Collaborative Design Approach

Technical Challenges/Considerations:

1. ***Content architecture approach:*** The design shows distinct sections like "About Us", “Our Services", and "Our Team", each with its own layout and content. This calls for a decision on how content will be managed in the backend, either with Custom Post Types (CPTs) with ACF field groups or by using Gutenberg block-based editing. A CPT + ACF approach gives more flexibility and scalability. It enables data consistency on theme changes. On the contrary, the Gutenberg/block-based setup gives editors more hands-on control over layout and flow. We may need to clarify this with the client.

2. ***Data Modelling:*** If we were to prefer the CPT-cased approach, then each content section would need its own Custom Post Type.This also shows the need for consistent modelling. Especially if any fields may be connected, example, there may be a relationship between the team members and the services they offer and we may want to use that.

3. ***Block styling:*** If we are using a block-based approach, then we need to ensure the styling for all blocks are consistent. Maintaining a well structured design could be a challenge in this case, especially if non-technical editors will be managing content.

4. ***Map Embeddings:*** We can see that the map is embedded twice, in the “About us” and before the “Latest News” section. Hence, we need to manage our API quota by avoiding loading twice.

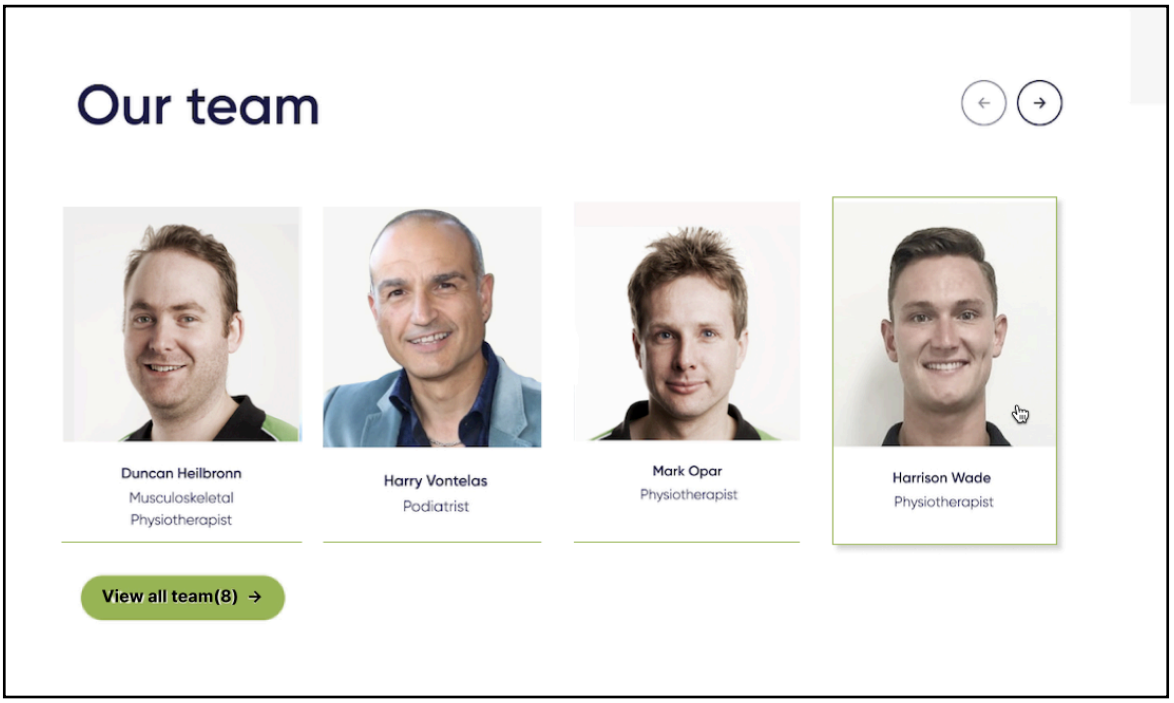
5. ***Data Loading for Carousels:*** The carousels show the need to consider data loading strategies to keep up efficiency. Example, for “Services” and “Team”, we may want to approach it in a different way as compared to the “Latest News” section as the amount of data in “Latest News” may grow over time. Hence, methods like pagination may be helpful. But for cases where there is not a lot of data (like the number of services offered), it may be an overkill and can lead to multiple API calls, which can overload our database with queries.

UI/UX and Accessibility Issues:

1. ***Button styling:*** The buttons need more consistency in their design. Example, the “Book a session” is coloured light green with white text in the navigation menu, but is white with pink text later in the page. Another issue here could be the button to text colour contrast used. We can consider a higher contrast pair for button and text. This could cause readability issues.

2. ***Section headings:*** The headings for different sections lack a consistent sizing. The “About us” section heading has a smaller font-size as compared to “Our Services” and “Our Team” where the headings are larger.

3. ***Carousel:*** The horizontal scroll for the carousel looks a bit confusing. It may be much simpler to allow users to either click the arrows or have the progress icons. It can be misleading from a user point-of-view to know where to click. It is also unclear if the items in the the carousel are clickable or not, maybe adding a highlight or some basic effect using simple JS on hovering over them could help. Using a slight margin between images in the “Our Team” section images would be helpful as done in the “Our Services” section above. Images in these sections have different aspect ratios as well. Hence, we need to ensure these are ratios are responsive in different screen sizes.



A **possible improvement** to the issues highlighted in points 3. This design keeps up the simplistic approach used initially while giving sufficient feedback to users and avoiding potential accessibility issues. We can highlight on hovering over the item if it’s clickable and have the scroll element to blur out, again, providing feedback to users.Another thing to note is the colour contrast for the button. A similar approach could be used for to other carousels as well for ensuring consistency and accessibility.

Efficiency In Build:

1. ***Minimise plugin usage:*** For projects like this where the pages simply display information, it is better if we use simple plugins or have our own custom plugins. This will reduce complexity and ease maintainability.
2. ***Singular Data Loading:*** Using a centralised approach to load and store repetitive details like contact details and map instances, example, an ACF Options page will help in cases like getting the data for the “Hero” and “Contact” sections.
3. ***Lazy-Loading:*** We can use techniques like lazy-loading or pagination in parts which could slow down our page.
4. ***Future-proofing:*** Use of plugins like Wordfence with auto-updates on would be helpful for avoiding future breaches. A common reason for sites/components to break is dependency updates. We can prevent this by regularly updating our themes and plugins.