1.create project starter

2. setup database

3. add jetstream/livewire :

composer require laravel/jetstream

php artisan jetstream:install livewire –teams

4. migrate tables to database:

Php artisan migrate

5. Create Models and migration :

* Page:
  + Title : string , nullable
  + Slug : string, nullable
  + Content: longtext , nullable
  + Create livewire components :
    - Php artisan make:livewire Pages
    - Php artisan make:livewire FrontPage
    - Add Trix editor :
      * <https://github.com/amaelftah/laravel-trix>

6. Setup Routes:

-create group route to access the pages through authentification middleware

-create modal and bind variable to it

7. Pages Controller:

* Create modal for article creation/modification
* Create modal for delete
* Bind modal for each case to a variable
* Change the value of the variable when a button clicked for the specific action
* Bind the variables in controller to modals then reset them after validation, submission of action

8. FrontPage:

Like single page application the frontend is one page that calls specific components when needed and call specific data from database

9. Create Navigation system for front end using tailwindcss / alpine.js

10. Create Navigation menus controller:

* Create modal for navigation menu creation and update
* Attach logic and variable bind the variables
* Do the same for the delete

11. User Access Management

* We add a column to user table to specify role : *php artisan make:migration add\_role\_to\_users\_table --table=users*
* create middleware to manage access : *php artisan make:middleware AccessByRole*
* add condition for each user role to access specific pages

12. Command to generate component:

* Create a customized command to generate a component from blueprint
* Create two basic blade views similar to admin/pages for users and user-permissions
* Create model and migration for user-permissions: *php artisan make:model UserPermission --migration*
  + Add two string nullable columns “role” and “route\_name”
  + Migrate to database
* To create command for generating our components we need to customize the stub:
  + Generate the stubs : php artisan stub:publish
  + Create command using: php artisan make:command CommandName
  + Put logic of the template Class and ClassModelName in Commands/CommandName.php
  + Create and edit stub that will be used by this CommandName class to generate our class and view