Productivity App - Repo Summary

Generated on 2026-02-19 from repository evidence only

What it is

Produktivitetsplanerare is a local-first web app for weekly planning, built with React, TypeScript, and Vite.
It schedules activity goals into free time around existing bookings and tracks completion status and weekly outcomes.

Who it’s for

- Primary user/persona (explicit): Not found in repo.
- Inferred from README and UI flows: people planning weekly activity goals around calendar commitments.

What it does

- Adds manual calendar bookings and supports recurring events in the weekly calendar.
- Creates activity goals with target minutes, session length, allowed days, and priority.
- Auto-plans weekly sessions using free slots, sleep/work constraints, and conflict reporting.
- Offers "Minimum viable day" fallback with 10-minute mini sessions when weeks are full.
- Imports busy events from iCal (.ics) and deduplicates overlapping duplicates by key.
- Integrates Google Calendar through OAuth and a local backend proxy for secure import.
- Persists state in localStorage and supports JSON export/import backups.

How it works (repo-backed architecture)

- Frontend: React SPA (src/App.tsx) with CalendarGrid, SidePanel, settings/report/detail modals.
- State: useStore (src/store/useStore.ts) centralizes CRUD, planning triggers, and week navigation.
- Planner: planWeek (src/utils/planWeek.ts) computes blocked intervals and places sessions by score.
- Storage: src/utils/storage.ts saves/loads versioned app state in browser localStorage.
- Imports: src/utils/icsImport.ts parses .ics; server/googleCalendarProxy.mjs handles OAuth/API proxy.
- Data flow: UI actions -> useStore -> planner/import helpers -> state -> localStorage -> rendered calendar.

How to run (minimal)

- Install and start frontend: ‘npm install‘ then ‘npm run dev‘.
- Open the printed Vite URL (example: http://localhost:5173).
- Optional Google sync: configure ‘.env.server‘, run ‘npm run dev:server‘, keep frontend running.