**Progetto di sviluppo “Catalogo Auto”**

**DESCRIZIONE GENERALE**

Si vuole realizzare una applicazione web per la gestione di vari marchi e modelli di auto .

Vi è la possibilità di aggiornare il catalogo auto , inserendo/modificando/eliminando sia le marche che i modelli associati

Tutti i modelli di auto sono catalogati per “brand” , per cui per inserire un modello bisogna far riferimento alla marca di quest’ultimo e , se non presente , procedere dapprima all’inserimento del brand.

**COME PROCEDERE**

L’architettura generale è già definita sia a livello di componenti che di servizi (implementare ulteriori components e services solo se esplictiamente indicato all’interno di questo documento).

Per importare ed utilizzare il progetto è necessario procedere con le seguenti operazioni :

* Attivare un Server NodeJS con un DataBase Oracle (consigliata versione 11g express).
  + Installare NodeJS e Angular
  + Eseguire su Oracle lo script vetture.sql presente nella cartella SQL.
* Aprire due istanze di VisualStudioCode
* Una che punta alla cartella node-ws-vetture :
  + - 1. eseguire 'npm install' e poi 'node server.js'
      2. Verificare sul browser all'indirizzo 127.0.0.1:3000 se il server risponde.
* Una che punta alla cartella ng-vetture :

eseguire 'npm install' e poi 'ng serve'.

L’esercitazione si focalizza principalmente sui servizi di business e la persistenza. Quindi realizzare e testare prima questa e poi realizzare la parte web

**STRUMENTI**

Per realizzare questa applicazione si devono utilizzare i seguenti strumenti:

* VisualStudio Code
* Angular
* Typescript
* SCSS
* Bootstrap

**LISTA FUNZIONALITA’**

Funzionalità per le marche :

1. Aggiunta marca
2. Modifica marca
3. Cancellazione marca
4. Visualizzazione delle marche presenti

Funzionalità per i modelli:

1. Aggiunta Modello
2. Modifica Modello
3. Cancellazione Modello
4. Visualizzazione dei modelli presenti

**Creazione Interfaccia Modello**

File model.ts con le seguenti proprietà :

idModello: number;

    nome: string;

    cilindrata: number;

    potenza: number;

    idMarca: number;

**Creazione ModelDataService con le seguenti functions :**

|  |
| --- |
| insertModel |
| http Method : POST |
| 'http://127.0.0.1:3000/modello/inserisci' |
|  |

|  |
| --- |
| getModelById |
| http Method : GET |
| http://127.0.0.1:3000/modello/leggi-per-id/${id} |

|  |
| --- |
| getModelByName |
| http Method : GET |
| `http://127.0.0.1:3000/modello/leggi-per-nome/${nome}` |

|  |
| --- |
| updateModel |
| http Method : PUT |
| http://127.0.0.1:3000/modello/aggiorna |

|  |
| --- |
| removeModel |
| http Method : DELETE |
| `http://127.0.0.1:3000/modello/rimuovi/${id}` |

|  |
| --- |
| getAllModels |
| http Method : GET |
| 'http://127.0.0.1:3000/modello/leggi-tutto' |

**Creazione components**

* **insertModel (criteri di validazione -> vedi Fig.1.4 cartella “Mockup”)**
* **searchModel**
* **showModels**
* **updateModel**
* **deleteModel**

**Le specifiche grafiche sono indicate nei file .png presenti nella cartella “Mockup” , per quel che riguarda l’implementazione dei components deve essere analoga ai components già presenti nel progetto , ad es. il component “showModels” deve permettere la visualizzazione, cancellazione e aggiornamento dei vari modelli di auto , così come avviene già per il component showBrands .**