

《JavaScript与jQuery开发技术》

门户网站的前端设计与开发

专 业： 计算机科学与技术

班 级： 计科2403

学 号 240601030075

姓 名： 黄展宽

完成日期： 2025.4.6

## 一、必做题

选取其中六题实现，实现步骤写在第二章内容中。

上机题目：

(1)福利彩票

需求说明：双色球投注区分为红色球号码区和蓝色球号码区，红色球号码区由1-33共三十三个号码组成，没有重复数字，蓝色球号码区由1-16共十六个号码组成。请根据需求实现双色球.html页面的JS逻辑。优化点：增加抽奖记录功能

(2)购物清单

需求说明：购物清单可根据输入产品信息和价格进行商品购买标记，生成的购买记录也可以进行对应的删除。请根据需求实现购物清单.html页面的JS逻辑。

(3)页面简易计算器

需求说明：开发基于web页面的简易计算器。功能不限，最少要包括加减乘除。

(4)导航功能

需求说明：鼠标放在导航上，可显示下一级目录；单击导航后，导航的背景色与其他导航不一致。请根据需求实现 导航.html页面的JS逻辑。

1. 页面注册

需求说明：①页面当中省份和城市需要一一对应，例如选择江苏省，那么应该在城市栏显示江苏省对应的城市；②点击注册按钮，需要对用户填写的信息进行检验：姓名不为空，密码长度不少于6位，邮箱地址要正确，否则提示信息错误。请按需求实现 注册.html页面的JS逻辑。

1. 轮播图

需求说明：使用任意jquery插件实现轮播图

(7)评论置顶

需求说明：完成论坛评论发布页面，要求输入用户名称及评论，发布后的评论有时间信息且评论可删除。如图1所示。

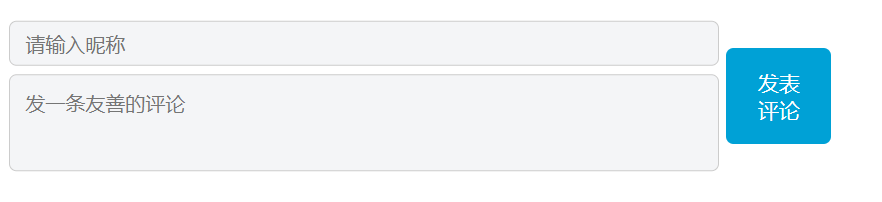


图1 评论发布页面

(8)图片切换

要求说明：在页面中有四幅小图片和一个图片展示区域，当鼠标移入某幅小图片时，在图片展示区域可看到其对应的大图。为凸显当前的小图状态，把其它三幅小图透明度设为0.5，如图2所示。



图2 图片切换页面

## 二、必做题实现步骤

2.1 (1)福利彩票

需求说明：双色球投注区分为红色球号码区和蓝色球号码区，红色球号码区由1-33共三十三个号码组成，没有重复数字，蓝色球号码区由1-16共十六个号码组成。请根据需求实现双色球.html页面的JS逻辑。优化点：增加抽奖记录功能

#### 2.1.1 JS技术说明

1. ****函数定义****：
   1. randomInt(min, max)：生成一个在指定范围内的随机整数，包括最小值和最大值。它使用 Math.random() 生成一个 0 到 1 之间的随机小数，然后通过公式转换为指定范围内的整数。
2. ****获取 DOM 元素****：
   1. const numbersDiv = document.getElementById('numbers');：获取页面中 ID 为 numbers 的元素，该元素包含用于显示号码的 HTML 元素。
   2. const generateButton = document.getElementById('generateButton');：获取页面中 ID 为 generateButton 的按钮元素，用于生成号码。
   3. const drawButton = document.getElementById('drawButton');：获取页面中 ID 为 drawButton 的按钮元素，用于模拟开奖。
   4. const recordContainer = document.getElementById('recordContainer');：获取页面中 ID 为 recordContainer 的元素，用于显示抽奖记录。
   5. const winningResult = document.getElementById('winningResult');：获取页面中 ID 为 winningResult 的元素，用于显示开奖结果。
3. ****生成号码函数****：
   1. generateLotteryNumbers()：该函数用于生成双色球的号码。
      1. 首先生成 6 个不重复的红球号码，范围是 1 到 33，并对这些号码进行排序。
      2. 然后生成 1 个蓝球号码，范围是 1 到 16。
      3. 接着将生成的号码显示在页面上对应的 HTML 元素中。
      4. 最后，将生成的号码记录添加到抽奖记录容器中。
4. ****开奖函数****：
   1. drawLottery()：该函数用于模拟开奖过程。
      1. 同样生成 6 个不重复的红球号码和 1 个蓝球号码。
      2. 拼接生成的号码字符串，用于显示开奖号码。
      3. 遍历抽奖记录，检查是否有与开奖号码相同的记录，如果有则表示有人中奖。
      4. 根据中奖情况在页面上显示相应的开奖结果信息。
5. ****事件监听器****：
   1. generateButton.addEventListener('click', generateLotteryNumbers);：为生成号码按钮添加点击事件监听器，当按钮被点击时，调用 generateLotteryNumbers 函数生成号码。
   2. drawButton.addEventListener('click', drawLottery);：为开奖按钮添加点击事件监听器，当按钮被点击时，调用 drawLottery 函数模拟开奖。
6. ****页面加载时执行****：
   1. window.onload = generateLotteryNumbers;：在页面加载完成时，自动调用 generateLotteryNumbers 函数生成一组初始的双色球号码并显示在页面上。

#### 2.1.2 关键代码

|  |
| --- |
| <!DOCTYPE html>  <html lang="zh-CN">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>双色球</title>  <style>  body {  font-family: Arial, sans-serif;  background: linear-gradient(135deg, #6a11cb, #85efa3);  margin: 0;  padding: 0;  display: flex;  justify-content: center;  align-items: center;  height: 100vh;  }  h1 {  color: #fff;  margin-bottom: 20px;  font-size: 2.5rem;  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);  }  .lottery-container {  background: rgba(255, 255, 255, 0.9);  padding: 30px;  border-radius: 15px;  box-shadow: 0 8px 20px rgba(0, 0, 0, 0.2);  text-align: center;  max-width: 500px;  width: 90%;  animation: fadeIn 1s ease-in-out;  height: 80vh;  overflow-y: auto;  }  .numbers {  display: flex;  justify-content: center;  margin: 30px 0;  flex-wrap: wrap;  gap: 10px;  }  .number {  width: 50px;  height: 50px;  line-height: 50px;  border-radius: 50%;  margin: 0 5px;  font-size: 20px;  font-weight: bold;  color: #fff;  text-align: center;  display: inline-block;  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.2);  transition: transform 0.2s ease, box-shadow 0.2s ease;  }  .number:hover {  transform: scale(1.1);  box-shadow: 0 6px 10px rgba(0, 0, 0, 0.3);  }  .red {  background-color: #e74c3c;  }  .blue {  background-color: #3498db;  }  button {  background-color: #2ecc71;  color: #fff;  border: none;  padding: 12px 24px;  font-size: 18px;  border-radius: 25px;  cursor: pointer;  transition: background-color 0.3s ease, transform 0.2s ease;  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.2);  margin: 5px;  }  button:hover {  background-color: #27ae60;  transform: translateY(-2px);  }  button:active {  transform: translateY(0);  }  .footer {  margin-top: 20px;  color: #777;  font-size: 14px;  }  .record-container {  margin-top: 20px;  border-top: 1px solid #ccc;  padding-top: 10px;  }  .record-item {  margin-bottom: 5px;  }  @keyframes fadeIn {  from {  opacity: 0;  transform: translateY(-20px);  }  to {  opacity: 1;  transform: translateY(0);  }  }  </style>  </head>  <body>  <div class="lottery-container">  <h1>双色球号码生成器</h1>  <div class="numbers" id="numbers">  <div class="number red"></div>  <div class="number red"></div>  <div class="number red"></div>  <div class="number red"></div>  <div class="number red"></div>  <div class="number red"></div>  <div class="number blue"></div>  </div>  <button id="generateButton">生成号码</button>  <button id="drawButton">开奖</button>  <div class="footer">  <p>试试你的运气吧！也许下一个大奖就是你的！</p>  </div>  <div class="record-container" id="recordContainer">  <h2>抽奖记录</h2>  </div>  <div id="winningResult"></div>  </div>  <script>  function randomInt(min, max) {  return Math.floor(Math.random() \* (max - min + 1)) + min;  }  const numbersDiv = document.getElementById('numbers');  const generateButton = document.getElementById('generateButton');  const drawButton = document.getElementById('drawButton');  const recordContainer = document.getElementById('recordContainer');  const winningResult = document.getElementById('winningResult');  function generateLotteryNumbers() {  const redNumbers = [];  while (redNumbers.length < 6) {  const num = randomInt(1, 33);  if (!redNumbers.includes(num)) {  redNumbers.push(num);  }  }  redNumbers.sort((a, b) => a - b);  const blueNumber = randomInt(1, 16);  const numberElements = numbersDiv.querySelectorAll('.number');  for (let i = 0; i < 6; i++) {  numberElements[i].textContent = redNumbers[i];  }  numberElements[6].textContent = blueNumber;  const recordItem = document.createElement('div');  recordItem.classList.add('record-item');  const redNumsStr = redNumbers.map(num => num.toString()).join(', ');  recordItem.textContent = `红球: ${redNumsStr}, 蓝球: ${blueNumber}`;  recordContainer.appendChild(recordItem);  }  function drawLottery() {  const redNumbers = [];  while (redNumbers.length < 6) {  const num = randomInt(1, 33);  if (!redNumbers.includes(num)) {  redNumbers.push(num);  }  }  redNumbers.sort((a, b) => a - b);  const blueNumber = randomInt(1, 16);  const redNumsStr = redNumbers.map(num => num.toString()).join(', ');  const winningNumbers = `红球: ${redNumsStr}, 蓝球: ${blueNumber}`;  const recordItems = recordContainer.querySelectorAll('.record-item');  let hasWinning = false;  recordItems.forEach(item => {  if (item.textContent === winningNumbers) {  hasWinning = true;  }  });  if (hasWinning) {  winningResult.textContent = `开奖号码: ${winningNumbers}，恭喜中奖！`;  } else {  winningResult.textContent = `开奖号码: ${winningNumbers}，很遗憾，无人中奖。`;  }  }  generateButton.addEventListener('click', generateLotteryNumbers);  drawButton.addEventListener('click', drawLottery);  window.onload = generateLotteryNumbers;  </script>  </body>  </html> |

#### 2.1.3 运行截图



2.2 页面简易计算器

需求说明：开发基于web页面的简易计算器。功能不限，最少要包括加减乘除。

#### 2.2.1 JS技术说明

### **全局变量与 DOM 元素获取**

* historyElement：借助 document.getElementById('history') 获取用于显示计算历史的 DOM 元素，后续用于存储和展示每次计算的记录。
* previousExpression：用于保存上一次的计算表达式，初始值为空字符串，在计算结果展示和历史记录添加时会用到。

### **函数功能说明**

#### **appendToDisplay(value)**

* ****功能****：向计算器显示框追加内容。
* ****参数****：value 为要追加的字符，如数字或运算符。
* ****实现方式****：通过 document.getElementById('display').value += value 将传入的 value 追加到显示框的当前内容之后。

#### **clearDisplay()**

* ****功能****：清空计算器显示框内容、计算历史记录以及上一次的计算表达式。
* ****实现方式****：
  + document.getElementById('display').value = ''：清空显示框的内容。
  + historyElement.innerHTML = ''：清空计算历史记录。
  + previousExpression = ''：重置上一次的计算表达式。

#### **calculate()**

* ****功能****：对显示框中的表达式进行计算，并处理结果显示和历史记录添加。
* ****实现方式****：
  + 获取显示框中的表达式 expression。
  + 使用 try...catch 结构进行错误处理：
    - try 块中：
      * 利用 eval(expression) 计算表达式的值。
      * 将计算结果显示在显示框中。
      * 若 previousExpression 不为空，创建一个 <p> 元素，将上一次的表达式和计算结果作为文本内容添加到该元素中，并将其添加到计算历史记录中。
      * 更新 previousExpression 为当前表达式。
    - catch 块中：若计算过程出现错误，将显示框内容设置为 '错误'。

### **事件绑定**

在 HTML 中通过 onclick 属性为按钮绑定了相应的 JavaScript 函数，当点击按钮时，会触发对应的函数执行相应操作，例如点击数字按钮调用 appendToDisplay 函数追加数字，点击 = 按钮调用 calculate 函数进行计算，点击 C 按钮调用 clearDisplay 函数清空内容。

#### 2.2.2 关键代码

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>简易计算器</title>  <style>  body {  display: flex;  justify-content: center;  align-items: center;  height: 100vh;  margin: 0;  background-color: #f4f4f9;  }  #calculator {  background-color: #fff;  border-radius: 8px;  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  padding: 20px;  }  #display {  width: 100%;  margin-bottom: 10px;  padding: 10px;  font-size: 24px;  text-align: right;  }  .buttons {  display: grid;  grid-template-columns: repeat(4, 1fr);  gap: 10px;  }  button {  padding: 10px;  font-size: 18px;  border: none;  background-color: #e0e0e0;  cursor: pointer;  border-radius: 4px;  }  button:hover {  background-color: #d0d0d0;  }  #history {  margin-top: 20px;  max-height: 200px;  overflow-y: auto;  }  </style>  </head>  <body>  <div id="calculator">  <input type="text" id="display" readonly>  <div class="buttons">  <button onclick="appendToDisplay('7')">7</button>  <button onclick="appendToDisplay('8')">8</button>  <button onclick="appendToDisplay('9')">9</button>  <button onclick="appendToDisplay('/')">/</button>  <button onclick="appendToDisplay('4')">4</button>  <button onclick="appendToDisplay('5')">5</button>  <button onclick="appendToDisplay('6')">6</button>  <button onclick="appendToDisplay('\*')">\*</button>  <button onclick="appendToDisplay('1')">1</button>  <button onclick="appendToDisplay('2')">2</button>  <button onclick="appendToDisplay('3')">3</button>  <button onclick="appendToDisplay('-')">-</button>  <button onclick="appendToDisplay('0')">0</button>  <button onclick="appendToDisplay('.')">.</button>  <button onclick="calculate()">=</button>  <button onclick="appendToDisplay('+')">+</button>  <button onclick="clearDisplay()">C</button>  </div>  <div id="history"></div>  </div>  <script>  const historyElement = document.getElementById('history');  let previousExpression = '';  function appendToDisplay(value) {  document.getElementById('display').value += value;  }  function clearDisplay() {  document.getElementById('display').value = '';  historyElement.innerHTML = '';  previousExpression = '';  }  function calculate() {  const expression = document.getElementById('display').value;  try {  const result = eval(expression);  document.getElementById('display').value = result;  if (previousExpression) {  const historyItem = document.createElement('p');  historyItem.textContent = `${previousExpression} = ${result}`;  historyElement.appendChild(historyItem);  }  previousExpression = expression;  } catch (error) {  document.getElementById('display').value = '错误';  }  }  </script>  </body>  </html> |

#### 2.2.3 运行截图520ea98af282ebb78cf8ac80c9ed391d

2.3购物清单

### 需求说明：购物清单可根据输入产品信息和价格进行商品购买标记，生成的购买记录也可以进行对应的删除。请根据需求实现购物清单.html页面的JS逻辑。

#### 2.3.1 JS技术说明

### **1. 获取 DOM 元素**

通过 document.getElementById 方法，分别获取了页面中 id 为 itemName（用于输入商品名称的输入框）、itemPrice（用于输入商品价格的输入框）、addItemButton（添加商品的按钮）和 shoppingTable（显示商品信息的表格）的 DOM 元素，并将它们分别赋值给对应的变量。这些变量后续用于操作相应的 DOM 元素。

### **2. 事件监听器绑定**

使用 addEventListener 方法为 addItemButton 按钮添加了一个 click 事件监听器。当按钮被点击时，会执行监听器中的回调函数，即后续的添加商品的操作逻辑。

### **3. 点击按钮后的操作逻辑**

使用 if 语句检查 itemName 和 itemPrice 是否都有值（即输入框不为空）。如果都有值，则执行 if 块内的代码，进行添加商品的操作。

### **4. 创建表格行和单元格**

创建一个 <td> 元素用于操作列，然后创建一个 <button> 元素，设置其文本为 '删除'，并为该按钮添加一个 click 事件监听器。当点击该按钮时，会调用回调函数 newRow.remove()，将包含该按钮的表格行从表格中移除。最后将按钮添加到操作单元格 actionCell 中。

### **5. 组装并添加表格行**

将前面创建的商品名称单元格 nameCell、商品价格单元格 priceCell 和操作单元格 actionCell 依次添加到新创建的表格行 newRow 中。然后将 newRow 添加到 shoppingTable 表格的 <tbody> 部分（通过 shoppingTable.tBodies[0] 获取 <tbody> 元素），从而在页面上显示新添加的商品信息行。

### **6. 清空输入框**

将 itemNameInput 和 itemPriceInput 输入框的值设置为空字符串，以便用户可以继续输入新的商品信息。

#### 2.3.2 关键代码

|  |
| --- |
| .input-group input {  width: calc(100% - 20px);  padding: 10px;  font-size: 16px;  border: 1px solid #ccc;  border-radius: 5px;  transition: border-color 0.3s ease, box-shadow 0.3s ease;  }  .input-group input:focus {  border-color: #6a11cb;  box-shadow: 0 0 8px rgba(106, 17, 203, 0.5);  outline: none;  }  .input-group button {  padding: 12px 24px;  font-size: 16px;  background-color: #28a745;  color: #fff;  border: none;  border-radius: 25px;  cursor: pointer;  transition: background-color 0.3s ease, transform 0.2s ease;  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.2);  }  .input-group button:hover {  background-color: #218838;  transform: translateY(-2px);  }  .input-group button:active {  transform: translateY(0);  }  table {  width: 100%;  border-collapse: collapse;  margin-top: 20px;  }  table th, table td {  padding: 12px;  text-align: left;  border-bottom: 1px solid #ddd;  }  table th {  background-color: #f8f9fa;  color: #555;  }  table td button {  padding: 8px 16px;  font-size: 14px;  background-color: #dc3545;  color: #fff;  border: none;  border-radius: 5px;  cursor: pointer;  transition: background-color 0.3s ease, transform 0.2s ease;  }  table td button:hover {  background-color: #c82333;  transform: translateY(-2px);  }  table td button:active {  transform: translateY(0);  }  @keyframes fadeIn {  from {  opacity: 0;  transform: translateY(-20px);  }  to {  opacity: 1;  transform: translateY(0);  }  }  .background {  position: absolute;  top: 0;  left: 0;  width: 100%;  height: 100%;  z-index: -1;  background: radial-gradient(circle, rgba(255, 255, 255, 0.1), rgba(255, 255, 255, 0.05));  animation: moveBackground 10s infinite alternate;  }  @keyframes moveBackground {  from {  transform: scale(1);  }  to {  transform: scale(1.2);  }  }  </style>  </head>  <body>  <div class="background"></div>  <div class="container">  <h1>🛒 超市购物表格 🛒</h1>  <div class="input-group">  <label for="itemName">商品名称</label>  <input type="text" id="itemName" placeholder="输入商品名称">  </div>  <div class="input-group">  <label for="itemPrice">商品价格</label>  <input type="number" id="itemPrice" placeholder="输入商品价格">  </div>  <div class="input-group">  <button id="addItemButton">添加商品</button>  </div>  <table id="shoppingTable">  <thead>  <tr>  <th>商品名称</th>  <th>商品价格</th>  <th>操作</th>  </tr>  </thead>  <tbody>  </tbody>  </table>  </div>  <script>  const itemNameInput = document.getElementById('itemName');  const itemPriceInput = document.getElementById('itemPrice');  const addItemButton = document.getElementById('addItemButton');  const shoppingTable = document.getElementById('shoppingTable');  addItemButton.addEventListener('click', () => {  const itemName = itemNameInput.value;  const itemPrice = itemPriceInput.value;  if (itemName && itemPrice) {  const newRow = document.createElement('tr');  const nameCell = document.createElement('td');  nameCell.textContent = itemName;  const priceCell = document.createElement('td');  priceCell.textContent = itemPrice;  const actionCell = document.createElement('td');  const deleteButton = document.createElement('button');  deleteButton.textContent = '删除';  deleteButton.addEventListener('click', () => {  newRow.remove();  });  actionCell.appendChild(deleteButton);  newRow.appendChild(nameCell);  newRow.appendChild(priceCell);  newRow.appendChild(actionCell);  shoppingTable.tBodies[0].appendChild(newRow);  itemNameInput.value = '';  itemPriceInput.value = '';  }  });  </script>  </body>  </html> |

#### 2.3.3 运行截图



2.4 页面注册

需求说明：①页面当中省份和城市需要一一对应，例如选择江苏省，那么应该在城市栏显示江苏省对应的城市；②点击注册按钮，需要对用户填写的信息进行检验：姓名不为空，密码长度不少于6位，邮箱地址要正确，否则提示信息错误。请按需求实现 注册.html页面的JS逻辑。

#### 2.4.1 JS技术说明

### **全局变量与初始化**

定义了一个全局变量 slideIndex 并初始化为 1，用于表示当前显示的幻灯片索引。

调用 showSlides 函数来显示初始幻灯片。

### **幻灯片切换功能**

plusSlides 函数：接收一个参数 n，用于控制幻灯片的切换方向和数量。它会更新 slideIndex 的值，并调用 showSlides 函数来显示相应的幻灯片。

currentSlide 函数：接收一个参数 n，将 slideIndex 设置为该值，然后调用 showSlides 函数来显示指定索引的幻灯片。

### **显示幻灯片功能**

showSlides 函数：

接收一个参数 n，表示要显示的幻灯片索引。

处理索引越界情况：如果 n 大于幻灯片数量，将 slideIndex 重置为 1；如果 n 小于 1，将 slideIndex 设置为幻灯片的最大数量。

隐藏所有幻灯片，并移除所有指示点的激活状态。

显示指定索引的幻灯片，并将对应的指示点设置为激活状态。

### **指示点创建功能**

createDots 函数：

获取所有幻灯片元素和指示点容器元素。

为每个幻灯片创建一个指示点元素，并为其添加点击事件监听器。点击指示点时，调用 currentSlide 函数显示相应的幻灯片。

将创建的指示点添加到指示点容器中，并将第一个指示点设置为激活状态。

### **自动切换功能**

使用 setInterval 函数，每 3000 毫秒（即 3 秒）调用一次 plusSlides(1) 函数，实现幻灯片的自动切换。

### **事件绑定**

在 HTML 中，为上一张和下一张按钮分别绑定了 plusSlides 函数，点击按钮时会触发相应的幻灯片切换操作。

在 createDots 函数中，为每个指示点绑定了点击事件监听器，点击指示点时会调用 currentSlide 函数切换到相应的幻灯片。

#### 2.4.2 关键代码

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>轮播图</title>  <script src="https://unpkg.com/@tailwindcss/browser@4"></script>  <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.7.2/css/all.min.css" rel="stylesheet">  <style>  .slider {  position: relative;  width: 100%;  max-width: 800px;  margin: 0 auto;  overflow: hidden;  }  .slides {  display: flex;  transition: transform 0.5s ease-in-out;  }  .slide {  min-width: 100%;  }  .slide img {  width: 100%;  height: auto;  }  .prev,  .next {  position: absolute;  top: 50%;  transform: translateY(-50%);  font-size: 24px;  background-color: rgba(0, 0, 0, 0.5);  color: white;  border: none;  cursor: pointer;  padding: 10px;  }  .prev {  left: 10px;  }  .next {  right: 10px;  }  .dots {  text-align: center;  margin-top: 10px;  }  .dot {  display: inline-block;  width: 10px;  height: 10px;  background-color: #bbb;  border-radius: 50%;  margin: 0 5px;  cursor: pointer;  }  .dot.active {  background-color: #717171;  }  </style>  </head>  <body>  <div class="slider">  <div class="slides" id="slides">  <div class="slide">  <img src="3A3D43331EFACE6505508E8CEFE22CD8.jpg" alt="Slide 1">  </div>  <div class="slide">  <img src="80DF8236DE00692E58FACBDAD89A5B36.jpg" alt="Slide 2">  </div>  <div class="slide">  <img src="EEBDF718866F552A835EF791E667660A.jpg" alt="Slide 3">  </div>  </div>  <button class="prev" onclick="plusSlides(-1)"><i class="fa-solid fa-angle-left"></i></button>  <button class="next" onclick="plusSlides(1)"><i class="fa-solid fa-angle-right"></i></button>  <div class="dots" id="dots"></div>  </div>  <script>  let slideIndex = 1;  showSlides(slideIndex);  function plusSlides(n) {  showSlides(slideIndex += n);  }  function currentSlide(n) {  showSlides(slideIndex = n);  }  function showSlides(n) {  let i;  let slides = document.getElementsByClassName("slide");  let dots = document.getElementsByClassName("dot");  if (n > slides.length) {  slideIndex = 1;  }  if (n < 1) {  slideIndex = slides.length;  }  for (i = 0; i < slides.length; i++) {  slides[i].style.display = "none";  }  for (i = 0; i < dots.length; i++) {  dots[i].className = dots[i].className.replace(" active", "");  }  slides[slideIndex - 1].style.display = "block";  dots[slideIndex - 1].className += " active";  }  function createDots() {  let slides = document.getElementsByClassName("slide");  let dotsContainer = document.getElementById("dots");  for (let i = 0; i < slides.length; i++) {  let dot = document.createElement("span");  dot.className = "dot";  dot.addEventListener("click", function () {  currentSlide(i + 1);  });  dotsContainer.appendChild(dot);  }  dotsContainer.children[0].classList.add("active");  }  createDots();  setInterval(function () {  plusSlides(1);  }, 3000);  </script>  </body>  </html> |

#### 2.4.3 运行截图fc8825e0b91faa93ad77c9eb25c429c9

2.5 页面注册

需求说明：①页面当中省份和城市需要一一对应，例如选择江苏省，那么应该在城市栏显示江苏省对应的城市；②点击注册按钮，需要对用户填写的信息进行检验：姓名不为空，密码长度不少于6位，邮箱地址要正确，否则提示信息错误。请按需求实现 注册.html页面的JS逻辑。

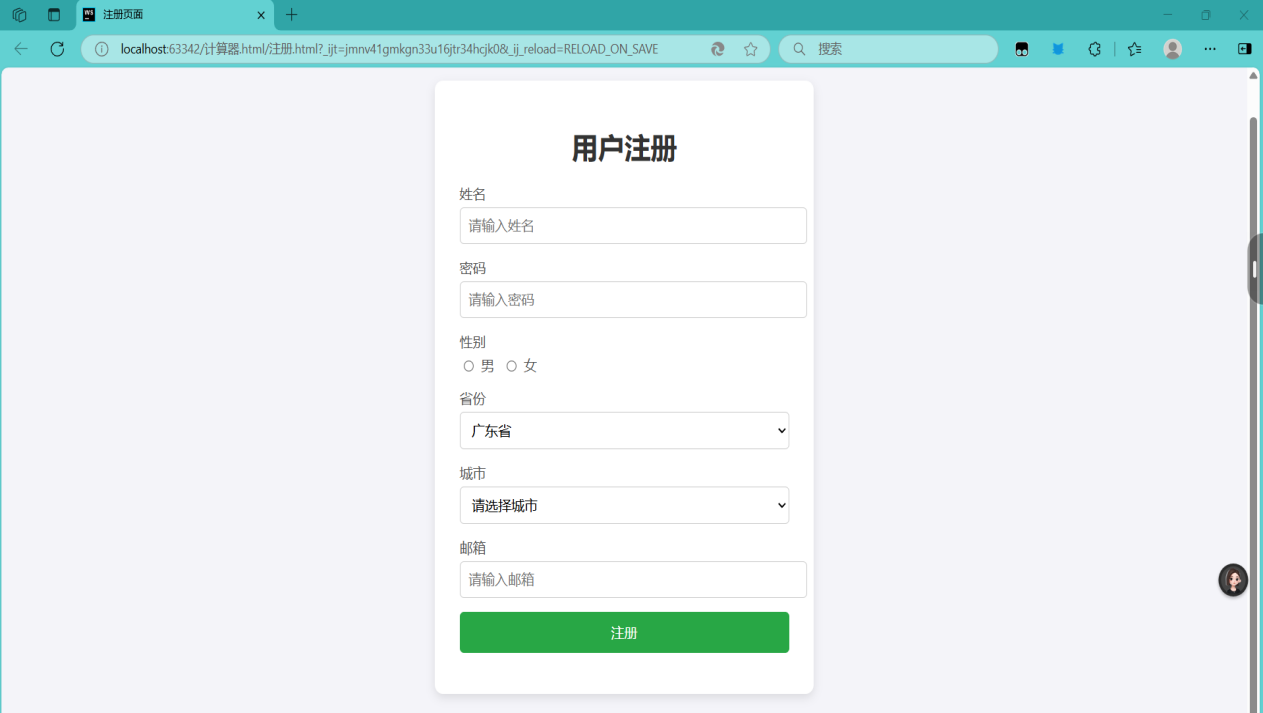
#### 2.5.1 JS技术说明

1. ****省份城市数据初始化****：定义对象provinceCities存储省份和城市对应关系。window.onload事件触发时，获取省份选择框元素，遍历provinceCities，为每个省份创建<option>元素并添加到选择框，实现页面加载时省份数据填充。
2. ****城市动态加载****：loadCities函数在省份选择框change事件触发时执行。获取选中省份，清空城市选择框，若有选中省份，则遍历对应城市数组，为每个城市创建<option>元素并添加到城市选择框，实现根据所选省份动态加载城市选项。
3. ****错误显示与隐藏****：showError函数通过获取指定id的错误提示元素，添加active类来显示错误信息；hideError函数则通过移除active类隐藏错误信息，控制错误提示的显示和隐藏。
4. ****表单验证****：validateForm函数对注册表单进行验证。获取各个表单字段的值，依次检查姓名、密码、性别、省份、城市和邮箱是否为空或符合格式要求。邮箱验证使用正则表达式/^[^\s@]+@[^\s@]+\.[^\s@]+$/，若不符合要求则调用showError显示错误信息，并将isValid设为false；全部字段验证通过则返回true。
5. ****表单提交****：submitForm函数在点击注册按钮时调用。先调用validateForm进行表单验证，验证通过则获取表单各字段值，在控制台打印数据并弹出注册成功提示框；验证不通过则不执行提交操作。

#### 2.5.2 关键代码

|  |
| --- |
| <!DOCTYPE html>  <html lang="zh-CN">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>注册页面</title>  <style>  body {  font-family: Arial, sans-serif;  background-color: #f4f4f9;  margin: 0;  padding: 20px;  display: flex;  justify-content: center;  align-items: center;  min-height: 100vh;  }  .container {  background-color: #fff;  padding: 30px;  border-radius: 10px;  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);  max-width: 400px;  width: 100%;  }  h1 {  color: #333;  margin-bottom: 20px;  text-align: center;  }  .form-group {  margin-bottom: 15px;  }  .form-group label {  display: block;  margin-bottom: 5px;  color: #555;  }  .form-group input,  .form-group select {  width: 100%;  padding: 10px;  font-size: 16px;  border: 1px solid #ccc;  border-radius: 5px;  }  .form-group input:focus,  .form-group select:focus {  border-color: #28a745;  outline: none;  }  .form-group .radio-group {  display: flex;  gap: 10px;  }  .form-group .radio-group label {  margin: 0;  display: flex;  align-items: center;  gap: 5px;  }  .form-group button {  width: 100%;  padding: 12px;  font-size: 16px;  background-color: #28a745;  color: #fff;  border: none;  border-radius: 5px;  cursor: pointer;  transition: background-color 0.3s ease;  }  .form-group button:hover {  background-color: #218838;  }  .error {  color: #dc3545;  font-size: 14px;  margin-top: 5px;  display: none;  }  .error.active {  display: block;  }  </style>  </head>  <body>  <div class="container">  <h1>用户注册</h1>  <form id="registerForm">  <div class="form-group">  <label for="name">姓名</label>  <input type="text" id="name" placeholder="请输入姓名">  <div id="nameError" class="error">姓名不能为空</div>  </div>  <div class="form-group">  <label for="password">密码</label>  <input type="password" id="password" placeholder="请输入密码">  <div id="passwordError" class="error">密码不能为空</div>  </div>  <div class="form-group">  <label>性别</label>  <div class="radio-group">  <label><input type="radio" name="gender" value="male"> 男</label>  <label><input type="radio" name="gender" value="female"> 女</label>  </div>  <div id="genderError" class="error">请选择性别</div>  </div>  <div class="form-group">  <label for="province">省份</label>  <select id="province" onchange="loadCities()">  <option value="">请选择省份</option>  </select>  <div id="provinceError" class="error">请选择省份</div>  </div>  <div class="form-group">  <label for="city">城市</label>  <select id="city">  <option value="">请选择城市</option>  </select>  <div id="cityError" class="error">请选择城市</div>  </div>  <div class="form-group">  <label for="email">邮箱</label>  <input type="email" id="email" placeholder="请输入邮箱">  <div id="emailError" class="error">请输入有效的邮箱地址</div>  </div>  <div class="form-group">  <button type="button" onclick="submitForm()">注册</button>  </div>  </form>  </div>  <script>  const provinceCities = {  "北京市": ["北京市"],  "上海市": ["上海市"],  "广东省": ["广州市", "深圳市", "珠海市"],  "浙江省": ["杭州市", "宁波市", "温州市"]  };  window.onload = function () {  const provinceSelect = document.getElementById('province');  for (const province in provinceCities) {  const option = document.createElement('option');  option.value = province;  option.textContent = province;  provinceSelect.appendChild(option);  }  };  function loadCities() {  const provinceSelect = document.getElementById('province');  const citySelect = document.getElementById('city');  const selectedProvince = provinceSelect.value;  citySelect.innerHTML = '<option value="">请选择城市</option>';  if (selectedProvince) {  const cities = provinceCities[selectedProvince];  for (const city of cities) {  const option = document.createElement('option');  option.value = city;  option.textContent = city;  citySelect.appendChild(option);  }  }  }  function showError(errorId) {  const errorElement = document.getElementById(errorId);  errorElement.classList.add('active');  }  function hideError(errorId) {  const errorElement = document.getElementById(errorId);  errorElement.classList.remove('active');  }  function validateForm() {  let isValid = true;  const name = document.getElementById('name').value;  const password = document.getElementById('password').value;  const gender = document.querySelector('input[name="gender"]:checked');  const province = document.getElementById('province').value;  const city = document.getElementById('city').value;  const email = document.getElementById('email').value;  if (!name) {  showError('nameError');  isValid = false;  } else {  hideError('nameError');  }  if (!password) {  showError('passwordError');  isValid = false;  } else {  hideError('passwordError');  }  if (!gender) {  showError('genderError');  isValid = false;  } else {  hideError('genderError');  }  if (!province) {  showError('provinceError');  isValid = false;  } else {  hideError('provinceError');  }  if (!city) {  showError('cityError');  isValid = false;  } else {  hideError('cityError');  }  if (!email || !/^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email)) {  showError('emailError');  isValid = false;  } else {  hideError('emailError');  }  return isValid;  }  function submitForm() {  if (validateForm()) {  const name = document.getElementById('name').value;  const password = document.getElementById('password').value;  const gender = document.querySelector('input[name="gender"]:checked').value;  const province = document.getElementById('province').value;  const city = document.getElementById('city').value;  const email = document.getElementById('email').value;  console.log({  name,  password,  gender,  province,  city,  email  });  alert('注册成功！');  }  }  </script>  </body>  </html> |

#### 2.5.3 运行截图



2.6 评论置顶

需求说明：完成论坛评论发布页面，要求输入用户名称及评论，发布后的评论有时间信息且评论可删除。如图1所示。

#### 2.6.1 JS技术说明

获取 DOM 元素：

通过 document.getElementById 方法分别获取了页面中 id 为 username（用户名称输入框）、comment（评论内容输入框）、submit-comment（发布评论按钮）和 comments（用于显示评论的容器）的 DOM 元素，并将它们赋值给相应的变量，以便后续操作。

事件监听器绑定：

使用 addEventListener 方法为 submitButton 按钮添加了一个 click 事件监听器。当按钮被点击时，会执行监听器中的回调函数，即后续的评论发布操作逻辑。

点击按钮后的操作逻辑：

在按钮点击回调函数内部，首先获取 usernameInput 和 commentInput 输入框的值，并分别赋值给 username 和 commentText 变量，用于后续处理。

使用 if 语句检查 username 和 commentText 是否都有值（即输入框不为空）。如果都有值，则执行 if 块内的代码，进行评论发布的操作。

创建评论元素：

创建一个新的 <div> 元素，并为其添加 comment 类名，作为评论的容器。

创建一个新的 <p> 元素，用于显示用户名称和评论内容，并将相应的文本内容设置为 userInfo 元素的 textContent。

创建一个新的 <p> 元素，添加 time 类名，用于显示评论的发布时间。获取当前时间，将其转换为本地字符串格式，并设置为 timeElement 元素的 textContent。

创建一个新的 <button> 元素，添加 delete-btn 类名，并设置其 innerHTML 为一个删除图标。为该按钮添加一个 click 事件监听器，当点击按钮时，会调用回调函数 commentDiv.remove()，将包含该按钮的评论容器从页面中移除。

组装并添加评论元素：

将前面创建的用户信息元素 userInfo、时间元素 timeElement 和删除按钮 deleteButton 依次添加到评论容器 commentDiv 中。然后将 commentDiv 添加到 commentsDiv 容器中，从而在页面上显示新发布的评论。

清空输入框：

将 usernameInput 和 commentInput 输入框的值设置为空字符串，以便用户可以继续输入新的评论。

#### 2.6.2 关键代码

|  |
| --- |
| <!DOCTYPE html>  <html lang="zh-CN">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>论坛评论发布</title>  <script src="https://cdn.tailwindcss.com"></script>  <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.7.2/css/all.min.css" rel="stylesheet">  <style>  body {  font-family: 'Inter', sans-serif;  background-color: #f4f4f9;  display: flex;  justify-content: center;  align-items: center;  min-height: 100vh;  margin: 0;  }  .container {  background-color: white;  padding: 2rem;  border-radius: 0.5rem;  box-shadow: 0 4px 6px -1px rgba(0, 0, 0, 0.1), 0 2px 4px -1px rgba(0, 0, 0, 0.06);  width: 100%;  max-width: 600px;  }  .input-group {  margin-bottom: 1rem;  }  .input-group label {  display: block;  margin-bottom: 0.25rem;  font-weight: 500;  color: #4a5568;  }  .input-group input,  .input-group textarea {  width: 100%;  padding: 0.5rem;  border: 1px solid #cbd5e0;  border-radius: 0.25rem;  transition: border-color 0.3s ease;  }  .input-group input:focus,  .input-group textarea:focus {  border-color: #667eea;  outline: none;  }  .btn {  padding: 0.5rem 1rem;  background-color: #667eea;  color: white;  border: none;  border-radius: 0.25rem;  cursor: pointer;  transition: background-color 0.3s ease;  }  .btn:hover {  background-color: #5a67d8;  }  .comment {  border: 1px solid #e2e8f0;  border-radius: 0.25rem;  padding: 1rem;  margin-bottom: 1rem;  position: relative;  }  .comment .delete-btn {  position: absolute;  top: 0.5rem;  right: 0.5rem;  background-color: #e53e3e;  color: white;  border: none;  border-radius: 0.25rem;  padding: 0.25rem 0.5rem;  cursor: pointer;  transition: background-color 0.3s ease;  }  .comment .delete-btn:hover {  background-color: #c53030;  }  .comment .time {  font-size: 0.875rem;  color: #718096;  margin-top: 0.25rem;  }  </style>  </head>  <body>  <div class="container">  <h1 class="text-2xl font-bold mb-4">论坛评论发布</h1>  <div class="input-group">  <label for="username">用户名称</label>  <input type="text" id="username" placeholder="请输入用户名称">  </div>  <div class="input-group">  <label for="comment">评论内容</label>  <textarea id="comment" rows="3" placeholder="请输入评论内容"></textarea>  </div>  <button class="btn" id="submit-comment">发布评论</button>  <div id="comments" class="mt-4"></div>  </div>  <script>  const usernameInput = document.getElementById('username');  const commentInput = document.getElementById('comment');  const submitButton = document.getElementById('submit-comment');  const commentsDiv = document.getElementById('comments');  submitButton.addEventListener('click', function() {  const username = usernameInput.value;  const commentText = commentInput.value;  if (username && commentText) {  const commentDiv = document.createElement('div');  commentDiv.classList.add('comment');  const userInfo = document.createElement('p');  userInfo.textContent = `${username}：${commentText}`;  const timeElement = document.createElement('p');  timeElement.classList.add('time');  const now = new Date();  const timeString = now.toLocaleString();  timeElement.textContent = `发布时间：${timeString}`;  const deleteButton = document.createElement('button');  deleteButton.classList.add('delete-btn');  deleteButton.innerHTML = '<i class="fa-solid fa-trash"></i>';  deleteButton.addEventListener('click', function() {  commentDiv.remove();  });  commentDiv.appendChild(userInfo);  commentDiv.appendChild(timeElement);  commentDiv.appendChild(deleteButton);  commentsDiv.appendChild(commentDiv);  usernameInput.value = '';  commentInput.value = '';  }  });  </script>  </body>  </html> |

#### 2.6.3 运行截图IMG_256

## 三、门户网站设计

**3.1 门户网站主题说明**

无畏契约小评论区

**3.2 网站设计需求分析（网站包含哪些页面，模块和功能）**

**登录注册页面、首页、评论页面、图片展示页面**

**3.3 网页实现使用的js技术说明**

①登录

关键JS代码：

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <script src="https://cdn.tailwindcss.com"></script>  <title>登录注册页面</title>  <style>  body {  font-family: 'Inter', sans-serif;  background: url('2023121403034070.png') no-repeat center center fixed;  background-size: cover;  display: flex;  flex-direction: column;  align-items: center;  justify-content: center;  min-height: 100vh;  margin: 0;  }  </style>  </head>  <body class="bg-gray-100 h-screen flex items-center justify-center">  <div class="bg-white p-8 rounded shadow-md w-full max-w-md">  <h2 class="text-2xl font-bold mb-4">登录注册</h2>  <ul class="flex space-x-4 mb-4">  <li class="cursor-pointer" id="loginTab">登录</li>  <li class="cursor-pointer" id="registerTab">注册</li>  </ul>  <!-- 登录表单 -->  <form id="loginForm" class="space-y-4">  <input type="text" id="loginUsername" placeholder="用户名" class="w-full p-2 border border-gray-300 rounded">  <input type="password" id="loginPassword" placeholder="密码" class="w-full p-2 border border-gray-300 rounded">  <button type="submit" class="w-full bg-blue-500 text-white p-2 rounded">登录</button>  </form>  <!-- 注册表单 -->  <form id="registerForm" class="space-y-4 hidden">  <input type="text" id="registerUsername" placeholder="用户名" class="w-full p-2 border border-gray-300 rounded">  <input type="email" id="registerEmail" placeholder="邮箱" class="w-full p-2 border border-gray-300 rounded">  <input type="password" id="registerPassword" placeholder="密码" class="w-full p-2 border border-gray-300 rounded">  <button type="submit" class="w-full bg-green-500 text-white p-2 rounded">注册</button>  </form>  </div>  <script>  const loginTab = document.getElementById('loginTab');  const registerTab = document.getElementById('registerTab');  const loginForm = document.getElementById('loginForm');  const registerForm = document.getElementById('registerForm');  const users = {}; // 模拟用户数据库  loginTab.addEventListener('click', () => {  loginForm.classList.remove('hidden');  registerForm.classList.add('hidden');  });  registerTab.addEventListener('click', () => {  registerForm.classList.remove('hidden');  loginForm.classList.add('hidden');  });  registerForm.addEventListener('submit', (e) => {  e.preventDefault();  const registerUsername = document.getElementById('registerUsername').value;  const registerPassword = document.getElementById('registerPassword').value;  if (registerUsername in users) {  alert('该用户名已被注册，请选择其他用户名。');  } else {  users[registerUsername] = registerPassword;  alert('注册成功，请登录。');  registerForm.reset();  loginForm.classList.remove('hidden');  registerForm.classList.add('hidden');  }  });  loginForm.addEventListener('submit', (e) => {  e.preventDefault();  const loginUsername = document.getElementById('loginUsername').value;  const loginPassword = document.getElementById('loginPassword').value;  if (users[loginUsername] === loginPassword) {  // 登录成功，跳转到指定页面  window.location.href = "首页.html";  } else {  alert('用户名或密码错误，请重试。');  }  loginForm.reset();  });  </script>  </body>  </html> |

②首页

关键JS代码：

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <script src="https://cdn.tailwindcss.com"></script>  <title>网站首页</title>  <style>  body {  font-family: 'Inter', sans-serif;  background: url('2023121403034070.png') no-repeat center center fixed;  background-size: cover;  display: flex;  flex-direction: column;  align-items: center;  justify-content: center;  min-height: 100vh;  margin: 0;  }  button {  margin: 10px;  padding: 10px 20px;  font-size: 16px;  border: none;  border-radius: 5px;  background-color: #007BFF;  color: white;  cursor: pointer;  transition: background-color 0.3s ease;  }  button:hover {  background-color: #0056b3;  }  /\* 设置图片样式，使其宽度与按钮相近 \*/  img {  width: 160px;  height: auto;  margin-bottom: 20px;  }  /\* 新增按钮框样式 \*/  .button-container {  background-color: white;  padding: 20px;  border-radius: 10px;  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);  display: flex;  flex-direction: column;  align-items: center;  }  </style>  </head>  <body>  <!-- 添加图片元素，将 src 属性替换为你实际的图片路径 -->  <img src="2023121403034070.png" alt="无畏契约">  <h1>欢迎来到我的网站</h1>  <!-- 新增按钮框 -->  <div class="button-container">  <button id="btn1">瓦友论坛</button>  <button id="btn2">枪皮展示</button>  <button id="btn3">跳转网站3</button>  <button id="backToLogin">返回登录界面</button>  </div>  <script>  const btn1 = document.getElementById('btn1');  const btn2 = document.getElementById('btn2');  const btn3 = document.getElementById('btn3');  const btn4 = document.getElementById('btn4');  const backToLogin = document.getElementById('backToLogin');  // 为每个按钮添加点击事件监听器  btn1.addEventListener('click', function () {  window.location.href = '瓦友论坛.html';  });  btn2.addEventListener('click', function () {  window.location.href = '枪皮展示.html';  });  btn3.addEventListener('click', function () {  window.location.href = '许愿池.html';  });  backToLogin.addEventListener('click', function () {  window.location.href = '登录.html';  });  </script>  </body>  </html> |

③评论区

关键JS代码：

|  |
| --- |
| <!DOCTYPE html>  <html lang="zh-CN">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>瓦友论坛</title>  <script src="https://cdn.tailwindcss.com"></script>  <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.7.2/css/all.min.css" rel="stylesheet">  <style>  body {  font-family: 'Inter', sans-serif;  background: url('2023121403034070.png') no-repeat center center fixed;  background-size: cover;  display: flex;  flex-direction: column;  align-items: center;  justify-content: center;  min-height: 100vh;  margin: 0;  }  .container {  background-color: white;  padding: 2rem;  border-radius: 0.5rem;  box-shadow: 0 4px 6px -1px rgba(0, 0, 0, 0.1), 0 2px 4px -1px rgba(0, 0, 0, 0.06);  width: 100%;  max-width: 600px;  }  .input-group {  margin-bottom: 1rem;  }  .input-group label {  display: block;  margin-bottom: 0.25rem;  font-weight: 500;  color: #4a5568;  }  .input-group input,  .input-group textarea {  width: 100%;  padding: 0.5rem;  border: 1px solid #cbd5e0;  border-radius: 0.25rem;  transition: border-color 0.3s ease;  }  .input-group input:focus,  .input-group textarea:focus {  border-color: #667eea;  outline: none;  }  .btn {  padding: 0.5rem 1rem;  background-color: #667eea;  color: white;  border: none;  border-radius: 0.25rem;  cursor: pointer;  transition: background-color 0.3s ease;  }  .btn:hover {  background-color: #5a67d8;  }  .comment {  border: 1px solid #e2e8f0;  border-radius: 0.25rem;  padding: 1rem;  margin-bottom: 1rem;  position: relative;  }  .comment .delete-btn {  position: absolute;  top: 0.5rem;  right: 0.5rem;  background-color: #e53e3e;  color: white;  border: none;  border-radius: 0.25rem;  padding: 0.25rem 0.5rem;  cursor: pointer;  transition: background-color 0.3s ease;  }  .comment .delete-btn:hover {  background-color: #c53030;  }  .comment .time {  font-size: 0.875rem;  color: #718096;  margin-top: 0.25rem;  }  .nav-buttons {  margin-bottom: 1rem;  display: flex;  justify-content: space-between;  }  </style>  </head>  <body>  <div class="container">  <div class="nav-buttons">  <button class="btn" id="backToHome">返回首页</button>  <button class="btn" id="backToLogin">返回登录页面</button>  </div>  <h1 class="text-2xl font-bold mb-4">瓦友论坛</h1>  <div class="input-group">  <label for="username">用户名称</label>  <input type="text" id="username" placeholder="请输入用户名称">  </div>  <div class="input-group">  <label for="comment">评论内容</label>  <textarea id="comment" rows="3" placeholder="请输入评论内容"></textarea>  </div>  <button class="btn" id="submit-comment">发布评论</button>  <div id="comments" class="mt-4"></div>  </div>  <script>  const usernameInput = document.getElementById('username');  const commentInput = document.getElementById('comment');  const submitButton = document.getElementById('submit-comment');  const commentsDiv = document.getElementById('comments');  const backToHomeButton = document.getElementById('backToHome');  const backToLoginButton = document.getElementById('backToLogin');  submitButton.addEventListener('click', function () {  const username = usernameInput.value;  const commentText = commentInput.value;  if (username && commentText) {  const commentDiv = document.createElement('div');  commentDiv.classList.add('comment');  const userInfo = document.createElement('p');  userInfo.textContent = `${username}：${commentText}`;  const timeElement = document.createElement('p');  timeElement.classList.add('time');  const now = new Date();  const timeString = now.toLocaleString();  timeElement.textContent = `发布时间：${timeString}`;  const deleteButton = document.createElement('button');  deleteButton.classList.add('delete-btn');  deleteButton.innerHTML = '<i class="fa-solid fa-trash"></i>';  deleteButton.addEventListener('click', function () {  commentDiv.remove();  });  commentDiv.appendChild(userInfo);  commentDiv.appendChild(timeElement);  commentDiv.appendChild(deleteButton);  commentsDiv.appendChild(commentDiv);  usernameInput.value = '';  commentInput.value = '';  }  });  backToHomeButton.addEventListener('click', function () {  window.location.href = '首页.html';  });  backToLoginButton.addEventListener('click', function () {  window.location.href = '登录.html';  });  </script>  </body>  </html> |

④皮肤图片展示

关键JS代码：

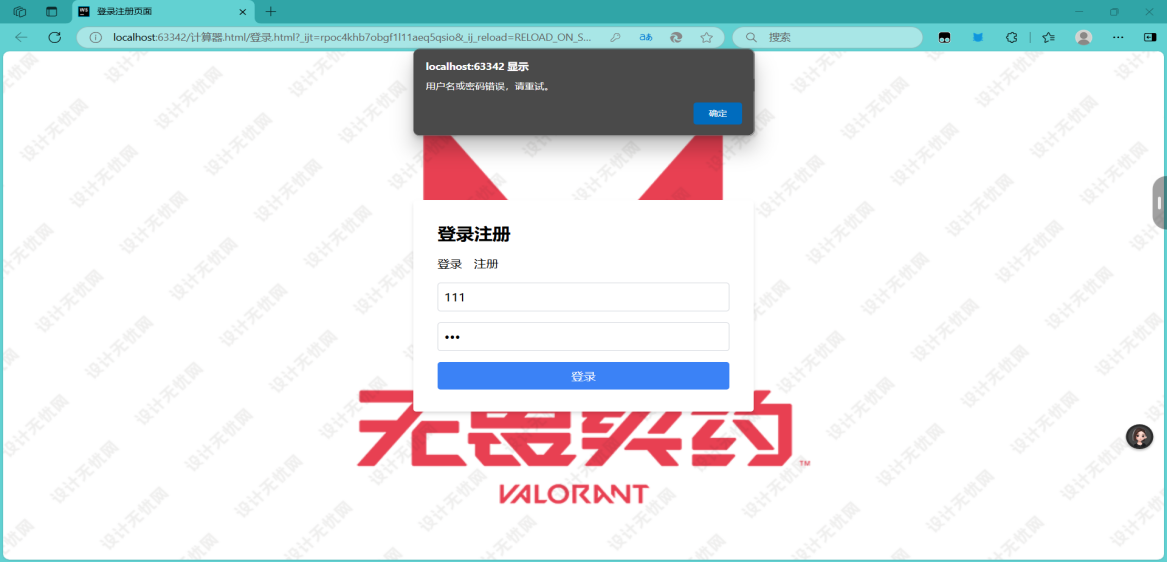
|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>轮播图</title>  <script src="https://unpkg.com/@tailwindcss/browser@4"></script>  <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.7.2/css/all.min.css" rel="stylesheet">  <style>  body {  font-family: 'Inter', sans-serif;  background: url('2023121403034070.png') no-repeat center center fixed;  background-size: cover;  display: flex;  flex-direction: column;  align-items: center;  justify-content: center;  min-height: 100vh;  margin: 0;  }  .nav-buttons {  margin-bottom: 2rem;  display: flex;  gap: 1rem;  }  .btn {  padding: 0.5rem 1rem;  background-color: #667eea;  color: white;  border: none;  border-radius: 0.25rem;  cursor: pointer;  transition: background-color 0.3s ease;  }  .btn:hover {  background-color: #5a67d8;  }  .slider {  position: relative;  /\* 减小最大宽度 \*/  width: 100%;  max-width: 400px;  margin: 0 auto;  overflow: hidden;  border-radius: 0.5rem;  box-shadow: 0 4px 6px -1px rgba(0, 0, 0, 0.1), 0 2px 4px -1px rgba(0, 0, 0, 0.06);  }  .slides {  display: flex;  transition: transform 0.5s ease-in-out;  }  .slide {  min-width: 100%;  position: relative;  }  .slide img {  width: 100%;  height: auto;  object-fit: cover;  border-radius: 0.5rem;  }  .prev,  .next {  position: absolute;  top: 50%;  transform: translateY(-50%);  font-size: 24px;  background-color: rgba(0, 0, 0, 0.5);  color: white;  border: none;  cursor: pointer;  padding: 10px;  border-radius: 50%;  transition: background-color 0.3s ease;  }  .prev {  left: 10px;  }  .next {  right: 10px;  }  .prev:hover,  .next:hover {  background-color: rgba(0, 0, 0, 0.8);  }  .dots {  text-align: center;  margin-top: 10px;  }  .dot {  display: inline-block;  width: 10px;  height: 10px;  background-color: #bbb;  border-radius: 50%;  margin: 0 5px;  cursor: pointer;  transition: background-color 0.3s ease;  }  .dot.active {  background-color: #717171;  }  </style>  </head>  <body>  <div class="nav-buttons">  <button class="btn" id="backToHome">返回首页</button>  <button class="btn" id="backToLogin">返回登录界面</button>  </div>  <div class="slider" id="slider">  <div class="slides" id="slides">  <div class="slide">  <img src="3A3D43331EFACE6505508E8CEFE22CD8.jpg" alt="Slide 1">  </div>  <div class="slide">  <img src="80DF8236DE00692E58FACBDAD89A5B36.jpg" alt="Slide 2">  </div>  <div class="slide">  <img src="EEBDF718866F552A835EF791E667660A.jpg" alt="Slide 3">  </div>  </div>  <button class="prev" onclick="plusSlides(-1)"><i class="fa-solid fa-angle-left"></i></button>  <button class="next" onclick="plusSlides(1)"><i class="fa-solid fa-angle-right"></i></button>  <div class="dots" id="dots"></div>  </div>  <script>  let slideIndex = 1;  showSlides(slideIndex);  let intervalId;  function plusSlides(n) {  showSlides(slideIndex += n);  }  function currentSlide(n) {  showSlides(slideIndex = n);  }  function showSlides(n) {  let i;  let slides = document.getElementsByClassName("slide");  let dots = document.getElementsByClassName("dot");  if (n > slides.length) {  slideIndex = 1;  }  if (n < 1) {  slideIndex = slides.length;  }  for (i = 0; i < slides.length; i++) {  slides[i].style.display = "none";  }  for (i = 0; i < dots.length; i++) {  dots[i].className = dots[i].className.replace(" active", "");  }  slides[slideIndex - 1].style.display = "block";  dots[slideIndex - 1].className += " active";  }  function createDots() {  let slides = document.getElementsByClassName("slide");  let dotsContainer = document.getElementById("dots");  for (let i = 0; i < slides.length; i++) {  let dot = document.createElement("span");  dot.className = "dot";  dot.addEventListener("click", function () {  currentSlide(i + 1);  });  dotsContainer.appendChild(dot);  }  dotsContainer.children[0].classList.add("active");  }  function startAutoSlide() {  intervalId = setInterval(function () {  plusSlides(1);  }, 3000);  }  function stopAutoSlide() {  clearInterval(intervalId);  }  const slider = document.getElementById('slider');  slider.addEventListener('mouseenter', stopAutoSlide);  slider.addEventListener('mouseleave', startAutoSlide);  createDots();  startAutoSlide();  const backToHomeButton = document.getElementById('backToHome');  const backToLoginButton = document.getElementById('backToLogin');  backToHomeButton.addEventListener('click', function () {  window.location.href = '首页.html'; // 请替换为实际的首页地址  });  backToLoginButton.addEventListener('click', function () {  window.location.href = '登录.html'; // 请替换为实际的登录页面地址  });  </script>  </body>  </html> |

⑤许愿池

关键JS代码：

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <script src="https://cdn.tailwindcss.com"></script>  <title>许愿池</title>  <style>  body {  font-family: 'Inter', sans-serif;  background: url('2023121403034070.png') no-repeat center center fixed;  background-size: cover;  display: flex;  flex-direction: column;  align-items: center;  justify-content: center;  min-height: 100vh;  margin: 0;  }  .container {  background-color: white;  padding: 2rem;  border-radius: 0.5rem;  box-shadow: 0 4px 6px -1px rgba(0, 0, 0, 0.1), 0 2px 4px -1px rgba(0, 0, 0, 0.06);  width: 100%;  max-width: 600px;  }  .input-group {  margin-bottom: 1rem;  }  .input-group label {  display: block;  margin-bottom: 0.25rem;  font-weight: 500;  color: #4a5568;  }  .input-group input {  width: 100%;  padding: 0.5rem;  border: 1px solid #cbd5e0;  border-radius: 0.25rem;  transition: border-color 0.3s ease;  }  .input-group input:focus {  border-color: #667eea;  outline: none;  }  .btn {  padding: 0.5rem 1rem;  background-color: #667eea;  color: white;  border: none;  border-radius: 0.25rem;  cursor: pointer;  transition: background-color 0.3s ease;  }  .btn:hover {  background-color: #5a67d8;  }  .wish-list {  margin-top: 2rem;  }  .wish-item {  border: 1px solid #e2e8f0;  border-radius: 0.25rem;  padding: 1rem;  margin-bottom: 1rem;  position: relative;  }  .nav-buttons {  margin-bottom: 1rem;  display: flex;  justify-content: space-between;  }  </style>  </head>  <body>  <div class="container">  <div class="nav-buttons">  <button class="btn" id="backToHome">返回首页</button>  <button class="btn" id="backToLogin">返回登录</button>  </div>  <h1 class="text-2xl font-bold mb-4">许下你的愿望</h1>  <div class="input-group">  <label for="wish">愿望内容</label>  <input type="text" id="wish" placeholder="请输入你的愿望">  </div>  <button class="btn" id="submit-wish">提交愿望</button>  <div class="wish-list" id="wish-list"></div>  </div>  <script>  const wishInput = document.getElementById('wish');  const submitButton = document.getElementById('submit-wish');  const wishList = document.getElementById('wish-list');  const backToHomeButton = document.getElementById('backToHome');  const backToLoginButton = document.getElementById('backToLogin');  submitButton.addEventListener('click', function () {  const wishText = wishInput.value;  if (wishText) {  const wishItem = document.createElement('div');  wishItem.classList.add('wish-item');  wishItem.textContent = wishText;  wishList.appendChild(wishItem);  wishInput.value = '';  }  });  backToHomeButton.addEventListener('click', function () {  window.location.href = '首页.html'; // 请替换为实际的首页地址  });  backToLoginButton.addEventListener('click', function () {  window.location.href = '登录.html'; // 请替换为实际的登录页面地址  });  </script>  </body>  </html> |

## 四、网站测试效果图

**

## c71e23d789dcba62ef80689dd17be86f

## c71e23d789dcba62ef80689dd17be86f

## 7e56c41eec81e2e2cf8e0658b8f30049

## 55960aa6a4f9cf0ecd120e5c073bed75

## 23270da207944798bd2b02e7093593569e8077296a307e8102ab0fbae4291f5aff55b68f5669e87eab09240f35dfe602

## 五、心得体会

①遇到问题:有一部分想法不知道如何用代码表示出来

解决方法：询问ai

感受到了前端技术的有趣，第一次接触面对对象的代码感到十分的新奇，也感受到了编程学习的不易，但是当代码打完实现出的一个个页面让我十分与成就感，也有几分快乐在其中，让我对于这个专业相关的行业有了一定的兴趣。