

**Cerințe proiect final
_ Examen Oral TM**

Condiții realizare proiect final

Testare Manuală

- Proiectul va fi realizat în limba română
- 15 va fi numărul maxim de slide-uri
- Prima pagină a proiectului va conține următoarele:
 - Titlu: Proiect final
 - Nume și prenume
 - Data examenului
- Ultima pagină - slide de mulțumire (simplu)
- Proiectul va conține **partea teoretică** (de răspuns la cerințele de la partea I de la examenul oral), cât și **partea practică**.

Atenție!

1. Proiectul practic **nu** trebuie să conțină teste pe unul din site-urile studiate la curs.
2. Proiectul practic se va realiza pe un site real.

Partea I

1. Explicați pe scurt ce sunt cerințele de business, la ce ne folosesc și cine le creează
2. Explicați diferența între un test condition și test case
3. Enumerați și explicați pe scurt etapele procesului de testare
4. Explicați diferența între retesting și regression testing
5. Explicați diferența între functional testing și non-functional testing
6. Explicați diferența între blackbox testing și whitebox testing
7. Enumerați tehnicile de testare și grupați-le în funcție de categorie (blackbox, whitebox, experience-based)
8. Explicați diferența între verification și validation
9. Explicați diferența între positive testing și negative testing și dați câte un exemplu din fiecare
10. Enumerați și explicați pe scurt nivelurile de testare

Partea II

Partea practică se va face la alegere pe baza următoarelor opțiuni:

Varianta 1: Se va testa o aplicație web pe care să o gestionați prin intermediul aplicației JIRA și al pluginului Zephyr

Această variantă trebuie să acopere următoarele componente:

- Cerințele pentru aplicația / website-ul ales - Se vor adăuga în prezentare DOUA cerințe de business (story-uri) care să reflecte atât titlul cât și descrierea story-ului. Se pot face poze direct la ce ați lucrat în Jira
- Condițiile de testare și cazurile de testare - Condițiile de testare vor reprezenta DOAR TITLUL testelor, iar test case-urile trebuie să conțină și pașii de execuție. Condițiile de testare trebuie să fie cel puțin 10, iar testele adăugate în prezentare trebuie să fie cel puțin două
- Matricea trasabilității - trebuie să reflecte în mod corect legătura între story-uri, teste și bug-uri. Ca să se întâmple asta trebuie să vă asigurați că testele sunt legate corect la story-uri și că bug-urile sunt legate corect la teste
- Raportul generat din aplicația Jira din secțiunea Dashboards
- Analiza de risc - va cuprinde riscurile de proiect și de produs identificate în scopul testării pe care ați făcut-o, NU IN GENERAL. De asemenea, riscurile vor fi prioritizate și schematizate la alegere fie pe o matrice a riscului fie pe un grafic PRISMA
- Raportul de defect (bug ticket) - prezentarea va trebui să conțină cel puțin două rapoarte de bug care să conțină următoarele elemente: summary (titlul bug-ului), precondiții (dacă există), pași de reproducere, rezultate așteptate și rezultate actuale
- Concluzii generale în urma testării - vor conține informații cu privire la numărul de story-uri totale, numărul de story-uri acoperite de teste, numărul total de teste scrise vs executate, numărul de bug-uri identificate împreună cu o explicație legată de severitatea acestora și cât de mult îl afectează pe utilizatorul final.

Partea II

Varianta 2: Se va testa un API prin intermediul aplicației Postman.

Această variantă trebuie să conțină următoarele informații:

1. Requesturi de GET, POST, oricare dintre PUT și PATCH, si DELETE
2. Explicația fiecărui request în parte
3. Codul de răspuns returnat de fiecare din metodele HTTP de mai sus
4. Cod Java Script scris direct în tab-ul de Tests din Postman
5. Raportul de execuție generat la alegere între Collection Runner (direct din Postman) sau Newman (prin intermediul terminalului)
6. Cel puțin două rapoarte de bug-uri (la alegere dacă să puneți poze din Jira sau să le scrieți direct în prezentare)
7. Tipul / tehnica de testare folosite pentru fiecare request în parte
8. Concluzii în urma testării (câte teste s-au făcut, care este procentajul acoperit al requesturilor, care este procentajul de teste passed, care este riscul de a lansa aplicația fără repararea bug-urilor)

Partea II

Varianta 3: Se va crea o bază de date pe un subiect la alegerea voastră

Această variantă trebuie să conțină următoarele componente:

- Instrucțiuni DDL (cel puțin una dintre CREATE, ALTER, DROP, TRUNCATE)
- Instrucțiuni de DML (INSERT, DELETE, UPDATE)
- Instrucțiuni DQL (select all, select câteva coloane, filtrare cu where, filtrări cu like, filtrări cu AND și OR, funcții agregate, filtrări pe funcții agregate, joinuri - inner join, left join, right join, cross join, limite, order by, chei primare, chei secundare)
- OPTIONAL (ofera bonus la examen) - folosirea subquery-urilor

Fiecare instrucțiune va trebui să aibă asociată o explicație a scenariului pe care l-ați acoperit.

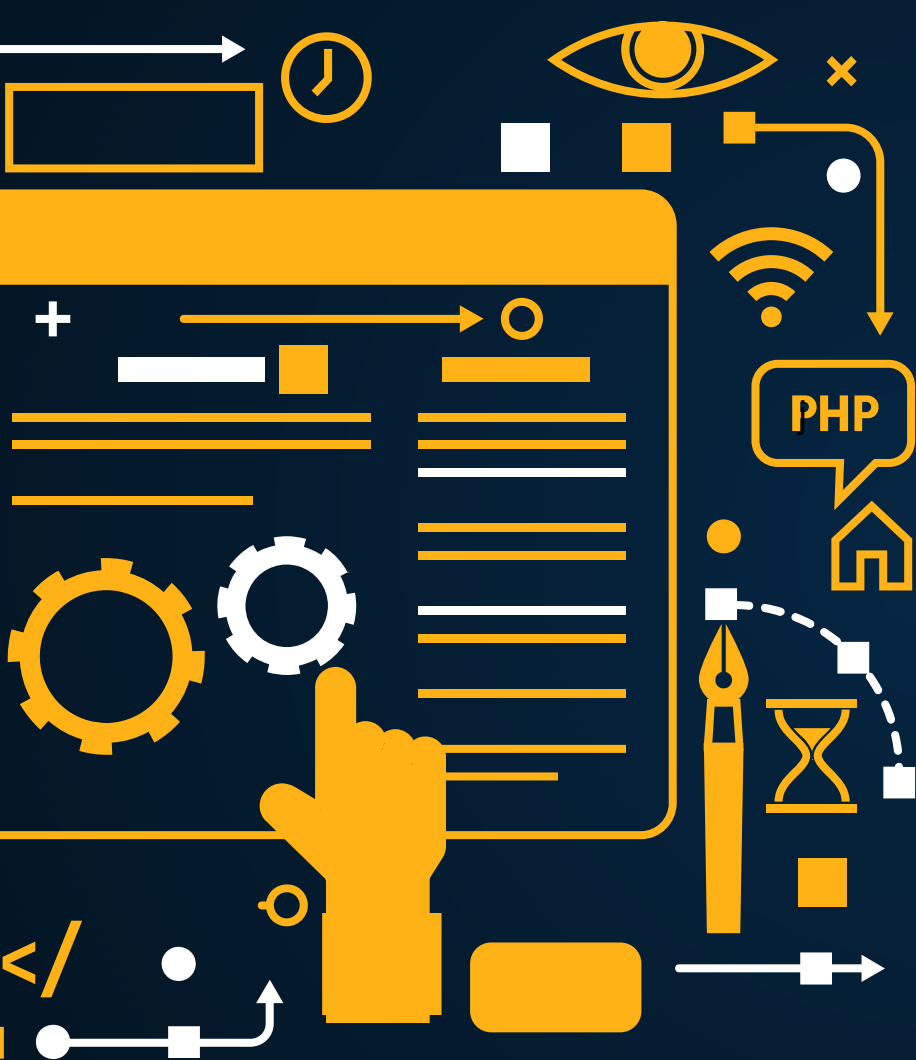
Exemple de scenarii:

- Crearea unei tabele entitate care să stocheze toate informațiile despre angajații din companie
- Crearea unei tabele de legătură între tabela produse și tabela comenzi
- Returnarea tuturor angajaților care au avut un bonus de performanță mai mare de \$1000
- Returnarea tuturor clienților care nu au avut comenzi în luna anterioară

De asemenea, fiecare două tabele legate prin chei primare și chei secundare vor trebui să aibă atașată o explicație legată de felul în care acestea sunt legate (care e cheia primară, care e cheia secundară) și respectiv care e relația între cele două tabele (1:1, 1,n, n,n)

Criterii de evaluare a proiectului final TM

- 1 punct pentru formatare: scrisul clar, culorile ușor de citit, să se respecte maximum de 15 slide-uri.
- 3 puncte pentru acoperirea cerințelor teoretice:
 - 1.5 pct corectitudine
 - 1.5 pct completitudine
- 6 puncte pentru acoperirea cerințelor practice:
 - 2 pct - îndeplinirea tuturor cerințelor practice
 - 2 pct - pentru adăugarea proiectului final în GIT (doar partea pentru examenul final. Exemplu: dacă examenul îl veți susține cu API testing veți încărca doar partea de API)
 - 2 pct - minim 10 teste scrise (doar pt JIRA și API)
 - pt SQL să aveți create cel puțin 5 tabele de SQL.
 - să aveți joinuri corecte între cel puțin 3 tabele.
 - să existe joinuri de diverse tipuri (left join, inner join și right join)
 - să se folosească corect funcțiile agregate cu clauza group by și filtrare pe funcții agregate cu having



Mult succes!

După finalizarea proiectului trimite-l pe
adresa de email:

support@itfactory.ro