

PROJECT NEXUS

Personal Project Command Center

STARK · WEBB EDITION

PREPARED FOR	STACK	VERSION	DATE
Personal Use	React · Persistent Storage	2.0	February 2026

A full-featured, visually stunning personal project tracking system built with React and persistent cloud storage. Inspired by the engineering precision of Tony Stark and the multi-dimensional energy of Spider-Verse. Track every project from idea to deployment — with repo links, live deploy status, tech stack badges, document attachments, activity logs, and dual Grid/Kanban views.

TABLE OF CONTENTS

01	FEATURE OVERVIEW	Complete list of all features and their descriptions
02	DATA ARCHITECTURE	Project data model and field definitions
03	COMPONENT BREAKDOWN	React component structure and responsibilities
04	UI DESIGN SYSTEM	Color palette, typography, and visual language
05	KEY INTERACTIONS	Drag & drop, file uploads, activity logging, persistence
06	HOW TO USE IT	Daily workflow guide for maximum productivity
07	FUTURE ROADMAP	Planned features and extension ideas

FEATURE OVERVIEW

CORE PROJECT FIELDS

Project Title & Description	Name and detailed overview of each project.
Status Tracking	Four states: Upcoming, Ongoing, Completed, Paused — each with a unique color and HUD icon.
Priority Levels	Low / Medium / High / Critical — visually coded with color badges on every card.
Deadline Tracking	Date field with auto-calculated days remaining. Turns red when urgent or overdue.
Progress Tracking	Manual slider OR auto-calculated from task completion percentage. Updates live.
Notes Field	Free-form text area for blockers, ideas, and quick reminders.
Tags	Custom keyword tags for filtering and organizing projects into categories.

DEVELOPER FIELDS

Repository Link	GitHub/GitLab URL stored per project. Opens in a new tab directly from the card.
Deployment Status	Three modes: LIVE (with URL link + green pulse dot), NOT DEPLOYED (red badge), or CUSTOM LABEL (e.g., 'Desktop Only', 'Beta Testing').
Tech Stack Badges	30+ preset technologies with color-coded icons. Shown on every card. Searchable selector in modal.
Document Attachments	Upload PDFs, images (PNG/JPG/GIF/WebP), and word docs (DOC/DOCX/TXT/MD). Stored as base64. 4MB per file limit.

PRODUCTIVITY FEATURES

Task Checklist	Add sub-tasks per project. Checking tasks auto-updates progress %. Completed tasks shown with strikethrough.
Activity Log	Automatic changelog per project. Logs: creation, status changes, task completions, doc uploads, and edits.
Grid View	Card-based overview of all projects. Drag cards to trigger kanban reorder.
Kanban Board View	Four columns (one per status). Full drag-and-drop between columns — dropping a card changes its status and logs the change.

View Toggle

Instantly switch between Grid and Kanban with the GRID/KANBAN toggle in the header.

Search & Filter

Search by title, tag, or tech stack. Filter bar for each status. Sort by deadline, priority, or progress.

Stats Dashboard

Live totals: Total · Ongoing · Upcoming · Completed · Overdue. Average progress ring across all projects.

Persistent Storage

All data saved in the browser's persistent storage layer. Survives page reloads and session closes.

DATA ARCHITECTURE

Each project is stored as a JavaScript object with the following schema:

FIELD	TYPE	DESCRIPTION	EXAMPLE
<code>id</code>	string	Unique random ID (8 chars)	a3f2c1d9
<code>title</code>	string	Project name	Portfolio Website
<code>description</code>	string	Detailed overview	Personal site built with...
<code>status</code>	enum	Upcoming Ongoing Completed Paused	Ongoing
<code>priority</code>	enum	Low Medium High Critical	High
<code>progress</code>	number 0-100	Manual % (overridden if tasks exist)	60
<code>tags</code>	string[]	Custom keyword labels	["Personal", "Design"]
<code>techStack</code>	string[]	Tech from 30+ preset list	["React", "Tailwind"]
<code>repoLink</code>	string	GitHub/GitLab URL	https://github.com/...
<code>deployLink</code>	string	Live URL (if deployed)	https://myapp.vercel.app
<code>deployStatus</code>	enum	live not-deployed custom	live
<code>deployLabel</code>	string	Custom label if deployStatus=custom	Desktop Only
<code>docs</code>	Doc[]	Attached files as base64	[{id, name, type, data, ...}]
<code>deadline</code>	string ISO date	Target completion date	2026-03-15
<code>createdAt</code>	string ISO date	Project creation date	2026-01-10
<code>tasks</code>	Task[]	Sub-task checklist items	[{id, text, done}]
<code>notes</code>	string	Freeform text notes	Focus on mobile...
<code>activityLog</code>	Log[]	Auto-generated history entries	[{ts, action}]

STORAGE KEY

All projects are serialized to JSON and stored under the key `stark_webb_tracker_v2` via the `window.storage` API. The storage is loaded on app mount and written after every create, edit, or delete operation. The app also falls back to sample demo projects if no stored data is found.

COMPONENT BREAKDOWN

App	Root	Main application shell. Manages all state (projects, view mode, filters, modal, drag). Handles storage load/save, drag-and-drop coordination, and project CRUD operations.
ProjectCard	Grid	Full-featured card for grid view. Shows: HUD bracket corners, halftone bg, status bar, progress ring, tech badges, tags, deploy badge, repo link, task count, deadline countdown. Draggable.
KanbanColumn	Kanban	One column per status in the Kanban board. Accepts drag-and-drop via HTML5 drag events. Shows project mini-cards. Drop triggers status update + activity log entry.
Modal	Overlay	Full create/edit form with 5 tabbed sections: Overview · Tech · Tasks · Docs · Activity. Handles all field state, file uploads, task management, tech selection, and activity log generation on save.
ProgressRing	UI	SVG ring chart. Takes progress (0-100), size, stroke, and color. Renders with a glow filter and smooth CSS transition. Text label centered.
TechBadge	UI	Colored badge for each tech in the stack. Looks up icon and color from TECH_LIST constant. Supports 'small' prop for compact card display.
DeployBadge	UI	Context-aware deploy status badge. LIVE mode: clickable link with pulsing green dot. NOT DEPLOYED: red badge with X icon. CUSTOM: shows user-typed label.
KanbanCard	Kanban	Compact card for inside kanban columns. Shows title, description snippet, tech badges, deploy badge, repo link, progress ring, and deadline.

UI DESIGN SYSTEM

VISUAL LANGUAGE

The aesthetic is a deliberate fusion of two iconic sci-fi worlds: the holographic HUD precision of Stark Industries and the kinetic halftone energy of Spider-Verse. Every UI decision reinforces this — from the corner bracket overlays on cards to the halftone dot grid in the background.

#070710	VOID BLACK	Primary background
#FF003C	SPIDER RED	Primary accent, status, buttons
#00BFFF	ARC BLUE	Secondary accent, links, tech
#FFD700	STARK GOLD	Completed state, high-value info
#AA55FF	QUANTUM PURPLE	Paused state
#00FF88	LIVE GREEN	Live deployment indicator
#F0EEFF	HOLOGRAM WHITE	Primary text
#0C0C1E	CARD DARK	Card and panel backgrounds

TYPOGRAPHY

Orbitron — Headers · Labels · HUD elements · Navigation · Futuristic geometric sans from Google Fonts

Rajdhani — Body text · Card titles · Form inputs · Technical, compact — perfect readability at small sizes

JetBrains Mono — Code · Data · Timestamps · Developer-native monospace for all data fields

SIGNATURE DESIGN ELEMENTS

- ◆ HUD Corner Brackets — thin L-shaped lines in each card corner, colored to match status
- ◆ Halftone Dot Grid — radial-gradient background pattern using red/blue dots at low opacity
- ◆ Status Color Bar — 2px gradient stripe across the top of every card
- ◆ Scan Line — animated horizontal line sweeps the full page in 8s loop
- ◆ Progress Ring — SVG donut chart with glow filter and smooth CSS transition
- ◆ Pulsing Live Dot — CSS animation on the green dot for live deployments

KEY INTERACTIONS

DRAG & DROP (KANBAN)

- ◆ Each project card has the HTML5 draggable attribute set to true.
- ◆ onDragStart stores the dragged project's ID in component state.
- ◆ Each KanbanColumn listens to onDragOver (preventDefault to allow drop) and onDrop.
- ◆ On drop: the project's status is updated to the target column's status.
- ◆ The status change is auto-logged in the project's activityLog with '(drag)' suffix.
- ◆ Works across all four columns: Upcoming, Ongoing, Completed, Paused.

FILE UPLOADS (DOCS)

- ◆ Click the dashed upload zone or the hidden file input to trigger browser file picker.
- ◆ Accepted types: .pdf, .png, .jpg, .jpeg, .gif, .webp, .doc, .docx, .txt, .md
- ◆ Files are read via the FileReader API as base64 data URLs.
- ◆ Each file is checked for a 4MB size limit before processing.
- ◆ Stored in the project's docs[] array: {id, name, type, size, data, uploadedAt}.
- ◆ VIEW button opens the file in a new tab via an iframe with the base64 data URL.

ACTIVITY LOG GENERATION

- ◆ On every project save, the app compares the new form data to the original project.
- ◆ Detected changes: status, priority, repo link added, deploy status, tasks completed, docs uploaded.
- ◆ A single log entry is created per save, summarizing all changes with ' · ' separator.
- ◆ Activity log entries are displayed in reverse-chronological order in the Activity tab.
- ◆ Each entry shows: action summary + formatted timestamp (Month Day · HH:MM AM/PM).

PERSISTENT STORAGE

- ◆ On app load (useEffect), the app reads from window.storage.get('stark_webb_tracker_v2').
- ◆ If no data found, the app seeds with two demo projects to show the UI in action.
- ◆ After every create/edit/delete, persist() serializes all projects to JSON and calls window.storage.set.
- ◆ The storage write is wrapped in try/catch — UI always updates even if storage fails.

HOW TO USE IT

DAILY WORKFLOW (RECOMMENDED)

■ MORNING

- ◆ Open the tracker. Switch to Kanban view to see everything by status at a glance.
- ◆ Sort Grid by Deadline to identify today's most urgent projects.
- ◆ Drag any 'Upcoming' project that you're starting today into 'Ongoing'.
- ◆ Check the Overdue count in the stats bar — address those first.

■ DURING WORK

- ◆ Click any project card to open it. Go to the Tasks tab and check off completed tasks.
- ◆ Progress updates automatically as tasks are completed — no manual slider needed.
- ◆ Use the Notes field for quick blockers, ideas, or context to pick up later.
- ◆ Upload relevant docs (specs, mockups, research PDFs) to keep everything in one place.

■ EVENING

- ◆ Mark completed projects as 'Completed' — either via drag in Kanban or in the modal.
- ◆ Check the Activity tab on any project to review what you did today.
- ◆ Set or update deadlines for upcoming projects to keep the urgency system accurate.
- ◆ Review the overall progress ring in the stats bar — your daily productivity at a glance.

PRO TIPS

- ◆ Use the 'Paused' status for projects on hold — keeps them visible but out of your active queue.
- ◆ Tech Stack badges are searchable — type 'py' to find Python, 'ts' for TypeScript, etc.
- ◆ The deploy status 'Custom Label' is perfect for mobile apps (App Store), desktop tools, internal tools, or CLI projects.
- ◆ Attach architecture diagrams as images — they'll render inline in the doc viewer.
- ◆ Use Tags like 'Freelance', 'Learning', 'Hackathon' to group projects and filter quickly.
- ◆ Add tasks even for small projects — the progress ring is much more satisfying when auto-calculated.

FUTURE ROADMAP

Ideas and features planned for future versions. These are organized by effort and impact.

PHASE 2 — QUICK WINS

Milestone System	Add milestone checkpoints within a project with their own dates and progress.
Color Themes	Let users switch between Stark Blue, Spidey Red, and custom accent colors.
Export to PDF	One-click project summary PDF for sharing status with teammates or clients.
Keyboard Shortcuts	n = new project, k = kanban view, g = grid view, Esc = close modal.
Project Templates	Save a project as a template (e.g., 'Web App Starter') to reuse task lists and tech stacks.

PHASE 3 — POWER FEATURES

GitHub API Integration	Auto-fetch repo stats (stars, last commit, language) from the GitHub API using the repo link.
Timeline / Gantt View	Third view option — horizontal timeline by deadline across all projects.
Daily Focus Mode	Pin 1-3 projects as 'Today's Mission' shown prominently at the top of the UI.
Recurring Tasks	Mark tasks as recurring (daily/weekly) to auto-reset on a schedule.
Time Tracker	Log time spent per session on a project. Show total hours in the card.

PHASE 4 — AI INTEGRATION

AI Next Steps	Button in modal: 'Suggest next steps' — uses Claude API to analyze project state and recommend actions.
Smart Deadline Estimation	Based on task count and priority, AI suggests a realistic deadline.
Auto-Tag from Description	Paste a project description and AI suggests relevant tags and tech stack.
Weekly Summary	AI-generated weekly digest of progress, overdue items, and upcoming deadlines.