1. Flutter Stateful Widget with get and Post method:

main.dart × pubspec.yaml

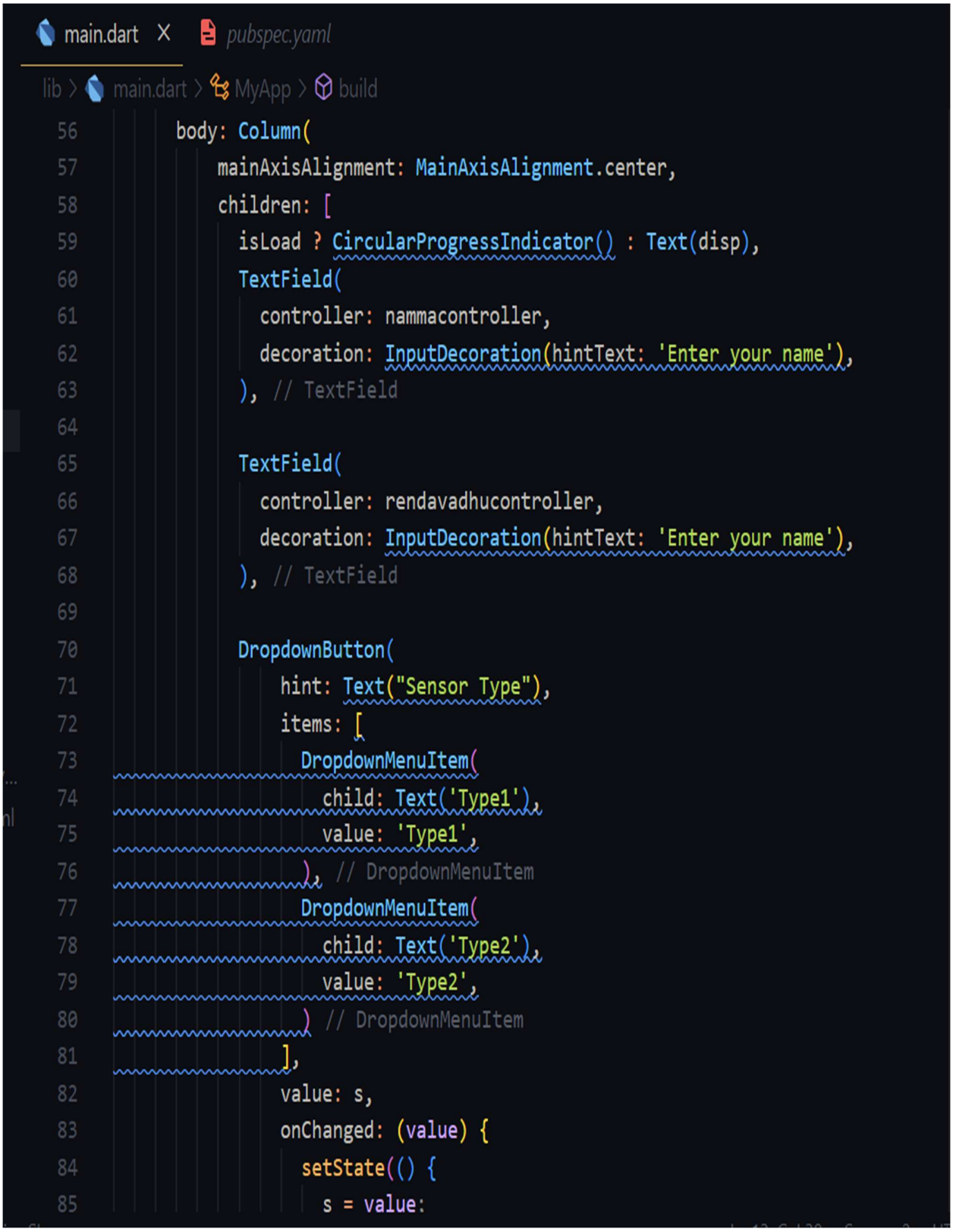
lib > main.dart > MyApp > build

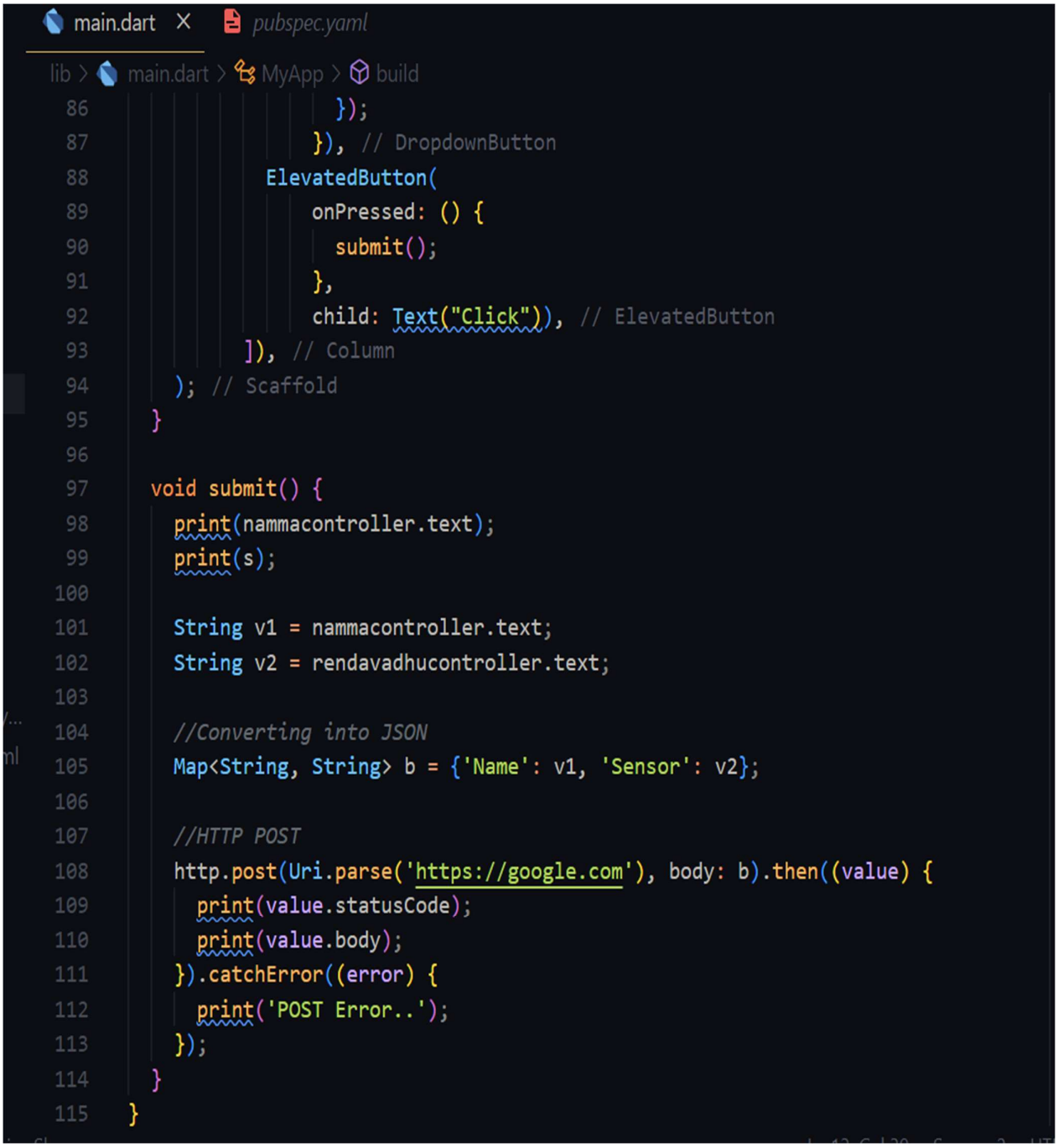
```dart
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;

Run | Debug | Profile
void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
@override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ), // ThemeData
      home: Demo(),
    ); // MaterialApp
  }
}

class Demo extends StatefulWidget {
  const Demo({super.key});

  @override
  State<Demo> createState() => _DemoState();
```

main.dart X pubspec.yaml

lib > main.dart > MyApp > build

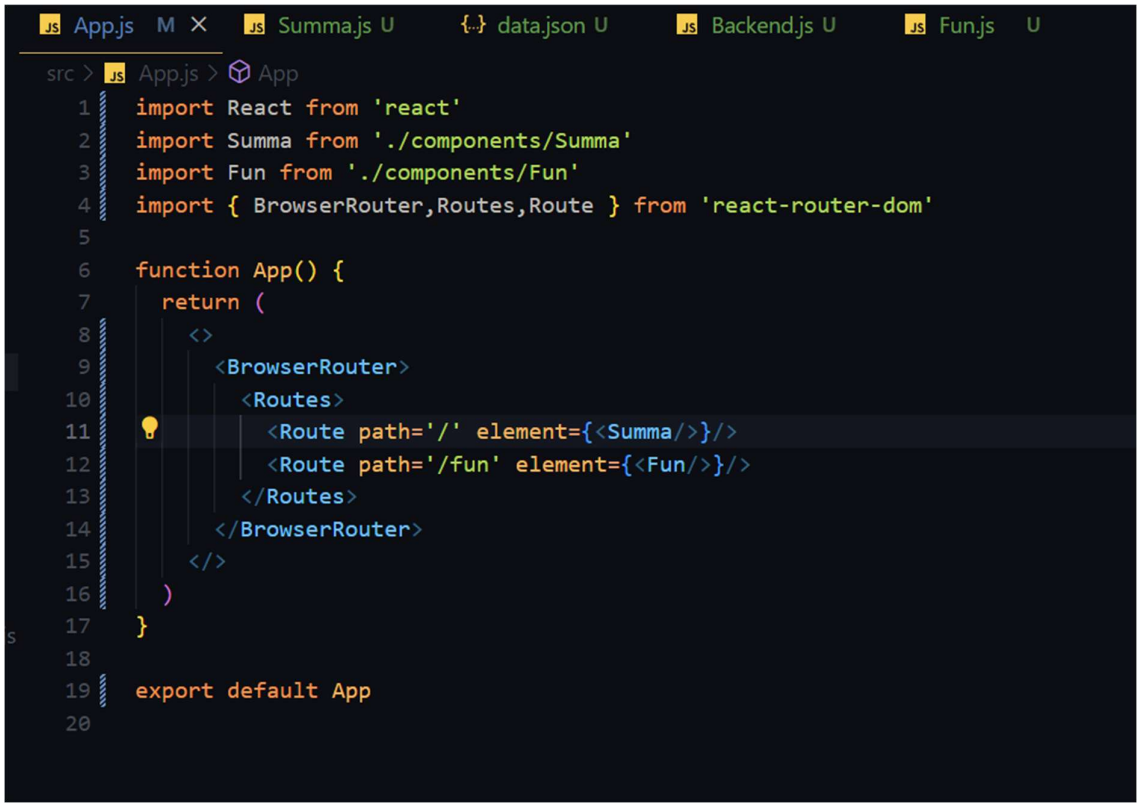
```dart
}

class _DemoState extends State<Demo> {
  bool isLoad = true;
  String disp = "";

  @override
  void initState() {
    http.get(Uri.parse('https://dummyjson.com/products/1')).then((value) {
      setState(() {
        isLoad = false;
        disp = value.body;
      });
    });
    super.initState();
  }

  TextEditingController nammacontroller = TextEditingController();
  TextEditingController rendavadhucontroller = TextEditingController();
  String? s;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Trial'),
      ), // AppBar
```

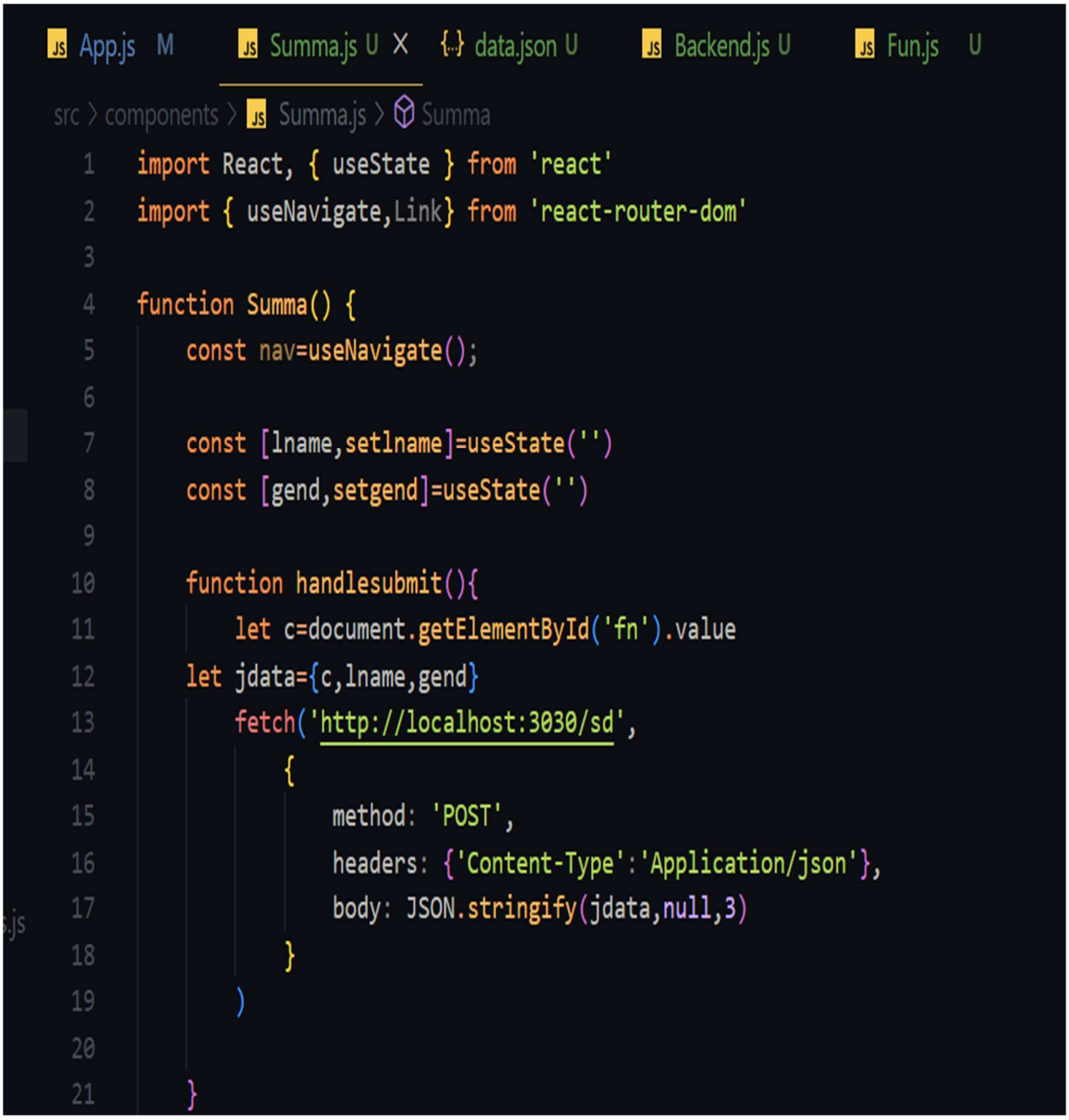

main.dart
pubspec.yaml
lib > main.dart > MyApp > build
body: Column(
mainAxisAlignment: MainAxisAlignment.center,
children: [
isLoad ? CircularProgressIndicator() : Text(disp),
TextField(
controller: nammacontroller,
decoration: InputDecoration(hintText: 'Enter your name'),
), // TextField
TextField(
controller: rendavadhucontroller,
decoration: InputDecoration(hintText: 'Enter your name'),
), // TextField
DropdownButton(
hint: Text("Sensor Type"),
items: [
DropdownMenuItem(
child: Text('Type1'),
value: 'Type1',
), // DropdownMenuItem
DropdownMenuItem(
child: Text('Type2'),
value: 'Type2',
) // DropdownMenuItem
],
value: s,
onChanged: (value) {
setState(() {
s = value:


main.dart
pubspec.yaml
lib > main.dart > MyApp > build
86 });
87 }), // DropdownButton
88 ElevatedButton(
89 onPressed: () {
90 submit();
91 },
92 child: Text("Click")), // ElevatedButton
93]), // Column
94); // Scaffold
95 }
96
97 void submit() {
98 print(nammacontroller.text);
99 print(s);
100
101 String v1 = nammacontroller.text;
102 String v2 = rendavadhucontroller.text;
103
104 //Converting into JSON
105 Map<String, String> b = {'Name': v1, 'Sensor': v2};
106
107 //HTTP POST
108 http.post(Uri.parse('https://google.com'), body: b).then((value) {
109 print(value.statusCode);
110 print(value.body);
111 }).catchError((error) {
112 print('POST Error..');
113 });
114 }
115 }

2. Sending form data from React to Node and Storing it JSON File..

React :




App.js M
Summa.js U X
data.json U
Backend.js U
Fun.js U
src > components > Summa.js > Summa
import React, { useState } from 'react'
import { useNavigate,Link} from 'react-router-dom'
function Summa() {
const nav=useNavigate();
const [lname,setlname]=useState('')
const [gend,setgend]=useState('')
function handlesubmit(){
let c=document.getElementById('fn').value
let jdata={c,lname,gend}
fetch('http://localhost:3030/sd',
{
method: 'POST',
headers: {'Content-Type':'Application/json'},
body: JSON.stringify(jdata,null,3)
}
)
}

App.js M | Summa.js U X | data.json U | Backend.js U | Fun.js U

src > components > Summa.js > Summa

```
23      const handleget=async()=>{
24          let result= await fetch('http://localhost:3030/sd',{method:'GET'})
25          result=await result.text()
26          document.getElementById('trial').innerHTML=result;
27      }
28
29      return (
30        <div>
31            <center>
32                firstname:
33                <input type='text' id='fn' placeholder='Enter your first name'/>
34                <br/>
35                lastname:
36                <input type='text' onBlur={(e)=>setlname(e.target.value)} placeholder='Enter your last name'/>
37                <br/>
38                Gender:
39                <select onChange={(e)=>setgend(e.target.value)}>
40                <option value=''>select</option>
41                    <option value='male'>Male</option>
42                    <option value='female'>Female</option>
43                </select>
44                <br/>
45                <button type='submit' onClick={handlesubmit}>Submit</button>
46            </center>
47            <button onClick={handleget}>Resp</button>
48            <div id='trial'>
49
50            </div>
51        </div>
52      )
```

```
53      }
54      export default Summa;
```

NODE:

App.js M | Summa.js U | data.json U | Backend.js U | Fun.js U

src > Backend.js > server > http.createServer() callback > <function>

```
const http=require('http');
const cors=require('cors');
const fs=require('fs');

const server=http.createServer((req,res)=>{
    cors()(req,res,()=>{

        if(req.method==='POST' && req.url==='/sd'){
            let body="";
            req.on('data',(chunk)=>{
                body+=chunk;
            })
            fs.open('data.json','a',(err,fd)=>{
                if(err){
                    res.writeHead(200)
                    res.end('error')
                }
                else{
                    fs.writeFileSync(fd,body)
                    res.end('Successfully Written to File')
                }
            })
        }

        else if(req.url==='/sd' && req.method==='GET'){
            let body="";
            body=fs.readFileSync('data.json',{encoding:'utf-8'})
            console.log(body);
            res.writeHead(200,{'content-type':'text/plain'})
            res.end(body);
        }
    })
})

server.listen(3030);

```

3. Post and Get form Data from React to Node

React:

App.js M | Summa.js U X | data.json U | Backend.js U | Fun.js U

src > components > Summa.js > Summa

```
1  import React, { useState } from 'react'
2  import { useNavigate,Link} from 'react-router-dom'
3
4  function Summa() {
5      const nav=useNavigate();
6
7      const [lname,setlname]=useState('')
8      const [gend,setgend]=useState('')
9
10     function handlesubmit(){
11         let c=document.getElementById('fn').value
12     let jdata={c,lname,gend}
13         fetch('http://localhost:3030/sd',
14             {
15                 method: 'POST',
16                 headers: {'Content-Type':'Application/json'},
17                 body: JSON.stringify(jdata,null,3)
18             }
19         )
20
21     }
```

App.js M Summa.js U data.json U Backend.js U Fun.js U

src > components > Summa.js > Summa

```
23    const handleget=async()=>{
24        let result= await fetch('http://localhost:3030/sd',{method:'GET'})
25        result=await result.text()
26        document.getElementById('trial').innerHTML=result;
27    }
28
29    return (
30      <div>
31          <center>
32              firstname:
33              <input type='text' id='fn' placeholder='Enter your first name'/>
34              <br/>
35              lastname:
36              <input type='text' onBlur={(e)=>setlname(e.target.value)} placeholder='Enter your last name'/>
37              <br/>
38              Gender:
39              <select onChange={(e)=>setgend(e.target.value)}>
40              <option value=''>select</option>
41                  <option value='male'>Male</option>
42                  <option value='female'>Female</option>
43              </select>
44              <br/>
45              <button type='submit' onClick={handlesubmit}>Submit</button>
46          </center>
47          <button onClick={handleget}>Resp</button>
48          <div id='trial'>
49
50          </div>
51      </div>
52    )
```

```
52        )
53      }
54      export default Summa;
```

NODE:

App.js M Summa.js U data.json U Backend.js U Fun.js U

src > Backend.js > server > http.createServer() callback > <function>

```
const http=require('http');
const cors=require('cors');
const fs=require('fs');
const { log } = require('console');
let body="";
const server=http.createServer((req,res)=>{
    cors()(req,res,()=>{

        if(req.method==='POST' && req.url==='/sd'){
            req.on('data',(chunk)=>{
                body+=chunk;
            })
            console.log('Received..');
        }
```

```
        else if(req.url==='/sd' && req.method==='GET'){
            res.writeHead(200,{'content-type':'text/plain'})
            res.end(body);
        }
    })
})

server.listen(3030);
```

4. **Sending form data from React to MongoDb**

Node:

Backend.jsx M

src > myComponents > Backend.jsx > server > http.createServer() callback > <function> > req.on('end') callback

```
const http = require('http');
const cors = require("cors");
const { MongoClient } = require('mongodb');

const url = "mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.0.2/StudentsDB";

const server = http.createServer((req, res) => {
  cors()(req, res, () => {
    let body = '';

    req.on('data', (chunk) => {
      body += chunk;
    });

    req.on('end', () => {
      try {
        const data = JSON.parse(body);
        let name = data.name;
        let email = data.email;
        let pass = data.password;
```

Backend.jsx M

src > myComponents > Backend.jsx > server > http.createServer() callback > <function> > req.on('end') callback

```
59    if (req.method === 'POST' && req.url === '/Register') {
60        MongoClient.connect(url)
61          .then(client => {
62            const db = client.db("StudentsDB");
63
64            db.collection("StudentsTable").insertOne({ Student_Name: name, EmailID: email, Password: pass })
65          .then(result => {
66            if (result) {
67              console.log("Register Successful");
68              res.writeHead(200,{"content-type" : "text/plain" })
69              res.end("Register Successful");
70
71            } else {
72              console.log("Register Unsuccessful");
73              res.writeHead(401,{"content-type" : "text/plain" })
74              res.end("Register Unsuccessful");
75
76            }
77          })
78          .catch(err => {
79            console.error("Error querying database:", err);
80            res.writeHead(500,{"content-type" : "text/plain" })
81            res.end("Error querying database");
82
83          })
84          .finally(() => {
85            client.close();
86          });
87      })
```

Backend.jsx M

src > myComponents > Backend.jsx > server > http.createServer() callback > <function>

```
88              .catch(err => {
89                console.error("Error connecting to the database:", err);
90                res.writeHead(500,{"content-type" : "text/plain" })
91                res.end("Error connecting to the database");
92
93              });
94              } else {
95                console.log("Error Post");
96                res.writeHead(404, { 'Content-Type': 'text/plain' });
97                res.end("Not Found");
98              }
99          } catch (err) {
100           console.error("Error parsing JSON:", err);
101           res.writeHead(400, { 'Content-Type': 'text/plain' });
102           res.end("Error parsing JSON");
103         }
104       });
105     });
106   });
107
108   server.listen(3002, () => {
109     console.log("Server is running on port 3002");
110   });
111
```

React:


Backend.jsx M
Register.jsx M
src > myComponents > Register.jsx > Register
import React, { useState } from 'react';
import axios from 'axios';
import './Login.css';
//import React from 'react';
function Register(){
const nav=useNavigate();
const [formData, setFormData] = useState({
name:'',
email: '',
password: '',
});
const onChangeHandler = (event) => {
setFormData((prevData) => ({
...prevData,
[event.target.name]: event.target.value,
}));
};
const onSubmitHandler = (event) => {
event.preventDefault()
axios.post("http://localhost:3002/Register",formData).then((response) => {
console.log(response.status, response.data);
alert(response.data);
if (response.data === "Register Successful") {
// Redirect to home page
nav('/login');
}

Backend.jsx M Register.jsx M

src > myComponents > Register.jsx > Register

```
      })
      .catch((error) => {
        console.error('Error:', error.message);
        alert('Error: ' + error.message);
      });
  };
  return (
    <div>
      <center>
        <form onSubmit={(e)=>onSubmitHandler(e)}>
        <div className='gbat'>
            <PersonIcon className='g2'/>
            <input name="name" type='text'  placeholder='Enter your Name'  onChange={onChangeHandler}
                  value={formData.name}
                    required
                  />
        </div>
       <div className='gbat'>
        <AccountCircleIcon className='g2'/>
        <input name="email"  placeholder='Enter your Email'  onChange={onChangeHandler}
                  value={formData.email}
                    required
                  />
        </div>
        <div className='gbat'>
        <KeyIcon className='g2'/>
        <input name='password' type='password'  placeholder='Enter your Password'  onChange={onChangeHandler}
                  value={formData.password}
                    required
                    />
        </div>
        <button >Register</button>
      </form>
        </center>
    </div>

  )
}

export default Register
```

5. **Flutter and Node integration (Sending Data from Mobile app and Receiving in Web App)**

Flutter:

```dart
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
Run | Debug | Profile
void main() {
  runApp(const MyApp());
}
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ), // ThemeData
      home: const Page(),
    ); // MaterialApp
  }
}
class Page extends StatefulWidget {
  const Page({super.key});
  @override
  State<Page> createState() => _PageState();
}
```

lib > main.dart > _PageState > getData

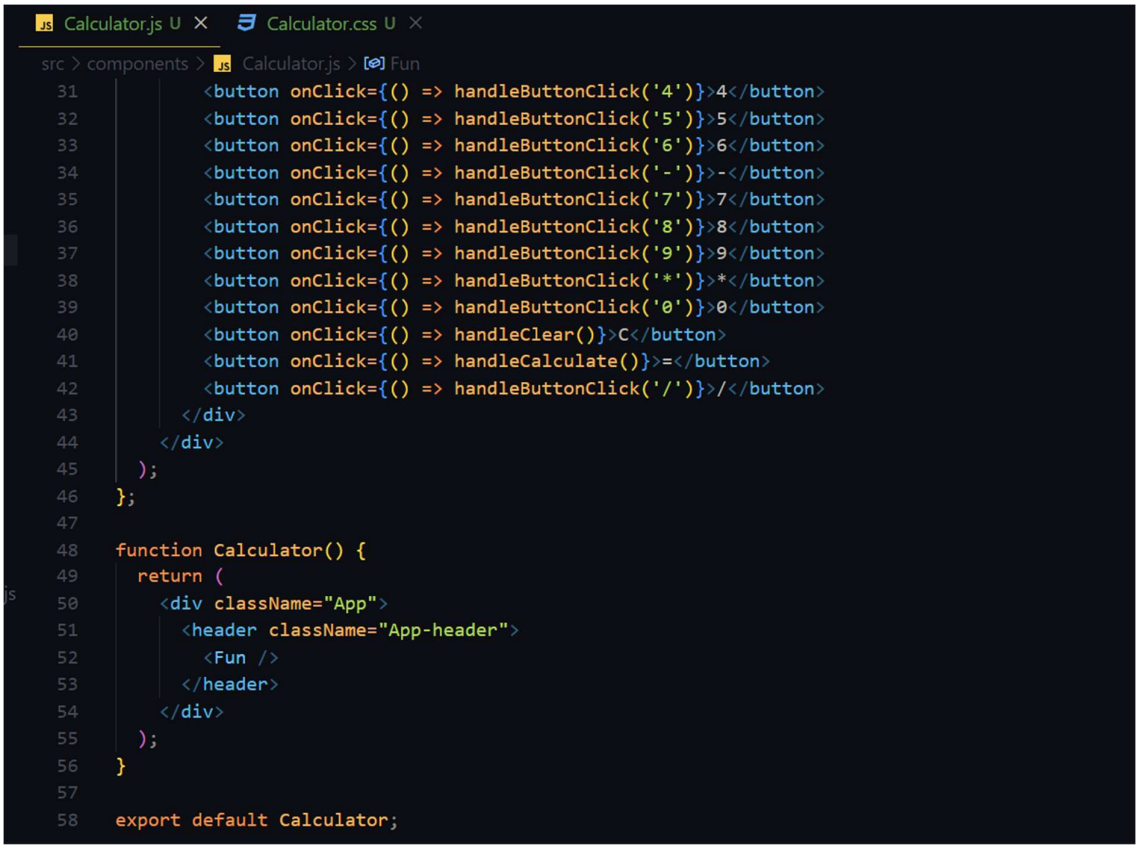
```dart
class _PageState extends State<Page> {
  TextEditingController textControl = TextEditingController();
  String? subName = 'Course';
  String data_from_GET = '';
  bool isGETting = true;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          TextField(
            controller: textControl,
            decoration: const InputDecoration(hintText: 'Your Input Text'),
          ), // TextField
          DropdownButton(
            hint: Text('Enter Subject:'),
            items: const [
              DropdownMenuItem(value: 'SDA', child: Text('SDA')),
              DropdownMenuItem(value: 'IIoT', child: Text('IIoT'))
            ],
            onChanged: (value) => subName = value,
          ), // DropdownButton
          ElevatedButton(onPressed: postData, child: const Text('POST')),
          ElevatedButton(onPressed: getData, child: const Text('GET')),
          isGETting ? const CircularProgressIndicator() : Text(data_from_GET)
        ],
      ), // Column
    ); // Scaffold
  }
  void getData() {
    http.get(Uri.parse('http://192.168.1.10:3030')).then((value) => {
          setState(() {
            isGETting = false;
            data_from_GET = value.body;
          })
        });
  }
  void postData() {
    Map<String, String?> data = {'name': textControl.text, 'course': subName};
    http
        .post(Uri.parse('http://192.168.1.10:3030'), body: data)
        .then((value) => print(value.body));
  }
}

```

Node:


Calculator.js U
Calculator.css U
App.js M
back.js U
src > back.js > server > http.createServer() callback > <function> > req.on('end') callback
const http = require('http')
const cors = require('cors')
let body = ''
const server = http.createServer((req,res)=>{
cors()(req,res,()=>{
if(req.method==='POST'){
req.on('data',(chunk)=>body+=chunk)
req.on('end',()=>{
console.log(body);
res.end(JSON.stringify({message:'Data Received'}))
})
}
else if(req.method==='GET'){
res.writeHead(200,{'Content-Type':'application/json'})
res.end(body)
}
})
})
server.listen(3030,()=>console.log('Server running on port 3030'))
server.on('request',()=>console.log('Request Received'))

6. Calculator Program with React

Calculator.js


Calculator.js U
Calculator.css U
src > components > Calculator.js > Fun
31 <button onClick={() => handleButtonClick('4')}>4</button>
32 <button onClick={() => handleButtonClick('5')}>5</button>
33 <button onClick={() => handleButtonClick('6')}>6</button>
34 <button onClick={() => handleButtonClick('-')}>-</button>
35 <button onClick={() => handleButtonClick('7')}>7</button>
36 <button onClick={() => handleButtonClick('8')}>8</button>
37 <button onClick={() => handleButtonClick('9')}>9</button>
38 <button onClick={() => handleButtonClick('*')}>*</button>
39 <button onClick={() => handleButtonClick('0')}>0</button>
40 <button onClick={() => handleClear()}>C</button>
41 <button onClick={() => handleCalculate()}>=</button>
42 <button onClick={() => handleButtonClick('/')}>/</button>
43
44
45);
46 };
47
48 function Calculator() {
49 return (
50
51 <header className="App-header">
52 <Fun />
53 </header>
54
55);
56 }
57
58 export default Calculator;

Calculator.css

Calculator.js U Calculator.css U

src > components > Calculator.css > button:hover

```css
.calculator {
  width: 300px;
  margin: 50px auto;
  border: 1px solid #ccc;
  border-radius: 5px;
  padding: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

.display {
  font-size: 20px;
  margin-bottom: 10px;
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
  text-align: right;
  height: 28px;
}

.buttons {
  display: grid;
  grid-template-columns: repeat(4, 1fr);
  gap: 10px;
}

button {
  font-size: 18px;
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
  cursor: pointer;
  background-color: #f8f8f8;
}

button:hover {
  background-color: #e0e0e0;
}
```