# Task One

**Could the client get the same or better website, using a different way of presenting this data?** **What alternatives, if any, would you suggest, and why?**

An alternative approach I would take would be to shrink the carousel a touch but keep it as the central feature piece. Remove the blue overlap at the bottom and use the new space underneath to dynamically show some products from the range you’re currently viewing on the carousel. If you were to use a framework such as Angular or React to design this website, you could create components to house these new features which you could easily feed in data from the backend.

# Task Two

**How would you connect to the data source?**

Because of my background in Node.js I would normally establish a connection to a database like MongoDB using it’s package, create some asynchronous functions to handle the general CRUD functionality and present those as easy to use API endpoints for interacting from the frontend, you can add some authentication to these endpoints using JWT to protect your database.

**How would you structure the query?**

In order to find the 3 most recent blog posts I'd create a query that sorts the blog collection by date in descending order, then set a limit to the number of returns to 3 to reduce the cost of the search. Manipulating the return results by using projection I can obtain only the relevant fields.

Such a query could look like:

query = {};

options = {

sort: { date: -1 },

projection: { \_id: 0, title: 1, thumbnail: 1, date: 1 }

}

cursor = collection.find(query, options).limit(3);

*It’s worth noting using this way of querying it’s easy to pass these options from the frontend*

**How would you transform the data into an operable format?**

Looping through the cursor using forEach, I could add the resulting objects of the query to an array, JSON stringify that array and send it back as a response to a fetch request from the frontend.

**How would you output the data to the browser?**

Turning the JSON string back into an array would allow me to loop through the resulting objects and dynamically create or update front end elements with the blog data.

**How would you optimize front-end code and assets for production?**

Generally reducing the size of your code and assets by using something like Webpack in production mode. This would minify your javascript and css files. You can also use lossless/lossy compression on the images on your site to drastically reduce the size and decrease load times. Another optimization would be to carefully monitor the number of requests the site is making to the backend, as well as caching some results that don’t need to be updated as frequently.

# Task Three

There’s a missing comma “,” on line 8 after: “title”: “hello back”

The array isn’t assigned to a variable.

No semicolon at the end, although it’s not necessarily required.