In this report for the work sheet I shall be dealing with the lecture review questions:

1. *Explain what is meant by the stream abstraction. What is the relationship between streams and the observer pattern? What are streams useful for modelling and when might you use them in Rich Web development?*

Streams are an abstraction, used to model asynchronous data sources. They are a powerful tool for when processing data and you do not know its size or arrival time. They are a time ordered list of items which you read and transform into other streams and values in some application specific order. They can be created over anything having a similar programming pattern to arrays.

Streams implement the observer pattern as they have been designed to work in the same way as it, they are incredibly useful in the ability they have allowing to take in vast amounts of data if needed when you are unsure of when the data may be used or inputted.

In the context of rich web app development, streams are useful as abstractions of asynchronous events, they also deal with solving the same issues as promises but can deal with a larger source of data synchronisation. The data they deal with is realised using the subscribe operation. So you would use streams when trying to deal with infinitely sized data, that you are unsure of when it will be delivered.

1. *Assume that you are building an interface to an API in your Rich Web App. Describe in detail how you could use the RxJS library to handle asynchronous network responses to API requests. In your opinion, what are the beneﬁts to using a streams library for networking over, say, promises? And what do you think are the downsides?*

A streams library for networking the application would give more synchronicity, it also allows for infinite data input. However adding an RxJs library also adds complexity into the API, that may or may not be needed, to add this as it is not widely supported, you need to import a full library to support, where with promises, you could develop an app solely with them, depending as well on size/expected size of the data that is to be inputted, if there is not a worry on size matters, and you are ok with a more simple synchronicity, then just using promises would be easier and less complex to implement.