

TASK 2 :

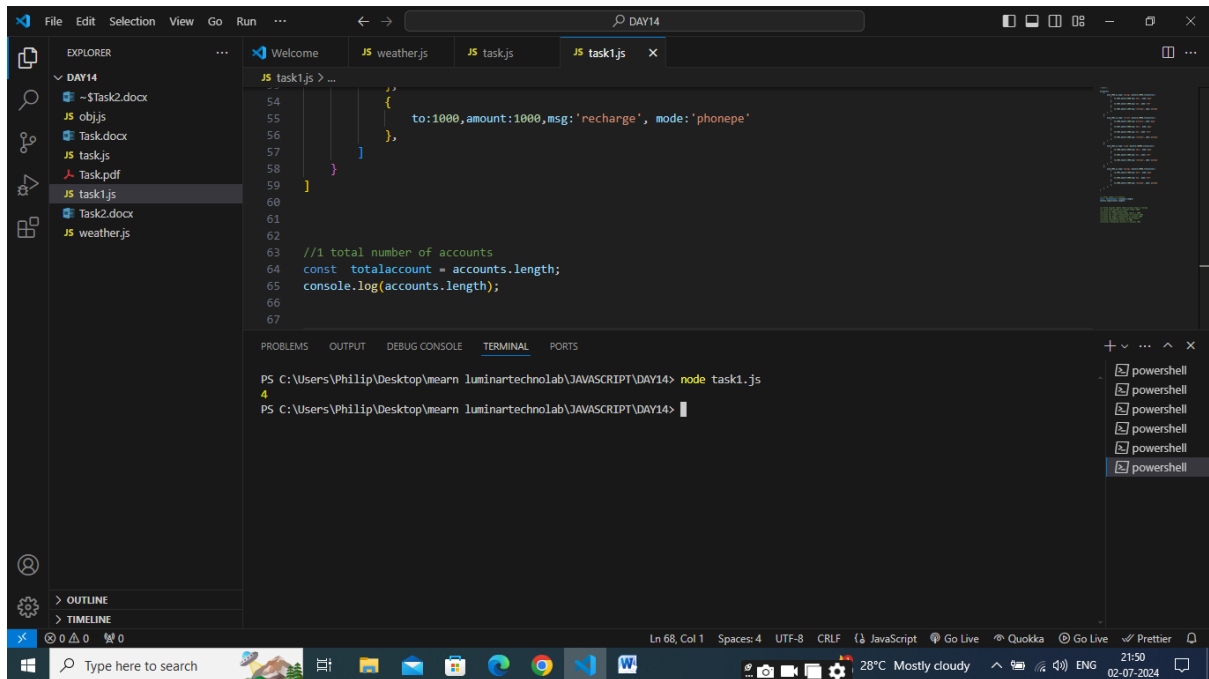
```
accounts=[
  {
    acno:1000,ac_type:'savings',balance:45000,transaction:[
      {
        to:1001,amount:6000,msg:'ebil', mode:'gpay'
      },
      {
        to:1002,amount:2000,msg:'emi', mode:'neft'
      },
      {
        to:1003,amount:1000,msg:'recharge', mode:'phonepe'
      },
    ]
  },
  {
    acno:1001,ac_type:'current',balance:30000,transaction:[
      {
        to:1000,amount:1000,msg:'grossary', mode:'gpay'
      },
      {
        to:1001,amount:5000,msg:'ebil', mode:'gpay'
      },
      {
        to:1002,amount:2000,msg:'emi', mode:'neft'
      },
      {
        to:1003,amount:1000,msg:'recharge', mode:'phonepe'
      },
    ]
  },
  {
    acno:1002,ac_type:'fixed',balance:100000,transaction:[
      {
        to:1000,amount:5000,msg:'ebil', mode:'gpay'
      },
      {
        to:1001,amount:2000,msg:'emi', mode:'neft'
      },
      {
        to:1003,amount:1000,msg:'recharge', mode:'phonepe'
      },
    ]
  },
]
```

```

{
  acno:1003,ac_type:'savings',balance:30000,transaction:[
    {
      to:1001,amount:7000,msg:'ebil', mode:'gpay'
    },
    {
      to:1002,amount:2000,msg:'emi', mode:'neft'
    },
    {
      to:1000,amount:1000,msg:'recharge', mode:'phonepe'
    },
  ],
}
]

```

//1 total number of accounts



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays a file tree for a project named 'DAY14', containing files like 'objjs', 'Task.docx', 'taskjs', 'Task.pdf', 'task1.js', 'Task2.docx', and 'weatherjs'. The main editor window is open to 'task1.js', showing a JavaScript array of account transactions and a comment '//1 total number of accounts'. Below the editor, the TERMINAL panel shows the command 'node task1.js' being executed in a PowerShell prompt, with the output '4' displayed, representing the length of the accounts array.

```

54      },
55      {
56        to:1000,amount:1000,msg:'recharge', mode:'phonepe'
57      }
58    ],
59  },
60 ],
61
62
63 //1 total number of accounts
64 const totalaccount = accounts.length;
65 console.log(accounts.length);
66
67

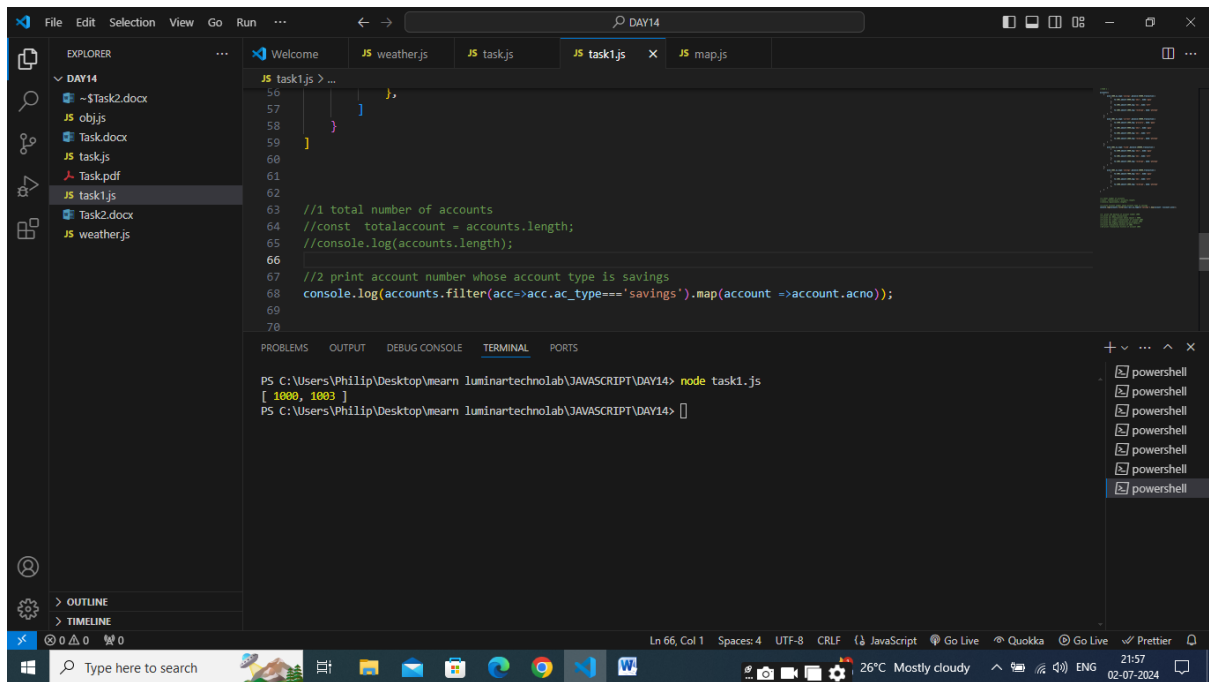
```

```

PS C:\Users\Philip\Desktop\mearn luminartechlab\JAVASCRIPT\DAY14> node task1.js
4
PS C:\Users\Philip\Desktop\mearn luminartechlab\JAVASCRIPT\DAY14>

```

//2 print account number whose account type is savings

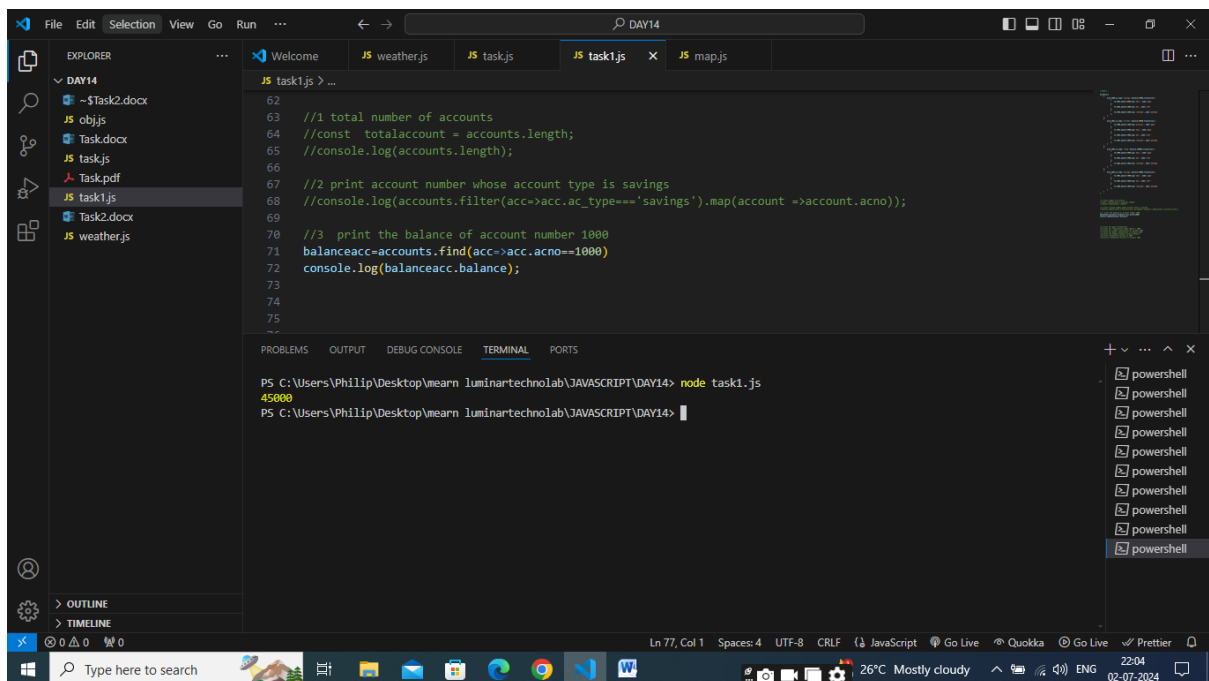


The image shows a Visual Studio Code editor window with a file named `task1.js` open. The code in the editor is as follows:

```
56  
57  
58  
59  
60  
61  
62  
63 //1 total number of accounts  
64 //const totalaccount = accounts.length;  
65 //console.log(accounts.length);  
66  
67 //2 print account number whose account type is savings  
68 console.log(accounts.filter(acc=>acc.ac_type==='savings').map(account =>account.acno));  
69  
70
```

The terminal at the bottom shows the command `node task1.js` being executed, resulting in the output `[1000, 1003]`.

//3 print the balance of account number 1000



The image shows the same Visual Studio Code editor window, but with an additional line of code added to `task1.js`:

```
62  
63 //1 total number of accounts  
64 //const totalaccount = accounts.length;  
65 //console.log(accounts.length);  
66  
67 //2 print account number whose account type is savings  
68 //console.log(accounts.filter(acc=>acc.ac_type==='savings').map(account =>account.acno));  
69  
70 //3 print the balance of account number 1000  
71 balanceacc=accounts.find(acc=>acc.acno==1000)  
72 console.log(balanceacc.balance);  
73  
74  
75
```

The terminal now shows the command `node task1.js` being executed, resulting in the output `45000`.

//4 print all gpay transactions

```
JS task1.js > ...
65 //console.log(accounts.length);
66
67 //2 print account number whose account type is savings
68 //console.log(accounts.filter(acc=>acc.ac_type==='savings').map(account =>account.acno));
69
70 //3 print the balance of account number 1000
71 //balanceacc=accounts.find(acc=>acc.acno==1000)
72 //console.log(balanceacc.balance);
73
74
75 //4 print all gpay transactions
76 const gpayTransactions = accounts.flatMap(account=>account.transaction).filter(txn=>txn.mode==='gpay');
77 console.log('Gpay Transactions:', gpayTransactions);
78
--
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

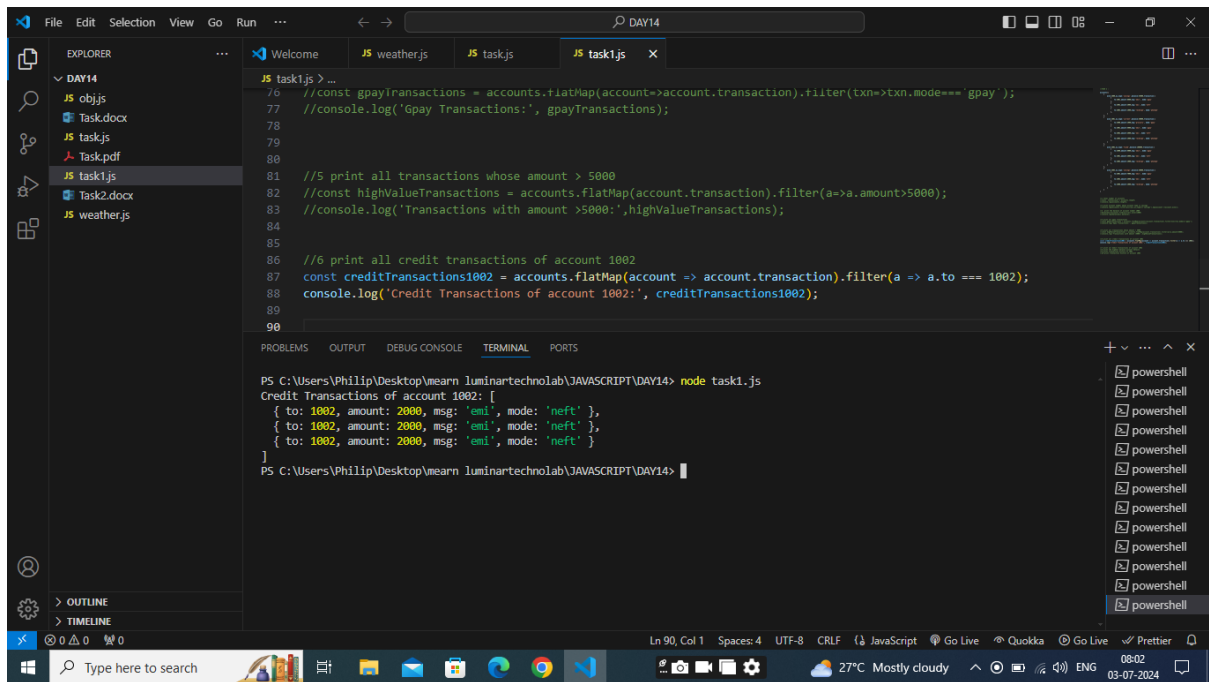
PS C:\Users\Philip\Desktop\mearn luminartechnolab\JAVASCRIPT\DAY14> node task1.js
Gpay Transactions: [
  { to: 1001, amount: 6000, msg: 'ebill', mode: 'gpay' },
  { to: 1000, amount: 1000, msg: 'grossary', mode: 'gpay' },
  { to: 1001, amount: 5000, msg: 'ebill', mode: 'gpay' },
  { to: 1000, amount: 5000, msg: 'ebill', mode: 'gpay' },
  { to: 1001, amount: 7000, msg: 'ebill', mode: 'gpay' }
]
PS C:\Users\Philip\Desktop\mearn luminartechnolab\JAVASCRIPT\DAY14>
```

//5 print all transactions whose amount > 5000

```
JS task1.js > ...
71 //balanceacc=accounts.find(acc=>acc.acno==1000)
72 //console.log(balanceacc.balance);
73
74
75 //4 print all gpay transactions
76 //const gpayTransactions = accounts.flatMap(account=>account.transaction).filter(txn=>txn.mode==='gpay');
77 //console.log('Gpay Transactions:', gpayTransactions);
78
79
80
81 //5 print all transactions whose amount > 5000
82 const highValueTransactions = accounts.flatMap(account => account.transaction).filter(a => a.amount > 5000);
83 console.log('Transactions with amount > 5000:', highValueTransactions);
84
--
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

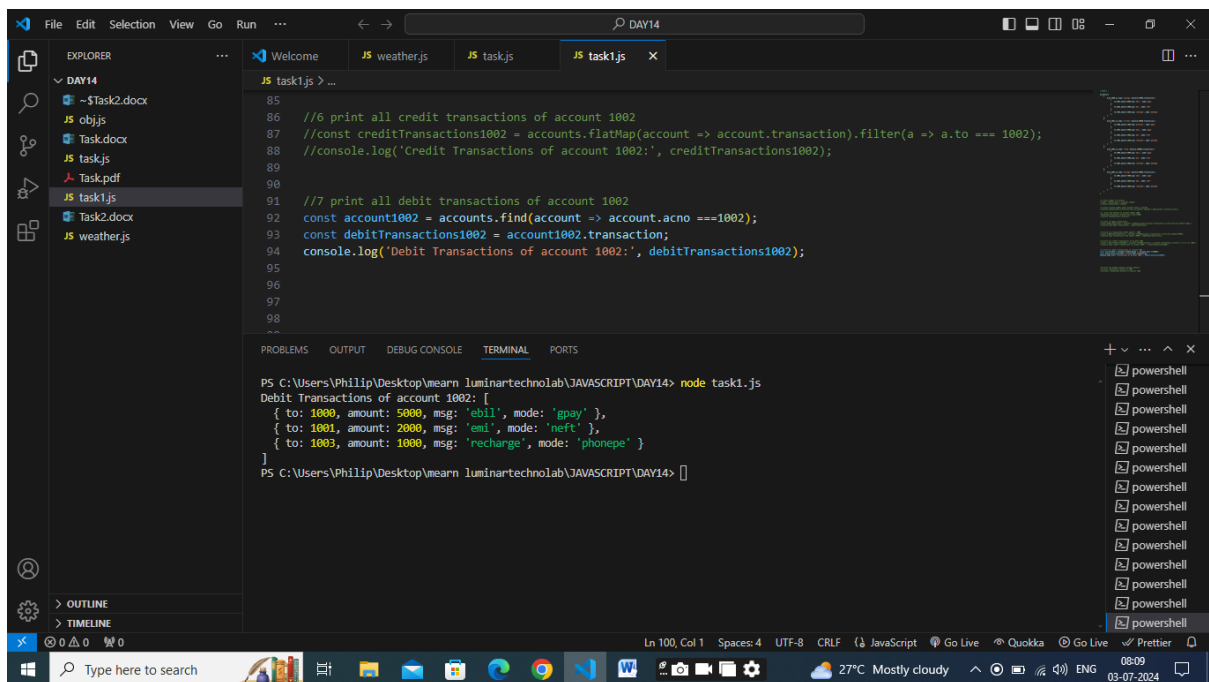
PS C:\Users\Philip\Desktop\mearn luminartechnolab\JAVASCRIPT\DAY14> node task1.js
Transactions with amount > 5000: [
  { to: 1001, amount: 6000, msg: 'ebill', mode: 'gpay' },
  { to: 1001, amount: 7000, msg: 'ebill', mode: 'gpay' }
]
PS C:\Users\Philip\Desktop\mearn luminartechnolab\JAVASCRIPT\DAY14>
```

//6 print all credit transactions of account 1002



```
File Edit Selection View Go Run ... DAY14
EXPLORER
  DAY14
    obj.js
    Task.docx
    task.js
    Task.pdf
    task1.js
    Task2.docx
    weather.js
  task1.js
task1.js
76 //const gpayTransactions = accounts.flatMap(account=>account.transaction).filter(txn=>txn.mode==='gpay');
77 //console.log('Gpay Transactions:', gpayTransactions);
78
79
80
81 //5 print all transactions whose amount > 5000
82 //const highValueTransactions = accounts.flatMap(account.transaction).filter(a=>a.amount>5000);
83 //console.log('Transactions with amount >5000:',highValueTransactions);
84
85
86 //6 print all credit transactions of account 1002
87 const creditTransactions1002 = accounts.flatMap(account => account.transaction).filter(a => a.to === 1002);
88 console.log('Credit Transactions of account 1002:', creditTransactions1002);
89
90
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Philip\Desktop\mearn luminartechnolab\JAVASCRIPT\DAY14> node task1.js
Credit Transactions of account 1002: [
  { to: 1002, amount: 2000, msg: 'emi', mode: 'neft' },
  { to: 1002, amount: 2000, msg: 'emi', mode: 'neft' },
  { to: 1002, amount: 2000, msg: 'emi', mode: 'neft' }
]
PS C:\Users\Philip\Desktop\mearn luminartechnolab\JAVASCRIPT\DAY14>
```

//7 print all debit transactions of account 1002



```
File Edit Selection View Go Run ... DAY14
EXPLORER
  DAY14
    ~$Task2.docx
    obj.js
    Task.docx
    task.js
    Task.pdf
    task1.js
    Task2.docx
    weather.js
  task1.js
task1.js
85
86 //6 print all credit transactions of account 1002
87 //const creditTransactions1002 = accounts.flatMap(account => account.transaction).filter(a => a.to === 1002);
88 //console.log('Credit Transactions of account 1002:', creditTransactions1002);
89
90
91 //7 print all debit transactions of account 1002
92 const account1002 = accounts.find(account => account.acno ===1002);
93 const debitTransactions1002 = account1002.transaction;
94 console.log('Debit Transactions of account 1002:', debitTransactions1002);
95
96
97
98
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Philip\Desktop\mearn luminartechnolab\JAVASCRIPT\DAY14> node task1.js
Debit Transactions of account 1002: [
  { to: 1000, amount: 5000, msg: 'ebil', mode: 'gpay' },
  { to: 1001, amount: 2000, msg: 'emi', mode: 'neft' },
  { to: 1003, amount: 1000, msg: 'recharge', mode: 'phonepe' }
]
PS C:\Users\Philip\Desktop\mearn luminartechnolab\JAVASCRIPT\DAY14>
```

//8 print the highest balance account details

```
86 //6 print all credit transactions of account 1002
87 //const creditTransactions1002 = accounts.flatMap(account => account.transaction).filter(a => a.to === 1002);
88 //console.log('Credit Transactions of account 1002:', creditTransactions1002);
89
90
91 //7 print all debit transactions of account 1002
92 // const account1002 = accounts.find(account => account.acno ===1002);
93 // const debitTransactions1002 = account1002.transaction;
94 // console.log('Debit Transactions of account 1002:', debitTransactions1002);
95
96
97 //8 print the highest balance account details
98 console.log(accounts.reduce((a,b)=> a.balance > b.balance? a: b ));
99
```

```
PS C:\Users\Philip\Desktop\mearn luminartechlab\JAVASCRIPT\DAY14> node task1.js
{
  acno: 1002,
  ac_type: 'fixed',
  balance: 100000,
  transaction: [
    { to: 1000, amount: 5000, msg: 'ebil', mode: 'gpay' },
    { to: 1001, amount: 2000, msg: 'emi', mode: 'neft' },
    { to: 1003, amount: 1000, msg: 'recharge', mode: 'phonepe' }
  ]
}
```

//9 print transaction history of 1002

```
100
101
102 //9 print transaction history of 1002
103
104 const account1002 = accounts.find(account => account.acno === 1002);
105 if (account1002) {
106   console.log('Transaction history of account ${account1002.acno}:');
107   account1002.transaction.forEach(transaction => {
108     console.log(` To: ${transaction.to}, Amount: ${transaction.amount}, Message: ${transaction.msg}, Mode: ${transaction.mode}`);
109   });
110 } else {
111   console.log('Account with acno 1002 not found.');
```

```
PS C:\Users\Philip\Desktop\mearn luminartechlab\JAVASCRIPT\DAY14> node task1.js
Transaction history of account 1002:
To: 1000, Amount: 5000, Message: ebil, Mode: gpay
To: 1001, Amount: 2000, Message: emi, Mode: neft
To: 1003, Amount: 1000, Message: recharge, Mode: phonepe
```

//10 print transaction history of account 1003

