FACULTATEA DE AUTOMATICA SI CALCULATOARE

MY FUTURE CAR

Document de proiectare arhitecturala

Student: Ion Luminita-Andreea

Grupa: 341C4

1. Descrierea temei

Proiectul consta in implementarea intr-o aplicatie de gestiune a unui magazin de automobile.

Aplicatia cuprinde:

* o baza de date creata in mySQL
* un server pentru backend creat in Python3 cu ajutorul bibiloteca Flask
* un server web creat in Dart folosind framework-ul Flutter

Graphical user interface, application

Description automatically generated

1. Descrierea bazei de date
   1. Diagrama bazei de date

Graphical user interface, application, Teams

Description automatically generated

* 1. Structura tabelelor

Baza de date este structurata in 6 tabele, dintre care unul principal care le foloseste pe celelalte 5 mai mici.

Text

Description automatically generatedTabelul CAR\_TYPE:

* Tine datele despre tipul masinii (sedan, break, pickup, truck, suv) -> CAR\_TYPE\_NAME
* Fiecarui tip i se asociaza un id cu care se va lucra mai departe -> CAR\_TYPE\_ID

Text

Description automatically generatedTabelul COMBUSTIBIL:

* Fiecarui tip de combustibil (motorina, benzina, electric, gas, hybrid) -> COMBUSTIBIL\_NAME ii este asociat un id unic -> COMBUSTIBIL\_ID

Text

Description automatically generatedTabelul GEARBOX:

* Aici tinem tipul cutiei de viteze pe care o masina il poate avea
* In GEARBOX\_CODE – tinem codul unic al unui tip de cutie de viteze
* In GEARBOX\_TYPE este tinuta denumirea efectiva a tipului de cutie de viteze (automata sau manuala)

Graphical user interface, text

Description automatically generated Tabelul BRAND:

* Aici tinem brandurile disponibile pe site pana acum
* BRAND\_CODE este de tipul varchar si va tine o prescurtare a numelui brandului
* BRAND\_NAME va fi campul in care se va tine numele intreg al brandului

Text

Description automatically generatedTabelul COLOR:

* Fiecare culoare (COLOR\_NAME) are asociat un id unic (COLOR\_CODE)

Text

Description automatically generated

Tabelul CARS

* Continue un id unic pentru fiecare masina introdusa (ID\_CAR)
* Campuri de tipul MODEL\_NAME, CAR\_YEAR, PRICE, NO\_KM, ENGINE\_POWER care sunt proprietati ale unei masini si care nu depind de alte tabele auxiliare
* Campuri de tipul BRAND\_CODE, COMBUSTIBIL\_ID, CAR\_TYPE\_ID, GEARBOX\_CODE, COLOR\_CODE, care reprezinta chei straine in tabelul Cars si care duc prin relatii de one-to-one catre celelalte tabele
  1. Descrierea constrângerilor de integritate

Ca si constrangeri de integritate avem:

* NOT NULL - specifica faptul ca o coloana nu poate contine o valoare NULL
* PRIMARY KEY - identifica unic fiecare inregistrare
* FOREIGN KEY - stabileste o legatura pe baza unei chei externe intre o coloanal din tabel şi o coloana din tabelul referit
  1. Descrierea procedurilor și funcțiilor

Proceduri: am creat proceduri pentru fiecare operatii de tipul insert, delete sau select pentru fiecare tabel.

Text

Description automatically generatedProcedurile de tip add intr-un tabel folosesc operatii de tip insert si sunt folosite pentru a adauga un nou element in tabelul dorit.

Text

Description automatically generatedProcedurile de tip delete folosesc operatii delete pentru a sterge din tabel un anumit element cu id-ul dat de la tastatura.

Text

Description automatically generated

Procedurile de tip select folosesc operatii de tip select pentru a selecta toate campurile dintr-un anumit tabel sau pentru a selecta anumite campuri dupa diferite criterii.

Text

Description automatically generated

Functii: In scriptul sql am doua functii, clasify\_by\_power care in functie de puterea pe care o are o masina (functia primeste puterea ca parametru) si intoarce in care din categoriile “Normal”, “Sport” sau “Passion” se incadreaza masina in functie de putere. Cea de-a doua functie este get\_car\_gearbox care primeste id-ul unei masini din baza de date ca parametru si intoarce ce fel de cutie de viteze are masina cu id-ul dat.

Text

Description automatically generatedTriggerul tgr\_car pune in tabelul mesaje care are un singur camp (MESAJ) un nou mesaj atunci cand s-a adaugat o noua masina in baza de date si cu ce pret s-a adaugat ea.

1. Descrierea aplicației

Aplicatia pune la dispozitie 6 ecrane in care se pot vizualiza masinile aflate in baza de date a magazinului, culorile disponibile, tipurile de masini exitent in baza de date, brandurile si inca 2 interfete care reprezinta de fapt 2 raporturi: ecranul Power Class unde putem filtra masinile dupa statusul pe care il primesc in functie de puterea motorului si My gearbox, unde poti afla ce fel de cutie automata are o anumita masina deja existenta in baza de date.

Diagram, table

Description automatically generated

Pe pagina Cars vedem toate masinile din baza de date si ce caracteristici au

Table

Description automatically generated

Pe pagina Car Type vedem ce tipuri de masini sunt in magazine si ce cod au

Graphical user interface, text, application, email

Description automatically generated

Pe pagina Brand vedem numele Brandurilor masinilor din magazine si codul lor asociat

Graphical user interface, text, application, email

Description automatically generated

Pe pagina Color vedem culorile masinilor si codul asociat lor

Graphical user interface, text, application, email

Description automatically generated

In ecranul Power Class putem vedea masinile filtrate dupa statusul pe care il au. Aici sunt selectate masinile Normale, adica cele care au puterea motorului mai mica sau egala cu 100.

A picture containing table

Description automatically generated

In ecranul My Gearbox avem o text inbox unde introducem id-ul unei masini si avem 2 cazuri:

* Id-ul introdus se afla in baza de date si putem determina ce tip de cutie de viteze are masina cu acel id
* Id-ul nu se afla in baza de date si ne zice asta

Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated

1. Prezentarea modului în care se face conexiunea cu baza de date

Baza de date este una create si populata local folosind un server mySql si m-am folosit de mysql.connector in Flask pentru a face conexiunea cu baza de date si a crea api-urile. Am folosit Flutter ca si framework pentru front-end, unde m-am conectat la url-ul dat de server si am concatenate cu rutele de la api necesare pentru fiecare pagina pe care am creat-o in front-end.

1. Concluzii

Aplicatia este una utila pentru a tine gestuinea unui magazin de masini pentru ca inainte de a vedea fizic masina, clientul se poate orienta dupa anumite caracteristici pe care le doreste folosind aplicatia MyFutureCar. Aplicatia se poate extinde usor, pentru ca orice adaugare in tabele deja existente in baza de date nu aduce consecintele modificarii in frontend, iar modificarea/adaugarea de pagini noi in frontend este destul de usoara pentru ca Flutter permite flexibilitatea si usurinta de a schimba diferite aspecte in pagina fara a scrie prea mult cod.

1. Bibliografie

<https://dev.mysql.com/>

<https://docs.flutter.dev/>