

# پروژه گروه درسی

آرمین ملک محمدی بیده‌ندی –

39912330117022

استاد کارگر، درس کامپایلر (چهارشنبه)

## قابلیت استفاده از پروژه:

این پروژه یک سایت برای دانش آموزان و دانشجویان است تا بتوانند در آن گروه های درسی ایجاد کرده و با ایجاد رویداد های درسی به خواندن درسی مشخص در کنار هم مشغول شوند.

## فن آوری های مورد استفاده:

Spring boot suite به عنوان فریمورک اصلی برای تولید این سایت استفاده شده است.

postgreSql به عنوان پایگاه داده مورد استفاده در سایت استفاده شده است.

برای فرانت از thymeleaf استفاده شده است.

## معماری و زیرساخت:

برای این پروژه از معماری MVC(model, view, control) استفاده شده است. پیاده سازی این معماری به این صورت است که کنترلر ها به عنوان کنترل کننده های url ها استفاده میشوند و همه چیز از جهت دهی url ها، چگونگی نشان دادن(چه اطلاعاتی به صفحه داده شود) صفحه های لیست، اطلاعات و... را کنترل میکنند.

لایه مدل در اصل لایه ای است که داده های ما در آن شکل میگیرند و در واقع به نوعی بخشی از این معماری هست که در آن با جداول پایگاه داده و تغییراتی که در آن صورت میگیرد سرو کار داریم.

لایه view(نمایش) لایه ای است که در آن نمایش داده ای که از پایگاه داده دریافت میکنیم را به عهده دارد.

برای مطالعه بیشتر در رابطه با این معماری می توانید به داکيومنتیشن اسپرینگ به نشانی روبه رو مراجعه کنید: [Getting Started | Serving Web Content with Spring MVC](#)

## نحوه اجرای کد:

برای اجرای فایل مورد نظر بعد از دانلود کردن فایل های پروژه نیاز داریم تا dependency های مورد نیاز پروژه در فایل pom را دانلود کنید. برای این کار با دستور `mvn dependency:resolve` برای همه dependency ها و دستور

برای `mvn dependency:get -Dartifact=<groupId>:<artifactId>:<version>` dependency مورد نظر میتوانید برنامه را برای اجرا آماده کنید.

بعد از این مرحله، میتوانید با اجرا کردن فایل `WebApplication` در پوشه `com.studygroup.web` برنامه را اجرا کرده و در مرورگر خود با آدرس `localhost:8080/groups` به صفحه اصلی سایت مراجعه کنید.

## تشریح اعضای اصلی سایت:

در ابتدا به تشریح فایل `pom.xml` میپردازیم:

این فایل فایلی هست که dependency های پروژه در آن وجود دارد، dependency ها به نوعی کتاب خانه های مورد نیاز ما می باشند که به وسیله این فایل مدیریت میشوند.

Dependency های استفاده شده در این فایل به شرح زیر میباشند:

- Spring boot starter data jpa
- Spring boot starter thymeleaf
- Spring boot starter web
- Thymeleaf layout dialect

- Spring boot devtools
- Postgre sql
- Lombok
- Spring boot starter test
- Spring boot starter security
- Thymeleaf extras springsecurity5

فایل application.properties :

این فایل فایلی است که در آن ما به اپلیکیشن خود پایگاه داده و تنظیمات اولیه بخش مربوط به امنیت را می شناسایم.

فایل مدل ها:

مدل های ما همان جداول پایگاه داده ما هستند، مدل ها به صورت کلاس های جاوا نوشته میشوند ولی نکته ای که آنها را از کلاس های معمولی جدا می کند انوتیشن هایی است که spring در اختیار ما قرار میدهد.

مدل های موجود در فایل مدل ها مدل های گروه، درس، نقش و کاربر هستند.

در هر مدل از انوتیشن های @Data, @NoArgsConstructor, @AllArgsConstructor, @Builder برای ایجاد getters و setters برای جداول داده خود استفاده میکنیم تا بتوانیم در ادامه با داده های موجود در پایگاه داده کار کنیم.

انوتیشن @Entity مدل ما را به اسپرینگ معرفی میکند تا بعد از اجرای برنامه جداول پایگاه داده را ایجاد کند.

در بعضی از مدل ها از انوتیشن @Table برای نام گذاری دلخواه جدول استفاده شده است.

این ها انوتیشن هایی بودند که برای خود مدل استفاده شده اند.  
درون مدل نیز از انوتیشن های دیگری استفاده شده که به آنها نیز می پردازیم.  
@Id : برای مشخص کردن کلید اصلی در پایگاه داده.

@GeneratedValue(strategy = GenerationType.IDENTITY) :

برای تولید شدن خودکار شناسه (Id) هر موجودیت درون جدول.

@ManyToMany, @OneToMany, @ManyToOne :

برای تعیین روابط بین جدول ها و تعیین کلید های خارجی هر جدول.

### فایل Repository:

این فایل شامل interface هایی میشود که شامل تابع های  
CRUD(create, read, update, delete) ما میشود که یعنی در اصل بعد از  
تولید شدن پایگاه داده، اعمال crud و تبدیل جداول به شیئی و فرستادن آن به لایه  
های بالاتر و نمایش آن را ممکن میسازد.

### فایل Dto:

این فایل ها همان کلاس های مدل ما هستند با این تفاوت که کار برد آنها در این  
است که تنها شامل ستون هایی از جدول اصلی ما هستند که ما تمایل به نشان  
دادن آنها به کاربر هستیم. در اصل شیئی که به لایه های بالاتر برای ارائه به  
کاربر فرستاده می شود از جنس Dto میباشد.

## فایل Service:

سرویس ها به عنوان لایه ای از انتزاع به کار میروند، در اصل این فایل سرویس های repository ما را برای مدل های مختلف ما بازنویسی میکند. به طور مثال برای استفاده از متد ساختن ریپازتوری ما ابتدا این متد را در فایل سرویس های خود برای مدل خود شخصی سازی کرده و سپس از آن استفاده میکنیم.

## فایل Controller:

کنترل کننده ها فایل هایی هستند که url های ما را کنترل میکنند و این که چه متد های http ای را می توانیم از url ها دریافت کنیم.