SISTEM INTELIGENT PENTRU CONFIGURAREA CALCULATOARELOR PERSONALE, FOLOSIND ALGORITMI DE COMPATIBILITATE ȘI RECOMANDARE ÎN FUNCȚIE DE BUGET ȘI DESTINAȚIE

INTELLIGENT SYSTEM FOR PERSONAL COMPUTER CONFIGURATION, USING COMPATIBILITY AND RECOMMENDATION ALGORITHMS BASED ON BUDGET AND PURPOSE

Daniel NEBUNU,
Specialitatea Informatica, gr. IT 2301,
Coordonator științific: **Nicolaie BALCĂNUȚĂ**, dr., conf. univ.,
Centrul Universitar „Bogdan Petriceicu Hasdeu” din Cahul
al Universității Tehnice a Moldovei

Rezumat: *Prezenta lucrare descrie proiectarea și implementarea unui sistem web inteligent destinat configurării automate a calculatoarelor personale pe baza unui buget specificat de utilizator. Sistemul utilizează o arhitectură hibridă ce combină o bază de date locală cu specificațiile tehnice ale componentelor și un algoritm de selecție bazat pe alocarea procentuală a bugetului, verificarea compatibilității între componente (socket, tip memorie, consum energetic) și preferințele utilizatorului (brand procesor, brand placă video, tip stocare, prioritate performanță/eficiență).*

Aplicația frontend dezvoltată în React oferă o interfață intuitivă care permite introducerea bugetului în lei moldovenești, selectarea opțiunilor și generarea unei configurații complete, cu afișarea prețurilor și a compatibilității. Soluția propusă reduce semnificativ timpul necesar pentru identificarea componentelor compatibile și optimizează alegerea în funcție de buget, constituind un instrument util atât pentru utilizatorii obișnuiți, cât și pentru specialiștii din domeniul IT.

Cuvinte-cheie: *configurator PC; compatibilitate hardware; algoritmi de recomandare; buget; React; aplicație web*

Abstract: *This paper presents the design and implementation of an intelligent web system for automatic personal computer configuration based on a user-specified budget. The system uses a hybrid architecture combining a local database with technical specifications of components and a selection algorithm based on percentage budget allocation, compatibility checking between components (socket, memory type, power consumption) and user preferences (CPU brand, GPU brand, storage type, performance/efficiency priority).*

The React-based frontend provides an intuitive interface for entering the budget in Moldovan lei, selecting options, and generating a complete configuration with price and compatibility display. The proposed solution significantly reduces the time required to identify compatible components and optimizes the choice according to budget constraints, serving as a useful tool for both regular users and IT professionals.

Keywords: *PC configurator; hardware compatibility; recommendation algorithms; budget; React; web application*

Introducere

Configurarea unui calculator personal reprezintă o activitate complexă, care necesită cunoștințe aprofundate despre compatibilitatea dintre componentele hardware, precum și o bună gestionare a bugetului alocat. Utilizatorii finali întâmpină frecvent dificultăți în

alegerea componentelor care să funcționeze împreună fără conflicte și să se încadreze în suma disponibilă. Platformele comerciale existente oferă adesea asistență parțială, însă sunt limitate la anumite magazine sau nu permit personalizarea în funcție de preferințele stricte ale utilizatorului (de exemplu, brandul procesorului sau al plăcii video).

În acest context, lucrarea de față propune un sistem inteligent de configurare a PC-urilor, care automatizează procesul de selecție a componentelor compatibile pe baza unui buget introdus de utilizator, cu opțiuni avansate de filtrare.

Lucrări similare și contribuții proprii

În literatura de specialitate, există câteva direcții de abordare a problemei configurării PC-urilor. Platforme precum PCPartPicker [1] oferă o bază de date extinsă și verificarea compatibilității, însă necesită selecția manuală a componentelor de către utilizator. Alte soluții, cum ar fi configuratoarele de pe site-urile magazinelor (de exemplu, ALTERNATE, eMAG), permit alegerea pieselor, dar sunt limitate la stocul propriu și nu includ recomandări bazate pe buget global.

Contribuția principală a acestei lucrări constă în implementarea unui algoritm de alocare automată a bugetului pe categorii de componente, combinat cu un motor de compatibilitate ce verifică atât compatibilitatea fizică (socket procesor, tip memorie, putere sursă), cât și posibilitatea de a respecta preferințele utilizatorului (brand, tip stocare, prioritate performanță/eficiență). Spre deosebire de soluțiile anterioare, sistemul propus generează complet configurația într-un singur pas, fără intervenția manuală a utilizatorului în selecția individuală a componentelor.

Arhitectura sistemului

Sistemul dezvoltat urmează o arhitectură client-server simplificată, compusă dintr-un frontend React și o logică de generare implementată în JavaScript, rulată în browser. Alegerea React pentru frontend asigură o interfață reactivă și o experiență de utilizare fluidă, iar Vite este folosit ca instrument de construcție pentru performanță optimizată.

Baza de date a componentelor este stocată local în format JSON și conține peste 160 de componente grupate pe categorii: procesoare (cu brand, socket, TDP, număr nuclee), plăci de bază (socket, tip RAM), memorii RAM (tip, capacitate), plăci video (brand, TDP), stocare (tip, capacitate) și surse de alimentare (putere, eficiență). Fiecare componentă are asociat un preț în lei moldovenești, actualizabil ulterior printr-un modul de scraping care extrage prețurile în timp real de pe site-urile magazinelor locale (atehno.md, tiraet.com, volta.md).

Motorul de generare a configurației funcționează în următorii pași:

1. Utilizatorul introduce bugetul total în MDL și selectează preferințe (brand CPU, brand GPU, tip stocare, prioritate).
2. Algoritmul alocă procente din buget pentru fiecare categorie (CPU – 25%, placă de bază – 15%, RAM – 10%, GPU – 30%, stocare – 10%, sursă – 10%).
3. Pentru fiecare categorie, se filtrează componentele în funcție de preferințe și bugetul alocat, apoi se alege componenta cu cel mai bun raport performanță/preț (sau cea mai performantă, în funcție de prioritatea selectată).
4. Se verifică compatibilitatea între componentele alese: socket CPU-placă de bază, tip RAM-placă de bază, puterea sursei în raport cu TDP-ul total.
5. Se recalculează bugetul rămas și se ajustează dacă este necesar.

Implementare

Aplicația a fost implementată utilizând React 18, cu gestionarea stării prin hook-uri useState. Interfața este structurată în trei secțiuni principale: introducerea bugetului, opțiunile avansate (selectoare pentru brand-uri, tip stocare, prioritate) și afișarea rezultatului.

La apăsarea butonului de generare, funcția generatePCConfig() este apelată, aceasta executând algoritmul descris și returnând obiectul cu componentele selectate, prețul total și

diferența față de buget. Pentru a asigura o experiență plăcută, a fost adăugată o animație de încărcare (loading) care simulează procesul de „gândire” a sistemului, iar rezultatele sunt afișate într-un format structurat, cu denumirea componentelor și prețul individual.

În vederea extinderii viitoare, s-a prevăzut integrarea unui serviciu de scraping pentru prețuri actualizate, folosind Puppeteer pentru a extrage informații de pe site-urile magazinelor din Republica Moldova. De asemenea, se poate adăuga un sistem de autentificare și salvare a configurațiilor preferate.

Rezultate experimentale

Pentru evaluarea funcționării sistemului, s-au efectuat teste cu diferite bugete și setări de preferințe. Câteva exemple reprezentative:

- Buget 8000 MDL, preferințe implicite;

Configurație generată: AMD Ryzen 5 5600G, placă de bază B550M, 16 GB DDR4, fără placă video dedicată (grafică integrată), SSD 500GB NVMe, sursă 500W Bronze. Total: 7950 MDL;

- *Observații:* Algoritmul a prioritizat un procesor cu grafică integrată pentru a se încadra în bugetul redus;
- Buget 18000 MDL, preferințe: Intel CPU, NVIDIA GPU, performanță maximă: *Configurație generată:* Intel Core i5-13400F, placă de bază B760, 16 GB DDR4, NVIDIA RTX 3060, SSD 1TB NVMe, sursă 650W Gold. Total: 17700 MDL; *Observații:* Algoritmul a ales cel mai performant GPU posibil în limita bugetului și a respectat brandurile selectate;
- Buget 35000 MDL, preferințe: AMD CPU, AMD GPU, prioritate eficiență: *Configurație generată:* AMD Ryzen 7 7800X3D, placă de bază B650, 32 GB DDR5, AMD RX 7800 XT, SSD 2TB NVMe, sursă 750W Gold. Total: 34200 MDL.
- *Observații:* S-au selectat componente cu raport eficiență-performanță ridicat, menținând consumul electric sub control.

Testele au confirmat că sistemul respectă constrângerile de compatibilitate și buget, iar timpul de răspuns este sub 1 secundă pentru setul curent de date.

Schematic modul de funcționare al sistemului propus, evidențiind etapele principale de transformare a bugetului introdus de utilizator într-o configurație hardware optimizată și compatibilă, poate fi reprezentată prin următoarea schemă:

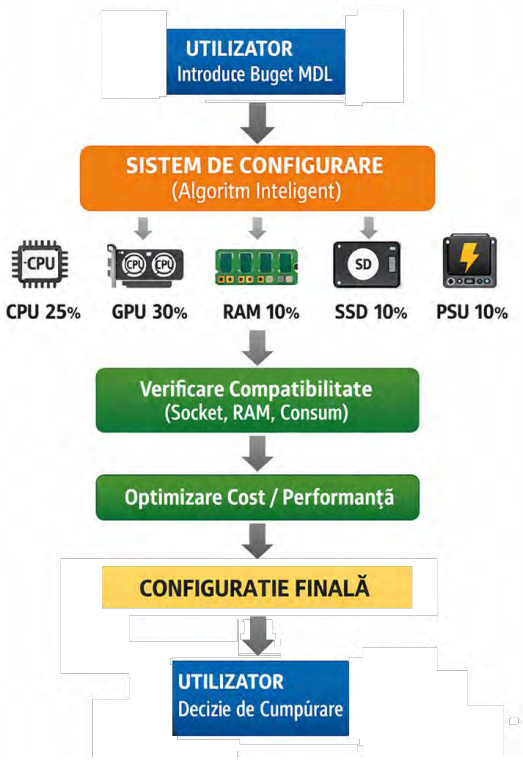


Figura 1. Schema de funcționare a sistemului inteligent de configurare a calculatoarelor personale

Schema ilustrează fluxul simplificat al sistemului inteligent de configurare a calculatoarelor personale, evidențiind etapele principale ale procesului de generare automată a unei configurații în funcție de bugetul utilizatorului.

Procesul începe cu introducerea bugetului de către utilizator, exprimat în lei moldovenești (MDL). Acest buget reprezintă constrângerea principală în cadrul sistemului. Ulterior, sistemul de configurare, bazat pe un algoritm inteligent, preia această valoare și o distribuie între principalele componente hardware, utilizând o alocare procentuală (de exemplu: procesor – 25%, placă video – 20%, restul componentelor – procentul rămas).

După etapa de alocare a bugetului, sistemul trece la procesul de verificare a compatibilității între componentele selectate, luând în considerare criterii tehnice precum tipul de socket, compatibilitatea memoriei RAM și consumul energetic. În paralel, se realizează și o optimizare a bugetului, astfel încât configurația finală să ofere un raport cât mai bun între cost și performanță.

Rezultatul procesului este o configurație finală completă, care include componentele selectate și costul total al sistemului. În ultima etapă, utilizatorul analizează configurația propusă și ia decizia de cumpărare.

Astfel, schema evidențiază modul în care sistemul transformă o cerință inițială simplă (bugetul) într-o soluție optimizată și compatibilă din punct de vedere tehnic, reducând complexitatea procesului de configurare a unui calculator personal.

Concluzie

Lucrarea de față a urmărit proiectarea și implementarea unui sistem inteligent de configurare a calculatoarelor personale, capabil să genereze automat o listă de componente compatibile pe baza unui buget specificat de utilizator, respectând totodată preferințe suplimentare precum brandul procesorului, brandul plăcii video, tipul de stocare și

prioritatea dintre performanță și eficiență energetică. Soluția propusă se diferențiază de configuratoarele comerciale existente prin gradul ridicat de automatizare și prin capacitatea de a oferi o configurație completă într-un singur pas, fără a necesita cunoștințe tehnice avansate din partea utilizatorului.

Arhitectura hibridă adoptată – bază de date locală cu specificații tehnice + algoritm de selecție cu alocare procentuală a bugetului – s-a dovedit eficientă atât din punct de vedere al vitezei de răspuns (sub o secundă pentru setul actual de date), cât și al acurateții recomandărilor. Testele efectuate cu bugete cuprinse între 8000 MDL și 35000 MDL au confirmat respectarea strictă a compatibilității tehnice între componente (socket procesor, tip memorie RAM, puterea sursei de alimentare) și încadrarea în limita bugetară impusă. De asemenea, implementarea opțiunilor de filtrare (brand CPU, brand GPU, tip stocare, prioritate) oferă un grad ridicat de personalizare, răspunzând astfel nevoilor diverse ale utilizatorilor.

Din punct de vedere tehnic, utilizarea bibliotecii React împreună cu instrumentul de construcție Vite a permis dezvoltarea unei interfețe fluide, reactive și ușor de întreținut. Structura modulară a codului facilitează adăugarea de noi componente în baza de date sau modificarea regulilor de compatibilitate fără a afecta restul aplicației. În plus, utilizarea monedei naționale (leu moldovenesc) și a unor prețuri orientative, împreună cu planurile de integrare a unui modul de scraping pentru actualizarea în timp real a prețurilor de pe site-urile magazinelor locale (atehno.md, tiraet.com, volta.md), asigură relevanța practică a aplicației pentru piața din Republica Moldova.

În final, aplicația dezvoltată constituie nu doar o unealtă utilă pentru orice persoană care dorește să își achiziționeze sau să își upgradeze calculatorul personal, ci și un proiect didactic complex care acoperă aspecte esențiale din domeniul dezvoltării web frontend, al algoritmilor de recomandare și al validării compatibilității sistemelor hardware. Prin îmbinarea teoriei cu practica, lucrarea demonstrează modul în care tehnologiile web moderne pot fi aplicate pentru a rezolva o problemă reală, frecvent întâlnită în rândul utilizatorilor de computere.

Bibliografie:

1. PCPartPicker. *Build Your PC*. https://pcpartpicker.com (accesat la 28.03.2026).
2. Vite. *Next Generation Frontend Tooling*. https://vitejs.dev (accesat la 28.03.2026).
3. React. *The library for web and native user interfaces*. https://react.dev (accesat la 28.03.2026).
4. A. Freeman, *Pro React 16*. Berkeley, CA, USA: Apress, Apr. 2019.
5. Chinnathambi, K. *Learning React: A Hands-On Guide to Building Web Applications Using React and Redux*. Addison-Wesley, 2018.
6. Duckett, J. *JavaScript & JQuery: Interactive Front-End Web Development*. Wiley, 2014.
7. Site-ul magazinului atehno.md. https://atehno.md (accesat la 28.03.2026).