

# BoHack黑客松主页PRD

文档版本: V1.0

编写人: BoHack团队

编写日期: 2025年10月4日

## 1. 项目概述

### 1.1 项目背景与目标

#### 背景:

随着技术社区的活跃和创新精神的日益增长，黑客松活动成为激发创意、培养团队协作和解决实际问题的重要平台。为了提升活动组织效率、优化用户体验，并确保信息传递的准确性与及时性，特此规划开发一个专用的黑客松活动网站。

#### 目标:

- 参赛方:** 提供流畅的报名、项目提交和信息获取流程。
- 主办方:** 自动化管理参赛人、项目、游客及嘉宾评委。
- 活动宣传:** 通过活动展示和结果公示，吸引更多参与者和关注者。
- 数据分析:** 收集用户和项目数据，为未来活动提供决策支持。

### 1.2 目标用户（人员身份）

- 游客:** 了解活动详情、日程、赞助商、新闻等。
- 参赛者:** 注册、报名、提交项目、查看个人/团队信息、接收通知。
- 工作人员:** 全面管理网站内容、用户、项目、通知、结果发布等。
- 嘉宾:**
  - 评委:** 查看分配的项目、进行评分、提交评价。
  - 导师 (可选):** 查看项目列表、提供在线指导或联系方式。
  - 赞助商:** 查看赞助权益展示、了解活动影响力。

## 2. 需求分析与功能模块

### 优先级定义:

- **P0 (核心/必须):** 网站上线运营的基础功能，缺失将导致核心业务无法进行
- **P1 (高优先级/重要):** 提升用户体验和管理效率的关键功能，强烈建议在第一版本中实现
- **P2 (中优先级/次要):** 锦上添花的功能，可以提升用户满意度或提供便利，可在后续迭代中实现
- **P3 (低优先级/可选):** 增强型或特定场景功能，可在有充足资源和时间时考虑

模块	功能点	详细描述	优先级	备注
2.1 核心功能模块				
2.1.1 用户与账户管理	注册	支持邮箱注册（需验证码），可考虑集成GitHub/Google等OAuth登录。	P0	用户进入网站的第一步
	登录/登出	用户凭邮箱/密码登录，并支持安全登出。	P0	游客仅能浏览公开信息
	找回密码	通过邮箱验证码或安全问题重置密码。	P1	提升用户体验，防止账户丢失
	个人资料编辑	用户可修改头像、昵称、个人简介、联系方式、职业/学校、技术栈等信息。	P1	完善用户信息，需再商议，可能会难以管理，建议在注册时注明不允许修改信息
	账户安全设置	修改密码、绑定/解绑第三方账户。	P2	增强账户安全性
	用户Dashboard	统一的个人中心，展示报名状态、团队信息、项目提交状态、消息通知等。	P1	提升用户体验，信息汇总入口
2.1.2 活动展示与信息发布	首页	活动Slogan、倒计时、报名入口、重要通知、热门项目/新闻预览。	P0	网站门户，吸引用户

模块	功能点	详细描述	优先级	备注
	关于活动	活动介绍、活动愿景、举办方介绍、历届回顾（若有）。	P1	增加活动吸引力和可信度
	活动日程	详细时间表（开始时间、结束时间、地点、环节名称、简介）。	P0	参赛者和访客了解活动安排的关键
	评委介绍	评委照片、姓名、头衔、简介。	P1	提升活动专业性和影响力
	赞助商展示	赞助商Logo、链接、赞助等级。	P1	回报赞助商，获取活动支持
	常见问题 (FAQ)	常见疑问与解答列表。	P2	减轻组织者客服压力
	新闻与公告	发布活动最新动态、重要通知（支持富文本编辑）。	P0	及时传达关键信息
	联系我们	联系方式、社交媒体链接、在线留言/客服入口。	P1	提供沟通渠道
	资源下载	活动手册、开发工具包、API文档等。	P2	提供额外辅助信息
2.1.3 报名问卷管理	报名入口	统一的报名页面。	P0	用户参与活动的第一步
	报名问卷	包含必填信息（姓名、联系信息（项目链接、是否组队）、协议签署等，后续拟定。	P0	收集参赛者信息，进行筛选与管理
	报名状态查询	用户可在Dashboard查看报名进度和状态（待审核/已通过）。	P1	提供用户反馈，了解报名结果。可由邮箱发送
	报名截止提醒	邮件/站内信提醒用户报名即将截止。	P2	提升用户体验，避免遗漏
2.1.4 项目提交与管理	项目提交表单	包含项目名称、简介、详细描述（富文本）、技术栈、演示视频/GitHub/在线演示链接、项目截图等。	P0	参赛者提交成果的核心途径

模块	功能点	详细描述	优先级	备注
	项目草稿保存	允许用户在正式提交前保存项目草稿，分多次编辑。	P1	提升用户体验，防止信息丢失
	项目编辑/更新	在提交截止日期前，用户可修改已提交的项目信息。	P1	允许用户完善项目内容
	项目列表（管理员/评委视角）	展示所有提交项目，可查看详情，后端导出也可	P0	管理员和评委管理和查看项目的基础
	项目搜索/筛选	根据团队名、项目名、赛道等条件进行搜索和筛选。	P1	方便管理员和评委快速定位项目
2.1.5 评判与结果公示	结果公示	公布获奖团队/项目列表（奖项、团队名、项目名），可链接到项目详情。可配置公示时间和可见性。	P0	活动成果的最终展示，提升活动影响力
2.1.6 后台管理系统	管理员 Dashboard	统计数据概览（注册人数、团队数、项目数、待处理事项）。	P2	管理员快速了解活动概况
	用户管理	查看、编辑、禁用、删除用户，导出用户数据。	P2	管理参赛者和访客信息
	项目管理	查看、编辑、审核、删除项目，导出项目数据，分配项目给评委。	P2	监管和处理所有提交项目
	内容管理	管理新闻、公告、FAQ、评委、导师、赞助商等信息（支持富文本编辑器）。	P2	维护网站内容，确保信息最新和准确
	活动日程管理	新增、编辑、删除活动环节。	P2	灵活调整和发布活动时间表
	评委管理	添加、删除评委，分配项目给评委，查看评委评分进度。	P2	管理评委资源和评判过程
	结果发布管理	设置奖项、配置获奖名单、发布/撤销结果。	P2	控制结果发布的时机和内容
	系统设置	活动时间、报名截止时间、项目提交截止时间、	P2	配置活动参数，提高管理

模块	功能点	详细描述	优先级	备注
		邮件模板、站点信息等。		灵活性
	数据导出	导出用户、团队、项目、评分等数据 (CSV/Excel)。	P2	数据备份和离线分析
	权限管理	设置不同管理员角色的权限。	P0	适用于多管理员协作场景，增强系统安全性
2.2 非功能性需求				
	性能与并发	页面加载速度快，尤其在报名高峰期能处理高并发请求。	P2	保证系统可用性，避免因卡顿造成用户流失
	安全性	用户数据加密存储、防止SQL注入、XSS攻击、CSRF攻击，HTTPS全站加密。	P0	保护用户隐私，维护网站可信度
	可扩展性	模块化设计，方便未来功能迭代和新增。	P1	适应未来业务变化，降低维护成本
	可用性 (UI/UX)	用户界面友好，操作流程简洁明了。	P0	提升用户体验，降低学习成本
	兼容性	支持主流浏览器 (Chrome, Firefox, Edge, Safari) 和移动设备访问。	P0	确保所有用户都能正常访问
	可维护性	代码结构清晰，文档完善，易于维护和二次开发。	P1	降低运营和开发成本
	国际化	支持多语言界面 (可选)。	P3	扩大用户范围 (除非有海外用户需求，否则优先级较低)
	通知机制	邮件通知、站内信通知 (报名成功、团队邀请、项目提交反馈等)。	P1	及时传递重要信息，提高用户参与度

## 3. 产品架构设计

### 3.1 业务流程图

#### 3.1.1 用户注册与报名流程

Error parsing Mermaid diagram!

Cannot read properties of null (reading 'getBoundingClientRect')

#### 3.1.2 团队管理与项目提交流程

Error parsing Mermaid diagram!

Cannot read properties of null (reading 'getBoundingClientRect')

#### 3.1.3 评委评审流程

Error parsing Mermaid diagram!

Cannot read properties of null (reading 'getBoundingClientRect')

..

### 3.2 系统功能模块图

Error parsing Mermaid diagram!

Cannot read properties of null (reading 'getBoundingClientRect')

## 4. 界面原型设计（可参考Adx和其他黑客松）

## 5. 数据接口设计参考 (API)

### 5.1 用户与账户管理

- 用户注册

- 请求: `POST /api/auth/register`
- 参数: `email`, `password`, `confirmPassword`, `captcha`, `agreeTerms`
- 响应: `{"code": 200, "message": "注册成功", "data": {"userId": "..."} }`

- 用户登录

- 请求: `POST /api/auth/login`
- 参数: `email/username`, `password`
- 响应: `{"code": 200, "message": "登录成功", "data": {"token": "...", "userInfo": {...}} }`

- 获取当前用户信息

- 请求: `GET /api/user/profile` (需携带JWT Token)
- 响应: `{"code": 200, "data": {"userId": "...", "name": "...", "email": "...", ...}} }`

- 更新用户信息

- 请求: `PUT /api/user/profile` (需携带JWT Token)
- 参数: `name`, `bio`, `phone`, `avatarUrl`, ...
- 响应: `{"code": 200, "message": "更新成功"} }`

- 发送邮箱验证码

- 请求: `POST /api/auth/send-verification-code`
- 参数: `email`
- 响应: `{"code": 200, "message": "验证码已发送"} }`

### 5.2 活动展示与信息发布

- 获取所有新闻/公告

- 请求: `GET /api/news`
- 响应: `{"code": 200, "data": [{"id": "...", "title": "...", "publishDate": "...", "summary": "..."}, ...]} }`

- 获取特定新闻详情

- 请求: `GET /api/news/{id}`
- 响应: `{"code": 200, "data": {"id": "...", "title": "...", "content": "...", ...}} }`

- 获取活动日程

- 请求: `GET /api/schedule`
- 响应: `{"code": 200, "data": [{"time": "...", "event": "...", "location": "...", ...}, ...]}`

- \* 获取评委列表

- 请求: `GET /api/judges`
- 响应: `{"code": 200, "data": [{"id": "...", "name": "...", "title": "...", "bio": "...", "avatar": "..."}, ...]}`

- \* 获取赞助商列表

- 请求: `GET /api/sponsors`
- 响应: `{"code": 200, "data": [{"id": "...", "name": "...", "logo": "...", "tier": "...", "website": "..."}, ...]}`

## 5.3 报名管理

- 提交报名问卷

- 请求: `POST /api/registration` (需携带JWT Token)
- 参数: `name`, `phone`, `school`, `experience`, `techStack`, `needTeam`, `specialRequest`, `agreeTerms`
- 响应: `{"code": 200, "message": "报名成功"}`

- \* 获取报名状态

- 请求: `GET /api/registration/status` (需携带JWT Token)
- 响应: `{"code": 200, "data": {"status": "approved/pending", "registrationDate": "..."}}`

## 5.4 项目提交与管理

- 提交项目

- 请求: `POST /api/project` (需携带JWT Token)
- 参数: `teamId`, `projectName`, `slogan`, `description`, `techStack`, `demoVideoUrl`, `githubRepoUrl`, `onlineDemoUrl`, `screenshots` (array of urls), `trackId`
- 响应: `{"code": 200, "message": "项目提交成功", "data": {"projectId": "..."}}`

- 获取我的项目列表

- 请求: `GET /api/project/my` (需携带JWT Token)
- 响应: `{"code": 200, "data": [{"projectId": "...", "projectName": "...", "status": "submitted/draft", ...}, ...]}`



- 获取项目详情
  - 请求: `GET /api/project/{projectId}`
  - 响应: `{"code": 200, "data": {"projectId": "...", "projectName": "...", "teamInfo": {...}, "description": "...", ...}}`
- 文件上传 (例如图片上传)
  - 请求: `POST /api/upload/image` (Multipart/form-data)
  - 参数: `file`
  - 响应: `{"code": 200, "data": {"url": "http://cdn.example.com/image.jpg"}}`

## 5.5 评判与结果公示

- \*获取结果公示
  - 请求: `GET /api/results/winners`
  - 响应: `{"code": 200, "data": [{"award": "一等奖", "projectId": "...", "projectName": "...", "teamName": "..."}, ...]}`
- 获取所有项目 (公开展示用, 仅限已公布结果后)
  - 请求: `GET /api/results/all-projects`
  - 响应: `{"code": 200, "data": [{"projectId": "...", "projectName": "...", "teamName": "...", "score": "...", ...}, ...]}`

## 5.6 后台管理系统 (待定)

- 管理员登录
  - 请求: `POST /api/admin/login`
  - 参数: `username`, `password`
  - 响应: `{"code": 200, "data": {"adminToken": "...", "adminInfo": {...}}}`
- 获取所有用户
  - 请求: `GET /api/admin/users` (需携带Admin Token)
  - 参数: `page`, `pageSize`, `search`
- 禁用/启用用户
  - 请求: `PUT /api/admin/user/{userId}/status` (需携带Admin Token)
  - 参数: `status`: "active"/"disabled"
- 导出用户数据
  - 请求: `GET /api/admin/users/export` (需携带Admin Token)

- 管理员发布新闻

- 请求: `POST /api/admin/news` (需携带Admin Token)
- 参数: `title`, `content`, `publishDate`

- 管理员发布结果

- 请求: `POST /api/admin/results/publish` (需携带Admin Token)
- 参数: `awardList`: `[{"award": "一等奖", "projectId": "...", "teamId": "..."}, ...]`

- 管理员分配评委给项目

- 请求: `POST /api/admin/judge/assign` (需携带Admin Token)
- 参数: `judgeId`, `projectIds`: `["id1", "id2", ...]`