

Auf der Webseite sollen Kopien von privaten Dokumenten gespeichert werden. Diese Dokumente sollen von Mitgliedern einer Gruppe eingesehen werden können. Ein Gruppenmitglied ist für die Verwaltung der Dokumente der Gruppe verantwortlich.

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.

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.

Mit

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

wird das Modell **Group**, der **GroupController** und eine Migrationsdatei zum Erstellen der benötigten Tabelle **groups** erstellt und 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
```

die Tabelle **groups** erstellt. Gruppen haben eine eindeutige Gruppennummer und eine entsprechende Bezeichnung.

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

```
\app\User.php
<?php namespace MyWeb;
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 = [ 'name', 'email', 'password',
        'admin', 'group_id', 'group_admin' ];
    protected $hidden = [ 'password', 'remember_token', ];
    public function group() {
        return $this->hasOne(Group::class);
    }
    public function person() {
        return $this->hasOne(Person::class);
    }
}
```

1

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);
    }
}
```

2

¹ **hasOne:** Ein User kann nur Mitglied einer einzigen Gruppe sein.

² **hasMany:** Eine Gruppe kann mehrere Gruppenmitglieder haben.

Der UserController muss angepasst werden, so dass die Werte der neuen Felder erfasst 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'));
    }
    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')
                ->subject('Registrierung auf www.MyWeb.ch');
        });
        $user = new User([
            'name' => $request->get('name'),
            'email' => $request->get('email'),
            'password' => bcrypt('NoPassword'),
            'group_id' => $request->get('group_id'),
            'admin' => $request->get('admin') == 1,
            'group_admin' => $request->get('group_admin') == 1
        ]);
        $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) {
        $u = User::find($user->id);
        $password = $u->password;
        $u->update([
            'name' => $request->get('name'),
            'email' => $request->get('email'),
            'password' => $password,
            'group_id' => $request->get('group_id'),
            'admin' => $request->get('admin') == 1,
            'group_admin' => $request->get('group_admin') == 1
        ]);
        return \Redirect::route('user.index');
    }
    public function destroy(User $user) {
        User::destroy($user->id);
        return \Redirect::route('user.index');
    }
}

```

Der Administrator kann beim Registrieren

```

\resources\views\users\u_create.blade.php
@extends('layouts.master')
@section('left')
    <div class="container">
        <div class="card">
            <div class="card-header">
                <b>Benutzerregistrierung</b>
                <notel0><br>
                Die Mail-Adresse muss eindeutig sein!
                </notel0>
            </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'])
                @include('partials.input_check',
                    ['label' => 'Webmaster', 'value' => 'admin'])
                @include('partials.input_check',
                    ['label' => 'Gruppenadmin', 'value' => 'group_admin'])
                @include('partials.input_range',
                    ['label' => 'Gruppennummer', 'value' => 'group_id',
                    'set' => $groups->count()])
                @include('partials.submit',
                    ['submitButtonText' => 'Speichern'])
                {!! Form::close() !!}
            </div>
        </div>
    </div>
    <br>
</div>
@endsection
@section('side')
    <b>Erfasste Personen</b>
    <notel0>
        @foreach($people as $person)
            <br>
            <b>{{ $person->email }}</b>
            @if ($person->user_id > 0)
                <b> Registriert</b>
            @endif
            {{ $person->name }}, {{ $person->street }}, {{ $person->town }}
            Tel: {{ $person->phone }}
        @endforeach
    </notel0>
    <br>
    <b>Registrierte Benutzer</b>
    <notel0>
        @foreach($users as $user)
            <br>
            <b>{{ $user->email }}</b> {{ $user->name }}
        @endforeach
    </notel0>
    <br>
    <strong>Gruppen</strong>
    <notel0>
        @foreach($groups as $group)
            <br>
            <b>{{ $group->group_id }}</b> {{ $group->group_name }}
        @endforeach
    </notel0>
</div>
@endsection

```

oder

Bearbeiten

```

\resources\views\users\u_edit.blade.php
@extends('layouts.master')
@section('left')
    <div class="panel card-default col-md-offset-1">
        <div class="card-body">
            @include('layouts.session_message')
            {!! Form::model($user, ['class'=>'form', 'method' => 'PUT',
                'route'=> ['user.update', $user->id]] ) !!}
            @include('partials.input_string',
                ['label' => 'Name Vorname', 'value' => 'name'])
            @include('partials.input_string',
                ['label' => 'Mailadresse', 'value' => 'email'])
            @include('partials.input_check',
                ['label' => 'Webmaster', 'value' => 'admin'])
            @include('partials.input_check',
                ['label' => 'Gruppenadmin', 'value' => 'group_admin'])
            @include('partials.input_range',
                ['label' => 'Gruppennummer', 'value' => 'group_id',
                    'set' => $groups->count()])
            @include('partials.submit',
                ['submitButtonText' => 'Änderungen speichern'])
            {!! Form::close() !!}
        </div>
        {!! Form::open(['method' => 'DELETE',
            'route' => ['user.destroy', $user->id]]) !!}
            @include('partials.submit', ['submitButtonText' => 'Datensatz löschen'])
        {!! Form::close() !!}
        <br>
        {!! Form::open(['method' => 'GET', 'route' => ['user.index', $user->id]]) !!}
            @include('partials.submit', ['submitButtonText' => 'zurück'])
        {!! Form::close() !!}
        </div>
    @endsection
    @section('side')
        <b>Erfasste Personen</b>
        <note10>
            @foreach($people as $person)
                <br>
                <b>{{ $person->email }}</b>
                @if ($person->user_id > 0)
                    <b> Registriert</b>
                @endif
                {{ $person->name }}, {{ $person->street }}, {{ $person->town }}
                Tel: {{ $person->phone }}
            @endforeach
        </note10>
        <br>
        <b>Registrierte Benutzer</b>
        <note10>
            @foreach($users as $user)
                <br>
                <b>{{ $user->email }}</b> {{ $user->name }}
            @endforeach
        </note10>
        <br>
        <strong>Gruppen</strong>
        <note10>
            @foreach($groups as $group)
                <br>
                <b>{{ $group->group_id }}</b> {{ $group->group_name }}
            @endforeach
        </note10>
    @endsection

```

einen Benutzers diesen einer Gruppe zuweisen und eventuelle angeben, dass der Benutzer Gruppenverantwortlicher ist.

MyWeb Administrator ▾

Benutzerregistrierung
Die Mail-Adresse muss eindeutig sein!

Name Vorname

Mailadresse

Webmaster

Gruppenadmin

Gruppennummer

Erfasste Personen
Web@FJKarli.ch Registriert Administrator, Hofstatt 10, CH-4528 Zuchwil Tel: +41 79 321 92 47
FJK@FJKarli.ch Registriert Franz-Josef Karli, Hofstatt 10, CH-4528 Zuchwil Tel: +41 79 321 92 47

Registrierte Benutzer
Web@FJKarli.ch Administrator
FJK@FJKarli.ch Karli - Ryffel Franz-Josef

Gruppen
1 Allgemein
3 Karli-Kano
2 Karli-Ryffel
4 leer

F. J. Karli © 2018
Zögern Sie nicht, das Kontaktformular [☑](#) zu verwenden!

Der GroupController

```

\app\Http\Controllers\GroupController.php
<?php namespace MyWeb\Http\Controllers;
use MyWeb\Group;
use MyWeb\User;
use Illuminate\Http\Request;
class GroupController extends Controller {
    public function index() {
        $groups = Group::all();
        return view('group.g_index', compact('groups'));
    }
    public function create() {
        $groups = Group::all();
        return view('group.g_create', compact('groups'));
    }
    public function store(Request $request) {
        $this->validate(request(), [
            'group_name' => 'required'
        ]);
        Group::create([
            'group_name' => request('group_name'),
            'group_id' => Group::count() + 1
        ]);
        return \Redirect::route('group.index');
    }
    public function show(Group $group) {
        return view('group.g_show', compact('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

```

\resources\views\group\g_create.blade.php
@extends('layouts.master')
@section('left')
    <div class="panel panel-default col-md-offset-1">
        <div class="card-header">
            <strong>Erfassen einer Gruppe</strong>
        </div>
        <div class="panel-body">
            @include('layouts.session_message')
            {!! Form::open(['class' => 'form', 'route' => 'group.store']) !!}
            @include('partials.input_string',
                ['label' => 'Gruppenname', 'value' => 'group_name'])
            @include('partials.submit', ['submitButtonText' => 'Gruppe speichern'])
            {!! Form::close() !!}
        </div>
    </div>
@endsection

```

```

@section('side')
    {!! Form::open(['class'=>'form', 'method' => 'GET',
        'route' => ['group.index']] !!)
    @include('partials.submit', ['submitButtonText' => 'Zurück!'])
    {!! Form::close() !!}
    <div class="col-md-offset-1 col-md-10">
        <br>
        <b>Bestehende Gruppen</b>
        <br>
        @forelse($groups as $group)
            <div class="blog-post">
                Gruppe {{ $group->group_id }}:
                <a href="{{ route('group.show', $group->id) }}">{{ $group->group_name }}</a>
            </div>
        @empty
        @endforelse
    </div>
@endsection

```

bearbeiten

```

\resources\views\group\g_edit.blade.php
@extends('layouts.master')
@section('left')
    <div class="panel panel-default col-md-offset-1">
        <div class="card-header">
            <strong>Bearbeiten einer Gruppe</strong>
        </div>
        <div class="panel-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'])
            @include('partials.input_number',
                ['label' => 'Gruppennummer', 'value' => 'group_id'])
            @include('partials.submit', ['submitButtonText' => 'Änderungen speichern'])
            {!! Form::close() !!}
        </div>
    </div>
@endsection
@section('side')
    {!! Form::open(['class'=>'form', 'method' => 'GET',
        'route' => ['group.index']] !!)
    @include('partials.submit', ['submitButtonText' => 'Zurück!'])
    {!! Form::close() !!}
@endsection

```

aufgelisten

```

\resources\views\group\g_index.blade.php
@extends('layouts.master')
@section('left')
    <div class="col-md-offset-1 col-md-10">
        @foreach($groups as $group)
            <div class="blog-post">
                Gruppe {{ $group->group_id }}:
                <a href="{{ route('group.show', $group->id) }}">{{ $group->group_name }}</a>
            </div>
        @endforeach
    </div>
@endsection
@section('side')
@endsection

```

oder einzeln ansehen

```

\resources\views\group\g_show.blade.php
@extends('layouts.master')
@section('left')
    <div class="panel col-md-offset-1">
        Gruppe: {{ $group->group_id }}<br>
        Gruppenname: {{ $group->group_name }}
        <note10>
            {{ $group->created_at->diffForHumans() }} /
            Version {{ $group->updated_at->format('d, m, Y') }}
        </note10>
    </div>
@endsection

```

```

@section('side')
  {!! Form::open(['class'=>'form', 'method' => 'GET',
    'route' => ['group.edit', $group->id]]) !!}
  @include('partials.submit', ['submitButtonText' => 'Gruppe bearbeiten'])
  {!! Form::close() !!}
  {!! Form::open(['class'=>'form', 'method' => 'DELETE',
    'route' => ['group.destroy', $group->id]]) !!}
  @include('partials.submit', ['submitButtonText' => 'Gruppe löschen'])
  {!! Form::close() !!}
  {!! Form::open(['class'=>'form', 'method' => 'GET',
    'route' => ['group.index']] !!}
  @include('partials.submit', ['submitButtonText' => 'Zurück'])
  {!! Form::close() !!}
@endsection

```

Mit

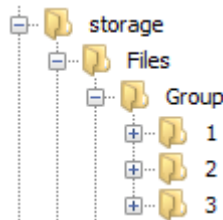
```

\routes\web.php
--
Route::resource('groups', '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'));
}
```

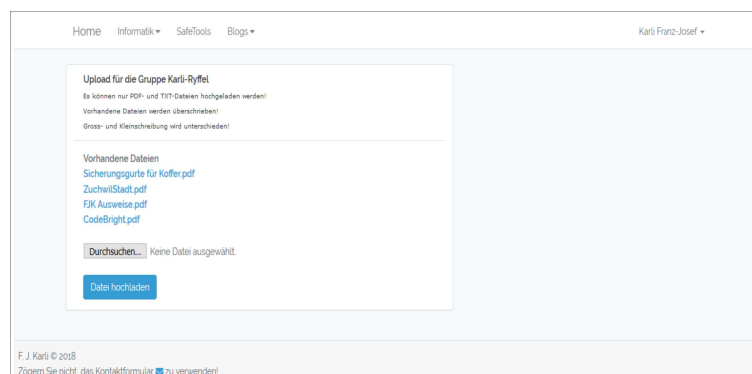
create übergibt Gruppennummer, Gruppenname und die Liste der erfasssten Gruppendokumente an die View **group.f_create**.

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

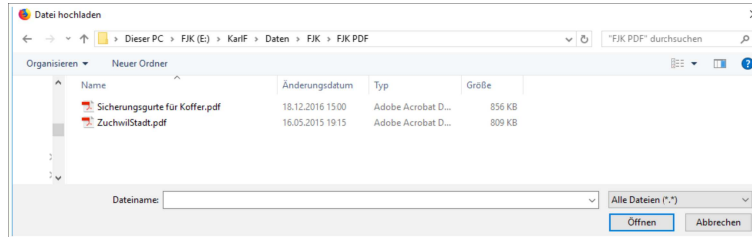
```
public function store(Request $request) {
    $id = Auth::user()->group_id;
    $file = $request->file('uploadedfile');
    switch ($file->getClientOriginalExtension()) {
        case 'pdf': case 'txt': case 'PDF': case 'TXT':
            $name = $file->getClientOriginalName();
            //Move 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 einer Tabelle gespeichert. Sollte eine Datei gleichen Namens hochgeladen werden wird die Tabelle nur nachgeführt und die existierende Datei überschrieben. Möglichweise wird beim Speichern ein neues Gruppenverzeichnis erstellt.

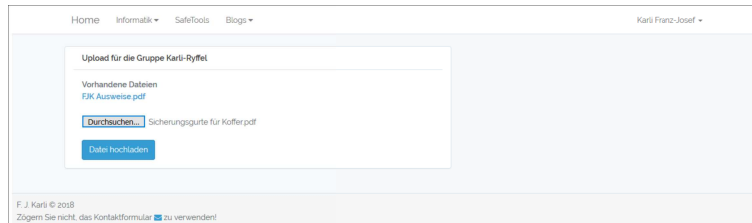
```
public function edit($id) {
    $groupFile = GroupFile::find($id);
    return view('group.f_edit', compact('groupFile'));
}
public function update(Request $request, $id) {
    GroupFile::find($id)->update(request(['file_doc']));
    return \Redirect::route('groupfile.index');
}
public function destroy($id) {
    $file = GroupFile::findOrFail($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');
}
}
```



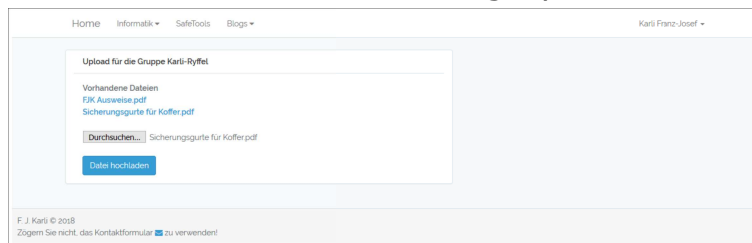
Ein Klick auf **Durchsuchen...** liefert das folgende Auswahlmenu.



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



kann diese hochgeladen und in der Datenbank gespeichert werden.



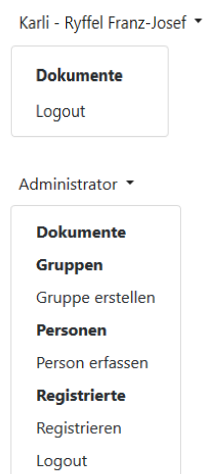
```
\routes\web.php
--
Route::resource('g_file', 'GroupFileController');
--
```

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
```



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

i_check

\resources\views\partials\input_check.blade.php

```
<div class="form-group row">
  <div class="col-md-4"></div>
  <div class="col-md-6">{{ Form::checkbox($value) }} {{ $label }}</div>
</div>
```

i_range

\resources\views\partials\input_range.blade.php

```
<div class="form-group row">
  <div class="col-md-4 col-form-label text-md-right">{{ Form::label($label) }}</div>
  <div class="col-md-6">{{ Form::selectRange($value, 1, $set) }}</div>
</div>
```

upload

https://www.tutorialspoint.com/laravel/laravel_file_uploading.htm

Uploading Files in Laravel is very easy. All we need to do is to create a view file where a user can select a file to be uploaded and a controller where uploaded files will be processed.

In a view file, we need to generate a file input by adding the following line of code.

```
Form::file('file_name');
```

In `Form::open()`, we need to add `'files'=>'true'` as shown below. This facilitates the form to be uploaded in multiple parts.

```
Form::open(array('url' => '/uploadfile', 'files'=>'true'));
```

Example

Step 1 – Create a view file called **resources/views/uploadfile.php** and copy the following code in that file.

```
resources/views/uploadfile.php
<html>
  <body>
    <?php
      echo Form::open(array('url' => '/uploadfile', 'files'=>'true'));
      echo 'Select the file to upload.';
      echo Form::file('image');
      echo Form::submit('Upload File');
      echo Form::close();
    ?>
  </body>
</html>
```

Step 2 – Create a controller called **UploadFileController** by executing the following command.

```
php artisan make:controller UploadFileController --plain
```

Step 3 – Copy the following code

```
app\Http\Controllers/UploadFileController.php
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Http\Requests;
use App\Http\Controllers\Controller;
class UploadFileController extends Controller {
    public function index(){
        return view('uploadfile');
    }
    public function showUploadFile(Request $request){
        $file = $request->file('image');
        //Display File Name
        echo 'File Name: '.$file->getClientOriginalName();
        echo '<br>';
        //Display File Extension
        echo 'File Extension: '.$file->getClientOriginalExtension();
        echo '<br>';
        //Display File Real Path
        echo 'File Real Path: '.$file->getRealPath();
        echo '<br>';
        //Display File Size
        echo 'File Size: '.$file->getSize();
        echo '<br>';
        //Display File Mime Type
        echo 'File Mime Type: '.$file->getMimeType();
        //Move Uploaded File
        $destinationPath = 'uploads';
        $file->move($destinationPath,$file->getClientOriginalName());
    }
}
```

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

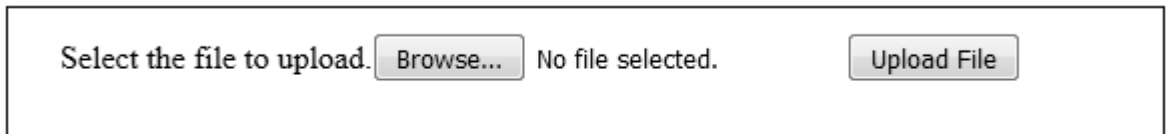
```
app/Http/routes.php
```

```
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 {{ $name }}</b><br>
            <note10>
                Mit Klick auf einen Filename kann die Beschreibung bearbeitet oder
                die Datei gelöscht werden<br>
            </note10>
        </div>
        <div class="panel-body">
            @foreach($files as $file)
                <a href="{{ route('g_file.edit', $file) }}">
                    <b>{{ $file->file_name }}</b>
                </a>: {{ $file->file_doc }}
                <br>
            @empty
            @endforeach
        </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 {{ $name }}</b><br>
            <note10>
                Es können nur PDF- und TXT-Dateien hochgeladen werden!
                Vorhandene Dateien werden überschrieben!
                Gross- und Kleinschreibung wird unterschieden!
            </note10><br>
        </div>
        <div class="card-body">
            @foreach($files as $file)
                <a href="{{ config('app.ownfiles', '') }}Group/{{ $id }}/{{ $file->file_name }}">
                    <b>{{ $file->file_name }}</b>
                </a>: {{ $file->file_doc }}
                <br>
            @empty
            @endforeach
        </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() !!}
            <br>
            {!! Form::open(['class' => 'form', 'method' => 'GET',
                'route' => ['g_file.index']]) !!}
            {!! Form::submit('Zurück', ['class' => 'btn btn-primary']) !!}
            {!! Form::close() !!}
        </div>
    </div>
```

@endsection