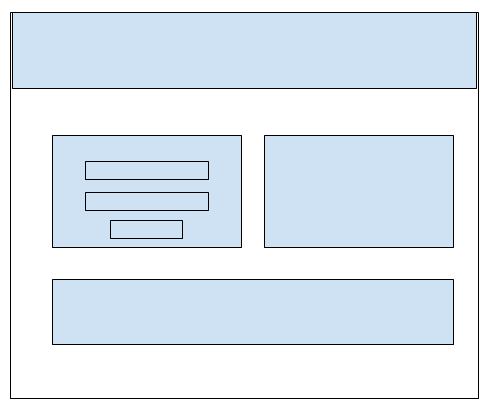
# Instructions – Assignment 6.4 – Input Properties

**Layout**

home



gpa-calculator-app, part 2

**Instructions**

* Make a copy of the gpa-calcualtor-app1 from Exercise 6.3 and add it to your week-6 directory
* Rename the application to gpa-calculator-app2
* Delete the node\_modules directory
* Delete the package-lock.json file
* Open the angular.json file and find and replace all “gpa-calculator-app1” entries with “gpa-calculator-app2”
* Open the package.json file and change the name to “gpa-calculator-app2”
* Run npm install and ng serve
  + You are doing this to test the application and confirm there are no errors
* base-layout.component.ts
  + Update the assignment variable’s value to “Assignment 6.4 - Input Properties”
* Generate two new components: gpa and grade-summary
  + ng g c gpa
  + ng g c grade-summary
  + Note: Our “home” page will consist of three parts, a form to input courses and grades, a grade summary area (grade-summary component), and a gpa section (gpa component)
* home.component.html
  + Using flex layout create one row with two divs side-by-side
    - In the left-hand div add a mat-card (this will be reserved for a form; also for the rest of this assignment we will refer to this section as the “grade-form”)
      * Give the mat-card-title a value of Grade Entry
    - In the right-hand div add a mat-card and in the mat-card-content add the app-grade-summary component
      * Give the mat-card-title a value of Transcript
  + Using flex layout create one column underneath the two side-by-side columns
    - In the div add a mat-card and in the mat-card-title add the app-gpa element after the word GPA:
      * Text align the mat-card-title to the center
  + The basic structure is
    - <div fxLayout=”row wrap” fxLayoutGap=”20px”><div fxFlex class=”grade-form”></div><div fxFlex class=”grade-summary”><app-grade-summary></app-grade-summary></div></div> <br /> <div fxLayout=”column” fxlayoutGap=”20px”><div fxFlex class=”gpa”><ap-gpa></app-gpa></div>
* app.module.ts
  + Add import statements for MatFormFieldModule, MatInputModule, MatListModule, FormsModule, and MatSelectModule
    - import { MatFormFieldModule } from ‘@angular/material/form-field’;
    - import { MatInputModule } from ‘@angular/material/input’;
    - import { MatListModule } from ‘@angular/material/list’;
    - import { FormsModule } from ‘@angular/forms’;
    - import { MatSelectModule } from ‘@angular/material/select’;
    - Note: we are using template-driven forms here; during week seven we will learn about reactive forms and data validation
  + Add the imported modules to the imports array
* Add a new file to the app directory and name it transcript.interface.ts
* transcript.interface.ts
  + Create a new interface and name it ITranscript with the following fields
    - course: string
    - grade: string
* home.component.ts
  + Add an import statement for the ITranscript interface
    - import { ITranscript } from “../transcript.interface’;
  + Add a variable named transcriptEntry of type ITranscript
    - transcriptEntry: ITranscript
  + Add a variable named selectableGrades of type Array<string> and assign it the following default values
    - A, A-, B+, B, B-, C+, C, C-, D+, D, D-, F
    - selectableGrades: Array<string> = [...] (replace the dots with the actual default grade values)
  + Add a variable name transcriptEntries of type Array<ITranscript> and set it to an empty array
    - transcriptEntries: Array<ITranscript> = []
  + In the components constructor create an empty ITranscript object and assign it to the transcriptEntry variable
    - this.transcriptEntry = {} as ITranscript;
  + Add a new function and name it saveEntry()
    - In the body of the function push this.transcriptEntry to the transcriptEntries array
      * this.transcriptEntries.push(this.transcriptEntry)
    - Next, set this.transcriptEntry as an empty ITranscript object
      * this.transcriptEntry = {} as ITranscript;
* home.component.html
  + grade-form
    - Add a new mat-card, mat-title, and mat-content
    - Give the mat-title a title of “Grade Entry”
    - Inside the mat-content add a form with with two form fields (input and mat-select), and mat-labels
      * Give the first input field an id of txtCourse. a label of “Course”, and map the transcriptEntry.course using the “Banana in a box”
        + [(ngModel)]=”transcriptEntry.course”
      * Give the second mat-form-field a mat-select
      * mat-select
        + Give it a name of “selectGrade” and map the transcriptEntry.grade using the “Banana in a box”

<mat-select [(ngModel)]=”transcriptEntry.grade” name=”selectGrade”>

* + - * + Inside the mat-select, add a mat-option and using \*ngFor to loop over the selectableGrades

<mat-option \*ngFor=”let grade of selectableGrades” [value]=”grade”> {{ grade }} </mat-option>

* + - * Add a button with a display name of Save (this goes inside the mat-content element)
      * Give the button an id of btnSave and add a click event to call the saveEntry() function
        + <button mat-raised-button color=”primary” (click)=”saveEntry()”>Save</button>
    - Grade-summary
      * Add an \*ngFor loop, iterate over the “transcriptEntries and assign the variables as input values
        + <app-grade-summary \*ngFor=”let entry of transcriptEntries” [course]=”entry.course” [grade]=”entry.grade”
* grade-summary.component.ts
  + Add an import statement for Input and add it to the @angular/core import
    - * import { Component, OnInit, Input } from ‘@angular/core’;
  + Add two input values: grade of type string and course of type string
    - @Input() grade: string;
    - @Input() course: string
* grade-summary.component.html
  + Using a mat-list, add a mat-list-item for the course and grade
    - <mat-list-item>{{ course }} | {{ grade }}</mat-list-item>
* Run and test the application
  + Test by adding multiple course/grade combos.  The results should be added to the grade-summary component
* home.component.ts
  + Add a new variable named gpaTotal of type number and set the initial value to 0
    - gpaTotal: number = 0;
  + Add a function called calculateResults()
  + calculateResults()
    - Add a local variable named gpa
      * let gpa: number = 0;
    - Add a for loop and iterate over the transcriptEntries array
      * In the body of the loop add a case statement that checks the transcriptEntry.grade and assigns the appropriate value (see the Assignment 6.4 - Grade Scale document for the grade scale)
        + switch (this.transcriptEntry.grade) { case: ‘A’: gpa += 4.0; break;}
    - Outside the body of the for loop, divide the gpa against the transcriptEntires array length  and assign the results to the gpaTotal variable
      * this.gpaTotal = gpa / this.transcriptEntries.length;
* gpa.component.ts
  + Add an import statement for Input properties
    - import { Component, OnInit, Input } from ‘@angular/core’;
  + Add an @Input variable named gpaTotal of type number
    - @Input() gpaTotal: number;
* gpa.component.html
  + Output the gpaTotal variable and use the pipe operator to round the value to two decimal places
    - {{ gpaTotal | number: ‘1.1-2’  }}
      * Note: the first one represents how many whole number digits we want to show.  The second 1 represents the minimum number of fractional decimals to display and the last value represents the maximum fractional decimals to display
* home.component.html
  + Add a button above the card in the gpa section and give it a (click) event to call the calculateResults() event and name the button “Calculate GPA
    - <button mat-button color=”primary” (click)=”calculateResults()”>Calculate GPA</button>
  + Add an input entry to the app-gpa component called [gpaTotal] and pass-in the gpaTotal value
* Run and test the application
  + Test the application by entering several grades and clicking on the “Calculate GPA” button.  You should see the GPA total in the bottom div.
* home.component.ts
  + Add a new function named clearEntries()
    - In the body of the clearEntries function set the transcriptEntries variable to an empty array and set the gpaTotal variable to 0
      * this.transcriptEntries = [];
      * this.gpaTotal = 0;
* home.component.html
  + Add a button above the mat-card in the grade-summary section, call the clearEntries() function and give it a title of “Clear Transcript”
    - <button mat-button color=”primary” (click)=”clearEntries()”>Clear Transcript</button>
* Go ahead and rerun the application and test the changes we made
  + Note you should be able to clear the grades in the Transcript section now