Logic thinking

-n -b

-s
-n } individually
-b

-s -n
-s -b } combination

mutually exclusive ←→ { -n -b → cannot exist together

-n ~~-b~~ or -b ~~-n~~

-n -s

-s -n -b
✓ ✓ ✗

-s -b -n
✓ ✓ ✗

first come, first serve

-s -n -b

optionArr = [-s, -n, -b] ✓

ion = 1
iob = 2

findoption

if(ion < iob) ✓
findoption = "-n";
}
else {
findOption = "-b";
}

if (findOption == "-n") {
n() ✓
}
else b(); ✓

function n(){

}

-b

function b(){

}

-b

-s -n

↳ [-s, -n] → if(ion != -1 && iob != -1) { ✓✓

{ if(ion < iob) || 1 < -1 → false
fo = -n
}
else fo = -b → blunders

ion = 1
iob = -1

[-s] → fo = " "

Extra features

1) create a file → node wcat.js -c filename

↓

fs.createFileSync(file) ✓

1.5

4-5

1) return ✓ ✓
2) || -
3) readme.md

2-3 technical 4 Manager
5 HR

↳ LTM
↳ OOPS
↳ Scaling
↳ DBMS
↳ CN
↳ DSA

↳ HTML
CSS
↓
internal use

fruitsArr → ["apple", "banana", "orange"]

fruitsArr.includes("apple") // true
fruitsArr.includes("grapes") // false