数据库修改：

在user模块中的用户模型类User中新增如下字段：

USER\_TYPE\_CHOICES = (

(1, '学习者'),

(2, '授业者'),

(3, '管理员')

)

nick\_name = models.CharField(max\_length=50, verbose\_name="昵称", default="")

birthday = models.DateField(verbose\_name="生日", blank=True, null=True)

mobile = models.CharField(max\_length=11, verbose\_name="手机", default="")

addr = models.CharField(max\_length=50, verbose\_name="地址", default="") # 这里是小地址，比如“北京市海淀区”

image = models.ImageField(verbose\_name="用户头像", upload\_to="users/%Y/%m", default="")

user\_type = models.SmallIntegerField(choices=USER\_TYPE\_CHOICES, default=1, verbose\_name='用户类型')

在order模块中的订单模型类OrderInfo中新增：

TEACHING\_MODE\_CHOICES = (

(1, '远程教育'),

(2, '现场授业')

)

teaching\_mode = models.SmallIntegerField(choices=TEACHING\_MODE\_CHOICES, default=1, verbose\_name='授业方式')

需要修改的地方：

发送短信的地方被我暂时注释了（已测试成功）。在user/views.py中的SendMessageView中，我把is\_success直接给设置成了True，需要把is\_success = send\_message(mobile, code)取消注释，并修改其内部参数。

短信接口使用了阿里云短信平台

接口：

|  |  |
| --- | --- |
| 功能 | 显示注册页面 |
| 调用地址 | /user/register |
| 调用形式 | 地址栏 |
| Get/Post | Get |
| 参数 |  |
| 参数形式 |  |
| 返回值 | register.html |
| 返回值形式 | 网页 |
| 说明 | <form method="post" action="/user/register">  {% csrf\_token %}<!--这个是必须要有的，防止跨域攻击-->  <ul>  <li>  <label>手机号码:</label>  <input type="text" placeholder="11位数的手机号码" id="mobile" name="mobile"/>  <span>提示信息</span>  </li>  <li>  <label>验证码:</label>  <input type="text" name="code" placeholder="请输入手机验证码">  <input type="button" value=" 获取验证码" id="zphone" onclick="test()"></li>  <li>  <label>密码:</label>  <input type="password" name="pwd" id="pwd">  <span>提示信息</span>  </li>  <li>  <label>确认密码:</label>  <input type="password" name="cpwd" id="cpwd">  <span>提示信息</span>  </li>  <li>  <input type="checkbox" name="allow" id="allow">  <label>协议“</label>  <span>提示信息</span>  </li>  <li>  <input type="submit" value="注 册" name="">  </li>  </ul>  </form> |

|  |  |
| --- | --- |
| 功能 | 发送短信验证码 |
| 调用地址 | /user/send\_message |
| 调用形式 | ajax |
| Get/Post | Get |
| 参数 | * mobile: 手机号 |
| 参数形式 | json |
| 返回值 | * {"status": -1, "msg": "请输入正确的手机号"} * {'status': 0, "msg": "该手机号已注册，请直接登录"} * {"status": 1, 'msg': '短信发送成功'} * {"status": 2, 'msg': '短信发送失败'} |
| 返回值形式 | json |
| 说明 |  |

|  |  |
| --- | --- |
| 功能 | 提交信息，进行用户注册 |
| 调用地址 | /user/register |
| 调用形式 | form表单（ajax也可以） |
| Get/Post | Post |
| 参数 | * mobile: 手机号 * pwd：密码 * code: 短信验证码 * allow:是否同意协议 |
| 参数形式 | json |
| 返回值 | * {'status': -1, 'msg': '数据不完整'} * {'status': 0, 'msg': '请同意协议'} * {'status': 1, 'msg': '登录成功'} * {'status': 2, 'msg': '验证码和手机号不匹配'} |
| 返回值形式 | json |
| 说明 | * 当返回值status为1，表示注册成功，会自动登录，并且会自动跳转到首页（index.html） * <form method="post" action="/user/register">   {% csrf\_token %}<!--这个是必须要有的，防止跨域攻击-->  </form>   * 不设置表单action时，提交表单时，会向浏览器地址栏中的地址提交数据 ，所以这里的action也可以不写 * allow 使用：   <input type="checkbox" name="allow" id="allow"> |

|  |  |
| --- | --- |
| 功能 | 显示登录页面 |
| 调用地址 | /user/login |
| 调用形式 | 地址栏 |
| Get/Post | Get |
| 参数 |  |
| 参数形式 |  |
| 返回值 | login.html |
| 返回值形式 | 网页 |
| 说明 | <form method="post">  {% csrf\_token %}  <input type="text" name="mobile" value="{{ mobile }}" placeholder="请输入手机号/用户名">  <input type="password" name="pwd" placeholder="请输入密码">  <div>  <input type="checkbox" name="remember" {{ checked }}>  <label>记住用户名</label>  </div>  <input type="submit" name="" value="登录">  </form> |

|  |  |
| --- | --- |
| 功能 | 提交登录信息，进行用户登录 |
| 调用地址 | /user/login |
| 调用形式 | 尽量通过form表单 |
| Get/Post | Post |
| 参数 | * mobile: 手机号 * pwd：密码 |
| 参数形式 | json |
| 返回值 | * {'status': -1, 'msg': '数据不完整'} * {'status': 0, 'msg': '用户名或密码错误'} |
| 返回值形式 | json |
| 说明 | * 登录成功没有返回值，会自动登录，并且会自动跳转到首页（index.html） * 目前仅可使用手机号登录，后期可根据需求并视时间丰富程度增加用户名/邮箱登录的功能 |

|  |  |
| --- | --- |
| 功能 | 退出登录 |
| 调用地址 | <a href="{% url 'user:logout' %}">退出</a> |
| 调用形式 | 把“调用地址”中的内容复制粘贴到网页上即可 |
| Get/Post | 点一下退出按钮即可 |
| 参数 |  |
| 参数形式 |  |
| 返回值 |  |
| 返回值形式 |  |
| 说明 | * 用户退出，并且自动跳转到首页 |

|  |  |
| --- | --- |
| 功能 | 显示网页顶部的欢迎信息 |
| 调用地址 | {% if user.is\_authenticated %}  <div>  欢迎您：{{ user.username }}  <a href="{% url 'user:logout' %}">退出</a>  </div>  {% else %}  <div>  <a href="{% url 'user:login' %}">登录</a>  <a href="{% url 'user:register' %}">注册</a>  </div>  {% endif %} |
| 调用形式 | 把“调用地址”中的内容复制粘贴到网页上即可 |
| Get/Post |  |
| 参数 |  |
| 参数形式 |  |
| 返回值 |  |
| 返回值形式 |  |
| 说明 | * 这个是django的语法，if-else表示判断，用户登录会显示用户名和一个退出按钮，用户没登录则显示登录和注册按钮 |

|  |  |
| --- | --- |
| 功能 | 获取收货地址列表 |
| 调用地址 | /user/address |
| 调用形式 | 地址栏 |
| Get/Post | Get |
| 参数 |  |
| 参数形式 |  |
| 返回值 | * user\_center.html * {'page': 'address', 'address\_list': address\_list} |
| 返回值形式 | address\_list为django形式，可使用如下形式在网页上获取：  <div>  {% if address\_list %}  {% for address in address\_list %}  <p>地址：{{ address.addr }} （{{ address.receiver }} 收） 电话：{{ address.phone }} 邮编：{{ address.zip\_code }}  {{address.is\_default}} </p>  {% endfor %}  {% else %}  无默认地址  {% endif %}  </div> |
| 说明 | * 如果无法掌握django形式的返回值，可提出转换为json形式的返回值 |

|  |  |
| --- | --- |
| 功能 | 收货地址的增加 |
| 调用地址 | /user/address |
| 调用形式 | form表单或ajax |
| Get/Post | Post |
| 参数 | * address\_id: 收货地址的id * receiver: 收件人 * addr: 收件地址 * zip\_code: 邮编（可空） * phone: 电话 * mode：1 |
| 参数形式 | json |
| 返回值 | * {'page': 'address', 'status': -1, 'msg': '数据不完整'} * {'page': 'address', 'status': -1, 'msg': '手机号格式不正确'} * {'page': 'address', 'status': 1, 'msg': '新增收货地址成功'} |
| 返回值形式 | json |
| 说明 | * 在参数中一定要加上mode，并且mode的值必须传1 * form表单记得添加{% csrf\_token %} * 如果使用ajax以post的形式传递数据，则需要在js文件中加上如下代码，以防止CSRF攻击： * // using jQuery * function getCookie(name) { * var cookieValue = null; * if (document.cookie && document.cookie !== '') { * var cookies = document.cookie.split(';'); * for (var i = 0; i < cookies.length; i++) { * var cookie = jQuery.trim(cookies[i]); * // Does this cookie string begin with the name we want? * if (cookie.substring(0, name.length + 1) === (name + '=')) { * cookieValue = decodeURIComponent(cookie.substring(name.length + 1)); * break; * } * } * } * return cookieValue; * } * var csrftoken = getCookie('csrftoken'); * function csrfSafeMethod(method) { * // 这些HTTP方法不要求CSRF包含 * return (/^(GET|HEAD|OPTIONS|TRACE)$/.test(method)); * } * $.ajaxSetup({ * beforeSend: function(xhr, settings) { * if (!csrfSafeMethod(settings.type) && !this.crossDomain) { * xhr.setRequestHeader("X-CSRFToken", csrftoken); * } * } * }); |

|  |  |
| --- | --- |
| 功能 | 收货地址的修改 |
| 调用地址 | /user/address |
| 调用形式 | form表单或ajax |
| Get/Post | Post |
| 参数 | * address\_id: 收货地址的id * receiver: 收件人 * addr: 收件地址 * zip\_code: 邮编（可空） * phone: 电话 * mode：2 |
| 参数形式 | json |
| 返回值 | * {'page': 'address', 'status': -1, 'msg': '数据不完整'} * {'page': 'address', 'status': -1, 'msg': '手机号格式不正确'} * {'page': 'address', 'status': 1, 'msg': '修改成功'} |
| 返回值形式 | json |
| 说明 | * 在参数中一定要加上mode，并且mode的值必须传2 |

|  |  |
| --- | --- |
| 功能 | 收货地址的删除 |
| 调用地址 | /user/address |
| 调用形式 | ajax或form表单 |
| Get/Post | Post |
| 参数 | * address\_id: 收货地址的id * mode：3 |
| 参数形式 | json |
| 返回值 | * {'page': 'address', 'status': 1, 'msg': '删除成功'} |
| 返回值形式 | json |
| 说明 | * 在参数中一定要加上mode，并且mode的值必须传3 |

|  |  |
| --- | --- |
| 功能 | 默认收货地址的设置 |
| 调用地址 | /user/address |
| 调用形式 | ajax或form表单 |
| Get/Post | Post |
| 参数 | * address\_id: 收货地址的id * mode：4 |
| 参数形式 | json |
| 返回值 | * {'page': 'address', 'status': 1, 'msg': '默认地址设置成功'} |
| 返回值形式 | json |
| 说明 | * 在参数中一定要加上mode，并且mode的值必须传4 |

|  |  |
| --- | --- |
| 功能 | 用户信息的获取 |
| 调用地址 | /user |
| 调用形式 | 地址栏 |
| Get/Post | Get |
| 参数 |  |
| 参数形式 |  |
| 返回值 | * 用户名——直接在网页需要的地方写上：{{ user.username }} * 手机号——直接网页需要的地方写上：{{ user.mobile }} |
| 返回值形式 |  |
| 说明 | * 目前用户信息只提供了显示功能，后续根据需要将提供用户信息的修改以及用户密码的修改 |

|  |  |
| --- | --- |
| 功能 | 首页的展示 |
| 调用地址 | /index 或者 直接输入网站地址:端口号 |
| 调用形式 | 地址栏 |
| Get/Post | Get |
| 参数 |  |
| 参数形式 |  |
| 返回值 | * index.html * {'types': 商品分类信息以及各个分类下的几个商品信息, 'goods\_banners': 首页轮播商品信息, 'promotion\_banners': 促销广告信息, 'cart\_count': 购物车商品数量} |
| 返回值形式 | json |
| 说明 | * 前端在网页上可通过如下的代码进行返回值的展示： * 获取购物车信息： * <label>您的购物车中有{{cart\_count}}件商品</label> * 获取其他信息： * <div>   <ul>  {% for type in types %}  <li><a href="#category0{{ forloop.counter }}" class="{{ type.logo }}">{{ type.name }}</a></li>  {% endfor %}  </ul>  <div>  {% for banner in goods\_banners %}  <ul>  <li><a href="{% url 'goods:detail' banner.sku.id %}"><img src="{{ banner.image.url }}" alt="轮播图"></a></li>  </ul>  {% endfor %}  </div>  <div>  {% for banner in promotion\_banners %}  <a href="{{ banner.url }}"><img src="{{ banner.image.url }}"></a>  {% endfor %}  </div>  </div>   * {% for type in types %}   <div>  <div>  <h3 id="category0{{ forloop.counter }}">{{ type.name }}</h3>  <div>  {% for banner in type.title\_banners %}  <a href="{% url 'goods:detail' banner.sku.id %}">{{ banner.sku.name }}</a>  {% endfor %}  </div>  <a href="{% url 'goods:list' type.id 1 %}" id="view\_more">查看更多 ></a>  </div>  <div>  <div><img src="{{ type.image.url }}"></div>  <ul>  {% for banner in type.image\_banners %}  <li>  <h4><a href="{% url 'goods:detail' banner.sku.id %}">{{ banner.sku.name }}</a></h4>  <a href="{% url 'goods:detail' banner.sku.id %}"><img src="{{ banner.sku.image.url }}"></a>  <div>¥ {{ banner.sku.price }}</div>  </li>  {% endfor %}  </ul>  </div>  </div>  {% endfor %} |

|  |  |
| --- | --- |
| 功能 | 详情的展示 |
| 调用地址 | /goods/商品的id |
| 调用形式 | 地址栏 |
| Get/Post | Get |
| 参数 |  |
| 参数形式 |  |
| 返回值 | * detail.html * {'sku': 商品的详细信息对象, 'types': 商品的分类信息, 'sku\_orders': 商品的评论信息, 'new\_skus': 当前商品分类下的其他商品推荐, 'same\_spu\_skus': 其他规格的商品信息, 'cart\_count': 购物车的数量} |
| 返回值形式 | json |
| 说明 | * 前端在网页上可通过如下的代码进行返回值的展示： * <!DOCTYPE html> * <html lang="en"> * <head> * <meta charset="UTF-8"> * <title>商品详情</title> * {% load staticfiles %} * <script type="text/javascript" src="{% static 'js/jquery-3.4.1.min.js' %}"></script> * </head> * <body> * <p>这里假设有个下拉菜单</p> * <div> * <h1>全部商品分类</h1> * <ul> * {% for type in types %} * <li><a href="{% url 'goods:list' type.id 1 %}" class="{{ type.logo }}">{{ type.name }}</a></li> * {% endfor %} * </ul> * </div> * <hr/> * <p>这里就是一个小导航</p> * <div> * <a href="#">全部分类</a> * <span>></span> * <a href="{% url 'goods:list' sku.type.id 1 %}">{{ sku.type.name }}</a> * <span>></span> * <a href="#">商品详情</a> * </div> * <hr/> * <div> * <p>这里是商品的图片，记得限制一下图片的大小</p> * <div><img width="200px" height="200px" src="{{ sku.image.url }}"></div> * <hr/> * <div> * <h3>商品名称：{{ sku.name }}</h3> * <p>商品简述：{{ sku.desc }}</p> * <div> * 商品价格：¥ <span id="goods\_price">{{ sku.price }}</span> * <span>单位：{{ sku.unite }}</span> * </div> * <div> * <label>想买的数量：</label> * <input id="goods\_num" type="text" value="1"/> * <p>商品数量这里可以放一个那种加减号的控件</p> * </div> * <div> * <label>其他规格:</label> * {% if same\_spu\_skus %} * <ul> * {% for sku in same\_spu\_skus %} * <li><a href="{% url 'goods:detail' sku.id %}">{{ sku.name }}</a></li> * {% endfor %} * </ul> * {% else %} * <label>无</label> * {% endif %} * </div> * <div>总价：<span id="total\_price"></span>元</div> * <div> * {% csrf\_token %} * <a href="javascript:;">立即购买</a> * <a href="javascript:;" sku\_id="{{ sku.id }}" id="add\_cart">加入购物车</a> * </div> * </div> * </div> * <hr/> * <div> * <div> * <p>同类商品推荐</p> * <ul> * {% for sku in new\_skus %} * <li> * <a href="{% url 'goods:detail' sku.id %}"><img width="200px" height="200px" src="{{ sku.image.url }}"></a> * <p><a href="{% url 'goods:detail' sku.id %}">{{ sku.name }}</a></p> * <p>￥{{ sku.price }}</p> * </li> * {% endfor %} * </ul> * </div> * <hr/> * <div> * <p>商品的介绍</p> * <p>{{ sku.goods.detail|safe }}</p> * </div> * <hr/> * <div> * <p>商品的评论</p> * {% if sku\_orders %} * {% for order in sku\_orders %} * <p>评论时间：{{ order.update\_time }}&nbsp;&nbsp;用户名:{{ order.order.user.username }}</p> * <p>评论内容:{{ order.comment }}</p> * {% endfor %} * {% else %} * <label>暂无评论</label> * {% endif %} * </div> * </div> * <script type="text/javascript"> * update\_goods\_amount() * // 计算商品的总价格 * function update\_goods\_amount() { * // 获取商品的单价和数量 * price = $('#goods\_price').text() * //alert(price) * count = $('#goods\_num').val() * //alert(count) * // 计算商品的总价 * price = parseFloat(price) * count = parseInt(count) * amount = price\*count * // 设置商品的总价 * $('#total\_price').text(amount.toFixed(2)) * } * // 手动输入商品的数量 * $('#goods\_num').blur(function () { * // 获取用户输入的数目 * count = $(this).val() * // 校验count是否合法 * if (isNaN(count) || count.trim().length==0 || parseInt(count) <=0){ * count = 1 * } * // 重新设置商品的数目 * $(this).val(parseInt(count)) * // 更新商品的总价 * update\_goods\_amount() * }) * $('#add\_cart').click(function(){ * // 获取商品id和商品数量 * sku\_id = $(this).attr('sku\_id') // attr prop * count = $('#goods\_num').val() * csrf = $('input[name="csrfmiddlewaretoken"]').val() * // 组织参数 * params = {'sku\_id':sku\_id, 'count':count, 'csrfmiddlewaretoken':csrf} * // 发起ajax post请求，访问/cart/add, 传递参数:sku\_id count * $.post('/cart/addcart', params, function (msg) { * }) * }) * </script> * </body> * </html> |

|  |  |
| --- | --- |
| 功能 | 商品搜索功能 |
| 调用地址 | /query?q=关键字 |
| 调用形式 | form表单 |
| Get/Post | GET |
| 参数 | * q: 关键字 |
| 参数形式 | <form method="get" action="/query">  <input type="text" name="q" placeholder="搜索商品">  <input type="submit" name="" value="搜索">  </form> |
| 返回值 | * search/search.html #search.html必须建立在search文件夹下 * query: GET过来的关键字 * page: 当前页的page对象 –>遍历page对象，获取到的是SearchResult类的实例对象，对象的属性object才是模型类的对象。 * paginator: 分页paginator对象 |
| 返回值形式 | django形式 |
| 说明 | * 首先需要新建一个search文件夹，然后在search文件夹下新建一个search.html文件（文件夹的名字和html文件的名字都是固定的） * 在search.html文件中，可以这样进行展示： * <!DOCTYPE html> * <html lang="en"> * <head> * <meta charset="UTF-8"> * <title>搜索结果</title> * </head> * <body> * <div> * <p>搜索的关键字是：{{ query }}</p> * </div> * <hr/> * <div> * <p>以下是搜索到的结果</p> * <ul> * {% for item in page %} * <li> * <a href="{% url 'goods:detail' item.object.id %}"><img src="{{ item.object.image.url }}"></a> * <p><a href="{% url 'goods:detail' item.object.id %}">{{ item.object.name }}</a></p> * <div> * <span>￥{{ item.object.price }}</span> * <span>{{ item.object.price}}/{{ item.object.unite }}</span> * <a href="#" title="加入购物车"></a> * </div> * </li> * {% endfor %} * </ul> * </div> * <hr/> * <div> * <p>以下是搜索结果的分页功能</p> * {% if page.has\_previous %} * <a href="/query?q={{ query }}&page={{ page.previous\_page\_number }}"><上一页</a> * {% endif %} * {% for pindex in paginator.page\_range %} * {% if pindex == page.number %} * <a href="/query?q={{ query }}&page={{ pindex }}" style="font-size:30px">{{ pindex }}</a> * {% else %} * <a href="/query?q={{ query }}&page={{ pindex }}">{{ pindex }}</a> * {% endif %} * {% endfor %} * {% if page.has\_next %} * <a href="/query?q={{ query }}&page={{ page.next\_page\_number }}">下一页></a> * {% endif %} * </div> * </body> * </html> |

|  |  |
| --- | --- |
| 功能 | 向购物车添加商品功能 |
| 调用地址 | /cart/addcart |
| 调用形式 | ajax |
| Get/Post | POST |
| 参数 | * sku\_id: 商品的id编号 * count: 本次添加到购物车的商品数量 |
| 参数形式 | json |
| 返回值 | * {'status': -1, 'msg': '请先登录'} * {'status': -1, 'msg': '数据不完整'} * {'status': -1, 'msg': '请输入正确的商品数量'} * {'status': -1, 'msg': '该商品不存在'} * {'status': -1, 'msg': '该商品库存不足'} * {'status': 1, 'cart\_count': total\_count, 'msg': '添加成功'} |
| 返回值形式 | Json |
| 说明 |  |

|  |  |
| --- | --- |
| 功能 | 购物车页面显示功能 |
| 调用地址 | /cart/ |
| 调用形式 | 地址栏 |
| Get/Post | GET |
| 参数 |  |
| 参数形式 |  |
| 返回值 | * cart.html * {'skus': 商品的对象列表（里面包含了该用户购物车中的全部信息）, 'cart\_count': 该用户购物车商品的总数量, 'cart\_price': 该用户商品的总价格} |
| 返回值形式 | django形式 |
| 说明 | * <!DOCTYPE html> * <html lang="en"> * <head> * <meta charset="UTF-8"> * <title>购物车</title> * {% load staticfiles %} * <script type="text/javascript" src="{% static 'js/jquery-3.4.1.min.js' %}"></script> * </head> * <body> * <div> * <p>全部商品 {{ cart\_count }} 件</p> * <form method="post" action="{% url 'order:orderpage' %}"> * {% csrf\_token %} * <p>如果新增了店铺表，商品有店铺的话，应该每个店铺有一个for循环</p> * <p>这里以后放上<a href="#">店铺名称</a></p> * <table border="1" cellspacing="0"> * <tr> * <th></th> * <th colspan="2">商品名称</th> * <th>商品规格</th> * <th>商品价格</th> * <th>数量</th> * <th>小计</th> * <th>操作</th> * </tr> * {% for sku in skus %} * <tr name="goods\_list"> * <td><input type="checkbox" name="sku\_ids" value="{{ sku.id }}" checked="checked"></td> * <td><a href="{% url 'goods:detail' sku.id %}"><img src="{{ sku.image.url }}"></a></td> * <td>{{ sku.name }}<br>{{ sku.price }}元/{{ sku.unite }}</td> * <td>{{ sku.unite }}</td> * <td>{{ sku.price }}元</td> * <td><input type="text" sku\_id="{{ sku.id }}" value="{{ sku.count }}"></td> * <td>{{ sku.amount }}元</td> * <td><a href="javascript:;">删除</a></td> * </tr> * {% endfor %} * <tr style="border:0px"> * <td><input type="checkbox" name="check\_all" checked="checked"></td> * <td colspan="4">全选</td> * <td colspan="2">合计(不含运费)： ¥ {{ cart\_price }} <br>已选商品 {{ cart\_count }}件</td> * <td><input type="submit" value="结算"></td> * </tr> * </table> * </form> * </div> * <script> * //全选，全不选 * $('input[name="check\_all"]').click(function() { * is\_checked = $(this).prop('checked') * $('input[name="sku\_ids"]').attr('checked',is\_checked) * } * ) * </script> * </body> * </html> |