**Modern Asynchronous Programming: Async/await**

**Instructions**

To complete this assignment, you will need to convert two functions to use async/await. In doing so, you'll have to make sure the code continues to work as intended.

When running the tests, most of them will pass! There are two tests at the end of the test suite that you will need to pass which check for the conversion.

**Fortune teller functions**

You'll be using the following functions provided in the ./utils/fortune-teller.js file. This library is just a way for you to practice working with promises. While you can see the source code in the file, you do not need to understand it fully or make any changes to it.

**welcome()**

The welcome() function returns a promise that, when resolved, will return a short welcome string.

welcome().then((response) => {

console.log(response);

*// "Provide me a question and I'll give you an answer..."*

});

**goodbye()**

The goodbye() function returns a promise that, when resolved, will return a short goodbye string.

goodbye().then((response) => {

console.log(response);

*// "Best of luck in the future..."*

});

**tell()**

The tell() function takes a string and returns a promise that, when resolved, will return a random fortune.

tell("Will the weather be warm today?").then((response) => {

console.log(response);

*// "Outlook good."*

});

If no argument is provided, then the function will return a rejection that returns a short error message.

tell().catch((response) => {

console.log(response);

*// "A question is required..."*

});

**Functions to complete**

Update the getFortune() and fullSession() functions to use async and await as well as try and catch. Note that these functions require you to log statements with console.log() and should continue to do so.

**Tips**

* You may complete this challenge on your own machine before uploading it to Qualified.
* Reference the related checkpoint for help on completing this Practice problem.
* If you need help, contact your mentor or speak with your peers in Slack.