

## 一.项目简介

本项目是一款面向移动端设计的智能规划助手，旨在将长期习惯目标与每日短周期待办任务进行高效整合。其核心特色在于自动化排程能力，包括内置的遗忘曲线智能复习功能以及任务冲突检测机制。

## 二.技术栈

框架:Vue.js3(GlobalBuild)

样式:TailwindCSS(CDN 版)

后端/数据库:Supabase(用于云端数据同步)

时间处理:dayjs(集成插件：isBetween,weekday,isSameOrBefore 等)

语言环境:简体中文

## 三. 系统类间关系分析

本项目的核心逻辑构建在几个关键实体及其相互作用之上。通过抽象化处理，其对象模型关系如下：

### 1. 核心实体模型

**计划项目(Project):** 作为最高层级的容器，负责定义长期的战略目标（如“考研复习”）。它包含了多个“里程碑”。

**里程碑(Milestone):** 项目的阶段性节点。它并不直接存储任务数据，而是作为一种“逻辑指针”，关联具体的任务或习惯。

**任务(Task):** 短周期的原子单位。具备时间戳、时长和优先级属性。

**习惯(Habit):** 循环周期的行为单位。与任务不同，它强调“连击”和

“频率”，并拥有独立的休息日逻辑。

## 2. 类间关系描述聚合关系：

一个项目可以聚合多个里程碑，里程碑的完成度直接决定项目的总进度。任务可以独立存在，也可以被挂载到某个里程碑下。一个“学习型”任务在完成后，会根据算法自动派生出多个“复习型”子任务，这些子任务在逻辑上指向同一个父级任务 ID。

## 四.核心设计说明

1. 自动冲突检测机制设计为了保证日程的科学性，系统设计了一套基于“时间区间”的碰撞检测逻辑。

设计思路：将每个任务视为时间轴上的一个闭区间  $[Start, Start+Duration]$ 。每当任务被创建、拖拽或修改时长时，系统会遍历当日所有任务序列。若两个区间的交集不为空，则标记为“冲突状态”。在 UI 层级，这种状态会通过视觉反馈（如红色边框）告知用户，但系统不强制禁止，以保留用户调整的灵活性。

2. 遗忘曲线复习算法这是本项目的智能化核心，旨在自动化处理“学习任务”的后续跟进。系统预设了一组基于艾宾浩斯遗忘曲线的时间偏移量（如 1,2,4,7,15 天）。当用户勾选完成一个带有“学习”标签的任务时，系统自动计算未来的复习日期，并批量生成关联任务。这些派生任务拥有特殊的“复习”标记，以便与普通待办区分。

3. 习惯连击与休息日容错逻辑为了解决传统打卡应用过于僵硬的问题，本项目设计了“人性化休息”逻辑。

设计思路：在习惯类中引入 **RestDays** 集合。在计算“当前连续天数”时，系统会检查中断的那一天是否为用户设定的休息日。若是，则跳过该日期继续向前追溯，从而保护用户的“心理连击感”，避免因合理的休息导致激励机制崩溃。

#### 四. 界面布局逻辑设计

1. 时间轴布局算法在日视图中，任务的显示不是简单的列表排布，而是基于“坐标转换”：纵坐标(Y 轴)：将任务开始时间（时:分）映射为相对于顶部的偏移像素。高度(Height)：将任务持续分钟数按比例转换为像素高度。这种设计使用户能够直观看到全天的时间分布空隙。

2. 状态机管理系统采用中心化的状态管理，确保当用户在“周视图”完成一个任务时，“统计图表”和“项目进度条”能够同步更新，避免了数据不一致的情况。