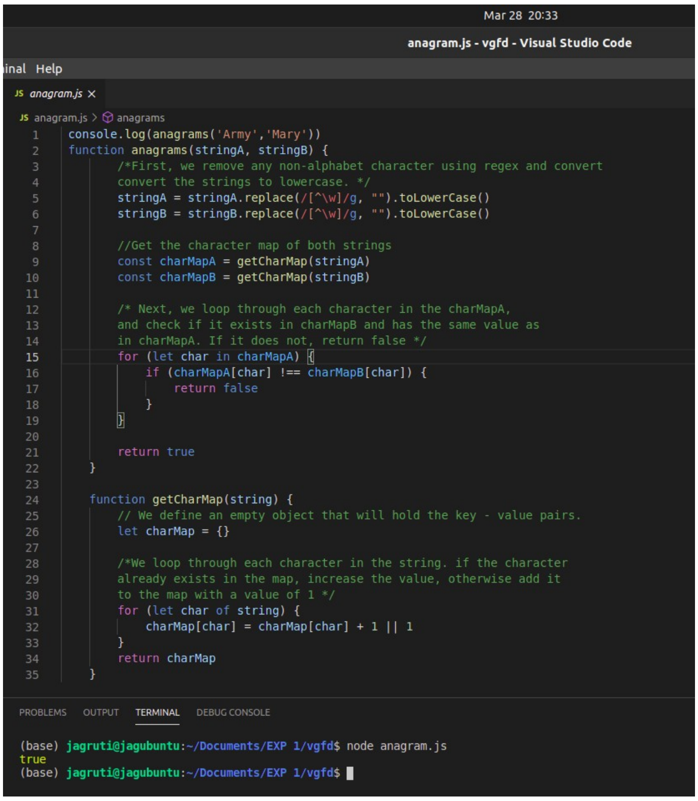
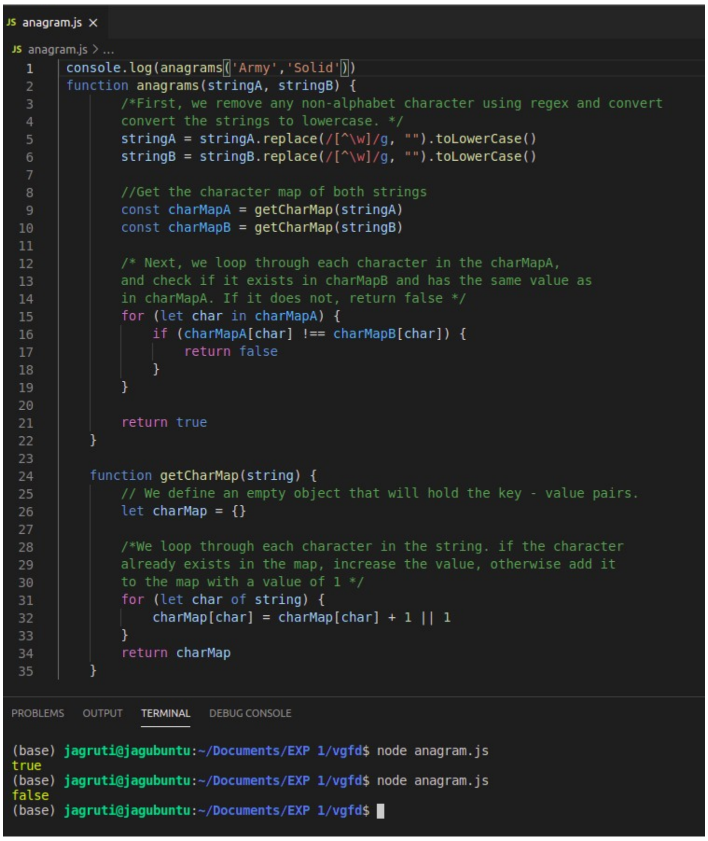
Output for the Fronend Task:
1.Anagram



```javascript
console.log(anagrams('Army','Mary'))
function anagrams(stringA, stringB) {
        /*First, we remove any non-alphabet character using regex and convert
        convert the strings to lowercase. */
        stringA = stringA.replace(/[^\w]/g, "").toLowerCase()
        stringB = stringB.replace(/[^\w]/g, "").toLowerCase()

        //Get the character map of both strings
        const charMapA = getCharMap(stringA)
        const charMapB = getCharMap(stringB)

        /* Next, we loop through each character in the charMapA,
        and check if it exists in charMapB and has the same value as
        in charMapA. If it does not, return false */
        for (let char in charMapA) {
            if (charMapA[char] !== charMapB[char]) {
                return false
            }
        }

        return true
    }

    function getCharMap(string) {
        // We define an empty object that will hold the key - value pairs.
        let charMap = {}

        /*We loop through each character in the string. if the character
        already exists in the map, increase the value, otherwise add it
        to the map with a value of 1 */
        for (let char of string) {
            charMap[char] = charMap[char] + 1 || 1
        }
        return charMap
    }
```

```
(base) jagruti@jagubuntu:~/Documents/EXP 1/vgfd$ node anagram.js
true
(base) jagruti@jagubuntu:~/Documents/EXP 1/vgfd$ 
```

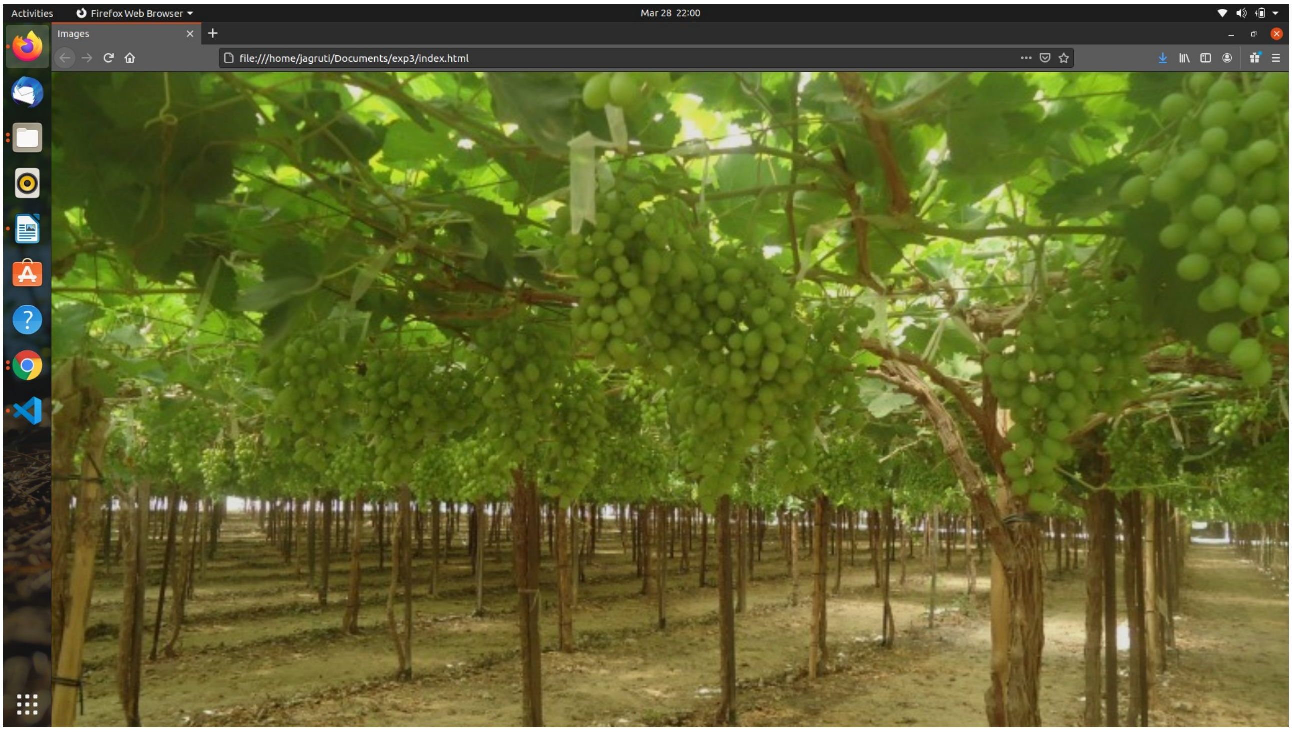
another example:



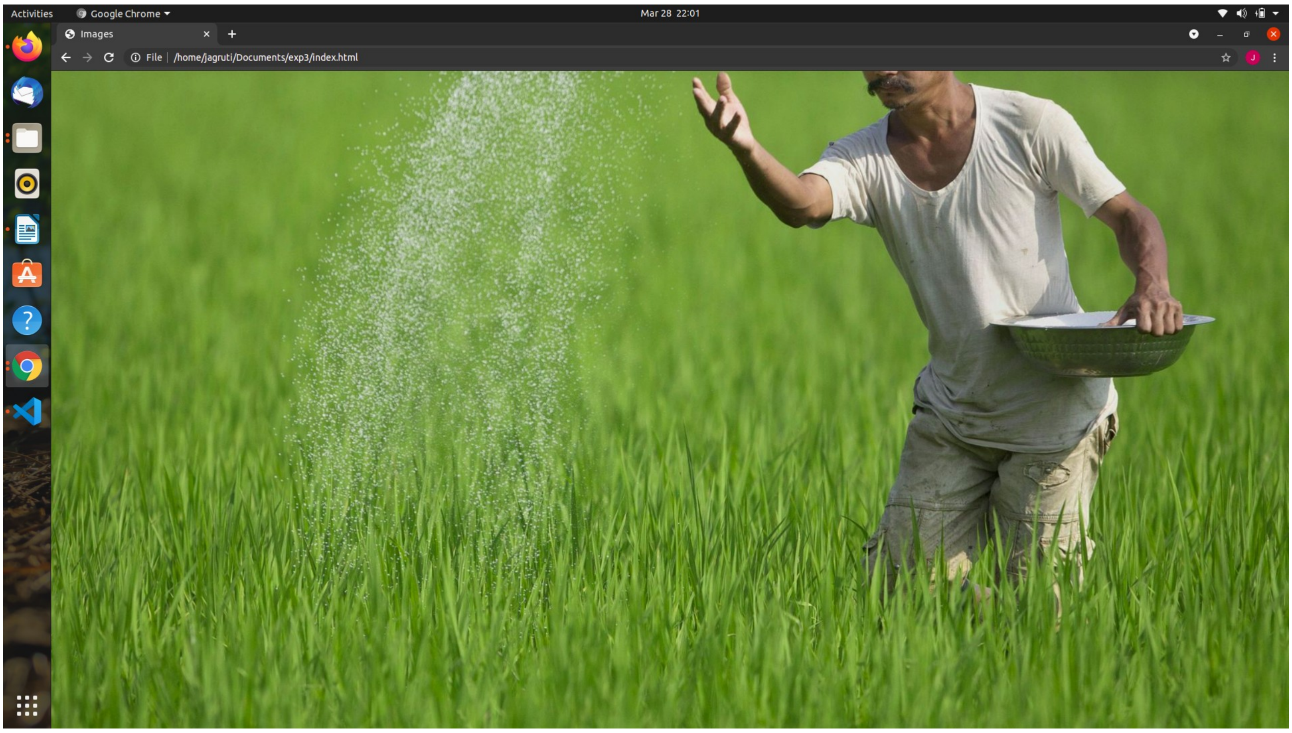
```javascript
console.log(anagrams('Army','Solid'))
function anagrams(stringA, stringB) {
        /*First, we remove any non-alphabet character using regex and convert
        convert the strings to lowercase. */
        stringA = stringA.replace(/[^\w]/g, "").toLowerCase()
        stringB = stringB.replace(/[^\w]/g, "").toLowerCase()

        //Get the character map of both strings
        const charMapA = getCharMap(stringA)
        const charMapB = getCharMap(stringB)

        /* Next, we loop through each character in the charMapA,
        and check if it exists in charMapB and has the same value as
        in charMapA. If it does not, return false */
        for (let char in charMapA) {
            if (charMapA[char] !== charMapB[char]) {
                return false
            }
        }

        return true
    }

    function getCharMap(string) {
        // We define an empty object that will hold the key - value pairs.
        let charMap = {}

        /*We loop through each character in the string. if the character
        already exists in the map, increase the value, otherwise add it
        to the map with a value of 1 */
        for (let char of string) {
            charMap[char] = charMap[char] + 1 || 1
        }
        return charMap
    }
```

```
(base) jagruti@jagubuntu:~/Documents/EXP 1/vgfd$ node anagram.js
true
(base) jagruti@jagubuntu:~/Documents/EXP 1/vgfd$ node anagram.js
false
(base) jagruti@jagubuntu:~/Documents/EXP 1/vgfd$ 
```

2.CSS for Cross Browser Compatibility
(Animation Images)
For firefox:



for chrome:


Activities
Google Chrome
Mar 28 22:01
Images
File | /home/jagruti/Documents/exp3/index.html