[Skip to main content](https://courses.funix.edu.vn/courses/course-v1:FUNiX+RJS301x_1.0-A_VN+2022_T7/courseware/7841b6bca89943be83bf49e469c3047b/66f4229127e9484bb0f1e350317faf56/?activate_block_id=block-v1%3AFUNiX%2BRJS301x_1.0-A_VN%2B2022_T7%2Btype%40sequential%2Bblock%4066f4229127e9484bb0f1e350317faf56" \l "main)

**[FUNiX Home Page](https://courses.funix.edu.vn/)**

**FUNiX:** RJS301x\_1.0-A\_VN Lập trình Web Front-End với React

1. **[Dashboard for:[https://courses.funix.edu.vn/static/public.site/images/profiles/default_50.76570a4d025e.png](https://courses.funix.edu.vn/dashboard)](https://courses.funix.edu.vn/dashboard)**

**[vinhdqFX17818](https://courses.funix.edu.vn/dashboard)**

1. More options

[**Bảng điểm**](https://portal.funix.edu.vn/)

[**Help**](https://funix.gitbook.io/funix-documentation/)

1. [Home](https://courses.funix.edu.vn/courses/course-v1:FUNiX+RJS301x_1.0-A_VN+2022_T7/info)

1. [Course, current location](https://courses.funix.edu.vn/courses/course-v1:FUNiX+RJS301x_1.0-A_VN+2022_T7/course/)

1. [Assessment Submission](https://courses.funix.edu.vn/courses/course-v1:FUNiX+RJS301x_1.0-A_VN+2022_T7/3ce3e9029d3a4024adcc67ca081920ef/)

1. [Progress](https://courses.funix.edu.vn/courses/course-v1:FUNiX+RJS301x_1.0-A_VN+2022_T7/progress)

[Course](https://courses.funix.edu.vn/courses/course-v1:FUNiX+RJS301x_1.0-A_VN+2022_T7/course/)  [Phần 1. Làm quen với ReactJS](https://courses.funix.edu.vn/courses/course-v1:FUNiX+RJS301x_1.0-A_VN+2022_T7/course/#block-v1:FUNiX+RJS301x_1.0-A_VN+2022_T7+type@chapter+block@7841b6bca89943be83bf49e469c3047b)  [Assignment 01: Xây dựng trang Web Booking](https://courses.funix.edu.vn/courses/course-v1:FUNiX+RJS301x_1.0-A_VN+2022_T7/course/#block-v1:FUNiX+RJS301x_1.0-A_VN+2022_T7+type@sequential+block@66f4229127e9484bb0f1e350317faf56)  Hướng dẫn hoàn thành dự án

 Previous

1. other Tổng quan dự án
2. other Hướng dẫn hoàn thành dự án
3. other Hướng dẫn nộp bài
4. other Tiêu chí đánh giá

Next

**Hướng dẫn hoàn thành dự án**

 Bookmark this page

**Yêu cầu dự án**

**1. Cài đặt các npm package cho Stater Code**

Ở file Starter code, bạn đã được tạo sẵn các Component như App, Home, Search, Hotel để tạo cấu trúc chung cho trang Web. Bạn có thể tham khảo thêm ở phần "**Tổng quan dự án**" để nắm rõ Starter code.

Để bắt đầu sử dụng được Starter Code, bạn sẽ cần cài đặt các Node Module phục vụ cho dự án. Bạn cần mở Terminal, truy cập vào thư mục Starter Code và chạy câu lệnh sau:

npm install

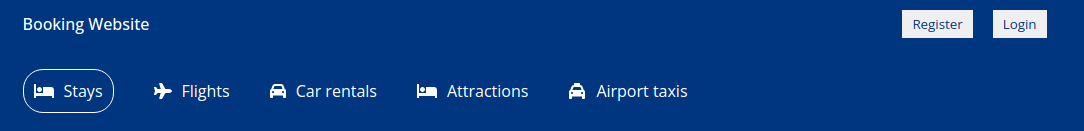
**Lưu ý:** Bạn cần cài đặt trước NodeJS cũng như NPM để hoàn thành yêu cầu này (Bạn có thể tham khảo lai bài Lab 1.1 để biết cách cài đặt và chuẩn bị môi trường).

Sau khi đã hoàn thành, nếu bạn thấy xuất hiện thêm thư mục node\_modules thì tức là bạn đã cài đặt thành công. Lúc này bạn có thể chạy dự án bằng câu lệnh:

npm start

**2. Tạo Component Navbar**

Tiếp theo, bạn sẽ bắt đầu tạo giao diện cho phần HomePage. Đầu tiên sẽ là phần Nav bar với thiết kế như sau:



Bạn sẽ cần tạo 1 Component tên là Navbar và có nhiệm vụ để render giao diện như trên. Để render được danh sách các tùy chọn như thiết kế, bạn sẽ sử dụng dữ liệu từ file **navBar.json** ở Starter Code, file này sẽ là một mảng chứa các phần tử đại diện cho 1 tùy chọn. Mỗi phần tử sẽ có cấu trúc như sau:

**{**

"type": "Stays",

"icon": "fa-bed",

"active": true,

**}**

Các thuộc tính mang ý nghĩa như sau:

* **type**: loại lựa chọn hay tên của item đó.
* **icon**: icon tương ứng với item.
* **active**: nếu là true thì bạn sẽ thêm border vào item đó, ngược lại thì không cần thêm border.

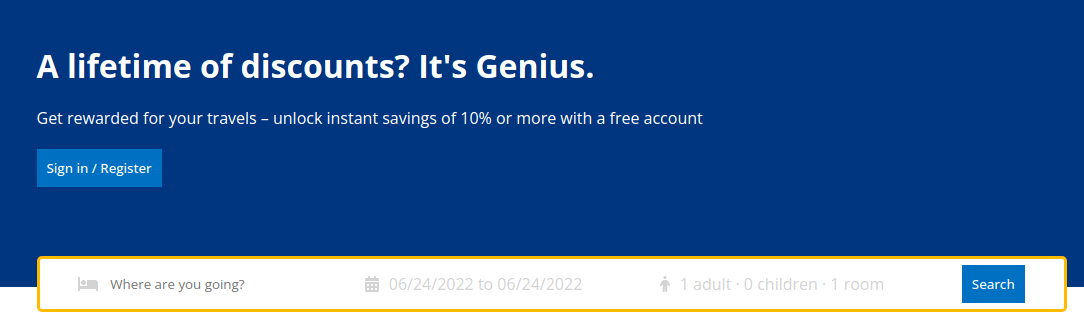
Để sử dụng các dữ liệu dạng file JSON này, bạn sẽ copy các dữ liệu trong file vào thẳng Component đó để thành một dữ liệu mẫu. Để sử dụng được các dữ liệu về icon, bạn có thể tham khảo lại các tạo và sử dụng **Font Awesome** cho Website.

Bạn sẽ cần sử dụng kiến thức về **Rendering Lists & Conditional Content** để có thể render ra theo đúng yêu cầu. Bạn có thể tạo một Component con gọi là **NavBarItem**, Component này sẽ có các Props để có thể truyền vào dữ liệu, sau đó bạn sẽ render Item dựa trên Props đã truyền vào.

Sau khi đã tạo được Component, bạn sẽ thêm Component này vào **Home.jsx** để có thể hiển thị ra khi người dùng truy cập vào trang chủ.

**3.Tạo Component Header cho Homepage**

Tiếp theo, bạn sẽ tạo Component Header cho Homepage với thiết kế như sau:



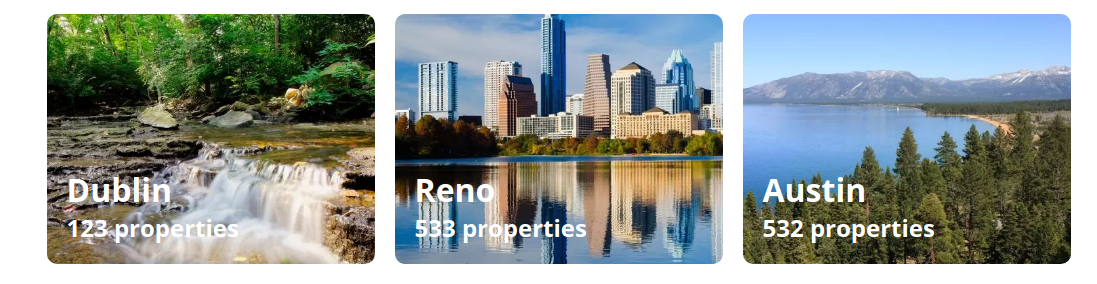
Bạn cần chú ý một số điển như sau:

* Các form nhập liệu sẽ sử dụng các thẻ **<input>** trong HTML.
* Khi click vào nút **Search** thì bạn sẽ được chuyển hướng sang trang **/seach** để hiển thị trang **Seach Page**. Để hoàn thành được yêu cầu này, bạn sẽ cần sử dụng [**onClick**](https://reactjs.org/docs/handling-events.html) trong React và[**window.location.replace**](https://www.w3schools.com/jsref/met_loc_replace.asp) để chuyển hướng do trang web.
* Các icon, bạn có thể tìm icon phù hợp ở trang **icon-list.html** trong Starter Code.

**4. Hiển thị danh sách trên trang Homepage**

Tiếp theo, bạn sẽ hiển thị danh sách trong trang Homepage với các thiết kế như sau.

a. Hiển thị các thành phố



Bạn sẽ render dựa trên các Dummy Data ở trong file **city.json.**File JSON này sẽ chứa mảng gồm các phần tử như sau:

**{**

"name"**:** "Dublin"**,**

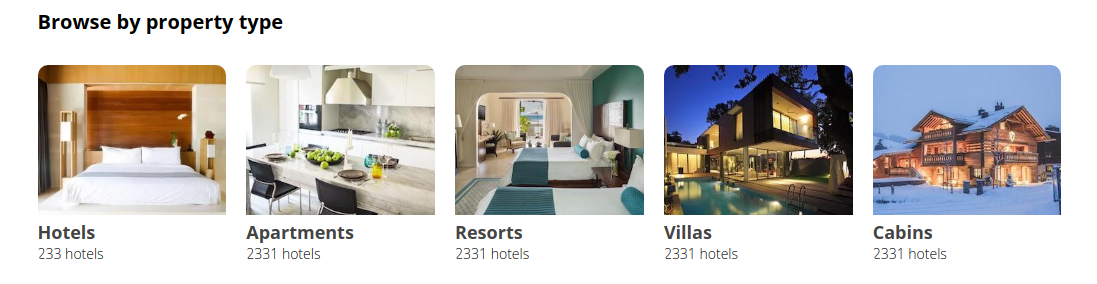
"subText"**:** "123 properties"**,**

"image"**:** "./images/city\_1.webp"**,**

**}**

b. Hiển thị các loại khách sạn

Sau đó, bạn sẽ hiển thị danh sách các loại khách sạn của hệ thống với thiết kế như sau



Bạn sẽ cần sử dụng dữ liệu từ file **type.json**, file này là một mảng với các dữ liệu có cấu trúc như sau:

**{**

"name"**:** "Hotels"**,**

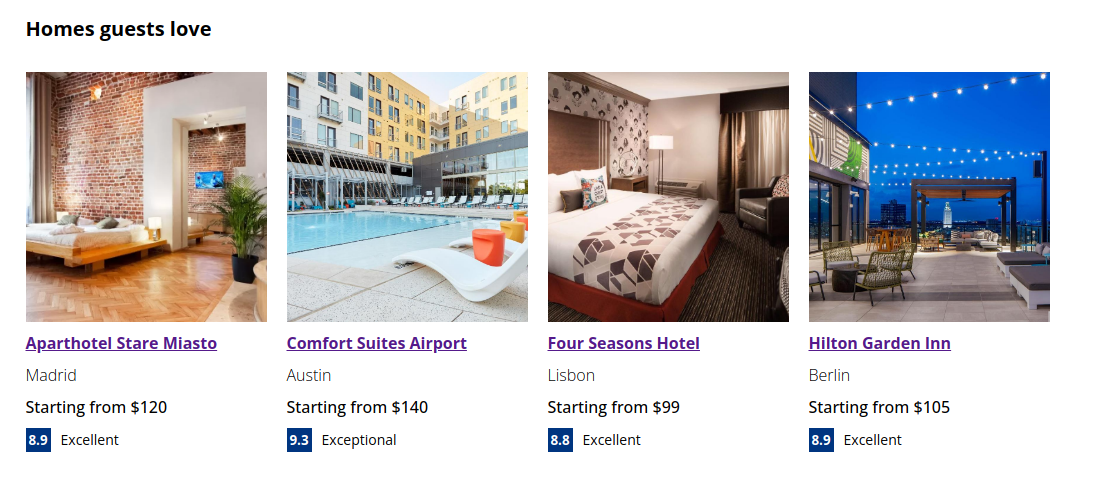
"count"**:** **233,**

"image"**:** "./images/type\_1.webp"**,**

**}**

c. Hiển thị các khách sạn

Tiếp theo sẽ là danh sách các khách sạn và thông tin cụ thể. Thiết kế sẽ như sau:



Bạn sẽ sử dụng dữ liệu ở file **hotel\_list.json** với các dữ liệu có cấu trúc như sau:

**{**

"name"**:** "Aparthotel Stare Miasto"**,**

"city"**:** "Madrid"**,**

"price"**:** **120,**

"rate"**:** **8.9,**

"type"**:** "Excellent"**,**

"image\_url"**:** "./images/hotel\_1.webp"

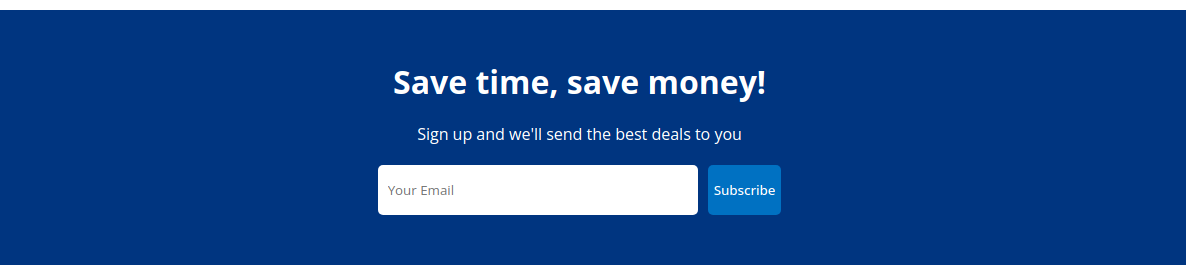
**}**

**Lưu ý:** Khi click vào tên của từng khách sạn thì người dùng sẽ được chuyển hướng để trang**Detail Page** để xem thông tin đầy đủ của khách sạn đó (đường link là **/detail**).

Mỗi loại danh sách sẽ được render bởi một Component độc lập. Bạn sẽ cần sử dụng kiến thức về **Rendering Lists & Conditional Content** để có thể render ra theo đúng yêu cầu.

**5. Tạo phần form đăng ký**

Tiếp theo, bạn sẽ tạo Component để hiển thị phần Form đăng ký với thiết kế như sau:

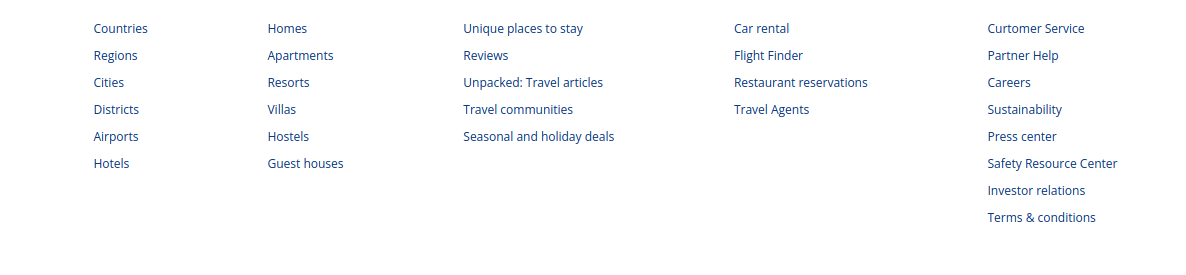


**Lưu ý:**

* Nội dung của phần này sẽ được căn giữa.
* Để tạo các Form nhập liệu, bạn cần sử dụng các thẻ **<input>** như trong HTML.

**6. Tạo Footer cho Homepage**

Tiếp theo, bạn sẽ tạo Component Footer để render phần chân trang với thiết kế như sau:

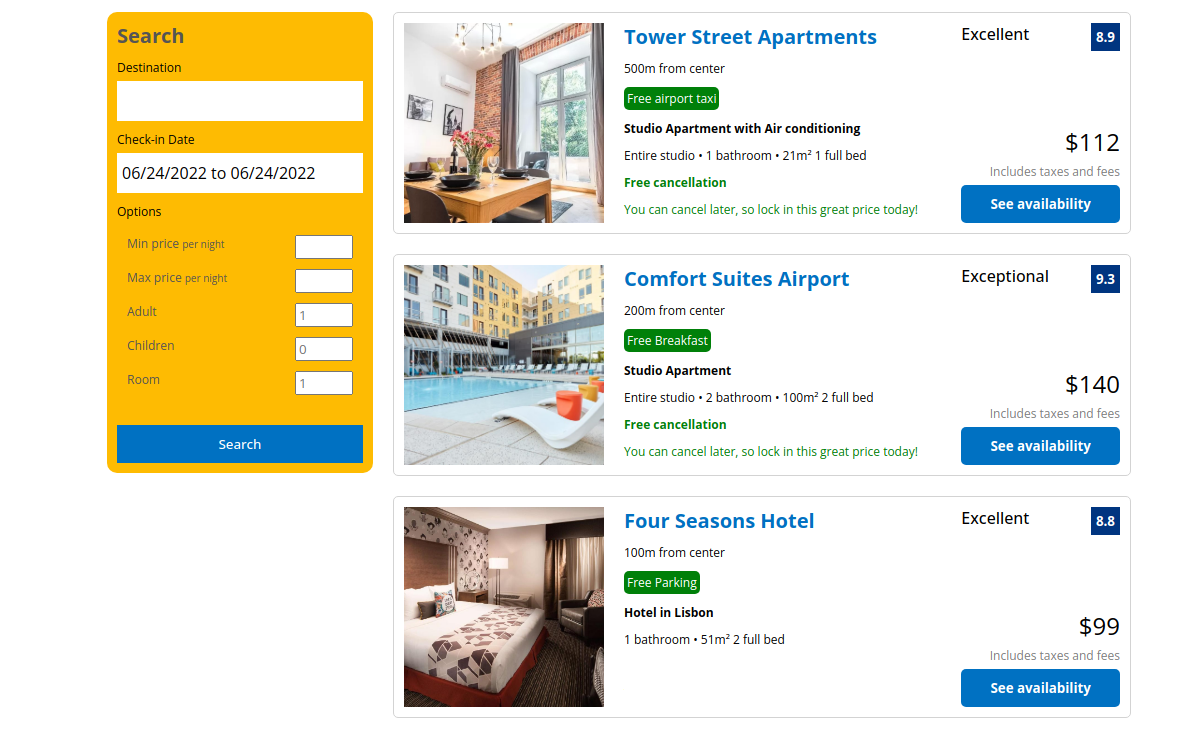


Để hoàn thành yêu cầu này, bạn sẽ render dựa theo dữ liệu từ file **footer.json**. FIle này sẽ bao gồm các phần tử, mỗi phần tử là dữ liệu của một ô trong Footer.

**7. Hiển thị danh sách trên trang Search Page**

Tiếp theo, bạn sẽ tạo một Page mới, đó là **Search Page** khi người dùng truy cập vào trong **/search** (để có thể chỉnh sửa được Page này, bạn sẽ sửa các đoạn code trong file Search.jsx trong Starter Code).

Phần Search Page sẽ có thể tận dụng lại được một số Component từ Home Page như NavBar hay Footer. Bạn sẽ cần tập trung vào việc phát triển nội dung của Page với thiết kế như sau:



Bạn sẽ cần tạo 2 Component con như sau:

* **SearchPopup**: Component để render phần popup nằm bên trái của Page.
* **SearchList**: Component để render các kết quả của việc tìm kiếm nằm ở bên phải của Page.

Ở Component **SearchList**, bạn sẽ render các kết quả dựa theo dữ liệu từ file **search.json**. File này sẽ chứa dữ liệu về các kết quả của việc tìm kiếm theo cấu trúc như sau:

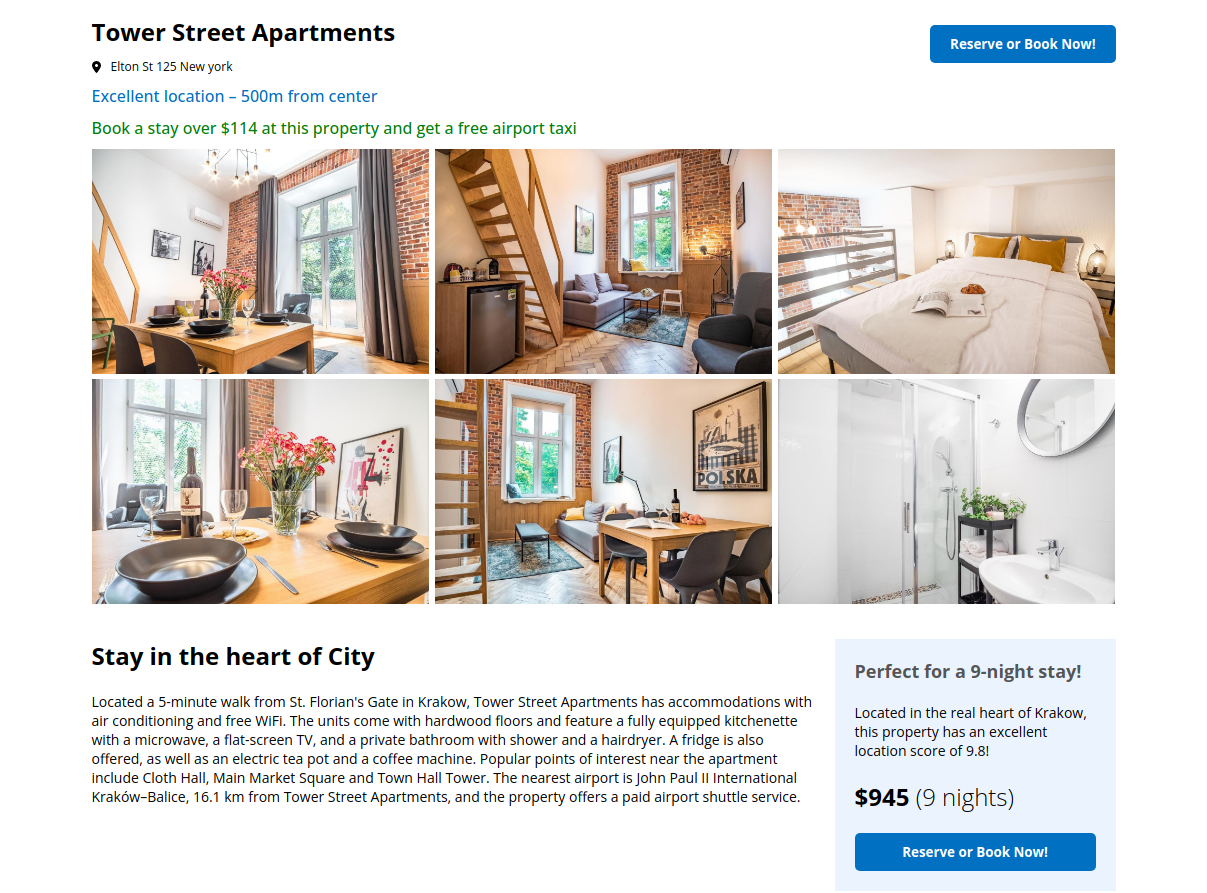
* **name:** tên khách sạn.
* **distance**: vị trí của khách sạn từ trung tâm.
* **tag:** tag của khách sạn để phân loại.
* **type**: loại khách sạn.
* **description**: thông tin về khách sạn.
* **free\_cancel:** là giá trị Boolean. Nếu bằng true thì sẽ được Cancel, ngược lại thì không.
* **price:** giá của khách sạn.
* **rate:** điểm đánh giá ở dạng số (trên thang 10.0).
* **rate\_text:** đánh giá của khách sạn ở dạng text.
* **image\_url**: link ảnh của khách sạn.

Sau đó bạn cần tạo thêm 1 Component con là **SearchListItem** gồm các Props với dữ liệu như trên để bắt đầu hiển thị.

**8. Xem chi tiết thông tin của khách sạn**

Cuối cùng, bạn sẽ tạo thêm 1 trang nữa để hiển thị thông tin cụ thể của 1 khách sạn. Bạn sẽ cần chỉnh sửa code ở file **Detail.jsx** để chỉnh sửa cho trang này.

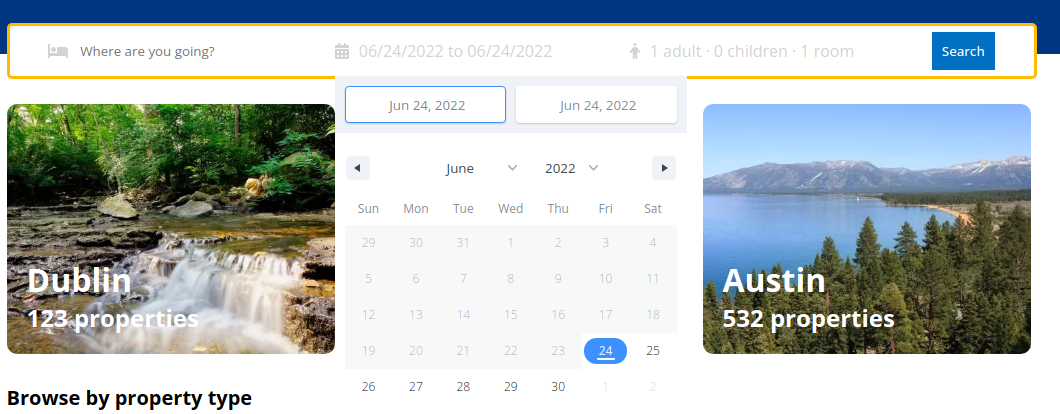
Phần Detail Page sẽ có thể tận dụng lại được một số Component từ Home Page như NavBar, Contact hay Footer. Bạn sẽ cần tập chung vào việc phát triển nội dung của Page với thiết kế như sau:



Dữ liệu về trang Detail bạn có thể tìm thấy ở file **detail.json**.

**9. (Nâng cao) Sử dụng Date Picker cho phần Search**

Ở yêu cầu này, bạn sẽ nâng cấp cho phần tìm kiếm. Nếu như ở yêu cầu cơ bản, phần tìm kiếm chỉ là nhập Text vào các ô ngày tháng, bạn sẽ sử dụng thư viện [**react-date-range**](https://www.npmjs.com/package/react-date-range)để hiển thị popup lựa chọn ngày tháng chuyên nghiệp hơn (thư viện này đã được cài đặt sẵn khi bạn sử dụng Starter Code).



Khi click vào phần chọn ngày tháng, bạn sẽ hiển thị ra một Modal để người dùng lựa chọn, sau đó sẽ lưu giá trị đó lại vào input. Bạn sẽ cần các kiến thức sau để hoàn thành yêu cầu:

* + Kiến thức về **State** của Component để thực hiện việc ẩn, hiện Modal. Bạn sẽ sử dụng [**useState**](https://reactjs.org/docs/hooks-state.html) hook để thực hiện phần này.
  + Sử dụng **DateRange** trong thư viện **react-date-range** để tạo modal. Bạn sẽ cần import Component đó với câu lệnh như sau:
  + **import** **{** DateRange **}** from "react-date-range"**;**

Sau đó, để thêm Component này vào trang Web, bạn sẽ sử dụng câu lệnh như sau

**<**DateRange

editableDateInputs**={true}**

moveRangeOnFirstSelection**={false}**

className**=**"date"

minDate**={new** Date**()}**

onChange**={...}**

ranges**={...}**

**/>**

Lưu ý: Bạn sẽ cần điền vào giá trị tương ứng ở các trường onChange và ranges, hãy đọc lại tài liệu hướng dẫn ở [**link sau**](https://www.npmjs.com/package/react-date-range) để biết giá trị phù hợp.

* Sử dụng [**onClick**](https://reactjs.org/docs/handling-events.html) để bắt sự kiện khi người dùng click vào ô nhập ngày tháng.

 PreviousNext

**About FUNiX**

[FUNiX LMS Home Page](https://courses.funix.edu.vn/)

* [About](https://funix.edu.vn/)

* [Blog](https://funix.edu.vn/blog/)

[](http://open.edx.org/)

**Báo lỗi**

Ask Mentors

Question nowHistory