

Lesson 4 Demo 1

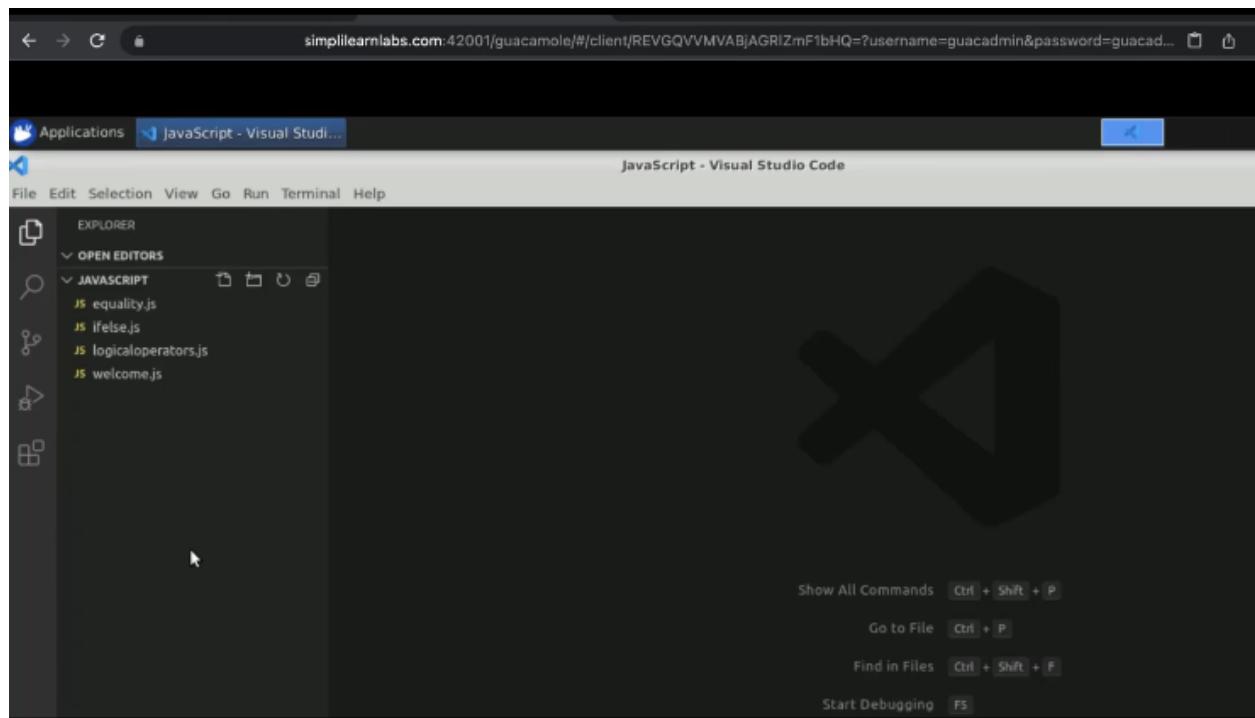
Creating Product User Object for eCommerce Store

Objective: To create a product user object for eCommerce store

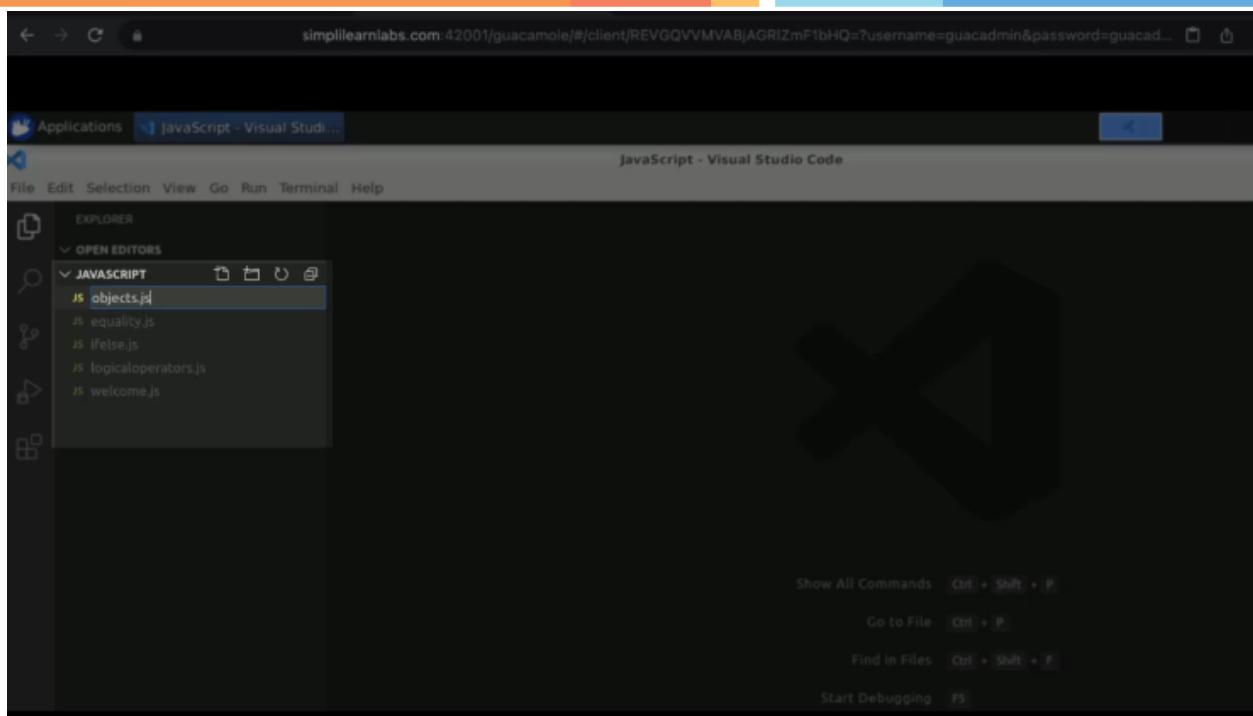
Prerequisites: None

Steps to perform:

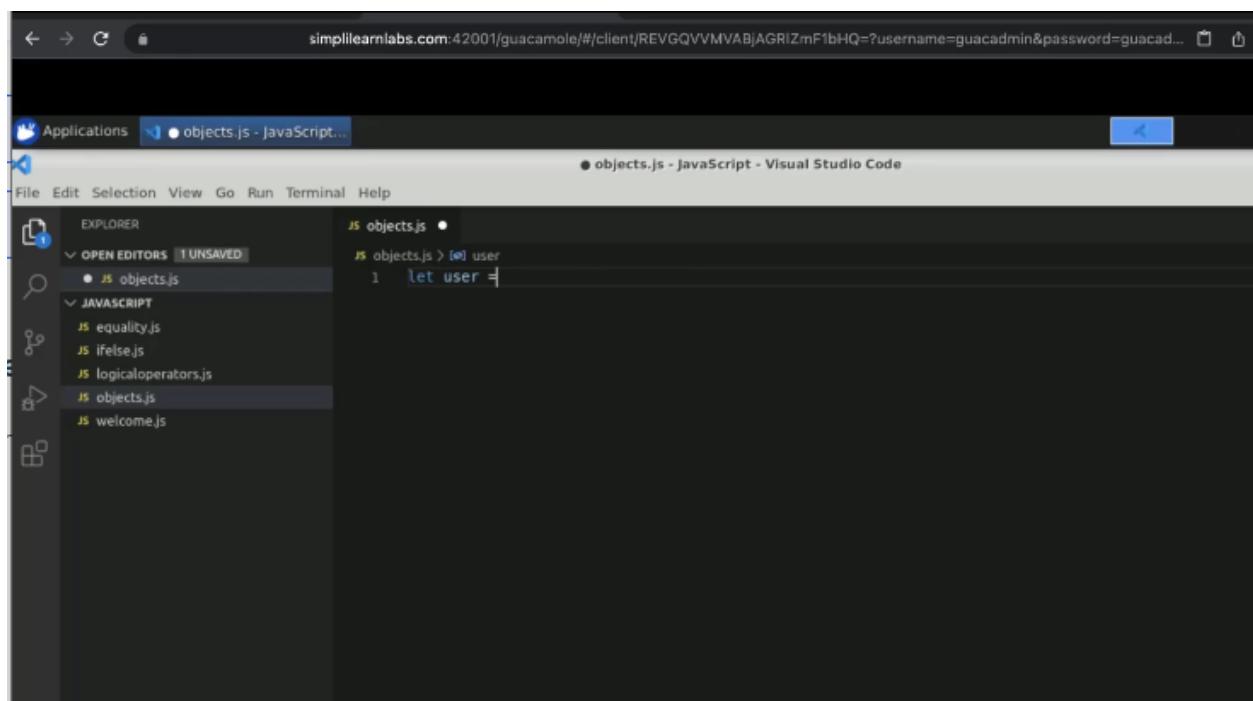
1. Open visual studio code.



2. Create a new file named 'object.js'.



3. Create an object as 'user'.



4. Add key and value for an object.

```

File Edit Selection View Go Run Terminal Help
EXPLORER JS objects.js •
OPEN EDITORS 1 UNSAVED
• JS objects.js
JAVASCRIPT JS equality.js
JS ifelse.js
JS logicaloperators.js
JS objects.js
JS welcome.js
JS objects.js > [0] user > [0] name
1 let user = []
2   name: 'John Watson'
3

```

5. Pass e-mail, phone number, and address, where the address can be in a nested form.

```

File Edit Selection View Go Run Terminal Help
EXPLORER JS objects.js •
OPEN EDITORS 1 UNSAVED
• JS objects.js
JAVASCRIPT JS equality.js
JS ifelse.js
JS logicaloperators.js
JS objects.js
JS welcome.js
JS objects.js > [0] user > [0] address > [0] zipCode
1 let user = {
2   name: 'John Watson',
3   email: 'john@example.com',
4   phone: '+91 99999 11111',
5   address: [
6     {
7       addressLine: '2144 Redwood Shores',
8       city: 'Banagalore',
9       zipCode: 520001
10    }
11  }

```

6. Add a log statement for object 'user'.

```

File Edit Selection View Go Run Terminal Help
OPEN EDITORS JS objects.js
JAVASCRIPT JS equality.js
JS ifelse.js
JS logicaloperators.js
JS objects.js
JS welcome.js
JS objects.js
let user = {
    name: 'John Watson',
    email: 'john@example.com',
    phone: '+91 99999 11111',
    address: {
        adrLine: '2144 Redwood Shores',
        city: 'Banagalore',
        zipCode: 520001
    }
}
console.log(user);
|

```

- Add one more log statement as 'type of user'.

```

File Edit Selection View Go Run Terminal Help
OPEN EDITORS JS objects.js
JAVASCRIPT JS equality.js
JS ifelse.js
JS logicaloperators.js
JS objects.js
JS welcome.js
JS objects.js
let user = {
    (property) email: string
    email: 'john@example.com',
    phone: '+91 99999 11111',
    address: {
        adrLine: '2144 Redwood Shores',
        city: 'Banagalore',
        zipCode: 520001
    }
}
console.log(user);
console.log(typeof user);
|

```

- Execute the program in the terminal.

applications objects.js - JavaScript - ...

Edit Selection View Go Run Terminal Help

EXPLORER JS objects.js x

OPEN EDITORS JS objects.js > ...

JAVASCRIPT JS equality.js JS ifelse.js JS logicaloperators.js JS objects.js JS welcome.js

```

JS objects.js > ...
1 let user = {
2   name: 'John Watson',
3   email: 'john@example.com',
4   phone: '+91 99999 11111',
5   address: {
6     adrLine: '2144 Redwood Shores',
7     city: 'Banagalore',
8     zipCode: 520001
9   }
10 }
11
12 console.log(user);
13 console.log(typeof user);
14
15

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: bash

```

email: 'john@example.com',
phone: '+91 99999 11111',
address: {
  adrLine: '2144 Redwood Shores',
  city: 'Banagalore',
  zipCode: 520001
}

```

> OUTLINE > TIMELINE object erishant@gmail.ip-172-31-90-232:~/Desktop/JavaScripts

9. Create another key called 'orders'.

applications objects.js - JavaScript - ...

Edit Selection View Go Run Terminal Help

EXPLORER JS objects.js ●

OPEN EDITORS 1 UNSAVED JS objects.js 1

JAVASCRIPT JS equality.js JS ifelse.js JS logicaloperators.js JS objects.js JS welcome.js

```

JS objects.js > [0] user > ↵ orders
1 let user = [
2   name: 'John Watson',
3   email: 'john@example.com',
4   phone: '+91 99999 11111',
5   address: {
6     adrLine: '2144 Redwood Shores',
7     city: 'Banagalore',
8     zipCode: 520001
9   },
10   orders: []
11 ]
12
13 console.log(user);
14 console.log(typeof user);
15
16

```

PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE 1: bash

```

email: 'john@example.com',
phone: '+91 99999 11111',
address: {
  adrLine: '2144 Redwood Shores',
  city: 'Banagalore',
  zipCode: 520001
}

```

> OUTLINE > TIMELINE object erishant@gmail.ip-172-31-90-232:~/Desktop/JavaScripts

10. Pass order id, amount and date in 'orders'.

applications objects.js - JavaScript... ● objects.js - JavaScript - Visual Studio Code

Edit Selection View Go Run Terminal Help

EXPLORER JS objects.js ●

OPEN EDITORS 1 UNSAVED ● JS objects.js

JAVASCRIPT JS equality.js JS ifelse.js JS logicaloperators.js JS objects.js JS welcome.js

```

JS objects.js > [0] user > [0] orders
1 let user = {
2   name: 'John Watson',
3   email: 'john@example.com',
4   phone: '+91 99999 11111',
5   address: {
6     adrLine: '2144 Redwood Shores',
7     city: 'Banagalore',
8     zipCode: 520001
9   },
10  orders:[
11    {
12      oid: 1,
13      amount: 2000,
14      date: '20th Nov, 2021'
15    }
16  ]

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE t: bash

```

email: 'john@example.com',
phone: '+91 99999 11111',
address: {
adrLine: '2144 Redwood Shores',
city: 'Banagalore',
zipCode: 520001
}

```

> OUTLINE > TIMELINE object erishant@gmail.ip-172-31-90-232:~/Desktop/JavaScript\$

11. Add another object under the 'orders'.

applications objects.js - JavaScript - ... objects.js - JavaScript - Visual Studio Code

Edit Selection View Go Run Terminal Help

EXPLORER JS objects.js ●

OPEN EDITORS X JS objects.js

JAVASCRIPT JS equality.js JS ifelse.js JS logicaloperators.js JS objects.js JS welcome.js

```

JS objects.js > [0] user > [0] orders > [0] date
9   },
10  orders:[
11    {
12      oid: 1,
13      amount: 2000,
14      date: '20th Nov, 2021'
15    },
16    [
17      {
18        oid: 11,
19        amount: 1000,
20        date: '25th Nov, 2021'
21      }
22    ]
23
24 console.log(user);

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE t: bash

```

email: 'john@example.com',
phone: '+91 99999 11111',
address: {
adrLine: '2144 Redwood Shores',
city: 'Banagalore',
zipCode: 520001
}

```

> OUTLINE > TIMELINE object erishant@gmail.ip-172-31-90-232:~/Desktop/JavaScript\$

12. Re-run the program.

```

applications(objects.js - JavaScript - ...)

Edit Selection View Go Run Terminal Help
EXPLORER JS objects.js ×
OPEN EDITORS JS objects.js > [0] user > [0] orders > [0] date
JAVASCRIPT JS equality.js
JS ifelse.js
JS logicaloperators.js
JS objects.js
JS welcome.js
JS objects.js
    9     },
    10    orders:[
    11        {
    12            oid: 1,
    13            amount: 2000,
    14            date: '20th Nov, 2021'
    15        },
    16        [
    17            {
    18                oid: 11,
    19                amount: 1000,
    20                date: '25th Nov, 2021'
    21            }
    22        ]
    23    }
    24    console.log(user);
        city: 'Banagalore',
        zipCode: 520001
    },
    orders: [
        { oid: 1, amount: 2000, date: '20th Nov, 2021' },
        { oid: 11, amount: 1000, date: '25th Nov, 2021' }
    ]
}
object
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScripts []

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

1: bash

13. Add a log again and access the attributes of the key.

```

applications(objects.js - JavaScript - ...)

Edit Selection View Go Run Terminal Help
EXPLORER JS objects.js ×
OPEN EDITORS JS objects.js > ...
JAVASCRIPT JS equality.js
JS ifelse.js
JS logicaloperators.js
JS objects.js
JS welcome.js
JS objects.js
    15     },
    16     [
    17         {
    18             oid: 11,
    19             amount: 1000,
    20             date: '25th Nov, 2021'
    21         }
    22     ]
    23
    24     console.log(user);
    25     console.log(typeof user);
    26
    27     console.log("User name is: "+user.name);
    28     console.log("Orders Place are:"+user['orders']);
    29
    30

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

1: bash

14. Execute the program.

The screenshot shows a Visual Studio Code interface with the following details:

- Title Bar:** applications objects.js - JavaScript - ...
- Header:** Edit Selection View Go Run Terminal Help
- Left Sidebar (EXPLORER):** OPEN EDITORS (JS objects.js), JAVASCRIPT (equality.js, ifelse.js, logicaloperators.js, objects.js, welcome.js)
- Editor Area:** JS objects.js (Content of the file is shown below)
- Bottom Navigation:** PROBLEMS, OUTPUT, TERMINAL, DEBUG CONSOLE
- Bottom Status Bar:** 1: bash
- Terminal Output:** object User name is: John Watson Orders Place are:[object Object],[object Object] erishant@gmail.com:ip-172-31-90-232:/Desktop/JavaScript\$

```
15     },
16     {
17       oid: 11,
18       amount: 1000,
19       date: '25th Nov, 2021'
20     }
21   ]
22 }
23
24 console.log(user);
25 console.log(typeof user);
26
27 console.log("User name is: "+user.name);
28 console.log("Orders Place are:"+user['orders']);
29
30
31 },
32 orders: [
33   { oid: 1, amount: 2000, date: '20th Nov, 2021' },
34   { oid: 11, amount: 1000, date: '25th Nov, 2021' }
35 ]
36 }
```

15. Use User of orders zero indexes.

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** applications • objects.js - JavaScript...
- File Explorer:** Shows the project structure under "OPEN EDITORS":
 - JS objects.js (1)
 - JAVASCRIPT
 - JS equality.js
 - JS ifelse.js
 - JS logicaloperators.js
 - JS objects.js (1)
 - JS welcome.js
- Code Editor:** The "objects.js" file is open, displaying the following code:

```
15      },
16      {
17          oid: 11,
18          amount: 1000,
19          date: '25th Nov, 2021'
20      }
21  ]
22 }
23
24 console.log(user);
25 console.log(typeof user);
26
27 console.log("User name is: "+user.name);
28 console.log("Orders Place are:"+user['orders'][0]);
29
30
```
- Bottom Status Bar:** object
User name is: John Watson
Orders Place are:[object Object],[object Object]
erishant@gmail.com@ip-172-31-90-232:~/Desktop/JavaScript\$

16. Save it and re-run the code.

```

applications(objects.js - JavaScript - ...)

Edit Selection View Go Run Terminal Help
EXPLORER JS objects.js ...
OPEN EDITORS JS objects.js ...
JAVASCRIPT JS equality.js
JS ifelse.js
JS logicaloperators.js
JS objects.js
JS welcome.js
JS objects.js > ...
15     },
16     {
17         oid: 11,
18         amount: 1000,
19         date: '25th Nov, 2021'
20     }
21 ]
22 }
23
24 console.log(user);
25 console.log(typeof user);
26
27 console.log("User name is: "+user.name);
28 console.log("Orders Place are:"+user['orders'][0]);
29
30
},
orders: [
    { oid: 1, amount: 2000, date: '20th Nov, 2021' },
    { oid: 11, amount: 1000, date: '25th Nov, 2021' }
]
}
object
User name is: John Watson
Orders Place are:[Object Object]
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ 1: bash

```

17. Nest the statement one more time with the attribute as 'amount'.

```

applications(objects.js - JavaScript - ...)

Edit Selection View Go Run Terminal Help
EXPLORER JS objects.js •
OPEN EDITORS 1 UNSAVED JS objects.js ...
JAVASCRIPT JS equality.js
JS ifelse.js
JS logicaloperators.js
JS objects.js
JS welcome.js
JS objects.js > ...
15     },
16     {
17         oid: 11,
18         amount: 1000,
19         date: '25th Nov, 2021'
20     }
21 ]
22 }
23
24 console.log(user);
25 console.log(typeof user);
26
27 console.log("User name is: "+user.name);
28 console.log("Orders Place are:"+user['orders'][0][0].amount);
29
30
},
orders: [
    { oid: 1, amount: 2000, date: '20th Nov, 2021' },
    { oid: 11, amount: 1000, date: '25th Nov, 2021' }
]
}
object
User name is: John Watson
Orders Place are:[Object Object]
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ 1: bash

```

18. Create an object called 'product'.

19. Add product id, name, brand, and price as attributes.

The screenshot shows a Visual Studio Code interface with the following details:

- Title Bar:** applications objects.js - JavaScript...
- File Explorer (Left):** Shows the project structure with files: JS objects.js (selected), JS equality.js, JS ifelse.js, JS logicaloperators.js, JS objects.js, and JS welcome.js.
- Code Editor (Center):** Displays the contents of the objects.js file.

```
JS objects.js > [0] product > ↵ price
20     }
21   ]
22 }
23
24 console.log(user);
25 console.log(typeof user);
26
27 console.log("User name is: "+user.name);
28 console.log("Orders Placed are:"+user['orders'][0]['amount']);
29
30 const product = [
31   {
32     pid: 101,
33     name: 'Alphabounce',
34     brand: 'Adidas',
35     price: 5000
36   }
37 ],
38 orders: [
39   { oid: 1, amount: 2000, date: '20th Nov, 2021' },
40   { oid: 11, amount: 1000, date: '25th Nov, 2021' }
41 ]
42
43 object
44 User name is: John Watson
45 Orders Place are:2000
46 erishant@gmail.com@ip-172-31-90-232:-/Desktop/JavaScript$
```
- Bottom Status Bar:** 1: bash

20. Add more attributes as 'inventory'. Pass objects within it such as; size, units, etc

The screenshot shows a Visual Studio Code interface with the following details:

- Title Bar:** applications objects.js - JavaScript...
- File Explorer:** Shows multiple JavaScript files: equality.js, ifelse.js, logicaloperators.js, objects.js (selected), and welcome.js.
- Code Editor:** Displays the contents of objects.js. The code defines a product object with a user name and orders, and an inventory array containing two items. Each item has a size and units attribute.

```

JS objects.js •
JS objects.js > [0] product > [0] inventory
25   [
26     {
27       console.log("User name is: "+user.name);
28       console.log("Orders Placed are:"+user['orders'][0]['amount']);
29     }
30   const product = {
31     pid: 101,
32     name: 'Alphabounce',
33     brand: 'Adidas',
34     price: 5000,
35     inventory: [
36       {
37         size: 7,
38         units: 9
39       }
40     ],
41   },
42   orders: [
43     { oid: 1, amount: 2000, date: '20th Nov, 2021' },
44     { oid: 11, amount: 1000, date: '25th Nov, 2021' }
45   ]
46 }
47 object
48 User name is: John Watson
49 Orders Place are:2000
      
```

- Terminal:** Shows the command t: bash.
- Status Bar:** PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

21. Manage inventory for more objects with different sizes and units.

The screenshot shows a Visual Studio Code interface with the following details:

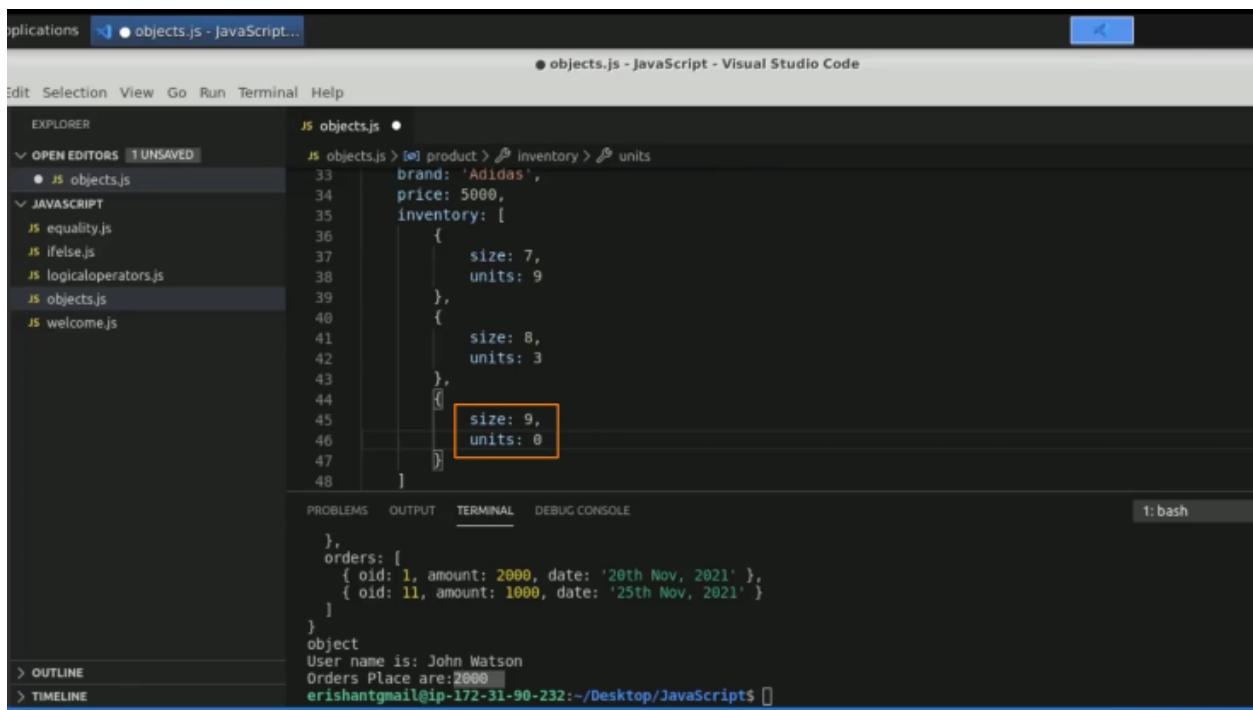
- Title Bar:** applications objects.js - JavaScript - ...
- File Explorer:** Shows multiple JavaScript files: equality.js, ifelse.js, logicaloperators.js, objects.js (selected), and welcome.js.
- Code Editor:** Displays the contents of objects.js. The code defines a product object with a user name and orders, and an inventory array containing two items. Each item has a size and units attribute. The second item's size and units are highlighted with a red box.

```

JS objects.js •
JS objects.js > [0] product > [0] inventory > [0] units
29
30   const product = {
31     pid: 101,
32     name: 'Alphabounce',
33     brand: 'Adidas',
34     price: 5000,
35     inventory: [
36       {
37         size: 7,
38         units: 9
39       },
40       {
41         size: 8,
42         units: 3
43       }
44     ],
45   },
46   orders: [
47     { oid: 1, amount: 2000, date: '20th Nov, 2021' },
48     { oid: 11, amount: 1000, date: '25th Nov, 2021' }
49   ]
50 }
51 object
52 User name is: John Watson
53 Orders Place are:2000
      
```

- Terminal:** Shows the command erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScripts \$.
- Status Bar:** PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

22. Some more objects can be as follows:

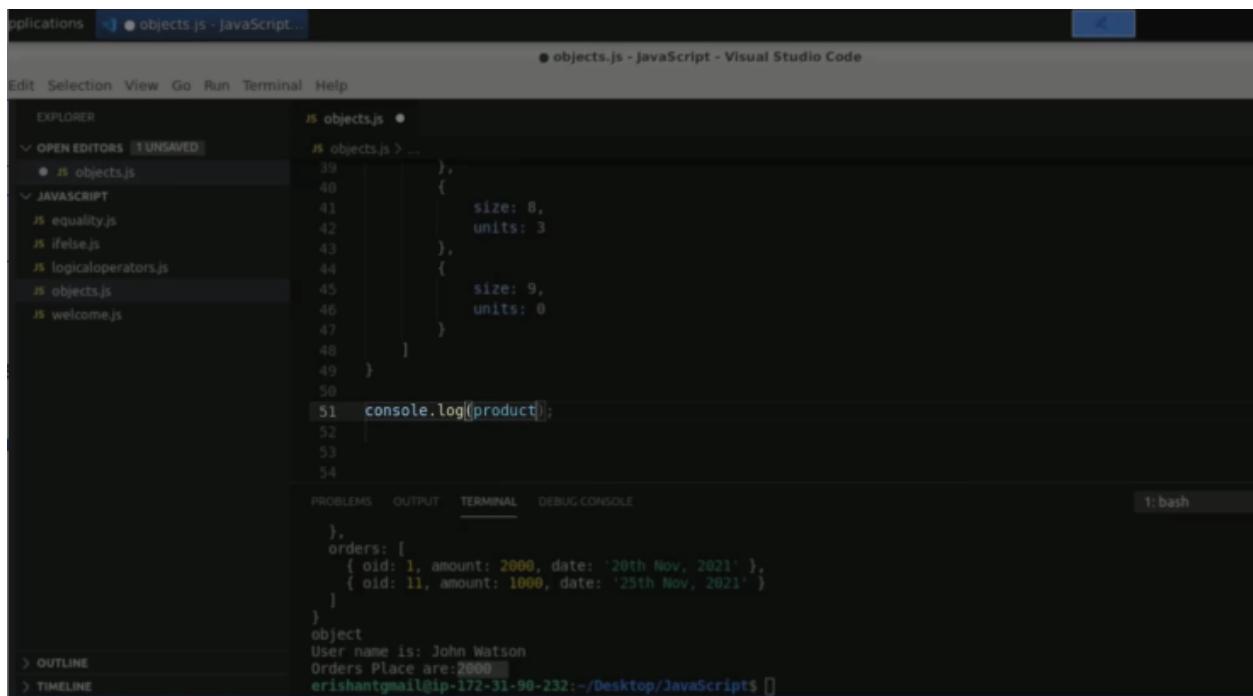


```

JS objects.js ●
JS objects.js > (o) product > (o) inventory > (o) units
33   brand: 'Adidas',
34   price: 5000,
35   inventory: [
36     {
37       size: 7,
38       units: 9
39     },
40     {
41       size: 8,
42       units: 3
43     },
44     [
45       {
46         size: 9,
47         units: 0
48       }
49     ]
50   ],
51   orders: [
52     { oid: 1, amount: 2000, date: '20th Nov, 2021' },
53     { oid: 11, amount: 1000, date: '25th Nov, 2021' }
54   ]
55 }
56 object
57 User name is: John Watson
58 Orders Place are:2000
59 erishant@gmail.ip-172-31-90-232:~/Desktop/JavaScript$ 

```

23. Log the product.



```

JS objects.js ●
JS objects.js > ...
39   },
40   {
41     size: 8,
42     units: 3
43   },
44   {
45     size: 9,
46     units: 0
47   }
48 ]
49 }
50
51 console.log([product]);
52
53
54
55 },
56 orders: [
57   { oid: 1, amount: 2000, date: '20th Nov, 2021' },
58   { oid: 11, amount: 1000, date: '25th Nov, 2021' }
59 ]
60 }
61 object
62 User name is: John Watson
63 Orders Place are:2000
64 erishant@gmail.ip-172-31-90-232:~/Desktop/JavaScript$ 

```

24. Save and run the code.

The screenshot shows the Visual Studio Code interface with the file 'objects.js' open. The code defines a product object with properties like pid, name, brand, price, and inventory. The inventory array has three elements: {size: 7, units: 9}, {size: 8, units: 3}, and {size: 9, units: 0}. A console.log statement is present at the end of the script. The terminal window shows the output: 'Orders Placed are:2000' followed by the object definition.

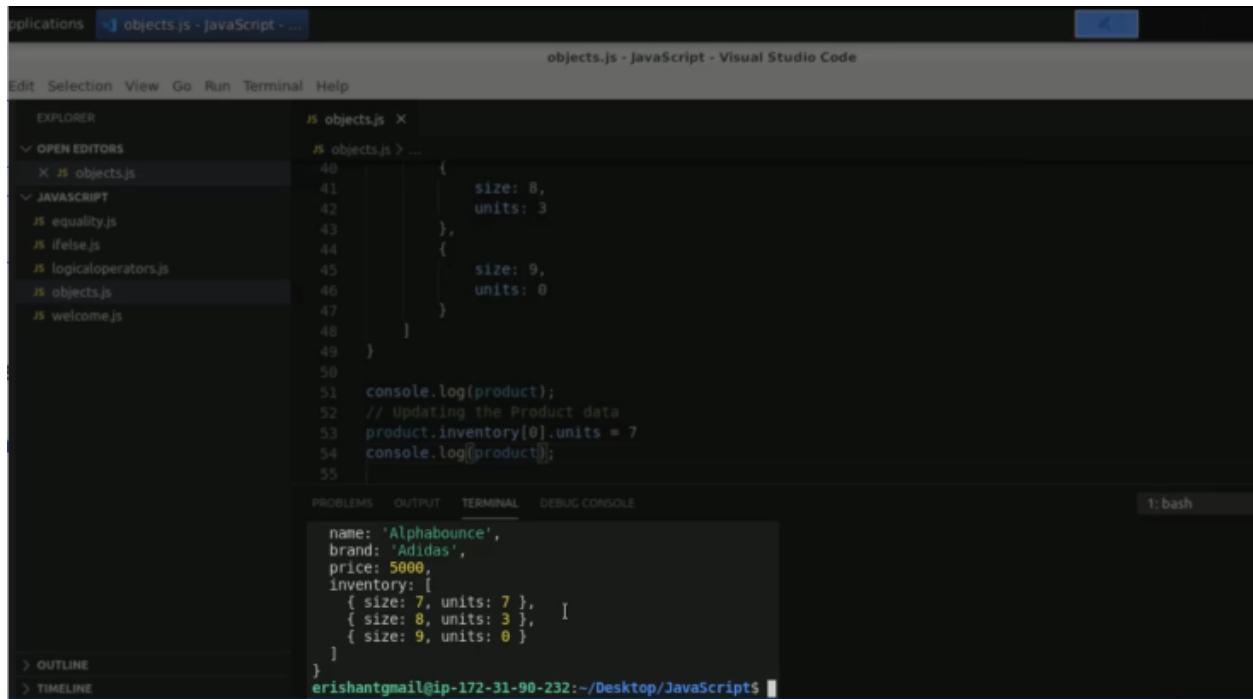
```
JS objects.js
39     },
40     {
41         size: 8,
42         units: 3
43     },
44     {
45         size: 9,
46         units: 0
47     }
48 }
49 }
50
51 console.log(product);
52
53
54
Orders Placed are:2000
{
  pid: 101,
  name: 'Alphabounce',
  brand: 'Adidas',
  price: 5000,
  inventory: [
    { size: 7, units: 9 },
    { size: 8, units: 3 },
    { size: 9, units: 0 }
```

25. Write product inventory as a zeroth index and update the units with seven.

The screenshot shows the Visual Studio Code interface with the file 'objects.js' open. The code is identical to the previous screenshot, but the inventory array now has its first element updated to {size: 7, units: 7}. A console.log statement is present at the end of the script. The terminal window shows the output: 'Orders Placed are:2000' followed by the object definition.

```
JS objects.js
39     },
40     {
41         size: 8,
42         units: 3
43     },
44     {
45         size: 9,
46         units: 0
47     }
48 }
49 }
50
51 console.log(product);
52 product.inventory[0].units = 7
53
54
Orders Placed are:2000
{
  pid: 101,
  name: 'Alphabounce',
  brand: 'Adidas',
  price: 5000,
  inventory: [
    { size: 7, units: 9 },
    { size: 8, units: 3 },
    { size: 9, units: 0 }
```

26. Run the program.

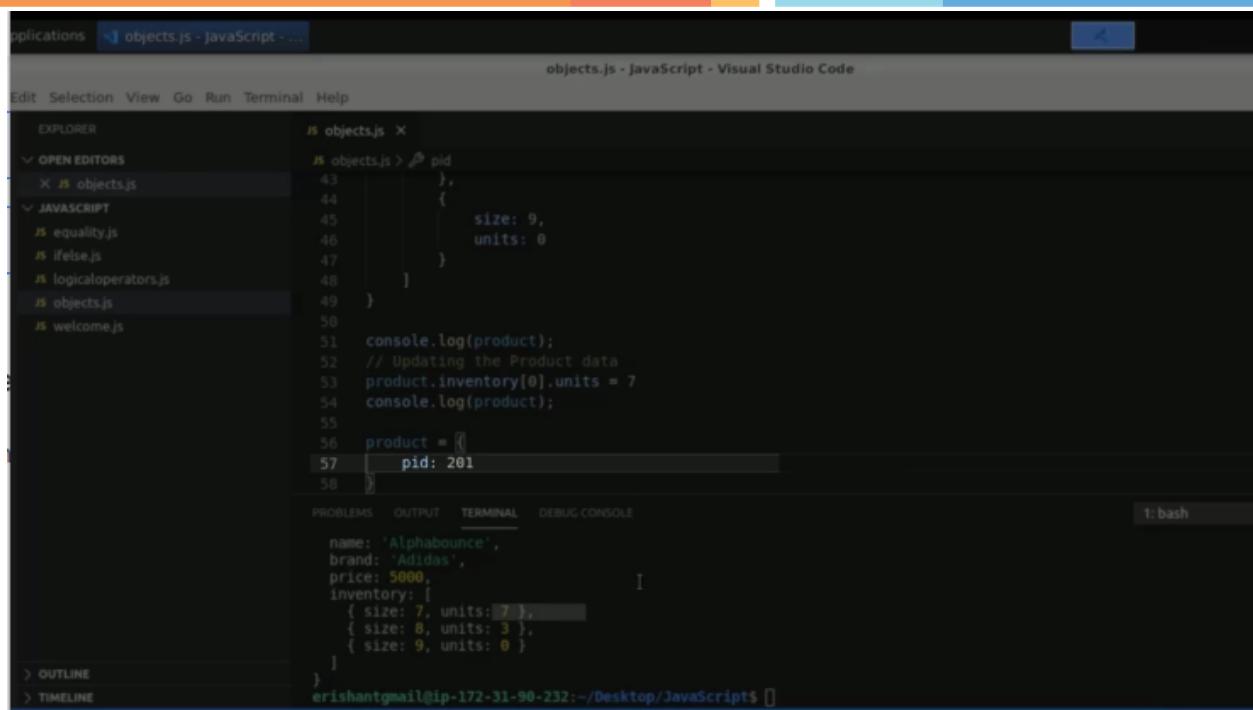


```
JS objects.js > ...
40     {
41         size: 8,
42         units: 3
43     },
44     {
45         size: 9,
46         units: 0
47     }
48 }
49 }
50
51 console.log(product);
52 // Updating the Product data
53 product.inventory[0].units = 7
54 console.log(product);
55
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
name: 'Alphabounce',
brand: 'Adidas',
price: 5000,
inventory: [
    { size: 7, units: 7 },
    { size: 8, units: 3 },
    { size: 9, units: 0 }
]
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScripts
```

27. Put a product id as 201 in a new product object.



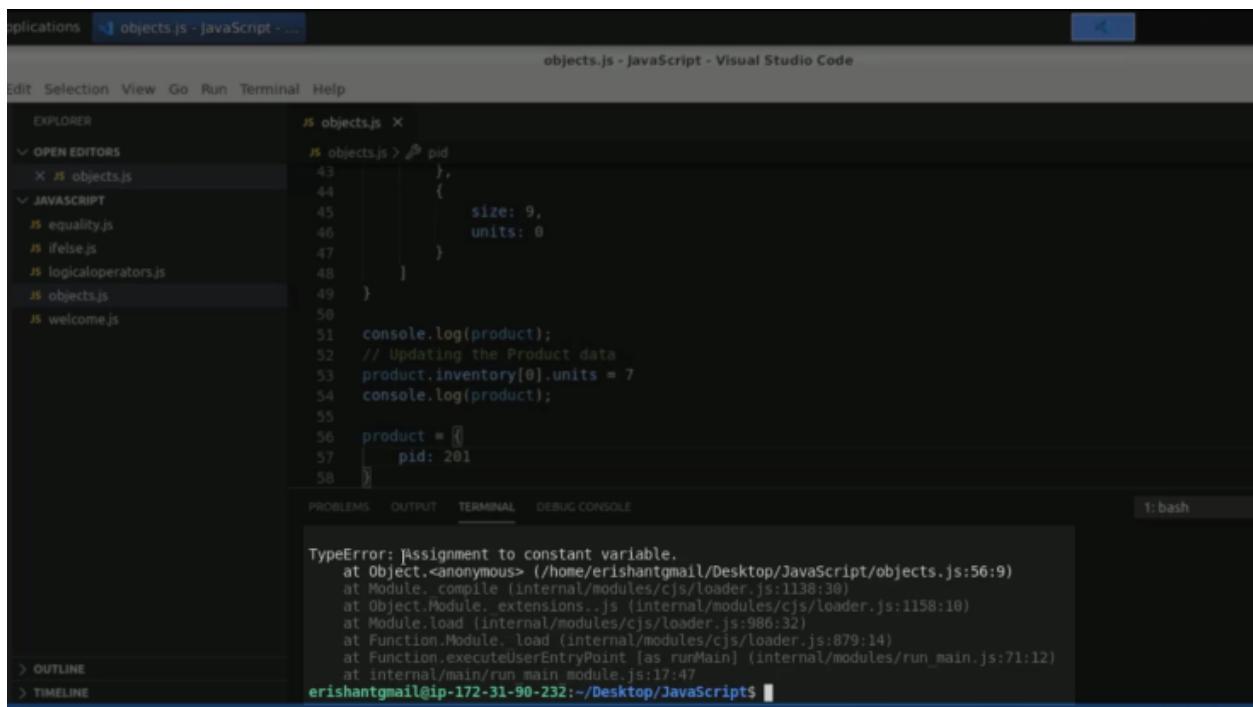
```

applications(objects.js - JavaScript - ...)

Edit Selection View Go Run Terminal Help
EXPLORER JS objects.js ×
OPEN EDITORS JS objects.js > pid
JAVASCRIPT equality.js
ifelse.js
logicaloperators.js
objects.js
welcome.js
JS objects.js
JS welcome.js
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
name: 'Alphabounce',
brand: 'Adidas',
price: 5000,
inventory: [
  { size: 7, units: 7 },
  { size: 8, units: 3 },
  { size: 9, units: 0 }
]
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ 1:bash
> OUTLINE
> TIMELINE

```

28. Run the program.



```

applications(objects.js - JavaScript - ...)

Edit Selection View Go Run Terminal Help
EXPLORER JS objects.js ×
OPEN EDITORS JS objects.js > pid
JAVASCRIPT equality.js
ifelse.js
logicaloperators.js
objects.js
welcome.js
JS objects.js
JS welcome.js
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
TypeError: Assignment to constant variable.
at Object.<anonymous> (/home/erishant@gmail/Desktop/JavaScript/objects.js:56:9)
at Module.compile (internal/modules/cjs/loader.js:1138:30)
at Object.Module._extensions..js (internal/modules/cjs/loader.js:1158:10)
at Module.load (internal/modules/cjs/loader.js:986:32)
at Function.Module._load (internal/modules/cjs/loader.js:879:14)
at Function.executeUserEntryPoint [as runMain] (internal/modules/run_main.js:71:12)
at internal/main/run_main_module.js:17:47
erishant@gmail@ip-172-31-90-232:~/Desktop/JavaScript$ 1:bash
> OUTLINE
> TIMELINE

```

