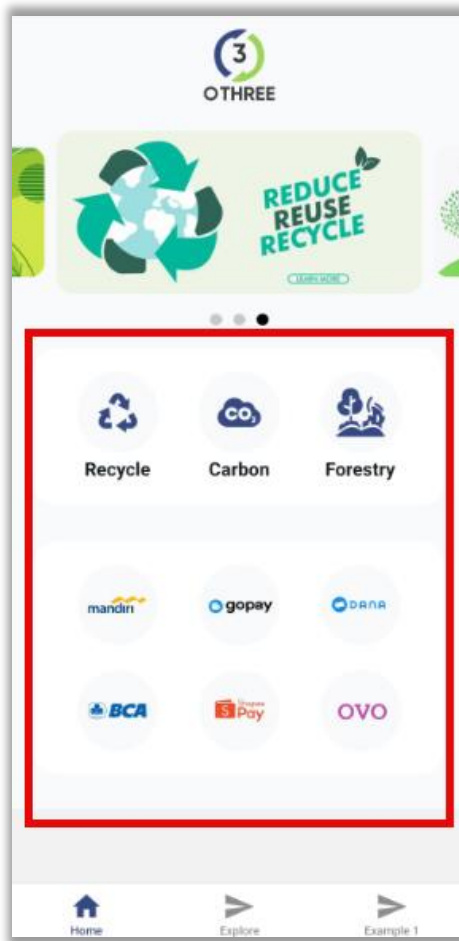


Create Images Icon Group



adjust color configuration in **constants\Colors.ts**:

/**

* Below are the colors that are used in the app. The colors are defined in the light and dark mode.

* There are many other ways to style your app. For example, [Nativewind](https://www.nativewind.dev/), [Tamagui](https://tamagui.dev/), [unistyles](https://reactnativeunistyles.vercel.app), etc.

*/

```
//const tintColorLight = "#0a7ea4";
const tintColorLight = "#374982";
const tintColorDark = "#fff";
```

```
export const Colors = {
  light: {
    text: "#11181C",
    text2: "#374982",
    background: "#F8FAFC",
    background2: "#FFFFFF",
    tint: tintColorLight,
    icon: "#687076",
    tabIconDefault: "#687076",
    //icon: "#198754",
    //tabIconDefault: "#198754",
```

```

        tabIconSelected: tintColorLight,
    },
    dark: {
        text: "#ECEDEE",
        text2: "#ECEDEE",
        background: "#151718",
        background2: "#555",
        tint: tintColorDark,
        icon: "#9BA1A6",
        tabIconDefault: "#9BA1A6",
        //icon: "#198754",
        //tabIconDefault: "#198754",
        tabIconSelected: tintColorDark,
    },
};

```

Edit `app\tabs\index.tsx` with this code:

```

import React, { useState, useEffect } from "react";

import { View, StyleSheet, ScrollView, Dimensions } from "react-native";
import { Image } from "expo-image";
import { ThemedView } from "@components/ThemedView";
import { ThemedText } from "@components/ThemedText";
import { SafeAreaView } from "react-native-safe-area-context";
import axios from "axios";
import Carousel, {
    ICarouselInstance,
    Pagination,
} from "react-native-reanimated-carousel";
import { useSharedValue } from "react-native-reanimated";
import { useColorScheme } from "@hooks/useColorScheme";
import { Colors } from "@constants/Colors";

//get display width
const width = Dimensions.get("window").width;

export default function HomeScreen() {
    //define empty array variable for images
    const [banners, setBanners] = useState([]);

    //set themes color according to device themes
    const colorScheme = useColorScheme();
    const themeCarousel =
        colorScheme === "light" ? styles.carouselLight : styles.carouselDark;

    const themeBG = colorScheme === "light" ? styles.bgLight : styles.bgDark;

    //call loadData function only once when the home page appears
    useEffect(() => {
        loadData();
    }, []);

    //get data from server
    const loadData = async () => {
        axios

```

```

    .get(`https://tempdev2.roomie.id/newsadmin/banner`, {})
    .then(function (response) {
        if (response.data.success == true) {
            //set banner variable with data from API
            setBanners(response.data.data);
        }
    });
};

const ref = React.useRef<ICarouselInstance>(null);
const progress = useSharedValue<number>(0);
const onPressPagination = (index: number) => {
    ref.current?.scrollTo({
        count: index - progress.value,
        animated: true,
    });
};

//content of the image banner to be rendered
const renderItem = ({ item }: { item: string }) => {
    return (
        <View
            style={{
                flex: 1,
                justifyContent: "center",
            }}
        >
            <Image
                style={styles.imageBanner}
                source={item}
                contentFit="contain"
            />
        </View>
    );
};

return (
    <SafeAreaView style={styles.saveContainer}>
        <ScrollView>
            <ThemedView style={styles.imageLogoContainer}>
                <Image
                    style={styles.imageLogo}
                    source={require("@/assets/images/logo.svg")}
                    contentFit="contain"
                />
            </ThemedView>

            <View style={[[themeCarousel]]>
                <Carousel
                    ref={ref}
                    width={width}
                    height={width / 2 - 22}
                    data={banners}
                    onProgressChange={progress}
                    renderItem={renderItem}
                    loop={true}
                    autoPlay

```

```

        autoPlayInterval={5000}
        mode="parallax"
    />

    <Pagination.Basic
        progress={progress}
        data={banners}
        dotStyle={{
            backgroundColor: "rgba(0,0,0,0.2)",
            borderRadius: 50,
        }}
        containerStyle={{ gap: 10, marginTop: 0 }}
        onPress={onPressPagination}
    />
</View>

<ThemedView style={styles.mainContainer}>
    <View style={[styles.containerInner, themeBG]}>
        <View style={[styles.container1]}>
            <View style={[styles.icon1, styles.iconTop]}>
                <Image
                    style={[styles.image1, styles.image1b]}
                    source={require("@/assets/images/button/recycle.svg")}
                    contentFit="contain"
                />
            </View>
            <View style={[styles.icon1, styles.iconTop]}>
                <Image
                    style={[styles.image1, styles.image1b]}
                    source={require("@/assets/images/button/carbon.svg")}
                    contentFit="contain"
                />
            </View>
            <View style={[styles.icon1, styles.iconTop]}>
                <Image
                    style={[styles.image1, styles.image1b]}
                    source={require("@/assets/images/button/forestry.svg")}
                    contentFit="contain"
                />
            </View>
        </View>
        <View style={[styles.container2]}>
            <View style={[styles.icon1]}>
                <ThemedText style={[styles.textBold]}>
                    Recycle
                </ThemedText>
            </View>
            <View style={[styles.icon1]}>
                <ThemedText style={[styles.textBold]}>
                    Carbon
                </ThemedText>
            </View>
            <View style={[styles.icon1]}>
                <ThemedText style={[styles.textBold]}>
                    Forestry
                </ThemedText>
            </View>
        </View>
    </View>

```

```

        </View>
    </View>

    <View style={[styles.containerInner, themeBG]}>
        <View style={[styles.container1, { marginBottom: 20 }]}>
            <View style={[styles.icon1, styles.iconTop]}>
                <Image
                    style={[styles.image1, styles.image1b]}
                    source={require("@/assets/images/button/b-mandiri.svg")}
                    contentFit="contain"
                />
            </View>
            <View style={[styles.icon1, styles.iconTop]}>
                <Image
                    style={[styles.image1, styles.image1b]}
                    source={require("@/assets/images/button/b-gopay.svg")}
                    contentFit="contain"
                />
            </View>
            <View style={[styles.icon1, styles.iconTop]}>
                <Image
                    style={[styles.image1, styles.image1b]}
                    source={require("@/assets/images/button/b-dana.svg")}
                    contentFit="contain"
                />
            </View>
        </View>
        <View style={[styles.container1]}>
            <View style={[styles.icon1, styles.iconTop]}>
                <Image
                    style={[styles.image1, styles.image1b]}
                    source={require("@/assets/images/button/b-bca.svg")}
                    contentFit="contain"
                />
            </View>
            <View style={[styles.icon1, styles.iconTop]}>
                <Image
                    style={[styles.image1, styles.image1b]}
                    source={require("@/assets/images/button/b-spay.svg")}
                    contentFit="contain"
                />
            </View>
            <View style={[styles.icon1, styles.iconTop]}>
                <Image
                    style={[styles.image1, styles.image1b]}
                    source={require("@/assets/images/button/b-ovo.svg")}
                    contentFit="contain"
                />
            </View>
        </View>
    </View>
</ThemedView>
</ScrollView>
</SafeAreaView>
    );
}

```

```

const styles = StyleSheet.create({
  saveContainer: {
    flex: 1,
  },
  imageLogoContainer: {
    flex: 1,
    flexDirection: "column",
    alignItems: "center",
    height: 80,
    paddingTop: 10,
    marginBottom: 0,
  },
  imageLogo: {
    width: 80,
    height: 80 * (144 / 165),
  },

  carousellLight: {
    backgroundColor: Colors.light.background,
  },
  carousellDark: { backgroundColor: Colors.dark.background },
  imageBanner: {
    width: "100%",
    height: "100%",
  },

  mainContainer: {
    padding: 20,
    flex: 1,
    flexDirection: "column",
    paddingBottom: 0,
  },

  containerInner: {
    borderRadius: 20,
    overflow: "hidden",
    padding: 20,
    marginBottom: 30,
  },

  bgLight: {
    backgroundColor: Colors.light.background2,
  },
  bgDark: { backgroundColor: Colors.dark.background2 },

  container1: {
    flexDirection: "row",
  },

  icon1: {
    flex: 1,
    alignItems: "center",
    justifyContent: "center",
  },

  iconTop: { height: 70 },
  image1: {

```

```

    flex: 1,
    width: "100%",
  },

  image1b: {},

  container2: {
    padding: 0,
    height: 25,
    marginTop: 0,
    flexDirection: "row",
  },

  textBold: {
    fontWeight: "bold",
  },
});

```

Run project with this command:

npx expo start

Result:

