Lesson 04 Demo 01

Creating Product and User Objects for eCommerce Store

Objective: To demonstrate the creation of objects in JavaScript and their usage in an eCommerce store scenario

Tools Required: Visual Studio Code and Node.js

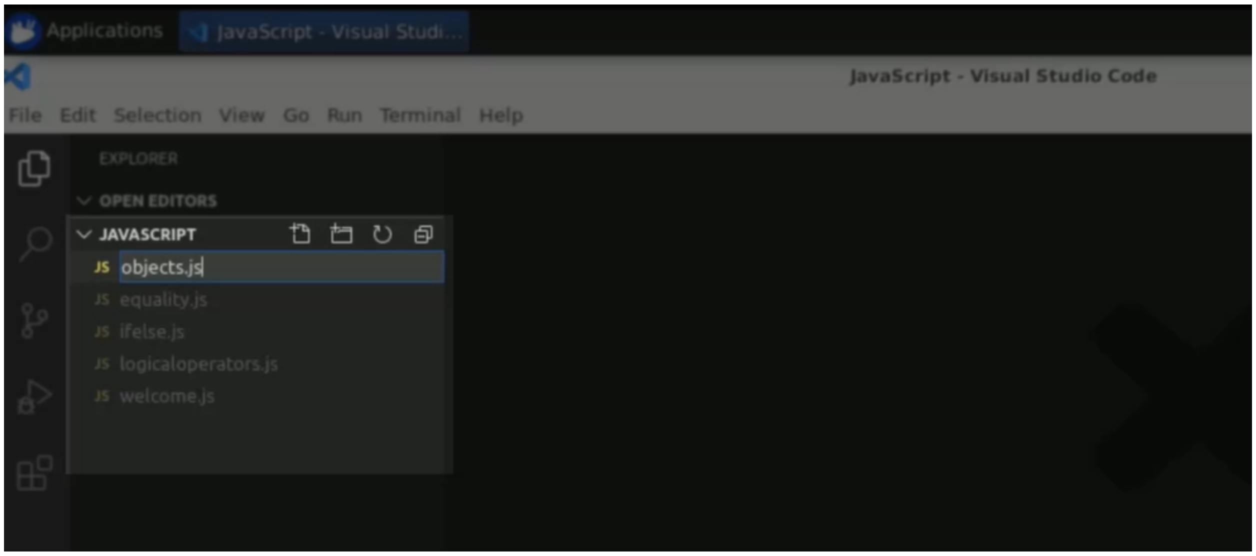
Prerequisites: None

Steps to be followed:

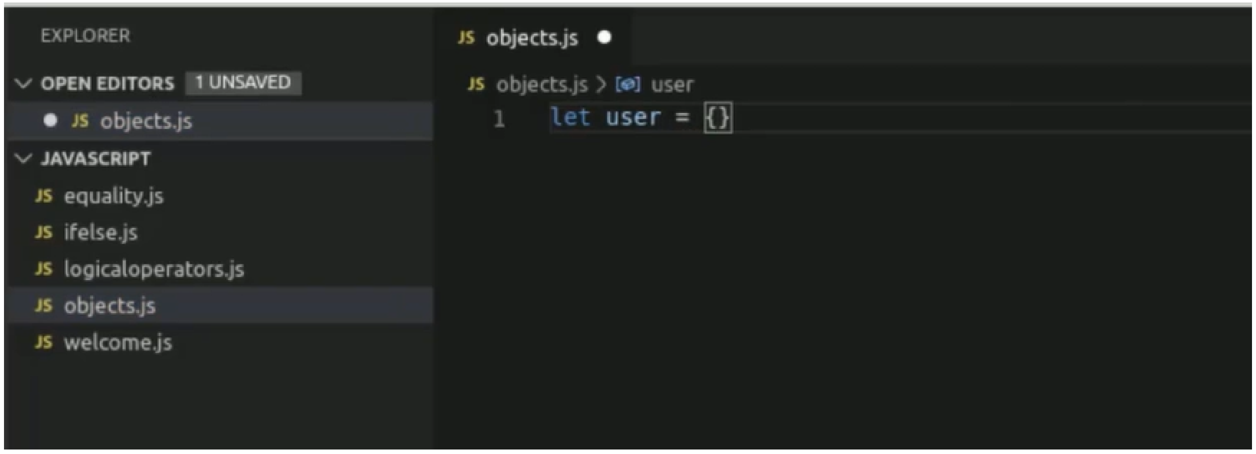
1. Create a user object
2. Create a product object

Step 1: Create a user object

1.1 Open Visual Studio Code and create a new file named **objects.js**

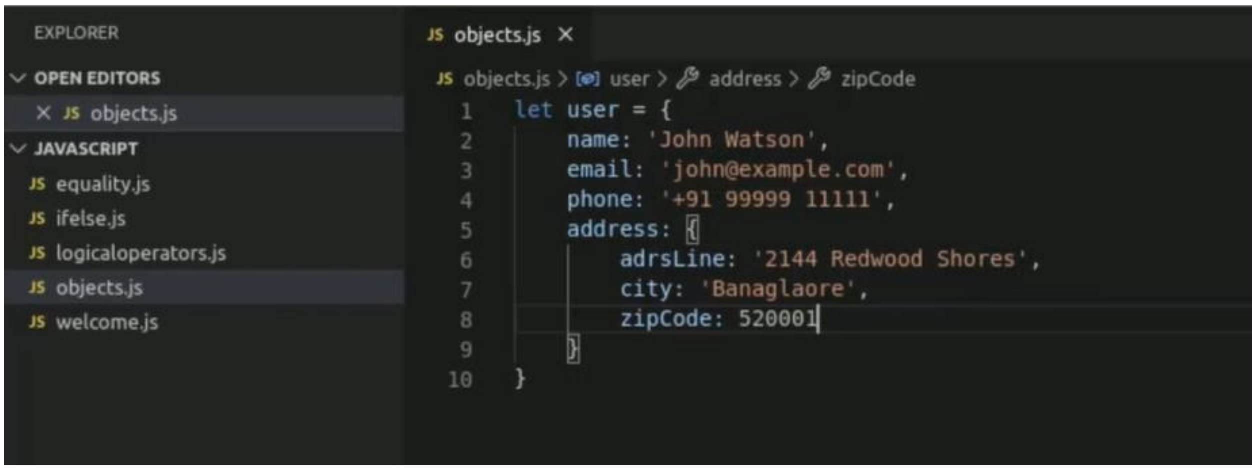

Applications
JavaScript - Visual Studi...
JavaScript - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
JAVASCRIPT
objects.js
equality.js
ifelse.js
logicaloperators.js
welcome.js

1.2 Declare an empty object called **user** using the object notation **{}**



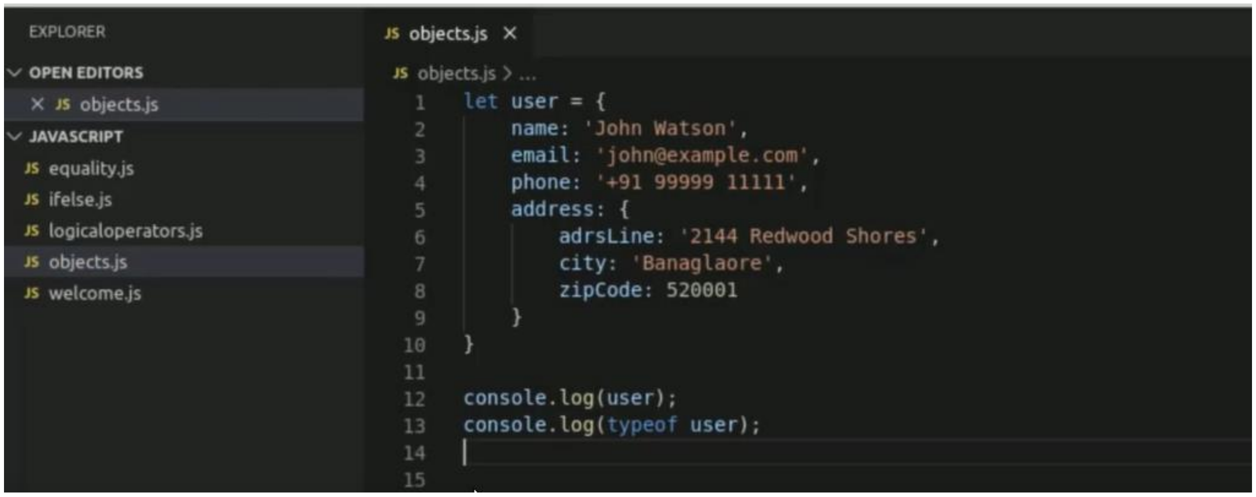
```
let user = {}
```

1.3 Add key-value pairs to the user object



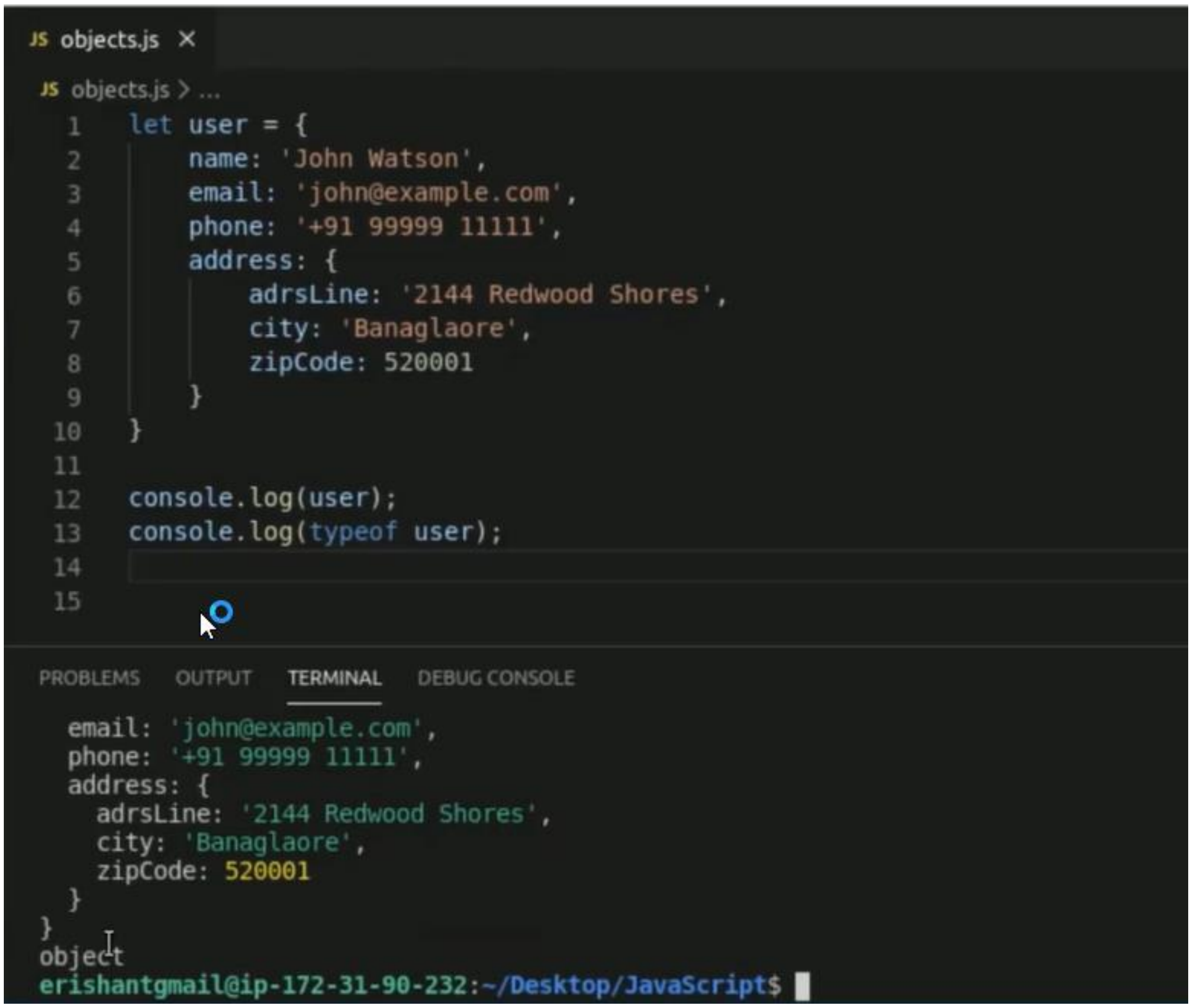
```
let user = {
    name: 'John Watson',
    email: 'john@example.com',
    phone: '+91 99999 11111',
    address: {
        adrsLine: '2144 Redwood Shores',
        city: 'Banaglaore',
        zipCode: 520001
    }
}
```

1.4 Use **console.log()** to print the user object



```
let user = {
    name: 'John Watson',
    email: 'john@example.com',
    phone: '+91 99999 11111',
    address: {
        adrsLine: '2144 Redwood Shores',
        city: 'Banaglaore',
        zipCode: 520001
    }
}

console.log(user);
console.log(typeof user);
```

1.5 Run the JavaScript program using **node objects.js**

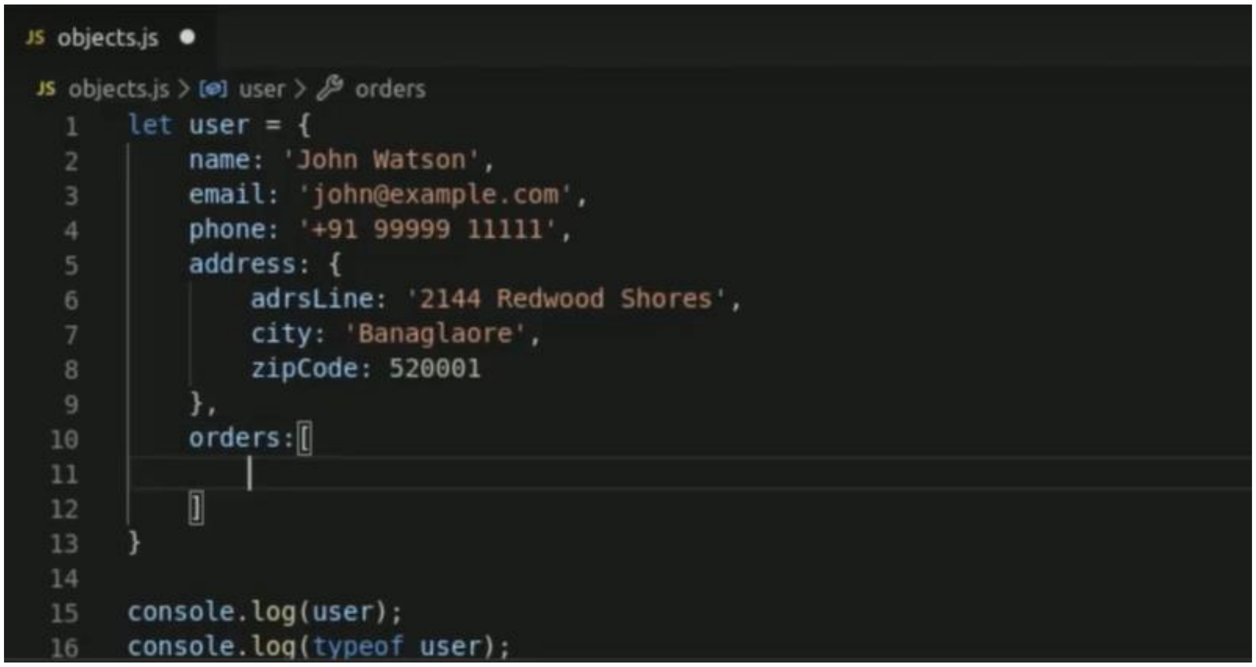


```javascript
let user = {
    name: 'John Watson',
    email: 'john@example.com',
    phone: '+91 99999 11111',
    address: {
        adrsLine: '2144 Redwood Shores',
        city: 'Banaglaore',
        zipCode: 520001
    }
}

console.log(user);
console.log(typeof user);
```

```
  email: 'john@example.com',
  phone: '+91 99999 11111',
  address: {
    adrsLine: '2144 Redwood Shores',
    city: 'Banaglaore',
    zipCode: 520001
  }
}
object
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$
```

Verify that the object contains the specified data

1.6 Add a key called **orders** to the user object and set the value of **orders** as an array **[]**

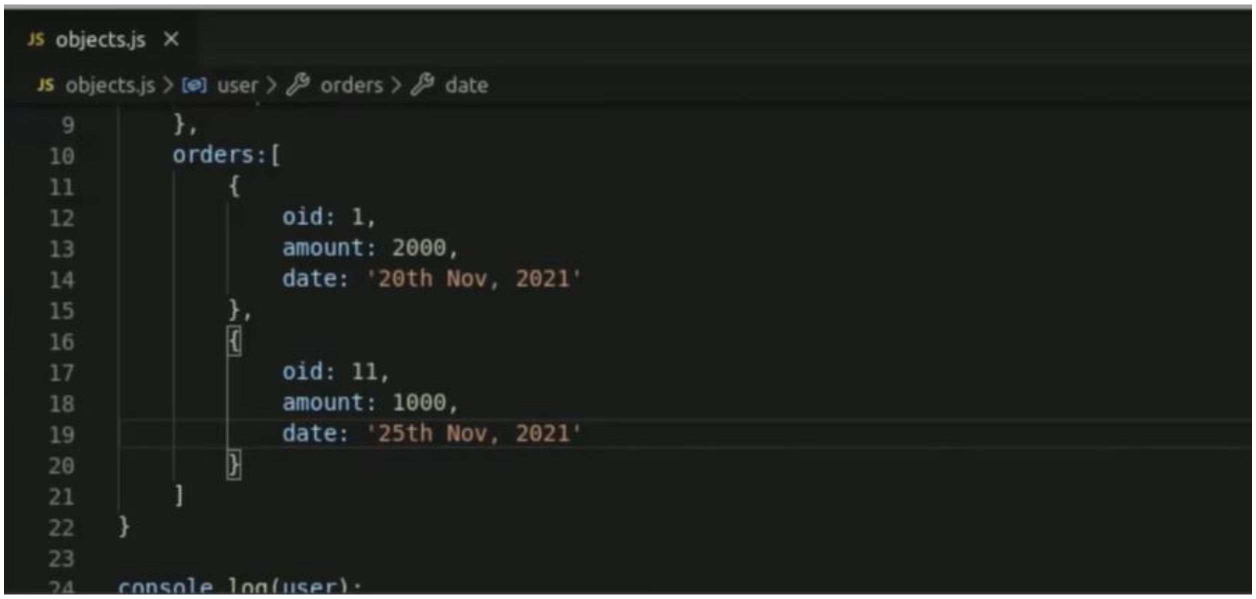


objects.js

objects.js > user > orders

```
let user = {
    name: 'John Watson',
    email: 'john@example.com',
    phone: '+91 99999 11111',
    address: {
        adrsLine: '2144 Redwood Shores',
        city: 'Banaglaore',
        zipCode: 520001
    },
    orders:[

    ]
}

console.log(user);
console.log(typeof user);
```

1.7 Add multiple objects representing orders to the **orders** array, each with properties such as **oid**, **amount**, and **date**

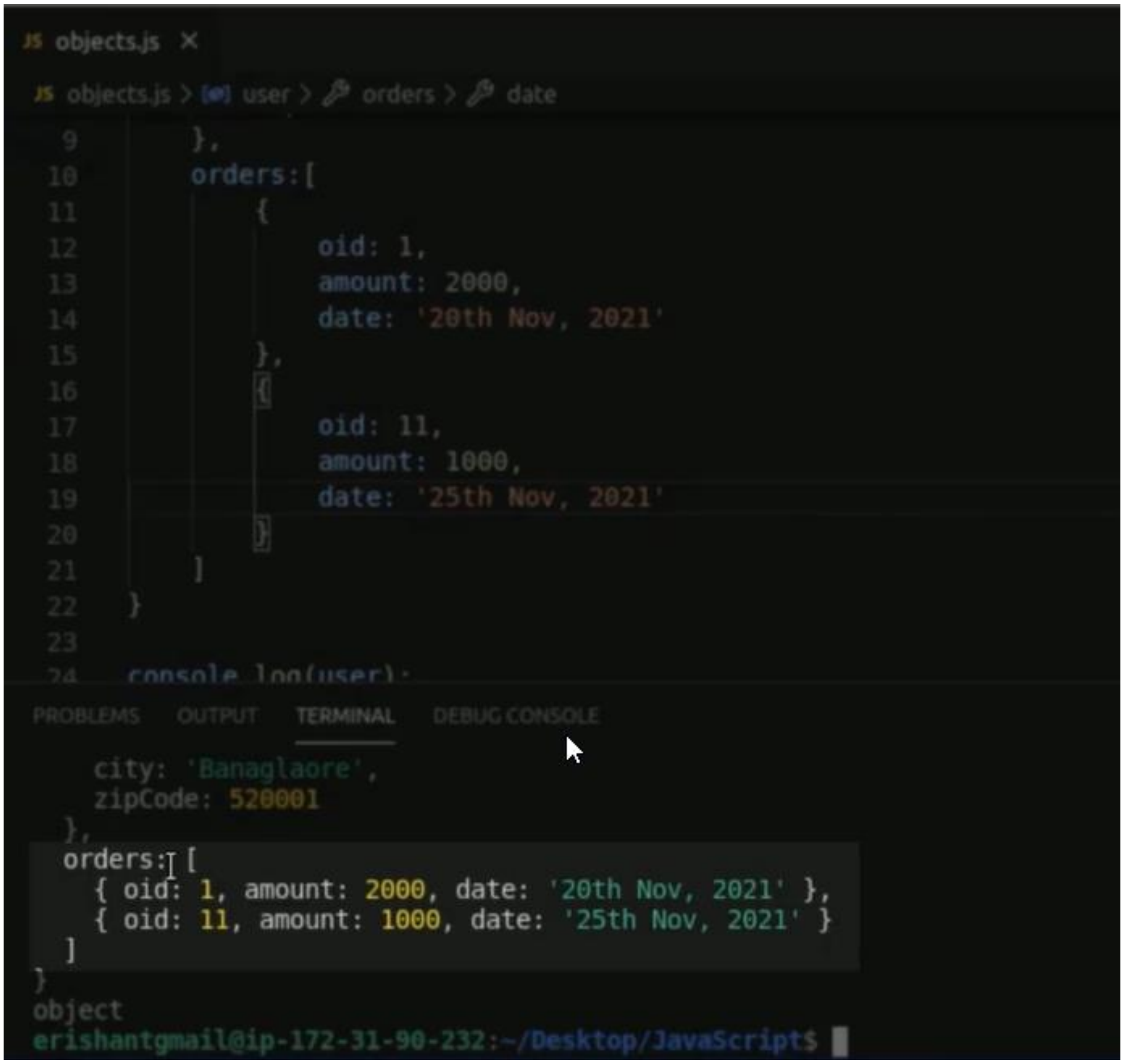


objects.js

objects.js > user > orders > date

```
    },
    orders:[
        {
            oid: 1,
            amount: 2000,
            date: '20th Nov, 2021'
        },
        {
            oid: 11,
            amount: 1000,
            date: '25th Nov, 2021'
        }
    ]
}

console.log(user);
```

1.8 Rerun the program and observe the output



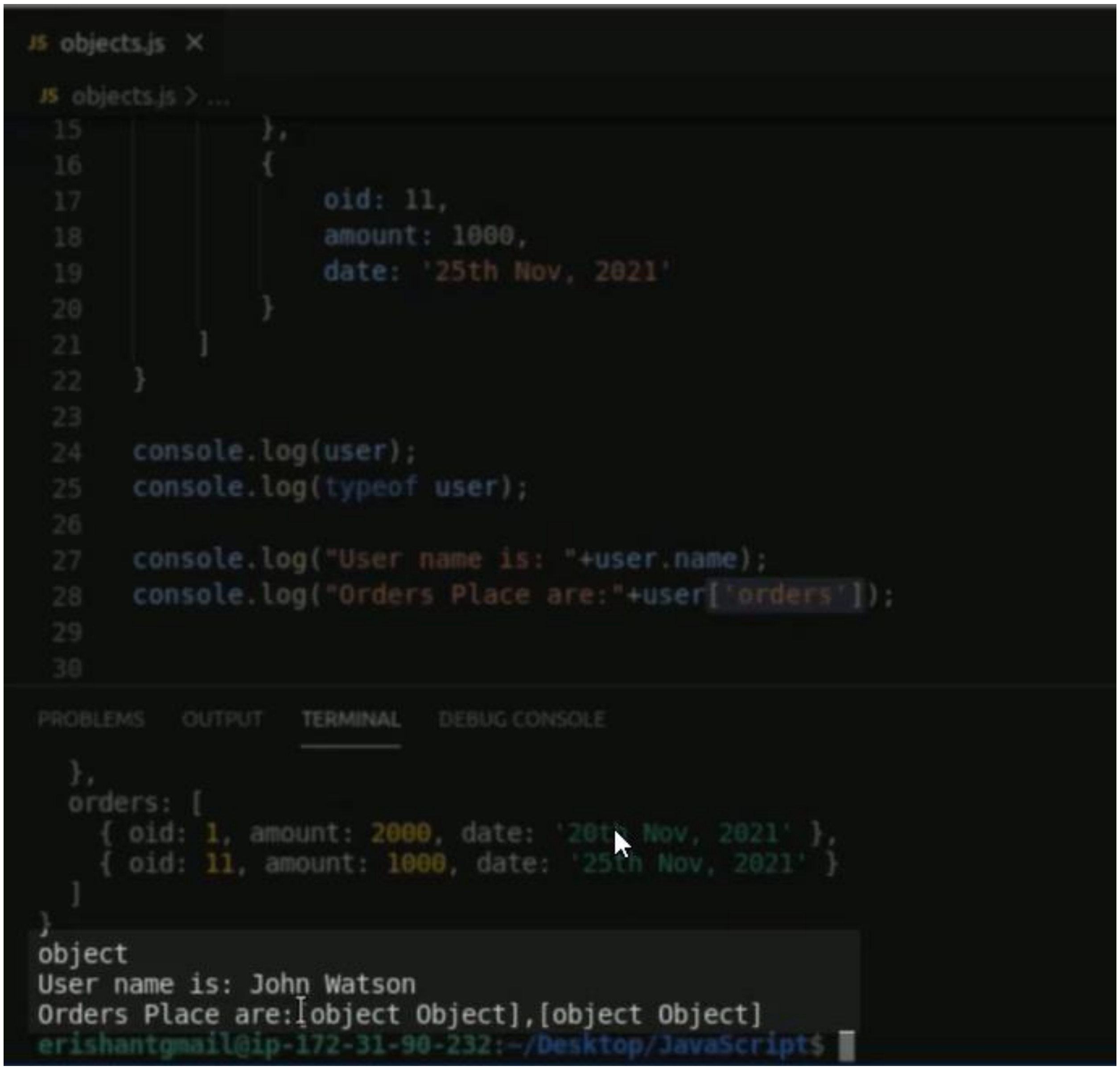
```
objects.js ×
objects.js > user > orders > date
        },
        orders:[
            {
                oid: 1,
                amount: 2000,
                date: '20th Nov, 2021'
            },
            {

                oid: 11,
                amount: 1000,
                date: '25th Nov, 2021'
            }
        ]
    }

console.log(user);
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
    city: 'Banaglaore',
    zipCode: 520001
  },
  orders: [
    { oid: 1, amount: 2000, date: '20th Nov, 2021' },
    { oid: 11, amount: 1000, date: '25th Nov, 2021' }
  ]
}
object
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$
```

Verify if the array includes the added objects in the specified order

1.9 Use dot notation or square brackets to access and print specific attributes of the user object, such as **name** and **orders**

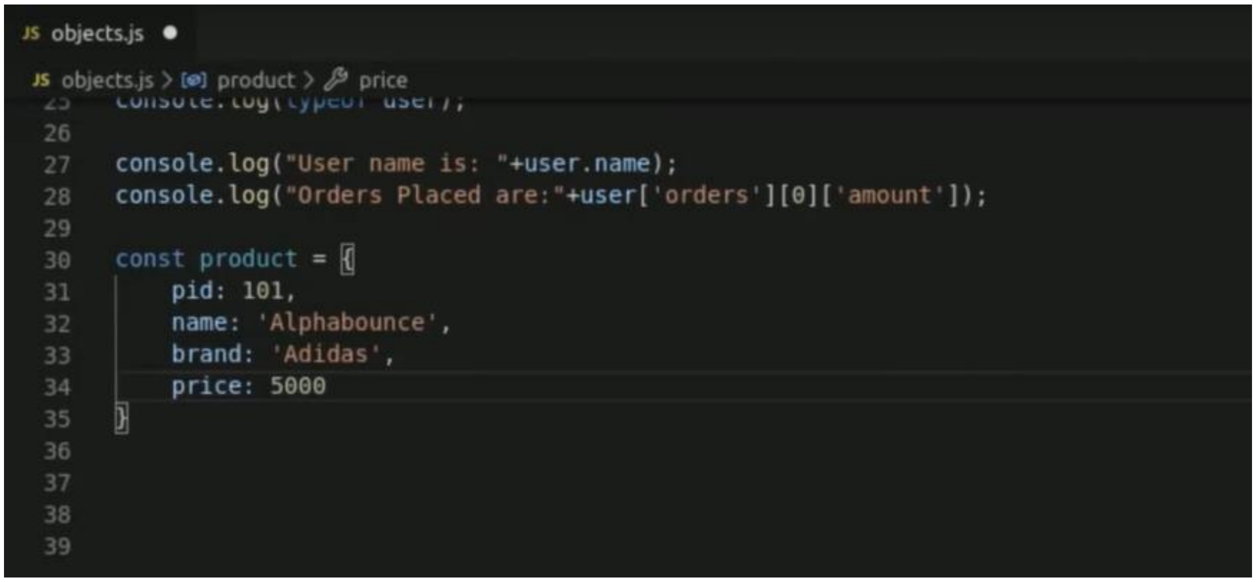


```
            },
            {
                oid: 11,
                amount: 1000,
                date: '25th Nov, 2021'
            }
        ]
}

console.log(user);
console.log(typeof user);

console.log("User name is: "+user.name);
console.log("Orders Place are:"+user['orders']);
```

```
  },
  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],[object Object]
erishantgmail@ip-172-31-90-232:~/Desktop/JavaScript$
```

Step 2: Create a product object

2.1 Declare a constant object called **product** using the **const** keyword. Assign key-value pairs to the product object, representing product details such as **pid**, **name**, **brand**, and **price**

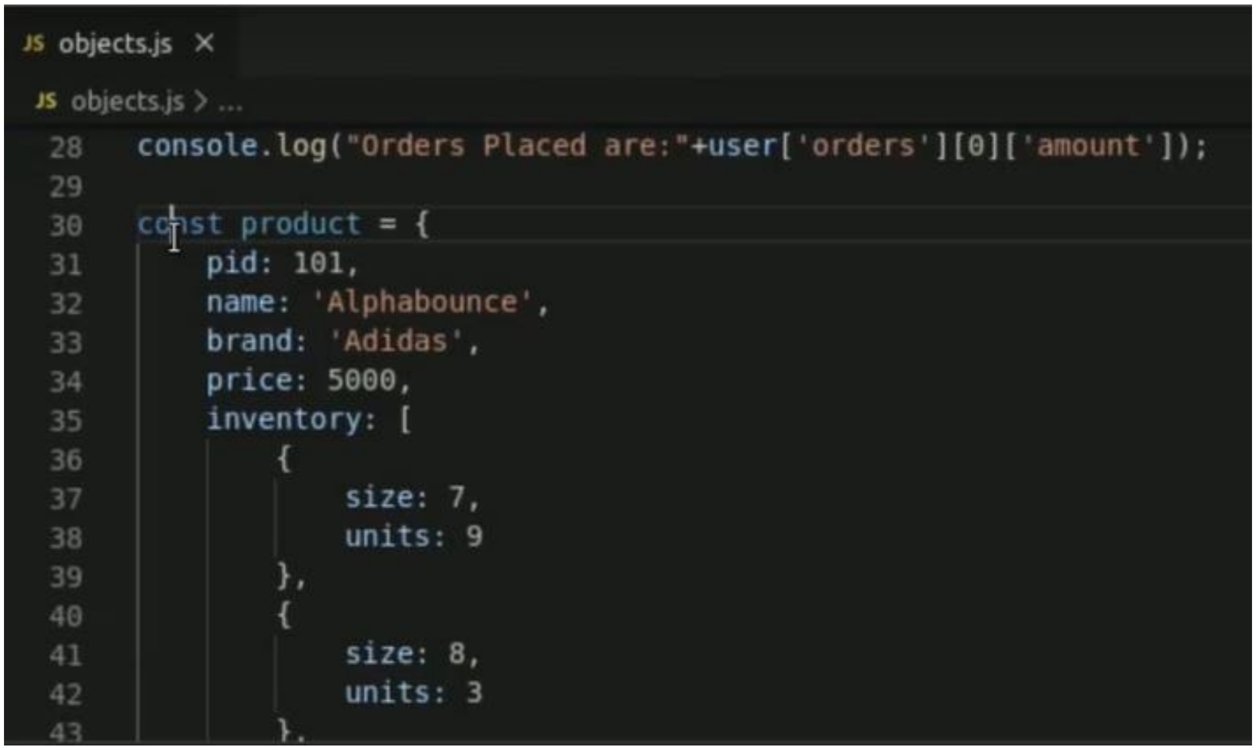


JS objects.js ●

JS objects.js > product > price

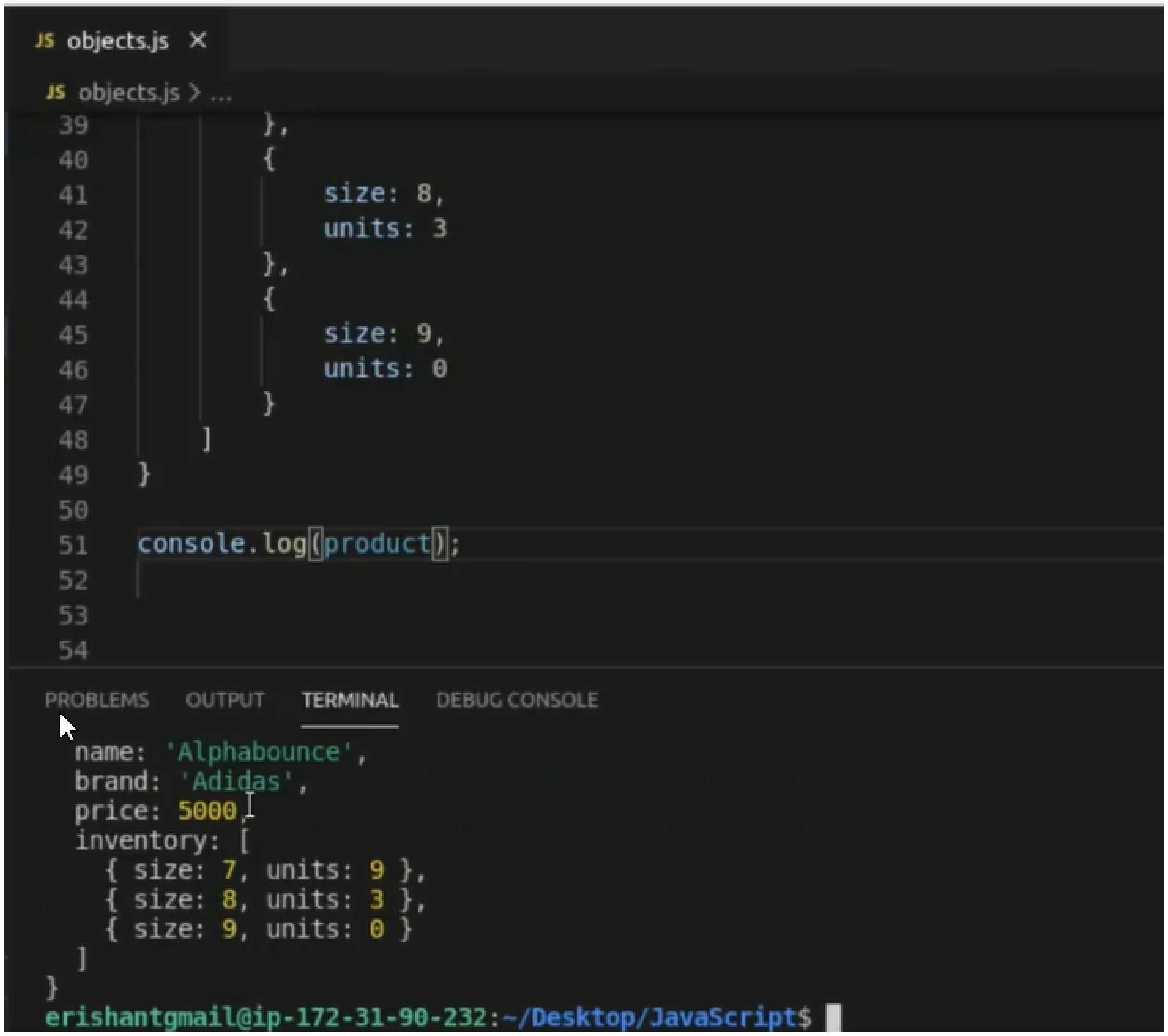
```
console.log(typeof user);

console.log("User name is: "+user.name);
console.log("Orders Placed are:"+user['orders'][0]['amount']);

const product = {
    pid: 101,
    name: 'Alphabounce',
    brand: 'Adidas',
    price: 5000
}
```

2.2 Add a key called **inventory** to the product object and add objects to the **inventory** array representing different sizes of the product and their available units



```
console.log("Orders Placed are:"+user['orders'][0]['amount']);

const product = {
    pid: 101,
    name: 'Alphabounce',
    brand: 'Adidas',
    price: 5000,
    inventory: [
        {
            size: 7,
            units: 9
        },
        {
            size: 8,
            units: 3
        },
```

2.3 Use console.log() to print the product object



objects.js ×

objects.js > ...

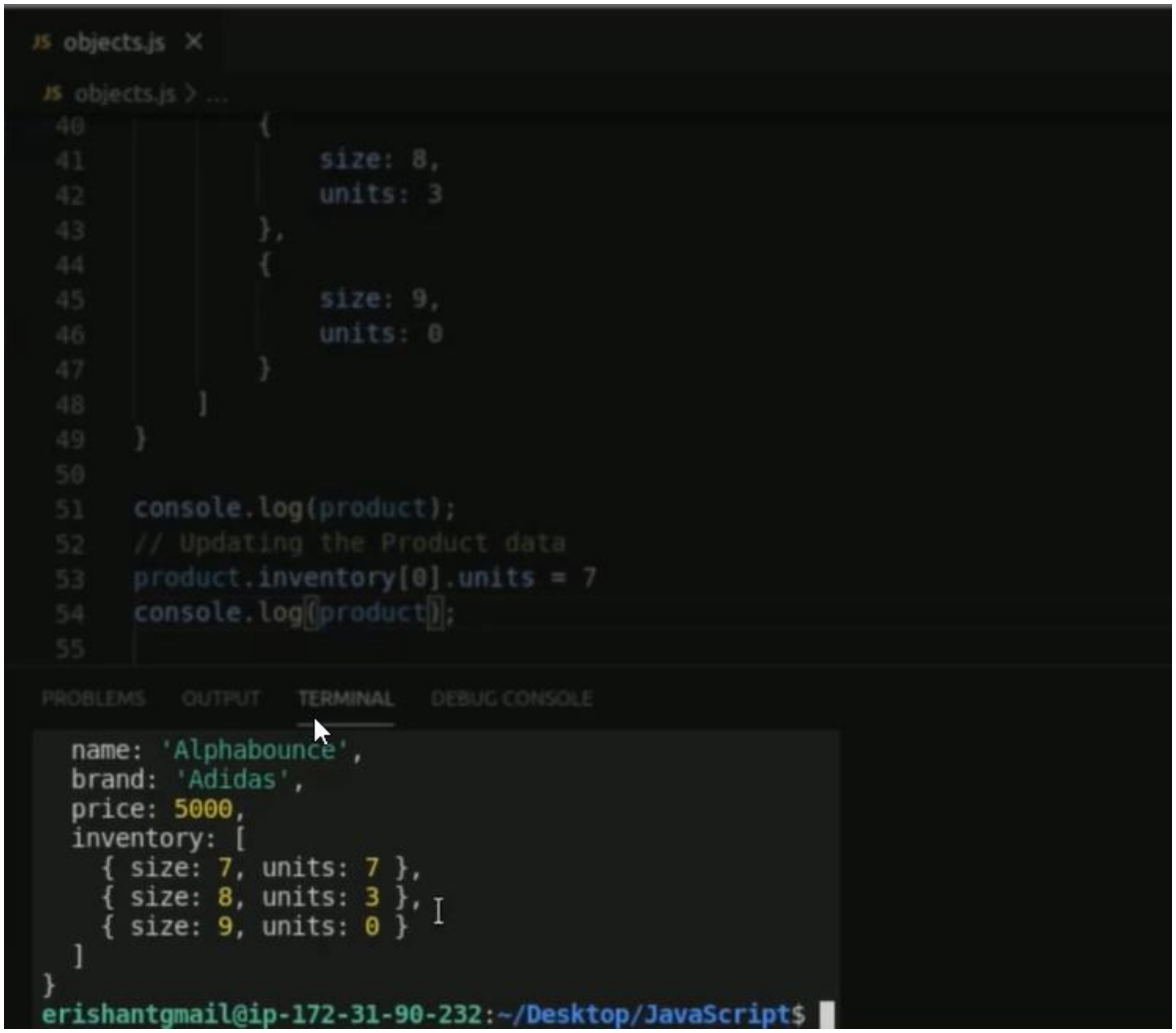
```
            },
            {
                size: 8,
                units: 3
            },
            {
                size: 9,
                units: 0
            }
        ]
}

console.log(product);



```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

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

Verify that the object contains the specified product details and inventory information

2.4 Update the value of a specific property, such as the available units for a particular size. Print the updated product object to verify the changes



```
            {
                size: 8,
                units: 3
            },
            {
                size: 9,
                units: 0
            }
        ]
}

console.log(product);
// Updating the Product data
product.inventory[0].units = 7
console.log(product);

```

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

By following these steps, you will be able to create objects, work with key-value pairs, and utilize nested objects and arrays in JavaScript.