

2025年夏季《移动软件开发》实验报告

姓名：陈怡冰 学号：23020007007

姓名和学号？	陈怡冰，23020007007
本实验属于哪门课程？	中国海洋大学25夏《移动软件开发》
实验名称？	实验1：第一个微信小程序
博客地址？	OUC-2025年夏季《移动软件开发》实验报告-实验一-CSDN博客
Github仓库地址？	ChenFirstIce/Mobile-software-development-for-the-summer-semester-of-2025: OUC-2025夏季学期《移动软件开发》课程的六个实验以及个人项目的课程报告以及代码。主要是使用微信小程序开发工具开发微信小程序，并涉及到一点鸿蒙系统。

(备注：将实验报告发布在博客、代码公开至 github 是 **加分项**，不是必须做的)

一、实验目标

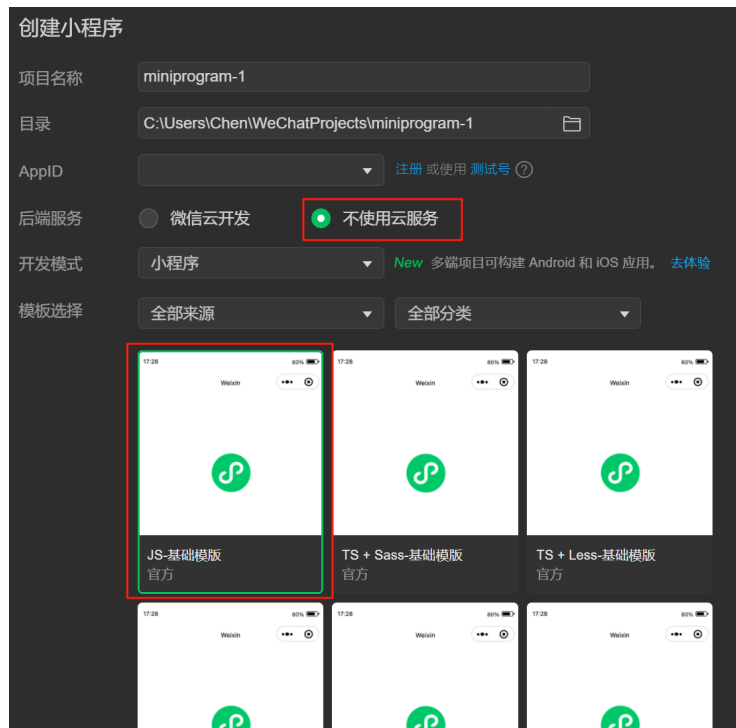
- 1、学习使用快速启动模板创建小程序的方法；
- 2、学习不使用模板手动创建小程序的方法。

二、实验步骤

2.1 自动生成小程序

1. 创建一个项目

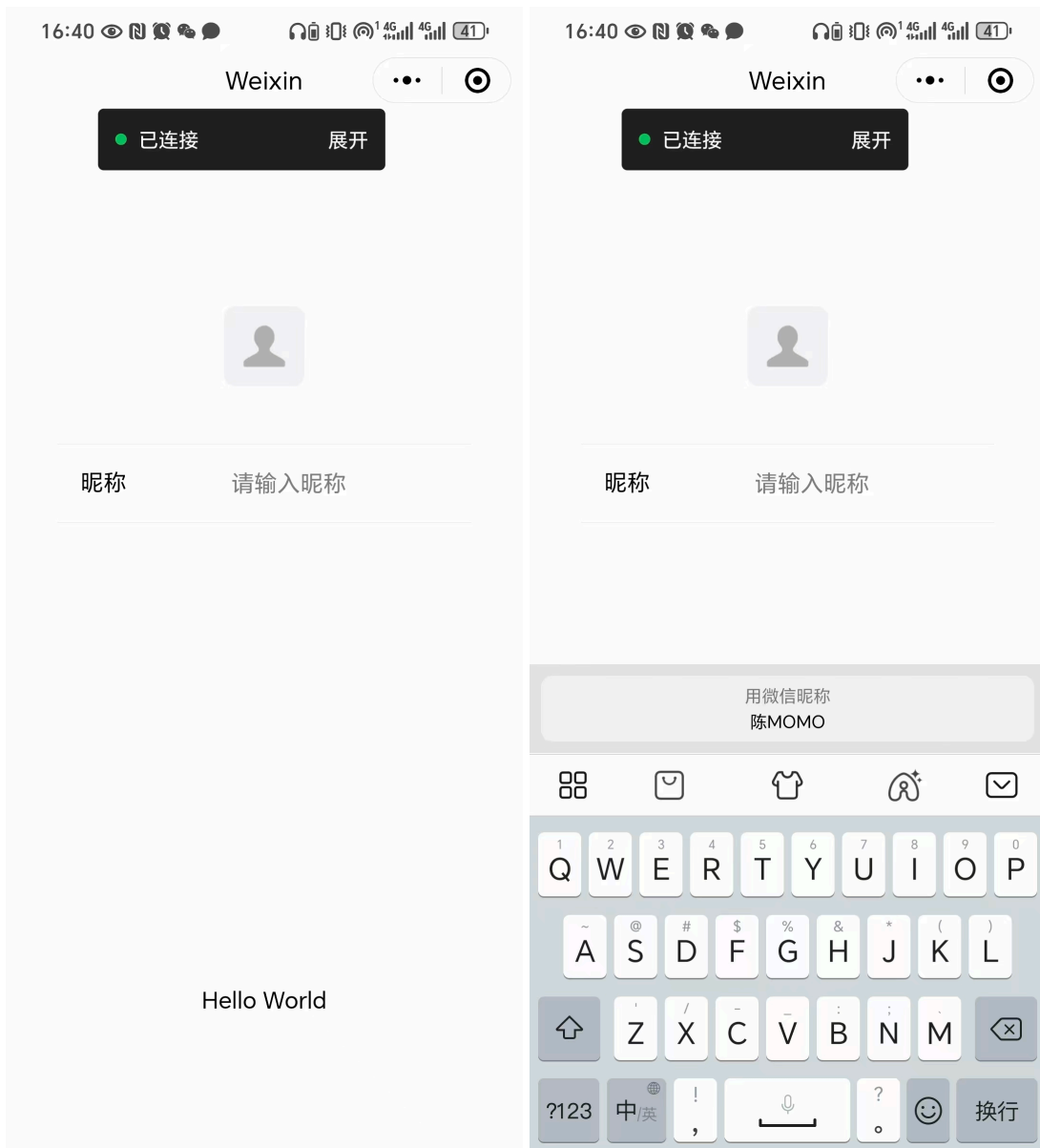
选择开发模式为“**小程序**”，并选中“**不使用云服务**”，然后选择合适的模板即可。（前面**加粗**的步骤指的是非必须项，只是为了找到符合实验内容的模板）



2. 真机预览

直接点击页面的**真机预览**就可以通过扫码的方式查看真机的界面。

效果如下：点击输入时会显示可以使用微信昵称



2.2 手动创建小程序

1. 项目创建 & 2. 页面配置

可以选择创建一个**空白项目**，我这里是选择模板项目并根据文档内容对项目中的内容进行删改：

- 将 app.json 文件内 pages 属性中的“pages/logs/logs”删除，并删除上一行末尾的逗号。
- 删除utils 文件夹及其内部所有内容
- 删除pages 文件夹下的 logs目录及其内部所有内容。
- 删除index.wxml和 index.wxss 中的全部代码。
- 删除index.js 中的全部代码,并且输入关键词“page”找到第二个选项按回车键让其自动补全函数。

```
//index.js
Page({
  /**
   * 页面的初始数据
   */
  data: {
```

```
},

/**
 * 生命周期函数--监听页面加载
 */
onLoad: function (options) {

},

/**
 * 生命周期函数--监听页面初次渲染完成
 */
onReady: function () {

},

/**
 * 生命周期函数--监听页面显示
 */
onShow: function () {

},

/**
 * 生命周期函数--监听页面隐藏
 */
onHide: function () {

},

/**
 * 生命周期函数--监听页面卸载
 */
onUnload: function () {

},

/**
 * 页面相关事件处理函数--监听用户下拉动作
 */
onPullDownRefresh: function () {

},

/**
 * 页面上拉触底事件的处理函数
 */
onReachBottom: function () {

},

/**
 * 用户点击右上角分享
 */
onShareAppMessage: function () {
```

```
},  
})
```

- 删除app.wxss中的全部代码
- 删除 app.js中的全部代码，并且输入关键词“app”找到第一个选项按回车键让其自动补全函数。

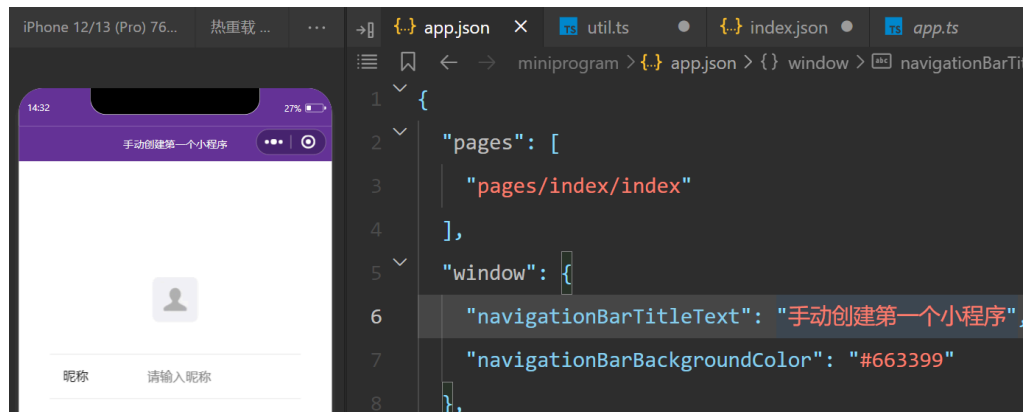
```
//app.js  
App({  
  
  /**  
   * 当小程序初始化完成时，会触发 onLaunch（全局只触发一次）  
   */  
  onLaunch: function () {  
  
  },  
  
  /**  
   * 当小程序启动，或从后台进入前台显示，会触发 onShow  
   */  
  onShow: function (options) {  
  
  },  
  
  /**  
   * 当小程序从前台进入后台，会触发 onHide  
   */  
  onHide: function () {  
  
  },  
  
  /**  
   * 当小程序发生脚本错误，或者 api 调用失败时，会触发 onError 并带上错误信息  
   */  
  onError: function (msg) {  
  
  }  
})
```

3. 视图设计

- 导航栏设计

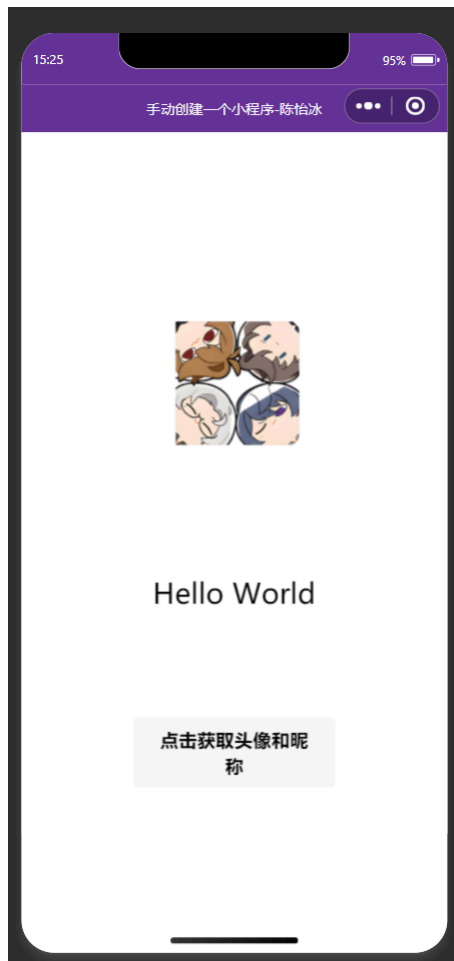
修改 app.json 文件中的page部分代码，把原模板中的 "pages/logs/logs" 删除，并修改 windows部分的代码，把导航栏根据教程换成白字紫底。

- 效果如下：



○ 页面设计

- 编写 `pages/index/index.wxml` 的代码，主要是定义当前页面应当出现的三个组件（包括微信头像[image]，微信昵称[text]，按钮[button]）的类。然后通过指定image属性 `src` 和 `mode` 模式得到静态图片，同样设置静态text。
- 编写 `pages/index/index.wxss` 设置container类，image、text元素的样式。
- 效果如下：



4. 逻辑实现

○ 获取微信用户实现

- 修改 `index.wxml` 的button组建的代码，设置获取用户信息的**激活事件**，这里的激活事件设置为**bindtap**（必须由用户主动点击事件直接触发）。这样点击后会直接出发 `getMyInfo` 函数。

```
<view class = 'container'>
  <image></image>
  <text>Hello world</text>
  <button bindtap="getMyInfo">
    点击获取头像和昵称
  </button>
</view>
```

- 编写 index.js 文件，主要是定义 **data** 和自定义函数 **getMyInfo**（这里会调用微信自带的 api: **wx.getUserProfile**），同时将获取到的 **wx.getUserProfile** 返回的值，并在 console 中显示，以便判断是否调用成功。

```
data: {
  src: '/images/logo.jpg',
  name: 'Hello world',
  hasUserInfo: false,
  canIUseGetUserProfile: wx.canIUse('getUserProfile'),
  canIUseNicknameComp: wx.canIUse('input.type.nickname'),
},
//获取用户信息
getMyInfo: function(e) {
  wx.getUserProfile({
    desc: '展示用户信息',
    success: (res) => {
      console.log('用户信息获取成功', res)
      this.setData({
        name: res.userInfo.nickName, // 修正: 从 userInfo 中获取昵称
        src: res.userInfo.avatarUrl, // 修正: 从 userInfo 中获取头像
        hasUserInfo: true
      })
    },
    fail: (res) => {
      console.log('用户信息获取失败', res)
    }
  })
}
```

```
用户信息获取成功 index.js? [sm]:18
{cloudID: undefined, encryptedData: "u0ZsCtnmu+E3U0VUoSOKr952T8vZD5s8HwuLzAD+IGMgryAOhb...iYvUKr6PSkvLZB11Y/zF2sJpMkXIXdQPkBeynRBoS1w...HsPIpQ", iv: "0ec58sYGUiTuTM895SHmDg==", signature: "69aa9ad12004409f8f4457f38b5d92c65f50431d", userInfo: {...}, ...}
cloudID: undefined
encryptedData: "u0ZsCtnmu+E3U0VUoSOKr952T8vZD5s8HwuLzAD+IGMgryAOhbT70y+BVPip9aQEgLMze94ZvyhWYzqrYi51+jKNFsux70BU04giZBKioy060CYC...
errMsg: "getUserProfile:ok"
iv: "0ec58sYGUiTuTM895SHmDg=="
rawData: '{"nickName":"微信用户","gender":0,"language":"","city":"","province":"","country":"","avatarUrl":"https://thirdwx.qlogo...
signature: "69aa9ad12004409f8f4457f38b5d92c65f50431d"
userInfo: {nickName: "微信用户", gender: 0, language: "", city: "", province: "", ...}
```

○ 使用动态数据显示头像和昵称

- 修改 index.wxml 中 **image** 和 **text** 组件的代码，使用 **{{}}** 做成动态数据；

```

<view class = 'container'>
  <image src='{{src}}' mode='widthFix'></image>
  <text>{{name}}</text>
  <button bindtap="getMyInfo">
    点击获取头像和昵称
  </button>
</view>

```

- 并在 `index.js` 的 `data` 中修改 `src` 和 `name` 的默认值。

```

data: {
  src: '/images/logo.jpg',
  name: 'Hello world',
  hasUserInfo: false,
  canIUseGetUserProfile: wx.canIUse('getUserProfile'),
  canIUseNicknameComp: wx.canIUse('input.type.nickname'),
},

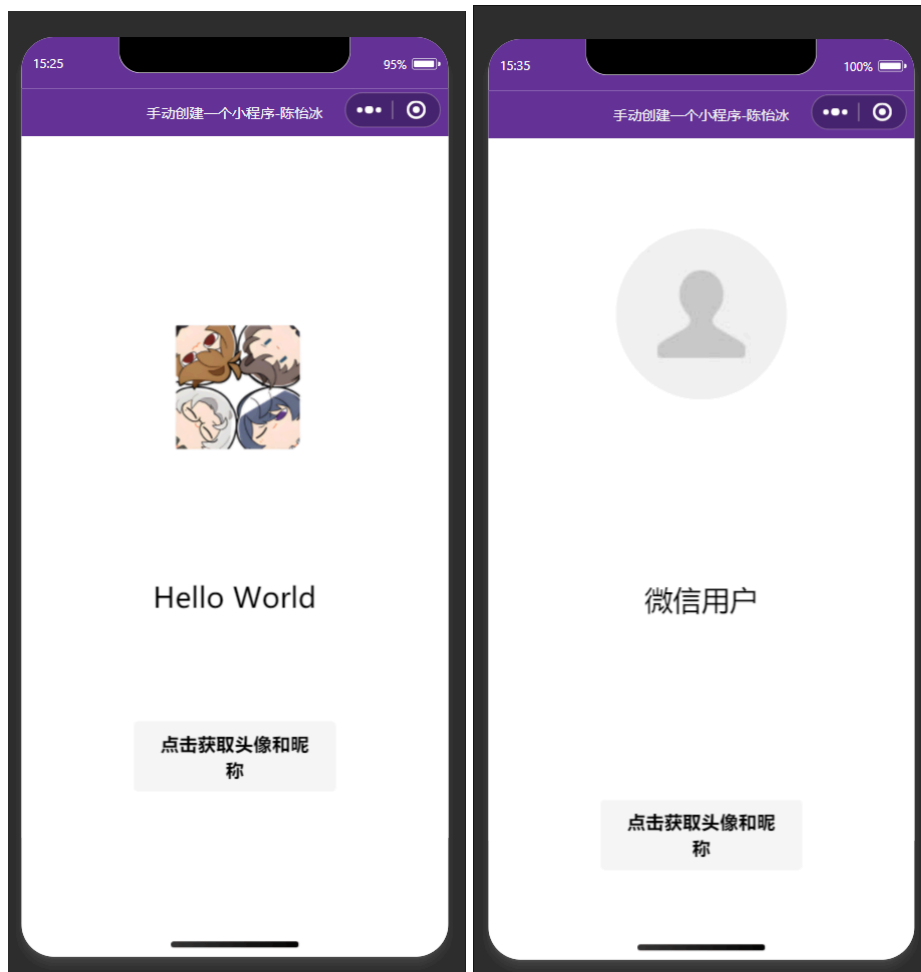
```

- 更新头像和昵称
 - 在 `getMyInfo` 函数中修改 `src` 和 `name`，效果见上图。

三、程序运行结果

最终结果：

左边为初始页面，点击按钮后**头像和昵称**改变成当前用户的信息。



四、问题总结与体会

问题一：无法获取用户信息

当使用 `getUserInfo` 作为激活事件时，无法获取用户的信息。此时 `index.js` 的 `getMyInfo` 函数中调用了 `wx.getUserProfile`。

```
<button open-type="getUserInfo" bindgetuserinfo="getMyInfo">
  点击获取头像和昵称
</button>
```

用户信息获取失败

```
{errMsg: "getUserProfile:fail can only be invoked by user TAP gesture."}
errMsg: "getUserProfile:fail can only be invoked by user TAP gesture."
constructor: (...)
hasOwnProperty: (...)
isPrototypeOf: (...)
propertyIsEnumerable: (...)
toLocaleString: (...)
toString: (...)
valueOf: (...)
defineGetter: (...)
```

解决方案

由于 `wx.getUserProfile` 的安全限制，使得该函数不能通过代码自动调用或其他事件触发。而 `open-type="getUserInfo"` 的触发逻辑由微信底层控制，与 `getUserProfile` 接口的调用规范不兼容。

```
<button bindtap="getMyInfo">
```

点击获取头像和昵称

```
</button>
```

```

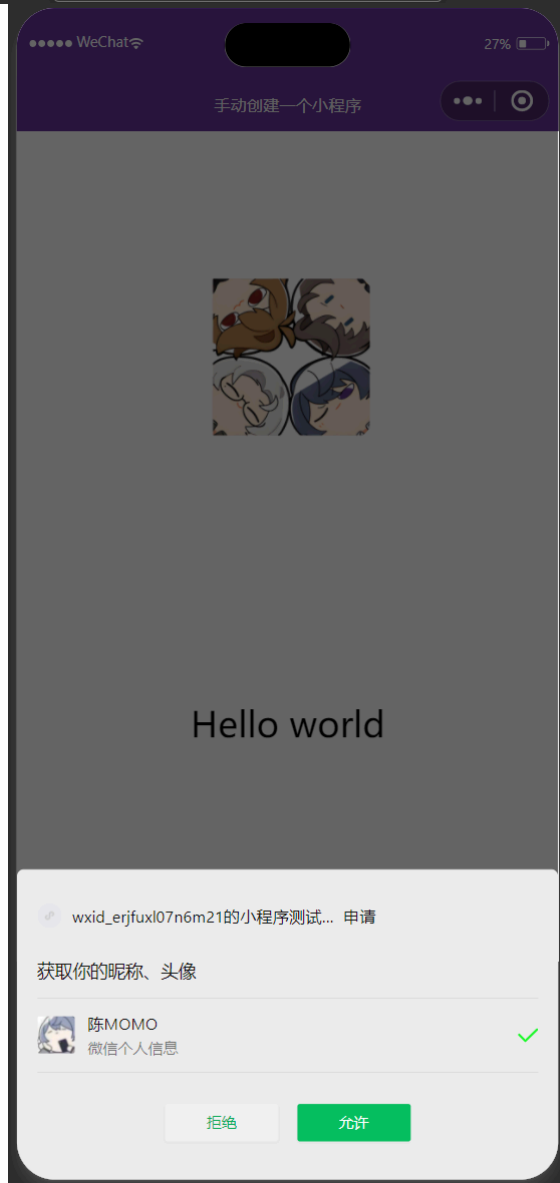
用户信息获取成功
index.js? [sm]:18
{cloudID: undefined, encryptedData: "uOZsCtnmu+E3U0VUoSOKr952T8vZD5s8HwuLzAD+IGMgrYAOhb...iYvUKr6PSkvLZBj1Y/zF2sjPmwXIXdQPkBeynRboS1w
HsPIpQ", iv: "Oec5BsYGUituTM895SHmDg==", signature: "69aa9ad12004409f8f4457f38b5d92c65f50431d", userInfo: {...}, ...}
cloudID: undefined
encryptedData: "uOZsCtnmu+E3U0VUoSOKr952T8vZD5s8HwuLzAD+IGMgrYAOhbT7Oy+BVPip9aQEg1Mze94ZvyhWYzqrYi51+jKNFsux70BU04giZBKioyO60CYC...
errMsg: "getUserProfile:ok"
iv: "Oec5BsYGUituTM895SHmDg=="
rawData: "{"nickName":"微信用户","gender":0,"language":"","city":"","province":"","country":"","avatarUrl":"https://thirdwx.qlogo...
signature: "69aa9ad12004409f8f4457f38b5d92c65f50431d"
userInfo: {nickName: "微信用户", gender: 0, language: "", city: "", province: "", ...}

```

问题二：没有拉取权限的显示

其实这个不算问题，用户信息都是可以获取成功的，只是没有显示。

做法就是把调试的基础库换成2.16.1（比较旧的版本应该都可以），之前是3.9.2。



体会

大概对 `wxml`、`js`、`wxss` 还有 `app` 的全局有了一定的了解，以下的表格是这次实验的一些体会，可能有疏漏或者不全面的地方，主要是白话的概括一下。

名称	大致用途
app	这个打头一般都是全页面不变的组件
js	做正事的，负责实现功能
wxml	这个页面有啥（除了app.wxml里的），并且可以定义激活事件，连接前端和后端
wxxx	wxml里的类或者组件长啥样，负责美观