前端任务：

1. 首页制作：
   1. 按GitHub上的设计图布局制作；
   2. GET请求后端今日推荐菜：菜名，主图片，简单描述；  
      API: http://localhost:9090/getTodayFood ；
   3. 点击选择今日推荐按钮后，进入菜谱页面；（需要配置动态路由）   
      API：http://localhost:9090/getDish
   4. 获取全库前三多阅读量的菜；  
      API：http://localhost:9090/getPopular
   5. 获取全库最多点赞的评论和菜的主图片；  
      API：http://localhost:9090/getTopComment
2. 问卷页制作：
   1. 按GitHub上的设计图布局制作；
   2. 第一题若选择是则不显示第二题；
   3. 第二题为多选；其余选项为单选框；
   4. POST请求后端获得老虎机菜品（共5个—菜名、主图片），进入老虎机页面；
   5. API：http://localhost:9090/customize
   6. 提交数据格式：

{

“q1”: true | false,

“q2”: {

“option1”: true | false,

“option2”: true | false,

“option3”: true | false,

“option4”: true | false

},

“q3”: true | false,

“q4”: {

“option1”: true | false,

“option2”: true | false,

“option3”: true | false,

“option4”: true | false

}

}

1. 老虎机页面制作：
   1. 动画完成后，直接进入相应的菜谱页面
2. 菜谱页制作：
   1. 按GitHub上的设计图布局制作；
   2. 根据菜名GET请求后端获取菜谱全部信息；  
      API：http://localhost:9090/getDish
   3. 根据菜名GET请求后端获取对应的评论信息；  
      API: http://localhost:9090/getComments ；
   4. 发送菜谱评论；  
      API: <http://localhost:9090/postComments>;

发送格式：

{

“nickname”: string,

“detail”: string

}

后端任务：

\*如果使用Java的话，建议使用springboot，其内置Tomcat服务器。（需使用IDEA正式版）

1. 首先修改端口号为9090；
2. 返回数据格式均为json；
3. /getTodayFood：
   1. 以当日日期为参数确定今日推荐菜today；
   2. 从数据库读取today的菜名、主图片路径、菜式简介；
   3. 返回格式：

{

“name”: string,

“address”: string,

“description”: string

}

1. /getDish：
   1. 以菜名为条件从数据库取出所有信息并返回；
   2. 返回格式：

{

“id”: number,

“name”: string,

“taste”: string,

“time”: number,

“address”: string,

“description”: string,

“likes”: number,

“views”: number,

“main”: [string],

“other”: [string],

“steps”: {

“descList”: [string],

“picAddList”: [string]

}

}

1. /getPopular：
   1. 返回前三阅读量的菜式；
   2. 返回格式：

{

“dish1”: {

“name”: string,

“address”: string,

“likes”: number.

“views”: number

},

“dish2”: {

“name”: string,

“address”: string,

“likes”: number.

“views”: number

},

“dish3”: {

“name”: string,

“address”: string,

“likes”: number.

“views”: number

}

}

1. /getTopComment：
   1. 从数据库获取最高点赞量的评论；
   2. 返回格式：

{

“nickname”: string,

“details”: string,

“picAddress”: string,

“dishname”: string

}

1. /customize：
   1. 根据用户要求，从数据库返回符合要求的5道菜的菜名和主图片；
   2. 返回格式：

{

“dish1”: {

“name”: string,

“address”: string

},

“dish2”: {

“name”: string,

“address”: string

},

“dish3”: {

“name”: string,

“address”: string

},

“dish4”: {

“name”: string,

“address”: string

},

“dish5”: {

“name”: string,

“address”: string

}

}

1. /getComments：
   1. 根据菜名获取相应的评论
   2. 返回格式：

{

“total”: number, //the number of comments；

“comments”: [comment]

}

\*\*“comment”: {

“nickname”: string,

“details”: string

}

数据库任务：

1. 根据下图建表，给后端提供数据库端口信息；
2. 数据库可以用MongoDB、MySQL等，和后端协调好就行；
3. 搜集菜谱信息，插入数据
4. 菜谱图片分主图片（封面）和步骤图；
   1. 图片资源放在sinocook/src/assets/picture目录下；
   2. 地址只需要相对地址即可，即“asset/picture…”;
   3. 每道菜的图片放一个文件夹，命名：Did（数字）；
   4. 主图片命名cover.jpg，步骤图命名：step1.jpg；

