# 面试官问12306项目怎么来的？

在校招前，想准备一个简历上项目，因为谷粒商城和瑞吉外卖已经被大家用的比较多了，所以我就没太考虑用这两个项目。

我在 GitHub 和 Gitee 中找了找类似的开源项目，找到了一个很符合我期望的项目：12306.因为 12306 是一个大规模的在线火车票订购平台，它涉及到许多复杂的业务逻辑和高并发处理，而且知名度也比较高，所以我觉得参与这个项目，我可以学到很多有关软件开发和系统设计方面的知识和技能。

而且，这个项目拥有比较丰富的前后端项目，架构分为 SpringBoot 单体版本和 SpringCloud 聚合版本，整体比较完善。为了完成这个项目，我首先深入学习了 SpringBoot 和 SpringCloud Alibaba 框架，以及与数据库交互的知识。

随后，我对 12306 系统进行了需求分析，包括用户的登录、车票的查询、订单的生成等功能。我在开发过程中遇到了一些挑战，比如如何处理并发访问和保证数据的一致性、RocketMQ 消息防止重复消费等等问题。但通过不断地学习和尝试，我逐渐解决了这些问题，并完成了一个基本功能完善的 12306 项目。

我觉得通过参与这个项目，不仅学到了很多技术知识，还锻炼了解决问题的能力和较多的代码设计能力。这个项目也让我更加了解了软件开发的流程和规范，以及如何在一个复杂的系统中做出合理的设计和优化。

回答核心

注意在回答时，强调你对项目的兴趣和学习的动力，以及你在项目中所学到的技术和经验，这样能更好地展现你的积极性和学习能力。

同时，可以适当强调你在解决问题的能力，这些都是面试官非常关注的点。如果你有小组的话，可以说和团队合作方面的事项，都是加分项。

# 其他项目

# 优惠券后台管理系统

项目描述：助力用户便捷领取和平台分发优惠券红包。平台包括优惠券创建、秒杀、分发、结算等业务。

* 通过责任链模式验证商家创建优惠券提交参数是否正确，保障验证代码高内聚、低耦合，保障了开闭原则。
* 为了支持创建大量优惠券，采用 ShardingSphere分库分表方案，并通过自定义分片算法避免数据倾斜，同时优化深分页查询 SQL，将极端情况下查询时间由4s降至100ms。
* 在使用 EasyExce! 解析 Excel创建优惠券分发任务时,通过线程池 ThreadPoo!Executor 异步执行解析任务降低创建优惠券推送接口响应时间。←
* 通过缓存预热、双重判定锁、空值缓存、布隆过滤器解决大量用户同时查询优惠券可能导致的缓存击穿、缓存穿透、缓存雪崩等问题，有效确保了服务的高可用性。
* 秒杀优惠券业务中，为避免数据库扣减库存成功后添加用户优惠券缓存失败，基于监听 Bin!og 机制异步添加用户优惠券缓存，并采用写后查询策略应对 Redis持久化的数据丢失问题。
* 为避免系统耦合，通过消息队列 Rocketm9 解耦用户优惠券推送和通知逻辑，后台管理模块发送推送消息并由分发模块进行消费。
* 通过创建用户优惠券表的唯一索引支持幂等逻辑,确保优惠券限制条件生效。发放优惠券时,如果用户已领取捕获数据库重复记录异常，并采用乐观自旋机制反复尝试，直至成功添加不重复的记录。
* 基于 CompletableFuture 的多线程并行处理方案,有效降低查询用户可用优惠券时的计算延时,提高系统响应速度。

# 微链

项目描述：帮助用户完成较长链接转换成短链接场景。系统支持高并发短链跳转场景，并提供了分析和跟踪功能帮助用户分析分享行为以及转化率等。

* 基于 Redis+ Session 实现用户登录认证，采用 Bcrypt 算法加密保存密码，权限模型采用 RBAC。
* 创建短链接时，通过 Redis的incrby 实现分布式号段发放，并采用本机原子 id 自增的方式，有效减少了Redis的请求次数，避免了Hash 算法生成短链碰撞而导致的循环生成检索性能开销，将创建短链接单机 QPS由1097 提升至7823。
* 利用容量规划及过期时间机制，实现号段循环重复利用，保障了系统的长期稳定运行。
* 通过 RocketMQ 延时消息支持短链设置有效期设置，异步清理数据库中失效短链接。
* 通过 Sentinel接口访问 QPS 限流保障短链接系统稳定运行，触发限流规则后进行降级处理。

# 虚拟现实多人影院

# 基于MIPS的多周期CPU设计 系统硬件设计

**项目描述**：实现一款兼容MIPS32 指令子集的处理器，完成多周期流水线 CPU 设计

**主要工作**： （1）实现各部分功能模块并进行连接

（2）实现单周期、多周期、带冒险处理的5级流水CPU

**个人收获**： （1）对数据通路图的设计与理解

（2）深入理解单周期、多周期、带冒险处理的5级流水CPU的原理

# 页式虚拟存储管理系统 系统软件设计

**项目描述**：建立一个请求分页存储管理系统的模型

**主要工作**： （1）设计存储空间的分配与回收算法

（2）模拟实现缺页中断支持的逻辑地址到物理地址转换

**个人收获**： （1）熟悉虚拟存储器与局部性原理

（2）熟悉请求分页存储管理方式

（3）掌握页面置换算法：FIFO、LRU

（4）了解页面抖动与工作集