Laravel Initial Steps (Windows)

1. Install Xampp Latest Version ⇒ https://www.apachefriends.org/download.html
2. Install Composer ⇒ https://getcomposer.org/download/
3. Run this command in CMD to install latest stable version of Laravel.

```
composer create-project laravel/laravel app_name
```

4. Run this command in CMD to install any specific version of Laravel.

   ⇒ For example, to install Laravel version 8.0.0, you would use the following command:

```
composer create-project --prefer-dist laravel/laravel:^8.0.0 app_name
```

5. Than set Database configuration in .env file.
6. Then after you can access Laravel project by http://laravel_project.test/public. (URL will change according to your folder and app name.)
7. If you want to install Authentication System out of the box then you can see further steps. There are many Authentication Library in Laravel but **Laravel Breeze** is better in my opinion. It uses Tailwind/Blade as a frontend and it will require Node.js install on your system.
8. Install Laravel Breeze using following command in CMD:

```
composer require laravel/breeze --dev
```

```
php artisan breeze:install
```

```
php artisan migrate
```

```
npm install
```

```
npm run dev
```

9. You can also use bootstrap authentication system using following command in CMD :

```
composer require laravel/ui
```

```
php artisan ui bootstrap --auth
```

```
php artisan migrate
```

```
npm install && npm run dev
```

10. http://projecturl.test - Now your basic website is live with simple Login/Register and Home page.
11. Now we will create one **Blog** module with Controller, Model, View and migrations.

Blog Module :

1. Create Migration - Using migration you can create table in database. It is basically a Schema of a table

   ⇒ you can create migration file using following command :

```
php artisan make:migration create_blogs_table
```

⇒ This will create a new migration file in the "database/migrations" directory. Open the file and update the "up" method to define the columns of the "blogs" table.

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

Then run the migration using the following command:

```
php artisan migrate
```

This will create the "blogs" table in your database.

2. Create a model for the blogs

   ⇒ To create a model for the "blogs" table, run the following command:

   ```
   php artisan make:model Blog
   ```

   ⇒ This will create a new model file in the "app/Models" directory. If you want to give custom table name then Open the file and update it to specify the table name:

   ```
   class Blog extends Model
   {
       protected $table = 'blog_posts';
   }
   ```

3. Create a controller for the blog module

   ⇒ To create a controller for the blog module, run the following command:

```
php artisan make:controller BlogController
```

This will create a new controller file in the "app/Http/Controllers" directory. Open the file and add the following methods:

```
class BlogController extends Controller
{
    public function index()
    {
        $posts = Post::all();
        return view('blog.index', ['posts' => $posts]);
    }

    public function show($id)
    {
        $post = Post::find($id);
        return view('blog.show', ['post' => $post]);
    }

    public function create()
    {
        return view('blog.create');
    }

    public function store(Request $request)
    {
        $post = new Post();
        $post->title = $request->input('title');
        $post->content = $request->input('content');
        $post->save();

        return redirect()->route('blog.index');
    }

    public function edit($id)
    {
        $post = Post::find($id);
        return view('blog.edit', ['post' => $post]);
    }

    public function update(Request $request, $id)
    {
        $post = Post::find($id);
        $post->title = $request->input('title');
        $post->content = $request->input('content');
        $post->save();

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

    public function destroy($id)
    {
        $post = Post::find($id);
```

```
        $post->delete();

        return redirect()->route('blog.index');
    }
}
```

4. Create view files in following location to display blogs in browser.

```
resources/views/blogs/index.blade.php
resources/views/blogs/create.blade.php
resources/views/blogs/edit.blade.php
```

```
// index.blade.php


@extends('layouts.app')

@section('content')
    <h1>Blog Posts</h1>

    @foreach ($blogs as $blog)
        <div>
            <h2>{{ $blog->title }}</h2>
            <p>{{ $blog->content }}</p>
            <a href="{{ route('blog.show', ['id' => $blog->id]) }}">Read more</a>
        </div>
    @endforeach
@endsection
```

```


// create.blade.php
@extends('layouts.app')

@section('content')
    <h1>Create Blog Post</h1>

    <form action="{{ route('blog.store') }}" method="POST">
        @csrf
        <div>
            <label for="title">Title:</label>
            <input type="text" name="title" id="title">
        </div>
        <div>
            <label for="content">Content:</label>
            <textarea name="content" id="content"></textarea>
        </div>
        <button type="submit">Create</button>
```

```
    </form>
@endsection
```

```
// edit.blade.php
@extends('layouts.app')

@section('content')
    <h1>Edit Blog Post</h1>

    <form action="{{ route('blog.update', ['id' => $blog->id]) }}" method="POST">
        @csrf
        @method('PUT')
        <div>
            <label for="title">Title:</label>
            <input type="text" name="title" id="title" value="{{ $blog->title }}">
        </div>
        <div>
            <label for="content">Content:</label>
            <textarea name="content" id="content">{{ $blog->content }}</textarea>
        </div>
        <button type="submit">Update</button>
    </form>
@endsection
```

5. Now we will set routes to access blog module through URL :

   ⇒ Open the routes/web.php file.
   ⇒ Add the following code at the bottom of the file:

```
Route::group(['prefix' => 'blogs'], function () {
    Route::get('/', [BlogController::class,'index'])->name('blog.index');
    Route::get('/create', [BlogController::class,'create'])->name('blog.create');
    Route::post('/store', [BlogController::class,'store'])->name('blog.store');
    Route::get('/{id}/show', [BlogController::class,'show'])->name('blog.show');
    Route::put('/{id}/update', [BlogController::class,'update'])->name('blog.updat
e');
    Route::delete('/{id}/destroy', [BlogController::class,'destroy'])->name('blog.
destroy');
    Route::get('/{id}/edit', [BlogController::class,'edit'])->name('blog.edit');
});
```