**1.Application Setting user\_id for new BlogPost**

When a user adds a new BlogPost the user\_id always appears null

we have to set up a user\_id when we create a new

in PostsController.php we need to write

$validateData['user\_id']=$request->user()->id;

user() would be currently authenticated user if user is authenticated

in PostsController.php

public function store(StorePost $request)

{

$validateData = $request->validated();

$validateData['user\_id'] = $request->user()->id;

$blogPost=BlogPost::create($validateData);

$request->session()->flash('status','Blog Post was created');

return redirect()->route('posts.show', ['post' => $blogPost->id]);

}

**2.GlobalQuery Posts Builder**

We want the recent posts to be on top

**in PostsController.php**

public function index()

{

$posts=BlogPost::withCount('comment')->orderBy('created\_at','desc')->get();

}

This is not a perfect solution

QueryScopes

We can define it 2 ways

in LatestScope.php

we need to write

**public function apply(Builder $builder, Model $model)**

**{**

**$builder->orderBy($model::CREATED\_AT, 'desc');**

**}**

**We have to use it at every model where we want to use LatestScope**

**in BlogPost.php**

**public function boot(){**

**static::addGlobalScope(new LatestScope);**

**}**

**3.Global Query Scopes and Sepcific issues**

Its not best idea to use LatestScope

Thsi is a hge problem and we get

**SQLSTATE[42000]: Syntax error or access violation: 1140 Mixing of GROUP columns (MIN(),MAX(),COUNT(),...) with no GROUP columns is illegal if there is no GROUP BY clause (SQL: select `blog\_posts`.\*, (select count(\*) from `comments` where `blog\_posts`.`id` = `comments`.`blog\_post\_id` and `comments`.`deleted\_at` is null order by `created\_at` desc) as `comment\_count` from `blog\_posts` where `blog\_posts`.`deleted\_at` is null order by `created\_at` desc)**

**Though it can be fixed by running strict:false but not recommened**

**4.Local Query**