

# 项目分工表现（Digital Landfill）

## 1) 角色与职责归属

成员	角色定位	核心职责	关键产出（可验证制品）
A	前端与可视化负责人 (UI/UX + Viz)	主页像素垃圾山、 三大仪表页 (User/Observer/Admin)、 统一样式与组件化	index_v2_corrected.html 像素山主页、personal/observer/admin 三页 mockup、前端组件库（按钮/卡片/ 图表容器）、可截图展示稿
B	后端与 API 负责人 (Flask + RBAC)	路由与认证、数据模型、 REST API、 错误码与分页、导出端点	Flask 项目骨架、/api/auth/* /api/profile/* /api/report/* /api/global/* /api/observer/* /api/admin/* 、API_Documentation.md
C	数据管线与分析负责人 (Data Pipeline + Analytics)	公共数据抓取与归一化、 热/冷计算、匿名聚合、 异常检测、参数管理	抓取与清洗脚本、settings 参数表、热/ 冷比例与时间序列入库、boxplot 统计、 异常日志与全局快照

## 2) 任务分解与完成标准（DoD）

模块	负责人	完成标准（Definition of Done）
像素垃圾山主页	A	热/冷双层像素堆叠；“YOU↘”自动对齐；顶部导航与工具按钮；可 1920×1080 截图清晰
个人档案页（User）	A	DPE 分数、时间线、导出按钮、CSV 上传位；样式与主页一致
观察员页 (Observer)	A、C	全局指数、热/冷双线趋势、粉丝组对比、boxplot 占位；示例数据可渲染
管理台（Admin）	B、A	参数设置、数据操作面板、异常告警、审计日志；路由与 RBAC 可切换视图
认证与 RBAC	B	登录/注销；User/Observer/Admin 权限隔离；未授权返回 401/403
用户报告接口	B	POST /api/report/submit 入库校验；GET /api/report/history 分页返回
全局趋势接口	B、C	GET /api/global/landfill 返回 hot/cold 比例与 YOU 映射；/trends 返回时间序列

模块	负责人	完成标准 (Definition of Done)
数据管线与参数	C	抓取→归一化→入库； settings 中可调能耗与衰减； 异常 spike 识别并写 audit_logs
文档与导出	B、A	API_Documentation.md （统一响应格式+错误码）； 导出 CSV/PDF/PNG 占位端点可返回样例

3) 周度里程碑与可交付物

周次	A（前端）	B（后端）	C（数据）	验收要点
W1	三页静态稿统一风格	Flask 骨架 + RBAC 模型	参数草案 + 字段映射	截图包 + 路由通
W2	User 页接 API 占位	/auth /profile /report	首批样本数据入库	前后端一次联调
W3	主页像素山接 API	/global/landfill /trends	生成 hot/cold 与时间序列	山体随 API 变动
W4	Observer 图表完善	/observer/summary/boxplot	boxplot 分位数与分组	三页可演示
W5	Admin 控制台完善	/admin/settings/users/logs	异常检测入审计	参数改动可见
W6	统一打磨与导出	导出端点与错误码统一	基线数据快照	Demo 彩排 + 文档齐全

4) 个人贡献亮点（对答式要点）

- **A**：像素山算法化堆叠、自动定位“YOU↘”、三页一致的设计系统、低成本可截图呈现。
- **B**：端到端 API 规范与实现、RBAC 权限隔离、统一响应/错误码、导出接口与分页策略。
- **C**：热/冷计算与参数化、外部数据归一化管线、boxplot 统计与异常检测、审计日志闭环。

5) 协作与验收机制

- **接口契约冻结**：W2 末冻结字段名与 JSON 结构；变更需 PR + 版本号。
- **数据契约样例**：每个关键端点 5 条样例（含异常），共享到 examples/ 。
- **可替代性**：三人环境启动脚本与 README，任何成员可一键跑通全栈 Demo。
- **评审清单**：截图（主页 + 三页），API 文档，数据库示意，1-2 分钟演示视频。

## 6) 风险与应对

- 外部源不稳定 → 本地快照与 mock JSON 保持接口不变 (C 提供)。
- 前端性能瓶颈 → 降低像素密度或分区渲染，优先截图交付 (A 调整)。
- 截止前 API 未齐 → B 先给 mock，A 接 mock 后无缝切真实。

## 一句话总结

A 负责「看得见的系统」，B 负责「连起来的系统」，C 负责「算得准的系统」。