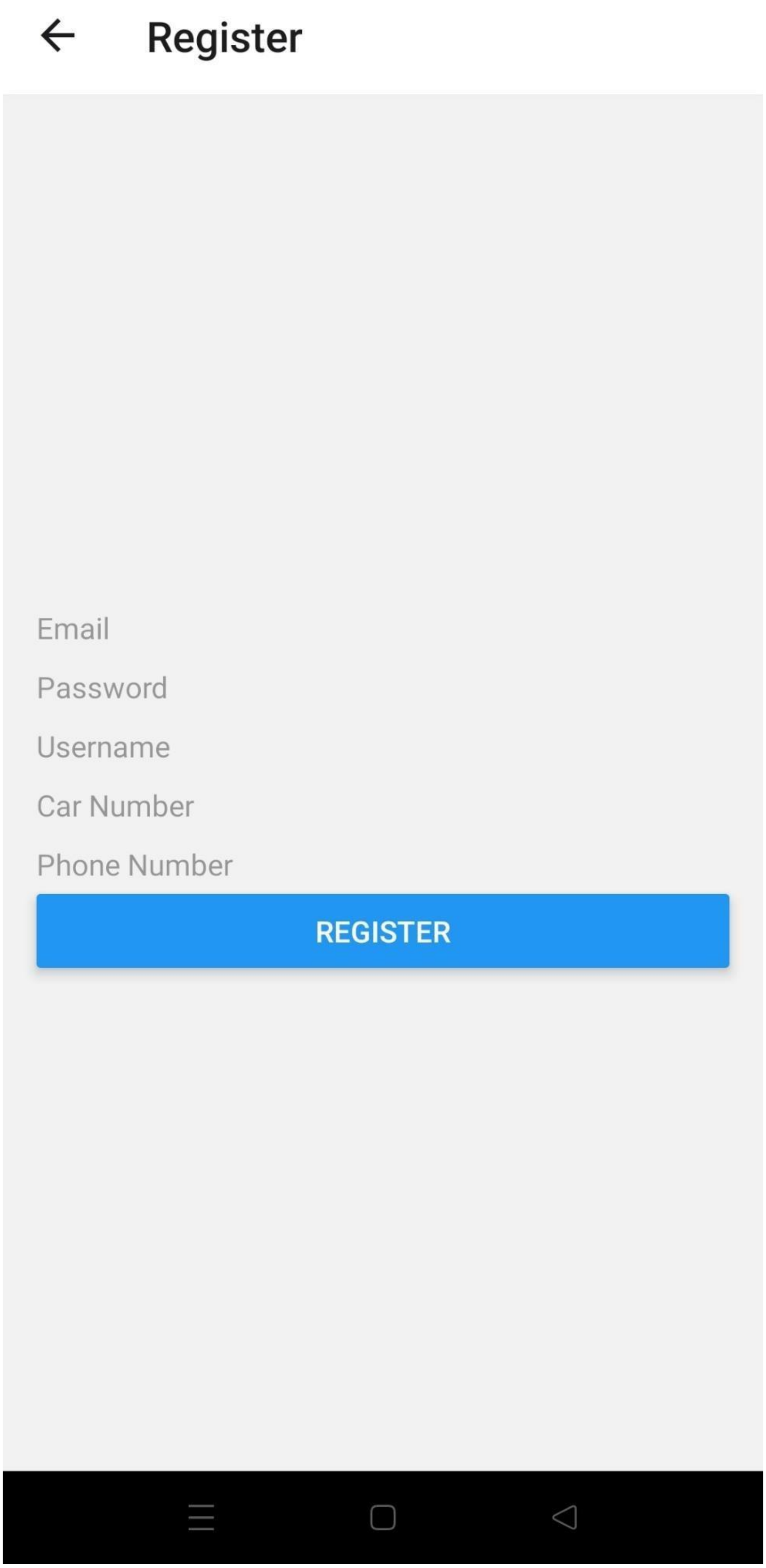
Login



Login

Email

Password



Register

Email

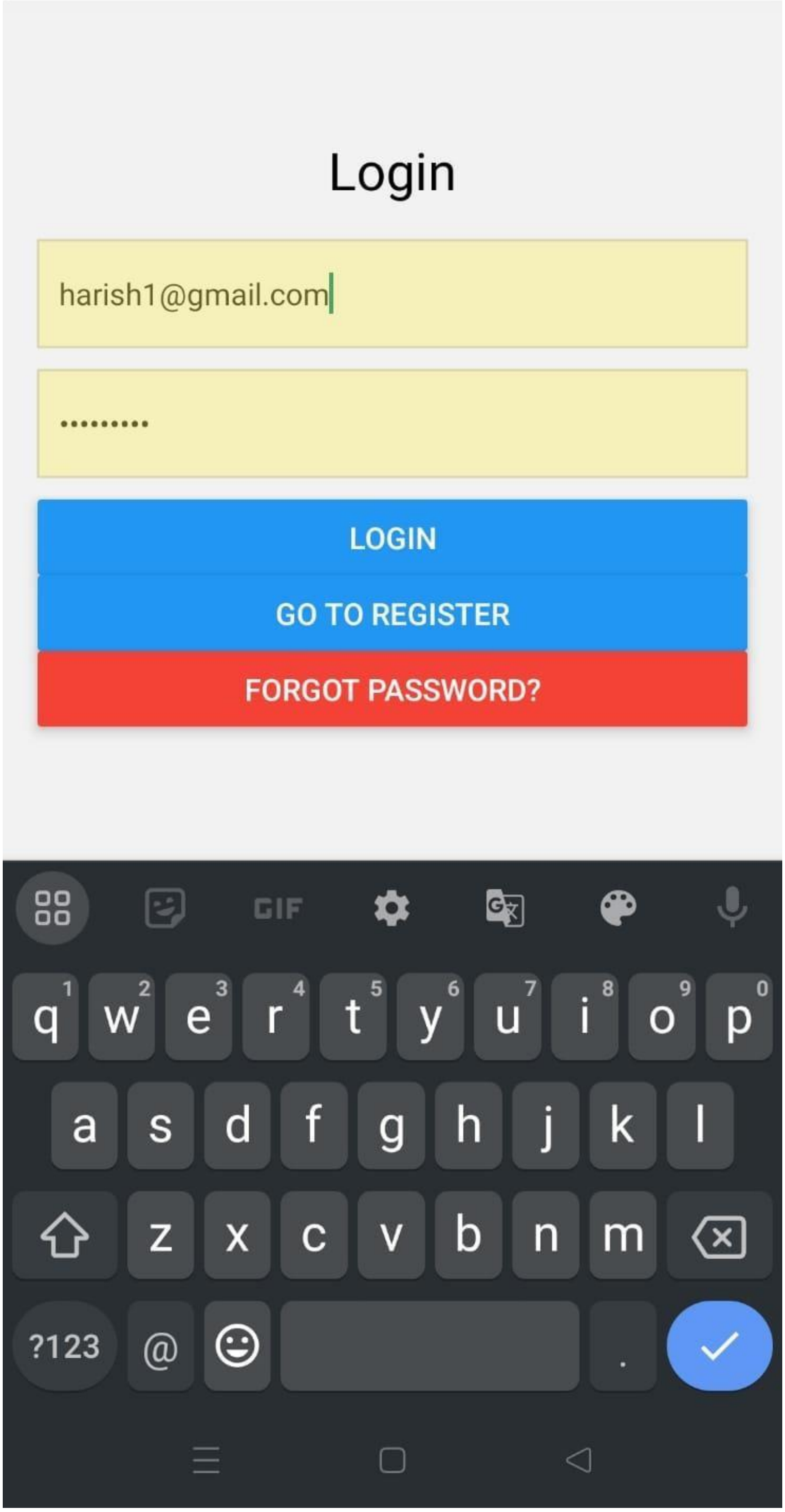
Password

Username

Car Number

Phone Number

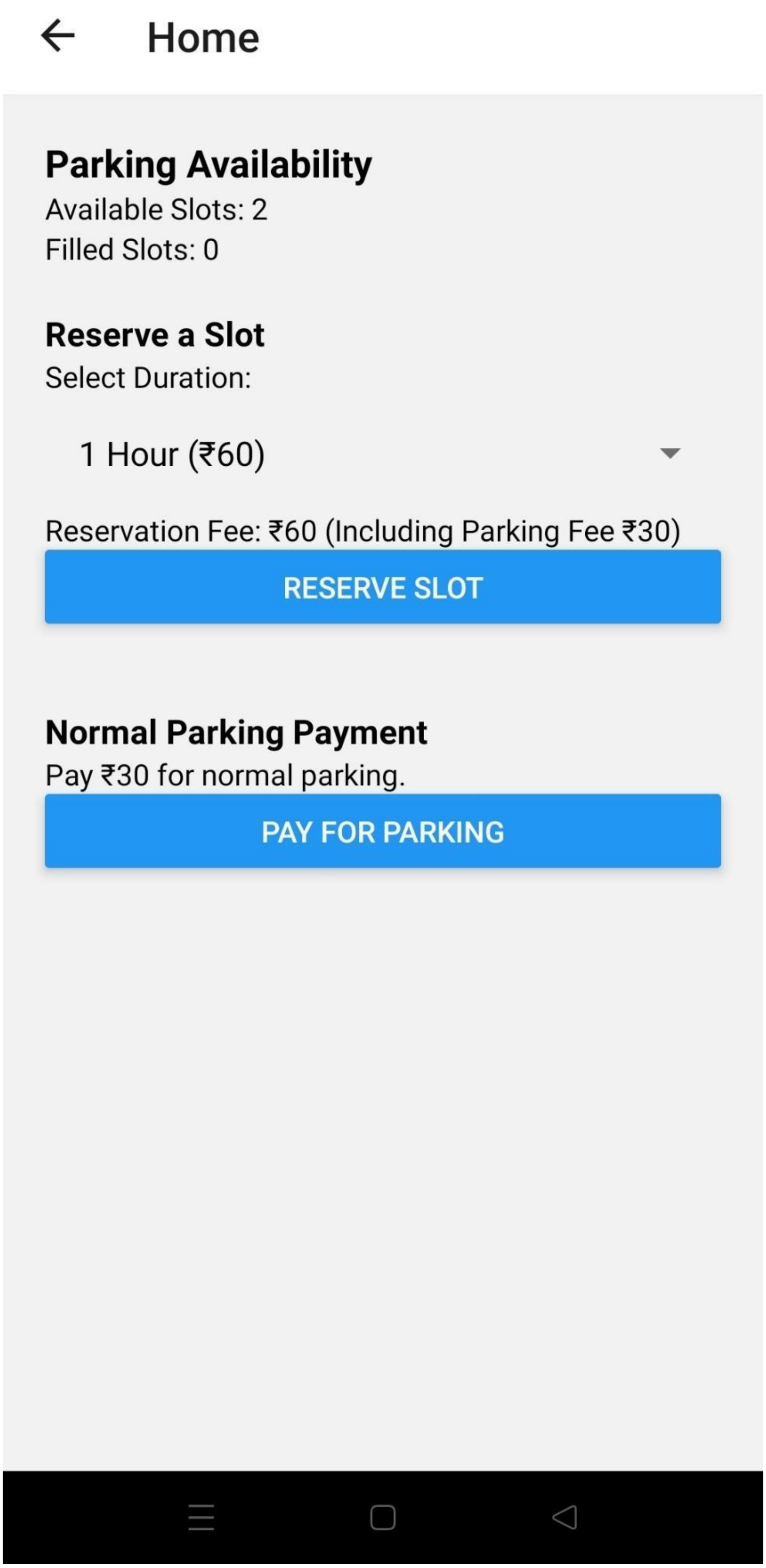
Login



Login

harish1@gmail.com

•••••••••



Home

Parking Availability

Available Slots: 2

Filled Slots: 0

Reserve a Slot

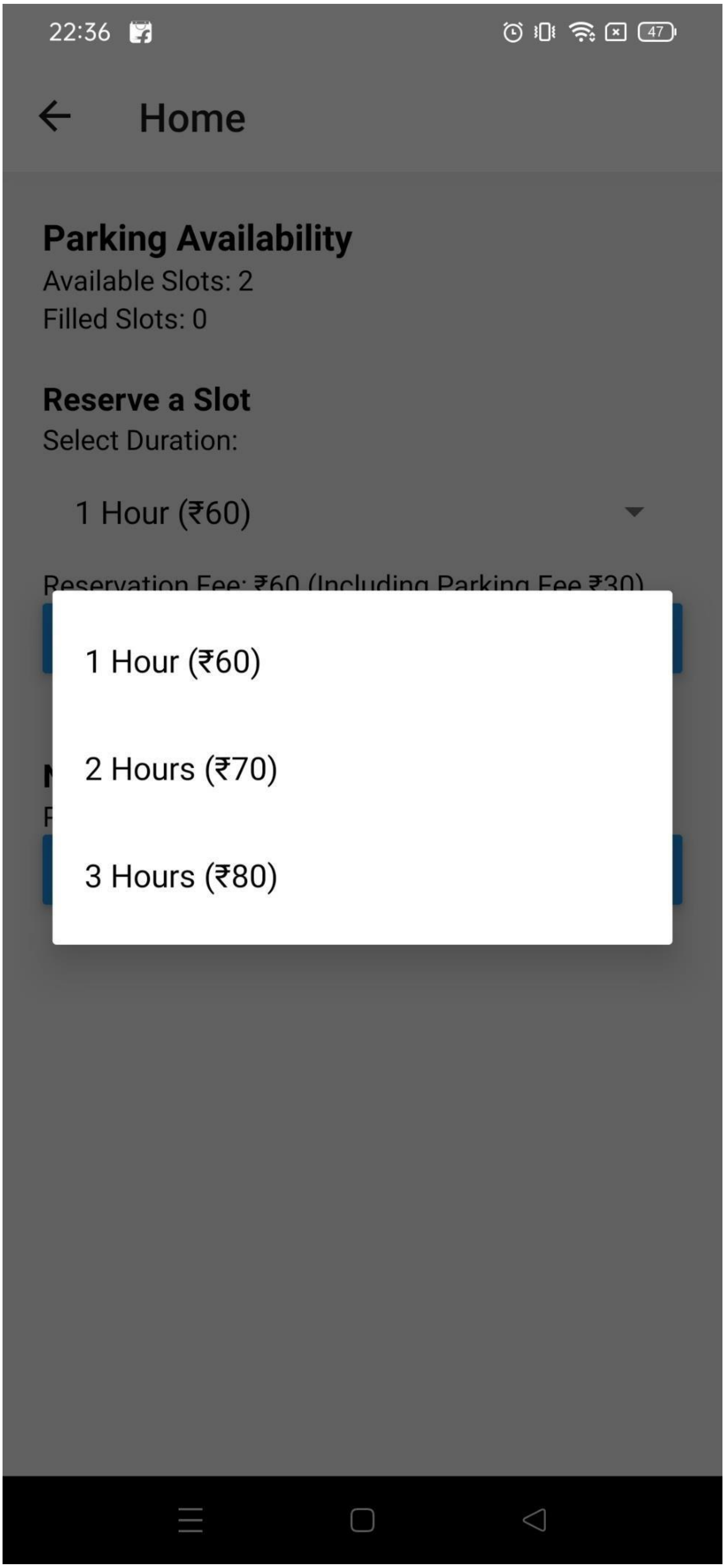
Select Duration:

1 Hour (₹60)

Reservation Fee: ₹60 (Including Parking Fee ₹30)

Normal Parking Payment

Pay ₹30 for normal parking.



Home

Parking Availability

Available Slots: 2

Filled Slots: 0

Reserve a Slot

Select Duration:

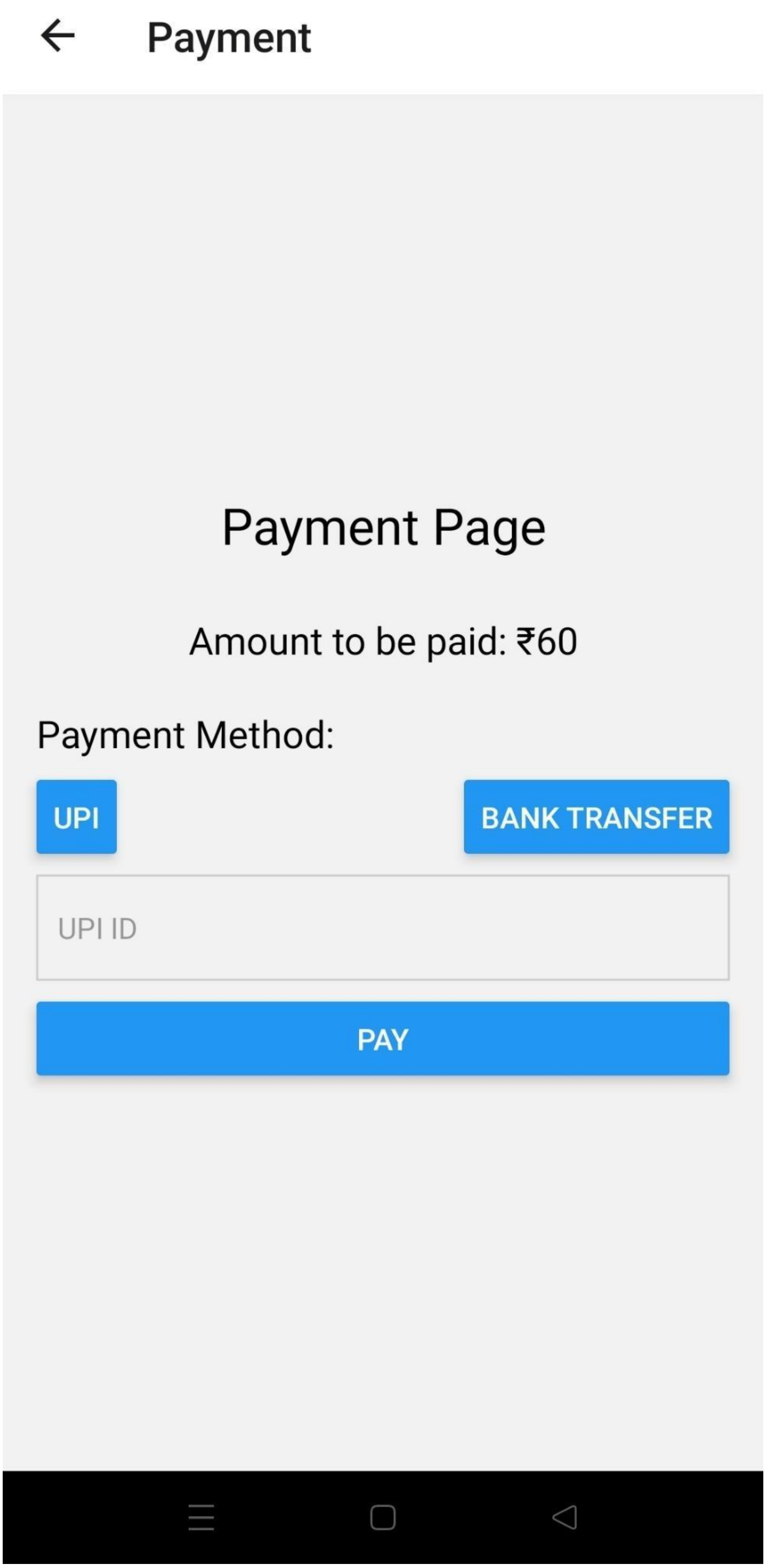
1 Hour (₹60)

Reservation Fee: ₹60 (Including Parking Fee ₹30)

1 Hour (₹60)

2 Hours (₹70)

3 Hours (₹80)

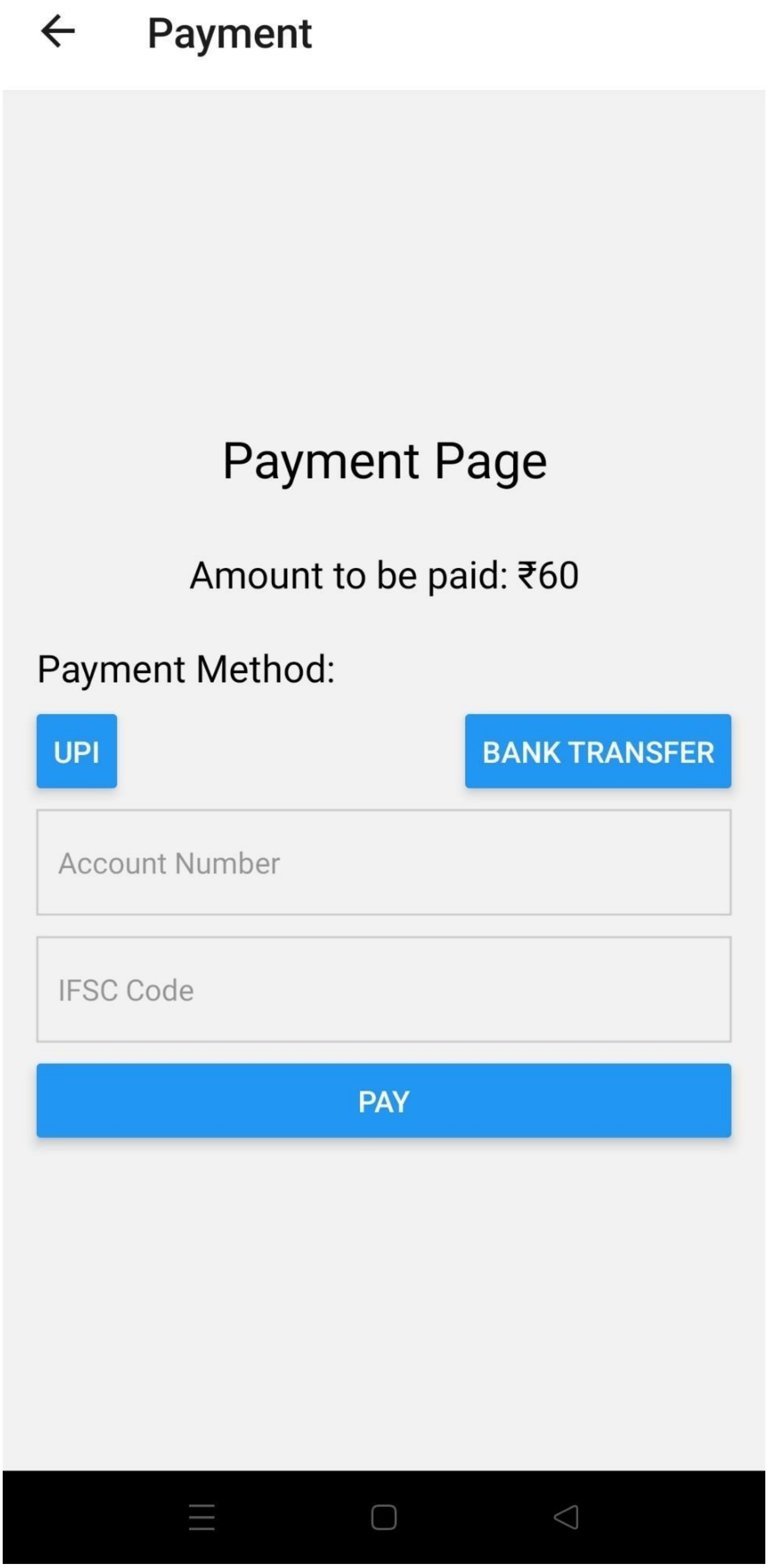


Payment

Payment Page

Amount to be paid: ₹60

Payment Method:



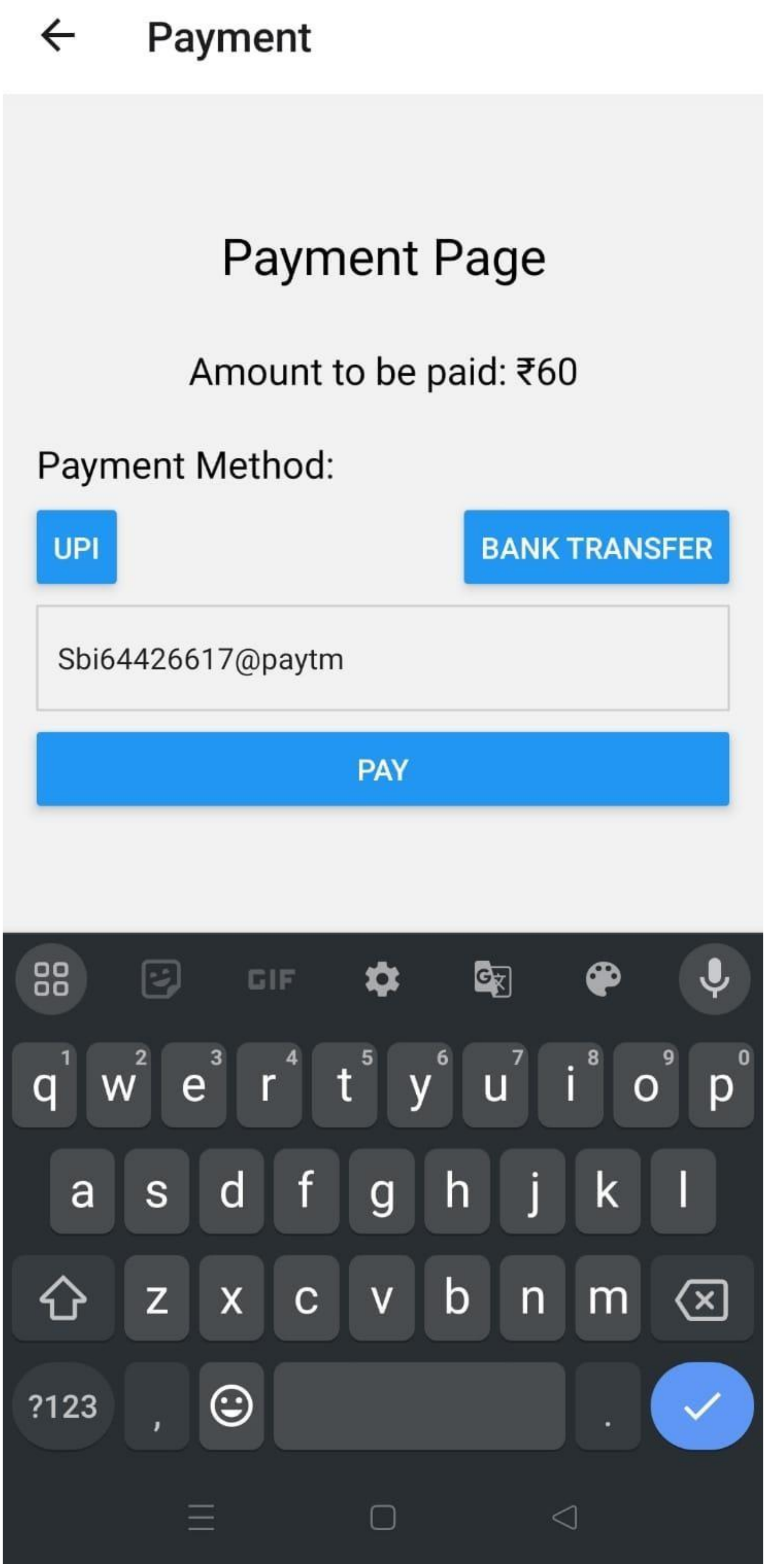
Payment Page

Amount to be paid: ₹60

Payment Method:

Account Number

IFSC Code



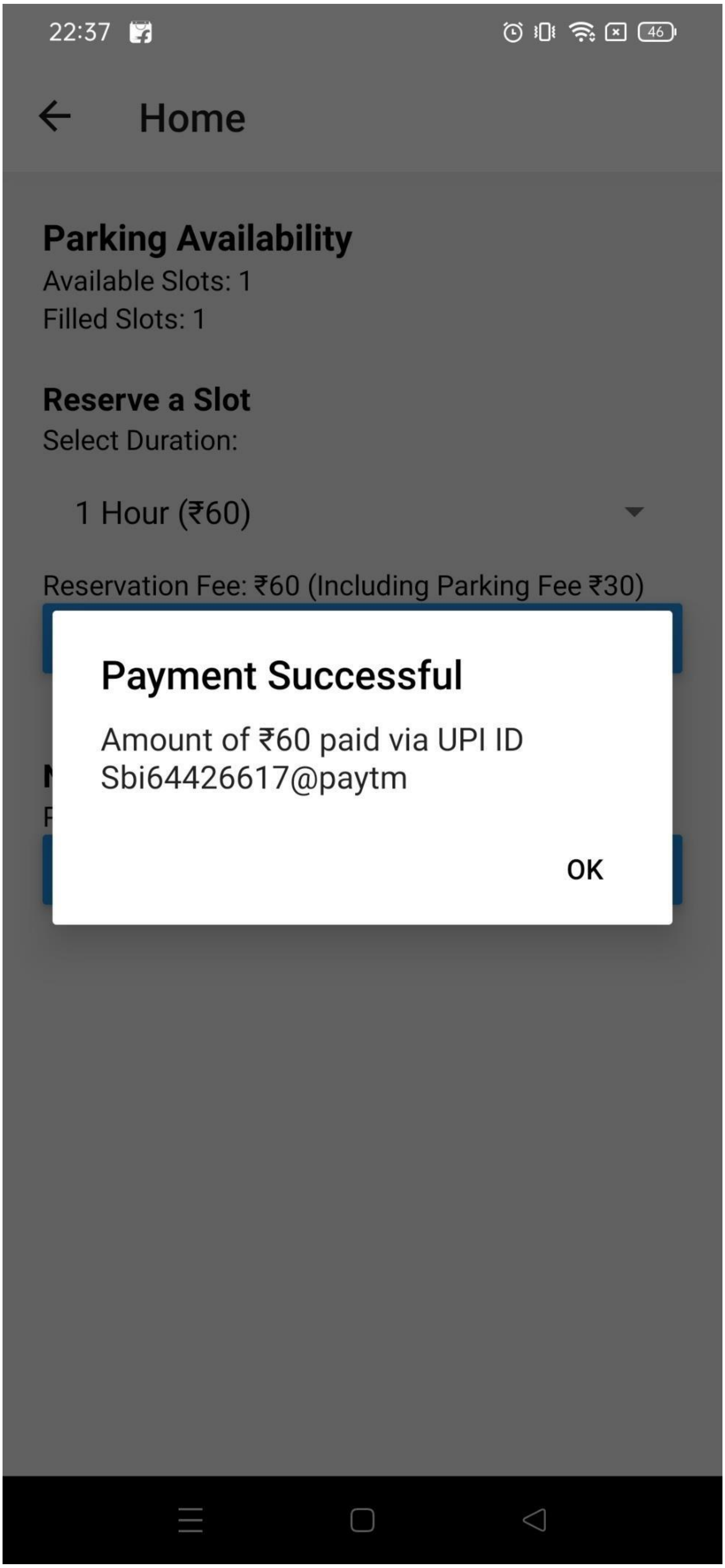
Payment

Payment Page

Amount to be paid: ₹60

Payment Method:

Sbi64426617@paytm



Parking Availability

Available Slots: 1

Filled Slots: 1

Reserve a Slot

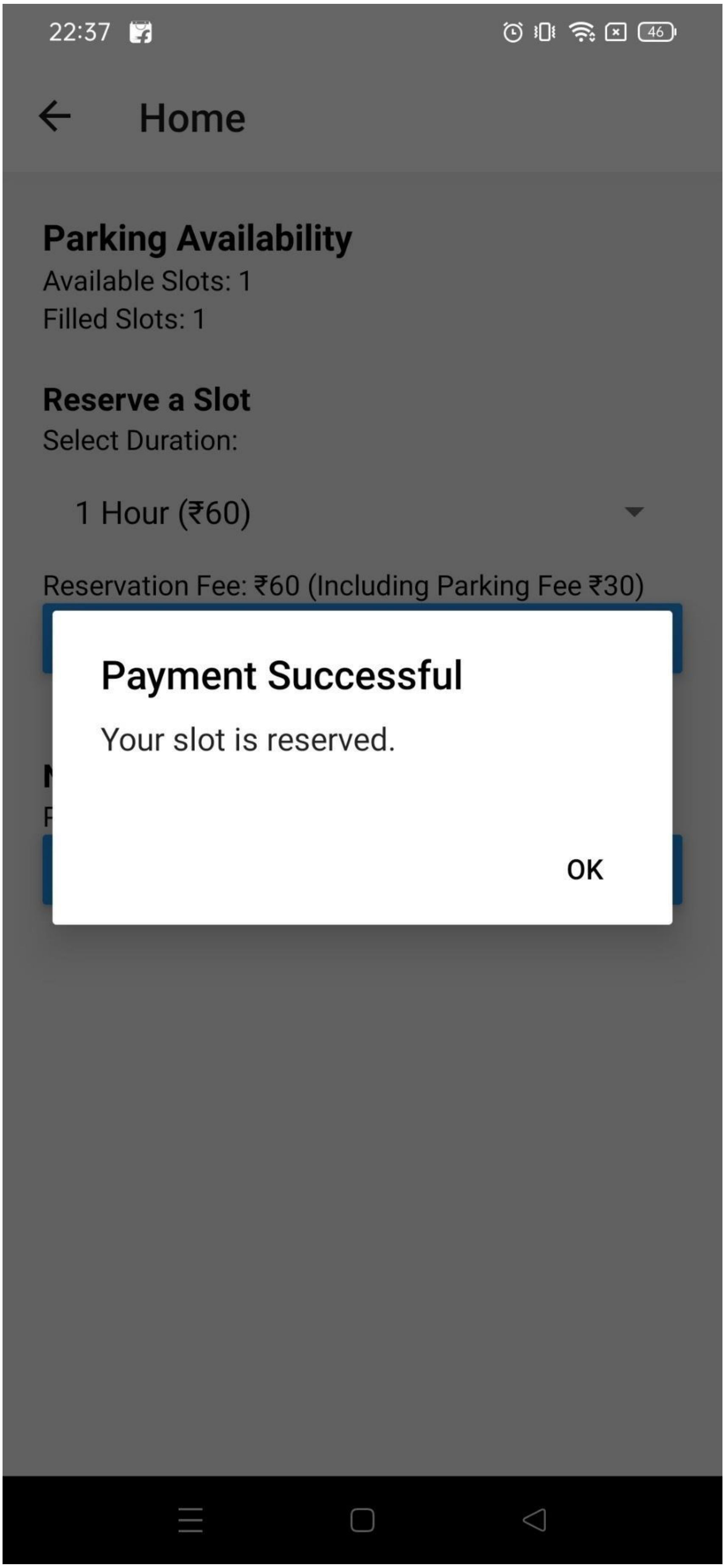
Select Duration:

1 Hour (₹60)

Reservation Fee: ₹60 (Including Parking Fee ₹30)

Payment Successful

Amount of ₹60 paid via UPI ID Sbi64426617@paytm



22:37

Home

Parking Availability

Available Slots: 1

Filled Slots: 1

Reserve a Slot

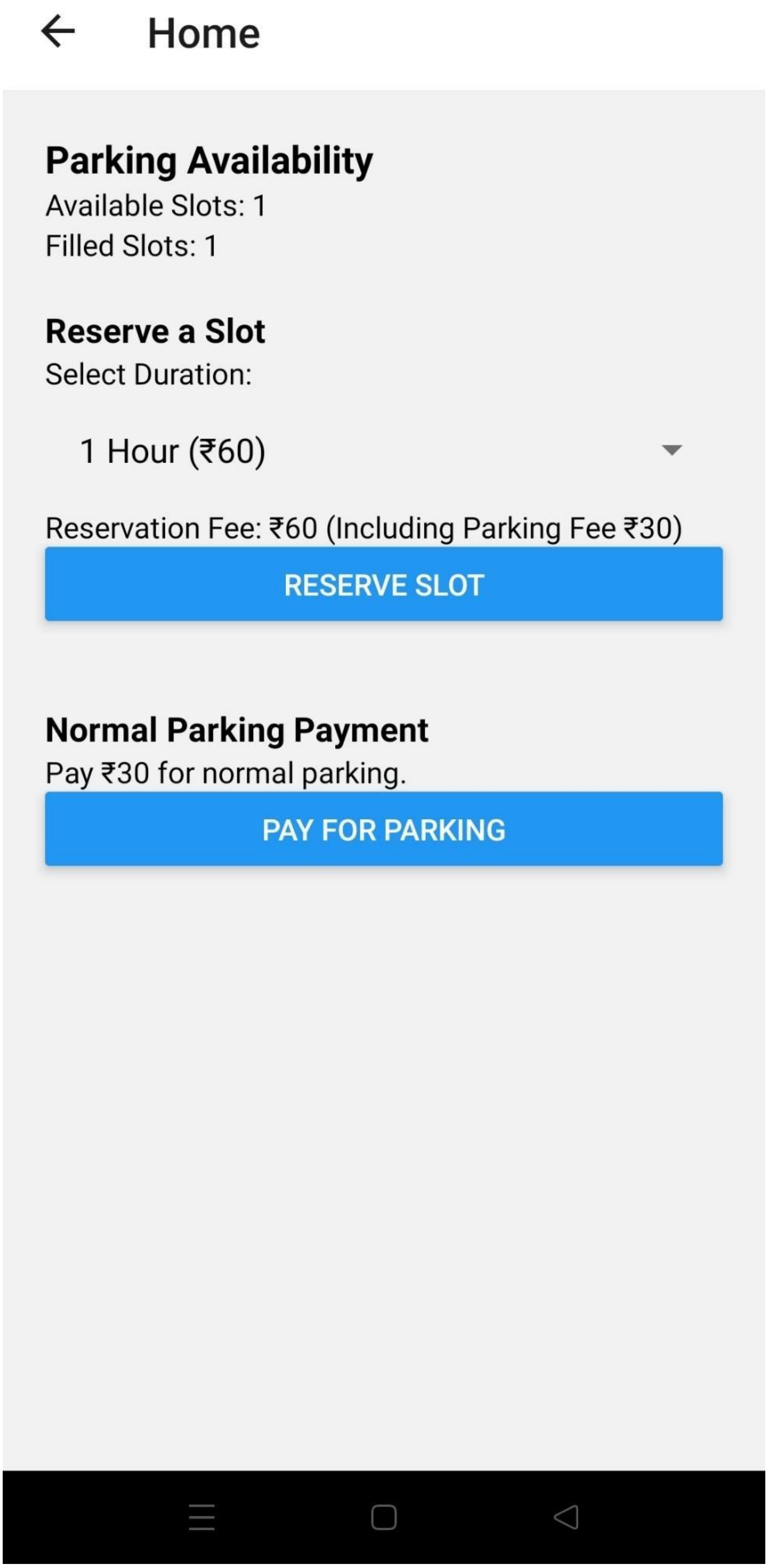
Select Duration:

1 Hour (₹60)

Reservation Fee: ₹60 (Including Parking Fee ₹30)

Payment Successful

Your slot is reserved.



Home

Parking Availability

Available Slots: 1

Filled Slots: 1

Reserve a Slot

Select Duration:

1 Hour (₹60)

Reservation Fee: ₹60 (Including Parking Fee ₹30)

Normal Parking Payment

Pay ₹30 for normal parking.