

# 项目管理文档

## 一、 成员 & 任务分配

江睿达：后端开发 & 功能测试

刘一苇：前端开发 & 功能测试

王子硕：文档撰写 & 项目管理

## 二、 开发计划安排

Task List 1 (第一阶段目标: 09/22 - 09/23 完成)

1. 项目创建: GitHub 仓库 & 项目初始化
2. 网页搭建: 完成基本页面框架和 UI
3. 多模态输入: 支持文本输入和图片上传
4. API 选择与接入: 敲定 API 服务商并完成接口调用测试
5. 下载功能: 实现 3D 模型下载, 验证文件格式正确性

Task List 2 (第二阶段目标: 09/24 - 09/25 完成)

1. API 集成 (Tripo): 完成任务提交、查询与结果获取, 前端可预览 GLB
2. Prompt 优化: 整理模板 (写实/低模/卡通等), 确定默认参数并生成对比结果
3. 评估系统: 制定项目评估指标和代码规范
4. PRD 定稿: 完善目标、范围、功能清单、流程与非功能指标

Task List 3 (第三阶段目标: 09/26 - 09/27 完成)

1. 模型保存与管理: 实现用户登录、模型保存至个人库、基本的历史列表查看功能
2. 模型纹理输出: 增加贴图生成功能, 前端可选择“下载带纹理”或“下载无纹理”模型。
3. 缓存与调用优化: 减少重复 API 调用。、

Task List 4 (第三阶段目标: 09/28 完成)

1. 文档工作完成
2. 前后端测试
3. Demo 视频录制

### 三、 代码规范

#### 1. 前端

- 命名方式:

普通变量, 函数: 小驼峰

组件、类型、接口: 大驼峰

- 格式:

缩进: 使用 2 个空格。

分号: 在语句末尾总是使用分号。

引号: 统一使用双引号。

换行与末尾逗号: 在长的 import 语句、类型定义和对象中都进行恰当的换行, 并且在多行结构中使用末尾逗号。

- 代码风格:

变量声明: 优先使用 const, 只在变量需要被重新赋值时才使用 let。

模块化: 统一使用 ES Module。

箭头函数: 回调函数中优先使用箭头函数。

异步处理: 优先使用 async/await。

类型:

为变量、函数参数和返回值明确添加类型。

组件 props 优先使用 interface, 其他情况优先使用 type 定义复杂数据结构。

尽量避免使用 any。

- 状态管理:

使用 useState 管理组件内状态。

使用 Zustand 管理跨组件状态, 通过定义 action 操作状态。

样式定义: 使用 TailwindCSS。

#### 2. 后端

- 命名方式

包: 全小写, 点分隔, 反向域名, 如 com.qiniuyun.aibased3dmodelgen;

分层: controller、service、model、config。

类：大驼峰；控制器以 Controller 结尾；服务类 Service / ServiceImpl；配置类 Config；异常类 Exception；枚举类 Enum。

方法/变量：小驼峰，方法表达功能清晰；布尔以 is/has 开头。

常量：全大写+下划线。

- 格式

缩进：4 空格，参数多时换行对齐。

空格：操作符前后加空格。

注解：类 (@Service)、方法 (@Override)、字段 (@Resource)、 Lombok (@Data)。

导入：字母序，静态导入在后，避免 \*。

- 代码风格

异常：自定义 BusinessException，统一 GlobalExceptionHandler，错误码用 ErrorCode。

日志：@Slf4j，关键操作 log.info()、log.error()。

响应：统一 BaseResponse<T>，ResultUtils 构建返回。

数据库：MyBatis-Flex；实体注解 @Table、@Id；服务继承 ServiceImpl。

配置：@Configuration，敏感信息放 .yml 且忽略提交。

文档：Knife4j 生成 API，控制器方法加说明。

依赖注入：优先 @Resource，必需依赖用构造器注入。

注释：类/方法用 JavaDoc，复杂逻辑写“原因”而非“内容”。