一、库、框架的选型（~/package.json）

1.构建、打包工具Webpack 4.x  
sass-loader/babel-loader/file-loader等loaders，web-webpack-plugin/mini-css-extract-plugin等plugins，建议模块化的代码编写、组织方式，对打包后的代码做些压缩、分割、按需加载等优化，webpack本身的优化特性可以活用，社区、github上有大量插件可选。  
2.业务使用到的第三方库、插件等  
axios/lodash,Vue全家桶，FullCalendar(排课插件),Echarts(可视化), ElementUI

二、业务流程（前端）

入口~ /index.js

1. 单页应用程序，请参考routes.js里的路由配置
2. 路由拦截：会根据该路由原信息、登录是否过期、还有是否欠费等条件判断路由的跳转逻辑，分两种情况：
   1. 路由验证通过，正常进入该路由
   2. 验证不通过，将会重定向到登录页面

无论哪种情况，进入后台相关的路由都会经过权限系统的逻辑过程，权限系统如下：

1. 权限系统： 权限系统前提是必须登录验证身份信息，权限就是登录用户的角色、业务类型，用户权限决定进入后台能够使用哪些功能，目前功能维度细分到左侧菜单栏的每一个Tab，具体逻辑由~\js\BLL\SideBarBLL.js控制过滤，拉取到角色对应的菜单列表被渲染到~\vue\management\Sidebar.vue组件，即可完成了权限校验
   1. 权限功能没有细分到单个Tab页内的按钮，这块目前是单独逻辑处理的
2. 排课系统：主要是课表控件，包括展示历史课程、创建新课程等等，课表控件基于FullCalendar插件上开发的，API文档请参考[fullcalendar](https://fullcalendar.io/docs),创建课程会有一些逻辑校验：现在时间、课程时长、角色所属业务是否有欠费等等，源码里都有
3. 其他：数据记录的查询、筛选、增删改查，表格展现，表单提交等等，主要基于ElementUI开发；数据统计的可视化展现，是基于echarts开发的。

三、不足、优化、扩展

现有项目在发布打包规范化、测试流程、重构、代码风格上面都有很大的发展潜力，这些对于项目的成长、维护是有用的。