**Career Guess who – Problems encountered and conquered**

**Backend**

* Node.js runs serverside, it doesn’t have access to the Web API and hence the browser for development environment – having to debug in VSCode terminal and couldn’t use browser
* Using NPM initialise puppeteer library and learning how to use File System (a Node.js module)
* Puppeteer uses asynchronous JS, having to learn about keywords await to ensure the async puppeteer functions were executing in order – look up for clarification
* Console logging errors whilst running Puppeteer API (for example, finding data of people with common names – Gary Kelly)
* Running a unit test on the extracted data to ensure it was clean/consistent and was suitable for use in frontend -> this prevented bugs from occurring in the frontend during data presentation as applying logic to consistent data sets vastly reduced error rates - TODO

**Frontend**

* Using dynamic imports (asynchronous) to import the correct set of data dependent on the result of the random number generator
* Manipulating string/array data types and using regular expressions to process the data into a presentable format on the frontend
* Dynamically generating HTML code in JS to create flexible tables
* Basic DOM manipulation to add functionality to the game and make the game intuitive to play
* Robust input validation on the user input field (forgiving as some players names are hard to spell)