Slide Note

Look at the image on the next slide and identify the components

Menu

Slide Note

Look at the page and identify some of the components.

You can give different answers such as:

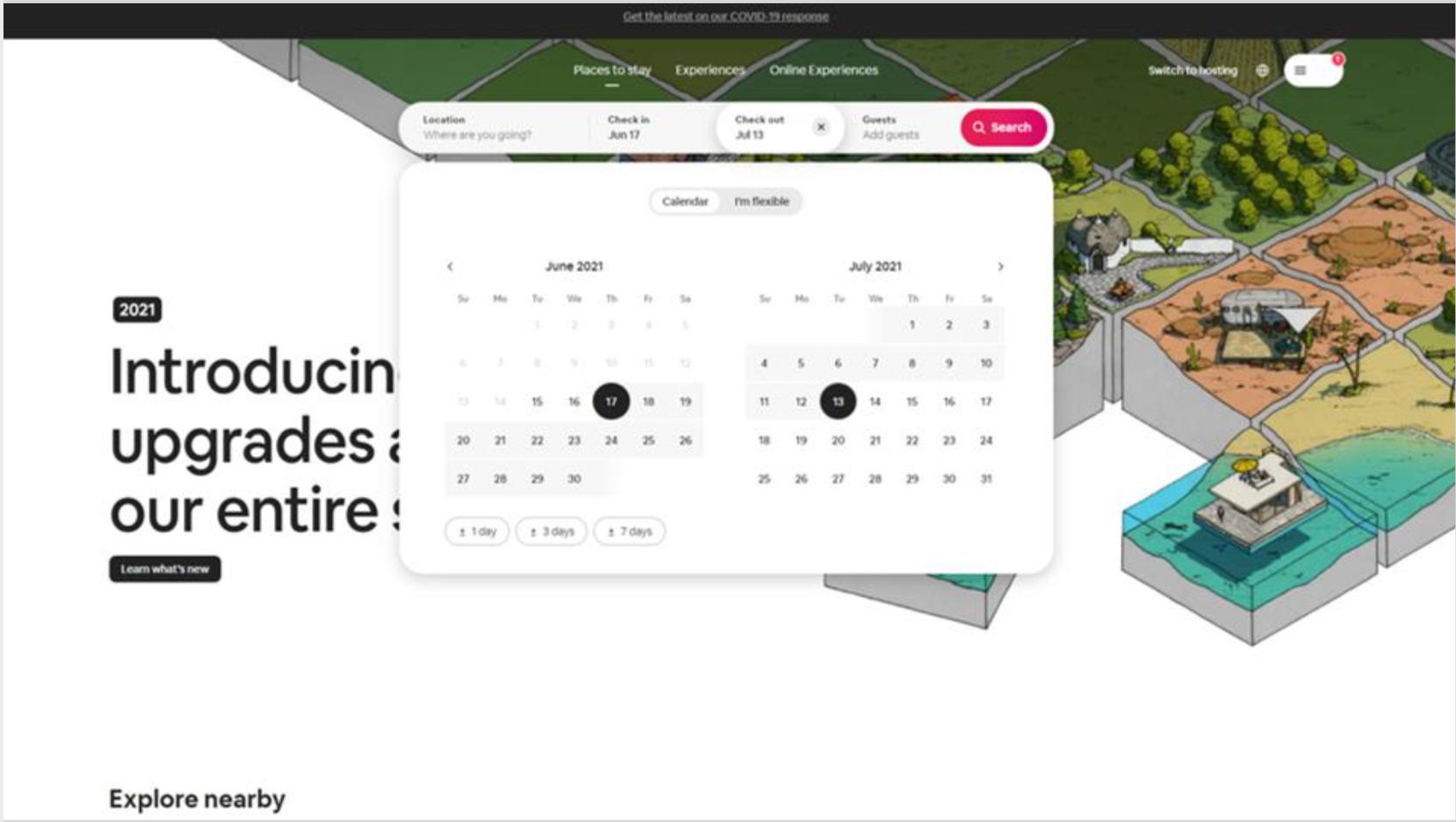
Search component

Toggle calendar button component

Calendar component

Day increment/decrement button components

Move to the next slide to discuss the identified components.



Menu

00:02 00:00:35 02/ 35

Slide Note

Identified components:

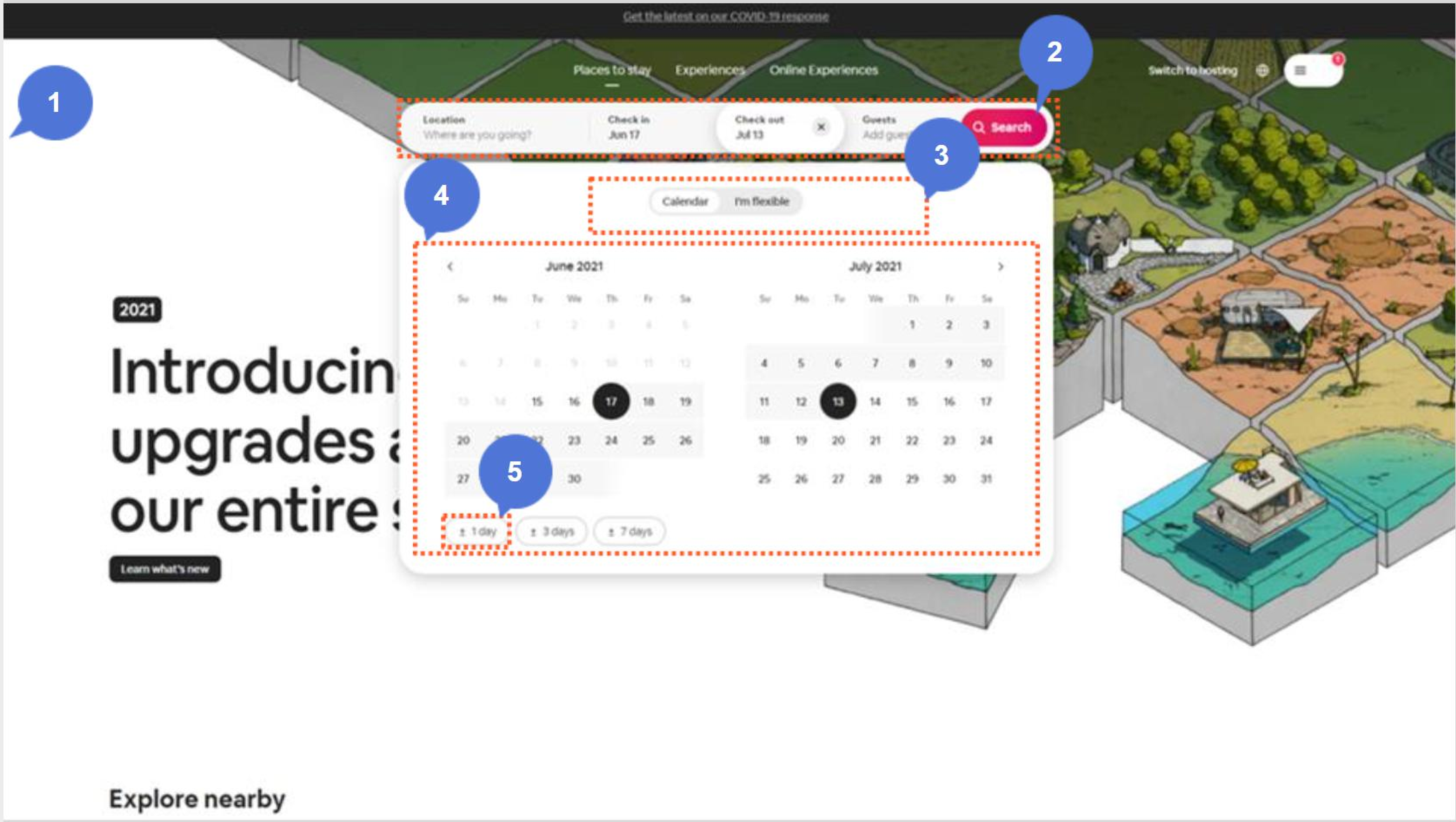
1 - Page

2 - Search view

3 - Calendar toggle button

4 - Calendar view

5 - Day increment/decrement button component



Menu

00:03 00:00:35 03/ 35

Slide Note

Look at another image and identify the components

Menu

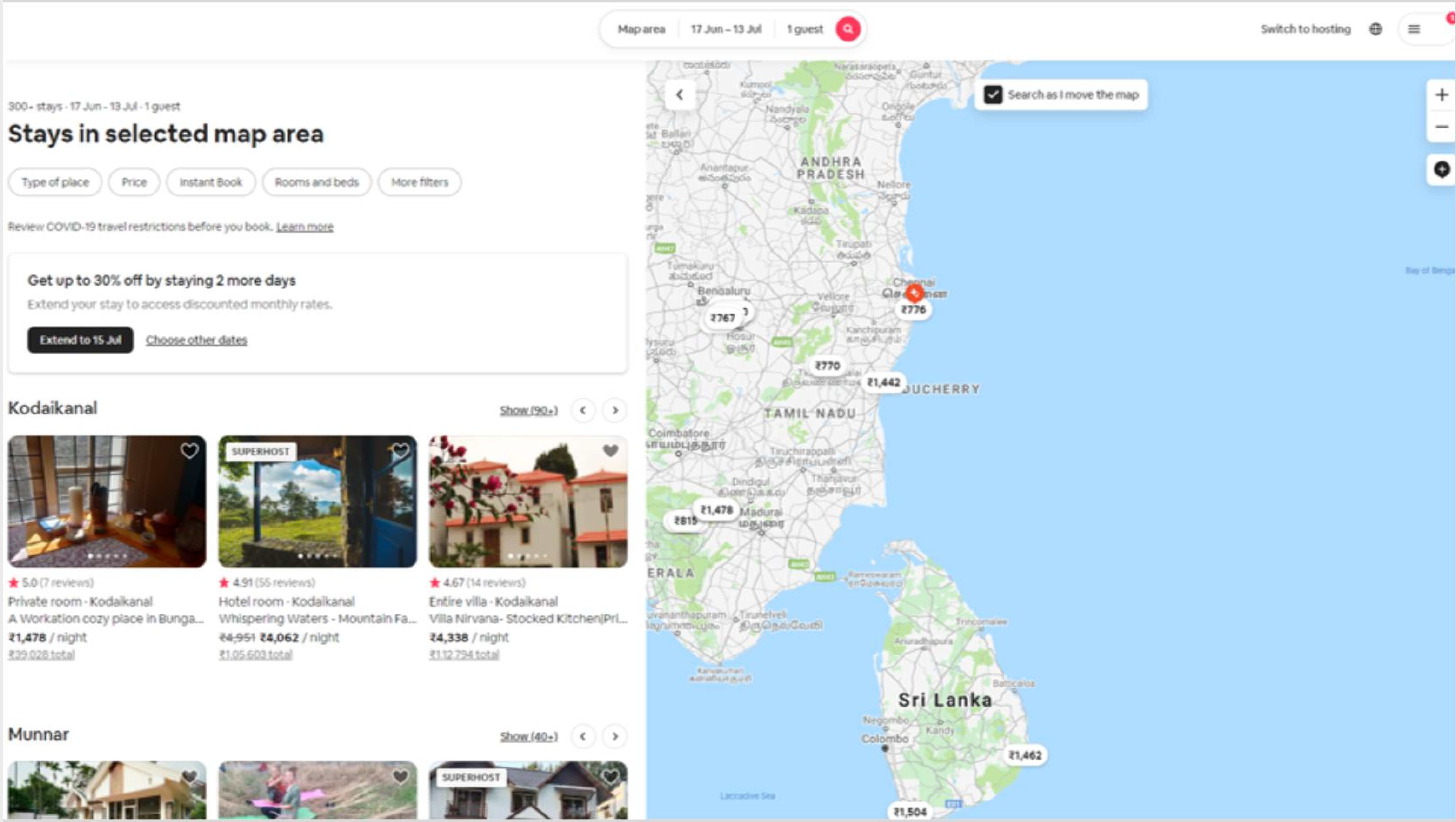
00:04:00:00:35 04/ 35

Slide Note

City search view and map view

Look at the page and identify some of the components available here

The same data pertaining to city/location is available in different views on the same page



Map area | 17 Jun – 13 Jul | 1 guest

Switch to hosting

300+ stays · 17 Jun - 13 Jul · 1 guest

Stays in selected map area

Type of place | Price | Instant Book | Rooms and beds | More filters

Review COVID-19 travel restrictions before you book. Learn more

Get up to 30% off by staying 2 more days

Extend your stay to access discounted monthly rates.

Extend to 15 Jul | Choose other dates

Kodaikanal Show (90+)

★ 5.0 (7 reviews)
Private room · Kodaikanal
A Workation cozy place in Bunga...
₹1,478 / night
₹39,028 total

SUPERHOST
★ 4.91 (55 reviews)
Hotel room · Kodaikanal
Whispering Waters - Mountain Fa...
~~₹4,951~~ ₹4,062 / night
₹1,05,603 total

★ 4.67 (14 reviews)
Entire villa · Kodaikanal
Villa Nirvana- Stocked Kitchen|Pri...
₹4,338 / night
₹1,12,794 total

Munnar Show (40+)

SUPERHOST

Menu

00:05 00:00:35 05/ 35

Slide Note

Identified components:

1-Page

2-Bnb Search View

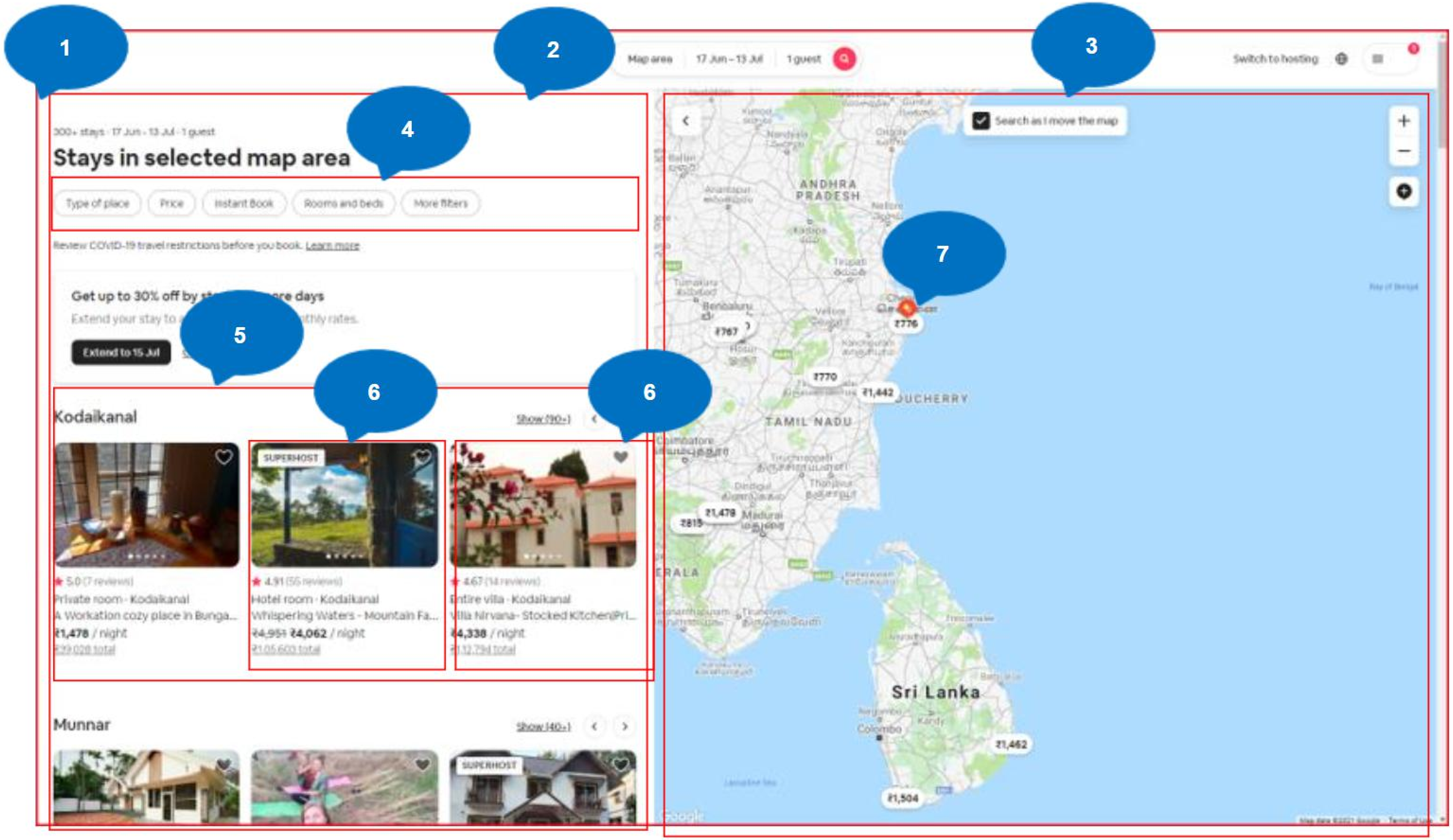
3-Bnb Map View

4-Filter Component

5- City Container view

6-City view

7-Map pins



Menu

00:06 / 00:00:35

06/ 35

Slide Note

A web page in an application is a tree of components having some hierarchy.

Are you writing code separately for each view?

No, the underlying data is the same, which is getting reused to display in each view.

Here, the same data of a city with its staying price is getting viewed in the city view using text, as well as in the map view using pins.

Think and Tell

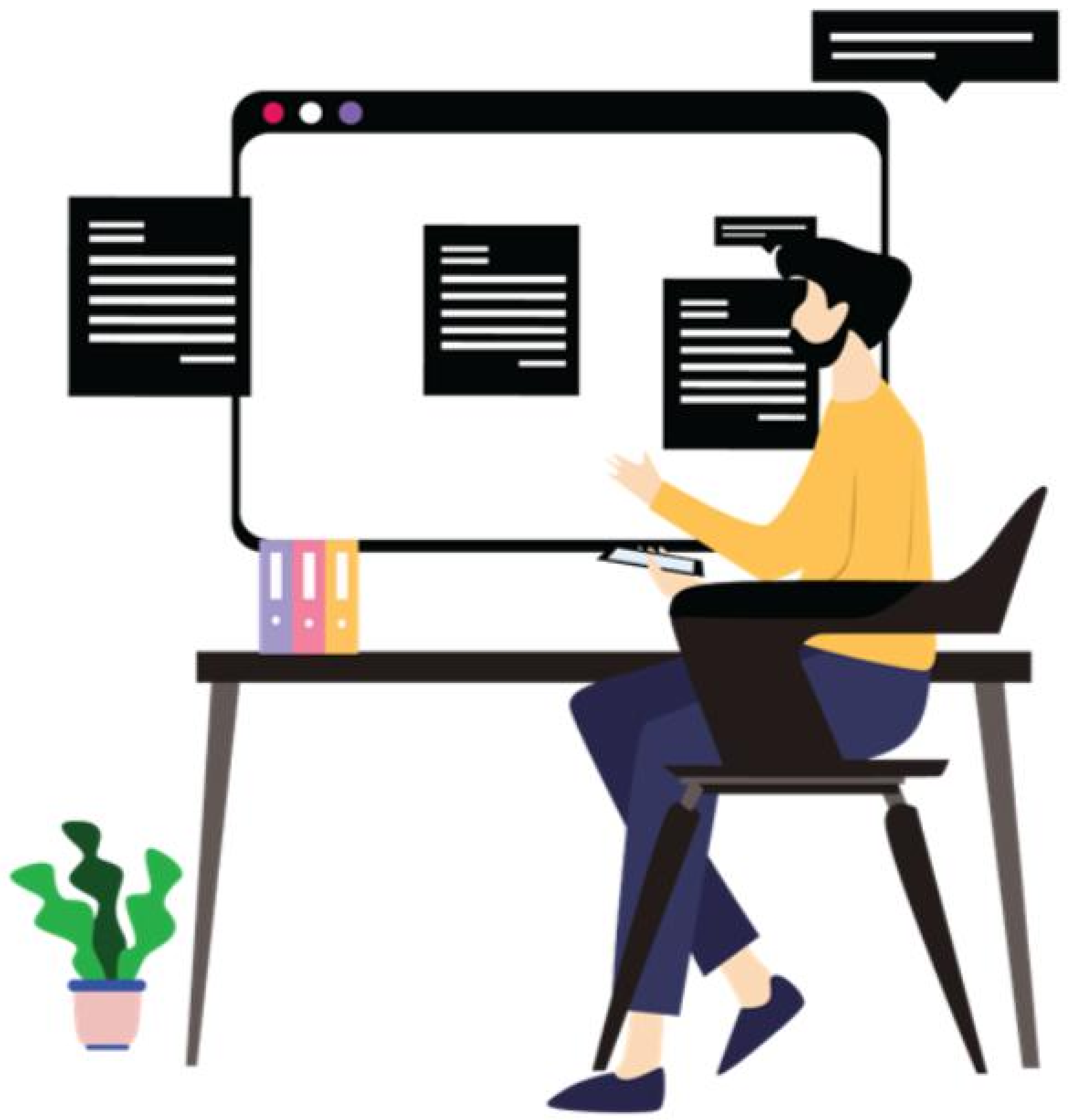
- How does the selection in the map view get updated in the search view?
- How is the user input from the Search View passed on to the City Container or the City View?
- How does the selection of accommodations in the City View get highlighted in the Map View?

Menu

00:07/00:00:35 07/ 35

Slide Note

Implement Interactions Between Angular Components



Menu

00:08:00:00:35 08/ 35

Slide Note

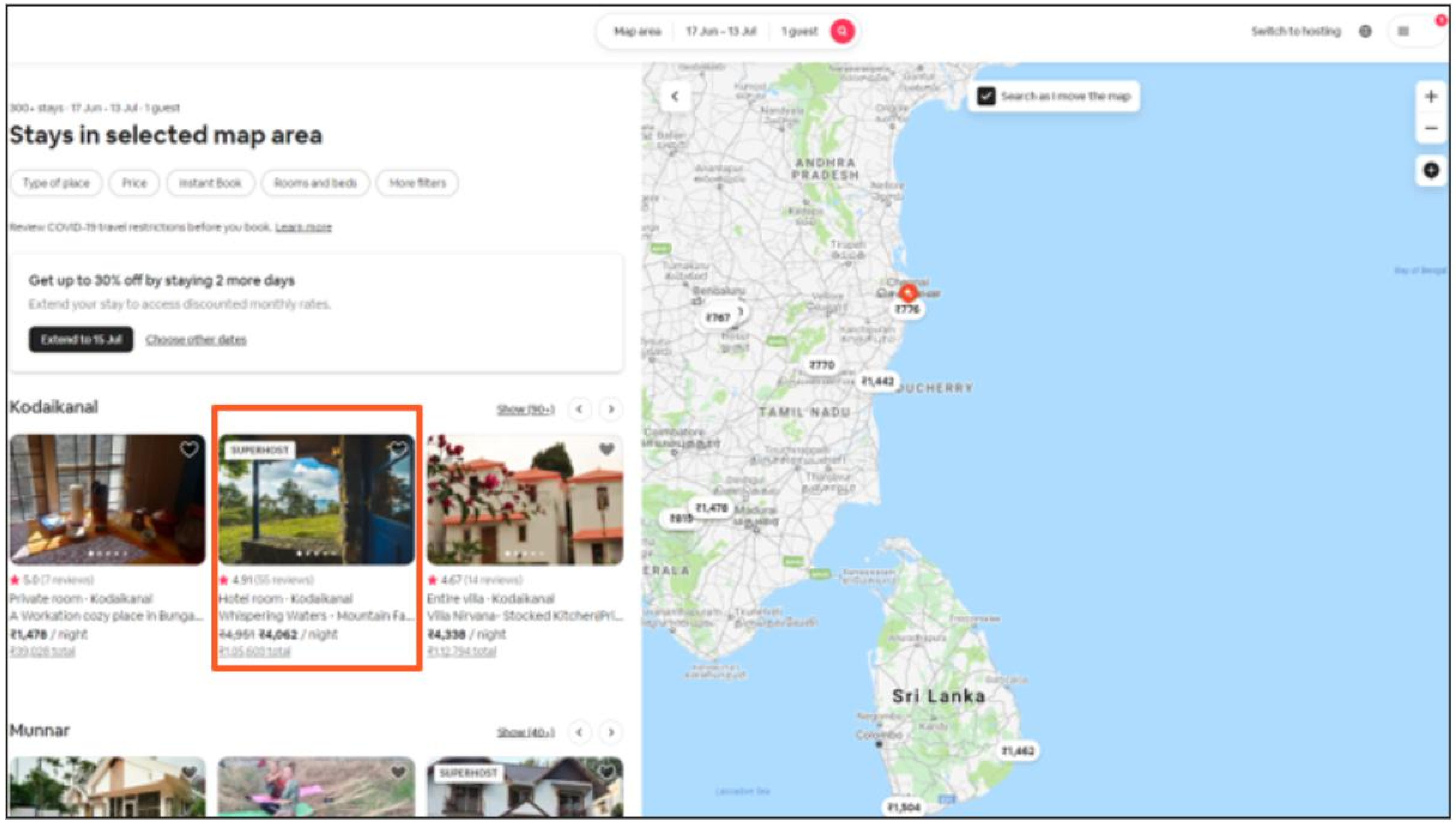
Learning Objectives

- Explain design principles for identifying components in an SPA
- Build a component hierarchy
- Explain the relationship among components in a component hierarchy
- Explain different roles of the component in a component hierarchy
- Use the `@Input()` decorator to share data from the parent component to the child component
- Use the `@Output()` decorator to share data from the child component to parent component

Menu

Slide Note

How Should a Web Page Update its Contents?


Map area | 17 Jun – 13 Jul | 1 guest

Switch to hosting

300+ stays · 17 Jun - 13 Jul · 1 guest

Stays in selected map area

Type of place | Price | Instant Book | Rooms and beds | More filters

Review COVID-19 travel restrictions before you book. Learn more

Get up to 30% off by staying 2 more days

Extend your stay to access discounted monthly rates.

Extend to 15 Jul | Choose other dates

Kodaikanal — Show (90+)

★ 5.0 (7 reviews)	SUPERHOST ★ 4.91 (55 reviews)	★ 4.67 (14 reviews)
Private room · Kodaikanal	Hotel room · Kodaikanal	Entire villa · Kodaikanal
A Workation cozy place in Bunga...	Whispering Waters - Mountain Fa...	Villa Nirvana- Stocked Kitchen|Pri...
₹1,478 / night	~~₹4,951~~ ₹4,062 / night	₹4,338 / night
₹39,026 total	₹1,05,600 total	₹1,12,794 total

Munnar — Show (40+)

Search as I move the map

What happens on the page when location selection is changed?

Should only the map view get reloaded with revised location, or the complete page should reload?

Menu

00:10 00:00:35 10/ 35

Slide Note

SPA – Resultant of a Component Hierarchy Build Up

- A Single Page Application, or SPA. is a web application that dynamically updates the web page by re-writing contents fetched from the server.
  - The part of the page, also known as view, whose content gets re-written is only loaded.
  - <u>The entire page is not reloaded.</u>
- An SPA contains a single `.html` file, the starting point for the rest of the application.
- The various views of the SPA are rendered through components.
- The components can have child components to create nested views.

Menu

Slide Note

How to Identify Components?

- How can we identify components in an SPA?
- Are there any design principles that can guide us in identifying the components?

Menu

00:12 00:00:35 12/ 35

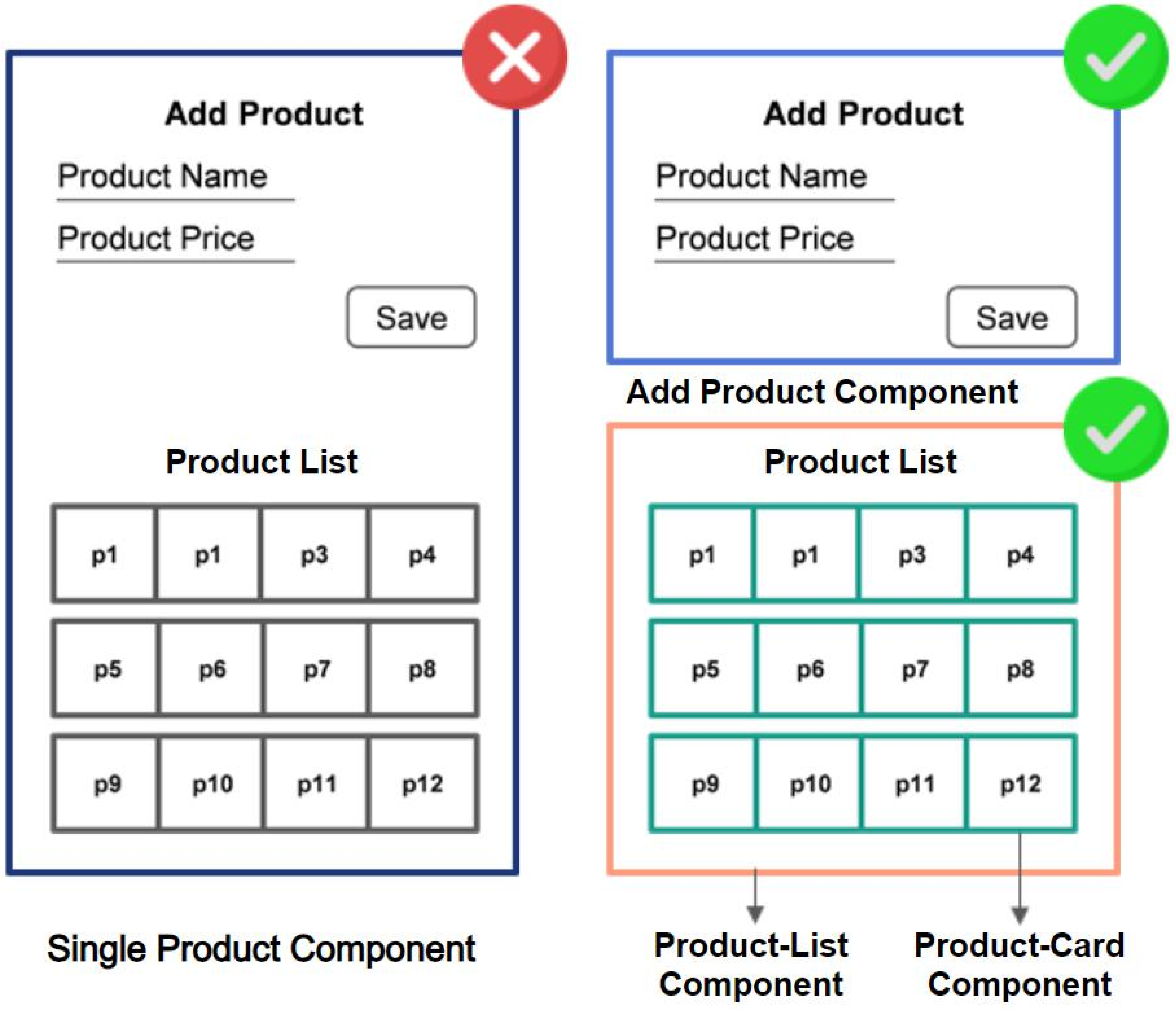
Slide Note

Add Product – adding product

Product List – container component provides product list data and renders product-cards

Product Card – displays one product item

Single Responsibility Principle or SRP



- This principle states that a component is expected to handle a single responsibility in an SPA.
- The image to the left shows a single component handling add and view responsibilities for product data.
  - It is a violation of SRP.
- The image to the right shows one component handling only one responsibility.

Slide Note

Don't Repeat Yourself (DRY) Principle

```
<app-media-icon>like</app-media-icon>
<app-media-icon>comment</app-media-icon>
<app-media-icon>share</app-media-icon>
```

A sample code to render the icon component with different image icons

- The Don't Repeat Yourself (DRY) principle encourages code reusability in an application.
- It states that the components should be designed in a way to be reusable.
- The three popular icons shown on the screen have a similar look and feel but differ in image content.
- One single component can be designed that accepts the identifier for an icon for display.
- Multiple instances of this component can be added to the template with different icons.

Slide Note

Guidelines to identify components

1. Use Single Responsibility Principle and Do not Repeat Yourself principle to identify components/

2. Header of the page should only display page headings with icons

Header would be one component.

1. There could be multiple icons in the header each serving a specific responsibility, with same look and feel.

Icon could be another component rendered by Header component.

1. The main section of the page, need to manage fruit data residing in Fruits array externally.

Fruit Manager component can be created that reads data from the external fruit array.

1. Following are the key tasks that would be performed on the fruit data:

Add fruit

Search fruit

Display fruit list

Each item in the list should be displayed with its caption, price and fruit image

1. Each of the above tasks would be handled as a single responsibility by following components:

Add-Fruit component

Search component

Fruit-List component

Fruit-Card component

For the image shown on slide:

1 – App Component

2 – Header Component

3 – Fruit-Manager Component

4 – Add Fruit Component

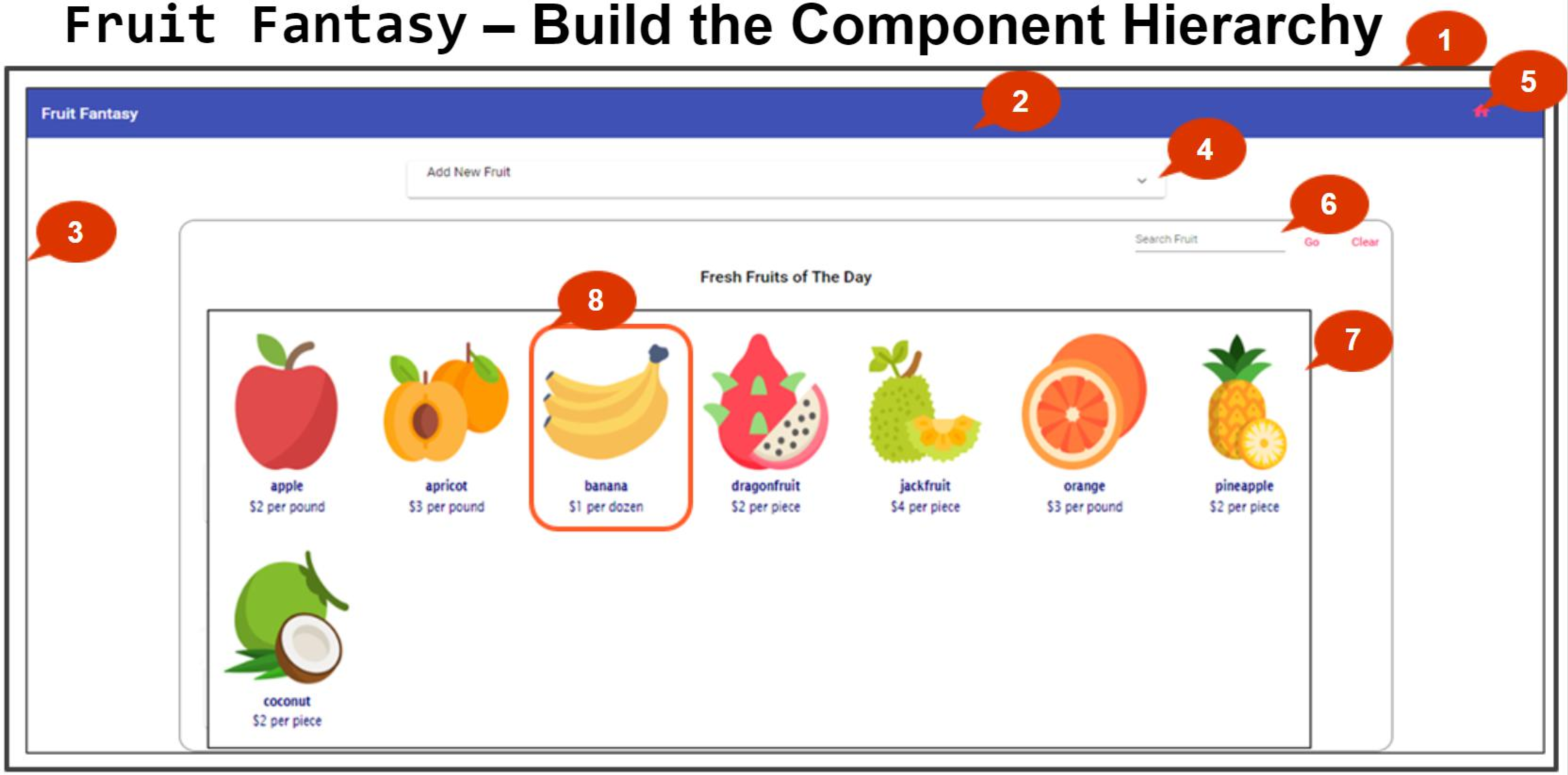
5 – Icon Component

6 – Search Component

7 – Fruit List Component

8 – Fruit Card Component

Fruit Fantasy – Build the Component Hierarchy



Menu

00:15 00:00:35 15/ 35

Slide Note

Create Components for the `Fruit-Fantasy` App

Create a component hierarchy for the `Fruit-Fantasy` app. Based on the hierarchy, create Angular components in the `Fruit-Fantasy` app.

Render the components as per the relationship depicted in the hierarchy.

Click here for the demo solution.

DEMO

Menu

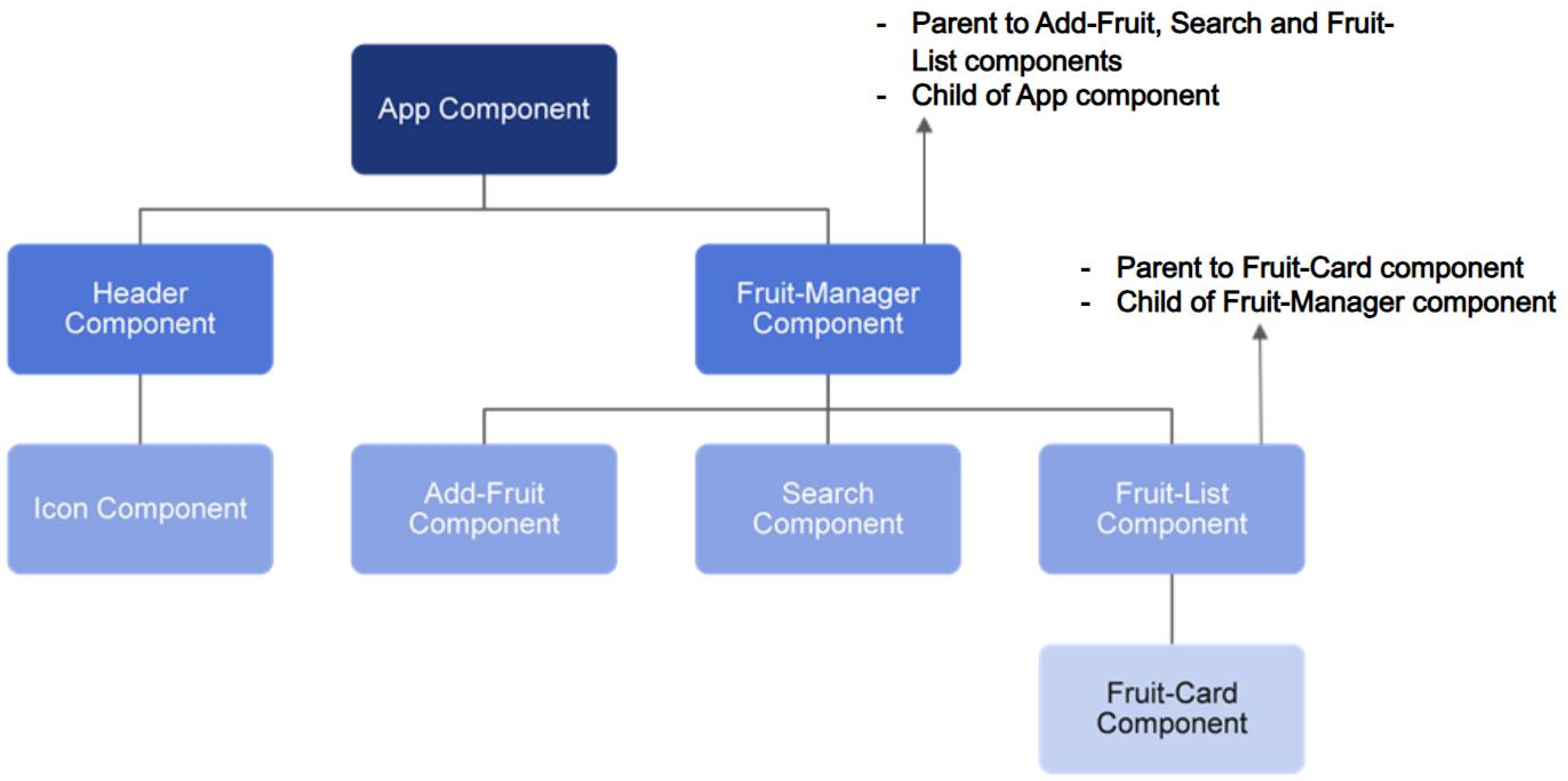
00:16:00:00:35 16/ 35

Slide Note

In a component hierarchy of an SPA, a component can be a child of another component, and a component can be a parent of another component.

Suppose you have a component called P. In the template of the P component, you write the selector of another component, say C. In this case, component C is the child of the P component, or P is the parent of C.

Component Relationship



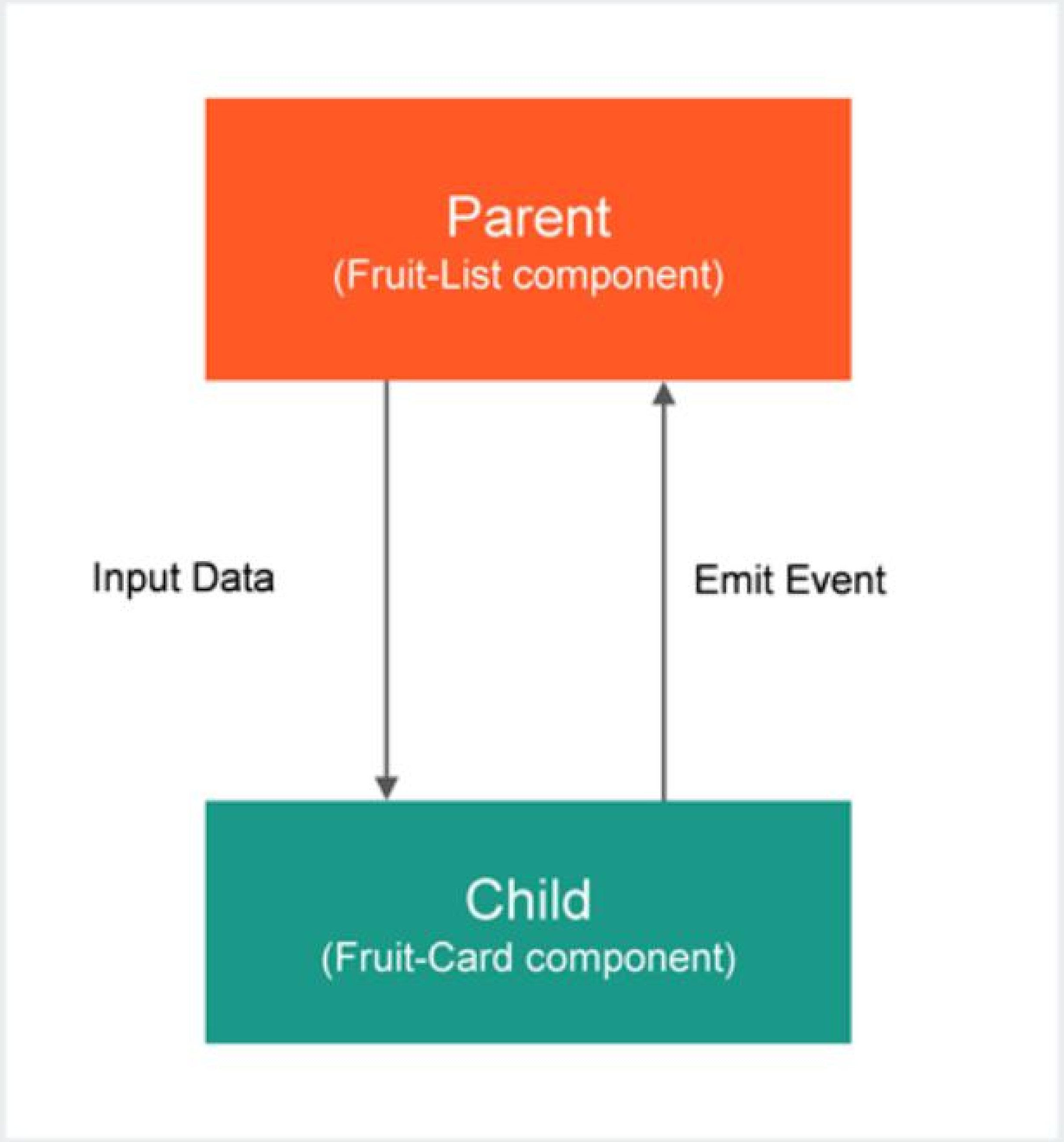
Slide Note

Role of Angular Components in SPA

- Components can be essentially classified into two types:
  - Smart or Container components
    - This component handles storing data (the data could be defined within the component or fetched from an external source).
    - The component is usually a parent component that passes this data to the child component for presentation.
    - Since it's bound to data, it would be very hard to reuse this component in another application.
  - Dumb or Presentation components
    - The component's responsibility is to present the data received from the parent component to the end user and not to fetch it from a particular location.
    - This component is highly reusable since it's not tightly bound with data but provides presentation logic to serve the data that it receives.

Menu

Slide Note



Component Interaction

- In a component hierarchy, parent and child components interact to share data.
- The data flow is bidirectional between Angular components.
- The parent component renders the child component and, while rendering, may pass data to the child component.
  - This data is input for the child component.
- When an action occurs in the child component, the child component may need to notify the parent about this action.
  - The child component notifies the parent component by emitting an event.

Menu

Slide Note

Data Flow – Parent to Child

- Here are a few instances related to the `Fruit-Fantasy` app that explain the need to enable data flow from parent to child components.

Parent (Source)	Data Sent	Child (Receiver)	Purpose
`Fruit-Manager`	List of Fruits (fruit array)	`Fruit-List`	`Fruit-Manager` provides a list of all fruits or filtered fruits to the `Fruit-List` component for rendering
`Fruit-List`	`Fruit` object	`Fruit-Card`	`Fruit-List` traverses the list and renders `Fruit-Cards`, one with each fruit item traversed

Menu

00:21:00:00:35 21/ 35

Slide Note

Think and Tell

- How does a parent component pass data to its child component?
- How does the child component receive data from its parent?

Menu

00:22 00:00:35 22/ 35

Slide Note

`Fruit-Fantasy` – Display List of Fruits

In the `Fruit-Fantasy` app, modify the `Fruit-Manager` component to create a list of fruit objects.

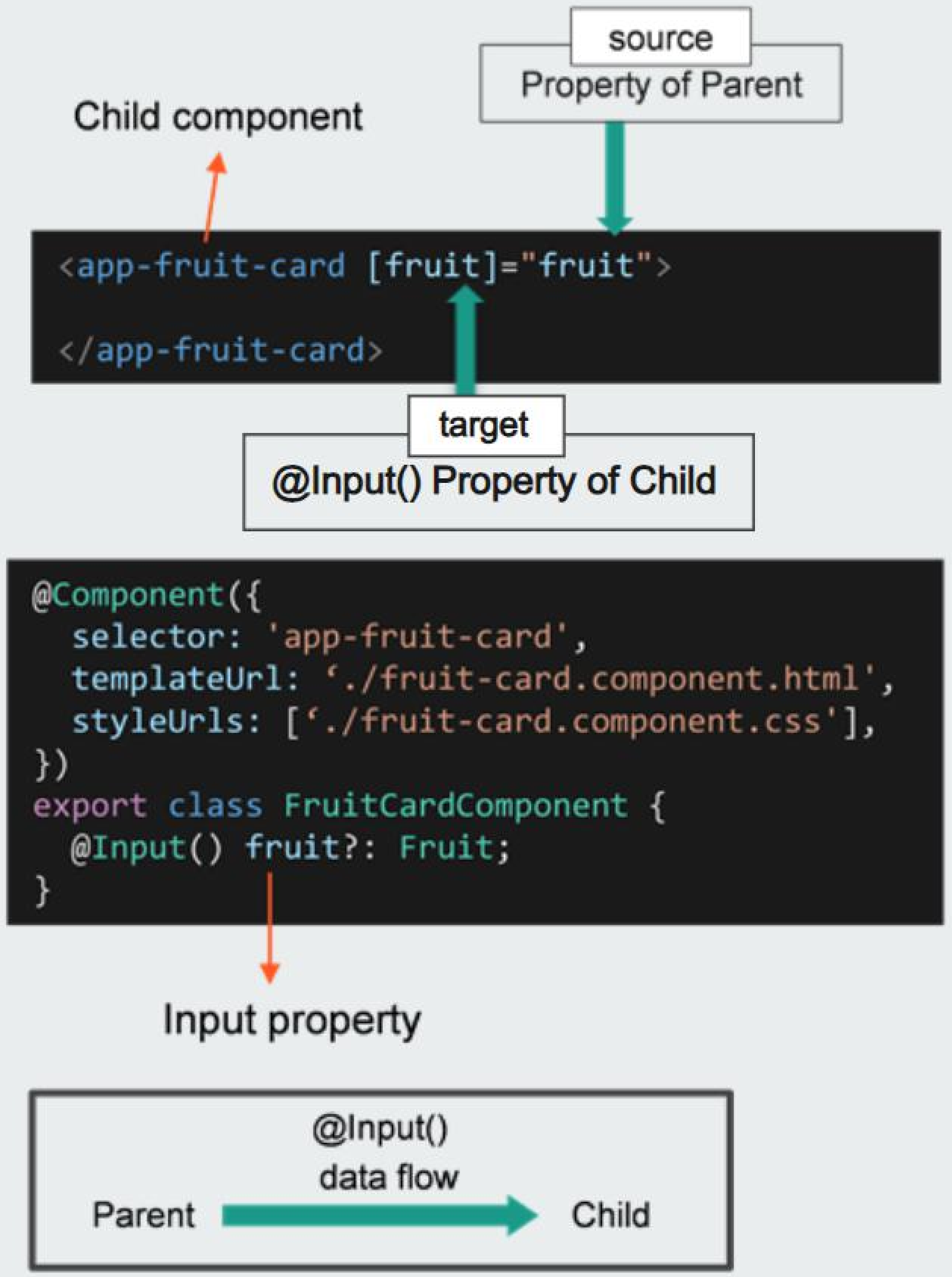
The `Fruit-Manager` should pass this list to the `Fruit-List` component, which will pass one item each to the `Fruit-Card` component to render the fruit details.

Click here for the demo solution.

DEMO

Menu

Slide Note



```
<app-fruit-card [fruit]="fruit">

</app-fruit-card>
```

```
@Component({
  selector: 'app-fruit-card',
  templateUrl: './fruit-card.component.html',
  styleUrls: ['./fruit-card.component.css'],
})
export class FruitCardComponent {
  @Input() fruit?: Fruit;
}
```

@Input() Decorator

- The child component accepts the value for its property from the parent component by annotating the property with the `@Input()` decorator. (Property with `@Input()` is also known as *Input Property*)
- In the snippet shown on the slide, the `app-fruit-card` is the child component selector used by the parent component to render the `Fruit-Card` component.
- While rendering, the parent component passes the value of the fruit to the child component, which is received by the `fruit` property of the child component.
- The `Fruit-Card` component declares the `fruit` property with the `@Input()` decorator to tell Angular that this property will be inputted with the value provided by the parent component.

Menu 00:24 00:00:35 24/ 35

Slide Note

Component Interactions – Child to Parent

- Here are a few instances related to the `Fruit-Fantasy` app that explain the need to enable data flow from child to parent components.

Child (Source)	Data Sent	Parent (Receiver)	Purpose
`Search`	Search text (fruit name)	`Fruit Manager`	The `Search` component notifies `Fruit-Manager` about the search request with the fruit name as the search text. The `Fruit-Manager` can use this input to filter the fruit list.
`Add-Fruit`	`Fruit` object	`Fruit Manager`	The `Add-Fruit` component notifies `Fruit-Manager` with the data about the new fruit being added. The `Fruit-Manager` can update the fruit list array with the fresh fruit item.

Menu

00:25 00:00:35 25/ 35

Slide Note

Think and Tell

- How does a child component notify the parent component about the execution of a task?
- How does a child component pass data to its parent?
- How does a parent component receive data from its child?

Menu

00:26 00:00:35 26/ 35

Slide Note

`Fruit-Fantasy` – Search Fruit in Fruit List

In the `Fruit-Fantasy` app, modify the `Fruit-Manager` component to filter the list of fruits based on the search input received.
The `Fruit-Manager` component should render a `Search` component that accepts user input with the fruit's name to search.
The `Search` component should emit an event with the fruit name and pass the data to the `Fruit-Manager` component.
The `Fruit-Manager` component should handle the event and use the fruit name passed to filter the fruit list by the fruit name.

Click here for the demo solution.

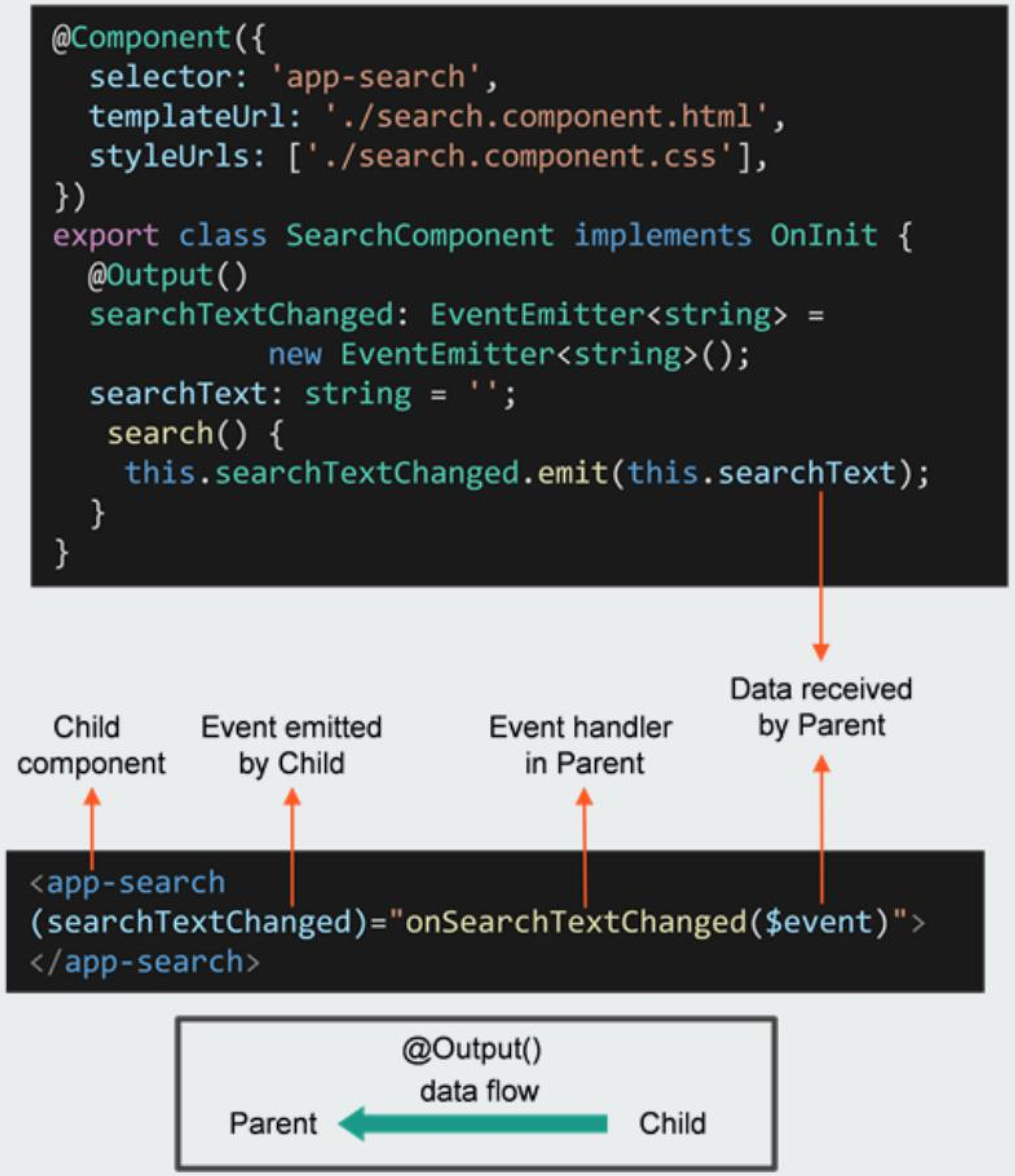
DEMO

Menu

00:27 00:00:35 27/ 35

@Output() Decorator

- The child component emits an event to notify the parent component about the triggered changes and shares the data with the event emitted.
- The event is an `EventEmitter` object declared by the child component by annotating the object with the `@Output()` decorator.
- In the snippet shown on the slide, the `app-search` is the child component selector used by the parent component to render the `Search` component.
- The `Search` component class emits a `searchTextChanged` event with the value of the `searchText` variable.
- The parent component, while rendering the `Search` component, associates the event with the event handler `onSearchTextChanged()` and receives the data passed via the `$event` argument.



```
@Component({
  selector: 'app-search',
  templateUrl: './search.component.html',
  styleUrls: ['./search.component.css'],
})
export class SearchComponent implements OnInit {
  @Output()
  searchTextChanged: EventEmitter<string> =
            new EventEmitter<string>();
  searchText: string = '';
   search() {
    this.searchTextChanged.emit(this.searchText);
  }
}
```

```
<app-search
(searchTextChanged)="onSearchTextChanged($event)">
</app-search>
```

Slide Note

Quick Check

In Angular, `@Input()` is a ________

1. Decorator
2. Component
3. Directive
4. Module

Menu

Slide Note

Quick Check: Solution

In Angular, `@Input()` is a ________

1. Decorator
2. Component
3. Directive
4. Module

Menu

00:30 00:00:35 30/ 35

Slide Note

Quick Check

In Angular, which decorator is used by the child component to use to share data with the parent component?

1. `@Input()`
2. `@Output()`
3. `@EventEmitter()`
4. `@Component()`

Menu

Quick Check: Solution

In Angular, which decorator is used by the child component to use to share data with the parent component?

1. `@Input()`
2. **`@Output()`**
3. `@EventEmitter()`
4. `@Component()`

Slide Note

Fill in the Blanks

- The `ProductList` component contains `products` property which is an array storing list of products. The `ProductList` will share these details with another component to present product details. In this context, the `ProductList` is a ________ component. (Smart/Dumb)
- The `Header` component is rendering the `Navigation` component with navigation links. In the component hierarchy, the `Header` component is the _____ component and the `Navigation` component is the ________ component. (Parent/Child)
- A `Cart` component in an online shopping app displays the product name, quantity requested, and the price of a product. The product details are provided to the `Cart` component by the `CartMaster` component. In this context, the `Cart` component is a _________ component (Smart/Dumb).
- For component A rendering component B, if component A wants to share data with component B, the property that will accept this data will be defined with the ____ (`@Input()/@Output()`) decorator in component ___ (A/B).
- For component A rendering component B, if component B wants to share data with component A, the property that will be emitting an event with this data will be defined with the ____ (`@Input()/@Output()`) decorator in component ___ (A/B).

Menu 00:33:00:00:35 33/ 35

Slide Note

Fill in the Blanks – Solution

- The `ProductList` component contains `products` property which is an array storing list of products. The `ProductList` will share these details with another component to present product details. In this context, the `ProductList` is a **Smart** component.
- The `Header` component is rendering the `Navigation` component with navigation links. In the component hierarchy, the `Header` component is the **Parent** component, and the `Navigation` component is the **Child** component.
- A `Cart` component in an online shopping app displays the product name, quantity requested, and the price of a product. The product details are provided to the `Cart` component by the `CartMaster` component. In this context, the `Cart` component is a **Dumb** component.
- For component A to render component B, if component A wants to share data with component B, the property that will accept this data will be defined with the `@Input()` decorator in component **B**.
- For component A rendering component B, if component B wants to share data with component A, the property that will be emitting an event with this data will be defined with the `@Output()` decorator in component B.

Menu