1. 从使用 Zepto.js 到使用 Vue-cli 脚手架搭建单页面应用。

2. 在 Vue.js 构建的单页面应用中，使用 VueRouter 作前端路由管理器、Vuex 进行状态管理、axios http库， 并在项目中运用 ES6 的解构、箭头函数、模板字符串等新特性、使用 async 函数解决异步问题；利用 npm 命令和配置 webpack 打包静态资源；实现前后端分离。

3. 熟练使用 CSS3 和动画库实现各种动效，以及利用 Flex 实现弹性布局。

4. 熟练使用 Fiddler 抓包，调试接口数据。

5. 熟练使用 jQuery 实现多级菜单展示、订阅、树型部门结构操作等多种功能，熟练使用 AJAX 技术。

6.熟练使用 Echarts、Highcharts 进行数据可视化的图表展示

## **AMD与CMD区别**

AMD 即Asynchronous Module Definition，中文名是异步模块定义的意思。它是一个在浏览器端模块化开发的规范

CMD 即Common Module Definition通用模块定义，CMD规范是国内发展出来的，就像AMD有个requireJS，CMD有个浏览器的实现SeaJS，SeaJS要解决的问题和requireJS一样，只不过在模块定义方式和模块加载（可以说运行、解析）时机上有所不同

关于这两个的区别网上可以搜出一堆文章，简单总结一下

最明显的区别就是在模块定义时对依赖的处理不同

****1、AMD推崇依赖前置，在定义模块的时候就要声明其依赖的模块****   
****2、CMD推崇就近依赖，只有在用到某个模块的时候再去require****   
这种区别各有优劣，只是语法上的差距，而且requireJS和SeaJS都支持对方的写法

AMD和CMD最大的区别是对依赖模块的执行时机处理不同，注意不是加载的时机或者方式不同

很多人说requireJS是异步加载模块，SeaJS是同步加载模块，这么理解实际上是不准确的，其实加载模块都是异步的，只不过AMD依赖前置，js可以方便知道依赖模块是谁，立即加载，而CMD就近依赖，需要使用把模块变为字符串解析一遍才知道依赖了那些模块，这也是很多人诟病CMD的一点，牺牲性能来带来开发的便利性，实际上解析模块用的时间短到可以忽略

为什么我们说两个的区别是依赖模块执行时机不同，为什么很多人认为ADM是异步的，CMD是同步的（除了名字的原因。。。）

同样都是异步加载模块，AMD在加载模块完成后就会执行改模块，所有模块都加载执行完后会进入require的回调函数，执行主逻辑，这样的效果就是依赖模块的执行顺序和书写顺序不一定一致，看网络速度，哪个先下载下来，哪个先执行，但是主逻辑一定在所有依赖加载完成后才执行

CMD加载完某个依赖模块后并不执行，只是下载而已，在所有依赖模块加载完成后进入主逻辑，遇到require语句的时候才执行对应的模块，这样模块的执行顺序和书写顺序是完全一致的

这也是很多人说AMD用户体验好，因为没有延迟，依赖模块提前执行了，CMD性能好，因为只有用户需要的时候才执行的原因

## **RESTful**

一种软件架构风格、设计风格，而不是标准，只是提供了一组设计原则和约束条件。它主要用于客户端和服务器交互类的软件。基于这个风格设计的软件可以更简洁，更有层次，更易于实现缓存等机制。

REST（英文：Representational State Transfer，简称REST）描述了一个架构样式的网络系统，比如 web 应用程序。