

تمام. هذه الخطوات النهائية النظيفة (بدون خلط قديم/جديد). نَقِّذْها حرفيًا في مستودع صديقك أو أي مستودع جديد.

الهدف النهائي

- . documents/: الملفات كلها داخل مجلد.
 - GitHub Actions عبر documents/index.json: يتولد ملف فهرسة تلقائيًا.
 - (PDF/ وتعرض الملفات documents/index.json تقرأ index.html صفحة صور/HTML/JSON...).
-

الخطوة 1: جهّز هيكل المستودع

داخل المستودع اجعل هذا الشكل:

```
/index.html
/favicon.ico    (اختياري)
/documents/
  مثال) pdf تعليمات
/.github/
  workflows/
    generate-documents-index.yml
```

وتقدر تسوي مجلدات داخله documents/ مهم: كل الملفات التي تريد عرضها تكون داخل مثل:

```
documents/أول ابتدائي/ملف1.pdf
documents/ثاني ابتدائي/ملف2.pdf
documents/...ملفات عامة
```

(تلقائيًا index.json توليد) GitHub Action الخطوة 2: أضف ملف الـ

أنشئ هذا الملف بالمسار التالي بالضبط:

.github/workflows/generate-documents-index.yml

loop) هذه النسخة "النهائية" وتمنع ال) ضع بداخله هذا المحتوى

```
name: Generate documents index.json on: push: branches: ["main"]
paths: - "documents/**" - ".github/workflows/generate-documents-
index.yml" paths-ignore: - "documents/index.json" workflow_dispatch:
permissions: contents: write jobs: build-index: runs-on: ubuntu-
latest steps: - name: Checkout uses: actions/checkout@v4 - name:
Generate documents/index.json run: | python3 - <<'PY' import json
import pathlib root = pathlib.Path("documents") files = [] if
root.exists(): for p in root.rglob("**"): if p.is_file() and not
p.name.startswith(".") and p.name != "index.json": rel =
p.relative_to(root).as_posix() parts = rel.split("/") folder =
parts[0] if len(parts) > 1 else "ملفات عامة" files.append({ "name":
p.name, "path": rel, "folder": folder }) files.sort(key=lambda x:
(x["folder"], x["name"])) out = root / "index.json"
out.write_text(json.dumps(files, ensure_ascii=False, indent=2),
encoding="utf-8") print(f"Wrote {out} with {len(files)} files") PY -
name: Commit index.json uses: stefanzweifel/git-auto-commit-
action@v5 with: commit_message: "Auto-generate documents/index.json"
file_pattern: "documents/index.json"
```

Action لتشغيل ال documents الخطوة 3: ارفع أي ملف داخل

documents/تعليمات.pdf: مثلاً ارفع

main إلى Push ثم اعمل

بعدها:

Actions تبويب → GitHub ادخل

شغال Workflow ستجد

documents/index.json: بعد نجاحه ستري ملف جديد ظهر تلقائياً

(index.html) الخطوة 4: ضع صفحة العرض

في جذر المستودع index.html انسخ نسختك النهائية التي تعمل عندك وضعها كملف

أن تكون هذه القيم موجودة HTML المهم داخل الـ

A) الفهرس

```
const INDEX_URL = "documents/index.json";
```

B) زر "فتح في تبويب جديد" للـ HTML

لازم يكون displayFile داخل

```
if (["html", "htm"].includes(ext)) { openTabBtn.href = fileUrl;
openTabBtn.style.display = "inline-flex"; } else {
openTabBtn.style.display = "none"; }
```

C) <pre class="json-viewer"> داخل JSON عرض

displayFile : لازم يكون في

```
if (ext === "json") { displayJson(fileUrl); return; }
```

CSS: وموجود

```
#viewer { overflow: auto; align-items: stretch; justify-content:
stretch; } .json-viewer { white-space: pre; overflow: auto;
direction: rtl; text-align: left; }
```

D) ترميز المسار (ضروري للأسماء العربية والمسارات)

```
function toUrlPath(filepath){ return
String(filepath).split("/").map(encodeURIComponent).join("/"); }
```

واستخدامه:

```
const fileUrl = `documents/${toUrlPath(filepath)}`;
```

GitHub Pages الخطوة 5: فَعِّل

من إعدادات المستودع

Settings → Pages

Source: **Deploy from a branch**

Branch: main

Folder: / (root)

GitHub Pages. ثم ادخل الرابط الذي يعطيه لك

تشغيل سريع للتأكد

- . افتح الموقع
 - . يجب يظهر → PDF اختر ملف
 - . يجب يظهر + زر فتح تبويب جديد يظهر → HTML اختر ملف
 - . يجب يظهر كنص منسق مع سكروول → JSON اختر ملف
-

ماذا تقول لصديقك باختصار؟

- "فقط documents/ ضع ملفاتك داخل"
 - "سيولده Action بيدك، ال index.json لا تكتب"
 - "سيحدّث الفهرس تلقائيًا documents/ أي إضافة/حذف داخل"
 - "في الجذر index.html صفحة العرض هي"
-

ثم يغير Template إذا تريدها أسهل على صديقك: قل له ينسخ مستودعك ك documents/ فقط محتوى