UserService

*/\*\*  
 \* Display popup to add project By Admin.  
 \*  
 \** ***@param*** *\Illuminate\Http\Request $request  
 \** ***@param*** *int $id  
 \** ***@return*** *\Illuminate\Http\Response  
 \*/* public function addProjectModal(Request $request)  
 {  
 $technologies = TechnologyStack::*all*();  
 $technology\_stack\_dropdown = view('utils.technology\_stack\_dropdown',['technologies'=>$technologies])->render();  
 $project\_managers = RoleUser::*with*('user')->where('role\_id',4)->get();  
 $containerId = $request->input('containerId', 'common\_popup\_modal');  
 $projectManagersDropDown = view('utils.project\_managers\_dropdown', ['project\_managers' => $project\_managers])->render();  
 $html = view('pages.admin.projects.\_partial.\_add\_project\_modal', ['id' => $containerId, 'data' => null,'technology\_stack\_dropdown'=>$technology\_stack\_dropdown, 'project\_managers\_dropdown' => $projectManagersDropDown])->render();  
 return $this->successResponse('success', ['html' => $html]);  
 }  
 */\*\*  
 \* Click Add to add project in the project list.  
 \*  
 \** ***@param*** *\Illuminate\Http\Request $request  
 \** ***@param*** *int $id  
 \** ***@return*** *\Illuminate\Http\Response  
 \*/* public function confirmAddProjectModal(Request $request)  
 {  
 $validator = Validator::make($request->all(), [  
 'project\_name' => 'required|min:3|max:30',  
 'project\_description' => 'required|max:100',  
 'project\_manager\_id' => 'required|numeric',  
 'technology\_stack\_id' => 'required|numeric',  
 ]);  
 if ($validator->fails()) {  
 return $this->error('Validation Failed', ['errors' => $validator->errors()]);  
 }  
 $project = new Project;  
 $project->name = $request->project\_name;  
 $project->description =$request->project\_description;  
 $project->user\_id =$request->project\_manager\_id;  
// $project->start\_date =Carbon::parse($request->date);  
  
 $project->save();  
 $project->technologystack()->attach(['technology\_stack\_id'=>$request->technology\_stack\_id]);  
 $projects = Project::*with*('users')->get();  
 $html = view('pages.admin.projects.\_partial.\_project\_list\_table\_html',['projects'=>$projects])->render();  
 return $this->successResponse('Project Added Successfully',['html'=>$html,'html\_section\_id'=>'project-list-section']);  
 }  
 */\*\*  
 \* Display popup to edit Project Attributes.  
 \*  
 \** ***@param*** *\Illuminate\Http\Request $request  
 \** ***@param*** *int $id  
 \** ***@return*** *\Illuminate\Http\Response  
 \*/* public function editProjectModal(Request $request)  
 {  
 $project\_id = $request->id;  
 $project = Project::*find*($project\_id);  
 $project\_manager\_id = $project->user\_id;  
 $project\_managers = RoleUser::*with*('user')->where('role\_id',4)->get();  
 $containerId = $request->input('containerId', 'common\_popup\_modal');  
 $projectManagersDropDown = view('utils.project\_managers\_dropdown', ['project\_managers' => $project\_managers,'project\_manager\_id'=>$project\_manager\_id])->render();  
 $html = view('pages.admin.projects.\_partial.\_edit\_project\_modal', ['id' => $containerId, 'data' => null, 'project\_managers\_dropdown' => $projectManagersDropDown,'project'=>$project])->render();  
 return $this->successResponse('success', ['html' => $html]);  
 }  
 */\*\*  
 \* Click Update Button to edit Project Attributes.  
 \*  
 \** ***@param*** *\Illuminate\Http\Request $request  
 \** ***@param*** *int $id  
 \** ***@return*** *\Illuminate\Http\Response  
 \*/* public function confirmEditProjectModal(Request $request)  
 {  
 $validator = Validator::make($request->all(), [  
 'project\_name' => 'required|min:3|max:30',  
 'project\_description' => 'required|max:30',  
 'project\_manager\_id' => 'required|numeric',  
 ]);  
 if ($validator->fails()) {  
 return $this->error('Validation Failed', ['errors' => $validator->errors()]);  
 }  
 $project\_id = $request->id;  
 $project = Project::*find*($project\_id);  
 $project->name = $request->project\_name;  
 $project->description =$request->project\_description;  
 $project->user\_id =$request->project\_manager\_id;  
 $project->start\_date =Carbon::*parse*($request->date);  
 $project->save();  
 $projects = Project::*with*('users')->get();  
 $html = view('pages.admin.projects.\_partial.\_project\_list\_table\_html',['projects'=>$projects])->render();  
 return $this->successResponse('Project Updated Successfully',['html'=>$html,'html\_section\_id'=>'project-list-section']);  
 }  
 */\*\*  
 \* Display Popup to delete Project form DB.  
 \*  
 \** ***@param*** *\Illuminate\Http\Request $request  
 \** ***@param*** *int $id  
 \** ***@return*** *\Illuminate\Http\Response  
 \*/* public function deleteProjectModal(Request $request)  
 {  
 $project\_id = $request->id;  
 $containerId = $request->input('containerId', 'common\_popup\_modal');  
 $html = view('pages.admin.projects.\_partial.\_delete\_project\_modal', ['id' => $containerId, 'project\_id' => $project\_id])->render();  
 return $this->successResponse('success', ['html' => $html]);  
 }  
 */\*\*  
 \* Click Delete button to delete project from DB.  
 \*  
 \** ***@param*** *\Illuminate\Http\Request $request  
 \** ***@param*** *int $id  
 \** ***@return*** *\Illuminate\Http\Response  
 \*/* public function confirmDeleteProjectModal(Request $request)  
 {  
 $project = Project::*find*($request->project\_id);  
 $project->delete();  
 $projects = Project::*with*('users')->get();  
 $html = view('pages.admin.projects.\_partial.\_project\_list\_table\_html',['projects'=>$projects])->render();  
 return $this->successResponse('Project Added Successfully',['html'=>$html,'html\_section\_id'=>'project-list-section']);  
 }  
 */\*\*  
 \* Display Developer Request Modal.  
 \*  
 \** ***@param*** *\Illuminate\Http\Request $request  
 \** ***@param*** *int $id  
 \** ***@return*** *\Illuminate\Http\Response  
 \*/* public function developersRequestModal(Request $request)  
 {  
 $project\_id = $request->id;  
 $containerId = $request->input('containerId', 'common\_popup\_modal');  
 $html = view('pages.projectManager.\_partial.\_developer\_request\_modal',['id' => $containerId,'project\_id'=>$project\_id])->render();  
 return $this->successResponse('success', ['html' => $html]);  
 }  
 */\*\*  
 \* Project Manager Request for Developer Request by using send button.  
 \*  
 \** ***@param*** *\Illuminate\Http\Request $request  
 \** ***@param*** *int $id  
 \** ***@return*** *\Illuminate\Http\Response  
 \*/* public function confirmDevelopersRequest(Request $request)  
 {  
 $project = Project::*where*('id',$request->project\_id)->first();  
 $project->number\_of\_developers = $request->no\_of\_developers;  
 $project->start\_date = Carbon::*parse*($request->start\_date);  
 $project->end\_date = Carbon::*parse*($request->end\_date);  
 $project->save();  
 $user\_id = $this->getAuthUserId();  
 $project\_lists = Project::*where*('user\_id',$user\_id)->get();  
 $html = view('pages.projectManager.\_partial.\_assign\_project\_list\_table\_html', ['project\_lists' => $project\_lists])->render();  
 return $this->successResponse('success',['html' => $html, 'html\_section\_id' => 'project-technology-stack-section']);  
 }

User Controller

*/\*\*  
 \* Display popup to add project By Admin.  
 \*  
 \** ***@param*** *\Illuminate\Http\Request $request  
 \** ***@param*** *int $id  
 \** ***@return*** *\Illuminate\Http\Response  
 \*/* public function addProjectModal(Request $request)  
 {  
 return $this->sendJsonResponse($this->projectManagerService->addProjectModal($request));  
 }  
 */\*\*  
 \* Click Add to add project in the project list.  
 \*  
 \** ***@param*** *\Illuminate\Http\Request $request  
 \** ***@param*** *int $id  
 \** ***@return*** *\Illuminate\Http\Response  
 \*/* public function confirmAddProjectModal(Request $request)  
 {  
 return $this->sendJsonResponse($this->projectManagerService->confirmAddProjectModal($request));  
 }  
 */\*\*  
 \* Display popup to edit Project Attributes.  
 \*  
 \** ***@param*** *\Illuminate\Http\Request $request  
 \** ***@param*** *int $id  
 \** ***@return*** *\Illuminate\Http\Response  
 \*/* public function editProjectModal(Request $request)  
 {  
 return $this->sendJsonResponse($this->projectManagerService->editProjectModal($request));  
 }  
 */\*\*  
 \* Click Update Button to edit Project Attributes.  
 \*  
 \** ***@param*** *\Illuminate\Http\Request $request  
 \** ***@param*** *int $id  
 \** ***@return*** *\Illuminate\Http\Response  
 \*/* public function confirmEditProjectModal(Request $request)  
 {  
 return $this->sendJsonResponse($this->projectManagerService->confirmEditProjectModal($request));  
 }  
 */\*\*  
 \* Display Popup to delete Project form DB.  
 \*  
 \** ***@param*** *\Illuminate\Http\Request $request  
 \** ***@param*** *int $id  
 \** ***@return*** *\Illuminate\Http\Response  
 \*/* public function deleteProjectModal(Request $request)  
 {  
 return $this->sendJsonResponse($this->projectManagerService->deleteProjectModal($request));  
 }  
 */\*\*  
 \* Click Delete button to delete project from DB.  
 \*  
 \** ***@param*** *\Illuminate\Http\Request $request  
 \** ***@param*** *int $id  
 \** ***@return*** *\Illuminate\Http\Response  
 \*/* public function confirmDeleteProjectModal(Request $request)  
 {  
 return $this->sendJsonResponse($this->projectManagerService->deleteProjectModal($request));  
 }  
 */\*\*  
 \* Display Assigning Project List to Project Manager.  
 \*  
 \** ***@param*** *\Illuminate\Http\Request $request  
 \** ***@param*** *int $id  
 \** ***@return*** *\Illuminate\Http\Response  
 \*/* public function index(Request $request)  
 {  
 $user\_id = $this->getAuthUserId();  
 $project\_lists = Project::*with*('technologystack')->where('user\_id',$user\_id)->get();  
 return view('pages.projectManager.projectManagers')->with('project\_lists', $project\_lists);  
// return $this->sendJsonResponse($this->projectManagerService->index($request));  
 }  
 */\*\*  
 \* Display Developer Request Modal.  
 \*  
 \** ***@param*** *\Illuminate\Http\Request $request  
 \** ***@param*** *int $id  
 \** ***@return*** *\Illuminate\Http\Response  
 \*/* public function developersRequestModal(Request $request)  
 {  
 return $this->sendJsonResponse($this->projectManagerService->developersRequestModal($request));  
 }  
 */\*\*  
 \* Project Manager Request for Developer Request by using send button.  
 \*  
 \** ***@param*** *\Illuminate\Http\Request $request  
 \** ***@param*** *int $id  
 \** ***@return*** *\Illuminate\Http\Response  
 \*/* public function confirmDevelopersRequest(Request $request)  
 {  
 $validator = Validator::make($request->all(), [  
 'no\_of\_developers' => 'required|numeric',  
 'start\_date' => 'required|date',  
 'end\_date' => 'date|after:start\_date', // |unique:users,email  
 ]);  
 if ($validator->fails()) {  
 return $this->error('Validation Failed', ['errors' => $validator->errors()]);  
 }  
 return $this->sendJsonResponse($this->projectManagerService->confirmDevelopersRequest($request));  
 }