**项目中期实验报告**

线上预览：<https://hppnw.github.io/aaaespa/>

**1. 项目概述**

·多媒体静态站点，聚焦 aespa 粉丝生态：专辑、成员、二创互动、大头贴自拍。

·纯前端 + JSON 资源 架构，未来通过 Serverless 注入动态能力。

·已完成三个核心页面（首页 / 专辑 / 成员）与多媒体交互；二创上传 & PhotoBooth 正在设计。

完整代码内容可查看仓库：<https://github.com/hppnw/aaaespa.git>

1.1 代码与目录结构

├─ index.html # 首页

├─ album.html # 专辑展示页

├─ member.html # 个人页

├─ photobooth.html # 大头贴页（预计）

**2. 项目实现情况**

2.1 首页 main.js

|  |  |
| --- | --- |
| 功能 | 关键代码 |
| 动态点击区域 | 根据 members.json 中的 faceArea 坐标，实时计算 <div.click-region> 位置 (setupClickAreas) |
| 成员过渡动画 | animateMemberTransition 抠出头像 → 视差放大 → sessionStorage 传参后跳转 |
| 专辑时间轴 | button.timeline-dot 添加 data-album; 点击后加遮罩 #page-transition-mask → 420 ms 后跳转 |

2.1.1 实现思路

加载 js/data/members.json 后调用 setupClickAreas() 动态计算四人海报成员热区，使用 元素实际尺寸 / 原图尺寸 的比例实现自适应。

点击热区触发 animateMemberTransition(member)：  
• 在离屏 <canvas> 裁剪头像并缓存至 sessionStorage。  
• 创建 blur-mask（背景虚化）与 transition-container（头像缩放）实现 0.8 s Material Motion 转场。  
• 结束后通过 window.location.href 跳转 member.html?member=id。

时间轴 button.timeline-dot 使用 data-album 存储目标专辑。

2.2 专辑页 album.js

·初始化：解析查询参数 album → 目录 assets/albums/{name}。

· CD 动效：<img ids="cd-cover"> + CSS spin；Hover 暂停通过 :hover 修改 animation-play-state。

· MV / Poster 轮播：official.json & tracks.json 数据驱动。

·曲目侧栏播放器：

let player = new Audio();function playTrack(src){player.src = src; player.play();}

切歌时暂停上一曲，维持单例。

·二创模块：galleryVideos 数组解析 fanmedia.json；updateGalleryVideo() 切换 B 站 bvid / iframe 外链。

2.2.1 核心函数与流程

loadAlbumData()：串行 / 并行获取 tracks.json、official.json、gallery.json 后调用多个渲染函数。异常时降级显示 Toast。

renderTracks(tracks)：为每首歌曲生成 <li> + 播放按钮；模块级单例 player 保证同一时刻只有一首歌播放。

renderOfficialMV() / updateMV(index)：根据 mvIndex 更新 <iframe>；通过 bindMVNavEvents() 绑定左右按钮。

renderPosterCarousel()：分页渲染3张海报，保证渐进加载；点击图片进入 showPosterInGallery() 全屏模式。

updateGalleryVideo()：遍历 fanmedia.json，当 type==='bilibili' 时拼接播放器 URL (danmaku=0)；否则直接嵌入外链。

交互：hamburgerBtn 切换 trackSidebar.active；监听 popstate 劫持浏览器返回键回首页，保持单页体验。

2.3 成员页 member.js

2.3.1 交互细节

（1）loadMemberInfo(data,id) 渲染信息面板；若检测到 aespa\_member\_cut 则把头像替换为无圆角版本，延续首页动效。

（2）背景图层 right-bg 使用首张写真 + filter: blur(32px) 制造深度。

（3）相册分页：galleryIndex 控制每页 3 张图片；updateNav() 切换按钮状态，使用 loading="lazy" 减少首屏资源。

（4）Solo MV：mvs 数组 iframe 自动播放，按数组分页，支持上一 / 下一切换，切换前先清空 src 以停止上一视频。

（5）如果首页 transition 存在，则从 sessionStorage 还原裁剪头像，保证视觉连贯。

**3. 当前进展**

|  |  |
| --- | --- |
| 模块 | 状态 |
| 核心静态页面 | 已完成 |
| JSON 数据模型 | 已完成 |
| 音频播放 | 已完成 |
| UI 动效 | 进行中（80%） |
| 构建 & 部署 | GitHub Pages |
| 二创上传 | 需求分析阶段 |
| 大头贴 | UI/技术选型阶段 |

**4. 待开发功能**

4.1 二创专区（用户上传链接）

目标：允许粉丝提交 BiliBili 等外链；后台审核后展示。

方案：

·前端表单 POST /api/fanmedia；校验 URL + 获取 oEmbed info。

Serverless (Firebase Functions) 写入 Firestore fanmedia 集合，字段：url,type,thumb,title,createdAt。

·读取时分页 (orderBy(createdAt) + limit).

·安全：正则过滤 + NoSQL Rule 限制匿名写入频次 (1/min)。

4.2 大头贴 (Photo Booth)

|  |  |  |
| --- | --- | --- |
| 阶段 | 技术 | 说明 |
| 模板选择 | <div class="template-grid"> | 本地模板图列，点击选用 |
| 摄像 | getUserMedia + MediaPipe | 实时抠像合成至 Canvas |
| 装饰 | hammer.js 手势 | 拖拽/缩放贴纸，双指旋转 |
| 导出 | canvas.toBlob | 下载或上传到 Storage |
| 邮件 | nodemailer (Function) | 发送含下载 URL 的邮件 |