

Auf der Webseite sollen Kopien von privaten Dokumenten gespeichert werden. Diese Dokumente sollen von Mitgliedern einer Gruppe eingesehen werden können. Mit

```
php artisan make:migration add_group_id_to_users_table --table=users
```

```
<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class AddGroupIdToUsersTable extends Migration {
    public function up() {
        Schema::table('users', function (Blueprint $table) {
            $table->smallInteger('group_id')->default(1);
        });
    }
    public function down() {
        Schema::table('users', function (Blueprint $table) {
            $table->dropColumn('group_id');
        });
    }
}
```

```
php artisan migrate
```

wird in der Tabelle **users** das Feld **group_id** eingefügt, um Benutzer einer Gruppe zuzuweisen.

Ein Gruppenmitglied ist für die Verwaltung der Dokumente der Gruppe verantwortlich. Mit

```
php artisan make:migration add_group_admin_to_users_table --table=users
```

```
<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class AddGroupAdminToUsersTable extends Migration {
    public function up() {
        Schema::table('users', function (Blueprint $table) {
            $table->boolean('group_admin')->default(false);
        });
    }
    public function down() {
        Schema::table('users', function (Blueprint $table) {
            $table->dropColumn('group_admin');
        });
    }
}
```

```
php artisan migrate
```

wird in der Tabelle **users** das Feld **group_admin** eingefügt, um einen Benutzer der Gruppe als Gruppenverantwortlichen zu bestimmen. Dieser ist für die Wartung der Gruppe verantwortlich.

Gruppen haben eine eindeutige Gruppennummer und eine entsprechende Bezeichnung.

```
php artisan make:model Group -cmr
```

liefert das Modell **Group**, der **GroupController** und eine Migrationsdatei zum Erstellen der benötigten Tabelle erstellt.

Mit

```
\database\Migrations\x_create_groups_table.php
<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreateGroupsTable extends Migration {
    public function up() {
        Schema::create('groups', function (Blueprint $table) {
            $table->increments('id');
            $table->tinyinteger('group_id')->default(1);
            $table->string('group_name');
            $table->timestamps();
        });
    }
    public function down() {
        Schema::dropIfExists('groups');
    }
}
```

php artisan migrate

wird die Tabelle **groups** erstellt.

Ein registrierter Benutzer kann Administrator, Mitglied einer Gruppe und Gruppenverantwortlicher sein.

```
\app\User.php
<?php namespace MyWeb;
use MyWeb\Blog;
use MyWeb\Group;
use MyWeb\Person;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\SoftDeletes;
use Illuminate\Notifications\Notifiable;
use Illuminate\Foundation\Auth\User as Authenticatable;
class User extends Authenticatable {
    use Notifiable;
    use SoftDeletes;
    protected $dates = ['deleted_at'];
    protected $fillable =
        [ 'admin', 'group_id', 'group_admin', 'email', 'name', 'password', ];
    protected $hidden = [ 'password', 'remember_token', ];
    public function blog() { return $this->hasMany(Blog::class); }
    public function group() { return $this->hasOne(Group::class); }
    public function publish(Blog $blog) { $this->blog()->save($blog); }
    public function person() { return $this->hasOne(Person::class); }
}
```

Eine Gruppe kann mehrere Mitglieder enthalten.

```
\app\Group.php
<?php namespace MyWeb;
use MyWeb\User;
use Illuminate\Database\Eloquent\Model;
class Group extends Model {
    protected $fillable = ['group_id', 'group_name'];
    public function user() { return $this->hasMany(User::class); }
}
```

Der UserController muss angepasst werden, so dass die Werte der neuen Felder erfasst respektive bearbeitet werden können.

```
\app\Http\Controllers\UserController.php
<?php namespace MyWeb\Http\Controllers;
use MyWeb\Group;
use MyWeb\Person;
use MyWeb\User;
use Illuminate\Http\Request;
class UserController extends Controller {

    public function index() {
        $users = User::orderBy('name')->get();
        return view('users.u_index', compact('users'));
    }
}
```

¹ **hasMany:** Eine Gruppe kann mehrere Gruppenmitglieder haben.

```

public function create() {
    $groups = Group::orderBy('group_name')->get();
    $people = Person::orderBy('name')->get();
    $users = User::get();
    return view('users.u_create', compact('groups', 'people', 'users'));
}
public function store(Request $request) {
    \Mail::to($request->get('email'), '');
    \Mail::send('users.u_mail', [
        'name' => $request->get('name'),
        'email' => $request->get('email')
    ],
    function($message) {
        $message->from('web@FJKarli.ch')
        $message->to('web@FJKarli.ch', 'Registrierung')
        ->subject('Registrierung auf www.MyWeb.ch');
    });
    $user = new User([
        'admin' => $request->get('admin') == 1,
        'email' => $request->get('email'),
        'group_admin' => $request->get('group_admin') == 1,
        'group_id' => $request->get('group_id'),
        'name' => $request->get('name'),
        'password' => bcrypt('NoPassword')
    ]);
    $user->save();
    return \Redirect::route('user.index');
}
public function show(User $user) { }
public function edit(User $user) {
    $groups = Group::orderBy('group_name')->get();
    $people = Person::orderBy('name')->get();
    $users = User::orderBy('name')->get();
    return view('users.u_edit', compact('groups', 'people', 'user', 'users'));
}
public function update(Request $request, User $user) {
    $password = $user->password;
    $user->update([
        'admin' => $request->get('admin') == 1,
        'email' => $request->get('email'),
        'group_admin' => $request->get('group_admin') == 1,
        'group_id' => $request->get('group_id'),
        'name' => $request->get('name'),
        'password' => $password
    ]);
    return \Redirect::route('user.index');
}
public function destroy(User $user) {
    $people = Person::get();
    $comment = PersonComment::get();
    foreach ($people as $person) {
        if ($person->user_id == $user->id) {
            $person->user_id = 0;
            Person::update($person);
        }
    };
    User::destroy($user->id);
    return \Redirect::route('user.index');
}
}
}

```

Der Administrator kann beim Registrieren

```

<!-- \resources\views\users\u_create.blade.php -->
@extends('layouts.layout')
@section('left')
<div class="container">
    <div class="row justify-content-center">
        <div class="col-md-11">
            <div class="card">
                <div class="card-header text-center"><b>Registrierung</b></div>
                <div class="card-body">
                    @include('layouts.session_message')
                    {!! Form::open(['class' => 'form', 'route' => 'user.store']) !!}
                    @include('partials.input_string',
                        ['label' => 'Name Vorname', 'value' => 'name'])
                    @include('partials.input_string',
                        ['label' => 'Mailadresse', 'value' => 'email'])
                </div>
            </div>
        </div>
    </div>
</div>

```



```

        'group_name' => request('group_name'),
        'group_id' => Group::count() + 1
    });
    return \Redirect::route('group.index');
}
public function show(Group $group) {}
public function edit(Group $group) {
    return view('group.g_edit', compact('group'));
}
public function update(Request $request, Group $group) {
    Group::find($group->id)->update(request(['group_name', 'group_id']));
    return \Redirect::route('group.index');
}
public function destroy(Group $group) {
    Group::destroy($group->id);
    return \Redirect::route('group.index');
}
}
}

```

sorgt dafür, dass der Administrator Gruppen erstellen, bearbeiten, auflisten oder löschen kann.

Mit

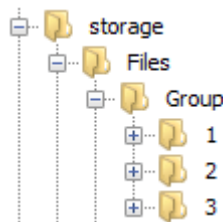
```

\routes\web.php
--
Route::resource('group', 'GroupController');
--

```

können die Funktionen des GroupController aufgerufen werden.

Wie Eingangs erwähnt, sollen Gruppenmitglieder Gruppendokumente hochladen und einsehen können. Gruppendokumente sollen in Gruppenverzeichnissen



gespeichert werden. Die Namen der Gruppendokumente werden mit deren Speicherorten in einer Tabelle gespeichert.

Das Beispiel aus dem Internet ([upload](#)) zeigt, wie Dateien in Laravel recht einfach hochgeladen werden können. Das Beispiel wird wie folgt übernommen. Mit

```
php artisan make:model GroupFile -cmr
```

wird ein Modell, ein Controller und eine Migrationsdatei erzeugt.

```

\database\Migrations\x_create_group_files_table.php
<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreateGroupFilesTable extends Migration {
    public function up() {
        Schema::create('group_files', function (Blueprint $table) {
            $table->increments('id');
            $table->integer('group_id')->default(1);
            $table->string('file_name');
            $table->string('file_path');
            $table->text('file_doc');
            $table->timestamps();
        });
    }
    public function down() {
        Schema::dropIfExists('group_files');
    }
}

```

```
php artisan migrate
```

group_id stellt die Zugehörigkeit zu der Gruppe her, **file_name** und **file_href** referenzieren die hochgeladene Datei.

Das Modell muss wie folgt angepasst werden.

```
\app\GroupFile.php
<?php namespace MyWeb;
use MyWeb\Group;
use Illuminate\Database\Eloquent\Model;
class GroupFile extends Model {
    protected $fillable = ['group_id', 'file_name', 'file_path'];
    public function group() {
        return $this->belongsTo(Group::class);
    }
}
```

belongsTo legt fest, dass eine Datei nur zu einer Gruppe gehört.

Der GroupFileController wird wie folgt ergänzt

```
app\Http\Controllers\GroupFileController.php
<?php namespace MyWeb\Http\Controllers;
use Auth;
use MyWeb\Group;
use MyWeb\GroupFile;
use MyWeb\User;
use Illuminate\Http\Request;
use League\Flysystem\Adapter\Local;
use League\Flysystem\Filesystem;
class GroupFileController extends Controller {
    public function __construct() {
        $this->middleware('auth');
    }
}
```

reserviert den Zugriff auf diesen Controller registrierten Benutzern

```
public function index() {
    $id = Auth::user()->group_id;
    $name = Group::findOrFail($id)->group_name;
    $files = GroupFile::where('group_id', $id)->get();
    return view('group.f_index', compact('id', 'name', 'files'));
}
```

index liefert die Liste der Dateien der Gruppe. Mit Klick auf einen Dateinamen, kann die entsprechende Datei dargestellt werden.

```
public function show(GroupFile $groupFile, $id) {
    $file = GroupFile::findOrFail($id);
    $file->delete();
    $adapter = new Local($file->file_path);
    $filesystem = new Filesystem($adapter, ['visibility' => 'public']);
    if ($filesystem->has($file->file_name)) {
        $filesystem->delete($file->file_name);
    };
    return \Redirect::route('groupfile.index');
}
```

show erlaubt dem Gruppenadministrator die gewählte Datei und deren Eintrag in der Tabelle **group_files** zu löschen

```
public function create() {
    $id = Auth::user()->group_id;
    $files = GroupFile::where('group_id', $id)->get();
    $name = Group::findOrFail($id)->group_name;
    return view('group.f_create', compact('id', 'name', 'files'));
}
```

create übergibt Gruppennummer, Gruppenname und die Liste der erfasssten Gruppenelemente an die View **group.f_create**.

Die erfassten Daten werden von der Funktion **store** bearbeitet

```
public function store(Request $request) {
    $id = Auth::user()->group_id;
    $file = $request->file('uploadedfile');
    switch ($file->getClientOriginalExtension()) {
        case 'pdf': case 'txt': case 'PDF': case 'TXT':
            $name = $file->getClientOriginalName();
            //Move Uploaded File
    }
}
```

```

$destinationPath = 'storage/Files/Group/' . $id . '/';
$file->move($destinationPath, $name);
if (GroupFile::where('file_name', $name)->exists()) {
    //update in group_files tabel
    $gf = GroupFile::where('file_name', $name);
    $gf->update([
        'group_id' => $id, 'file_name' => $name,
        'file_path' => $destinationPath, 'file_doc' => request('file_doc')
    ]);
} else {
    //save in group_files table
    $upload = new GroupFile;
    $upload->group_id = $id; $upload->file_name = $name;
    $upload->file_path = $destinationPath;
    $upload->file_doc = request('file_doc');
    $upload->save();
}
}
return back();
}

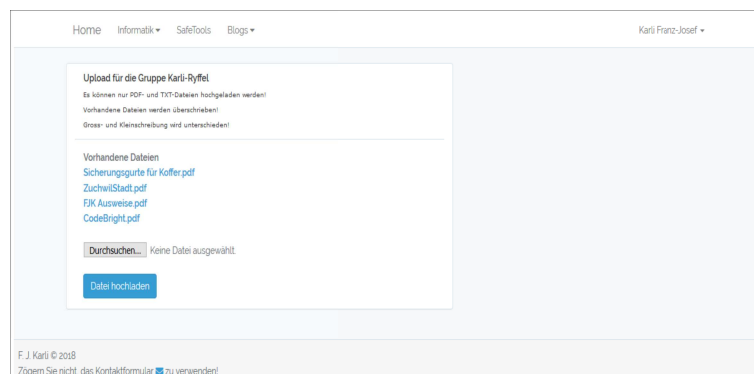
```

Falls eine Datei hochgeladen wird, werden die entsprechenden Zugriffsdaten in der Tabelle **group_files** gespeichert. Sollte eine Datei gleichen Namens hochgeladen werden wird die Tabelle aktualisiert und die existierende Datei überschrieben. Möglicherweise wird beim Speichern ein neues Gruppenverzeichnis erstellt.

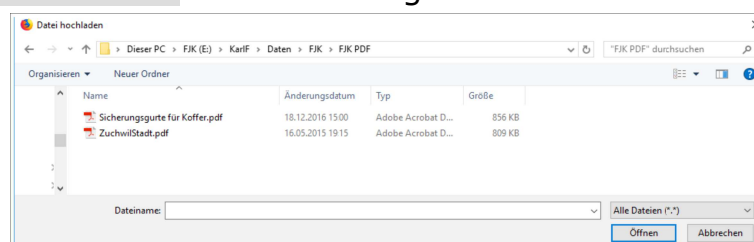
```

public function edit($id) {
    $groupFile = GroupFile::find($id);
    return view('group.f_edit', compact('groupFile'));
}
public function update(Request $request, $id) {
    GroupFile::find($id)->update(request(['file_doc']));
    return \Redirect::route('groupfile.index');
}
public function destroy($id) {
    $file = GroupFile::findOrFail($id);
    $file->delete();
    $adapter = new Local($file->file_path);
    $filesystem = new Filesystem($adapter, ['visibility' => 'public']);
    if ($filesystem->has($file->file_name)) {
        $filesystem->delete($file->file_name);
    };
    return \Redirect::route('groupfile.index');
}
}
}

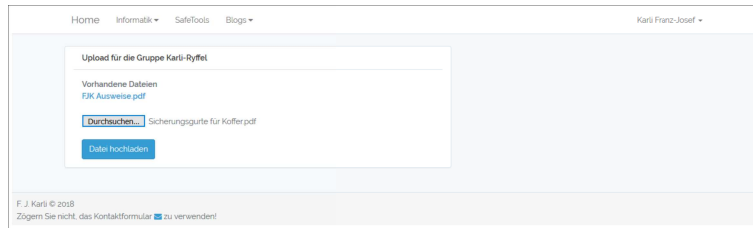
```



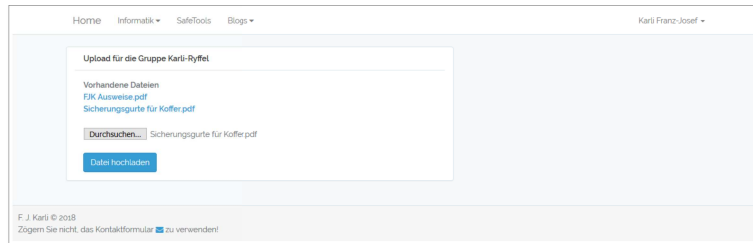
Ein Klick auf **Durchsuchen...** liefert das folgende Auswahlmennü.



Nach dem Markieren der gewünschten Datei und Klick auf **Öffnen**



kann diese hochgeladen und in der Datenbank gespeichert werden.



```
\routes\web.php
--
Route::resource('groupfile', 'GroupFileController');
--
```

sorgen dafür, dass Dateien hochgeladen, gespeichert, gelöscht, usw. werden können.

Diese Seiten stehen ausschliesslich registrierten Benutzern zur Verfügung.

Mit

```
\resources\views\layouts\nav.blade.php
--
<li>
  <a class="dropdown-item"
    href="{{ route('groupfile.create') }}">
    <b>Dokumente</b>
  </a>
</li>
@if (Auth::user()->admin)
  <li>
    <a class="dropdown-item"
      href="{{ route('group.index') }}">
      <b>Gruppen</b>
    </a>
  </li>
  <li>
    <a class="dropdown-item"
      href="{{ route('group.create') }}">
      Gruppe erstellen
    </a>
  </li>
@endif
```

Karli - Ryffel Franz-Josef ▾

Dokumente

Logout

Administrator ▾

Dokumente

Gruppen

Gruppe erstellen

Personen

Person erfassen

Registrierte

Registrieren

Logout

wird die Menüzeile so angepasst, dass sie mit Klick auf **Dokumente** Gruppendokumente gewählt oder hochgeladen können.

u_mail

```
<!-- \resources\views\users\u_mail.blade.php -->
<b>{{ $name }} wurde registriert!</b>
<br>
Wählen Sie <b>Login</b> auf www.FJKarli.ch
<br>
Wählen Sie 'Passwort vergessen' und fordern Sie für {{ $email }} einen Reset-Link an.
<br>
Sie erhalten diesen Reset-Link mit einer eMail.
<br>
Melden Sie sich mit diesem Link an und geben Sie Ihr gewünschtes Passwort ein.
</p>
Viel Erfolg
Gruss FJK
```

i_check

```
\resources\views\partials\input_check.blade.php
<div class="row">
  <label class="col-md-4 text-md-right">{{ $label }}</label>
  <div class="col-md-2">{{ Form::checkbox($value) }}</div>
</div>
```

i_range

```
\resources\views\partials\input_range.blade.php
<div class="row">
  <label class="col-md-4 text-md-right">{{ $label }}</label>
  <div class="col-md-6">
    {{ Form::selectRange($value, 1, $set) }}
  </div>
</div>
```

p_side

```
<!-- \resources\views\users\p_side.blade.php -->
<b>Erfasste Personen</b>
<note10>
  <ul>
    @foreach($people as $person)
      <li>
        <a href="{{ route('person.edit', $person->id) }}">
          <b>{{ $person->name }}</b>
        </a>
        <note10>
          - {{ $person->email }}, {{ $person->street }}, {{ $person->town }}
          Tel: {{ $person->phone }}
        </note10>
      </li>
    @endforeach
  </ul>
</note10>
```

u_side

```
<!-- \resources\views\users\u_side.blade.php -->
<b>Registrierte Benutzer</b>
<note10>
  <ul>
    @foreach($users as $user)
      <li>
        @if ($user->admin)
          <span class="fa fa-plus"></span>
        @endif
        <a href="{{ route('user.edit', $user->id) }}"> <b>{{ $user->name }}</b>
        </a> - {{ $user->email }}
      </li>
    @endforeach
  </ul>
</note10>
```

g_side

```
<!-- \resources\views\groups\g_side.blade.php -->
<!-- \resources\views\users\g_side.blade.php -->
<b>Gruppen</b>
<note10>
  <ul>
    @foreach($groups as $group)
      <li>
        <b>{{ $group->group_id }}</b>
        <a href="{{ route('group.edit', $group->id) }}"> {{ $group->group_name }} </a>
      </li>
    @endforeach
  </ul>
</note10>
```

g_index

```
<!-- \resources\views\group\g_index.blade.php -->
@extends('layouts.layout')
@section('content')
```

```

<div class="container">
  <div class="row justify-content-center">
    <div class="col-md-12">
      <div class="card">
        <div class="card-header text-center"><b>Liste der erfassten Gruppen</b></div>
        <div class="card-body">
          @forelse($groups as $group)
            <div class="blog-post">
              Gruppe {{ $group->group_id }}:
              <a href="{{ route('group.edit', $group->id) }}">
                {{ $group->group_name }}
              </a>
            </div>
          @empty
            Keine Gruppen erfasst
          @endforelse
        </div>
        <div class="card-footer">
          @include('partials.hint',
            ['hints' => ['Daten können mit Klick auf den Namen bearbeitet werden']])
        </div>
      </div>
    </div>
  </div>
</div>
@endsection

```

g_create

<!-- \resources\views\group\g_create.blade.php -->

```

@extends('layouts.layout')
@section('left')
<div class="container">
  <div class="row justify-content-center">
    <div class="col-md-8">
      <div class="card">
        <div class="card-header text-center">
          <b>Erstellen einer Gruppe</b>
        </div>
        <div class="card-body">
          @include('layouts.session_message')
          {!! Form::open(['class' => 'form', 'route' => 'group.store']) !!}
          @include('partials.input_string',
            ['label' => 'Gruppenname', 'value' => 'group_name'])
          @include('partials.input_number',
            ['label' => 'Gruppennummer', 'value' => 'group_id'])
          @include('partials.submit', ['submitButtonText' => 'Gruppe speichern'])
          {!! Form::close() !!}
        </div>
        <div class="card-footer">
          {!! Form::open(['class'=>'form', 'method' => 'GET',
            'route' => ['group.index']]) !!}
          @include('partials.submit', ['submitButtonText' => 'Zurück'])
          {!! Form::close() !!}
        </div>
      </div>
    </div>
  </div>
</div>
@endsection
@section('side')
  @include('groups.g_side')
@endsection

```

g_edit

<!-- \resources\views\group\g_edit.blade.php -->

```

@extends('layouts.layout')
@section('left')
<div class="container">
  <div class="row justify-content-center">
    <div class="col-md-8">
      <div class="card">
        <div class="card-header text-center">
          <b>Bearbeiten der Gruppendaten</b>
        </div>
        <div class="card-body">
          @include('layouts.session_message')
          {!! Form::model($group, ['class'=>'form', 'method' => 'PUT',
            'route'=> ['group.update', $group->id] ] ) !!}
          @include('partials.input_string',
            ['label' => 'Gruppenname', 'value' => 'group_name'])
        </div>
      </div>
    </div>
  </div>
</div>

```



```

        //Move Uploaded File
        $destinationPath = 'uploads';
        $file->move($destinationPath,$file->getClientOriginalName());
    }
}

```

Step 4 – Add the following lines in **app/Http/routes.php**.

```

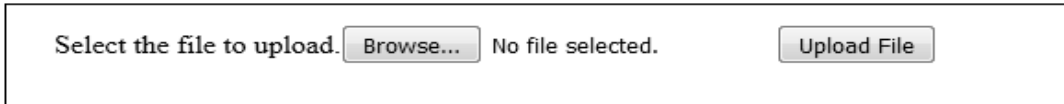
app/Http/routes.php
Route::get('/uploadfile','UploadFileController@index');
Route::post('/uploadfile','UploadFileController@showUploadFile');

```

Step 5 – Visit the following URL to test the upload file functionality.

http://localhost:8000/uploadfile

Step 6 – You will receive a prompt as shown in the following image.



f_index

\resources\views\group\f_index.blade.php

```

@extends('layouts.master')
@section('left')
    <div class="panel panel-default col-md-offset-1">
        <div class="card-header">
            <b>Verwaltung Gruppendateien der Gruppe {{ $name }}</b><br>
            <notel0>
                Mit Klick auf einen Filename kann die Beschreibung bearbeitet oder
                die Datei gelöscht werden<br>
            </notel0>
        </div>
        <div class="panel-body">
            @forelse($files as $file)
                <a href="{{ route('g_file.edit', $file) }}">
                    <b>{{ $file->file_name }}</b>
                </a>: {{ $file->file_doc }}
                <br>
            @empty
            @endforelse
        <br>
        </div>
    </div>
@endsection

```

f_create

\resources\views\group\f_create.blade.php

```

@extends('layouts.master')
@section('left')
    <div class="panel card-default col-md-offset-1">
        <div class="card-header">
            <b>Upload für die Gruppe {{ $name }}</b><br>
            <notel0>
                Es können nur PDF- und TXT-Dateien hochgeladen werden!
                Vorhandene Dateien werden überschrieben!
                Gross- und Kleinschreibung wird unterschieden!
            </notel0><br>
        </div>
        <div class="card-body">
            @forelse($files as $file)
                <a href="{{ config('app.ownfiles', '') }}Group/{{ $id }}/{{ $file-
                >file_name }}">
                    <b>{{ $file->file_name }}</b>
                </a>: {{ $file->file_doc }}
                <br>
            @empty
            @endforelse
        </div>
        <div class="card-footer">
            <br>
            {!! Form::open(['class' => 'form', 'files'=>'true', 'method' => 'POST',
            'route' => ['g_file.store', $id]]) !!}
            {{ Form::file('uploadedfile') }}<br>
            {{ Form::label('Beschreibung:') }}
            {{ Form::textarea('file_doc', null,
            ['class' => 'form-control', 'rows' => '3', 'required']) }}<br>
            {!! Form::submit('Datei hochladen', ['class' => 'btn btn-primary']) !!}
            {!! Form::close() !!}
        </div>
    </div>

```

```
<br>
  {!! Form::open(['class'=>'form', 'method' => 'GET',
    'route' => ['g_file.index']] !!)
    {!! Form::submit('Zurück', ['class' => 'btn btn-primary']) !!}
  {!! Form::close() !!}
</div>
</div>
@endsection
```