Praktična vežba: nivoi testiranja

Cilj: Na istoj HTML/JavaScript aplikaciji razlikovati jedinično, integraciono, sistemsko i prihvatno testiranje. Nije potrebna instalacija Node.js-a niti Vitest-a.

Zadatak 1. Priprema projekta

U VS Code-u napravite folder TestiranjePopusta i u njemu tri fajla: index.html, script.js i test.js. Sačuvajte sve fajlove u istom folderu. Otvorite index.html u pregledaču.

Zadatak 2. Napravite korisnički interfejs (index.html)

```
<!DOCTYPE html>
<html lang="sr">
<head>
    <meta charset="UTF-8">
    <title>Testiranje popusta</title>
</head>
<body>
    <h2>Obracun popusta</h2>
    <label>Pocetna cena:</label>
    <input type="number" id="cena">
    <br><br>
    <label>Popust (%):</label>
    <input type="number" id="popust">
    <br><br>
    <button onclick="PrikaziCenu()">Izracunaj</button>
    <p id="rezultat"></p>
    <hr>
    <button onclick="PokreniTestove()">Pokreni testove</button>
    <p id="testRezultati"></p>
    <script src="script.js"></script>
    <script src="test.js"></script>
</body>
</html>
```

Proverite: Da li se prikazuju dva polja i dva dugmeta? Zašto se script.js učitava pre test.js?

Zadatak 3. Napišite funkcije koje testiramo (script.js)

```
function IzracunajPopust(cena, procenat) {
    return cena * procenat / 100;
}

function KonacnaCena(cena, procenat) {
    var popust = IzracunajPopust(cena, procenat);
    return cena - popust;
}

function PrikaziCenu() {
    var poljeCena = document.getElementById("cena");
    var poljePopust = document.getElementById("popust");
    var cena = Number(poljeCena.value);
    var procenat = Number(poljePopust.value);

    if (poljeCena.value == "" || poljePopust.value == "" ||
        cena < 0 || procenat < 0 || procenat > 100) {
        document.getElementById("rezultat").innerHTML =
            "Neispravan unos!";
        return;
    }

    var rezultat = KonacnaCena(cena, procenat);
    document.getElementById("rezultat").innerHTML =
        "Konacna cena: " + rezultat + " RSD";
}
```

Isprobajte: Unesite cenu 2000 i popust 20. Kliknite na „Izracunaj“. Očekivani rezultat je 1600 RSD. Zatim unesite popust 120 i proverite poruku o grešci.

Zadatak 4. Jedinični test (test.js)

Jedinični test proverava jednu funkciju. Proveravamo samo IzracunajPopust(), bez korišćenja forme.

```
function TestJedinicni() {
    var stvarni = IzracunajPopust(2000, 20);
    var ocekivani = 400;

    if (stvarni == ocekivani) {
        return "PASSED";
    }
    else {
        return "FAILED";
    }
}
```

Zadatak 5. Integracioni test (nastavak test.js)

Testiramo saradnju funkcija KonacnaCena() i IzracunajPopust(). Ovo je pojednostavljen školski primer integracionog testiranja.

```
function TestIntegracioni() {
    var stvarni = KonacnaCena(2000, 20);
    var ocekivani = 1600;

    if (stvarni == ocekivani) {
        return "PASSED";
    }
    else {
        return "FAILED";
    }
}
```

Zadatak 6. Pokretanje oba testa (nastavak test.js)

```
function PokreniTestove() {
    var rezultat1 = TestJedinicni();
    var rezultat2 = TestIntegracioni();

    document.getElementById("testRezultati").innerHTML =
        "Jedinicni test: " + rezultat1 +
        "<br>Integracioni test: " + rezultat2;
}
```

Pokrenite: Osvežite stranicu i kliknite na „Pokreni testove“. Očekujete PASSED za oba testa.

Zadatak 7. Namerno izazovite grešku

U fajlu script.js privremeno promenite deljenje sa 100 u deljenje sa 10:

```
return cena * procenat / 10;
```

Sačuvajte fajl, osvežite stranicu i ponovo pokrenite testove. Zapišite koji su testovi pali i zašto. Zatim vratite / 100 da bi aplikacija ponovo bila ispravna.

Zadatak 8. Samostalno dodajte još dva testa

U test.js napravite funkcije TestJedinicni2() i TestIntegracioni2() za cenu 3000 i popust 10%. Očekivani iznos popusta je 300, a konačna cena 2700. Dopunite PokreniTestove() tako da prikaže rezultate sva četiri testa.

Zadatak 9. Sistemsko testiranje - proverite celu aplikaciju

Otvorite stranicu u pregledaču. Za svaki red unesite podatke u formu, kliknite na „Izracunaj“ i upišite stvarni rezultat. Proveravate unos, izračunavanje i prikaz kao jednu celinu.

Cena	Popust	Očekivani rezultat	Stvarni rezultat / PASSED ili FAILED
2000	20	1600 RSD	_____
1000	0	1000 RSD	_____
500	100	0 RSD	_____
-100	10	Neispravan unos!	_____
1000	120	Neispravan unos!	_____
prazno	20	Neispravan unos!	_____

Zadatak 10. Prihvatno testiranje - uloga naručioca

Zamislite da ste vlasnik prodavnice koji je naručio aplikaciju. Proverite da li aplikacija ispunjava sledeće zahteve i označite DA ili NE.

Zahtev naručioca	Ispunjeno?
Računa cenu nakon popusta.	DA / NE
Prihvata procenat popusta od 0 do 100.	DA / NE
Odbija negativnu početnu cenu.	DA / NE
Prikazuje konačnu cenu korisniku.	DA / NE