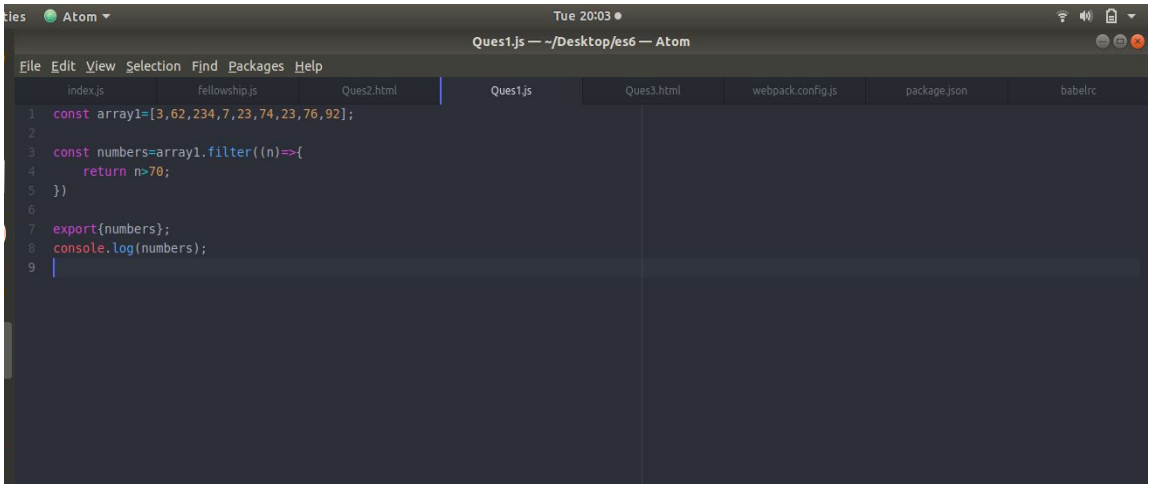
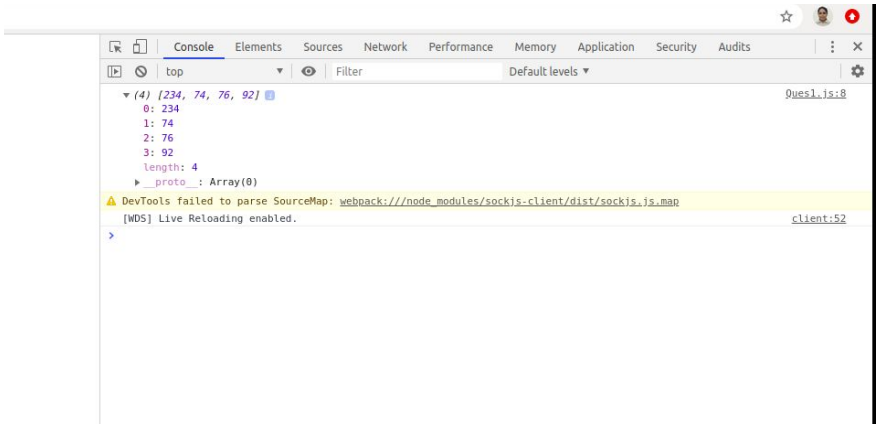
# ES6:Part6

**Q1. Given this array: `[3,62,234,7,23,74,23,76,92]`, Using arrow function, create an array of the numbers greater than `70`.**




OUTPUT:



**Q2.**

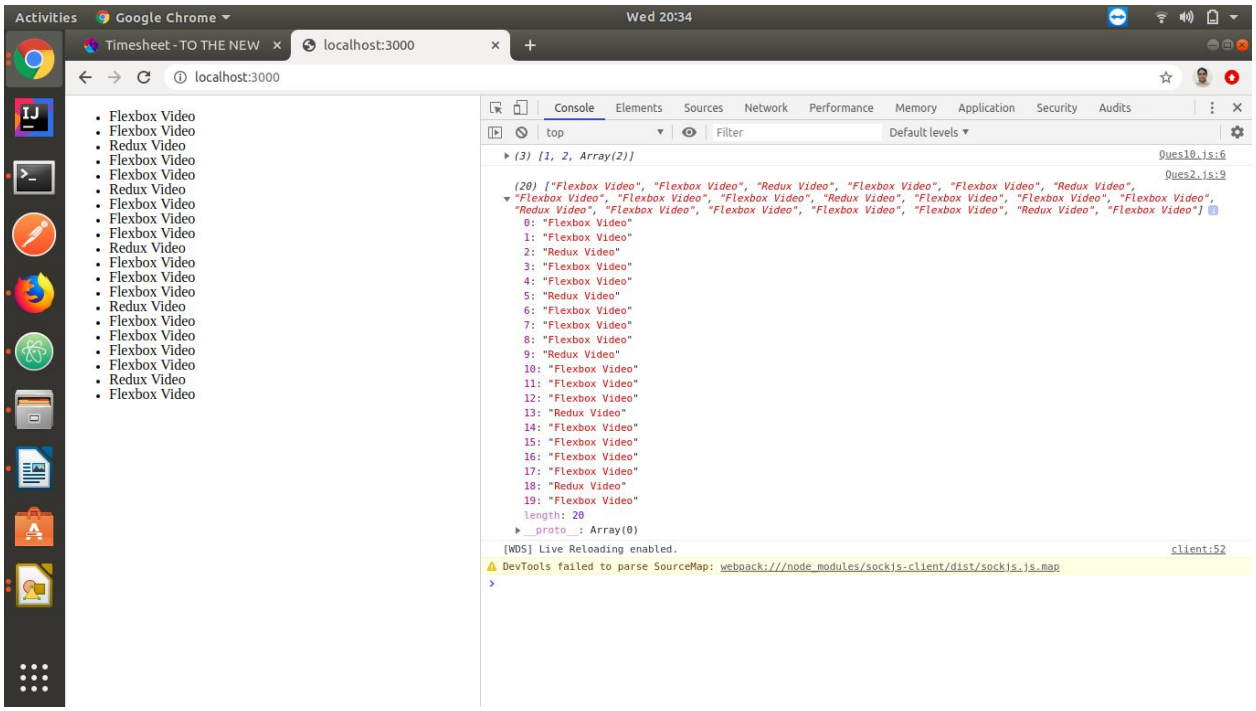
```
<ul>
  <li data-time="5:17">Flexbox Video</li>
  <li data-time="8:22">Flexbox Video</li>
  <li data-time="3:34">Redux Video</li>
  <li data-time="5:23">Flexbox Video</li>
  <li data-time="7:12">Flexbox Video</li>
  <li data-time="7:24">Redux Video</li>
  <li data-time="6:46">Flexbox Video</li>
  <li data-time="4:45">Flexbox Video</li>
  <li data-time="4:40">Flexbox Video</li>
  <li data-time="7:58">Redux Video</li>
  <li data-time="11:51">Flexbox Video</li>
  <li data-time="9:13">Flexbox Video</li>
  <li data-time="5:50">Flexbox Video</li>
  <li data-time="5:52">Redux Video</li>
  <li data-time="5:49">Flexbox Video</li>
  <li data-time="8:57">Flexbox Video</li>
  <li data-time="11:29">Flexbox Video</li>
  <li data-time="3:07">Flexbox Video</li>
  <li data-time="5:59">Redux Video</li>
  <li data-time="3:31">Flexbox Video</li>
</ul>
```

**a. Select all the list items on the page and convert to array.**

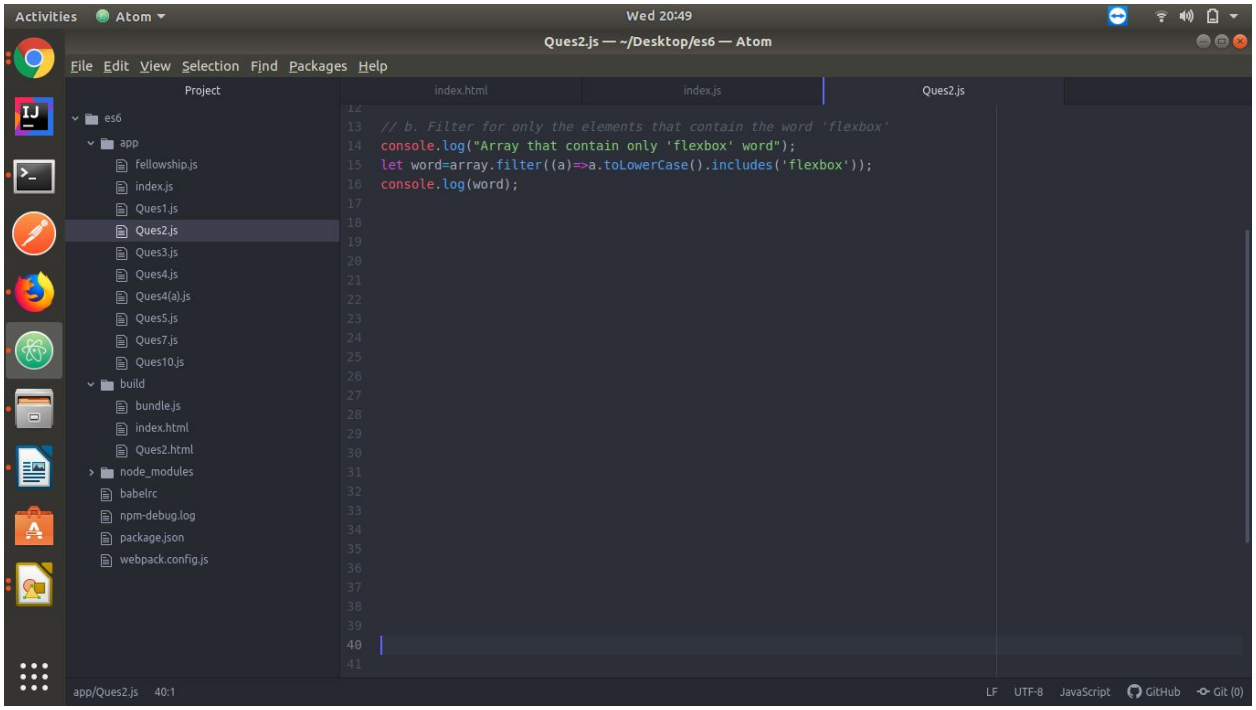
```html
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title></title>
  </head>
  <body>
    <ul id="List">
  <li data-time="5:17">Flexbox Video</li>
  <li data-time="8:22">Flexbox Video</li>
  <li data-time="3:34">Redux Video</li>
  <li data-time="5:23">Flexbox Video</li>
  <li data-time="7:12">Flexbox Video</li>
  <li data-time="7:24">Redux Video</li>
  <li data-time="6:46">Flexbox Video</li>
  <li data-time="4:45">Flexbox Video</li>
  <li data-time="4:40">Flexbox Video</li>
  <li data-time="7:58">Redux Video</li>
  <li data-time="11:51">Flexbox Video</li>
  <li data-time="9:13">Flexbox Video</li>
  <li data-time="5:50">Flexbox Video</li>
  <li data-time="5:52">Redux Video</li>
  <li data-time="5:49">Flexbox Video</li>
  <li data-time="8:57">Flexbox Video</li>
  <li data-time="11:29">Flexbox Video</li>
  <li data-time="3:07">Flexbox Video</li>
  <li data-time="5:59">Redux Video</li>
  <li data-time="3:31">Flexbox Video</li>
</ul>
    <script type="text/javascript"src="bundle.js"></script>
```

```javascript
// a. Select all the list items on the page and convert to array.

let array=[];
let lists=document.getElementById('List').getElementsByTagName("li");
for(var i=0;i<lists.length;i++){
  array.push(lists[i].innerHTML);
}
console.log(array);
```

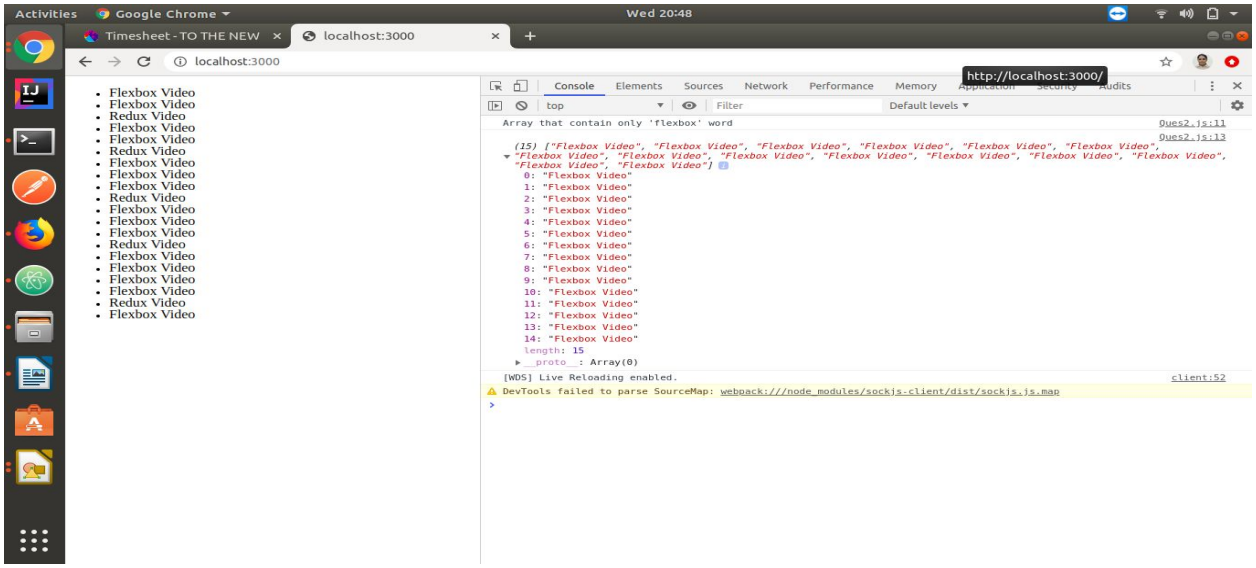
**OUTPUT:**


Activities
Google Chrome
Wed 20:34
Timesheet - TO THE NEW
localhost:3000
localhost:3000
Flexbox Video
Flexbox Video
Redux Video
Flexbox Video
Flexbox Video
Redux Video
Flexbox Video
Flexbox Video
Flexbox Video
Redux Video
Flexbox Video
Flexbox Video
Flexbox Video
Flexbox Video
Redux Video
Flexbox Video
Flexbox Video
Flexbox Video
Redux Video
Flexbox Video
Console
Elements
Sources
Network
Performance
Memory
Application
Security
Audits
top
Filter
Default levels
(3) [1, 2, Array(2)]
Ques10.js:6
Ques2.js:9
(20) ["Flexbox Video", "Flexbox Video", "Redux Video", "Flexbox Video", "Flexbox Video", "Redux Video", "Flexbox Video", "Flexbox Video", "Flexbox Video", "Redux Video", "Flexbox Video", "Flexbox Video", "Flexbox Video", "Flexbox Video", "Redux Video", "Flexbox Video", "Flexbox Video", "Flexbox Video", "Flexbox Video", "Redux Video", "Flexbox Video"]
0: "Flexbox Video"
1: "Flexbox Video"
2: "Redux Video"
3: "Flexbox Video"
4: "Flexbox Video"
5: "Redux Video"
6: "Flexbox Video"
7: "Flexbox Video"
8: "Flexbox Video"
9: "Redux Video"
10: "Flexbox Video"
11: "Flexbox Video"
12: "Flexbox Video"
13: "Redux Video"
14: "Flexbox Video"
15: "Flexbox Video"
16: "Flexbox Video"
17: "Flexbox Video"
18: "Redux Video"
19: "Flexbox Video"
length: 20
__proto__: Array(0)
[WDS] Live Reloading enabled.
client:52
DevTools failed to parse SourceMap: webpack:///node_modules/sockjs-client/dist/sockjs.js.map

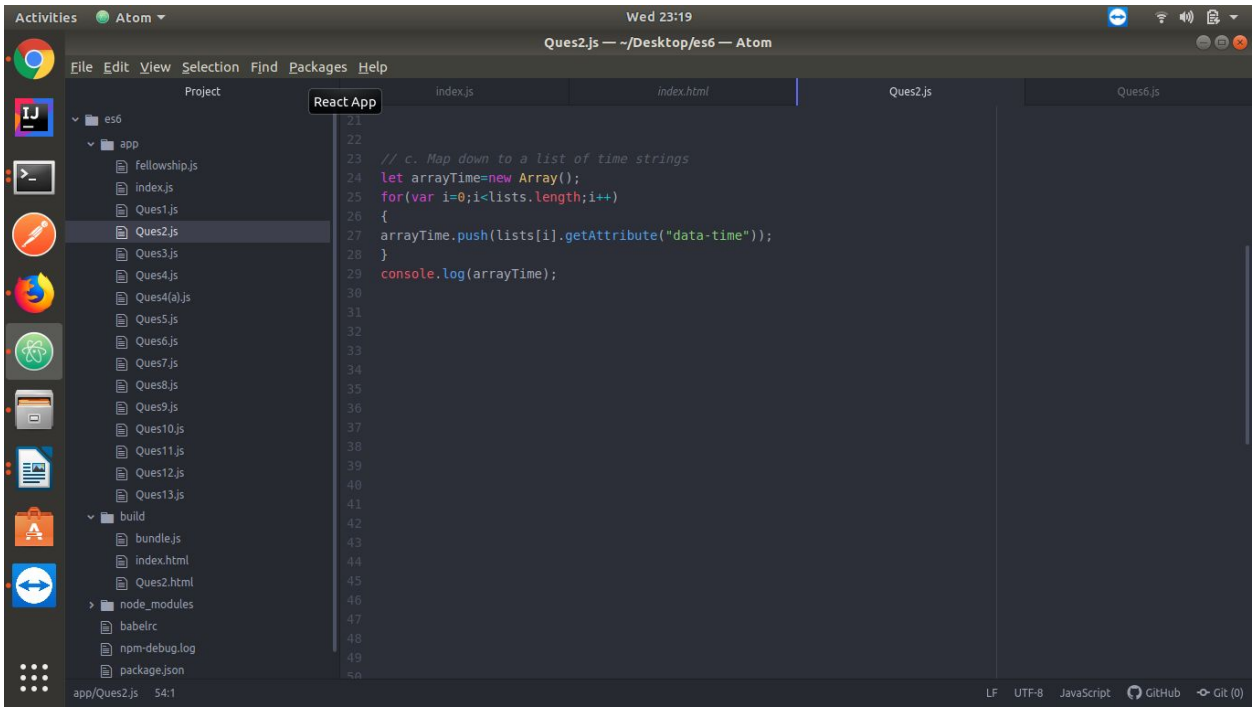
**b. Filter for only the elements that contain the word 'flexbox'.**



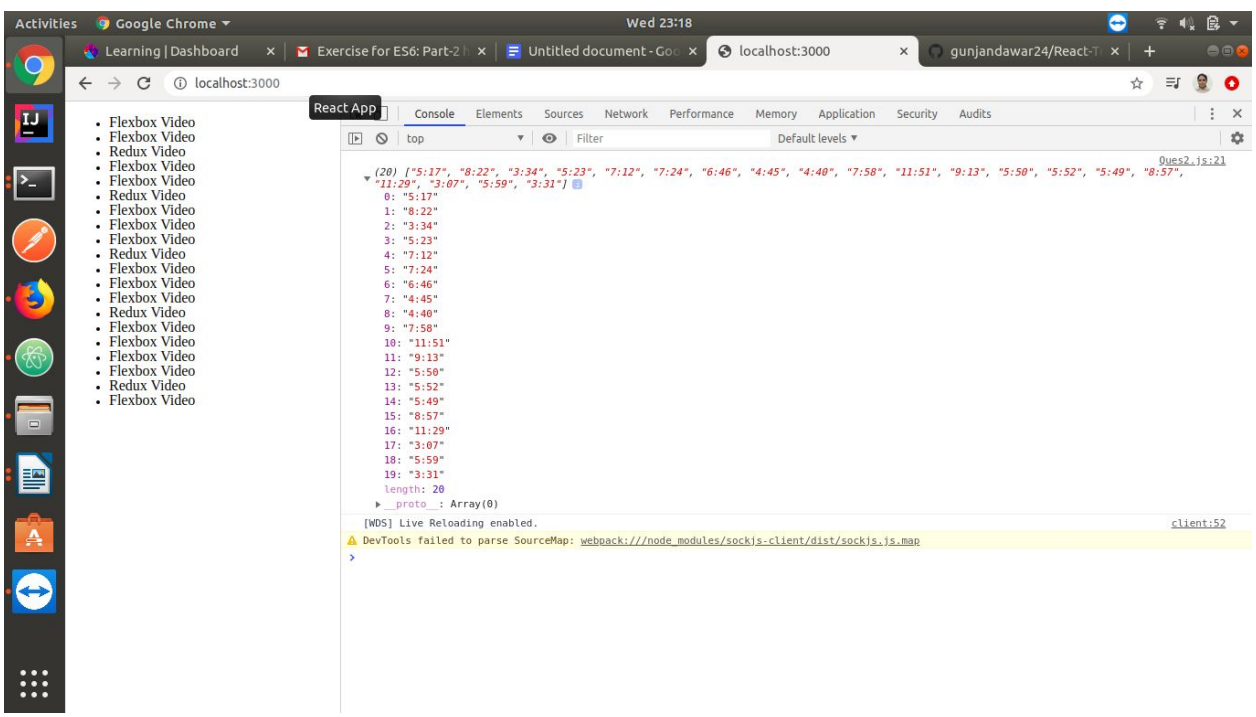

**OUTPUT:**



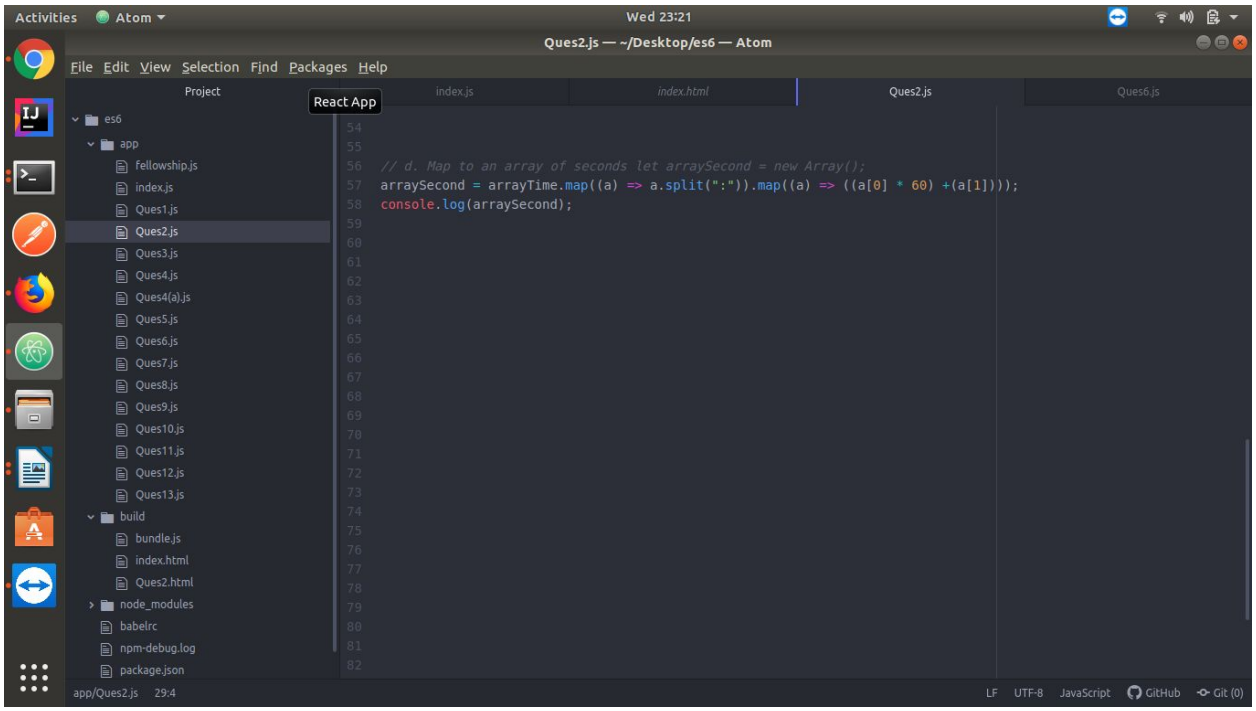
## c. Map down to a list of time strings



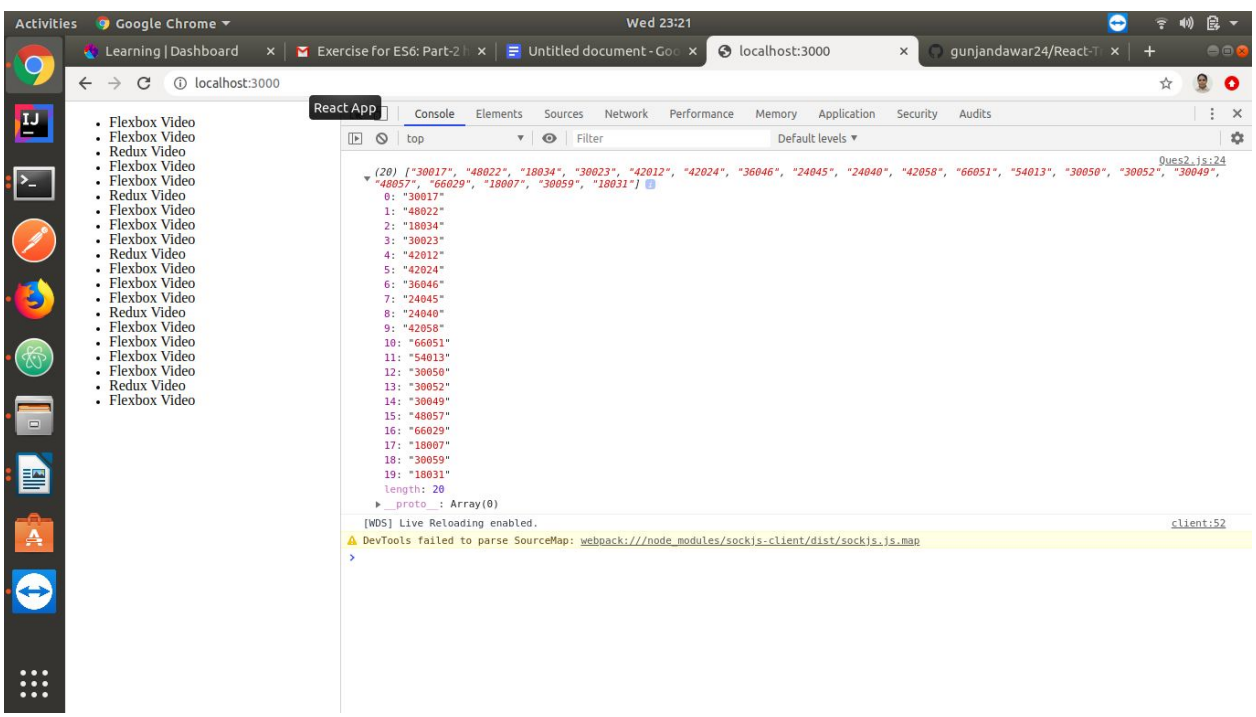

## OUTPUT:



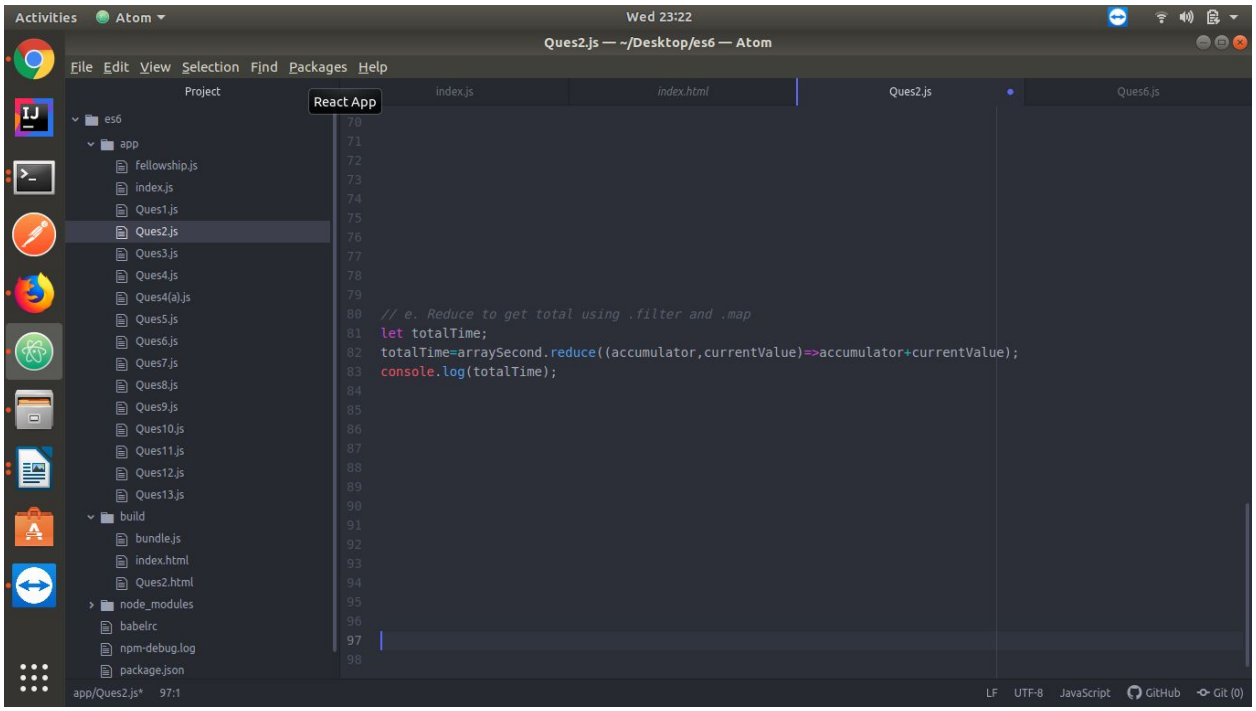
## d. Map to an array of seconds



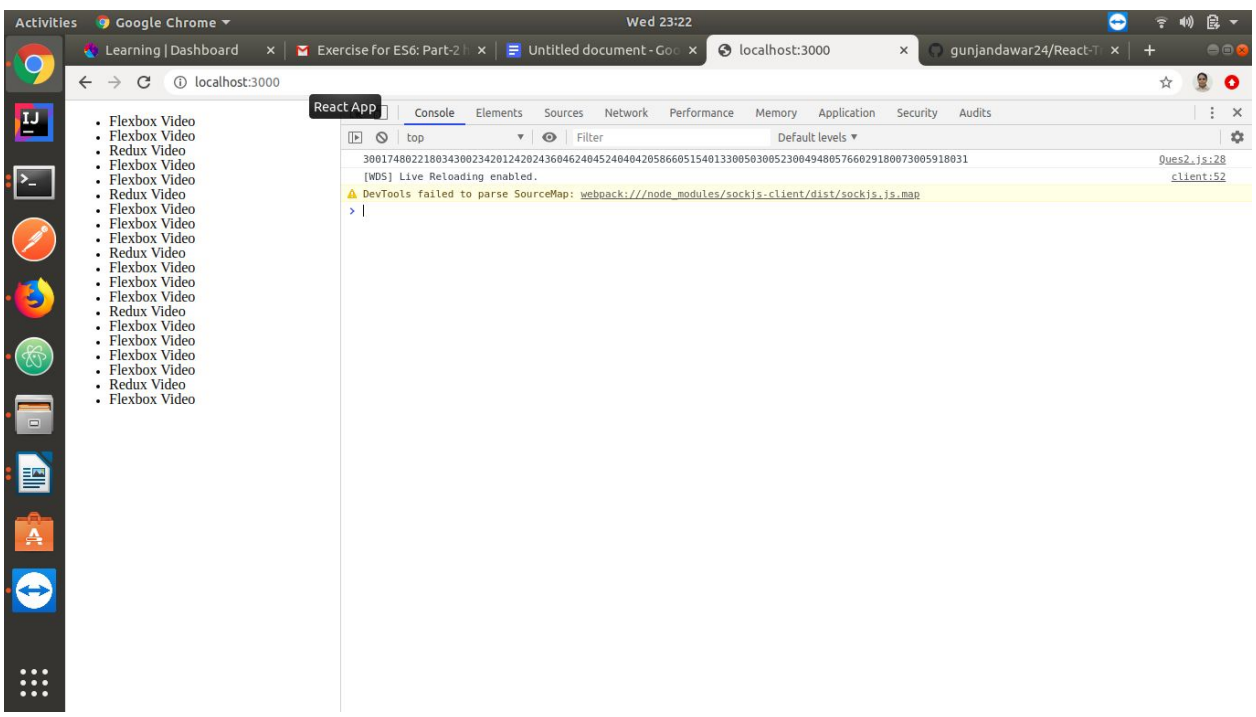

## OUTPUT:



## e. Reduce to get total using .filter and .map




## OUTPUT:



**Q3. Create a markup template using string literal**

**const song = {**

**name: 'Dying to live',**

**artist: 'Tupac',**

**featuring: 'Biggie Smalls'**

**};**

**Result:**

**" **

** **

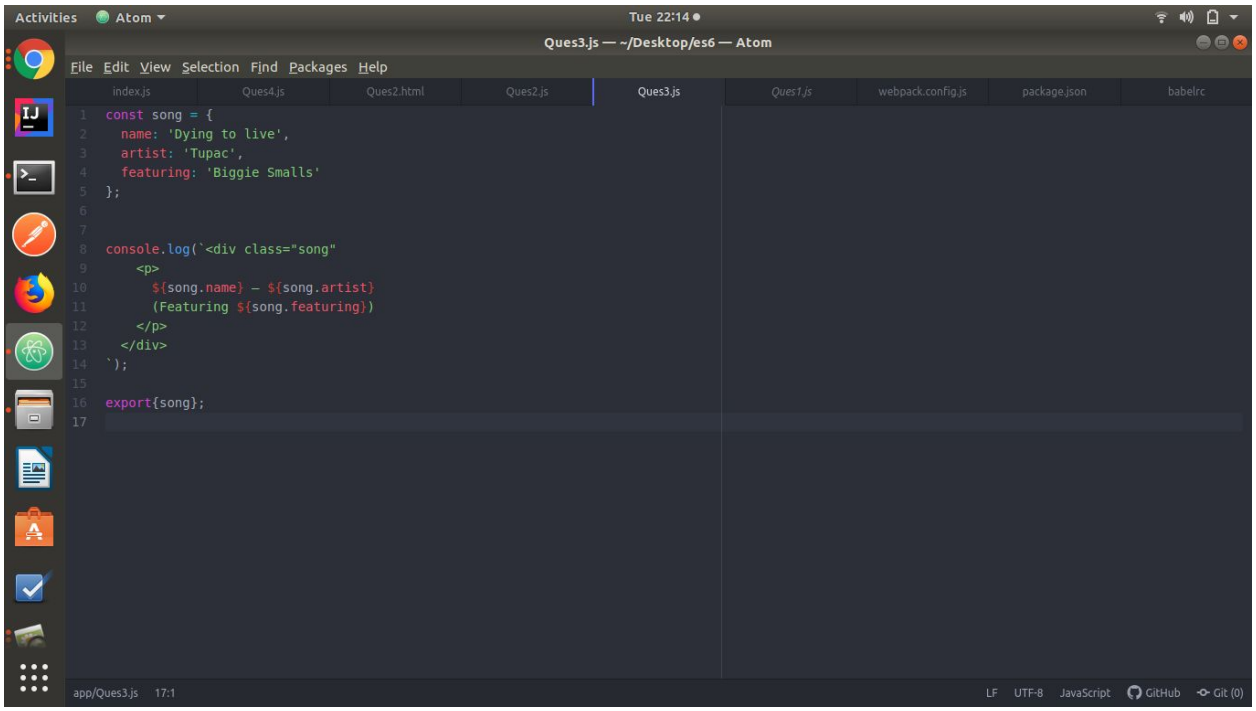
**Dying to live — Tupac**

**(Featuring Biggie Smalls)**

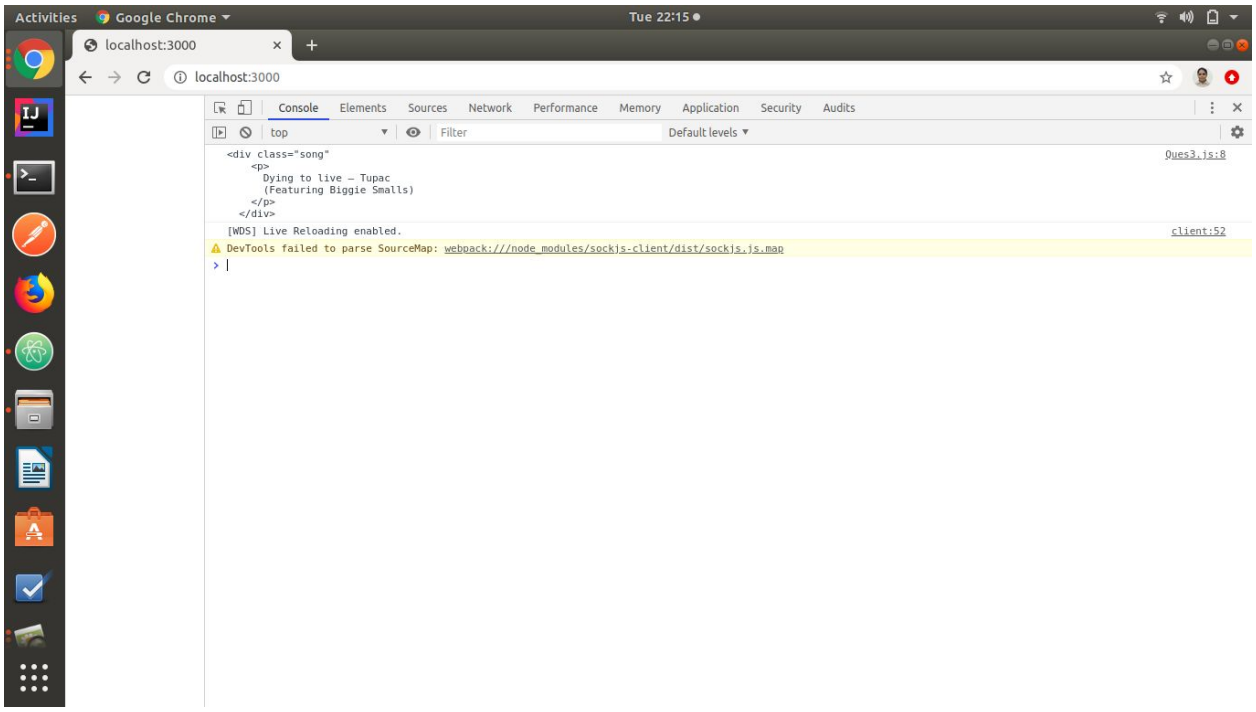
** **

** **

**“**



## OUTPUT:


Activities
Google Chrome
Tue 22:15
localhost:3000
localhost:3000
Console
Elements
Sources
Network
Performance
Memory
Application
Security
Audits
top
Filter
Default levels
<div class="song"

Dying to live – Tupac
(Featuring Biggie Smalls)

Ques3.js:8
[WDS] Live Reloading enabled.
client:52
DevTools failed to parse SourceMap: webpack:///node_modules/sockjs-client/dist/sockjs.js.map

**Q4. Extract all keys inside address object from user object using destructuring ?**

**const user = {**

**firstName: ‘Sahil’,**

**lastName: ‘Dua’,**

**Address: {**

**Line1: ‘address line 1’,**

**Line2: ‘address line 2’,**

**State: ‘Delhi’,**

**Pin: 110085,**

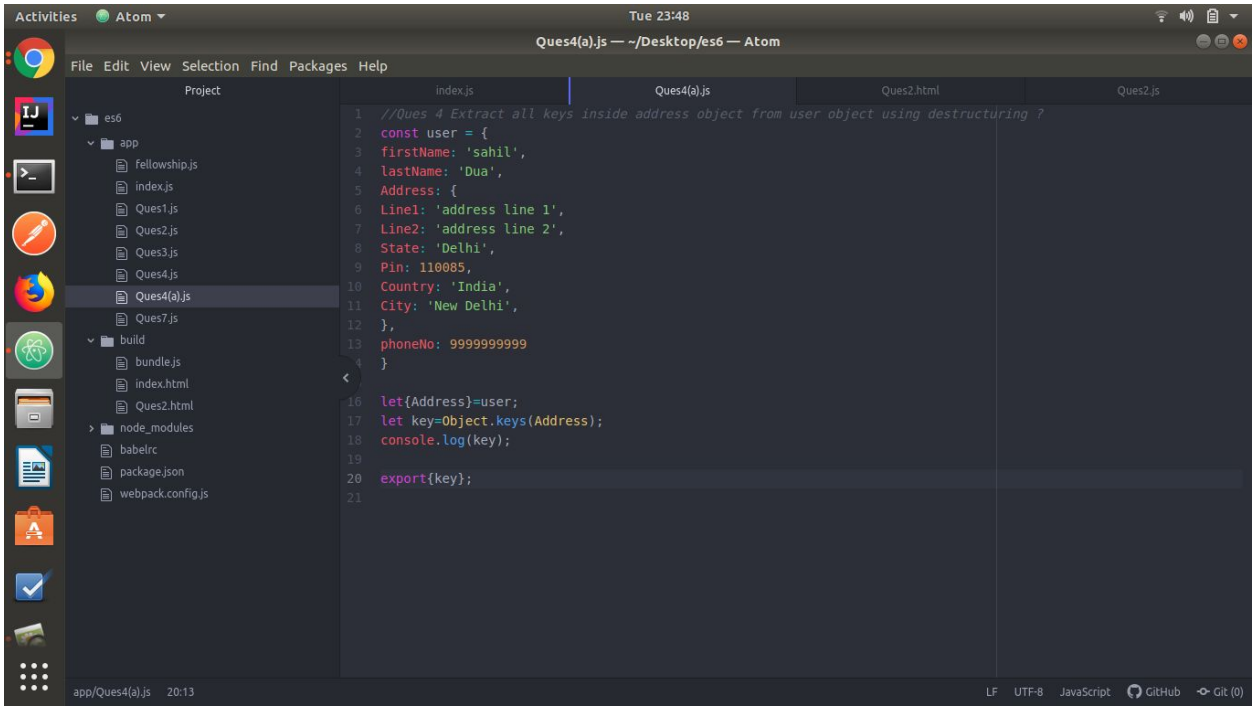
**Country: ‘India’,**

**City: ‘New Delhi’,**

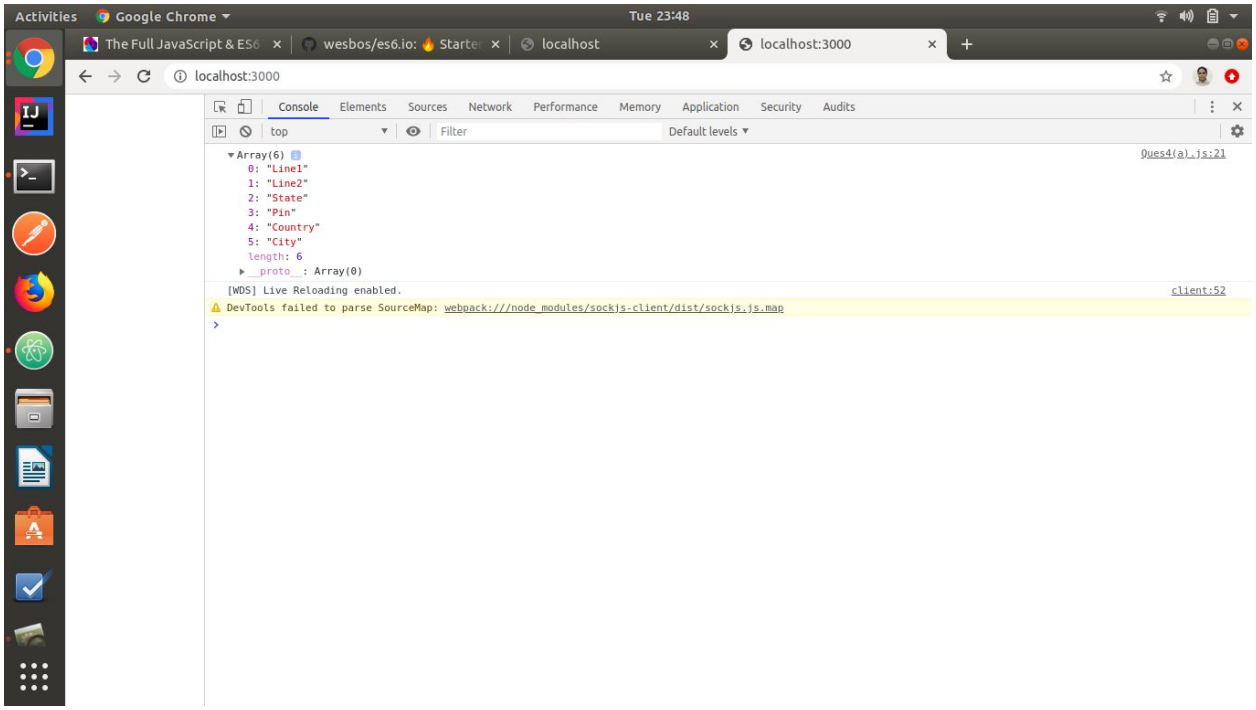
**},**

**phoneNo: 9999999999**

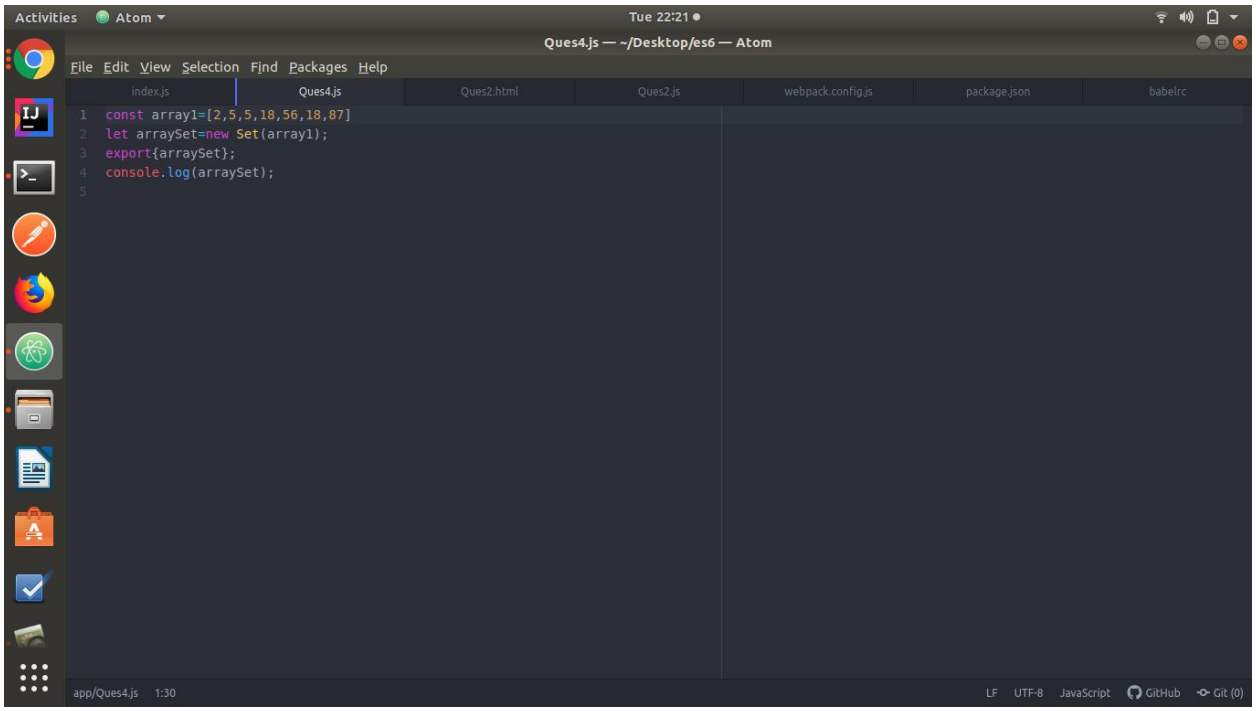
**}**



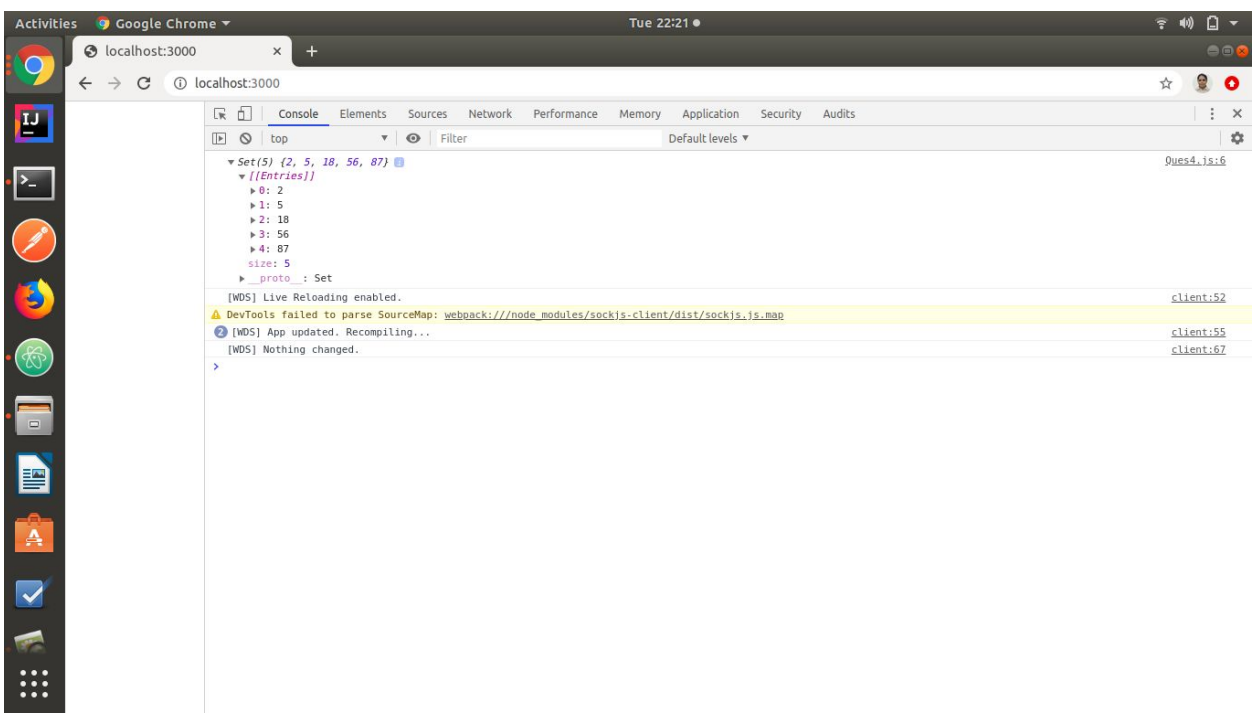
**OUTPUT:**


Activities
Google Chrome
Tue 23:48
The Full JavaScript & ES6
wesbos/es6.io: Starte
localhost
localhost:3000
localhost:3000
Console
Elements
Sources
Network
Performance
Memory
Application
Security
Audits
top
Filter
Default levels
Ques4(a).js:21
Array(6)
0: "Line1"
1: "Line2"
2: "State"
3: "Pin"
4: "Country"
5: "City"
length: 6
__proto__: Array(0)
[WDS] Live Reloading enabled.
client:52
DevTools failed to parse SourceMap: webpack:///node_modules/sockjs-client/dist/sockjs.js.map

## Q4. Filter unique array members using Set.




## OUTPUT:

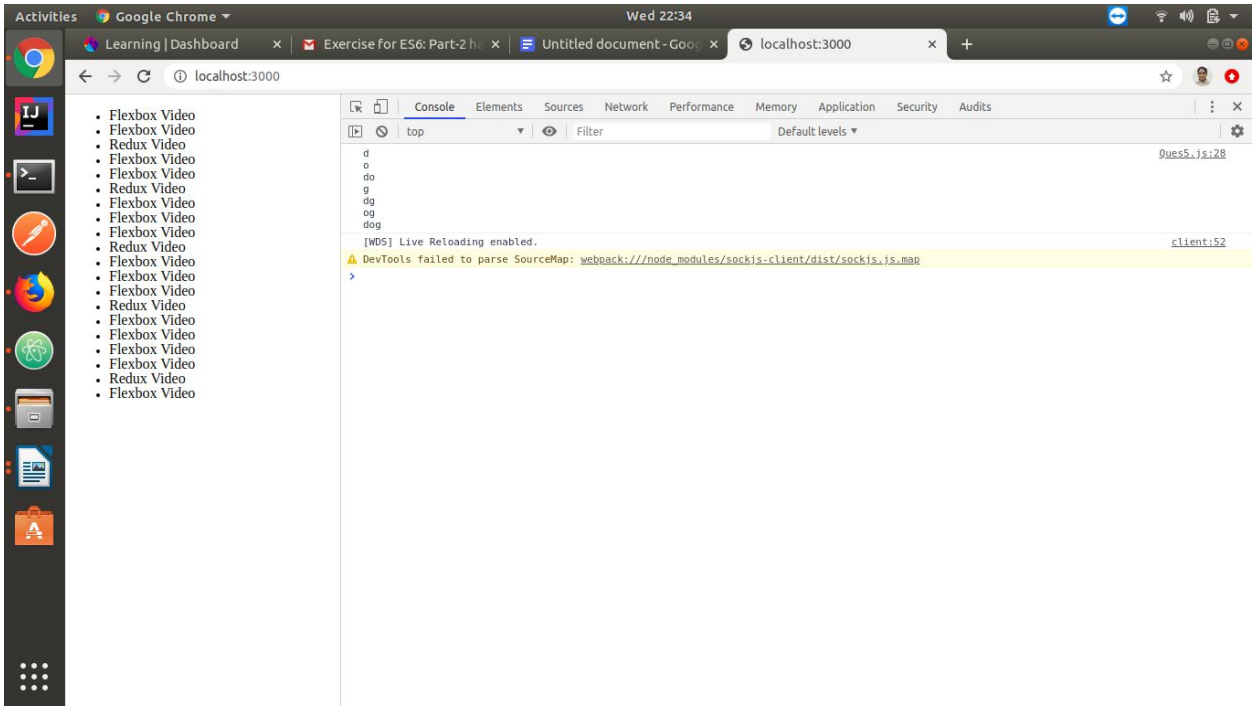


**Q5. Find the possible combinations of a string and store them in a MAP?**

```
function substrings(str1)
{
var array1 = [];
  for (var x = 0, y=1; x < str1.length; x++,y++)
  {
   array1[x]=str1.substring(x, y);
    }

var combi = [];
var temp= "";
var slent = Math.pow(2, array1.length);

for (var i = 0; i < slent ; i++)
{
    temp= "";
    for (var j=0;j<array1.length;j++) {
        if ((i & Math.pow(2,j))){
            temp += array1[j];
        }
    }
    if (temp !== "")
    {
        combi.push(temp);
    }
}
  console.log(combi.join("\n"));
}
```

```
substrings("dog");

export{substrings};
```

**OUTPUT:**



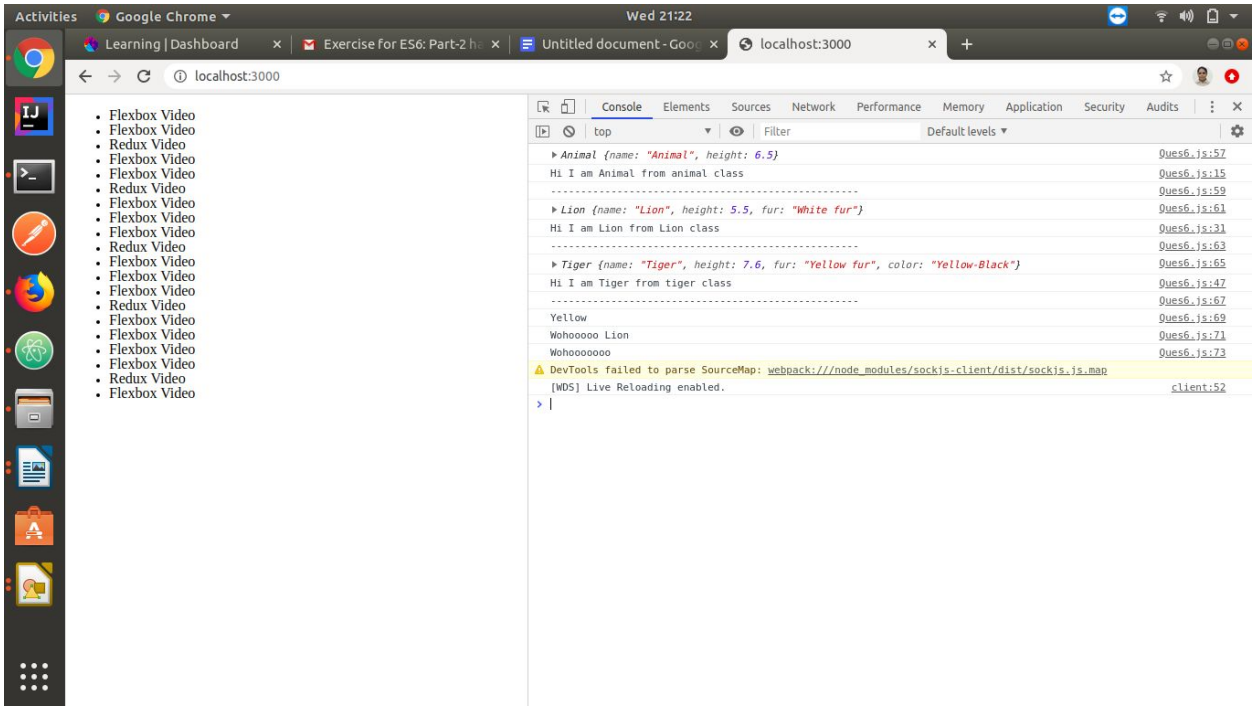
**Q6. Write a program to implement inheritance upto 3 classes.The Class must  public variables and static functions.**

```
class Animal {
    constructor(name, height) {
        this.name = name;
        this.height = height;
    }

    hello() {
        console.log(`Hi I am ${this.name} from animal class`);
    }

    static sound() {
        return "Wohoooooo";
    }
}



class Lion extends Animal {
    constructor(name, height, fur) {
        super(name, height);
        this.fur = fur;
    }
    hello() {
        console.log(`Hi I am ${this.name} from Lion class`);
    }
    static sound() {
        return "Wohooooo Lion"
    }
```

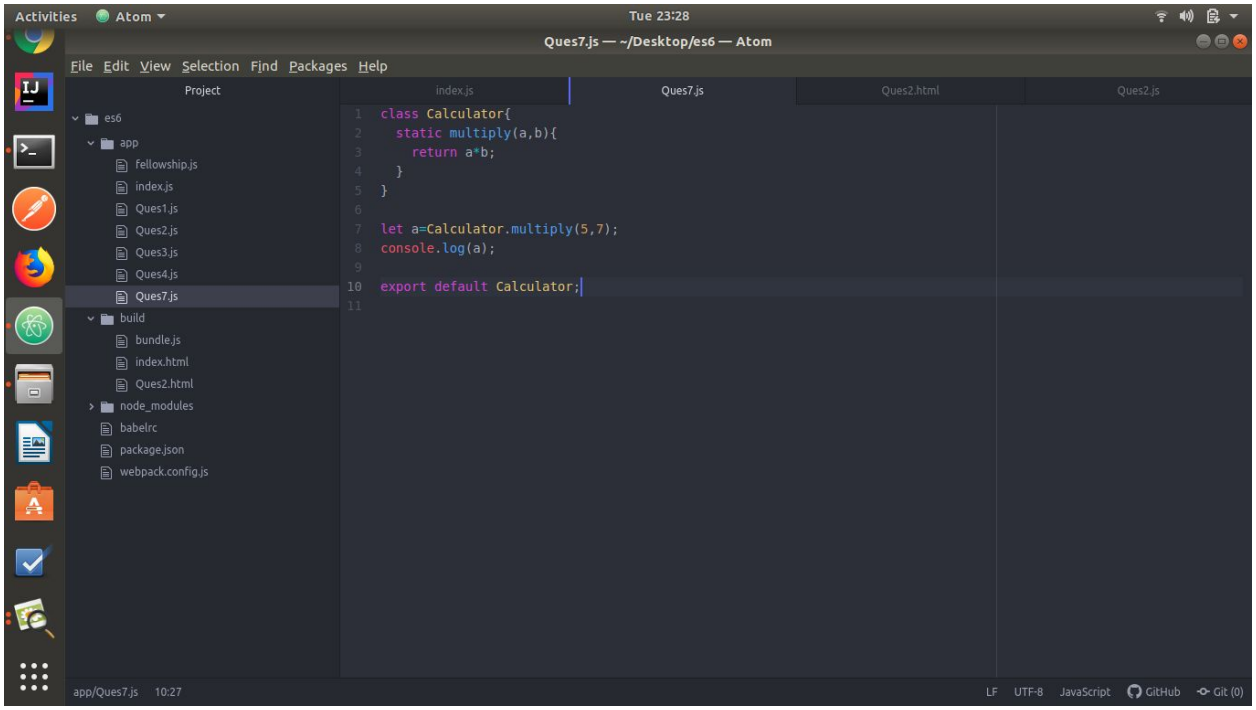
```
}



class Tiger extends Lion {
    constructor(name, height, fur, color) {
        super(name, height, fur)
        this.color = color;
    }
    hello() {

        console.log(`Hi I am ${this.name} from tiger class`);
    }
    static colour() {
        return "Yellow";
    }
}



let king = new Animal("Animal", 6.5);
console.log(king);
king.hello();
console.log("--------------------------------------------------");

let lion = new Lion("Lion", 5.5, "White fur");
console.log(lion);
lion.hello();
console.log("--------------------------------------------------");
```

```
let tiger = new Tiger("Tiger", 7.6, "Yellow fur", "Yellow-Black");
console.log(tiger);
tiger.hello();
console.log("-----------------------------------------------");

let staticMethod = Tiger.colour();
console.log(staticMethod);
let staticMethodLion = Lion.sound();
console.log(staticMethodLion);
let staticMethodAnimal = Animal.sound();
console.log(staticMethodAnimal);

export {
    king,
    lion,
    tiger,
    staticMethod,
    staticMethodLion,
    staticMethodAnimal
};
```

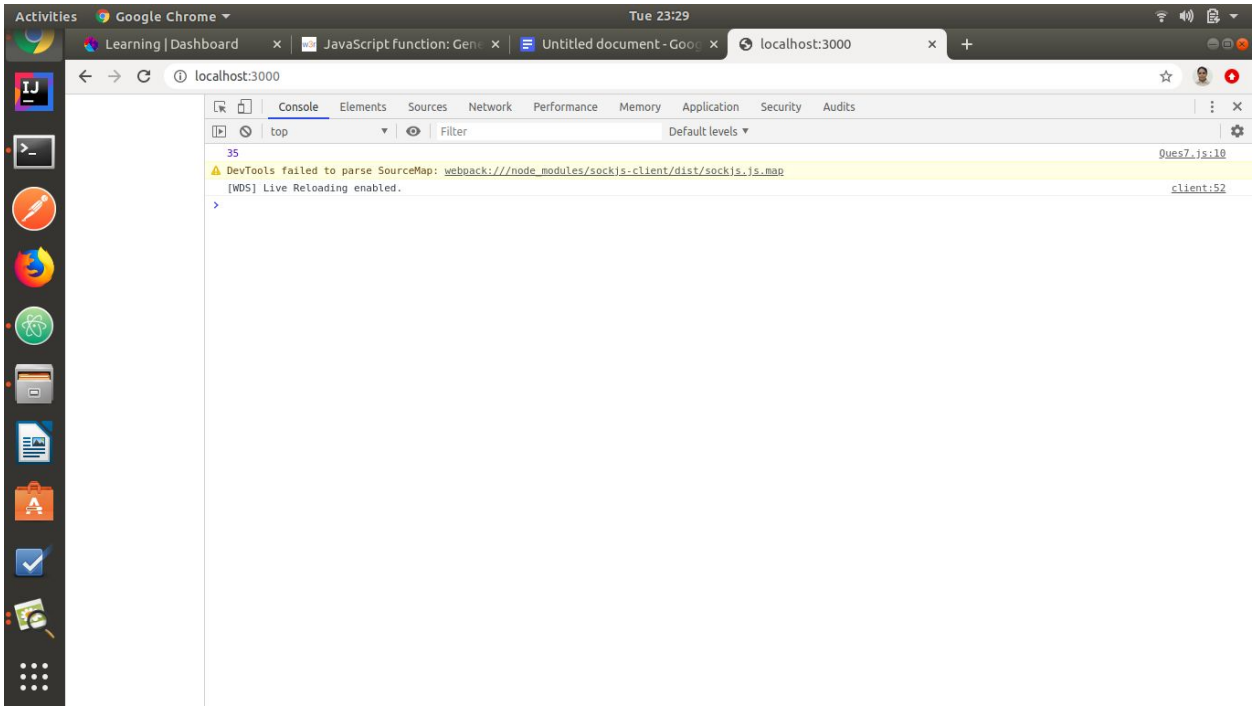
OUTPUT:


Activities
Google Chrome
Wed 21:22
Learning | Dashboard
Exercise for ES6: Part-2 h
Untitled document - Goo
localhost:3000
localhost:3000
Flexbox Video
Flexbox Video
Redux Video
Flexbox Video
Flexbox Video
Redux Video
Flexbox Video
Flexbox Video
Flexbox Video
Redux Video
Flexbox Video
Flexbox Video
Flexbox Video
Redux Video
Flexbox Video
Flexbox Video
Flexbox Video
Flexbox Video
Redux Video
Flexbox Video
Console
Elements
Sources
Network
Performance
Memory
Application
Security
Audits
top
Filter
Default levels
Animal {name: "Animal", height: 6.5} Ques6.js:57
Hi I am Animal from animal class Ques6.js:15
---------- Ques6.js:59
Lion {name: "Lion", height: 5.5, fur: "White fur"} Ques6.js:61
Hi I am Lion from Lion class Ques6.js:31
---------- Ques6.js:63
Tiger {name: "Tiger", height: 7.6, fur: "Yellow fur", color: "Yellow-Black"} Ques6.js:65
Hi I am Tiger from tiger class Ques6.js:47
---------- Ques6.js:67
Yellow Ques6.js:69
Wohooooo Lion Ques6.js:71
Wohoooooo Ques6.js:73
DevTools failed to parse SourceMap: webpack:///node_modules/sockjs-client/dist/sockjs.js.map
[WDS] Live Reloading enabled. client:52

## Q7. Write a program to implement a class having static functions.



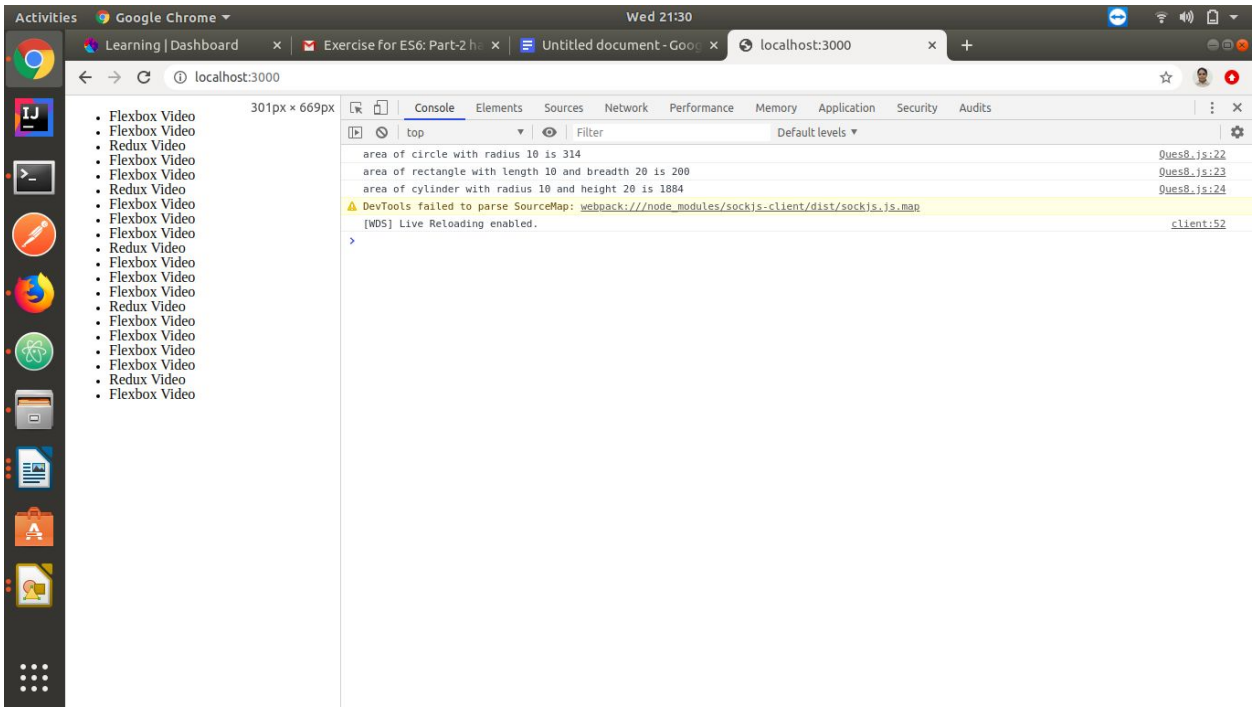

OUTPUT:



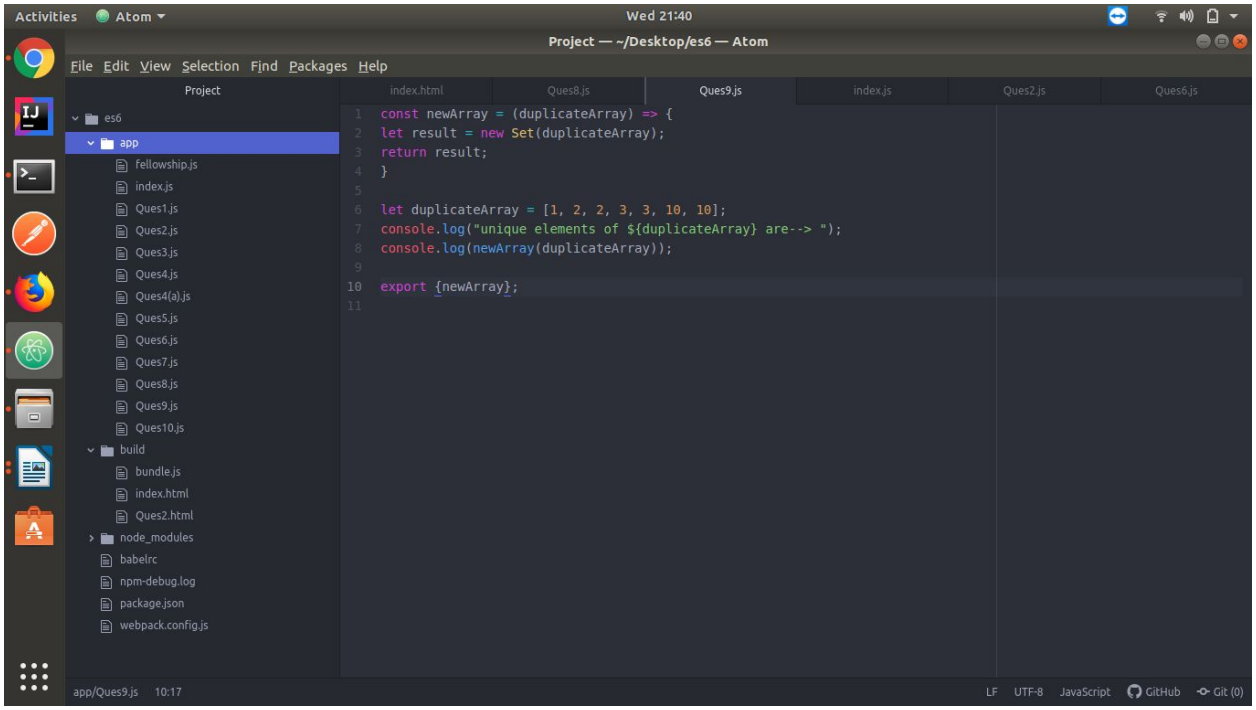
**Q8. Import a module containing the constants and method for calculating area of circle, rectangle, cylinder.**

```
const areaCircle=(radius)=>{
const pi=3.14;
let area=pi*(radius*radius);
return `area of circle with radius ${radius} is ${area}`
}


const areaRectangle=(length,breadth)=>{
return `area of rectangle with length ${length} and breadth ${breadth} is ${length*breadth}`
}


const areaCylinder=(radius,height)=>{
const pi=3.14;
let area=(2*pi*radius*height)+(2*pi*radius*radius);
return `area of cylinder with radius ${radius} and height ${height} is ${area}`
}

export{areaCircle,areaCylinder,areaRectangle};

console.log(areaCircle(10));
console.log(areaRectangle(10, 20));
console.log(areaCylinder(10, 20));
```

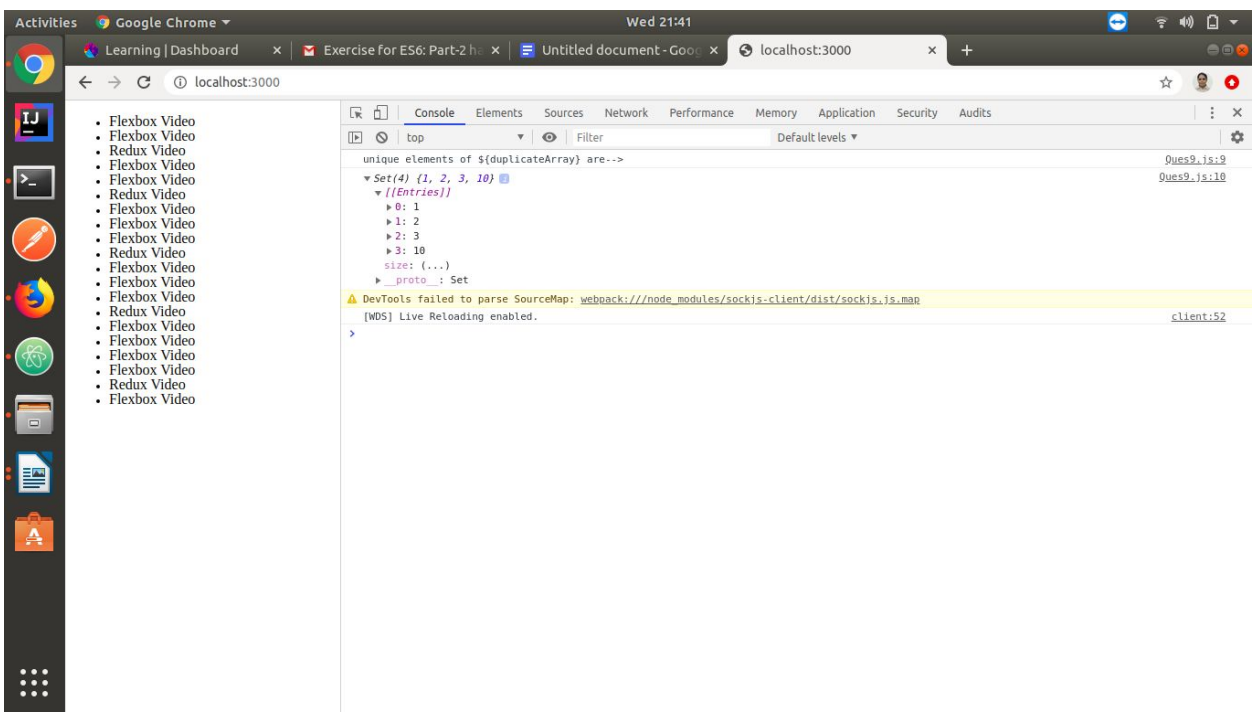
OUTPUT:


Activities
Google Chrome
Wed 21:30
Learning | Dashboard
Exercise for ES6: Part-2
Untitled document - Goog
localhost:3000
localhost:3000
301px × 669px
Flexbox Video
Flexbox Video
Redux Video
Flexbox Video
Flexbox Video
Redux Video
Flexbox Video
Flexbox Video
Flexbox Video
Redux Video
Flexbox Video
Flexbox Video
Flexbox Video
Redux Video
Flexbox Video
Flexbox Video
Flexbox Video
Flexbox Video
Redux Video
Flexbox Video
Console
Elements
Sources
Network
Performance
Memory
Application
Security
Audits
top
Filter
Default levels
area of circle with radius 10 is 314
Ques8.js:22
area of rectangle with length 10 and breadth 20 is 200
Ques8.js:23
area of cylinder with radius 10 and height 20 is 1884
Ques8.js:24
DevTools failed to parse SourceMap: webpack:///node_modules/sockjs-client/dist/sockjs.js.map
[WDS] Live Reloading enabled.
client:52

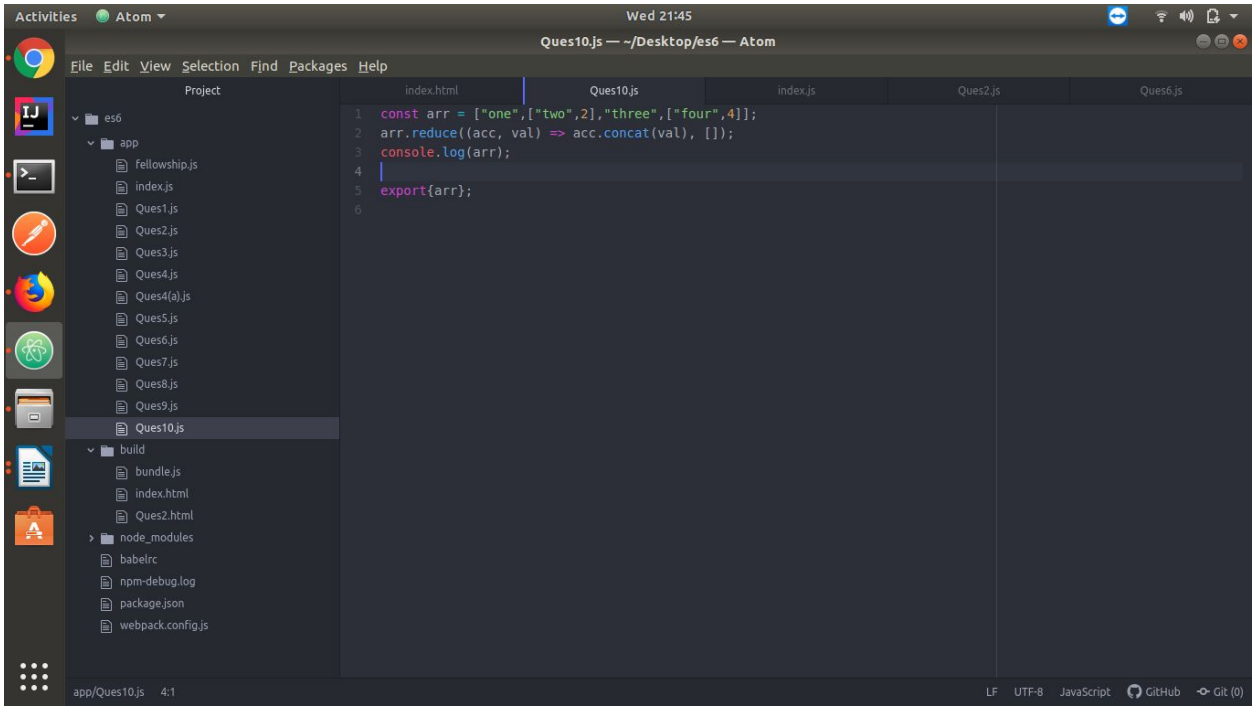
Q9. Import a module for filtering unique elements in an array.

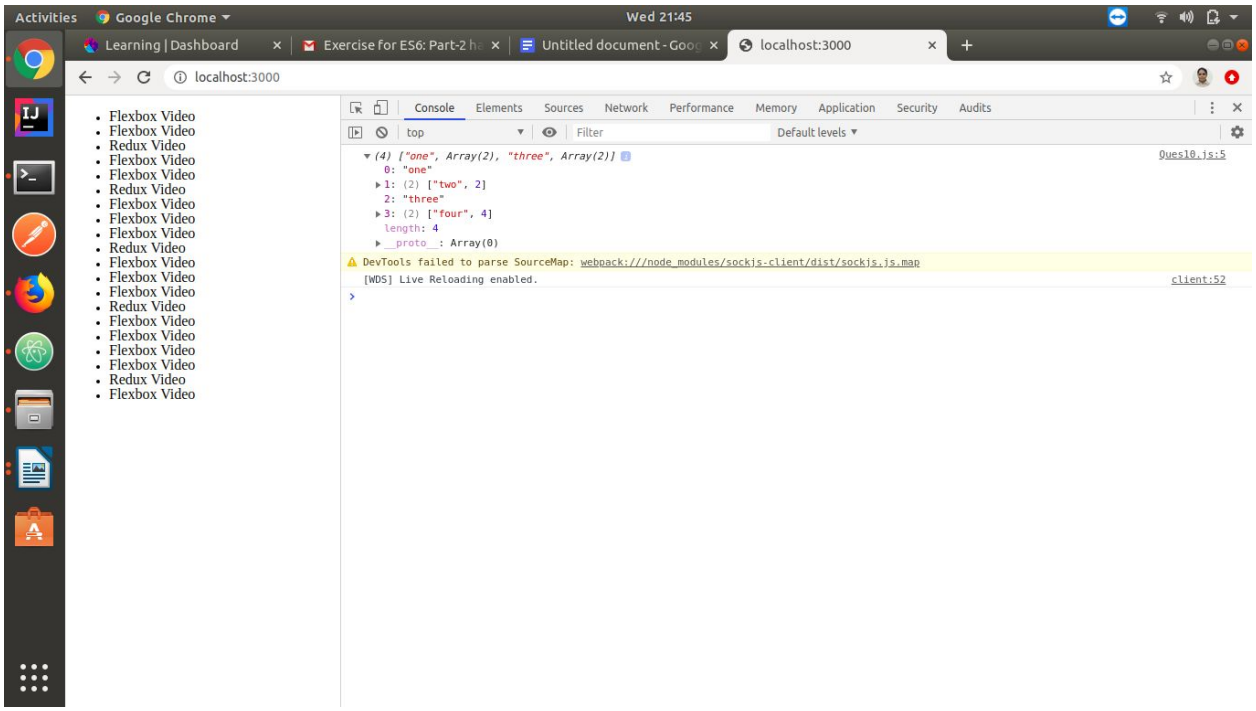



OUTPUT:



Q10. Write a program to flatten a nested array to single level using arrow functions.



**Q11. Implement a singly linked list in es6 and implement addFirst() addLast(), length(), getFirst(), getLast(). (without using array)**

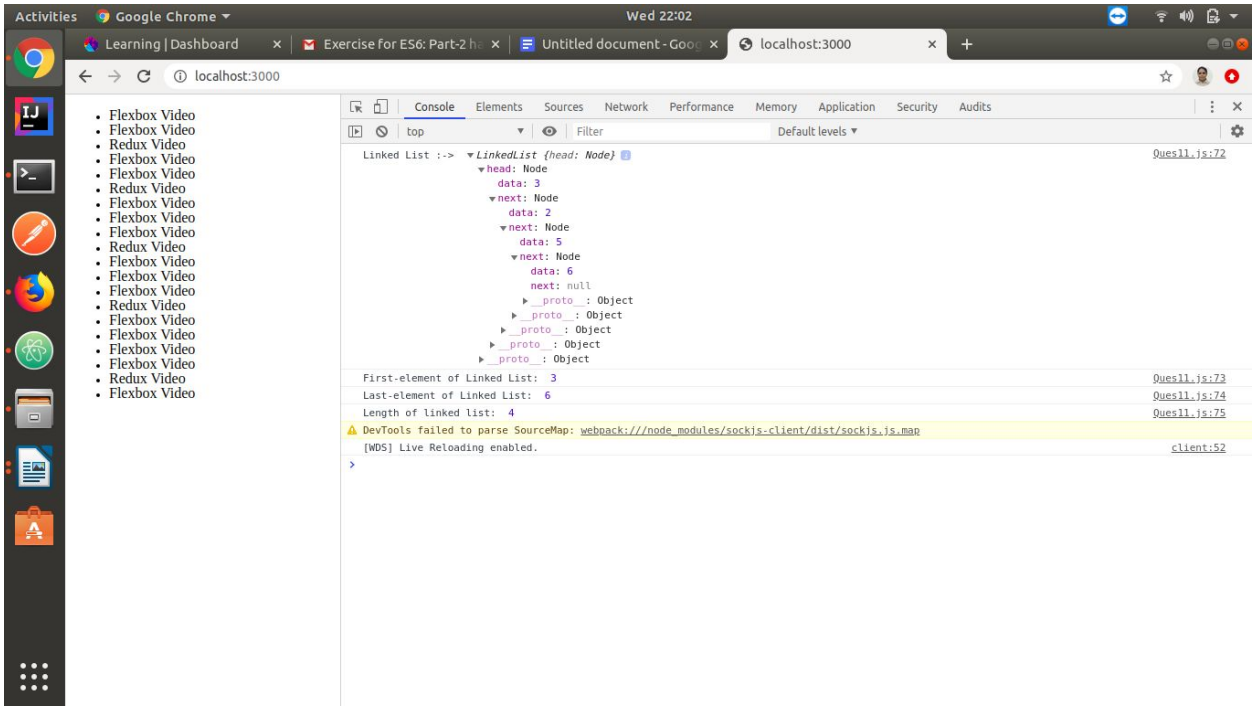
```
class Node {
constructor(data, next = null) {
this.data = data;
this.next = next;
}
}

class LinkedList {
constructor() {
this.head = null;
}

addFirst(data) {
let newNode = new Node(data);
//The pointer next is assigned head pointer so that both pointers now point at the same node.
newNode.next = this.head;
//As we are inserting at the beginning the head pointer needs to now point at the newNode.
this.head = newNode;
}


addLast(data) {
let newNode = new Node(data);
// When head = null i.e. the list is empty, then head itself will point to the newNode.
if (!this.head) {
this.head = newNode;
}
```

// Else, traverse the list to find the tail (the tail node will initially be pointing at null), and update the tail's next pointer.

```
let tail = this.head;

while (tail.next != null) {

tail = tail.next;

}

tail.next = newNode;

}


lengthOfList() {

var count = 1;

let tail = this.head;

while (tail.next != null) {

tail = tail.next;

count += 1;

}

return count;

}



getLast() {

let tail = this.head;

while (tail.next != null)

tail = tail.next;

return tail.data;

}



getFirst() {
```

```
return this.head.data;
}
}


let list=new LinkedList();
list.addFirst(5);
list.addFirst(2);
list.addFirst(3);
list.addLast(6);
console.log("Linked List :-> ",list);
console.log("First-element of Linked List: ",list.getFirst());
console.log("Last-element of Linked List: ",list.getLast());
console.log("Length of linked list: ",list.lengthOfList());
export {LinkedList}
```

OUTPUT:

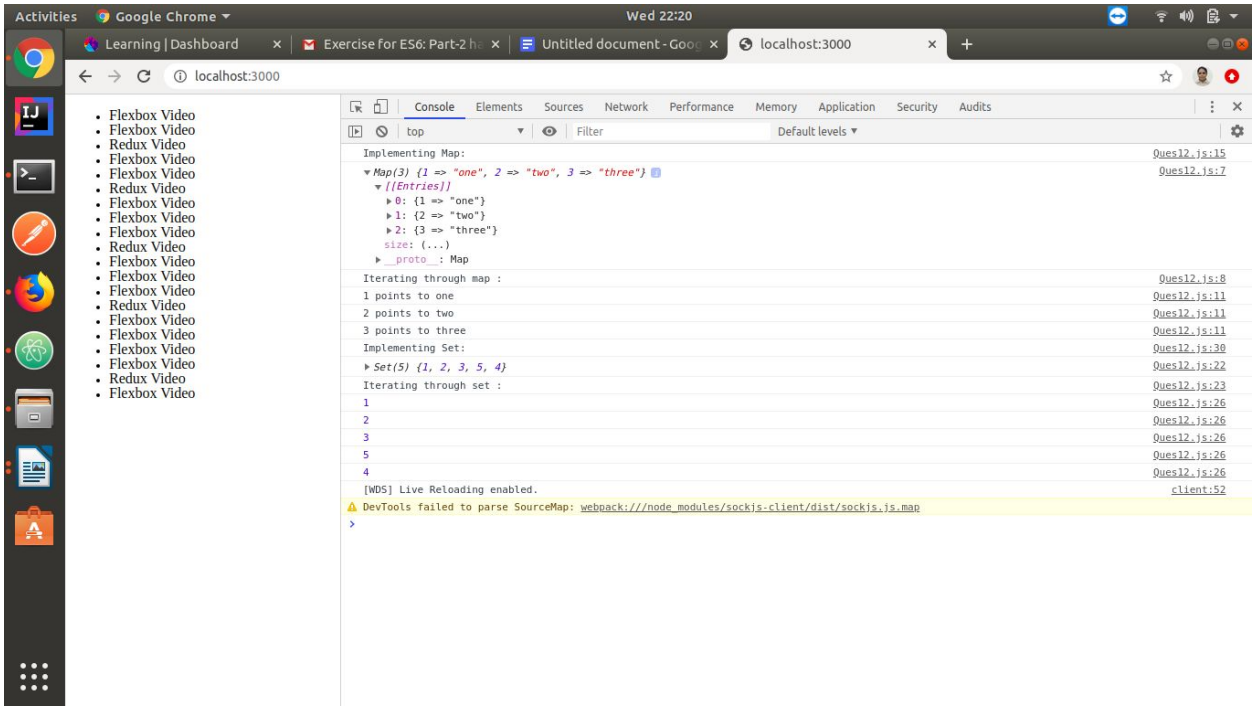


**Q12. Implement Map and Set using Es6**

```
// Q12. Implement Map using Es6
const arrayToMap = (array) => {
let valMap = new Map(array);
console.log(valMap);
console.log("Iterating through map : ")
for (let [key, value] of valMap.entries()) {
  console.log(`${key} points to ${value}`);
}}

console.log("Implementing Map: ");
let array1=[[1,"one"],[2,"two"],[3,"three"]];
arrayToMap(array1);

export {arrayToMap};
//Q12. Implement Set using Es6
const arrayToSet = (array) => {
let valSet = new Set(array);
console.log(valSet);
console.log("Iterating through set : ")
for (let elements of valSet.values()) {
console.log(elements);
}}
console.log("Implementing Set: ");
let array2=[1,2,3,5,4,1];
arrayToSet(array2);
```

export {arrayToSet};

OUTPUT:



**Q13. Implementation of stack (using linked list)**

```
function stackUsingLL() {

let Node = function (elm) {
this.element = elm;
this.next = null;
}

//To keep track of the size
let length = 0;

//To keep track of the list
let head = null;

//Push data in the stack
this.push = function (elm) {
//Create a new node
let node = new Node(elm);
var current;
//Add the new node at the top
current = head;
node.next = current;
head = node;
length++;
}


//Pop the item from the stack
```

```
this.pop = function () {
let current = head;
//If there is item then remove it
//and make the next element as the first
if (current) {
let elm = current.element;
current = current.next;
head = current;
length--;
return elm;
}
return null;
}


//Return the first element in the stack
this.peek = function () {
if (head) {
return head.element;
}
return null;
}


//Convert the stack to an array
this.toArray = function () {
let arr = [];
let current = head;
```

```
while (current) {
arr.push(current.element);
current = current.next;
}
return arr;
}


//Check if stack is empty
this.isEmpty = function () {
return length === 0;
}


//Return the size of the stack
this.size = function () {
return length;
}


//Clear the stack
this.clear = function () {
head = null; length = 0;
}
}
```

```
let stack = new stackUsingLL();
stack.push(10);
stack.push(20);
stack.push(30);
console.log("Stack : ",stack.toArray());
console.log("top element of stack",stack.peek());
console.log("is stack empty",stack.isEmpty());
console.log("stack size-",stack.size());
console.log("pop function->",stack.pop());
console.log("stack after pop",stack.toArray());
console.log("stack size-",stack.size());
console.log("clearing the stack");
stack.clear(); //clear the stack
console.log("is stack empty",stack.isEmpty());
export{stackUsingLL}
```

OUTPUT: