من أول سطر كود إلى أول مشروع حقيقي

FRONT-END DEVELOPER ROADMAP

دليلك العملي لتعلّم الـ Front-End من الصفر

خطوات واضحة، كورسات مجانية، مشاريع للتطبيق، ومهارات مهمة للشغل الحقيقي.

HTML & CSS | JavaScript | React | Next.js | Production Skills

إعداد: أحمد سمير

Full-Stack Developer & Instructor

قبل ما نبدأ

الموضوع مش عدد كورسات

الهدف إنك تتعلّم إزاي تفكر وتطبّق وتبني شغل حقيقي، مش تحفظ أسماء مكتبات.

السلام عليكم ورحمة الله وبركاته 🤍

في الأول متنساش تصلي على النبي عليه أفضل الصلاة والسلام.

لو ناوي تبدأ **Front-End**، متخليش أول خطوة إنك تدفع فلوس في كورس قبل ما تفهم الطريق صح. أنا **أحمد سمير**، Full-Stack Developer بخبرة أكتر من 4 سنين، وكنت في يوم من الأيام تايه ومش عارف أبدأ منين، زي ناس كتير دلوقتي.

عملت الـ Roadmap دي علشان أوفر عليك وقت الحيرة، وأوضحلك الفرق بين إنك **تتعلم فعلاً** وبين إنك تستهلك محتوى وخلاص. أنا مش ضد الكورسات المدفوعة، لكن المهم تعرف إمتى تحتاجها، وإمتى التطبيق أهم من فيديو جديد.

إزاي تستخدم الدليل ده؟

01
اتعلم
اختار المصدر المقترح لكل مرحلة واتدرّج من غير استعجال.

02
طبق
وقف الفيديو، اكتب الكود بإيدك، ودوّن النقاط المهمة.

03
ابنِ
نفّذ مشروع من دماغك قبل ما تفتح المرحلة اللي بعدها.

04
انشر
ارفع الكود على GitHub وانشر المشروع برابط مباشر.

> ﴿ وَأَن لَّيْسَ لِلْإِنسَانِ إِلَّا مَا سَعَىٰ ۝ وَأَنَّ سَعْيَهُ سَوْفَ يُرَىٰ ۝ ثُمَّ يُجْزَاهُ الْجَزَاءَ الْأَوْفَىٰ ﴾
>
> سورة النجم: 39–41

افتكر: النتيجة مش أهم من إنك تسعى وتكمّل. التقدّم الحقيقي بيتقاس بالحاجات اللي قدرت تبنيها بنفسك.

الخريطة العامة

10 محطات للـ Front-End

الطريق الأساسي زي ما هو: HTML → CSS → JavaScript → React → Next.js، مع مهارات الشغل المهمة بينهم.

01 **HTML** — هيكل الصفحة والمعنى الدلالي	02 **CSS + Responsive** — تنسيق الواجهات والموبايل
03 **Git / GitHub** — حفظ الشغل والتعاون	04 **JavaScript** — المنطق والتفاعل
05 **APIs + Browser** — التعامل مع البيانات	06 **TypeScript + Tooling** — تنظيم الكود والـ Types
07 **React** — Components و State	08 **React at Scale** — Routing و Server State
09 **Next.js** — App Router و Rendering	10 **Ship & Portfolio** — Testing و Performance و Deployment

قاعدة مهمة

المراحل 01 و02 و04 و07 و09 هي عمود الطريق الأساسي. باقي المهارات هتتعلمها في وقتها، ومش مطلوب تخلص كل الأدوات قبل ما تبدأ مشاريع.

ده ترتيب تعلّم مقترح، مش شرط يكون خط سير ثابت لكل شخص. ما تستعجلش الانتقال لمجرد إن عدد فيديوهات الكورس خلص.

المرحلة 01 / الأساس

HTML — بنبني الهيكل

ابدأ بكورس الـ HTML اللي اخترته في النسخة الأصلية، وطبّق بعد كل جزء واكتب Notes لو ده بيساعدك.

ابدأ من هنا

أول مرحلة هي HTML. ركّز على إنك تفهم كل Element بتستخدمه وليه، مش تحفظ شكل التاج وخلاص.

اتعلّم إيه؟

- Semantic HTML: الفرق بين `header` و `main` و `section` و `article`.
- Forms، Tables، Lists، Images، Links، Headings و Labels.
- أساسيات SEO: عنوان الصفحة، وصفها، وترتيب العناوين الصحيح.
- Accessibility من أول يوم: `alt` للصور و `label` لكل input.

المصادر

- كورس HTML الأصلي — زي ما هو ↗
- MDN: تعلّم HTML خطوة بخطوة ↗

تطبيق المرحلة

ابنِ صفحة شخصية بسيطة + صفحة تسجيل حساب فيها Form صحيح. جرّب تخليها مفهومة حتى لو قفلت الـ CSS تمامًا.

جاهز تنتقل لما...

تقدر تعمل صفحة HTML نظيفة من غير ما تنسخ الكود، وتشرح استخدام Semantic Tags و Forms بشكل صحيح.

المرحلة 02 / تصميم الواجهة

CSS + Responsive Design

هنا هنظبط شكل الموقع، وتجربة الاستخدام، وطريقة ظهوره على الشاشات المختلفة.

الـ CSS بحر كبير، بس مش مطلوب تكون حافظ كل Property. الأهم إنك تفهم الـ Layout كويس وتعرف تبني واجهة ثابتة ومتجاوبة.

المطلوب تفهمه كويس

- Box Model، Cascade، Specificity، Selectors، Typography والألوان.
- Flexbox و CSS Grid و Positioning.
- Responsive Design بنظام Mobile First و Media Queries.
- States زي hover/focus، وأساسيات الـ Animations من غير إفراط.

الكورسات والمراجع

- كورس CSS الأصلي — عبد الرحمن جمال ↗
- Responsive Design — MDN ↗
- Accessibility — MDN ↗

Bootstrap و Tailwind

اتعلم أساسيات CSS الأول، بعد كده اختار **واحد** للتطبيق وجرّب التاني. Bootstrap إطار UI جاهز، وTailwind أسلوب Utility Classes.

- Bootstrap — الكورس العربي الأصلي ↗
- Bootstrap — الكورس الإنجليزي الأصلي ↗
- Tailwind CSS — التوثيق الرسمي ↗
- Roadmap Projects — ملف المشاريع الأصلي ↗

تحدّي المرحلة

نفّذ **10 صفحات أو Components مختلفة** غير مشاريع الكورس، وراجع الموبايل والتابلت والديسكتوب. متدخلش JavaScript قبل ما تقدر تنفّذ Layout محترم.

المرحلة 03 / مهارة شغل

Git & GitHub

دي أول إضافة كبيرة على الطريق الأصلي: مهارة يومية لأي Developer بيشتغل ضمن فريق.

من هنا ابدأ تستخدم Git في كل مشروع، حتى لو مشروع بسيط. في الشغل الحقيقي، Version Control مش اختيار.

لازم تتعلم

- إيه الفرق بين Git و GitHub و Repository.
- `init` ، `add` ، `commit` ، `status` ، `log` .
- `branch` ، `merge` ، `pull` ، `push` وحل تعارض بسيط.
- كتابة README واضح، وملف `.gitignore` ، وفتح Pull Request للمراجعة.

مصادر مجانية رسمية

Git — دروس + فيديوهات للمبتدئين ↗

GitHub Skills — تمارين عملية ↗

تطبيق عملي

خد مشروع CSS من اللي عملتهم: افتح له Repository، اعمل 3 Commits مفهومة، Branch لتعديل، Pull Request، وبعدها Merge.

في الشغل هتحتاج ده ليه؟

تقدر ترجع لتعديل قديم، تشتغل مع فريق، وتتابع التغييرات من غير ما تضيع الشغل أو تبعت ملفات باسم final_final_v2.

اتعلم Git بالتوازي مع باقي المراحل، مش لازم توقف التطور لحد ما تتقن كل أوامره.

المرحلة 04 / عقل المبرمج

JavaScript — أهم منعطف

لو المرحلة دي أخدت وقت أطول من المتوقع ده طبيعي. الفهم بييجي بالتجربة، مش بالمشاهدة بس.

هنا فعلاً هتبدأ تفكر كمبرمج. خد وقتك، ومتقلقش لو في حاجات لسه مش واضحة من أول مرة. المهم تكتب كود وتحل مشاكل.

المفاهيم الأساسية

- Variables، Data Types، Conditions، Loops و Functions.
- Arrays، Objects، Destructuring، Spread، Modules و ES6+.
- Scope، Closures، Array Methods و Error Handling.
- DOM Manipulation، Events، Validation و Local Storage.
- Promises، `async/await` ومبادئ Asynchronous JavaScript.

الكورسات

- JavaScript — الكورس العربي الأصلي ↗
- JavaScript — الكورس الإنجليزي الأصلي ↗
- JavaScript Guide — MDN ↗
- Roadmap Projects — ملف المشاريع الأصلي ↗

مشاريع المرحلة

To-Do List، Quiz App، Expense Tracker، أو Product Filter ببحث وترتيب. لازم تعرف تكتب المنطق بنفسك من غير ما تبص على الحل.

اختبار بسيط لنفسك

لو اتعرضت عليك فكرة جديدة، تقدر تقسّمها Functions، وتتعامل مع Events وأخطاء المستخدم، وتعرف تدور على المشكلة في Console.

المرحلة 05 / ربط البيانات

Browser APIs + REST APIs

الجزء ده هو اللي بينقل JavaScript من صفحات بسيطة لتطبيقات فيها بيانات حقيقية.

كتير من مشاريع الشغل بتعرض بيانات جاية من Backend. عشان كده لازم تفهم إزاي الـ Front-End بيكلم API بشكل آمن ومنظم.

تتعلم إيه؟

- فكرة Client / Server و HTTP Requests و Status Codes.
- طرق الطلب الأساسية `GET` و `POST` و `PUT/PATCH` و `DELETE` .
- JSON و `fetch()` و `async/await` .
- حالات Loading و Empty و Error و Retry بدل ما الشاشة تسكت.
- الفرق بين Authentication و Authorization، ومبدأ إن الـ Client مش مصدر ثقة.

مصادر جديدة

- Fetch & Network Requests — MDN ↗
- HTTP Basics — MDN ↗
- Chrome DevTools — فحص الأخطاء والشبكة ↗

تطبيق عملي

ابنِ صفحة منتجات فيها Pagination، Filters، Search، Loading، ورسالة واضحة عند فشل الـ API. جرّب طلبًا ناجحًا وآخر بيرجع Error.

أهم نقطة

متستخدمش بيانات سرية أو API Keys خاصة على الـ Front-End، ومتعتبرش إخفاء زر في الواجهة حماية حقيقية.

المرحلة 06 / تنظيم الكود

TypeScript + Dev Tooling

دي أدوات هتفرق في المشاريع الكبيرة، وفي سهولة قراءة وصيانة الكود.

بعد ما تبقى مرتاح في JavaScript، ابدأ تتعلم TypeScript وتفهم الأدوات اللي هتستخدمها في React.

TypeScript

- Primitive Types، Arrays، Objects، Unions و Type Narrowing.
- `type` و `interface` ، Typing Functions و Optional Properties.
- إزاي تكتب Type لبيانات بترجع من API وتقلل أخطاء الـ Runtime.
- بلاش تعتمد على `any` كل مرة تقف قدامك مشكلة.

الأدوات اللي هتحتاجها

- Node.js و npm لفهم Packages و `package.json` .
- Vite لتشغيل مشاريع JavaScript أو React محليًا.
- DevTools و Network Tab و Console لتتبّع المشاكل.

TypeScript Handbook — رسمي ↗

Vite — دليل البداية ↗

npm — دليل المفاهيم الأساسية ↗

مشروع صغير

حوّل مشروع JavaScript سابق لـ TypeScript: اكتب Types للـ Products والـ API Response وتعامل مع القيم الناقصة.

مش لازم تتقن Generics المعقدة قبل React؛ الأساسيات القوية كفاية للانتقال.

المرحلة 07 / الواجهات الحديثة

React.js — طريقة تفكير جديدة

خليك فاكر إن React مش بديل JavaScript. كل ما الأساس عندك أقوى، React هتبقى أسهل.

بعد ما تبقى فاهم JavaScript كويس، ادخل React. هنا هتبدأ تبني UI من Components صغيرة قابلة لإعادة الاستخدام.

الأساسيات التي لا تتخطّاها

- JSX، Components، Props، State و Events.
- Conditional Rendering، Lists & Keys و Forms.
- `useState` و `useEffect` مع فهم إمتى تحتاج Effect وإمتى لأ.
- Composition و Lifting State Up، ثم Custom Hooks.
- الفصل بين UI Component و Business Logic.

المصادر

- React — الكورس العربي الأصلي ↗
- React — الكورس الإنجليزي الأصلي ↗
- React Learn — التوثيق الرسمي ↗
- Thinking in React — تدريب عملي ↗

مشروع المرحلة

اعمل Task Management App: إضافة وتعديل وحذف، Local Storage، Forms، Filters، تقسيم Components، ورسائل خطأ واضحة.

قبل ما تعدّي

لازم تعرف تشرح ليه الـ Component اتعمل، الـ State مكانها فين، وإيه اللي بيحصل لما المستخدم يضغط على زر.

المرحلة 08 / تطبيقات أكبر

React — من Tutorial إلى Application

الفرق بين مشروع Demo وتطبيق إنتاجي هو تنظيم البيانات، الأخطاء، التجربة، وقابلية الصيانة.

المرحلة دي اختيارية في البداية، لكنها مهمة علشان تبدأ تعمل تطبيقات شبه اللي بتتبني جوه الشركات.

مفاهيم الشغل الحقيقي

- Routing: صفحات متعددة وتنقل وتعامُل مع 404.
- Context و Custom Hooks؛ متدخلش مكتبة Global State إلا لما تحتاجها فعلًا.
- Server State: Caching، Refetching، Mutations و Error States.
- Forms Validation، Component Architecture و Reusable UI.
- إعادة استخدام Components من غير Over-Engineering.

المصادر

- React Router — المسار التعليمي الرسمي ↗
- TanStack Query — إدارة Server State ↗
- React + TypeScript — دليل رسمي ↗

تطبيق واقعي

ابنِ Dashboard لطلبات متجر: جدول Orders، بحث وفلاتر، تفاصيل، Loading و Errors، وتنقل بين صفحات. تعامل مع API وهمي أو تجريبي.

فكرة هتوفر عليك وقت

مش كل State محتاجة Redux أو Zustand. اسأل: هي State محلية؟ مشتركة بين Components؟ ولا جاية من Server؟ وبعدها اختار الأداة.

المرحلة 09 / Next.js

Next.js — خطوة أقرب للـ Production

الهدف هنا إنك تفهم الـ Rendering والـ Routing وتعرف تختار الحل المناسب، مش تجمع Features لمجرد التجربة.

هنا نبدأ Next.js بعد React. ركّز على المفاهيم الأساسية للـ App Router، وتابع التوثيق الرسمي لأن تفاصيل Frameworks بتتطور بسرعة.

المفاهيم اللي تهمك

- File-based Routing، Nested Layouts و Dynamic Routes.
- Server Components و Client Components، وإمتى تستخدم كل واحد.
- Data Fetching، Loading UI، Error Boundaries و Caching حسب السيناريو.
- Metadata و SEO و تحسين الصور والخطوط.
- تأمين التعامل مع بيانات المستخدم، وفصل الكود اللي يشتغل على السيرفر.

مصادر التعلّم

- Next.js — الكورس الأصلي ↗
- Next.js Learn — مشروع Dashboard ↗
- Next.js App Router — التوثيق ↗
- Roadmap Projects — ملف المشاريع الأصلي ↗

المشروع النهائي

موقع شركة أو منصة صغيرة فيها صفحات عامة، Blog أو Dashboard، Products لإدارة المحتوى، SEO، Search و Deployment.

ملحوظة مهمة

لو فيديو قديم بيستخدم طريقة Routing أو API مختلفة، طبّق المفهوم وافتح الـ Docs عشان تتأكد من الأسلوب المناسب لنسختك.

المرحلة 10 / الجودة

Testing + Accessibility + Performance

الجزء اللي بيفرق بين واجهة شكلها حلو وواجهة فعلاً جاهزة لمستخدمين حقيقيين.

قبل ما تقول المشروع خلص، لازم تكون متأكد إنه شغّال صح، سريع، ومتاح لأكبر عدد من المستخدمين.

Testing — اختبر سلوك المستخدم

- Unit Tests للـ Functions المهمة باستخدام Vitest.
- Component Tests بتجربة أقرب لتفاعل المستخدم باستخدام React Testing Library.
- E2E Tests لسيناريوهات أساسية زي تسجيل الدخول وإضافة منتج باستخدام Playwright.

↗ Vitest — Getting Started

↗ React Testing Library

↗ Playwright — E2E Testing

Performance + Accessibility

- فهم LCP و CLS و INP، وتقليل تحميل JavaScript والصور الثقيلة.
- التحقق من Keyboard Navigation و Labels و Color Contrast و Focus.
- تجربة الموقع على موبايل حقيقي أو محاكي، ومع شبكة بطيئة.

↗ Web Performance — web.dev

↗ Accessibility — MDN

تطبيق عملي

اختبر حالة Critical في مشروعك، أصلح 3 مشاكل من Lighthouse، وجرب الموقع بالكامل بالكيبورد فقط.

المرحلة 10 / تسليم الشغل

Deployment + SEO + Security Basics

متفكرش في الشغل كمجرد كود على جهازك؛ فكر فيه كمنتج لازم يوصل للمستخدم بصورة سليمة.

الشغل ما بيخلصش عند `npm run dev`. لازم تبقى عارف تبني نسخة Production، تنشرها، وتراجعها بعد النشر.

قائمة النشر الأساسية

- شغّل Production Build وتأكد إن مفيش Errors.
- راجع Environment Variables؛ متحطش Secrets في كود المتصفح.
- اضبط عنوان الصفحة و Description و Open Graph.
- جرّب Refresh على الصفحات الداخلية وراجع 404.
- راجع الروابط، الصور، Form Errors، و responsive layouts.
- ارفع المشروع على GitHub وانشره على Hosting مناسب.

المصادر الجديدة

- Vercel — Deployments ↗
- Google — SEO Starter Guide ↗
- MDN — Web Security Basics ↗

Deliverable حقيقي

كل مشروع في الـ Portfolio لازم يكون له Live URL و GitHub Repository و README: بيحل إيه، اتعمل بإيه، وصور من الشغل.

اختبار الاستلام

ابعت رابط المشروع لشخص غيرك. لو عرف يفتحه ويستخدمه من الموبايل من غير تعليمات منك، أنت قربت جدًا من معنى الجاهزية.

تطبيق من غير تشتت

10 أفكار مشاريع مرتبة بالمستوى

دي أمثلة إضافية عملية، غير المشاريع الموجودة في ملف Google Drive الأصلي.

01	**Personal Profile** صفحة شخصية منظمة بالـ Semantic HTML.	HTML
02	**Landing Page** صفحة منتج Responsive بتقسيم واضح.	HTML / CSS
03	**Restaurant Menu** قائمة طعام بكروت وفلاتر شكلية.	CSS
04	**Pricing Section** ثلاث باقات بأسعار وتجاوب جيد.	CSS
05	**Dashboard Layout** Sidebar + Header + Cards على أحجام مختلفة.	CSS Grid
06	**To-Do App** CRUD + Local Storage.	JavaScript
07	**Quiz App** أسئلة وتقييم وTimer اختيارياً.	JavaScript
08	**Product Explorer** بحث وفلاتر وحالات تحميل وأخطاء.	JS + API
09	**Task Manager** Forms + Routing + State.	React / TS
10	**Business Dashboard** API + Auth UI + Tables + Deploy.	React / Next

ملف مشاريعك الأصلي

ما زال هو المرجع الرئيسي اللي تقدر ترجعله في كل مرحلة. دي قائمة إضافية تساعدك تختار مشاريع من غير تشتت:

افتح ملف Roadmap Projects الأصلي ↗

مش مطلوب تعمل كل المشاريع بنفس المستوى. اختار 3 أو 4 من الأقوى، واشتغل عليهم بجد لحد ما يبقوا مناسبين للعرض.

هدف الرحلة

من أول كورس لأول فرصة

التعلّم مش سباق. أهم حاجة تكون عارف تعمل إيه بإيدك، وتعرف تحكي عنه بثقة.

بعد ما تبني أساس قوي وتعمل مشاريع، جهّز نفسك لعرض شغلك بطريقة يفهمها Recruiter أو Client بسرعة.

Portfolio قوي محتاج إيه؟

- تعريف مختصر أنت بتعمل إيه، ومهاراتك الأساسية.
- من 3 إلى 4 مشاريع متقنة بدل 20 مشروع Tutorial.
- شرح المشكلة والحل ودورك في المشروع والتحديات.
- روابط Live و Code، و Screenshots واضحة.
- تصميم Responsive، وأداء جيد، وروابط شغالة.

قبل ما تقدم على شغل

- ☐ JavaScript و DOM و async/await و API requests
- ☐ HTML + CSS + Responsive بدون اعتماد كلي على Framework
- ☐ Git / GitHub و README و Pull Requests
- ☐ React Components و State و Forms و Routing
- ☐ Next.js و SEO و Deployment
- ☐ TypeScript أساسيات و Projects منظمة
- ☐ مشروعين على الأقل منشورين و قابلين للشرح
- ☐ Testing أساسيات و Accessibility و Performance

متنساش

كل شوية ارجع للمشاريع القديمة وحسّنها باستخدام اللي اتعلمته. ده بيطلع مستوى شغلك أسرع من إنك تبدأ مشروع جديد كل يوم.

قناتي على YouTube — الرابط الأصلي ↗

الـ Portfolio — الرابط الأصلي ↗

لو استفدت، متنساش تدعيلي وتصلي على النبي. وربنا يوفقك في كل خطوة. — **أحمد سمير**

ملحق A / لا تغيير في الروابط

كل روابط الكورسات الأصلية

14 عنوان URL فريدًا بالملف الأصلي، منهم 12 رابطًا للمصادر أو القناة أو الـ Portfolio، بالإضافة لرابطين تلقائيين.

الروابط دي تم نقلها **بنفس عناوين الـ URL الموجودة في ملف الـ PDF الأصلي حرفيًا**، من غير تبديل قوائم التشغيل أو حذف معاملات الرابط.

CSS — الكورس الأصلي

افتح الرابط الأصلي ↗

https://youtube.com/playlist?list=PLknwEmKsW8OuN04Odt2sJqt4aAnkp-iYA&si=3jDEwPDTg7Xl_s9N

HTML — الكورس الأصلي

افتح الرابط الأصلي ↗

https://youtu.be/Dv39fDYei9A?si=f9_EAWVYPUN1QMQ9

Bootstrap — English

افتح الرابط الأصلي ↗

https://youtube.com/playlist?list=PL4cUxeGkcC9joIM91nLzd_qaH_AimmdAR&si=iK_qnqJjMlauE_IO

Bootstrap — عربي

افتح الرابط الأصلي ↗

https://youtube.com/playlist?list=PLDoPjvoNmBAyvm7f--dc6XqkpfDcen_vQ&si=NMYSdq1TW4ULBv67

JavaScript — عربي

افتح الرابط الأصلي ↗

https://youtube.com/playlist?list=PLknwEmKsW8OuTqUDaFRBiAViDZ5uI3VcE&si=xMr0F2uToqXm3usj

Roadmap Projects — Google Drive

افتح الرابط الأصلي ↗

https://drive.google.com/file/d/1I6FfnocDC_o31gRmQ36Cgxt5txovjBNq/view?usp=sharing

React — كورس عربي

افتح الرابط الأصلي ↗

https://youtube.com/playlist?list=PLYyqC4bNbCIdSZ-JayMLl4WO2Cr995vyS&si=d8Ln19Ut-ONCcR8i

JavaScript — English

افتح الرابط الأصلي ↗

https://youtube.com/playlist?list=PL4cUxeGkcC9haFPT7J25Q9GRB_ZkFrQAc&si=tv-stBEwegKhldII

Next.js — الكورس الأصلي

افتح الرابط الأصلي ↗

https://youtube.com/playlist?list=PLxbVBWjVdAEjcc65Mk0-y52yVX6yb3ZAS&si=E0rFYeZqCSQ5S2YZ

React — كورس إنجليزي

افتح الرابط الأصلي ↗

https://youtube.com/playlist?list=PL4cUxeGkcC9gZD-Tvwfod2gaISzfRiP9d&si=FYF22Q92I2mtPuLk

Portfolio

افتح الرابط الأصلي ↗

https://my-portfolio-new-b84u.vercel.app/

قناة YouTube

افتح الرابط الأصلي ↗

https://www.youtube.com/@alaydecoding

معلومة عن ملفك الأصلي

كان فيه كمان رابطان تلقائيان غير تعليميين نتيجة تحويل أسماء التقنيات إلى URLs. احتفظنا بيهم في الصفحة التالية كما هما، لكن لا ننصح باستخدامهم كمصادر.

ملحق B / المصادر المضافة

مصادر إضافية — الجزء الأول

دي الروابط الجديدة لتعلم الأدوات والمهارات اللي اتضافت على ترتيبك الأصلي.

المصادر الجديدة دي **إضافات** للنسخة القديمة. أغلبها توثيق رسمي تقدر ترجعله لما تقف معاك نقطة، ومش لازم تقرأه كله مرة واحدة.

البداية وتصميم الواجهات

- MDN — HTML ↗
- MDN — مسار تعلّم الويب ↗
- Accessibility — MDN ↗
- Responsive Design — MDN ↗
- Bootstrap Docs ↗
- Tailwind CSS ↗

Git & GitHub والأدوات

- GitHub Skills ↗
- Git Learn ↗
- npm Basics ↗
- Chrome DevTools ↗
- Vite Guide ↗

JavaScript & APIs

- Network Requests & Fetch ↗
- JavaScript Guide — MDN ↗
- HTTP Basics ↗

الرابطان التلقائيان من PDF الأصلي

تم الاحتفاظ بهما حرفيًا لأغراض التطابق مع الملف الأصلي، مع التنبيه إنهما ليسا روابط توثيق رسمية:

- http://react.js ↗
- http://next.js ↗

ملحق C / المصادر المضافة

مصادر إضافية — الجزء الثاني

المراجع الرسمية للتطبيقات الحديثة ومرحلة الجودة والتسليم.

ركز الأول على الـ Core Skills، وبعدها ارجع للمصادر المتقدمة في الوقت المناسب. Animation Libraries مثل Motion اختيارية، ومش بديل لفهم CSS Animations.

TypeScript & React

- ↗ React Learn
- ↗ TypeScript Handbook
- ↗ React Router
- ↗ Thinking in React
- ↗ React + TypeScript
- ↗ TanStack Query

Next.js & Delivery

- ↗ Next.js Official Docs
- ↗ Next.js Learn — App Router
- ↗ Google SEO Guide
- ↗ Vercel Deployments

Testing, Performance & Extras

- ↗ React Testing Library
- ↗ Vitest
- ↗ web.dev Performance
- ↗ Playwright
- ↗ Motion for React — اختياري
- ↗ Web Security — MDN

آخر نصيحة

خد كورس واحد للمرحلة، ومرجع واحد لما تحتاج تفهم بعمق، ونفّذ مشروع. كفاية مصادر على قد ما تكفيك تتحرك، مش على قد ما تقدر تجمع Links.

بالتوفيق في رحلتك — AHMED SAMIR / ALAYD CODING