1. composer create-project laravel/laravel example-app
2. go inside the file composer update --ignore-platform-reqs
3. php artisan serve
4. php artisan key:generate
5. php artisan config:cache
6. php artisan make:controller name

note

views file are always in blade.php

7.Route file trigger the function of view when a previous fixed link encountered

8.make necessary function inside the controller file(bookcontroller).[bookcontroller::class,’welcome’]

9.Making a folder inside view where all the necessary view file will be stored.(books.index)

Making index page

1.first make a model book by typing command

artisan make:model “name”

then in controller file

2. $book=model\_name::paginate(15);

3.return view(‘books.index’)->

With (’ container\_name’,$ the variable created)

4.now create the index.blade.php file in the view.books and configure it (making table)

<ul>

@foreach($books as $meow)

<td> meow->id</td>

@endforeach

Write {{$meows->links()}}

Making the view button

1.make another function in controller name show

Public function show($book\_id)

2.make a route using get method and url contain /books/{bookid}/show

Here pass the value bookid to show declared method inside the class

3.add new column in index page table call action(for view) and inside the foreach add <td> <a href=”the address declared previous”></td>

But don’t do this;

Name each route like this way

Route()->name(‘ proper\_name’);

Then replace the href ={{ route(‘routename’,$parameter)}}

Parameter is in meow->id form

Then go to the controller and in the current function

Create $xyz=book::find(book\_id) // book created in previous index page

Then return a view page

Return view(‘books.show) ->with(‘book’,$xyz);

Then create the show.blade.php in the view…books

Create the table and a back button

//optional

@extends(‘filename’) @section(‘sectionname’)

implement

@endsection

Making button

Same way create a route,name it and make the class

Return view(books.create)

Make the view page

=>create a form (action=”{{ books.store}}”; +inside form @csrf

=> take help from bootstrap

Each <div> add nested <div>{{$errors->first(‘title’);}</div>

…. Lastly make a submit button which triggers a link and the link is routed to a class by post method

Implement the store function

Public function store(Request $request)

{

//make rules

$rules={ ‘title’==>’condition1|con2|condition3’

}

}

$this->validate($request,$rules);

$tempbook=new Book;

$tempbook->title=$request->title;….

$tempbook->save();

Return redirect()->route(name and parameters)

Edit books:

Make function edit($bookid)

{

$book=Book::find(bookid);

Return view page ->with(‘book’.$book) ;

}

Copy the insert html page

Need id as extra

So add<input type=’hidden’ name=’id’ value={{book->id}};

Then $book=$book.find($request->id);

$book->title=$request->title;

$book->author=$request->author…

$book.save();

Return redirect()->route(‘….’,$book->id)

Make a database

Php artisan make:migration database

Php artisan migrate

Php artisan make:factory bookfacroty

Write code

Goto database seeder

Book::factory