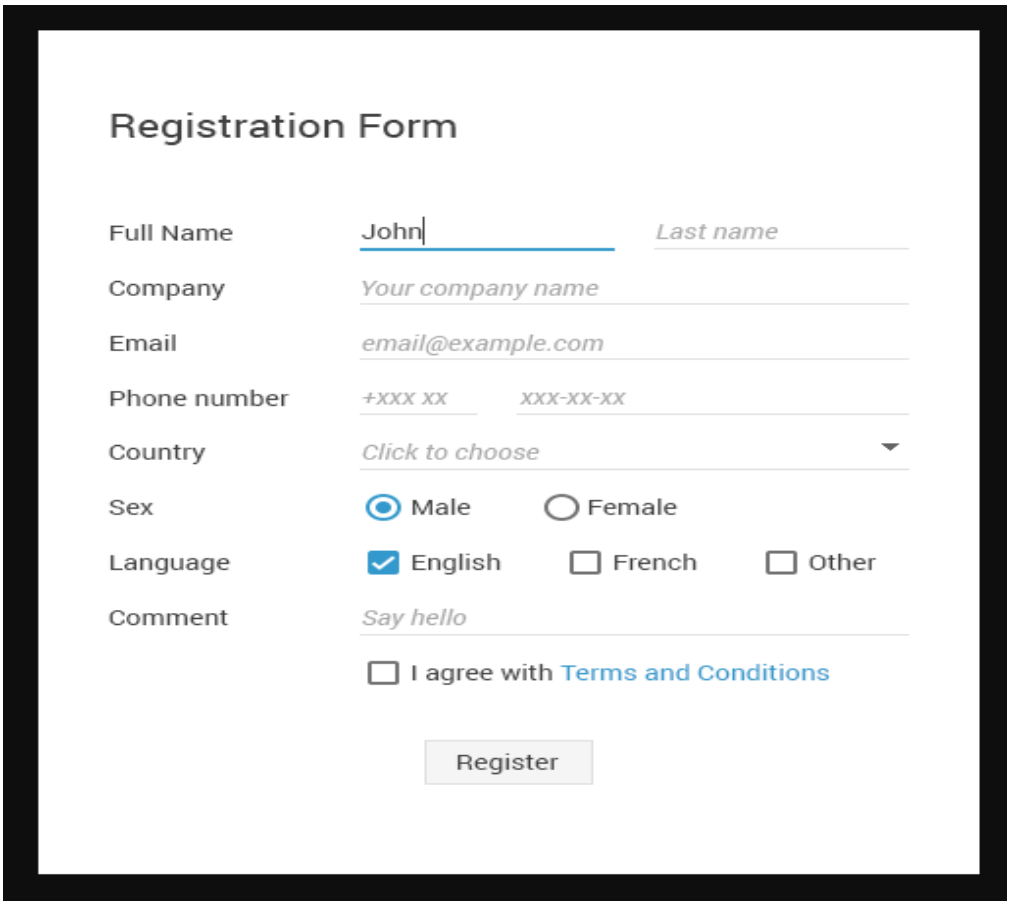
JavaScript Questions

1. Write program to create below kind of form with all field's validation required and on Register button show registration successfully message.



2. Write program to create a function that takes a string representing a function and converts between an arrow function and a regular function.

- If the input is a regular function, return an equivalent arrow function.
- If the input is a arrow function, return an equivalent regular function.

Examples:

// Function to arrow

> "function () {}" → "() => {}"

> "function name() {}" → "const name = () => {}"
>
> "function name(str) { console.log(str); }" → "const name = (str) =>
> { console.log(str); }"
>
> **// Arrow to function**
>
> "() => {}" → "function () {}"
>
> "const name = () => {}" → "function name() {}"
>
> "let name = (str) => { console.log(str); }" → "function name(str)
> { console.log(str); }"

Notes

- The body of the function can be multiline.
- Be careful with spaces, sometimes they're optional.
- An arrow function can be declared using `const` / `let` / `var` or nothing at all. However, when converting a regular function to an arrow, always use `const` (unless you're dealing with an anonymous function).

3. Write program to create a function that takes numbers as arguments, adds them together, and returns the product of digits until the answer is only 1 digit long.

Examples:

sumDigProd(16, 28) → 6

> // 16 + 28 = 44
> // 4 * 4 = 16
> // 1 * 6 = 6
>
> sumDigProd(0) → 0
>
> sumDigProd(1, 2, 3, 4, 5, 6) → 2

Notes

The input of the function is at least one number.

4. Write program to using javascript and ajax which call API and show data into HTML page in well format.

API End Point: https://mboum.com/api/v1/ne/news/?symbol=AAPL&apikey=demo

5. Write program to create a function to determine whether the second promise is faster than the first. If it is, return `true`. Otherwise, return `false`.

Examples:

speedTest(promise1, promise2) → true

```
// promise2 resolves faster


speedTest(promise1, promise2) → false
// promise1 resolves faster
```

Notes

Make sure you `await` both promises when executing them.