## Angular

* Ng new app-name
* Adding css links
* Ng serve – to run the app
* Define route: app-routing.module.ts is the route of the app. So any component that needs to be displayed should be mentioned in this master file
* Or you can just use the selector of that component to specifically mention where to use it

Text

Description automatically generated

* Ng generate component component\_name to create component. Different html pages are called component
* Any redirections among the component pages are linked in ”app-routing.module.ts”
* There, the path is imported and the path name is given – ‘/new-list’

### Service

* Ng generate service service\_name.
  + Service will provide the data the components need
  + To implement any logic. Like functions that can be reused among components.
  + External interaction ( angular to interact with the express)
  + Extension – file\_name.service.ts
* Create a service called ‘WebRequest’ that handles the web request logics and declare the API URL we are going to use ( basically linking the API with Angular). Declare all the methods (GET, POST, PATCH, DELETE) we are going to use with that URL
* Visit web-request.service.ts for all the server request logics
* Next visit task.service.ts
* Next visit new-list.component.ts
* Next visit new-list.component.html
* Next visit app.module.ts
  + First create service to handle http request – “web-request.service.ts”
  + Next create service to do the function
  + This function is linked with the html using component.ts and not the component.html itself

## CSS

* Display: flex container expands items to fill available free space or shrinks them to prevent overflow.
* Use justify-content to align along the x axis. Use align-items to align along the y axis.
* Width will only go full on any screen. Max width will adapt to screen size
* Flex direction: which direction to fill
* Flex-grow: if previous flex is specified, this remaining flex will fill accordingly
* cursor: pointer; to change cursor as clickable

## Express

* npm init
* npm I express –save
* npm install -g nodemon
* nodemon app.js will automatically run and refresh the server
* npm i body-parser --save - to use the middleware body-parser during req.body

## Mongoose

* npm install mongoose –save