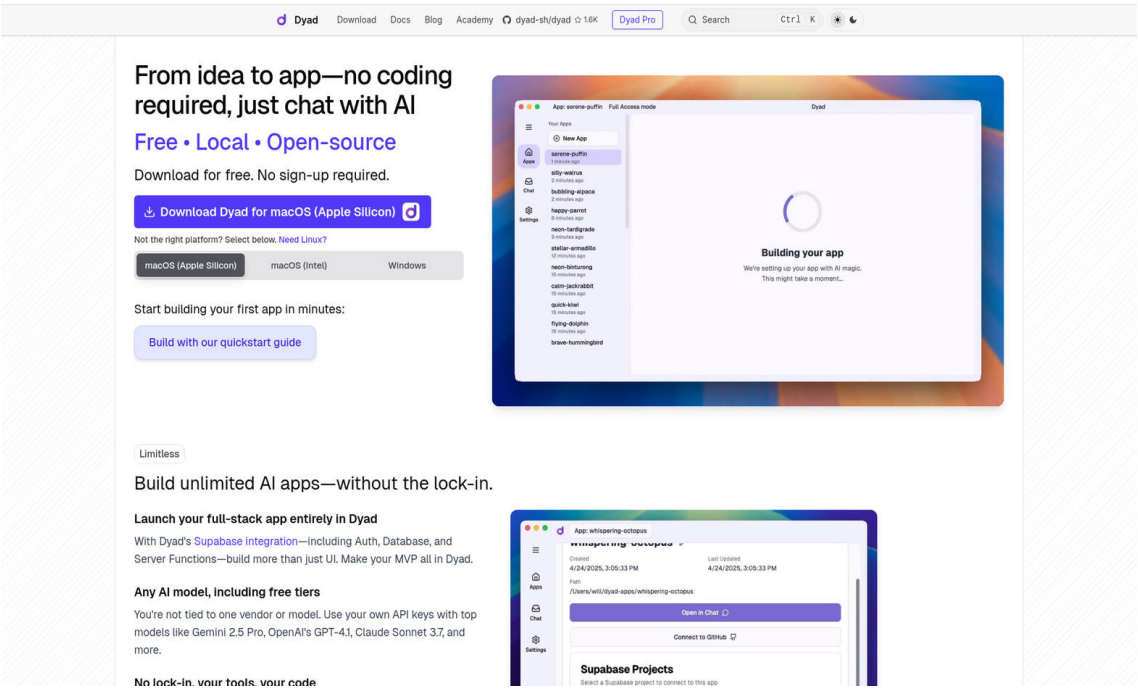

Offizielle Dyad-Oberfläche (Homepage & Editor)

DYAD

Ausführliche Infomappe 2026

Der lokale, open-source AI App Builder für Power-User

Was ist Dyad?
Ein vollständig lokaler, open-source Desktop-App-Builder (macOS & Windows), mit dem du durch einfache Chat-Prompts professionelle Full-Stack-Web-Apps (inkl. Backend, Datenbank, Auth) erstellen kannst – ohne Vendor-Lock-in, ohne monatliche Cloud-Kosten und mit voller Code-Kontrolle. Alternative zu Lovable, v0, Bolt.new & Replit.

Erstellt: Juni 2026 | Basierend auf aktuellen Quellen (dyad.sh, GitHub, Reviews)

Inhaltsverzeichnis

1. Was ist Dyad? – Hintergrund & Philosophie

Dyad (dyad.sh) ist ein lokaler, open-source AI App Builder, der 2025/2026 von Will Chen (ehemals Google/OpenAI) entwickelt wurde und sich schnell zu einem der beliebtesten Tools seiner Art entwickelt hat (über 21.000 GitHub-Sterne). Es positioniert sich bewusst als Gegenentwurf zu cloud-basierten Tools wie Lovable, v0.dev, Bolt.new oder Replit.

Kern-Philosophie: Du sollst dich wie ein Owner fühlen, nicht wie ein Mieter. Deshalb läuft alles lokal auf deinem Rechner, der generierte Code gehört dir vollständig, und es gibt keinerlei Vendor-Lock-in. Du kannst jederzeit exportieren, manuell editieren und mit deinem bevorzugten Stack (VS Code, Cursor etc.) weiterarbeiten.

Wichtige Alleinstellungsmerkmale:

- 100% lokal & privat – Keine Daten verlassen deinen Computer (außer bei gewählten Cloud-Modellen)
- Modell-agnostisch – Nutze Claude 4, GPT-4.1, Gemini, DeepSeek, Qwen oder komplett lokale Modelle via Ollama/LM Studio
- Kein Lock-in – Sauberer, production-ready Code + ein-Klick-Export
- Full-Stack nativ – Supabase-Integration für Datenbank, Authentifizierung und Server-Funktionen direkt im Chat
- Kostenlos & unlimited im Free-Tier (mit eigenem API-Key oder lokalen Modellen)
- Open Source – Transparenz und Community-Driven (43+ Contributors)

Technisch: Electron-Desktop-App (macOS Apple Silicon + Intel, Windows). Der generierte Code basiert typischerweise auf modernen Web-Stacks (React/Next.js-ähnlich, Vite, Drizzle ORM etc.).

2. Was kann Dyad? – Features, Capabilities & Spiele

Dyad ist extrem vielseitig und eignet sich hervorragend für Rapid Prototyping, MVPs, interne Tools, Landing Pages, Dashboards, CRUD-Apps und sogar Web-basierte Spiele.

Bewährte Anwendungsfälle (aus User-Reports & Reviews):

- Komplette Sleep-Tracking-App mit Auth, Datenbank, CRUD, Visualisierung & responsivem Design in ~10 Minuten (Free-Modell DeepSeek)
- Poker-Game – erfolgreich von Usern gebaut und als funktionsfähig bestätigt
- Task-Management-Apps, Maze-Generatoren (für Game Jams), Admin-Dashboards
- Mobile-fähige Apps via PWA oder Capacitor (hybride iOS/Android-Apps)
- Datenintensive Apps mit persistenter Supabase-Datenbank & Auth
- Tools mit natürlicher Sprach-Query-Funktion für Datenmanipulation

Kann Dyad Spiele bauen? Ja – Web-Games sind möglich!

Dyad kann Browser-basierte Spiele (HTML5 Canvas, JavaScript, React-Komponenten) erstellen. Beispiele: Poker, einfache 2D-Spiele, Maze-Generatoren. Für komplexere Spiele (z.B. mit Phaser.js, Three.js oder Multiplayer) musst du ggf. manuell nachhelfen oder spezifische Libraries per Prompt einbinden. Es ist kein dedizierter Game-Engine-Builder wie Unity/Unreal, aber für Indie-Web-Games und Prototypen hervorragend geeignet. Viele User berichten von positiven Ergebnissen bei Game-Jam-ähnlichen Projekten.

Feature-Übersicht:

Feature	Beschreibung	Verfügbarkeit
Lokale Ausführung	Vollständig auf deinem Rechner – keine Cloud-Abhängigkeit	Free + Pro
Open Source	Vollständig einsehbarer Code auf GitHub (21k+ Stars)	Free
Beliebige AI-Modelle	Claude, GPT, Gemini, Ollama, LM Studio, OpenRouter u.v.m.	Free (BYOK)
Full-Stack Apps	Frontend + Backend mit Supabase/Neon (DB, Auth, Functions)	Free + Pro
Echtzeit-Vorschau	Live-Preview während der Code-Generierung	Free
Code-Export & IDE-Integration	Sauberer, editierbarer Code für VS Code etc.	Free
Agent-Modus	Autonomes Debuggen, Kontext-Sammlung, Website-Cloning	Pro (Basic Free limitiert)
Sicherheits-Scan	Prüfung auf SQL-Injection, XSS etc.	Free
Deployment	Ein-Klick zu Vercel, GitHub Pages oder eigenem Hosting	Free
Mobile Support	PWA + Capacitor für hybride Mobile-Apps	Free
MCP Tools	Erweiterbar mit Browser-Tools, Search etc.	Free + Pro

3. Technische Architektur & Modell-Unterstützung

Dyad läuft als Electron-App und generiert moderne Web-Anwendungen. Der Stack ist flexibel, aber typischerweise modern und production-ready.

Unterstützte AI-Modelle & Provider:

- Cloud (empfohlen für Qualität): Anthropic Claude (beste Coding-Qualität), OpenAI GPT-4.1 / o1, Google Gemini (günstig/free Tier)
- Kostenlose Cloud: OpenRouter (DeepSeek, Qwen etc. – oft sehr starke Free-Modelle)
- Komplett lokal & offline: Ollama oder LM Studio (z.B. Qwen 2.5 32B, Llama 3.3, DeepSeek-Coder) – perfekte Privatsphäre, aber hardwareabhängig
- Hybrid: Du kannst pro Projekt oder sogar pro Nachricht das Modell wechseln

Wichtige Integrationen:

- Supabase / Neon: Native, tief integrierte Full-Stack-Unterstützung (Datenbank-Schema per Prompt, Auth, RLS, Edge Functions)
- GitHub: Version Control, Deployment-Workflows
- Vercel: Ein-Klick-Deployment mit Environment-Variablen
- MCP-Server: Erweiterbarkeit durch Tools (z.B. Chrome DevTools, Brave Search, eigene Server)
- Capacitor: Für hybride Mobile-Apps

4. Installation & Erste Schritte

Sehr einfach – keine Anmeldung nötig!

Schritte:

1. Gehe auf https://www.dyad.sh oder GitHub Releases (https://github.com/dyad-sh/dyad/releases)
2. Lade die passende Version herunter (macOS Apple Silicon empfohlen, Windows verfügbar)
3. Installiere & starte die App (Electron-basiert)
4. In den Settings: Füge deinen AI-API-Key hinzu (OpenAI, Anthropic, Google oder OpenRouter) ODER installiere Ollama für lokale Modelle
5. Optional: Supabase-Projekt anlegen und verbinden (für Full-Stack)
6. Neues Projekt starten oder bestehendes importieren
7. Im Chat loslegen: „Erstelle eine Task-Management-App mit Kategorien, Prioritäten und Supabase-Auth“

> Tipp: Für den Einstieg empfehlen viele User Gemini (günstig/free) oder Claude Sonnet/Opus für beste Code-Qualität. Lokale Modelle sind super für sensible Daten oder Offline-Nutzung (z.B. im Flugzeug).

5. Optimale Nutzung – Best Practices & Profi-Tipps

Aus Reviews, AMA mit dem Creator und Community-Erfahrungen:

Prompting-Tipps:

- Sei spezifisch: Statt „Baue eine App“ → „Baue eine moderne Task-App mit Kategorien, Fälligkeitsdaten, Prioritäten, Dark-Mode und Supabase-Persistenz“
- Iteriere schrittweise: Baue Grundgerüst → „Füge Login mit Supabase Auth hinzu“ → „Mache es responsiv für Mobile“ → „Füge Charts mit Recharts hinzu“
- Nutze den Preview: Schau dir jede Änderung sofort an und korrigiere direkt im Chat („Die Buttons sind zu klein, mach sie größer und mit besserem Hover-Effekt“)
- „Fix error with AI“ Button nutzen – Dyad debuggt oft selbstständig sehr gut
- Für komplexe Logik: Beschreibe gewünschtes Verhalten genau oder lass Dyad erst einen Plan machen („Erstelle zuerst einen Architektur-Plan...“)

Workflow-Empfehlungen:

- Starte mit einfachen Projekten, um das Tool kennenzulernen
- Für große Projekte: Nutze Pro-Plan (Smart Context, Turbo Models, bessere Agent-Fähigkeiten)
- Kombiniere AI + manuelles Editing: Lass Dyad 80-90% machen, feile den Rest selbst (oder mit Cursor/Claude Code)
- Nutze GitHub-Integration für Versionierung und Teamarbeit
- Deploy früh zu Vercel, um echte User-Feedback zu bekommen

Hardware-Tipps für lokale Modelle:

Für Ollama/LM Studio brauchst du einen starken Rechner (mind. 16-32 GB RAM, gute GPU mit viel VRAM für große Modelle). Ansonsten: Cloud-Modelle (besonders Gemini oder günstige OpenRouter-Modelle) sind oft die bessere Balance aus Qualität, Geschwindigkeit und Kosten.

6. Preismodelle & Kostenübersicht

Dyad ist eines der fairsten und günstigsten Tools auf dem Markt. Der Free-Tier ist wirklich nutzbar – besonders wenn du eigene API-Keys mitbringst (BYOK).

Plan	Preis	AI-Credits / Monat	Agent-Modus	Extras
Free	0 €	Unlimited (eigener Key) + Basic Agent (5 Msg/Tag)	Basic limitiert	Vollständige Core-Features Lokale Modelle (Ollama)
Pro	20 €/Monat	200 Credits	Voll (Smart Context, Turbo Models)	Dyad Academy Priorisierte Features
Max	79 €/Monat	900 Credits (+ Nachkauf)	Voll + Priority	Office Hours Alle Extras

Real-World Kosten: Viele User kommen mit 5–15 €/Monat für API-Nutzung aus (je nach Nutzung und Modell). Mit lokalen Modellen oder sehr günstigen Free-Tiers (Gemini, OpenRouter) oft nahezu kostenlos. Im Vergleich zu Lovable/Bolt (oft 20–50 €+/Monat bei intensiver Nutzung) spart man massiv.

7. Grenzen & Limitationen

Ehrliche Einschätzung – kein Tool ist perfekt:

- Primär Web-Apps: Sehr gut für moderne Web-Apps. Native Mobile-Apps nur hybrid via Capacitor (kein reines Swift/Kotlin).
- Hardware-Abhängigkeit bei lokalen Modellen: Schwache Rechner → langsame Generierung oder kleinere Modelle nötig.
- Agent-Modus limitiert im Free: Nur 5 Messages/Tag im Basic Agent. Volle Power erst im Pro.
- Sehr große/complexe Codebases: Ohne Pro-Features (Smart Context) kann der Kontext verloren gehen → mehr manuelles Nachhelfen nötig.
- Kein dedizierter Game-Engine: Web-Games ja (Poker, Canvas-Spiele etc.), aber keine AAA- oder komplexen 3D/Physics-Spiele ohne manuelle Integration von Engines wie Phaser/Three.js.
- Qualität hängt vom Prompt & Modell ab: Gute Ergebnisse brauchen gute Beschreibungen. Halluzinationen möglich (wie bei allen AI-Tools) → immer Code reviewen.
- Pro-Code teilweise Fair-Source: Der Core ist Open Source, einige Pro-Features unter Functional Source License (kommerziell eingeschränkt).
- Keine eingebaute Hosting-Infrastruktur: Du deployest selbst (Vercel etc.). Gut für Control, aber extra Schritt.
- Neues Tool: Sehr aktiv entwickelt (v1.3.0 im Juni 2026), aber noch nicht so „polished“ wie einige Cloud-Konkurrenten in allen Edge-Cases.

> Wichtig: Dyad ist kein Ersatz für fundierte Programmierkenntnisse bei komplexen Produktions-Apps. Es ist ein extrem starker Accelerator für Prototyping, Lernen und schnelle MVPs. Der generierte Code sollte immer geprüft und ggf. optimiert werden.

8. Vergleich mit Alternativen

Dyad glänzt besonders bei Privatsphäre, Kosten, Code-Ownership und Modell-Freiheit. Cloud-Tools gewinnen bei Convenience und manchmal bei poliertem UI/Agent-Erfahrung.

Kriterium	Dyad	Lovable / Bolt / v0
Ausführung	Lokal (Electron)	Cloud-basiert
Open Source	Ja (Apache 2.0 + Fair Source)	Nein (proprietär)
Kosten Free Tier	Unlimited (mit eigenem Key) oder komplett kostenlos mit Ollama	Stark limitiert (Credits/Token)
Lock-in	Kein – voller Code-Export	Teilweise / Branding / Public Projects
Datenschutz	Maximal (alles lokal)	Daten in Cloud
Modell-Freiheit	Jedes Modell (inkl. lokal)	Eingeschränkt auf eigene Modelle
Agent-Funktionen	Stark (Pro)	Ja, aber cloud-limitiert
Full-Stack + DB	Exzellent (Supabase native)	Gut, aber oft extra Setup
Performance große Projekte	Gut mit Pro-Features	Abhängig von Token-Limits

Kurz-Empfehlung:

Wähle Dyad, wenn: Du Privatsphäre, volle Code-Kontrolle, niedrige/langfristig günstige Kosten und Open-Source schätzt.
Wähle Lovable/Bolt/v0, wenn: Du maximale Convenience, starke Cloud-Agenten und schnelle polished Results ohne lokales Setup willst (und bereit bist, für Credits zu zahlen).

9. Community, Ressourcen & Weiterentwicklung

- Offizielle Seite: https://www.dyad.sh – Download, Docs, Blog, Academy
- GitHub: https://github.com/dyad-sh/dyad (21k+ Stars, aktiv)
- Reddit Community: r/dyadbuilders – sehr aktiv, Projekte teilen, Hilfe, AMA mit Creator Will
- X/Twitter: @dyad_sh
- Docs & Quickstart: Auf der Website verfügbar (getting-started)
- Releases: Regelmäßig neue Features (Terminal-Integration, MCP OAuth etc. in v1.3.0)

Das Tool wird sehr aktiv weiterentwickelt. Der Creator kommuniziert offen in der Community. Viele Features (z.B. bessere Agenten, Mobile-Support) kamen durch User-Feedback.

10. Fazit & Empfehlungen

Dyad ist aktuell eines der besten Tools für alle, die einen lokalen, kontrollierten und kostengünstigen AI-gestützten App-Builder suchen. Es ist besonders stark für Entwickler, Indie-Hacker, Privacy-Bewusste und alle, die keine monatlichen hohen Abos für Cloud-Tools zahlen wollen.

Stärken-Zusammenfassung:

- Hervorragendes Preis-Leistungs-Verhältnis (oft fast kostenlos)
- Maximale Privatsphäre & Code-Ownership
- Sehr gute Full-Stack-Fähigkeiten mit Supabase
- Flexibel bei AI-Modellen (inkl. komplett lokal)
- Kann echte, deploybare Apps inkl. Spiele-Prototypen bauen
- Starke, wachsende Community

Für wen besonders geeignet?

- Indie-Hacker & Solopreneure
- Entwickler, die schnelle Prototypen wollen
- Privacy-Fokussierte & Offline-Nutzer
- Lernende / Studenten (kostenlos ausprobieren)
- Teams, die Code-Kontrolle behalten wollen
- Web-Game-Entwickler für Browser-Spiele

> Empfehlung: Lade Dyad einfach aus und probiere es aus – es kostet nichts und ist in 5 Minuten einsatzbereit. Starte mit einem kleinen Projekt (z.B. Todo-App oder ein einfaches Spiel). Du wirst schnell merken, ob es zu deinem Workflow passt. Für die meisten Power-User ist es aktuell die beste Balance aus Macht, Freiheit und Kosten.

Quellen & Stand: Offizielle Website dyad.sh, GitHub-Repo, NoCode MBA Review (Mai 2026), OpenReplay Blog, Reddit r/dyadbuilders, diverse YouTube-Vergleiche & Community-Posts (Stand Juni 2026).