Titlu: Online Videogame Store

Smart, Darren James

Aceasta este o aplicație web Flask scrisă în Python, HTML, CSS, JavaScript, SQL și folosește PostgreSQL ca sistem de gestionare a bazei de date.

Aplicația conține rute către o pagină "Home", o pagină "All Products", o pagină "Contact", o pagină de produs "Product/<prod\_id>" pentru un produs selectat în cadrul rutei "All Products", o rută "Category" accesată de pe pagina oricărui produs ales, o rută "Search" și o rută spre coș "Basket". Există și o rută suplimentară ’ascunsă’ numită "Easter Egg" care redă automat un film tematic de joc video.

Toate paginile sunt renderizate ca extensii ale fișierului "Layout.html" și populate cu date folosind motorul Jinja 2.

Funcțiile create special în fișierul "image\_converter.py" au fost create pentru a permite convertirea imaginilor în date de tip bytea și stocarea acestora direct în baza de date. La rularea aplicației din fișierul "Online\_store\_webapp.py", funcția "Filefinder()" creează mai întâi o listă cu toate căile imaginilor din /database/images/ folosind funcția "os.walk()".

Funcția "img\_to\_bytea()" cu modulele "psycopg2" și "io" se conectează apoi la baza de date și, folosind o buclă "for" cu o declarație "if" pentru a preveni duplicarea, traverseaza lista căilor de imagine și convertește fiecare imagine în date de tip bytea înainte de a le stoca în tabela "images", împreună cu un nume pentru fiecare imagine extras din numele fișierului.

"bytea\_to\_img()" acceptă date binare de imagine și utilizează funcțiile "base64encode()" și "decode('utf-8')" din modulul base64 pentru a converti datele bytea într-un șir codificat în base64, pe care browser-ul poate să-l interpreteze ca imagine.

"dict\_maker()" creează un dicționar al tuturor informațiilor despre produse, inclusiv imagini, pentru utilizare în ruta "Toate produsele". Aceasta se face prin interogarea bazei de date pentru toate rândurile legate de toate produsele și crearea de elemente în dicționar unde cheia este id-ul produsului și valoarea este un dicționar cu toate datele legate de acel id de produs; cu datele binare ale imaginii convertite într-un format utilizabil folosind "bytea\_to\_img()". Apoi, o buclă "for" Jinja 2 în "all\_products.html" traduce toate datele pentru fiecare produs din dicționar pentru a fi afișate pe pagină.

Au fost create 3 script-uri JavaScript mici în acest proiect:

"script.js" verifică pur și simplu dacă browser-ul este în modul "Light" sau "Dark" folosind "window.matchMedia" și schimbă favicon-ul la versiunea luminoasă sau întunecată.

"image.js" a fost creat pentru a permite afisarea/ascunderea imaginii afișate în pagina "Produs/<prod\_id>".

"basket.js" utilizează un "EventListener" pentru a actualiza stocul disponibil afișat după scăderea numărului de articole adăugate în coș și afișează o alertă pentru a confirma adăugarea articolului (sau articolelor) în coș.

"styles.css" conține toate stilurile utilizate în proiect.