

Car Form

توضیحات:

این سایت یک فرم درباره ماشین است که با استفاده از فریمورک spring جاوا نوشته شده است.

این اپلیکیشن به یک دیتابیس mysql وصل میشود و اطلاعات ماشینها را در آن ثبت و از آن واکنشی میکند.

template های این سایت با استفاده از thymeleaf نوشته شده اند.

برای برقراری ارتباط با دیتابیس نیز از jpa و

برای ساخت مدل ها و getter/setter و constructor آنها از lombok استفاده کرده ایم.

صفحه اول:

در صفحه اول این سایت شما با یک فرم مواجه میشوید که در صورتی که اطلاعات را به درستی وارد نمایید، با کلیک کردن بر روی دکمه ی submit اطلاعات وارد شده در دیتابیس رجیستر میشود و شما به صفحه ی بعدی redirect میشوید. شما میتوانید با استفاده از دکمه های view form و یا search بدون وارد کردن اطلاعات و ذخیره یک ماشین جدید به صفحات دیگر بروید.

(فرم ۴ ویژگی درباره ماشین را از شما میخواهد.)

صفحه دوم:

در صفحه ی cars شما میتوانید جدولی که شامل لیست تمامی ماشین های در دیتابیس است را ببینید. همچنین میتوانید با استفاده از دکمه های پایین صفحه به فرم و یا صفحات دیگر بروید.

صفحه سوم:

در صفحه ی search شما میتوانید ماشین های واقع در دیتابیس را بر اساس رنگشان جست و جو نمایید. رنگ مورد نظر خود را (برای مثال Red) وارد کنید و دکمه ی search را کلیک کنید یا enter را فشار دهید. این کار شما را به صفحه ی results هدایت میکند.

صفحه چهارم:

در صفحه ی results شما نتایج جست و جوی خود را میبینید. برای مثال اگر رنگ Blue را جست و جو کرده باشید، تمامی ماشین های آبی داخل دیتابیس را خواهید دید. در این صفحه نیز با استفاده از دکمه های پایین صفحه میتوانید به صفحات دیگر بروید.

توضیحات فایلها:

در پوشه ی `src/main/java/cars/carform` شما ۵ فایل را مشاهده میکنید که هر یک وظیفه ی خاصی را بر عهده دارند.

در `Car.java` مدل ما را با `Entity` های `annotation` و `Data` ساخته ایم. `Entity` مدل ما را به عنوان یک جدول در `db` میسازد و `data`، `getter/setter` و `constructor` ها را میسازد.

در `CarRepository.java` ما یک `interface` از `JpaRepository` ساخته ایم. که وظیفه برقراری ارتباط بین مدل و `DataLayer` یا دیتابیس را دارد.

در `CarController.java` ما یک کنترلر برای مدل ما ساخته ایم که وظیفه ی `handle` کردن `request` ها و دادن اطلاعات از دیتابیس به `template` ها را دارد. برای مثال مشخص میکند که اگر یک `get request` به صفحه ی `search` آمد چه چیزی به `client` برگردانده شود.

هر یک از `GetMapping` و `SetMapping` ها اطلاعات را با استفاده از `CarService` و `CarRepository` از دیتابیس گرفته و به `template` های `thymeleaf` که ساختیم میدهند و صفحه را به `client` میدهند.

در `CarService.java` تمامی `database query` ها پیاده سازی شده اند.

در آخر نیز `CarformApplication.java` برنامه ی ما را اجرا میکند.

در `template` ها نیز اطلاعاتی که از `controller` گرفتیم را در قالب `html` که با `css` آن را `style` کردیم به کاربر نمایش میدهیم. و یا با استفاده از عباراتی مانند `th field` بیان میکنیم که برای مثال مقدار یک `input` به کدام ستون جدول دیتابیس تعلق دارد.