(software testing)

(مقدمة)

هذا الشرح سينقسم لـ 7 اجزاء :

① الجزء الاول هنشرح المبادئ والـ concepts الاساسية
② ~ ~ الثاني ~ الـ unit testing .
③ ~ ~ الثالث ~ الـ integration testing
④ ~ ~ الرابع ~ الـ E2E testing
⑤ ~ ~ الخامس ~ الـ TDD والـ BDD .
⑥ ~ ~ السادس ~ الـ performance testing
⑦ ~ ~ السابع ~ الـ Jest framework

① concepts :

*** دة اهم سيكشن لانه هنتعرف فيه على الاساسيات العامة للـ software testing وهنعرف الفرق بين الـ manual testing والـ automation وهنعرف المميزات اللي هناخدها من الـ automated testing وهنعرف انواع الـ testing المختلفة والـ tools اللي هنستخدمها علشان اقدر اعمل testing للـ application بتاعي .

1- ايه هو الـ software testing ؟

هو طريقة بنختبر بيها اداء الـ application اللي عملناه ونتأكد بيها ان كل الـ features شغالة بدون اي bug ونستطلع الناتج اللي متوقع منه .

علشان نقدر نعمل testing فيه طريقتين :

① يا manual testing
② ~ ~ automated testing

NO.: رقم :
Date : التاريخ :

ROXI

e- ايه الفرق بين إنك اعمل testing لل software باستخدام طريقة الـ manual testing أو باستخدام طريقة الـ automated testing؟

① طريقة الـ manual testing هو انى بيدا استغل الـ application واجرب كل feature فيه بنفسى اكنى انا الـ user واراقب هل الـ feature شغال ويطلع الناتج الى مطلوب منه ولا فيه مشاكل.

② طريقة الـ automated testing بيحصل لنفس غرض الـ manual وهو انى اجرب الـ features واتأكد انها كان بيأدى الوظيفة الى مطلوبة منه ولا لا، ولكن بشكل أوتوماتيك مش يدوى بدون تدخل العامل البشرى اى بمعنى آخر في طريقة الـ automated testing إننا "هنكتب code يختبر الـ code بتاعنا".

٢- اية مميزات الـ automated عن الـ manual؟

* لا يحتاج إلى عامل بشرى عكس الـ manual
* عملية الـ testing هتاخد وقت أقل من الـ manual
* في خلال دقايق بقدر اعمل testing لكامل الـ application.
* بيساعدنى اكتب كود اصغر وادق بحيث يخلينى أدرك مدى تعقيد ومدى نظافة الكود الى بحاول اختبره.
* الـ cost الى هيتم صرفه على عملية الـ testing مع زيادة حجم الكود هيكون اقل مقارنة بالجهد الى هيكلفنى لو عملت manual testing

NO:
Date:

٤- ايه الانواع المختلفة لل software testing ؟

يوجد أكثر من نوع مختلف ولكن الذي يهمنا كـ developer هو ٣ أنواع فقط وهما:
① ال unit testing
② ال integration testing
③ ال End-2-End testing

٥- ايه الفرق بين ال unit والـ integration وال E2E ؟

- ال unit testing هو إني أكتب test cases بـ test بيتر unit معينة في الـ application وال unit دي بتكون معزولة عن الـ dependencies بتوعها
- ال unit testing هو software development techniques هو طريقة بنقسم بيها ال application بتاعتنا إلى units منعزلة عن بعضها ونكتب automated test لكل unit منهم.
- مثال على ال unit دي مثلاً: ال login function، أو عموماً أي function بتنفذ feature معينة.

- مشكلة ال unit test إنك بتعمل ال test لل unit من غير ال dependencies اللي بتعتمد عليهم زي مثلاً إنك تكلم api،
يعني لو فيه test لل login functionality بتاعة ال app، فأنا مش هقدر أعمل test غير على ال pure login function بتاعتي ال function ومش هقدر اعمل test على أي side effect زي ال api مثلاً

- ال unit testing سهل وسريع ولكن مش هتدينى الثقة الكاملة اذا كان ال feature شغالة بشكل سليم فى ال app ولا لا، لأن ال unit test مابتشوفش بعضها الا كل حاجة لوحدها.

- فى ال integration testing هنبدأ نـ test ال app بال dependency متنوعة سواء كانت api أو database أو file أو أى حاجة.

عيبه انه بياخد وقت اطول من ال unit test وميزته انه عكس ال unit test بحيث انه بيتأكد ان ال feature شغالة بالكامل وبتعمل test لكل حاجة لوحدها من غير ال dependency زى ال unit test.

- ال end-2-end test ومن اسمه هو end-2-end يعنى بعمل test عال flow بتاع ال feature بالكامل من أول ما ال user يفتح ال UI بتاع ال app لغاية آخر حاجة المفروض تحصل من ال feature دية.

- فى ال e2e احنا بنعمل test عال feature من خلال ال UI بحيث ننفذ ال senario اللى هيحصل بالضبط مع ال user.

- ال e2e شبه فكرة ال manual ولكن هيتم بشكل automated، يعنى انا هعمل user robot يمر على ال flow بتاعت ال app بشكل كامل من البداية للنهاية.

• ميزة الـ e2e إنه بيخليك تمر بالسيناريو الكامل
اللي الـ user هيمر بيه.

6- كل نوع من أنواع الـ testing ليه مميزاته وعيوبه، بالتالي إيه أكتر نوع لازم استخدمه؟ بمعنى آخر إيه الـ pyramid اللي لازم اتبعه؟

يفضل اعمل unit test أكتر، واكتب شوية integrate وشوية أقل من الـ e2e.

الـ unit سهل وسريع ولكن مش بيسلط الـ test الـ feature بالكامل.
الـ integrate أبطأ من الـ unit لانه بيـ test الـ feature بالكامل يعني بيـ test الـ feature بالـ side effect بتاعه، بالتالي هيديني ثقة أكبر

الـ e2e أبطأ بكتير ولكن بيديني التجربة الكاملة اللي هيبرز الـ end user من أول ما يفتح الـ UI لحد آخر نقطة هيوصلها من الـ feature.

7- ايه الـtools اللي بنستخدمها علشان نعمل unit test و integrate test ؟

في الـ js بنستخدم tools كتيرة :

① jasmine ودي اقدم library للـunit test

② mocha ودي مش بتديني كل حاجة زي الـjasmine بالتالي لازم استخدم مكاتب تانية اسمها chai

③ jest ودي مبنية على الـ jasimine ونودت feature اكتر وظيفتها كتابة الـtest cases الـel

* الـ jest هو اشهر testing tool بنعمل بيها unit test و integration للـ js programs.

* puppeteer دي الـ tool اللي بنعمل بيها e2e
* cucumber " " " " BDD
* JMeter " " " " performance testing

---

② Unit Testing :

* هنستخدم tool اسمها jest ودي tool بنعمل بيها unit test, integration test على مشاريع الـ كل.
* هنعملها install من الـ npm
* الـ jest command ده امر هيدخل جوه ملفات الـ js او الـ ts اللي بتحتوي على test. special

١- كل ملف تزود بنعمله ملف نحطه .test.js علشان نكتب فيه الـ unit test الخاصة بيها الـ functions اللي جواها.

٢- علشان اعمل unit test على أي function، اولاً بعمل import للـ function دي عندي جوه ملف الـ test، وبعدها اعمل test cases للـ function دي

Test case (جزء الكود):

```
test('sum-should return 2+3 = 5', ()=>{
    const result = sum(3, 2);
    expect(result).toBe(5);
});
```

دي بيتحول عليها الـ matcher

* دي عبارة عن test case بيتوقع ان ناتج الـ sum بـ 2 و 3 لازم يكون 5.

٣- علشان نشغل ملفات الـ test بنكتب
>> jest
او >> npm test

٤- الـ describe() زي الـ test ولكن بنعمل بيها group مجموعة test مع بعض علشان يبقى منظم أكتر من الـ log

الـ describe() هو اللي container لتنظيم شكل الـ tests، هو الـ container اللي نجمع فيه مجموعة من الـ test cases مع بعض.

٥- بدل كلمة test() ممكن نستخدم it() والاتنين هما نفس بعض بالضبط، الاتنين بنعمل بيهم test cases العرق الوحيد إن كلمة it أوضح لما تقرا الـ test case بس أكتر ولا أقل ده الأمر استخدم اللي يريحك.

٦- تذكر: الـ unit test بنعمله على الـ pure function اللي مالهوش side effect.

الـ integration test بنعمله على الـ function اللي بتعتمد على dependency خارجي فيها side effect.

احنا غالباً كـ backend dev أو frontend-dev هنعمل integration test لإن غالباً بنعتمد على non-pure functions.

٧- ازاي نكتب test case؟
بنستخدم test() أو it()، وبنبعت 2parameter
الأول: هو رسالة
التاني: هو الـ callback اللي هنكتب فيه كود الـ test

```
it('', () => { expect().toBe() });
```

الـ matcher

رسالة الـ test اللي بتوضح بيعمل وظيفة الـ test

NO.: رقم:
Date: التاريخ:

٨- الـ describe هو function كل وظيفتها إنها بتعمل grouping لمجموعة من الـ test،

إحنا بنستخدمها علشان نعمل group للـ tests اللي بتتعامل مع نفس الـ unit / الـ function علشان يكون تنظيمي مش أكتر.

٩- إيه هو الـ matchers؟

هو function بنستخدمها في الـ test case علشان نقارن بين الـ actual result اللي راجع من الـ unit اللي بنعمل عليها test وبين الـ expected result

ومن بيساعدنا علشان نحدد هل الـ test نجح ولا فشل.

إحنا نقدر نعمل custom matcher لو ملقيناش الـ built-in matcher اللي يناسب الحالة اللي بنعمل عليها test.

١٠- إيه هو الـ mock؟

هو function بنستخدمها بترجع fake data بدلاً من الـ real data اللي بتجيلك من الـ external dependency بالتالي ضمنت إن الـ test code بتاعي هيبقى معزول عن العوامل الخارجية بالتالي مش هيتأثر عليهم

```
ex: const m = jest.fn(); → mock function هترجع
m("b17");  → هيرجع الـ fake data اللي هترجع
expect(m).toBe( );
```

NO.: رقم:
Date: التاريخ:

11- ليه ظهر ال mock؟

لأن ال unit test بيتم على pure function ولا تعتمد على dependencies خارجية، بالتالي لو ال function بتعتمد على data من مصدر خارجي زي ال db انا أقدر أعمل مصدر بديل / fake data بحيث أقدر أعمل unit test على non-pure function.

12- إيه هو ال code coverage؟

* دي tool تبع ال jest بتعمل report بتوضح بيه نسبة الـ test، اللي انا عاملة على الكود بتاعك.
* ماهو إيه الاداة قياس بتقيس بيها نسبة الأكواد اللي استخدمها من testing المشروع

>> npm test -- --coverage

* الامر ده هيعمل folder اسمه coverage يحتوي على web page بيعرض تقرير يوضح نسبة الأكواد اللي استخدمتها test.

13- إيه هو ال setup functions؟

هي function بيستخدم run تلقائياً قبل أو بعد ال test وذلك لتجهيز ال environment أو تجهيز ال variables أو ال mocking function اللي سيتم استخدامها مع ال test cases.

١٤- لازم تفكر فيه قبل ما تكتب unit test؟

قبل ما تكتب الـ testing لازم تفكر من ٣ حاجات:

① هل الـ unit / function اللي هعمل عليها test، هل بتشتغل sync code ولا async code؟

② هي الـ unit دي بتشتغل، انا متوقع منها ايه؟

③ ايه الـ prerequisites الطلوبة علشان الـ unit دي تشتغل صح؟ زي مثلاً في data جاية من نوع الـ unit قبل ما تخش الـ unit.

١٥- ايه الحاجات اللي افكر فيها بعد ما كتبت الـ unit test علشان اتأكد اذا انا كتبتها مظبوط ولا لأ؟

① ايه الـ unit اللي انا هشتغل عليها؟
② ايه الـ test case بتاعتها؟ description
③ ايه هو الـ actual output اللي طلع لما عملت testing؟
④ ايه هو الـ expected output؟

الحاجات دي لازم تكون واضحة وانا بعمل unit testing علشان لو حصل failure اعرف اعرف الحاجة اللي