

Using rightmove.co.uk as an example: Create a client-side web application that enables a user to search properties by:

- type (house, flat, any),
- price (min price, max price)
- bedrooms (min bedroom, max bedroom)
- date added (after specified date or between two given dates)
- postcode area (1st part of the postcode, e.g. BR1, NW1)

The form for the user to enter their search criteria should be written in HTML5 and make use of React JS UI components/lifecycles, libraries/widgets to ensure consistency, reliability, and accessibility.

The search criteria will be used to search properties in a JSON format. A JSON file has been created for you to contain 2 properties. You will add an additional 5 properties to cover a range of properties (type, price, number of bedrooms and date added, postcode area), to allow the search to return a range of results. (5%). You will need to incorporate your JSON data into your Frontend as we are not using a server.

Make use of React Widgets on your form to enhance ALL your form elements. (10%)

Display the properties that correspond to the search made by the user on the search page. 10% for search functionality – 10% for the displaying the results in an effective and pleasant way (picture, short description, price) **(20%)**

Note:

- the search should work for searching using any number of criteria (i.e. 1 criteria only, 2 simultaneous criteria, 3, 4 or all 5)

- if you cannot get the search functionality working, still provide a list of properties in an effective and pleasant way to be able to get the marks associated with the layout and display of these results.

Each property listed on the result page should have a link that allows the user to visit the corresponding property page.

The property page should contain a large image, thumbnail images, and a short description of the property (property type, price, location). Using your own code or a plugin of your choice, add the facility to view all the property images. **(10%)** You should have 6 to 8 pictures for each property.

Using react tabs, display the property long description, the floor plan and the google map. **(10%)**

Add the facility to save the property to your favourite list of properties by dragging the property to a favourites list on the side(5) **AND** by pressing a “favourite” button or icon(5). (10% to ensure each property can only be added once. 5% otherwise). **(10%)**

The user should also be able to remove a property from the favourite list (by dragging it out of the list (5) as well as by pressing a delete button(3)) and clear the favourite list (2). **(10%)**

Add the facility to display your favourite property list on the search page. **(5%)**

Make your search page and your results page responsive by changing the layout on smaller devices, using hand-written media queries as well as **flex or grid layout**. If you are using a framework such as Bootstrap, then you might not need to use the media queries, but you should be able to tell how it is being implemented if asked during viva. You are required to have two layouts (defined by your hand-written media query or bootstrap): large screen and smaller than iPad landscape width for at least the search page with the results, so covering the form section as well as the results section (4) and the individual property details (4). You will get another 2 marks for your solution justification **(10%)**

Pay attention to Aesthetics, as described below: **(5%)**

- the inclusion of visual clues based on groupings; related items or links are grouped together while unrelated items are separated.
- use of headings and subheadings to allow visual scanning of content.
- use of headings, subheadings, font sizes, bold fonts and italic fonts in proportion to the importance of the item.

- alignment of elements on a page so that they are all visually connected; size all elements on the page to create balance and unity; nothing should look out of place unless you have a specific reason for the effect.

- choose a font style that supports the site atmosphere and stick to it; limit styles to 2 at most.

- use images and photos for visual appeal and to communicate ideas.

- use one set of design elements across your website.

- direct the user's attention to the important information

- harmony and balance

Ensure your site is protected against client-side hacking through the use of CSP, html encoding, etc. **(5%)**

Clarification: You should have a total of 7 properties. You are not permitted to use existing templates. You are not permitted to

use any server-side solution. Your solution must work in a browser without the use of a server, and this is how it will be tested. You may however specify which browser to use to test the solution if the solution works better/looks better in a particular browser but try to achieve consistency across multiple browsers.

Ensure good coding standard is used throughout using good comments, appropriate indentation, and correctness. **Up to 10 marks will be deducted** for lack of comments and incorrectness. Ensure to use any standards that are better for proposed work but do not use a readymade solution for this.

Rubric:

Each section will be graded based on the criteria outlined below, with final scoring determined by the student's ability to explain their approach and code during the viva. Marks may be adjusted if students are unable to adequately explain their solutions.

1. JSON Data - You will add an additional 5 properties to cover a range of types, price, size etc), to allow the search to return a range of results. 1 mark for each property that covers the range.

2. React UI - Make use of React UI widgets on your form to enhance all your form elements (2 for each of: type, size, price, date, and search button, to a max of 10%).

3. Search Functionality - Retrieve the properties that correspond to the search made by the user on the search page. 10% for search functionality over more than one criteria. 5% for the ability to search ONLY 1 criteria at a time.

4. Search facility - displaying the results in an effective and pleasant way (picture, short description, price)

5. The properties page - Image plugin - Using your own code or a plugin of your choice, add the facility to view all the property images. (10%) You should have 5 pictures for each property

6. The properties page - React UI Tabs - Using React UI tabs/depends on the solution (different students might create these components differently e.g. react -tabs, react-bootstrap or taking inspiration from the demo project) (4), display the long description(2), the floor plan (2) and the google map (2) (Total 10%)

7. Save to local storage - Add the facility to save the property to your local browser storage and favourite list of properties on the page bydragging the property to a favourites list on the side(5), as well as by pressing a "favourite" icon or save button (5).

8. Remove from / clear local storage - The user should also be able to remove a property from the favourite list (by dragging it out of the list (5) or pressing a delete button(3)) and clear the favourite list (2). (10%)

9. Read from local storage - Add the facility to display your favourite property list on the search component. (5%)

10.RWD - Make your search page section, the results section and your property components responsive by changing the layout on smaller devices, using hand-written media queries as well as flex or grid layout.

You are required to have two layouts: large screen and smaller than iPad landscape width for the search section (2%) results section (2%) and individual property pages (4%). Media queries that would be present by default with frameworks used by the student, e.g. Bootstrap, does count. You can also justify your choice (2%).

11. Aesthetics - will be evaluated for a total of 5%: 2% points for clear visual hierarchy, 1% point for proper alignment and grouping, 1% point for a consistent color scheme, and 1% point for effective use of visual elements.

12. Security (5%) - Ensure your site is protected against client-side hacking using CSP (this is normally done on server side so student should be able to explain this), html encoding (in react app everything should be enclosed in JSX {} when rendering), etc. 2.5% each.

Notes:

All work must be demonstrated in the viva where you will receive feedback and an indicative mark. The viva is a mandatory component of the coursework, and students are required to answer the questions posed during the session (will be scheduled after the submission deadline).

This allows students to demonstrate that they have completed the work independently and understand the functionality of their code.

Work marked without a viva shall receive between 0-30% maximum.

This is individual work – do not collaborate.