**深 圳 大 学 实 验 报 告**

**课程名称： 基于Web的编程**

**实验项目名称： JS综合实验**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术**

**指导教师： 蔡树彬**

**报告人：梁润宇 学号： 2021220003 班级： 2班**

**实验时间： 2023年11月15日至2023年12月26日**

**实验报告提交时间： 2023年12月23日**

**教务处制**

|  |
| --- |
| **实验目的与要求：**  **目的：**使用JS，调研并设计、实现图片上传和拖拽排序模块。  要求: 1. 调研相关主流网站的图片上传和拖拽排序模块的设计要点。   1. 设计实现自己网站的图片上传和拖拽排序模块。 2. 要求该图片上传和排序模块用户体验良好。 |
| **方法、步骤：**  要完成本实验，依据实验要求进行分解，需要完成的实验步骤是：   1. 主流网站图片上传和拖拽排序功能的调研   当前大部分UGC网站均有图片上传和排序功能。请选择一个具有这些功能，并有较大流量和一定规模的网站进行调研。  提示：上传时，是否允许用户选择多张图片或非图片的内容？如何处理上传文件过大或网络传输异常问题；如何对已上传的文件进行排序和归类？如何查阅已上传文件？   1. 自己网站的图片上传和拖拽排序模块的设计   前述调研对你有什么启发，你将如何设计、实现自己网站的图片上传和拖拽排序功能？   1. 基于JS的用户体验检查和改进   请使用列表的形式，列举说明你使用JS，对图片上传和拖拽排序模块的哪些用户体验进行了优化、设计实现？ |
| **实验过程及内容：**   1. 主流网站图片上传和拖拽排序功能的调研。   当前大部分UGC网站均有图片上传和排序功能。选择百度贴吧进行调研。    图1.百度贴吧回复页面  从图中可以看到，百度贴吧支持发送文字，图片，表情，涂鸦（本质上是用户当场绘制的图片）。其中，图片支持本地上传和网络图片。一个回复最多可以发送10张图片。  对于本地上传，打开文件资源管理器进行选择。对于网络图片，输入url进行解析。    图2.图片上传    图3.文件资源管理器页面   * 1. 上传多张图片或非图片的内容   百度贴吧回复栏允许上传最多10张图片，如果在上传图片的窗口选择非图片内容，则上传失败。   * 1. 上传文件过大或网络传输异常   选择一个20Mb的图片上传，提示上传失败。    图4.上传大文件  断开网线后，文件上传中途卡住。    图5.网络传输异常  1.3 图片进行拖拽排序  百度贴吧回复栏支持图片的拖拽排序，把鼠标移动到图片的上方就可以拖动，拖动到指定位置松开鼠标左键就完成排序。图6中，正在拖拽中间的图片。    图6.图片拖拽排序  2. 自己网站的图片上传和拖拽排序模块的设计  2.1文件上传和预览区域：  通过<input type="file">创建文件上传输入框，允许选择多个图片文件。  使用<div>元素作为预览区域，具有sortable类，表示这里可以进行拖拽排序。  2.2图片上传事件处理：  当用户选择图片文件后，触发handleImageUpload函数。  函数遍历选定的文件，为每个文件创建一个包含图片和删除按钮的容器，并将其添加到预览区域。    图7. 图片上传事件处理  在这个函数里，会先检测上传的文件类型时候为图片。若不为图片，则跳过处理。一次性可以上传多张图片。  读取图片之后，创建图片容器，设置允许拖拽，规定预览的最大高度和宽度避免图片超出视线。  每张图片创建一个删除按钮，位于图片右上角。然后把图片和删除按钮一起添加到图片容器内。最后为图片添加拖拽事件，并显示在页面上。  2.3拖拽功能：  (1)拖拽事件监听器：  在每个图片容器上添加了两个拖拽事件监听器：dragstart 和 dragend。  dragstart 事件在用户开始拖拽时触发。  dragend 事件在用户停止拖拽时触发。    图8.监听拖拽目标  (2)dragstart 事件处理：  当用户开始拖拽时，设置了数据传输的类型为纯文本，因为在我们的例子中，实际数据并不是很重要，只需知道发生了拖拽。  添加了 dragging 类，通过这个类可以在拖拽过程中对正在被拖拽的元素应用一些样式变化，例如降低不透明度。  (3)dragend 事件处理：  当用户停止拖拽时，移除了 dragging 类，恢复元素原始样式。  这样可以使拖拽过程中应用的样式效果消失，不再影响元素的外观。  (4)拖拽排序：  监听了预览区域的 dragover 事件，该事件在拖拽元素在目标容器上方移动时触发。  在 dragover 事件中，阻止了默认行为，即浏览器的默认处理方式，以允许我们自定义拖拽效果。  (5)获取拖拽后的位置：  使用 getDragAfterElement 函数计算出拖拽后的位置。  这个函数获取了所有非拖拽元素（.image-container:not(.dragging)），并找到最接近拖拽位置的元素。  在没有找到元素的情况下，即拖拽到了预览区域的末尾，直接将拖拽元素追加到末尾。  在找到元素的情况下，插入拖拽元素到这个元素之前。  2.4删除功能：  删除按钮是一个 button 元素，其文本内容为 "❌"。每个删除按钮添加了一个点击事件监听器。当用户点击删除按钮时，触发相应的事件处理函数。该函数通过 img.remove() 移除了包含图片和删除按钮的图片容器。  效果：点击删除按钮后，相应的图片容器即被从预览区域中移除，用户可以轻松地删除不需要的图片。  2.5样式：  使用CSS设置了一些样式，使得图片容器可识别、可拖拽，以及删除按钮的位置。    图9.CSS样式设计  3.基于JS的用户体验检查和改进   |  |  | | --- | --- | | 问题 | 优化措施 | | 上传的不是图片怎么办？ | 检查文件格式，非图片跳过处理并提示 | | 想一次性上传多张图片怎么办？ | 允许一次性上传多张图片 | | 上传错了怎么办？ | 每张图片都可以删除 | | 图片尺寸过大怎么办？ | 限制图片最大宽度，高度 | | 想调节图片顺序怎么办？ | 实现拖拽排序功能 |  1. 实验结果     图10.实验结果  测试了如下功能：   1. 上传非图片格式的文件：提示“请选择图片文件” 2. 一次上传1张图片，上传2次。上传后删除。 3. 一次上传2张图片，拖拽后改变顺序，然后删除。 |
| **实验结论：**   1. 完成实验后，当你看到自己刚做完的这个功能模块，这次，你还有何感想？   与实验1和实验2相比，本次实验是对一个功能模块进行设计。在实际的网站中，图片上传可以用在头像、背景的设置，富文本框等场合。在网站设计的过程中，开发团队分工对各个模块分别进行实现，一个大型的完善的网站也是由一个个小型的功能米快组装起来的。 |
| **心得体会：**  JavaScript是一门高级、解释型的编程语言，主要用于在网页上实现交互式的效果。它是一门弱类型语言，具有动态性和灵活性，使得它在前端开发中非常流行。以下是一些我对JavaScript的理解和感受：  1.前端开发的基石： JavaScript是前端开发的基础，它负责处理网页上的动态内容和用户交互。通过JavaScript，我们可以动态地修改页面内容、响应用户的操作，并与后端服务器进行通信。  2.异步编程： JavaScript中广泛使用的异步编程模型，例如回调函数、Promise和async/await，使得处理异步操作变得更加高效和方便。这对于处理网络请求、定时器等场景非常重要。  3.生态系统： JavaScript有一个庞大而活跃的生态系统，有丰富的第三方库和框架，如React、Vue、Angular等，它们大大简化了前端开发的复杂性。  4.动态类型和灵活性： JavaScript是一门弱类型语言，这意味着变量的类型可以在运行时改变。这种灵活性有时可以提高开发效率，但也需要开发者注意避免潜在的错误。  5.跨平台： JavaScript不仅限于浏览器环境，还可以通过Node.js在服务器端运行。这种跨平台的特性使得前端和后端可以使用相似的语言和技术栈。  6.不断演进： JavaScript语言一直在不断演进，每年都有新的语法和功能被引入。这种持续改进使得JavaScript能够应对不断变化的前端开发需求。 |
| 指导教师批阅意见：  成绩评定：  指导教师签字：蔡树彬  2023年12月30日 |
| 备注： |