***Form CRUD***

**আমরা কি করব** :

আমরা করবো হচ্ছে Form CRUD।

সর্বপ্রথম আমরা একটি Form নিবো create.blade.php ফাইলে। সেই form এর মধ্যে ফিল্ড থাকবে

<**form** action=**"**{{ route(**'forms.store'**) }}**"** enctype=**"multipart/form-data"** method=**"post"**>

১. name

২. file

৩. date

৪. radiobox

৫. checkbox

৬. select(multiple)

৭. textarea

৮. button

**কীভাবে করব :** আমরা যেহেতু Form CRUD করব সেজন্য আমাদেরকে forms টেবিল তৈরি করতে হবে । এখন আমরা migration file create করব ।

প্রথমে ধারাবাহিকভাবে migration file তৈরি করব। Migration create করার command হচ্ছে :

**php artisan make:migration create\_forms\_table**

**public function up()**

**{**

**Schema::create('categories', function (Blueprint $table) {**

**$table->increments('id');**

**$table->string('name')->nullable();**

**$table->string('picture')->nullable();**

**$table->date('date\_of\_birth')->nullable();**

**$table->string('gender')->nullable();**

**$table->string('hobby')->nullable();**

**$table->text('skills')->nullable();**

**$table->longText('bio')->nullable();**

**$table->timestamps();**

**});**

**}**

এখন আমরা টেবিলকে migrate করব

migrate করার command হচ্ছে :

**php artisan migrate**

আমাদের এখন model create করতে হবে। Model create করার command হচ্ছে :

**php artisan make:model Form**

আমরা Category model এর class মধ্যে এই Code লিখব ।

**protected $fillable**=[**'name', 'picture', 'date\_of\_birth', 'gender', 'hobby', 'skills', 'bio'**];

বা এইটিও ব্যবহার করতে পারি:

**protected $guarded** = [];

আমাদেরকে **controller create** করতে হবে। কমান্ড হল:

**php artisan make:controller FormController**

এখন আমরা **route** তৈরি করব। আমরা *resource route use* করব।

Route::*resource*(**'forms'**, **'FormController'**);

এইখানে আমরা validation এর জন্য Request controller ব্যবহার করবো।

তার জন্য কমান্ড হলো : **php artisan make:request FormRequest**

**FormRequest controller** class এর মধ্যে মেথড টি হলো

**public function** authorize()

{

**return true**;

}

**public function** rules()

{

**return** [

**'name'** => **'required|max:20|min:5'**,

**'picture'** => **'required'**,

**'date\_of\_birth'** => **'required'**,

**'gender'** => **'required'**,

**'hobby'** => **'required'**,

**'skills'** => **'required'**,

**'bio'** => **'required|max:200|min:15'**,

];

}

**app/Http/Controller** এর **FormController** মধ্যে  **View File** এ ডাটা দেখানোর জন্য এই মেথডটি বানাতে হবে।

validation এর জন্য controller এর মধ্যে এইটি use করতে হবে।

**use** Illuminate\Http\Request;

**public function** index(Request $request)

{

*// dd($request->all());*

**return** view(**'forms.index'**, compact(**'forms'**));

}

**Form create** করার মেথড হল :

**public function** create()

{

**return** view(**'forms.create'**);

}

আমরা error handling করবো। তার জন্য আমাদেরকে try catch ব্যবহার করা লাগবে।

তার জন্য সর্বপ্রথম create.blade.php এর মধ্যে body এর পর এবং form tag শুরু করার আগে error handling করতে হবে। তার জন্য code হলো :

**@if (**$errors->any()**)**

<**div** class=**"alert alert-danger"**>

<**ul**>

**@foreach (**$errors->all() **as** $error**)**

<**li**>{{ $error }}</**li**>

**@endforeach**

</**ul**>

</**div**>

**@endif**

এর পর index.blade.php এর মধ্যে store করবো। তাই সেই file এর মধ্যে একটি table বানাতে হবে।

<**form** action=**"**{{ route(**'forms.index'**) }}**"**>

<**select** name=**"gender"**>

<**option** value=**""**>Select Gender</**option**>

<**option** value=**"Male"**>Male</**option**>

<**option** value=**"Female"**>Female</**option**>

</**select**>

<**button** type=**"submit"**>Filter</**button**>

</**form**>

(Filter এর জন্য এইটি করা লাগবে)

<**form** action=**"**{{ route(**'forms.index'**) }}**"**>

<**select** name=**"sort\_by\_name"**>

<**option** value=**""**>Select One</**option**>

<**option** value=**"asc"**>Ascending</**option**>

<**option** value=**"desc"**>Descending</**option**>

</**select**>

<**button** type=**"submit"**>Filter</**button**>

</**form**>

(Ascending and Descending এর জন্য এইটি index.blade.php file এর মধ্যে table এর আগে দিতে হবে।)

এবং তার জন্য মেথড হবে এইটি যা create function মেথড এর মধ্যে দিতে হবে।

**if**(**isset**($request->**gender**)){

$forms = Form::*where*(**'gender'**, $request->**gender**)->paginate(2);

}**elseif** (**isset**($request->**sort\_by\_name**)){

$forms = Form::*orderBy*(**'name'**, $request->**sort\_by\_name**)->paginate(2);

}**else**{

$forms = Form::*paginate*(2);

}

**@php** $sl=0 **@endphp**

এইটি হলো সিরিয়াল নাম্বার এর জন্য। যাতে প্রতিটি সিরিয়াল অনুযায়ী থাকে।

**@foreach(**$forms **as** $form**)**

<**tr**>

<**td**>{{ ++$sl }}</**td**>

<**td**>{{ $form->**name** }}</**td**>

<**td**>

**@if(**file\_exists(storage\_path(**'app/public/users/'**).$form->**picture**) && (!is\_null($form->**picture**))**)**

<**img** src=**"**{{asset(**'storage/users/'**.$form->**picture**)}}**"** width=**"100px"**

height=**"100px"**>

**@else**

<**img** src=**"**{{ asset(**'default.png'**) }}**"** alt=**"Default Picture"** height=**"100"**>

**@endif**

</**td**>

(picture upload করার জন্য উপরের মেথড টি follow করা লাগবে। তার জন্য storage path এর সাথে public path এর লিংক করা লাগবে। তার জন্য কমান্ড হলো :: **php artisan storage:link**)

<**td**>

<**a** href=**"**{{ route(**'forms.show'**, $form->**id**) }}**"**>Show</**a**> |

<**a** href=**"**{{ route(**'forms.edit'**, $form->**id**) }}**"**>Edit</**a**> |

<**form** action=**"**{{ route(**'forms.destroy'**, $form->**id**) }}**"** method=**"post"** style=**"**display: **inline**;**"**>

**@csrf**

**@method(**'delete'**)**

<**button** type=**"submit"** onclick=**"return** confirm(**'Are You Sure ?'**)**"**>Delete</**button**>

</**form**>

</**td**>

</**tr**>

**@endforeach**

**Form store** করার মেথড হল :

**public function** store(FormRequest $request)

{

**try**{

$data = $request->except(**'picture'**, **'skills'**, **'hobby'**);

**if**($request->hasFile(**'picture'**)){

$data[**'picture'**] = $this->uploadImage($request->**picture**, $request->**name**);

}

$data[**'skills'**] = serialize($request->**skills**);

$data[**'hobby'**] = serialize($request->**hobby**);

Profile::*create*($data);

**return** redirect()->route(**'forms.index'**)->withStatus(**'Created Successfully!'**);

}**catch** (QueryException $exception){

**return** redirect()->back()->withInput()->withErrors($exception->getMessage());

}

}

**private function** uploadImage($file, $title)

{

$extension = $file->getClientOriginalExtension();

$fileName = date(**'y-m-d'**) . **'\_'** . time() . **'\_'** . $title . **'.'** . $extension;

*// $file->move(storage\_path('app/public/products/'), $fileName);*

Image::*make*($file)->resize(300, 300)->save(storage\_path(**'app/public/users/'**) . $fileName);

**return** $fileName;

}

**Single Form Show** করার মেথড হল :

Form এর প্রতিটি user এর single show এর জন্য একটি file লাগবে show.blade.php নামের।

সেই ফাইলে একটি table করা লাগবে।

<**table**>

<**tr**>

<**td**>Name</**td**>

<**td**>{{ $form>**name** }}</**td**>

</**tr**>

<**tr**>

<**td**>Picture</**td**>

<**td**></**td**>

</**tr**>

<**tr**>

<**td**>Gender</**td**>

<**td**>{{ $form>**gender** }}</**td**>

</**tr**>

<**tr**>

<**td**>Date Of Birth</**td**>

<**td**>{{ $form>**date\_of\_birth**->toFormattedDateString() }}</**td**>

</**tr**>

<**tr**>

<**td**>Hobby</**td**>

<**td**>

<**ul**>

**@foreach(**unserialize($form>**hobby**) **as** $hobby**)**

<**li**>{{ $hobby }}</**li**>

**@endforeach**

</**ul**>

</**td**>

</**tr**>

<**tr**>

<**td**>Skills</**td**>

<**td**>

**@foreach(**unserialize($form>**skills**) **as** $skill**)**

<**li**>{{ $skill }}</**li**>

**@endforeach**

</**td**>

</**tr**>

<**tr**>

<**td**>Bio data</**td**>

<**td**>{!! $form>**bio** !!}</**td**>

</**tr**>

</**table**>

এবং মেথড টি হলো:

**public function** show(Form $form)*//route model binding*

{

*// $user = Profile::findOrFail($id);*

*// dd($profile);*

**return** view(**'forms.show'**, compact(**'form'**));

}

**Form edit** করার মেথড হল :

edit করার জন্য একটি edit.blade.php ফাইল বানাতে হবে।

error handling এর জন্য ::

**@if (**$errors->any()**)**

<**div** class=**"alert alert-danger"**>

<**ul**>

**@foreach (**$errors->all() **as** $error**)**

<**li**>{{ $error }}</**li**>

**@endforeach**

</**ul**>

</**div**>

**@endif**

ফাইল এর মধ্যে হবে একটি form যা create.blade.php তে ছিলো।

<**form** action=**"**{{ route(**'forms.update'**, $form>**id**) }}**"** enctype=**"multipart/form-data"** method=**"post"**>

**name** এর value=**"**{{ $form>**name** }}**"**

**date of birth** এর value=**"**{{\Carbon\Carbon::*parse*($form>**date\_of\_birth**)->format(**'Y-m-d'**) }}**"**

**gender** এর value=**"Male"** {{ $form>**gender**==**'Male'** ? **'checked'** : **''** }}

value=**"Female"** {{ $form>**gender**==**'Female'** ? **'checked'** : **''** }}

**hobby** এর name=**"hobby[]"** value=**"Listening"** {{ in\_array(**'Listening'**, unserialize($form>**hobby**)) ? **'checked'** : **''** }}

name=**"hobby[]"** value=**"Reading"** {{ in\_array(**'Reading'**, unserialize($form>**hobby**)) ? **'checked'** : **''** }}

name=**"hobby[]"** value=**"Coding"** {{ in\_array(**'Coding'**, unserialize($form>**hobby**)) ? **'checked'** : **''** }}

<**select** name=**"skills[]"** multiple>

<**option** value=**""**>Select One</**option**>

<**option** value=**"PHP"** {{ in\_array(**'PHP'**, unserialize($form>**skills**)) ? **'selected'** : **''** }}>PHP</**option**>

<**option** value=**"JavaScript"** {{ in\_array(**'JavaScript'**, unserialize($form>**skills**)) ? **'selected'** : **''** }}>JavaScript</**option**>

<**option** value=**"MySql"** {{ in\_array(**'MySql'**, unserialize($form>**skills**)) ? **'selected'** : **''** }}>MySql</**option**>

</**select**>

<**textarea** name=**"bio"** id=**"bioData"**>

{{ $form>**bio** }}

</**textarea**>

**public function** edit(Profile $form)

{

**return** view(**'forms.edit'**, compact(**'form'**));

}

**Form update** করার মেথড হল :

**public function** update(Request $request, Form $form)

{

**try**{

$data = $request->except(**'picture'**, **'skills'**, **'hobby'**);

**if**($request->hasFile(**'picture'**)){

$data[**'picture'**] = $this->uploadImage($request->**picture**, $request->**name**);

}**else**{

$data[**'picture'**] = $form>**picture**;

}

$data[**'skills'**] = serialize($request->**skills**);

$data[**'hobby'**] = serialize($request->**hobby**);

$form>update($data);

**return** redirect()->route(**'forms.index'**)->withStatus(**'Updated Successfully !'**);

}**catch** (QueryException $exception){

**return** redirect()->back()->withInput()->withErrors($exception->getMessage());

}

}

**Form destroy** করার মেথড হল :

**public function** destroy(Form $form)

{

**try**{

$form->delete();

**return** redirect()->route(**'forms.index'**)->withStatus(**'Deleted Successfully !'**);

}**catch** (QueryException $exception){

**return** redirect()->back()->withInput()->withErrors($exception->getMessage());

}

}

***FORM ATTRIBUTE***

## **<input name="name" type="text">** <input type="button"> <input type="checkbox"> <input type="color"> <input type="date"> <input type="datetime-local"> <input type="email"> <input type="file"> <input type="hidden"> <input type="image"> <input type="month"> <input type="number"> <input type="password"> <input type="radio"> <input type="range"> <input type="reset"> <input type="search"> <input type="submit"> <input type="tel"> <input type="text"> <input type="time"> <input type="url"> <input type="week">

## **<select name="cars">**

## <option value="fiat" selected>Fiat</option>

## <select name="cars" size="4" **multiple**>

1. <textarea name="message" rows="10" cols="30">
2. <button type="button" onclick="alert('Hello World!')">Click Me!</button>