

My Day - 任务管理应用

一个现代化的任务管理Web应用，采用极简扁平设计风格，帮助您高效管理日常任务和时间。

核心特性

任务管理

- **任务池管理**：添加、编辑、删除预设任务
- **分类系统**：支持工作、个人、健康、学习、购物、其他等分类
- **优先级设置**：低、中、高、紧急四个级别
- **任务描述**：详细的任务备注功能

今日计划

- **智能规划**：从任务池中选择任务添加到今日计划
- **进度跟踪**：实时显示任务完成状态和进度
- **状态管理**：轻松标记任务完成/未完成
- **进度可视化**：美观的进度条和统计图表

数据分析

- **今日概览**：任务总数、完成数、完成率统计
- **分类统计**：按任务分类显示分布情况
- **实时更新**：数据实时同步更新

数据管理

- **本地存储**：数据安全存储在本地浏览器

- **数据导出：**支持备份所有任务数据
- **数据导入：**从备份文件恢复任务数据
- **清空功能：**一键清空所有数据

设计特色

极简扁平风格

- **现代化UI：**采用Material Design设计理念
- **扁平化设计：**去除多余装饰，专注功能本身
- **配色方案：**清爽的蓝紫色系，护眼舒适
- **图标系统：**统一的Lucide React图标库

流畅动画效果

- **页面切换：**平滑的路由转场动画
- **元素交互：**按钮悬停和点击效果
- **列表展示：**任务卡片的渐入动画
- **状态变更：**进度条和状态的过渡效果

响应式设计

- **多设备适配：**完美支持桌面、平板、手机
- **弹性布局：**自适应不同屏幕尺寸
- **触摸友好：**针对移动设备优化的交互

技术架构

前端技术栈

- **React 18.3**: 现代化前端框架
- **TypeScript**: 类型安全的开发体验
- **TailwindCSS**: 原子化CSS框架
- **React Router**: 客户端路由管理
- **Lucide React**: 优质图标库

开发工具

- **Vite**: 快速构建工具
- **ESLint**: 代码质量检查
- **PostCSS**: CSS处理工具
- **pnpm**: 高效包管理器

数据存储

- **LocalStorage**: 本地浏览器存储
- **JSON格式**: 结构化数据管理
- **实时同步**: 状态自动保存

页面结构

1. 今日概览 (/)

- 显示今日任务统计概览
- 任务完成进度条
- 分类分布统计

- 最近任务列表

2. 任务池 (/pool)

- 管理所有预设任务
- 任务创建和编辑
- 高级筛选功能
- 批量操作

3. 今日计划 (/today)

- 专注管理今日任务
- 任务状态切换
- 进度实时跟踪
- 完成情况总结

4. 设置 (/settings)

- 数据统计信息
- 导出/导入功能
- 应用配置选项
- 使用说明

快速开始

环境要求

- Node.js >= 16.0.0
- pnpm >= 7.0.0

安装依赖

```
cd my-day  
pnpm install
```

开发模式

```
pnpm dev
```

应用将在 `http://localhost:5173` 启动

生产构建

```
pnpm build
```

构建文件将输出到 `dist` 目录

预览构建

```
pnpm preview
```

使用指南

基本操作流程

1. 创建任务

- 进入"任务池"页面
- 点击"新建任务"按钮
- 填写任务信息（标题、描述、分类、优先级）
- 选择是否添加到今日计划
- 保存任务

2. 管理今日计划

- 在任务池中选择任务，添加到今日计划
- 进入"今日计划"页面查看当日任务
- 点击复选框标记任务完成状态
- 查看实时进度统计

3. 任务筛选

- 使用搜索框按标题或描述搜索
- 按分类、优先级、状态筛选
- 清除筛选条件重置视图

4. 数据管理

- 进入"设置"页面
- 导出数据备份到本地文件
- 从备份文件导入数据
- 查看数据统计信息

键盘快捷键





- **Enter**：保存表单
- **Esc**：关闭模态框
- **Space**：切换任务完成状态（聚焦时）

核心功能详解

任务分类系统

- 工作 : 工作相关任务
- 个人 : 个人生活事务
- 健康 : 健身、医疗相关
- 学习 : 学习、培训任务
- 购物 : 购物清单
- 其他 : 其他类型任务

优先级管理

-  紧急: 需要立即处理的任务
-  高: 重要且紧急的任务
-  中: 重要但不紧急的任务
-  低: 可以延后处理的任务

数据安全

- 本地存储: 所有数据存储在用户浏览器本地
- 隐私保护: 不上传到任何服务器
- 数据备份: 支持手动导出备份
- 恢复功能: 可从备份文件恢复数据

自定义配置

主题定制

应用采用TailwindCSS，可以通过修改 `tailwind.config.js` 来自定义主题：

```
// 自定义颜色
colors: {
  primary: {
    50: '#eff6ff',
    500: '#3b82f6',
    700: '#1d4ed8',
  }
}
```

添加新分类

在 `src/types/index.ts` 中扩展 `TaskCategory` 枚举：

```
export enum TaskCategory {
  // ... 现有分类
  CUSTOM = 'custom', // 新分类
}
```


数据格式

任务数据结构

```
interface Task {  
  id: string;           // 唯一标识符  
  title: string;        // 任务标题  
  description?: string; // 任务描述  
  category: TaskCategory; // 任务分类  
  priority: Priority;    // 优先级  
  completed: boolean;    // 完成状态  
  createdAt: Date;       // 创建时间  
  completedAt?: Date;    // 完成时间  
  isToday: boolean;      // 是否在今日计划  
}
```

部署说明

静态文件部署

应用是纯前端SPA，可部署到任何静态文件服务器：

- **Netlify**：拖拽 `dist` 文件夹即可
- **Vercel**：连接Git仓库自动部署
- **GitHub Pages**：使用Actions自动化部署
- **传统服务器**：上传 `dist` 文件夹到web目录

环境变量

应用无需任何环境变量配置，开箱即用。

开发指南

项目结构

```
my-day/
├── src/
│   ├── components/           # 可复用组件
│   │   ├── Layout/          # 布局组件
│   │   ├── Task/            # 任务相关组件
│   │   └── ui/               # 基础UI组件
│   ├── hooks/                # 自定义Hooks
│   ├── lib/                  # 工具函数
│   ├── pages/                # 页面组件
│   ├── types/                # TypeScript类型定义
│   ├── App.tsx               # 主应用组件
│   ├── main.tsx              # 应用入口
│   └── index.css              # 全局样式
├── public/                   # 静态资源
├── dist/                     # 构建输出
└── package.json               # 项目配置
```

代码规范

- 使用TypeScript严格类型检查
- 遵循ESLint规则
- 组件采用函数式组件 + Hooks
- 使用自定义Hooks管理状态
- CSS采用TailwindCSS原子类

性能优化

- 组件懒加载

- 图标按需导入
- 代码分割
- 资源压缩
- 缓存策略

已知问题

1. **间歇性保存问题**：部分用户可能遇到任务创建时的间歇性保存失败，重试通常可以解决
2. **大数据量性能**：当任务数量超过1000个时，可能出现性能下降
3. **IE兼容性**：不支持Internet Explorer浏览器

路线图

即将推出

- ☐ 任务标签系统
- ☐ 任务时间提醒
- ☐ 统计报表导出
- ☐ 主题切换功能
- ☐ 键盘快捷键

未来计划

- ☐ 团队协作功能
- ☐ 云端同步
- ☐ 移动端App
- ☐ API接口
- ☐ 插件系统

贡献指南

欢迎提交Pull Request和Issue!

提交规范

- feat: 新功能
- fix: 修复问题
- docs: 文档更新
- style: 代码格式调整
- refactor: 代码重构
- test: 测试相关
- chore: 构建/工具相关

许可证

MIT License

作者

开发者: Frontend Engineering Expert

致谢

感谢以下开源项目:

- React & React Router
- TailwindCSS
- Lucide React
- Vite
- TypeScript

 让每一天都更高效! 开始使用 My Day 管理您的任务吧!