University of Westminster
School of Electronics and Computer Science

5COSC026W Advanced Client-Side Web Development 2025/26)

Module leader	Ebad Majeed
Unit	Final Coursework
Weighting:	60%
Qualifying mark	30%
Description	Responsive Web Application (SPA)
Learning Outcomes Covered in this Assignment:	Assessed coursework is a highly interactive/responsive website that incorporates the features and techniques that are demonstrated by the syllabus coupled with a viva, during which the students will demonstrate a thorough knowledge of issues related to the solution (LO1 - 5).
Handed Out:	**06th November 2025**
Due Date	**05th January 2026**
Expected deliverables	One Zip Electronic file containing all necessary HTML, JS and CSS files, GitHub repo link and URL of deployed version Once you are done with your project, remove the node modules folder but make sure that package.json is present and then zip it and upload.
Method of Submission:	**Electronic submission on BB via a provided link close to the submission time.**
Type of Feedback and Due Date:	Verbal feedback and marks on the viva date, up to 2 week after submission All marks will remain provisional until formally agreed by an Assessment Board.
BCS Criteria meeting in this assignment	2.1.1 Knowledge and understanding of facts, concepts, principles & theories 2.1.2 Use of such knowledge in modelling and design 2.1.3 Problem solving strategies 2.1.9 Knowledge of information security issues 2.2.1 Specify, design or construct computer-based systems

Assessment regulations

Refer to section 4 of the “How you study” guide for undergraduate students for a clarification of how you are assessed, penalties and late submissions, what constitutes plagiarism etc.

Penalty for Late Submission

If you submit your coursework late but within 24 hours or one working day of the specified deadline, 10 marks will be deducted from the final mark, as a penalty for late submission, except for work which obtains a mark in the range 40 – 49%, in which case the mark will be capped at the pass mark (40%). If you submit your coursework more than 24 hours or more than one working day after the specified deadline you will be given a mark of zero for the work in question unless a claim of Mitigating Circumstances has been submitted and accepted as valid.

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website: http://www.westminster.ac.uk/study/current-students/resources/academic-regulations

5COSC026W Advanced Client-side Web Development
Final Coursework
Estate Agent Client-side Web Application

Set Date: 06th November 2025
Due date: 05th January 2026 1pm on Blackboard,
Viva date TBC

Objective: 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.

Core Requirements:

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. (4%) 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. (8%)

Display the properties that correspond to the search made by the user on the search page. 10% for search functionality – 7% for displaying the results in an effective and pleasant way (picture, short description, price) (17% total)

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. (5%)

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. (7%)

Add the facility to save the property to your favourite list of properties by dragging the property to a favourites list on the side AND by pressing a "favourite" button or icon. (8% total, with marks allocated to ensure each property can only be added once)

The user should also be able to remove a property from the favourite list (by dragging it out of the list as well as by pressing a delete button) and clear the favourite list. (7%)

Add the facility to display your favourite property list on the search page. (3%)
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. You are required to have two layouts: large screen and smaller than iPad landscape width for at least the search page with the results and the individual property details. You will get marks for your solution justification. (8%)
Pay attention to Aesthetics, as described below: (4%)

- 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
- 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. (3%)

- **GitHub & Version Control:** You must host your code on GitHub with a clear, logical commit history demonstrating incremental development. (5%)
- **Deployment:** The application must be deployed to a live URL using GitHub Pages or another cloud service. (5%)
- **Testing:** Implement a minimum of five (5) meaningful tests using the JEST testing framework. (12%)

Technical Constraints & Standards:
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.

Ensure good coding standard is used throughout using good comments, appropriate indentation, and correctness. (4%)

Submission & Viva:
You must submit your GitHub repository URL and the live application URL. All work will be demonstrated in a viva, which is mandatory. Failure to attend the viva would result in a minimum of 0% or maximum mark of 30%. This is individual work.

Mark Distribution Summary:

- JSON Data: 4%
- React UI Widgets: 8%
- Search Functionality: 10%
- Results Display: 7%
- Property Page Gallery: 5%
- Property Page Tabs: 7%
- Add to Favourites: 8%
- Remove/Clear Favourites: 7%
- Display Favourites: 3%
- Responsive Design: 8%
- Aesthetics: 4%
- Security: 3%
- Code Quality: 4%
- GitHub & Commits: 5%
- Deployment: 5%
- JEST Testing: 12%
- **Total: 100%**

Detailed Rubric (will be added on blackboard):

1. JSON Data (4%)

- **Fail (0 - 1): Fewer than 4 properties added, or data is completely unsuitable for searching.**
- **Pass (2): 5 properties added, but with limited variety that makes some search criteria ineffective.**
- **Good (3): 5 properties added with good variety, covering most search criteria effectively.**
- **Distinction (4): 5 diverse properties expertly cover all search types, prices, bedrooms, dates, and postcodes.**

2. React UI Widgets (8%)

- **Fail (0-1): No React Widgets used on the form.**
- **Pass (2-4): Approximately half of the form elements use enhanced widgets with basic implementation.**
- **Good (5-6): Most form elements use appropriate React Widgets, significantly improving user experience.**

- **Distinction (7-8): All form elements use enhanced, appropriate React Widgets creating superior user experience.**

3. Search Functionality (10%)

- **Fail (0-2): Search functionality is completely broken or non-functional.**
- **Pass (3-5): Search works but only with single criteria, or has major bugs when combining criteria.**
- **Good (6-8): Search works well with multiple criteria but may have minor, non-critical bugs.**
- **Distinction (9-10): Search works flawlessly with any combination of 1 to 5 criteria with robust logic.**

4. Results Display (7%)

- **Fail (0-1): Results are poorly structured, difficult to read, or missing key information.**
- **Pass (2-3): Results displayed functionally but lack visual appeal with poor layout.**
- **Good (4-5): Results displayed clearly and pleasantly with images, descriptions, and prices.**
- **Distinction (6-7): Highly effective, visually pleasing professional layout that enhances usability.**

5. Property Page - Gallery (5%)

- **Fail (0): No functional image gallery implemented.**
- **Pass (1-2): Basic gallery implemented but clunky or has fewer than 6 images.**
- **Good (3-4): Gallery works well with 6-8 images but lacks advanced features.**
- **Distinction (5): Fully functional, intuitive gallery with thumbnails and smooth navigation.**

6. Property Page - Tabs (7%)

- **Fail (0): Tabs not used to organize content.**
- **Pass (1-3): Tabs implemented but buggy or missing required sections.**
- **Good (4-5): Tabs work well to organize description, floor plan, and Google Map.**
- **Distinction (6-7): Flawless tab implementation creating seamless user interface.**

7. Add to Favourites (8%)

- **Fail (0-1): Favourites feature completely broken.**
- **Pass (2-4): Only one method works reliably, or duplicates not prevented.**
- **Good (5-6): Both methods work but with minor issues in duplicate prevention.**
- **Distinction (7-8): Both methods work perfectly with robust duplicate prevention.**

8. Remove/Clear Favourites (7%)

- **Fail (0-1): Removal features completely broken.**
- **Pass (2-3): Only one removal method and clear list work reliably.**
- **Good (4-5): All removal functions work but one method may be less polished.**
- **Distinction (6-7): All removal methods work perfectly with excellent user experience.**

9. Display Favourites (3%)

- **Fail (0): Favourites list not displayed on search page.**
- **Pass (1): Favourites list displayed but basic with minor issues.**
- **Good (2): Favourites list functional and updates as expected.**
- **Distinction (3): Well-integrated, real-time updating list prominently displayed.**

10. Responsive Design (8%)

- **Fail (0-1): Site not responsive on different screen sizes.**
- **Pass (2-4): Basic responsiveness attempted but inconsistent or broken.**
- **Good (5-6): Site responsive across all pages with two distinct layouts.**
- **Distinction (7-8): Fully responsive with polished layouts and clear technical justification.**

11. Aesthetics (4%)

- **Fail (0): Design messy, inconsistent, and difficult to use.**
- **Pass (1-2): Functional but bland design lacking visual hierarchy.**
- **Good (3): Pleasant, consistent design with good visual hierarchy.**
- **Distinction (4): Professional, cohesive design creating engaging user experience.**

12. Security (3%)

- **Fail (0): No client-side security measures implemented.**
- **Pass (1): Basic security attempt made but incorrect or incomplete.**
- **Good (2): One security measure well-implemented and functional.**
- **Distinction (3): Both CSP and JSX encoding correctly implemented and explainable.**

13. Code Quality (4%)

- **Fail (0): Code messy, difficult to follow, and uncommented.**
- **Pass (1-2): Functional but poorly formatted with insufficient comments.**
- **Good (3): Readable, well-structured with adequate comments.**
- **Distinction (4): Exceptionally well-structured, formatted, and commented code.**

14. GitHub & Commits (5%)

- **Fail (0-1): No repository or single massive commit history.**
- **Pass (2): Sparse history with large commits and poor messages.**

- **Good (3-4): Good history with frequent, descriptive commit messages.**
- **Distinction (5): Professional repository with clear, logical progression and excellent commits.**

15. Deployment (5%)

- **Fail (0): No deployment or completely broken live site.**
- **Pass (1-2): Deployed but with significant functional issues.**
- **Good (3-4): Deployed and mostly functional with minor differences.**
- **Distinction (5): Successfully deployed, fully functional, and mirrors repository.**

16. JEST Testing (12%)

- **Fail (0-2): 0-1 tests implemented or all tests failing.**
- **Pass (3-5): 2-4 basic tests covering simple functionality.**
- **Good (6-9): 5 tests passing but not covering critical application logic.**
- **Distinction (10-12): 5+ meaningful tests covering critical functions with good coverage.**

Total: 100%

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.