

Node.js Section 3

Merhaba bugün Node.js serimize devam ediyoruz ve 3.bölümdeyiz. Geçen haftaki Section2 folderini Section2-3 olarak değiştirip ve app.js yi de index.js olarak değiştirip oradan devam edeceğiz.

notes.js adında bir file oluşturalım ve title ile body yi alıp kaydetme işlemlerini burada yapalım.

Şimdi notları eklemek için addNote, notlardan silmek için ise removeNote adında functionlar oluşturalım.

Section2-3 - notes.js

```
function addNote(title, body) {}  
function removeNote(title) {}
```

Birden fazla fonksiyonları export etmek istiyorsak bunları bir nesne aracılığıyla export edebiliriz.

```
function addNote(title, body) {}  
function removeNote(title) {}  
  
module.exports = {  
    addNote: addNote, // ! sağ kısmında yazan bu dosyadaki ile eşleşmeli  
    removeNote: removeNote, // ! sol kısmında yazan ise başka dosyalarda kullanılmak üzere isimlendirilir.  
};
```

Bu notes.js file ini index.js içinde kullanmak istiyoruz. Önce add command inden başlayacağız. Console yazdırırmak yerine notes.js deki add note functionu nu kullanalım.

Notes.js i index.js e eklemek için :

```
const notes = require("./notes.js")
```

Bu kod parçacığını ekleyelim. Ve add commandindeki handler functionunu şu şekilde güncelleyelim.

```
Section2-3 - index.js
```

```
handler: function (argv) {
    notes.addNote(argv.title, argv.body)
},
```

```
Section2-3 - notes.js
```

```
function addNote(title, body) {
    console.log(title, body, "notes.js file içerisindeyiz")
}
```

Şimdi bir deneme yapalım console.log ile aynı deneme şekli ama notes.js deki kod çalışacaktır burada.

The screenshot shows the VS Code interface with the following details:

- EXPLORER**: Shows a project structure with **SECTION2-3** containing **node_modules**, **playground**, **notes.js**, **index.js**, **package-lock.json**, and **package.json**.
- NOTES**: The **notes.js** file is open in the editor, displaying code for adding and removing notes.
- TERMINAL**: Shows the command `node index.js add --title="Burak" --body="Gül"` being run and its output: "Burak Güls notes.js file içerisindeyiz".
- STATUS BAR**: Shows the current file is **notes.js**, the line is **Ln 10, Col 1**, spaces are **Spaces: 4**, encoding is **UTF-8**, and the file type is **JavaScript**.

Şimdi dosyaya veri ekleme şeklimiz şu şekilde ilerleyecektir.

- 1- Verileri JSON şeklinde tutacağız.
- 2- Dosyaya veri eklemeden önce JSON dosyasından verileri çekip kullanabileceğimiz formata dönüştürmeliyiz ki (parse etmek) nasıl ekleyebileceğimizi özelleştirelim.
- 3- Veri eklemeden önce dönüştürüduğumuz verileri kontrol ederek ona göre ekleme yapabiliriz. Mesela aynı title a sahip olmayan notlar eklemeye çalışacağımız bu örnekte. Title propertymiz uniq olacaktır yani.
- 4- Daha sonra verileri kaydedeceğiz.

Verili çekme işlemini sürekli yapacağımızdan dolayı `loadNotes` adında bir function oluşturalım ve dosyadan veri çekeceğimiz için `fileSystem` i ekleyelim file imzına.

The code editor shows the following code:

```
Section2-3 - notes.js
const fs = require('fs');
```

Daha sonra `loadNotes` içi de şöyle olsun

Section2-3 - notes.js

```
function loadNotes() {  
    // read from file  
    const dataBuffer = fs.readFileSync("notes.json");  
    // Convert to JSON format  
    const dataJSON = dataBuffer.toString();  
    // Parse the string and return  
    return JSON.parse(dataJSON);  
}
```

Bu fonksiyon ne yapar peki ?

- 1-notes.json fileinden verileri alır.
- 2- aldığı veriler JSON şeklindedir. Bu veriyi stringler, string hale dönüştürür.
- 3-daha sonra bu stringe dönümüş veriyi kullanabileceğimiz şekilde parse eder.

Şimdi bu function'u add note içinde kullanalım ve hata aldığımızı görelim.

Section2-3 - notes.js

```
function addNote(title, body) {  
    console.log(title, body, "notes.js file içerisindeyiz");  
    const notes = loadNotes();  
}
```

Çalıştırmak için
node index.js add --title="Burak" --body="Gül" i çalıştıralım.

The screenshot shows a VS Code interface with the following details:

- File Explorer:** Shows two files: `notes.js` and `index.js`. `notes.js` has a status bar indicating `U` (untracked).
- Terminal:** The terminal tab is active, displaying the following error message:

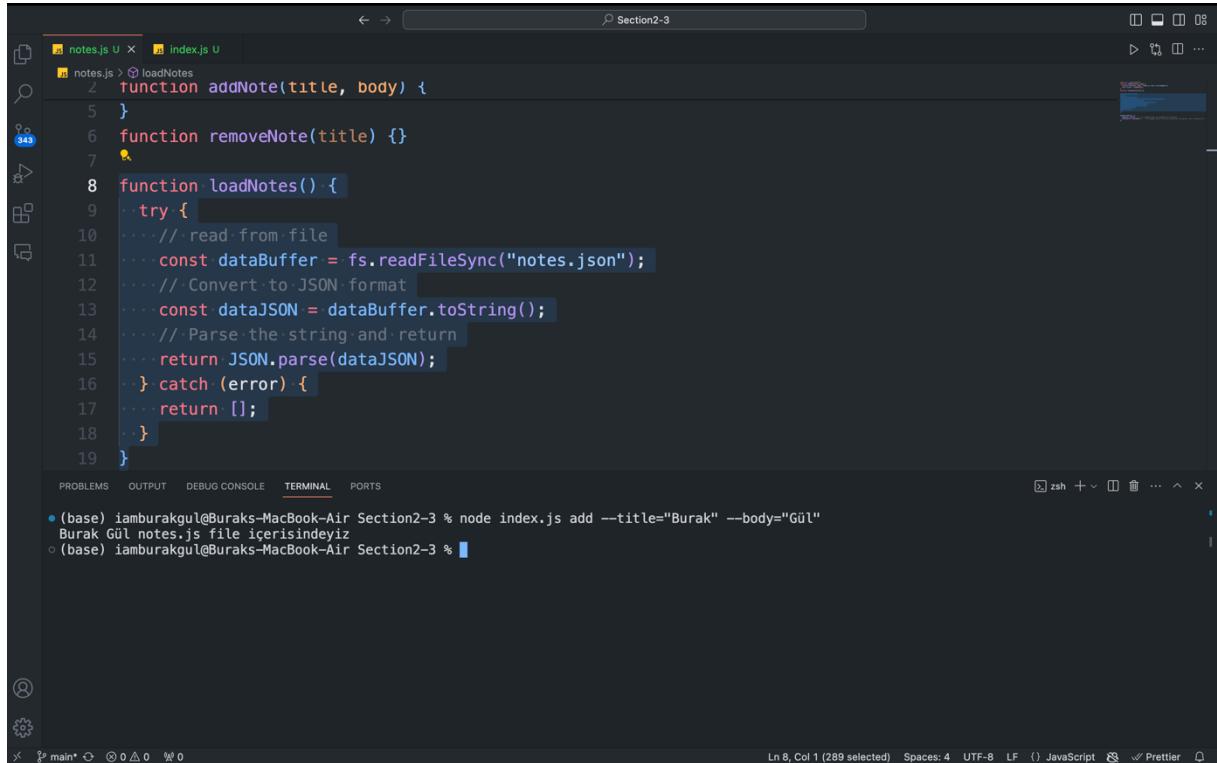
```
Error: ENOENT: no such file or directory, open 'notes.json'
at Object.openSync (node:fs:581:18)
at Object.readFileSync (node:fs:457:35)
at loadNotes (/Users/iamburakgul/Desktop/PROGRAMMING/My-NodeJS-Notes/Section2-3/notes.js:10:25)
at Object.addNote (/Users/iamburakgul/Desktop/PROGRAMMING/My-NodeJS-Notes/Section2-3/notes.js:4:17)
at Object.handler (/Users/iamburakgul/Desktop/PROGRAMMING/My-NodeJS-Notes/Section2-3/index.js:21:11)
at /Users/iamburakgul/Desktop/PROGRAMMING/My-NodeJS-Notes/Section2-3/node_modules/yargs/build/index.cjs:1:8993
at j (/Users/iamburakgul/Desktop/PROGRAMMING/My-NodeJS-Notes/Section2-3/node_modules/yargs/build/index.cjs:1:4
956)
  at _handleValidationAndGetResult (/Users/iamburakgul/Desktop/PROGRAMMING/My-NodeJS-Notes/Section2-3/node_modu
les/yargs/build/index.cjs:1:8962)
  at _applyMiddlewareAndGetResult (/Users/iamburakgul/Desktop/PROGRAMMING/My-NodeJS-Notes/Section2-3/node_modu
les/yargs/build/index.cjs:1:9604)
  at _runCommand (/Users/iamburakgul/Desktop/PROGRAMMING/My-NodeJS-Notes/Section2-3/node_modules/yargs/build/in
dex.cjs:1:7231) {
  errno: -2,
  code: 'ENOENT',
  syscall: 'open',
}
```
- Status Bar:** Shows "Ln 15, Col 2" and "JavaScript".

Gördüğümüz gibi dosyayı bulamadı bu fonksiyonda hata olabilir o yüzden fonksiyonumuzu refactor edelim.

Section2-3 - notes.js

```
function loadNotes() {  
  try {  
    // read from file  
    const dataBuffer = fs.readFileSync("notes.json");  
    // Convert to JSON format  
    const dataJSON = dataBuffer.toString();  
    // Parse the string and return  
    return JSON.parse(dataJSON);  
  } catch (error) {  
    return [];  
  }  
}
```

Şimdi yeniden çalıştırınca ise sorun yok gördüğünüz gibi.Dosya yok ise bize boş bir dizi döndürmektedir ayrıca.



```
notes.js
1 // Importing the loadNotes function from notes.js
2 const loadNotes = require('./notes');
3
4 function addNote(title, body) {
5   // Adding a new note to the notes array
6   notes.push({
7     title: title,
8     body: body
9   });
10
11   // Logging the updated notes array to the console
12   console.log(notes);
13 }
14
15 function removeNote(title) {
16   // Removing a note from the notes array by title
17   const noteIndex = notes.findIndex(note => note.title === title);
18   if (noteIndex !== -1) {
19     notes.splice(noteIndex, 1);
20   }
21 }
22
23 function loadNotes() {
24   try {
25     // Reading the notes.json file
26     const dataBuffer = fs.readFileSync('notes.json');
27     // Converting the buffer to a string
28     const dataJSON = dataBuffer.toString();
29     // Parsing the JSON string into an array
30     return JSON.parse(dataJSON);
31   } catch (error) {
32     // If there's an error, return an empty array
33     return [];
34   }
35 }
36
37 module.exports = {
38   addNote,
39   removeNote,
40   loadNotes
41 };

index.js
1 // Importing the notes module
2 const notes = require('./notes');
3
4 notes.addNote('Burak', 'Gül');
5
6 notes.loadNotes();
```

(base) iamburakgul@Buraks-MacBook-Air Section2-3 % node index.js add --title="Burak" --body="Gül"
Burak Gül notes.js file içerisindeyiz
(base) iamburakgul@Buraks-MacBook-Air Section2-3 %

Şimdi dizeye verimizi ekleyelim ve yazdırıralım bakalım.



```
function addNote(title, body) {
  const notes = loadNotes();
  notes.push({
    title: title,
    body: body
  })
  console.log(notes)
}
```

Notes.js deki addNote fonksiyonu bu şekilde şuanlık.

Notes dizisine nesne eklerken soldaki title ve body dizi içerisindeki objelerin propertylerini belirtir sağdaki title ve body ise değerleridir bu değerler de fonksiyondan gelen argümanlardır.

The screenshot shows a VS Code interface with the following details:

- Code Editor:** The main area displays the `notes.js` file content. The code uses ES6 syntax to interact with the file system using the `fs` module. It includes functions for adding a note with a title and body, loading notes from a JSON file, and printing the notes array to the console.
- Terminal:** The bottom panel shows a terminal window with the command `node index.js add --title="Burak" --body="Gül"` and its output: `[{ title: 'Burak', body: 'Gül' }]`.
- Sidebar:** On the left, there's a sidebar with various icons for file operations like copy, paste, and search, as well as a "343" badge indicating untracked files.
- Status Bar:** The bottom bar shows file navigation (main*, index.js), status (Ln 6, Col 14), and other settings (Spaces: 4, UTF-8, LF, JavaScript, Prettier).

Şimdi ben bu veriyi yazdırmaak yerine kaydedelim.

Bunun için saveNotes adında bir fonksiyon yaratalım ve parametre olarak bir dizi alınsın.

```
Section2-3 - notes.js

const saveNotes = function (notes) {
  const dataJSON = JSON.stringify(notes);
  fs.writeFileSync("notes.json", dataJSON);
}


```

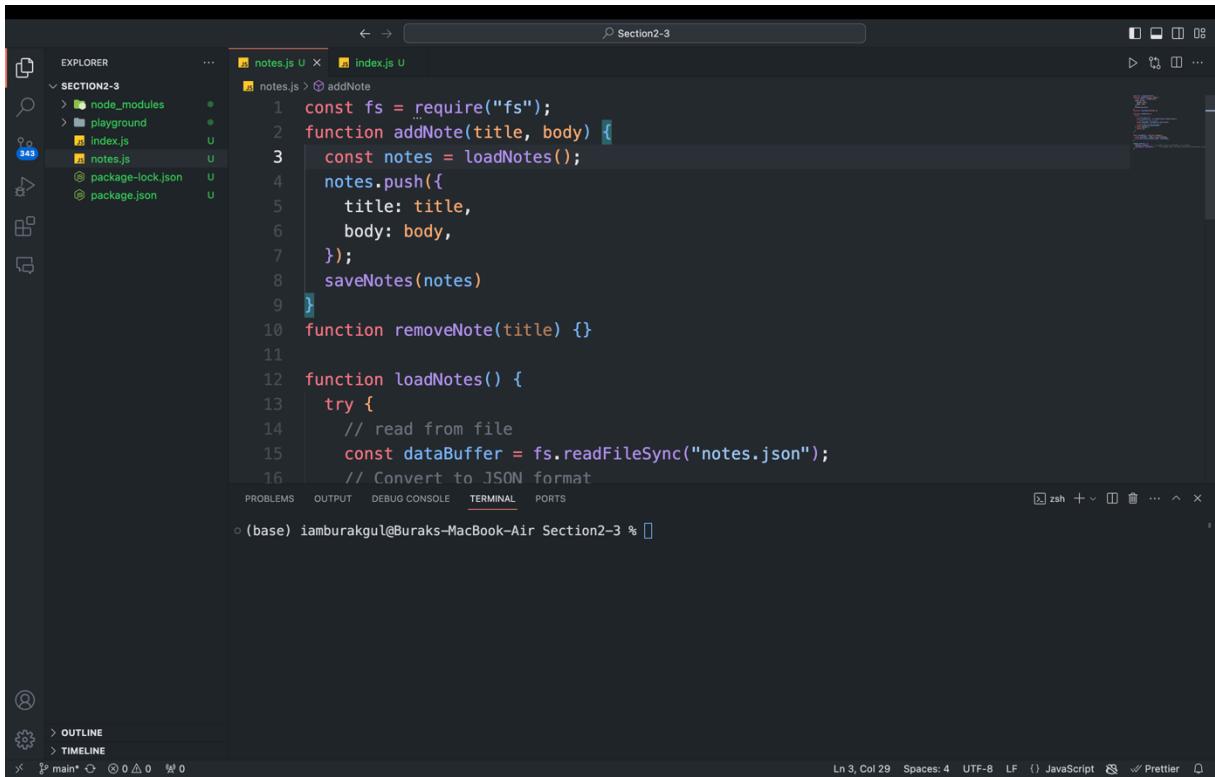
bu fonksiyonumuz ise veriyi JSON stringi olarak isliyor ve notes.json dosyasina yaziyor.

Simdi ise addNote fonksiyonunda console ile yazdırma yerine kaydedelim dosyaya.

```
Section2-3 - notes.js

function addNote(title, body) {
  const notes = loadNotes();
  notes.push({
    title: title,
    body: body,
  });
  saveNotes(notes)
}
```

Çalıştırmadan önceki hali gördüğümüz üzere notes.json dosyası yok.



```
notes.js U X Index.js U
notes.js > addNote
1 const fs = require("fs");
2 function addNote(title, body) {
3   const notes = loadNotes();
4   notes.push({
5     title: title,
6     body: body,
7   });
8   saveNotes(notes)
9 }
10 function removeNote(title) {}
11
12 function loadNotes() {
13   try {
14     // read from file
15     const dataBuffer = fs.readFileSync("notes.json");
16     // Convert to JSON format
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
(base) iamburakgul@Buraks-MacBook-Air Section2-3 % zsh
```

Şimdi ise notes.json geldi oluşturuldu.

The screenshot shows the VS Code interface with the following details:

- EXPLORER** sidebar: Shows a project structure named "SECTION2-3" containing "node_modules", "playground", "index.js", "notes.js", "notes.json", "package-lock.json", and "package.json". A file icon with the number "344" is visible.
- EDITOR**: The "notes.js" file is open, displaying the following code:

```
1 const fs = require("fs");
2 function addNote(title, body) {
3     const notes = loadNotes();
4     notes.push({
5         title: title,
6         body: body,
7     });
8     saveNotes(notes)
9 }
10 function removeNote(title) {}
11
12 function loadNotes() {
13     try {
14         // read from file
15         const dataBuffer = fs.readFileSync("notes.json");
16         // Convert to JSON format
```
- TERMINAL**: The terminal shows the command being run: `(base) iamburakgul@Buraks-MacBook-Air Section2-3 % node index.js add --title="Burak" --body="Gül"`.
- STATUS BAR**: Shows "Ln 7, Col 6" and "Spaces: 4" and "UTF-8 LF". It also indicates the file is a JavaScript file and has been prettified.

Şimdi biz verimizi de kaydediyoruz ama aynı title a sahip 2 tane note olamaz dedik bu şartı sağlayalım şimdi.

Mantık olarak bu loadNotes fonksiyonunun dönderdiği dizideki elemanları tek tek gezerim ve yeni ekleyeceğim note un title i ile eşleşiyorsa tekrarlanan diziye atarım o elemanı ve tekrarlanan dizi boş ise eklemek istediğim veriyi eklerim, değilse eklemem.

Section2-3 - notes.js

```
function addNote(title, body) {
  const notes = loadNotes();
  const duplicateNotes = notes.filter(function (note) {
    return note.title === title;
  })
  if (duplicateNotes.length === 0) { // ! no duplicate
    notes.push({ title: title, body: body });
    saveNotes(notes);
    console.log("New note added!");
  } else { // ! there is duplicate
    console.log("Note title taken");
  }
}
```

Burada tek tek gezmek içi for,forEach kullanmak yerine filter fonksiyonunu kullandık.

İlerleyen kısımlarda onu göreceğimiz için işin detayına girmiyorum şuanlık ama aynı işi yapıyor yani.

Ve deneyelim bakalım kodumuzu çalışacakmı . Daha önceden Burak gül şeklinde girmiştım. Bizim için title önemli bunu görmek için body kısmını değiştirsem de bir şey olmaz.

```
notes.js U X index.js U
notes.js > addNote
1 const fs = require("fs");
2 function addNote(title, body) {
3     const notes = loadNotes();
4     const duplicateNotes = notes.filter(function (note) {
5         return note.title === title;
6     })
7     if (duplicateNotes.length === 0) { // ! no duplicate
8         notes.push({ title: title, body: body });
9         saveNotes(notes);
10        console.log("New note added!");
11    } else { // ! there is duplicate
12        console.log("Note title taken");
13    }
14}
15 function removeNote(title) {}
```

(base) iamburakgul@Buraks-MacBook-Air Section2-3 % node index.js add --title="Burak" --body="Batuhan"

(base) iamburakgul@Buraks-MacBook-Air Section2-3 %

Şimdi ise başarılı bir veri ekleyelim.

```
notes.js U X index.js U
notes.js > addNote
1 const fs = require("fs");
2 function addNote(title, body) {
3     const notes = loadNotes();
4     const duplicateNotes = notes.filter(function (note) {
5         return note.title === title;
6     })
7     if (duplicateNotes.length === 0) { // ! no duplicate
8         notes.push({ title: title, body: body });
9         saveNotes(notes);
10        console.log("New note added!");
11    } else { // ! there is duplicate
12        console.log("Note title taken");
13    }
14}
15 function removeNote(title) {}
```

(base) iamburakgul@Buraks-MacBook-Air Section2-3 % node index.js add --title="Batuhan" --body="Gül"

(base) iamburakgul@Buraks-MacBook-Air Section2-3 %

Şimdi burada verileri ekleyince terminalde yeşil arka planda ekleyemeyince kırmızı arka planda yazmasını sağlayalım.

Bunun için npm i chalk@4 çalıştırarak chalk'ı indirelim ve
const chalk = require("chalk") ile kullanmak üzere notes.js file'ına ekleyelim.
son olarak addNote fonksiyonundaki console kısımlarını güncelleyelim.

```
Section2-3 - notes.js

function addNote(title, body) {
  const notes = loadNotes();
  const duplicateNotes = notes.filter(function (note) {
    return note.title === title;
  })
  if (duplicateNotes.length === 0) { // ! no duplicate
    notes.push({ title: title, body: body });
    saveNotes(notes);
    console.log(chalk.green.inverse( "New note added!"))
  } else { // ! there is duplicate
    console.log(chalk.red.inverse("Note title taken"))
  }
}
```

İşte addNote fonksiyonunun en son hali.

Örnek yapalım şimdi de.

The screenshot shows a VS Code interface with the following details:

- EXPLORER** sidebar: Shows a project structure named "SECTION2-3" containing "node_modules", "playground", "index.js", "notes.js", "notes.json", "package-lock.json", and "package.json".
- TERMINAL** tab: Displays the output of running node.js commands:
 - (base) iamburakgul@Buraks-MacBook-Air Section2-3 % node index.js add --title="Batuhan" --body="Gül"
Note title taken
 - (base) iamburakgul@Buraks-MacBook-Air Section2-3 % node index.js add --title="BTU" --body="BM"
New note added!
 - (base) iamburakgul@Buraks-MacBook-Air Section2-3 %
- STATUS BAR**: Shows "Ln 16, Col 30" and "Spaces: 4, UTF-8, LF, JavaScript, Prettier".

Ekleme işi tamam olduğuna göre silme işlemine geçelim.

Remove için command kısmını düzenleyelim şimdi de.

Section2-3 - index.js

```
yargs.command({  
  command: "remove",  
  describe: "Remove a new note",  
  builder: {  
    title: {  
      describe: "Note title",  
      demandOption: true,  
      type: "string",  
    },  
  },  
  handler: function (argv) {  
    notes.removeNote(argv.title)  
  },  
})
```

Notes.js içindeki removeNote fonksiyonunu da şu şekilde yapıp çalıştığını görelim.

Section2-3 - notes.js

```
function removeNote(title) {  
  console.log(title,"notes.js file içindeyiz")  
}
```

The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows a project structure with files: node_modules, playground, index.js, notes.js, notes.json, package-lock.json, and package.json.
- Code Editor:** The notes.js file is open, displaying code for a notes application. It includes functions for adding, removing, and loading notes from a JSON file.
- Terminal:** The terminal shows the command `node index.js remove --title="BTU"` being run, with the output "notes.js file içindeyiz".
- Status Bar:** Shows the current file is notes.js, the line number is 16, the column number is 1, 79 lines are selected, and the file is in JavaScript mode.

Şimdi bu removeNote fonksiyonunu düzeltelim ve kullanalım.

Nasıl çalışmalıdır bu fonksiyon ?

- 1- Önce dosyadan verileri almalıyız.
- 2- Aldığımız verilerin içerisinde gezmemeli ve bu verilerin title özelliği ile bizim silmek istediğimiz için yazdığımız title ile karşılaşmalıdır.
- 3- Karşılaştırdığımız verilerden uymayanları yeni bir diziye atayalım ve bu diziyi file a yeniden yazalım.

Terminale de silme işlemi başarılı olunca arka planı yeşil şekilde başarısız olunca ise kırmızı şekilde yazdırılacak.

İşte removeNote fonksiyonumuz.

```
Section2-3 - notes.js

function removeNote(title) {
  const notes = loadNotes();
  const notesToKeep = notes.filter(function (note) {
    return note.title !== title;
  });
  if (notes.length > notesToKeep.length) {
    console.log(chalk.green.inverse("Note removed"));
    saveNotes(notesToKeep);
  } else {
    console.log(
      chalk.red.inverse("The note you wanted to delete was not found")
    );
  }
}
```

Burada tek tek gezmek yerine yani forEach kullanmak yerine filter fonksiyonunu kullanalım yazması ve anlaması daha rahat olmaktadır.

The screenshot shows the VS Code interface with the following details:

- EXPLORER** sidebar: Shows the project structure with files: node_modules, playground, index.js, notes.js, notes.json, package-lock.json, and package.json.
- EDITOR**: The notes.js file is open, showing the code for adding and removing notes. The removeNote function uses the filter method to keep notes that don't match the title.
- TERMINAL**: The terminal window shows the execution of the script and its output:
 - (base) iamburakgul@Buraks-MacBook-Air Section2-3 % node index.js remove --title="BTU"
 - Note removed
 - (base) iamburakgul@Buraks-MacBook-Air Section2-3 % node index.js remove --title="Burakk"
 - The note you wanted to delete was not found
 - (base) iamburakgul@Buraks-MacBook-Air Section2-3 %
- STATUS BAR**: Shows the current line (Ln 18, Col 1), character count (386 selected), spaces setting (Spaces: 4), encoding (UTF-8), file type (JavaScript), and Prettier status.