Task 2

In AJAX applications, JSON is lighter and more efficient than XML. Since JSON is valid JavaScript, it meshes much better with Ajax/JavaScript than XML does. It is also simpler than XML to parse. To fetch a JSON string, we can simply use JSON.Parse. To do the same thing to an XML document, we need to use the XML DOM to loop through the document and extract all values and store them in variables.

Although JSON seems like the superior choice, there are some features that it lacks compared to XML. For instance, XML har Schema support which gives us the ability to specify the format of a document so that will be used to control that supplied data is in the correct, specified format. This is important when we’re passing data between separate systems were the wrong data format may stop it from being processed.

XML also has namespace support, which gives us the ability to mix data that is intended to be read/written by multiple sources in the same document.

Task 3

Processing a request asynchronously avoids the delay while the retrieval of data happens. This allows a user to continue to use the web page while the data is being fetched. The data is then processed when it is ready, and the response is displayed to the user. In most situations, this is the superior choice as it minimizes delays and lets the user be occupied with other things on the page while the data loads.