

Lab Assignment 6

[Edit](#)[New Page](#)[Jump to bottom](#)

madhurisarda edited this page 7 minutes ago · 2 revisions

Lab Assignment 6

Name: Madhuri Sarda ID: 16291679 Class ID: 21 Lab ID: 6_2

The agenda of this assignment is to perform CRUD operations on the customer notes. Add, update, delete and read the customerID, customerName, customerEmail.

```
1
2  const fs = require('fs');
3  const yargs = require('yargs');
4
5  const notes = require('./notes.js');
6
7  // ----- Begin - command configuration -----
8  const customerID = {
9    describe: 'ID of Customer',
10    demand : true,
11    alias : 'id'
12  }
13
14  const customerName = {
15    describe: 'Name of Customer',
16    demand : true,
17    alias : 'name'
18  }
19
20  const customerEmail = {
21    describe: 'Email ID of Customer',
22    demand : true,
23    alias : 'email'
24  }
25
26  const argv = yargs
27
28  .command('add', 'Add a new note', {
29    ID: customerID,
30    Name: customerName,
```

```
31 // ----- End - command configuration -----
32 })
33 .command('update','Update an existing note',{
34     ID: customerID,
35     Name: customerName,
36     Email: customerEmail
37 })
38 .command( cmd: 'list', description: 'List all notes')
39 .command( cmd: 'read', description: 'Read a note', builder: {
40     ID: customerID,
41     Name: customerName,
42     Email: customerEmail
43 })
44 .command( cmd: 'remove', description: 'Remove a Note', builder: {
45     ID: customerID
46 })
47 .help()
48 .argv;
49
50
51 // ----- End - command configuration -----
52
53
54 var command = yargs.argv._[0];
55
56
57 if (command === 'add'){
58     var note = notes.addNote(argv.ID,argv.Name, argv.Email);
59     if (note){
60         notes.logNote(note); //add a new note
61     } else{
```

```
66  else if (command === 'update'){
67      var note = notes.updateNotes(argv.ID,argv.Name, argv.Email);
68      if (note){
69          notes.logNote(note);                                //add a new note
70      } else{
71          console.log("Note was updated");
72      }
73  }
74
75  else if (command === 'list') {
76      var AllNotes = notes.getAll();
77      console.log(`Printing ${AllNotes.length} note(s).`);
78      AllNotes.forEach((note)=>{                            //list all note(s)
79          notes.logNote(note);
80      });
81  }
82
83  else if (command === 'read') {
84      var note = notes.getNote(argv.ID);
85      if(note){
86          notes.logNote(note);                                //read a note
87      }
88      else{
89          console.log("Note not found");
90      }
91  }
92  else if (command === 'remove') {
93      var noteRemoved = notes.remove(argv.ID);
94      var message = noteRemoved ? 'Note was removed' : 'Note not found';
95      console.log(message);
```

```
1  const fs = require('fs');
2
3
4  // -----Begin of Reusable functions -----
5
6  var fetchNotes = () => {
7    try {                                //if file won't exist
8      var notesString = fs.readFileSync('notes-data.json');
9      return JSON.parse(notesString);
10   } catch(e){
11     return [];
12   }
13 };
14
15 var saveNotes = (notes) => {
16   fs.writeFileSync('notes-data.json',JSON.stringify(notes));
17 };
18
19
20 // -----End of Reusable functions -----
21
22 var updateNotes = (ID,Name,Email) => {
23   remove(ID); // update the value for existing notes
24   var customerNote = {ID,Name,Email};
25   var filteredNotes = fetchNotes();
26   filteredNotes.push(customerNote);
27   saveNotes(filteredNotes);
28   return customerNote
29 }
30 // to add a new note
```

```
30 // to add a new note
31
32 var addNote = (ID,Name,Email) => {
33   var notes = fetchNotes();
34   var customerNote = {ID,Name,Email}
35   var duplicateNotes = notes.filter((note) => { // to check if note already exists
36     if(note.ID === ID) {
37       updateNotes(ID,Name,Email);
38     }
39     return note.ID === ID;
40   });
41   if (duplicateNotes.length === 0){
42     notes.push(customerNote);
43     saveNotes(notes);
44     return customerNote
45   }
46
47 };
48
49 //to list all the notes
50
51 var getAll = () => {
52   return fetchNotes();
53 };
54
55
56 // to read a note
57
58 var getNote = (ID) => {
```

```
59 var getNote = (ID) => {
60   var notes = fetchNotes();
61
62   var getNotes = notes.filter((note) => { // to check if note exists and return note
63     return note.ID === ID;
64   });
65
66   return getNotes[0]
67 };
68
69
70 // to delete a note
71
72 var remove = (ID) => {
73   var notes = fetchNotes(); // reusable func
74
75   var filteredNotes = notes.filter((note) => { // will return all other notes other than "note to be removed"
76     return note.ID !== ID;
77   });
78
79   saveNotes(filteredNotes); //save new notes array
80
81   return notes.length !== filteredNotes.length
82 };
83
84 // function just to print out note to screen
```

```

84     return notes.length !== filteredNotes.length
85
86   });
87
88   // function just to print out note to screen
89
90   var logNote = (note) => {
91     console.log('--');
92     console.log(`Customer ID: ${note.ID}`);
93     console.log(`Customer Name: ${note.Name}`);
94     console.log(`Customer Email: ${note.Email}`);
95   });
96
97   // add new function names here to be accessible from other modules
98
99   module.exports = {
100     addNote, getAll, remove, getNote, logNote, updateNotes
101   };
102

```

```

package.json × notes-data.json × app.js × notes.js ×
1 [{"ID":1,"Name":"madhuri","Email":"ms6dt@mail.umkc.edu"}, {"ID":2,"Name":"udaya","Email":"udaya@mail.umkc.edu"}]

```

Output screens:

```

Terminal: Local × +
C:\Users\madhuri's PC\Documents\GitHub\ASE\Lab6_YargsModule\Lab6_YargsModule>node app.js add --id "1" command --name "madhuri" command --email "ms6dt@mail.umkc.edu"
--
Customer ID: 1
Customer Name: madhuri
Customer Email: ms6dt@mail.umkc.edu
C:\Users\madhuri's PC\Documents\GitHub\ASE\Lab6_YargsModule\Lab6_YargsModule>

Terminal: Local × +
C:\Users\madhuri's PC\Documents\GitHub\ASE\Lab6_YargsModule\Lab6_YargsModule>node app.js list
Printing 2 note(s).
--
Customer ID: 1
Customer Name: madhuri
Customer Email: ms6dt@mail.umkc.edu
--
Customer ID: 2
Customer Name: madhuri1
Customer Email: madhursarda@mail.umkc.edu
C:\Users\madhuri's PC\Documents\GitHub\ASE\Lab6_YargsModule\Lab6_YargsModule>

Terminal: Local × +
C:\Users\madhuri's PC\Documents\GitHub\ASE\Lab6_YargsModule\Lab6_YargsModule>node app.js read --id "1"
--
Customer ID: 1
Customer Name: madhuri
Customer Email: ms6dt@mail.umkc.edu
C:\Users\madhuri's PC\Documents\GitHub\ASE\Lab6_YargsModule\Lab6_YargsModule>

Terminal: Local × +
C:\Users\madhuri's PC\Documents\GitHub\ASE\Lab6_YargsModule\Lab6_YargsModule>node app.js remove --id "1"
Note was removed
C:\Users\madhuri's PC\Documents\GitHub\ASE\Lab6_YargsModule\Lab6_YargsModule>node app.js list
Printing 1 note(s).
--
Customer ID: 2
Customer Name: madhuri1
Customer Email: madhursarda@mail.umkc.edu
C:\Users\madhuri's PC\Documents\GitHub\ASE\Lab6_YargsModule\Lab6_YargsModule>

```

The link to the document with the code snippet is

<https://github.com/madhursarda/ASE/wiki/Lab-Assignment-6>

The link to the document with the code snippet is

https://github.com/madhurisarda/ASE/blob/master/Lab6_YargsModule/SourceCode_Lab6_CR_UD_Operations_using_Yargs_Module.docx

+ Add a custom footer

▼ Pages 5

[Home](#)

[Lab Assignment 1](#)

[Lab Assignment 2](#)

[Lab Assignment 3](#)

[Lab Assignment 6](#)

+ Add a custom sidebar

Clone this wiki locally

