

Answer To the Question of assignment-19

1. Simple Blog Project using Laravel Eloquent ORM & Axios

- *Blog Post List By
- *Blog Post Details
- *Free comments system

Ans:

Step 1: Set up Laravel project

Assuming you have already installed Laravel, open a terminal and navigate to your desired project directory. Run the following command to create a new Laravel project:

```
composer create-project laravel/laravel blog-project
```

Step 2: Set up the database

Update the .env file with your database credentials. Create a new database for your project.

Name blog-project

Step 3: Generate the necessary models, migrations, and controllers

Run the following commands to generate the necessary models, migrations, and controllers:

```
php artisan make:model Post -mc
```

```
php artisan make:model Comment -mc
```

This will generate the Post and Comment models along with their corresponding migrations and controllers.

Step 4: Define the database schema

Open the generated migration files in the database/migrations directory and define the table structures. For example, in the create_posts_table.php migration file, you can define the posts table as follows:

```
public function up()
{
    Schema::create('posts', function (Blueprint $table) {
        $table->id();
        $table->string('title');
        $table->text('content');
        $table->timestamps();
    });
}
```

Similarly, define the schema for the comments table in the create_comments_table.php migration file.

After defining the schema, run the migration command to create the tables:

```
php artisan migrate
```

Step 5: Define the relationships between models

Open the Post and Comment models and define the relationships between them. In the Post model, add the following method:

```
public function comments()
{
    return $this->hasMany(Comment::class);
}
```

In the Comment model, add the following method:

```
public function post()
{
    return $this->belongsTo(Post::class);
}
```

Step 6: Define routes and controllers

Open the routes/web.php file and define the routes for the blog posts and comments. Here's an example:

```
use App\Http\Controllers\PostController;

use App\Http\Controllers\CommentController;


Route::get('/posts', [PostController::class, 'page']);
Route::get('/projectsData', [PostController::class, 'projectsData']);
Route::get('/posts/{id}', [PostController::class, 'show']);
Route::post('/posts/{id}/comments', [CommentController::class, 'store']);
```

Now, open the app/Http/Controllers/PostController.php file and define the methods index and show:

```
use App\Models\Post;

public function page(Request $request){
    return view('pages.projects');
}

function projectsData(Request $request){
    return $posts = Post::all();
}

public function show($id)
{
    $post = Post::findOrFail($id);
    return view('pages.postdetail', compact('post'));
}
```

Similarly, open the app/Http/Controllers/CommentController.php file and define the store method:

```
use App\Models\Post;
use App\Models\Comment;
use Illuminate\Http\Request;

public function store(Request $request, $id)
{
    $post = Post::findOrFail($id);

    $comment = new Comment();
    $comment->content = $request->input('content');
    $comment->post()->associate($post);
    $comment->save();

    return redirect()->back();
}
```

Step 7: Create views for blog posts and comments

Create the following view files in the resources/views/posts directory:

index.blade.php: This file will display the list of blog posts.

```
@foreach($posts as $post)
    <h2>{{ $post->title }}</h2>
    <p>{{ $post->content }}</p>
    <a href="/posts/{{ $post->id }}">View details</a>
@endforeach
```

show.blade.php: This file will display the details of a specific blog post.

```
<h2>{{ $post->title }}</h2>
```

```
<p>{{ $post->content }}</p>
```

```
<h3>Comments</h3>
```

```
@foreach($post->comments as $comment)
```

```
    <p>{{ $comment->content }}</p>
```

```
@endforeach
```

```
<form action="/posts/{{ $post->id }}/comments" method="post">
```

```
    @csrf
```

```
    <textarea name="content"></textarea>
```

```
    <button type="submit">Submit Comment</button>
```

```
</form>
```

Step 8: Compile assets and start the server

Run the following commands to compile your assets and start the development server:

```
npm install
```

```
npm run dev
```

```
php artisan serve
```

That's it! You now have a simple blog project using Laravel's Eloquent ORM and Axios. You can access your blog by visiting <http://localhost:8000/posts> in your browser.

Note: This example assumes you have Axios already installed and configured in your project.

In addition for better: Certainly! Here's an example of how you can apply HTML, CSS, and Bootstrap to the views in your simple blog project:

1. index.blade.php (list of blog posts):

```
@extends('layouts.app')
```

```
@section('content')
```

```
    <div class="container">
```

```

<h1>Blog Posts</h1>

<div class="row">
  @foreach($posts as $post)
    <div class="col-md-4">
      <div class="card mb-4">
        <div class="card-body">
          <h5 class="card-title">{{ $post->title }}</h5>
          <p class="card-text">{{ $post->content }}</p>
          <a href="/posts/{{ $post->id }}" class="btn btn-primary">View Details</a>
        </div>
      </div>
    </div>
  @endforeach
</div>
@endsection

```

2. show.blade.php (details of a specific blog post):

```
@extends('layouts.app')
```

```
@section('content')
```

```
<div class="container">
```

```
  <h1>{{ $post->title }}</h1>
```

```
  <p>{{ $post->content }}</p>
```

```
<h3>Comments</h3>
```

```
<ul class="list-group">
```

```
  @foreach($post->comments as $comment)
```

```
    <li class="list-group-item">{{ $comment->content }}</li>
```

```
  @endforeach
```

```
</ul>
```

```
<form action="/posts/{{ $post->id }}/comments" method="post">
```

```

@csrf

<div class="form-group">

    <textarea class="form-control" name="content" rows="3"></textarea>

</div>

<button type="submit" class="btn btn-primary">Submit Comment</button>

</form>

</div>

```

@endsection

3. app.blade.php (layout template):

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <title>Simple Blog</title>
</head>

<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <div class="container">
            <a class="navbar-brand" href="/posts">Simple Blog</a>
        </div>
    </nav>

    @yield('content')

    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>

</html>

```

Axios part

Include in

```
<head>
<script src="{{asset('js/axios.min.js')}}"></script>
</head>
```

****Go to view/components/project-list.blade.php and post below code

```
<section class="py-5">
  <div class="container px-5 mb-5">
    <div class="text-center mb-5">
      <h1 class="display-5 fw-bolder mb-0"><span class="text-gradient d-inline">Blog Post
List</span></h1>
    </div>
    <div class="row gx-5 justify-content-center">
      <div id="project-list" class="col-lg-11 col-xl-9 col-xxl-8">

        </div>
      </div>
    </div>
  </section>

<script>
  GetProjectList();
  async function GetProjectList(){
    let URL="/projectsData"

    try {
      // Loader Show Content Hide
      document.getElementById('loading-div').classList.remove('d-none');
      document.getElementById('content-div').classList.add('d-none');
```



```

const response = await axios.get(URL);

document.getElementById('loading-div').classList.add('d-none');
document.getElementById('content-div').classList.remove('d-none');

response.data.forEach((item)=>{

    document.getElementById('project-list').innerHTML+=(`<div class="card overflow-
hidden shadow rounded-4 border-0 mb-5">
    <div class="card-body p-0">
        <div class="d-flex align-items-center">
            <div class="p-5">
                <h2 class="fw-bolder">${item['title']}</h2>
                <a href="/posts/${item['id']}">View details</a>
            </div>
        </div>
    </div>`)

})

} catch (error) {
    alert(error)
}

}
</script>

```

***Go to view/components/detail.blade.php and post below code

```

<div class="container px-5 my-5">
    <div class="row gx-5 justify-content-center">
        <div class="col-lg-11 col-xl-9 col-xxl-8">
            <section>
                <h2 class="text-secondary fw-bolder mb-4">Selected Post Detail</h2>
                <!-- Education Card 1-->

                <div id="education-list">

```

```

        <h2>{{ $post->title }}</h2>
        <h2>{{ $post->excerpt }}</h2>
        <p>{{ $post->description }}</p>

        <h3>Comments</h3>
        @foreach($post->comments as $comment)
            <p>{{ $comment->content }}</p>
        @endforeach

        <form action="/posts/{{ $post->id }}/comments" method="post">
            @csrf
            <textarea name="content"></textarea></br>
            <button type="submit">Submit Comment</button>
        </form>
    </div>

</section>
</div>
</div>
</div>

***Go to view/app.blade.php and post below code

<!DOCTYPE html>
<html lang="en" data-bs-theme="light">
<head>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no"/>

    <link rel="icon" type="image/x-icon" href="{{asset('/favicon.ico')}}" />
    <link href="{{asset('https://cdn.jsdelivr.net/npm/bootstrap-icons@1.8.1/font/bootstrap-
icons.css')}}" rel="stylesheet" />
    <link href="{{asset('css/bootstrap.css')}}" rel="stylesheet" />
    <link href="{{asset('css/style.css')}}" rel="stylesheet" />
    <script src="{{asset('js/axios.min.js')}}"></script>
</head>

<body class="d-flex flex-column h-100">

    @include('components.navbar')

```

```
@include('components.loader')
<div class="" id="content-div">
    @yield('content')
</div>
```

```
@include('components.footer')
```

```
<script src="{{asset('js/bootstrap.bundle.min.js')}}"></script>
</body>
</html>
```

In the above example, Bootstrap classes are used to style the elements, such as cards for the blog posts, form groups, and buttons. The layout file (app.blade.php) contains a navbar with a dark background, and the content of each view is wrapped in a container for better spacing and alignment.

Remember to include the CSS and JavaScript files for Bootstrap from the respective CDN links in the app.blade.php file.

With this setup, your blog project will have a responsive and visually appealing design using Bootstrap. Feel free to customize the styles further according to your preferences.