**理科专用**

未命名

**本科生毕业论文（设计）册**

黑体，3号

学 院： 软件学院

专 业： H5

年 级： 2016

学生姓名： 白凯发

指导教师： 王勇

**河北师范大学本科生毕业论文（设计）任务书**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **姓 名** |  | **学院** |  | | **专业** |  | **年级（班）** |  |
| **毕业论文（设计）题目** | |  | | | | | | |
| **指导教师** |  | **职称** |  | **教研室** | |  | **研究方向** |  |
| **论文（设计）基本要求：**包括论文（设计）的基本内容、应完成的基本环节及各环节要求、学生应遵循的学术规范等 | | | | | | | | |
| **论文（设计）研究目标：** | | | | | | | | |
| **主要参考文献：** | | | | | | | | |

指 导 教师： 年 月 日

教研室主任： 年 月 日

**河北师范大学本科生毕业论文（设计）开题报告书**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **姓 名** | |  | **学院** |  | **专业** |  | **年级（班）** |  |
| **论文题目** | |  | | | | | **指导教师** |  |
| **课题论证** |  | | | | | | | |
| **方案设计** |  | | | | | | | |
| **进度计划** |  | | | | | | | |
| **指**  **导**  **教**  **师**  **意见** | **指导教师：**  **年 月 日** | | | | | | | |
| **教研室意见** | **教研室主任：**  **年 月 日** | | | | | | | |

**河北师范大学本科生毕业论文（设计）文献综述**

|  |
| --- |
| 包括国内外研究现状、发展趋势、存在问题，对文献资料进行概括、分析。 |

**河北师范大学本科生毕业论文（设计）翻译文章**

|  |
| --- |
| 列出参阅的外文文献资料的篇目，对其中与研究课题相关的重要文献进行翻译，注明原文的出处并附原文（附在后面）。 |

未命名

**本科生毕业论文（设计）**

黑体，2号

题目：

黑体，3号

学生姓名： 白凯发

指导教师： 王勇

学 院： 软件学院

专 业： H5

年 级： 2016

楷体，小3

完成日期： 年 月 日

学位论文原创性声明

本人所提交的学位论文 ，是在导师的指导下，独立进行研究工作所取得的原创性成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中标明。

本声明的法律后果由本人承担。

论文作者（签名）： 指导教师确认（签名）：

年 月 日 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解河北师范大学有权保留并向国家有关部门或机构送交学位论文的复印件和磁盘，允许论文被查阅和借阅。本人授权河北师范大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其它复制手段保存、汇编学位论文。

已申请保密的论文须添加本句。

保密的学位论文在\_\_\_\_\_年解密后适用本授权书

论文作者（签名）： 指导教师（签名）：

年 月 日 年 月 日

摘 要（黑体小3，居中）

空2格

标题与正文间，空1行

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX。（宋体小4，首行缩进2字符）

空2格

空2格

**关键词（宋体小4，加粗，顶格）：** XXXX XXXX XXXX（宋体小4）

**Abstract（Times New Roman小3，居中）**

标题与正文间，空1行

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.（Times New Roman小4，首行缩进2字符）

空2格

空2格

Key words（Times New Roman小4，加粗，顶格）: XXXX XXXX XXXX（Times New Roman小4）

目 录（黑体小3，居中）

目录中，所有标题序号与标题文字间，空1格

空2格

第1章 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*（黑体小4，顶格） 1

第2章 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 2

2.1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*（宋体小4，缩进两格） 2

2.2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 2

2.3 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* .. 3

2.3.1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*（宋体小4，缩进四格） .. 3

2.3.2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* .. 3

第3章 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 4

3.1 \*\*\*\*\*\*\*\*\*\*\* 5

3.2 \*\*\*\*\*\*\*\*\*\* 5

3.3 \*\*\*\*\*\*\*\*\*\* 7

参考文献 8

附录 ..9

致谢．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．．． 10

论文题目（黑体3号）

1. 一级标题（黑体小3，居中）

1.1 二级标题1（黑体4号，顶格）

正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容[1]。

参考文献标注

1.2 二级标题2

1,2.1三级标题1（黑体小4，顶格）

1.2.2三级标题2

1.2.3三级标题3

（1）四级标题1（宋体小4，缩进两格）

（2）四级标题2

正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容正文内容[[1]](#footnote-0)。（宋体小4，首行缩进2字符）

注释标注

参考文献（黑体小3，居中）

标题与正文间，空1行

[1]作者.文章题目[J]．期刊名称，出版年份，卷号（期数）:起止页码.

[2]著者.书名[M]．版次（注：第一版不标注）.出版地：出版单位，出版年份.

[3]作者.文章题目[A]．主编.论文集名[C],出版地：出版单位，出版年份：起止页码.

[4]作者.论文题目[D]．保存地:保存单位,年份.

宋体（Times New Roman）5号，顶格，标点全部为半角。

[5]报告者.报告题目[R]．报告地:报告会主办单位,报告年份.

[6]作者.文章题目[N]．报纸名,出版日期(版次).

如一篇文章作者人数较多，列举前三位作者姓名，加“等”。

游戏玩家在线交流平台设计实现

项目选型：

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。

功能选型：

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。

着手开发：

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。

后端开发（1）

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。

后端开发（2）

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)

后端开发（3）

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。然给用户返回注册成功。

后端开发（4）

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

前端搭建（1）

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。

前端开发（2）

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。

前端开发（3）

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。登录功能做好后我们开始写用户个人中心页面，其中用邮箱作为用户的头像，然后主要记录用户的好友数，回帖数和自己发布的主题数，外加上注册时间和上次登录时间接口作为补充。通过我们用户的登录后的token，利用jwt-decode可以解析出我们用户的所有信息，里面包含有我们的创建日期，我们将创建日期格式化处理一下，输出到我们的前端页面中，至于我们的上次登录时间，新建一个this\_time接口，使用Scheme的save()方法首先保存我们上次的登录时间，然后当我们需要的时候调用,

ThisTime.find().sort({\_id:-1}).then((time) => {

    res.json(time[0])

  })

这是将表进行排序，第一个数据便是我们上次登录的时间。

前端开发（4）

在将我们的个人中心页面简单搭建之后，我们开始搭建主页面，左边放一个我们的轮播图，彰显出目前主要的几个活动，方便用户交流，右边决定放置几个热点新闻，补充当下的一些实时热点。

1. 注释内容（宋体小5） [↑](#footnote-ref-0)