

大作业 1

一、作业内容

设计并实现一套基于 Spring Boot 的新能源汽车销售系统，该系统面向管理员和普通用户（购车客户）两类核心角色，需满足以下功能需求：

1) 管理员核心功能

- 车辆信息管理：添加、编辑、删除新能源汽车基础信息，包括车型名称、品牌、续航里程、电池类型、指导价、库存数量、上市时间、配置参数（如快充时间、智能驾驶等级）等。
- 库存与销售管理：统计库存不足（库存 ≤ 5 ）的车型数量及详情；查看所有销售订单（含订单编号、购车用户、车型、下单时间、支付状态、提车状态等），支持订单状态更新（如确认支付、标记提车）。
- 用户信息管理：添加、编辑、删除用户基础信息（姓名、手机号、身份证号、地址、注册时间），可查看用户的购车记录及浏览偏好。
- 促销活动管理：新增、修改、下架促销活动（如车型折扣、购车送充电桩、置换补贴），关联具体车型并设置活动有效期。
- 数据统计分析：统计指定时间段内的销量 TOP5 车型、销售额、新增用户数；分析不同品牌 / 续航里程车型的销售占比。

2) 普通用户（购车客户）核心功能

- 车辆检索与筛选：支持按品牌、价格区间、续航里程、电池类型等条件搜索 / 筛选新能源汽车；可按价格、续航里程升 / 降序排序。
- 车辆详情查看：查看车型的完整信息（含配置参数、库存状态、促销活动）；查看其他用户对该车型的评价（评分、文字评论、晒单）。
- 订单与购车操作：选择心仪车型下单（需填写提车门店、购车数量、联系方式），支持查看订单状态（待支付、已支付、待提车、已完成）；取消未支付的订单。
- 个人中心管理：查看自己的购车订单历史、待提车提醒；修改个人基础信息（如收货地址、手机号）；查看自己的评价记录。
- 车辆评价：对已完成提车的车型进行评分（1-5 星）和文字评价，支持上传提车 / 用车照片。

- 预约试驾：选择车型和门店，预约试驾时间，提交试驾申请；查看自己的试驾预约记录及审核状态（管理员审核后反馈）。

3) 前端页面要求

- 区分管理员端和用户端登录入口，登录后展示对应角色的功能菜单；
- 页面流程需符合用户操作习惯（如用户检索车辆→查看详情→下单→查看订单的完整链路）；
- 核心页面（如车辆列表、订单详情、个人中心）需布局清晰，交互友好。

4) 选做功能（可加分）

- 实现车辆配置对比功能：用户可选择 2-3 款车型，直观对比核心参数（续航、价格、配置等）。
- 实现智能推荐功能：基于用户浏览记录/筛选偏好，推荐相似车型或高性价比车型。
- 实现购车金融方案关联：为车型匹配不同的金融方案（如首付比例、分期期数、利率），用户可计算月供并选择金融方案下单。
- 实现库存预警提醒：用户可订阅心仪车型的库存提醒，当该车型库存补货后自动推送通知。

二、作业要求

1) 整体设计需遵循数据库系统设计规范，基于 Spring Boot 框架完成开发。后端可以整合 MyBatis/MyBatis-Plus 实现数据库交互，前端可选用 Vue/React/HTML+CSS+JS 等技术栈（无强制要求）。

2) 需提交完整的设计文档，包含：

- 数据字典（所有表的字段名称、数据类型、长度、主键 / 外键、约束条件、字段说明）；
- E-R 图（清晰展示实体（车辆、用户、订单、促销活动等）、实体属性及实体间关系）；
- 数据库物理模型（表结构设计，含表名、字段、索引、主键 / 外键关联）；
- 系统架构说明（Spring Boot 核心模块划分、前后端交互流程）；

3) 提交的代码需保证可正确运行，无语法错误，需包含：

- 完整的项目工程代码（后端 Spring Boot 工程 + 前端页面代码）；
- 数据库初始化脚本（建表语句、基础测试数据）；
- 代码中需添加必要的注释（如核心接口、工具类、业务逻辑方法的功能说明）；

4) 最终提交物清单

- 设计文档（Word/PDF 格式）；
- 完整系统代码（压缩包形式，含工程目录、依赖配置文件）；
- 功能演示录屏（时长 5-8 分钟，展示核心功能操作流程，如管理员添加车辆、用户下单等）；
- 实验报告（含需求分析、设计过程、实现细节、测试用例及结果、遇到的问题及解决方法、总结等）。

三、作业考核标准

1) **功能实现：**核心功能（管理员车辆管理、用户检索购车、订单管理等）全部实现且运行正常，无功能缺失；选做功能完成度（每完成 1 项选做功能可额外加分，总分不超过 10 分）；交互体验（页面无明显卡顿、操作流程无逻辑漏洞）。

2) **数据库设计、E-R 图、物理模型设计合理**，实体及关系映射准确，符合业务逻辑；数据字典完整、规范，字段约束（主键、外键、非空、唯一等）设置合理，无冗余字段；数据库操作（增删改查）高效，核心表添加合理索引，无明显性能问题。

3) 文档与代码规范

- 设计文档、实验报告条理清晰，内容完整，符合学术规范；
- 代码结构清晰（例如，按 Controller/Service/Mapper 等分层设计），命名规范，注释完整且有意义；
- 数据库初始化脚本可直接执行，测试数据覆盖核心功能场景。

四 提交

- 1) 截止日期：2026 年 2 月 5 日
- 2) 将所有文件打包命名为组长姓名-递交组员学号-递交组员名字”（如李四-3111111111-张三.zip）
- 3) 压缩包内附上组员的信息以及每位同学的具体工作