

# Grant Master - Complete Build Prompt

Use this prompt to recreate the entire Grant Master SaaS application from scratch.

## System Overview

Build **Grant Master**, a production-ready SaaS platform that helps researchers write, review, and resubmit NIH/NSF/DoD grant applications using AI assistance.

### Tech Stack:

- Frontend: Next.js 14+ (App Router), React 18+, TypeScript, TailwindCSS
- Backend: Next.js API Routes, Neon PostgreSQL (Drizzle ORM)
- AI: OpenAI GPT-4o
- Auth: Custom JWT-based authentication
- Deployment: Vercel + Neon

## Core Features

### 1. Authentication System

- Email/password registration and login
- Password strength indicator (Weak/Medium/Strong with progress bar)
- Real-time email validation with visual feedback
- Forgot password flow with email reset
- JWT tokens stored in HTTP-only cookies
- Route protection middleware for `/dashboard` and `/application/*`

### 2. Dashboard

- List all user's grant applications
- Create new application (select grant type: R01, R21, R03, K-series, etc.)
- Quick access cards to:
  - Writing Assistant
  - Review Simulation
  - CRE Analysis
  - Biosketch Generator

- Resubmission Assistant

### 3. Grant Editor

- Section-based editing for NIH grant structure:
- Specific Aims
- Significance
- Innovation
- Approach
- Timeline
- Budget
- AI Writing Assistant (contextual suggestions)
- Auto-save functionality
- Export to DOCX/PDF

### 4. AI Features

#### 4.1 AI Suggestions ( `/api/ai-suggest` )

- Analyze grant text section by section
- Provide 3-5 actionable improvements
- Focus on clarity, scientific rigor, compliance

#### 4.2 Review Simulation ( `/api/review-simulate` )

- Simulate NIH study section review
- Score on 5 criteria (1-9 scale):
- Significance
- Innovation
- Approach
- Investigators
- Environment
- Generate reviewer-style critiques
- Identify strengths and weaknesses

#### 4.3 CRE Analysis ( `/api/cre-analysis` )

- Check Criterion Review Elements compliance
- Flag missing required components
- Suggest improvements per criterion

## 4.4 Statistical Check ( `/api/statistical-check` )

- Analyze statistical methodology
- Verify power calculations
- Check appropriate test selection

## 5. Biosketch Generator & Verifier

### 5.1 Generator Tab

- Form inputs: Name, eRA Commons ID, Position, Institution, Education, Research Focus, Publications
- AI generates NIH-compliant biosketch
- Copy/Download/Edit options

### 5.2 Upload & Verify Tab ( `/api/biosketch-verify` )

- Upload PDF/DOCX biosketch
- AI checks against NIH requirements:
- Section A: Personal Statement (required)
- Section B: Positions, Scientific Appointments
- Section C: Contributions to Science (max 5, max 4 pubs each)
- Section D: Research Support
- 5-page limit
- No URLs/hyperlinks
- Proper citation format (PMCID/PMID)
- Pass/fail results with actionable suggestions

## 6. Resubmission Assistant (Modular Feature)

**Location:** `/src/features/resubmission/` (isolated, removable module)

### 6.1 Document Upload (Step 1)

- Upload original grant (PDF/DOCX)
- Upload Summary Statement (PDF)
- Input previous score
- Select submission type (A0→A1 or A1→New)
- Warning for A1 "last chance" submissions

### 6.2 Summary Statement Parser (Step 2)

- Auto-extract from PDF:

- Overall impact score
- Individual criterion scores
- Reviewer critiques organized by criterion
- Resume/Synopsis section
- Categorize: "Must Address" vs "Consider"

### **6.3 AI Independent Audit (Step 3)**

- Analyze beyond reviewer comments
- Check against NIH criteria
- Identify gaps reviewers missed
- Flag preliminary data weaknesses
- Suggest NEW data needed

### **6.4 Response Strategy Generator (Step 4)**

- Point-by-point response suggestions
- Priority ranking (Critical/Important/Minor)
- Recommended structural changes
- Page budget calculator

### **6.5 Section Rewriter (Step 5)**

- Side-by-side original vs suggested
- Track changes view
- Page count tracker with warnings
- Address specific critiques

### **6.6 Cover Letter Generator (Step 6)**

- NIH-compliant 1-page format
- Summary of changes
- Point-by-point response to major critiques
- Export as DOCX

### **6.7 Quality Check (Step 7)**

- Simulated re-review scoring
- Improvement delta visualization
- Remaining risk areas
- Pre-submission checklist

**API:** Single endpoint `/api/resubmission` with action parameter

## Disclaimer on ALL outputs:

"This tool provides suggestions only. No guarantee of funding outcomes."

---

## 7. Publications & Manuscript Module (Modular Feature)

**Location:** `/src/features/publications/` (isolated, removable module)

**Entry Points:** Publications can be the FIRST interaction (import existing work) OR the OUTPUT of funded grants.

### 7.1 Import Publications

- Enter DOI → fetch metadata from CrossRef
- Search PubMed by author name → select papers
- Manual entry for preprints/unpublished
- Bulk import support

### 7.2 Research Profile Builder

From imported publications, AI extracts:

- Research themes/expertise areas
- H-index calculation
- Top cited works
- Collaboration network
- Auto-generate Biosketch Sections C & D

### 7.3 Manuscript Assistant (6 Steps)

- **Step 1: Setup** - Link to grant, target journal, co-authors, deadline
- **Step 2: Literature Helper** - Gap finder, citation suggester, novelty check
- **Step 3: IMRAD Builder** - AI assistance for Introduction, Methods, Results, Discussion
- **Step 4: Journal Fit** - Semantic matching to find best-fit journals
- **Step 5: Compliance Check** - Word limits, figure limits, formatting
- **Step 6: Pre-Submission** - Cover letter, highlights, reviewer suggestions

### 7.4 Journal Fit & Intelligence Engine

**Input:**

- Abstract (required)
- Keywords (optional)
- Manuscript metadata: sample size, model systems, validation cohorts, clinical data

**Scoring:**

- Semantic fit score (embedding similarity with journal scope)
- Impact Competitiveness Score per tier (High/Mid/Emerging)
- 4 factors: Dataset Scale, Mechanistic Depth, Validation, Translational Depth

**Output:**

- Top 5 ranked journals with fit score (0-100)
- Impact tier badges
- Competitiveness bands: Prestige Mismatch (0-40), Stretch (41-65), Competitive (66-80), Strongly Aligned (81-100)
- Color-coded indicators (Red/Yellow/Green)
- Similar PubMed articles per journal

**Static Journal Database:** 20-25 curated journals with:

- name, publisher, impact\_factor, impact\_tier, aims\_scope, word\_limit, figure\_limit, open\_access\_fee, submission\_url

**Disclaimer:** "Heuristic competitiveness index. Not acceptance probability."

## 7.5 Peer Review Response (Reuses Resubmission Pattern)

- Upload decision letter + reviewer comments
- AI parses each reviewer's critiques
- Categorize: Major | Minor | Editorial
- Generate point-by-point response draft
- Track manuscript changes
- Generate response letter

## 7.6 Grant-Publication Linker

- Connect publications back to funding source
- Track deliverables (promised vs published)
- Auto-generate acknowledgment text with grant numbers
- Progress reports for renewals

**API:** Single endpoint `/api/publications` with actions:

- "import-doi" | "search-pubmed" | "journal-fit" | "build-profile" | "literature-gaps" | "cover-letter" | "revision-response"
- 

# Production Hardening

## Security

- Input validation and sanitization on all API routes
- Environment variable validation at startup

- Rate limiting on AI endpoints
- CORS configuration

## Error Monitoring

- Sentry integration (client + server)
- Structured error logging

## Legal

- Terms of Service page ( </terms> )
- Privacy Policy page ( </privacy> )
- Cookie consent (if needed)

## Performance

- Lazy loading for feature modules
  - Dynamic imports for heavy components
  - API response timeouts (<10s)
  - Bundle size monitoring
-

# Database Schema (Neon + Drizzle)

```

-- Users
CREATE TABLE users (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  email VARCHAR(255) UNIQUE NOT NULL,
  password_hash VARCHAR(255) NOT NULL,
  name VARCHAR(255),
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

-- Applications
CREATE TABLE applications (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  user_id UUID REFERENCES users(id) ON DELETE CASCADE,
  title VARCHAR(500) NOT NULL,
  grant_type VARCHAR(50) NOT NULL,
  status VARCHAR(50) DEFAULT 'draft',
  content JSONB DEFAULT '{}',
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

-- Resubmissions
CREATE TABLE resubmissions (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  user_id UUID REFERENCES users(id) ON DELETE CASCADE,
  application_id UUID REFERENCES applications(id),
  original_grant_text TEXT,
  summary_statement_text TEXT,
  previous_score DECIMAL(3,1),
  submission_type VARCHAR(20),
  parsed_critiques JSONB,
  audit_results JSONB,
  response_strategy JSONB,
  cover_letter TEXT,
  quality_score JSONB,
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
);

-- Usage Tracking
CREATE TABLE usage_logs (

```

```

    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID REFERENCES users(id),
    action VARCHAR(100) NOT NULL,
    tokens_used INTEGER,
    created_at TIMESTAMP DEFAULT NOW()
);

-- Publications
CREATE TABLE publications (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID REFERENCES users(id) ON DELETE CASCADE,
    pmid VARCHAR(20),
    pmcid VARCHAR(20),
    doi VARCHAR(100) UNIQUE,
    title TEXT,
    authors JSONB,
    journal VARCHAR(255),
    year INTEGER,
    citation_count INTEGER,
    abstract TEXT,
    keywords JSONB,
    research_themes JSONB,
    created_at TIMESTAMP DEFAULT NOW()
);

-- Manuscripts
CREATE TABLE manuscripts (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    user_id UUID REFERENCES users(id) ON DELETE CASCADE,
    grant_id UUID REFERENCES applications(id),
    title VARCHAR(500),
    status VARCHAR(50) DEFAULT 'draft',
    target_journal VARCHAR(255),
    co_authors JSONB,
    content JSONB,
    submission_date DATE,
    created_at TIMESTAMP DEFAULT NOW()
);

-- Grant-Publication Links
CREATE TABLE grant_publications (
    grant_id UUID REFERENCES applications(id),

```

```
publication_id UUID REFERENCES publications(id),
relationship VARCHAR(50),
PRIMARY KEY (grant_id, publication_id)
);
```

---

## Environment Variables

```
# Database
DATABASE_URL=postgresql://...

# Authentication
JWT_SECRET=your-secret-key

# OpenAI
OPENAI_API_KEY=sk-...

# App
NEXT_PUBLIC_APP_URL=https://your-domain.com

# Sentry (optional)
SENTRY_DSN=https://...
NEXT_PUBLIC_SENTRY_DSN=https://...
```

---

# File Structure

```
/src
├─ app/
│   ├─ api/
│   │   ├─ ai-suggest/route.ts
│   │   ├─ auth/[...]/route.ts
│   │   ├─ biosketch-verify/route.ts
│   │   ├─ cre-analysis/route.ts
│   │   ├─ resubmission/route.ts
│   │   ├─ review-simulate/route.ts
│   │   └─ statistical-check/route.ts
│   ├─ application/[id]/page.tsx
│   ├─ dashboard/page.tsx
│   ├─ resubmission/page.tsx
│   ├─ privacy/page.tsx
│   ├─ terms/page.tsx
│   └─ page.tsx (login/register)
├─ components/
│   ├─ AuthForm.tsx
│   ├─ BiosketchGenerator.tsx
│   ├─ Dashboard.tsx
│   ├─ GrantEditor.tsx
│   └─ ...
├─ contexts/
│   └─ AuthContext.tsx
├─ features/
│   ├─ resubmission/
│   │   ├─ components/
│   │   ├─ hooks/
│   │   └─ types.ts
│   └─ publications/
│       ├─ components/
│       ├─ hooks/
│       ├─ data/journals.ts
│       └─ types.ts
├─ lib/
│   ├─ db.ts
│   ├─ env.ts
│   ├─ schema.ts
│   ├─ validate.ts
│   └─ ...
├─ middleware.ts
└─ instrumentation.ts
```

---

# Monetization Ready

## Stripe Integration Points

- Subscription tiers (Free/Pro/Enterprise)
- Usage-based billing (AI tokens)
- Feature gating by plan

## Suggested Pricing

Plan	Price	Features
Free	\$0	3 applications, basic AI (100 requests/mo)
Pro	\$29/mo	Unlimited apps, full AI, resubmission tools
Enterprise	\$99/mo	Team accounts, priority support, API access

---

## Success Criteria

- ☐ User can register, login, reset password
  - ☐ User can create and edit grant applications
  - ☐ AI suggestions work on all sections
  - ☐ Review simulation provides realistic scores
  - ☐ Biosketch generator and verifier functional
  - ☐ Resubmission 7-step wizard complete
  - ☐ Publications import (DOI/PubMed) works
  - ☐ Journal Fit scoring with tier badges
  - ☐ Manuscript wizard functional
  - ☐ Grant-publication linking works
  - ☐ All disclaimers displayed
  - ☐ Route protection working
  - ☐ Error monitoring active
  - ☐ Legal pages present
  - ☐ Mobile responsive
  - ☐ <3s page load time
  - ☐ Zero console errors
-

# Deployment Checklist

1. Push to GitHub
2. Connect Vercel to repository
3. Set environment variables in Vercel
4. Run database migrations in Neon
5. Configure custom domain
6. Enable Sentry alerts
7. Test all flows on production
8. Monitor for 24h before launch

---

**Built with AI assistance. No funding outcomes guaranteed.**