



FZUGO

服务于FZUer的校园导航



# 目录

01 Bug修复及优化

02 完成情况

03 测试情况

04 未来规划



# 第一部分

## Bug修复及优化

Bug改进情况已通过修复事件绑定、优化性能、确保数据实时更新和增强用户反馈功能等措施显著提升了系统的稳定性和用户体验。同时对部分界面进行了优化。

## Bug1: 搜索功能未绑定事件处理程序

代码片段:

```
<input class="card-top-input" value="{{start.name}}"  
placeholder="当前地点/起点" disabled="true" data-search_id="  
{{1}}" bindtap="tosearch" />  
<input class="card-top-input" value="{{end.name}}"  
placeholder="请选择终点" disabled="true" data-search_id="{{w}}"  
bindtap="tosearch" />
```

问题:

bindtap="tosearch" 事件可能未正确绑到 tosearch 函数，导致用户点击输入框时搜索功能失败。

解决方案:

```
// 确保在后端逻辑中有对应的tosearch函数  
Page({  
  tosearch: function(event) {  
    // 搜索逻辑  
  }  
});
```

## Bug2: 类别选择没有响应

代码片段:

```
<view class="card-bottom-category-label" {{index == category ?  
'choose' : ''}} wx:for="{{site_data}}" wx:key="id" id="  
{{index}}" bindtap="changeCategory">  
  {{item.name}}  
</view>
```

问题:

bindtap="changeCategory" 事件可能没有正确绑定到 changeCategory 函数，导致用户点击类别标签时无法切换类别。

解决方案:

```
// 确保在后端逻辑中有对应的 changeCategory 函数  
Page({  
  changeCategory: function(event) {  
    // 类别切换逻辑  
  }  
});
```

## Bug3:轮播图性能问题

代码片段:

```
<swiper class="list-landscape-container-swiper" next-  
margin="20px" indicator-dots="{{indicatorDots}}" autoplay="  
{{autoplay}}" circular="{{circular}}" interval="{{interval}}"  
duration="{{duration}}">  
  <block wx:for="{{background}}" wx:key="*this">  
    <swiper-item>  
      <view class="list-landscape-container-swiper-item">  
        <image class="list-landscape-container-swiper-item-  
img" mode="widthFix" bindload="imgHeight" src="{{item}}"  
bindtap="lookPhoto" data-src="{{item}}"/></image>  
      </view>  
    </swiper-item>  
  </block>  
</swiper>
```

问题:

如果 background 数组中的图片过大，可能会导致轮播图性能问题，如加载缓慢或卡顿。

解决方案:

```
// 优化图片加载逻辑  
Page({  
  imgHeight: function(event) {  
    // 根据实际需要调整图片高度  
  }  
});
```

## Bug4:天气信息未及时更新

代码片段:

```
<view class="card-container-weather-tmp">  
  <view>{{now.temp}}°</view>  
  <image class="card-container-weather-tmp-icon"  
src="https://icons.qweather.com/assets/icons/{{now.icon}}.svg"  
/></image>  
</view>
```

问题:

天气信息可能未实时更新，导致显示过时的数据。

解决方案:

```
// 定时更新天气数据  
Page({  
  updateWeather: function() {  
    // 获取最新天气数据并更新视图  
  }  
});
```

## Bug5:地图漏洞修补及优化

问题1：  
地图重复初始化。

解决方案：

```
import { useEffect, useRef } from 'react';
import { Map, View } from 'ol';
import TileLayer from 'ol/layer/Tile';
import OSM from 'ol/source/OSM';

function MapComponent() {
  const mapRef = useRef<HTMLDivElement>(null);
  const mapInstance = useRef<Map | null>(null);

  useEffect(() => {
    // 修复地图初始化时的内存泄漏问题
    if (!mapRef.current || mapInstance.current) return;

    const map = new Map({
      target: mapRef.current,
      layers: [
        new TileLayer({
          source: new OSM(),
        }),
      ],
      view: new View({
        center: [0, 0],
        zoom: 2,
      }),
    });
    mapInstance.current = map;
  });

  // 修复地图销毁时的清理问题
  return () => {
    if (mapInstance.current) {
      mapInstance.current.setTarget(undefined);
      mapInstance.current = null;
    }
  };
}
```

问题2：  
处理空坐标输入的边界情况：计算地图的边界范围，  
以防止因空数组引起的程序崩溃。

解决方案：

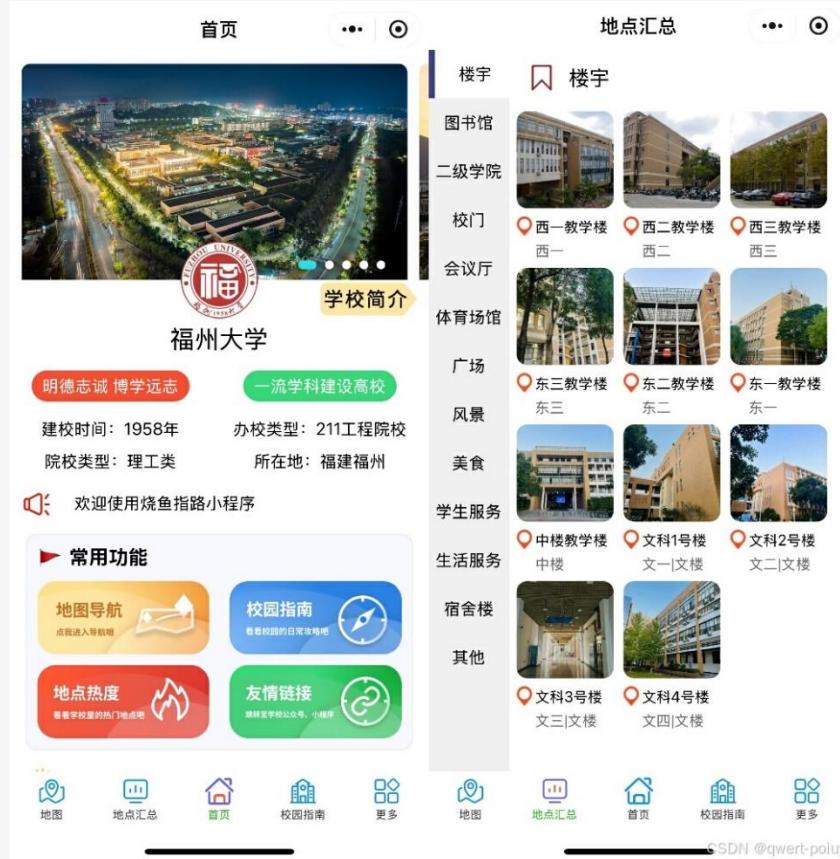
```
// 地图工具函数，包含常见bug修复
export const calculateMapBounds = (coordinates: number[][][]) => {
  if (!coordinates || coordinates.length === 0) {
    return null; // 修复空坐标输入导致的崩溃问题
  }

  let minX = Infinity;
  let minY = Infinity;
  let maxX = -Infinity;
  let maxY = -Infinity;

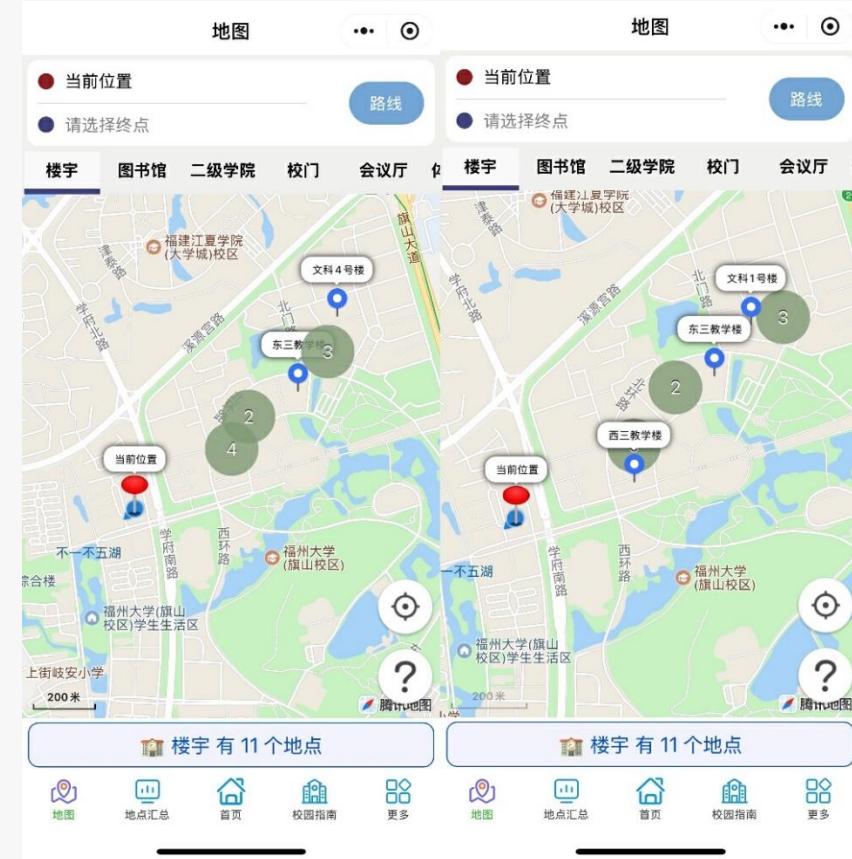
  coordinates.forEach(([x, y]) => {
    minX = Math.min(minX, x);
    minY = Math.min(minY, y);
    maxX = Math.max(maxX, x);
    maxY = Math.max(maxY, y);
  });

  return [minX, minY, maxX, maxY];
};
```

# 主页UI升级优化

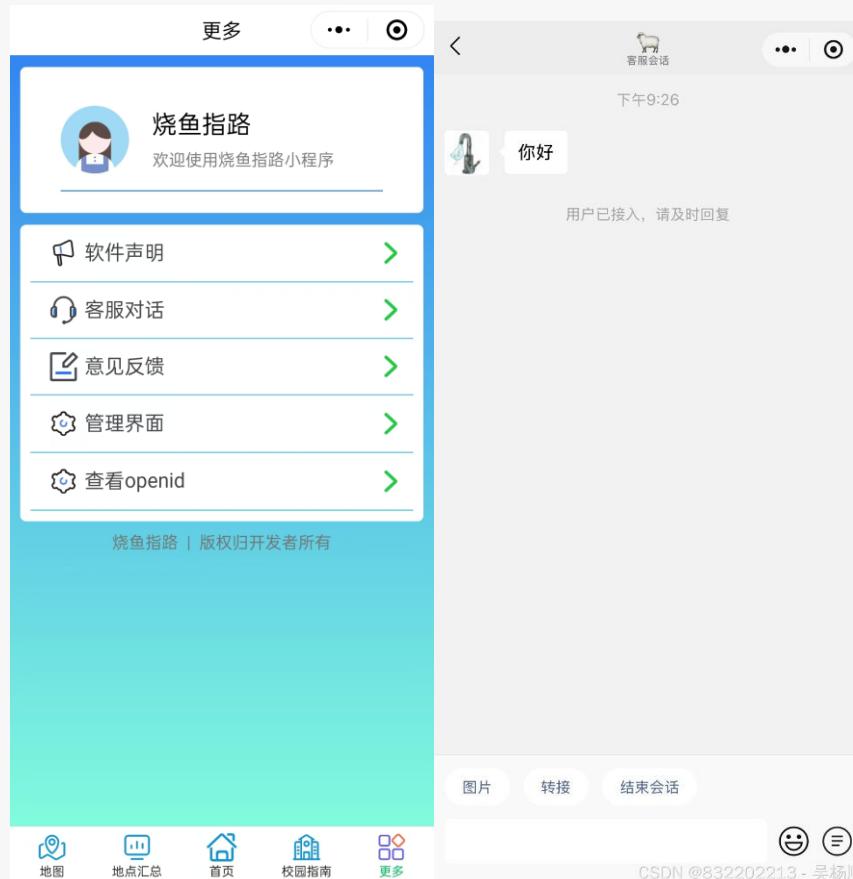


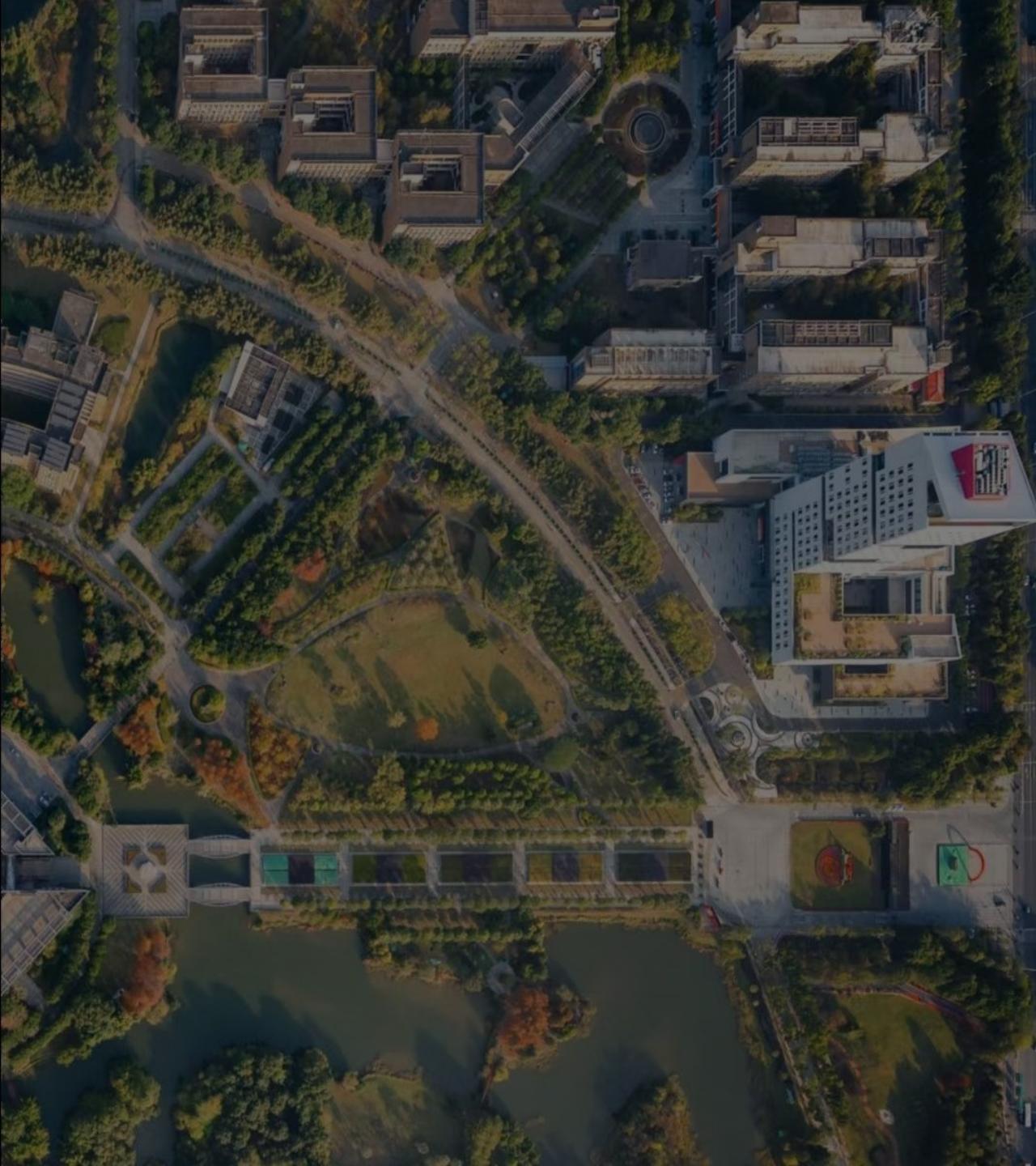
# 地图导航界面优化



# 用户反馈功能改进

添加用户反馈界面和存储机制。我们的存储机制将确保每一条反馈都能被跟踪和分析，以便我们的团队能够及时做出回应和采取行动。





## 第二部分 完成情况

项目已顺利完成，所有预定模块和功能均按计划实现。校园导航小程序，包括主页设计、地图导航、位置汇总和校园指南等核心功能已顺利完成。此外，后端数据库与API接口的构建也已完成，并进行了全面的测试，确保系统的稳定性与准确性。

# 项目界面展示



# 小程序主界面



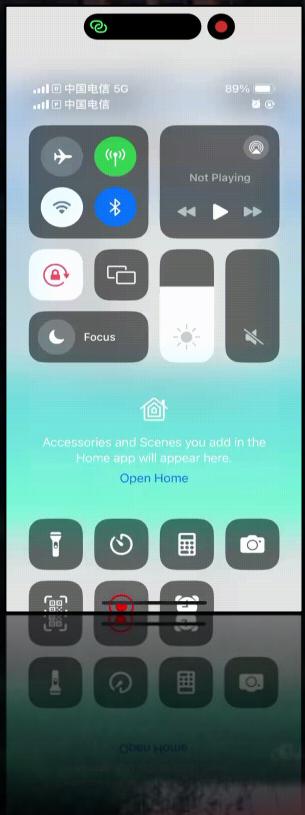
# 地图导航界面



# 校园地点展示



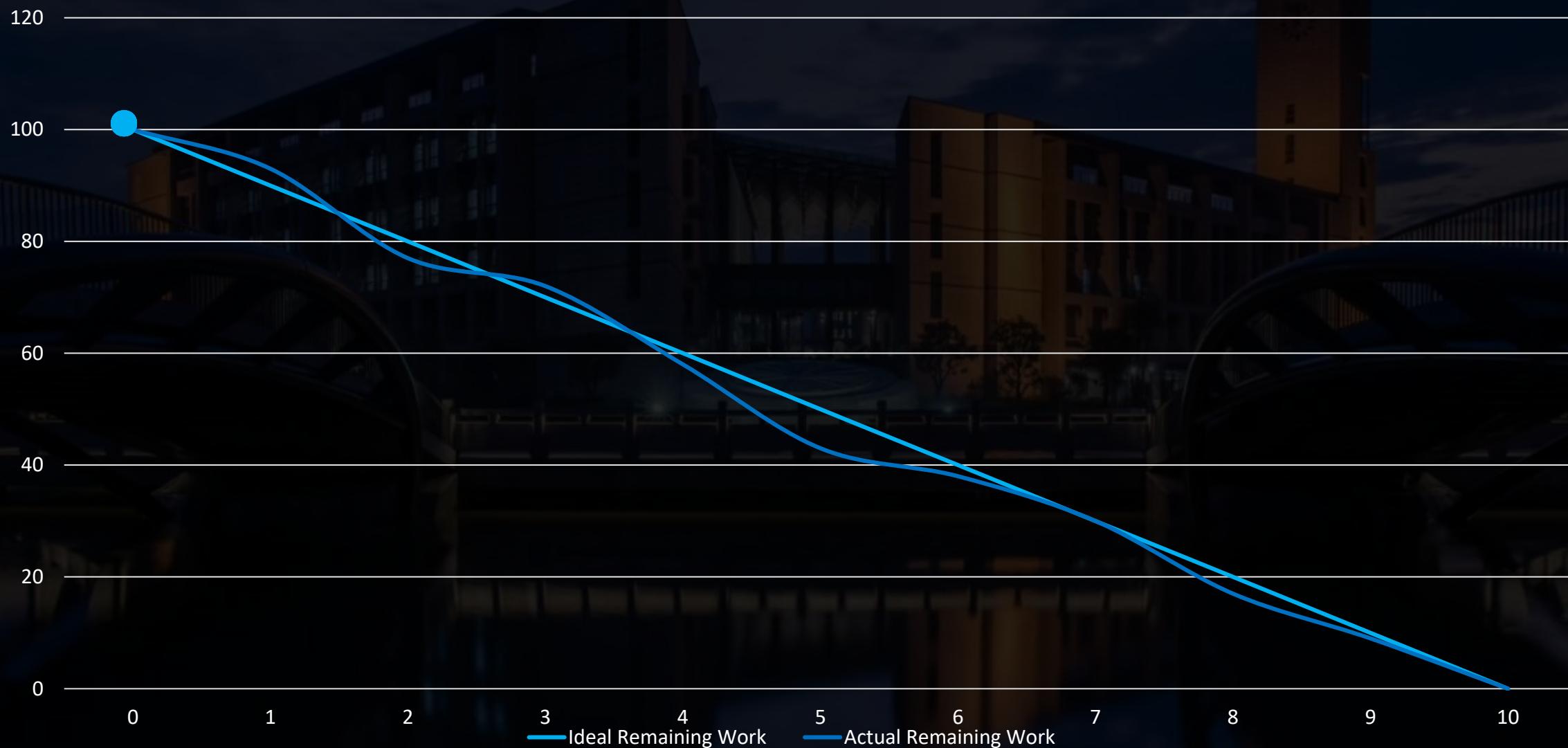
# 校园指南展示



# 微信客服功能



# 燃尽图





## 第三部分 测试情况

最终测试结果显示，小程序在启动性能、运行时性能、兼容性和网络性能方面均表现优秀，为正式上线做好了准备。

# 测试情况

云测

首页 / 小程序质检 / 质检任务报告 烧鱼指路 分享报告

## 烧鱼指路 小程序质检报告

报告总览 启动性能 运行性能 兼容性 网络性能 自动化详情

小程序类型: 线上版本 小程序版本: 1.0.0 提测人: 懒大王 创建时间: 2024-12-06 00:36:05  
测试计划: Monkey 智能化Monkey 任务耗时: 31分55秒 扣除时长: 0秒  
设备完成率: 100% ① 页面覆盖率: 64.71% ②

### 综合评估

评估结果: 优秀

启动性能 > 运行性能 > 兼容性 > 网络性能 >

启动性能: 优秀 运行性能: 良好 兼容性: 优秀 网络性能: 优秀

### 优化建议

启动性能: 未开启初始渲染缓存  
运行性能: setData传入未绑定在WXML的变量, 短时间太多图片请求  
兼容性: 暂无  
网络性能: 存在耗时过长的请求

### 启动性能

冷启动 (需下载代码包)		已下载代码包	
总耗时	722 ms	总耗时	414 ms
代码包下载	104 ms	代码注入	22 ms
代码注入	23 ms	首页初次渲染	19 ms
首页初次渲染	21 ms		

MiniTest Designed by 微信测试中心 v3.3.1

15°C 多云 搜索

CSDN @882202208刘叶妮 1:54 2024/12/6

# 改进建议

## 问题

未开启初始渲染缓存

setData传入未绑定在WXML的变量

短时间太多图片请求

存在耗时过长的请求

## 解决方案

在应用启动时，考虑启用初始渲染缓存功能，以加快页面加载速度，提高用户体验。通过缓存常用数据和资源，可以减少首次加载时的延迟。

对于通过 setData 方法传入的数据，确保所有传入的变量都已在 WXML 中进行正确绑定。避免使用未绑定的变量，以避免潜在的运行时错误和数据展示问题。

尽量减少短时间内的多次图片请求，考虑使用懒加载技术，按需加载图片。可以通过合理的图片尺寸和格式优化加载速度，或考虑将多个请求合并为一个请求，降低网络负担。

对请求时长过长的接口进行性能分析，找出瓶颈并优化。可以考虑使用数据缓存策略，异步加载非关键请求，或者优化后端服务以减少延迟。



## 第四部分

### 未来规划

未来，我们计划持续收集用户反馈，优化功能并定期更新产品。同时，扩大市场推广，吸引更多用户，并加强团队技术培训，以提升开发水平和应对新挑战。

## 功能扩展:

开发用户反馈系统和多语言支持，增加智能推荐模块。

## 用户社区建设:

创建用户社区平台，推动用户交流。

## 测试与优化:

完成新功能的全面测试，确保系统稳定。

## 性能优化:

提升数据库查询效率和前端加载性能。

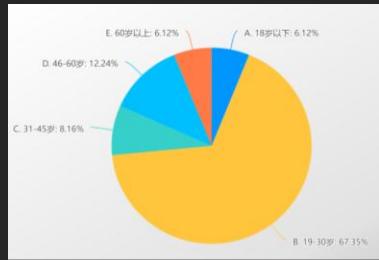
## 开发新功能:

实现用户反馈系统和多语言切换功能。

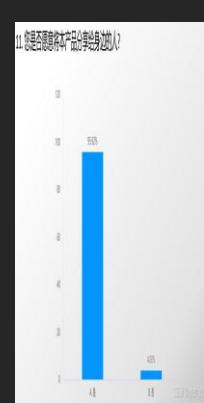
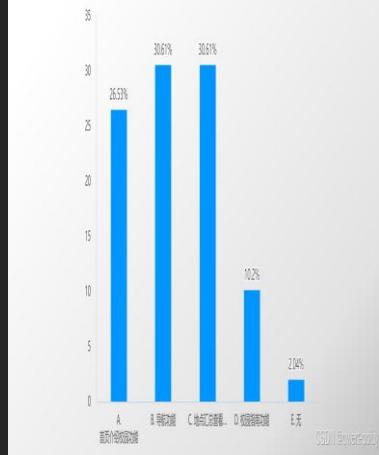
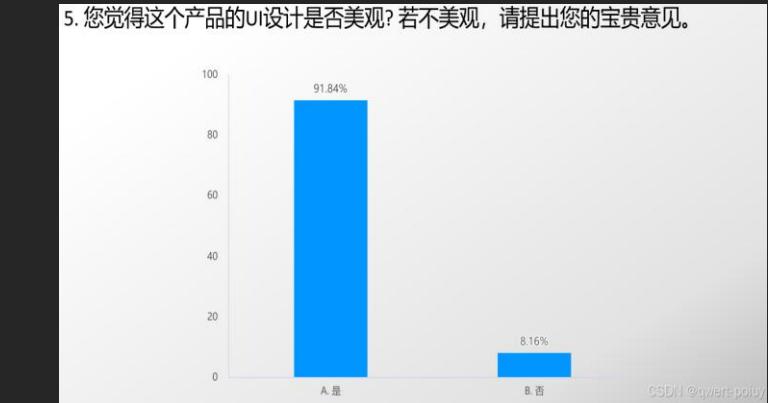
## 文档更新:

更新功能和测试文档，为新版本发布做准备。

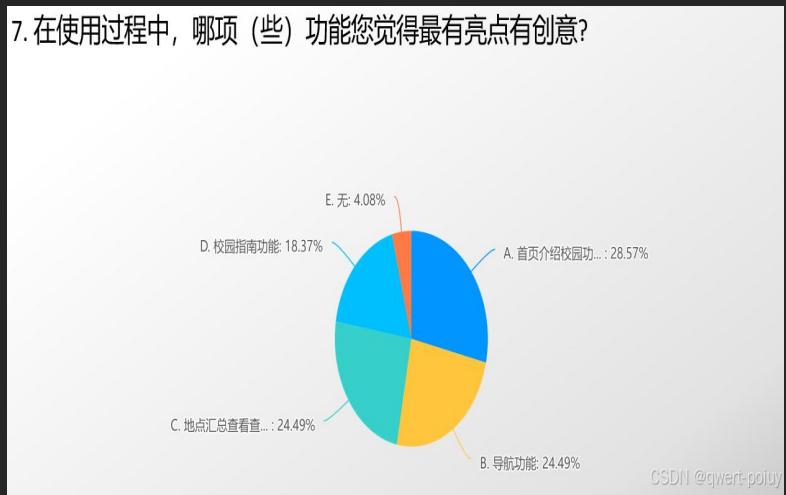
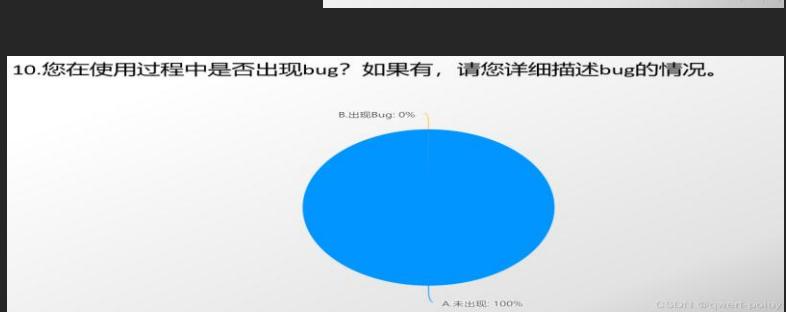
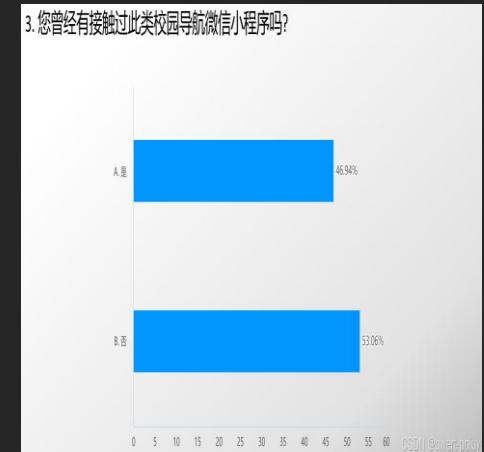
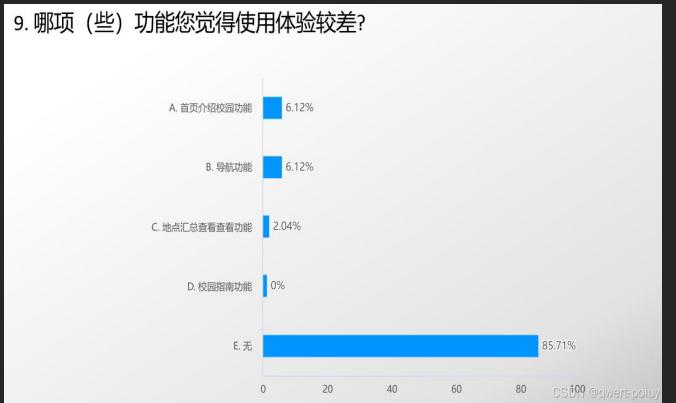
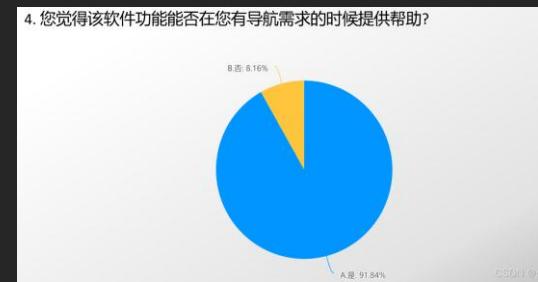
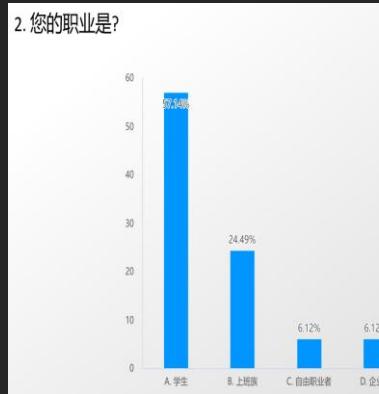
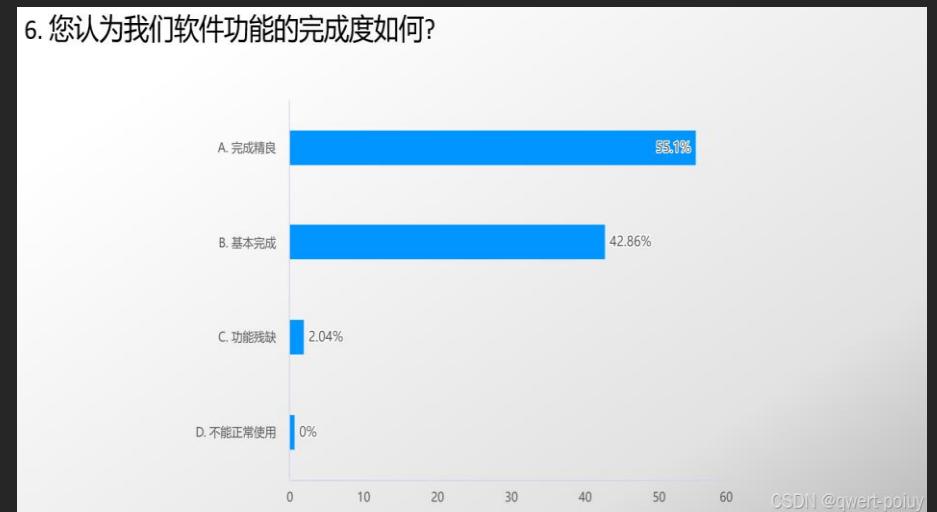
# 用户反馈

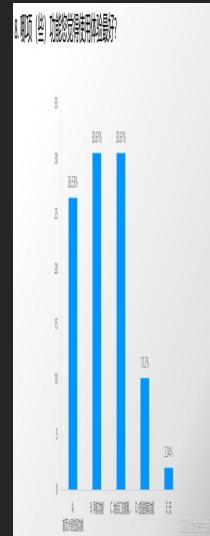
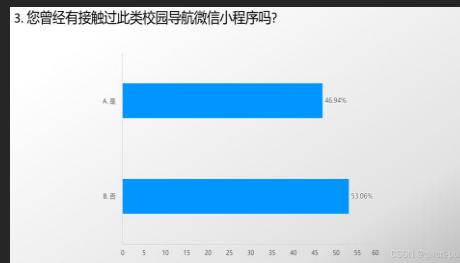
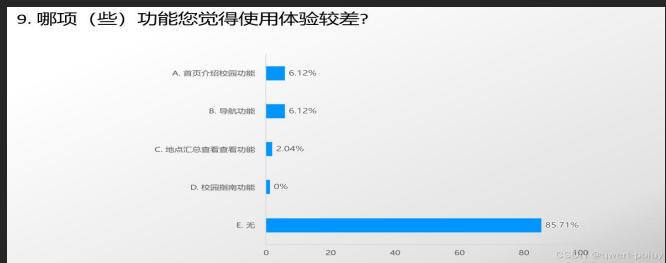
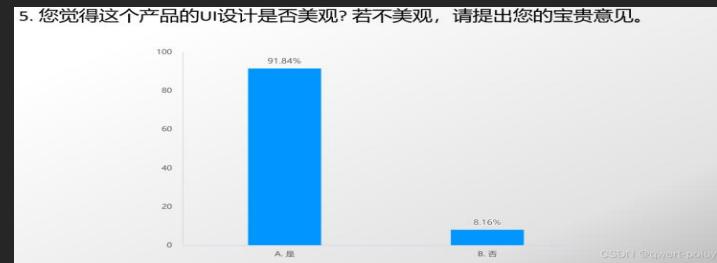
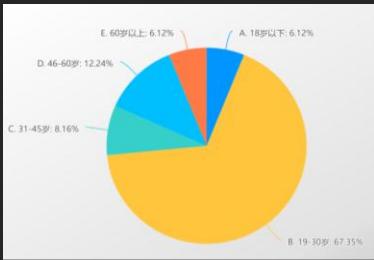


8.哪项(些)功能您觉得使用体验最好?

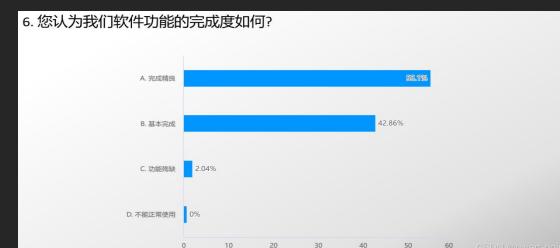
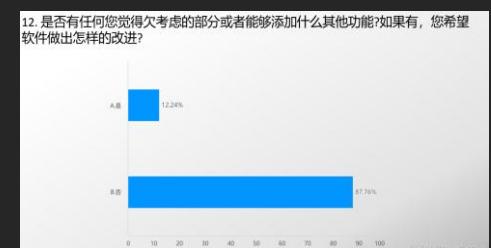
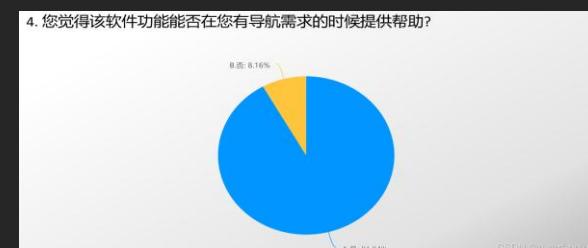
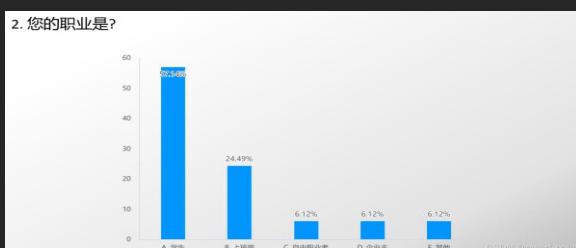
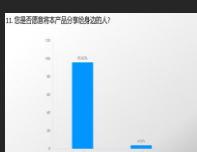
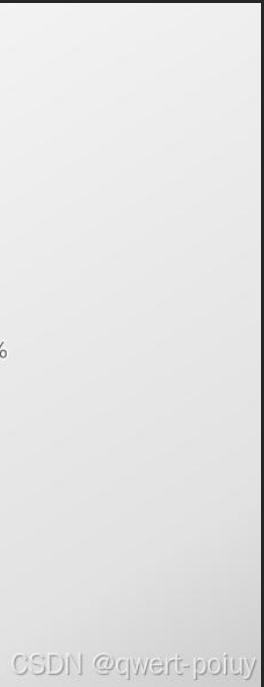
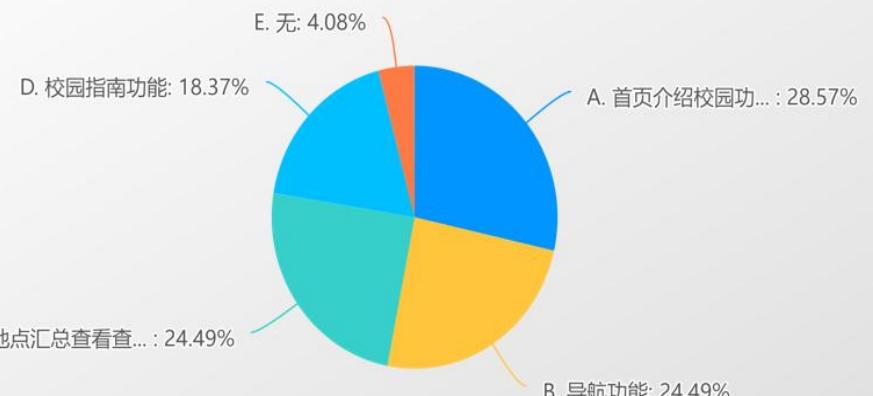


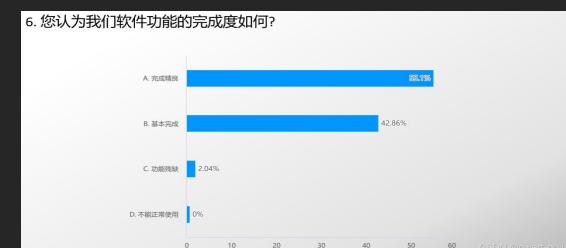
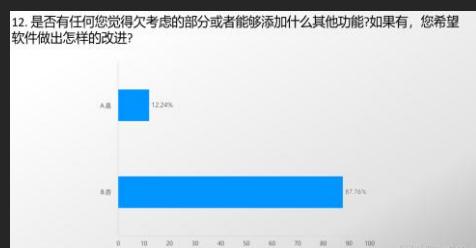
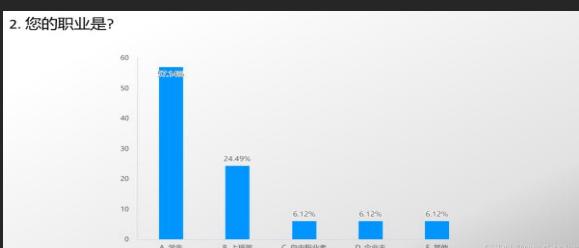
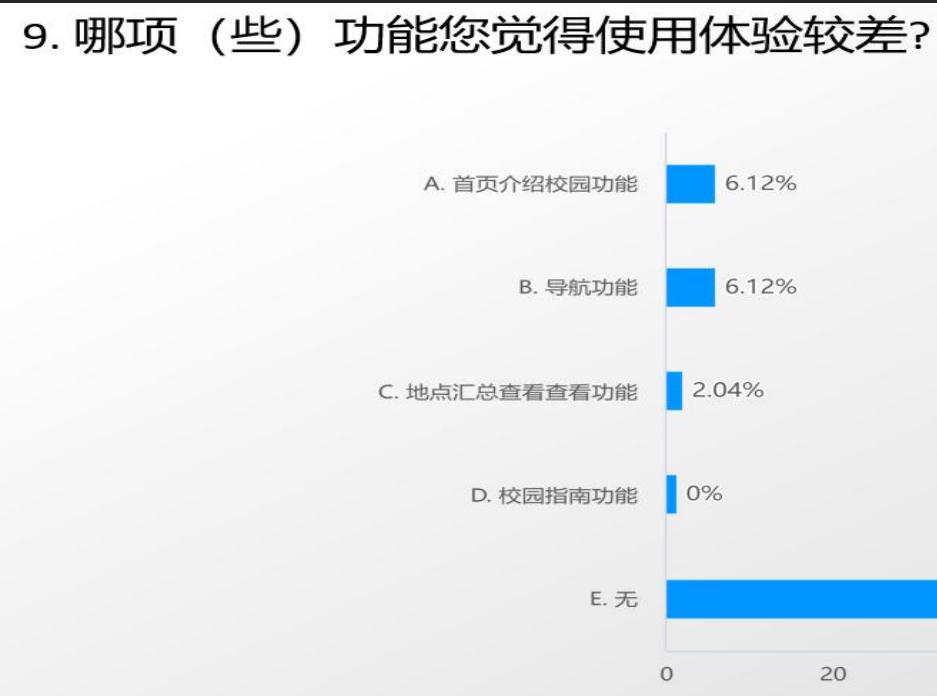
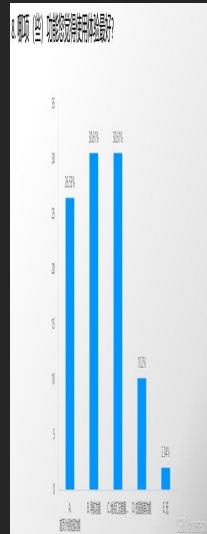
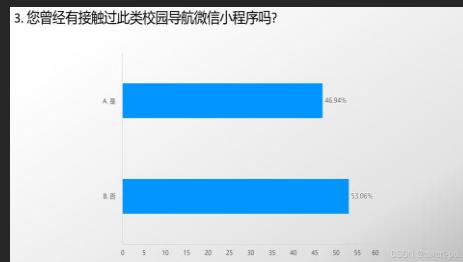
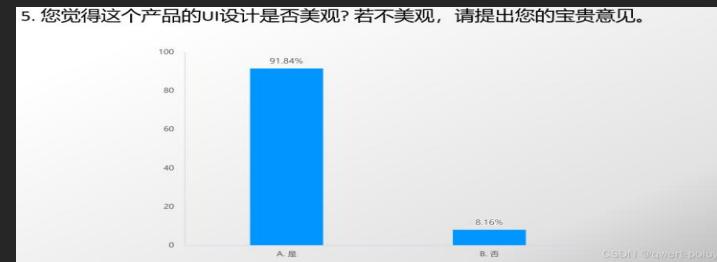
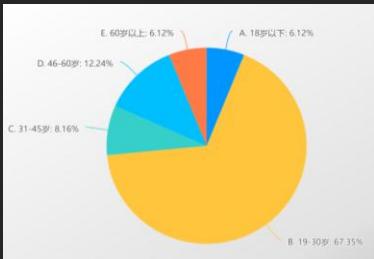
6. 您认为我们软件功能的完成度如何?





## 7. 在使用过程中,哪项(些)功能您觉得最有亮点有创意?



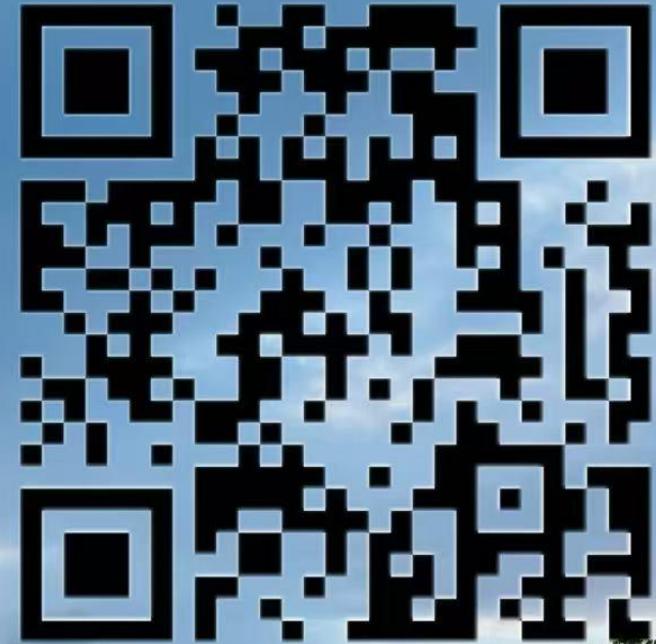


# 第一步



FZUGO福大导航小程序

# 第二步



用户使用情况调查问卷





THANKS TO SEE

致力于更好的校园导航