1. **用户模块任务**
2. 用户注册：通过手机号注册，注册成功后设置密码，密码使用密文存储

技术栈：

Spring Boot：提供RESTful API，处理注册请求。

MySQL：存储用户信息（手机号、密码等）。

BCrypt：对密码进行加密存储。

实现：

用户通过手机号注册，后端使用Spring Boot接收请求，验证手机号唯一性。

使用BCrypt对密码进行哈希加密后存储到MySQL。

流程：

用户输入账号，密码，昵称，选择学校后，点击接收验证码，后端接收账号密码数据，并发送验证码信息

用户收到验证码后，点击注册提交表单后端处理：

1：验证验证码是否正确，正确进行下一步，否则返回验证码错误

2：验证手机号是否存在，若存在返回账号已存在异常，否则在数据库表中注册用户信息

1. 用户登录：用户可以根据手机号和密码登录，也可以根据手机号和验证码登录，后端需要实现登录jwt验证。

技术栈：

Spring Security：实现用户认证和授权。

JWT（JSON Web Token）：生成和验证Token，实现无状态登录。

Redis：缓存验证码，支持短信登录。

实现：

用户通过手机号和密码登录时，Spring Security验证密码，生成JWT返回给客户端。

短信登录时，从Redis中获取验证码进行验证。

流程：

选择账号密码登录或者短信验证码登录

账号密码登录：

输入账号密码，与后端数据进行校验，校验通过返回token，失败返回错误信息

短信验证码登录：

用户输入手机号码，先排查手机号是否存在，若不存在显示账号未注册，请先注册，若存在则发送验证码，验证通过后，返回token

1. 绑定学生学校信息，用于区分学校，来展示互助任务

技术栈：

MySQL：存储学校信息和用户-学校绑定关系。

Spring Boot：提供绑定接口。

实现：

用户选择学校后，后端将学校ID与用户ID绑定，存储到MySQL。

1. 查看和编辑个人信息（昵称、头像、联系方式、学校）

技术栈：

Spring Boot：提供RESTful API。

MySQL：存储用户个人信息。

实现：

用户通过API获取和更新个人信息，后端操作MySQL进行数据读写。

1. 修改密码

技术栈：

Spring Security：验证原密码。

BCrypt：对新密码进行加密存储。

实现：

用户输入原密码和新密码，后端验证原密码后，使用BCrypt加密新密码并更新到MySQL。

1. 个人完成或者发布任务接收的评价以及信誉积分展示

技术栈：

MySQL：存储评价和信誉积分。

Spring Boot：提供查询接口。

实现：

用户查看评价和信誉积分时，后端从MySQL中查询数据并返回。

1. 提供忘记密码找回功能

技术栈：

Redis：缓存重置密码的验证码。

Spring Boot：提供验证和重置接口。

实现：

用户通过手机号获取验证码，后端生成验证码并存储到Redis。

用户输入验证码后，后端验证并允许重置密码。

1. **核心功能模块**
2. 发布任务（任务标题、任务分类、任务描述、赏金金额、发布时间，预期完成时间、截至时间、任务优先级（可根据优先级调整赏金金额）、可接单人员信誉等级、任务附加信息图片视频备注等）

技术栈：

Spring Boot：提供任务发布接口。

MySQL：存储任务信息。

Redis：缓存任务列表，提升查询性能。

实现：

用户填写任务信息后，后端将任务数据存储到MySQL，并更新Redis缓存。

1. 任务接单

技术栈：

Spring Boot：提供接单接口。

MySQL：更新任务状态和接单者信息。

实现：

用户接单时，后端更新任务状态为“进行中”，并记录接单者ID

1. 任务状态管理（待接单、进行中、已完成、以取消）

技术栈：

Spring Boot：提供状态更新接口。

MySQL：存储任务状态。

实现：

任务状态变更时，后端更新MySQL中的任务状态字段。

1. 任务展示与筛选

技术栈：

Elasticsearch：支持任务全文搜索和复杂筛选。

Redis：缓存热门任务列表。

实现：

用户搜索任务时，后端从Elasticsearch中查询数据。

热门任务列表从Redis中获取，减少数据库压力。

1. 取消任务

技术栈：

Spring Boot：提供取消接口。

MySQL：更新任务状态为“已取消”。

实现：

用户取消任务时，后端更新任务状态并解冻赏金。

1. **交易模块**
2. 用户虚拟钱包管理（用户的虚拟金额）

技术栈：

MySQL：存储用户钱包余额。

Spring Boot：提供钱包查询和更新接口。

实现：

用户查看钱包余额时，后端从MySQL中查询数据。

1. 赏金的支付与退回

技术栈：

Spring Boot：提供支付和退回接口。

MySQL：更新钱包余额和交易流水。

Redis：缓存交易流水，提升查询性能

实现：

任务完成后，后端从发布者钱包扣除赏金，转账给接单者，并记录交易流水。

1. 交易的流水记录

技术栈：

MySQL：存储交易流水。

Spring Boot：提供流水查询接口。

实现：

用户查看交易流水时，后端从MySQL中查询数据。

1. **通知模块**
2. 消息通知

技术栈：

WebSocket：实现实时消息推送。

Redis：缓存未读消息。

实现：

任务状态变更时，后端通过WebSocket推送消息给用户。

1. 用户间私信

技术栈：

WebSocket：实现实时聊天。

MySQL：存储聊天记录。

实现：

用户发送私信时，后端通过WebSocket推送消息，并存储聊天记录到MySQL。

**5.评价模块**

1. 任务完成后双方互评

技术栈：

Spring Boot：提供评价提交接口。

MySQL：存储评价数据。

实现：

任务完成后，双方提交评价，后端将评价数据存储到MySQL。

2. 信用分计算

技术栈：

Spring Boot：实现信用分计算逻辑。

MySQL：存储信用分。

实现：

根据评价数据计算信用分，并更新到MySQL。

3. 个人评价与信用分展示

技术栈：

Spring Boot：提供查询接口。

MySQL：存储评价和信用分。

实现：

用户查看评价和信用分时，后端从MySQL中查询数据。

1. **管理端模块**

1. 用户管理

技术栈：

Spring Boot：提供用户管理接口。

MySQL：存储用户信息。

实现：

管理员查看和封禁用户时，后端操作MySQL进行数据更新。

2. 任务管理

技术栈：

Spring Boot：提供任务管理接口。

MySQL：存储任务信息。

实现：

管理员审核和删除任务时，后端操作MySQL进行数据更新。

3. 数据统计

技术栈：

Spring Boot：提供统计接口。

MySQL：存储用户、任务和交易数据。

ECharts：前端数据可视化。

实现：

管理员查看数据统计时，后端从MySQL中查询数据，前端使用ECharts展示图表。