

**青 岛 工 学 院**

**产品设计大作业报告**

**（2021-2022学年第1学期）**

题目

**学生姓名 学号**

**指导教师**

**学 院 信息工程学院**

**专 业 软件工程 年级 2019级**

**青 岛 工 学 院**

# 仿糗事百科微信小程序

自律养成记，是一款记录自己自律习惯为主题的微信小程序，它的操作简单，画面有趣，在年轻人中很流行。

在自律养成记中可以定制目标并分享自己的日常，如图1，2，3所示。

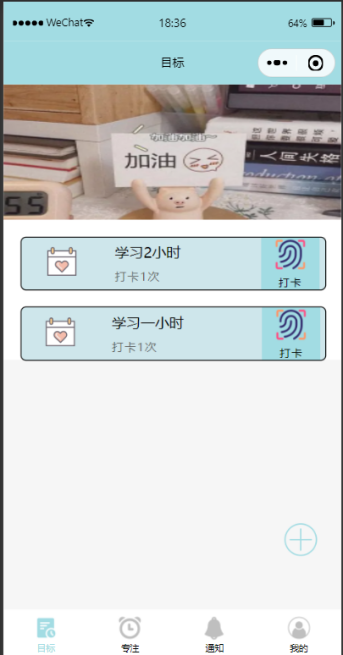
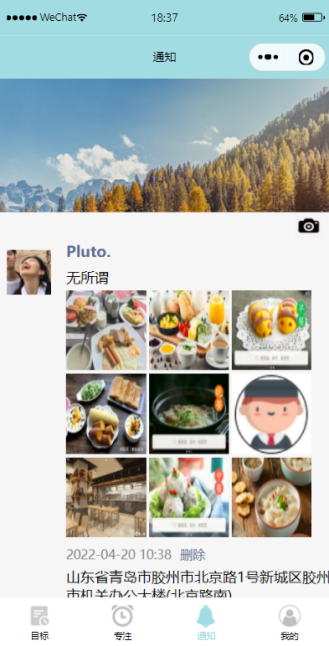
  

图1 目标 图2专注 图3 分享

## 1． 需求描述

自律养成记微信小程序主要完成以下功能。

（1）实现底部菜单切换效果，如图4所示。



图4 底部菜单

（2）实现底部菜单切换效果，菜单选中时字体以及图片变为蓝色，同时对应的内容也相应变化，如图5所示。



图5 我的界面

（3）实现我的界面信息列表设计，包括用户头像、人昵称，日历等其他信息。

## 2． 设计思路及相关知识点

### 2.1 设计思路

2.1.1 目标

（1）实现轮播图，需要使用swiper轮播组件。

（2）点击添加符号，进行页面跳转

（3）选择标签进行表单选择，写下目标。

（4）时间选择器和日期选择器来进行选择完成目标时间。

2.1.2 专注

（1）实现时间滑动条选择器，需要使用slider滑动条组件，设置其show-value属性，让value值显示出来；

（2）选中任务，图标和字体颜色就会进行相应的切换；

（3）点击开始专注按钮就会进入专注模式，根据选择的时间开始倒计时；

2.1.3 我的

（1）实现微信授权功能，实现点击获取微信头像及昵称。

（2）菜单点击功能，需要页面跳转等js。

（3）日历功能，需要js方法，获取日期并进行定位。

（4）绑定手机号上传数据库功能，需要进行云开发。

（5）个性化功能，需要点击按钮进行按钮转换。

（6）提问功能，点击按钮可以进行提问。

（7）吐槽功能，点击可以上传图片及文字。

### 2.2 相关知识点

2.2.1 目标

（1）轮播图运用swiper轮播组件

（2）菜单添加点击功能，需要wx.navigateTo等页面跳转js

（3）新建创建功能把数据传入数据库

2.2.2 专注

（1）在界面布局的时候，会用到微信小程序的组件，包括view视图容器组件、image图片组件、slider滑动条组件、canvas组件等组件的使用；

（2）界面样式设计，需要使用wxss样式进行界面的美化和渲染；

2.2.3 我的

（1）实现微信授权功能，需要借助getUserInfo: function (e)及setUserInfo: function (userInfo)等js

（2）菜单点击功能，需要wx.navigateTo等页面跳转js

（3）日历功能，需要onLoad: function ()、dateInit: function (setYear, setMonth)、lastMonth: function ()、nextMonth: function ()等方法

（4）绑定手机号上传数据库功能，需要submit: function (e)、addRname(event)等方法

（5）个性化功能，需要button按钮进行js以及wxss。

（6）提问功能，点击按钮可以进行提问，需要formSubmit(e)、radioChange(e)等方法。

（7）吐槽功能，点击可以上传图片及文字，需要handleTabsItemChange(e)等方法。

## 3． 准备工作

（1）首先需要准备一个AppID，如果没有AppID也没有关系，只不过不能在手机上进行项目的预览，但是在开发工具上开发是没有任何问题的。

（2）在设计列表页时，需要用到一些图标，放置在images文件夹下，如图7所示。



图 1 图标（1）



图 2 图标

## 4． 设计流程

我们首先来设计自律小程序目标，专注，分享，我的页面的每个具体流程。

### 目标

### 目标页面设计

实现目标默认页面的设计，如图7所示。



图7 目标页面

（1）在pages目录下，新建一个mubiaoyemian，然后进入到pages/mubiao/mubiao目录下，清空mubiao.wxml、mubiao.js、mubiao.wxss文件默认生成的内容。

（2）进入到pages/mubiao/mubiao.wxml文件里，设计swpier轮播图，历史记录，添加目标，数据库数据显示内容方面，具体代码如下所示

<view class="content">

<!-- 轮播图 -->

<view class="img">

<swiper autoplay interval='6000' duration='3000'>

<swiper-item>

<image src="/images/mubiao/top-3.webp" style="width:100%;height:230px;"></image></swiper-item>

<swiper-item>

<image src="/images/mubiao/top-2.webp" style="width:100%;height:230px;"></image></swiper-item>

<swiper-item>

<image src="/images/mubiao/top-1.webp" style="width:100%;height:230px;"></image></swiper-item>

</swiper>

</view>

<!-- 数据存放位置 -->

<view class="all-body">

<view class="body" wx:for="{{list}}" wx:key="list">

<view class="one">

<image src="/images/mubiao/yundong-0.png " style="width: 40px;height:40px;"></image></view>

<view class="two">

<view class="title">{{item.userName}}</view>

<view class="num">打卡{{total}}次</view>

</view>

<view class="daka" bindtap="daka">

<view class="img"><image src="/images/mubiao/daka-2.png" style="width: 38px;height:38px;"></image></view>

<view class="daka\_text">打卡</view>

</view>

</view>

</view>

<!-- 底部添加按钮 -->

<view class="add" bindtap="add">

<image src="/images/添加.png" style="width:40px;height: 40px;"></image>

</view>

</view>

（3）进入到pages/mubiao/mubiao.wxss文件里，给页面内容添加样式。

（4）进入到pages/mubiao/mubiao.js文件里，给历史记录绑定一个list事件等，具体代码如下所示。

add:function(){

wx.navigateTo({

url: '../mubiao-xinjian/mubiao-xinjian',

})

},

onLoad:function(options){

var \_this=this;

//引用数据库

const db=wx.cloud.database({

//环境Id

env:'cloud1-6gbr6dbdd0f97800'

})

db.collection('list').get({

success:res=>{

console.log(res.data)

//给list赋值,否则不会显示

this.setData({

list:res.data

})

}

})

},

daka(){

this.setData({

total:this.data.total+1

})

### 添加目标页面设计

添加目标页面是用来记录用户制定的目标及类型和时间如图8所示

图8 添加目标页面 图9 日期选择器

（1）在pages目录下新建一个mubiao-xinjian文件，然后进入到pages/mubiao-xinjian/mubiao-xinjian目录下，清空wxml、js、wxss文件默认生成的内容。

（2）进入到pages/mubiao-xinjian/mubiao-xinjianwxml页面，有相关的新建目标按钮及相关信息，具体代码如下所示。

<view class="item">

<view>目标类型</view>

<view>

<radio-group class="radin-group" bindchange="radioChange" name="id">

<radio value="0" checked="true">学习</radio>

<radio value="1">运动</radio>

<radio value="2">其他</radio>

</radio-group>

</view>

</view>

<view class="line"></view>

<view class="item">

<view>目标</view>

<view><input type="text" placeholder="请填写您的目标"placeholder-class="holder" name="userName"value="{{name}}" /></view>

</view>

<view class="line"></view>

<view class="item">

<view>完成日期选择</view>

<view><picker mode="date" value="{{date}}" bindchange="bindDateChange"><input name="shijian"type="text"placeholder-class="holder"date="remark" value="{{date}}"/></picker></view>

</view>

<view class="line"></view>

<view class="item">

<view>时间设定</view>

<view><picker mode="time" value="{{time}}" start="09:01"end="21:01" bindchange="bindTimeChange"><input type="text" placeholder-class="holder" name="timer" value="{{time}}"/> </picker></view>

</view>

<view class="line"></view>

<view class="btn"><button form-type="submit">创建</button></view>

</form>

<view class="btn">

<button bindtap='getData'>更新</button>

</view>

</view>

### 项目预览

如果项目中有AppID的，就可以将项目进行上传，在手机上就可以浏览自律养成记微信小程序，如图15所示。



图10 目标页面

### 专注

### 专注页面设计

实现专注默认页面的设计中，如图11所示。



图11 专注页面

（1）在pages目录下，新建一个zhuangzhu问价，然后进入到pages/zhuangzhu/zhuangzhu目录下，清空zhuangzhu.wxml、zhuangzhu.js、zhuangzhu.wxss文件默认生成的内容。

（2）进入到pages/zhuangzhu/zhuangzhu.wxml文件里，设计slider滑动条时间选择器，历史记录，任务选中，开始专注按钮设计包括4方面内容：slider滑动条、任务、历史记录、开始专注按钮，具体代码如下所示。

<view>

<view class="lishi" bindtap="history">历史记录</view>

<view class="slider">

<slider min="0" max="60" show-value activeColor="#A2DCE3" backgroundColor="#666666" value="{{time}}" bindchange="slideChange" style="margin-top: 10px;"></slider>

</view>

<view class="task\_prompt">

<view class="task\_choice">选择一个任务</view>

<view class="task\_jishi">在接下来的{{time}}分钟内，你将专注做这件事</view>

<view hidden="true">

<input type="text" value="{{time}}"/>

</view>

</view>

<view class="task\_list">

<view wx:for="{{taskList}}" wx:key="task" class="tasklist" bindtap="choicetask" data-index="{{index}}">

<view class="task\_icon">

<image src="{{index==taskActive?item.imgUrl:item.image}}"></image>

</view>

<view class='task\_text {{index==taskActive ? "task\_text\_active":""}}'>{{item.text}}</view>

<view hidden="true">

<input type="text" class='task\_text {{index==taskActive ? "task\_text\_active":""}}' value="{{item.text}}"/>

</view>

</view>

<view class="start" bindtap="start">开始专注</view>

</view>

</view>

（3）进入到pages/zhuangzhu/zhuangzhu.wxss文件里，给页面内容添加样式。

（4）进入到pages/zhuangzhu/zhuangzhu.js文件里，给历史记录绑定一个history事件、给slider滑动条绑定一个slideChange事件、给taskList数组绑定一个choicetask事件、给开始专注按钮绑定一个start事件，具体代码如下所示。

Page({

history:function(){

wx.navigateTo({

url: '../history/history',

})

},

slideChange:function(e){

this.setData({

time:e.detail.value

})

},

choicetask:function(e){

this.setData({

taskActive:e.currentTarget.dataset.index

})

var index = parseInt(e.currentTarget.dataset.index)

const db =wx.cloud.database()

db.collection('history').add({

data:{

times:this.data.time+'min',

text:this.data.taskList[index].text,

},

success:res=>{

console.log('[数据库][新增记录]成功',res);

},

fail:err=>{

console.error('[数据库][新增记录]失败',err);

}

})

},

start:function(){

var that=this

if(that.data.time>0){

wx.navigateTo({

url: '../startdown/startdown?timeStr='+that.data.time,

})

}else{

wx.showToast({

title: '时间不能为0，请调整时间',

})

}

},

})

这样就可以实现专注页面的设计效果，如图12所示。



图12 专注页面

### 专注页面模式设计

选择好时间和任务，点击专注页面的开始专注按钮，进入到专注模式页面，开启专注模式，根据选择的时间进行倒计时，在倒计时期间可以选择暂停，或者放弃

（1）在pages目录下新建一个shartdown文件，然后进入到pages/startdown/startdown目录下，清空startdown.wxml、startdown.js、startdown.wxss文件默认生成的内容。

（2）进入到pages/startdown/startdown.wxml页面，使用canvas绘图和view视图组件，具体代码如下所示。

<view class="clock" style="height:{{clockHeight}}rpx;">

<view class="progress">

<canvas canvas-id="progress\_bg" class="progress\_bg"></canvas>

<canvas canvas-id="progress\_active" class="progress\_active"></canvas>

<view class="progress\_text">{{timeStr}}</view>

</view>

<view class="btns">

<view class="okBtn" bindtap="ok" wx:if="{{okShow}}">返回</view>

<view class="pauseBtn" bindtap="pause" wx:if="{{pauseShow}}">暂停</view>

<view class="continueCancelBtn" wx:if="{{continueCancelShow}}">

<view class="continueBtn" bindtap="continue">继续</view>

<view class="cancelBtn" bindtap="cancel">放弃</view>

</view>

</view>

</view>

（3）进入到pages/startdown/startdown.wxss页面，给页面内容添加样式。

（4）进入到pages/zhuangzhu/zhuangzhu.js文件里，给历史记录绑定一个history事件、给slider滑动条绑定一个slideChange事件、给taskList数组绑定一个choicetask事件、给开始专注按钮绑定一个start事件，具体代码如下所示。

// pages/startdown/startdown.js

const app = getApp()

const util = require('../../utils/util')

Page({

onLoad: function (options) {

var res = wx.getSystemInfoSync();

var rate = 750 / res.windowWidth;

var that =this

console.log(options);

console.log(options.timeStr);

that.setData({

rate:rate,

clockHeight:rate \* res.windowHeight,

mTime:options.timeStr\*60\*1000,

timeStr:parseInt(options.timeStr)>=10 ? options.timeStr+':00':'0'+options.timeStr+':00',

time:options.timeStr

})

this.drawBg();

this.drawActive();

},

drawBg:function(){

var lineWidth = 6 / this.data.rate;

var ctx1 = wx.createCanvasContext('progress\_bg');

ctx1.setLineWidth(lineWidth);

ctx1.setStrokeStyle('#000000'); // 描边颜色

ctx1.setLineCap('round');

ctx1.beginPath(); // 开始绘制路径

ctx1.arc(400/this.data.rate/2,400/this.data.rate/2,400/this.data.rate/2-2\*lineWidth,0,2\*Math.PI,false);

ctx1.stroke();// 绘制一条路径

ctx1.draw();

},

// 动态画圆

drawActive:function(){

var \_this = this

// 开启一个定时器

var timer = setInterval(function(){

var angle = 1.5+2\*(\_this.data.time\*60\*1000 - \_this.data.mTime)/(\_this.data.time\*60\*1000)

var currentTime = \_this.data.mTime - 100

\_this.setData({

mTime:currentTime

})

if(angle < 3.5){

if(currentTime % 1000 == 0){

var timeStr1 = currentTime / 1000;// s

var timeStr2 = parseInt(timeStr1 / 60);// m

var timeStr3 = (timeStr1 - timeStr2\*60) >= 10 ? (timeStr1 - timeStr2\*60) :

'0'+(timeStr1 - timeStr2\*60);

var timeStr2 = timeStr2 >= 10 ? timeStr2 : '0'+timeStr2;

\_this.setData({

timeStr:timeStr2+':'+timeStr3

})

}

var lineWidth = 6 / \_this.data.rate;

var ctx2 = wx.createCanvasContext('progress\_active')

ctx2.setLineWidth(lineWidth);

ctx2.setStrokeStyle('#ffffff');

ctx2.setLineCap('round')

ctx2.beginPath();

ctx2.arc(400/\_this.data.rate/2,400/\_this.data.rate/2,400/\_this.data.rate/2-2\*lineWidth,

1.5\*Math.PI,angle\*Math.PI,false);

ctx2.stroke();

ctx2.draw();

}else{

var logs = wx.getStorageSync('logs') || [];

logs.unshift({

date:util.formatTime(new Date),

task:\_this.data.taskActive,

time:\_this.data.time

});

wx.setStorageSync('logs', logs);

\_this.setData({

timeStr:'00:00',

okShow:true,

pauseShow:true,

continueCancelShow:false

});

clearInterval(timer);

}

},100)

\_this.setData({

timer:timer

})

},

ok:function(){

clearInterval(this.data.timer);

wx.reLaunch({

url: '../zhuanzhu/zhuangzhu',

})

},

pause:function(){

clearInterval(this.data.timer);

this.setData({

pauseShow:false,

continueCancelShow:true,

okShow:false

})

},

continue:function(){

this.drawActive();

this.setData({

pauseShow:true,

continueCancelShow:false,

okShow:true

})

},

cancel:function(){

clearInterval(this.data.timer);

wx.reLaunch({

url: '../zhuanzhu/zhuangzhu',

})

}

})

这样就可以实现专注模式页面的设计效果，如图13所示。

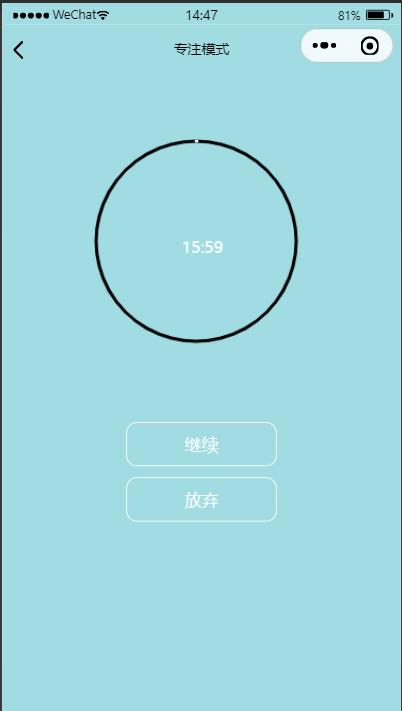
 

图13 专注模式页面

### 历史记录页面设计

历史记录页面是用来记录用户专注了多长时间和多少次，如果记录太多可以选择清空记录，如图14所示

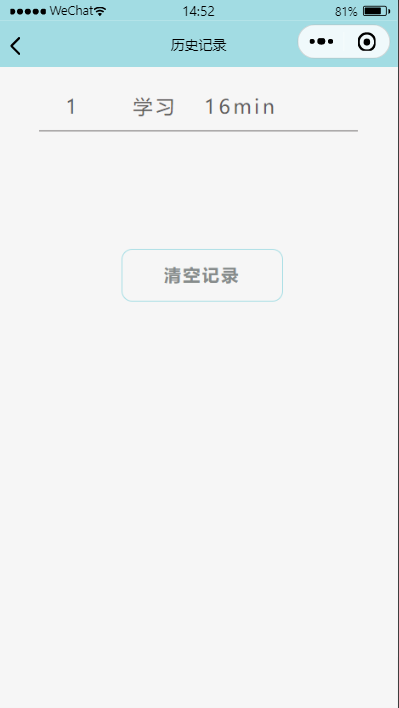


图14 历史记录页面

（1）在pages目录下新建一个history文件，然后进入到pages/history/history目录下，清空history.wxml、history.js、history.wxss文件默认生成的内容。

（2）进入到pages/history/history.wxml页面，有相关的历史记录和清空记录按钮，具体代码如下所示。

<view class="content" wx:for="{{histories}}">

<view class="index">{{index+1}}</view>

<view class="images">{{item.image}}</view>

<view class="text">{{item.text}}</view>

<view class="text">{{item.times}}</view>

</view>

<view class="delete" bindtap="delete" id="{{item.id}}">清空记录</view>

（3）进入到pages/history/history.wxss页面，给页面内容添加样式。

（4）进入到pages/zhuangzhu/zhuangzhu.js文件里，绑定一个清空记录按钮具体代码如下所示。

onLoad: function (options) {

var \_this=this;

// 引用数据库

const db = wx.cloud.database({

// 环境ID

env:'cloud1-7gvfbdsx591cc716'

})

// 开始查询数据 news对应的是集合名称

db.collection('history').get({

// 查询成功

success:res=>{

console.log(res.data);

// 很重要，给histories数组赋值，没有这一步，前台就不会显示值

\_this.setData({

histories:res.data

})

}

})

},

delete:function(){

var \_this=this

const db = wx.cloud.database({

// 环境ID

env:'cloud1-7gvfbdsx591cc716'

})

db.collection('history').where({all:null}).remove()

}

### 项目预览

如果项目中有AppID的，就可以将项目进行上传，在手机上就可以浏览自律养成记微信小程序，如图15所示。



图15 专注页面

### 我的

### 微信授权功能设计

自律养成记的头像可以获得微信授权，如图10所示。

图10

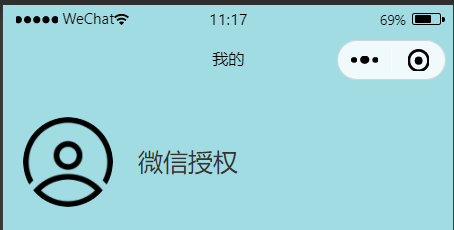
（1）新建一个zilv项目，AppID为wx0720a9d856aaef3d，把准备好的图片放置在zilv项目里；

图 16 微信授权

（2）进入到wode.wxss界面里，将窗口背景色设置为蓝色（#A2DCE3），标题改为我的，字体颜色设置为黑色（#333333），具体代码如下所示。

.head {

display:flex;

width: 100%;

background-color: #A2DCE3;

align-items: center;

}

.userinfo-nickname {

color: #333333;

font-size: 42rpx;

}

（3）进入到pages/index/index目录下，清空index.wxml、index.js、index.wxss文件默认生成的内容。

（4）进入到pages/wode/wode.wxml文件里，设计顶部头像昵称，包括两方面内容：微信头像、微信昵称，具体代码如下所示。

<view class='head head-height'>

<block wx:if="{{!hasUserInfo && canIUse}}">

<view class="userinfo">

<image class="userinfo-avatar" src="../../images/tubiao/user/用户 (1).png" mode="cover"></image>

<text class="userinfo-nickname">微信授权</text>

</view>

<button class="login-button head-height" open-type="getUserInfo" bindgetuserinfo="getUserInfo"> 获取头像昵称 </button>

</block>

<block wx:else>

<view class="userinfo">

<image bindtap="bindViewTap" class="userinfo-avatar" src="{{userInfo.avatarUrl}}" mode="cover"></image>

<text class="userinfo-nickname">{{userInfo.nickName}}</text>

</view>

</block>

</view>

（5）进入到pages/wode/wode.wxss文件里，给头像内容添加样式，具体代码如下所示。

.userinfo {display: flex;flex-direction: row;align-items: center;width: 100%;margin-left: 200rpx;}

.userinfo-avatar {width: 150rpx;height: 150rpx;border-radius: 50%;margin-left: -80px;

margin-right: 20px;}

这样就可以实现微信授权效果效果，如图11所示。

图11

### 菜单点击功能设计

菜单可以进行点击，实现菜单切换效果，单击不同的页签，，同时菜单对应的内容也随着进行切换。

（1）在菜单里设计样式，绑定单击事件wx.navigateTo；

（2）进入到pages/wode/wode.wxml文件，编写大概代码，具体代码如下所示。

<view class="mine-wrapper">

<view class="list-wrapper">

<view class="alllists">

<block wx:for="{{menuitems}}" wx:key="menu\_for">

<navigator url="{{item.url}}" class="lists">

<view class='content' bindtap="{{item.num1}}">

<view class="listimg">

<image src="{{item.icon}}"></image>

</view>

<view class="listtext">{{item.text}}</view>

<view class="listicon">{{item.tips}}</view>

<view class="arrows">

<image src="{{item.arrows}}"></image>

</view>

</view>

</navigator>

<view class="divLine"></view>

</block>

</view>

</view>

</view>

1. 添加页签菜单单击绑定事件wx.navigateTo，具体代码如下所示。

data: {

userInfo: {},

hasUserInfo: false,

canIUse: wx.canIUse('button.open-type.getUserInfo'),

menuitems: [

{ text: '日历', url: '#', icon: '/images/tubiao/user/日历.png', tips: '',

arrows: '/images/tubiao/user/箭头.png',num1:'num2'

},{ text: '安全', url: '#', icon: '/images/tubiao/user/安全.png', tips: '',

arrows: '/images/tubiao/user/箭头.png',num1:'num6'

},{ text: '关注服务号及提醒设置', url: '#', icon: '/images/tubiao/user/服务号-营销推广.png', tips: '',

arrows: '/images/tubiao/user/iconfont-icon (1).png',num1:'num7'

},{ text: '个性化', url: '#', icon: '/images/tubiao/user/个性化.png', tips: '',

arrows: '/images/tubiao/user/箭头.png',num1:'num8'

},{ text: '操作说明', url: '#', icon: '/images/tubiao/user/操作日志.png', tips: '',

arrows: '/images/tubiao/user/箭头.png' ,num1:'num3'

},{ text: '隐私政策', url: '#', icon: '/images/tubiao/user/隐私协议.png', tips: '',

arrows: '/images/tubiao/user/箭头.png' ,num1:'num4'

},{ text: '吐槽建议', url: '#', icon: '/images/tubiao/user/建议.png', tips: '',

arrows: '/images/tubiao/user/箭头.png' ,num1:'num5'

}]},

num2: function (options) {wx.navigateTo({url: '/pages/rili/rili',})},

num3: function (options) {wx.navigateTo({url: '/pages/caozuo/caozuo',}) },

num4: function (options) {wx.navigateTo({url: '/pages/yinsi/yinsi',}) },

num5: function (options) {wx.navigateTo({url: '/pages/jianyi/jianyi',}) },

num6: function (options) {wx.navigateTo({url: '/pages/anquan/anquan',}) },

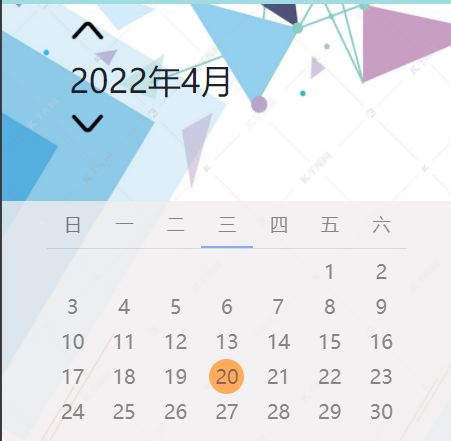
num7: function (options) {wx.navigateTo({url: '/pages/guanzhu/guanzhu',}) },

num8: function (options) {wx.navigateTo({url: '/pages/gexing/gexing',}) },

这样点击菜单会进行跳转

### 日历功能设计

日历页面是用来显示日期的页面，显示出星期以及日期对应以及当天日期高亮，点击箭头可以进行日历翻页等功能，如图12所示。

图12

1. 在pages目录下面，新建一个rili文件夹，然后进入到pages/rili/rili.wxml，具体代码如下所示。

<view class='calendar\_title flex\_nowrap'>

<view class='icon' bindtap='lastMonth'>

<image src='/images/tubiao/rili/num7.png' />

</view>

<view>{{year}}年{{month}}月</view>

<view class='icon' bindtap='nextMonth'>

<image src='/images/tubiao/rili/num8.png' />

</view>

</view>

<view class='calendar'>

<view class='header'>

<view wx:for='{{date}}' wx:key='' class='{{(index == todayIndex) && isTodayWeek ? "weekMark" : ""}}'>{{item}}

<view></view>

</view>

</view>

<view class='date-box'>

<block wx:for='{{dateArr}}' wx:key=''>

<view class='{{isToday == item.isToday ? "nowDay" : ""}}'>

<view class='date-head' bindtap='lookHuoDong' data-year='{{year}}' data-month='{{month}}' data-datenum='{{item.dateNum}}'>

<view>{{item.dateNum}}</view>

</view>

</view>

</block>

</view>

</view>

<image class='background' src="/images/tubiao/rili/back2.jpg" mode="aspectFill"></image>

（2）进入到pages/rili/rili.wxss文件，给日历添加样式，具体代码如下所示。

.calendar\_title{width: 70%; margin: 10rpx auto; justify-content: space-between;font-size: 20pt;color: rgb(30, 31, 44);}

.calendar\_title .icon image{width: 60rpx; height: 60rpx; margin: 5rpx auto;}

.calendar{width: 100%;margin-top: 75rpx;background: rgb(240, 237, 237);border-bottom-right-radius: 20rpx;border-bottom-left-radius: 20rpx;padding: 0 0 20rpx;font-family: FZZhunYuan-M02S;opacity: 0.80; filter: alpha(opacity=10);}

.header{font-size: 0;width: 80%;margin: 0 auto;}

.header>view{display: inline-block;width: 14.285%;color: #666;font-size: 30rpx;text-align: center;border-bottom: 1px solid #D0D0D0;padding: 20rpx 0;}

.weekMark{position: relative;width: 80%;margin: 0 auto;}

.weekMark view{position: absolute;bottom: 0;left: 0;width: 100%;border-bottom: 2px solid #69f;}

.date-box{font-size: 0;padding: 10rpx 0;width: 80%;margin: 0 auto;}

.date-box>view{position: relative; display: inline-block;width: 14.285%;color: #666;text-align: center;vertical-align: middle;}

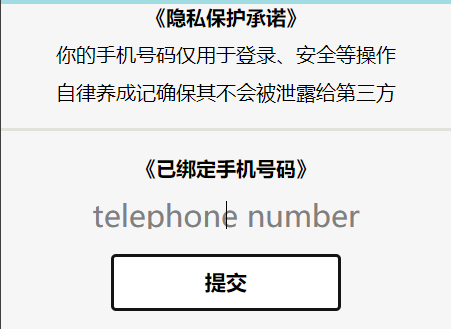
.date-head{height: 60rpx;line-height: 60rpx;font-size: 12pt;}

.nowDay .date-head{width: 60rpx;border-radius: 50%;text-align: center;color: rgb(107, 54, 54);background-color: #ff9933;margin: 0 auto;}

.background {width: 100%;height: 100%;position:fixed;background-size:100% 100%;z-index: -1;}

### 绑定手机号上传数据库能设计

绑定手机号功能是将手机号传到数据库中，需要云数据库，如图13所示。

图13

1. 在pages目录下面新建一个文件夹anquan，具体代码如下所示。

<view class="num1">

<view class="num2">《隐私保护承诺》</view>

<view class="num3">你的手机号码仅用于登录、安全等操作</view>

<view class="num4">自律养成记确保其不会被泄露给第三方</view>

</view>

<view class="divLine"></view>

<view class="num5">《已绑定手机号码》</view>

<view class="selfTwo" wx:for="{{feelList}}" wx:key="{{index}}">

<view class="right">{{item.feel}}</view>

</view>

<input type="number" placeholder="telephone number" name="haoma" value="{{haoma}}" class="num6" bindinput="addRname"></input>

<button bindtap="submit" class="register-btn">提交</button>

1. wxss添加样式，具体代码如下所示。

.num2{text-align: center;font-weight: bold;}

.num3{text-align: center;margin-top: 20rpx;}

.num4{text-align: center;margin-top: 20rpx;}

.num5{text-align: center;font-weight: bold;margin-top: 40rpx;}

.num6{text-align: center;font-size: 50rpx;margin-top: 35rpx;}

.divLine{background: #E0E3DA;width: 100%;height: 5rpx;margin-top: 40rpx;}

.register-btn{border: 3px solid #141414; background-color: white;margin-top: 20px;}

.register-btn::after{border: none; }

1. 在js中添加代码，使用云数据库，具体代码如下所示。

data: {haoma:''},

addRname(event) {console.log("电话：", event.detail.value)

Rname1 = event.detail.value },

submit: function (e) {let that = this;

db.add({

data: {Rname: Rname1,},

success: function (res) {console.log("上传成功", res)}}) },

resetSalaryBtnClick: function (e) {

this.setData({haoma: "",})}

### 个性化功能设计

个性化功能可以调整小程序的细节，通过点击按钮来进行操作，如图14所示。

图14 页面分享

进入到pages/gexing/gexing.wxml文件里，具体代码如下所示。

<view style="background-color:#faf9f9;height:600px;" class="num1">

<view style="padding-top: 10px;"></view>

<view style="display:flex;flex-direction: row;background-color: #ffffff;height:50px;line-height: 50px;">

<view style="font-weight: bold;" class="num2">打卡震动</view>

<view style="position: absolute;right:10px;">

<switch type="switch" checked />

</view>

</view>

<view style="height:1px;background-color: #f2f2f2;opacity:0.2"></view>

<view style="display: flex;flex-direction: row;background-color: #ffffff;height:50px;line-height: 50px;">

<view style="font-weight: bold;" class="num3">打卡音效</view>

<view style="position: absolute;right:10px">

<switch type="switch" />

</view>

</view>

<view style="height:1px;background-color: #f2f2f2;opacity: 0.2;"></view>

<view style="margin-top: 20px;"></view>

<view style="height:1px;background-color: #f2f2f2;opacity: 0.2;"></view>

<view style="display: flex;flex-direction: row;background-color: #ffffff;height:50px;line-height: 50px;">

<view style="font-weight:bold" class="num4">已完成目标划线</view>

<view style="position: absolute;right:10px;">

<switch type="checkbox" checked />

</view>

</view>

<view style="height:1px;background-color: #f2f2f2;opacity: 0.2;"></view>

<view style="height:1px;background-color: #f2f2f2;opacity: 0.2;"></view>

<view style="display: flex;flex-direction: row;background-color: #ffffff;height:50px;line-height: 50px;">

<view style="font-weight:bold" class="num5">目标列表中隐私状态图标</view>

<view style="position: absolute;right:10px;">

<switch type="checkbox" checked />

</view>

</view>

<view style="height:1px;background-color: #f2f2f2;opacity: 0.2;"></view>

<view style="margin-top: 20px;"></view>

<view style="height:1px;background-color: #f2f2f2;opacity: 0.2;"></view>

<view style="height:1px;background-color: #f2f2f2;opacity: 0.2;"></view>

<view style="display: flex;flex-direction: row;background-color: #ffffff;height:50px;line-height: 50px;">

<view style="font-weight:bold" class="num6">朋友圈照片更新</view>

<view style="position: absolute;right:10px;">

<switch type="checkbox" checked />

</view>

</view>

<view style="height:1px;background-color: #f2f2f2;opacity: 0.2;">

</view>

</view>

微信开发者单击按钮时，界面效果如图15所示。

图15

### 提问功能设计

（1）进入tiwen.wxml文件，基础代码如下。

<view class='baseInfo'>

<image src='/images/tubiao/user/建议.png'></image>

<text>提问</text>

</view>

<form bindsubmit="formSubmit">

<view class='name'>

<text >您的疑问：</text>

<textarea name="name" type='text' placeholder-style='font-size:28rpx;' placeholder='请输入...'></textarea>

</view>

<button form-type='submit'>提交</button>

</form>

<!-- 模态框 -->

<modal class="modal" title='要提交的信息' hidden="{{modalHidden}}" bindcancel='modalCancel' bindconfirm='modalConfirm'>

<view>你的提问是:{{information.name}}</view>

</modal>

1. 进入tiwen.wxss文件，基础代码如下。

page{background: #eee;}

.baseInfo{padding:20rpx 10rpx;}

.baseInfo image{width: 10rpx;height: 30rpx;vertical-align: middle;margin-right: 10rpx;}

.baseInfo text{font-size: 28rpx;color: #6f6f6f;}

.list{color: #6f6f6f;font-size: 32rpx;}

.icon{background: #fff;display: flex;align-content: center;align-items: center;justify-content: space-between;padding: 20rpx;border-bottom: 1rpx solid #eee;}

.icon-img{display: flex;align-content: center;align-items: center;}

.icon-head{width: 120rpx;height: 120rpx;border-radius: 50%;margin-right: 30rpx;}

.icon-r{width: 20rpx;height: 30rpx;}

.name,.sex,.phone,.info{background:#fff;display:flex;justify-content:space-between;align-content: center;align-items: center;padding: 20rpx;border-bottom: 1rpx solid #eee;}

.name input,.phone input,.info input{text-align:right;}

.sex{background: #fff;}

.radio radio{margin-left: 30rpx;}

button{border: 3px solid #141414; border-radius: 15px; background-color: white;margin-top: 20px;}

button::after{border: none; }

1. 进入tiwen.wxss文件，基础代码如下。

data: {

touxiang: 'https://manager.diandianxc.com/diandianxc/mrtx.png',

icon\_r: 'https://manager.diandianxc.com/mine/enter.png',

isSex:"0",information:[],userSex:'',modalHidden:true},

radioChange(e){console.log(e.detail.value);var sexName=this.data.isSex

this.setData({isSex:e.detail.value})},

formSubmit(e){console.log(e);var userSex=this.data.isSex==0?'男':'女';

var information= e.detail.value;console.log(userSex);

this.setData({

information: e.detail.value,userSex,modalHidden:false});},

modalCancel(){wx.showToast({title: '取消提交',icon:'none'})

this.setData({modalHidden:true,})},

modalConfirm() {wx.showToast({title: '提交成功',icon:'success'})

this.setData({modalHidden: true})}

### 吐槽功能设计

吐槽功能能够发送图片以及文字进入后台，并得到已发送的建议，若未输入文字或图片时，会显示输入错误，如图16所示

图16

（1）进入tiwen.wxml文件，基础代码如下。

<Tabs tabs="{{tabs}}" bindtabsItemChange="handleTabsItemChange">

<view class="fb\_main">

<view class="fb\_title"></view>

<view class="fb\_tips">

<text >功能建议</text><text >购买遇到问题</text>

<text >性能问题</text><text >其他</text>

</view>

<view class="fb\_content">

<textarea value="{{textValue}}" bindinput = "handleTextInput" placeholder="请描述一下问题"></textarea><view class="fb\_tool">

<button bindtap = "handleimg" >+</button>

<view class="up\_img\_item" wx:for="{{chooseimg}}" wx:key="\*this" bindtap = "handleremoveimg" data-index="{{index}}">

<UpImg src="{{item}}"></UpImg>

</view></view></view>

<view class="form\_btn\_wrap">

<button bindtap="handlefrom">

<icon type="success\_no\_circle"color="lightblue"></icon>

提交

</button></view></view></Tabs>

（2）进入tiwen.wxml文件，基础代码如下。

page{background-color: #d9edf3;}

.fb\_main {padding:20rpx;color: #666;}

.fb\_tips {display: flex;flex-wrap: wrap;}

text {width: 30%;padding: 10rpx;background-color: #fff;text-align: center;margin:20rpx 10rpx;}

.fb\_content {background-color: #fff;margin-top: 20rpx;}

textarea {padding: 10rpx;}

.fb\_tool {display: flex;flex-wrap: wrap;padding-bottom: 30rpx;}

button{margin: 0;width: 90rpx;height: 90rpx;font-size: 60rpx;padding: 0;display: flex;justify-content: center;align-items: center;margin-left: 20rpx;margin-top: 20rpx;color: rgb(22, 20, 20);}

.up\_img\_item{margin-left: 20rpx;margin-top: 20rpx;}

.form\_btn\_wrap {margin-top: 20rpx;display: flex;justify-content: flex-end;}

button{margin: 0;width: 38%;color: lightblue;background-color: var(--themeColor); }

（3）进入tiwen.wxml文件，基础代码如下。

data: {

tabs: [{id: 0,value: "体验问题",isActive: true,},

{id: 1,value: "商品、商家投诉",isActive: false,},],

textValue: "",chooseimg: [],upLoadImg: "",},

handleTabsItemChange(e) {const { index } = e.detail;let { tabs } = this.data;tabs.forEach((v, i) =>i === index ? (v.isActive = true) : (v.isActive = false));

this.setData({tabs,});},

handleimg(e) {console.log(e);wx.chooseImage({count: 9,sizeType: ["original", "compressed"], sourceType: ["album", "camera"], success: (result) => {console.log(result);

this.setData({chooseimg: [...this.data.chooseimg, ...result.tempFilePaths],});},

fail: () => {},complete: () => {},});},

handleremoveimg(e) {console.log(e);

const { index } = e.currentTarget.dataset;

let { chooseimg } = this.data;

chooseimg.splice(index, 1);

this.setData({chooseimg,});},

handleTextInput(e) {this.setData({textValue: e.detail.value,});},

handlefrom(e) {const { textValue, chooseimg } = this.data;

if (!textValue.trim()) {wx.showToast({title: "输入有误",icon: "none",mask: true,});

return;}

wx.showLoading({title: "正在上传中",mask: true,});

if (chooseimg.length != 0) {chooseimg.forEach((v, i) => {wx.uploadFile({

url: "https://imgchr.com/", filePath: v, name: "file",formData: {},

success: (result) => {console.log(result);

if (i === chooseimg.length - 1) {console.log("提交给后台，将回到上一界面");wx.hideLoading();

this.setData({textValue: "",chooseimg: [],});

wx.navigateBack({delta: 1,});}},});

wx.getNetworkType({success: (result) => {console.log(result)},

fail: () => {},

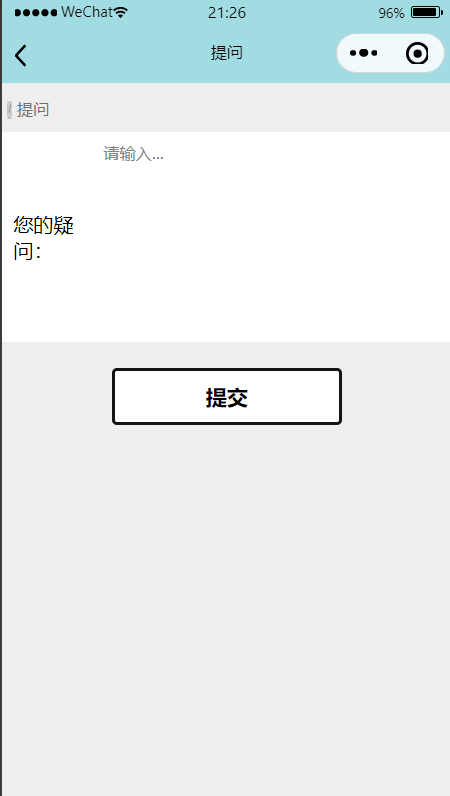
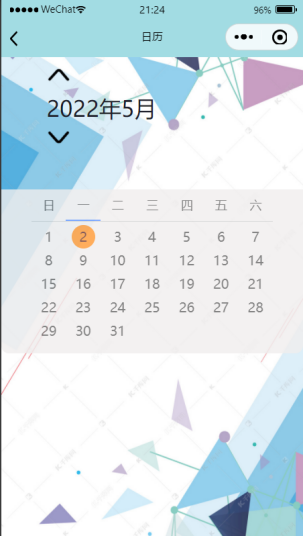
complete: () => {}});});

} else {wx.hideLoading();console.log("只提交了文本");

wx.navigateBack({delta: 1,});}},

### 项目预览

如果项目中有AppID的，就可以将项目进行上传，在手机上就可以浏览自律养成记微信小程序，如图18所示。

图18

### 分享

### 自律通知页面设计

通知页面是用来显示自律朋友发布的日常内容，每条内容包括4项内容：发布人头像、发布人昵称、发布人所发内容、发布人发表时间、发布人所发地址；如图 所示。



图19 通知页面内容

（1）在pages/index目录下面，新建一个index.wxml文件，然后进入到pages/index/index.wxml文件里，添加轮播图，具体代码如下所示。

<view class="img">

<swiper indicator-dots autoplay interval='6000' duration='3000'>

<swiper-item>

<image src="../image/01.jpg" style="width: 100%;height: 230px;"></image></swiper-item>

<swiper-item>

<image src="../image/02.jpg" style="width:100%;height:230px;"></image></swiper-item>

<swiper-item>

<image src="../image/03.jpg" style="width:100%;height:230px;"></image></swiper-item>

</swiper>

</view>

（2）进入到pages/index/index.wxml文件，设计发布人头像、发布人昵称、发布人地址、发布人发布的照片、图标，具体代码如下所示。

<block wx:for="{{actionsList}}" wx:for-index="numm" wx:key="index">

<view class="message">

    <view class="left">

       <image class="face-img" src="{{item.faceImg}}"></image>

    </view>

    <view class="right">

    <view class="nickname">{{item.nickName}}</view>

       <view>

        {{item.text}}

      </view>

      <view class="imags-wrap">

<block wx:if="{{item.images.length == 1}}" wx:for="{{item.images}}" wx:for-item="imgItem" wx:key="index" wx:for-index="indexx">

  <!-- 单图布局 -->

  <image src="{{imgItem}}" class="image-1"  bindtap="previewImg" data-src="{{imgItem}}" data-index="{{numm}}"></image>

</block>

<block wx:if="{{item.images.length == 2 || item.images.length ==3}}" wx:for="{{item.images}}" wx:for-item="imgItem" wx:key="index">

  <!-- 2-3张图布局 -->

  <image src="{{imgItem}}" class="image-2-3" bindtap="previewImg" data-src="{{imgItem}}" data-index="{{numm}}"></image>

</block>

<block wx:if="{{item.images.length == 4}}" wx:for="{{item.images}}" wx:for-item="imgItem" wx:key="index">

  <!-- 4图布局 -->

  <image src="{{imgItem}}" class="image-4" bindtap="previewImg" data-src="{{imgItem}}" data-index="{{numm}}"></image>

</block>

<block wx:if="{{item.images.length >= 5}}" wx:for="{{item.images}}" wx:for-item="imgItem" wx:key="index">

  <!-- 5-9张图布局 -->

  <image src="{{imgItem}}" class="image-5-9" bindtap="previewImg" data-src="{{imgItem}}" data-index="{{numm}}"></image>

</block>

     </view>

       <view class="time-delete">

       <view class="time"> {{item.time}}</view>

       <view class="delete"  wx:if="{{Openid == item.\_Openid}}" catchtap="deleteAction" data-id="{{item.\_id}}">删除</view>

       </view>

       <view wx:if="{{item.address}}" class="addreess" bindtap="opendLocation" data-index="{{numm}}">{{item.address}}</view>

       <view class="hudong-box">

           <view class="hudong-cell">

           <image src="../image/780.jpg" class="hudong-img"></image>

           <view>分享</view>

           </view>

           <view class="hudong-cell">

            <image src="../image/10.png" class="hudong-img"></image>

           <view>评论</view>

           </view>

           <view class="hudong-cell">

            <image src="../image/11.png" class="hudong-img"></image>

           <view>点赞</view>

           </view>

       </view>

    </view>

</view>

</block>

（3）进入到pages/index/index.wxss文件里，发布人头像、发布人昵称、发布人地址、发布人发布的照片、图标添加样式，具体代码如下所示。

.face-img{

  width: 100rpx;

  height: 100rpx;

  padding: 20rpx 20rpx 40rpx 20rpx;

}

.message{

  display: flex;

  flex-direction: row;

}

.left{

  margin-right: 15rpx;

}

.nickname{

  color: #586B94;

  font-weight: 550;

  font-size: 34rpx;

  margin-bottom: 15rpx;

}

.time-delete{

  display: flex;

  flex-direction: row;

  font-size: 28rpx;

}

.time-delete{

  margin: 10rpx 0rpx 10rpx 0rpx;

}

.time{

  color: #7f7f7f;

  margin-right: 20rpx;

}

.delete{

  color:#586B94;

}

.hudong-img{

  width: 40rpx;

  height: 40rpx;

}

.hudong-box{

  display: flex;

  flex-direction: row;

  justify-content: space-between;

  color: #7f7f7f;

  margin: 10rpx 0prx 10rpx 0rpx;

}

.hdong-cell{

  display: flex;

  flex-direction: row;

  align-items: center;

}

.imags-wrap{

  margin-top: 10rpx;

}

.image-1{

  height:500rpx;

  width: 550rpx;

}

.image-2-3 {

  width: 180rpx;

  height: 180rpx;

  margin-right: 6rpx;

}

.image-4 {

  width: 240rpx;

  height: 240rpx;

  margin-right: 6rpx;

}

.image-5-9 {

  width: 180rpx;

  height: 180rpx;

  margin-right: 6rpx;

}

.prize-box{

  background-color: #f7f7f7;

  margin-top: 20rpx;

  display: flex;

  flex-direction: row;

  flex-wrap: wrap;

  align-items: center;

  padding: 20rpx 20rpx;

  color: #586B94;

  font-size: 30rpx;

  border-bottom:1rpx solid #ededed ;

}

.prize-img{

  width: 30rpx;

  height: 30rpx;

  margin-right: 10rpx;

}

.comment-box{

  background-color: #f7f7f7;

  padding: 20rpx  20rpx;

  font-size: 30rpx;

}

.comment-nickname{

  color: #586b94;

}

.camer-box{

  text-align: right;

  margin: 5rpx 20rpx 0rpx 0rpx;

}

.camer-img{

  width: 64rpx;

  height: 55rpx;

}

界面效果如图 所示。



图20 发布人头像、昵称、位置、图标

（4）进入到pages/auth/auth.js文件里，设计授权登录，具体代码如下所示。

<button bindtap="getInfo">授权登录</button>

（5）进入到pages/auth/auth文件里，可以实现获取用户信息，让后成功返回通知也买你具体代码如下所示。

 getInfo(e){

       wx.getUserProfile({

           desc:'获取用户信息',

           success(res){

               console.log(res)

                app.globalData.userInfo =res.userInfo

            wx.navigateBack({

                success(res){

                    wx.showToast({

                    title: '授权成功',

                    })

                }

            })

           }

       })

    }

界面效果如图 所示

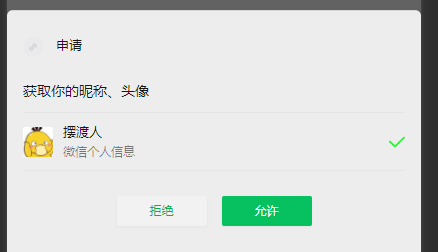


图21 获取用户信息内容

控制台如图 所示

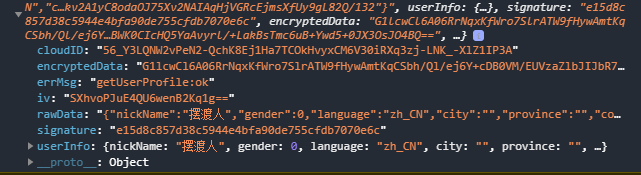


图22 控制台内容

（6）进入到pages/publish/publish.wxml文件里，设计发布框、添加按钮、位置选择器，具体代码如下所示。

<view class="input-wary">

  <textarea placeholder="这一刻的idear..... " class="input-text" maxlength="2000" bindinput="getValue"></textarea>

</view>

//发布内容

<view class="img-wary">

  <block wx:for="{{cloudImgList}}" wx:key="index">

  <view class="one-img">

  <image src="{{item}}" class="img-choose"></image>

  <image src="../image/delete.png" class="close-img" bindtap="deleteImg" data-index="{{index}}"></image>

  </view>

  </block>

  <view class="one-img" bindtap="chooseImage"  wx:if="{{cloudImgList.length < 9}}">

  <image src="../image/add.png" class="img-choose"></image>

  </view>

//图片内容

</view>

<view bindtap="chooseAddress" class="choose-localtion">{{localtion}}</view>

//位置内容

（7）进入到pages/publish/publish.wxss文件，给发布框、添加按钮、位置选择器添加样式，具体代码如下所示。

.pub-wary{

    display: flex;

    flex-direction: row;

    justify-content: flex-end;

    margin-right: 20rpx;

}

.pub-text{

    background-color: #07c160;

    color: #fff;

    width: 120rpx;

    height: 60rpx;

    border-radius: 10rpx;

    text-align: center;

    line-height: 60rpx;

    font-size: 28rpx;

}

.input-wary{

    margin: 20rpx 40rpx;

}

.input-text{

    height: 200rpx;

    width: 100%;

}

.img-wary{

    display: flex;

    flex-direction: row;

    flex-wrap: wrap;

    margin: 20rpx 30rpx;

}

.img-choose{

    width: 210rpx;

    height: 210rpx;

}

.one-img{

    position: relative;

    margin:4rpx 8rpx;

}

.close-img{

    width: 30rpx;

    height: 30rpx;

    position: absolute;

    right: 4rpx;

    top: 4rpx;

}

.imgItem {

    margin: 6rpx;

    position: relative;

    width: 176rpx;

    height: 176rpx;

    background: gainsboro;

  }

  .photo {

    width: 50%;

    height: 50%;

    margin: 25%;

  }

  .choose-localtion {

    margin: 50rpx 40rpx;

    font-size: 36rpx;

    color: rgb(90, 90, 90);

  }

  .addreess {

    color: #0f4ed8;

    font-size: 30rpx;

  }

界面效果如图 所示。



图23 发布内容

### 发布动态页面设计。

（1）在pages/publish目录下面新建一个页面publish.js，用来实现位置的获取，具体代码如下所示。

data: {

        cloudImgList:[],

        localtion:'请选择位置',

        addresssInfo:{}

    },

    chooseAddress(){

        var that = this;

        wx.chooseLocation({

          success(res){

              console.log(res)

              that.setData({

                  localtion:res.address+res.name,

                  addresssInfo:res

              })

              if(res.address==""){

                that.setData({

                  localtion:"请选择位置"

                })

              }

          }

        })

（2）在pages/publish目录下面新建一个页面publish.js，用来实现图片的上传和判断图片数量的获取，具体代码如下所示

 uploadImages(){

        var that = this;

        for(var i = 0 ; i< this.data.tempImgList.length; i++){

          wx.cloud.uploadFile({

            cloudPath:`actionImages/${Math.random()}\_${Date.now()}.${this.data.tempImgList[i].match(/\.(\w+)$/)[1]}`,

            filePath: this.data.tempImgList[i],

            success(res){

              console.log(res.fileID)

              that.data.cloudImgList.push(res.fileID)

              that.setData({

                cloudImgList:that.data.cloudImgList

              })

            }

          })

        }

      },

1. 在pages/publish目录下面新建一个页面publish.js，用来实现删除图片，具体代码如下所示

 deleteImg(e){

        console.log(e.currentTarget.dataset.index)

        this.data.cloudImgList.splice(e.currentTarget.dataset.index,1)

        this.setData({

          cloudImgList: this.data.cloudImgList

        })

      },

### 实现发布成功设计

通知页面功能设计，在发布页面中，用户成功上传信息到云后台，再通过云后台来实现到页面如图 所示。

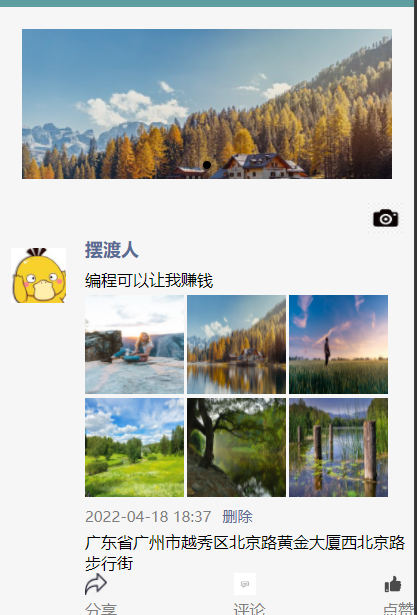


图24 成功发布

进入到pages/index/index.js文件里，添加onPullDownRefresh来实现下拉刷新，具体代码如下所示。

 onPullDownRefresh:function(){

    this.getActionsList()

  }

getActionsList(){

    var that = this

    wx.cloud.database().collection('actions').orderBy('time','desc').get({

      success(res){

        console.log(res)

        var list = res.data

        for(var l in list){

          list[l].time = util.formatTime(new Date(list[l].time))

        }

        that.setData({

          actionsList :list

        })

      }

    })

  },

获取数据库内容，数据库如图 所示。

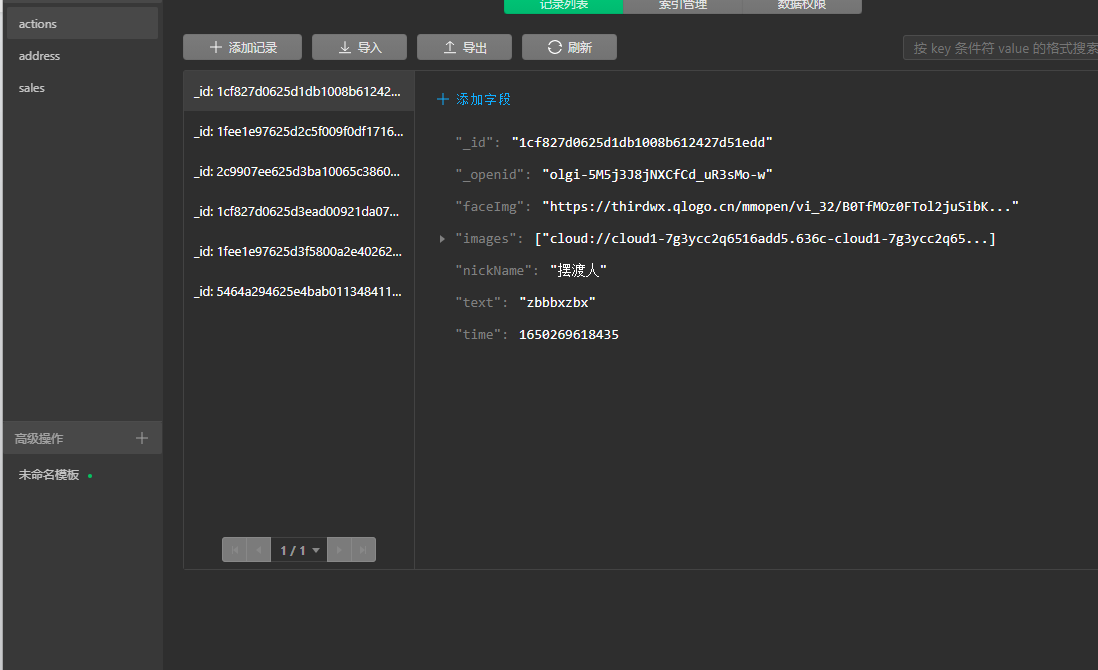


图25 后台数据库

### 动态删除设计

如果动态中有用户不喜欢的信息，可以选择删除，再用户删除后自动刷新页面。如图 所示。





图27 删除

## 5． 小结

本章主要设计了自律养成记微信小程序，重点掌握以下内容：

1. 学会布局和给界面添加相关样式。
2. 学会canvas绘图和定时器设置。
3. 学会一个页面向另一个页面进行传值。

（4）实现微信授权功能，需要借助getUserInfo: function (e)及setUserInfo: function (userInfo)等js，菜单点击功能，需要wx.navigateTo等页面跳转js，日历功能，需要onLoad: function ()、dateInit: function (setYear, setMonth)、lastMonth: function ()、nextMonth: function ()等方法，绑定手机号上传数据库功能，需要submit: function (e)、addRname(event)等方法，个性化功能，需要button按钮进行js以及wxss。提问功能，点击按钮可以进行提问，需要formSubmit(e)、radioChange(e)等方法。吐槽功能，点击可以上传图片及文字，需要handleTabsItemChange(e)等方法。