Ned - Executive Function Assistant for Students

An AI-powered homework and organization app for students with ADHD/ADD

📋 Project Overview

What it is: Ned helps students (initially Willy, age 13, 7th grade) stay organized with homework, tests, and daily responsibilities.

Why it exists: Traditional executive function tools don't work for kids with ADD. Ned provides automated, contextual support that meets them where they are.

Current Status: MVP deployed and in testing

🎯 Current Features (MVP)

- ✅ Daily homework checklist
- ✅ Weekly calendar view
- ✅ Test tracking and study tips
- ✅ Morning prep checklist (what to pack)
- ✅ Dad jokes + soccer trivia
- ✅ Progress tracking
- ✅ Deployed on Netlify
- ✅ Installed on Willy's iPhone

Live URL: https://polite-dasik-8c85da.netlify.app

🏗️ Tech Stack

Current (MVP)

- **Frontend:** Static HTML/CSS/JavaScript
- **Hosting:** Netlify (free tier)
- **Storage:** Browser localStorage

Phase 2 (In Progress)

- **Frontend:** React or vanilla JS
- **Backend:** Vercel serverless functions
- **Database:** Supabase (PostgreSQL)
- **APIs:** Canvas LMS, Gmail (future), Claude AI (future)

📁 Project Structure

```
ned-project/
├── code/
│   ├── ned-app.html          # Current MVP
│   └── supabase-schema.sql   # Database setup
├── docs/
│   ├── README.md             # This file
│   ├── parking-lot.md        # Feature backlog
│   ├── architecture.md       # System design
│   └── decisions.md          # Technical decisions
├── diagrams/
│   ├── hld-current.mmd       # Current architecture
│   ├── hld-phase2.mmd        # Target architecture
│   └── database-erd.mmd      # Database design
└── credentials/
    └── api-keys.txt          # KEEP SECURE! Git-ignored
```

🚀 Quick Start

View the MVP

1. Open https://polite-dasik-8c85da.netlify.app
2. Add to iPhone home screen for app-like experience

Set Up Development (Phase 2)

1. Clone this repo (if on GitHub)
2. Install dependencies: `npm install`
3. Copy `.env.example` to `.env` and add credentials
4. Run locally: `npm run dev`

📊 Architecture

Current State

```
User's Phone → Netlify (Static HTML) → Browser localStorage
```

Phase 2 Target

```
User's Phone → Frontend (Vercel/Netlify)
                    ↓
              Supabase Database
                    ↑
           Backend (Vercel Functions)
                    ↑
           Canvas API + Gmail API
```

See `/diagrams/` for detailed architecture diagrams.

🗺️ Roadmap

Phase 1: MVP ✅ COMPLETE

- Basic checklist interface
- Manual homework entry
- Deployed and tested

Phase 2: Automation 🚧 IN PROGRESS

- ☐ Canvas API integration (auto-pull homework)
- ☐ Supabase database setup
- ☐ Scheduled syncing (every 3 hours)
- ☐ Voice input for adding tasks

Phase 3: Smart Features 📅 PLANNED

- ☐ Email parsing (school newsletters)
- ☐ Push notifications (morning/evening)
- ☐ AI coaching (Claude API)
- ☐ Pattern detection (forgot lunchbox analysis)

Phase 4: Scale 💡 FUTURE

- ☐ Multi-child support (Simone)
- ☐ Parent dashboard

- □ Multiple LMS platforms (Google Classroom, etc.)
- □ Co-parenting features (meal planning, activities)

See `docs/parking-lot.md` for complete feature list.

👥 Users

Primary User

- **Willy** - 7th grade, ADD, uses Canvas
- Needs: Homework tracking, reminders, executive function support

Secondary User

- **Dad (Peter)** - Divorced, co-parenting, wants to stay involved
- Needs: Visibility into Willy's progress, automated updates

Future Users

- **Simone** - 11, different school (different LMS)
- Other families with ADHD/ADD kids

🔐 Security & Credentials

Stored Securely (Environment Variables)

- Canvas API token
- Supabase credentials (URL, service_role key)
- Claude API key (future)
- Gmail API credentials (future)

Public (OK to expose)

- Supabase anon/public key
- Netlify/Vercel URLs

⚠️ **NEVER commit credentials to Git!**

📝 Key Decisions

Why SQL (Supabase) over NoSQL (Firebase)?

- Better for complex queries (homework patterns, analytics)
- Easier migration to self-hosted if needed
- Standard PostgreSQL (universal developer skill)

Why Vercel for backend?

- Simplest serverless setup
- Great free tier
- Built-in cron jobs for scheduled tasks
- Easy environment variable management

Why build vs. buy?

- Existing tools don't address ADD-specific needs
- Opportunity to learn/build something meaningful
- Potential to help other families

See `docs/decisions.md` for detailed rationale.

🧪 Testing Plan

Week 1 (Nov 11-15)

- Willy tests MVP
- Gather feedback on daily usage
- Identify pain points

Week 2 (Nov 18-22)

- If positive feedback → Build Phase 2
- If mixed → Iterate on MVP
- If negative → Pivot

📚 Documentation

- **Architecture:** `/diagrams/` folder
- **API Docs:** (TBD - Phase 2)
- **Database Schema:** `code/supabase-schema.sql`
- **Feature Backlog:** `docs/parking-lot.md`
- **Meeting Notes:** `docs/meetings/`

🤝 Contributing

Currently a family project. If this grows:

- Code contributions welcome
- Feature suggestions: Open an issue
- Bug reports: Open an issue

📞 Contact

- **Project Lead:** Peter Hammond
- **Primary User:** Willy (testing & feedback)

📄 License

Private project (for now). If open-sourced later, likely MIT License.

🎯 Vision

Short-term: Help Willy stay organized and succeed in school.

Long-term: Build a tool that helps thousands of ADHD/ADD kids develop executive function skills through AI-powered support, reducing parent burden and improving student outcomes.

Last Updated: November 10, 2025
Current Phase: MVP Testing → Phase 2 Planning