PROIECT FINAL

PARTEA I – NOȚIUNI TEORETICE

Explicați ce sunt variabilele. Dar constantele?

- **Variabilele** – reprezintă adrese din memorie care stochează valori, acestea au un nume unic pentru a fi identificate ulterior și se creează în momentul în care îi se atribuie o valoare
- **Constantele** – reprezintă adrese din memorie care stochează valori însă nu își pot schimba valoarea pe parcursul execuției programului

Enumerați tipurile de date și dați câte un exemplu din fiecare

- **int** – un tip de date ce poate defini un număr intreg,
  - Ex: **ziua = 14, luna = 10, anul = 2023**
- **float** – un tip de date ce poate defini un număr zecimal (separatorul zecimal este caracterul "."),
  - Ex: **distanta_dintre_pamant_si_soare_in_minute_lumina = 8.3, circumferinta_pamantului_la_ecuator_in_km = 40075.017**
- **bool** – un tip de date ce poate defini doar două valori: **True**/**False**,
  - Ex: **am_parcurs_cursul_de_testare_automata = TRUE, am_eșuat_la_testul_final = FALSE**
- **string** – un tip de date ce poate defini un sir de caractere de la tastatura delimitate de ' ' sau " ",
  - Ex: **rezultat_final = "Felicitări ai absolvit examenul final!"**

Explicați cum se folosește condiționalul if else

- **IF** – este o structura alternativa ce se folosește când vrem să executăm un set de instrucțiuni atunci când o condiție este adevărata iar un alt set de instrucțiuni este falsa.
- **Elif** – este o modalitate prin care putem sa evaluam mai multe condiții într-o structură alternativă if. Sistemul va rula codul secvențial, de sus in jos, până când va găsi prima condiție adevărata și va rula codul aferent acesteia.
- **Else** – este o modalitate prin care instruim sistemul să execute o instrucțiune în mod implicit atunci când niciuna din condițiile evaluate anterior nu a fost evaluate ca fiind adevărată.

Ce este un API și care este rolul lui?

- **API** este prescurtarea de la **Application Programming Interface** si reprezinta un set de proceduri, funcții și alte elemente pe care un sistem le pune la dispoziție pentru a facilita comunicarea cu un alt sistem având ca **rol** de a permite comunicarea și schimbul de date între diferite aplicații sau servicii realizând posibilitatea de a se integra și colabora între diverse componente software, chiar dacă acestea rulează pe platforme sau tehnologii diferite.

PARTEA I – NOȚIUNI TEORETICE

Enumerați structurile de date și caracteristicile fiecăruia

- **Liste** – Sunt structuri de date care pot salva mai multe elemente in aceeași locație; sunt ordonate, indexate (indexul începe de la 0) și permit adăugarea și ștergerea a mai multor elemente aparținând unor tipuri de date diferite; se definesc cu paranteze drepte „ [ ] ”
  - **Ex: lista_elemente = ["Ana", "Anton", 2, 45.6, True]**
- **Seturi** – Sunt structuri de date care pot salva mai multe elemente in aceeași locație; nu sunt ordonate sau indexate; nu putem adăuga elemente la o anumita poziție și nu putem să extragem un singur element; se definesc prin folosirea acoladelor „ **{ }** ”
  - **Ex: set_elemente = {"Ana","Anton",2,45.6, True}**
- **Tupluri** – Sunt structuri de date care pot salva mai multe elemente in aceeași locație; similar listelor dar nu permit adăugarea și/sau ștergerea a mai multor elemente; se definesc cu paranteze rotunde „ ( )”
  - **Ex: tuplu_elemente = ("Ana", "Anton", 2, 45.6, True)**
- **Dicționare** – Sunt structuri de date care pot salva mai multe elemente de tip **cheie:valoare**; sunt ordonate și permit adăugarea și ștergerea a mai multor elemente aparținând unor tipuri de date diferite prin intermediul cheilor; se prin folosirea acoladelor și structuri cheie:valoare „ **{cheie:valoare}** ”
  - **Ex: dictionar_elemente = {"nume":„Ion","prenume":"Anton","durata_angajare":2,"varsta":45.6, "inca_angajata":True}**

Explicați ce este o funcție și ce este un parametru

- **Funcțiile** – reprezinta blocuri de cod reutilizabile care sunt definite cu scopul de a limita numărul de linii de cod pe care le scriem si a face programul sa fie mai modular si respectiv mai ușor de citit si de gestionat. O funcție are următoarele componente:
  - Inceputul functiei: **def**
  - Numele functiei → free text (se recomanda formatul **snake_case**)
  - Inceputul corpului functiei → marcat de caracterul “**:**”
  - Corpul functiei → marcat de spatiu de la marginea fisierului (indentare). Atunci cand se iese din indentare se iese din corpul functiei
  - Parametri → optionali
  - Instructiune return → optionala

PARTEA I – NOȚIUNI TEORETICE

Explicați ce este o funcție și ce este un parametru

- **Parametri** – reprezintă adrese de memorie temporare care se populează atunci când funcția este apelată. Sunt scriși intre paranteze rotunde iar la definirea metodei pot fi de doua feluri:
  - Parametri expliciți – vor primi obligatoriu valoare la apelare
  - Parametri impliciți – caz în care nu vor primi valoare la apelare ci se va folosi in mod implicit valoarea specificata la definirea funcției

Explicati diferenta dintre o clasă și un obiect

- **Clasa** – reprezinta un set de atribute care descriu cum ar trebui să arate entitatea (ce proprietăți să aibă) și metode care descriu ce ar trebui să facă entitatea (cum să se comporte); acestea determină cum vor arăta și ce vor putea să facă obiectele
- **Obiect** – este o reprezentare reala a unui tipar descris de o clasa și se mai numește și o instanță a clasei; iar când se instanțială un obiect se rezervă un spațiu de memorie care este reprezentat de numele obiectului, și in acel spațiu de memorie se vor stoca toate atributele din clasa cu valori specifice

La ce ne ajută un selector? Cate tipuri cunoști?

- Un selector este un sir de caractere care are rolul de a identifica unul sau mai multe elemente într-o pagina web cu scopul de a interacționa cu ele in procesul de automatizare.
- Exista mai multe tipuri de selectori, printre care:
  - ID – reprezinta un identificator unic pentru un elementsunt in general cel mai usor de folosit deoarece garantează unicitatea elementelor pe site
  - Class – identifică elemente in pagina web pe baza perechilor de tip class=valoare care reprezintă elementul de clasă
  - Name – identifică elemente in pagina web pe baza perechilor de tip name=valoare care reprezintă elementul de nume
  - Link Text – identifică elemente in pagina web pe baza perechilor de tip href=valoare care reprezintă elementul de link text
  - Partial Link Text– identifică elemente in pagina web pe baza perechilor de tip href=valoare care reprezintă text unicat parțial din link text
  - XPATH – este un selector care ne permite navigarea într-un cod HTML

PARTEA I – NOȚIUNI TEORETICE

Ce este TDD, care sunt avantajele și ce este testarea unitară?

- **TDD** este prescurtarea de la **Test Driven Development** si este un proces de dezvoltare software care se bazează pe transformarea cerințelor de business in teste înainte de a avea codul sursa dezvoltat
- **Avantajele** utilizării unui TDD:
  - Ajuta la crearea minimului de cod optim necesar implementării unei funcționalități
  - Se concentrează pe teste, asigurând astfel o aplicație mai apropiata de nevoile clientului
  - Asigura o acoperire mai mare a aplicației prin teste
  - Codul este mai ușor de întreținut
- **Testarea unitară** (sau de componente) e o modalitate prin care fiecare bucată individuală de cod este testată pentru a verifica dacă este pregatită pentru utilizare. Un test unitar reprezintă testarea celei mai mici bucăți funcționale dintr-o aplicație cum ar fi funcții, clase, proceduri, interfețe.

Ce este BDD, care sunt avantajele? Ce este sintaxa gherkin și la ce ne ajută?

- **BDD (Behavior Driven Development)** este un proces de dezvoltare software derivata din TDD care se bazează pe o atenție mai mare asupra scenariilor de testare
- **Avantajele** sunt similare TDD însă prin faptul că adaugă peste codul de testare automata fișierele descriptive ale scenariilor de business care sunt scrise într-un limbaj pe care sa îl înțeleagă și utilizatorii care nu au cunoștințe tehnice numite **feature files**
- **Gherkin** este un limbaj descriptiv folosit in procesul de BDD pentru a putea implementa scenariile de business într-un limbaj natural, scris într-o engleză simplă astfel ajutând să fie înțeles de către toate persoanele implicate

Care sunt principalele metode HTTP? O propoziție explicativă pentru fiecare

- Metodele principale de HTTP sunt:
  - **GET** – este o metoda de HTTP prin intermediul careia putem solicita extragerea de informatii din baza de date
  - **POST** – este o metoda de HTTP prin intermediul careia putem solicita scrierea de informații în baza de date
  - **PUT** – este o metoda de HTTP prin intermediul careia putem actualiza resursele existente sau pentru a crea resurse noi dacă nu există deja
  - **PATCH** – este o metoda de HTTP prin intermediul căreia putem solicita modificarea de informații în baza de date prin modificarea anumitor părți dintr-un obiect

PARTEA II – ASPECTE PRACTICE

DETALII DESPRE APLICAȚIA TESTATĂ. CE FACE SITE/API ENDPOINT/CLASĂ?

- Site-ul **edu.ro** este site-ul oficial al **Ministerului Educației,** care este instituția publică centrală responsabilă de elaborarea și implementarea politicilor și strategiilor naționale în domeniul educației și cercetării în România.
- Scopul acestui site este de a oferi informații utile și actualizate
  - despre sistemul educațional românesc,
  - despre programele,
  - proiectele și inițiativele ministerului,
  - despre legislația și reglementările în vigoare,
  - despre resursele educaționale disponibile, precum și
  - despre oportunitățile de finanțare,
  - burse și mobilități pentru elevi, studenți, profesori și cercetători.
- Este structurat în mai multe secțiuni, care corespund celor mai importante domenii de activitate ale ministerului:
  - Minister,
  - Învățământ preuniversitar,
  - Învățământ superior,
  - „România Educată”,
  - Cooperare Internațională,
- Pe site se pot găsi și
  - Comunicate de presă,
- Română
- Engleză
- Franceză
- Germană
- Bulgară
- Maghiară
- Italiană
- Sârbă
- Spaniolă
- Turcă
- Ucraineană

PARTEA II – ASPECTE PRACTICE

CE LIMBAJ AI FOLOSIT? CE IDE AI FOLOSIT? CE LIBRĂRII AI ALES ȘI CUM SE INSTALEAZĂ ACESTEA?

- În acest proiect am folosit limbajul de cod **Python**, acesta a fost creat în anii 1990 și este cunoscut pentru sintaxa sa ușor de învățat și citit. **Python** este utilizat într-o gamă largă de domenii, cum ar fi dezvoltarea web, analiza datelor, inteligența artificială, automatizarea, dezvoltarea de jocuri și multe altele. Este apreciat pentru comunitatea sa activă, bibliotecile puternice și abilitatea de a fi utilizat pe multiple platforme.

- **PyCharm** este un mediu de dezvoltare integrat (**IDE**) pentru Python dezvoltat de JetBrains. Acest IDE oferă o suită de instrumente puternice pentru dezvoltatori Python, inclusiv funcții de completare automată a codului, depanare, integrare cu sisteme de control al versiunilor, gestionarea proiectelor și multe altele.

- Ca librărie am folosit **Selenium** ce conține o suită de instrumente pentru automatizarea testelor de aplicații web, ce permite dezvoltatorilor să simuleze acțiunile utilizatorului pe un browser web real sau simulat, permițând astfel testarea automată a funcționalității aplicațiilor web.
- Instalarea acestor librării poate fi realizată folosind pip, care este gestionarul de pachete Python standard. Iată cum se instalează câteva dintre librăriile menționate: Pentru instalarea Selenium, puteți utiliza comanda:

PARTEA II – ASPECTE PRACTICE

DE CE ESTE IMPORTANTĂ TESTAREA AUTOMATĂ? AVANTAJE?

- Testarea automată este un proces esențial în dezvoltarea software, care aduce numeroase beneficii. Iată câteva avantaje ale testării automate:

PARTEA II – ASPECTE PRACTICE

DACĂ AI ALES O ANUMITĂ METODOLOGIE, SCRIE CÂTEVA CUVINTE DESPRE EA. CE REPREZINTĂ ȘI DE CE AI ALES-O? CARE SUNT AVANTAJELE?

PARTEA II – ASPECTE PRACTICE

DACĂ AI ALES UN ANUMIT DESIGN PATTERN EXPLICĂ CARE ESTE ACESTA ȘI CARE SUNT AVANTAJELE ALEGERII ACESTUIA?

PARTEA II – ASPECTE PRACTICE

CUM SE FOLOSEȘTE PROIECTUL? CUM SE CLONEAZĂ?
CUM SE RULEAZĂ TESTELE? CUM SE ACCESEAZĂ RAPOARTELE?

PARTEA II – ASPECTE PRACTICE

SĂ APARĂ SCREENSHOTS DIN COD CU EXPLICAȚII MINIME ȘI POZE CU RAPOARTELE

În imaginile alăturare se poate observa rezultatul obținut în urma rulări a celor :

- **5 Feature-uri** ce conține:
  - **9 Scenarii**
  - **1 Scenariu Outtline**

PARTEA II – ASPECTE PRACTICE

POȚI DESCRIE SCENARIILE DE TEST IMPLEMENTATE.

PARTEA II – ASPECTE PRACTICE

POȚI DESCRIE SCENARIILE DE TEST IMPLEMENTATE.

VĂ MULȚUMESC PENTRU ATENȚIE!