

TaskFlow - Project Overview

TaskFlow is a production-ready Single Page Application (SPA) for task and team management, built entirely client-side using React 18.3.1 and TypeScript. It features an intuitive dashboard, full CRUD task management, team member tracking, dynamic chart visualizations, and robust support for responsive design, theming, and smooth animations. TaskFlow can be deployed as static files on any hosting platform and requires no backend servers.

1. What is TaskFlow?

TaskFlow is a modern, interactive dashboard application that helps users track projects, manage tasks, and collaborate with team members. It is designed for project managers, team leads, and collaborative environments needing lightweight, reliable, and accessible task organization tools without server dependencies.

2. Tech Stack & Architecture

Core Framework & Language

- **React 18.3.1:** Modern UI library with hooks and functional components.
- **TypeScript:** Type-safe JavaScript ensures code quality and tooling support.
- **Vite:** Fast build tool and development server (runs on port 8080).

Styling & UI Libraries

- **Tailwind CSS:** Utility-first CSS with custom design tokens for colors, spacing, and typography.
- **shaden/ui:** 50+ accessible UI components (Button, Dialog, Card, etc.).
- **Framer Motion:** Animation library for smooth, staggered transitions.
- **Lucide React:** Versatile icon library (Menu, Moon, Sun, Trash, etc.).

Routing & State Management

- **React Router v6:** Client-side routing for all main pages (/dashboard, /tasks, /team).
- **TanStack Query:** Configured for server state (not currently used, ready for future API integration).
- **localStorage:** Persistently stores tasks and team member data client-side.

Data Visualization

- **Recharts:** Renders BarChart (tasks by status) and PieChart (tasks by priority).

3. How It Works

Entry Point: `src/main.tsx` mounts the React app, wrapped in all necessary providers for state, notifications, and routing.

App Layout: Defined in `src/App.tsx`, using a layered component hierarchy:

- `QueryClientProvider` (TanStack Query)
- `TooltipProvider` (shadcn)
- Toast & Sonner (notifications)
- `BrowserRouter`
- Main Layout (sidebar, navbar)
- Route definitions for Dashboard, Tasks, Team, NotFound

Sidebar & Navbar: Sidebar navigation adapts for mobile (overlay) and desktop (fixed). Navbar features theme toggling (dark/light).

Pages:

- **Dashboard:** Shows task stats, Bar & Pie charts, animated cards.
- **Tasks:** Search, filter, table view, real-time CRUD operations via `TaskDialog` modals.
- **Team:** Grid cards with avatars, name, role, email, and task count, animated on load.

4. Data Management

Storage Keys (`localStorage`):

- `taskflow_tasks`: List of tasks (id, title, description, priority, status, assignee, dueDate, createdAt)
- `taskflow_team`: Team members (id, name, email, role, avatar)

CRUD Functions in `src/lib/storage.ts`:

- `getTasks()` / `saveTasks()`: Handle all task operations.
- `getTeam()` / `saveTeam()`: Handle team member operations.

Initial Data: 5 demo tasks & 4 demo team members (Alice, Bob, Charlie, Diana).

5. Key Features

Dashboard

- Stat cards: Total, Completed, In Progress, Pending tasks.

- Animated BarChart and PieChart: Tasks by status and priority (Framer Motion).
- Adaptive layout.

Tasks

- Search and filter: Title, description, status, priority.
- Table UI: Edit, delete, sort, modal dialog for creation/editing.
- Real-time updates (instant filtering, sorting).

Team

- Grid layout: Avatar, name, role, email, task count.
- Animated cards, responsive grid (1–4 columns depending on screen size).

Theme System

- Light/Dark toggle (persisted via localStorage, design tokens auto-adapt).

Responsive Design

- Mobile-first approach.
- Sidebar overlays on mobile, fixed on desktop.
- Grid layouts adapt by breakpoints: sm (640px), md (768px), lg (1024px), xl (1280px).

Animations

- Staggered Framer Motion reveals for cards/charts.
- Tailwind animations for interaction feedback.

6. Development Workflow

```
npm install    # Install dependencies
npm run dev    # Start dev server (localhost:8080)
npm run build  # Production build for deployment
npm run preview # Preview production build
```

7. Project Structure

```
src/
├── components/
│   ├── layout/
│   └── tasks/
```

	└─ ui/	# 50+ shadow components
	└─ hooks/	# use-theme, use-toast, use-mobile
	└─ lib/	# storage.ts (data CRUD), utils.ts
	└─ pages/	# Dashboard, Tasks, Team, NotFound
	└─ App.tsx	# Routing and main layout
	└─ main.tsx	# React entry point
	└─ index.css	# Global styles, design tokens

8. Design Patterns

- **Component Composition:** Small, focused, reusable UI components.
- **Custom Hooks:** Isolate logic for theme switching, toast notifications, and mobile detection.
- **Type Safety:** All data models and props typed, ensuring runtime safety.
- **Design System:** Semantic color tokens, typography, and spacing for consistent theme.
- **No Backend:** Uses localStorage, ensuring instant responses, offline capability, and easily deployable static assets.

9. Extensibility

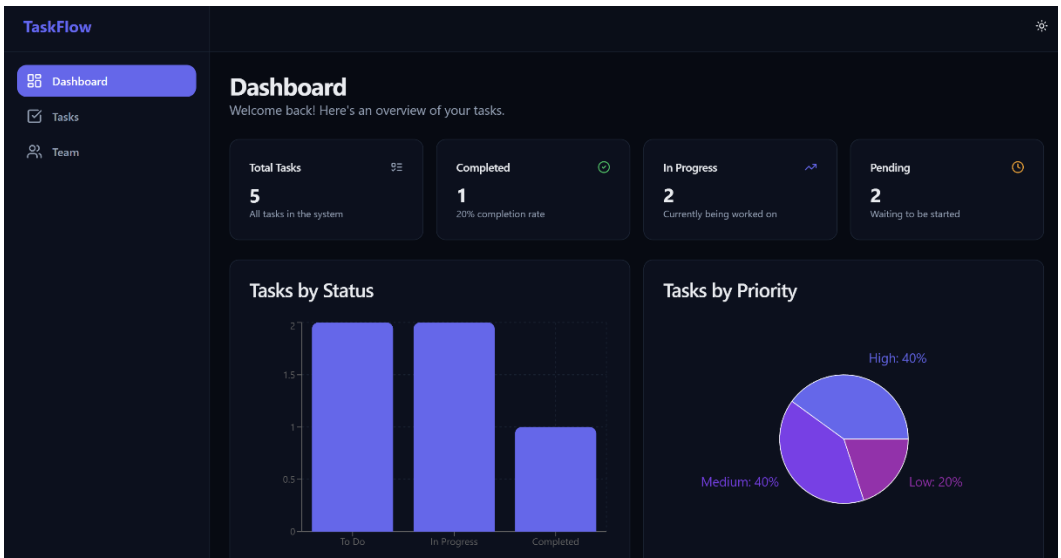
Current strengths:

- Modular, type-safe, easily maintainable.
- Ready for backend integration (future APIs, authentication).
- Advanced UI with animations and charts.

Future enhancements:

- Real API backends for multi-user sync, authentication, and collaboration.
- Project hierarchies, dependencies, notifications, audit logs.
- Export features, time tracking, file attachments.
- Mobile app versions (React Native).

10. Screenshots of Final Project



All Statuses

✓ All Statuses

To Do

In Progress

Completed

All Priorities

✓ All Priorities

High

Medium

Low

Create New Task

Add a new task to your workflow.

Title

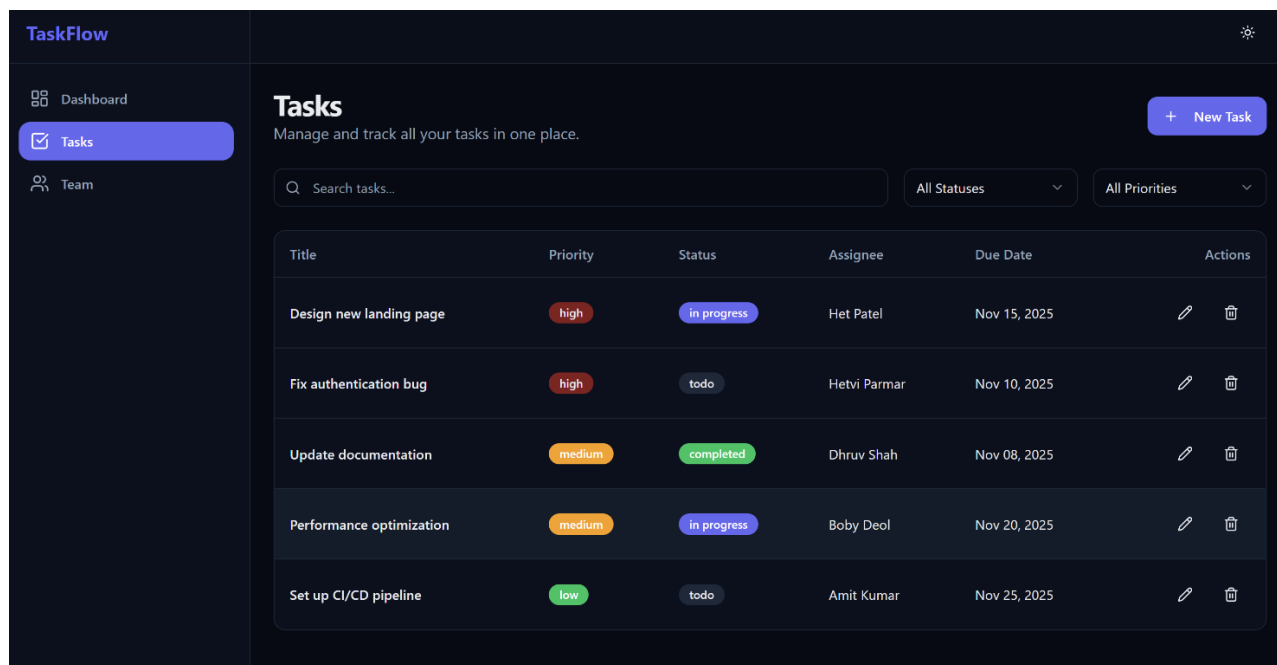
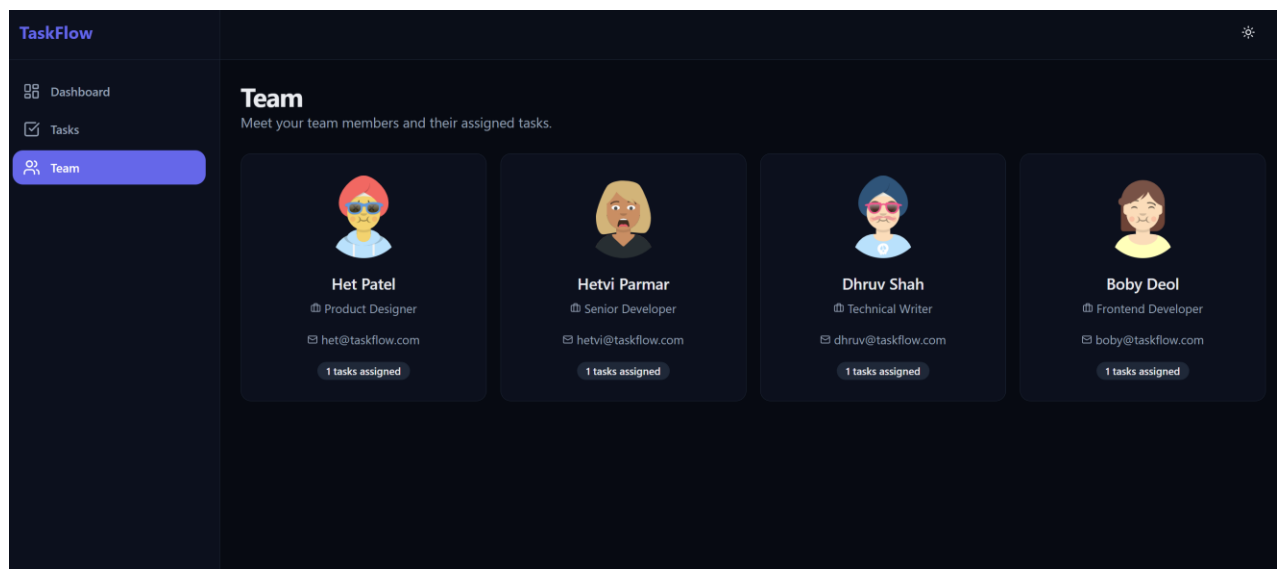
Description

Priority Medium **Status** To Do

Assignee Select a team member

Due Date dd-mm-yyyy

Cancel Create Task



10. Conclusion

TaskFlow exemplifies modern, modular SPA design using the latest technologies for speed, flexibility, and user experience. Its robust architecture, client-side persistence, and rich feature suite make it suitable for both personal productivity and collaborative team management. The application is scalable for future backend integration and extensions.

TaskFlow is a highly adaptable template for project management solutions.