**Laravel Assignment done:**

**Dashboard:**

* Use front-end theme like dashlite (**used adminalt theme**)
* Use webpack for installing theme (**done**)
* Basic Laravel Auth: ability to log in as administrator (**using breeze when login as super admin “credential” you can access all permissions and when login as sub admin credential you can access only below mention permission and also user can write comment and give review to product** )
* Use database seeds to create first user with email admin@admin.com and password “password” (done)

1. Admin: email: [admin@admin.com](mailto:admin@admin.com) , pass: password
2. Subadmin: email: [subadmin@admin.com](mailto:subadmin@admin.com) , pass: Pass@123
3. User: register user with api and welcome email send to them

* CRUD functionality (Create / Read / Update / Delete) for two menu items: **Categories** and **Products**. (done)
* Option to add multiple categories with one product. (done)
* Option to add multiple image with product (no need to use any plugin , use simple multiple image upload functionality) (done)

**Frontend:**

* Create one page where show category list, (done)
* On category click show all products which belongs to this category (done)
* Create product detail pagewhere user can comment and give feedback (done)

**API:**

* Register User (done)
* When new user register send email to admin (done but email send to new register user)
* Category listing api (done)
* Product listing api (done)
* Api response must be formatted (use api resource and collection ) (done, it is new for me, I learn it)

**Important notes:**

* breeze for dashboard authentication (yes)
* sanctum for api authentication (we have no login api on that reason I do not use this)
* all api’s must be paginated (yes)
* email’s must be queued (yes)
* Store images in **storage/app/public** folder and make them accessible from public(yes)
* Use intervention package for image compression (yes, it is new for me, I learn it )
* All crud’s should be ajax based (yes)
* Form validation using laravel requests (yes)
* Use helper class for common functions(I have not seen common functionality )
* Use data tables for tables (yajra & jquery datatables) (dashboard)(yes, use yajra )

**No need to design anything, just focus on functionality**