**学生承诺书**

摘 要

本文以生活中需要的常用信息（图书，音乐，电影）为研究对象，设计开发了在线生活百科系统，有效的实现了信息的查询。本设计还可以针对登录后的用户可以使用收藏功能。

本设计主要使用React+Redux+Webpack+Babel+Bootrap技术实现前端页面与逻辑的构建。该系统核心分为两部分内容：一，三大模块（图书，音乐，电影）的信息显示，具体条目的细节显示。二，查询模块，根据关键字查询需要的内容。

关键词：生活百科 前端 React Redux Webpack

目录

**未找到目录项。**

1. 前言
   1. 课题设计背景，意义及目的
      1. 课题设计背景

90年代的音乐黄金时代过去后，华语乐坛就一蹶不振，究其原因并不都是因为互联网时代的到来，而是由于中国大陆对于版权的保护制度还有实施的措施不到位，完全更不上技术的发展而导致的。而且，由于历史原因，国人一直都有“免费听歌”的思想惯性。不过，2015年以后，国内的音乐产业迎来了生机，各大音乐平台进行的“版权之战”，加上不断完善的版权制度，在互联网的大背景之下，音乐产业进行着革新，例如：付费专辑，现场的演出直播，网络平台直播等。都大大提高了音乐人的生存环境。

近几年国内发展迅速，电影市场在未来也将保持快速发展。单单2015年期间国内的票房收入就总共达到440多亿元，共有12多亿次观影记录。不过，中国电影票房规模总量虽大，但人均观影次数还是达不到一次，所以国内电影市场发展潜力还是非常的大。

* + 1. 课题设计的意义及目的

随着人们生活质量水平提高，娱乐产业也越来越得到重视。例如：音乐，电影等越来越流行。人们不再只是单纯的为了听而听，看而看。而是挑着看，挑有质量的听。人们对于产品的质量要求越来越高，但同时找不到一个很好了解这些东西的渠道。如果有一个专门提供一些生活中经常需要的东西（图书，音乐和电影等信息）的网站，人们就可以很方便的得到这方面的信息。

* 1. 国内研究概况

近年来，随着互联网的快速发展，微博、微信等开始成为图书，音乐与电影营销的新渠道。在传统营销模式逐渐落于窠臼时，以网络为核心的新媒体营销异军突起，为处于蓬勃发展的中国市场注入了新的血液。作为媒体的新生力量，新媒体使音乐和电影营销的手段更加多样化，将会彻底颠覆既定的营销方式，呈现出新技术下的新气象。一批以点石娱乐为代表的新媒体营销团队，助力中国向着市场化的进程迈进。

1. 可行性分析
   1. 社会可行性

生活百科系统设计是以实时便捷的在线查询为设计思路。在日常生中，人们随时都可能需要查询相关的信息。无论是在家里，路上还是其他地方。只要想查询，就很容易的通过本网站查询到。考虑到移动端的庞大流量，本网站还采取了响应式的开发，兼容不同屏幕的显示问题。完美解决移动端的小屏幕显示问题。

* 1. 技术可行性

生活百科系统设计的大部分是前端的技术。采用SPA（单页面应用程序）大大提到了用户体验。数据都是采用JSON进行传输。系统的主要由：React+Redux+Webpack+Babel+Bootstrap 构建的。React完美释放DOM操作，所以本网站只涉及很少的直接DOM操作（DOM操作交给Jquery）。而且使用Redux全局统一管理数据（例如当前登录用户）。使用Webpack一体化的构建项目，解决文件的依赖，合并，代码实时更新等大多数问题。还有结合Babel翻译ECMA2015语法规范，使开发更加便捷。同时使用Bootstrap作为前端页面的样式的构建框架。大大减轻了重复造轮子的问题。

网站结合MVC思想，React负责渲染View层，Redux（controller）负责管理module。

* 1. 操作可行性

生活百科系统是基于Bomb第三方作为后台服务器的。所以不用担心服务器搭建问题，只要有网络就可以使用。而且由于采用的是B/S（Browser/Server）结构开发。只要终端上有浏览器，就可以随时随地使用。而且大部分功能不需要登录即可使用。有效提高了用户的体验。

1. 系统技术介绍及环境搭建
   1. 系统技术介绍
      1. JavaScript

JavaScript是一个用于浏览器的动态脚本语言。它不需要严格的数据结构定义，支持面向对象（函数）和函数式编程。近几年由于HTML5的出现，前端无疑是最火的，作为前端核心的JavaScript，也出现了非常多有用的框架，诸如：Angular，Vue，React。不过自从NodeJS出现以后，使得JavaScript不再仅限于操作浏览器，现在也能当做一门后台语言作为构建服务器的语言了。

* + 1. Jquery

Jquery是前端发展前期最著名的类库。它能够兼容不同的浏览器，便捷的选择器方法，简单的操作DOM节点，绑定事件等特色功能。它还包装了Ajax请求方法，不用我们自己封装。不过随着前端的快速发展，在JS代码中操作DOM已经不提倡，所以渐渐的现在Jquery的使用数已大大降低，本系统之所以还使用Jquery是因为Bootstrap的JS文件是基于Jquery的，还有少部分不可避免的DOM操作

* + 1. Bootstrap

之所以采用Bootstrap，主要是因为Bootstrap提供了一整套样式还有自定义组件。免去了自己写HTMl结构还有CSS样式。最重要的是Bootstrap是响应式开发的。可以完美解决各种终端不同大小的屏幕显示问题

* + 1. NodeJS

前端的快速发展，离不开NodeJS。NodeJS是一个JavaScript的运行环境。自从NodeJS出现后，前端项目构建就离不开步它。诸如：npm，bower，gulp，webpack，grunt等工具都是基于NodeJS的

* + 1. Webpack

Webpack是当前最流行的前端构建工具了。它具有gulp的大部分功能，诸如代码压缩，自动更新，解析less/sass文件等功能，也同时拥有自己强大的功能：合并文件，打包第三方库等特色功能

* + 1. Babel

传统的JavaScript虽然说也是支持面向对象的，但底层实质还是面向函数编程。但是在2015年发布了ECMA2015（es6，JavaScript标准）以后，JavaScript拥有了很多新功能，诸如：1.同步方式解决异步API调用。2.新增定义变量关键字let，const。3.数据结构的解构等等。大大提高了JavaScript开发效率。不过话说如此，新的东西还是没那么快普及，大部分浏览器都只能部分支持es6新特性，所以需要一个翻译器，这就是Babel。Babel不仅可以实时翻译新语法，还支持先翻译在运行。

* + 1. React

本系统的核心技术，React是一个声明式的前端框架，来源于Facebook的内部项目。React的思想是网站是一个大组件，由一个个小的组件组合起来的。所以React中最小单元就是组件，通过单向数据绑定，可以实时跟新数据和视图。由于React采用虚拟的DOM，只有在数据发生变化时，React只会渲染发生变化的部分。所以渲染效率极高

* + 1. React-Router

作为一个SPA（单页面应用程序），关键点无异于路由部分。本系统使用React-Router解决路由管理问题。整个应用都处于不更新状态，只通过路由的跳转选择渲染不同的组件。提高用户体验。

* + 1. Redux

React与其他两大框架Angular，Vue对比来说，React只用于页面的渲染，就是View层。Module层是做不了的。所以需要结合Redux，Redux是一个应用数据流框架，最主要作用是应用的状态管理。本系统采用了与React结合更好的React-Redux

* 1. 系统开发平台及运行环境
     1. 系统开发平台

本系统后台基于Bmob这个第三方网站解决。能使我更加专注于前端的开发。还有部分API来源于豆瓣的API公共接口。

* + 1. 系统运行环境

操作系统Windows，大部分主流手机系统。

浏览器版本要求：IE 9.0+，Firefox 3.5+，Opera 10.5+，Chrome 5.0+，Safari 3.0+