

Auf der Webseite sollen Kopien von privaten Dokumenten (Ausweise, Kopien von Flugscheinen, usw.) gespeichert werden. Diese Dokumente sollen von Mitgliedern einer Gruppe eingesehen werden können. Pro Gruppe wird ein Verantwortlicher bestimmt. Dieser ist dafür verantwortlich, dass veraltete Dokumente der Gruppe gelöscht werden. Um Gruppen speichern zu können, braucht es ein neues Modell.

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

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

```
\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');
    }
}
```

Um registrierte Benutzer - User - einer Gruppe zuweisen zu können, muss in der Tabelle **users** ein Feld **group** eingeführt werden.

```
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 die Tabelle **groups** erstellt und in der Tabelle **users** das Feld **group_id** eingefügt. Ebenso muss in der Tabelle **users** ein Feld **group_admin** eingeführt werden.

```
<?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');
        });
    }
}
```

Gespeicherte Dokumente müssen durch einen Gruppenverantwortlichen gelöscht werden können. Die Tabelle **users** muss daher um eine neue Spalte erweitert werden.

```
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
```

Ein User kann Administrator, ein Mitglied einer Gruppe und Gruppenverantwortlicher sein. Eine Gruppe kann mehrere Mitglieder haben. Dies wird in den Modellen **User** und **Group** wie folgt angegeben.

```
\app\User.php
<?php namespace FJKarli;
--
use FJKarli\Group;
--
class User extends Authenticatable {
--
    protected $fillable = [ 'name', 'email', 'password',
        'admin', 'group_id', 'group_admin' ];
--
    public function group() { return $this->hasOne(Group::class); }
}
```

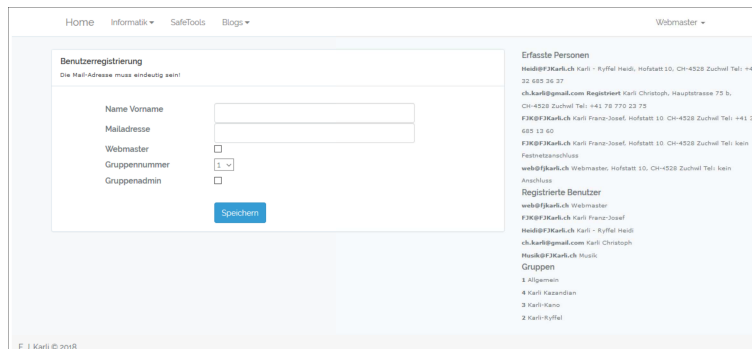
hasOne sagt aus, dass ein User nur Mitglied einer einzigen Gruppe sein kann.

```
\app\Group.php
<?php namespace FJKarli;
use Illuminate\Database\Eloquent\Model;
class Group extends Model {
}

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

hasMany sagt aus, dass in einer Gruppe mehrere User sein können.

Die Registrierung und Bearbeitung von registrierten Benutzern muss überarbeitet werden. Er kann beim Registrieren (`u_create`) oder Bearbeiten (`u_edit`) eines Benutzers diesen einer Gruppe zuweisen.



Dabei wurden zwei neue Input-Felder eingeführt:

```
\resources\views\partials\input_check.blade.php
<div class="form-group">
  <div class="col-md-offset-1 col-md-3">{{ Form::label($label) }}</div>
  <div class="col-md-3">
    {{ Form::checkbox($value) }}
  </div>
  <div class="col-md-9">
  </div>
</div>
```

`input_check` prüft ob des entsprechende Feld markiert ist.

```
\resources\views\partials\input_range.blade.php
<div class="form-group">
  <div class="col-md-offset-1 col-md-3">{{ Form::label($label) }}</div>
  <div class="col-md-3">
    {{ Form::selectRange($value, 1, $set) }}
  </div>
  <div class="col-md-9">
  </div>
</div>
```

`input_range` bietet eine Auswahl der erfassten Gruppennummern.

Der UserController (`u_contr`) muss angepasst werden, so dass die Werte der neuen Felder erfasst werden können.

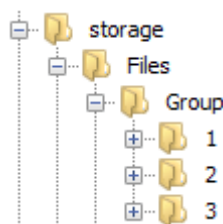
Der GroupController (`g_controll`) sorgt dafür, dass der Administrator Gruppen erstellen (`g_create`), bearbeiten (`g_edit`), aufgelisten (`g_index`) oder einzeln ansehen (`g_show`) kann.

Mit der Anpassung

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

können die Funktionen der 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 (^{upload}) aus dem Internet 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->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 FJKarli;
use FJKarli\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 (**gf_contr**) und

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

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

```
app\Http\Controllers\GroupFileController.php
<?php namespace FJKarli\Http\Controllers;
use Auth; use FJKarli\Group;
use FJKarli\GroupFile;
use FJKarli\User;
use Illuminate\Http\Request;
use League\Flysystem\Adapter\Local;
use League\Flysystem\Filesystem;
class GroupFileController extends Controller {
    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)) {
```

```

        if ($filesystem->delete($file->file_name)) {
        } else {
            dd('file not deleted');
        };
    };
    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'));
}

```

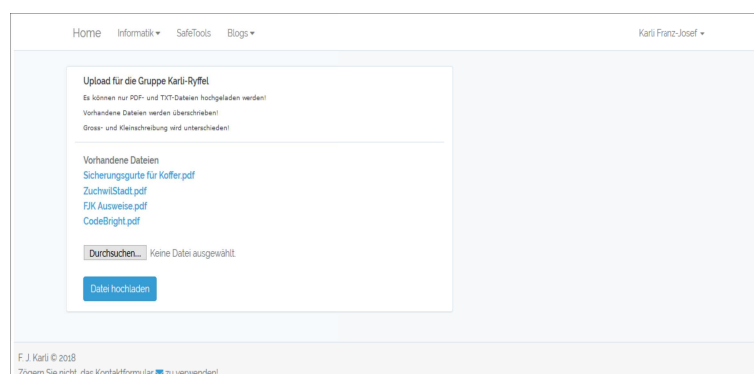
create übergibt Gruppennummer, Gruppenname und die Liste der erfassten Gruppenelemente an die View **group.f_create**. Die erfassten Daten werden von der Funktion **store** bearbeitet

```

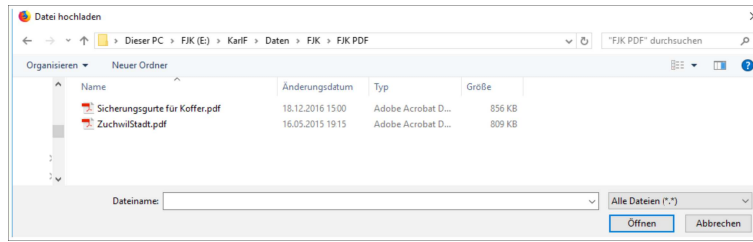
public function store(Request $request) {
    $id = Auth::user()->group_id;
    $file = $request->file('uploadedfile');
    switch ($file->getClientOriginalExtension()) {
        case 'pdf':
        case 'txt':
        case 'PDF':
        case 'TXT':
            $name = $file->getClientOriginalName();
            //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
                ]);
            } else {
                //save in group_files table
                $upload = new GroupFile;
                $upload->group_id = $id;
                $upload->file_name = $name;
                $upload->file_path = $destinationPath;
                $upload->save();
            }
    }
    return back();
}

```

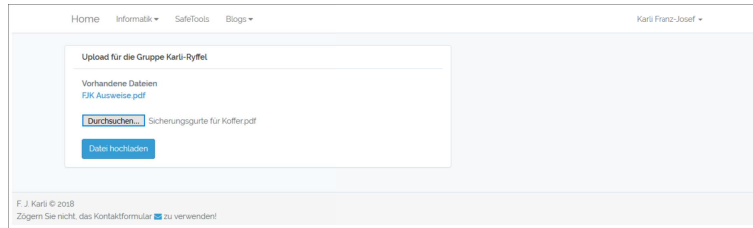
Falls eine Datei hochgeladen wird, werden die entsprechenden Zugriffsdaten in einer Tabelle gespeichert. Sollte eine Datei gleichen Namens hochgeladen werden wird die Tabelle nur nachgeföhrt und die existierende Datei überschrieben. Möglichweise wird beim Speichern ein neues Gruppenverzeichnis erstellt.



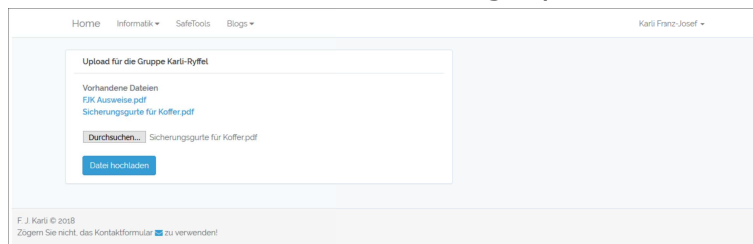
Ein Klick auf **Durchsuchen...** liefert das folgende Auswahlmenü.



Nach dem Markieren der gewünschten Datei und Klick auf Öffnen



kann diese hochgeladen und in der Datenbank gespeichert werden.

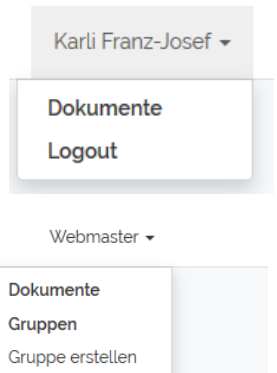


Diese Seiten stehen ausschliesslich registrierten Benutzern zur Verfügung.

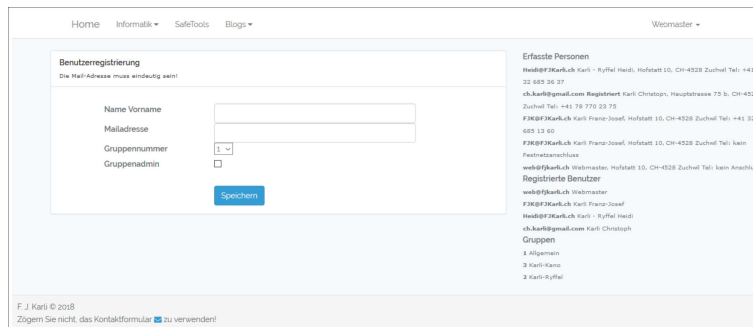
Mit

```
\resources\views\layouts\nav.blade.php
```

```
--
<li>
<li>
<a href="{{ route('groupfile.create') }}">
<b>Dokumente</b>
</a>
</li>
@if (Auth::user()->admin)
<li>
<a href="{{ route('group.index') }}">
<b>Gruppen</b>
</a>
</li>
</li>
<li>
<a href="{{ route('group.create') }}">
Gruppe erstellen
</a>
</li>
</li>
```



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



Der Administrator kann

- Administrator-Rechte vergeben¹,
- Benutzer einer Gruppe zuweisen und
- Ein Gruppenmitglied als Gruppenverantwortlicher bezeichnen²

Zu diesem Zweck müssen UserController (UserC) und die Formulare zum Erfassen (u_create) respektive Bearbeiten (u_edit) von Benutzerdaten müssen angepasst werden.

Die Zuweisung zu einer Gruppe erfolgt mit:

```
\resources\views\partials\input_range.blade.php
<div class="form-group">
<div class="col-md-offset-1 col-md-3">{{ Form::label($label) }}</div>
<div class="col-md-3">
{{ Form::selectRange($value, 1, $set) }}
</div>
<div class="col-md-9"> </div>
</div>
```

Der Aufruf sieht wie folgt aus:

```
@include('partials.input_range',
['label' => 'Gruppennummer', 'value' => 'group_id', 'set' => $groups->count()])
```

Die Auswahl ist mit der Anzahl der vorhandenen Gruppen begrenzt.

¹ Mit Vorteil wird der Vergabe von Administrator-Rechten besonderes Augenmerk geschenkt.

² Dieser ist für die Wartung der hochgeladenen Dateien, usw. verantwortlich.

Die Vergabe Administratorrechten respektive die Bezeichnung als Gruppenverantwortlicher erfolgt mit

```
\resources\views\partials\input_check.blade.php
<div class="form-group">
  <div class="col-md-offset-1 col-md-3">{{ Form::label($label) }}</div>
  <div class="col-md-3"> {{ Form::checkbox($value) }} </div>
</div class="col-md-9"> </div>
</div>
```

respektive mit

```
@include('partials.input_check', ['label' => 'Webmaster', 'value' => 'admin'])

@include('partials.input_check', ['label' => 'Gruppenadmin', 'value' => 'group_admin'])
```

Die Angaben werden im UserController mit den Funktionen **store**

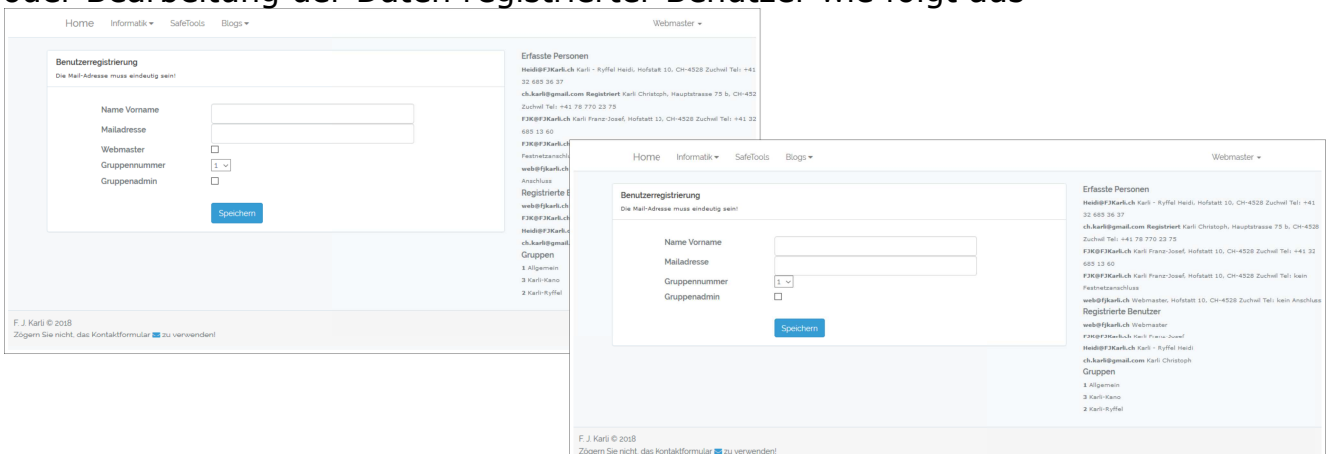
```
public function store(Request $request) {
  \Mail::to($request->get('email'), '');
  \Mail::send('users.u_mail', [
    'name' => $request->get('name'),
    'email' => $request->get('email')
  ],
  function($message) {
    $message->from('web@FJKarli.ch')
    ->subject('Registrierung auf www.MyWeb.ch');
  });
  $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');
}
```

respektive **update** in der Datenbank gespeichert.

```
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');
}
```

Zur Unterstützung des Administrators wird die Liste der aktuellen Gruppen ausgegeben.

Mit diesen Anpassungen sehen die Formular zur Registrierung eines Benutzers oder Bearbeitung der Daten registrierter Benutzer wie folgt aus



Mit der Anpassung der Menüzeile

```
<li>
  <a href="{{ route('groupfile.create') }}"><b>Dokumente</b></a>
</li>
@if (Auth::user()->group_admin)
  <li>
    <a href="{{ route('group.index') }}">Dokumente verwalten</a>
  </li>
@endif
```

hat der Gruppenverantwortliche hat nun einen neuen Link.



Mit Klick auf Dokumente verwalten

u_create

\resources\views\users\u_create.blade.php

```
@extends('layouts.master')
@section('left')
    <div class="panel panel-default col-md-offset-1">
        <div class="panel-heading">
            <strong>Benutzerregistrierung</strong>
            <notel0><br>
                Die Mail-Adresse muss eindeutig sein!
            </notel0>
        </div>
        <div class="panel-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_range',
                ['label' => 'Gruppennummer', 'value' => 'group_id',
                 'set' => $groups->count()])
            @include('partials.input_check',
                ['label' => 'Gruppenadmin', 'value' => 'group_admin'])
            @include('layouts.submit',
                ['submitButtonText' => 'Speichern'])
            {!! Form::close() !!}
        </div>
    </div>
@endsection
@section('side')
    <strong>Erfasste Personen</strong>
    <notel0>
        @foreach($people as $person)
            <br>
            <b>{{ $person->email }}</b>
            @if ($person->user_id > 0)
                <strong> Registriert</strong>
            @endif
            {{ $person->name }}, {{ $person->street }}, {{ $person->town }}
            Tel: {{ $person->phone }}
        @endforeach
    </notel0>
    <br>
    <strong>Registrierte Benutzer</strong>
    <notel0>
        @foreach($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>
@endsection
```

u_edit

\resources\views\users\u_create.blade.php

```
@extends('layouts.master')
@section('left')
    <div class="panel panel-default col-md-offset-1">
        <div class="panel-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_range',
                ['label' => 'Gruppennummer', 'value' => 'group_id',
                 'set' => $groups->count()])
        </div>
    </div>
@endsection
```

```

        @include('partials.input_check',
            ['label' => 'Gruppenadmin', 'value' => 'group_admin'])
        @include('layouts.submit',
            ['submitButtonText' => 'Änderungen speichern'])
        {!! Form::close() !!}
    </div>
    {!! Form::open(['method' => 'DELETE',
        'route' => ['user.destroy', $user->id]]) !!}
    @include('layouts.submit', ['submitButtonText' => 'Datensatz löschen'])
    {!! Form::close() !!}
    {!! Form::open(['method' => 'GET', 'route' => ['user.index', $user->id]]) !!}
    @include('layouts.submit', ['submitButtonText' => 'zurück'])
    {!! Form::close() !!}
</div>
@endsection
@section('side')
<strong>Erfasste Personen</strong>
<note10>
    @foreach($people as $person)
        <br>
        <b>{{ $person->email }}</b>
        @if ($person->user_id > 0)
            <strong> Registriert</strong>
        @endif
        {{ $person->name }}, {{ $person->street }}, {{ $person->town }} Tel:
    {{ $person->phone }}
    @endforeach
</note10>
<br>
<strong>Registrierte Benutzer</strong>
<note10>
    @foreach($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

```

u_contr

\app\Http\Controllers\UserController.php

```

<?php namespace FJKarli\Http\Controllers;
use FJKarli\Group;
use FJKarli\Person;
use FJKarli\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) {
    return view('users.u_show', compact('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');
}
}
}

```

g_controll

\app\Http\Controllers\GroupController.php

```

<?php namespace FJKarli\Http\Controllers;
use FJKarli\Group;
use FJKarli\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');
    }
}
}

```

g_create

\resources\views\group\g_create.blade.php

```

@extends('layouts.master')
@section('left')
    <div class="panel panel-default col-md-offset-1">
        <div class="panel-heading">

```

```

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

```

g_edit

\resources\views\group\g_edit.blade.php

```

@extends('layouts.master')
@section('left')
    <div class="panel panel-default col-md-offset-1">
        <div class="panel-heading">
            <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('layouts.submit', ['submitButtonText' => 'Änderungen speichern'])
            {!! Form::close() !!}
        </div>
    </div>
@endsection
@section('side')
    {!! Form::open(['class'=>'form', 'method' => 'GET',
        'route' => ['group.index']]) !!}
    @include('layouts.submit', ['submitButtonText' => 'Zurück'])
    {!! Form::close() !!}
@endsection

```

g_index

\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

```

g_show

\resources\views\group\g_show.blade.php

```

@extends('layouts.master')
@section('left')
    <div class="panel col-md-offset-1">

```

```

        Gruppe: {{ $group->group_id }}<br>
        Gruppename: {{ $group->group_name }}
    <notel0>
        {{ $group->created_at->diffForHumans() }} /
        Version {{ $group->updated_at->format('d, m, Y') }}
    </notel0>
</div>
@endsection
@section('side')
    {!! Form::open(['class'=>'form', 'method' => 'GET',
        'route' => ['group.edit', $group->id]]) !!}
    @include('layouts.submit', ['submitButtonText' => 'Gruppe bearbeiten'])
    {!! Form::close() !!}
    {!! Form::open(['class'=>'form', 'method' => 'DELETE',
        'route' => ['group.destroy', $group->id]]) !!}
    @include('layouts.submit', ['submitButtonText' => 'Gruppe löschen'])
    {!! Form::close() !!}
    {!! Form::open(['class'=>'form', 'method' => 'GET',
        'route' => ['group.index']] !!}
    @include('layouts.submit', ['submitButtonText' => 'Zurück'])
    {!! Form::close() !!}
@endsection

```

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**.

```

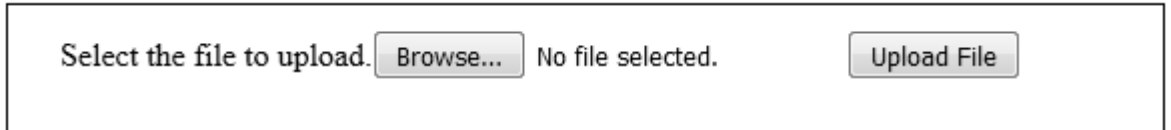
app/Http/routes.php
Route::get('/uploadfile','UploadFileController@index');
Route::post('/uploadfile','UploadFileController@showUploadFile');

```

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.



gf_contr

```

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'));
}

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
                ]);
            } else {
                //save in group_files tabel
                $upload = new GroupFile;
                $upload->group_id = $id;
                $upload->file_name = $name;
                $upload->file_path = $destinationPath;
                $upload->save();
            }
        }
    }
    return back();
}

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)) {
        if ($filesystem->delete($file->file_name)) {
            } else {
                dd('file not deleted');
            }
        };
    };
    return \Redirect::route('groupfile.index');
}

public function edit(GroupFile $groupFile) { }
public function update(Request $request, GroupFile $groupFile) { }

```

```

    public function destroy(GroupFile $groupFile) { }
}

```

f_index

\resources\views\group\f_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

```

f_create

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

```

@extends('layouts.master')
@section('content')
    <div class="panel panel-default col-md-offset-1 col-md-6">
        <div class="panel-heading">
            <b>Upload für die Gruppe {{ $name }}</b><br>
            <notel0>
                Es können nur PDF- und TXT-Dateien hochgeladen werden!<br>
                Vorhandene Dateien werden überschrieben!<br>
                Gross- und Kleinschreibung wird unterschieden!
            </notel0>
        </div>
        <div class="panel-body">
            <b>Vorhandene Dateien</b><br>
            @foreach($files as $file)
                <a href="{{ config('app.ownfiles', '') }}Group/{{ $id }}/{{ $file->file_name }}">
                    <b>{{ $file->file_name }}</b>
                </a>
            <br>
            @endforeach
            <br>
            {!! Form::open(['class' => 'form', 'files'=>'true', 'method' => 'POST', 'route' => ['groupfile.store', $id]]) !!}
            {!! Form::file('uploadedfile') !!}<br>
            {!! Form::submit('Datei hochladen', ['class' => 'btn btn-primary']) !!}
            {!! Form::close() !!}
        </div>
    </div>
@endsection

```

UserC

app\Http\Controllers\UserController.php

```

<?php namespace FJKarli\Http\Controllers;
use FJKarli\Group;
use FJKarli\Person;
use FJKarli\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) {
    return view('users.u_show', compact('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');
}
}
}

```

u_create

\resources\views\users\user_create.blade.php

```

@extends('layouts.master')
@section('left')
    <div class="panel panel-default col-md-offset-1">
        <div class="panel-heading">
            <strong>Benutzerregistrierung</strong>
            <notel0><br>
                Die Mail-Adresse muss eindeutig sein!
            </notel0>
        </div>
        <div class="panel-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_range',
                ['label' => 'Gruppennummer', 'value' => 'group_id',
                 'set' => $groups->count()])
            @include('partials.input_check',
                ['label' => 'Gruppenadmin', 'value' => 'group_admin'])
            @include('layouts.submit',
                ['submitButtonText' => 'Speichern'])
            {!! Form::close() !!}
        </div>
    </div>
@endsection
@section('side')
    <strong>Erfasste Personen</strong>
    <notel0>
        @foreach($people as $person)
            <br>
            <b>{{ $person->email }}</b>
            @if ($person->user_id > 0)
                <strong> Registriert</strong>
            @endif
            {{ $person->name }}, {{ $person->street }}, {{ $person->town }}
        </notel0>
    </strong>

```

```

        Tel: {{ $person->phone }}
    @endforeach
</notel10>
<br>
<strong>Registrierte Benutzer</strong>
<notel10>
    @foreach($users as $user)
        <br>
        <b>{{ $user->email }}</b> {{ $user->name }}
    @endforeach
</notel10>
<br>
<strong>Gruppen</strong>
<notel10>
    @foreach($groups as $group)
        <br>
        <b>{{ $group->group_id }}</b> {{ $group->group_name }}
    @endforeach
</notel10>
@endsection

```

u_edit

\resources\views\users\u_edit.blade.php

```

@extends('layouts.master')
@section('left')
    <div class="panel panel-default col-md-offset-1">
        <div class="panel-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_range',
                ['label' => 'Gruppennummer', 'value' => 'group_id',
                    'set' => $groups->count()])
            @include('partials.input_check',
                ['label' => 'Gruppenadmin', 'value' => 'group_admin'])
            @include('layouts.submit',
                ['submitButtonText' => 'Änderungen speichern'])
            {!! Form::close() !!}
        </div>
        {!! Form::open(['method' => 'DELETE',
            'route' => ['user.destroy', $user->id]]) !!}
        @include('layouts.submit', ['submitButtonText' => 'Datensatz löschen'])
        {!! Form::close() !!}
        {!! Form::open(['method' => 'GET', 'route' => ['user.index', $user->id]]) !!}
        @include('layouts.submit', ['submitButtonText' => 'zurück'])
        {!! Form::close() !!}
    </div>
@endsection
@section('side')
    <strong>Erfasste Personen</strong>
    <notel10>
        @foreach($people as $person)
            <br>
            <b>{{ $person->email }}</b>
            @if ($person->user_id > 0)
                <strong> Registriert</strong>
            @endif
            {{ $person->name }}, {{ $person->street }}, {{ $person->town }}
            Tel: {{ $person->phone }}
        @endforeach
    </notel10>
    <br>
    <strong>Registrierte Benutzer</strong>
    <notel10>
        @foreach($users as $user)
            <br>
            <b>{{ $user->email }}</b> {{ $user->name }}
        @endforeach
    </notel10>
    <br>
    <strong>Gruppen</strong>
    <notel10>
        @foreach($groups as $group)

```

```
<br>
  <b>{{ $group->group_id }}</b> {{ $group->group_name }}
@endforeach
</note10>
@endsection
```