

wla-outreach-platform — Repository Bootstrap

Below is a ready-to-commit repository layout for deploying the **WLA Outreach Platform prototype** to **GitHub Pages** (project pages). Replace `YOUR_GITHUB_ORG_OR_USER` where noted if you want a copy-paste CI config.

Default stack: **Vite + React + TypeScript + TailwindCSS**. CI: **GitHub Actions** (deploys to `gh-pages`).

File tree

```
wla-outreach-platform/
├── .github/
│   └── workflows/
│       └── deploy.yml
├── public/
│   └── favicon.svg
├── src/
│   ├── main.tsx
│   ├── App.tsx
│   ├── components/
│   │   └── InfoCard.tsx
│   ├── pages/
│   │   ├── HomePage.tsx
│   │   ├── PreCamp.tsx
│   │   ├── DuringCamp.tsx
│   │   ├── PostCamp.tsx
│   │   ├── OutreachCalculator.tsx
│   │   ├── KeysAndGames.tsx
│   │   ├── DeerAging.tsx
│   │   ├── HabitatBuilder.tsx
│   │   ├── JobsPage.tsx
│   │   ├── PolicyPage.tsx
│   │   └── PrivacyPage.tsx
│   ├── styles.css
│   └── vite-env.d.ts
├── index.html
├── package.json
├── tsconfig.json
├── tsconfig.app.json
├── postcss.config.js
└── tailwind.config.js
```

```
├ vite.config.ts
├ .gitignore
├ LICENSE
└ README.md
```

GitHub Actions — `.github/workflows/deploy.yml`

```
name: Deploy to GitHub Pages

on:
  push:
    branches: [ main ]
  workflow_dispatch:

permissions:
  contents: read
  pages: write
  id-token: write

concurrency:
  group: "pages"
  cancel-in-progress: true

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout
        uses: actions/checkout@v4

      - name: Setup Node
        uses: actions/setup-node@v4
        with:
          node-version: "20"
          cache: npm

      - name: Install
        run: npm ci

      - name: Build
        env:
          VITE_BASE: "/wla-outreach-platform/" # change if repo name differs
        run: npm run build
```

```

    - name: Upload artifact
      uses: actions/upload-pages-artifact@v3
      with:
        path: ./dist

  deploy:
    needs: build
    runs-on: ubuntu-latest
    environment:
      name: github-pages
      url: ${ steps.deployment.outputs.page_url }
    steps:
      - name: Deploy to GitHub Pages
        id: deployment
        uses: actions/deploy-pages@v4

```



package.json

```

{
  "name": "wla-outreach-platform",
  "version": "0.1.0",
  "private": true,
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview --port 5173"
  },
  "dependencies": {
    "lucide-react": "^0.469.0",
    "react": "^18.3.1",
    "react-dom": "^18.3.1"
  },
  "devDependencies": {
    "@types/react": "^18.3.4",
    "@types/react-dom": "^18.3.0",
    "autoprefixer": "^10.4.20",
    "postcss": "^8.4.47",
    "tailwindcss": "^3.4.13",
    "typescript": "^5.6.2",
    "vite": "^5.4.8"
  }
}

```



vite.config.ts

```
import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react'

// Base path helps assets resolve on GitHub Pages project sites.
const base = process.env.VITE_BASE || '/wla-outreach-platform/'

export default defineConfig({
  plugins: [react()],
  base,
})
```

If your repo name changes, either set `VITE_BASE` in the workflow or edit the constant above.



tailwind.config.js

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



postcss.config.js

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



index.html

```
<!doctype html>
<html lang="en">
  <head>
```

```

<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<link rel="icon" href="/wla-outreach-platform/favicon.svg" />
<title>WLA Outreach Platform – Prototype</title>
</head>
<body>
  <div id="root"></div>
  <script type="module" src="/src/main.tsx"></script>
</body>
</html>

```

The favicon path assumes Pages base `/wla-outreach-platform/`. When developing locally, Vite rewrites as needed.



`src/main.tsx`

```

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

createRoot(document.getElementById('root')!).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
)

```



`src/styles.css`

```

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

/* App base tweaks */
html, body, #root { height: 100%; }

```

`src/App.tsx`

```

import React, { useMemo, useState } from "react";
import { Bird, Bug, Castle, CheckCircle2, FileDown, FileText, Fish, Footprints,

```

```

Home, Landmark, LogIn, LogOut, Map, Microscope, Settings, Shield, Sparkles,
Squirrel, User, Users, CloudUpload, Camera, Mic, Printer } from "lucide-react";
import HomePage from './pages/HomePage'
import PreCamp from './pages/PreCamp'
import DuringCamp from './pages/DuringCamp'
import PostCamp from './pages/PostCamp'
import OutreachCalculator from './pages/OutreachCalculator'
import KeysAndGames from './pages/KeysAndGames'
import DeerAging from './pages/DeerAging'
import HabitatBuilder from './pages/HabitatBuilder'
import JobsPage from './pages/JobsPage'
import PolicyPage from './pages/PolicyPage'
import PrivacyPage from './pages/PrivacyPage'

function useMockAuth(){
  const [user, setUser] =
  useState<{name:string,email:string,role:"student"|"mentor"|"staff"|null}>
  null>(null);
  return { user, signIn:(p:"google"|"microsoft"|"meta")=> setUser({name:"Alex
Student", email:`alex.student@wla.example`, role:"student"}), signOut:()=>
setUser(null) };
}

export default function App(){
  const auth = useMockAuth();
  const [tab, setTab] = useState("home");
  return (
    <div className="min-h-screen bg-gray-50">
      <header className="sticky top-0 z-10 border-b bg-white/70 backdrop-blur">
        <div
          className="mx-auto flex max-w-7xl items-center justify-between px-4 py-3">
          <div className="flex items-center gap-2">
            <Sparkles className="text-emerald-600"/>
            <h1 className="text-lg font-semibold">WLA Outreach Platform 
            Prototype</h1>
          </div>
          <div className="flex items-center gap-2">
            {auth.user ? (
              <>
                <span className="hidden text-sm text-gray-600
md:inline">{auth.user.name}  {auth.user.role}</span>
                <button onClick={()=>auth.signOut()} className="inline-flex
items-center gap-1 rounded-xl border px-3 py-1.5 hover:bg-gray-50"><LogOut
size={16}/> Sign out</button>
              </>
            ) : (
              <div className="flex gap-2">
                <button onClick={()=>auth.signIn("google")} className="inline-

```

```

flex items-center gap-1 rounded-xl border px-3 py-1.5 hover:bg-gray-50"><LogIn
size={16}/> Google</button>
      <button onClick={()=>auth.signIn("microsoft")}
className="inline-flex items-center gap-1 rounded-xl border px-3 py-1.5
hover:bg-gray-50"><LogIn size={16}/> Outlook</button>
      <button onClick={()=>auth.signIn("meta")}
className="inline-flex items-center gap-1 rounded-xl border px-3 py-1.5
hover:bg-gray-50"><LogIn size={16}/> Meta</button>
    </div>
  )}
</div>
</div>
</header>

<main className="mx-auto max-w-7xl px-4 py-6">
  <Nav tab={tab} setTab={setTab} />
  <div className="mt-6 rounded-2xl border bg-white p-4 shadow-sm">
    {tab==="home" && <HomePage/>}
    {tab==="pre" && <PreCamp/>}
    {tab==="camp" && <DuringCamp/>}
    {tab==="post" && <PostCamp/>}
    {tab==="outreach" && <OutreachCalculator/>}
    {tab==="keys" && <KeysAndGames/>}
    {tab==="deer" && <DeerAging/>}
    {tab==="habitat" && <HabitatBuilder/>}
    {tab==="jobs" && <JobsPage/>}
    {tab==="policy" && <PolicyPage/>}
    {tab==="privacy" && <PrivacyPage/>}
  </div>
</main>
</div>
);
}

function Nav({tab, setTab}:{tab:string,setTab:(t:string)=>void}){
  const items = [
    {id:"home", label:"Home", icon: Home},
    {id:"pre", label:"Pre-Camp", icon: Users},
    {id:"camp", label:"During Camp", icon: Castle},
    {id:"post", label:"Post-Camp", icon: FileText},
    {id:"outreach", label:"Outreach Calculator", icon: CheckCircle2},
    {id:"keys", label:"Keys & Games", icon: Microscope},
    {id:"deer", label:"Deer Aging", icon: Squirrel},
    {id:"habitat", label:"Habitat Builder", icon: Fish},
    {id:"jobs", label:"Jobs", icon: Footprints},
    {id:"policy", label:"Policy", icon: Landmark},
    {id:"privacy", label:"Privacy & Settings", icon: Shield},
  ]

```

```

];
return (
  <nav className="grid grid-cols-2 gap-2 md:grid-cols-6 lg:grid-cols-11">
    {items.map(it=> (
      <button key={it.id} onClick={()=>setTab(it.id)} className={`inline-flex
items-center justify-center gap-2 rounded-xl border px-3 py-2 text-sm $
{tab===it.id? 'bg-emerald-50 border-emerald-200' : 'hover:bg-gray-50'}}`>
        <it.icon size={16}/> {it.label}
      </button>
    ))}
  </nav>
);
}

```

Components & pages

Paste the following into their respective files.

src/components/InfoCard.tsx

```

import React from 'react'
export default function InfoCard({title, icon:Icon, children}:{title:string,
icon:any, children:any}){
  return (
    <div className="rounded-2xl border p-4">
      <div className="mb-2 flex items-center gap-2"><Icon size={18}
className="text-emerald-600"/><h3 className="font-medium">{title}</h3></div>
      <div className="text-sm text-gray-700">{children}</div>
    </div>
  );
}

```

src/pages/HomePage.tsx

```

import React from 'react'
import { Shield, Sparkles } from 'lucide-react'
import InfoCard from '../components/InfoCard'
export default function HomePage(){
  return (
    <div className="grid gap-3 md:grid-cols-2">
      <InfoCard title="What this is" icon={Sparkles}>
        A preview shell for a privacy-first, youth-safe WLA outreach web app.
        Sign in via OAuth (Google/Outlook/Meta), offline-friendly PWA, with pre/during/

```


post-camp workflows, calculators, natural-history games, reporting exports, jobs, and policy lookup.

```
</InfoCard>
<InfoCard title="Data hygiene (short)" icon={Shield}>
  <ul className="list-disc pl-5">
    <li>Collect minimum PII, encrypt at rest/in transit.</li>
    <li>Pseudonymous IDs; parental consent workflow.</li>
    <li>Row-level security by role (student/mentor/staff).</li>
    <li>Local-first media; opt-in sync with explicit scopes.</li>
  </ul>
</InfoCard>
</div>
);
}
```

src/pages/PreCamp.tsx

```
import React from 'react'
import { Users, FileText } from 'lucide-react'
import InfoCard from '../components/InfoCard'
export default function PreCamp(){
  return (
    <div className="grid gap-4 md:grid-cols-2">
      <InfoCard title="Checklists" icon={Users}>
        <ul className="list-disc pl-5">
          <li>Forms & permissions</li>
          <li>Kit issuance (e.g., rod/reel, field journal)</li>
          <li>Safety modules (OSHA 10, First Aid/CPR)</li>
          <li>Tech check (PWA install, camera/mic access)</li>
        </ul>
      </InfoCard>
      <InfoCard title="Micro-courses" icon={FileText}>
        <ul className="list-disc pl-5">
          <li>Leave No Trace + near-water safety</li>
          <li>Intro to PA ecoregions</li>
          <li>Outreach 101 (what counts, with examples)</li>
        </ul>
      </InfoCard>
    </div>
  );
}
```

src/pages/DuringCamp.tsx

```
import React from 'react'
import { Users, Map } from 'lucide-react'
import InfoCard from '../components/InfoCard'
export default function DuringCamp(){
  return (
    <div className="grid gap-4 md:grid-cols-2">
      <InfoCard title="Daily plan & roster" icon={Users}>Team assignments,
      roles, and tailgate JHAs.</InfoCard>
      <InfoCard title="Field logging" icon={Map}>Waypoints, photo points, and
      artifacts (offline, sync later).</InfoCard>
    </div>
  );
}
```

src/pages/PostCamp.tsx

```
import React from 'react'
import { FileText, FileDown } from 'lucide-react'
import InfoCard from '../components/InfoCard'
export default function PostCamp(){
  return (
    <div className="grid gap-4 md:grid-cols-2">
      <InfoCard title="Outreach portfolio" icon={FileText}>Talks, articles,
      media, events, interpretation.</InfoCard>
      <InfoCard title="Exports" icon={FileDown}>Google Drive / OneDrive /
      local .csv, .json, .pdf.</InfoCard>
    </div>
  );
}
```

src/pages/OutreachCalculator.tsx

```
import React, { useMemo, useState } from 'react'
import { CloudUpload, FileDown, FileText, Map, Camera, Mic, Printer } from
'lucide-react'
const CODES = [
  { id: 'SAF', label: 'Safety' },
  { id: 'AQ', label: 'Aquatic Access' },
  { id: 'TRL', label: 'Trails/Rec Infra' },
  { id: 'HAB', label: 'Habitat' },
  { id: 'OPS', label: 'Ops/Visitor' },
  { id: 'GIS', label: 'GIS/Data' },
]
```

```

    { id: 'INT', label: 'Interpretation' },
    { id: 'ENG', label: 'Engagement' },
  ]
function minutesToHHMM(mins:number){ const h=Math.floor(mins/60), m=mins%60;
return `${h}:${m.toString().padStart(2,'0')}`; }
export default function OutreachCalculator(){
  const PRESETS: Record<string, { minutes:number, primary:string,
notes:string }[]> = {
    'Sayers 3-hr Habitat Build': [
      { minutes:15, primary:'SAF', notes:'Tailgate JHA; PPE; roles' },
      { minutes:60, primary:'AQ', notes:'Shoreline structure install' },
      { minutes:30, primary:'TRL', notes:'Approach tread repair; ADA' },
      { minutes:15, primary:'GIS', notes:'Waypoints + photos' },
      { minutes:30, primary:'INT', notes:'Wayfinding mock-up' },
      { minutes:30, primary:'ENG', notes:'Stakeholder debrief' },
    ],
    'Outreach Talk (1.5 hr)': [
      { minutes:15, primary:'ENG', notes:'Room setup; accessibility check' },
      { minutes:45, primary:'ENG', notes:'Public talk delivery' },
      { minutes:30, primary:'INT', notes:'Display boards / media Q&A' },
    ],
    'Trail Day (2 hr)': [
      { minutes:15, primary:'SAF', notes:'Tailgate JHA; tool check' },
      { minutes:60, primary:'TRL', notes:'Tread work; drainage dips' },
      { minutes:30, primary:'HAB', notes:'Invasives pull along corridor' },
      { minutes:15, primary:'GIS', notes:'Photo-points + waypoints' },
    ],
  }
  const [preset, setPreset] = useState<string>('Sayers 3-hr Habitat Build')
  const [slices, setSlices] = useState(PRESETS[preset])
  const [artifacts, setArtifacts] =
useState<{id:string,type:'photo'|'audio'|'doc'|'map',name:string,ts:string,note:string,url?:string}>([])
  const total = slices.reduce((a,r)=>a+r.minutes,0)
  const byCode:Record<string,number> = useMemo(()=>{
    const m:Record<string,number>={}; CODES.forEach(c=>m[c.id]=0);
    slices.forEach(s=> m[s.primary]+=s.minutes); return m;
  },[slices])
  function loadPreset(name:string){ setPreset(name); setSlices(PRESETS[name]); }
  function addSlice(){ setSlices(prev=> [...prev, {minutes:15, primary:'SAF',
notes:''}]); }
  function addArtifact(type:'photo'|'audio'|'doc'|'map', file?:File){
    const name = file?.name || `${type.toUpperCase()}_${new
Date().toISOString().slice(0,19).replace(/[:T]/g,'')}`
    const url = file ? URL.createObjectURL(file) : undefined
    setArtifacts(prev=> [...prev, {id:crypto.randomUUID(), type, name, ts: new
Date().toISOString(), note: '', url, size: file?.size}])

```

```

    }
    function handleFileInput(ev:React.ChangeEvent<HTMLInputElement>,
type:'photo'|'audio'|'doc'|'map'){
        const file = ev.target.files?.[0]; if(!file) return; addArtifact(type,
file); ev.target.value = ''
    }
    function exportArtifactsCSV(){
        const header = ['Type','Name','Timestamp','Size(bytes)','Note','URL']
        const rows = artifacts.map(a=> [a.type,a.name,a.ts,a.size??'',`"$
{(a.note||'').replace(/"/g, '""')}`",a.url??''].join(','))
        const csv = [header.join(','), ...rows].join('\n')
        const blob = new Blob([csv],{type:'text/csv'})
        const a = document.createElement('a')
        a.href = URL.createObjectURL(blob); a.download = `WLA_Artifacts_${new
Date().toISOString().slice(0,10)}.csv`; a.click()
    }
    function stubDriveExport(){ alert('Drive export stub: In production request
Google Drive scope and upload CSV/JSON and media with explicit consent.') }
    function stubOneDriveExport(){ alert('OneDrive export stub: In production
request Microsoft Graph scope and push selected files to a WLA folder.') }
    function printSummary(){
        const win = window.open('', '_blank'); if(!win) return
        const rows = slices.map(s=> `<tr><td>${s.primary}</td><td>${s.minutes}</
td><td>${s.notes||''}</td></tr>`).join('')
        const arts = artifacts.map(a=> `<tr><td>${a.type}</td><td>${a.name}</
td><td>${a.ts.replace('T',' ').slice(0,16)}</td><td>${a.note||''}</td></
tr>`).join('')
        win.document.write(`<!doctype html><html><head><title>WLA Outreach Summary</
title>
        <style>body{font-family:ui-sans-serif,system-ui; padding:24px} h1{font-
size:18px} table{width:100%; border-collapse:collapse} th,td{border:1px solid
#ddd; padding:6px; font-size:12px} th{background:#f6f6f6}</style>
        </head><body>
        <h1>Outreach Session Summary</h1>
        <h2>Slices</h2>
        <table><thead><tr><th>Primary code</th><th>Minutes</th><th>Notes</th></
tr></thead><tbody>${rows}</tbody></table>
        <h2 style="margin-top:16px">Artifacts</h2>
        <table><thead><tr><th>Type</th><th>Name</th><th>Timestamp</th><th>Note</
th></tr></thead><tbody>${arts||'<tr><td colspan="4">No artifacts</td></tr>'}</
tbody></table>
        <p style="margin-top:16px">Totals by code (minutes): $
{Object.entries(byCode).map(([k,v])=>`${k}: ${v}`)}.join(', ')</p>
        <script>window.onload = () => { window.print(); }</script>
        </body></html>`)
        win.document.close()
    }
    return (

```

```

<div className="grid gap-4 lg:grid-cols-2">
  <div className="rounded-2xl border p-4">
    <div className="mb-2 flex flex-wrap items-center gap-2">
      <h3 className="font-medium">Slices</h3>
      <span className="ml-auto text-xs text-gray-600">Preset:</span>
      <select value={preset} onChange={e=> loadPreset(e.target.value)}
className="rounded border px-2 py-1 text-sm">
        {Object.keys(PRESETS).map(k=> <option key={k} value={k}>{k}</
option>)}
      </select>
    </div>
    <ul className="space-y-2">
      {slices.map((s,idx)=> (
        <li key={idx} className="flex flex-col gap-2 md:flex-row md:items-
center">
          <select value={s.primary} onChange={e=>{ const v=e.target.value;
setSlices(prev=> prev.map((p,i)=> i===idx? {...p, primary:v}:p)); }}
className="rounded border px-2 py-1 w-44">
            {CODES.map(c=> <option key={c.id} value={c.id}>{c.id}</option>)}
          </select>
          <input type="number" min={15} step={15} value={s.minutes}
onChange={e=>{ const v=Number(e.target.value); setSlices(prev=>
prev.map((p,i)=> i===idx? {...p, minutes:v}:p)); }} className="w-28 rounded
border px-2 py-1"/>
          <input value={s.notes} onChange={e=>{ const v=e.target.value;
setSlices(prev=> prev.map((p,i)=> i===idx? {...p, notes:v}:p)); }}
className="flex-1 rounded border px-2 py-1" placeholder="Specific verbs:
install, anchor, survey..."/>
        </li>
      ))}
    </ul>
    <div className="mt-3 flex gap-2">
      <button onClick={()=> setSlices(prev=> [...prev, {minutes:15,
primary:'SAF', notes:''}])} className="rounded-xl border px-3 py-1.5 hover:bg-
gray-50">+ Add 15 min slice</button>
      <div className="ml-auto text-sm text-gray-600 flex items-center
gap-2">
        <span>Total credited</span>
        <span className="font-mono font-semibold">{minutesToHHMM(total)} h</
span>
      </div>
    </div>
    <div className="mt-4 grid grid-cols-2 gap-1 text-sm">
      {CODES.map(c=> (
        <div key={c.id} className="flex items-center justify-between">
          <span className="text-gray-600">{c.id}</span>
          <span className="font-mono">{minutesToHHMM(byCode[c.id]||0)} h</

```

```

span>
    </div>
  )})
</div>
</div>
<div className="rounded-2xl border p-4">
  <h3 className="mb-2 font-medium">Artifacts checklist</h3>
  <ul className="space-y-2 text-sm">
    <li className="flex items-center gap-2"><input type="checkbox"
className="h-4 w-4"/> JHA photo (tailgate)</li>
    <li className="flex items-center gap-2"><input type="checkbox"
className="h-4 w-4"/> 1 geo□tagged photo per hour</li>
    <li className="flex items-center gap-2"><input type="checkbox"
className="h-4 w-4"/> Waypoint map / mobile form receipt</li>
    <li className="flex items-center gap-2"><input type="checkbox"
className="h-4 w-4"/> Stakeholder debrief notes</li>
  </ul>
  <div className="mt-4 grid grid-cols-2 gap-2">
    <label className="inline-flex items-center gap-2 rounded-xl border
px-3 py-2 hover:bg-gray-50 cursor-pointer"><Camera size={16}/> Add photo<input
onChange={e=>handleFileInput(e,'photo')} type="file" accept="image/*"
className="hidden"/></label>
    <label className="inline-flex items-center gap-2 rounded-xl border
px-3 py-2 hover:bg-gray-50 cursor-pointer"><Mic size={16}/> Add audio<input
onChange={e=>handleFileInput(e,'audio')} type="file" accept="audio/*"
className="hidden"/></label>
    <label className="inline-flex items-center gap-2 rounded-xl border
px-3 py-2 hover:bg-gray-50 cursor-pointer"><Map size={16}/> Add map<input
onChange={e=>handleFileInput(e,'map')} type="file" className="hidden"/></
label>
    <label className="inline-flex items-center gap-2 rounded-xl border
px-3 py-2 hover:bg-gray-50 cursor-pointer"><FileText size={16}/> Add doc<input
onChange={e=>handleFileInput(e,'doc')} type="file" className="hidden"/></
label>
  </div>
  <div className="mt-3 rounded-xl border overflow-x-auto">
    <table className="w-full text-sm">
      <thead className="bg-gray-50"><tr><th className="p-2 text-
left">Type</th><th className="p-2 text-left">Name</th><th className="p-2 text-
left">Timestamp</th><th className="p-2 text-left">Note</th><th className="p-2
text-left">Open</th></tr></thead>
      <tbody>
        {artifacts.map(a=> (
          <tr key={a.id} className="border-t">
            <td className="p-2 uppercase text-gray-600">{a.type}</td>
            <td className="p-2">{a.name}</td>
            <td className="p-2 font-mono">{a.ts.replace('T',' ').slice(0,

```


```

16)}}</td>
      <td className="p-2"><input className="w-full rounded border
px-2 py-1" value={a.note} onChange={e=> setArtifacts(prev=> prev.map(x=>
x.id===a.id? {...x, note: e.target.value}: x) )}/></td>
      <td className="p-2">{a.url? <a href={a.url} target="_blank"
rel="noreferrer" className="text-emerald-700 underline">view</a> : <span
className="text-gray-400">n/a</span>}</td>
    </tr>
  )}}
  {artifacts.length===0 && <tr><td className="p-3 text-gray-500"
colSpan={5}>No artifacts yet.</td></tr>}
</tbody>
</table>
</div>
<div className="mt-3 flex flex-wrap gap-2">
  <button onClick={exportArtifactsCSV} className="inline-flex items-
center gap-2 rounded-xl border px-3 py-2 hover:bg-gray-50"><FileDown size={16}/
> Export artifacts (.csv)</button>
  <button onClick={stubDriveExport} className="inline-flex items-center
gap-2 rounded-xl border px-3 py-2 hover:bg-gray-50"><CloudUpload size={16}/>
Push to Drive (stub)</button>
  <button onClick={stubOneDriveExport} className="inline-flex items-
center gap-2 rounded-xl border px-3 py-2 hover:bg-gray-50"><CloudUpload
size={16}/> Push to OneDrive (stub)</button>
  <button onClick={printSummary} className="inline-flex items-center
gap-2 rounded-xl border px-3 py-2 hover:bg-gray-50"><Printer size={16}/>
Print / Save PDF</button>
</div>
</div>
</div>
)
}

```

src/pages/KeysAndGames.tsx

```

import React, { useState } from 'react'
import { Bug, Bird, Microscope } from 'lucide-react'
import InfoCard from '../components/InfoCard'
export default function KeysAndGames(){
  return (
    <div className="grid gap-4 lg:grid-cols-3">
      <div className="lg:col-span-2 rounded-2xl border p-4">
        <h3 className="mb-2 font-medium flex items-center gap-2"><Bug size={16}/
> Macroinvertebrate Key  Mini Game</h3>
        <MacroKeyGame/>
      </div>

```

```

    <InfoCard title="Plants" icon={LeafIcon}>Key by leaf/branch traits;
optional camera capture.</InfoCard>
    <InfoCard title="Insects" icon={Bug}>Wing/antenna/size keys; upload photo
for ID via iNat  style API.</InfoCard>
    <InfoCard title="Bird song" icon={Bird}>Record/upload clips; submit to
partner API; return candidate species.</InfoCard>
    <InfoCard title="Photo ID" icon={Microscope}>Upload  suggest taxa;
student confirms and adds notes.</InfoCard>
  </div>
)
}
function LeafIcon(props:any){ return <svg {...props} className={`h-4 w-4 $
{props.className} || ''`} viewBox="0 0 24 24" fill="none" stroke="currentColor"
strokeWidth="2" strokeLinecap="round" strokeLinejoin="round"><path d="M11 3a8 8
0 0 0 8 8c0 7-4 10-8 10S3 18 3 11 6 3 13 3"/></svg>; }
function MacroKeyGame(){
  type Node = { id:string, q?:string, options?: {label:string, next:string}[],
result?: string };
  const TREE:Record<string, Node> = {
    start: { id:'start', q:'Does the aquatic insect have 3 distinct tails
(cerci)?', options:[ {label:'Yes', next:'threeTails'}, {label:'No',
next:'notThree'} ] },
    threeTails: { id:'threeTails', q:'Do you see plate-like gills along the
sides of the abdomen?', options:[ {label:'Yes', next:'mayfly'}, {label:'No',
next:'stoneflyCheck'} ] },
    mayfly: { id:'mayfly', result:'Likely <b>Mayfly nymph</b> (Order
Ephemeroptera): 3 tails, lateral gills.' },
    stoneflyCheck: { id:'stoneflyCheck', q:'Are there 2 tails and gills under
the thorax (not along abdomen)?', options:[ {label:'Yes', next:'stonefly'},
{label:'No', next:'caddisCheck'} ] },
    stonefly: { id:'stonefly', result:'Likely <b>Stonefly nymph</b> (Order
Plecoptera): 2 tails, thoracic gills.' },
    caddisCheck: { id:'caddisCheck', q:'Is the insect in a portable case of
sticks/sand (a "case")?', options:[ {label:'Yes', next:'caddis'}, {label:'No',
next:'odonateCheck'} ] },
    caddis: { id:'caddis', result:'Likely <b>Caddisfly larva</b> (Order
Trichoptera): case-building or free-living.' },
    odonateCheck: { id:'odonateCheck', q:'Are the lower mouthparts a large
hinged mask covering the face?', options:[ {label:'Yes', next:'odonata'},
{label:'No', next:'beetleWormCheck'} ] },
    odonata: { id:'odonata', result:'Likely <b>Dragonfly/Damselfly nymph</b>
(Order Odonata): extendable labium (mask).' },
    beetleWormCheck: { id:'beetleWormCheck', q:'Is the body worm-like without
true legs, or does it have hardened wing covers?', options:[ {label:'Worm-like
(no true legs)', next:'flyLarva'}, {label:'Hardened covers (elytra)/distinct
legs', next:'beetle'} ] },
    flyLarva: { id:'flyLarva', result:'Likely <b>Fly larva</b> (Order Diptera):

```



```

no true legs, worm-like.' },
  beetle: { id:'beetle', result:'Likely <b>Aquatic beetle/larva</b> (Order
Coleoptera): hardened wing covers or distinct legs.' },
  notThree: { id:'notThree',
q:'Does it have 2 long tails and a very flattened body?', options:[
{label:'Yes', next:'stonefly'}, {label:'No', next:'caddisCheck'} ] },
  };
  const [cur, setCur] = useState<Node>(TREE.start);
  const [trail, setTrail] = useState<string[]>(['start']);
  function choose(nextId:string){ setCur(TREE[nextId]); setTrail(prev=>
[...prev, nextId]); }
  function reset(){ setCur(TREE.start); setTrail(['start']); }
  return (
    <div className="space-y-3">
      {cur.result ? (
        <div className="rounded-xl border p-3">
          <p className="text-sm" dangerouslySetInnerHTML={{__html: cur.result}}/>
        </div>
        <div className="mt-2 text-xs text-gray-600">Path: {trail.join(' → ')}</div>
        <button onClick={reset} className="mt-3 rounded-xl border px-3 py-1.5
hover:bg-gray-50">Start over</button>
      ) : (
        <div className="rounded-xl border p-3">
          <div className="mb-2 text-sm font-medium">{cur.q}</div>
          <div className="flex flex-wrap gap-2">
            {cur.options?.map(opt=> (
              <button key={opt.label} onClick={()=>choose(opt.next)}
className="rounded-xl border px-3 py-1.5 hover:bg-gray-50">{opt.label}</button>
            ))}
          </div>
          <div className="mt-2 text-xs text-gray-600">Path: {trail.join(' → ')}</div>
        </div>
      )}
      <div className="text-xs text-gray-600">Training tool; confirm IDs with an
instructor or field guide.</div>
    </div>
  )
}

```

src/pages/DeerAging.tsx

```

import React from 'react'
import InfoCard from '../components/InfoCard'

```

```
import { Squirrel } from 'lucide-react'
export default function DeerAging(){
  return (
    <div className="grid gap-4 md:grid-cols-2">
      <InfoCard title="Body cues" icon={Squirrel}>Gallery of body shapes (fawn
      □ mature). Students select age class.</InfoCard>
      <InfoCard title="Jaw/teeth" icon={Squirrel}>Upload jaw photo; mark molars/
      premolars; guided rubric.</InfoCard>
    </div>
  )
}
```

src/pages/HabitatBuilder.tsx

```
import React from 'react'
import InfoCard from '../components/InfoCard'
import { Fish } from 'lucide-react'
export default function HabitatBuilder(){
  return (
    <div className="grid gap-4 md:grid-cols-2">
      <InfoCard title="Coldwater (Appalachian)" icon={Fish}>Choose structures
      (log, rootwad, rock vane), set flows, banks, buffer plants. Export a plan with
      BOM.</InfoCard>
      <InfoCard title="Warmwater" icon={Fish}>Shoreline complexity, spawning
      beds, access pads, invasive checks.</InfoCard>
    </div>
  )
}
```

src/pages/JobsPage.tsx

```
import React from 'react'
export default function JobsPage(){
  return (
    <div className="space-y-3">
      <div className="rounded-2xl border p-4">
        <h3 className="mb-1 font-medium">Weekly Job Feed</h3>
        <p className="text-sm text-gray-600">Filter by agency, county, and
        skill tags; subscribe to alerts.</p>
        <ul className="mt-3 grid gap-2 md:grid-cols-2">
          <li className="rounded-xl border p-3"><b>Seasonal Fisheries Tech</b>
          □ PFBC □ Centre Co. □ Tags: AQ/GIS</li>
          <li className="rounded-xl border p-3"><b>Trail Crew Member</b> □ DCNR
          □ Tioga Co. □ Tags: TRL/HAB</li>
        </ul>
      </div>
    </div>
  )
}
```

```

    </div>
  </div>
)
}

```

src/pages/PolicyPage.tsx

```

import React from 'react'
export default function PolicyPage(){
  return (
    <div className="space-y-3">
      <div className="rounded-2xl border p-4">
        <h3 className="mb-1 font-medium">Your County Who Represents You</h3>
        <p className="text-sm text-gray-600">Auto lookup by county/ZIP; list
federal/state/local reps with contact methods; show active bills on
conservation/outdoors.</p>
        <ul className="mt-2 list-disc pl-5 text-sm">
          <li>Click to draft a respectful outreach email; logs as ENG hours
(with mentor approval).</li>
        </ul>
      </div>
    </div>
  )
}

```

src/pages/PrivacyPage.tsx

```

import React from 'react'
export default function PrivacyPage(){
  return (
    <div className="space-y-3">
      <div className="rounded-2xl border p-4">
        <h3 className="mb-1 font-medium">Privacy controls</h3>
        <ul className="list-disc pl-5 text-sm">
          <li>Consent mode: <b>Minor</b> / <b>18+</b> . Parental e-consent
workflow</li>
          <li>Share level: <b>Private</b> (local only) / <b>Mentor</b> /
<b>Program</b></li>
          <li>PII redaction: show only first name + last initial in shared
spaces</li>
          <li>Media retention: auto-purge after N days unless kept in
portfolio</li>
        </ul>
      </div>
      <div className="rounded-2xl border p-4">

```

```
    <h3 className="mb-1 font-medium">Exports</h3>
    <p className="text-sm text-gray-600">Download .csv/.json; connect
Google Drive/OneDrive to push reports.</p>
  </div>
</div>
)
}
```



vite-env.d.ts

```
/// <reference types="vite/client" />
```



.gitignore

```
node_modules
.dist
.DS_Store
.env
.env.*
.vscode
```



LICENSE

MIT License

Copyright (c) 2025 Wildlife Leadership Academy

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



README.md

WLA Outreach Platform – Prototype

Privacy-first, youth-safe outreach app for Wildlife Leadership Academy (WLA). Built with Vite + React + TypeScript + Tailwind. Deploys to GitHub Pages via Actions.

Scripts

- ``npm run dev`` – local dev
- ``npm run build`` – production build
- ``npm run preview`` – preview build locally

Deploy

Pushing to ``main`` triggers the Pages workflow. If your repo name is not ``wla-outreach-platform``, update the ``VITE_BASE`` in ``.github/workflows/deploy.yml`` or set a repository secret/environment variable.

Notes

- This is a prototype shell: mock auth, stubbed exports, offline-friendly artifact capture (local URLs).
- For production, wire real OAuth (OIDC), storage with row-level security, and API adapters (iNaturalist/Bird audio, etc.).



public/favicon.svg

```
<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2" stroke-linecap="round" stroke-linejoin="round"><path d="M22 12s-3-4-10-4-10 4 3 4 10 4 10-4 10-4Z"/><path d="M22 12s-4 2-10 2-10-2-10-2"/><path d="M2 12a10 10 0 0 0 20 0"/></svg>
```