AI-Powered Web DAW - GitHub Ready Project

This project contains all files for a full-stack AI-powered digital audio workstation (DAW) with frontend (React), backend (Node.js/Express), and Python AI services.

Project Structure

```
ai-daw/
│
├─ frontend/
│  ├─ public/
│  │  └─ index.html
│  ├─ src/
│  │  ├─ components/
│  │  │  ├─ Timeline.tsx
│  │  │  ├─ TrackControls.tsx
│  │  │  ├─ EffectsPanel.tsx
│  │  │  └─ TransportControls.tsx
│  │  ├─ App.tsx
│  │  ├─ index.tsx
│  │  └─ styles.css
│  └─ package.json
│
├─ backend/
│  ├─ controllers/
│  │  └─ audioController.js
│  ├─ routes/
│  │  └─ api.js
│  ├─ services/
│  │  └─ aiProcessing.js
│  ├─ app.js
│  └─ package.json
│
├─ ai-services/
│  ├─ vocal_cleanup.py
│  ├─ beat_generation.py
│  └─ requirements.txt
│
├─ README.md
└─ .gitignore
```

Setup Instructions

Frontend

```
cd frontend
npm install
npm start
```

Backend

```
cd backend
npm install
node app.js
```

AI Services

```
cd ai-services
pip install -r requirements.txt
python vocal_cleanup.py
python beat_generation.py
```

Notes

- The frontend provides timeline editing, track controls, AI effects panel, and transport controls.
- The backend handles audio upload, vocal cleanup, and beat generation requests.
- Python AI services currently have placeholder functionality; replace with Spleeter, TensorFlow/PyTorch models, or custom DSP for production.
- Node.js backend communicates with Python AI services via HTTP requests or child processes.
- Audio files can be uploaded, processed, and returned for playback or export.

This folder is **GitHub-ready**. Just upload the `ai-daw` folder and push to your repository. After that, you can expand AI processing and mastering features to complete your full DAW.