**学汉语拼音网站项目**

**设计说明书**

**版本：1.0**

**作者：贺伟强**

**机构：山东大学泰山学堂**

**创建日期：2019年3月24日**

目录

[1.项目概述 2](#_Toc4361168)

[1.1 背景介绍 2](#_Toc4361169)

[1.2 项目目标 2](#_Toc4361170)

[2.软件架构 3](#_Toc4361171)

[2.1 需求功能的划分 3](#_Toc4361172)

[2.2 软件层次的划分或开发模式的考虑 3](#_Toc4361173)

[2.3 安全的考虑 4](#_Toc4361174)

[2.4 接口或插件的使用 4](#_Toc4361175)

[2.5 扩展性的考虑 4](#_Toc4361176)

[3.部分代码的实现 5](#_Toc4361177)

[3.1 与数据库的连接实现出题功能 5](#_Toc4361178)

[3.2 kiana助手对应功能的实现 7](#_Toc4361179)

[3.3 一串拼音的播放 10](#_Toc4361180)

[3.4 查看成绩单弹窗的实现 10](#_Toc4361181)

[3.5 其他 11](#_Toc4361182)

[4.总结 12](#_Toc4361183)

[4.1 总结 12](#_Toc4361184)

# 1.项目概述

## 1.1 背景介绍

汉语拼音是学习现代汉字发音的基础，基本的组成方式是声母+韵母（声调）。虽然汉语拼音可以较好的帮助外国人学习汉语，但许多外国人缺乏较好的学习方法和手段。

## 1.2 项目目标

本网站建设的目标是帮助外国人轻松、愉悦、有效的学习汉语拼音，进而加深对汉语的掌握。

# 2.软件架构

## 2.1 需求功能的划分

## 2.2 软件层次的划分或开发模式的考虑

本网站项目采用nodejs搭建服务器，拼音测试功能链接的数据库采用MySQL，运用Ajax实现前端与后端服务器的交流，配合 JQuery等客户端框架，实现个人设计的附加功能。

界面层：

使用HTML+JavaScript+css进行web界面的表现，并运用了JQueryUI来配合css进行某些画面的实现。

业务逻辑层：

使用Ajax进行交流。

数据访问层：

采用MySQL数据库，在nodejs内创建connection与MySQL连接。

## 2.3 安全的考虑

本网站绝大多数数据均存在服务器上，广泛使用Ajax，使用多次传输最后验证的方式，使用户无法从html中获取更多的信息。

## 2.4 接口或插件的使用

1、汉字拼音转换工具<https://github.com/hotoo/pinyin>

2、崩坏学园Kiana页面宠物<https://github.com/Ling97/kiana>

注：该插件功能仅仅实现最基本功能，对于用户操作的反应、对话框的加入、美化等等功能均为本人自己添加

3、home部分又下角的仓鼠<http://abowman.com/google-modules/hamster/>

4、flash小游戏<http://www.17yy.com/s/flash_game/>

5、多个nodejs模块的引用

## 2.5 扩展性的考虑

1、可以进一步美化网站页面

2、可以加入登陆功能，实现历史信息的记录，确保错题数据的安全性

3、可以更加丰富contact，并进行分类

# 3.部分代码的实现

## 3.1 与数据库的连接实现出题功能

|  |
| --- |
| //与mysql数据库的连接  var connection = mysql.createConnection({  host : 'localhost',  user : 'root',  password : '123456',  database : 'Chinese\_character'  }); |

首先与数据库连接。

|  |
| --- |
| //路由234，随一道题返回，包含一个汉字和四个选项  app.get('/234',function(req,res){  connection.query("select \* from chinese\_character.table1 order by rand() limit 1",function getABCD(err,result){  if(err) console.log("Error!");  else{  var x=result[0];  x["A"]=String(pinyin(x.word));  getapinyin()  .then(function getB(data){  x["B"]=data;  //console.log("reach B");  if(Math.random()>0.2&&Levenshtein(x["A"],x["B"])>Math.pow(Math.random()\*10,2)/10)  return getapinyin().then(getB);  else return getapinyin();  })  .then(function getC(data){  x["C"]=data;  //console.log("reach C");  if(Math.random()>0.2&&Levenshtein(x["A"],x["C"])>Math.pow(Math.random()\*10,2)/10)  return getapinyin().then(getC);  else return getapinyin();  })  .then(function getD(data){  x["D"]=data;  //console.log("reach D");  if(Math.random()>0.2&&Levenshtein(x["A"],x["D"])>Math.pow(Math.random()\*10,2)/10)  getapinyin().then(getD);  })  .then(function(){  if(x["A"]==x["B"]||x["A"]==x["C"]||x["A"]==x["D"]||x["B"]==x["C"]||x["B"]==x["D"]||x["C"]==x["D"])  getABCD(err,result);  else{  var tt;  switch(Math.floor(Math.random()\*4)){  case 0:tt=x["A"];x["A"]=x["B"];x["B"]=tt;break;  case 1:tt=x["A"];x["A"]=x["C"];x["C"]=tt;break;  case 2:tt=x["A"];x["A"]=x["C"];x["C"]=tt;break;  }  res.send(result[0]);  }  });  }  });  }); |

接着是具体的实现，getapinyin()函数在后面的代码中有实现，功能是与数据库连接，返回一个对应的promise。

此处有回调问题不好解决，因为与数据库的连接是一个异步的回调函数，因此多次不同逻辑的嵌套或顺序使用难以进行下去。

解决的办法可以是使用函数的递归，但这回大大的降低代码的可读性并增大代码长度。

此处我最终采用了promise的写法，promise对象用于异步操作，它表示一个尚未完成且预计在未来完成的异步操作，可以使用返回promise然后then的写法使代码逻辑清晰，而每一个then的板块内又可以自己递归，用起来效率极高。

此外，此处使用了函数Levenshtein()，该函数采用JavaScript编写，使用了动态规划算法，能够得出两个字符串得编辑距离。在生成选项得过程中，多次采用该函数，能够使得生成得ABCD四个选项的拼音字符串尽量的相似。从而达到更好的出题效果。

## 3.2 kiana助手对应功能的实现

|  |
| --- |
| //设置index  $(this).css("z-index", "999999");  //拖动效果  $(this).dragging({  move: 'both',  randomPosition: data.randomPosition //位置是否随机  });  //创建一个div包括所有内容  $(this).append("<div class='kiana'></div>");  //创建语言气泡div  $(".kiana").append("<div class='kianaLanguage'></div>");  //设置字体样式  $(".kianaLanguage").css({  position: "absolute",  top: -20,  left: -100,  display: "none",  opacity: 0.7,  backgroundColor: "#213C7F",  color: "#fff",  padding: "5px 10px",  fontSize: "14px",  width: 130,  wordWrap: "break-word",  zIndex: 10050,  borderRadius: 4  });  //创建图片div class="kianaImgDiv"  $(".kiana").append("<div class='kianaImgDiv'></div>");  //向图片div里创建图片，用于显示kiana  $(".kianaImgDiv").append("<img id='kianaImg' />");  //kianaImg 默认src=第一张  $("#kianaImg").prop("src", kianaImg1);  //创建mp3 div class="kianaMP3Div"  $(".kiana").append("<div class='kianaMP3Div'></div>");  //向MP3div里创建audio  $(".kianaMP3Div").append("<audio id='kianaAudio'></audio>"); |

这是kiana对应的结构，后面的代码都是在此基础上运作的，大概可以用下图表示。

然后下面是资源配置：

|  |
| --- |
| //图片  var kianaImg1 = "image/kiana-master/src/img/kiana-1.png";  var kianaImg2 = "image/kiana-master/src/img/kiana-2.png";  var kianaImg3 = "image/kiana-master/src/img/kiana-3.gif";  var kianaImg4 = "image/kiana-master/src/img/kiana-4.png";  //MP3  var dragMp3 = "image/kiana-master/src/mp3/kiana\_drag.mp3";  var kiana\_4Mp3 = "image/kiana-master/src/mp3/kiana\_4.mp3";  var kiana\_2Mp3 = "image/kiana-master/src/mp3/kiana\_2.mp3";  var kiana\_11Mp3 = "image/kiana-master/src/mp3/kiana\_11.mp3";  var kiana\_10Mp3 = "image/kiana-master/src/mp3/kiana\_10.mp3";  var kiana\_8Mp3 = "image/kiana-master/src/mp3/kiana\_8.mp3";  //语言  var lan =  [  "呀~~",  "变态！",  "去死去死去死去死！！",  "烦死啦走开走开啦！！！",  "锵锵~",  "要加油哦",  ];  //存语言和MP3的json对象,每个语言对应一个mp3  var LanMp3 = {  "呀~~": dragMp3,  "变态！": kiana\_4Mp3,  "烦死啦走开走开啦！！！": kiana\_11Mp3,  "去死去死去死去死！！": kiana\_2Mp3,  "锵锵~": kiana\_10Mp3,  "要加油哦": kiana\_8Mp3,  }; |

这些对应的图片、音频资源都存在于src文件夹中，将他们事先用变量定义出来。

接下来是kiana的鼠标动作的一系列JQuery代码，这部分较长，且不是我编写的，这里就不展示了。

最后是两个个性化的函数，他们分别在用户答对答错题是被调用，以产生用户答对题，kiana就会兴奋的“锵锵”，打错时kiana就会说“要加油哦”给你加油鼓劲的效果。

|  |
| --- |
| function playwin(){  var o = {  "mp3Src": LanMp3[lan[4]],  "lan": lan[4]  };  Mp3PlayAndShowMsg(o.mp3Src, o.lan);  }  function playfail(){  var o = {  "mp3Src": LanMp3[lan[5]],  "lan": lan[5]  };  Mp3PlayAndShowMsg(o.mp3Src, o.lan);  } |

## 3.3 一串拼音的播放

|  |
| --- |
| function play(arr,now)  {  var x=document.createElement("AUDIO");  x.setAttribute("src","./voice/"+arr[now]+".mp3");  console.log(x.src);  x.playbackRate=2.0;  x.currentTime=0.1;  x.play();  x.addEventListener('ended', function () {  if(now+1<arr.length) play(arr,now+1);  else{  isplaying=false;  $(document).ready(function(){  $("button#next").show();  });  }  }, false);  } |

该函数play()，运用递归的方式一个一个播放拼音音频，并且添加音频的监听器，使得上个拼音播放完后下一个拼音才会继续播放。因为该音频的播放速度不快，影响体验，故此处更改了audio对象的playbackRate属性和currentTime属性。

## 3.4 查看成绩单弹窗的实现

|  |
| --- |
| <link rel="stylesheet" href="/jquery-ui-1.12.1.custom/jquery-ui.min.css" />  <script src="/jquery-ui-1.12.1.custom/jquery-ui.min.js"></script> |

首先调用JqueryUI。

|  |
| --- |
| $("#dialog").dialog({  autoOpen: false,  width: 850,  height: 600,  show: {  effect: "blind",  duration: 1000  },  hide: {  effect: "explode",  duration: 1000  }  }); |

然后只需要简单的使用JqueryUI就能实现炫酷的弹窗效果了。

上面的show和hide定义的是弹窗出现和消失的方式，从上到下的出现，块状扩散消失。

## 3.5 其他

其他的功能实现，比如绘制扇形图、下载、内嵌视频游戏、仓鼠宠物等等，因为篇幅有限，这里就不给出具体的实现了。

# 4.总结

## 4.1 总结

总的来说，本学汉语拼音网站是功能齐全，有小助手与小宠物减少做题过程的枯燥，并且题目的生成采用数据库与算法，既有做题功能又有出题功能，还能给出可点击的统计图与错题本数据下载。缺点也很明显，与一些学习汉语拼音网站相比，本站风格相对跟为简约，不够“炫酷”，并且没有登录登出功能，使得网站安全性仍有待加强。