predev. Academy

خريطة المسار

# تطوير الويب باستخدام React و Next.js

تبني وتنشر تطبيق ويب حقيقيا فيه تسجيل دخول ولوحة تحكم ودعم للعربية والإنجليزية واختبارات، باستخدام React 19 و Next.js 15.

واجهات الويب

| المدة | المحاضرات | المواعيد | طريقة الحضور | المستوى | لغة الشرح |
|---|---|---|---|---|---|
| 10 أسابيع | 20 محاضرة مباشرة، 60 ساعة | مرتين في الأسبوع، 3 ساعات لكل محاضرة، مع ساعة office hours أونلاين كل أسبوع | حضوري أو أونلاين، لايف | مبتدئ، بعد Tech Compass | العربية المصرية، مع المصطلحات التقنية بالإنجليزية |

### لمن هذا المسار

- طلاب علوم الحاسب ونظم المعلومات الذين يريدون مهارة frontend حقيقية قبل التخرج
- الخريجون الجدد الذين يعرفون أساسيات البرمجة ويبحثون عن أول وظيفة في الويب
- من يغيّرون مجالهم بعد إنهاء Tech Compass واختيار تطوير الويب
- المصممون ومن يعملون على WordPress ويريدون كتابة كود React حديث

### قبل اليوم الأول

- إنهاء مقدمة Tech Compass، أو إثبات معرفة أساسيات البرمجة (المتغيرات، الشروط، التكرار، الدوال)
- لابتوب بذاكرة 8 GB RAM على الأقل (ويفضل 16 GB)، وقرص SSD، ونظام Windows 10/11 أو macOS أو Linux
- تثبيت قبل اليوم الأول: Node.js 24 LTS و Git و VS Code و Google Chrome
- حسابات مجانية: GitHub و Vercel وحساب Google للملفات المشتركة
- القدرة على قراءة توثيق بسيط بالإنجليزية (والشرح يكون بالعربية)

## في نهاية المسار تستطيع

- بناء صفحات accessible و responsive باستخدام HTML semantic و CSS حديث تعمل بالعربية RTL وبالإنجليزية
- كتابة JavaScript و TypeScript حديث ونظيف، بما في ذلك الكود الـ async والتعامل مع الـ API باستخدام fetch
- بناء واجهات React 19 بالـ components والـ props والـ state والـ effects والـ custom hooks
- تنظيم تطبيق Next.js 15 App Router بالـ layouts والـ nested routes والـ server و client components والـ server actions
- بناء نماذج فيها validation باستخدام React Hook Form و Zod، وإدارة بيانات الخادم باستخدام TanStack Query
- إضافة تسجيل الدخول والـ sessions وصفحات محمية حسب دور المستخدم
- إطلاق واجهة بالعربية والإنجليزية مع RTL layout صحيح
- كتابة اختبارات unit و component و end-to-end، واستخدامها شبكة أمان عند كتابة الكود بالـ AI
- قياس الـ Core Web Vitals وتحسينها وإضافة SEO metadata

- نشر تطبيق production على Vercel أو على VPS باستخدام Coolify، وعرضه في GitHub portfolio

## التقنيات المستخدمة

**أساسيات الويب**

| | |
|---|---|
| **HTML5** | هيكل semantic تفهمه محركات البحث وقارئات الشاشة |
| **CSS (Flexbox, Grid, logical properties)** | تصميمات responsive تعمل بالعربية RTL وبالإنجليزية LTR |
| **JavaScript (ES2024)** | لغة المتصفح ولغة أي تطبيق React |
| **TypeScript 5** | يكشف الأخطاء قبل التشغيل، ومطلوب في أغلب إعلانات الوظائف |

**الـ FRAMEWORK وواجهة المستخدم**

| | |
|---|---|
| **React 19** | أكثر مكتبة frontend مطلوبة في مصر وفي الوظائف عن بعد |
| **Next.js 15 App Router** | الـ routing والـ server rendering وجلب البيانات والـ server actions في framework واحد |
| **Tailwind CSS 4** | تنسيق سريع ومتسق تستخدمه الفرق يوميا |
| **shadcn/ui** | مكونات accessible تملكها وتستطيع تعديلها، مبنية على Radix |

**البيانات والنماذج وتسجيل الدخول**

| | |
|---|---|
| **TanStack Query 5** | الـ caching وحالات التحميل والـ mutations لبيانات الـ client |
| **React Hook Form + Zod** | نماذج سريعة بـ validation schema واحدة للـ client والـ server |
| **Auth.js** | تسجيل الدخول والـ sessions والصفحات المحمية دون كتابة auth من الصفر |
| **next-intl** | روابط ورسائل بالعربية والإنجليزية، وتنسيق الأرقام والتواريخ |

**الجودة والنشر**

| | |
|---|---|
| **Vitest + Testing Library** | اختبارات unit و component تعمل في ثوان |
| **Playwright** | اختبار end-to-end واحد لأهم مسار للمستخدم في متصفح حقيقي |
| **GitHub و Git** | إدارة النسخ والـ pull requests ومراجعة الكود كما في أي فريق حقيقي |
| **VPS على Coolify أو Vercel** | نشر حقيقي للعامة مع environment variables ونطاق خاص |
| **Cursor أو Claude Code** | كتابة الكود بمساعدة الـ AI مع مراجعة واختبارات وحدود واضحة |

## مشروع المسار: Mawid: تطبيق ويب لحجز العيادات مع لوحة تحكم

Mawid منصة حجز لسلسلة عيادات صغيرة: يختار المريض الفرع والطبيب والموعد، ويدير الموظفون كل شيء من لوحة تحكم. يشبه المشروع لوحات الحجز والإدارة التي تبنيها predev. Solutions لعملائها، لذلك يعمل الطلاب على نفس المشكلات التي يواجهها أي فريق حقيقي: تسجيل الدخول، الصلاحيات، النماذج، الجداول، الفلاتر، ومستخدمون بالعربية والإنجليزية.

- صفحة رئيسية عامة تعرض الفروع والأطباء والخدمات، مع SEO metadata
- إنشاء حساب وتسجيل دخول للمريض مع صفحات محمية
- مسار الحجز: اختيار الفرع والطبيب واليوم وموعد متاح
- صفحة حجوزاتي مع الإلغاء وتغيير الموعد
- لوحة تحكم للإدارة فيها بطاقات إحصائيات و chart بسيط
- جدول الحجوزات مع البحث والفلاتر والترتيب والـ pagination
- إدارة الأطباء والخدمات بنماذج للإضافة والتعديل والحذف
- صلاحيات حسب الدور: admin و receptionist و patient
- دعم كامل للعربية والإنجليزية مع RTL layout وزر لتغيير اللغة

- مظهر فاتح وداكن، وحالات التحميل والأخطاء، و toasts
- اختبارات unit و component واختبار Playwright لمسار الحجز
- نشر production مع environment variables ونطاق خاص

## مراحل المسار

الأسابيع 1-3
**الأساسيات**
كتابة HTML semantic و CSS responsive و JavaScript و TypeScript حديث، واستخدام Git و GitHub كل يوم.

الأسابيع 4-7
**الـ Framework الأساسي**
بناء تطبيق الحجز Mawid خطوة بخطوة باستخدام React 19 و Next.js 15 و App Router و Tailwind والنماذج وجلب البيانات وتسجيل الدخول والعربية والإنجليزية والاختبارات.

الأسابيع 8
**البناء باستخدام الـ AI**
استخدام أدوات الـ AI في كتابة الكود للعمل بسرعة أكبر، مع البقاء في موقع التحكم من خلال المراجعة والاختبارات والحدود الواضحة.

الأسابيع 9
**مشروع التخرج**
بناء تطبيق ويب خاص بك من brief ونشره على production.

الأسابيع 10
**أسبوع التوظيف**
الاستعداد للتقديم: CV و LinkedIn و GitHub portfolio والمقابلات والعمل الحر ويوم العرض.

## الخطة أسبوعًا بأسبوع

### الأساسيات · الأسابيع 1-3

#### الأسبوع 1 كيف يعمل الويب، HTML، و Git

**المحاضرة 1 الويب والمتصفح و HTML semantic**

- ماذا يحدث عند فتح URL: الـ DNS والـ HTTP request والـ response والـ rendering
- أدوات المتصفح DevTools: Elements و Console و Network
- هيكل مستند HTML، والـ head والـ meta tags، والخاصيتان lang و dir
- العناصر الـ semantic: header و nav و main و section و article و footer
- الروابط، والصور مع الـ alt text، والقوائم، والجداول
- النماذج: أنواع الـ input والـ labels والأزرار والـ validation المدمج

**تطبيق عملي في المحاضرة:** نبني صفحة semantic لعيادة فيها header وقائمة خدمات وجدول أطباء ونموذج تواصل، ونراجعها باستخدام الـ accessibility tree في DevTools.

**المحاضرة 2 Git و GitHub وعادات حل المشكلات**

- أساسيات الـ terminal: المجلدات والملفات وتشغيل الأوامر
- Git: init و add و commit و log و diff، وكتابة commit messages واضحة
- الـ branches والـ merge وحل merge conflict بسيط
- GitHub: الـ repos والـ push والـ pull وملفات README و GitHub Pages
- حل المشكلات: اقرأ المشكلة، واكتب الخطوات بكلام بسيط، ثم اكتب الكود
- كيف تبحث في التوثيق وتقرأ رسائل الأخطاء

**تطبيق عملي في المحاضرة:** نرفع صفحة العيادة إلى GitHub repo باستخدام feature branch و pull request، ثم ننشرها على GitHub Pages.

**الواجب:** ابن صفحة تعريف شخصية بـ HTML semantic، وارفعها على GitHub بخمسة commits واضحة على الأقل، وانشرها على GitHub Pages.

## الأسبوع 2 تنسيق CSS والتصميم الـ responsive والـ RTL

### المحاضرة 1 أساسيات CSS و Flexbox

- الـ selectors والـ specificity والـ cascade
- الـ box model والـ margin والـ padding والـ border والـ box-sizing
- الألوان والوحدات (rem و % و vw) والخطوط باستخدام web fonts
- Flexbox: الاتجاه والمحاذاة والـ wrapping والـ gap
- CSS variables للألوان والمسافات

**تطبيق عملي في المحاضرة:** ننسق صفحة العيادة: navbar وبطاقات الأطباء باستخدام Flexbox، مع نظام ألوان ومسافات في CSS variables.

### المحاضرة 2 Grid والتصميم الـ responsive والعربية RTL

- CSS Grid: الأعمدة والصفوف والـ areas و auto-fit مع minmax
- تصميم mobile first والـ media queries
- الـ logical properties: margin-inline و padding-block و inset-inline-start
- "dir="rtl والخطوط العربية وعكس الأيقونات بشكل صحيح
- أساسيات الـ accessibility: تباين الألوان، وشكل الـ focus، والـ labels الظاهرة

**تطبيق عملي في المحاضرة:** نحوّل صفحة العيادة إلى grid يعمل من عرض 360px حتى شاشة الحاسوب، ثم نحولها إلى العربية RTL دون أن يختل التصميم.

**الواجب:** أعد بناء صفحة landing لخدمة مصرية حقيقية (اختر واحدة) من screenshot، بحيث تكون responsive وتعمل RTL و LTR، وانشرها على GitHub Pages.

## الأسبوع 3 JavaScript الحديثة وأساسيات TypeScript

### المحاضرة 1 JavaScript التي ستستخدمها كل يوم

- let و const، والأنواع، والـ template strings، والقيم truthy و falsy
- الدوال والـ arrow functions والـ scope
- الـ arrays والـ objects: map و filter و reduce و find والـ spread والـ destructuring
- الـ modules: import و export
- الـ DOM: اختيار العناصر، والـ events، وتحديث الصفحة
- تمارين قصيرة في حل المشكلات على الـ arrays والـ strings

**تطبيق عملي في المحاضرة:** نبني صفحة بحث عن الأطباء بـ JavaScript فقط: تصفية قائمة الأطباء بالاسم والتخصص أثناء الكتابة.

### المحاضرة 2 JavaScript الـ async و fetch و TypeScript

- الـ event loop ببساطة، والـ Promises، و async و await
- fetch و JSON وأنواع الـ HTTP methods والـ status codes
- معالجة الأخطاء بـ try و catch، وحالات التحميل والنتائج الفارغة
- TypeScript: سبب وجوده، والأنواع الأساسية، و type و interface
- الـ union types والحقول الاختيارية وتحديد أنواع الـ parameters والـ return
- تشغيل TypeScript في مشروع Vite

**تطبيق عملي في المحاضرة:** نجلب الأطباء من JSON API حقيقي ونعرضهم مع حالات التحميل والخطأ، ثم نحول الكود إلى TypeScript مع أنواع واضحة لردود الـ API.

**الواجب:** حل 6 مسائل قصيرة في حل المشكلات بـ TypeScript (arrays و strings و objects)، وابن تطبيقا صغيرا typed يتصل بـ API عام ويعرض النتائج مع فلاتر.

## الأسبوع 4 أساسيات React 19

### المحاضرة 1 ال Components وال props و JSX

- لماذا React: ال components وال UI ال declarative
- إنشاء تطبيق React 19 باستخدام Vite و TypeScript
- قواعد JSX، وعرض القوائم مع ال keys، والعرض المشروط
- ال props وتحديد أنواعها بـ TypeScript
- تقسيم الصفحة إلى components صغيرة قابلة لإعادة الاستخدام
- React DevTools

**تطبيق عملي في المحاضرة:** نبني صفحة الأطباء في Mawid على شكل components: DoctorCard و DoctorList و SpecialtyBadge، مع props محددة الأنواع وبيانات تجريبية.

### المحاضرة 2 ال State وال events وال effects وال hooks

- useState والتعامل مع ال events
- رفع ال state إلى المكون الأب وتمرير ال callbacks
- useEffect: متى تستخدمه ومتى تتجنبه
- ال controlled inputs والنماذج البسيطة
- useRef و useContext وكتابة custom hook
- قواعد ال hooks والأخطاء الشائعة

**تطبيق عملي في المحاضرة:** نضيف بحثا وفلترا بالتخصص وقائمة أطباء مفضلين محفوظة في context، ونستخرج منطق التصفية في custom hook باسم useDoctorFilter.

**الواجب:** ابن component لاختيار الموعد يعرض المواعيد المتاحة لليوم المختار ويميز الموعد المحدد، مع props محددة الأنواع وتقسيم نظيف لل components.

**مرحلة في المشروع:** قائمة الأطباء واختيار الموعد في Mawid يعملان في React ببيانات تجريبية.

## الأسبوع 5 Next.js 15 App Router و Tailwind 4 و shadcn/ui

### المحاضرة 1 ال Routing وال layouts والتنسيق

- إنشاء مشروع Next.js 15 وهيكل مجلد app
- ال routing بالملفات وال dynamic routes وال route groups
- ال layouts و loading.tsx و error.tsx و not-found.tsx
- Tailwind CSS 4: ال utility classes وال theme tokens وال responsive variants
- shadcn/ui: إضافة Button و Card و Dialog و Table و Form
- المظهر الفاتح والداكن

**تطبيق عملي في المحاضرة:** ننقل Mawid إلى Next.js: layout عام، و layout للوحة التحكم فيه sidebar، وصفحة الأطباء وصفحة تفاصيل الطبيب، بتنسيق Tailwind و shadcn/ui.

### المحاضرة 2 ال Server وال client components وجلب البيانات

- الفرق بين ال server components وال client components وتوجيه use client
- جلب البيانات في ال server components باستخدام async و await
- أساسيات ال caching: ال rendering الثابت والديناميكي و revalidate
- ال route handlers لبناء API صغير
- ال streaming مع Suspense وال skeleton loaders
- ال environment variables في Next.js

**تطبيق عملي في المحاضرة:** نجلب الفروع والأطباء والخدمات من Mawid API داخل ال server components، ونضيف skeleton أثناء التحميل، ونعالج أخطاء ال API باستخدام error.tsx.

**الواجب:** ابن صفحة الخدمات وصفحة تفاصيل الطبيب باستخدام dynamic route و metadata وحالات تحميل وخطأ، وسلّمها في pull request للمراجعة.

**مرحلة في المشروع:** Mawid يعمل على Next.js بالصفحات العامة وهيكل لوحة التحكم، ويقرأ بيانات حقيقية من ال API.

## الأسبوع 6 النماذج والـ mutations وبيانات الـ client

### المحاضرة 1 النماذج باستخدام React Hook Form و Zod والـ server actions

- React Hook Form: register والأخطاء وحالة الإرسال
- Zod schemas واستنتاج أنواع TypeScript منها
- استخدام نفس الـ schema في الـ client والـ server
- الـ server actions للإضافة والتعديل والحذف
- useActionState وحالات الانتظار والـ toasts
- رسائل أخطاء و labels سهلة الوصول في النماذج

**تطبيق عملي في المحاضرة:** نبني نموذج الحجز (الفرع، الطبيب، اليوم، الموعد، الملاحظات) ونموذج إضافة طبيب للإدارة، مع validation في الطرفين باستخدام الـ server actions.

### المحاضرة 2 TanStack Query وجداول لوحة التحكم

- متى تستخدم TanStack Query في تطبيق Next.js
- الـ queries والـ query keys والـ caching
- الـ mutations والـ invalidation والـ optimistic updates
- البحث والفلاتر والترتيب والـ pagination باستخدام URL search params
- جداول البيانات باستخدام shadcn/ui Table
- تطبيق debounce على حقل البحث

**تطبيق عملي في المحاضرة:** نبني جدول الحجوزات للإدارة مع البحث وفلتر الحالة والترتيب والـ pagination محفوظة في الـ URL، مع أزرار إلغاء وتأكيد تستخدم optimistic updates.

**الواجب:** أكمل صفحة حجوزاتي للمريض مع الإلغاء وتغيير الموعد، وصفحة إدارة الخدمات مع الإضافة والتعديل والحذف.

**مرحلة في المشروع:** يستطيع المريض الحجز والإلغاء وتغيير الموعد، وتستطيع الإدارة إدارة الحجوزات والأطباء والخدمات.

## الأسبوع 7 تسجيل الدخول والعربية والإنجليزية والـ accessibility والاختبارات

### المحاضرة 1 تسجيل الدخول والصلاحيات وتعدد اللغات

- كيف تعمل الـ sessions والـ tokens، وأين نخزنها بأمان
- Auth.js مع تسجيل دخول credentials على Mawid API
- حماية الصفحات بالـ middleware والتحقق من الصلاحيات على الخادم
- صلاحيات admin و receptionist و patient
- next-intl: روابط حسب اللغة وملفات الرسائل وزر تغيير اللغة
- الـ RTL layout باستخدام الـ logical utilities في Tailwind، والتواريخ والأرقام بالعربية

**تطبيق عملي في المحاضرة:** نضيف تسجيل الدخول وإنشاء الحساب، ونحمي لوحة التحكم حسب الدور، ونترجم Mawid إلى العربية والإنجليزية مع RTL layout يعمل بشكل صحيح.

### المحاضرة 2 الـ Accessibility والاختبارات والأداء والـ SEO

- فحص الـ accessibility: التنقل بلوحة المفاتيح والـ focus وأساسيات ARIA و Lighthouse
- Vitest و Testing Library: اختبار الـ components والـ hooks
- اختبار النماذج والـ Zod schemas
- اختبار Playwright واحد end-to-end لمسار الحجز
- الـ Core Web Vitals: LCP و INP و CLS، و next/image و next/font
- الـ SEO: الـ Metadata API و Open Graph والـ sitemap والـ robots

**تطبيق عملي في المحاضرة:** نكتب اختبارات لمكون اختيار الموعد و schema الحجز ونموذج الحجز، ونضيف اختبار Playwright من تسجيل الدخول حتى تأكيد الحجز، ثم نصلح أهم ملاحظات Lighthouse.

**الواجب:** صل إلى 10 اختبارات ناجحة على الأقل، ودرجة accessibility في Lighthouse لا تقل عن 90، وترجمة عربية كاملة، و SEO metadata في كل صفحة عامة.

**مرحلة في المشروع:** Mawid مكتمل: تسجيل دخول وصلاحيات، عربية وإنجليزية مع RTL، اختبارات، و SEO، وجاهز للنشر.

## البناء باستخدام الـ AI · الأسابيع 8

### الأسبوع 8 البناء باستخدام الـ AI بالطريقة الصحيحة

#### المحاضرة 1 أدوات الـ AI للكود في مشروع حقيقي

- كيف يعمل Claude Code و Cursor: الـ context والـ prompts وملفات قواعد المشروع
- كتابة مهمة واضحة: الهدف، والملفات، والقيود، وطريقة التحقق من النتيجة
- طلب خطة قبل الكود، والعمل على خطوات صغيرة
- قراءة الـ diff سطرا سطرا قبل قبوله
- أخطاء الـ AI الشائعة في React و Next.js: حدود server و client خاطئة، و APIs قديمة، و packages غير موجودة
- إبعاد الأسرار وبيانات العملاء عن الـ prompts

**تطبيق عملي في المحاضرة:** نستخدم Claude Code أو Cursor لإضافة تقويم لمواعيد الأطباء في Mawid، ونراجع كل diff، ونرفض أو نصلح اقتراحين خاطئين على الأقل.

#### المحاضرة 2 الاختبارات شبكة أمان، ومتى لا تثق في الـ AI

- اكتب الاختبار أولا، ثم دع الـ AI ينفذ
- استخدام الـ AI في كتابة الاختبارات، والتأكد من أن الاختبارات يمكن أن تفشل
- إعادة تنظيم الكود بالـ AI مع بقاء الاختبارات ناجحة
- الـ debugging بالـ AI: أعطه الخطأ والكود وما جربته
- متى لا تثق فيه: تسجيل الدخول، والدفع، والأمان، والتراخيص
- شرح كل سطر تسلمه أثناء الـ code review

**تطبيق عملي في المحاضرة:** نكتب اختبارات فاشلة لميزة تذكير بالحجز، ونترك الـ AI ينفذها حتى تنجح الاختبارات، ثم نجري code review في أزواج ونشرح كل تغيير.

**الواجب:** سلّم ميزة جديدة في Mawid بمساعدة الـ AI من خلال pull request يتضمن الـ prompts المستخدمة والاختبارات المضافة، وملاحظة قصيرة عما رفضته ولماذا.

**مرحلة في المشروع:** في Mawid ميزة بُنيت بمساعدة الـ AI ومعها اختبارات و pull request تمت مراجعته.

## مشروع التخرج · الأسابيع 9

### الأسبوع 9 بناء مشروع التخرج ونشر حقيقي

#### المحاضرة 1 التخطيط والبناء

- اختيار الـ brief وتقليص الـ scope ليكتمل في أسبوع
- كتابة user stories وخريطة الصفحات والـ routes
- نموذج البيانات وعقد الـ API
- تجهيز الـ repo: TypeScript و linting و Prettier وهيكل المجلدات
- خطة يومية و GitHub issues كلوحة مهام

**تطبيق عملي في المحاضرة:** تعرض خطة مشروعك في 5 دقائق وتتلقى feedback، ثم تبني الصفحات الأساسية والمسار الرئيسي داخل المحاضرة مع متابعة المدرب.

#### المحاضرة 2 النشر على production

- النشر على Vercel: المشاريع والـ preview deployments والـ environment variables
- البديل: النشر على VPS باستخدام Coolify
- النطاق الخاص و HTTPS
- فحوصات الـ production: صفحات الأخطاء والـ logs والمتابعة الأساسية
- كتابة README جيد فيه screenshots وخطوات التشغيل

**تطبيق عملي في المحاضرة:** تنشر مشروع التخرج و Mawid على production مع environment variables ونطاق خاص أو subdomain، ثم تشغل اختبار Playwright على الرابط الحقيقي.

**الواجب:** أكمل مشروع التخرج: المسار الرئيسي يعمل من البداية للنهاية، والاختبارات ناجحة، ورابط يعمل، و README فيه screenshots، وفيديو demo مدته 3 دقائق.

**مرحلة في المشروع:** مشروع التخرج منشور ويعمل برابط عام و README كامل.

## الأسبوع 10 أسبوع التوظيف ويوم العرض

### المحاضرة 1 الـ CV و LinkedIn و GitHub والمقابلات

- CV من صفحة واحدة لـ frontend developer يجتاز أنظمة الـ ATS
- ملف LinkedIn: الـ headline وقسم about والمشاريع المميزة
- GitHub portfolio: الـ pinned repos و profile README وروابط تعمل
- موضوعات مقابلات الـ frontend: JavaScript و React و Next.js و CSS والمتصفح
- مقابلة تقنية تجريبية فيها live coding
- أولى خطوات العمل الحر: مستقل و Upwork، وتسعير مشروع صغير، وكتابة عرض

**تطبيق عملي في المحاضرة:** تحدّث الـ CV و LinkedIn وملف GitHub داخل المحاضرة بمراجعة المدرب، ثم تجري مقابلة تقنية تجريبية لمدة 30 دقيقة مع زميل.

### المحاضرة 2 يوم العرض

- تنظيم عرض للمنتج في 5 دقائق
- شرح قراراتك التقنية والـ tradeoffs
- الإجابة عن أسئلة المهندسين
- feedback من مهندسي predev. Solutions والضيوف
- خطة الخطوات القادمة والتخرج

**تطبيق عملي في المحاضرة:** تعرض مشروع التخرج مباشرة أمام مهندسي predev. Solutions والضيوف، وتجيب عن أسئلتهم، وتحصل على feedback مكتوب.

**الواجب:** قدّم على 10 وظائف frontend أو مشاريع عمل حر على الأقل خلال الأسبوعين التاليين، وشارك تقدمك في مجموعة الخريجين.

## مشروع التخرج

يبني كل طالب تطبيق ويب خاصا به وينشره، انطلاقا من brief يختاره مع المدرب. يجب أن يحل مشكلة حقيقية لمستخدمين في مصر، وأن يتضمن تسجيل دخول ولوحة تحكم، وأن يستخدم نفس الـ stack المستخدم في Mawid.

### المتطلبات

- Next.js 15 App Router مع TypeScript و Tailwind CSS 4 و shadcn/ui
- تسجيل دخول بدورين للمستخدمين على الأقل وصفحات محمية
- مسار واحد على الأقل للإضافة والتعديل والحذف مع validation باستخدام React Hook Form و Zod
- صفحة لوحة تحكم فيها جدول مع البحث والفلاتر والـ pagination
- العربية والإنجليزية مع RTL layout صحيح
- حالات التحميل والنتائج الفارغة والأخطاء في كل صفحة بيانات
- 8 اختبارات unit أو component على الأقل واختبار Playwright واحد end-to-end
- SEO metadata في الصفحات العامة ودرجة accessibility في Lighthouse لا تقل عن 90
- منشور على Vercel أو على VPS باستخدام Coolify، مع environment variables ورابط يعمل
- Git history نظيف و README فيه screenshots وخطوات التشغيل والرابط

### أفكار مقترحة

- اشتراكات صالة رياضية وحجز حصص مع لوحة تحكم للمدرب
- مركز دروس خصوصية: مجموعات وحصص وحضور للمدرسين وأولياء الأمور
- حجز طاولات في مطعم مع شاشة إدارة لمدير الفرع
- مركز صيانة سيارات: حجز موعد صيانة ومتابعة حالة السيارة
- إعلانات عقارات في كمبوندات مصر مع لوحة تحكم للوسيط العقاري
- فواتير العاملين المستقلين ومتابعة العملاء وحالة الدفع

## معايير تقييم مشروع التخرج

| | |
|---|---|
| يعمل من البداية للنهاية على الرابط الحقيقي | 25% |
| جودة الكود: الهيكل و TypeScript والـ components وتقسيم server و client | 20% |
| الـ UI والـ UX: responsive والعربية RTL والـ accessibility | 15% |
| التعامل الصحيح مع النماذج والبيانات وتسجيل الدخول | 15% |
| الاختبارات: unit و component واختبار end-to-end واحد | 10% |
| النشر والـ README والـ Git history | 10% |
| العرض في يوم العرض والإجابة عن الأسئلة | 5% |

## طريقة التقييم

| | |
|---|---|
| الواجب الأسبوعي<br>يسلَّم في صورة pull requests على GitHub ويراجَع بقائمة تحقق قصيرة، والتسليم المتأخر يحصل على نصف الدرجة | 20% |
| مراحل المشروع المستمر<br>مراحل Mawid في الأسابيع 4-8، تراجَع داخل المحاضرة مقابل قائمة المميزات | 25% |
| الاختبارات القصيرة ومراجعة الكود المباشرة<br>اختبارات قصيرة في الأسابيع 3 و 5 و 7، مع شرح كودك أثناء المراجعة | 10% |
| تكليف أسبوع الـ AI<br>pull request يتضمن الـ prompts والاختبارات وملاحظات عن الاقتراحات المرفوضة | 5% |
| مشروع التخرج<br>يقيَّم بمعايير مشروع التخرج | 30% |
| تسليمات أسبوع التوظيف<br>CV و LinkedIn وملف GitHub محدثة، والمقابلة التجريبية | 10% |

## قائمة المهارات

### HTML و CSS

- [ ] يكتب HTML semantic مع headings و labels و alt text صحيحة
- [ ] يبني تصميمات responsive باستخدام Flexbox و Grid بأسلوب mobile first
- [ ] يستخدم الـ logical properties ليعمل التصميم RTL و LTR

### JavaScript و TypeScript

- [ ] يستخدم دوال الـ arrays والـ destructuring والـ spread والـ modules
- [ ] يكتب كودا async باستخدام async و await ويعالج الأخطاء
- [ ] يحدد أنواع الـ props وردود الـ API والدوال باستخدام TypeScript

### React

- [ ] يقسم الواجهة إلى components صغيرة قابلة لإعادة الاستخدام مع props محددة الأنواع
- [ ] يدير الـ state والـ effects والـ context ويكتب custom hooks
- [ ] يتجنب الـ effects والـ re-renders غير الضرورية

### Next.js

- [ ] ينظم الـ routes والـ layouts وملفات التحميل والأخطاء في الـ App Router
- [ ] يختار بين الـ server components والـ client components
- [ ] يجلب البيانات على الخادم ويستخدم الـ server actions للتعديل
- [ ] يفهم أساسيات الـ caching والـ revalidation

### النماذج والبيانات وتسجيل الدخول

- [ ] يبني نماذج باستخدام React Hook Form و Zod على الـ client والـ server
- [ ] يستخدم TanStack Query للـ queries والـ mutations والـ invalidation
- [ ] يحمي الصفحات ويتحقق من الصلاحيات على الخادم

### الجودة والنشر

- [ ] يكتب اختبارات Vitest و Testing Library واختبار Playwright واحدا
- [ ] يفحص الـ accessibility والـ Core Web Vitals والـ SEO metadata
- [ ] يستخدم Git branches و pull requests، وينشر على Vercel أو Coolify
- [ ] يستخدم أدوات الـ AI مع المراجعة والاختبارات، ويستطيع شرح كل سطر

## بعد التخرج

**الشهور 1-3 بعد التخرج**

- التقديم على وظائف junior frontend و internship كل أسبوع
- تحسين Mawid ومشروع التخرج، وإضافة مشروع صغير آخر
- التدرب على أسئلة مقابلات JavaScript و React ومسائل برمجة قصيرة
- تنفيذ مشروع عمل حر صغير للحصول على أول تقييم من عميل

**الشهور 4-6**

- تعلم backend لواجهتك: Node.js 24 LTS مع REST API و PostgreSQL 17
- التعمق في الـ caching والـ server actions والأداء في Next.js
- المساهمة بإصلاحات صغيرة في مشروع React مفتوح المصدر
- تعلم إدارة الـ state للتطبيقات الأكبر، مثل Zustand

**الشهور 7-12: نحو مستوى mid-level**

- تولي features كاملة داخل فريق: التخطيط والكود والاختبارات والنشر
- تعلم أساسيات الـ CI وعادات الـ code review والمتابعة في الـ production
- تجربة React Native إذا أردت إضافة تطوير الموبايل
- البدء في مساعدة المبتدئين وكتابة منشورات تقنية قصيرة على LinkedIn

## الوظائف التي يقود إليها المسار

Junior Frontend Developer (React) | Junior Next.js Developer | React Developer في شركات البرمجيات والـ startups | Trainee أو Frontend Intern | Freelance Web Developer على مستقل و Upwork | Remote Junior Frontend Developer لشركات في الخليج وأوروبا | Junior Full Stack Developer بعد إضافة backend

## مصادر


- MDN Web Docs: HTML: https://developer.mozilla.org/en-US/docs/Web/HTML
- MDN Web Docs: CSS: https://developer.mozilla.org/en-US/docs/Web/CSS
- MDN Web Docs: JavaScript: https://developer.mozilla.org/en-US/docs/Web/JavaScript
- The Modern JavaScript Tutorial: https://javascript.info/
- TypeScript Handbook: https://www.typescriptlang.org/docs/
- توثيق React: https://react.dev/learn
- توثيق Next.js: https://nextjs.org/docs

- دورة Next.js Learn: https://nextjs.org/learn
- توثيق Tailwind CSS: https://tailwindcss.com/docs
- توثيق shadcn/ui: https://ui.shadcn.com/docs
- React Hook Form: https://react-hook-form.com/
- Zod: https://zod.dev/
- TanStack Query: https://tanstack.com/query/latest
- Auth.js: https://authjs.dev/
- next-intl: https://next-intl.dev/
- Vitest: https://vitest.dev/
- React Testing Library: https://testing-library.com/docs/react-testing-library/intro/
- Playwright: https://playwright.dev/
- web.dev: Core Web Vitals: https://web.dev/articles/vitals
- W3C: مقدمة في Web Accessibility: https://www.w3.org/WAI/fundamentals/accessibility-intro/
- توثيق Git: https://git-scm.com/doc
- GitHub Docs: https://docs.github.com/en
- توثيق Vercel: https://vercel.com/docs
- توثيق Coolify: https://coolify.io/docs/
- توثيق Claude Code: https://docs.anthropic.com/en/docs/claude-code/overview