<?php

namespace App\Http\Controllers;

use App\Group;

use App\Requestt;

use App\Student;

use Auth;

use DB;

use Illuminate\Http\Request;

class CreateGroupController extends Controller

{

/\*\*

\* Display a listing of the resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function index()

{

//Getting current user info

$s = Student::find(Auth::user()->id);

//Getting current user's group iniformation

$CurrGroup = $s->group\_id;

//Getting number of students in group.

$temp = Student::where('group\_id', $CurrGroup)->where('group\_id', '!=', null)->count();

if ($temp > 2) {

return redirect('/')->with('WarningMessage', 'Your group is full !');

} else

if (Auth::user()->user\_role == 5) {

//Get ID of Current user

$Curruser = Auth::user()->id;

//Make query to see if student is elligble to make a request

$dept = DB::table('students')->where('id', $Curruser)->value('department\_id');

$students = DB::select(DB::raw("SELECT students.id, students.department\_id,students.group\_id, students.first\_name, students.last\_name

from students WHERE students.id != $Curruser AND students.department\_id = $dept AND students.id NOT IN

(SELECT students.id FROM students, requestts WHERE requestts.sender\_id = $Curruser AND requestts.Rec\_id = students.id AND requestts.status !=2 ) AND students.id NOT IN

(SELECT students.id FROM students, requestts WHERE requestts.sender\_id = students.id AND requestts.Rec\_id = $Curruser ) "));

$Myreq = DB::select(DB::raw("SELECT requestts.id, students.first\_name, students.last\_name, requestts.status FROM requestts, students WHERE sender\_id = $Curruser AND requestts.Rec\_id = students.id"));

$Myreq1 = DB::select(DB::raw("SELECT requestts.id,students.id AS STUID , students.first\_name, students.last\_name, students.group\_id FROM requestts, students WHERE sender\_id = students.id AND requestts.Rec\_id = $Curruser AND requestts.status = 3"));

return view('CreateGroup', compact('students', 'Myreq', 'Myreq1'));

}

//Do some coding here.

else {

return 'False';

}

}

/\*\*

\* Show the form for creating a new resource.

\*

\* @return \Illuminate\Http\Response

\*/

public function create()

{

//

}

/\*\*

\* Store a newly created resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @return \Illuminate\Http\Response

\*/

public function store(Request $request)

{

//

$requestt = new Requestt();

$requestt->id = null;

$requestt->Sender\_id = Auth::user()->id;

$requestt->Rec\_id = $request->StudentID;

$requestt->status = 3;

$requestt->save();

return redirect('CreateGroup');

//return $request->StudentID;

}

/\*\*

\* Display the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function show($id)

{

//

}

/\*\*

\* Show the form for editing the specified resource.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function edit($id)

{

//

}

/\*\*

\* Update the specified resource in storage.

\*

\* @param \Illuminate\Http\Request $request

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function update(Request $request, $id)

{

//

}

/\*\*

\* Remove the specified resource from storage.

\*

\* @param int $id

\* @return \Illuminate\Http\Response

\*/

public function destroy($id)

{

//

}

public function Approve(Request $request)

{

$Curruser = Auth::user()->id;

$CurrUserGroup = DB::table('students')->where('id', $Curruser)->value('group\_id');

$OtherUserGroup = DB::table('students')->where('id', $request->student\_id)->value('group\_id');

if (!empty($CurrUserGroup)) {

// Check weather the current user already has a group.

$s = Student::find($request->student\_id);

$OtherStuGroup = $s->group\_id;

//Numnber of student within the group

$temp = Student::where('group\_id', $OtherStuGroup)->where('group\_id', '!=', null)->count();

//Check if the group of the current user's memebers are less than 3

if ($temp < 3) {

//Adding 1 to the members of the group

$group = Group::find($CurrUserGroup);

$group->NumberOfStudents++;

$group->save();

//Adding the group id to the current user

$student = Student::find($request->student\_id);

$student->group\_id = $CurrUserGroup;

$student->save();

}

//return an error if the student accepts a request while he's in a group.

// return redirect('CreateGroup')->with('ErrorMessage', 'Can not accept request, you are already a group member !');

} else {

//else means, that the current user doesn't have a group yet.

$tempp = Requestt::find($request->requestt\_id);

$tempp->status = 1;

$tempp->save();

/\*\* Adding a specific student to an already existing group \*/

if (!empty($OtherUserGroup)) {

$student = Student::find($Curruser);

$student->group\_id = $OtherUserGroup;

$student->save();

$group = Group::find($OtherUserGroup);

$group->NumberOfStudents++;

$group->save();

return redirect('\home')->with('SuccessMessage', 'You have been added to the group !');

} else {

//Generating rndom id for a new group

$temp\_GID = rand(1, 1000);

$group = new Group();

$group->id = $temp\_GID;

$group->NumberOfStudents = 2;

$group->save();

$student = Student::find($Curruser);

$student->group\_id = $temp\_GID;

$student->save();

$student = Student::find($request->student\_id);

$student->group\_id = $temp\_GID;

$student->save();

$group = Group::find($temp\_GID);

$group->leader = $Curruser;

$group->save();

return redirect('\CreateGroup')->with('SuccessMessage', 'You have been added to the group !');

}

}

}

public function Reject(Request $request)

{

// When reject button is clicked the status field of that request is updated

$requestt = Requestt::find($request->requestt\_id);

$requestt->status = 2;

$requestt->save();

return redirect('\CreateGroup')->with('SuccessMessage', 'Request rejected !');

}

}