Develop a React Native application that features a digital clock displaying the current time in 24 hours, with the ability to switch between dark and light themes. Additionally, implement a timer with popup features for the application.

* Describe the initial steps to set up a new React Native project
* Describe how you would create a digital clock that displays the current time in hours, minutes, and seconds format.
* Explain how to implement dark and light theme functionality in the React Native application.
* Explain how to use components like Modal to display the timer popup.

Step - 1: Run **npx create-expo-app digitalClock**

Step-2: Run **cd digitalClock**

**App.js**

import { StatusBar } from 'expo-status-bar';

import { StyleSheet, Text, View, Modal, TouchableOpacity } from 'react-native';

import { useState, useEffect } from 'react';

export default function App() {

// Theme Toggle

const [isButtonClicked, setIsButtonClicked] = useState(false);

const toggleStyles = () => {

setIsButtonClicked(!isButtonClicked);

};

const containerStyles = {

backgroundColor: isButtonClicked ? 'white' : 'black',

};

const textStyles = {

color: isButtonClicked ? 'black' : 'white',

};

const btnStyles = {

color: isButtonClicked ? 'white' : 'black',

};

const btnBgStyles = {

backgroundColor: isButtonClicked ? 'black' : 'white',

};

// Date and Time Display

const [currentTime, setCurrentTime] = useState(new Date());

const [currentDate, setCurrentDate] = useState(new Date());

useEffect(() => {

const intervalId = setInterval(() => {

setCurrentTime(new Date());

}, 1000);

return () => {

clearInterval(intervalId);

};

}, []);

const formattedTime = currentTime.toLocaleTimeString('en-US', { hour12: false });

// Set the timezone to 'Asia/Kolkata' (India's timezone)

const options = {

timeZone: 'Asia/Kolkata',

weekday: 'long',

year: 'numeric',

month: 'long',

day: 'numeric',

};

const formattedDate = currentDate.toLocaleDateString('en-US', options);

// Modal and Timer

const [modalVisible, setModalVisible] = useState(false);

const [timer, setTimer] = useState(0);

const [isRunning, setIsRunning] = useState(false);

const startTimer = () => {

setIsRunning(true);

};

const stopTimer = () => {

setIsRunning(false);

};

const cancelTimer = () => {

setIsRunning(false);

setTimer(0);

setModalVisible(false);

};

useEffect(() => {

let interval;

if (isRunning) {

interval = setInterval(() => {

setTimer((prevTimer) => prevTimer + 1);

}, 1000);

} else {

clearInterval(interval);

}

return () => {

clearInterval(interval);

};

}, [isRunning]);

return (

<View style={[styles.container, containerStyles]}>

<Text style={[styles.time, textStyles]}>{formattedTime}</Text>

<Text style={[styles.date, textStyles]}>{formattedDate}</Text>

<TouchableOpacity onPress={() => setModalVisible(true)}>

<Text style={[styles.timer, btnStyles, btnBgStyles]}>Timer</Text>

</TouchableOpacity>

<TouchableOpacity onPress={toggleStyles} >

<Text style={[styles.toggle, btnStyles, btnBgStyles]}>Toggle Colors </Text>

</TouchableOpacity>

<Modal visible={modalVisible} animationType="slide">

<View style={styles.modal}>

<Text style={styles.time1}>Timer: {timer} seconds</Text>

{isRunning ? (

<TouchableOpacity onPress={stopTimer}>

<Text style={styles.stop}>Stop</Text>

</TouchableOpacity>

) : (

<TouchableOpacity onPress={startTimer}>

<Text style={styles.start}>Start</Text>

</TouchableOpacity>

)}

<TouchableOpacity onPress={cancelTimer}>

<Text style={styles.cancel}>Cancel</Text>

</TouchableOpacity>

</View>

</Modal>

<StatusBar style="auto" />

</View>

);

}

const styles = StyleSheet.create({

container: {

flex: 1,

alignItems: 'center',

justifyContent: 'center',

},

time: {

fontFamily: 'monospace',

fontSize: 50,

},

time1: {

fontFamily: 'monospace',

fontSize: 30,

color: '#fff',

},

date: {

fontFamily: 'monospace',

fontSize: 20,

},

timer: {

margin: 20,

fontSize: 20,

padding: 5,

width: 100,

textAlign: 'center',

borderRadius: 20,

fontFamily: 'monospace',

},

toggle: {

fontSize: 20,

padding: 5,

width: 200,

textAlign: 'center',

borderRadius: 20,

fontFamily: 'monospace',

},

modal: {

flex: 1,

justifyContent: 'center',

alignItems: 'center',

backgroundColor: '#000',

},

start: {

color: '#000',

backgroundColor: '#fff',

margin: 20,

fontSize: 20,

padding: 5,

width: 100,

textAlign: 'center',

borderRadius: 20,

fontFamily: 'monospace',

},

stop: {

color: '#000',

backgroundColor: '#fff',

margin: 20,

fontSize: 20,

padding: 5,

width: 100,

textAlign: 'center',

borderRadius: 20,

fontFamily: 'monospace',

},

cancel: {

color: '#000',

backgroundColor: '#fff',

fontSize: 20,

padding: 5,

width: 100,

textAlign: 'center',

borderRadius: 20,

fontFamily: 'monospace',

},

});