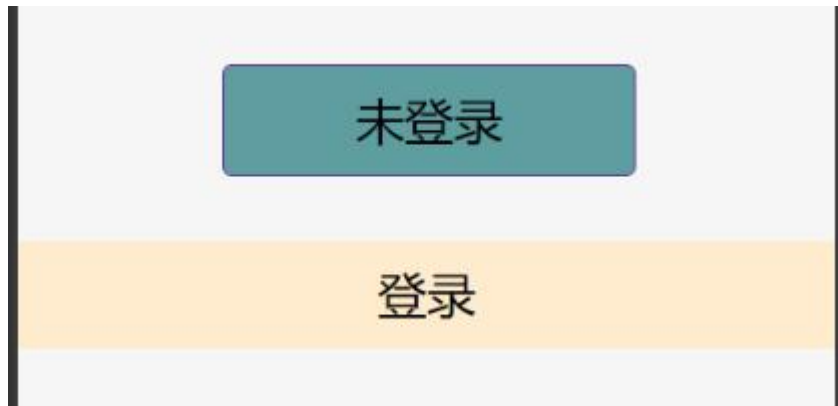


1. 需求和功能说明：

小程序默认是未登录状态，此时的文化传播和普法宣传功能都可以正常的使用，其他页面如下所示



此时登录所在的区域可以点击，点击后就可以进入登录界面。

登录节目中手机号只保证了能够输入数字，且最长长度为 11，在点击登录按钮的时候还会从新检测手机号长度是否为 11，过长过短都会阻止登录；在密码栏限制长度为 15，且输入的时候不会直接显示字符。

The image shows a login form. It has two input fields: the first is labeled '手机号' (Phone number) and contains the text '12312312312'; the second is labeled '密码' (Password) and contains four dots. Below the input fields is a green button with the text '登录' (Login). At the bottom, there is a teal-colored button with the text '还没注册? 点击这里' (Not registered? Click here).

如果输入信息出现错误，也会有相应的提升



手机号
12312312312

密码
...

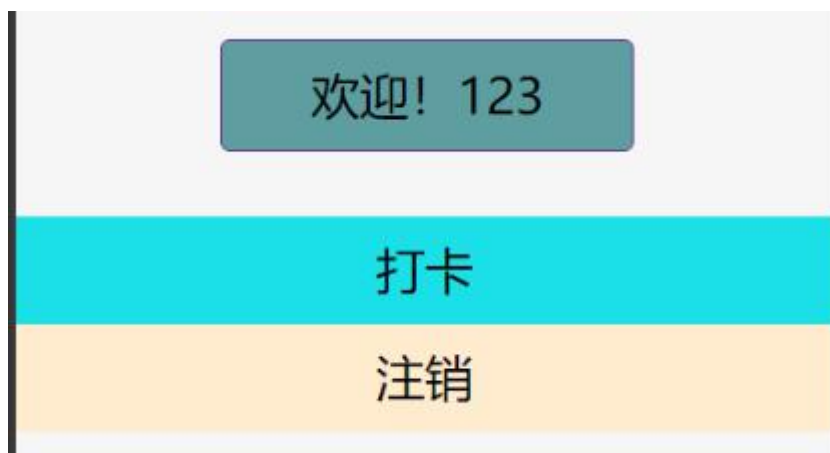
还没注册?

登录

!

信息有误

输入的信息正确之后，就自动跳转到了其他页面，此时的效果就是登录后的效果



欢迎! 123

打卡

注销

隐藏的打卡选项会暴露出来，且只出现一个注销按钮，上面的也会变成欢迎! + 用户名的模式；点击注销，就会重新回到第一幅图的未登录状态。

点击注册按钮进入注册界面，手机号还是限制 11 位的数字，用户名和密码也都有长度上的限制。



用户名
请输入用户名

手机号
请输入手机号

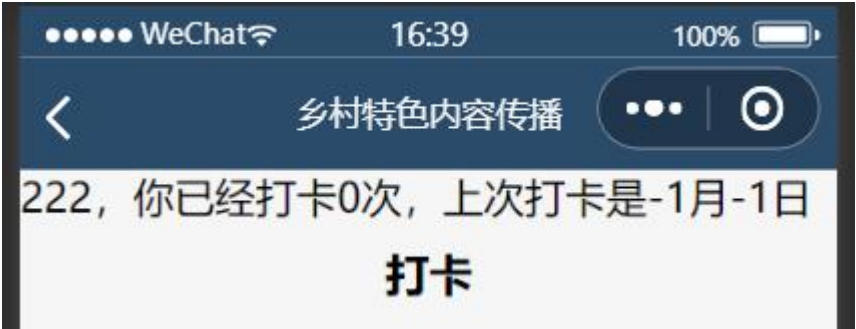
密码
请输入密码

注册

如果都合法的话，手机号也没有被注册过，那么就会出现如下画面，在 2.5s 之后跳转到第一幅图的未登录的状态，可以使用新注册的账号登录。



在登录状态下点击打卡，进入打卡页面，打卡页面还没有进行美化，只是以文本的形式给出了上次打卡的时间和打卡的总次数



如果打卡 0 次，不会出现有效的月份和日期，点击打卡后，出现提示界面



如果今天还没打卡，那么会出现打卡成功，对应的数据发生改变，2.5s 后回到登录界面；如果已经打卡一次，那么会提示打卡失败，也会退回到登录界面



2. 技术实现说明

实现登录，注册，注销等功能；通过在小程序中通过 **Mobx** 实现全局数据共享，在不同的页面下调用相关数据的操作函数，来保证小程序的运行过程中的数据的共享；共享的数据包括当前用户的状态，用户名，唯一标识用户的号码等。

访问数据库的操作都是在小程序的 **js** 代码中实现的，在某些逻辑中，使用 **async** 和 **await** 来规定异步操作的顺序，防止产生冲突；比如注册时现在数据库中查找是否使用过当前注册的号码，在查询结果得出后，才针对结果确定是否插入这个注册的数据到数据库中。

3. 产品实现说明

```
export const store = observable({
  isLogIn:false,
  userName:'',
  phone:'',
  changeStatus:action(function(){
    this.isLogIn=!this.isLogIn
  }),
  setName:action(function(name){
    this.userName=name
  }),
  setPhone:action(function(_phone){
    this.phone=_phone
  }),
  getPhone:action(function(){
    return this.phone
  })
})
```

用来保存全局共享的变量

```
async register(){  
  let phone=this.data.phone  
  let pass=this.data.password  
  
  if(phone.length!=11){ ...  
  }  
  
  if(pass.length>15){ ...  
  }  
  
  var count=await db.collection('user').where({ ...  
  }).count()  
  
  if(count.total>0){ ...  
  }  
  
  else{ ...  
  }  
  
},
```

实现注册的函数，保证了两次数据库访问的顺序防止发生错误。

测试用例的手机号和密码

1.

手机号：12312312312

密码：1111

2.

手机号：18812511931

密码：11111