

app/(home)/(SportTest)/Appointment/index.tsx

```
import React from 'react';
```

```
import { Box } from '@components/ui/box';
```

```
import { Text } from '@components/ui/text';
```

```
export default function Appointment() {
```

```
  return (
```

```
    <Box>
```

```
      <Text>体测预约页面</Text>
```

```
      { /* 这里添加预约表单或日历组件 */ }
```

```
    </Box>
```

```
  );
```

```
}
```

app/(home)/(SportTest)/MyScore/index.tsx

```
import { ScrollView } from '@components/ui/scroll-view';
```

```
import React from 'react';
```

```
import { VStack } from '@components/ui/vstack';
```

```
import ScoreCard from '@components/ScoreCard';
```

```
import { Text } from '@components/ui/text';
```

```
import { Box } from '@components/ui/box';
```

```
import { Center } from '@components/ui/center';
```

```
export default function MyScore() {
```

```
  // 用户信息
```

```
  const userInfo = {
```

```
    name: "张三",
```

```
    id: "1234567890",

    score: 80,

    rankText: "你超过了 80%的人",

    avatarUrl: "https://images.unsplash.com/photo-1620403724159-40363e84a155?q=80&w=2646&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D",

  };
```

// 成绩信息

```
const scoreData = [

  { project: "1000 米", score: 85, weightedScore: 25.5 },

  { project: "引体向上", score: 90, weightedScore: 27 },

  { project: "仰卧起坐", score: 95, weightedScore: 28.5 },

  { project: "立定跳远", score: 88, weightedScore: 26.4 },

];
```

```
return (

  <ScrollView className="rounded-lg overflow-hidden" key="score-container">

    <VStack className="p-4 w-full" key="score-content">

      <Text>2023-2024 春夏学期</Text>

      <ScoreCard userInfo={userInfo} scoreData={scoreData}

key="card1"/>

      <Text>2023-2024 春夏学期</Text>

      <ScoreCard userInfo={userInfo} scoreData={scoreData}

key="card2"/>

    </VStack>

  </ScrollView>
```

```

    );
}

app/(home)/(SportTest)/Record/index.tsx

import React from 'react';

import { Box } from '@components/ui/box';
import { Text } from '@components/ui/text';

export default function Record() {
  return (
    <Box>
      <Text>预约记录页面</Text>
      { /* 这里添加预约历史记录列表 */ }
    </Box>
  );
}

```

```

app/(home)/(SportTest)/Standard/index.tsx

import React from 'react';

import { Box } from '@components/ui/box';
import { VStack } from '@components/ui/vstack';
import { Heading } from '@components/ui/heading';
import StandCard from '@components/StandCard';

export default function Standard() {
  const scores = [
    [

```

```

    { heading: '50 米', score: 10 },
    { heading: '1000 米', score: 10 },
    { heading: '仰卧起坐', score: 10 },
  ],
  [
    { heading: '引体向上', score: 10 },
    { heading: '立定跳远', score: 10 },
    { heading: '肺活量', score: 10 },
  ],
  [
    { heading: '坐位体前屈', score: 10 },
    { heading: '体重指数', score: 10 },
    { heading: '800 米', score: 10 },
  ],
];

```

```

const handlePressItem = (itemHeading: string) => {
  console.log(`${itemHeading} clicked!`);
};

```

```

return (
  <Box>
    <VStack className="p-4 w-full" key="standard-content">
      <Heading className='text-center'>项目权重</Heading>
      <StandCard scores={scores} onPressItem={handlePressItem} />
    </VStack>
  </Box>
)

```

```
);  
}
```

app/(home)/_layout.tsx

```
import { Stack } from 'expo-router';  
import React from 'react';  
import { View } from 'react-native';
```

```
export default function HomeLayout() {  
  return (  
    <Stack>  
      <Stack.Screen name="(SportTest)"  
        options={{  
          title: '体育测试',  
          headerTitleAlign: 'center',  
          headerShown: false  
        }}  
      />  
      <Stack.Screen name="MyCourse"/>  
      <Stack.Screen name="SportTool"/>  
    </Stack>  
  )  
}
```

App/(home)/index.tsx

```
import { Link, Stack } from 'expo-router';  
import React from 'react';
```

```
import { Image, Text, View, StyleSheet } from 'react-native';
```

```
export default function Index() {  
  return (  
    <View style={styles.container}>  
      <Text>Home Screen</Text>  
    </View>  
  );  
}
```

```
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    alignItems: 'center',  
    justifyContent: 'center',  
  },  
  image: {  
    width: 50,  
    height: 50,  
  },  
});
```

App/(home)/MyCourse.tsx

```
import { Link, Stack } from 'expo-router';
```

```
import React from 'react';
```

```
import { Image, Text, View, StyleSheet } from 'react-native';
```

```
export default function Index() {  
  return (  
    <View style={styles.container}>  
      <Text>Home Screen</Text>  
    </View>  
  );  
}
```

```
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    alignItems: 'center',  
    justifyContent: 'center',  
  },  
  image: {  
    width: 50,  
    height: 50,  
  },  
});
```

app/(home)/SportTool.tsx

```
import CourseTag from '@components/CourseTagProps';  
import FeatureItem from '@components/FeatureItem';  
import {PackageMinus, PackagePlus, Settings } from 'lucide-react-native';  
import React from 'react';
```

```

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

export default function SportTool() {
  return (
    <View>
      { /* <Text>SportTest</Text> */ }

      <FeatureItem
        icon={PackagePlus}
        text="器材领用"
        onPress={() => console.log('General Settings pressed')}
      />

      <FeatureItem
        icon={PackageMinus}
        text="器材归还"
        onPress={() => console.log('General Settings pressed')}
      />
    </View>
  );
}

```

app\login\index.tsx

```

import { Toast, ToastTitle, ToastDescription, useToast } from '@components/ui/toast';
import { InputField } from '@components/ui/input';
import React, { useState } from "react";
import { VStack } from "@components/ui/vstack";

```



```
import { Input } from "@components/ui/input";

import { Text } from "@components/ui/text";

import { Button, ButtonText } from "@components/ui/button";

import { FormControl } from "@components/ui/form-control";

import { Heading } from '@components/ui/heading';

import { InputSlot } from '@components/ui/input';

import { InputIcon } from '@components/ui/input';

import { CheckIcon, EyeIcon, EyeOffIcon } from 'lucide-react-native';

import { Checkbox, CheckboxIndicator, CheckboxIcon, CheckboxLabel } from
'@components/ui/checkbox';

import { ImageBackground } from '@components/ui/image-background';

import { HStack } from '@components/ui/hstack';

import { Link, LinkText } from '@components/ui/link';

import { useRouter } from 'expo-router';

const Login = () => {

  const toast = useToast();

  const router = useRouter();

  const [rememberMe, setRememberMe] = useState(true);

  const [agreePolicy, setAgreePolicy] = useState(true);

  const [showPassword, setShowPassword] = useState(false);

  const [username, setUsername] = useState("admin");

  const [password, setPassword] = useState("iot421");

  const handleLogin = () => {

    const TEST_USERNAME = "admin";

    const TEST_PASSWORD = "iot421";

    if (!agreePolicy) {

      console.log("请先同意政策");
```

```

toast.show({
  placement: 'top',
  duration: 2000,
  render: ({
    id
  }) => {
    const toastId = `toast-${id}`;
    return <>
      <Toast nativeID={toastId} action={"warning"}
variant={"outline"}>
        <VStack space="xs">
          <ToastTitle>登陆失败</ToastTitle>
          <ToastDescription>
            请先同意政策
          </ToastDescription>
        </VStack>
      </Toast>
    </>;
  }
});
} else {
  if (username === TEST_USERNAME && password ===
TEST_PASSWORD) {
    console.log("登录成功");
    toast.show({
      placement: 'top',
      duration: 2000,

```

```

        render: ({
            id
        }) => {
            const toastId = `toast-${id}`;

            return <>
                <Toast nativeID={toastId} action={"success"}
variant={"outline"}>

                    <VStack space="xs">

                        <ToastTitle>登录成功</ToastTitle>

                        <ToastDescription>

                            即将跳转首页

                        </ToastDescription>

                    </VStack>

                </Toast>

            </>;
        }
    });

    router.replace('/(tabs)');
} else {
    toast.show({
        placement: 'top',
        duration: 2000,
        render: ({
            id
        }) => {
            const toastId = `toast-${id}`;

            return <>

```

```

        <Toast nativeID={toastId} action={"error"}
variant={"outline"}>

            <VStack space="xs">

                <ToastTitle>登陆失败</ToastTitle>

                <ToastDescription>

                    账号或密码错误

                </ToastDescription>

            </VStack>

        </Toast>

    </>;

}

});

}

}

};

return (

    <ImageBackground source={require('@assets/images/background.png')}
className='h-screen w-screen p-4 '>

        <VStack space='lg'>

            <VStack className='justify-center flex items-center h-64 mt-4'>

                <Heading className="text-typography-0 text-5xl p-4">智云体
育</Heading>

            </VStack>

            <VStack className='justify-center flex'>

                <FormControl className="p-4 rounded-lg border-outline-300
bg-typography-300/45">

                    <VStack space="xl">

```

```

<VStack space="xs">
  <Text className="text-typography-black">账号
</Text>

  <Input className='border-outline-500'>
    <InputField value={username} // 将状态绑定
    到 InputField

    onChangeText={setUsername} // 在这里
    更新状态

    className='h-20' />
  </Input>
</VStack>

<VStack space="xs">
  <Text className="text-typography-black">密码
</Text>

  <Input className="border-outline-500">
    <InputField value={password} // 绑定密码状
    态

    onChangeText={setPassword} // 更新密
    码状态

    type={showPassword ? "text" :
    "password"} className='h-20' />
    <InputSlot className="pr-3" onPress={() =>
    setShowPassword(!showPassword)}>
      <InputIcon as={showPassword ?
      EyeIcon : EyeOffIcon} />
    </InputSlot>
  </Input>
</VStack>

```

```

        { /* Remember Me Checkbox */

        <Checkbox size="md" value="rememberMe"
className="ml-auto" isChecked={rememberMe} onChange={checked =>
setRememberMe(checked)}>

            <CheckboxIndicator>

                <CheckboxIcon as={CheckIcon} />

            </CheckboxIndicator>

            <CheckboxLabel className='text-typography-
black'>记住密码</CheckboxLabel>

        </Checkbox>

        { /* Login Button */

        <Button onPress={handleLogin}>

            <ButtonText className="text-typography-0">登
录</ButtonText>

        </Button>

        </VStack>

        </FormControl>

    </VStack>

    { /* Agree Policy Checkbox */

    <VStack className='items-center'>

        <Checkbox size="md" value="agreePolicy" className=""
isChecked={agreePolicy} onChange={checked => setAgreePolicy(checked)}>

            <CheckboxIndicator>

                <CheckboxIcon as={CheckIcon} />

            </CheckboxIndicator>

```

```

        <CheckboxLabel className='text-typography-black '>
            <HStack>
                <Text size="lg">点击查看并同意</Text>
                <Link onPress={() =>
router.push('/terms_of_service')}>
                    <HStack>
                        <LinkText size="lg">用户协议
</LinkText>
                    </HStack>
                </Link>
                <Text size='lg'>和</Text>
                <Link onPress={() =>
router.push('/privacy_policy')}>
                    <HStack>
                        <LinkText size="lg">隐私政策
</LinkText>
                    </HStack>
                </Link>
            </HStack>
        </CheckboxLabel>
    </Checkbox>
</VStack>
</VStack>
</ImageBackground>
);
};
export default Login;

```

app\login\privacy_policy.tsx

```
import { View } from "@components/ui/view";
```

```
import React from "react";
```

```
import { Text } from "@components/ui/text";
```

```
export default function privacy_policy() {
```

```
  return (
```

```
    <View>
```

```
      <Text>这是隐私政策</Text>
```

```
    </View>
```

```
  );
```

```
}
```

app\login\terms_of_service.tsx

```
import { View } from "@components/ui/view";
```

```
import React from "react";
```

```
import { Text } from "@components/ui/text";
```

```
export default function terms_of_service() {
```

```
  return (
```

```
    <View>
```

```
      <Text>这是用户协议</Text>
```

```
    </View>
```

```
  );
```

```
}
```


app\\login_layout.tsx

```
import { Stack } from 'expo-router';
```

```
import React from 'react';
```

```
export default function LoginLayout() {
```

```
  return (
```

```
    <Stack>
```

```
      <Stack.Screen
```

```
        name='index'
```

```
        options={{
```

```
          headerShown:false
```

```
        }}
```

```
      />
```

```
      <Stack.Screen name='privacy_policy' />
```

```
      <Stack.Screen name='terms_of_service' />
```

```
    </Stack>
```

```
  )
```

```
}
```

app\\tabs\\aibot.tsx

```
import { StyleSheet } from 'react-native';
```

```
import { WebView } from 'react-native-webview';
```

```
import { View } from '@components/Themed';
```

```
import React from 'react';
```

```
export default function AibotScreen() {
```

```

return (
  <View style={styles.container}>
    <WebView
      source={{ uri: 'http://175.24.183.106/#/chat' }}
      style={styles.webview}
    />
  </View>
);
}

```

```

const styles = StyleSheet.create({
  container: {
    flex: 1,
  },
  webview: {
    flex: 1,
    width: '100%',
  }
});

```

app\tabs\index.tsx

```

import { StyleSheet, Image, ScrollView, TouchableOpacity } from 'react-native';
import { Text, View } from '@components/Themed';
import { LandPlot, Info, Dumbbell, BookOpen, Video, Heart, PenSquare, User ,BotIcon}
from 'lucide-react-native';

```

```
import MenuCard from '@components/MenuCard';

import ImageCarousel from '@components/ImageCarousel';

import React from 'react';

import { useRouter } from 'expo-router';

interface MenuItem {

  icon: any; // 或者使用更具体的类型

  label: string;

}

export default function TabOneScreen() {

  const router = useRouter();

  const menuItems = [

    { icon: User, label: '体育测试' },

    { icon: Info, label: '课程信息' },

    { icon: LandPlot, label: '场馆预约' },

    { icon: Dumbbell, label: '体育器材' },

    { icon: Video, label: '教学视频' },

    { icon: BookOpen, label: '健康常识' },

    { icon: Heart, label: '急救知识' },

    { icon: PenSquare, label: '特殊生申请' },

  ];

  const carouselImages = [

    "https://upload-images.jianshu.io/upload_images/5809200-a99419bb94924e6d.jpg?imageMogr2/auto-orient/strip%7CimageView2/2/w/1240",

    "https://upload-images.jianshu.io/upload_images/5809200-736bc3917fe92142.jpg?imageMogr2/auto-orient/strip%7CimageView2/2/w/1240",

    "https://upload-images.jianshu.io/upload_images/5809200-7fe8c323e533f656.jpg?imageMogr2/auto-orient/strip%7CimageView2/2/w/1240",

  ]
```

```
];
```

```
const handleMenuPress = (item: MenuItem, index: number) => {
```

```
  console.log(`点击了${item.label}, 索引为${index}`);
```

```
  if(index===0){
```

```
    router.push('/(home)/SportTest/SportTest');
```

```
    // router.push('/SportTest');
```

```
    // router.push('/(home)/index');
```

```
  }if(index===1){
```

```
    router.push('/(home)/MyCourse');
```

```
  }
```

```
  if(index===3){
```

```
    router.push('/(home)/SportTool');
```

```
  }
```

```
};
```

```
return (
```

```
  <ScrollView style={styles.scrollView}>
```

```
    <View style={styles.container}>
```

```
      <ImageCarousel images={carouselImages} />
```

```
      <MenuCard
```

```
        items={menuItems}
```

```
        onPress={handleMenuPress}
```

```
      />
```

```
    </View>
```

```
        </ScrollView>

    );
}
```

```
const styles = StyleSheet.create({

  scrollView: {

    flex: 1,

    backgroundColor: '#fff',

  },

  container: {

    flex: 1,

    backgroundColor: '#fff',

  },

});
```

```
app\tabs\two.tsx
```

```
import { ScrollView, StyleSheet } from 'react-native';

import EditScreenInfo from '@components/EditScreenInfo';

import { Text, View } from '@components/Themed';

import MessageCard from '@components/MessageCard';

import React from 'react';
```

```
export default function TabTwoScreen() {
```

```
  const messages = [

    {

      id: '1',
```

```
term: '2024-2025 学年, 秋冬学期',  
content: '体测时间已开始'  
},  
{  
  id: '2',  
  term: '2024-2025 学年, 秋冬学期',  
  content: '成绩已经录入, 您的成绩为 90 分'  
},  
{  
  id: '3',  
  term: '2024-2025 学年, 秋冬学期',  
  content: '请注意查看最新的课程安排'  
},  
{  
  id: '4',  
  term: '2024-2025 学年, 秋冬学期',  
  content: '图书馆将于下周一进行维修'  
},  
{  
  id: '5',  
  term: '2024-2025 学年, 秋冬学期',  
  content: '校车时刻表已更新'  
},  
{  
  id: '6',  
  term: '2024-2025 学年, 秋冬学期',  
  content: '新的学生宿舍楼即将开放'
```

```
},  
  
{  
  id: '7',  
  term: '2024-2025 学年, 秋冬学期',  
  content: '体育馆新设备已到货'  
},  
  
{  
  id: '8',  
  term: '2024-2025 学年, 秋冬学期',  
  content: '明天下午有一场篮球友谊赛'  
},  
  
{  
  id: '9',  
  term: '2024-2025 学年, 秋冬学期',  
  content: '学生会招新活动即将开始'  
},  
  
{  
  id: '10',  
  term: '2024-2025 学年, 秋冬学期',  
  content: '期末考试时间表已发布'  
},  
  
{  
  id: '11',  
  term: '2024-2025 学年, 秋冬学期',  
  content: '毕业典礼细节已确定'  
},  
  
{
```

```
      id: '12',
      term: '2024-2025 学年, 秋冬学期',
      content: '新的交换生项目现已开放申请'
    },
    {
      id: '13',
      term: '2024-2025 学年, 秋冬学期',
      content: '校园网络升级通知'
    },
    {
      id: '14',
      term: '2024-2025 学年, 秋冬学期',
      content: '疫情防控最新通知'
    },
    {
      id: '15',
      term: '2024-2025 学年, 秋冬学期',
      content: '志愿者活动报名开始'
    },
    // 可以添加更多消息...
  ];

const handleMessagePress = (id: string) => {
  console.log('Clicked message with id:', id);
};

return (
```



```

<ScrollView style={styles.container}>
  <View style={styles.messageList}>
    {messages.map((message) => (
      <MessageCard
        key={message.id}
        id={message.id}
        term={message.term}
        content={message.content}
        onPress={handleMessagePress}
      />
    ))}
  </View>
</ScrollView>
);
}

```

```

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#f5f5f5',
  },
  messageList: {
    padding: 16,
  },
});

```

app\tabs\UserProfile.tsx

```
import React from 'react';

import { View, Text, Image, StyleSheet, ScrollView } from 'react-native';

import FeatureItem from '@components/FeatureItem';

import { useRouter } from 'expo-router';

import { BookUser, History, Cog } from 'lucide-react-native';

const UserProfile: React.FC = () => {

  const router=useRouter();

  return (

    <View style={styles.container}>

      {/* 用户信息区域 */}

      <View style={styles.profileSection}>

        <Image

          source={{uri:'http://175.24.183.106:9000/test/user.jpg'}}

          style={styles.avatar}

        />

        <Text style={styles.userName}>Test_001</Text>

        <Text style={styles.userId}>00000001</Text>

      </View>

      {/* 功能菜单区域 */}

      <ScrollView style={styles.menuSection}>

        {/* 使用功能条目组件 */}

        <FeatureItem

          icon={BookUser}

          text="个人资料修改"

          onPress={() => router.push('/EditProfile')}

        />

      </ScrollView>

    </View>

  );
}
```

```
    />

    <FeatureItem

      icon={History}

      text="我的历史成绩"

      onPress={() => console.log('我的历史成绩')}

    />

    <FeatureItem

      icon={Cog}

      text="系统设置"

      onPress={() => console.log('系统设置')}

    />

  </ScrollView>

</View>

);

};
```

```
const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#f9f9f9',

  },

  header: {

    flexDirection: 'row',

    alignItems: 'center',

    padding: 15,

    backgroundColor: '#fff',

  },

});
```

```
headerTitle: {  
  fontSize: 18,  
  fontWeight: 'bold',  
  marginLeft: 10,  
},  
profileSection: {  
  alignItems: 'center',  
  backgroundColor: '#f0f4f8',  
  paddingVertical: 20,  
},  
avatar: {  
  width: 80,  
  height: 80,  
  borderRadius: 40,  
  marginBottom: 10,  
},  
userName: {  
  fontSize: 20,  
  fontWeight: 'bold',  
  color: '#000',  
},  
userId: {  
  fontSize: 14,  
  color: '#555',  
  marginTop: 5,  
},  
menuSection: {
```

```
    marginTop: 10,  
    backgroundColor: '#fff',  
  },  
});
```

```
export default UserProfile;
```

```
app\(\tabs)\_layout.tsx
```

```
import React from 'react';  
import FontAwesome from '@expo/vector-icons/FontAwesome';  
import { Link, Tabs } from 'expo-router';  
import { Pressable } from 'react-native';  
import Colors from '@constants/Colors';  
import { useColorScheme } from '@components/useColorScheme';  
import { useClientOnlyValue } from '@components/useClientOnlyValue';  
  
// You can explore the built-in icon families and icons on the web at  
https://icons.expo.fyi/  
  
function TabBarIcon(props: {  
  name: React.ComponentProps<typeof FontAwesome>['name'];  
  color: string;  
}) {  
  return <FontAwesome size={28} style={{ marginBottom: -3 }} {...props} />;  
}
```

```
export default function TabLayout() {
```

```
const colorScheme = useColorScheme();

return (
  <Tabs
    screenOptions={{
      tabBarActiveTintColor: Colors[colorScheme ?? 'light'].tint,
      // Disable the static render of the header on web
      // to prevent a hydration error in React Navigation v6.
      headerShown: useClientOnlyValue(false, true),
      headerStyle: {
        height: 50,
        borderBottomWidth: 0,
        shadowOpacity: 0,
        elevation: 0,
      },
    }}
  >
    <Tabs.Screen
      name="index"
      options={{
        tabBarLabel: '首页',
        headerTitle: '智云体育',
        headerTitleAlign: 'center',
        tabBarIcon: ({ color }) => <TabBarIcon name="home" color={color} />,
        headerRight: () => (
          <Link href="/modal" asChild>
```

```

        <Pressable>
          {{{ pressed }} => (
            <FontAwesome
              name="info-circle"
              size={25}
              color={Colors[colorScheme ?? 'light'].text}
              style={{ marginRight: 15, opacity: pressed ? 0.5 : 1 }}
            />
          )}
        </Pressable>
      </Link>
    ),
  }}
/>
<Tabs.Screen
  name="two"
  options={{
    tabBarLabel: '通知',
    headerTitle: '我的消息',
    headerTitleAlign: 'center',
    tabBarIcon: ({ color }) => <TabBarIcon name="envelope" color={color} />,
  }}
/>
<Tabs.Screen
  name="aibot"
  options={{
    tabBarLabel: 'AI 助手',

```

```

        headerTitle: 'AI 助手',
        headerTitleAlign: 'center',
        tabBarIcon: ({ color }) => <TabBarIcon name="code" color={color} />,
      }}
    />
  <Tabs.Screen
    name="UserProfile"
    options={{
      tabBarLabel: '我的',
      headerTitle: '个人空间',
      headerTitleAlign: 'center',
      tabBarIcon: ({ color }) => <TabBarIcon name="user" color={color} />,
    }}
  />
</Tabs>
);
}

```

app\+html.tsx

```
import { ScrollViewStyleReset } from 'expo-router/html';
```

```
import React from 'react';
```

```
// This file is web-only and used to configure the root HTML for every
```

```
// web page during static rendering.
```

```
// The contents of this function only run in Node.js environments and
```

```
// do not have access to the DOM or browser APIs.
```

```
export default function Root({ children }: { children: React.ReactNode }) {
```



```

return (
  <html lang="en">
    <head>
      <meta charSet="utf-8" />
      <meta httpEquiv="X-UA-Compatible" content="IE=edge" />
      <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no" />

      {/*
        Disable body scrolling on web. This makes ScrollView components work
        closer to how they do on native.

        However, body scrolling is often nice to have for mobile web. If you want to
        enable it, remove this line.
      */}

      <ScrollViewStyleReset />

      {/* Using raw CSS styles as an escape-hatch to ensure the background color
        never flickers in dark-mode. */}

      <style dangerouslySetInnerHTML={{ __html: responsiveBackground }} />

      {/* Add any additional <head> elements that you want globally available on
        web... */}

    </head>
    <body>{children}</body>
  </html>
);
}

const responsiveBackground = `

```

```
body {  
  background-color: #fff;  
}  
  
@media (prefers-color-scheme: dark) {  
  body {  
    background-color: #000;  
  }  
};
```

app\+not-found.tsx

```
import { Link, Stack } from 'expo-router';
```

```
import { StyleSheet } from 'react-native';
```

```
import { Text, View } from '@components/Themed';
```

```
import React from 'react';
```

```
export default function NotFoundScreen() {  
  return (  
    <>  
      <Stack.Screen options={{ title: 'Oops!' }} />  
      <View style={styles.container}>  
        <Text style={styles.title}>This screen doesn't exist.</Text>  
  
        <Link href="/(tabs)" style={styles.link}>  
          <Text style={styles.linkText}>Go to home screen!</Text>  
        </Link>  
    </>  
  );  
}
```

```
        </View>

      </>

    );
  }
}
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: 'center',
    justifyContent: 'center',
    padding: 20,
  },
  title: {
    fontSize: 20,
    fontWeight: 'bold',
  },
  link: {
    marginTop: 15,
    paddingVertical: 15,
  },
  linkText: {
    fontSize: 14,
    color: '#2e78b7',
  },
});
```

app\EditProfile.tsx

```
import { Modal, ModalBackdrop, ModalBody, ModalCloseButton, ModalContent,
ModalFooter, ModalHeader } from '@components/ui/modal';

import { Image } from '@components/ui/image';

import { Actionsheet, ActionsheetIcon, ActionsheetBackdrop, ActionsheetContent,
ActionsheetDragIndicator, ActionsheetDragIndicatorWrapper, ActionsheetItem,
ActionsheetItemText, ActionsheetFlatList, ActionsheetScrollView, ActionsheetSectionList,
ActionsheetSectionHeaderText, ActionsheetVirtualizedList } from
'@components/ui/actionsheet';

import { ButtonText } from "@components/ui/button";

import { FormControl } from "@components/ui/form-control";

import { Input, InputField, InputSlot, InputIcon } from "@components/ui/input";

import { VStack } from "@components/ui/vstack";

import React, { useState } from "react";

import { Button } from "@components/ui/button";

import { Text } from "@components/ui/text";

import { HStack } from "@components/ui/hstack";

import * as ImagePicker from 'expo-image-picker';

import { Avatar, AvatarFallbackText, AvatarImage } from "@components/ui/avatar";

import { Divider } from "@components/ui/divider";

const EditProfile = () => {

  const [showAvatar, setShowAvatar] = useState(false);

  const [showActionsheet, setShowActionsheet] = React.useState(false);

  const [avatar, setAvatar] =
useState<string>('http://175.24.183.106:9000/test/user.jpg');

  const [name, setName] = useState('Test_001');

  const [studentId, setStudentId] = useState('0000001');

  const [college, setCollege] = useState('计算学院');
```

```
const [major, setMajor] = useState('计算机');

const [phone, setPhone] = useState('123456');

const handleClose = () => {
    setShowActionsheet(false);
};

//提交按钮点击

const handleSubmit = () => {
    const profileData = {
        avatar,
        name,
        studentId,
        college,
        major,
        phone
    };

    console.log('个人资料:', profileData);

    //Alert.alert('提交成功', '个人资料已提交');
};

const handleAvatarShow = () => {
    setShowActionsheet(false);

    setShowAvatar(true);

    console.log(showAvatar);
};

// 处理头像上传

const handleAvatarUpload = async () => {
    const permissionResult = await
```

```

ImagePicker.requestMediaLibraryPermissionsAsync();

    if (!permissionResult.granted) {

        console.log('请允许访问相册以上传头像');

        //Alert.alert('权限被拒绝', '请允许访问相册以上传头像');

        return;

    }

    const pickerResult = await ImagePicker.launchImageLibraryAsync({

        mediaTypes: 'images',

        // 直接使用 'images' 来选择图片类型

        allowsEditing: true,

        aspect: [1, 1],

        quality: 1

    });

    // 检查是否取消了选择

    if (!pickerResult.canceled) {

        setAvatar(pickerResult.assets[0].uri); // 访问 assets[0].uri

    }

    setShowActionsheet(false);

};

return <VStack className="" space="lg">

    <VStack className='h-full w-full flex justify-between'>

        <FormControl className='bg-typography-100/45'>

            <HStack className="items-center px-4 py-3">

                <Text className="text-typography-black">头像: </Text>

                <Button onPress={() => setShowActionsheet(true)}

```

```
className='ml-auto rounded-full aspect-square bg-background-300/0' size='sm'>
```

```
    <Avatar size="sm">
```

```
        <AvatarImage source={{
```

```
            uri: avatar
```

```
        }} />
```

```
    </Avatar>
```

```
</Button>
```

```
    <Actionsheet isOpen={showActionsheet}  
onClose={handleClose}>
```

```
        <ActionsheetBackdrop />
```

```
        <ActionsheetContent>
```

```
            <ActionsheetDragIndicatorWrapper>
```

```
                <ActionsheetDragIndicator />
```

```
            </ActionsheetDragIndicatorWrapper>
```

```
            <ActionsheetItem onPress={handleAvatarShow}>
```

```
                <ActionsheetItemText>查看头像
```

```
            </ActionsheetItemText>
```

```
            </ActionsheetItem>
```

```
            <ActionsheetItem onPress={handleAvatarUpload}>
```

```
                <ActionsheetItemText>上传头像
```

```
            </ActionsheetItemText>
```

```
            </ActionsheetItem>
```

```
            <ActionsheetItem onPress={handleClose}>
```

```
                <ActionsheetItemText>取消
```

```
            </ActionsheetItemText>
```

```
            </ActionsheetItem>
```

```
        </ActionsheetContent>
```

```
    </Actionsheet>
```

```
</HStack>

<Divider className='bg-typography-300 shadow-
inner'></Divider>

<HStack space="xs" className="items-center pl-4 py-3">

  <Text className="text-typography-black">姓名： </Text>

  <Input className='border-0 w-64 ml-auto'>

    <InputField value={name} // 将状态绑定到 InputField

      onChangeText={setName} // 在这里更新状态

      className='h-20 text-right' />

  </Input>

</HStack>

<Divider className='bg-typography-300 shadow-
inner'></Divider>

<HStack space="xs" className="items-center pl-4 py-3">

  <Text className="text-typography-black">学号： </Text>

  <Input className='border-0 w-64 ml-auto'>

    <InputField value={studentId} // 将状态绑定到 InputField

      onChangeText={setStudentId} // 在这里更新状态

      className='h-20 text-right' />

  </Input>

</HStack>

<Divider className='bg-typography-300 shadow-
inner'></Divider>

<HStack space="xs" className="items-center pl-4 py-3">

  <Text className="text-typography-black">学院： </Text>

  <Input className='border-0 w-64 ml-auto'>

    <InputField value={college} // 将状态绑定到 InputField

      onChangeText={setCollege} // 在这里更新状态
```



```

        className='h-20 text-right' />

    </Input>

</HStack>

<Divider className='bg-typography-300 shadow-
inner'></Divider>

<HStack space="xs" className="items-center pl-4 py-3">

    <Text className="text-typography-black">专业： </Text>

    <Input className='border-0 w-64 ml-auto'>

        <InputField value={major} // 将状态绑定到 InputField

            onChangeText={setMajor} // 在这里更新状态

            className='h-20 text-right' />

        </Input>

    </HStack>

<Divider className='bg-typography-300 shadow-
inner'></Divider>

<HStack space="xs" className="items-center pl-4 py-3">

    <Text className="text-typography-black">电话： </Text>

    <Input className='border-0 w-64 ml-auto'>

        <InputField value={phone} // 将状态绑定到 InputField

            onChangeText={setPhone} // 在这里更新状态

            className='h-20 text-right' />

        </Input>

    </HStack>

<Divider className='bg-typography-300 shadow-
inner'></Divider>

</FormControl>

<VStack className='p-4'>

    <Button onPress={handleSubmit} className="w-full ">

```

```

        <ButtonText className="text-typography-0">保存
    </ButtonText>

    </Button>

    </VStack>

</VStack>

<Modal
    isOpen={showAvatar}
    onClose={() => { setShowAvatar(false); }}
    className='p-0 rounded-none'
>
    <ModalBackdrop className="" />
    <ModalContent className='w-full aspect-square p-0 border-0
rounded-none'>
        <ModalBody className='w-full m-0'>
            <VStack className='bg-background-500 aspect-square w-
full'>
                <Image source={{ uri: avatar }} alt='头像无法显示'
className='w-full h-full'></Image>
            </VStack>
        </ModalBody>
    </ModalContent>
</Modal>
</VStack>;
};

export default EditProfile;

app\modal.tsx

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

```

```

import { Platform, StyleSheet } from 'react-native';

import EditScreenInfo from '@components/EditScreenInfo';
import { Text, View } from '@components/Themed';
import React from 'react';

export default function ModalScreen() {
  return (
    <View style={styles.container}>
      <Text style={styles.title}>Modal</Text>
      <View style={styles.separator} lightColor="#eee"
darkColor="rgba(255,255,255,0.1)" />
      <EditScreenInfo path="app/modal.tsx" />

      { /* Use a light status bar on iOS to account for the black space above the modal
*/ }

      <StatusBar style={Platform.OS === 'ios' ? 'light' : 'auto'} />
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    alignItems: 'center',
    justifyContent: 'center',
  },

```

```
title: {  
  fontSize: 20,  
  fontWeight: 'bold',  
},  
separator: {  
  marginVertical: 30,  
  height: 1,  
  width: '80%',  
},  
});
```

app\test.tsx

```
import React from 'react';  
import { Text } from 'react-native';
```

```
export default function Test() {  
  return <Text>Test</Text>;  
}
```

app_layout.tsx

```
import FontAwesome from '@expo/vector-icons/FontAwesome';  
import "@/global.css";  
import { GluestackUIProvider } from "@components/ui/gluestack-ui-provider";  
import { DarkTheme, DefaultTheme, ThemeProvider } from '@react-navigation/native';  
import { useFonts } from 'expo-font';  
import { Stack } from 'expo-router';
```

```
import * as SplashScreen from 'expo-splash-screen';

import { useEffect } from 'react';

import 'react-native-reanimated';

import { useColorScheme } from '@components/useColorScheme';

import React from 'react';

import { useRouter } from 'expo-router';

import { ChevronLeft } from 'lucide-react-native';

export {

  // Catch any errors thrown by the Layout component.

  ErrorBoundary,

} from 'expo-router';

export const unstable_settings = {

  // Ensure that reloading on `/modal` keeps a back button present.

  initialRouteName: '(login)',

};

// Prevent the splash screen from auto-hiding before asset loading is complete.

SplashScreen.preventAutoHideAsync();

export default function RootLayout() {

  const [loaded, error] = useFonts({

    SpaceMono: require('../assets/fonts/SpaceMono-Regular.ttf'),

    ...FontAwesome.font,

  });
```

```

// Expo Router uses Error Boundaries to catch errors in the navigation tree.

useEffect(() => {
  if (error) throw error;
}, [error]);

useEffect(() => {
  if (loaded) {
    SplashScreen.hideAsync();
  }
}, [loaded]);

if (!loaded) {
  return null;
}

return <GluestackUIProvider mode="light"><RootLayoutNav
/></GluestackUIProvider>;
}

function RootLayoutNav() {
  const colorScheme = useColorScheme();

  return (
    <GluestackUIProvider mode="light"><GluestackUIProvider
mode="light"><ThemeProvider value={colorScheme === 'dark' ? DarkTheme :
DefaultTheme}>
      <Stack>
        <Stack.Screen

```

```
name="(login)"

options={{
    headerTitle:"login",
    headerShown:false
}}
/>

<Stack.Screen
    name="(tabs)"
    options={{
        headerShown: false,
    }}
/>

<Stack.Screen
    name="(home)"
    options={{
        headerShown: false,
    }}
/>

<Stack.Screen
    name="modal"
    options={{
        presentation: 'modal',
    }}
/>

<Stack.Screen
    name="EditProfile"
    options={{
```

```
        headerTitle:"修改资料",
      }}
    />

  </Stack>

</ThemeProvider></GluestackUIProvider></GluestackUIProvider>

);
}
```

components\CourseTagProps.tsx

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

```
interface CourseTagProps {
  year: string; // 学年，例如 "2024-2025"
  term: string; // 学期，例如 "秋冬学期"
  courseName: string; // 课程名称，例如 "射箭"
  time: string; // 上课时间，例如 "(周一第 6, 7 节)"
  teacher: string; // 授课教师
  location: string; // 上课地点
  onPress?: () => void; // 点击事件处理函数
}
```

```
const CourseTag: React.FC<CourseTagProps> = ({ year, term, courseName, time,
teacher, location, onPress }) => {
  return (
```



```

<TouchableOpacity onPress={onPress} activeOpacity={0.7}>
  <View style={styles.card}>
    {/* 学年与学期 */}
    <Text style={styles.termText}>
      {year}学年, {term}
    </Text>
    {/* 课程详细信息 */}
    <View style={styles.details}>
      <Text style={styles.courseName}>{courseName}</Text>

      <Text style={styles.teacher}>老师: {teacher}</Text>
      <Text style={styles.time}>上课时间: {time}</Text>
      <Text style={styles.location}>上课地点: {location}</Text>

    </View>
  </View>
</TouchableOpacity>
);
};

```

```

const styles = StyleSheet.create({
  card: {
    backgroundColor: '#f7faff', // 卡片背景颜色
    borderRadius: 8,
    marginTop: 16,
    marginLeft: 8,
    marginRight: 8,

```

```
    overflow: 'hidden', // 防止内容溢出

    shadowColor: '#000',

    shadowOffset: { width: 0, height: 2 },

    shadowOpacity: 0.1,

    shadowRadius: 4,

    elevation: 2, // Android 阴影
  },

  termText: {

    backgroundColor: '#d9eafc', // 顶部背景色

    color: '#333', // 字体颜色

    fontSize: 14,

    paddingVertical: 8,

    paddingHorizontal: 12,
  },

  details: {

    padding: 12,
  },

  courseName: {

    fontSize: 16,

    fontWeight: 'bold',

    color: '#333',

    marginBottom: 4,
  },

  time: {

    fontSize: 14,

    color: '#555',

    marginBottom: 4,
```

```
    },  
    teacher: {  
      fontSize: 14,  
      color: '#555',  
      marginBottom: 4,  
    },  
    location: {  
      fontSize: 14,  
      color: '#555',  
    },  
  },  
});
```

```
export default CourseTag;
```

```
components\EditScreenInfo.tsx
```

```
import React from 'react';
```

```
import { StyleSheet } from 'react-native';
```

```
import { ExternalLink } from './ExternalLink';
```

```
import { MonoText } from './StyledText';
```

```
import { Text, View } from './Themed';
```

```
import Colors from '@constants/Colors';
```

```
export default function EditScreenInfo({ path }: { path: string }) {
```

```
  return (
```

<View>

<View style={styles.getStartedContainer}>

<Text

style={styles.getStartedText}

lightColor="rgba(0,0,0,0.8)"

darkColor="rgba(255,255,255,0.8)">

Open up the code for this screen:

</Text>

<View

style={[styles.codeHighlightContainer, styles.homeScreenFilename]}

darkColor="rgba(255,255,255,0.05)"

lightColor="rgba(0,0,0,0.05)">

<MonoText>{path}</MonoText>

</View>

<Text

style={styles.getStartedText}

lightColor="rgba(0,0,0,0.8)"

darkColor="rgba(255,255,255,0.8)">

Change any of the text, save the file, and your app will automatically update.

</Text>

</View>

<View style={styles.helpContainer}>

<ExternalLink

style={styles.helpLink}

```
      href="https://docs.expo.io/get-started/create-a-new-app/#opening-the-app-on-your-phonetablet">
```

```
      <Text style={styles.helpLinkText} lightColor={Colors.light.tint}>
```

```
        Tap here if your app doesn't automatically update after making changes
```

```
      </Text>
```

```
    </ExternalLink>
```

```
  </View>
```

```
</View>
```

```
);
```

```
}
```

```
const styles = StyleSheet.create({
```

```
  getStartedContainer: {
```

```
    alignItems: 'center',
```

```
    marginHorizontal: 50,
```

```
  },
```

```
  homeScreenFilename: {
```

```
    marginVertical: 7,
```

```
  },
```

```
  codeHighlightContainer: {
```

```
    borderRadius: 3,
```

```
    paddingHorizontal: 4,
```

```
  },
```

```
  getStartedText: {
```

```
    fontSize: 17,
```

```
    lineHeight: 24,
```

```
    textAlign: 'center',
```

```

    },
    helpContainer: {
      marginTop: 15,
      marginHorizontal: 20,
      alignItems: 'center',
    },
    helpLink: {
      paddingVertical: 15,
    },
    helpLinkText: {
      textAlign: 'center',
    },
  },
});

```

components\ExternalLink.tsx

```

import { Link } from 'expo-router';
import * as WebBrowser from 'expo-web-browser';
import React from 'react';
import { Platform } from 'react-native';

export function ExternalLink(
  props: Omit<React.ComponentProps<typeof Link>, 'href'> & { href: string }
) {
  return (
    <Link
      target="_blank"

```

```

    {...props}

    // @ts-expect-error: External URLs are not typed.
    href={props.href}
    onPress={(e) => {
      if (Platform.OS !== 'web') {
        // Prevent the default behavior of linking to the default browser on native.
        e.preventDefault();
        // Open the link in an in-app browser.
        WebBrowser.openBrowserAsync(props.href as string);
      }
    }}
  />
);
}

```

components\FeatureItem.tsx

```

import React from 'react';
import { View, Text, TouchableOpacity, StyleSheet } from 'react-native';
import { ChevronRight, LucideIcon } from 'lucide-react-native';

interface FeatureItemProps {
  icon: LucideIcon; // 图标名称
  text: string; // 条目文本
  onPress: () => void; // 点击事件
}

```

```

const FeatureItem: React.FC<FeatureItemProps> = ({ icon: Icon, text, onPress }) => {
  return (
    <TouchableOpacity style={styles.container} onPress={onPress}>
      <View style={styles.leftSection}>
        {/* 使用 Lucide 图标组件 */}
        <Icon size={24} color="#666" />
        <Text style={styles.text}>{text}</Text>
      </View>
      {/* 这里也需要使用 Lucide 的 ChevronRight 图标 */}
      <ChevronRight size={24} color="#000" />
    </TouchableOpacity>
  );
};

```

```

const styles = StyleSheet.create({
  container: {
    flexDirection: 'row',
    alignItems: 'center',
    justifyContent: 'space-between',
    paddingVertical: 15,
    borderBottomWidth: 1,
    borderBottomColor: '#888',
    paddingHorizontal: 15,
    backgroundColor: '#e6f2ff'
  },
  leftSection: {
    flexDirection: 'row',

```



```
        alignItems: 'center',
      },
      text: {
        fontSize: 16,
        marginLeft: 10,
        color: '#000',
      },
    });
```

```
export default FeatureItem;
```

```
components\ImageCarousel.tsx
```

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

```
import { View, Image, StyleSheet, Dimensions, FlatList, TouchableOpacity } from 'react-native';
```

```
interface ImageCarouselProps {
  images: string[];
}
```

```
const { width } = Dimensions.get('window');
```

```
export default function ImageCarousel({ images }: ImageCarouselProps) {
```

```
  const [activeIndex, setActiveIndex] = useState(0);
```

```
  const flatListRef = useRef<FlatList | null>(null);
```

```
  useEffect(() => {
```

```

const interval = setInterval(() => {
  if (activeIndex === images.length - 1) {
    flatListRef.current?.scrollToIndex({ index: 0, animated: true });
  } else {
    flatListRef.current?.scrollToIndex({ index: activeIndex + 1, animated: true });
  }
}, 3000); // Change image every 3 seconds

return () => clearInterval(interval);
}, [activeIndex, images.length]);

```

```

const handleImagePress = (index: number) => {
  console.log(`Image ${index + 1} clicked`);
  // Here you can add navigation logic or any other action
  // For example: navigation.navigate('ImageDetails', { imageIndex: index });
};

```

```

const renderItem = ({ item, index }: { item: string; index: number }) => (
  <View style={styles.carouselContainer}>
    <TouchableOpacity onPress={() => handleImagePress(index)}>
      <Image source={{ uri: item }} style={styles.image} />
    </TouchableOpacity>
  </View>
);

```

```

const onViewableItemsChanged = useRef(({ viewableItems }: any) => {
  if (viewableItems.length > 0) {

```

```

        setActiveIndex(viewableItems[0].index);
    }
    }).current;

return (
    <View style={styles.container}>
        <FlatList
            ref={flatListRef}
            data={images}
            renderItem={renderItem}
            horizontal
            pagingEnabled
            showsHorizontalScrollIndicator={false}
            onViewableItemsChanged={onViewableItemsChanged}
            viewabilityConfig={{ itemVisiblePercentThreshold: 50 }}
            keyExtractor={(item, index) => index.toString()}
        />
        <View style={styles.pagination}>
            {images.map((_, index) => (
                <View
                    key={index}
                    style={[
                        styles.paginationDot,
                        index === activeIndex ? styles.paginationDotActive : null,
                    ]}
                />
            ))}
        </View>
    </View>
);

```

```

        </View>
    </View>
);
}

const styles = StyleSheet.create({
  container: {
    height: 200,
    margin: 16,
  },
  image: {
    width: width - 32, // Subtract the total horizontal margin
    height: 200,
    resizeMode: 'cover',
    borderRadius: 12,
  },
  pagination: {
    flexDirection: 'row',
    position: 'absolute',
    bottom: 10,
    alignSelf: 'center',
  },
  paginationDot: {
    width: 8,
    height: 8,
    borderRadius: 4,
    backgroundColor: 'rgba(255, 255, 255, 0.5)',
  },
});

```

```

        marginHorizontal: 4,
    },
    paginationDotActive: {
        backgroundColor: '#fff',
    },
    carouselContainer: {
        borderRadius: 12,
        shadowColor: '#000',
        shadowOffset: {
            width: 0,
            height: 2,
        },
        shadowOpacity: 0.25,
        shadowRadius: 3.84,
        elevation: 5,
        backgroundColor: '#fff', // Required for Android shadow
    },
});

```

components\MenuCard.tsx

```

import React from 'react';
import { StyleSheet, TouchableOpacity, View } from 'react-native';
import { Text } from '@components/Themed';
import { LucideIcon } from 'lucide-react-native';

```

```

interface MenuItem {
    icon: LucideIcon;

```

```
    label: string;
  }
```

```
interface MenuCardProps {
  items: MenuItem[];
  onPress?: (item: MenuItem, index: number) => void;
}
```

```
export default function MenuCard({ items, onPress }: MenuCardProps) {
  return (
    <View style={styles.cardContainer}>
      <View style={styles.card}>
        <View style={styles.menuGrid}>
          {items.map((item, index) => (
            <TouchableOpacity
              key={index}
              style={styles.menuitem}
              onPress={() => onPress?.(item, index)}
            >
              <View style={styles.iconCircle}>
                <item.icon size={24} color="#666" />
              </View>
              <Text style={styles.menuText}>{item.label}</Text>
            </TouchableOpacity>
          )))}
        </View>
      </View>
    </View>
  )
}
```

```
    </View>

  );
}
```

```
const styles = StyleSheet.create({
  cardContainer: {
    padding: 16,
  },
  card: {
    backgroundColor: '#e6f2ff',
    borderRadius: 16,
    padding: 16,
    shadowColor: '#000',
    shadowOffset: {
      width: 0,
      height: 2,
    },
    shadowOpacity: 0.25,
    shadowRadius: 3.84,
    elevation: 5,
  },
  menuGrid: {
    flexDirection: 'row',
    flexWrap: 'wrap',
    justifyContent: 'space-between',
  },
  menuItem: {
```

```
      width: '25%',
      alignItems: 'center',
      marginBottom: 20,
    },
    iconCircle: {
      width: 50,
      height: 50,
      borderRadius: 25,
      backgroundColor: '#fff',
      justifyContent: 'center',
      alignItems: 'center',
      shadowColor: '#000',
      shadowOffset: {
        width: 0,
        height: 2,
      },
      shadowOpacity: 0.1,
      shadowRadius: 2,
      elevation: 3,
    },
    menuText: {
      marginTop: 8,
      fontSize: 12,
      color: '#333',
      textAlign: 'center',
    },
  },
});
```


components\MessageCard.tsx

```
import React from 'react';
```

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

```
import { List } from 'lucide-react-native';
```

```
interface MessageCardProps {
```

```
  id: string;
```

```
  term: string;
```

```
  content: string;
```

```
  onPress: (id: string) => void;
```

```
}
```

```
export default function MessageCard({ id, term, content, onPress }: MessageCardProps)
```

```
{
```

```
  return (
```

```
    <TouchableOpacity onPress={() => onPress(id)}>
```

```
      <View style={styles.card}>
```

```
        <View style={styles.header}>
```

```
          <Text style={styles.termText}>{term}</Text>
```

```
          <List size={20} color="#666" />
```

```
        </View>
```

```
        <Text style={styles.contentText}>{content}</Text>
```

```
      </View>
```

```
    </TouchableOpacity>
```

```
  );
```

```
}
```

```
const styles = StyleSheet.create({  
  card: {  
    backgroundColor: '#e6f2ff',  
    borderRadius: 20,  
    padding: 16,  
    marginBottom: 16,  
    width: '100%',  
    shadowColor: "#000",  
    shadowOffset: {  
      width: 0,  
      height: 2,  
    },  
    shadowOpacity: 0.25,  
    shadowRadius: 3.84,  
    elevation: 5,  
  },  
  header: {  
    flexDirection: 'row',  
    justifyContent: 'space-between',  
    alignItems: 'center',  
    marginBottom: 8,  
  },  
  termText: {  
    fontSize: 14,  
    color: '#666',  
  },  
},
```

```
    contentText: {  
      fontSize: 16,  
      color: '#333',  
    },  
  });
```

components\ScoreCard.tsx

```
import React from "react";  
  
import { Table, TableBody, TableData, TableHead, TableHeader, TableRow,  
TableFooter } from '@components/ui/table';  
  
import { Box } from '@components/ui/box';  
  
import { Text } from '@components/ui/text';  
  
import { Card } from '@components/ui/card';  
  
import { Avatar, AvatarFallbackText, AvatarImage } from '@components/ui/avatar';  
  
import { VStack } from '@components/ui/vstack';  
  
import { Heading } from '@components/ui/heading';  
  
import { HStack } from '@components/ui/hstack';  
  
import { Divider } from '@components/ui/divider';
```

```
interface UserInfo {  
  name: string;  
  id: string;  
  score: number;  
  rankText: string;  
  avatarUrl: string;  
}
```

```
interface ScoreItem {  
    project: string;  
    score: number;  
    weightedScore: number;  
}
```

```
interface ScoreCardProps {  
    userInfo: UserInfo;  
    scoreData: ScoreItem[];  
}
```

```
const ScoreCard: React.FC<ScoreCardProps> = ({ userInfo, scoreData }) => {  
    return (  
        <Card className="bg-background-200 rounded-2xl p-0 my-4">  
            { /* 用户信息部分 */ }  
            <Box className="flex flex-row items-center container-none p-5">  
                <Avatar className="mr-4">  
                    <AvatarFallbackText>  
                        {userInfo.name.slice(0, 2).toUpperCase()}  
                    </AvatarFallbackText>  
                    <AvatarImage source={{ uri: userInfo.avatarUrl }} />  
                </Avatar>  
                <VStack>  
                    <Heading size="md" className="mb-1">  
                        {userInfo.name}  
                    </Heading>  
                    <Text size="sm">{userInfo.id}</Text>  
                </VStack>  
            </Box>  
        </Card>  
    )  
}
```

```

</VStack>

<VStack className="ml-auto">

    <HStack>

        <Heading size="4xl">{userInfo.score}</Heading>

        <Text size="sm" className="mt-5">分</Text>

    </HStack>

    <Text size="sm">{userInfo.rankText}</Text>

</VStack>

</Box>

<Divider className="bg-indigo-500 object-cover" />

{/* 成绩表格部分 */}

<Box className="rounded-2xl container-none p-5">

    <Table className="w-full rounded-2xl">

        <TableHeader>

            <TableRow className="border-b-0 bg-background-0">

                <TableHead className="font-bold">项目

            </TableHead>

                <TableHead className="font-bold">成绩

            </TableHead>

                <TableHead className="font-bold">权重分

            </TableHead>

        </TableRow>

    </TableHeader>

    <TableBody>

        {scoreData.map((item, index) => (

            <TableRow

                key={`score-row-${index}`}

                className={`border-b-0 ${

```

```

            index % 2 === 0 ? "bg-typography-200" : ""
        }}
    >
        <TableData>{item.project}</TableData>
        <TableData>{item.score}</TableData>
        <TableData>{item.weightedScore}</TableData>
    </TableRow>
  )}}
</TableBody>
</Table>
</Box>
</Card>
);
};

```

```
export default ScoreCard;
```

components\StandCard.tsx

```
import React from 'react';
```

```
import { VStack } from './ui/vstack';
```

```
import { HStack } from './ui/hstack';
```

```
import StandItem from './StandItem'; // 引入封装的 StandItem 组件
```

```
import { Card } from './ui/card';
```

```
// 定义单个分数项的类型
```

```
interface ScoreItem {
```

```
    heading: string; // 标题, 例如 "50 米"

    score: number;    // 分数
  }
```

// 定义组件的 Props 类型

```
interface StandCardProps {

    scores: ScoreItem[][]; // 二维数组, 表示分组的分数数据

    onPressItem: (heading: string) => void; // 点击事件的回调函数
}
```

```
const StandCard: React.FC<StandCardProps> = ({ scores, onPressItem }) => {

    return (

        <Card className="bg-typography-100 rounded-2xl p-4 my-4">

            <VStack className="flex justify-center" space="3xl">

                {scores.map((row, rowIndex) => (

                    <HStack key={rowIndex} className="m-100 flex justify-center w-full"
space="lg">

                        {row.map((item, itemIndex) => (

                            <StandItem

                                key={itemIndex}

                                heading={item.heading}

                                score={item.score}

                                onPress={() => onPressItem(item.heading)}

                            />

                        ))}

                    </HStack>

                ))}

            </VStack>

        </Card>

    )
}
```

```

        </VStack>

      </Card>

    );
  };

  export default StandCard;

```

components\StandItem.tsx

```

import React from 'react';

import { Pressable, Text } from 'react-native';

import { Card } from './ui/card'; // 替换为你的 Card 组件路径

import { Heading } from './ui/heading'; // 替换为你的 Heading 组件路径

```

```

type ScoreCardProps = {

  heading: string;

  score: number;

  onPress: () => void;

};

```

```

const StandItem: React.FC<ScoreCardProps> = ({ heading, score, onPress }) => {

  return (

    <Pressable onPress={onPress}>

      {{{ pressed }}} => (

        <Card className={pressed ? 'w-32 bg-primary-100' : 'w-32'}>

          <Heading>{heading}</Heading>

          <Text className="text-primary-800">权重分: {score}</Text>

        </Card>

```



```
    })  
  </Pressable>  
  
  );  
};
```

```
export default StandItem;
```

```
components\StyledText.tsx
```

```
import React from 'react';
```

```
import { Text, TextProps } from './Themed';
```

```
export function MonoText(props: TextProps) {  
  return <Text {...props} style={[props.style, { fontFamily: 'SpaceMono' }] } />;  
}
```

```
components\Themed.tsx
```

```
/**
```

```
 * Learn more about Light and Dark modes:
```

```
 * https://docs.expo.io/guides/color-schemes/
```

```
 */
```

```
import { Text as DefaultText, View as DefaultView } from 'react-native';
```

```
import Colors from '@constants/Colors';
```

```
import { useColorScheme } from './useColorScheme';
```

```
import React from 'react';
```

```
type ThemeProps = {  
  lightColor?: string;  
  darkColor?: string;  
};
```

```
export type TextProps = ThemeProps & DefaultText['props'];  
export type ViewProps = ThemeProps & DefaultView['props'];
```

```
export function useThemeColor(  
  props: { light?: string; dark?: string },  
  colorName: keyof typeof Colors.light & keyof typeof Colors.dark  
) {  
  const theme = useColorScheme() ?? 'light';  
  const colorFromProps = props[theme];  
  
  if (colorFromProps) {  
    return colorFromProps;  
  } else {  
    return Colors[theme][colorName];  
  }  
}
```

```
export function Text(props: TextProps) {  
  const { style, lightColor, darkColor, ...otherProps } = props;  
  const color = useThemeColor({ light: lightColor, dark: darkColor }, 'text');
```

```
    return <DefaultText style={{ color }, style}} {...otherProps} />;
  }
```

```
export function View(props: ViewProps) {
  const { style, lightColor, darkColor, ...otherProps } = props;

  const backgroundColor = useThemeColor({ light: lightColor, dark: darkColor },
    'background');

  return <DefaultView style={{ backgroundColor }, style}} {...otherProps} />;
}
```

components\useClientOnlyValue.ts

// This function is web-only as native doesn't currently support server (or build-time) rendering.

```
export function useClientOnlyValue<S, C>(server: S, client: C): S | C {
  return client;
}
```

components\useClientOnlyValue.web.ts

import React from 'react';

// `useEffect` is not invoked during server rendering, meaning

// we can use this to determine if we're on the server or not.

```
export function useClientOnlyValue<S, C>(server: S, client: C): S | C {
```

```
const [value, setValue] = React.useState<S | C>(server);

React.useEffect(() => {

    setValue(client);

}, [client]);


return value;

}
```

```
components\useColorScheme.ts

export { useColorScheme } from 'react-native';
```

```
components\useColorScheme.web.ts

// NOTE: The default React Native styling doesn't support server rendering.

// Server rendered styles should not change between the first render of the HTML

// and the first render on the client. Typically, web developers will use CSS media

queries

// to render different styles on the client and server, these aren't directly supported in

React Native

// but can be achieved using a styling library like Nativewind.

export function useColorScheme() {

    return 'light';

}
```

```
constants\Colors.ts

const tintColorsLight = '#2f95dc';
```

```
const tintColorDark = '#fff';
```

```
export default {
```

```
  light: {
```

```
    text: '#000',
```

```
    background: '#fff',
```

```
    tint: tintColorLight,
```

```
    tabIconDefault: '#ccc',
```

```
    tabIconSelected: tintColorLight,
```

```
  },
```

```
  dark: {
```

```
    text: '#fff',
```

```
    background: '#000',
```

```
    tint: tintColorDark,
```

```
    tabIconDefault: '#ccc',
```

```
    tabIconSelected: tintColorDark,
```

```
  },
```

```
};
```