



Challenge

Style a Single

Page Application

Using Angular

Material



Challenge

- Challenge: Style the Keep-Note application using Angular Material

Points to Remember

- Appropriate Angular Material components should be used for styling the application.
- Pre-built theme color palettes should be applied consistently across all the Angular Material components in the application.
- The Angular Material navigation schematics should be used to quickly add a toolbar and a side navigation bar to the application.
- The Angular Material component selectors should not be used to define the CSS style properties.
 - Instead, define `id` or `class` attributes on the elements and use them as CSS selectors to define the style rules.

Instructions for Challenge

- [Click here](#) for the boilerplate.
- Read the README.md file in the boilerplate for further instructions about the challenge.
- Fork the boilerplate into your own workspace.
- Clone the boilerplate into your local system.
- Run the command `npm install` to install the dependencies.
- Use the solution code of the Keep-Note application developed in the challenge of the sprint, Sprint 5: Build Reusable Application Logic Using Angular Services of the Angular Course: Building Single Page Applications Using Angular .

Notes:

1. The solution of this challenge will undergo an automated evaluation on the CodeReview platform. (Local testing is recommended prior to the testing on the CodeReview platform)
2. The test cases are available in the boilerplate.

Context

As you are aware, Keep-Note is a web application that allows users to maintain notes. It is developed as a single-page application using multiple components.

Note: The stages through which the development process will be carried out are shown below:

Stage 1: Create basic Keep-Note application to add and view notes.

Stage 2: Implement unit testing for the Keep-Note application.

Stage 3: Create Keep-Note application with multiple interacting components to add, view and search notes.

Stage 4: Implement persistence in the Keep-Note application.

Stage 5: Style the Keep-Note application using Material design.

Stage 6: Create simple form with validation in the Keep-Note application.

Stage 7: Create complex form with validation in the Keep-Note application.

Stage 8: Enable navigation in the Keep-Note application.

Stage 9: Secure routes in the Keep-Note application

Context (cont'd)

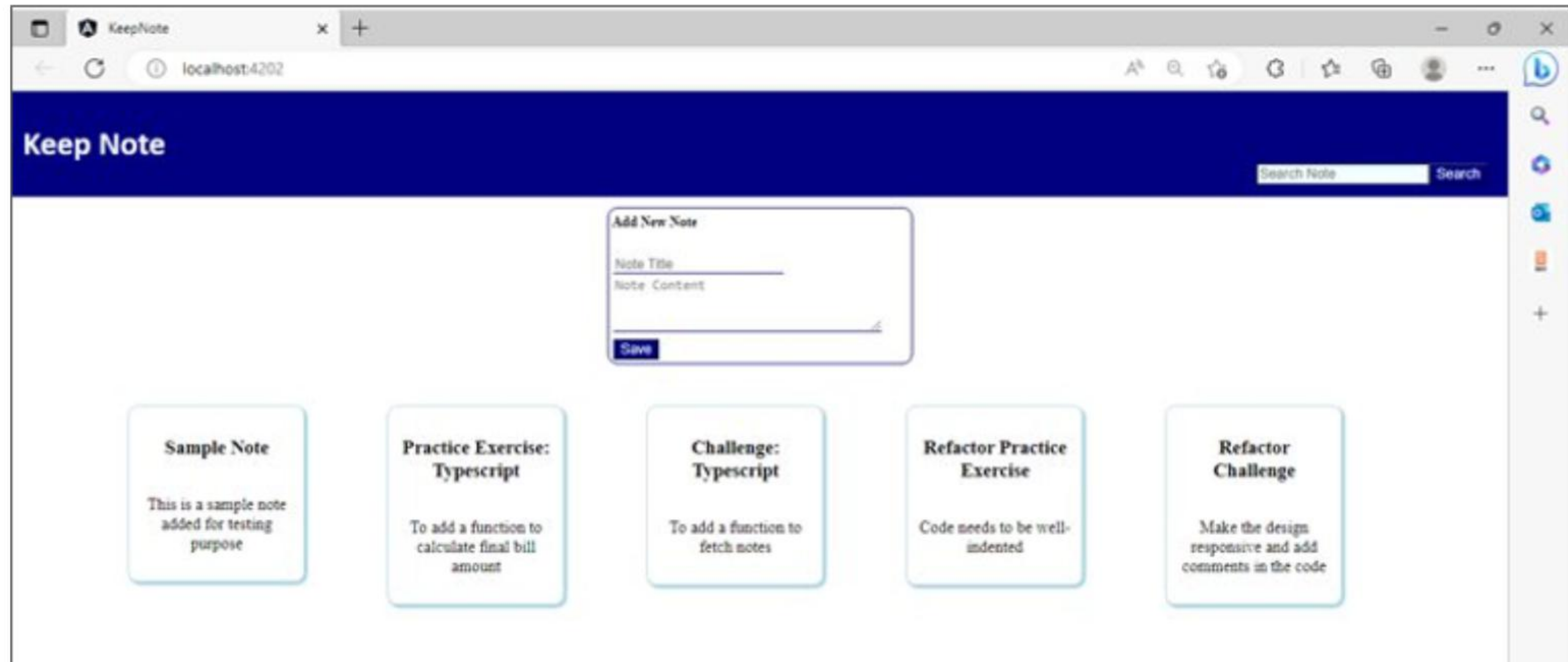
In this sprint, we are at Stage 5.

Till this stage, the layouts and appearance of UI of the Keep-Note application were defined using the CSS style properties.

In this stage, for pleasing aesthetics and a rich user experience, the looks of the Keep-Note application should be enhanced by using the Angular Material library.

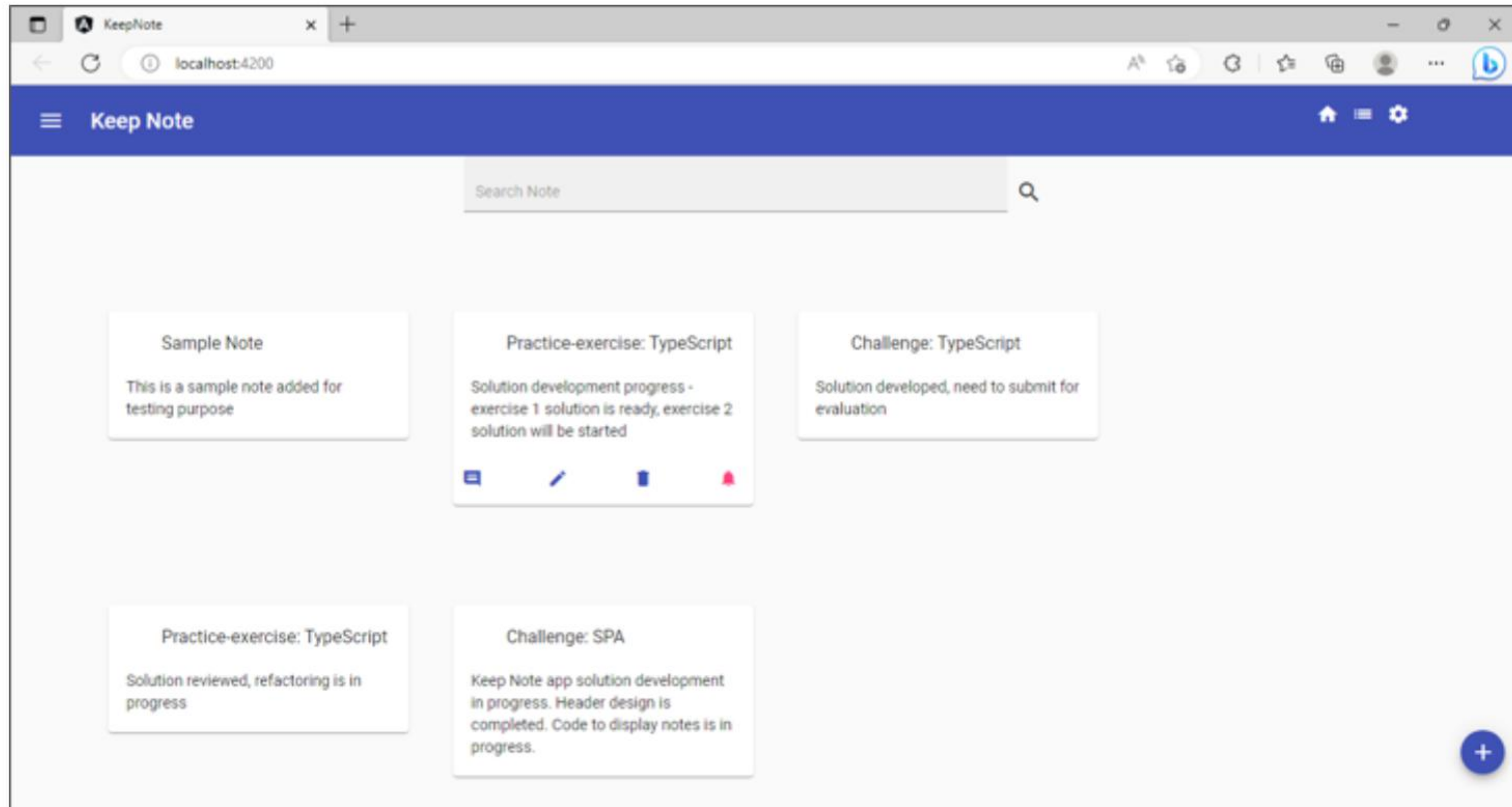
Output Without Angular Material

- The image below shows the sample output of the Keep-Note application without using Angular Material for styling:



Output with Angular Material

- The image below shows the sample output of the Keep-Note application after styling using Angular Material:



Style the Keep-Note Application using Angular Material

Style the Keep-Note application using the Angular Material components, schematics and a pre-built theme.

Note: The tasks to style the application are given in the upcoming slide.

CHALLENGE



Tasks

To develop the solution for the Keep-Note application, following tasks need to be completed:

- Task 1: Install Angular Material.
- Task 2: Style the HeaderComponent.
- Task 3: Style the NoteAddComponent.
- Task 4: Style the NoteViewComponent.
- Task 5: Style the SearchComponent.
- Task 6: Add NavBarComponent using the navigation schematics.

Note: The partial instructions to complete the tasks are given in the upcoming slides.

Task 1: Install Angular Material

- Install Angular Material packages using the Angular CLI command.
- During installation, select a pre-built theme, allow Angular Material typography styles to be installed, and opt for `Include` and `enable animations`.

Task 2: Style the HeaderComponent

- The HeaderComponent should be styled using the Angular Material Toolbar component.
- It should contain the application title Keep Note.
- It should also contain navigation icons designed using Angular Material Icon component.
- Refer to the sample output given below of the expected styled HeaderComponent.



Note: Click on the links given below to know more about these Angular Material components:

- [Toolbar component](#).
- [Icon component](#).

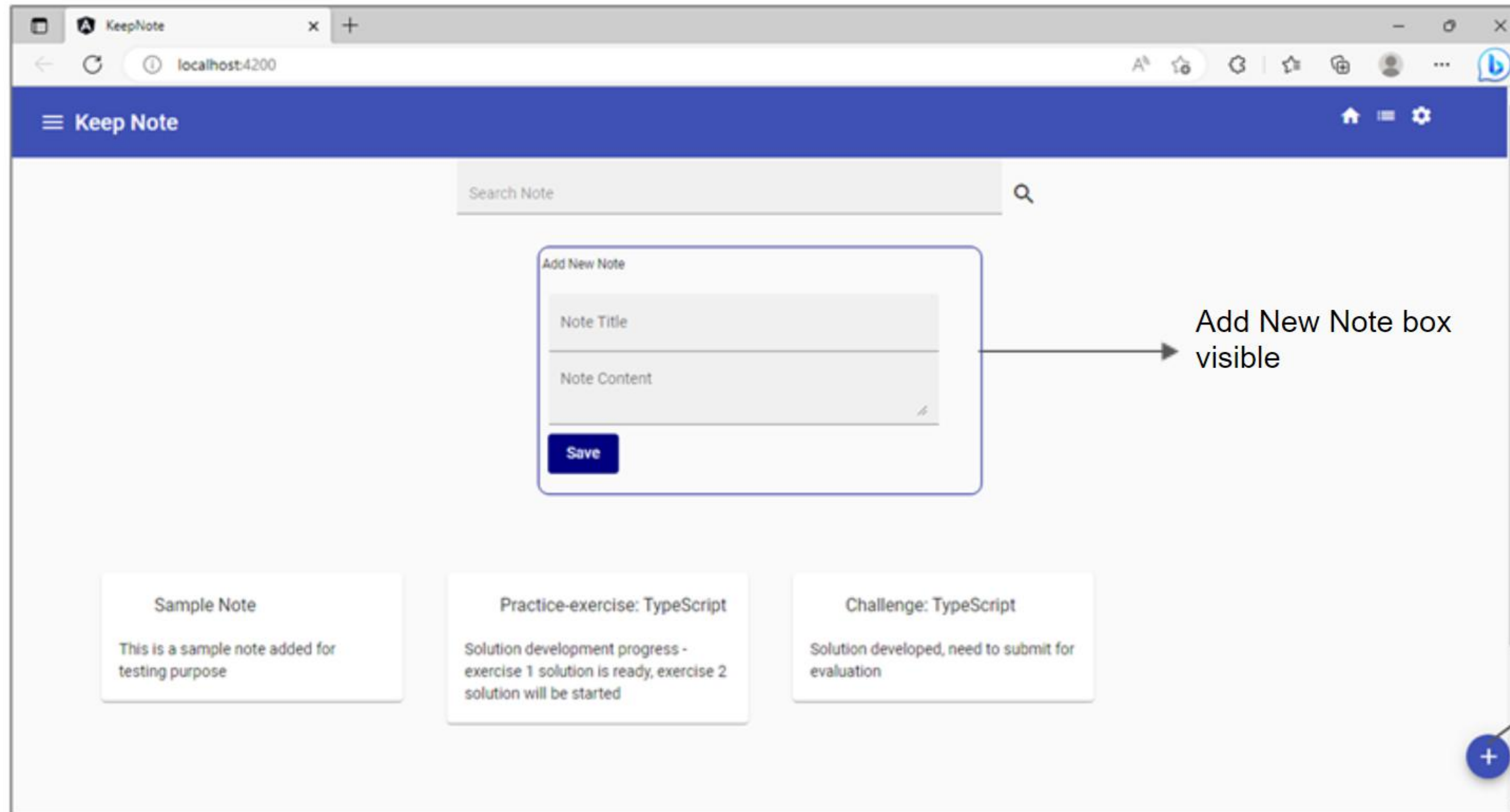
Task 3: Style the NoteAddComponent

- Use Angular Material Input and Button components for styling the NoteAddComponent.
 - It is recommended that, to keep the UI clutter-free, the view for the NoteAddComponent should be hidden/collapsed and visible only when the user clicks on a button.
- The input field should contain appropriate labels to instruct the user about the input required.

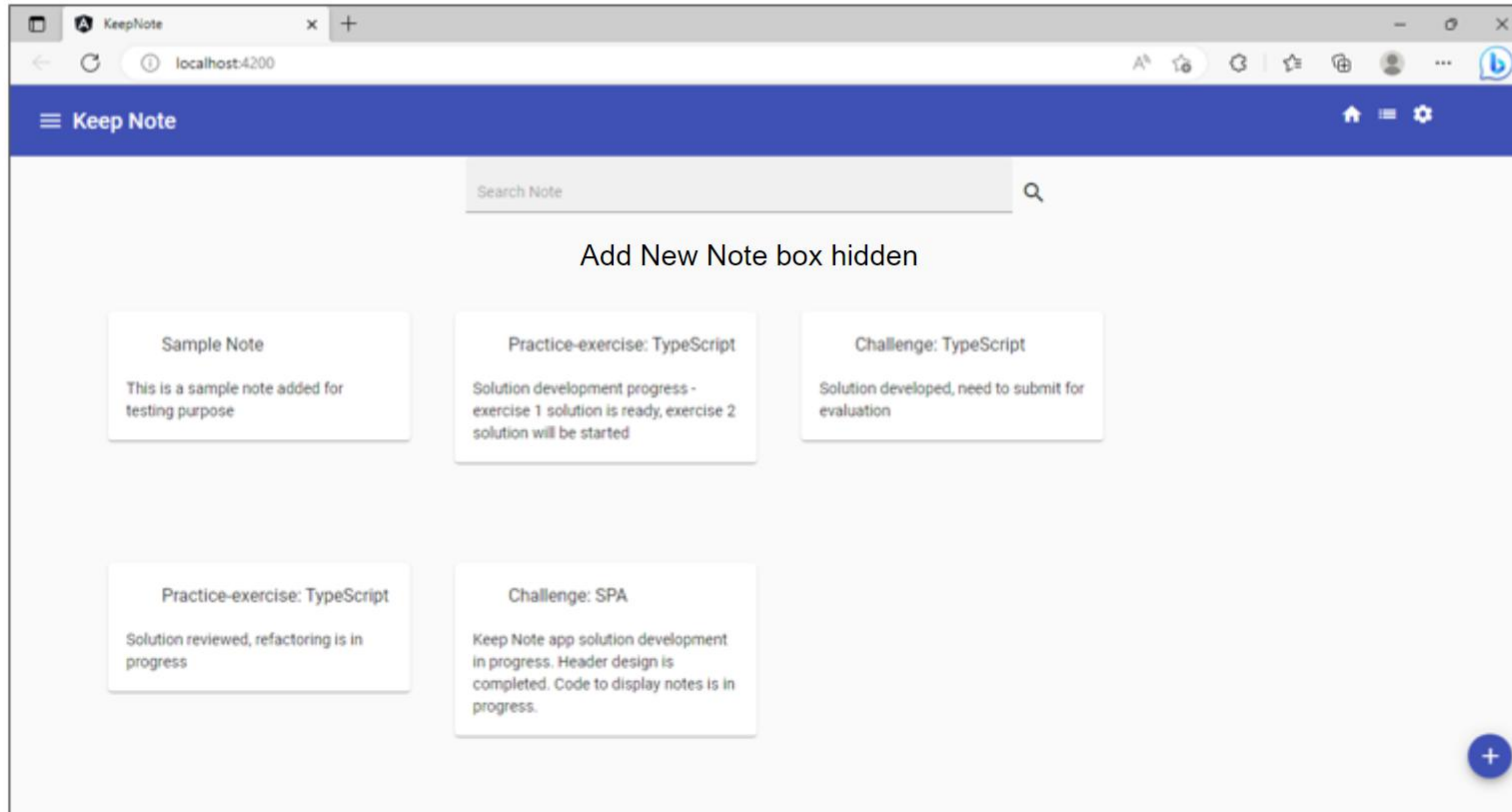
Note: Click on the links given below to know more about these Angular Material components:

- [Form field component.](#)
- [Button component.](#)

Task 3: Expected Output (+ Button Clicked)



Task 3: Expected Output (+ Button Clicked Again)



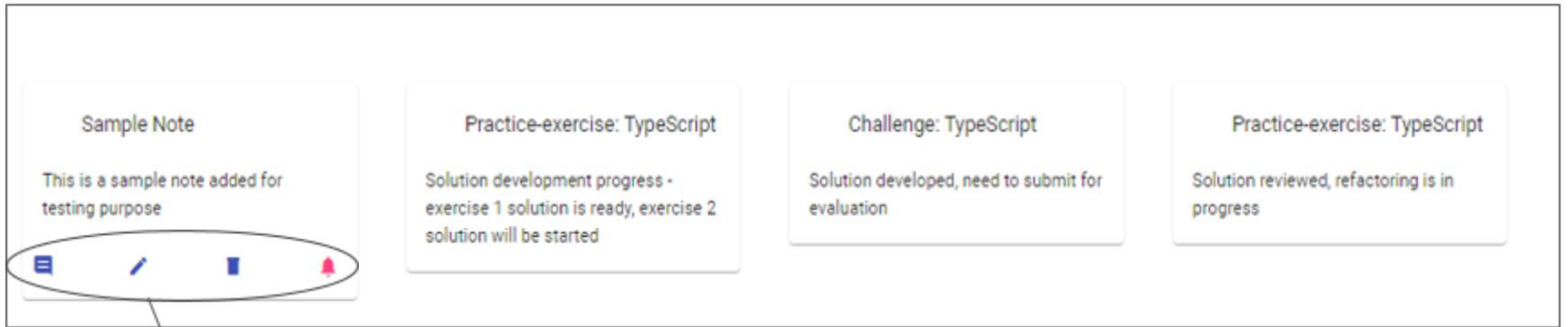
Task 4: Style the NoteViewComponent

- Use Angular Material Card component to display the notes in the card format on the NoteViewComponent.
- Add more icon buttons, using Angular Material Button and Icon components, in the card footer for commenting, editing, deleting, and setting up reminders on the selected note.
- For a clutter-free UI, the buttons on the note card can be kept hidden, and they are visible only when the note is selected or hovered.

Notes:

1. Click on the link given below to know more about the Angular Material Card component:
 - [Card component](#).
2. Refer to the link below to explore suitable Material Icons that should be used in the Note card:
 - [Material Symbols and Icons - Google Fonts](#)

Task 4: Expected Output



Footer icons on card are visible only when a note card is hovered

Task 5: Style the SearchComponent

- The SearchComponent design can be kept simple by having only an input field that accepts note title as the input to search.
- A search icon can be used with the search input field to suggest its intended purpose.
- Use Angular Material Form Field, Button and Icon components for styling the SearchComponent.
- Refer to the sample output below of the expected styled SearchComponent.

A sample output of the styled SearchComponent. It consists of a light gray rectangular input field with a thin black border. Inside the field, the text "Search Note" is displayed in a light gray font. To the right of the input field, there is a magnifying glass icon, also in a light gray color.

Task 6: Add NavBarComponent using the Navigation Schematics

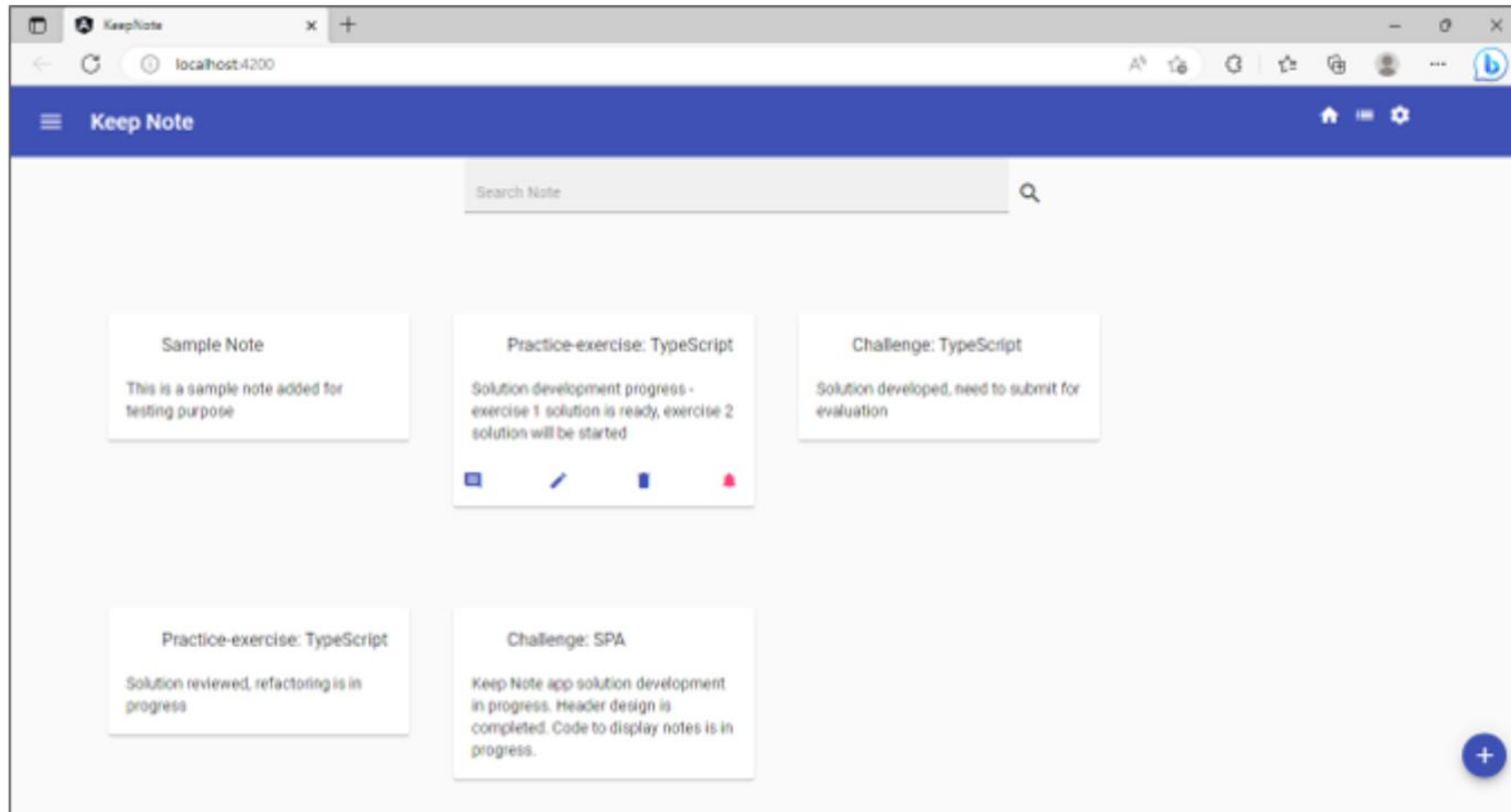
- Use the Angular Material Navigation schematics to add a component with the name NavBar to the Keep-Note application.
- Refactor the existing code so that the newly added component displays a collapsible sidebar, a toolbar with the application title and a content area for rendering the note contents.
- The sidebar should display following links:
 - Home
 - New Note
 - Archives
 - Settings

Notes:

1. The links should be accompanied with the appropriate icons..
2. Refer to the link below to explore suitable Material Icons that should be used in the sidebar:
 - [Material Symbols and Icons - Google Fonts.](#)

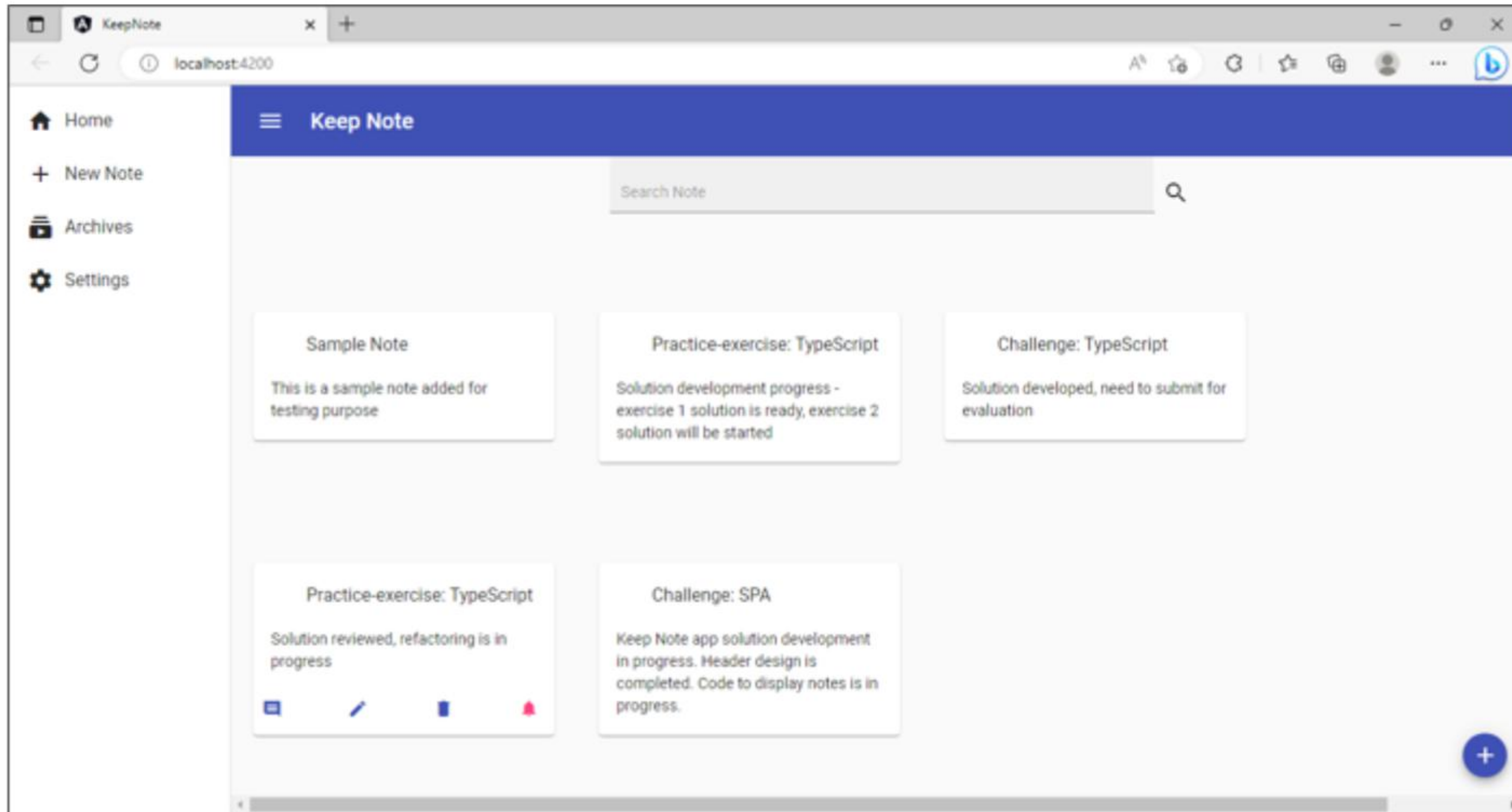
Final Expected Output (Sidebar Closed)

- The sample layout suggested for the Keep-Note application with sidebar closed, is provided in the image below:



Final Expected Output (Sidebar Opened)

- The sample layout suggested for the Keep-Note application with sidebar opened, is provided in the image below:



Test the Solution Locally

Test the solution first locally and then on the CodeReview platform. Steps to test the code locally are:

- From the command line terminal, set the path to the folder containing cloned boilerplate code.
- Run the command `ng test` or `npm run test` to test the solution locally and ensure all the test cases pass.
- Refactor the solution code if the test cases are failing and do a re-run.
- Finally, push the solution to git for automated testing on the CodeReview platform.