**Todo List**

I have created a todo list app using laravel 8, bootstarp and javascript. This is a single page application which lets every user to create, delete, edit and control access rights which means only those user will be able to see tasks who has been given the permission to see any task.

**Controllers**

To make this app I have made to 2 controllers HomeController and TodoController. HomeController is responsible to load the home page with all visible tasks.

TodoController is responsible for containing the logic for creating, updating and deleting tasks. I have used to conventional naming for the method inside the controllers. So the store method has been used to make a new task, update method has been used to change any old task and destroy method has been used to make a delete any task.

**Routing**

I have used Laravel’s Auth::routes() for basic authentications. All the http requests including requests made with ajax has to pass through the auth middleware except for login request.

**Models**

To make the app I need only two models one for the users the User model and another for the tasks which called Todo.

The Todo model has three methods one is *user()* method which links the Todo model this User model using belongsTo relation since each task has only one user or can be created by one user so one-to-many relation will be applied here.

*Second* method is *permittedUsers()* which returns all the users that are permitted to access that particular task. It uses the pivot table todo\_user which is used to store the task access permission. todo\_user has three column id, user\_id, task\_id. User\_id column hold the id of the user who has the right to view that particular task that is being pointed by the task\_id on that specific row.

*Finally* the visibilitytag() method is used to return label for each task which are “public”, “only me” and ‘custom”.

**Relationships**

There are two types of relationships which has been used in the app. First is one-to-many and second is many-to-many relationship.

The one-to-many relationship is used because one user can have multiple task but each task can have only one creator therefore one-to-many relation.

The many-to-many relation is needed to track the view right of each task. Since each task can be viewed by many users and each user can have right to view many tasks therefore to achieve this we need a many-to-many relationship. For many-to-many relationship we need a pivot table, I named the table todo\_user. It contains the three column id, user\_id, todo\_id.

**Gates**

I have used one gate name “*modify*” which detects if any user has the authority to update and delete the any task. The gate has has been declared in the boot method of the AuthServiceProvider.

**How the App Works**