Technical documentation

The purpose of this document is generic Technical document for the use of Task Management web application.

The Application features:

- view the list of existing tasks separated in categories.

- view each task details.

- edit task details: name, description, priority, category and dueDate.

- view list of subtasks of a task.

- edit subtask's name and check/uncheck.

Frameworks:

In order to build the single web application Angular framework has been used, which is JavaScript-based open-source front-end web application framework.

Bootstrap also has been used, which is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.

Data Structure:

The data comes from an existing API in format of json object which contains arrays like: tasks, subtasks , category and users. Using some services and requests to GET and PATCH from and to the server.

Project Structure:

The Model View Controller (MVC) software design pattern has been implemented for the project structure.

A Model View Controller pattern is made up of the following three parts:

- Model − It is the lowest level of the pattern responsible for maintaining data.
- View − It is responsible for displaying all or a portion of the data to the user.
- Controller − It is a software Code that controls the interactions between the Model and View.

The model part in our project exists in the services which get and post data for the server which can be found under this directory: ./public/services/taskManagerServices.js.
One example of getting data :

```
app.service('getOneTask',['$resource','baseURL',function($resource,baseURL){
    this.getFeedback=function(id){
      return $resource(baseURL+"tasks/"+id,null,{
        'update':{
          method:'GET'
        }
      });
    };
}])
```

This service is getting one task by requesting get method from the baseURL which is in our case http://localhost:3000//tasks/{taskId}

The view part exists in the html files.

The controller part which conatis the functions and the logic of the tasks manager can be found under this directory: ./public/tasksView/taskManagerController.js.

User Interaction:

The application start at the login page which contains two inputs fields for username and password which are hardcoded for the user which has id equal 1.
Then after clicking on the login button it will direct the user to the tasks view page.

The table below shows the tasks separated by category type colored by green for example Personal, and you can expand and collapse the category to show tasks belong to it.
The tasks are sorted by their priority (high, medium, low) and you can filter each column by typing in the input field at each column's header.

Category	Name	Description	Due Date	Priority
No category (1)				
Personal (3)				
Personal	Shopping			high
Personal	Cooking			low
Personal	Cleaning			medium
Work (1)				

Each row is a task and you can click on the task row to show task details and subtasks which you can also update tasks and subtasks as shown below: