Byggplan för Flashcards-App

# 0) Förutsättningar

• Node 20+, npm 10+  
• Git + GitHub/GitLab  
• Docker Desktop (för senare steg)

# 1) Initiera projekt

1. npm create vite@latest flashcards -- --template react-ts  
2. cd flashcards && git init && git add -A && git commit -m "chore: init"  
3. Lägg till grundläggande konfig (.editorconfig, .nvmrc, README.md)  
 - tsconfig.json: aktivera "strict": true, "paths" alias @/\*

# 2) Installera beroenden

npm i zustand localforage zod framer-motion  
npm i -D tailwindcss postcss autoprefixer @types/localforage vite-plugin-pwa \  
 eslint @typescript-eslint/{eslint-plugin,parser} eslint-config-prettier \  
 vitest jsdom @testing-library/{react,user-event} @types/node  
  
Initiera Tailwind: npx tailwindcss init -p och koppla i index.css.  
Lägg PWA senare (steg 10).

# 3) Mappstruktur

src/  
 app/ # layout, routes  
 components/ # UI (Card, Uploader, EditorPanel, Stats)  
 features/  
 decks/ # deckRepo (IndexedDB), deckService, deckStore (Zustand)  
 train/ # repetitionlogik, vyer  
 lib/ # helpers (uuid, sanitize, keyboard)  
 types/ # Deck/Card typer + Zod schema  
 styles/  
 pages/ # Home, Train, Edit, Manage  
 main.tsx, App.tsx  
public/  
 runtime/ # config.json (skapad av docker-entrypoint)

# 4) Typer & schema (validering)

Skapa types/deck.ts med TS-typer och Zod-schema för uppladdad JSON.

# 5) Persistens (IndexedDB)

features/decks/deckRepo.ts med localforage: saveDeck, listDecks, deleteDeck, getDeck.

# 6) Laddning av data

deckService.ts hanterar seed, import från fil, import från URL.

# 7) State (Zustand)

deckStore.ts (activeDeckId, filters), progressStore.ts (bucket, ease, interval, dueAt).

# 8) Kort & flip-UI

Card.tsx med Framer Motion flip, tangentbordsstöd, aria-labels.

# 9) Övningsläge

MVP: 3 buckets (ny, lärs, behärskat). Senare: SM-2.

# 10) Editor

EditorPanel för redigering, autospara, forka seed-lek.

# 11) Filter, sök & vyer

Filter på kategori/taggar, sök, vyer: Bläddra, Öva, Quiz.

# 12) Import/Export

Exportera deck + progress. Import: ersätt eller behåll progress.

# 13) PWA (offline)

vite-plugin-pwa, manifest.json, install prompt i UI.

# 14) Testning

Vitest, React Testing Library, test av logik och UI.

# 15) Kvalitet

ESLint, Prettier, CI med lint/typecheck/test, a11y.

# 16) Bygg för produktion

npm run build, npm run preview.

# 17) Docker

Tvåstegs Dockerfile (build → Nginx). EntryPoint genererar config.json.

# 18) CI/CD

GitHub Actions: lint, test, build, release Docker image.

# 19) Definition of Done (MVP)

[ ] Ladda upp JSON i UI  
[ ] Visa/flip kort  
[ ] IndexedDB  
[ ] 3-bucket repetition  
[ ] Export/import  
[ ] Docker med seed

# 20) Nästa steg (post-MVP)

Multival, statistik, URL-import, Anki/CSV, AI-assist.