

# Proiect web dinamic cu baze de date

- Proiect facultativ la disciplina Programare Java -

## Motivație

Exersarea tehnologiilor *Spring Boot*, *Spring framework*, *Spring MVC*, *Spring Data JPA*, *Spring Security*, etc care au fost studiate la curs și laborator și care sunt utilizate în proiect determină creșterea șanselor de a obține un *Internship* sau un post de programator Java ca și Junior pentru că sunt tehnologii utilizate în industrie.

Proiectul pune bazele necesare dezvoltării unui proiect de diplomă mai complex.

Implementarea proiectului presupune completarea studiului făcut la școală, în cadrul disciplinei, cu studiu individual pentru lămurirea unor aspecte necesare dezvoltării.

Studentii care rezolvă complet proiectul primesc un punct în plus la nota de la examen. Pentru rezolvări parțiale se acordă 0.5 în plus.

## Modul de lucru sugerat

Se recomandă utilizarea unuia din IDE-urile IntelliJ sau Eclipse.

Se recomandă utilizarea sistemului de versionare Git și a platformei GitHub, proiectul fiind urcat într-un depozit **privat** în **GitHub**. Se recomandă salvarea periodică a modificărilor aduse proiectului în depozitul **Git** local și în depozitul la distanță **GitHub**, prin utilizarea comenzilor **Commit** și **Push**. În acest fel poate să fie urmărit istoricul dezvoltării și se poate reveni la una din versiunile mai vechi.

## Predarea și evaluarea proiectului

Proiectul se predă cel târziu în săptămâna 12, având timp la dispoziție pentru implementare trei săptămâni, dar poate să fie predat mai devreme de termenul limită în cadrul orelor de laborator. Evaluarea proiectului presupune rularea acestuia și demonstrarea bunei funcționări, prezentarea codului și a istoricului dezvoltării, răspunsul la întrebări legate de modul de dezvoltare.

## Important

Studentii care doresc să rezolve proiectul facultativ discută cu cadrul didactic și își anunță intenția până la finalul săptămânii 9. Nu se acordă bonus studenților care cu foarte puțin timp înainte de termenul limită afirmă "si eu l-am rezolvat...".

## Enunț proiect

Să se realizeze un proiect web dinamic care permite gestiunea simplificată a mașinilor din parcul auto al unei firme. Accesul în aplicație se realizează prin utilizator și parolă. Persoanele autorizate să acceseze aplicația pot avea următoarele roluri:

- rolul de utilizator (ROLE\_USER) care poate să vizualizeze datele despre mașini și poate le filtreze aplicând filtre dinamice
- rolul de editor (ROLE\_EDITOR) poate vizualiza și filtra datele despre mașini, dar în plus poate și adăuga, modifica sau șterge mașini.

Pentru simplitate, conturile utilizatorilor se vor salva în baza de date prin comenzi SQL completate în fișierul *data.sql* (opțional poate să fie făcut încă un rol, cel de administrator, care gestionează conturile). Tot opțional se poate dezvolta și posibilitatea de înregistrare în aplicație în vederea obținerii unui cont de utilizator.

Aplicația va avea două tabele MySQL:

- **utilizatori** cu câmpurile id\_utilizator – cheie primara si autoincrement, nume, utilizator, parola - se va salva criptograma parolei obținută prin criptarea cu *BCryptPasswordEncoder()*, rolul (cu valorile posibile *ROLE\_USER*, *ROLE\_EDITOR*)
- **masini** (nr\_inmatriculare – cheie primara), id\_utilizator (id-ul utilizatorului care a adăugat mașina), marca, modelul, culoarea, anul\_fabricatiei, capacitatea\_cilindrica, tipul\_de\_combustibil, puterea, cuplul, volumul\_portbagajului, pretul

Aplicația va avea un aspect asemănător cu cel din capturile de mai jos:

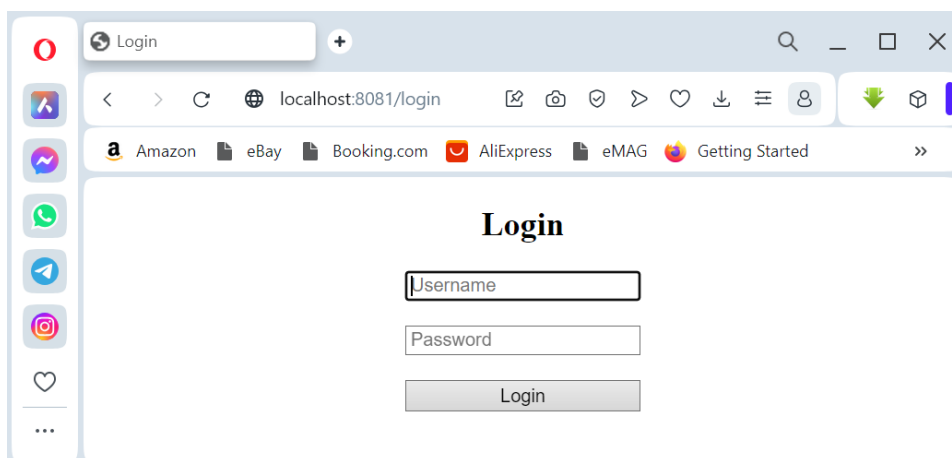


Fig. 1. Accesul în program

Dacă se introduce greșit utilizatorul și / sau parola se va afișa un mesaj corespunzător:

## Login

Invalid username or password.

Fig. 2. Accesul nepermis în aplicație

După autentificarea cu succes se va afișa un mesaj de primire însoțit de numele celui care s-a autentificat (vezi figurile 3 și 6).

Pentru utilizatorul autentificat cu succes care are rolul *ROLE\_USER* se afișează mașinile din parcul auto, acesta putând să le filtreze după trei câmpuri la alegere. Se vor alege 3 câmpuri care vor fi folosite pentru a filtra datele. Deasupra de tabel se vor afișa mesaje care prezintă ce afișează tabelul (de exemplu: Toate mașinile (vezi figura 3), Mașinile cu marca=ford și combustibil=benzina (vezi figura 4), etc). Se va construi o interogare dinamică în funcție de valorile introduse în casete. Dacă într-o anumită casetă de text nu s-a introdus nici o valoare, atunci câmpul corespunzător nu apare în interogare. Dacă se introduc valori în două casete, câmpurile corespunzătoare vor apărea în interogare unite prin și logic. Apăsarea butonului de filtrare fără a completa nici un fel de informație în casetele de filtrare va determina afișarea tuturor mașinilor din parcul auto.

The screenshot shows a web browser window with the address bar displaying 'localhost:8081/masini'. The page title is 'Lista masini'. The main content area displays a welcome message 'Bine ai venit, Ion Ionescu' in a large, bold font. Below the message, there are three input fields labeled 'marca', 'culoare', and 'combustibil', followed by a 'Filtrează' button. Underneath the filters, the text 'Toate mașinile' is displayed. A table with 11 columns and 7 rows of car data is shown. The columns are: Număr înmatriculare, Marca, Model, Culoare, An fabricație, Capacitate cilindrică, Combustibil, Putere, Cuplu, Volum portbagaj, and Preț. The rows contain data for various car models like ford mondeo, seat leon, and ford focus. At the bottom of the page, there is a 'Logout' link.

Număr înmatriculare	Marca	Model	Culoare	An fabricație	Capacitate cilindrică	Combustibil	Putere	Cuplu	Volum portbagaj	Preț
TM12AAA	ford	mondeo	albastru	2014	2000	electric	140	180	600	9000
TM14CSS	seat	leon	galben	2015	1800	benzina	150	120	350	10500
TM14ZYA	seat	leon	negru	2015	1800	motorina	150	120	350	11000
TM20RRR	ford	focus	negru	2010	1997	motorina	170	136	400	8888
TM30XZY	ford	mondeo	negru	2014	2000	benzina	140	180	600	9500
TM99JKW	ford	focus	albastru	2010	1997	benzina	170	136	400	9999

Fig.3 Nivelul de acces utilizator

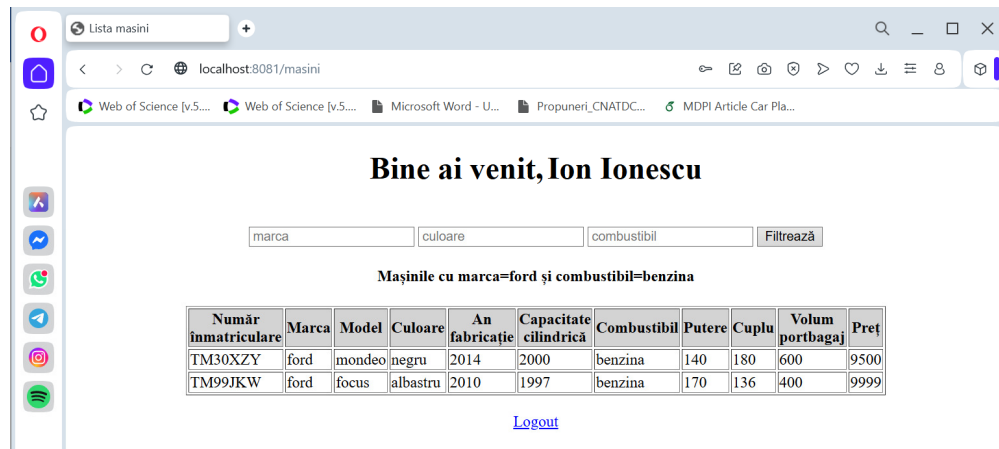


Fig.4 Filtrare după marcă și combustibil

Aționarea butonului logout va determina închiderea sesiunii și redirectarea către pagina de login, însoțită de mesajul de mai jos:

### Login

You have been logged out.




Fig. 5 Ieșirea din program

Utilizatorul cu rolul de editor va beneficia de o interfață asemănătoare cu cea din captura de ecran de mai jos. Operația de modificare actualizează datele mașinii cu numărul de înmatriculare specificat în caseta de text. Se vor completa doar casetele de text aferente câmpurilor care se modifică, celelalte vor rămâne necompletate. Operația de ștergere utilizează doar informația completată în caseta de text număr de înmatriculare.

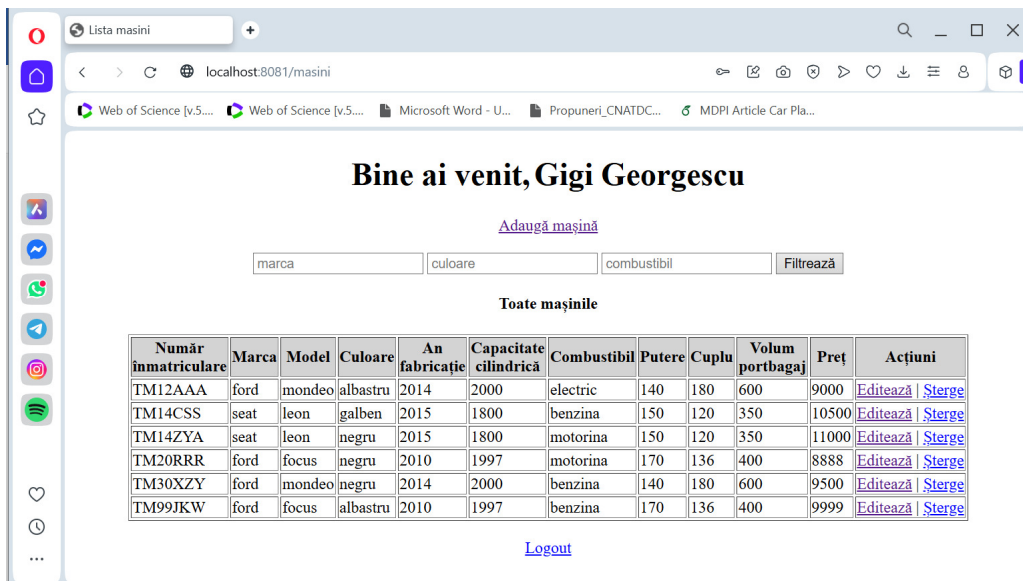


Fig. 6. Nivelul de acces editor

## Indicații de rezolvare

Se va genera un proiect Spring Boot utilizând *Spring Initializr*. Proiectul generat va fi un proiect Maven care utilizează limbajul Java, versiunea 17. Se recomandă utilizarea următoarelor dependențe:

**Dependencies** ADD DEPENDENCIES... CTRL + B

**Spring Web** WEB  
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

**Spring Data JPA** SQL  
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

**MySQL Driver** SQL  
MySQL JDBC driver.

**Lombok** DEVELOPER TOOLS  
Java annotation library which helps to reduce boilerplate code.

**Spring Boot DevTools** DEVELOPER TOOLS  
Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

**Thymeleaf** TEMPLATE ENGINES  
A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

**Spring Security** SECURITY  
Highly customizable authentication and access-control framework for Spring applications.

Fig. 6 Dependențele proiectului

Operațiile asupra bazei de date *MySQL* se vor realiza folosind *Spring Data JPA*. Se recomandă dezvoltarea front-end-ului proiectului web dinamic utilizând HTML + Thymeleaf, dar se acceptă dezvoltarea front-end-ului utilizând React, Angular sau Java Server Pages (JSP-uri).

Se recomandă structurarea aplicației pe mai multe nivele, făcându-se câte un pachet pentru fiecare nivel:

- Nivelul entităților - conține entitățile (clasele persistente) care se mapează la tabele
- Nivelul repository – urilor - conține repository-urile Spring Data JPA, oferă acces la date, operează asupra entităților oferind operațiile CRUD
- Nivelul serviciilor - utilizează repository-urile pentru a interacționa cu baza de date și implementează buisness logic-ul aplicației

- Nivelul controllerelor - gestionează cererile și răspunsurile HTTP și expune API-uri pentru a interacționa cu serviciile și a manipula datele.
- Nivelul de securitate care protejează datele și controlează accesul la resursele aplicației.

Se recomandă dezvoltarea unor unități de testare a aplicației.

Se apreciază completarea funcționalităților minime de mai sus cu noi funcționalități.

## **Enunț alternativ**

Se pot accepta propuneri de proiect din partea studenților dacă acestea respectă modul de lucru sugerat, tehnologiile propuse și sunt cel puțin la fel de complexe.