**I. Project Scope & Objectives**

**Core Features**

* **CRUD operations (Create, Read, Update, Delete tasks)**
* **Data persistence (Priority: Browser LocalStorage)**
* **Responsive UI for desktop/mobile**
* **Basic debugging & cross-browser validation**

**Optional Extended Features**

* **Category/Label management**
* **Deadline reminders (Browser notifications)**

**Tech Stack**

* **Core: HTML5/CSS3/JavaScript (ES6+)**
* **Optional Framework: Vue.js or React (Recommended after native JS implementation)**

**II. Project Phases & Tasks**

**Phase 1: Setup & Initialization (0.5 Day)**

1. **Tech Stack Finalization**
   * **Choose framework approach (Native JS/Vue/React)**
   * **Select storage method (LocalStorage vs. SQLite)**
2. **Project Setup**
   * **Create folder structure (index.html, styles/, scripts/)**
   * **Add UI dependencies (e.g., Bootstrap or vanilla CSS)**

**Phase 2: Core Feature Development (3-4 Days)**

**Task List Functionality**

* **UI: Display tasks (title, status, labels, deadlines)**
* **Interactions: Toggle completion, delete/edit tasks  
  Task Form Functionality**
* **UI: Input fields (title, description, labels, deadline)**
* **Logic: Form validation & LocalStorage integration  
  Data Persistence**
* **Implement loadTasks() and saveTasks()**
* **Real-time UI updates via data observers**

**Phase 3: Testing & Optimization (1-2 Days)**

1. **Functional Testing**
   * **Manual CRUD & persistence testing**
   * **Debug using Chrome DevTools (Network/Storage)**
2. **Compatibility Testing**
   * **Verify Chrome/Firefox/Safari support**
3. **Performance Tuning**
   * **Optimize DOM updates (e.g., batch rendering)**

**Phase 4: Extended Features (Optional, 2-3 Days)**

**Label Management**

* **UI: Tag selector (preset + custom tags)**
* **Logic: Filter tasks by labels  
  Deadline Reminders**
* **Logic: Daily deadline checks (e.g., 9:00 AM alerts)**
* **Integration with Notification API (user permission required)**

**III. Timeline**

| **Phase** | **Tasks** | **Duration** |
| --- | --- | --- |
| **1** | **Initialization** | **0.5 Day** |
| **2** | **Core Features** | **3 Days** |
| **3** | **Testing** | **1 Day** |
| **4** | **Extended Features** | **2 Days** |
| **Total: 1-2 weeks (2-3 hours/day)** |  |  |

**IV. Deliverables**

1. **Base Version**
   * **Deployed web app (GitHub Pages)**
   * **Source code repository (with comments)**
2. **Documentation**
   * **README.md: Setup guide & usage instructions**
   * **Technical report: Key challenges & solutions**

**V. Risks & Mitigation**

| **Risk** | **Mitigation** |
| --- | --- |
| **LocalStorage limit (~5MB)** | **Auto-clean old data or switch to IndexedDB** |
| **Browser notification denial** | **Fallback to in-page alerts** |
| **Cross-browser issues** | **Feature detection (e.g., localStorage check)** |

**VI. Future Iterations**

1. **User authentication (Firebase/Supabase)**
2. **Data import/export (JSON files)**
3. **Voice input/hotkey support**

**一、项目范围与目标**

**核心功能**

* **支持任务的增删改查（CRUD）**
* **数据持久化存储（优先使用浏览器 LocalStorage）**
* **响应式 UI 适配桌面与移动端**
* **基础调试与浏览器兼容性验证**

**可选扩展功能（亮点）**

* **任务分类标签管理**
* **截止日期提醒功能（浏览器通知）**

**技术栈**

* **基础：HTML5/CSS3/JavaScript（ES6+）**
* **可选框架：Vue.js 或 React（建议先完成原生 JS 版本后再尝试）**

**二、项目阶段与任务分解**

**阶段 1：项目启动与基础搭建（1 天）**

1. **技术选型**
   * **确定是否使用框架（原生 JS / Vue / React）**
   * **选择数据存储方案（LocalStorage 或 SQLite）**
2. **初始化项目**
   * **创建项目文件夹结构（index.html, styles/, scripts/）**
   * **引入基础依赖（如 Bootstrap 或纯 CSS 实现 UI）**

**阶段 2：核心功能开发（3-4 天）**

**任务列表功能**

* **界面：任务列表展示（标题、状态、分类标签、截止日期）**
* **交互：勾选完成状态、删除任务、编辑任务入口  
  任务表单功能**
* **界面：输入框（标题、描述、分类标签、截止日期）**
* **逻辑：表单提交验证、数据存储到 LocalStorage  
  数据持久化**
* **实现 loadTasks() 和 saveTasks() 方法**
* **监听数据变化实时更新 UI（响应式设计）**

**阶段 3：测试与优化（1-2 天）**

1. **功能测试**
   * **手动测试 CRUD 操作与数据持久化**
   * **使用 Chrome DevTools 调试网络与存储问题**
2. **兼容性测试**
   * **验证主流浏览器（Chrome/Firefox/Safari）支持**
3. **性能优化**
   * **减少 DOM 操作频率（如批量渲染任务列表）**

**阶段 4：扩展功能开发（可选，2-3 天）**

**分类标签功能**

* **界面：标签选择器（预设标签 + 自定义标签）**
* **逻辑：按标签过滤任务列表  
  截止日期提醒**
* **逻辑：定时检查临近任务（如每日 9:00 弹窗提醒）**
* **集成浏览器通知 API（需用户授权）**

**三、时间计划表**

| **阶段** | **任务** | **预计耗时** |
| --- | --- | --- |
| **1** | **项目初始化** | **0.5 天** |
| **2** | **核心功能开发** | **3 天** |
| **3** | **测试与优化** | **1 天** |
| **4** | **扩展功能开发** | **2 天** |
| **总时长：1-2 周（按每天 2-3 小时计）** |  |  |

**四、交付成果**

1. **基础版本**
   * **可运行的 Web 应用（GitHub Pages 部署）**
   * **完整代码仓库（含注释）**
2. **文档**
   * **README.md：项目说明与用户指南**
   * **技术总结报告（关键问题与解决方案）**

**五、风险与应对**

| **风险** | **应对方案** |
| --- | --- |
| **LocalStorage 存储容量限制（约 5MB）** | **定期清理旧数据或切换至 IndexedDB** |
| **浏览器通知权限被拒绝** | **提供备选提醒方式（页面内弹窗）** |
| **跨浏览器兼容性问题** | **使用特性检测（如 localStorage 是否存在）** |

**六、后续迭代建议**

1. **添加用户账户系统（Firebase 或 Supabase）**
2. **支持数据导出/导入（JSON 文件）**
3. **集成语音输入或快捷键操作**