

**本科毕业设计（论文）**

GRADUATION DESIGN(THESIS)

**学生记录册**

题 目:

姓 名:

学 号:

学 院:

专业班级:

指导教师:

兰州交通大学教务处

**目录**

1.指导教师职责……………………………………………………1

2.学生做毕业设计(论文)要求……………………………………1

3.本科毕业设计（论文）相关规定（节选）………………………2

4.本科毕业设计（论文）学生记录册填写说明……………………4

5.兰州交通大学本科毕业设计（论文）记录…………………5

# 指导教师职责

1.负责确定毕业设计题目，指导学生收集完成毕业设计所需的数据、图纸及相关参考文献资料。

2. 完成开题讲座，下发设计任务书，讲解题目要求、设计或论文撰写方法，并说明有关注意事项。

3.负责安排设计进度，检查每个学生在各阶段完成任务情况，及时督促和调整。

4.督促学生按照任务计划顺利完成毕业设计（论文）工作。

5.因材施教，启发引导，充分调动学生的积极性和创造性，努力培养学生独立分析问题和解决问题以及团结协作能力。

6.严格要求学生，不放任自流，坚决杜绝毕业设计（论文）抄袭、替做等现象的发生。

7.督促学生认真填写《本科毕业设计（论文）学生记录册》中学生记录。

8.在指导毕业设计期间应定期向系（教研室）主任汇报毕业设计进展情况及存在问题，系（教研室）主任应定期向学院主管领导汇报相关问题，并及时解决。

9.应在答辩前一周对学生在整个设计期间的有关表现写出评语，给定指导教师成绩，提交答辩委员会作为答辩成绩的参考。

# 学生做毕业设计(论文)要求

1.学生须按任务书的规定，在教师的指导下独立完成所要求的内容，严禁抄袭。

2.设计（论文）方案合理、可行，图面质量符合规定，说明书文理通顺，书写整洁。

3.论点明确，论据充分，结论正确。

4.做毕业设计（论文）过程中要加强团队协助和创新能力的锻炼。

# 本科毕业设计(论文)相关规定（节选）

根据学校毕业设计（论文）相关规定，本科毕业设计（论文）强调以下要求：

1. 退回制度

（1）教师有督促学生按指定时间、地点到场答疑的义务，也有退回指导学生的权利；

（2）对于不按要求认真做设计的学生，毕业设计工作开始后5周内，教师有权利将学生毕业设计指导任务退回学院，由学院二次指定指导教师，被二次指定教师的学生毕业设计成绩最高评定为中等；

（3）学院二次指定，但未指定到指导教师的学生，其毕业设计（论文）只能随下一年级重做。

2. 查重检测

本科毕业设计论文查重检测是防范学术作假的重要手段，我校本科生毕业设计（论文）须进行查重检测。

（1）所有申请参加毕业设计（论文）答辩的本科生均须参加毕业设计（论文）查重检测（除小语种、建筑和城规专业外）；

（2）查重检测分两次进行，第一次为学生试查重，第二次为正式查重，查重标准为去除本人已发表文献复制比；

（3）对于复制比≤30%的毕业设计（论文），视为通过查重检测；

（4）对于30%<复制比≤50%的毕业设计（论文），需进行修改，并由指导教师审核，学院毕业设计（论文）领导小组认可后再给一次查重机会，再次查重后复制比仍大于30%的视为未通过查重检测；

（5）对于50%<复制比≤60%的毕业设计（论文），视为未通过查重检测，疑似有抄袭行为，学院组织由非学生本人指导教师组成的专家组进行认定，认定有严重抄袭行为的，取消学位申请资格；

（6）对于复制比60%的毕业设计（论文），视为未通过查重检测，认定有严重抄袭行为，取消学位申请资格。

3. 教师、答辩委员会两方决定最终成绩

（1）毕业设计（论文）最终成绩由指导教师评定成绩和答辩成绩两部分组成，两部分成绩各占50%；

（2）两部分成绩按对应百分制给出后再相加平均对应给出总评等级，其中任意一方评定为不及格，总评成绩为不及格；

（3）指导教师评语和成绩在答辩前给出，不及格学生没有答辩资格；

4. 随机抽取答辩

（1）毕业设计（论文）答辩分为校级公开答辩、院级公开答辩和小组答辩三级答辩，其中校、院两级公开答辩要求全体学生到场参加，现场抽取名单，现场给出答辩成绩并计算出总评成绩；

（2）现场抽到的学生不论何种原因缺席者，答辩成绩按0分记；

（3）校、院两级公开答辩中抽取到的参加答辩学生，未通过答辩者，不能参加小组答辩；

（4）校级、院级及小组答辩均不设二次答辩，每位同学只有一次答辩机会，未通过者只能在下一年度重新做毕业设计、再次答辩。

5. 以下情况之一，毕业设计（论文）不及格

（1）没有完成任务书规定的主要工作量；

（2）设计原则与方案选择缺乏足够的依据，存在较大的错误；

（3）说明书概念不清，图纸有严重错误，计算不准确；

（4）答辩时，所做报告不能说明基本问题，回答问题时有原则性错误；

（5）被指导教师退回学院，学院未二次指定到指导教师的学生；

（6）固定指导时间不到场答疑，累计超过三分之一时间；

（7）存在论文作假行为、抄袭行为；

（8）论文（说明书）查重复制比大于60%；

（9）指导教师给定成绩不及格；

（10）答辩成绩不及格。

# 本科毕业设计（论文）学生记录册填写说明

1. 学校给每名做毕业设计（论文）学生发一本《本科毕业设计（论文）学生记录册》，学生每周应将教师指导情况填写在学生记录部分；由教师填写要求及建议，并签字认可，以达到全程记录学生完成学习任务和教师指导情况的目的。
2. 《本科毕业设计（论文）学生记录册》记录页包括两部分：

（1）学生记录部分：本部分由学生填写，写明本周第几次答疑、答疑时间、地点、上次答疑至本次答疑期间所做工作，毕业设计(论文)进度情况，需要解答的问题；

（2）毕业设计指导教师要求及建议部分：本部分由指导教师填写要求及建议、写明学生之前做设计存在问题及下次答疑前应完成任务，指导教师每次指导都须签字、标记指导时间。

1. 学生每周至少答疑2次，《本科毕业设计（论文）学生记录册》中记录页按周设计，同一周的答疑按次写在一页内，书写不下时，写到下一页；答疑次数较多，页数不够的学生可在学校教务处网站下载打印记录页，填写后粘贴在记录册最后。
2. 记录页中学生记录和毕业设计指导教师要求及建议内容必须手写，禁止打印、禁止抄袭，指导教师在每次审核时需核实学生记录内容真实性。
3. 校外做毕业设计（论文）学生的记录册由校外指导教师负责填写要就及建议，学生回校答辩前由校内指导教师审核。
4. 《本科毕业设计（论文）学生记录册》应在答辩时呈交答辩委员会，作为答辩委员会评分的参考资料。
5. 《本科毕业设计（论文）学生记录册》须作为学生毕业设计（论文）的一部分与毕业设计（论文）一起存档保存。

第 4 周

|  |
| --- |
| 学生记录：  本周第一次答疑，2016-03-21，网上答疑  由于之前在老师的帮助下，已经进行过实战项目的开发，因此毕业设计课题定为游戏玩家在线交流平台设计实现,在初次分析系统功能之后，我从前端三种技术选型中决定使用虽然最后解除但最为简单易上手且实战最多的mobx框架，对比Vue与React框架，mobx只需要创建仓库并注入使用即可，然而Vue与React都需要派发请求最后改变数据，操作相对麻烦，在开发效率上，显然mobx更加得天独厚，对于后台，基于我只会nodejs这一种，所以没有选择，但是对于数据库的选择，我选择了我更加熟悉的mongodb，并且使用studio3t作为图形化管理工具。 |
| 毕业设计指导教师要求及建议： |

第 4 周

|  |
| --- |
| 学生记录：  本周第二次答疑，2016-03-23，网上答疑  技术选型是确定了，但是对于具体要实现哪些功能我还并没有想好，不过既然是玩家交流平台，那么帖子，实时交流必然是少不了的内容，经过一番思考，我觉得既然是玩家交流平台，那么前台必然要有帖子板块，帖子详细内容，帖子发表及回复功能，除此之外，还要有个人中心，以此来管理用户自己发布的帖子，对于每个系统必须要有的内容，自然就是用户登录与用户注册板块。说完了前台功能，后台功能自然是对数据的维护，首先肯定要有用户管理，对用户的增删改查，其次还要有对帖子内容，以及评论的增删操作，防止出现不和谐的内容，当然还要有对图片的维护功能，定期更新轮播图的内容，实时更新热点。 |
| 毕业设计指导教师要求及建议： |

第 5 周

|  |
| --- |
| 学生记录：  本周第一次答疑，2016-03-28，网上答疑  确定了功能，建立了数据库模型，接下里就可以着手开发工作了。我首先设计了一张用户表，包含邮箱和密码，还有唯一的id，以及头像，表格的设计非常重要，但是也不是不可以修改，但是尽量包含必须的信息，我想通过面向流程的开发思路来完成各项工作，所以暂时先设计这一张表。我搭建了后台环境，主要使用在学校学过的nodejs，使用express框架，第一步自然是从网上下载下来nodejs和mongodb数据库，nodejs安装好之后就可以直接使用了，最好安装稳定版本，不然最新版本可能出现未知的bug，然后就是尝试连接mongodb数据库，首先要在D盘新建一个名字为data的文件夹，然后cd data再新建一个叫做db的文件夹，这是用来存储mongodb的数据用的，如果没有这两个文件夹，mongodb不知道数据应该放在那里，因此无法使用，然后就是启动电脑的mongodb服务了，进入mongodb的安装路径，然后再进入bin目录下，然后分别双击下面的mongo.exe,mongod.exe与mongos.exe,这是为了保证电脑上的mongodb服务处于运行状态，这样我们才能连接上数据库。 |
| 毕业设计指导教师要求及建议： |

第 5 周

|  |
| --- |
| 学生记录：  本周第二次答疑，2016-03-30，网上答疑  数据库连接好了之后，我们可以开始正式写nodejs代码了，打开首先安装express框架，使用cnpm i express -S ，cnpm是淘宝的镜像，对于国内的小伙伴使用是比较友好的，使用npm安装可能会因为墙的原因失败，-S是保存到package.json文件中，并且应用到上线项目中，这样我们下次克隆项目的时候就可以直接使用cnpm i命令来安装所有依赖了。  紧接着就是mongoose,passport,cookieParser等的安装了，与安装express类似，这里就不在赘述了，其中连接mongoose的核心代码是  var url = 'mongodb://localhost:27017/houtai' //local表示数据库的名称  mongoose.connect(url, { useNewUrlParser: true })      .then(() => { console.log('连接成功') })      .catch(err => console.log(err))      server.listen(port, () => {      console.log(`Server running on port ${port}`);  })  ，这样我们就连接上了mongodb数据库，紧接着开始应用我们的中间件，cookieParsers是用来解析我们的cookie使用的，可以实现一段时间免登录的功能，bodyParser是用来解析我们的url使用的，passport是用来产生token使用的，当我们的中间件使用好后，我们可以开始配置路由了。 |
| 毕业设计指导教师要求及建议： |

第 6 周

|  |
| --- |
| 学生记录：  本周第一次答疑，2016-04-05，网上答疑  如何配置路由呢？首先我新建一个routes目录，然后新建一个文件夹名为api，在api中新建n个我需要的js文件，然后在server.js中引入它，例如  app.use("/api/users", users);  这样我就使用了users这个路由。接入上面创建user表这个话题，对于user，我需要一个User的模型，用来表示user有哪些数据结构，所以我新建一个modes文件夹，下面新建一个User.js文件，在里面写入  const mongoose  = require('mongoose');  const Schema = mongoose.Schema;  //Create Schema  const UserSchema = new Schema({      name: {          type: String,      },      email: {          type: String,      },      password: {          type: String,      },      avatar: {          type: String,      },      date: {          type: Date,          default: Date.now      }  })  module.exports = User = mongoose.model("users", UserSchema) |
| 毕业设计指导教师要求及建议： |

第 6 周

|  |
| --- |
| 学生记录：  本周第二次答疑，2016-04-07，网上答疑  这样便建立了一个User的模型，可以看到里面有姓名，邮箱，密码等等，但是实际过程中我不一定要都用到这些信息，但是写全的话方便以后补充，接下来在user.js中我们写入创建用户的代码首先是/register接口,在注册邮箱之前，首先要查询这个邮箱是否已经存在，如果已经存在我们要给用户返回邮箱已被注册，然后我们就可以根据用户输入的email和password来创建我们的用户了，由于我们在model中并没有使用required字段，因此即使参数少于我们之前定义的model中的字段数量，我们依然可以创建用户，我们还需要bcrypt，用来对用户的密码进行加密，再存储到数据库中，增加我们的数据库的安全性，防止万一我们的数据库被人偷之后，用户的密码直接泄露。然后我们再调用save方法，然后给用户返回注册成功。 |
| 毕业设计指导教师要求及建议： |

第 7 周

|  |
| --- |
| 学生记录：  本周第一次答疑，2016-04-11，网上答疑  然后随之我们就可以写我们的登录功能，这时候就要用到我们前面提到的cookie了，我们创建一个cookie，给它规定3天的有效期，过了这个有效期，之前的token就自动失效了，需要用户重新登录以此来获取token，这样，在一定程度上保证了用户的安全。  紧接着，我们在数据库中查询前端用户发来的请求，比对我们的email，如果没有这个user，我们返回用户不存在，由于前面我们前面使用了bcrypt，因此在比对的时候我们也要使用bcrypt的compare，如果isMatch，这时候我们使用jwt给用户返回token。这时我们就可以在前端存入这个token，进行后续的操作。 |
| 毕业设计指导教师要求及建议： |

第 7 周

|  |
| --- |
| 学生记录：  本周第二次答疑，2016-04-13，网上答疑    搭建了后端环境，紧接着我们可以搭建前端的环境，为了给我们的页面增添趣味，于是使用video标签增加了一个视频作为背景，让我们的页面看起来不那么空洞，然后使用antd的login组件，在搭建前端架构中，我选用了mobx中在2019年中新加入的react-mobx-lite，这同样是一款公共存储仓库，但是与之前不同的是，它允许我们在hooks中使用，由于hooks是2019年react推出的重大更新版本，因此我决定跟随时代潮流，深入学习一下hooks给前端带来的好处。 |
| 毕业设计指导教师要求及建议： |

第 8 周

|  |
| --- |
| 学生记录：  本周第yi次答疑，2016-04-18，网上答疑  接着回到store，在初次探测过程中，我使用了类似原来的context方法来在上下文中去注入store，但是发现这种方法非常的繁琐，而且，每一个store的注入都要再次在app.js中去注入，索性后来看到可以使用useObserver与useStores,使用原来的方式，然后在组件中引入store，类似原来装饰器的注释方法，用起来和以前一样舒服，并且支持了hooks写法，兼容了过去，指引了未来。搞好这一框架的基础工作之后，我便开始正式的写前端的代码了，写好登录页面后，同样在antd的form表单中找到注册的表单，然后进行适当的修改，取到我所需要的邮箱和密码，同样适用之前的video背景，便完成了我的登录注册页面。接下来介绍下我的项目目录文件的原则，我使用了components,containers,router,service,store,utils,assets这几个文件夹，其中components主要存放几个页面公有的组件，containers存放所有的页面，页面里面每个页面都要新建一个文件夹,例如，有一个主要的home.js,然后是home.module.less文件,然后  在同级目录下新建imgs用来存放图片，components存放home.js所需要的组件，这样做的好处是，结构清晰，那个文件依赖于那个文件一目了然，当然，我们多个文件都用到的图片资源，存储到,assets下的imgs中。这样的好处主要是写起来效率很高，找文件和图片非常方便.其它的目录类似。 |
| 毕业设计指导教师要求及建议： |

第 周

|  |
| --- |
| 学生记录：  关于登录的一些细节还有必要描述一下，在登录之后需要使用jwt\_decode来解析我们的用户信息，还需要将token存储到localStorage中，当解析出用户信息之后，我们需要在header中显示出用户名，及邮箱 |
| 毕业设计指导教师要求及建议： |

第 周

|  |
| --- |
| 学生记录： |
| 毕业设计指导教师要求及建议： |

第 周

|  |
| --- |
| 学生记录： |
| 毕业设计指导教师要求及建议： |

第 周

|  |
| --- |
| 学生记录： |
| 毕业设计指导教师要求及建议： |

第 周

|  |
| --- |
| 学生记录： |
| 毕业设计指导教师要求及建议： |

第 周

|  |
| --- |
| 学生记录： |
| 毕业设计指导教师要求及建议： |

第 周

|  |
| --- |
| 学生记录： |
| 毕业设计指导教师要求及建议： |

第 周

|  |
| --- |
| 学生记录： |
| 毕业设计指导教师要求及建议： |

第 周

|  |
| --- |
| 学生记录： |
| 毕业设计指导教师要求及建议： |

第 周

|  |
| --- |
| 学生记录： |
| 毕业设计指导教师要求及建议： |

第 周

|  |
| --- |
| 学生记录： |
| 毕业设计指导教师要求及建议： |