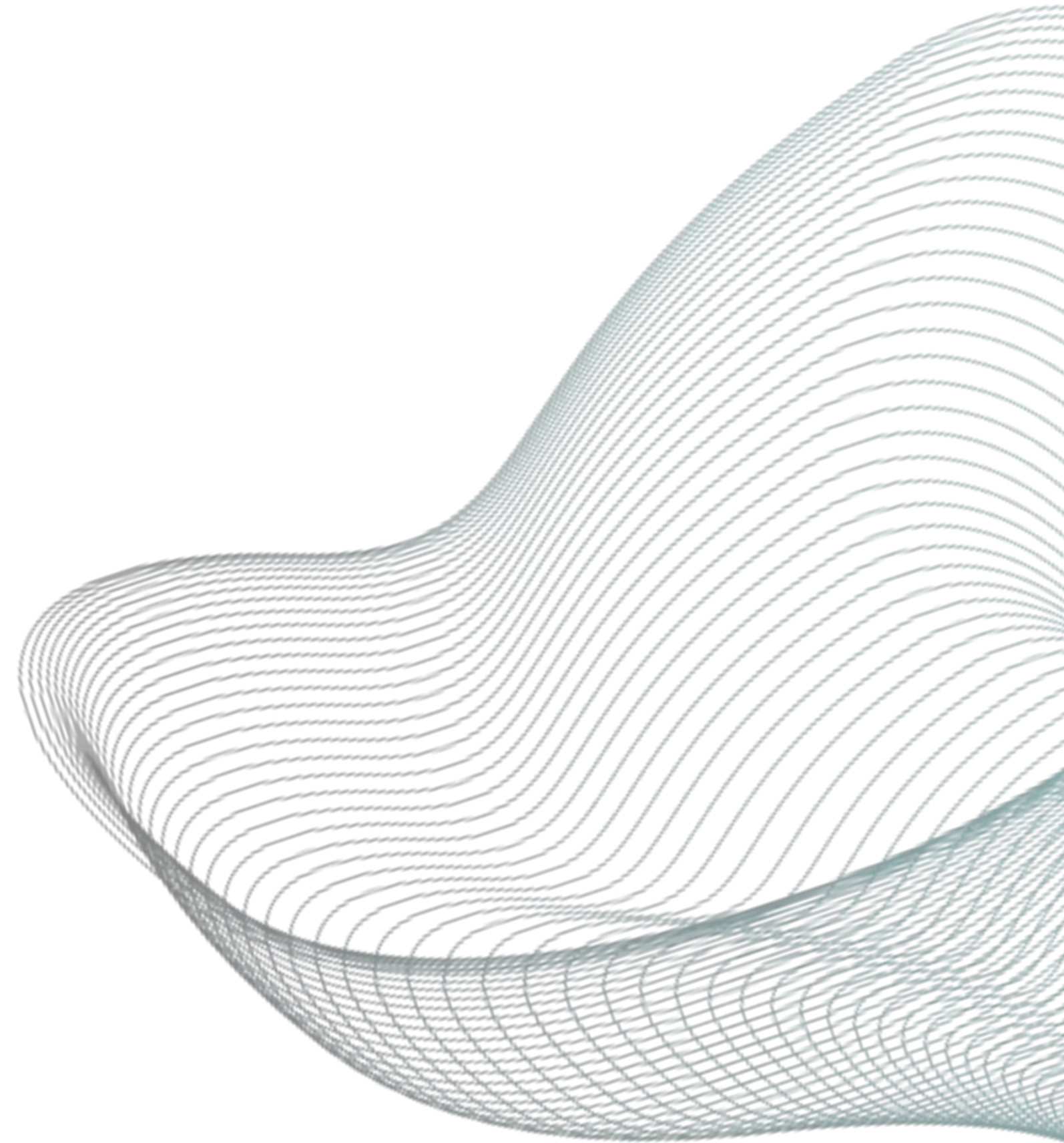


PROIECT FINAL

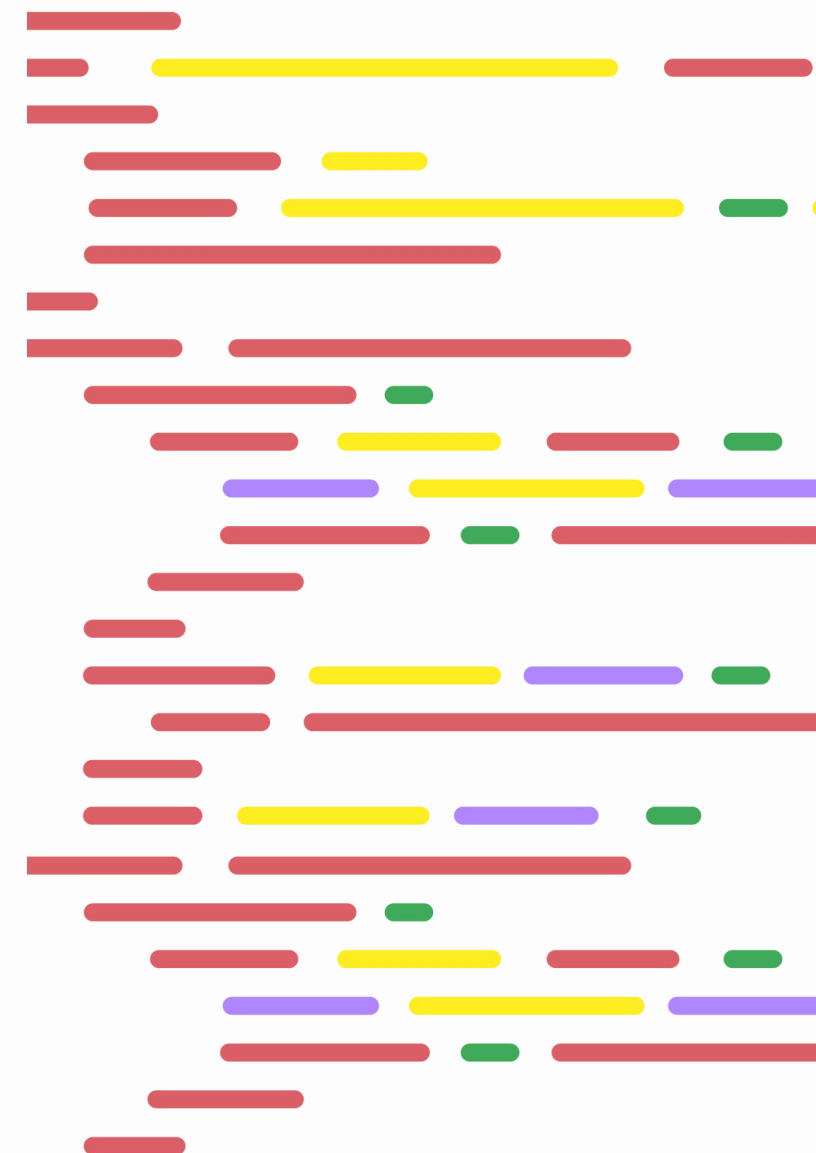
NUME CURSANT

- dată examen -



Cuprins

01. Variabile / Constante/ Tipuri de Date / IF
02. Structuri de Date
03. Funcții și Parametri / Clase și Obiecte
04. Selectorii / TDD / BDD / Gherkin
05. Testare Unitara / API / Metode HTTP
06. Descriere Aplicație
07. Implementare Proiect
08. Avantaje Testare Automată
09. Structură Proiect
10. Poze Cod - Teste
11. Raport Bug + Raport Execuție
12. Concluzii



PARTEA I - TEORIE

Variable / Constante / Tipuri de Date



Variable

Definire Variabile



Constante

Definire Constante



Tipuri de Date

Enumerare tipuri de Date

Structura Alternativă IF



Definirea structurii IF

It would help if you were comfortable delegating, but there's still much work involved.



Elif

It would help if you were comfortable delegating, but there's still much work involved.



Else

It would help if you were comfortable delegating, but there's still much work involved.

PARTEA I - TEORIE

Structuri de Date



Liste

Definire liste



Tupluri

Definire tupluri



Seturi

Definire seturi



Dicționare

Definire dicționare

Funcții și Parametri / Clase și Obiecte



Funcții

Definire funcții



Parametri

Definire parametri



Clase

Definire clasă

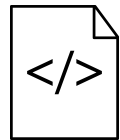


Obiecte

Definire obiect

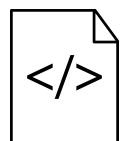
PARTEA I - TEORIE

Selectorii / TDD



Selectorii

Definire Selectorii



TDD

Definire TDD + Avantaje

BDD / Gherkin



BDD

Definire BDD + Avantaje



Gherkin

Definire Gherkin

PARTEA I - TEORIE

Testare Unitară + API



Testare Unitară

Definire Testare Unitară



API

Definire API

Metode HTTP

Enumerare metode HTTP + Explicații + Status Code

PARTEA II - PRACTICĂ

Nume aplicație

Specificare nume

Descriere aplicație

descriere aplicație + obiectivele ei

Module Testate

Enumerare module testate

Link github

https://github.com/utilizator/automation_testing_python_framework_aplicație

Logo Aplicație Testată

IMPLEMENTARE ȘI UTILIZARE PROIECT

IDE FOLOSIT

IDE-ul pe care l-am folosit pentru
implementarea proiectului a fost
nume_ide

LIMBAJ DE PROGRAMARE

Proiectul de testare automată a fost
dezvoltat în Python conform
informațiilor acumulate pe parcursul
cursului.

FRAMEWORK

Framework-ul folosit a fost
nume_framework împreună cu
design pattern-ul
nume_design_pattern

UTILIZARE PROIECT

Cum se clonează proiectul?

Ce librării au fost instalate și cum au fost instalate?

E nevoie de ceva pași preliminari pentru a putea rula proiectul (ex: creare
de cont, generare de token etc)?

Cum se rulează proiectul?

Cum se generează raportul de execuție?



Titlu Avantaj

Descriere Avantaj



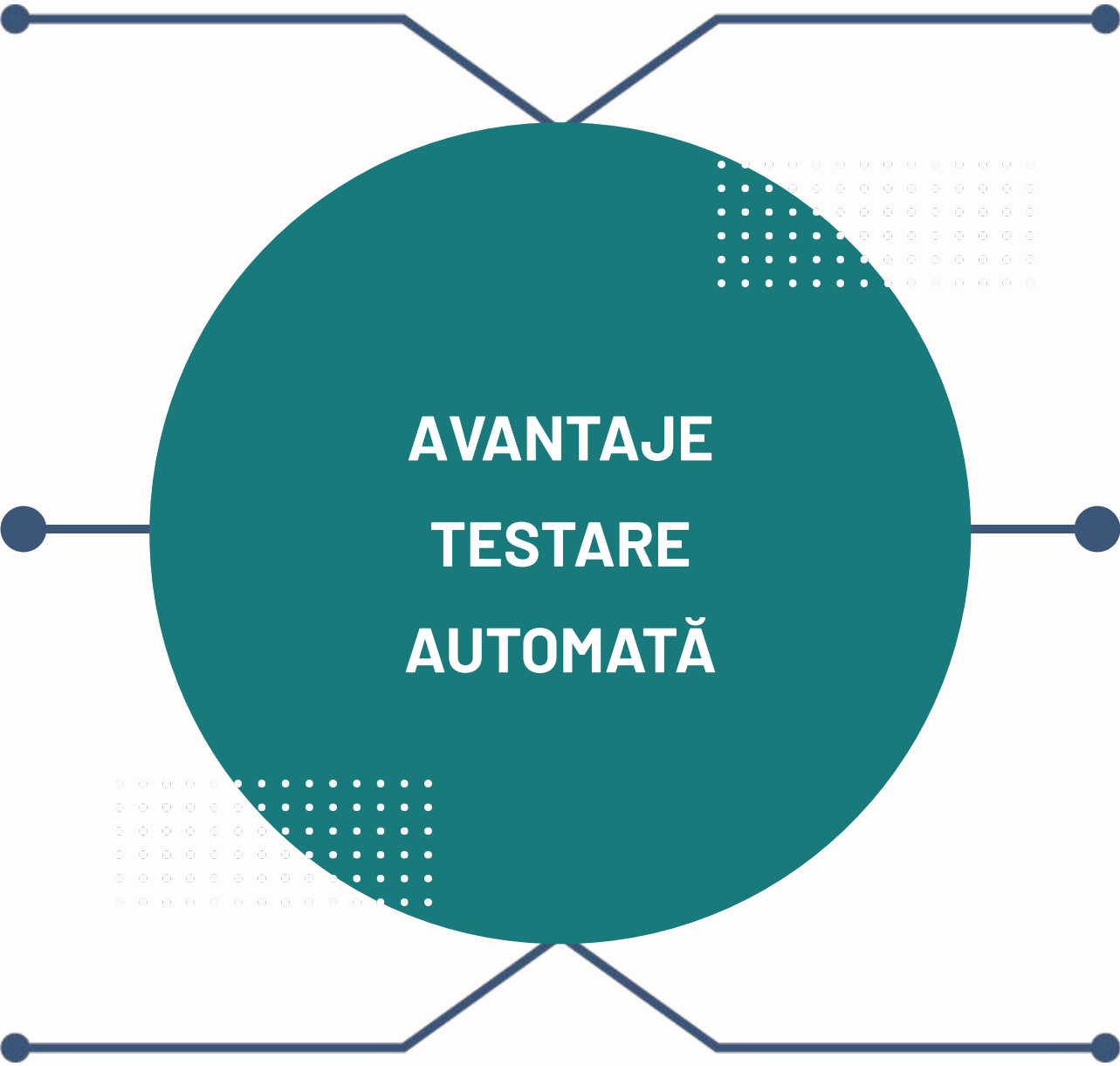
Titlu Avantaj

Descriere Avantaj



Titlu Avantaj

Descriere Avantaj



Titlu Avantaj

Descriere Avantaj



Titlu Avantaj

Descriere Avantaj



Titlu Avantaj

Descriere Avantaj



STRUCTURĂ PROIECT



FOLDER 1

Descriere conținut
folder 1

FOLDER 2

Descriere conținut
folder 2

FOLDER 3

Descriere conținut
folder 3

FOLDER 4

Descriere conținut
folder 4

Poze Structură Proiect (setup / teardown
/ features folder / steps folder / pages
folder / requests folder / tests folder)



Poză Test 1

Nume Test 1

Descriere Test 1

Status



Poză Test 2

Nume Test 2

Descriere Test 2

Status

Titlu Bug

Pași de reproducere

Descriere pași de reproducere

Rezultate așteptate

Descriere rezultate așteptate

Rezultate actuale

Descriere rezultate actuale

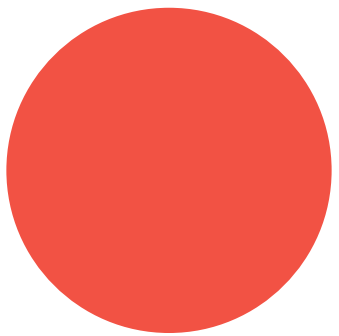
Poză Raport de Execuție

Interpretare Raport de Execuție



Concluzii Proiect

Descriere pe scurt a rezultatelor testării, câte teste au fost create și executate, ce procentaj aproximativ din cerințele în scop au fost acoperite, dacă există vreo funcționalitate pe care nu ai apucat să o testezi, dacă bug-urile găsite impactează lansarea produsului în producție sau se pot fixa și ulterior, dacă ai identificat riscuri de produs care trebuie mitigate, dacă e vreo reecomandare pe care vrei să o faci pentru lansare, dacă sunt ceva lessons learned de care trebuie să se țină cont la proiectele viitoare etc.





Multumesc