HydraHunt Deployment Guide

🚀 Quick Start (Local Development)

Prerequisites

- Node.js 18+
- npm or bun package manager

Setup Steps

1. **Clone and Install**
   ```bash
   cd hydrahunt
   npm install
   ```

2. **Configure Environment**
   ```bash
   # Copy example env file
   cp .env.example .env

# Edit .env with your values:
# - DATABASE_URL (default SQLite is fine for development)
# - NEXTAUTH_SECRET (generate with: openssl rand -base64 32)
# - ABACUS_API_KEY (for AI features)
```

1. **Initialize Database**
   ```bash
   npx prisma generate
   npx prisma db push
   ```

2. **Run Development Server**
   ```bash
   npm run dev
   ```

3. **Open http://localhost:3000**

🌐 Production Deployment

Option 1: Vercel (Recommended)

1. **Push to GitHub**
   ```bash
   git init
   git add .
   git commit -m "Initial commit"
   git remote add origin <your-repo-url>
   git push -u origin main
   ```

2. **Connect to Vercel**
   - Go to vercel.com (https://vercel.com)
   - Import your GitHub repository
   - Configure environment variables:

Variable	Description
`DATABASE_URL`	PostgreSQL connection string (use Vercel Postgres or Supabase)
`NEXTAUTH_SECRET`	Random 32-char secret
`NEXTAUTH_URL`	Your Vercel app URL (e.g., https://hy-drahunt.vercel.app)
`ABACUS_API_KEY`	Your Abacus.AI API key
`GOOGLE_CLIENT_ID`	(Optional) Google OAuth client ID
`GOOGLE_CLIENT_SECRET`	(Optional) Google OAuth secret

1. **Deploy**
   - Vercel auto-deploys on push to main
   - The `vercel.json` is pre-configured

Option 2: Netlify

1. **Push to GitHub** (same as above)

2. **Connect to Netlify**
   - Go to netlify.com (https://netlify.com)
   - Import your repository
   - Build settings are in `netlify.toml`

3. **Configure Environment Variables**
   - Same variables as Vercel

4. **Deploy**

Option 3: Self-Hosted (Docker)

```
# Dockerfile
FROM node:18-alpine AS base

WORKDIR /app
COPY package*.json ./
RUN npm ci

COPY . .
RUN npx prisma generate
RUN npm run build

EXPOSE 3000
CMD ["npm", "start"]
```

```
# Build and run
docker build -t hydrahunt .
docker run -p 3000:3000 --env-file .env hydrahunt
```

🗄️ Database Setup

For Production (PostgreSQL)

1. **Update Prisma Schema**

   ```
   prisma
     // prisma/schema.prisma
     datasource db {
       provider = "postgresql"
       url      = env("DATABASE_URL")
     }
   ```

2. **Run Migrations**

   ```
   bash
     npx prisma migrate dev --name init
     npx prisma generate
   ```

Recommended Providers

- **Vercel Postgres**: Integrated with Vercel
- **Supabase**: Free tier available
- **PlanetScale**: Serverless MySQL
- **Railway**: PostgreSQL with easy setup

🔐 Authentication Setup

Email/Password

Works out of the box. Users can sign up and login.

Google OAuth (Optional)

1. Go to Google Cloud Console (https://console.cloud.google.com)
2. Create a new project
3. Enable Google+ API
4. Create OAuth 2.0 credentials
5. Add authorized redirect URI: `https://your-domain.com/api/auth/callback/google`
6. Copy Client ID and Secret to environment variables

🤖 AI Features Setup

The platform uses Abacus.AI for:
- Resume parsing

- ATS optimization
- Career transition analysis
- Resume beautification

Get Abacus.AI API Key

1. Sign up at abacus.ai (https://abacus.ai)
2. Create an API key in your dashboard
3. Add to `ABACUS_API_KEY` environment variable

📁 Project Structure

```
hydrahunt/
├── src/
│   ├── app/
│   │   ├── api/            # API routes
│   │   │   ├── auth/       # Authentication
│   │   │   ├── analyze/    # AI analysis endpoints
│   │   │   └── resumes/    # Resume CRUD + upload
│   │   ├── login/          # Login page
│   │   └── page.tsx        # Main app
│   ├── components/         # UI components
│   ├── contexts/           # React contexts
│   ├── hooks/              # Custom hooks
│   ├── lib/                # Utilities (db, auth, ai)
│   └── types/              # TypeScript types
├── prisma/
│   └── schema.prisma       # Database schema
├── vercel.json             # Vercel config
├── netlify.toml            # Netlify config
└── .env.example            # Environment template
```

🔧 API Endpoints

Authentication

- `POST /api/auth/signup` - Create account
- `POST /api/auth/[...nextauth]` - NextAuth.js handlers

Resumes

- `GET /api/resumes?userId=<id>` - List user's resumes
- `POST /api/resumes` - Create resume
- `GET /api/resumes/[id]` - Get single resume
- `PUT /api/resumes/[id]` - Update resume
- `DELETE /api/resumes/[id]` - Delete resume
- `POST /api/resumes/upload` - Upload & parse resume (PDF/DOCX/TXT)
- `GET /api/resumes/[id]/export?template=<name>&format=html` - Export resume

Analysis

- `POST /api/analyze` - Analyze resume (ats/general/beautify/optimize)

- `POST /api/analyze/transition` - Career transition analysis

⚠️ Known Limitations

1. **PDF Export**: Currently generates HTML that can be printed to PDF via browser
2. **File Storage**: Files are parsed but not stored (consider adding S3/Cloudinary)
3. **OAuth**: Google OAuth requires setup; email/password works immediately
4. **SQLite**: Fine for development; use PostgreSQL for production

---

🐛 Troubleshooting

"Prisma Client not generated"

```
npx prisma generate
```

"Database connection failed"

- Check `DATABASE_URL` is correct
- For SQLite, ensure the file path is writable

"AI features not working"

- Verify `ABACUS_API_KEY` is set
- Check API key has proper permissions

"NextAuth errors"

- Ensure `NEXTAUTH_SECRET` is set
- For production, set `NEXTAUTH_URL` to your domain

📞 Support

For issues, please open a GitHub issue with:
- Error message/screenshot
- Steps to reproduce
- Environment (local/Vercel/Netlify)