**1. Explică pe scurt fiecare nivel de testare, având ca referință testing pyramid.**

**Unit tests**

Testarea modulară este primul nivel al piramidei de testare. Testele modulare au un domeniu de aplicare limitat și se asigură că unitățile de cod izolate funcționează conform așteptărilor. Testele modulare ar trebui să evalueze o singură variabilă și nu se bazeze/să nu depindă de factori externi.

**Integration tests**

Testarea de integrare reprezintă cel de-al doilea nivel al piramidei de testare software. Testarea modulară verifică porțiuni mici ale unei baze de cod. Echipa Dev ar trebui să efectueze teste de integrare pentru a verifica interacțiunea acestui cod cu alte coduri din aplicație, care cuprind în general tot software-ul. Aceste teste verifică modul în care un cod interacționează cu componente externe.

**End-to-end tests**

Testarea end-to-end este cel de-al treilea nivel al piramidei de testare și verifică întreaga aplicație. Folosește un mediu și date de testare care să simuleze funcționalitatea din lumea reală. Metoda ideală de abordare a acestei testări din perspectiva utilizatorilor finali.

**2. Să presupunem că avem o aplicație bancară în dezvoltare și business-ul decide să implementeze o bară de căutare (search) în aplicație. Este creat un task de dezvoltarea a acestei funcții, iar ție îți este atribuit task-ul de a o testa. Descrie procesul de la A la Z, acoperind aspecte precum scrierea testelor, testarea efectivă, închiderea task-ului și ce consideri că trebuie sa fie îndeplinit pentru a finaliza task-ul.**

***Scrierea testelor***

Funcționalități și cum ar trebui să funcționeze individual  
1. Câmpul de intrare  
- ar trebui să permită introducerea textului manual sau cu voce (dacă această opțiune este valabilă).  
- acest câmp trebuie să fie vizibil și așezat în pagină astfel încât să nu fie nevoie să se deruleze pagina în jos pentru a se ajunge la el.  
  
2. Text substituent:  
- Textul substituent este de obicei afișat în interiorul câmpului de intrare atunci când este gol.  
- Oferă utilizatorilor un indiciu sau un exemplu cu privire la ceea ce pot căuta.  
- Textul trebuie să fie clar și nu prea lung, dispărând odată ce utilizatorul începe să tasteze.  
  
3. Pictograma de căutare (lupă):  
- Pictograma de căutare este de obicei reprezentată de o lupă.  
- Este de obicei plasată în interiorul sau la sfârșitul câmpului de intrare.  
- Făcând clic pe această pictogramă sau apăsând "Enter" ar trebui să trimită interogarea de căutare.  
  
4. Butonul Clear (golire) (opțional):  
- Un buton Clear, adesea reprezentat de o pictogramă "X" sau similară, permite utilizatorilor să elimine rapid câmpul de intrare.  
- Apare de obicei după ce utilizatorul a început să tasteze în câmpul de intrare.  
- Dacă faceți clic, ar trebui să ștergeți imediat textul și, eventual, să vă concentrați asupra câmpului de intrare pentru o nouă căutare.  
  
5. Pictograma microfon (opțional):  
- Dacă este acceptată căutarea vocală, poate fi prezentă o pictogramă de microfon.  
- Făcând clic pe această pictogramă ar trebui să se activeze intrarea vocală, permițând utilizatorilor să spună cu voce tare ceea ce caută.  
  
6. Completare automată/sugestii (opțional):  
- Pe măsură ce utilizatorii tastează, o listă de sugestii sau opțiuni de completare automată poate apărea sub câmpul de intrare.  
- Acestea se bazează de obicei pe căutări populare sau date istorice.  
- Utilizatorii pot face clic pe aceste sugestii pentru a finaliza rapid interogarea lor sau pentru a explora interogări conexe.  
  
7. Căutare avansată/filtre (opțional):  
- Unele bare de căutare pot avea o opțiune de a accesa opțiuni de căutare avansate sau filtre.  
- Acest lucru ar putea fi reprezentat de o pictogramă de viteze, un link sau un buton care spune "căutare avansată".  
- Făcând clic aici s-ar putea permite utilizatorilor să-și rafineze căutarea după diverse criterii (intervalul de date, categoria, etc.).  
  
8. Butonul Trimitere (opțional):  
- Deși nu este întotdeauna prezent (făcând clic pe pictograma de căutare sau apăsând "Enter" poate trimite interogarea), unele bare de căutare pot include un buton dedicat "căutare".  
- Acest buton trebuie să fie clar marcat și să indice acțiunea de a efectua căutarea.

***Testarea efectivă***

- Testare pentru a vedea dacă atunci când se introduce un cuvânt/cuvinte cheie și apoi se apăsă pe butonul de search, se afișează rezultatele. / Testare introducând cuvinte care să includă numere sau caractere speciale pentru a vedea cum se comportă.

- Testarea barei de căutare cu numărul minim și maxim de caractere pentru a vedea dacă se comportă conform așteptărilor.  
- Testarea barei de căutare când nu găsesc rezultate sau se returnează prea multe rezultate.  
  
- Testarea barei de căutare de pe platforme diferite (Chrome, Firefox, Safari, etc.) pentru a se asigura că funcționează la fel pe toate platformele.   
- Testarea barei de căutare de pe diferite dipozitive (desktop, telefon, tabletă) pentru a verifica dacă apar probleme de design.  
  
- Testarea performanței barei de căutare prin introducerea un număr mare de interogări pentru a vedea dacă sunt întârzieri în afișarea rezultatelor. - Testarea bare de căutare în diferite condiții de rețea pentru a vedea dacă funcționează bine cu diferite viteze.   
  
- Testarea barei de căutare cu utilizatori reali pentru a aduna feedback cu privire la ușurința sa de utilizare, claritatea instrucțiunilor și experiența generală a utilizatorului.

- Testarea mesajelor de eroare din bara de căutare și asigurarea că sunt clare și utile pentru utilizatori.

- Testarea barei de căutare pentru orice vulnerabilități, cum ar fi atacurile de injecție SQL sau vulnerabilitățile de scriptare încrucișată (XSS).

- Testarea funcției de completare automată a barei de căutare pentru a verifica eventualele probleme de scurgere de date.

- Testarea barei de căutare cu diferite limbi și seturi de caractere pentru a vă asigura că acceptă cerințele de internaționalizare.

- Testarea barei de căutare cu diferite setări regionale pentru a vă asigura că afișează corect rezultatele localizate.

- Testarea barei de căutare după fiecare modificare sau implementare a codului pentru a ne asigura că noile funcții sau remedieri de erori nu îi afectează funcționalitatea.

- Menținerea un set de cazuri de testare de regresie pentru a le rula periodic pentru a ne asigura că bara de căutare continuă să funcționeze conform așteptărilor.

***Închiderea task-ului***

După ce au fost efectuate cazurile de testare pentru bara de căutare, bug-urile au fost remediate, restestate și s-a constatat funcționarea corectă a barei în cel puțin 90% din teste.

***Finalizarea task-ului***

Dacă bara de căutare îndeplinește următoarele criterii, se poate considera că taskul e finalizat:

- îndeplinirea obiectivului sau rezultatului dorit

- respectă standardele de calitate

- s-a încadrat în deadline

- s-a încadrat în buget

- a primit aprobarea din partea stakeholder-ilor

- funcția de căutare este acum documentată în mod corespunzător

- funcția trebuie să fie fost supusă testării, validării și verificării cu succes pentru a se asigura că funcționează corect și îndeplinește cerințele specificate.

**3. Care este avantajul Unit testing-ului față de End-to-end/system testing?**

Testarea modulară este de obicei mai rapidă și mai eficientă decât testarea de integrare. Aceasta deoarece este mai ușor și mai rapid să se testeze o singură componentă în izolare decât mai multe componente împreună.

**4. Care este avantajul End-to-end/system testing-ului față de Unit testing?**

Reducerea numărului de erori

Unele erori apar mai mult ca sigur în timpul dezvoltării sistemelor complexe. Testarea sistemului verifică codul unui sistem și funcționalitatea acestuia în raport cu cerințele, astfel încât erorile nedetectate în timpul testării de integrare și modulară pot să iasă la iveală în timpul testării sistemului.

Economii

**5. Ai următoarele funcționalități în aplicație:**

* **calcul rată;**
* **postare incident în aplicație (erori de plăți, de funcționalitate, de afișare a aplicației);**
* **toate câmpurile din pagina de înregistrare sunt obligatorii;**
* **plăți IBAN;**
* **investiții prin aplicația bancară.**

**Stabilește ce nivele de testare se potrivesc pentru fiecare funcționalitate și argumentează alegerea făcută. O funcționalitate poate avea mai multe niveluri de testare (unit, integration, end-to-end).**

**calcul rată** - unit testing unde se va vedea că se calculează procente, rata, dobânda fixă, variabilă, etc. – se vor verifica toate formulele

- integration testing dacă avem un sistem avansat prin care, atunci când cineva merge la bancă pentru un împrumut, se verifică venitul și se spune suma la care acea persoană se încadrează să primească.

- end-to-end testing – persoana vrea un credit pe x ani, iar banca spune că depășește gradul de îndatorare

**postare incident în aplicație (erori de plăți, de funcționalitate, de afișare a aplicației)**- unit testing – se verifică codul

- end-to-end testing – eroare de afișaj apare în etapa finală a dezvoltării produsului.

**toate câmpurile din pagina de înregistrare sunt obligatorii**

- unit testing – se verifică codul

- integration testing – se verifică dacă în urma completării corecte a câmpurilor se poate ajunge la funcționalitatea dorită

**plăți IBAN**

- integration testing – se verifică încadrarea cu alte componente

- end-to-end testing – integrarea componentelor în sistem

**investiții prin aplicația bancară**

- end-to-end testing - integrarea componentelor în sistem