

Disability Hearing Companion — Full-Stack Starter Code

A lightweight, accessible starter you can paste into your project. It includes: - **Backend (Node + Express)**: hearings/cases, evidence uploads, script generator, reminders stub. - **Frontend (React + Vite + Tailwind + shadcn/ui)**: Dashboard, Hearings, Evidence, Scripts Generator, Calendar. - **Keyboard & screen-reader friendly** (semantic HTML, ARIA, focus rings, skip links).

Folder Tree

```
hearing-companion/  
  backend/  
    package.json  
    .env.example  
    src/  
      server.js  
      db.js  
      routes/  
        cases.js  
        hearings.js  
        evidence.js  
        scripts.js  
  frontend/  
    index.html  
    package.json  
    postcss.config.js  
    tailwind.config.js  
    src/  
      main.jsx  
      App.jsx  
      lib/api.js  
      components/  
        Nav.jsx  
        Topbar.jsx  
        Card.jsx  
        DataTable.jsx  
        FileDropzone.jsx  
      pages/  
        Dashboard.jsx  
        Hearings.jsx  
        Evidence.jsx  
        Scripts.jsx
```

Calendar.jsx
styles.css

Backend

backend/package.json

```
{
  "name": "hearing-companion-backend",
  "version": "0.1.0",
  "type": "module",
  "scripts": {
    "dev": "node --watch src/server.js",
    "start": "node src/server.js"
  },
  "dependencies": {
    "cors": "^2.8.5",
    "dotenv": "^16.4.5",
    "express": "^4.19.2",
    "multer": "^1.4.5-lts.1",
    "nanoid": "^5.0.7"
  }
}
```

backend/.env.example

```
PORT=4000
UPLOAD_DIR=./uploads
```

backend/src/db.js

```
import { nanoid } from 'nanoid';

// In-memory data store (swap for SQLite later)
export const db = {
  cases: [],
  hearings: [],
  evidence: [],
  scripts: []
};

export function seed() {
```

```

    const caseId = nanoid();
    db.cases.push({ id: caseId, title: 'SSA Disability Claim', status: 'draft',
claimant: 'You', createdAt: new Date().toISOString() });
    db.hearings.push({ id: nanoid(), caseId, date: new Date(Date.now()
+86400000).toISOString(), location: 'Phone (ALJ)', notes: 'Prep RFC letter;
bring med list', status: 'scheduled' });
}

```

backend/src/routes/cases.js

```

import { Router } from 'express';
import { db } from '../db.js';
import { nanoid } from 'nanoid';

const r = Router();

r.get('/', (req, res) => {
  res.json(db.cases);
});

r.post('/', (req, res) => {
  const { title, claimant } = req.body;
  const item = { id: nanoid(), title, claimant, status: 'draft', createdAt: new
Date().toISOString() };
  db.cases.push(item);
  res.status(201).json(item);
});

r.patch('/:id', (req, res) => {
  const item = db.cases.find(c => c.id === req.params.id);
  if (!item) return res.status(404).json({ error: 'Not found' });
  Object.assign(item, req.body, { updatedAt: new Date().toISOString() });
  res.json(item);
});

export default r;

```

backend/src/routes/hearings.js

```

import { Router } from 'express';
import { db } from '../db.js';
import { nanoid } from 'nanoid';

const r = Router();

```

```

r.get('/', (req, res) => {
  const { caseId } = req.query;
  const rows = caseId ? db.hearings.filter(h => h.caseId === caseId) :
db.hearings;
  res.json(rows);
});

r.post('/', (req, res) => {
  const { caseId, date, location, notes } = req.body;
  const item = { id: nanoid(), caseId, date, location, notes, status:
'scheduled' };
  db.hearings.push(item);
  res.status(201).json(item);
});

r.patch('/:id', (req, res) => {
  const item = db.hearings.find(h => h.id === req.params.id);
  if (!item) return res.status(404).json({ error: 'Not found' });
  Object.assign(item, req.body);
  res.json(item);
});

export default r;

```

backend/src/routes/evidence.js

```

import { Router } from 'express';
import multer from 'multer';
import fs from 'fs';
import path from 'path';
import { fileURLToPath } from 'url';
import { db } from '../db.js';
import { nanoid } from 'nanoid';

const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);

const r = Router();

const uploadDir = process.env.UPLOAD_DIR || path.join(__dirname, '../..//
uploads');
if (!fs.existsSync(uploadDir)) fs.mkdirSync(uploadDir, { recursive: true });

const upload = multer({ dest: uploadDir });

r.get('/', (req, res) => {

```

```

    const { caseId } = req.query;
    const rows = caseId ? db.evidence.filter(e => e.caseId === caseId) :
db.evidence;
    res.json(rows);
  });

  r.post('/', upload.single('file'), (req, res) => {
    const { caseId, tag, notes } = req.body;
    const file = req.file;
    const item = { id: nanoid(), caseId, tag, notes, filename: file?.filename,
originalName: file?.originalname, mimetype: file?.mimetype, size: file?.size,
uploadedAt: new Date().toISOString() };
    db.evidence.push(item);
    res.status(201).json(item);
  });

  export default r;

```

backend/src/routes/scripts.js

```

import { Router } from 'express';
import { nanoid } from 'nanoid';
import { db } from '../db.js';

const TONES = {
  calm: (msg) => `Calm/Factual:\n${msg}`,
  firm: (msg) => `Firm/Boundary:\n${msg}`,
  warm: (msg) => `Warm/Relational:\n${msg}`
};

// basic generator using prompt parts (you can wire to your AI later)
function generateScript({ audience = 'Disability Hearing', need = 'State daily
limits and safety risks', tone = 'calm' }) {
  const base = `For $
{audience}: I need to be clear that due to my conditions, I cannot work reliably
or safely. ${need}. Please record this in the file.`;
  return TONES[tone]?.(base) || base;
}

const r = Router();

r.post('/generate', (req, res) => {
  const { audience, need, tone } = req.body;
  const text = generateScript({ audience, need, tone });
  const item = { id: nanoid(), audience, need, tone, text, createdAt: new
Date().toISOString() };

```

```
    db.scripts.push(item);
    res.status(201).json(item);
  });

  r.get('/', (req, res) => {
    res.json(db.scripts);
  });

  export default r;
```

backend/src/server.js

```
import 'dotenv/config';
import express from 'express';
import cors from 'cors';
import cases from './routes/cases.js';
import hearings from './routes/hearings.js';
import evidence from './routes/evidence.js';
import scripts from './routes/scripts.js';
import { seed } from './db.js';

const app = express();
app.use(cors());
app.use(express.json());

app.get('/', (req, res) => res.json({ ok: true, name: 'Hearing Companion API' }));
app.use('/cases', cases);
app.use('/hearings', hearings);
app.use('/evidence', evidence);
app.use('/scripts', scripts);

seed();

const PORT = process.env.PORT || 4000;
app.listen(PORT, () => console.log(`API on http://localhost:${PORT}`));
```

Frontend

Vite + React + Tailwind. Uses simple components; you can add shadcn/ui later.

frontend/package.json

```
{
  "name": "hearing-companion-frontend",
  "version": "0.1.0",
  "private": true,
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "react": "^18.3.1",
    "react-dom": "^18.3.1"
  },
  "devDependencies": {
    "@vitejs/plugin-react": "^4.3.1",
    "autoprefixer": "^10.4.19",
    "postcss": "^8.4.39",
    "tailwindcss": "^3.4.10",
    "vite": "^5.4.8"
  }
}
```

frontend/index.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Disability Hearing Companion</title>
  </head>
  <body class="bg-zinc-950 text-zinc-100">
    <a href="#main" class="sr-only focus:not-sr-only focus:absolute focus:top-2
focus:left-2 bg-emerald-600 text-white px-3 py-1 rounded">Skip to content</a>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

frontend/tailwind.config.js

```
/** @type {import('tailwindcss').Config} */
export default {
  content: ['./index.html', './src/**/*.js,jsx'],
  theme: {
    extend: {}
  },
  plugins: []
};
```

frontend/postcss.config.js

```
export default {
  plugins: {
    tailwindcss: {},
    autoprefixer: {}
  }
};
```

frontend/src/styles.css

```
@tailwind base;
@tailwind components;
@tailwind utilities;

:focus-visible { outline: 2px solid #22c55e; outline-offset: 2px; }
```

frontend/src/lib/api.js

```
const API = import.meta.env.VITE_API_URL || 'http://localhost:4000';

async function req(path, options={}) {
  const res = await fetch(`${API}${path}`, { headers: { 'Content-Type':
'application/json' }, ...options });
  if (!res.ok) throw new Error(await res.text());
  return res.json();
}

export const api = {
  listCases: () => req('/cases'),
  createCase: (data) => req('/cases', { method: 'POST', body:
JSON.stringify(data) }),
}
```



```

listHearings: (caseId) => req(`/hearings${caseId}?caseId=${caseId}`:''),
createHearing: (data) => req('/hearings', { method: 'POST', body:
JSON.stringify(data) }),

listEvidence: (caseId) => req(`/evidence${caseId}?caseId=${caseId}`:''),
uploadEvidence: async ({ caseId, file, tag, notes }) => {
  const form = new FormData();
  form.append('caseId', caseId);
  if (tag) form.append('tag', tag);
  if (notes) form.append('notes', notes);
  form.append('file', file);
  const res = await fetch(`${API}/evidence`, { method: 'POST', body: form });
  if (!res.ok) throw new Error(await res.text());
  return res.json();
},

listScripts: () => req('/scripts'),
generateScript: (data) => req('/scripts/generate', { method: 'POST', body:
JSON.stringify(data) })
};

```

frontend/src/main.jsx

```

import React from 'react';
import { createRoot } from 'react-dom/client';
import './styles.css';
import App from './App.jsx';

createRoot(document.getElementById('root')).render(<App />);

```

frontend/src/components/Nav.jsx

```

import React from 'react';

export default function Nav({ current, onNavigate }) {
  const items = [
    { id: 'dashboard', label: 'Dashboard' },
    { id: 'hearings', label: 'Hearings' },
    { id: 'evidence', label: 'Evidence' },
    { id: 'scripts', label: 'Scripts' },
    { id: 'calendar', label: 'Calendar' }
  ];
  return (
    <nav aria-label="Primary" className="w-56 shrink-0 border-r border-zinc-800

```

```



p-3">
  <h1 className="text-lg font-semibold mb-3">Hearing Companion</h1>
  <ul className="space-y-1">
    {items.map(i => (
      <li key={i.id}>
        <button
          className={`w-full text-left px-3 py-2 rounded hover:bg-zinc-800
focus:bg-zinc-800 ${current===i.id?'bg-zinc-900':''}`}
          aria-current={current===i.id ? 'page' : undefined}
          onClick={() => onNavigate(i.id)}
        >{i.label}</button>
      </li>
    ))}
  </ul>
</nav>
);
}

```

frontend/src/components/Topbar.jsx

```

import React from 'react';

export default function Topbar() {
  return (
    <header
      className="sticky top-0 z-10 border-b border-zinc-800 bg-zinc-950/80 backdrop-
blur">
      <div className="mx-auto max-w-6xl px-4 py-3 flex items-center justify-
between">
        <span className="text-sm text-zinc-400">Accessible  Keyboard-first 
Private</span>
        <a className="underline text-emerald-400" href="#main">Skip to content</
a>
      </div>
    </header>
  );
}

```

frontend/src/components/Card.jsx

```

import React from 'react';
export default function Card({ title, children, actions }) {
  return (
    <section className="rounded-2xl border border-zinc-800 bg-zinc-900 p-4
focus-within:ring-2 focus-within:ring-emerald-500">

```

```

    <div className="flex items-center justify-between mb-2">
      <h2 className="text-base font-semibold">{title}</h2>
      <div>{actions}</div>
    </div>
    {children}
  </section>
);
}

```

frontend/src/components/DataTable.jsx

```

import React from 'react';

export default function DataTable({ columns, rows, empty="No data" }) {
  return (
    <div className="overflow-auto rounded-lg border border-zinc-800">
      <table className="min-w-full text-sm">
        <thead className="bg-zinc-800 text-zinc-300">
          <tr>
            {columns.map(c => (
              <th key={c.key} scope="col" className="px-3 py-2 text-left font-
medium">{c.label}</th>
            ))}
          </tr>
        </thead>
        <tbody>
          {rows.length===0 && (
            <tr><td colspan={columns.length} className="px-3 py-4 text-
zinc-400">{empty}</td></tr>
          )}
          {rows.map((r, idx) => (
            <tr key={r.id || idx} className="odd:bg-zinc-950 even:bg-zinc-900">
              {columns.map(c => (
                <td key={c.key} className="px-3 py-2 align-top">{c.render?
c.render(r[c.key], r) : String(r[c.key] ?? '')}</td>
              ))}
            </tr>
          ))}
        </tbody>
      </table>
    </div>
  );
}

```

frontend/src/components/FileDropzone.jsx

```
import React, { useRef, useState } from 'react';

export default function FileDropzone({ onFile }) {
  const inputRef = useRef();
  const [drag, setDrag] = useState(false);
  return (
    <div
      onDragOver={(e)=>{e.preventDefault(); setDrag(true);}}
      onDragLeave={()=>setDrag(false)}
      onDrop={(e)=>{e.preventDefault(); setDrag(false);
onFile(e.dataTransfer.files[0]);}}
      className={`rounded-xl border-2 border-dashed p-4 text-center ${drag?
'border-emerald-500 bg-emerald-500/10': 'border-zinc-700'}`}
      role="region" aria-label="Upload evidence"
    >
      <p className="mb-2">Drop a file here or</p>
      <button className="underline" onClick={()=>inputRef.current?.click()}
>browse</button>
      <input ref={inputRef} type="file" className="sr-only" onChange={(e)=>
onFile(e.target.files[0]) } />
    </div>
  );
}
```

frontend/src/pages/Dashboard.jsx

```
import React, { useEffect, useState } from 'react';
import Card from '../components/Card.jsx';
import DataTable from '../components/DataTable.jsx';
import { api } from '../lib/api.js';

export default function Dashboard() {
  const [cases, setCases] = useState([]);
  const [hearings, setHearings] = useState([]);

  useEffect(()=>{
    api.listCases().then(setCases);
    api.listHearings().then(setHearings);
  }, []);

  return (
    <div className="grid gap-4 md:grid-cols-2" id="main" tabIndex={-1}>
      <Card title="Your Cases">
        <DataTable
```

```

        columns={[{key: 'title', label: 'Title'}, {key: 'status', label: 'Status'},
{key: 'createdAt', label: 'Created'}}]
        rows={cases}
        empty="No cases yet"
      />
    </Card>
    <Card title="Upcoming Hearings">
      <DataTable
        columns={[{key: 'date', label: 'Date'}, {key: 'location', label: 'Location'},
{key: 'notes', label: 'Notes'}}]
        rows={hearings}
        empty="No hearings scheduled"
      />
    </Card>
    <Card title="Prep Checklist" actions={<span className="text-sm text-
zinc-400">low-spoon</span>>
      <ul className="list-disc ml-6 space-y-1 text-sm text-zinc-300">
        <li>Print med list & symptoms (POTS, hEDS, pain).
        </li>
        <li>Draft 3 one-minute statements: daily limits, safety risks,
typical bad day.</li>
        <li>Upload RFC letters & imaging under Evidence.</li>
      </ul>
    </Card>
    <Card title="Quick Scripts">
      <div className="text-sm text-zinc-300 space-y-2">
        <p><strong>Calm:</strong> I need it on record that my conditions
prevent reliable, safe work. Please note this in the file.</p>
        <p><strong>Firm:</strong> I require accommodations and a clear record
of my daily functional limits.</p>
        <p><strong>Warm:</strong> I'm doing everything I can; benefits are
necessary for safe, independent living.</p>
      </div>
    </Card>
  </div>
);
}

```

frontend/src/pages/Hearings.jsx

```

import React, { useEffect, useState } from 'react';
import Card from '../components/Card.jsx';
import DataTable from '../components/DataTable.jsx';
import { api } from '../lib/api.js';

export default function Hearings() {

```

```

const [caseId, setCaseId] = useState('');
const [cases, setCases] = useState([]);
const [hearings, setHearings] = useState([]);
const [form, setForm] = useState({ date: '', location: '', notes: '' });

useEffect(()=>{ api.listCases().then(setCases); },[]);
useEffect(()=>{ api.listHearings(caseId || undefined).then(setHearings); },
[caseId]);

async function create(e){
  e.preventDefault();
  if (!caseId) return alert('Pick a case');
  const item = await api.createHearing({ caseId, ...form });
  setHearings(h=>[item,...h]);
  setForm({ date:'', location:'', notes:'' });
}

return (
  <div className="space-y-4" id="main" tabIndex={-1}>
    <Card title="Schedule Hearing">
      <form onSubmit={create} className="grid gap-2 md:grid-cols-4">
        <label className="col-span-1">Case
          <select className="mt-1 w-full bg-zinc-900 border border-zinc-700
rounded p-2" value={caseId} onChange={e=>setCaseId(e.target.value)} aria-
label="Select case">
            <option value="">Select 

```

```

        </div>
      </form>
    </Card>
    <Card title="Hearings">
      <DataTable columns={[{key: 'date', label: 'Date'},
{key: 'location', label: 'Location'}, {key: 'notes', label: 'Notes'},
{key: 'status', label: 'Status'}]} rows={hearings} empty="No hearings" />
    </Card>
  </div>
);
}

```

frontend/src/pages/Evidence.jsx

```

import React, { useEffect, useState } from 'react';
import Card from '../components/Card.jsx';
import DataTable from '../components/DataTable.jsx';
import FileDropzone from '../components/FileDropzone.jsx';
import { api } from '../lib/api.js';

export default function Evidence() {
  const [cases, setCases] = useState([]);
  const [caseId, setCaseId] = useState('');
  const [rows, setRows] = useState([]);
  const [meta, setMeta] = useState({ tag: '', notes: '' });

  useEffect(() => { api.listCases().then(setCases); }, []);
  useEffect(() => { api.listEvidence(caseId || undefined).then(setRows); },
[caseId]);

  async function onFile(file){
    if (!caseId) return alert('Pick a case first');
    const item = await api.uploadEvidence({ caseId, file, ...meta });
    setRows(r=>[item, ...r]);
  }

  return (
    <div className="space-y-4" id="main" tabIndex=-1>
      <Card title="Upload Evidence">
        <div className="grid gap-2 md:grid-cols-4">
          <label className="col-span-1">Case
            <select className="mt-1 w-full bg-zinc-900 border border-zinc-700
rounded p-2" value={caseId} onChange={e=>setCaseId(e.target.value)}>
              <option value="">Select  </option>
              {cases.map(c=> <option key={c.id} value={c.id}>{c.title}</
option>)}
            </select>

```

```

        </select>
      </label>
      <label className="col-span-1">Tag
        <input className="mt-1 w-full bg-zinc-900 border border-zinc-700
rounded p-2" placeholder="RFC, imaging, meds" value={meta.tag}
onChange={e=>setMeta(m=>({...m,tag:e.target.value}))} />
      </label>
      <label className="col-span-2">Notes
        <input className="mt-1 w-full bg-zinc-900 border border-zinc-700
rounded p-2" value={meta.notes}
onChange={e=>setMeta(m=>({...m,notes:e.target.value}))} />
      </label>
    </div>
    <div className="mt-3">
      <FileDropzone onFile={onFile} />
    </div>
  </Card>
  <Card title="Evidence List">
    <DataTable
      columns={[
        {key:'uploadedAt',label:'Uploaded'},
        {key:'tag',label:'Tag'},
        {key:'originalName',label:'File'},
        {key:'notes',label:'Notes'}
      ]}
      rows={rows}
      empty="No files yet"
    />
  </Card>
</div>
);
}

```

frontend/src/pages/Scripts.jsx

```

import React, { useEffect, useState } from 'react';
import Card from '../components/Card.jsx';
import DataTable from '../components/DataTable.jsx';
import { api } from '../lib/api.js';

export default function Scripts() {
  const [rows, setRows] = useState([]);
  const [form, setForm] = useState({ audience: 'Disability Hearing', need:
'State daily limits and safety risks', tone: 'calm' });

  useEffect(()=>{ api.listScripts().then(setRows); },[]);

```



```

async function generate(e){
  e.preventDefault();
  const item = await api.generateScript(form);
  setRows(r=>[item, ...r]);
}

return (
  <div className="space-y-4" id="main" tabIndex={-1}>
    <Card title="Generate Script">
      <form onSubmit={generate} className="grid gap-2 md:grid-cols-3">
        <label>Audience
          <input className="mt-1 w-full bg-zinc-900 border border-zinc-700
rounded p-2" value={form.audience}
onChange={e=>setForm(f=>({...f,audience:e.target.value}))} />
        </label>
        <label>Need
          <input className="mt-1 w-full bg-zinc-900 border border-zinc-700
rounded p-2" value={form.need}
onChange={e=>setForm(f=>({...f,need:e.target.value}))} />
        </label>
        <label>Tone
          <select className="mt-1 w-full bg-zinc-900 border border-zinc-700
rounded p-2" value={form.tone}
onChange={e=>setForm(f=>({...f,tone:e.target.value}))}>
            <option value="calm">Calm/Factual</option>
            <option value="firm">Firm/Boundary</option>
            <option value="warm">Warm/Relational</option>
          </select>
        </label>
        <div className="col-span-3">
          <button className="mt-2 bg-emerald-600 hover:bg-emerald-500 px-4
py-2 rounded">Generate</button>
        </div>
      </form>
    </Card>
    <Card title="Saved Scripts">
      <DataTable columns={[{key: 'createdAt', label: 'Created'},
{key: 'tone', label: 'Tone'}, {key: 'audience', label: 'Audience'},
{key: 'text', label: 'Text'}]} rows={rows} empty="Nothing yet" />
    </Card>
  </div>
);
}

```

frontend/src/pages/Calendar.jsx

```
import React, { useEffect, useState } from 'react';
import Card from '../components/Card.jsx';
import DataTable from '../components/DataTable.jsx';
import { api } from '../lib/api.js';

export default function Calendar() {
  const [rows, setRows] = useState([]);
  useEffect(() => { api.listHearings().then(setRows); }, []);

  return (
    <div className="space-y-4" id="main" tabIndex={-1}>
      <Card title="All Hearings">
        <DataTable columns={[{key: 'date', label: 'Date'},
        {key: 'location', label: 'Location'}, {key: 'notes', label: 'Notes'},
        {key: 'status', label: 'Status'}]} rows={rows} empty="No hearings" />
      </Card>
    </div>
  );
}
```

frontend/src/App.jsx

```
import React, { useState } from 'react';
import Nav from '../components/Nav.jsx';
import Topbar from '../components/Topbar.jsx';
import Dashboard from '../pages/Dashboard.jsx';
import Hearings from '../pages/Hearings.jsx';
import Evidence from '../pages/Evidence.jsx';
import Scripts from '../pages/Scripts.jsx';
import Calendar from '../pages/Calendar.jsx';

export default function App(){
  const [page, setPage] = useState('dashboard');
  return (
    <div className="min-h-screen">
      <Topbar />
      <div className="mx-auto max-w-6xl flex">
        <Nav current={page} onNavigate={setPage} />
        <main className="flex-1 p-4" id="main" tabIndex={-1}>
          {page==='dashboard' && <Dashboard/>}
          {page==='hearings' && <Hearings/>}
          {page==='evidence' && <Evidence/>}
          {page==='scripts' && <Scripts/>}
          {page==='calendar' && <Calendar/>}
        </main>
      </div>
    </div>
  );
}
```

```
        </main>
      </div>
    </div>
  );
}
```

Run It

1) Backend

```
cd backend
cp .env.example .env
npm i
npm run dev
```

API at `http://localhost:4000`

2) Frontend

```
cd ../frontend
npm i
# Point frontend to your API if needed
# echo "VITE_API_URL=http://localhost:4000" > .env
npm run dev
```

Next Up (Optional Enhancements)

- Replace in-memory DB with SQLite (better-sqlite3) or Postgres.
- Add auth (Clerk/Supabase) and per-user encryption for evidence metadata.
- Calendar ICS export and Google Calendar push.
- AI: summarize evidence & generate hearing briefs from your script packs.
- Import your existing `.docx` advocacy scripts and surface inside **Scripts** page.