Step1: Run the follwing command to run composer:

composer dump-autoload

Step2: Run the follwing command to install the Laravel Package

composer create-project laravel/laravel myappname

Step3: cleck the running of myappname

http://localhost/myappname

Step4: Create a virtal Host by using "Add a Virtual Host" at localhost WAMP/LAMP

Step5: It may possible afater creating Apache Virtual Host it still not opening the webpage, so that case please check that all the letter are in same order and don't forgot it is a case senstive in nature. like

C:/wamp64/www/awsapp/public & c:/wamp64/www/awsapp/public

are two different things becouse in one case c is in small letters while in another cases c is in capitial letters

IF it gives again and again same issue then open httpd-vhost.config file at below listed locations:

("c:/wamp64/bin/apache/apache2.4.33/conf/extra/httpd-vhosts.conf")

Then update case senstive information and restart the server.. & ejoy your virtual locallost

Step6: run

npm insatll

Step7: To insatting of Authentication for laravel run following commands

php artian make:auth

step8: If you do not have the applications key. let's generate it with following commands

php artisan key:generate

(Restart the server if require)

Step9: Step to insatll Bootstrap 4

npm insatll bootstrap

Step10: go to the resources/assets/sass/ and open app.scss file and change

the @import path from "~bootstrap-sass/assets/stylesheets/bootstrap" to

"node\_modules/bootstrap/scss/bootstrap"

Step11: Go to the resources/assets/sass/ and open \_variables.scss file and change

$font-size-base:14px; to $font-size-base:1rem;

Step12: go to the resources/assets/js/ and open file bootstrap.js and change

require('bootstrap-sass'); to require('bootstrap');

Step13: Now recompile the project by using following command:

npm run dev

Step14: Change the App\_NANE= yuorApp\_name at .env file this will update your application title

npm run dev

Step 15: create pagesController

php artisan make:controller PagesController

New PagesController is created inside the app/Http/Controllers/

Inside this class you are required to write the following code:

 public function index(){

        return view('pages.index');

   }

This code will say it will look for the index.blade.php inside the pages folder under recources/view/pages/index.blade.php

Step 16: Update the web.php file inside the route folder: here you an call the page which you have created inside the view/pages folder with name like index.blade.php and then you have called then inside the PagesController.php under app/Http/Controllers

Route::get('/', 'PagesController@index');

Step 17: Call the AppName to tile tag of webApp

<title>{{config(‘app.name’,’My Default App Title’)}}</title>

Step 18: Call Bootstrap css file from the app directory

<link rel=”stylesheet” href=”{{asset(‘css/app.css’)}}” >

Step 19: Add Bootstrap JavaScript file

<script src="{{ asset('js/app.js') }}"></script>

Step 20: Create a Layout folder inside the /resources/view/ & create a file name app.blade.php

Copy the basic structure of HTML code inside this page like:

<!doctype html>

<html lang="{{ app()->getLocale() }}">

    <head>

        <meta charset="utf-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="{{asset('css/app.css')}}">

    <title>{{config('app.name','aws-App')}}</title>

    <head>

    <body>

        <div class="container"> <!-- <div class="container-fluid"> -->

                @include('inc.navbar')<br>

                <div class="row" style="//background:white;">

                        <div class="col-sm-3">@include('inc.sidenav')</div>

                        <div class="col-sm-9">@yield('content') </div>

                </div>

                @include('inc.fotter')

        </div>

    </body>

    <script src="{{ asset('js/app.js') }}"></script>

</html>

Here you can include page like navbar.blade.php, side.blade.php, fotter.blade.php, etc using @include(‘inc.navber’) and so on.

This inc.navbar mean inside the folder resource/view/inc/ there is a file named navbar.blade.php which is called during the runtime by app.blade.php file in a position where it placed according to the website style i.e. under <Div></div> Tags.

Step 21: Step to write inside the index.balde.php

@extends('layouts.app')

 @section('content')

 <div class="jumbotron">

      <h1> aws-app template with bootstrap 4 </h1>

      <p> WellCome to aws app page with  HTML, CSS...</p>

    </div>

 @endsection

Step 22: Content of Contact.balde.php, about.blade.php ans on

@extends('layouts.app') <!-- calling Inside the layout folder the app.blade.php file -->

@section('content')

    <h1>About </h1>

    <p> WellCome to aws app About   </p>

@endsection