1. 静态网页vs动态网页  
   静态网页：任何人在任何时间访问，内容都是一成不变的；主要技术有HTML,CSS,JS  
   动态网页:不同的人在不同的时间访问，内容都可能发生改变；一般都需要后台数据库的支持，以及服务器端编程语言的支持
2. Iweb在线课堂项目必需的数据库表结构

企业级应用中的数据，一般保存在关系型数据库中，例如：DB2、Oracle、SQLSever、MYSQL等，这些数据产品都使用基本类似的操作语言：SQL语言，提供了增删改查数据库中数据的功能

本项目计划采用MySQL数据库管理系统。  
  
方式1 :安装独立的MySQL服务器，在Windows服务控制台中启动和停止。如果想连接该服务器，还需要再安装独立的客户端工具。例如Navicaat for MySQL工具  
  
方式2:安装-款MySQL服务器套装，例如XAMPP，包含Apache服务器、PHP解释器、MySQL/MariaDB 数据库、phpMyAdmin客户端等等工具

XAMPP下载地址：xampp.org

MYSQL数据库管理系统的结构：

①服务器端：真实的存储着所有的数据，用户不能直接查看或操作

②客户端：用户通过客户端连接到服务器端，发起增删查改从而操作服务器中的数据，常用的MYSQL服务器客户端有：

形式1：命令行版本的工具

形式2：桌面版本工具，例如：NavicatForMySQL

形式3：网页版本的工具，例如PHPMyAdmin,新浪云等云数据库采用的都是此种形式

MySQL数据库中提供的加密函数

Password(参数)

①只要提供一个任意长度的字符串或者数字都可以加密得到另一个固定长度的密文

②原文不同的话，得到密文也不一样

③通过密文很难推出原文

④原文发生一点改变，密文会产生巨大的改变

使用服务器编程语言：Node.js

Node.js是一个运行于V8引擎上的js执行环境，提供了远超过经典客户端JavaScriptAPI.

Node.js虽然名字中带着js但与传统的js完全不同——传统的js运行于客户端浏览器中，主要用于实现页面交互效果Node.js不属于客户端的范畴，与Java/php/.net语言类似，运行服务器端主要用于数据库操作、服务端文件IO，网络访问等等

英文官网：nodejs.org

中文镜像：nodejs.cn

1. 使用Node.js创建服务器项目的步骤

①在服务器上安装Node.js解释器，确保在命令行中可以运行

node -v

②创建项目必须的文件夹，例如iweb\_backend,进入此文件夹，创建Node.js项目必须的描述文件：package.json  
 可以手工创建该文件，并用记事本进行编辑  
 也可以使用简化工具来创建此文件：npm init

③在项目根目录下，下载并安装第三模块，例如：

npm i mysql

npm i express

......

④创建项目的入口文件，创建Web服务器，处理客户端提交的HTTP请求，并返回响应消息  
⑤打开cmd命令行工具，使用node解释器运行入口文件

 node main.js  
⑥使用客户端浏览器，发起请求

NPM与Node的关系

Node.js的扩展模块有很多，为了简便开发者的查找和使用，有人专门创建了一个Node.js的扩展模块仓库（npmjs.com），并提供了一个命令行下载管理工具：Node Packag Manager,简称npm

后来版本的Node.js安装包中，已经默认吧npm安装在一起作为官方的扩展模块工具

小知识：

可以使用npm i xxx自动从npmjs.com仓库中下载指定的第三方模块；也可以使用npm i 自动读取package.json的指定的“依赖列表（depenecies）”查找必须的第三方依赖包，并自动从npmjs.com上下载

项目组常见的矛盾：前后端的联动调试

跨域请求问题

如果从一个页面资源中需要请求另一个页面的资源，  
 若这两个资源的协议名或者端口号三者中有任何一个不相同，就称为“跨域请求”，  
 例如  
 页面1：http://127.0.0.1:8848/index.html  
 页面2：<http://127.0.0.1:5050/course/newest>

客户浏览器基于安全原因，默认禁止XHR和Fetch请求跨域资源

经典面试题：请问您有那些方法可以解决浏览器对于XHR的跨域限制？

经典面试题：请问您有哪些方法可以解决浏览器对于fetch的跨域限制？

面试题：客户如何发起异步请求获取动态web服务器中的数据

方式1：传统的XMLHttpRequest对象

优点：浏览器兼容性最好

不足：使用繁琐，可能产生回调地狱

方式2：jQuery.ajax()简化函数

优点：底层就是xhr，兼容性好使用方便

不足：可能产生回调地狱；jquery目前的行情在走下坡路

方式3：使用第三方工具Axios

优点：底层有XHR，基于ES6 Promise，不会产生回调地狱

不足：老IE浏览器不支持Promise；需要下载第三方工具

方式4：使用Fetch

优点：W3C委员会推荐的新技术，用于取代XHR对象；基于ES6 Promise，使用简单

不足：新技术存在兼容性问题，暂时标准中还没提供请求进度查询、请求中途取消等功能

使用fetch方案发起异步请求的方式

①发起GET请求

Fetch（url）

.then(response=>response.json())

.then(data=>{

//data就是响应消息主体转换得到的js数据

})

①发起POST请求

Fetch（url,{method:’POST,body:’}）

.then(response=>response.json())

.then(data=>{

//data就是响应消息主体转换得到的js数据

})

ES6小知识：模板字符串  
let age=20  
let name='Tom'  
let str='用户姓名：name 用户年龄：age'//错误  
let str='用户姓名:'+name+'用户年龄：'+age//正确  
let str=**`**用户姓名: ${name} 用户年龄：+${age}**`**//正确

1. Git和Github  
      项目中，经常需要文件共享、内容合并、历史回溯、代码找回、开发过程追踪等功能，必需有相关的“源代码管理系统（Source Code Manager)”软件。常见的SCM软件有：CVS、SVN、Git.

Git Repository=File + History

本地仓库：每个开发者在自己电脑上维护的文件和历史内容

远程仓库：所有的开发者都可以将自己的本地仓库中的内容合并到一个公共的远程仓库

每个公司可以建立自己的远程仓库，也可以将本公司的项目远程仓库托管到类似github.com网站  
  Git的使用步骤：  
  ①下载并安装Git客户端软件  
  ②运行Git客户端软件，告诉它当前开发者姓名和邮箱  
  ③创建一个用于保存项目文件的目录，将其初始化为git仓库；也可以克隆远程的一个git仓库到本地，也可以一个项目目录。

④开发者在本地项目仓库中创建/删除/修改文件或目录，添加到本地仓库缓存区，最后再一次性提交到本地仓库

⑤晚上下班前，记得把本地仓库中的提交历史推送到远程公共仓库中

⑥第二天上班时，每个开发者都应该从公共仓库中拉取昨天所有的更新到本地仓库