1. Install Angular Cli

sudo npm install -g @angular/cli

2. Create your application

ng new myKeepApp

3. serve the application

cd myKeepApp

ng serve –open

4. to create components like header and note ..

cd src/app/

ng generate component header

```
jp@jp-ThinkPad-L450:~/Workspace/myKeepApp/src/app$ ng generate component header
  create src/app/header/header.component.css (0 bytes)
  create src/app/header/header.component.html (25 bytes)
  create src/app/header/header.component.spec.ts (628 bytes)
  create src/app/header/header.component.ts (269 bytes)
  update src/app/app.module.ts (398 bytes)
```

the above command had automatically included the below line into app.module.ts

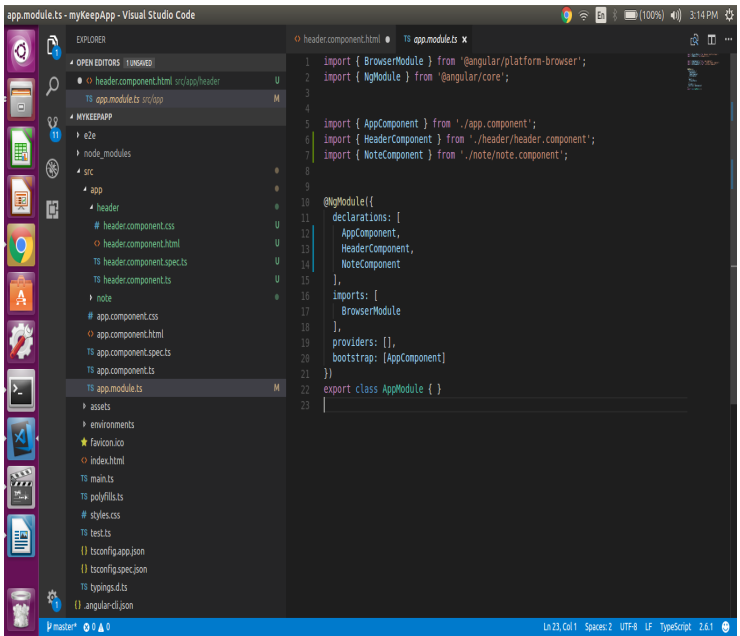
```
import { HeaderComponent } from './header/header.component';
```

5 let us try to create another component note

ng g c note

this will also inclue the below line automatically in app.module.ts

```
import { NoteComponent } from './note/note.component';
```



6. in app.component.html file.. you can comment exiting code and just add
<app-header></app-header>
<app-note></app-note>

save it and then run the application by giving the below command

ng server –open

7. now let us use material ui to make this ui look much better

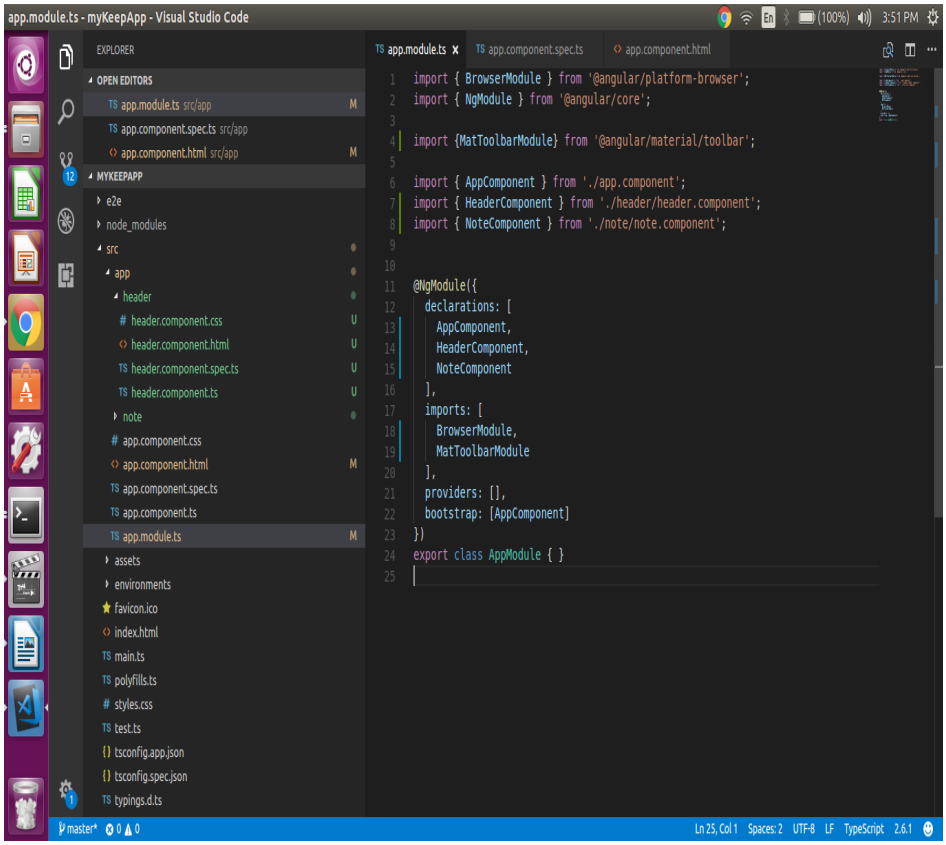
go to https://material.angular.io/ and browse for various options available. To use material ui in your application.. type the below command in terminal

npm install --save @angular/material @angular/cdk

8. <mat-toolbar> is a container for headers, titles, or actions. To include that in our app.. copy the below line to app.component.ts file

import {MatToolbarModule} from '@angular/material/toolbar';

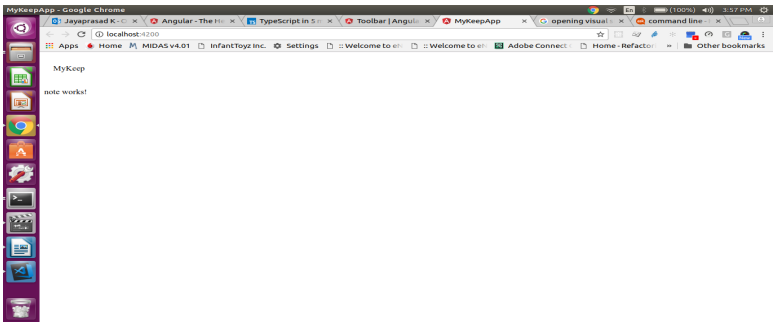
and also add “MatToolbarModule ” in imports array..



9. to make this tool bar to be part of the header component.. goto header.component.html file and add the below code

<mat-toolbar>MyKeep</mat-toolbar>

you can see the below output. The header appears normal without any style



to get the styles.. install angular animations by using the below command

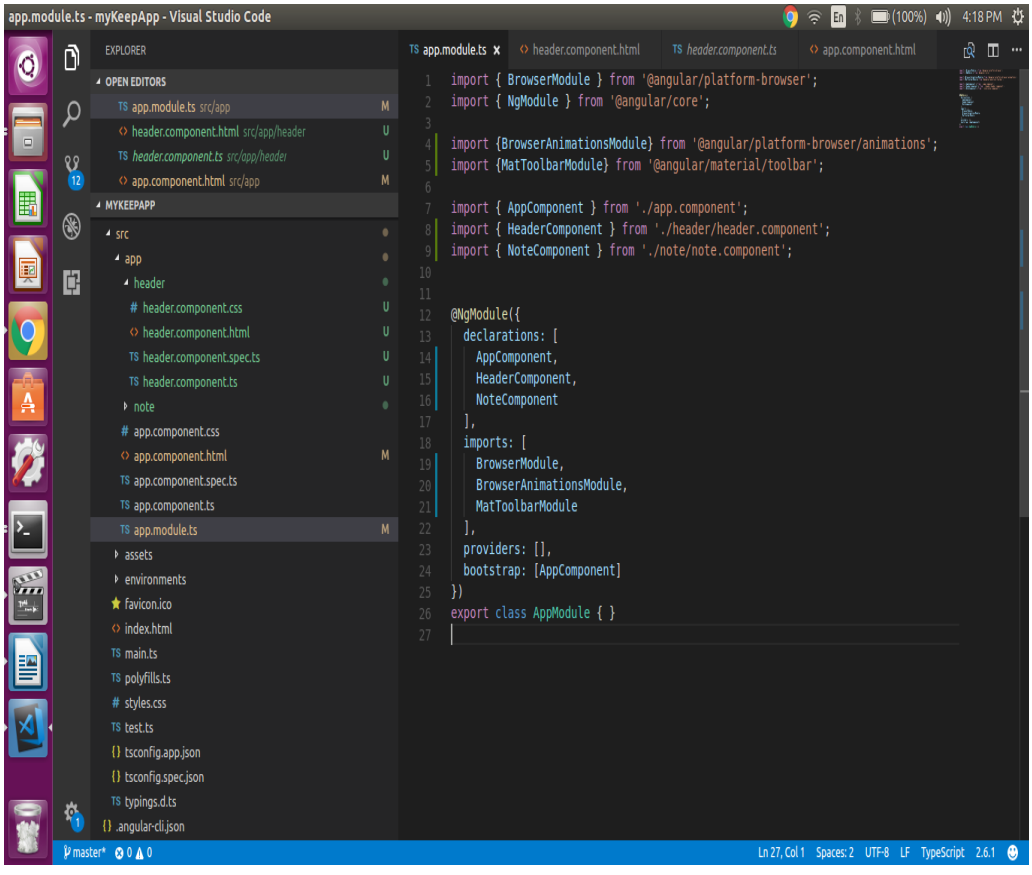
Some Material components depend on the Angular animations module in order to be able to do more advanced transitions. If you want these animations to work in your app, you have to install the @angular/animations module and include the BrowserAnimationsModule in your app.

npm install --save @angular/animations

copy the below command in app.module.ts

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

and then add “BrowserAnimationsModule” to the imports array as shown below:

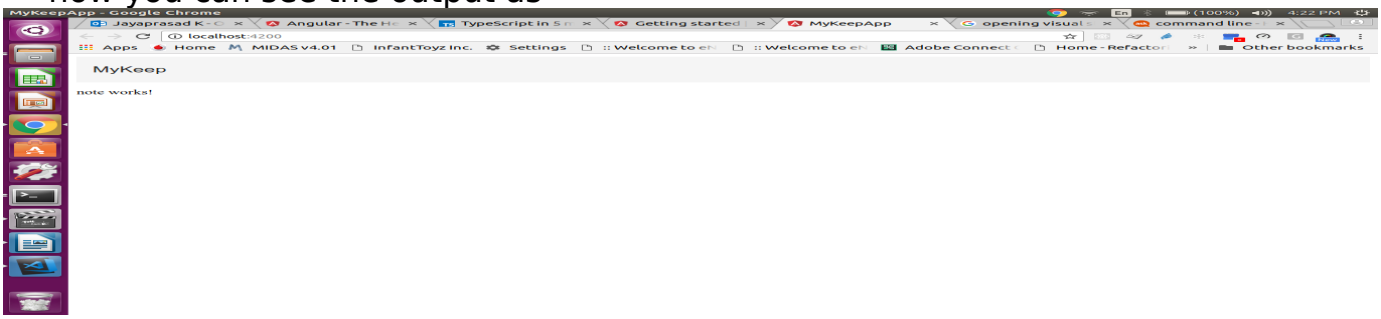


Including a theme is required to apply all of the core and theme styles to your application.

To get started with a prebuilt theme, include one of Angular Material's prebuilt themes globally in your application. If you're using the Angular CLI, you can add this to your styles.css:

```
@import "~@angular/material/prebuilt-themes/indigo-pink.css";
```

now you can see the output as



to change the colour of the header.. in your header.component.html file..

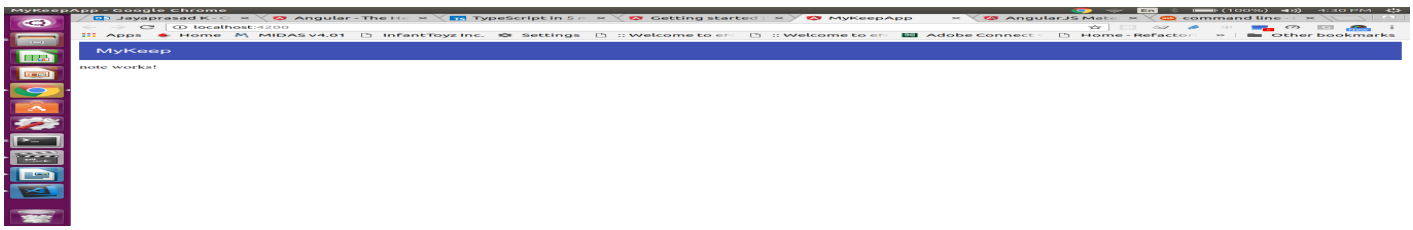
change the existing line of code

```
<mat-toolbar>MyKeep</mat-toolbar>
```

as

```
<mat-toolbar color="primary">MyKeep</mat-toolbar>
```

now the output is



10. To eliminate the margins above the header.. in styles.css file.. include the below code

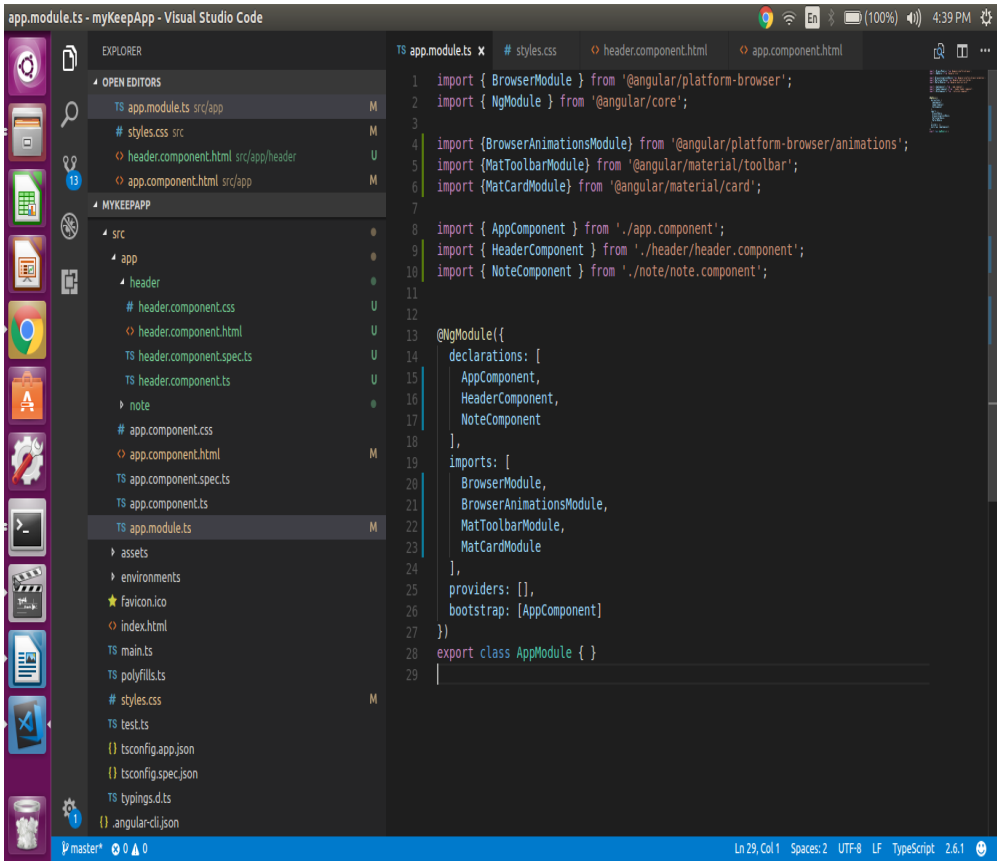
```
body {  margin : 0;}
```

11.Now let us add card from material ui for note component

copy and paste the below line in app.component.ts

import {MatCardModule} from '@angular/material/card';

and add “MatCardModule” in imports section as given below



12. editing note.component.html file ...

delete the exiting code and paste the below code to see the card

```
<mat-card>
<mat-card-title>MyNotes</mat-card-title>
<mat-card-content>This is my first notes</mat-card-content>
</mat-card>
```

13, let us style the notes

in the note.component.css file type the below code

```
:host{
 display: “flex”;
 padding: “1em”;
}
```

https://angular.io/guide/component-styles

Special selectors

Component styles have a few special selectors from the world of shadow DOM style scoping (described in the CSS Scoping Module Level 1 page on the W3C site). The following sections describe these selectors.

:host

Use the :host pseudo-class selector to target styles in the element that hosts the component (as opposed to targeting elements inside the component's template).

```
content_copy:host {
    display: block;
    border: 1px solid black;
}
```

The :host selector is the only way to target the host element. You can't reach the host element from inside the component with other selectors because it's not part of the component's own template. The host element is in a parent component's template.

Use the function form to apply host styles conditionally by including another selector inside parentheses after :host.

The next example targets the host element again, but only when it also has the active CSS class.

```
content_copy:host(.active) {
    border-width: 3px;
}
```

Let us work on the note.component to make it much more better

https://material.angular.io/components/expansion/api

let us copy the below code into app.module.ts

import {MatExpansionModule} from '@angular/material/expansion';

and add "MatExpansionModule" to imports sections

in the app.component.html file comment the <app-note></app-note> tag and put the below code

```
<app-header></app-header>
<!-- <app-note></app-note> -->
<mat-expansion-panel>
  <mat-expansion-panel-header>
     <mat-panel-title>
        This is my Note
      </mat-panel-title>
  </mat-expansion-panel-header>

</mat-expansion-panel>
```

Sofar we were using static text in the notes.. now it is time to get the input from the user.. hence let us use form fields

https://material.angular.io/components/form-field/api

copy and paste the below line to app.module.ts

```
import {MatFormFieldModule} from '@angular/material/form-field';
```

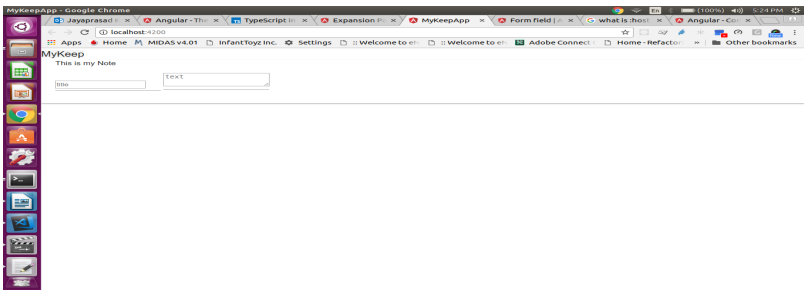
and add "MatFormFieldModule" to imports sections

_____-

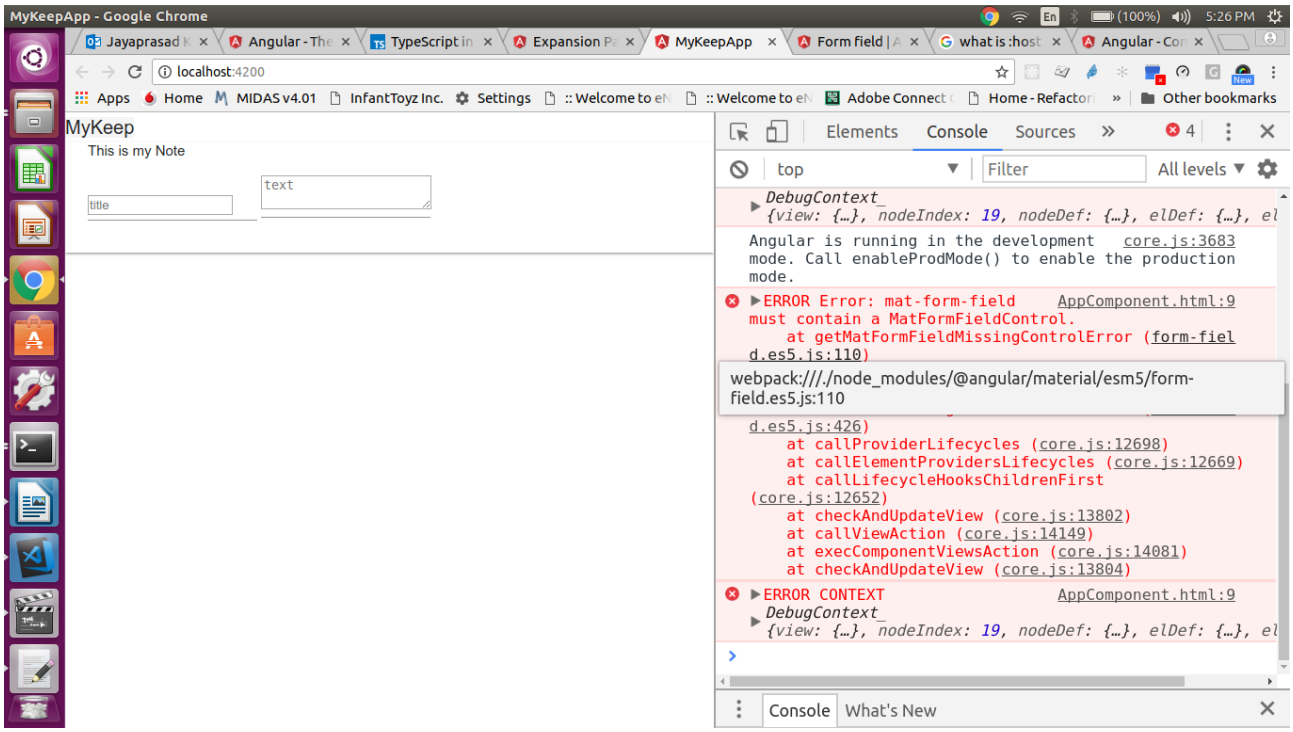
in app.component.html file.. type the below code

```
<app-header></app-header>
<!-- <app-note></app-note> -->
<mat-expansion-panel>
  <mat-expansion-panel-header>
     <mat-panel-title>
        This is my Note
      </mat-panel-title>
  </mat-expansion-panel-header>
  <mat-form-field>
    <input  placeholder="title" type="text">
  </mat-form-field>
  <mat-form-field>
    <textarea placeholder="text"></textarea>
  </mat-form-field>
</mat-expansion-panel>
```

the output will be

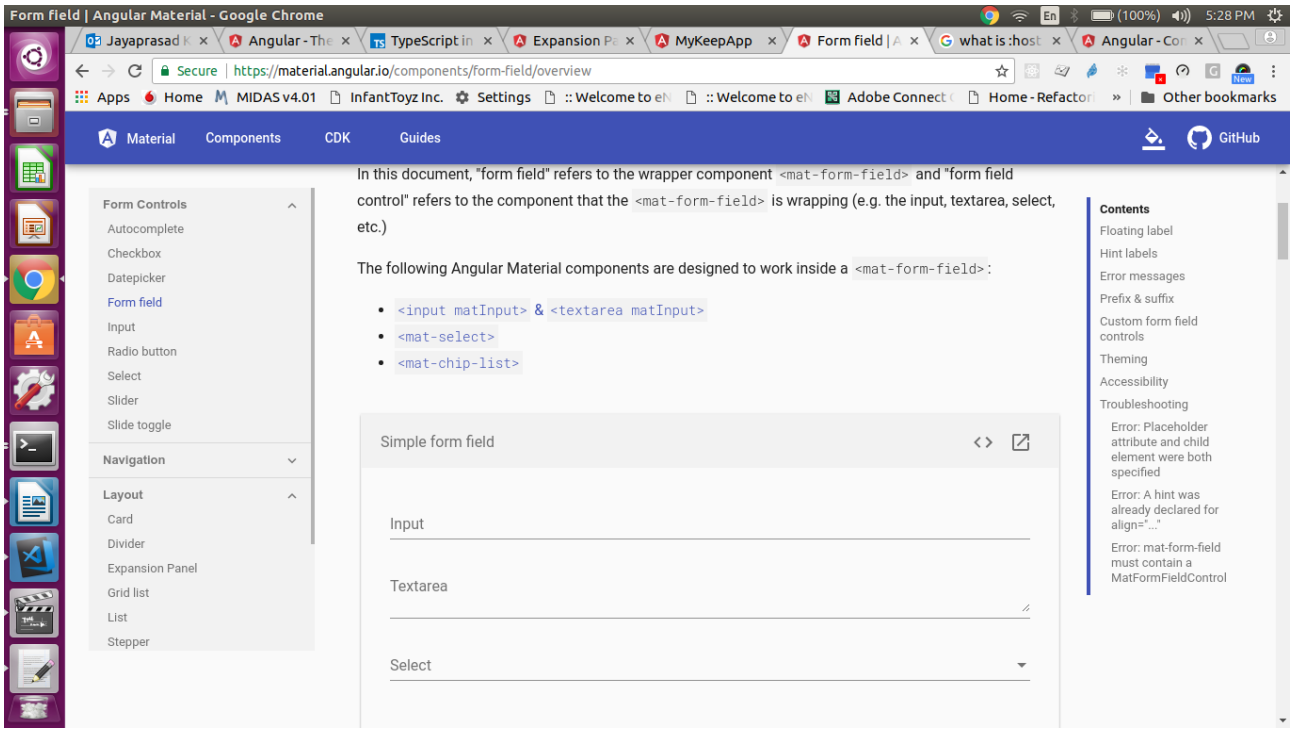


when you inspect you can see some error



https://material.angular.io/components/form-field/overview

the above link has the error description in the bottom lower corner



Error: mat-form-field must contain a MatFormFieldControl

This error occurs when you have not added a form field control to your form field. If your form field contains a native <input> or <textarea> element, make sure you've added the

matInput directive to it and have imported MatInputModule. Other components that can act as a form field control include <mat-select>, <mat-chip-list>, and any custom form field controls you've created.

Hence you need to add the below entry in your app.module.ts

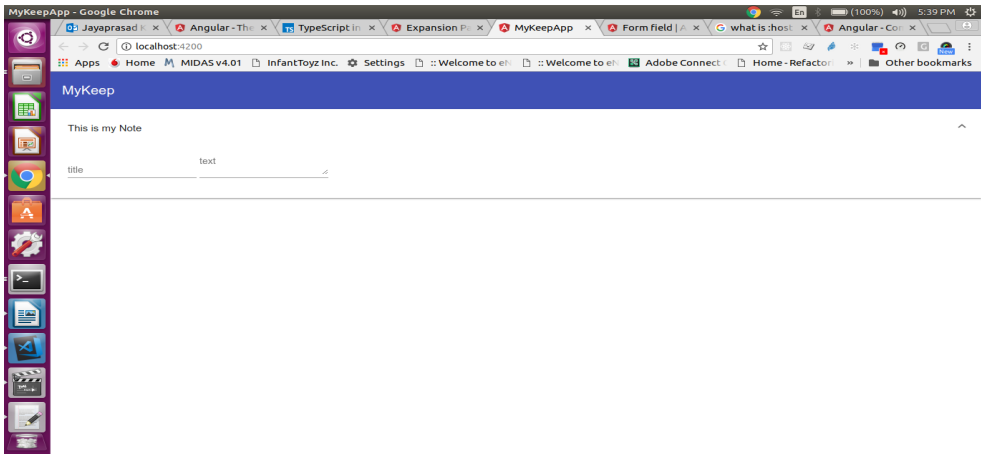
import {MatInputModule} from '@angular/material/input';

and the add “MatInputModule ” to imports sections

in the app.component.html you must add “mapInput” property to the <input> & <textaread> tags as given below :

```
<app-header></app-header>
<!-- <app-note></app-note> -->
<mat-expansion-panel>
  <mat-expansion-panel-header>
     <mat-panel-title>
        This is my Note
      </mat-panel-title>
  </mat-expansion-panel-header>
  <mat-form-field>
    <input matInput placeholder="title" type="text">
  </mat-form-field>
  <mat-form-field>
    <textarea matInput placeholder="text"></textarea>
  </mat-form-field>
</mat-expansion-panel>
```

the output will be like below:



The controls are coming adjacent to each other.. we need to change them to come one below the anther. Inorder to do this .. let us put these controls in div element.. give a class name to it and then apply style.

app.component.html

```
<app-header></app-header>
<!-- <app-note></app-note> -->
<div class="keep-comp-note-taker-panel">
   <mat-expansion-panel>
      <mat-expansion-panel-header>
         <mat-panel-title>
            This is my Note
          </mat-panel-title>
      </mat-expansion-panel-header>
      <div class="keep-comp-note-taker-wrapper">
         <mat-form-field>
            <input matInput placeholder="title" type="text">
          </mat-form-field>
          <mat-form-field>
            <textarea matInput placeholder="text"></textarea>
          </mat-form-field>
      </div>

     </mat-expansion-panel>
</div>
```

app.component.css

```
.keep-comp-note-taker-panel{
   width: 50vw;
   margin : auto;
}
.keep-comp-note-taker-wrapper{
   display: flex;
   flex-direction: column;
}
```

---

to add button now.. in app.module.ts add below

https://material.angular.io/components/button/api

import {MatButtonModule} from '@angular/material/button';

and then add "MatButtonModule" in imports section

now the app.component.html is

```
<app-header></app-header>
<!-- <app-note></app-note> -->
<div class="keep-comp-note-taker-panel">
   <mat-expansion-panel>
```

```
        <mat-expansion-panel-header>
            <mat-panel-title>
                This is my Note
            </mat-panel-title>
        </mat-expansion-panel-header>
        <div class="keep-comp-note-taker-wrapper">
            <mat-form-field>
                <input matInput placeholder="title" type="text">
            </mat-form-field>
            <mat-form-field>
                <textarea matInput placeholder="text"></textarea>
            </mat-form-field>
            <div class="keep-comp-button">
                <button mat-button>Click me!</button>
            </div>
        </div>

    </mat-expansion-panel>
</div>
```

app.component.css

```
.keep-comp-note-taker-panel{
    width: 50vw;
    margin : auto;
}
.keep-comp-note-taker-wrapper{
    display: flex;
    flex-direction: column;
}
.keep-comp-button{
```

```
  display: flex;
  justify-content: flex-end;
}
```

with this we have completed the html design part. Now we should get on to the data extract part

---

in **app.module.ts** .. add

```
import { FormsModule } from '@angular/forms';
```

and then update imports section accordingly

Add below code to

app.component.ts

```
import { Component } from '@angular/core';

class Note{
  title:string;
  text: string;
  constructor(){
    this.title="";
    this.text="";
  }
}

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
```

```
  //title = 'app';

  note:Note = new Note();
  takeNotes(){
    console.log(this.note.title+" "+this.note.text);
  }
}
```

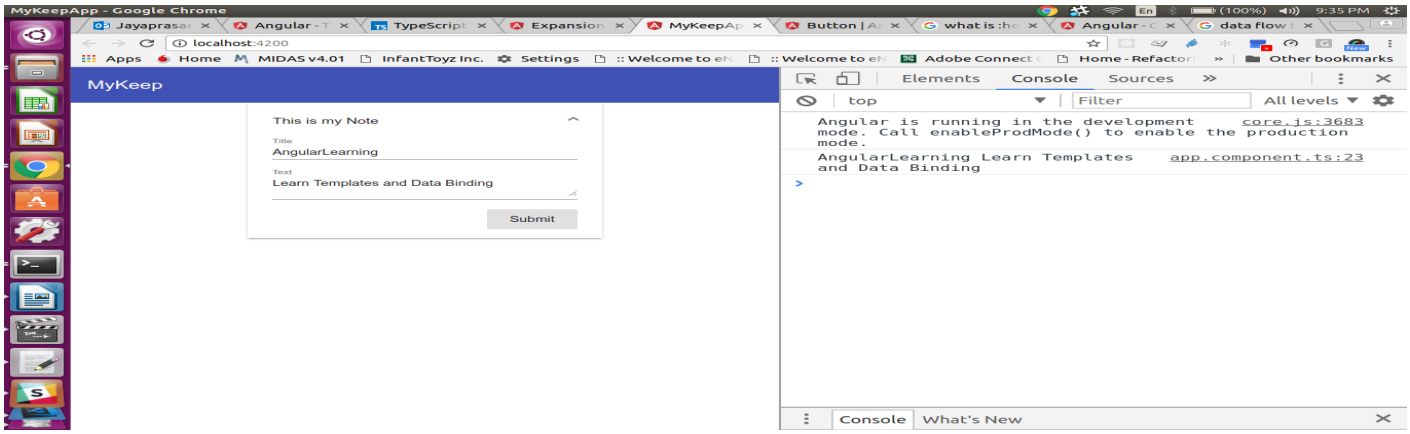
app.component.html

```
<app-header></app-header>
<!-- <app-note></app-note> -->
<div class="keep-comp-note-taker-panel">
    <mat-expansion-panel>
        <mat-expansion-panel-header>
            <mat-panel-title>
                This is my Note
            </mat-panel-title>
        </mat-expansion-panel-header>
        <div class="keep-comp-note-taker-wrapper">
            <mat-form-field>
                <input [(ngModel)]="note.title" matInput placeholder="Title" type="text">
            </mat-form-field>
            <mat-form-field>
                <textarea [(ngModel)]="note.text" matInput placeholder="Text"></textarea>
            </mat-form-field>
            <div class="keep-comp-button">
                <button mat-button (click)="takeNotes()"> Submit</button>
            </div>
        </div>
```

```
    </mat-expansion-panel>
</div>
```

---

the output is some thing like this..

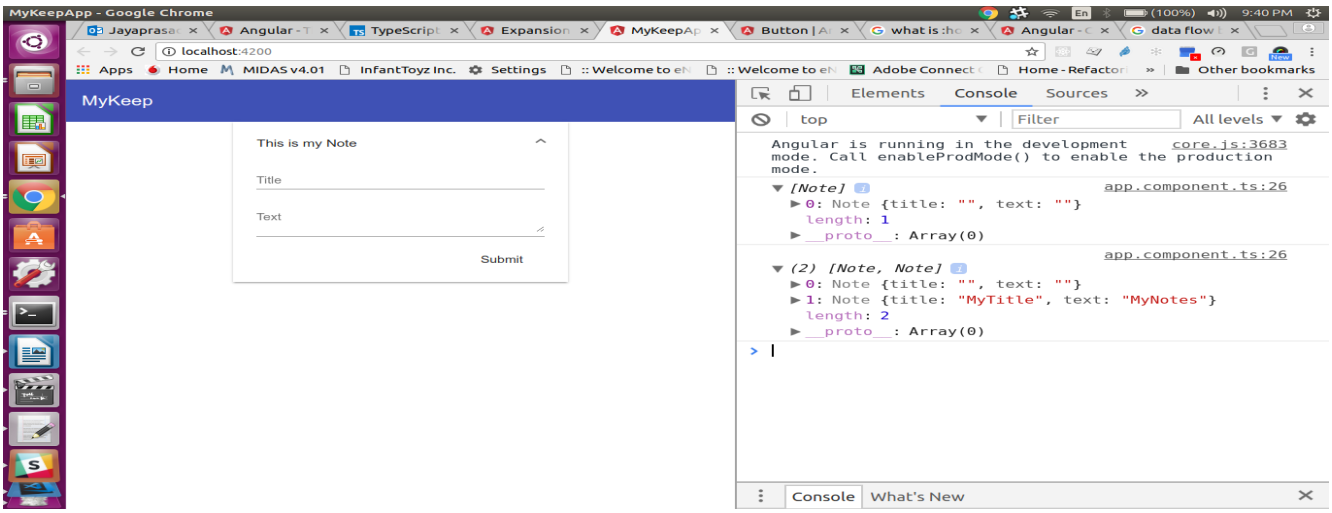


We need to have multiple notes hence we have to have a list of notes..

app.component.ts

```
import { Component } from '@angular/core';

class Note{
  title:string;
  text: string;
  constructor(){
    this.title="";
    this.text="";
  }
}
```

```
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  //title = 'app';

  note:Note = new Note();
  //declare an array
  notes:Array<Note> = [];

  takeNotes(){
  this.notes.push(this.note);
    console.log(this.notes);
    this.note=  new Note(); // creating a new note object for next entry
  //  console.log(this.note.title+" "+this.note.text);
  }
}
```



Let us try to show the same details in the html page

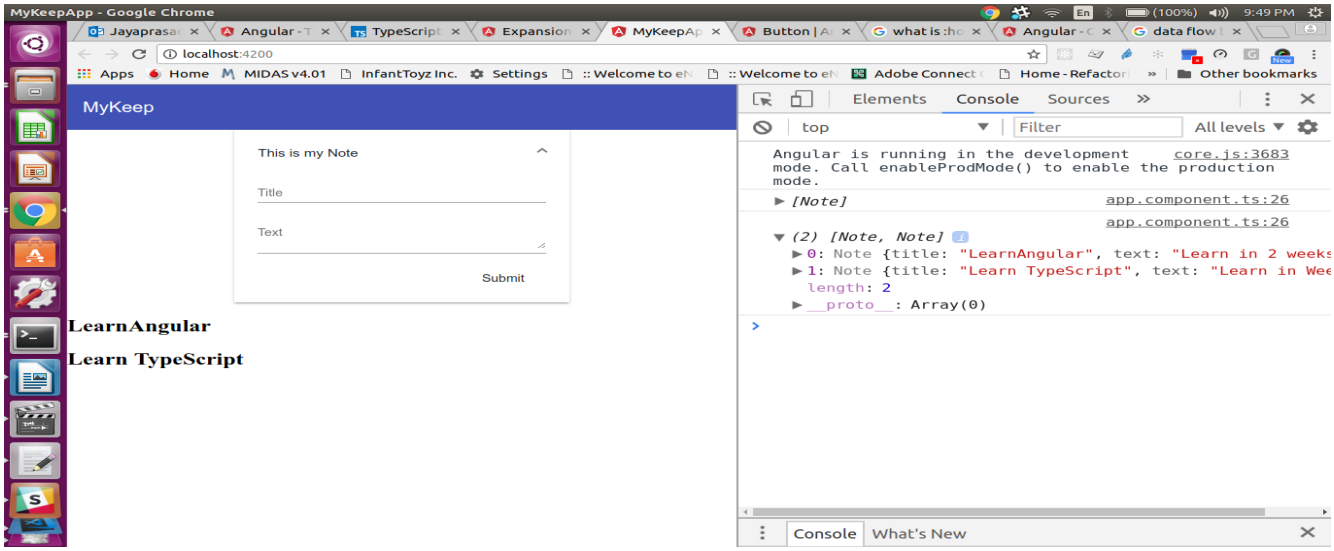
in your app.component.html add the below code snippet

<!-- Notes Container -->

<div *ngFor="let note of notes">

<h2>{{note.title}}</h2>

output is like



Let us make it much more better.. let us copy the code from note.component.html and modify it

```
<mat-card>
  <mat-card-title>MyNotes</mat-card-title>
  <mat-card-content>This is my first notes</mat-card-content>
</mat-card>
```

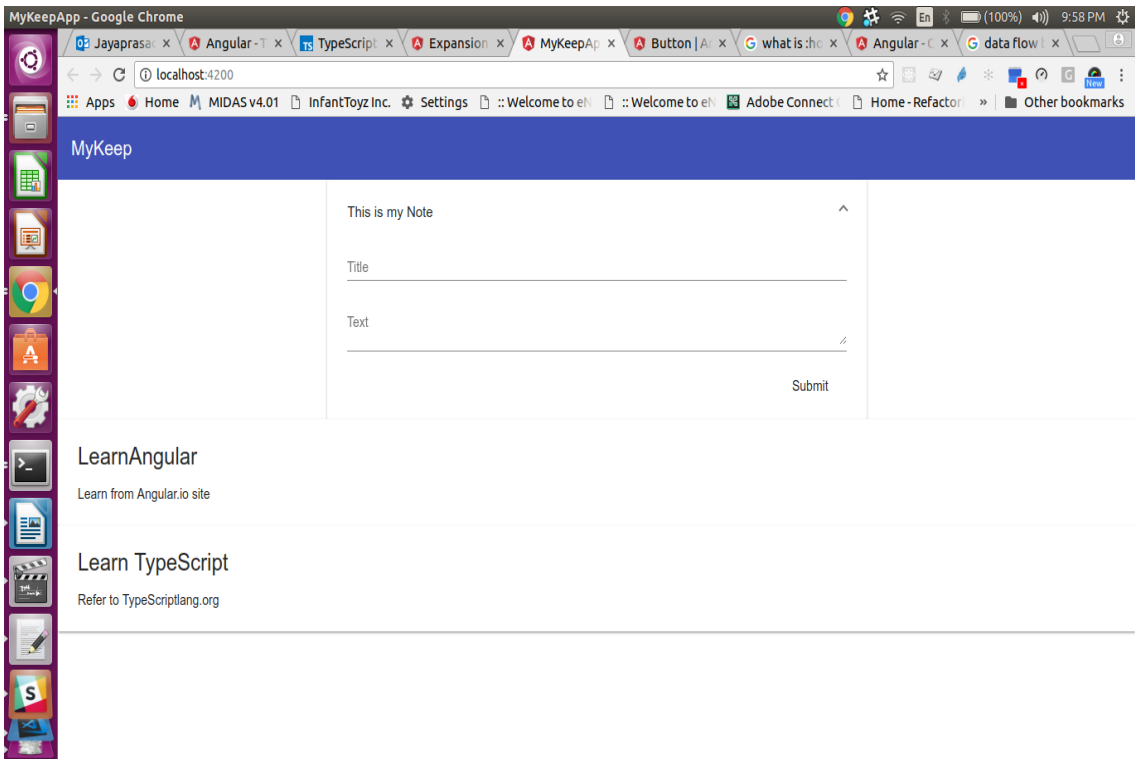
as ..

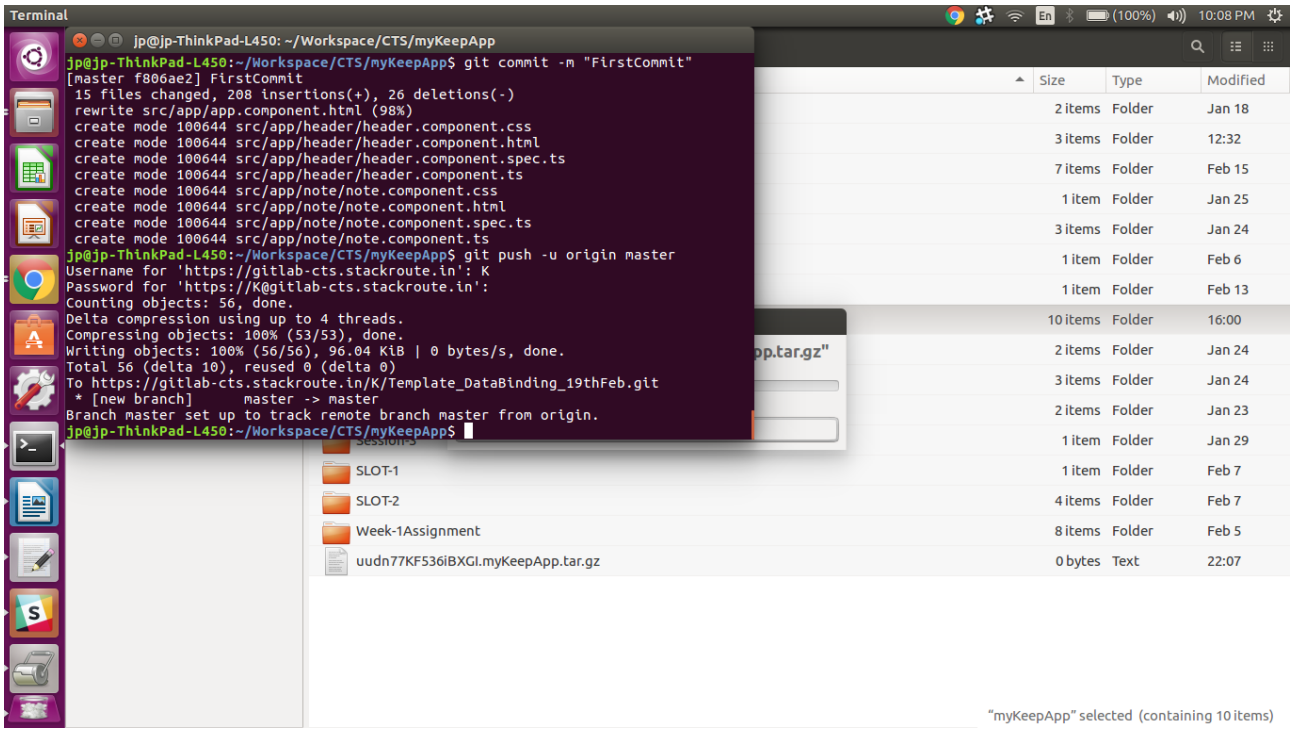
```
<mat-card  *ngFor="let note of notes">
  <mat-card-title>{{note.title}}</mat-card-title>
  <mat-card-content>{{note.text}}</mat-card-content>
</mat-card>
```

and place it in app.component.html

```
<app-header></app-header>
<!-- <app-note></app-note> -->

<!-- Note Taker -->
<div class="keep-comp-note-taker-panel">
    <mat-expansion-panel>
        <mat-expansion-panel-header>
            <mat-panel-title>
                This is my Note
              </mat-panel-title>
        </mat-expansion-panel-header>
        <div class="keep-comp-note-taker-wrapper">
            <mat-form-field>
                <input [(ngModel)]="note.title" matInput placeholder="Title" type="text">
```

```
            </mat-form-field>
            <mat-form-field>
              <textarea [(ngModel)]="note.text" matInput placeholder="Text"></textarea>
            </mat-form-field>
            <div class="keep-comp-button">
               <button mat-button (click)="takeNotes()"> Submit</button>
            </div>
        </div>

    </mat-expansion-panel>
</div>

<!-- Notes Container -->
<!--
<div *ngFor="let note of notes">
<h2>{{note.title}}</h2>
</div>

-->
<div>
   <mat-card  *ngFor="let note of notes">
      <mat-card-title>{{note.title}}</mat-card-title>
      <mat-card-content>{{note.text}}</mat-card-content>
    </mat-card>
</div>
```


MyKeepApp - Google Chrome
localhost:4200
MyKeep
This is my Note
Title
Text
Submit
LearnAngular
Learn from Angular.io site
Learn TypeScript
Refer to TypeScriptlang.org


Terminal
jp@jp-ThinkPad-L450: ~/Workspace/CTS/myKeepApp
jp@jp-ThinkPad-L450:~/Workspace/CTS/myKeepApp$ git commit -m "FirstCommit"
[master f806ae2] FirstCommit
15 files changed, 208 insertions(+), 26 deletions(-)
rewrite src/app/app.component.html (98%)
create mode 100644 src/app/header/header.component.css
create mode 100644 src/app/header/header.component.html
create mode 100644 src/app/header/header.component.spec.ts
create mode 100644 src/app/header/header.component.ts
create mode 100644 src/app/note/note.component.css
create mode 100644 src/app/note/note.component.html
create mode 100644 src/app/note/note.component.spec.ts
create mode 100644 src/app/note/note.component.ts
jp@jp-ThinkPad-L450:~/Workspace/CTS/myKeepApp$ git push -u origin master
Username for 'https://gitlab-cts.stackroute.in': K
Password for 'https://K@gitlab-cts.stackroute.in':
Counting objects: 56, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (53/53), done.
Writing objects: 100% (56/56), 96.04 KiB | 0 bytes/s, done.
Total 56 (delta 10), reused 0 (delta 0)
To https://gitlab-cts.stackroute.in/K/Template_DataBinding_19thFeb.git
* [new branch] master -> master
Branch master set up to track remote branch master from origin.
jp@jp-ThinkPad-L450:~/Workspace/CTS/myKeepApp$
Size
Type
Modified
SLOT-1
SLOT-2
Week-1Assignment
uudn77KF536iBXGI.myKeepApp.tar.gz
"myKeepApp" selected (containing 10 items)