predev. Academy

خريطة المسار

تطوير تطبيقات الموبايل باستخدام Flutter

تبني تطبيق Android و iOS حقيقي باستخدام Flutter وتختبره وتنشره، من أول سطر Dart حتى نسخة على مسار الاختبار الداخلي في المتاجر.

الموبايل

المدة	المحاضرات	المواعيد	المكان	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	العامية المصرية، والمصطلحات التقنية بالإنجليزية

لمن هذا المسار

- طلاب حاسبات وهندسة يريدون مهارة موبايل يمكنهم العمل بها قبل التخرج
- خريجون جدد يريدون أول وظيفة Junior Flutter في مصر أو عن بعد
- من يغيّرون مجالهم وأنهوا Tech Compass ويريدون بناء تطبيقات ينزّلها الناس من المتجر
- مطورو ويب يريدون إضافة تطبيقات الموبايل إلى الأعمال التي يقبلونها

قبل اليوم الأول

- إنهاء Tech Compass، أو معرفة أساسيات منطق البرمجة: المتغيرات والشروط والتكرار
- لابتوب بذاكرة 8 GB RAM على الأقل (يفضل 16 GB)، ومساحة فارغة 30 GB، ونظام Windows 10/11 أو macOS أو Linux
- موبايل Android وكابل USB، أو لابتوب يستطيع تشغيل Android emulator
- حسابات مجانية جاهزة قبل أول يوم: GitHub و Google (من أجل Firebase)
- برامج مثبتة قبل أول يوم: Git و VS Code و Flutter SDK و Android Studio (نساعدك في محاضرة الإعداد لو ظهرت مشكلة)
- جهاز Mac اختياري: بناء iOS ورفعه على TestFlight يعرضه المدرب عرضا عمليا

في نهاية المسار تستطيع

- تكتب كود Dart 3 نظيفا باستخدام null safety والـ classes و async/await و records و pattern matching
- تبني شاشات Flutter متجاوبة من تصميم جاهز مع Material 3 theming و dark mode
- تنظم التطبيق حسب الـ feature وتستخدم Riverpod لإدارة الحالة و go_router للتنقل
- تتصل بـ REST API باستخدام dio وتحول JSON إلى models محددة النوع وتتعامل مع حالات التحميل والفراغ والخطأ
- تنفذ تسجيل الدخول بالتوكن المحفوظ في secure storage مع شاشات محمية وتسجيل خروج

- ترسل إشعارات push وتتعامل معها باستخدام Firebase Cloud Messaging
- تطلق تطبيقا بالعربية والإنجليزية مع اتجاه RTL صحيح
- تكتب اختبارات unit و widget وتستخدمها لفحص الكود الذي يولده الذكاء الاصطناعي
- تجهز flavors للـ dev والـ production، وتوقّع نسخة release وترفعها على مسار الاختبار الداخلي في Play Console
- تعرض مشروعا منشورا على GitHub وفي المقابلة، وتجيب عن أسئلة Flutter الشائعة في المقابلات

التقنيات المستخدمة

اللغة والإطار

Dart 3	null safety والكود غير المتزامن و records و patterns تُستخدم في كل ملف Flutter
Flutter 3	كود واحد لـ Android و iOS، وتستخدمه أغلب فرق الموبايل التي توظف Juniors في مصر
Material 3	نظام التصميم الافتراضي في Flutter، ومعه theming و dark mode جاهزين

البيانات والشبكة

dio	HTTP client يدعم interceptors للتوكن والـ logging ومعالجة الأخطاء
json_serializable / freezed	توليد كود تحويل JSON و models غير قابلة للتعديل بدل كتابتها يدويا
flutter_secure_storage	حفظ access tokens في Keychain و Keystore بدل حفظها كنص عادي
shared_preferences / sqflite	حفظ الإعدادات وعمل cache للبيانات حتى يعمل التطبيق مع اتصال ضعيف

المعمارية وإدارة الحالة

Riverpod	الأداة الأساسية لإدارة الحالة في الكورس: سهلة الاختبار، آمنة وقت الترجمة، وتتعامل جيدا مع البيانات غير المتزامنة
Bloc (نظرة عامة)	منتشرة في الشركات المصرية، فتتعلم قراءتها وتعرف متى تختارها الفرق
go_router	حزمة التنقل الرسمية: named routes و deep links وتحويل المستخدم لتسجيل الدخول

الخدمات

Firebase (FlutterFire)	إعداد المشروع و Cloud Messaging (FCM) لإشعارات push
flutter_localizations / intl	العربية والإنجليزية، واتجاه RTL، والتواريخ والأرقام بالشكل الصحيح

الجودة والنشر

flutter_test / mocktail	اختبارات unit و widget تكتشف الأخطاء قبل أن يراها المستخدم
Flavors و --dart-define	نسخة dev ونسخة production منفصلتان بعناوين API ومشاريع Firebase مختلفة
Google Play Console / App Store Connect	التوقيع ومسارات الاختبار الداخلي و TestFlight
Git و GitHub	التحكم في النسخ يوميا، و pull requests، والبورتفوليو العام الخاص بك
Cursor أو Claude Code	أدوات برمجة بالذكاء الاصطناعي تُستخدم مع المراجعة والاختبارات كما تستخدمها الفرق الحقيقية

مشروع المسار: Sehha: تطبيق حجز العيادات

من الأسبوع 4 إلى الأسبوع 8 تبني تطبيقا واحدا: تطبيق للمرضى لحجز مواعيد الأطباء، قريب من تطبيق الرعاية الصحية الذي أطلقته predev باسم NileMed ومن بنية تطبيقاتنا الأخرى على المتاجر مثل Rakeez و iSpeaker. اخترناه لأنه يغطي ما يطلبه العملاء في المشاريع الحقيقية: تسجيل دخول، وقوائم من API، وبحث، وخطوات حجز، وإشعارات push، وواجهة عربية أولا.

- إنشاء حساب وتسجيل دخول وخروج مع حفظ التوكن في secure storage
- شاشة رئيسية بالتخصصات والأطباء المميزين من REST API
- قائمة أطباء مع بحث وفلاتر و pagination و pull to refresh
- صفحة الطبيب مع المواعيد المتاحة
- خطوات الحجز مع التحقق من بيانات النموذج وشاشة تأكيد
- شاشة مواعيدي بالحجوزات القادمة والسابقة مع إمكانية الإلغاء
- إشعار push لتأكيد الحجز يفتح الشاشة الصحيحة
- العربية والإنجليزية مع دعم كامل لـ RTL، و theme فاتح وداكن
- Offline cache لآخر قائمة أطباء تم تحميلها وحفظ إعدادات المستخدم
- Flavors لـ dev والـ production، واختبارات unit و widget، ونسخة Android release موقعة

مراحل المسار

الأسابيع 1-3
الأساسيات
تكتب كود Dart صحيحا، وتستخدم Git يوميا، وتحل مسائل صغيرة قبل أن تبدأ واجهات Flutter.

الأسابيع 4-7
إطار العمل الأساسي
تبني تطبيق Sehha لحجز العيادات خطوة بخطوة باستخدام Flutter و Riverpod و go_router و dio و Firebase.

الأسابيع 8
البناء باستخدام الذكاء الاصطناعي
تستخدم أدوات البرمجة بالذكاء الاصطناعي لتسريع العمل على كود حقيقي، مع فحص كل تغيير بالمراجعة والاختبارات.

الأسابيع 9
مشروع التخرج
تبني تطبيقك الخاص من وصف مختصر وتطلقه على مسار اختبار داخلي حقيقي.

الأسابيع 10
أسبوع التوظيف
تحوّل مشاريعك إلى مقابلات عمل وأول عملاء في العمل الحر.

الخطة أسبوعًا بأسبوع

الأساسيات · الأسابيع 1-3

الأسبوع 1 أساسيات Dart وتجهيز بيئة العمل

المحاضرة 1 الإعداد وأول برامج Dart

- كيف يعمل تطبيق Flutter: كود Dart والـ engine و Android و iOS
- تثبيت Flutter و Dart والتأكد منهما باستخدام flutter doctor
- إعداد VS Code: إضافات Dart و Flutter، والـ formatter، والاختصارات
- المتغيرات var و final و const والأنواع الأساسية
- النصوص والـ interpolation والعمليات
- تشغيل الكود في DartPad ومن الـ terminal

تطبيق عملي في المحاضرة: تبني برنامج command line لتقسيم الفاتورة، يقرأ الإجمالي وعدد الأشخاص ويطبع نصيب كل شخص مع البقشيش.

المحاضرة 2 التحكم في التدفق والدوال وأساسيات Git

- if/else و switch والتكرار
- الدوال: المعاملات المسماة والاختيارية، و arrow syntax
- Lists و sets و maps وأهم الدوال الخاصة بها (map و where و fold)
- ما هو Git: الـ repository والـ commit والـ history
- git init و add و commit و log وأول push على GitHub
- كتابة README قصير

تطبيق عملي في المحاضرة: تكتب برنامج درجات الطلاب الذي يحسب المتوسط والتقدير من map، مع commit لكل خطوة ثم push على repository جديد في GitHub.

الواجب: تحل 5 تمارين Dart قصيرة (مسائل نصوص وقوائم) في repository واحد، مع commit لكل تمرين، وترسل رابط GitHub.

الأسبوع 2 البرمجة الكائنية في Dart و null safety

المحاضرة 1 الـ Classes والـ Objects

- الـ classes والحقول والـ constructors والـ named constructors
- Getters و setters والأعضاء الخاصة
- الوراثة والـ abstract classes والـ interfaces
- Mixins و extension methods
- Enums بحقول ودوال
- factory constructors وكتابة fromJson و toJson يدويا

تطبيق عملي في المحاضرة: تصمم سلة صيدلية: classes باسم Product و CartItem و Cart مع إضافة وحذف وحساب الإجمالي، و fromJson constructor لـ Product.

المحاضرة 2 Null safety و records و patterns

- الأنواع nullable و non nullable، و ? و ! و ?? و late
- Records لإرجاع أكثر من قيمة
- Pattern matching و switch expressions
- sealed classes لتمثيل الحالات
- الاستثناءات: try/catch/finally والاستثناءات المخصصة
- قراءة أخطاء الـ compiler وإصلاحها

تطبيق عملي في المحاضرة: تمثل حالة الطلب كـ sealed class (Pending و Shipped و Delivered و Cancelled) وتطبع رسالة لكل حالة باستخدام switch expression.

الواجب: تبني نظام مكتبة صغيرا في Dart (كتب وأعضاء واستعارات) باستخدام classes و sealed result type لمحاولات الاستعارة وبدون أخطاء null. ترفعه مع README يشرح طريقة التشغيل.

الأسبوع 3 Dart غير المتزامن، وطريقة العمل بـ Git، وأول تطبيق Flutter

المحاضرة 1 الكود غير المتزامن وحل المسائل

- لماذا تحتاج التطبيقات كودا غير متزامن: الشبكة والملفات والمؤقتات
- Future و async/await ومعالجة الأخطاء
- Streams والاستماع للأحداث
- تحويل JSON من API عام داخل برنامج Dart
- تقسيم المسألة إلى خطوات قبل كتابة الكود
- أنماط شائعة: العد والبحث والترتيب والتجميع باستخدام maps

تطبيق عملي في المحاضرة: تكتب برنامج Dart يجلب posts من JSON API عام، ويجمعها حسب المستخدم، ويطبع أعلى 3 مستخدمين في عدد الـ posts.

المحاضرة 2 Git مع فريق وأول تطبيق Flutter

- الـ branches والـ merge وحل conflict بسيط
- Pull requests و code review على GitHub
- ملف .gitignore. وعادات كتابة رسائل الـ commit
- إنشاء مشروع Flutter وفهم بنية المجلدات
- التشغيل على emulator وعلى موبايل حقيقي، و hot reload و hot restart
- شجرة الـ widgets: MaterialApp و Scaffold و Text و Column

تطبيق عملي في المحاضرة: في مجموعات من اثنين، تنشئان تطبيق Flutter على repository مشترك، ويضيف كل طالب جزءا من الشاشة على branch، ثم يفتح pull request ويراجع pull request زميله.

الواجب: تحل 3 مسائل problem solving بلغة Dart من القائمة المرفقة، وتبني تطبيق Flutter من شاشة واحدة لبطاقة تعريف شخصية (صورة واسم ونبذة وروابط). ترفع الاثنين وتفتح pull request للمراجعة.

الأسبوع 4 الـ Widgets والتخطيط والتنقل

المحاضرة 1 الـ Widgets والتخطيط

- StatelessWidget و StatefulWidget، ومتى يكفي setState
- Row و Column و Stack و Expanded و Flexible و Spacer
- كيف تعمل الـ constraints: لماذا يحدث overflow وكيف تصلحه
- ListView و GridView والـ builders للقوائم الطويلة
- الصور والأيقونات والخطوط المخصصة والـ assets
- تقسيم الشاشة إلى widgets صغيرة قابلة لإعادة الاستخدام

تطبيق عملي في المحاضرة: تبني الشاشة الرئيسية لـ Sehha من تصميم Figma: شريط بحث، وشبكة تخصصات، وقائمة بطاقات أطباء ببيانات ثابتة.

المحاضرة 2 الـ Theme وبنية المشروع و go_router

- Material 3 ThemeData و ColorScheme وأنماط النصوص
- Theme فاتح وداكن من مكان واحد
- تخطيط متجاوب باستخدام MediaQuery و LayoutBuilder
- بنية مجلدات حسب الـ feature لتطبيق حقيقي
- go_router: الـ routes والـ path parameters والـ nested routes و shell لـ bottom navigation
- تمرير البيانات بين الشاشات

تطبيق عملي في المحاضرة: تضيف شاشة الطبيب وشاشة المواعيد، وتربطهما بـ go_router و bottom navigation bar، وتنقل كل الألوان والخطوط إلى الـ theme.

الواجب: تنهي الواجهات الثابتة لشاشات تسجيل الدخول وإنشاء الحساب والحجز بنفس الـ theme. تسجل فيديو للشاشة مدته 30 ثانية وتضيفه إلى الـ README.

مرحلة في المشروع: واجهات Sehha الثابتة: كل الشاشات الرئيسية جاهزة ويمكن التنقل بينها باستخدام go_router، مع theme فاتح وداكن.

الأسبوع 5 إدارة الحالة باستخدام Riverpod و REST APIs

المحاضرة 1 Riverpod لحالة التطبيق

- الحالة المحلية مقابل حالة التطبيق، ولماذا لا يكفي setState بعد حد معين
- Provider و Notifier و AsyncNotifier
- ref.watch و ref.read و ref.listen
- AsyncValue: حالات التحميل والبيانات والخطأ في الواجهة
- Family و autoDispose providers
- Bloc باختصار: كيف تقارن بـ Riverpod وأين ستقابلها

تطبيق عملي في المحاضرة: تنقل قائمة الأطباء والفلاتر إلى Riverpod providers، مع فلتر تخصص يحدّث القائمة بدون setState.

المحاضرة 2 التعامل مع REST API باستخدام dio

- أساسيات HTTP: الـ methods والـ status codes والـ headers و JSON bodies
- إعداد dio: base options والـ timeouts و logging interceptor
- Models باستخدام json_serializable و freezed و build_runner
- Repository pattern بين الـ API والواجهة
- معالجة الأخطاء: عدم وجود إنترنت، والـ timeout، و 4xx و 5xx
- حالات تحميل وفراغ وخطأ يفهمها المستخدم

تطبيق عملي في المحاضرة: تستبدل البيانات الثابتة بـ Sehha practice API: تحمّل التخصصات والأطباء من خلال repository وتعرض حالات التحميل والفراغ والخطأ.

الواجب: تضيف بحثا مع debounce و pagination (تحميل المزيد عند التمرير) لقائمة الأطباء، مع pull to refresh. تعمل commit على branch وتفتح pull request.

مرحلة في المشروع: Sehha يحمّل بيانات حقيقية: التخصصات والأطباء والبحث والـ pagination من الـ API عن طريق Riverpod.

الأسبوع 6 النماذج وتسجيل الدخول والتخزين المحلي

المحاضرة 1 النماذج وتسجيل الدخول بالتوكن

- Form و TextFormField والـ validators والتحكم في الـ focus
- عرض أخطاء التحقق القادمة من السيرفر بجانب الحقل الصحيح
- تسجيل الدخول وإنشاء الحساب باستخدام access و refresh tokens
- حفظ التوكن باستخدام flutter_secure_storage
- Interceptor في dio يضيف التوكن ويتعامل مع 401
- Redirect في go_router لحماية الشاشات، وتسجيل الخروج

تطبيق عملي في المحاضرة: تبني تسجيل الدخول وإنشاء الحساب على الـ API، وتحفظ التوكن بشكل آمن، وتُبقي المستخدم مسجلا بعد إعادة التشغيل، وتحوّل الزائر إلى شاشة الدخول.

المحاضرة 2 خطوات الحجز والبيانات بدون إنترنت

- حجز على عدة خطوات: اختيار اليوم، ثم الموعد، ثم التأكيد
- Date و time pickers وتنسيق المدخلات
- طلبات POST وتحديث الواجهة بعد النجاح
- shared_preferences لإعدادات مثل الـ theme واللغة
- Offline cache بسيط باستخدام sqflite لآخر قائمة أطباء
- Snackbars و dialogs وشاشات التأكيد

تطبيق عملي في المحاضرة: تبني خطوات الحجز وشاشة مواعيدي مع الإلغاء، وتعمل cache لقائمة الأطباء حتى تظهر والموبايل بدون إنترنت.

الواجب: تضيف شاشة profile يعدّل فيها المستخدم الاسم ورقم الموبايل مع التحقق، وشاشة إعدادات تحفظ اختيار الـ theme. تكتب 3 حالات اختبار ستحولها إلى اختبارات آلية الأسبوع القادم.

مرحلة في المشروع: Sehha يعمل من البداية للنهاية لمستخدم مسجل: دخول، وتصفح، وحجز، وعرض المواعيد وإلغاؤها، مع offline cache.

الأسبوع 7 Firebase و flavors والعربية والاختبارات

المحاضرة 1 إشعارات push و flavors

- إعداد مشروع Firebase باستخدام FlutterFire CLI
- Firebase Cloud Messaging: الـ tokens والصلاحيات على Android و iOS
- التعامل مع الإشعار والتطبيق مفتوح أو في الخلفية أو مغلق
- فتح الشاشة الصحيحة عند الضغط على الإشعار
- Flavors للـ dev والـ production بأسماء ومعرّفات تطبيق مختلفة
- قيم البيئة باستخدام --dart-define، وإبعاد الأسرار عن Git

تطبيق عملي في المحاضرة: تربط Sehha بـ Firebase، وتستقبل إشعار تأكيد حجز مرسلا من Firebase console، وتفتح تفاصيل الموعد عند الضغط عليه. تنشئ flavors للـ dev والـ prod.

المحاضرة 2 العربية و RTL والاختبارات

- flutter_localizations و gen-l10n وملفات ARB
- تخطيط RTL: Directionality، واستخدام start/end بدل left/right، وعكس الأيقونات
- التواريخ والأرقام وصيغ الجمع باستخدام intl
- Unit tests للـ repositories والـ notifiers باستخدام mocktail
- Widget tests: إيجاد الـ widgets والضغط عليها والتحقق من النتيجة
- عمل override لـ Riverpod providers داخل الاختبارات

تطبيق عملي في المحاضرة: ترجم Sehha إلى العربية مع زر لتغيير اللغة، وتصلح كل مشاكل تخطيط RTL، ثم تكتب unit tests لـ auth repository و widget test لشاشة الدخول.

الواجب: تصل إلى 10 اختبارات ناجحة على الأقل في التطبيق، وتترجم كل النصوص المتبقية، وتصلح تحذيرات الـ lint. تعمل tag للـ repository باسم v1.0.0.

مرحلة في المشروع: Sehha v1.0: إشعارات push، والعربية والإنجليزية مع RTL، و flavors للـ dev والـ prod، ومجموعة اختبارات ناجحة.

البناء باستخدام الذكاء الاصطناعي · الأسابيع 8

الأسبوع 8 أدوات الذكاء الاصطناعي بالطريقة الصحيحة

المحاضرة 1 العمل مع Claude Code أو Cursor

- ما الذي تجيده أدوات الذكاء الاصطناعي وأين تخطئ في مشاريع Flutter
- إعطاء السياق: ملف قواعد للمشروع، وبنية المجلدات، والأنماط الموجودة
- طلب خطة أولا ثم تغييرات صغيرة
- قراءة الـ diff سطرا بسطر قبل قبوله
- التأكد من أسماء الحزم وإصداراتها على pub.dev
- إبعاد مفاتيح الـ API والبيانات الشخصية عن الـ prompts

تطبيق عملي في المحاضرة: تستخدم Claude Code أو Cursor لإضافة ميزة المفضلة إلى Sehha (حفظ الأطباء وعرضهم وحفظهم محليا) بنفس أنماط المشروع، ثم تراجع الـ diff مع زميل وترفض أي جزء لا تستطيع شرحه.

المحاضرة 2 الاختبارات كشبكة أمان، ومتى لا تثق في الذكاء الاصطناعي

- كتابة الاختبار أولا ثم طلب الكود من الذكاء الاصطناعي
- اكتشاف Flutter APIs القديمة والدوال غير الموجودة
- تتبع الأخطاء مع الذكاء الاصطناعي: إعطاؤه الخطأ والـ stack trace والكود
- إعادة هيكلة الكود بالذكاء الاصطناعي مع بقاء الاختبارات ناجحة
- فحص الأمان والتراخيص في الكود المولد
- شرح مشروع التخرج ونطاقه والتخطيط له

تطبيق عملي في المحاضرة: تصلح 5 أخطاء مزروعة في نسخة من Sehha: تكتب اختبارا فاشلا لكل خطأ، وتستخدم الذكاء الاصطناعي لاقتراح الإصلاح، وتحتفظ فقط بالإصلاحات التي تنجح في الاختبارات والمراجعة.

الواجب: تكتب خطة مشروع التخرج في صفحة واحدة: الشاشات والـ API والـ data models والحزم وجدول يومي. تعتمدها في الـ office hours.

مرحلة في المشروع: دمج ميزة المفضلة بعد المراجعة ومعها اختبارات، واعتماد خطة مشروع التخرج.

مشروع التخرج · الأسابيع 9

الأسبوع 9 بناء مشروع التخرج وإطلاقه

المحاضرة 1 إطلاق Android ويوم البناء

- أيقونة التطبيق والـ splash screen واسم التطبيق لكل flavor
- الإصدارات: version name و build number
- إنشاء upload keystore وإعدادات التوقيع، والحفاظ عليه
- بناء Android App Bundle (AAB)
- Play Console: إنشاء التطبيق، ومسار الاختبار الداخلي، وقائمة المختبرين
- أساسيات صفحة المتجر: الصور والوصف وسياسة الخصوصية ونموذج data safety

تطبيق عملي في المحاضرة: توقّع نسخة release من تطبيق مشروع التخرج، وتشاركها مع المختبرين عبر Firebase App Distribution، وتتابع مع المدرّب رفعًا حقيقيًا على Play Console.

المحاضرة 2 شرح iOS عمليا ومراجعة مشاريع التخرج

- ما يحتاجه إطلاق iOS: جهاز Mac و Xcode وحساب Apple Developer
- Bundle ID والـ certificates والـ provisioning باختصار
- عمل archive والرفع على App Store Connect
- المختبرون الداخليون والخارجيون في TestFlight
- أسباب الرفض الشائعة في المراجعة وكيف تتجنبها
- Code review لمشاريع التخرج في مجموعات صغيرة

تطبيق عملي في المحاضرة: تشاهد رفعا مباشرا على TestFlight يقوم به المدرب، ثم تنهي مشروع التخرج، وتصلح ملاحظات المراجعة، وتجهز عرضا مدته 5 دقائق.

الواجب: تنهي مشروع التخرج: كل الميزات المطلوبة، و README بصور ورابط تثبيت، و 8 اختبارات على الأقل، و release عليها tag في GitHub.

مرحلة في المشروع: تطبيق مشروع التخرج متاح على مسار اختبار داخلي (Play Console internal testing أو TestFlight) ويمكن للمختبرين تثبيته.

الأسبوع 10 الحصول على وظيفة

المحاضرة 1 الـ CV و LinkedIn و GitHub والعمل الحر

- CV من صفحة واحدة لـ Junior Flutter يجتاز الفرز السريع
- عنوان LinkedIn وقسم About ونشر مشاريعك
- GitHub profile README، والـ repositories المثبتة، والصور و GIFs للعرض
- أين تُنشر وظائف Flutter في مصر وللعمل عن بعد
- أول خطوات العمل الحر: مستقل وخمسات و Upwork، وتحديد نطاق تطبيق صغير وتسعيره
- كتابة عرض عمل والتعامل مع أول عميل

تطبيق عملي في المحاضرة: تعيد كتابة الـ CV وملف LinkedIn في المحاضرة مع ملاحظات المدرب، وتثبت Sehha ومشروع التخرج على GitHub مع README مناسب لكل منهما.

المحاضرة 2 مقابلات تجريبية ويوم العرض

- أسئلة Dart و Flutter الشائعة في المقابلات
- شرح إدارة الحالة و widget lifecycle ومعمارية التطبيق بصوت مسموع
- كتابة widget صغير مباشرة في وقت محدد
- الأسئلة السلوكية والحديث عن مشاريعك
- يوم العرض: تقديم مشروع التخرج للضيوف وشركاء التوظيف

تطبيق عملي في المحاضرة: تجري مقابلة تقنية تجريبية مدتها 30 دقيقة مع ملاحظات مكتوبة، ثم تعرض تطبيق مشروع التخرج في يوم العرض.

الواجب: تتقدم إلى 10 وظائف Flutter أو مشاريع عمل حر على الأقل خلال الأسبوعين القادمين، وترسل القائمة للمدرب للمتابعة في الـ office hours.

مشروع التخرج

يبني كل طالب تطبيق Flutter أصليا لحالة استخدام مصرية حقيقية من وصف مختصر، بنفس معمارية Sehha (Riverpod و go_router و dio و secure storage)، ويطلقه على مسار اختبار داخلي في Google Play أو TestFlight.

المتطلبات

- 5 شاشات على الأقل مع تنقل باستخدام go_router
- تسجيل دخول مع حفظ التوكن في secure storage، أو سبب واضح يعتمده المدرب إذا كان التطبيق بلا حسابات
- بيانات من REST API أو Firebase مع حالات التحميل والفراغ والخطأ
- إدارة الحالة باستخدام Riverpod
- نموذج واحد على الأقل مع التحقق من المدخلات
- العربية والإنجليزية مع تخطيط RTL صحيح
- 8 اختبارات unit و widget ناجحة على الأقل
- Flavors للـ dev والـ production، وبدون أسرار داخل الـ repository
- نسخة release موقعة على Play Console internal testing أو TestFlight
- Repository عام على GitHub مع README وصور وخطوات التشغيل

أفكار مقترحة

- تطبيق حجز خدمات منزلية للسباكين والكهربائيين
- تطبيق طلب من الصيدلية مع رفع صورة الروشتة وتتبع الطلب
- تطبيق اشتراكات جيم وحجز الحصص
- مخطط رحلات بالميكروباص والمترو مع حفظ المسارات
- تطبيق قائمة طعام وحجز طاولات لسلسلة مطاعم محلية
- تطبيق للبحث عن مدرس خصوصي لمواد المدرسة والجامعة

معايير تقييم مشروع التخرج

يعمل من البداية للنهاية على جهاز حقيقي	25%
بنية الكود والمعمارية وسهولة القراءة	20%
إدارة الحالة والـ API ومعالجة الأخطاء	15%
جودة الواجهة ودعم RTL العربي وسهولة الاستخدام	15%
الاختبارات	10%
نسخة release على مسار اختبار	10%
العرض وشرح القرارات	5%

طريقة التقييم

الواجبات الأسبوعية ترفع على GitHub قبل الجمعة التالية، وتُقيّم على الكود العامل والـ commits والـ README	20%
التطبيقات العملية في المحاضرة والمشاركة إنهاء التطبيق العملي بنهاية المحاضرة أو في الـ office hours	10%
مراحل مشروع Sehha تُراجع في الأسابيع 4-7 حسب قائمة المرحلة، مع code review قصير	25%
مخرجات أسبوع الذكاء الاصطناعي ميزة المفضلة وإصلاح الأخطاء، وتُقيّم على جودة المراجعة ونجاح الاختبارات	10%
مشروع التخرج يُقيّم حسب معايير مشروع التخرج	30%
أسبوع التوظيف CV وملف LinkedIn و GitHub محدثة، بالإضافة إلى المقابلة التجريبية	5%

قائمة المهارات

Dart

- [ ] تستخدم null safety بدون فرض القيم بـ !
- [ ] تكتب classes و sealed types و enums و extension methods
- [ ] تستخدم async/await و Futures و Streams مع معالجة صحيحة للأخطاء
- [ ] تستخدم records و switch expressions لكود واضح

الواجهات

- [ ] تبني أي شاشة من تصميم Figma باستخدام layout widgets في Flutter
- [ ] تصلح أخطاء الـ overflow والـ constraints
- [ ] تطبق Material 3 theme بوضع فاتح وداكن
- [ ] تدعم RTL العربي والإنجليزية في نفس التطبيق

المعمارية وإدارة الحالة

- [ ] تنظم الكود حسب الـ feature مع repository layer
- [ ] تدير الحالة باستخدام Riverpod Notifier و AsyncNotifier
- [ ] تقرأ كودا مبنيا على Bloc
- [ ] تتنقل باستخدام go_router، بما في ذلك الـ redirects والـ deep links

البيانات والـ Backend

- [ ] تستدعي REST APIs باستخدام dio و interceptors
- [ ] تولد JSON models باستخدام json_serializable أو freezed
- [ ] تحفظ التوكن بشكل آمن وتتعامل مع انتهاء الجلسة
- [ ] تعمل cache للبيانات محليا باستخدام shared_preferences و sqflite
- [ ] تستقبل إشعارات push باستخدام Firebase Cloud Messaging

الجودة والنشر

- [ ] تكتب unit و widget tests باستخدام mocktail و provider overrides
- [ ] تجهز flavors للـ dev والـ production
- [ ] توقّع Android App Bundle وترفعه على Play Console
- [ ] تشرح خطوات إطلاق iOS على TestFlight
- [ ] تعمل بـ Git branches و pull requests
- [ ] تراجع الكود المولد بالذكاء الاصطناعي قبل دمجه

بعد التخرج

الشهور 1-3 بعد التخرج

- تحسّن Sehha ومشروع التخرج، وتصلح الملاحظات، وتنشر تطبيقا واحدا على الأقل في production على Google Play
- تتقدم لوظائف Junior Flutter والتدريب كل أسبوع
- تأخذ مهمة عمل حر صغيرة مدفوعة لتتدرب على التعامل مع عميل
- تقرأ أجزاء توثيق Riverpod و Flutter التي لم تقرأها

الشهور 4-6

- تتعلم Bloc جيدا لأن شركات مصرية كثيرة تستخدمها
- تضيف animations وخرائط ومدفوعات (مثل Paymob أو Fawry SDKs) إلى مشروع جانبي
- تتعلم طبقات clean architecture و dependency injection
- تجهز CI للاختبارات والـ builds (GitHub Actions أو Codemagic)

الشهور 7-12: نحو مستوى Mid-level

- تتولى ميزة كاملة من التصميم حتى النشر على المتجر داخل فريق
- تتعلم أساسيات الـ native: platform channels و Kotlin و Swift لعمليات ربط صغيرة
- تقيس الأداء باستخدام Flutter DevTools وتقلل حجم التطبيق
- تتعلم خيار backend مثل Firebase بعمق أو REST API بسيط حتى تسلم تطبيقات كاملة
- تساهم في حزمة Flutter مفتوحة المصدر أو تكتب عما تبنيه

الوظائف التي يقود إليها المسار

Junior Flutter Developer | Mobile Application Developer (Flutter) | متدرب Flutter في شركة برمجيات أو شركة ناشئة | مطور تطبيقات موبايل مستقل (مستقل وخمسات و Upwork) | Junior Flutter Developer عن بعد لشركات في الخليج أو أوروبا أو أمريكا | Cross platform developer في فريق منتج

مصادر

- جولة في لغة Dart: https://dart.dev/language
- توثيق Flutter: https://docs.flutter.dev
- دليل تثبيت Flutter: https://docs.flutter.dev/get-started/install
- توثيق Riverpod: https://riverpod.dev
- مكتبة Bloc: https://bloclibrary.dev

- حزمة go_router: https://pub.dev/packages/go_router
- حزمة dio: https://pub.dev/packages/dio
- حزمة json_serializable: https://pub.dev/packages/json_serializable
- حزمة flutter_secure_storage: https://pub.dev/packages/flutter_secure_storage
- إعداد Firebase مع Flutter: https://firebase.google.com/docs/flutter/setup
- دعم اللغات في Flutter: https://docs.flutter.dev/ui/accessibility-and-internationalization/internationalization
- نظرة عامة على الاختبارات في Flutter: https://docs.flutter.dev/testing/overview
- بناء تطبيق Android وإطلاقه: https://docs.flutter.dev/deployment/android
- بناء تطبيق iOS وإطلاقه: https://docs.flutter.dev/deployment/ios
- Google Play Console: https://play.google.com/console/about/
- TestFlight: https://developer.apple.com/testflight/
- Material Design 3: https://m3.material.io
- توثيق Git: https://git-scm.com/doc
- توثيق GitHub: https://docs.github.com
- نظرة عامة على Claude Code: https://docs.anthropic.com/en/docs/claude-code/overview