# Angular Fundamentals - Homework

1. Setup a project using the Angular CLI
2. Create a module – posts
3. Create the specific components in that module “posts” so you have a working CRUD (create, read, update, delete post entity) :

* **Index** – List all the resources in a table with heading (id, title, description, etc.). The last column should be named “actions” and there you should have 3 buttons – Details, Edit, Delete. "Details” leads to the view component page, “Edit” leads to the edit component page. “Delete” triggers the window.confirm alert and if “OK” is clicked then you send a request to the API with DELETE method and remove the deleted row from the table.
* **View** – The detailed view of a single resource, you go to this page by clicking the id or the button “details” of the resource row from index table. This page should show only the data for the clicked post, each data on new line:

*Id: 2*

*Title: Neshto si*

*Description: Neshto si opisanie*

* Edit – Page with a form (same as add post component form) but with prefilled data coming from the server, ex. <http://api.com/posts/2> you get the data for the post and show it in the form. The form buttons should be "Cancel" (redirect to index) and "Update" (update the resource data and go back to index).

Above the index table you should have a button “Add post”, when you click it you are redirected to the add-post component with a form and submit button. If form is valid and submit is triggered, you are redirected back to the index table.

Use a fake API to fetch data from server, ex. <https://jsonplaceholder.typicode.com/> and documentation - <https://github.com/typicode/jsonplaceholder#how-to>

**Note: Read and use the** [**https://angular.io/guide/reactive-forms**](https://angular.io/guide/reactive-forms) **as they are used on almost every project**

**Note: Feel free to use whatever public API you want. The idea is to be able to create and fetch resources**

1. Create a component “home” outside the posts module and in its view show the link to posts index (/posts). Initially and by default when you run the app, the home component should show.

**Extra mile – Bonus points**

1. Integrate material design in Angular and use the styles from there for the forms / tables and etc. ( <https://material.angular.io/> )
2. Create a component “loader” which shows and hides the text “Loading...” on the top of the page. Load that component in the app and show it before the request to the server is being sent and hide it when the request to the server is done.