Lecture 21 | Web Scraping and Excel Data Extraction

JSON `JavaScript Object Notation`

- JSON
- JavaScript Object Notation, just like JavaScript Objects
- Stores data in a key value pair format

JSON Data

```json
{
  "name": "Milind Mishra",
  "age": 23,
  "height": "5'11",
  "weight": "85kgs",
  "isAvenger": false,
  "isCaptain": false,
  "address": {
    "city": "Bangalore",
    "state": "Karnataka",
    "country": "India"
  }
}
```

> The following example shows a possible JSON representation describing a person.

```json
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [],
  "spouse": null
}
```

`JSON` things to know

- JSON is a text format
- **JSON** is an acronym for `JavaScript Object Notation`
- in JSON, the keys are always strings enclosed in "double quotes", and not enclosed in single quotes
- in JSON we cant have comments
- both key value pairs are enclosed in curly braces, and itself in double quotes
- for a boolean value in JSON, we use true or false without quotes
- for a null value in JSON, we use null without quotes
- after last property we do not need a comma
- if a comma is given after a property JSON expects another property after it
- to keep multiple objects in a single JSON file, we use an **array**

```
[
  {
    "name": "Milind Mishra",
    "age": 23,
    "height": "5'11",
    "weight": "85kgs",
    "isAvenger": true,
    "isCaptain": false,
    "address": {
      "city": "Bangalore",
      "state": "Karnataka",
      "country": "India"
    }
  },
  {
    "name": "Rajesh Sharma",
    "age": 25,
    "height": "5'10",
    "weight": "80kgs",
    "isAvenger": true,
    "isCaptain": false,
    "address": {
      "city": "Bangalore",
      "state": "Karnataka",
      "country": "India"
    }
  }
]
```

- A *good coding practice* while making multiple object JSON while using an array and not change the sequence of key value pairs

API *Application Programming Interface*

- This simple JSON that we created can be called a small API
- It is a way to communicate with other programs
- It can be used to fetch details here in this case we are using it to fetch details of a `person`
- JSON API examples

> `funWithJson.js`

```
const log = console.log;
const fs = require("fs");

// whenever a file is read it gets read as a buffer data
let buffer = fs.readFileSync("example.json");
log(buffer);
// to parse the buffer we need to convert it to a string
let json = buffer.toString();
log(json);
```

> Output :

```
> node funWithJson.js
<Buffer 5b 0a 20 20 7b 0a 20 20 20 20 22 6e 61 6d 65 22 3a 20 22 4d 69 6c
69 6e 64 20 4d 69 73 68 72 61 22 2c 0a 20 20 20 20 22 61 67 65 22 3a 20 32
33 2c 0a ... 449 more bytes>
[
  {
    "name": "Milind Mishra",
    "age": 23,
    "height": "5'11",
    "weight": "85kgs",
    "isAvenger": true,
    "isCaptain": false,
    "address": {
      "city": "Bangalore",
      "state": "Karnataka",
      "country": "India"
    }
  },
  {
    "name": "Rajesh Sharma",
    "age": 25,
    "height": "5'10",
    "weight": "80kgs",
    "isAvenger": true,
    "isCaptain": false,
    "address": {
      "city": "Bangalore",
      "state": "Karnataka",
      "country": "India"
    }
  }
]
```

> Manupilating an JSON file using the fs module

```
const log = console.log;
const fs = require("fs");

// whenever a file is read it gets read as a buffer data
let buffer = fs.readFileSync("example.json");
// log(buffer);
// to parse the buffer we need to convert it to a string
// let data = buffer.toString();
// to parse JSON we use the JSON.parse() method
let data = JSON.parse(buffer);
// log(data);

// since data is now an array of objects we can manupulate it using array
methords

data.push({
  name: "Manish Malhotra",
  age: "28",
  height: "5.8",
  weight: "65",
  isAvenger: true,
  isCaptain: false,
  address: {
    city: "Bangalore",
    state: "Karnataka",
    country: "India",
  },
});

log(data);

// to convert the data back to a buffer we use the JSON.stringify() method
let stringData = JSON.stringify(data);
log(stringData);

// the purpose to stringify was to use fs.writeFileSync() since
fs.writeFileSync() only accepts strings
fs.writeFileSync("example.json", stringData);
log("File written successfully");
```

> Output :

```
> node funWithJson.js
[
  {
    name: 'Milind Mishra',
    age: 23,
    height: "5'11",
    weight: '85kgs',
    isAvenger: true,
    isCaptain: false,
    address: { city: 'Bangalore', state: 'Karnataka', country: 'India' }
  },
  {
    name: 'Rajesh Sharma',
    age: 25,
    height: "5'10",
    weight: '80kgs',
    isAvenger: true,
    isCaptain: false,
    address: { city: 'Bangalore', state: 'Karnataka', country: 'India' }
  },
  {
    name: 'Manish Malhotra',
    age: '28',
    height: '5.8',
    weight: '65',
    isAvenger: true,
    isCaptain: false,
    address: { city: 'Bangalore', state: 'Karnataka', country: 'India' }
  }
]
[{"name":"Milind
Mishra","age":23,"height":"5'11","weight":"85kgs","isAvenger":true,"isCapt
ain":false,"address":
{"city":"Bangalore","state":"Karnataka","country":"India"}},
{"name":"Rajesh
Sharma","age":25,"height":"5'10","weight":"80kgs","isAvenger":true,"isCapt
ain":false,"address":
{"city":"Bangalore","state":"Karnataka","country":"India"}},
{"name":"Manish
Malhotra","age":"28","height":"5.8","weight":"65","isAvenger":true,"isCapt
ain":false,"address":
{"city":"Bangalore","state":"Karnataka","country":"India"}}]
File written successfully
```

Alternatively without using fs module use `const jsonFile = require("./example.json");`

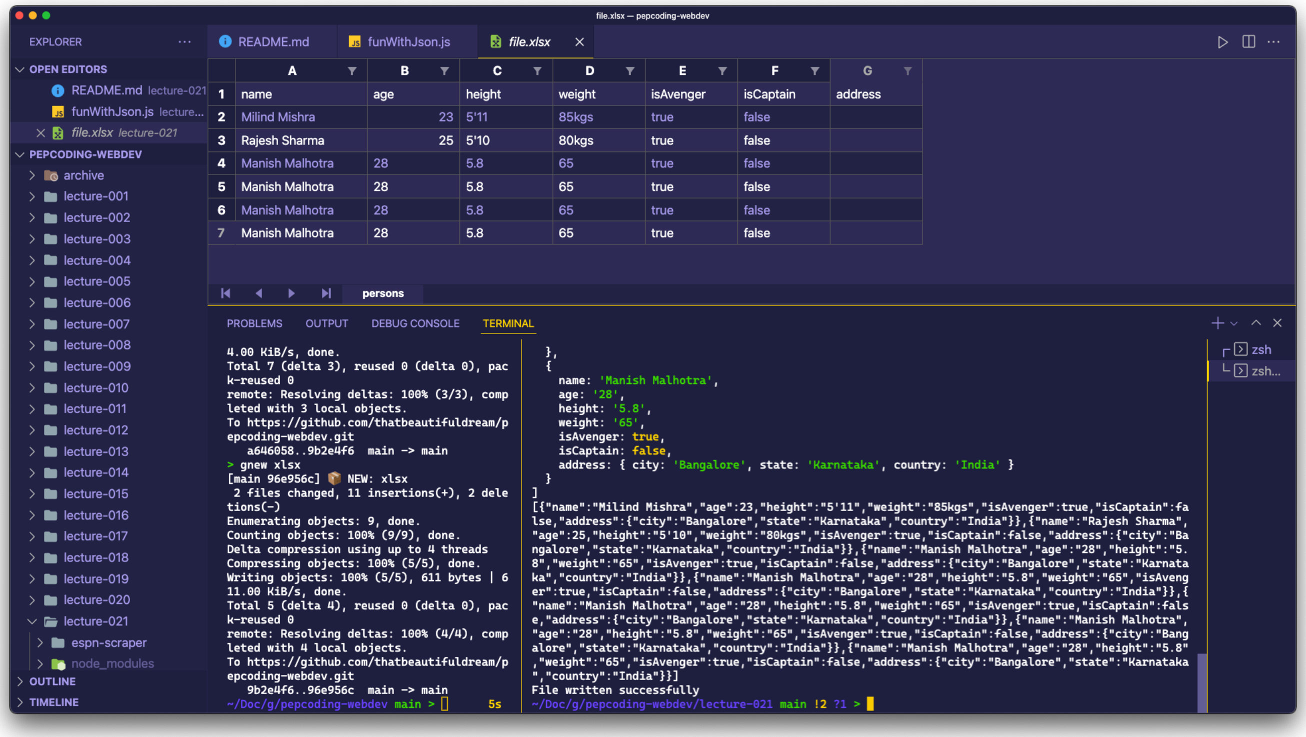
```
// all the buffer data conversion to string and all strings conversion to
JSON is now not required
const jsonFile = require("./example.json");
```

```
// all data is now an array of objects stored in jsonFile variable

const log = console.log;

jsonFile.push({
  name: "Manish Malhotra",
  age: "28",
  height: "5.8",
  weight: "65",
  isAvenger: true,
  isCaptain: false,
  address: {
    city: "Bangalore",
    state: "Karnataka",
    country: "India",
  },
});

log(jsonFile);
```

Working with Excel files

- Install xlsx library using `npm install xlsx`

```
const log = console.log;
const fs = require("fs");
const xlsx = require("xlsx");

// all the buffer data conversion to string and all strings conversion to
JSON is now not required
const jsonFile = require("./example.json");
// all data is now an array of objects stored in jsonFile variable

// whenever a file is read it gets read as a buffer data
let buffer = fs.readFileSync("example.json");
// log(buffer);
// to parse the buffer we need to convert it to a string
// let data = buffer.toString();
// to parse JSON we use the JSON.parse() method
let data = JSON.parse(buffer);
// log(data);

// since data is now an array of objects we can manupulate it using array
methords

data.push({
  name: "Manish Malhotra",
  age: "28",
  height: "5.8",
  weight: "65",
  isAvenger: true,
    isCaptain: false,
    address: {
      city: "Bangalore",
      state: "Karnataka",
      country: "India",
    },
  });

log(data);

// to convert the data back to a buffer we use the JSON.stringify() method
let stringData = JSON.stringify(data);
log(stringData);

// the purpose to stringify was to use fs.writeFileSync() since
fs.writeFileSync() only accepts strings
fs.writeFileSync("example.json", stringData);
log("File written successfully");

// flow : workbook -> worksheet (convert into rows and cols) expects a
JSON data -> excel file

// writing to excel file

let newWB = xlsx.utils.book_new(); // create a new workbook file

let newWS = xlsx.utils.json_to_sheet(jsonFile); // convert the json data
to a worksheet (sheet format)
xlsx.utils.book_append_sheet(newWB, newWS, "persons"); // append the
worksheet to the workbook
xlsx.writeFile(newWB, "file.xlsx"); // write the workbook to a file
```

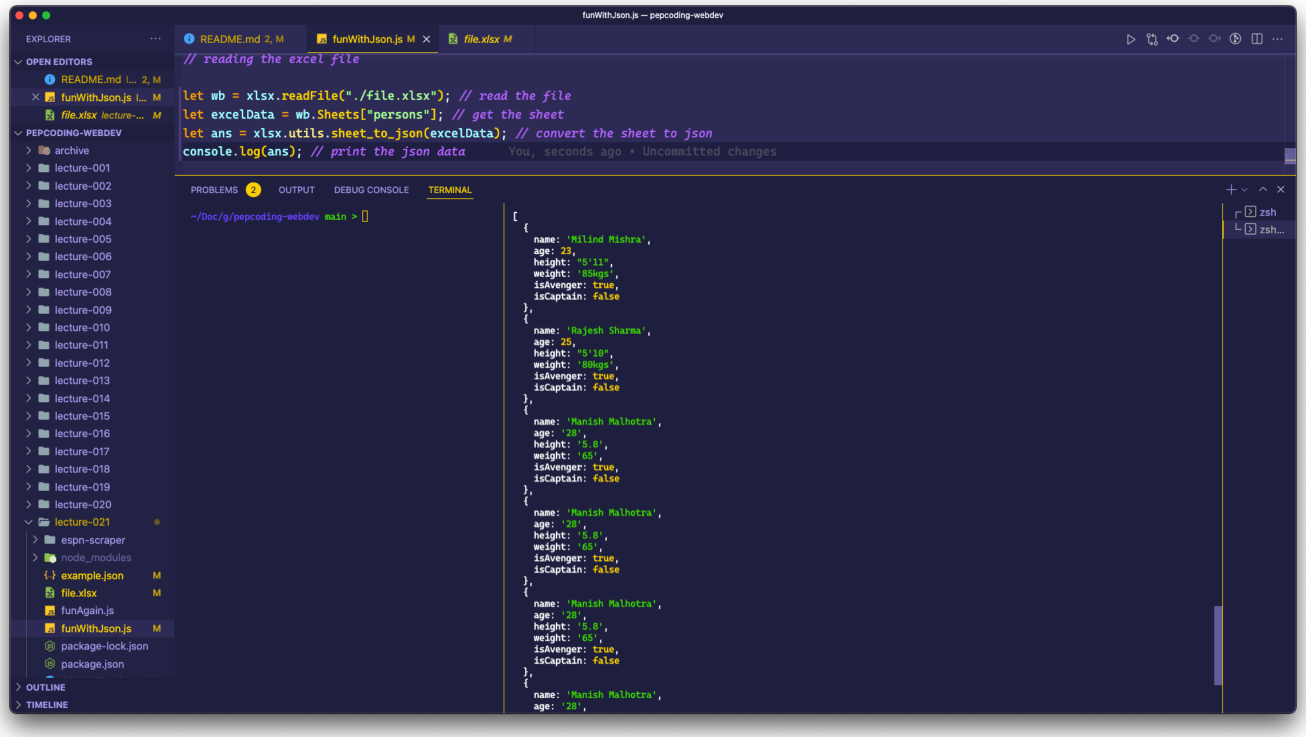


	A	B	C	D	E	F	G
1	name	age	height	weight	isAvenger	isCaptain	address
2	Milind Mishra	23	5'11	85kgs	true	false	
3	Rajesh Sharma	25	5'10	80kgs	true	false	
4	Manish Malhotra	28	5.8	65	true	false	
5	Manish Malhotra	28	5.8	65	true	false	
6	Manish Malhotra	28	5.8	65	true	false	
7	Manish Malhotra	28	5.8	65	true	false	

```
4.00 KiB/s, done.
Total 7 (delta 3), reused 0 (delta 0), pac
k-reused 0
remote: Resolving deltas: 100% (3/3), comp
leted with 3 local objects.
To https://github.com/thatbeautifuldream/p
epcoding-webdev.git
   a646058..9b2e4f6  main -> main
> gnew xlsx
[main 96e956c] 📦 NEW: xlsx
 2 files changed, 11 insertions(+), 2 dele
tions(-)
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 611 bytes | 6
11.00 KiB/s, done.
Total 5 (delta 4), reused 0 (delta 0), pac
k-reused 0
remote: Resolving deltas: 100% (4/4), comp
leted with 4 local objects.
To https://github.com/thatbeautifuldream/p
epcoding-webdev.git
   9b2e4f6..96e956c  main -> main
~/Doc/g/pepcoding-webdev main >          5s
```

```
  },
  {
    name: 'Manish Malhotra',
    age: '28',
    height: '5.8',
    weight: '65',
    isAvenger: true,
    isCaptain: false,
    address: { city: 'Bangalore', state: 'Karnataka', country: 'India' }
  }
]
[{"name":"Milind Mishra","age":23,"height":"5'11","weight":"85kgs","isAvenger":true,"isCaptain":fa
lse,"address":{"city":"Bangalore","state":"Karnataka","country":"India"}},{"name":"Rajesh Sharma",
"age":25,"height":"5'10","weight":"80kgs","isAvenger":true,"isCaptain":false,"address":{"city":"Ba
ngalore","state":"Karnataka","country":"India"}},{"name":"Manish Malhotra","age":"28","height":"5.
8","weight":"65","isAvenger":true,"isCaptain":false,"address":{"city":"Bangalore","state":"Karnata
ka","country":"India"}},{"name":"Manish Malhotra","age":"28","height":"5.8","weight":"65","isAveng
er":true,"isCaptain":false,"address":{"city":"Bangalore","state":"Karnataka","country":"India"}},{
"name":"Manish Malhotra","age":"28","height":"5.8","weight":"65","isAvenger":true,"isCaptain":fals
e,"address":{"city":"Bangalore","state":"Karnataka","country":"India"}},{"name":"Manish Malhotra",
"age":"28","height":"5.8","weight":"65","isAvenger":true,"isCaptain":false,"address":{"city":"Bang
alore","state":"Karnataka","country":"India"}},{"name":"Manish Malhotra","age":"28","height":"5.8"
,"weight":"65","isAvenger":true,"isCaptain":false,"address":{"city":"Bangalore","state":"Karnataka
","country":"India"}}]
File written successfully
~/Doc/g/pepcoding-webdev/lecture-021 main !2 ?1 >
```

```
// reading the excel file

let wb = xlsx.readFile("./file.xlsx"); // read the file
let excelData = wb.Sheets["persons"]; // get the sheet
let ans = xlsx.utils.sheet_to_json(excelData); // convert the sheet to
json
console.log(ans); // print the json data
```



```
// reading the excel file

let wb = xlsx.readFile("./file.xlsx"); // read the file
let excelData = wb.Sheets["persons"]; // get the sheet
let ans = xlsx.utils.sheet_to_json(excelData); // convert the sheet to json
console.log(ans); // print the json data     You, seconds ago • Uncommitted changes
```

```
~/Doc/g/pepcoding-webdev main >
```

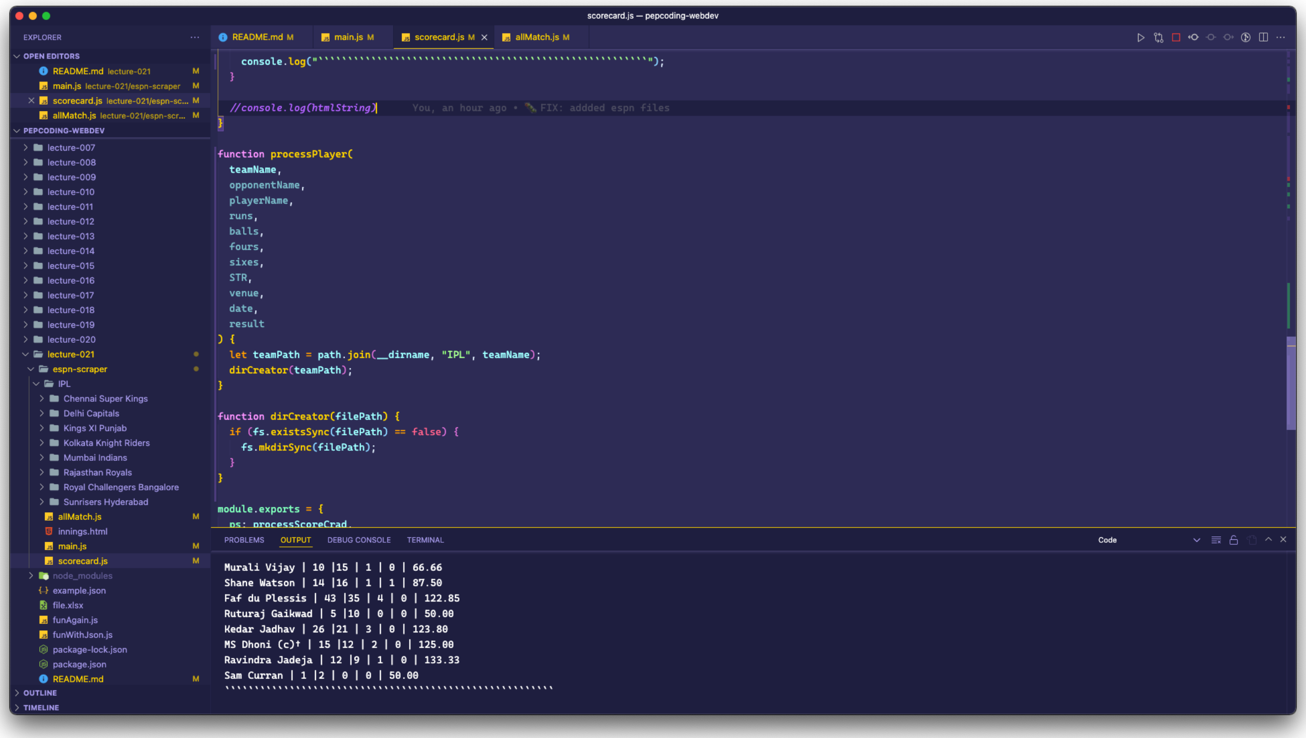
```
[
  {
    name: 'Milind Mishra',
    age: 23,
    height: "5'11",
    weight: '85kgs',
    isAvenger: true,
    isCaptain: false
  },
  {
    name: 'Rajesh Sharma',
    age: 25,
    height: "5'10",
    weight: '80kgs',
    isAvenger: true,
    isCaptain: false
  },
  {
    name: 'Manish Malhotra',
    age: '28',
    height: '5.8',
    weight: '65',
    isAvenger: true,
    isCaptain: false
  },
  {
    name: 'Manish Malhotra',
    age: '28',
    height: '5.8',
    weight: '65',
    isAvenger: true,
    isCaptain: false
  },
  {
    name: 'Manish Malhotra',
    age: '28',
    height: '5.8',
    weight: '65',
    isAvenger: true,
    isCaptain: false
  },
  {
    name: 'Manish Malhotra',
    age: '28',
```

ESPN Scraper and Excel Integration

> Making individual folders with team names fetched from ESPN website using function : `processPlayer()`

```
function processPlayer(
  teamName,
  opponentName,
  playerName,
  runs,
  balls,
  fours,
  sixes,
  STR,
  venue,
  date,
  result
) {
  let teamPath = path.join(__dirname, "IPL", teamName);
  dirCreator(teamPath);
}

function dirCreator(filePath) {
  if (fs.existsSync(filePath) == false) {
    fs.mkdirSync(filePath);
  }
}
```

> Got all the individual team folders using `dirCreator()` function



```
    console.log("``````````````````````````````````````````````````````````");
  }

  //console.log(htmlString)        You, an hour ago • FIX: addded espn files
}

function processPlayer(
  teamName,
  opponentName,
  playerName,
  runs,
  balls,
  fours,
  sixes,
  STR,
  venue,
  date,
  result
) {
  let teamPath = path.join(__dirname, "IPL", teamName);
  dirCreator(teamPath);
}

function dirCreator(filePath) {
  if (fs.existsSync(filePath) == false) {
    fs.mkdirSync(filePath);
  }
}

module.exports = {
  ps: processScoreCrad,
```

```
Murali Vijay | 10 |15 | 1 | 0 | 66.66
Shane Watson | 14 |16 | 1 | 1 | 87.50
Faf du Plessis | 43 |35 | 4 | 0 | 122.85
Ruturaj Gaikwad | 5 |10 | 0 | 0 | 50.00
Kedar Jadhav | 26 |21 | 3 | 0 | 123.80
MS Dhoni (c)† | 15 |12 | 2 | 0 | 125.00
Ravindra Jadeja | 12 |9 | 1 | 0 | 133.33
Sam Curran | 1 |2 | 0 | 0 | 50.00
``````````````````````````````````````````````````````````
```

## Excel read and write using function : `excelReader()` & `excelWriter()`

```
// writing the excel file
function excelWriter(file) {
  let newWB = xlsx.utils.book_new(); // create a new workbook file
  let newWS = xlsx.utils.json_to_sheet(jsonFile); // convert the json data to a worksheet (sheet format)
  xlsx.utils.book_append_sheet(newWB, newWS, "persons"); // append the worksheet to the workbook
  xlsx.writeFile(newWB, "file.xlsx"); // write the workbook to a file
}

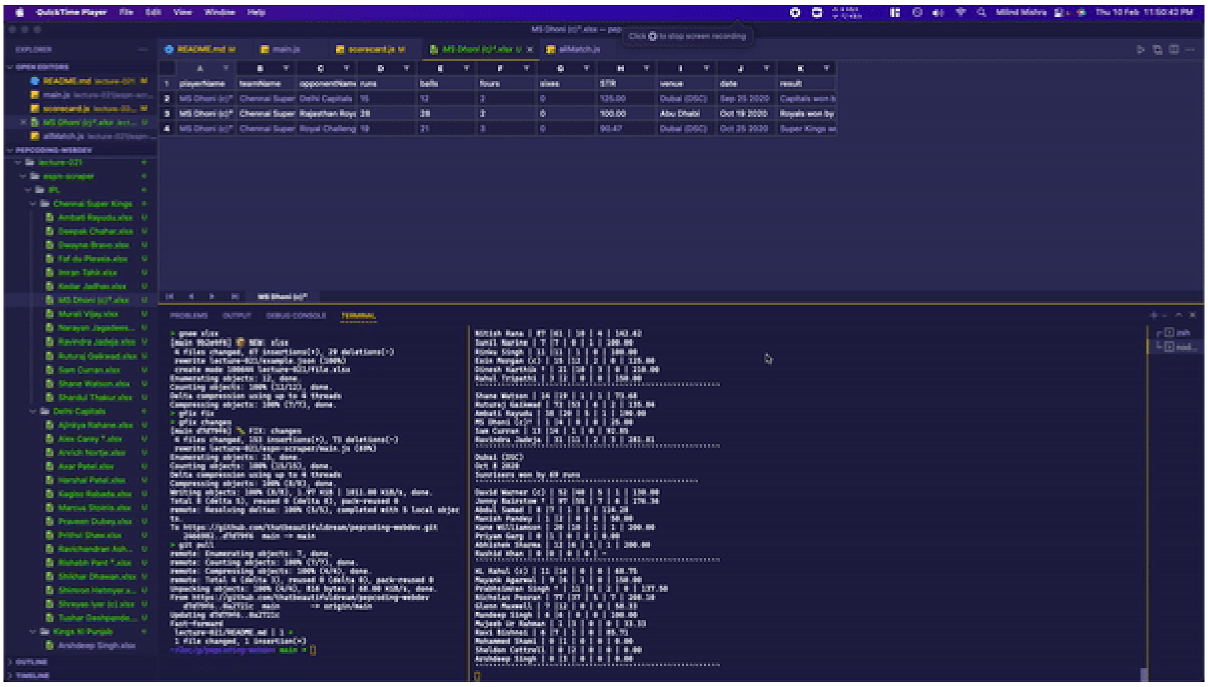
// reading the excel file
function excelReader(file, sheetName) {
  if (fs.existsSync(file) == false) {
    return [];
  }
  let wb = xlsx.readFile("./file.xlsx"); // read the file
  let excelData = wb.Sheets["persons"]; // get the sheet
  let ans = xlsx.utils.sheet_to_json(excelData); // convert the sheet to
```
```

```
json
  return ans; // return the json data
}
```

> Adding individual team data to the excel file using `excelWriter()` function

```
function processPlayer(
  teamName,
  opponentName,
  playerName,
  runs,
  balls,
  fours,
  sixes,
  STR,
  venue,
  date,
  result
) {
  let teamPath = path.join(__dirname, "IPL", teamName);
  dirCreator(teamPath);

  let filePath = path.join(teamPath, playerName + ".xlsx");

  let content = excelReader(filePath, playerName);
  [];

  let playerObj = {
    playerName,
    teamName,
    opponentName,
    runs,
    balls,
    fours,
    sixes,
    STR,
    venue,
    date,
    result,
  };

  content.push(playerObj);

  excelWriter(filePath, playerName, content);
}

function dirCreator(folderPath) {
  if (fs.existsSync(folderPath) == false) {
    fs.mkdirSync(folderPath);
  }
}
```

```
function excelWriter(fileName, sheetName, jsonData) {
  let newWB = xlsx.utils.book_new();
  // Creating a new WorkBook
  let newWS = xlsx.utils.json_to_sheet(jsonData);
  // Json is converted to sheet format (rows and cols)
  xlsx.utils.book_append_sheet(newWB, newWS, sheetName);
  xlsx.writeFile(newWB, fileName);
}

function excelReader(fileName, sheetName) {
  if (fs.existsSync(fileName) == false) {
    return [];
  }
  let wb = xlsx.readFile(fileName);

  let excelData = wb.Sheets[sheetName];
  let ans = xlsx.utils.sheet_to_json(excelData);
  return ans;
}

module.exports = {
  ps: processScoreCrad,
};
```



	A	B	C	D	E	F	G	H	I	J	K
1	playerName	teamName	opponentName	runs	balls	fours	sixes	STR	venue	date	result
2	MS Dhoni (c)*	Chennai Super	Delhi Capitals	15	12	2	0	125.00	Dubai (DSC)	Sep 25 2020	Capitals won b
3	MS Dhoni (c)*	Chennai Super	Rajasthan Roys	28	28	2	0	100.00	Abu Dhabi	Oct 19 2020	Royals won by
4	MS Dhoni (c)*	Chennai Super	Royal Challeng	19	21	3	0	90.47	Dubai (DSC)	Oct 25 2020	Super Kings w