**https://github.com/AAA123GOOD/ABR.git**

**1. 整体设计理念**

本网页以"我的大学生活"为主题，采用单页应用形式，通过导航栏切换不同内容区块。

主体页面设计思路：以蓝色系主色调体现信息安全专业特色，

多样化布局展示：每个板块采用不同CSS布局技术，展示出不同的效果

**2. 页面结构与数量**

共设计5个页面，通过导航切换显示：

区块ID 标题 布局技术 核心内容

campus 校园风采 网格布局 校园建筑图文介绍

major 专业特色 弹性盒子布局 专业优势、课程表

life 课余生活 浮动布局 吃喝玩乐志愿活动

wuhan 武汉风光 定位布局 武汉景点图文展示+联系表单

1. **各页面详细设计说明**

**页眉样式分析 (site-header)**

css

页眉整体样式

.site-header {

background: linear-gradient(135deg, #1a5f9c, #0d3b66); 135度蓝渐变背景

color: white; 白色文字

padding: 32px 0; 上下32px内边距

text-align: center; 内容居中

}

页眉内容容器

.header-content {

max-width: 1200px; 最大宽度限制

margin: 0 auto; 水平居中

}

主标题样式

.site-header h1 {

font-size: 35.2px; 字体大小35.2px

margin-bottom: 8px; 底部8px外边距

}

}

**页脚设计 (site-footer)**

css

.site-footer {

background-color: #2c3e50; 设置深蓝色背景，与导航栏保持一致

color: white; 白色文字确保高对比度可读性

text-align: center; 内容居中对齐

padding: 24px; 24像素内边距，保证内容呼吸空间

margin-top: 32px; 顶部32像素外边距，与主体内容分离

**}**

**3.1首页 (index.html)**

布局特点：网格布局(Grid)

核心内容：网站概览+四大板块入口

HTML结构：

<div class="grid-container">

<div class="grid-item">...</div> x3

</div>

CSS技术：

**整体架构设计**

主内容区域容器样式

.main-content {

max-width: 1200px; 设置最大宽度为1200px，保证内容不会过宽

margin: 32px auto; 上下32px外边距，左右自动居中

padding: 0 16px; 左右16px内边距，防止移动端内容贴边

}

内容区块基础样式

.page-section {

margin-bottom: 48px; 区块底部48px外边距，创建内容分隔

padding: 24px; 24px内边距保证内容呼吸空间

background-color: white; 白色背景与页面底色形成对比

}

二级标题样式

h2 {

color: #1a5f9c; 使用品牌蓝色

margin-bottom: 24px; 底部24px外边距

padding-bottom: 8px; 底部8px内边距

border-bottom: 2px solid #1a5f9c; 蓝色下划线作为装饰

}

三级标题样式

h3 {

color: #2c3e50; 深灰色文字

margin: 16px 0 8px; 上16px，下8px外边距

}

段落文本样式

p {

color: #555; 中灰色文字，提升可读性

margin-bottom: 16px; 段落间16px间距

}

图片基础样式

img {

max-width: 100%; 图片最大宽度不超过容器

height: auto; 高度自适应，保持比例

display: block; 消除图片底部间隙

}

**导航栏技术实现**

导航栏容器

.main-nav {

background-color: #2c3e50; 深蓝背景

position: sticky; 粘性定位

top: 0; 顶部固定

z-index: 100; 确保在最上层

}

导航菜单列表

.main-nav ul {

display: flex; 弹性布局

justify-content: center; 水平居中

list-style: none; 移除列表样式

max-width: 1200px; 最大宽度限制

margin: 0 auto; 水平居中

padding: 0; 移除默认内边距

}

导航菜单项

.main-nav li {

padding: 16px 24px; 内边距16px/24px

}

导航链接

.main-nav a {

color: white; 白色文字

text-decoration: none; 移除下划线

font-weight: bold; 加粗字体

transition: all 0.3s; 过渡动画0.3秒

padding: 8px 16px; 内边距8px/16px

}

导航链接交互状态

.main-nav a:hover,

.main-nav a.current {

color: #4fc3f7; 悬停/当前项亮蓝色

background-color: rgba(255,255,255,0.1); 半透明白背景

**3.2 校园风采 (campus)**

布局特点：网格布局，使用网格系统实现图片排列。

HTML结构：

<div class="grid-container">

<div class="grid-item">

<img><div class="grid-item-content">...</div>

</div> x6

</div>

**CSS技术：**

网格布局容器 - 定义Grid布局系统

.grid-container {

display: grid;

自动适配列数，每列最小宽度 300px，最大占满剩余空间（1fr 表示等份分配可用空间 ）

让布局在不同设备宽度下，自动调整列的数量，实现响应式

grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));

网格项（子元素）之间的间距，水平和垂直方向间距都是 24px

gap: 24px;

}

.grid-item 是网格布局里的每一个子项，作为弹性容器，让内部元素垂直排列

.grid-item {

border: 1px solid #ddd; 给网格项加浅灰色边框，区分不同项

overflow: hidden; 溢出内容隐藏，避免内部元素（如图片）超出容器破坏布局

display: flex; 开启弹性布局

flex-direction: column; 弹性元素垂直排列（从上到下）

}

网格项内部图片的样式

.grid-item img {

width: 100%; 图片宽度占满父容器（.grid-item）

height: 250px; 初始设置图片高度，不过实际会结合 object-fit 适配

min-height: 150px; 最小高度，防止屏幕过窄时图片被压得太扁

object-fit: cover; 保持图片比例，裁剪填充容器，优先显示中间内容

border-bottom: 1px solid #eee; 图片底部加浅灰色边框，与下方内容区分隔

}

网格项内部内容区域（文字等）的样式

.grid-item-content {

padding: 16px; 内边距，让文字等内容和容器边缘有间隔，更美观

flex-grow: 1; 让内容区域自动填充 .grid-item 剩余的垂直空间，保证每个网格项高度一致

}展示出的内容组织：

主标题+导语段落

6个校园地点卡片（图片+文字描述）

**3.3 专业特色 (major)**

布局特点：弹性盒子布局

内容组成：

1.3个等宽图文卡片（专业特色介绍）

2.课程表格

HTML结构：

<div class="flex-container">...</div>

<table class="course-table">...</table>

CSS技术：

弹性盒子容器 - 定义Flex布局容器

..flex-container {

display: flex; 开启弹性布局

flex-direction: column; 弹性元素垂直排列（从上到下）

gap: 20px; 弹性子元素之间的间距，垂直方向间距 20px

}

.flex-item 是弹性布局里的子项，内部再用弹性布局实现图片和文字并排

.flex-item {

display: flex; 开启弹性布局，让内部图片和文字水平排列

}

弹性布局子项内部图片的样式

.flex-item img {

width: 55%; 图片宽度占 .flex-item 容器的 55%

height: auto; 高度自动，保持图片原有宽高比

margin-right: 15px; 图片右侧外边距，和文字保持间隔

object-fit: contain; 保持图片比例，完整显示在容器内，不裁剪，可能会有空白

}

弹性布局子项内部文字内容区域的样式

.flex-item-content {

flex: 1; 让文字区域占满 .flex-item 剩余的水平空间

min-width: 40%; 最小宽度，防止屏幕过窄时文字被挤没，保证可读性

} 表格标题样式

.course-table caption {

font-weight: bold; 标题文字加粗

margin-bottom: 16px; 标题下方16像素外边距

text-align: left; 标题文字左对齐

}

表头和单元格共用样式

.course-table th,

.course-table td {

padding: 12px 15px; 单元格内边距：上下12px，左右15px

text-align: left; 内容左对齐

border-bottom: 1px solid #ddd; 底部1像素浅灰色边框

}

表头特殊样式

.course-table th {

background-color: #1a5f9c; 深蓝色背景

color: white; 白色文字

}

表格斑马纹效果 - 偶数行背景色

.course-table tr:nth-child(even) {

background-color: #f9f9f9; 浅灰色背景

}

**3.4 课余生活 (life)**

布局特点：浮动布局

结构特点：

每行2个浮动元素

使用clearfix清除浮动

核心内容：学生活动展示

HTML结构：

<div class="float-container">

<div class="float-item">...</div> x4

<div class="clearfix"></div>

</div>

CSS技术：

.浮动布局容器 - 设置底部外边距24像素，用于与后续内容保持间距

.float-container {

margin-bottom: 24px;

}

浮动项基础样式 -

宽度占容器48%，左浮动，

右侧留4%外边距，底部24像素间距，

1像素浅灰色边框，隐藏溢出内容

.float-item {

width: 48%;

float: left;

margin-right: 4%;

margin-bottom: 24px;

border: 1px solid #ddd;

overflow: hidden;

}

每行第二个浮动项 - 移除右侧外边距，确保两列布局正确换行

.float-item:nth-child(2n) {

margin-right: 0;

}

浮动项内图片 -

宽度100%填满容器，固定高度250像素，

cover模式保持图片比例裁剪

.float-item img {

width: 100%;

height:250px;

object-fit: cover;

}

浮动项内容区 - 设置16像素内边距保证内容间距

.float-item-content {

padding: 16px;

}

清除浮动 -

通过伪元素在容器末尾添加空内容，

设置为表格显示并清除两侧浮动，

解决浮动元素导致的高度塌陷问题

.clearfix::after {

content: "";

display: table;

clear: both;

}

**3.5 武汉风光 (wuhan)**

布局特点：定位布局

核心内容：本地景点+联系表单

HTML结构：

<div class="position-container">

<div class="position-item item1">...</div>

<div class="position-item item2">...</div>

</div>

定位布局

css

定位容器

.position-container {

position: relative;

height: 500px;

margin-bottom: 24px;

}

定位项基础样式

.position-item {

position: absolute;

width: 300px;

border: 1px solid #ddd;

overflow: hidden;

}

定位项图片

.position-item img {

width: 100%;

height: 200px;

object-fit: cover;

}

定位项内容区

.position-item-content {

padding: 8px 16px;

}

左上角定位

.item1 {

top: 0;

left: 0;

}

水平居中定位

.item2 {

top: 100px;

left: 50%;

transform: translateX(-50%);

}

右上角定位

.item3 {

top: 0;

right: 0;

}

**联系表单样式 (contact-form)**

css

联系表单容器样式

.contact-form {

background-color: white; 白色背景

padding: 32px; 四周32px内边距

}

表单组样式 - 包裹每个输入字段

.form-group {

margin-bottom: 24px; 底部24px外边距分隔表单项

}

表单标签样式

.form-group label {

display: block; 块级显示，独占一行

margin-bottom: 8px; 标签与输入框间距8px

font-weight: bold; 加粗字体

}

输入框和文本域共用样式

.form-group input,

.form-group textarea {

width: 100%; 宽度100%填满容器

padding: 12.8px; 12.8px内边距(0.8rem)

border: 1px solid black; 1px黑色边框

font-size: 16px; 16px字体大小(1rem)

}

文本域特殊样式

.form-group textarea {

resize: vertical; 只允许垂直调整大小

min-height: 120px; 最小高度120px

}

提交按钮基础样式

button[type="submit"] {

background-color: #1a5f9c; 品牌蓝色背景

color: white; 白色文字

padding: 12.8px 24px; 内边距：上下12.8px，左右24px

border: none; 无边框

cursor: pointer; 鼠标指针变为手形

font-size: 16px; 16px字体大小

}

提交按钮悬停状态

button[type="submit"]:hover {

background-color: #0d3b66; 深蓝色背景

}