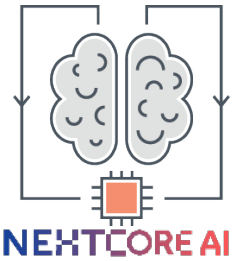
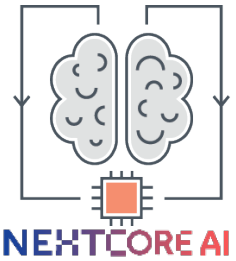


Angular Material

- How to configure Angular Material
- Upgrade the UX with Angular Material



Add Angular Material Automatically

1. Execute the add command, as shown.

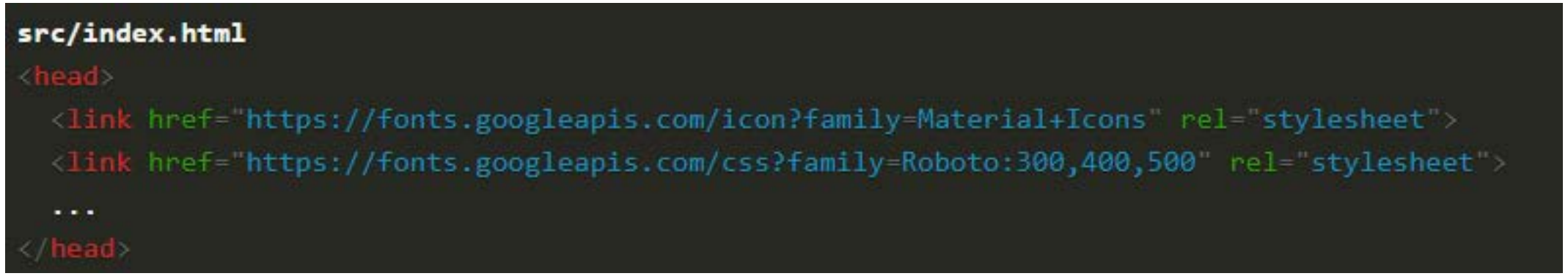
```
$ npx ng add @angular/material
Installing packages for tooling via npm.

+ @angular/material@6.0.1
added 1 package in 15.644s
Installed packages for tooling via npm.
UPDATE package.json (1381 bytes)
UPDATE angular.json (3694 bytes)
UPDATE src/app/app.module.ts (502 bytes)
UPDATE src/index.html (474 bytes)
UPDATE node_modules/@angular/material/prebuilt-themes/indigo-pink.css (56678 bytes)
added 1 package in 13.031s
```

Add Angular Material

- Note that index.html has been modified to add icons library/fontCountry

src/index.html

```
<head>
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
  <link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500" rel="stylesheet">
  ...
</head>
```



Automatic Material

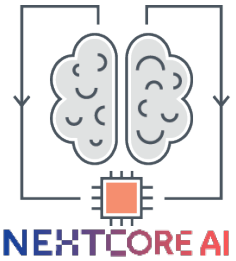
- app.module.ts has been updated to import BrowserAnimationsModule, as demonstrated:

src/app/app.module.ts

```
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    ...
    BrowserAnimationsModule
  ],
```

2. Start your app and ensure that it works correctly:

```
$ npm start
```

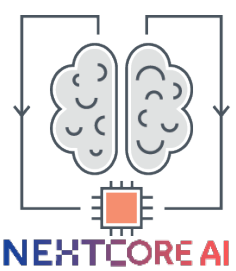


Manually Install Material

1. In the terminal, execute `npm install @angular/material @angular/cdk @angular/animations hammerjs`
2. Observe package.json versions:

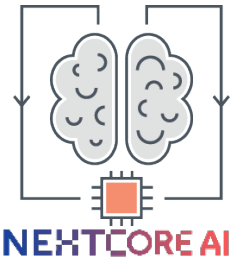
package.json

```
 "dependencies": {
    "@angular/animations": "6.0.0",
    "@angular/cdk": "6.0.0",
    "@angular/material": "6.0.0",
    "hammerjs": "^2.0.8",
    ...
```



Material's components

- `@angular/material` is the official Material 2 library.
- `@angular/cdk` is a peer-dependency, not something you directly use unless you intend to build your own components.
- `@angular/animations` enables some of the animations for some Material 2 modules. It can be omitted to keep app size minimal. You may use `NoopAnimationsModule` to disable animations in the modules that require this dependency. As a result, you will lose some of the UX benefits of Angular Material.
- `hammerjs` enables gesture support; it's critical if you're targeting any touch-enabled device, not just phones and tablets, but also hybrid-laptops.



Manually configuring Angular Material

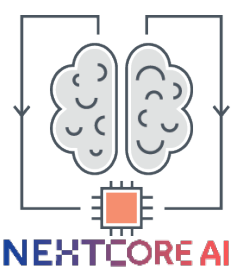
- Now that the dependencies are installed, let's configure Angular Material in our Angular app. Note that if you used `ng add @angular/material` to install Angular Material, some of this work will be done for you.

Importing modules

- We will start by creating a separate module file to house all our Material module imports:

1. Execute the following command in the terminal to generate `material.module.ts`:

```
$ npx ng g m material --flat -m app
```

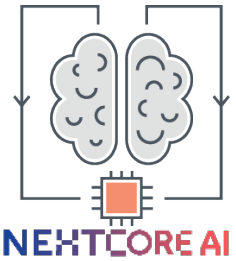


Importing modules

2. Observe the newly created file material.module.ts

src/app/material.module.ts

```
import { NgModule } from '@angular/core'
import { CommonModule } from '@angular/common'

@NgModule({
  imports: [CommonModule],
  declarations: [],
})
export class MaterialModule {}
```

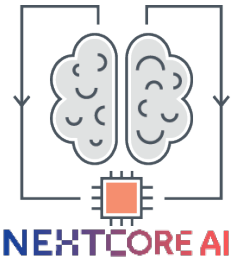


Importing Modules

3. Ensure that the module has been imported into app.module.ts:

src/app/app.module.ts

```
import { MaterialModule } from './material.module'
...
@NgModule({
  ...
    imports: [..., MaterialModule],
}
```

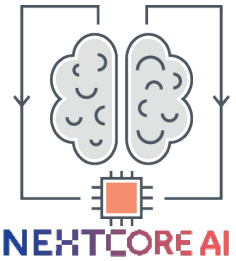


Importing Modules

4. Add animations and gesture support (optional, but necessary for mobile device support):

src/app/app.module.ts

```
import 'hammerjs'
import { BrowserAnimationsModule } from '@angular/platform-browser/animations'

@NgModule({
  ...
  imports: [..., MaterialModule, BrowserAnimationsModule],
}
```

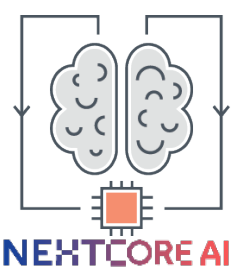


Importing Modules

5. Modify material.module.ts to import basic components for Button, Toolbar, and Icon
6. Remove CommonModule:

src/app/material.module.ts

```
import { MatButtonModule, MatToolbarModule, MatIconModule } from
'@angular/material'
import { NgModule } from '@angular/core'

@NgModule({
  imports: [MatButtonModule, MatToolbarModule, MatIconModule],
  exports: [MatButtonModule, MatToolbarModule, MatIconModule],
})
export class MaterialModule {}
```



Importing Base Theme

A base theme is necessary in order to use Material components. We can define or change the default theme in angular.json:

```
angular.json
...
"styles": [
  {
    "input": "node_modules/@angular/material/prebuilt-themes/indigo-pink.css"
  },
  "src/styles.css"
],
...
```

1. Choose a new option from here:

```
deeppurple-amber.css
indigo-pink.css
pink-bluegrey.css
purple-green.css
```

2. Update angular.json to use the new Material theme

Material Icon font

https://www.google.com/design/icons/

Material Icon font- You can get access to a good default set of iconography by adding the **Material Icon** web font to your application. Clocking in at 48 kb in size, this is a very lightweight library.

src/index.html

```
<head>
  ...
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
</head>
```


```
src/index.html
<head>
  ...
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
</head>
```