## 说明

以下是对期末报告中自评分各项的具体说明，参见(单击右键下载)[报告模板](http://www.openxy.com/books/web_front/about/%E6%8A%A5%E5%91%8A%E6%A8%A1%E6%9D%BF-%5b%E5%AD%A6%E5%8F%B7%5d-%5b%E5%A7%93%E5%90%8D%5d.docx)

项目总计100分，分为以下各项(每项一般为5分)

## 内容(思政)

* 站点的内容具有正能量，1分
* 有使用价值，值得一看或一用，2分
* 内容与技术之间结合自然紧密，2分

## 网站结构

包含常见的页面类型，如：首页、列表页、详细页、表单页、其它页（如关于、提示页面等），各1分

## 基本语义结构

合理使用语义标记刻画文档结构，以markdown所支持的标记为基本子集，如：

* 基本文档标记 <h1-h6>、<p>、<em> 1分
* 二维类标记 <ul>、<table>等 2分
* 需要转义处理的相关标记 <pre>、<code> 1分
* 容器类标记 <div>、<span>、<article>、<section>、<header>、<main>、<footer>、<section>、<aside>、<nav> 1分
* 元标记:<title>、<meta> 1分

## 多媒体

正确表达和处理各类多媒体资源，如图片、音频、视频等

音频与视频文件请使用外链或限制文件尺寸。github上有限制

## 表单控件

包含五类控件，如单行文本、多行文本、单选、多选、下拉列表等，各1分

## 基本排版

基本的版面效果适合阅读，重点如下

* 字体声明：安全字体（fallback机制） 1分
* 字体单位的合理使用：em、rem等 1分

基于normalize.css的基本样式扩展 3分

## 定位

使用绝对定位、固定定位、相对定位等任一方式实现前端特效，如菜单、悬浮窗等等不限，以实现的技巧性或复杂度为标准

## 布局

实现横向布局，只允许使用float或flexbox这两种布局技术

* 基本的三行两列布局：2分
* 嵌套布局或复杂的布局：3分

## 响应式设计

站点对不同设备或尺寸的自适应性的解决

* 移动优先的设计原则和viewport元标记的设置 1分
* media query的使用 1分
* 断点breakpoint的设置 1分
* 响应式设计效果，如不同尺寸屏幕采用不同布局或图片集设置等 2分

## DOM

结合 DOM API 实现页面元素的动态增删改，涉及使用5类api

## 数据存取

结合 AJAX、本地存储等技术实现数据的动态读取与存储

* 基于ajax数据的GET 2分
* 数据的存储和读取（基于本地存储） 2分
* 基于ajax数据的POST 1分

为保证项目部署到github上仍能正常运行，请使用本地存储。如果你希望使用自建后端或json-server之类的方案，可在验收时进行演示，作为另外的加分项

## 表单处理

基于客户端脚本，实现表单的本地提交与处理，包括：

* 表单填写流程的处理与跳转(正确、错误等情况) 3分
* 客户端字段验证 1分
* 表单填写错误的信息提示 1分

## 前端模板

合理使用前端模板引擎库，如站点中各处的重复部分、动态内容的渲染等。评分标准：

* 模板的设计和实现，如对模板语言特性的充分使用、模板文件的划分(独立性、共享性)等 3分
* 模板的调用 2分

## 前端路由

基于浏览器api或第三方插件，实现站点各页面的前端路由化，即可分享、可收藏。评分标准：

* 页面路由的合理设计，即路由表的设计、各路由url的命名有规律。 2分
* 单页化spa：整个站点只有一个入口页面(即index.html)，但可动态生成多个页 3分

## 项目管理与部署

* 使用git进行代码管理，基本的git设置(如.gitignore等),有连续的commit且规范, 2分

托管在github或类似站点上 1分

* 使用github pages等进行静态站点部署 2分

## 前端项目包管理

* 利用node和npm进行开发，如使用simple http server 之类的本地Web服务器进行开发 1分
* 引用的第三方javascript库，如Jquery等，统统通过npm的方式引入和管理 2分
* 编写简单的构建脚本对项目进行构建（譬如sass之类文件的转译、代码压缩等 ） 2分

## 代码模块化与规范化

* 文件布局合理有规律，符合主流约定，可参看vue项目等文件布局及文件命名等 1分

文件名命名遵行一定的规则，如：index.html、 show.html、new.html、edit.html、form.html等

css、js等基本为外联文件，且归类放置在各目录下,如 /css，/js等，考察Web路径的使用

图片等附件文件有统一的格式，譬如png或jpg，且放置在统一的目录下，如/img等

代码规范：如js的各类命名([standard](https://standardjs.com/readme-zhcn.html))、css class的命名(如BEM规范) 1分

* 模块化：代码的封装、共享、重用等 2分
* 代码检查工具的使用，如eslint等工具的引入和配置 1分

## 扩展技术

总分15分。合理使用(不可以为用而用)以下任意技术，如：

* 雪碧图 3分

图标字体（含字体集定制） 3分

CSS渐变、转换 3分

* CSS动画 3分

cookies 3分

* jquery plugin(仅限一个) 3分
* 各类浏览器api(仅限一个) 3分
* sass 3分
* js的高级特性，如定时器、异步处理等 3分
* 兼容性问题，如对IE10的兼容 3分
* 可访问性：如对听觉设备、打印设备的优化 3分
* 参照上述分值，使用到的其它任意技术，可自行评分并写明理由

样式相关的累计分值不可超过7分（含）。

上述分值为参考，可依具体的实现难度申请增分

## 其它约定

* 仅允许使用jquery、ES6（但需转换为plain js）
* 不允许使用vue、bootstrap等库
* 只依赖Web服务器，不依赖jsp、php等后端脚本技术