Walchand College of Engineering, Sangli
Department of Computer Science and Engineering

Class: Third Year (Computer Science and Engineering)

Year: 2020-21 **Semester:** 2

Course: Advanced Web and Mobile Application Development Lab (4CS381)

Practical No. 7

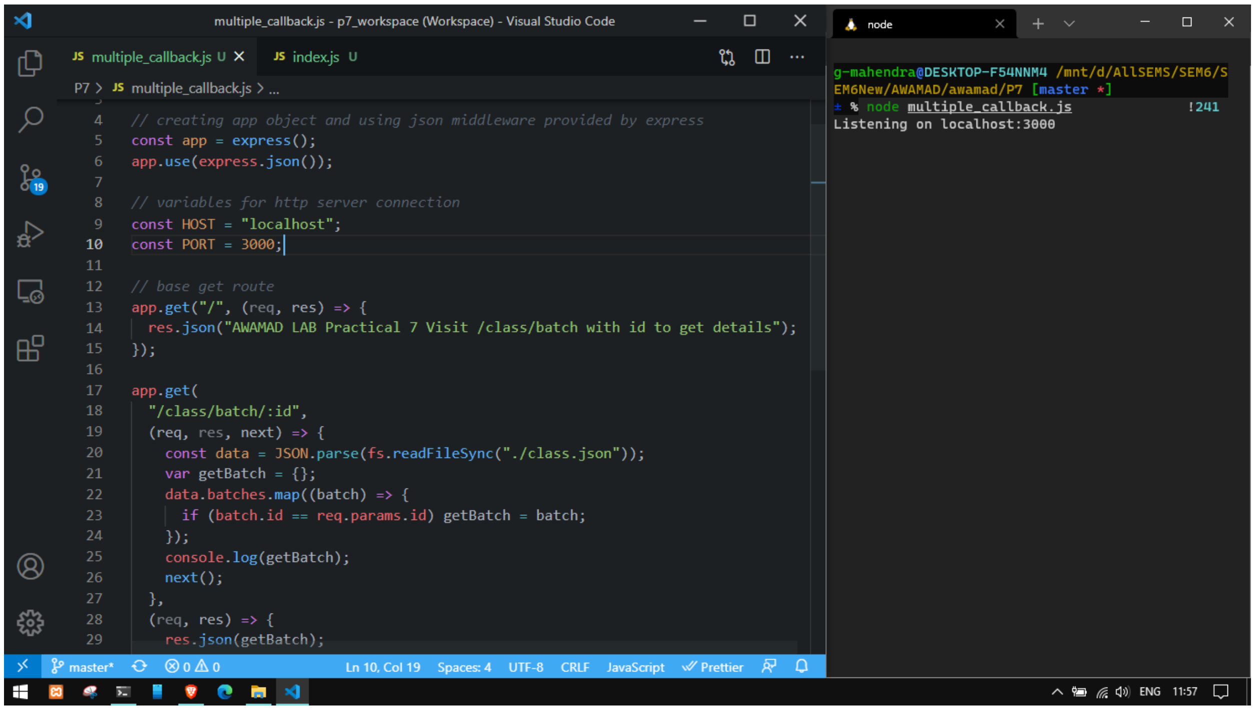
Exam Seat No:

1. 2018BTECS00033- Mahendra Bhimrao Gharge
2. GitHub URL: https://github.com/g-mahendra/awamad

Problem Statement 1:

1. Write an Express app that uses the multiple call-back functions to handle a request.
2. Use different routes to show the use of call-back functions.
3. Use at least two call-back functions.
4. Use array of functions as an argument.

Screenshot 1:



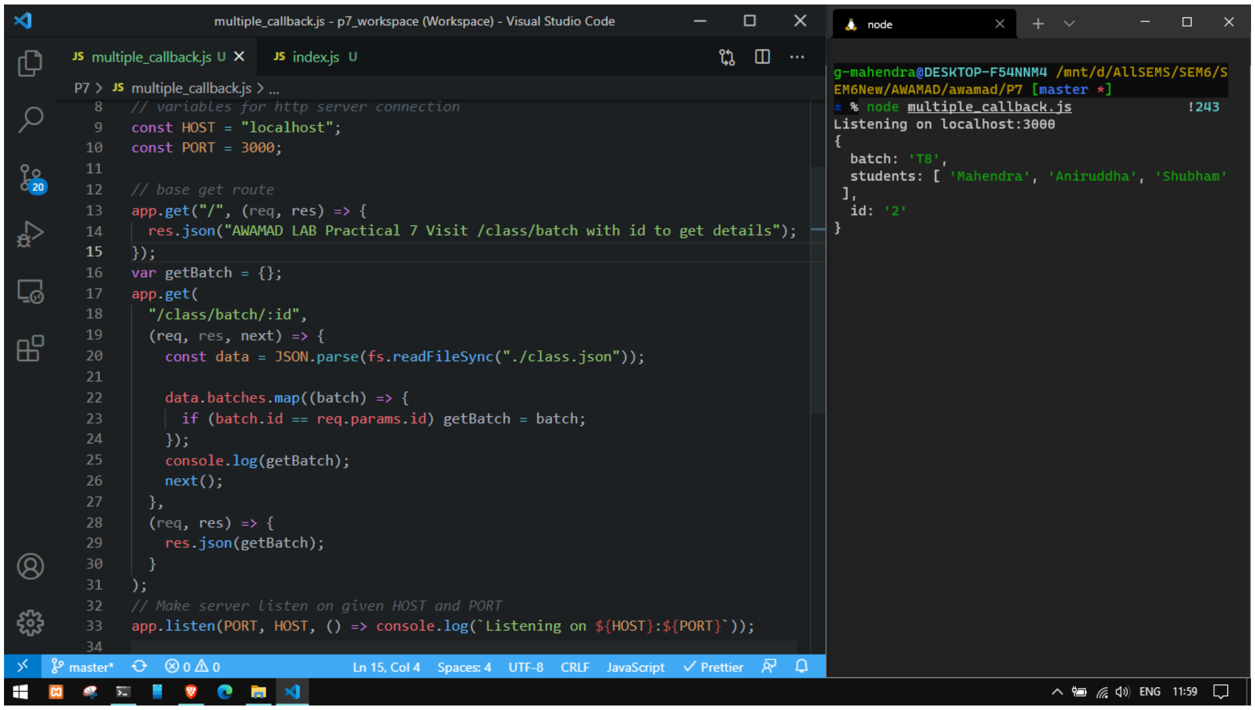
multiple_callback.js - p7_workspace (Workspace) - Visual Studio Code

```javascript
// creating app object and using json middleware provided by express
const app = express();
app.use(express.json());

// variables for http server connection
const HOST = "localhost";
const PORT = 3000;

// base get route
app.get("/", (req, res) => {
  res.json("AWAMAD LAB Practical 7 Visit /class/batch with id to get details");
});

app.get(
  "/class/batch/:id",
  (req, res, next) => {
    const data = JSON.parse(fs.readFileSync("./class.json"));
    var getBatch = {};
    data.batches.map((batch) => {
      if (batch.id == req.params.id) getBatch = batch;
    });
    console.log(getBatch);
    next();
  },
  (req, res) => {
    res.json(getBatch);
```

```
g-mahendra@DESKTOP-F54NNM4 /mnt/d/AllSEMS/SEM6/SEM6New/AWAMAD/awamad/P7 [master *]
± % node multiple_callback.js                                    !241
Listening on localhost:3000
```

Information 1: Installed express with **npm i express**

Screenshot 2:



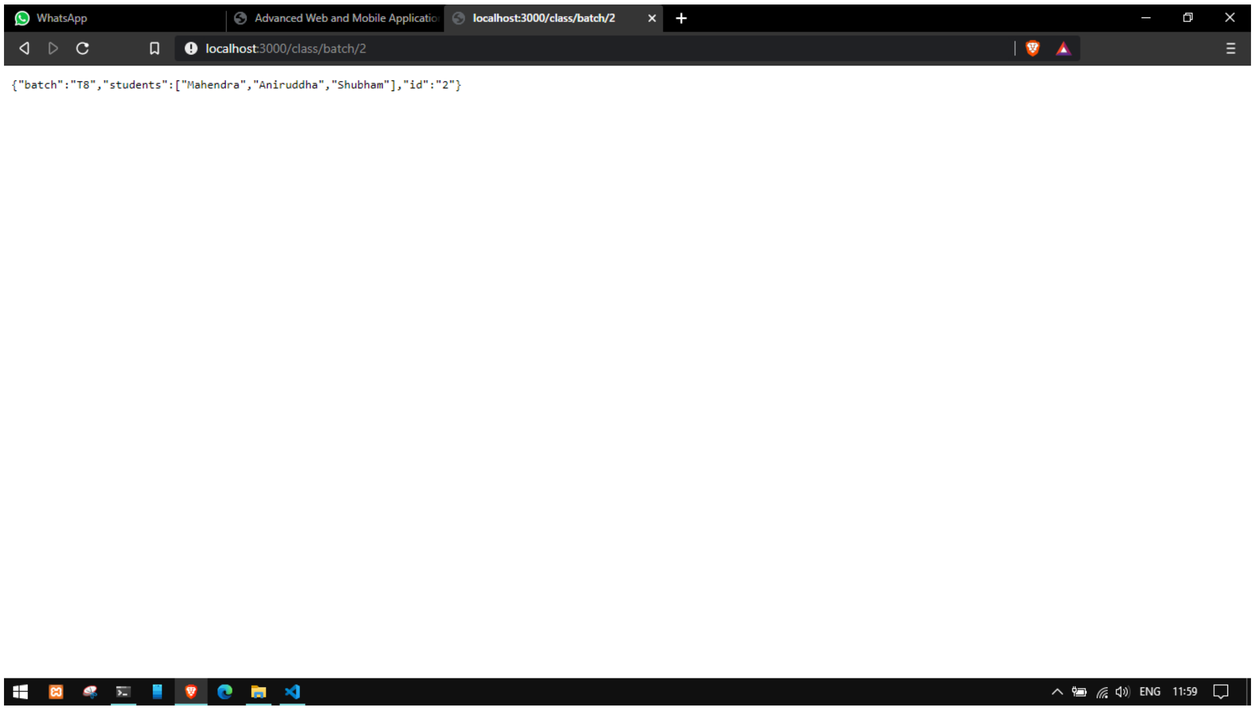
multiple_callback.js - p7_workspace (Workspace) - Visual Studio Code

```
// variables for http server connection
const HOST = "localhost";
const PORT = 3000;

// base get route
app.get("/", (req, res) => {
  res.json("AWAMAD LAB Practical 7 Visit /class/batch with id to get details");
});
var getBatch = {};
app.get(
  "/class/batch/:id",
  (req, res, next) => {
    const data = JSON.parse(fs.readFileSync("./class.json"));

    data.batches.map((batch) => {
      if (batch.id == req.params.id) getBatch = batch;
    });
    console.log(getBatch);
    next();
  },
  (req, res) => {
    res.json(getBatch);
  }
);
// Make server listen on given HOST and PORT
app.listen(PORT, HOST, () => console.log(`Listening on ${HOST}:${PORT}`));
```

```
g-mahendra@DESKTOP-F54NNM4 /mnt/d/AllSEMS/SEM6/SEM6New/AWAMAD/awamad/P7 [master *]
± % node multiple_callback.js                                           !243
Listening on localhost:3000
{
  batch: 'T8',
  students: [ 'Mahendra', 'Aniruddha', 'Shubham'
 ],
  id: '2'
}
```

Information 2: Using multiple callbacks for retrieving and sending JSON data.

Screenshot 3:

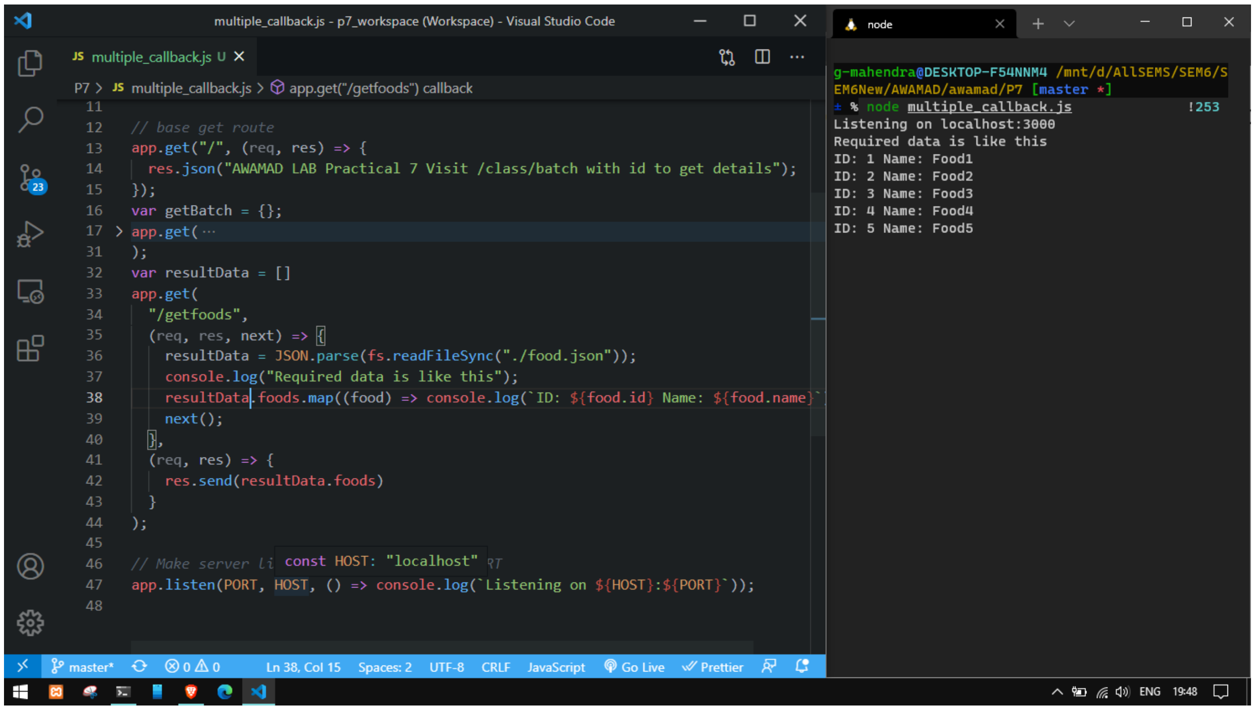


localhost:3000/class/batch/2

```
{"batch":"T8","students":["Mahendra","Aniruddha","Shubham"],"id":"2"}
```

Information 3: Output of data sent using multiple callbacks

Screenshot 4:

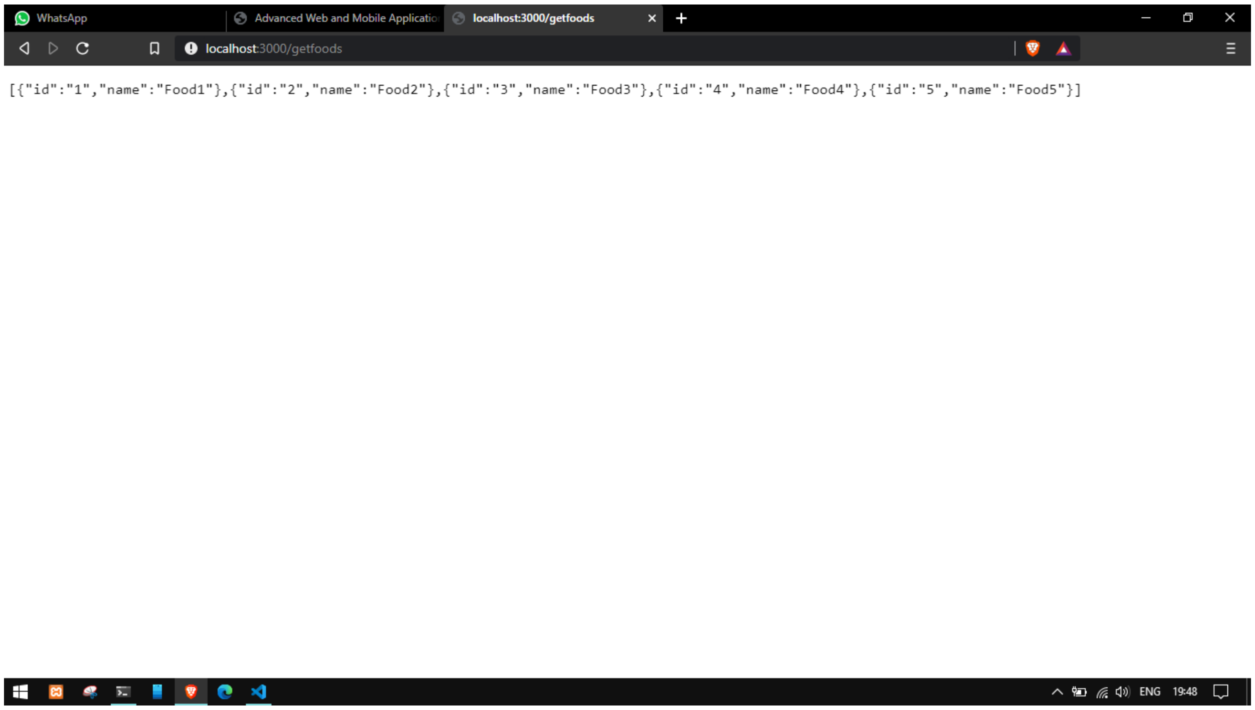


```
// base get route
app.get("/", (req, res) => {
  res.json("AWAMAD LAB Practical 7 Visit /class/batch with id to get details");
});
var getBatch = {};
app.get(⋯
);
var resultData = []
app.get(
  "/getfoods",
  (req, res, next) => {
    resultData = JSON.parse(fs.readFileSync("./food.json"));
    console.log("Required data is like this");
    resultData.foods.map((food) => console.log(`ID: ${food.id} Name: ${food.name}`
    next();
  },
  (req, res) => {
    res.send(resultData.foods)
  }
);

// Make server li const HOST: "localhost" RT
app.listen(PORT, HOST, () => console.log(`Listening on ${HOST}:${PORT}`));
```

```
g-mahendra@DESKTOP-F54NNM4 /mnt/d/AllSEMS/SEM6/SEM6New/AWAMAD/awamad/P7 [master *]
± % node multiple_callback.js                                    !253
Listening on localhost:3000
Required data is like this
ID: 1 Name: Food1
ID: 2 Name: Food2
ID: 3 Name: Food3
ID: 4 Name: Food4
ID: 5 Name: Food5
```

Information 4: Using route /getfoods to send data of foods using two call back functions

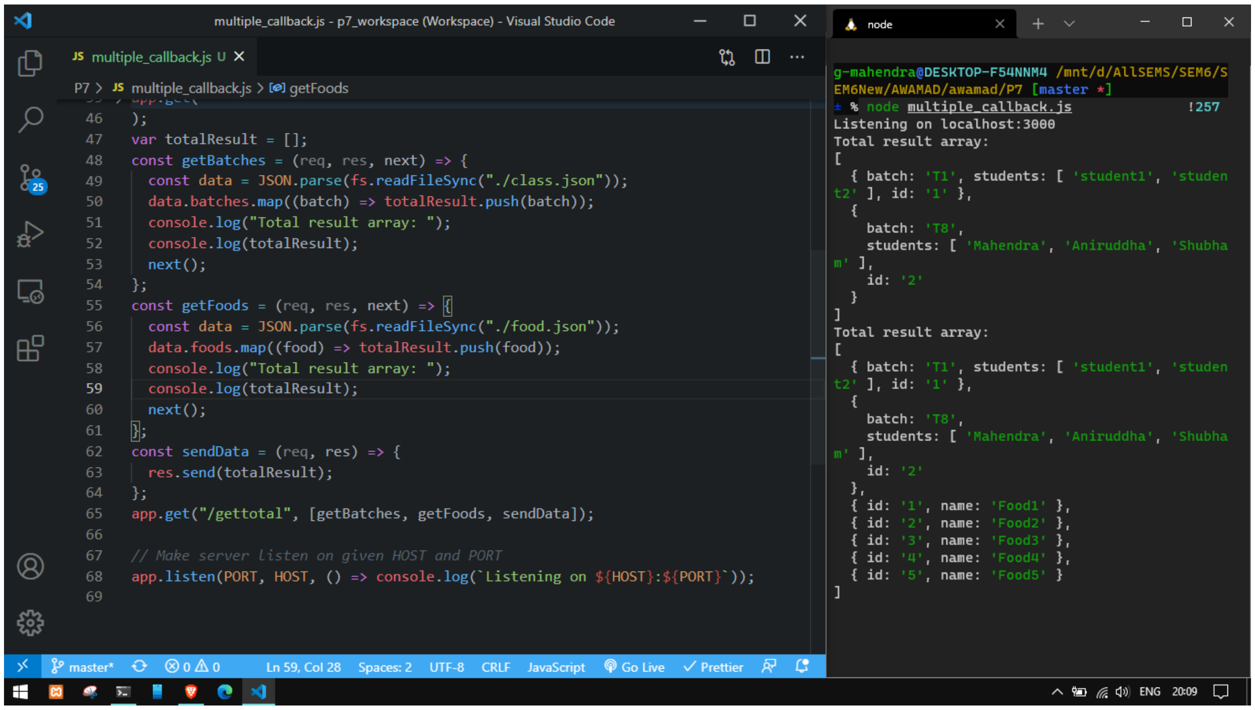
Screenshot 5:



```
[{"id":"1","name":"Food1"},{"id":"2","name":"Food2"},{"id":"3","name":"Food3"},{"id":"4","name":"Food4"},{"id":"5","name":"Food5"}]
```

Information 5: Output of getFoods route

Screenshot 6:

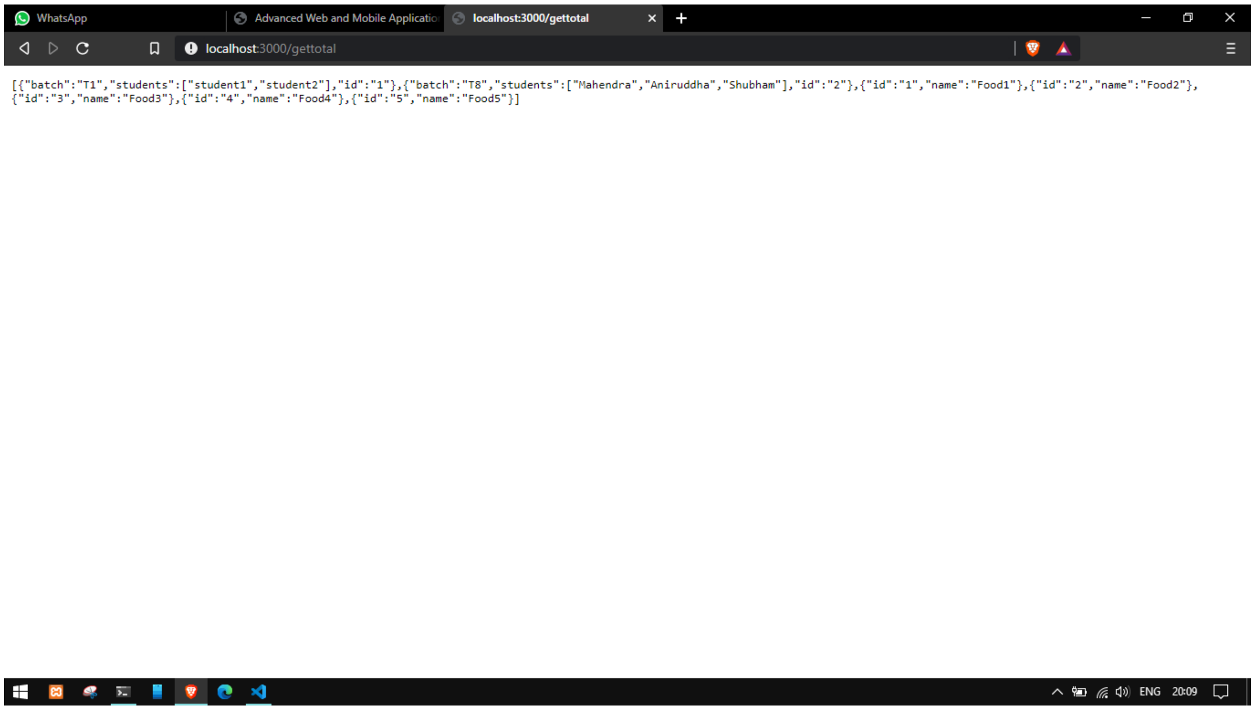


```
);
var totalResult = [];
const getBatches = (req, res, next) => {
  const data = JSON.parse(fs.readFileSync("./class.json"));
  data.batches.map((batch) => totalResult.push(batch));
  console.log("Total result array: ");
  console.log(totalResult);
  next();
};
const getFoods = (req, res, next) => {
  const data = JSON.parse(fs.readFileSync("./food.json"));
  data.foods.map((food) => totalResult.push(food));
  console.log("Total result array: ");
  console.log(totalResult);
  next();
};
const sendData = (req, res) => {
  res.send(totalResult);
};
app.get("/gettotal", [getBatches, getFoods, sendData]);

// Make server listen on given HOST and PORT
app.listen(PORT, HOST, () => console.log(`Listening on ${HOST}:${PORT}`));
```

```
g-mahendra@DESKTOP-F54NNM4 /mnt/d/AllSEMS/SEM6/SEM6New/AWAMAD/awamad/P7 [master *]
± % node multiple_callback.js                                    !257
Listening on localhost:3000
Total result array:
[
  { batch: 'T1', students: [ 'student1', 'student2' ], id: '1' },
  {
    batch: 'T8',
    students: [ 'Mahendra', 'Aniruddha', 'Shubham' ],
    id: '2'
  }
]
Total result array:
[
  { batch: 'T1', students: [ 'student1', 'student2' ], id: '1' },
  {
    batch: 'T8',
    students: [ 'Mahendra', 'Aniruddha', 'Shubham' ],
    id: '2'
  },
  { id: '1', name: 'Food1' },
  { id: '2', name: 'Food2' },
  { id: '3', name: 'Food3' },
  { id: '4', name: 'Food4' },
  { id: '5', name: 'Food5' }
]
```

Information 6: Using three callbacks with array of callbacks

Screenshot 7:



```
[{"batch":"T1","students":["student1","student2"],"id":"1"},{"batch":"T8","students":["Mahendra","Aniruddha","Shubham"],"id":"2"},{"id":"1","name":"Food1"},{"id":"2","name":"Food2"},
{"id":"3","name":"Food3"},{"id":"4","name":"Food4"},{"id":"5","name":"Food5"}]
```

Information 7: output of using three callbacks with array of callbacks