# 技术文档

## 一、项目概述

本项目旨在借助前端开发领域的常用工具和技术，实现高效的数据可视化与特定的图像效果处理。项目运用 HTML、CSS、JavaScript 构成的前端三剑客搭建页面结构、设计样式并实现交互逻辑；引入 Echarts 实现各类可视化图表，直观呈现数据信息；借助 Flexible.js 完成移动端适配，确保页面在不同屏幕尺寸下的显示效果；利用 jQuery.js 简化 DOM 操作，提升开发效率；运用高斯泼溅模型处理计算机视觉（CV）相关技术，制作3D画面。

## 二、技术架构

### 前端三剑客

1. **HTML（超文本标记语言）**：作为页面结构的基础，定义了页面的各类元素，如标题、段落、图像、表格等。通过合理的标签嵌套，构建出清晰、语义化的页面布局，为后续的样式设计和交互实现提供基础框架。
2. **CSS（层叠样式表）**：负责页面的样式设计，包括颜色、字体、布局、动画等方面。利用 CSS 的盒模型、选择器和属性，实现页面元素的精准定位与美化，结合 Flexbox 和 Grid 布局，创建响应式页面，确保在不同设备上都能呈现良好的视觉效果。
3. **JavaScript**：实现页面的交互逻辑，通过操作 DOM 元素，响应用户的各种事件，如点击、输入、滚动等。借助 JavaScript 的异步编程、模块化开发等特性，提升页面的性能和可维护性。

### Echarts 可视化图表

Echarts 是一款基于 JavaScript 的开源可视化库，提供了丰富多样的图表类型，如柱状图、折线图、饼图、散点图等。通过简单的配置项，可轻松创建出交互式的可视化图表。在本项目中，Echarts 从后端获取数据，依据业务需求选择合适的图表类型进行展示，并通过事件绑定实现图表的交互操作，如数据提示、区域缩放、图例切换等。

### Flexible.js

Flexible.js 是专门用于移动端适配的 JavaScript 库，通过动态设置 HTML 的 font - size，实现 rem 单位的灵活运用。在项目开发过程中，Flexible.js 根据设备的屏幕宽度，计算并设置合适的 font - size 值，使得页面元素的尺寸能够根据屏幕大小进行自适应调整，从而保证页面在不同移动端设备上的一致性和可用性。

### jQuery.js

jQuery 是一个快速、简洁的 JavaScript 库，极大地简化了 HTML 文档遍历、事件处理、动画效果和 AJAX 操作。在本项目中，使用 jQuery 选择器快速定位 DOM 元素，绑定事件处理函数，简化复杂的 DOM 操作。

### 高斯泼溅模型

高斯泼溅模型在计算机视觉领域常用于图像模糊、噪声去除等处理。在本项目中，通过对图像应用高斯分布算法，模拟泼溅效果，改变图像的像素分布，生成独特的视觉效果。具体实现过程中，利用 JavaScript 编写算法，对图像的每个像素点进行计算，依据高斯分布的原理调整像素值，从而达到预期的泼溅效果。

## 三、项目实现

### 可视化图表实现

1. **引入 Echarts 库**：在 HTML 文件中，通过<script>标签引入 Echarts 的 JavaScript 文件。
2. **准备 DOM 容器**：在 HTML 中创建一个具有固定尺寸的<div>元素，作为图表的容器。
3. **配置图表参数**：在 JavaScript 文件中，定义图表的配置项，包括图表类型、数据、标题、坐标轴等信息。
4. **渲染图表**：使用 Echarts 的init方法初始化图表，并调用setOption方法将配置项应用到图表上，实现图表的渲染。

### 页面适配

1. **引入 Flexible.js 库**：在 HTML 文件的<head>标签中，通过<script>标签引入 Flexible.js 文件。
2. **使用 rem 单位**：在 CSS 样式表中，将元素的尺寸单位统一设置为 rem，根据设计稿和 Flexible.js 的配置，计算出相应的 rem 值，实现页面元素的自适应。

### DOM 操作与数据交互

1. **引入 jQuery.js 库**：在 HTML 文件中，通过<script>标签引入 jQuery 的 JavaScript 文件。
2. **利用 jQuery 选择器定位元素**：使用 jQuery 的选择器，如$('selector')，快速定位页面中的 DOM 元素。
3. **绑定事件处理函数**：通过$(selector).on('event', function())方法，为元素绑定各种事件处理函数，如点击、鼠标悬停等。

### 高斯泼溅模型实现

1. **获取图像数据**：使用 HTML5 的<canvas>元素获取图像的像素数据。
2. **应用高斯泼溅算法**：编写 JavaScript 函数，实现高斯分布算法，对图像的像素数据进行处理。
3. **更新图像**：将处理后的像素数据重新绘制到<canvas>元素上，显示高斯泼溅效果。

## 四、项目维护

1. **代码审查**：定期对项目代码进行审查，检查代码质量、规范性和可维护性，及时发现并修复潜在问题。
2. **性能优化**：通过分析页面加载时间、资源使用情况等指标，对项目进行性能优化，如图片压缩、代码懒加载等。
3. **功能迭代**：根据业务需求和用户反馈，对项目功能进行迭代升级，添加新功能或改进现有功能。

# 使用说明

**一、系统要求**

正常电脑都可以进行使用，本项目主要使用到TML、CSS、JavaScript 以及 Echarts，主流浏览器通常都支持，但各浏览器的兼容性和性能表现略有不同，在项目运行时，对浏览器的要求具体如下：

1. Chrome 浏览器：建议使用最新稳定版本，Chrome 对 HTML5、CSS3 和 JavaScript 新特性支持良好，能确保 Echarts 可视化效果流畅展示，且在性能和稳定性上表现出色。其内置的开发者工具，可助力用户快速诊断和解决页面加载、交互方面的问题。
2. Firefox 浏览器：同样推荐使用最新版本，Firefox 以对网页标准支持严格而闻名，在渲染 HTML 和 CSS 时能准确呈现设计效果，JavaScript 引擎性能良好，为 Echarts 图表交互提供高效支持。
3. Edge 浏览器：随着技术的更新，Edge 基于 Chromium 内核，对 HTML、CSS、JavaScript 和 Echarts 的支持能力大幅提升，使用最新版可确保项目功能的正常使用。

鉴于不同版本的浏览器对新技术支持程度存在差异，若遇到兼容性问题，可更换至推荐的浏览器进行尝试 。

**二、操作指南​**

**首页：**项目首页清晰展示三大核心板块：古代农业发展时间轴、三维博物馆和可视化大屏。用户点击各板块，即可便捷跳转至对应的专属页面。此外，首页右上角有跳转到古代农业发展与弘扬相关时政新闻的按钮。

**古代农业发展时间轴：**在一维时间轴的设计中，以朝代为划分依据，梳理古代农业发展脉络，展示各朝代农业发展成就，并搭配代表性图片。当鼠标悬浮于图片之上，可获取详细介绍信息。为提升页面美观度，突出展示效果，将对页面中心位置的图片及相关信息进行正常显示，对其余部分实施模糊处理，引导用户聚焦关键信息。点击右上角返回按钮返回首页。

**三维博物馆：**三维博物馆借助知识图谱，对 3D 展示页面进行科学分类。用户只需在图谱中点击相应节点，便可直接跳转至感兴趣的 3D 页面，快速浏览体验对应内容。点击右上角返回按钮返回首页。

**可视化大屏：**可视化大屏围绕 “三才理论” 构建，划分为 “天”“地”“人” 三个板块。各板块分别展示对应内容的数据图表，用户将鼠标悬停在图表上，即可查看详细数据。 在大屏左右两侧，设有切换按钮，用户点击即可在不同可视化大屏间进行跳转。为引导用户系统浏览图表数据，仅在最后 “人” 板块的可视化大屏上设置返回按钮，便于用户结束浏览时，一键返回首页。

**运行手册**

1. **文件获取：**下载项目的全部文件压缩包。下载完成后，解压压缩包，获取完整的项目文件目录。
2. **项目启动：**在解压后的文件夹中，找到名为“HOME”的文件夹并打开。在此文件夹内，找到 “home.html” 文件，直接双击该文件，系统将默认使用您电脑上安装的浏览器打开项目。或者，您也可以先打开浏览器，通过浏览器的 “文件” 菜单，选择 “打开文件” 选项，手动定位到 “home.html” 文件，从而启动项目。
3. **操作指南：**项目成功打开后，可依据使用说明进行操作。这份说明详细介绍了项目各个功能板块的使用方法，能帮助您快速熟悉并充分利用项目的各项功能。