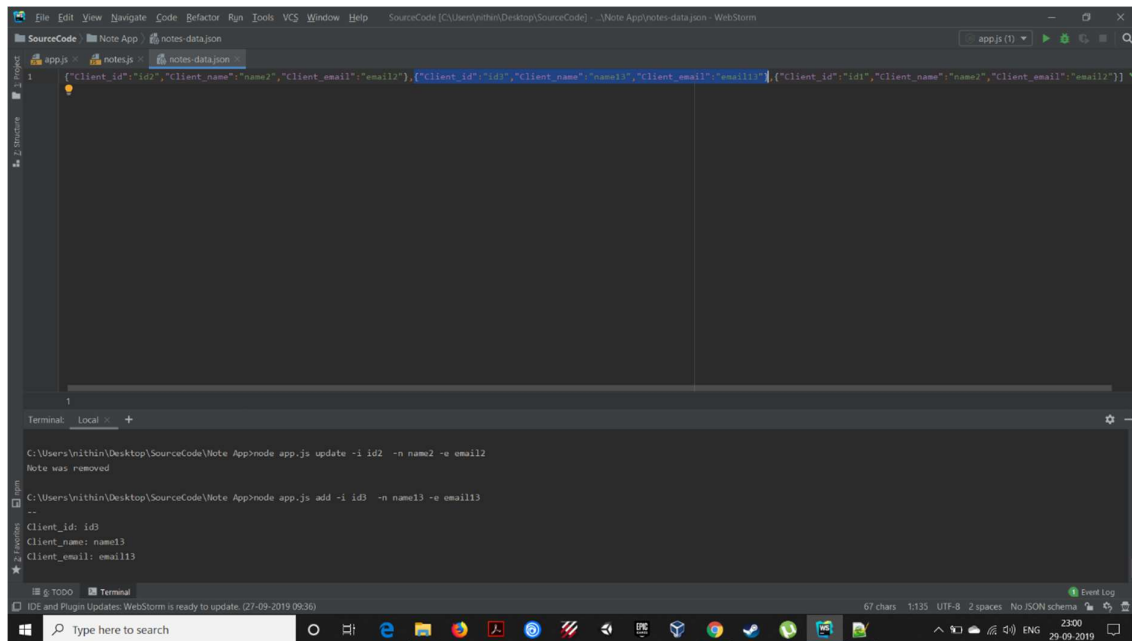


ASE

LAB 6

ADDING RECORD



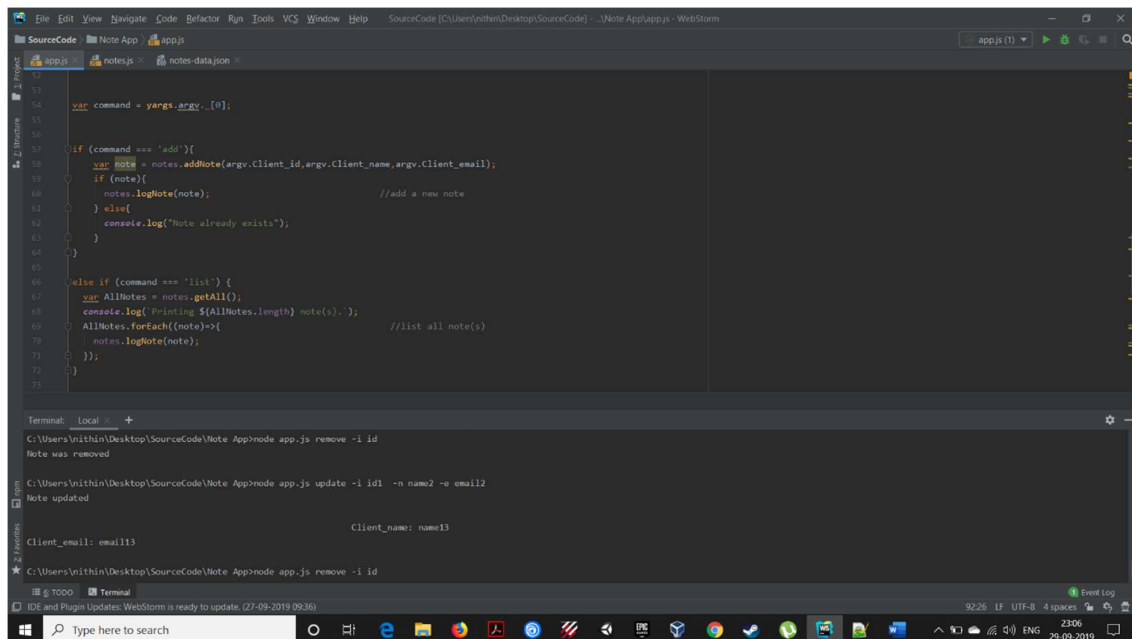
The screenshot shows the VS Code editor with a file named `notes-data.json` open. The JSON array contains three objects, with the last one highlighted in blue. The terminal below shows the command `app.js update -i id2 -n name2 -e email2` and its output, followed by `app.js add -i id3 -n name3 -e email3` and its output, which matches the new record in the JSON file.

```
[{"Client_id":"id2","Client_name":"name2","Client_email":"email2"}, {"Client_id":"id3","Client_name":"name3","Client_email":"email3"}, {"Client_id":"id1","Client_name":"name1","Client_email":"email1"}]
```

```
C:\Users\nithin\Desktop\SourceCode\Note App>node app.js update -i id2 -n name2 -e email2
Note was removed

C:\Users\nithin\Desktop\SourceCode\Note App>node app.js add -i id3 -n name3 -e email3
--
Client_id: id3
Client_name: name3
Client_email: email3
```

In the above picture I have added a record to the JSON file, I have also highlighted the record which got added to the JSON file.



The screenshot shows the `app.js` file in the VS Code editor. The code includes logic for adding a new note and listing all notes. The terminal shows the command `app.js remove -i id` and its output, followed by `app.js update -i id1 -n name2 -e email2` and its output, which matches the new record in the JSON file.

```
var command = yargs.argv._[0];

if (command === 'add') {
  var note = notes.addNote(argv.Client_id, argv.Client_name, argv.Client_email);
  if (note) {
    notes.logNote(note);
  } else {
    console.log("Note already exists");
  }
}

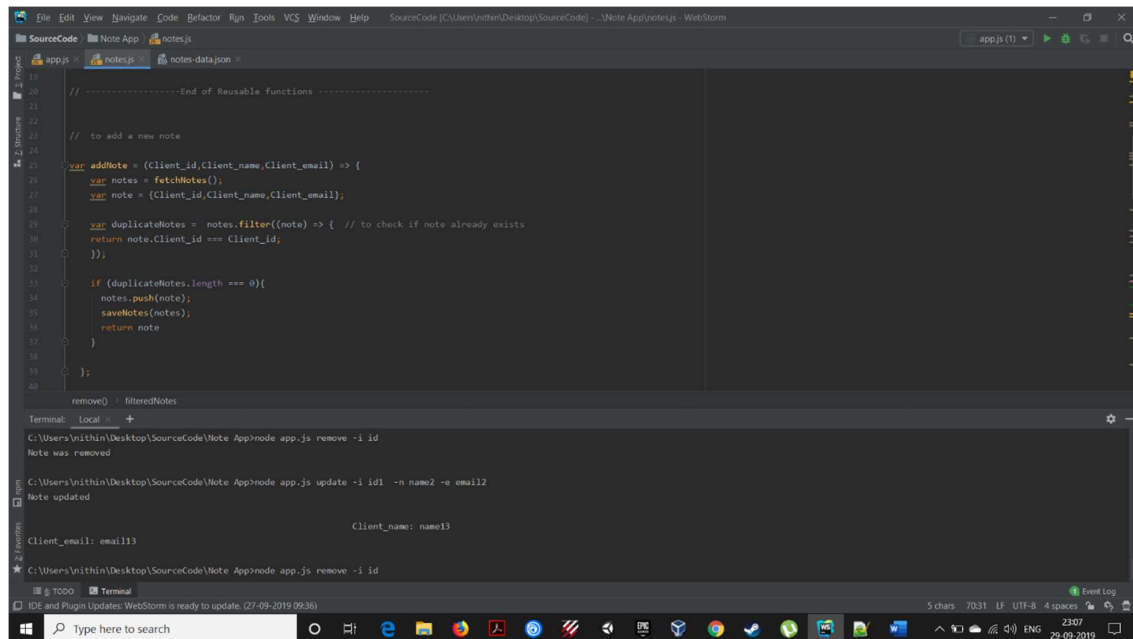
else if (command === 'list') {
  var AllNotes = notes.getAll();
  console.log("Printing " + AllNotes.length + " note(s).");
  AllNotes.forEach((note) => {
    notes.logNote(note);
  });
}
```

```
C:\Users\nithin\Desktop\SourceCode\Note App>node app.js remove -i id
Note was removed

C:\Users\nithin\Desktop\SourceCode\Note App>node app.js update -i id1 -n name2 -e email2
Note updated

Client_name: name3
Client_email: email3
```

The above is the code for trigger the add command, and this function also calls the required functions.

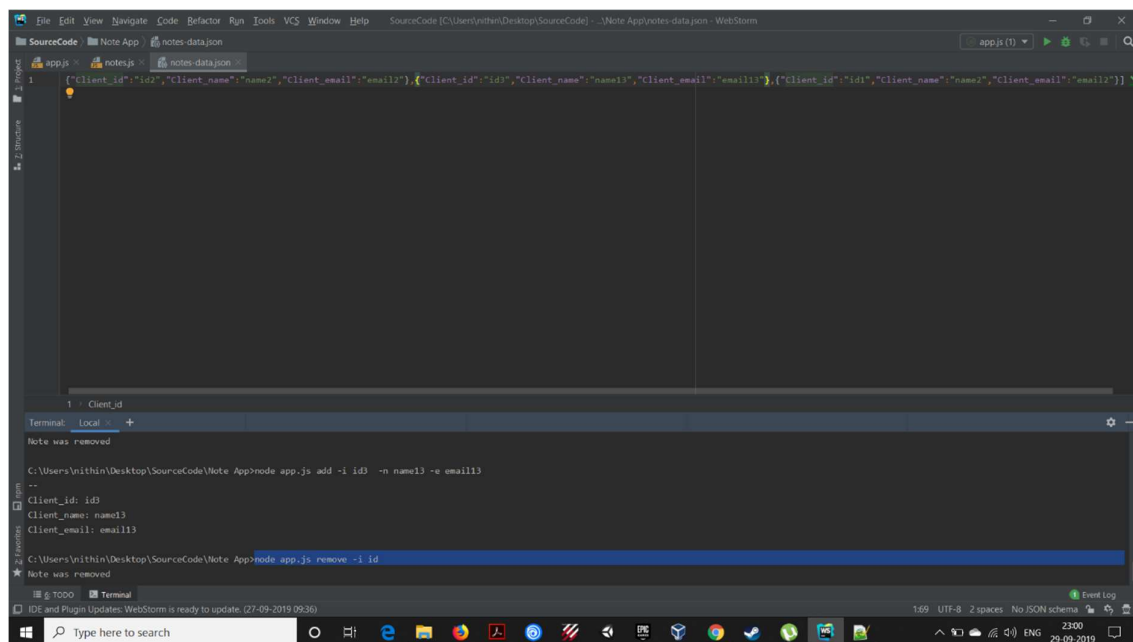


```
// -----End of Reusable functions -----  
  
// to add a new note  
  
var addNote = (Client_id,Client_name,Client_email) => {  
  var notes = fetchNotes();  
  var note = {Client_id,Client_name,Client_email};  
  
  var duplicateNotes = notes.filter(note => { // to check if note already exists  
    return note.Client_id === Client_id;  
  });  
  
  if (duplicateNotes.length === 0){  
    notes.push(note);  
    saveNotes(notes);  
    return note  
  }  
};  
  
remove() > filteredNotes
```

```
Terminal: Local +  
C:\Users\nitin\Desktop\SourceCode\Note App\node app.js remove -i id  
Note was removed  
  
C:\Users\nitin\Desktop\SourceCode\Note App\node app.js update -i id1 -n name2 -e email2  
Note updated  
  
Client_name: name13  
Client_email: email13  
  
C:\Users\nitin\Desktop\SourceCode\Note App\node app.js remove -i id
```

It is actual code for adding a new record to the JSON file.

Deleting a Record



```
[{"Client_id":"id2","Client_name":"name2","Client_email":"email2"}, {"Client_id":"id3","Client_name":"name13","Client_email":"email13"}, {"Client_id":"id1","Client_name":"name2","Client_email":"email2"}]
```

```
Terminal: Local +  
Note was removed  
  
C:\Users\nitin\Desktop\SourceCode\Note App\node app.js add -i id3 -n name13 -e email13  
--  
Client_id: id3  
Client_name: name13  
Client_email: email13  
  
C:\Users\nitin\Desktop\SourceCode\Note App\node app.js remove -i id  
Note was removed
```

remove -i id command is used in order to remove the record with employee id as id.

```

else if (command === 'remove') {
  var noteRemoved = notes.remove(argv.Client_id);
  var message = noteRemoved ? 'Note was removed' : 'Note not found';
  console.log(message);
}

```

This code is used to trigger the word remove and call the required functions.

```

64 var remove = (Client_id) => {
65   var notes = fetchNotes(); // reusable func
66
67   var filteredNotes = notes.filter((note) => { // will return all other notes other than "note to be removed"
68     return note.Client_id !== Client_id;
69   });
70   saveNotes(filteredNotes); //save new notes array
71   return notes.length !== filteredNotes.length
72 };

```

This code is used in order to remove the record from the JSON file.

Updating a Record

```

C:\Users\nithin\Desktop\SourceCode\Note App>node app.js update -i id1 -n name2 -e email2
Note updated
Client_name: name13
Client_email: email13

```

The above command is used in order to add a record to the JSON file.

```

80 else if (command === 'update') {
81   var noteRemove = notes.remove(argv.Client_id);
82   var message = noteRemove ? 'Note updated' : 'Note not found';
83   console.log(message);
84   var note = notes.addNote(argv.Client_id, argv.Client_name, argv.Client_email);
85 }

```

I have used the add and remove function in order to update my record in JSON file.

Contribution: we both worked on it individually and we got the work done.

Team 3_2

Dukkipati, Sri Sai Nithin Chowdary (class id 05)

Skoglund,Rod (Class id 23)