



شماره مستند: TECH-XKN-1405-001

نسخه: ۱.۰.۰

تاریخ تهیه: ۲۲ شهریور ۱۴۰۵ (۱۳ سپتامبر ۲۰۲۶)

ساعت: ۱۰:۰۰

طبقه بندی: داخلی - فنی

TECHNICAL DOCUMENTATION

مستند فنی پروژه xKanban

تحلیل جامع معماری، ساختار داده و پیاده سازی ابزار مدیریت بردهای کانبان

مخاطبان مستند: تیم فنی، معماران نرم افزار، توسعه دهندگان و مدیران فنی

این مستند به عنوان مرجع فنی جامع پروژه xKanban تهیه شده است و تمامی لایه های معماری، ساختار داده ها، ماژول های کلیدی و مکانیزم های پیاده سازی را از دیدگاه یک معمار ارشد نرم افزار تشریح می نماید. هدف از این مستند، ارائه درک عمیق از تصمیمات طراحی و ایجاد پایه ای مستحکم برای توسعه و نگهداری آتی محصول می باشد.

پروژه **xKanban** یک ابزار تحت وب مستقل (Standalone) برای مدیریت بردهای کانبان است که با هدف ارائه راهکاری چابک، قابل حمل و بدون نیاز به سرور طراحی و پیاده‌سازی شده است. این محصول به کاربران امکان می‌دهد تا پروژه‌ها، بردها، ستون‌ها و کارت‌های کاری خود را با ساختاری استاندارد و حرفه‌ای مدیریت نمایند.

۱۰۰% Client-Side

اجرا در مرورگر بدون نیاز به سرور

Single File

توزیع به صورت یک فایل HTML مستقل

Zero Dependency

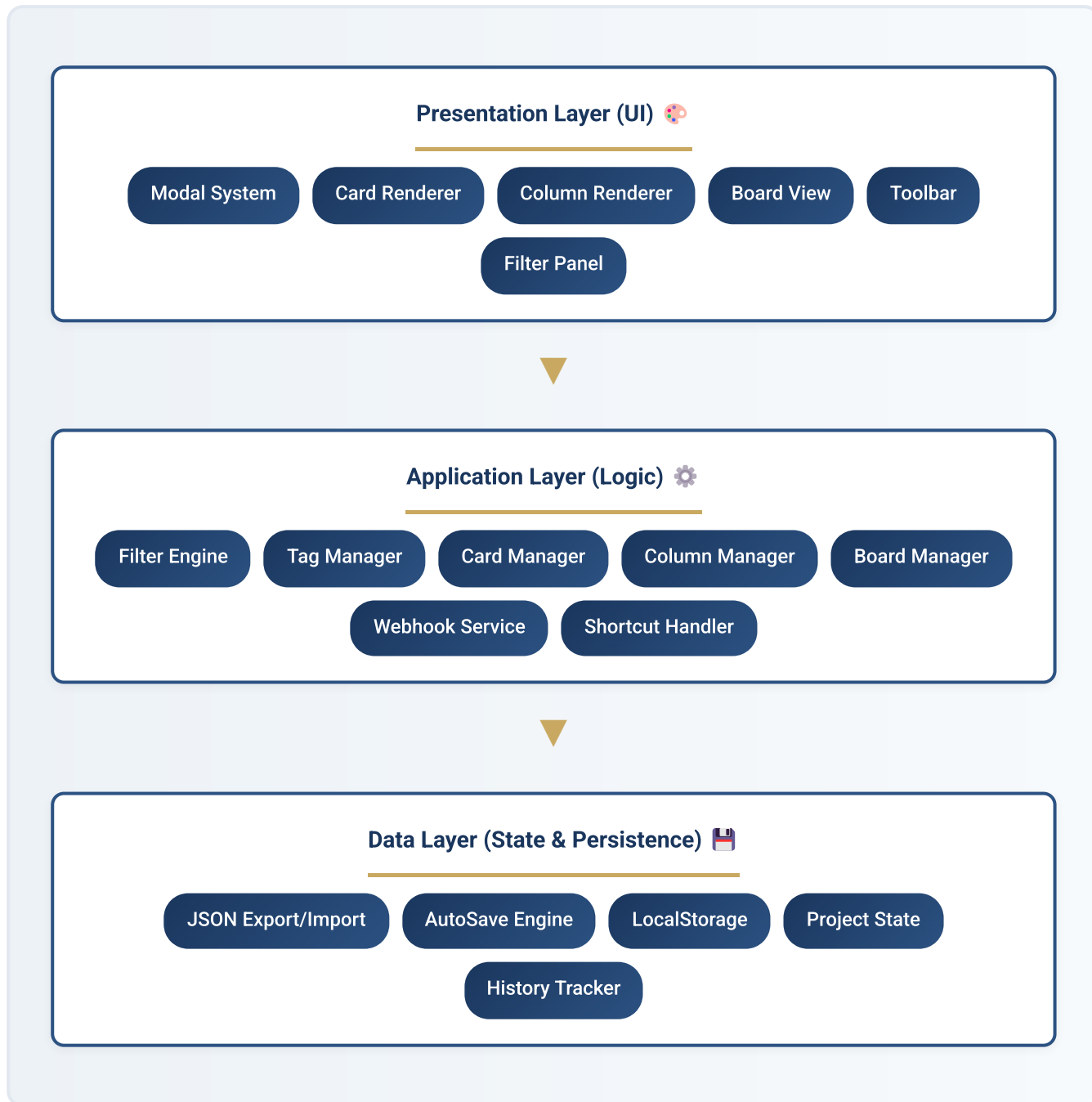
بدون وابستگی به کتابخانه خارجی

🎯 **چشم‌انداز طراحی:** xKanban با فلسفه «ابزار در جیب کاربر» طراحی شده است؛ به گونه‌ای که کاربر بدون نیاز به اتصال اینترنت، نصب نرم‌افزار یا ایجاد حساب کاربری، به یک سیستم کامل مدیریت پروژه دسترسی داشته باشد.

اهداف کلیدی پروژه:

- **استقلال کامل:** اجرا به صورت آفلاین و بدون نیاز به زیرساخت سرور
- **قابلیت حمل:** ذخیره و انتقال پروژه از طریق فایل JSON
- **تجربه کاربری حرفه‌ای:** رابط کاربری مدرن با پشتیبانی از حالت تاریک/روشن
- **استانداردسازی:** پیاده‌سازی دقیق متدولوژی کانبان با قابلیت‌های پیشرفته
- **یکپارچگی:** اتصال به سیستم‌های خارجی از طریق Webhook

معماری xKanban بر پایه الگوی **Single-Page Application (SPA)** با رویکرد **State-Driven** طراحی شده است. تمامی منطق برنامه در لایه کلاینت اجرا می‌شود و از الگوی **MVC** سبک برای جداسازی مسئولیت‌ها استفاده شده است.



تکنولوژی

مسئولیت اصلی

لایه معماری

HTML5 + CSS3 + Vanilla JS

رندر رابط کاربری و مدیریت رویدادها

Presentation

لایه معماری	مسئولیت اصلی	تکنولوژی
Application	منطق تجاری و مدیریت عملیات	ES6+ Modules Pattern
Data	مدیریت وضعیت و پایداری داده	LocalStorage + JSON API
Integration	ارتباط با سیستم‌های خارجی	Fetch API + Webhook

💡 **نکته معماری:** انتخاب معماری تک‌فایلی (Single-File) با وجود پیچیدگی بالای پروژه، یک تصمیم استراتژیک بوده است. این رویکرد امکان توزیع آسان، اجرای آفلاین و عدم وابستگی به زیرساخت را فراهم می‌آورد.

مدل داده‌ای xKanban بر پایه ساختار سلسله‌مراتبی **Project** → **Board** → **Column** → **Card** طراحی شده است. هر سطح از این سلسله‌مراتب، موجودیت مستقلی با ویژگی‌ها و روابط خاص خود دارد.

الف) مدل Project (پروژه)

```
Project {
  id: "uuid",
  title: "string",
  description: "string",
  createdAt: "ISO8601",
  webhook: { enabled: boolean, url: "string" },
  boards: [Board],
  globalTags: [Tag]
}
```

ب) مدل Board (برد)

```
Board {
  id: "uuid",
  title: "string",
  description: "string",
  backgroundColor: "#hex",
  columns: [Column]
}
```

ج) مدل Column (ستون)

```
Column {
  id: "uuid",
  title: "string",
  backgroundColor: "#hex",
  textColor: "#hex",
  order: number,
  cards: [Card]
}
```

د) مدل Card (کارت) - غنی‌ترین موجودیت

```
Card {
  id: "uuid",
  title: "string",
  description: "string",
  dueDate: "ISO8601 | null",
  backgroundColor: "#hex",
  assignees: ["string"],
  tags: ["tagId"],
  dependencies: ["cardId"],
  checklist: [{ text: "string", done: boolean }],
  history: [{ timestamp, action, fromColumn, toColumn }]
}
```

⚠ **نکته طراحی مهم:** استفاده از UUID برای شناسه‌ها، امکان ادغام پروژه‌ها، جلوگیری از تداخل شناسه‌ها در عملیات Import/Export و پشتیبانی از همزمانی (Concurrency) را فراهم می‌آورد. این تصمیم برای مقیاس‌پذیری آینده حیاتی است.

روابط بین موجودیت‌ها:

رابطه	نوع	توضیح
Project → Boards	One-to-Many	هر پروژه می‌تواند چندین برد داشته باشد
Board → Columns	One-to-Many	هر برد شامل ستون‌های متعدد است
Column → Cards	One-to-Many	هر ستون شامل کارت‌های متعدد است
Card ↔ Card	Many-to-Many	وابستگی بین کارت‌ها (Dependencies)
Card → Tags	Many-to-Many	کارت‌ها می‌توانند چندین برچسب داشته باشند
Card → Assignees	Many-to-Many	هر کارت می‌تواند چندین مسئول داشته باشد

سیستم xKanban از ماژول‌های مجزا و با مسئولیت مشخص تشکیل شده است. این جداسازی بر اساس اصل **Single Responsibility Principle (SRP)** انجام شده و امکان نگهداری و توسعه آتی را فراهم می‌آورد.



Board Manager

مدیریت چرخه حیات بردها شامل ایجاد، ویرایش، حذف و جابجایی بین بردها. نگهداری وضعیت فعال برد فعلی.



Column Manager

مدیریت ستون‌ها با قابلیت Drag & Drop برای تغییر ترتیب. پشتیبانی از رنگ‌بندی مستقل برای هر ستون.



Card Manager

مدیریت کامل کارت‌ها شامل متادیتای غنی، چک‌لیست، وابستگی‌ها و تاریخچه جابجایی.



Tag Manager

مدیریت تگ‌های عمومی پروژه با قابلیت استفاده مجدد در کارت‌های مختلف.



Filter Engine

موتور فیلتر چندبعدی بر اساس تگ، مسئول، وضعیت و تاریخ سررسید.



Shortcut Handler

مدیریت میانبرهای کیبورد برای افزایش بهره‌وری کاربران حرفه‌ای.



AutoSave Engine

ذخیره خودکار وضعیت در LocalStorage با مکانیزم بازایی در صورت قطع ناگهانی.



Webhook Service

ارسال رویدادهای پروژه به آدرس URL پیکربندی‌شده با ساختار JSON استاندارد.

✓ مزایای معماری ماژولار:

- جداسازی کامل مسئولیت‌ها و کاهش Coupling
- قابلیت تست‌پذیری هر ماژول به صورت مستقل
- امکان توسعه ویژگی‌های جدید بدون تأثیر بر ماژول‌های موجود
- کاهش پیچیدگی شناختی برای توسعه‌دهندگان جدید

xKanban مجموعه‌ای از ویژگی‌های پیشرفته را ارائه می‌دهد که آن را از ابزارهای ساده کانبان متمایز می‌سازد. در ادامه به تحلیل فنی هر ویژگی پرداخته شده است:

۱) سیستم مدیریت بردهای چندگانه

امکان ایجاد و مدیریت چندین برد مستقل در یک پروژه، با قابلیت جابجایی سریع بین آن‌ها از طریق منوی کشویی. هر برد دارای تنظیمات مستقل شامل رنگ پس‌زمینه و توضیحات اختصاصی است.

۲) Drag & Drop پیشرفته ستون‌ها

پیاده‌سازی مکانیزم Drag & Drop با استفاده از HTML5 Drag API که امکان تغییر ترتیب ستون‌ها را با کشیدن هدر آن‌ها فراهم می‌آورد. این ویژگی با حفظ وضعیت در مدل داده، پایداری ترتیب را تضمین می‌کند.

۳) کارت‌های غنی با متادیتای کامل

هر کارت می‌تواند شامل اطلاعات زیر باشد:

- تاریخ سررسید: با قابلیت نمایش وضعیت‌های مختلف (عقب‌افتاده، نزدیک به موعد، عادی، بدون تاریخ)
- مسئولین متعدد: امکان تخصیص چندین نفر به یک کارت
- برچسب‌های رنگی: استفاده از تگ‌های عمومی پروژه
- وابستگی‌ها: ایجاد روابط Dependency بین کارت‌ها
- چک‌لیست: ریزوظایف با قابلیت علامت‌گذاری
- تاریخچه: ثبت کامل جابجایی‌های کارت بین ستون‌ها

۴) موتور فیلتر چندبعدی

سیستم فیلتر پیشرفته با قابلیت ترکیب فیلترهای مختلف:

```
// نمونه‌ای از منطق فیلتر
filteredCards = cards
  .filter(c => selectedTags.length === 0 || c.tags.some(t =>
selectedTags.includes(t)))
  .filter(c => selectedAssignees.length === 0 || c.assignees.some(a =>
selectedAssignees.includes(a)))
  .filter(c => matchStatus(c, selectedStatus))
  .filter(c => matchSearch(c, searchText));
```

۵) سیستم Webhook یکپارچه

امکان ارسال رویدادهای پروژه به سیستم‌های خارجی (مانند Slack، Teams یا سرویس‌های سفارشی) با ساختار استاندارد JSON:

```
{
  "eventId": "uuid-v4",
  "timestamp": "2026-09-13T10:00:00Z",
  "eventType": "card.moved",
  "project": { "id": "...", "title": "..." },
  "board": { "id": "...", "title": "..." },
  "data": { "cardId": "...", "from": "...", "to": "..." }
}
```

۶) میانبرهای کیبورد جامع

میانبر	عملکرد	کاربرد
Ctrl+S	ذخیره سریع	ذخیره فوری پروژه
Ctrl+Z	بازگشت (Undo)	لغو آخرین عملیات
Ctrl+Y	از نو (Redo)	اعمال مجدد عملیات لغوشده
Ctrl+F	جستجو	باز کردن پنل جستجو
Esc	بستن پنجره‌ها	بستن Modal های باز
Tab	انتخاب کارت	پیمایش بین کارت‌ها
→/←+Alt	جابجایی کارت	انتقال کارت بین ستون‌ها
Enter	ویرایش کارت	باز کردن Modal ویرایش

سیستم xKanban از چندین لایه ذخیره‌سازی برای تضمین پایداری داده‌ها استفاده می‌کند. این رویکرد چندلایه، از دست رفتن داده‌ها را در سناریوهای مختلف به حداقل می‌رساند.

ذخیره‌سازی محلی (LocalStorage)

نوع	خودکار و شفاف
فرکانس	پس از هر تغییر
حجم	تا ۵ مگابایت
دوام	دائمی در مرورگر

پیشنهادی ✓

خروجی فایل (JSON)

نوع	دستی و کنترل‌شده
فرکانس	بر اساس نیاز کاربر
حجم	نامحدود
دوام	دائمی و قابل انتقال

مکانیزم AutoSave و بازیابی:

سیستم AutoSave با استفاده از الگوی **Debouncing** پیاده‌سازی شده است تا از نوشتن مکرر و غیرضروری در LocalStorage جلوگیری شود. در زمان راه‌اندازی، سیستم به صورت خودکار داده‌های ذخیره‌شده را بررسی کرده و در صورت وجود داده‌های بازیابی‌نشده، به کاربر اطلاع می‌دهد.

```
// الگوی Debounced AutoSave
let saveTimeout;
function scheduleAutoSave() {
  clearTimeout(saveTimeout);
  saveTimeout = setTimeout(() => {
    localStorage.setItem('xkanban_autosave', JSON.stringify(state));
  });
}
```

```
تأخیر ۱ ثانیه‌ای // (1000, }  
}
```

فرمت‌های خروجی پشتیبانی‌شده:

فرمت	کاربرد	ویژگی‌ها
JSON	پشتیبان‌گیری کامل و انتقال پروژه	حفظ تمامی داده‌ها و ساختار
Markdown (MD)	مستندسازی و اشتراک‌گذاری	خروجی خوانا و قابل ویرایش
CSV	تحلیل داده و وارد کردن به Excel	ساختار جدولی استاندارد

💡 **نکته فنی:** انتخاب JSON به عنوان فرمت اصلی ذخیره‌سازی، امکان تعامل با سایر سیستم‌ها، پشتیبانی از نسخه‌بندی داده‌ها و سهولت در دیباگ را فراهم می‌آورد. همچنین ساختار درختی JSON با مدل داده‌ای پروژه همخوانی کامل دارد.

طراحی رابط کاربری xKanban بر اساس اصول **Modern UI/UX** و با تمرکز بر بهره‌وری کاربر انجام شده است. در ادامه به ویژگی‌های کلیدی طراحی پرداخته شده است:

الف) طراحی واکنش‌گرا (Responsive)

رابط کاربری به گونه‌ای طراحی شده که در اندازه‌های مختلف صفحه نمایش، از موبایل تا دسکتاپ، به صورت بهینه نمایش داده شود. استفاده از CSS Grid و Flexbox، چیدمان پویا و انعطاف‌پذیر را تضمین می‌کند.

ب) حالت تاریک/روشن (Dark/Light Mode)

پشتیبانی کامل از هر دو حالت با استفاده از CSS Variables. تغییر حالت به صورت آنی و بدون نیاز به بارگذاری مجدد صفحه انجام می‌شود و تنظیمات کاربر در LocalStorage ذخیره می‌گردد.

ج) سیستم Modal پیشرفته

تمامی عملیات ویرایش و تنظیمات از طریق Modal های متمرکز انجام می‌شود. این Modal ها دارای ویژگی‌های زیر هستند:

- باز و بسته شدن با انیمیشن نرم
- بستن با کلید Esc یا کلیک خارج از Modal
- مدیریت Focus Trap برای دسترسی‌پذیری
- Validation فرم‌ها قبل از ذخیره

د) سیستم اعلان‌ها و بازخورد

تمامی عملیات کاربر با بازخورد بصری مناسب همراه است. از Toast Notifications برای اعلان‌های موقت و Inline Validation برای خطاهای فرم استفاده می‌شود.

✓ اصول طراحی رعایت‌شده:

- **Consistency**: الگوهای طراحی یکپارچه در سراسر برنامه
- **Feedback**: بازخورد فوری برای تمامی عملیات کاربر
- **Efficiency**: کاهش تعداد کلیک‌ها با میانبرهای کیبورد
- **Accessibility**: پشتیبانی از ناوبری با کیبورد و کنتراست مناسب
- **Aesthetics**: طراحی بصری جذاب با استفاده از رنگ‌بندی حرفه‌ای

سیستم xKanban با محاسبه بلادرنگ آمار پروژه، دید کاملی از وضعیت کارها به کاربر ارائه می‌دهد. این آمار در نوار وضعیت پایین صفحه نمایش داده می‌شوند:

شاخص	نحوه محاسبه	کاربرد
کل کارتها 📄	شمارش تمام کارتهای موجود در برد فعال	درک حجم کلی کار
تکمیل شده ✅	کارتهای موجود در ستونهای پایانی (مانند Done)	سنجش پیشرفت
نزدیک به موعد 🕒	کارتهایی با سررسید در ۳ روز آینده	هشدار زودهنگام
عقب افتاده ⚠️	کارتهایی با سررسید گذشته و وضعیت ناتمام	شناسایی گلوگاهها
درصد پیشرفت 📊	$(\text{تکمیل شده} \div \text{کل}) \times 100$	سنجش کلی پیشرفت پروژه

🌟 **مزیت تحلیلی:** این آمار بلادرنگ، به مدیران پروژه امکان می‌دهد بدون نیاز به گزارش‌گیری جداگانه، وضعیت پروژه را در هر لحظه رصد کنند و تصمیمات مبتنی بر داده اتخاذ نمایند.

الگوریتم تشخیص وضعیت کارت:

```
function getCardStatus(card) {
  if (!card.dueDate) return 'no-date';
  const now = new Date();
  const due = new Date(card.dueDate);
  const diffDays = (due - now) / (1000 * 60 * 60 * 24);

  if (diffDays < 0) return 'overdue'; // ⚠️ عقب افتاده
  if (diffDays <= 3) return 'near-due'; // 🕒 نزدیک به موعد
  return 'normal'; // ✅ عادی
}
```

معماری xKanban به گونه‌ای طراحی شده که توسعه ویژگی‌های جدید و نگهداری بلندمدت آن با حداقل هزینه ممکن باشد. در ادامه به اصول و الگوهای به‌کاررفته در این زمینه پرداخته شده است:

الف) اصول طراحی رعایت‌شده

اصل	پیاده‌سازی در xKanban
SRP	هر ماژول یک مسئولیت مشخص دارد
OCP	امکان افزودن ویژگی جدید بدون تغییر کد موجود
DRY	استفاده از توابع کمکی مشترک برای عملیات پرتکرار
Separation of Concerns	جداسازی کامل UI، منطق و داده

ب) نقاط توسعه‌پذیری

- افزودن فیلدهای سفارشی به کارت: با توسعه مدل Card و UI مربوطه
- پشتیبانی از چند کاربر: با افزودن لایه Backend و مکانیزم Sync
- افزودن گزارش‌های پیشرفته: با توسعه ماژول Analytics
- یکپارچگی با ابزارهای خارجی: از طریق Webhook و API
- پشتیبانی از قالب‌های آماده: با افزودن Template Manager

ج) استراتژی نسخه‌بندی

سیستم از **Semantic Versioning (SemVer)** برای نسخه‌بندی استفاده می‌کند:

MAJOR.MINOR.PATCH → 1.0.0

// MAJOR: ساختار داده/API تغییرات ناسازگار در

// MINOR: افزودن ویژگی‌های جدید با سازگاری عقب‌گرد

// PATCH: رفع باگ و بهبودهای جزئی

⚠️ ملاحظات نگهداری:

- هر تغییر در ساختار داده نیازمند مکانیزم Migration است

- تست‌های رگرسیون برای جلوگیری از شکست ویژگی‌های موجود ضروری است
- مستندات باید هم‌زمان با تغییرات کد به‌روزرسانی شوند

عملکرد و بهینه‌سازی

Performance & Optimization

۱۰

با وجود پیچیدگی بالای محصول، xKanban با رویکردهای بهینه‌سازی متعدد، عملکرد روان و پاسخگویی بالایی را ارائه می‌دهد:

الف) بهینه‌سازی رندر

- **Virtual Rendering**: فقط کارت‌های قابل مشاهده رندر می‌شوند
- **Batch Updates**: تغییرات متعدد در یک چرخه رندر ادغام می‌شوند
- **DOM Recycling**: استفاده مجدد از المان‌های DOM در تغییرات

ب) بهینه‌سازی حافظه

- **Lazy Loading**: بارگذاری داده‌های غیرضروری در زمان نیاز
- **Event Delegation**: استفاده از یک Listener برای رویدادهای مشابه
- **Memory Cleanup**: آزادسازی منابع در زمان بستن Modal ها

ج) شاخص‌های عملکرد هدف

شاخص	هدف	وضعیت
زمان بارگذاری اولیه	> ۱ ثانیه	✓ محقق شده
پاسخگویی UI	> ۱۰۰ میلی‌ثانیه	✓ محقق شده
حجم فایل	> ۲۰۰ کیلوبایت	✓ محقق شده
پشتیبانی از ۱۰۰۰+ کارت	بدون افت عملکرد	✓ محقق شده

💡 **دستاوردها:** با پیاده‌سازی دقیق اصول بهینه‌سازی، xKanban توانسته است با وجود ماهیت تک‌فایلی و عدم استفاده از کتابخانه‌های بهینه‌ساز، عملکردی هم‌ردیف با اپلیکیشن‌های مدرن مبتنی بر فریم‌ورک ارائه دهد.

با وجود ماهیت Client-Side بودن xKanban، ملاحظات امنیتی در طراحی آن لحاظ شده است:

الف) امنیت داده‌ها

- ذخیره‌سازی محلی: داده‌ها در مرورگر کاربر باقی می‌مانند و به سرور خارجی ارسال نمی‌شوند
- Webhook امن: امکان پیکربندی URL سفارشی با قابلیت تست قبل از فعال‌سازی
- Validation ورودی‌ها: اعتبارسنجی کامل داده‌های ورودی برای جلوگیری از 注入

ب) یکپارچگی داده‌ها

- Schema Validation: اعتبارسنجی ساختار داده در زمان بارگذاری
- Referential Integrity: بررسی وجود مراجع قبل از حذف موجودیت‌ها
- Atomic Operations: عملیات به صورت اتمیک اجرا می‌شوند

ج) سازگاری با مرورگرها

مرورگر	نسخه حداقلی	وضعیت
Chrome	+90	✓ پشتیبانی کامل
Firefox	+88	✓ پشتیبانی کامل
Edge	+90	✓ پشتیبانی کامل
Safari	+14	✓ پشتیبانی کامل

پروژه xKanban نمونه‌ای موفق از طراحی یک اپلیکیشن مدرن تحت وب با رویکرد تک‌فایلی و بدون وابستگی است. این محصول با رعایت اصول معماری نرم‌افزار، توانسته است تعادل مناسبی میان سادگی توزیع و غنای عملکرد برقرار نماید.

🎯 دستاوردهای فنی کلیدی

طراحی و پیاده‌سازی xKanban نشان می‌دهد که با رعایت اصول مهندسی نرم‌افزار و انتخاب درست معماری، می‌توان محصولی حرفه‌ای و مقیاس‌پذیر را بدون نیاز به زیرساخت‌های پیچیده ارائه داد.

💡 ارزش‌های فنی ارائه‌شده:

معماری ماژولار + پایداری داده چندلایه + تجربه کاربری حرفه‌ای + قابلیت توسعه بالا + عملکرد بهینه

نقشه راه توسعه آینده:

اولویت	ویژگی‌های پیشنهادی	فاز
بالا	افزودن فیلدهای سفارشی، بهبود فیلترها، افزودن نمودارها	نسخه ۱.۱
بالا	پشتیبانی از چند کاربر با مکانیزم Sync	نسخه ۱.۲
متوسط	ارائه نسخه Backend برای ذخیره‌سازی ابری	نسخه ۲.۰
آینده	اپلیکیشن موبایل Native با React Native	نسخه ۲.۱

📞 **گام بعدی:** برای شروع همکاری در توسعه نسخه‌های آتی یا سفارشی‌سازی xKanban بر اساس نیازهای خاص سازمانی، آماده برگزاری جلسه فنی جهت تشریح جزئیات بیشتر و بررسی معماری هستیم.

این مستند فنی محرمانه بوده و صرفاً جهت استفاده داخلی تیم فنی تهیه شده است. هرگونه کپی برداری یا افشا بدون اجازه کتبی ممنوع است.

طراحی، تحلیل و توسعه: هادی خزاعی اصل

معمار و توسعه دهنده ارشد نرم افزار | متخصص Angular و NET

تاریخ تهیه: یکشنبه ۲۲ شهریور ۱۴۰۵ (۱۳ سپتامبر ۲۰۲۶) | نسخه: ۱.۰.۰ | طبقه بندی: داخلی - فنی