data.json

```json
{
    "friends": [
        {"id": 1, "name": "Jashanpreet Singh"},
        {"id": 2, "name": "Ridamnoor Kaur"},
        {"id": 3, "name": "Brijesh Das"},
        {"id": 4, "name": "Mohit Kumar"},
        {"id": 5, "name": "Ramesh Apte"},
        {"id": 6, "name": "Kamlesh Sharma"}
    ]
}
```
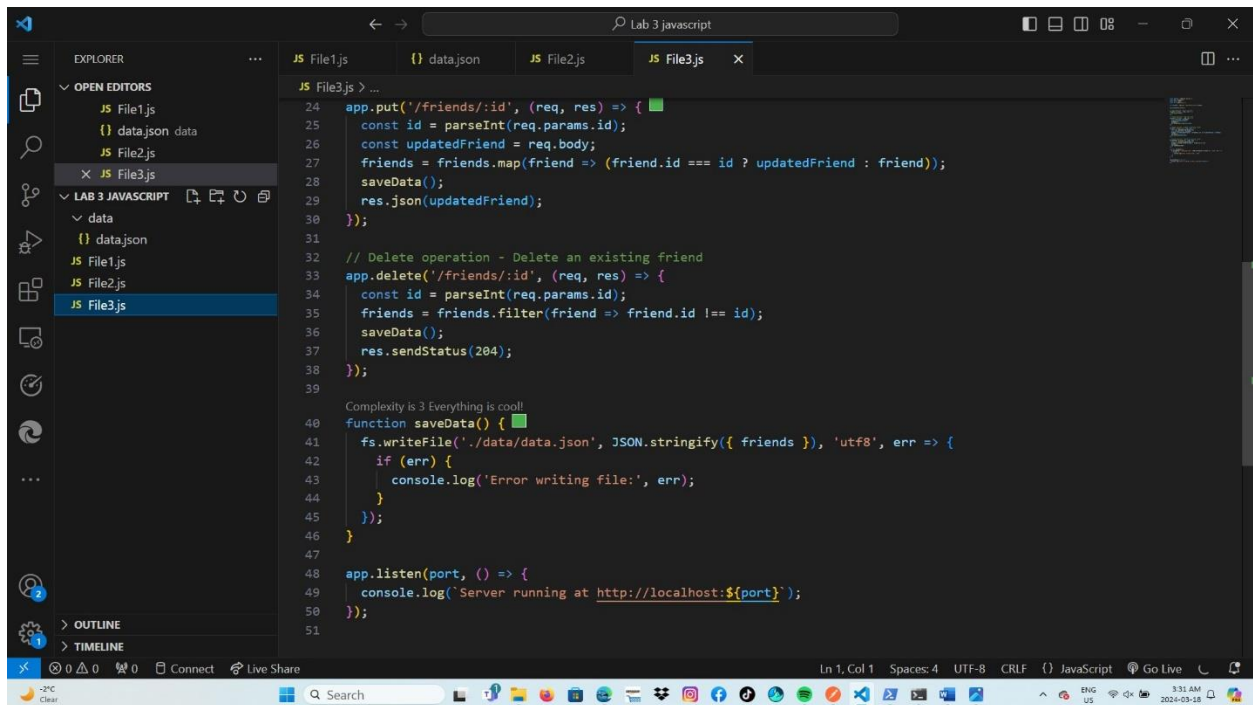
File1.js

```javascript
const express = require('express');
const app = express();
const port = 3001;
const jsonData = require('./data/data.json');

const groupNames = jsonData.friends.map(friend => friend.name);

app.get('/', (req, res) => {
  res.send(`<h1>Group Names:</h1><ul>${groupNames.map(name => `<li>${name}</li>`).join('')}</ul>`);
});

app.listen(port, () => {
  console.log(`Server running at http://localhost:${port}`);
});
```
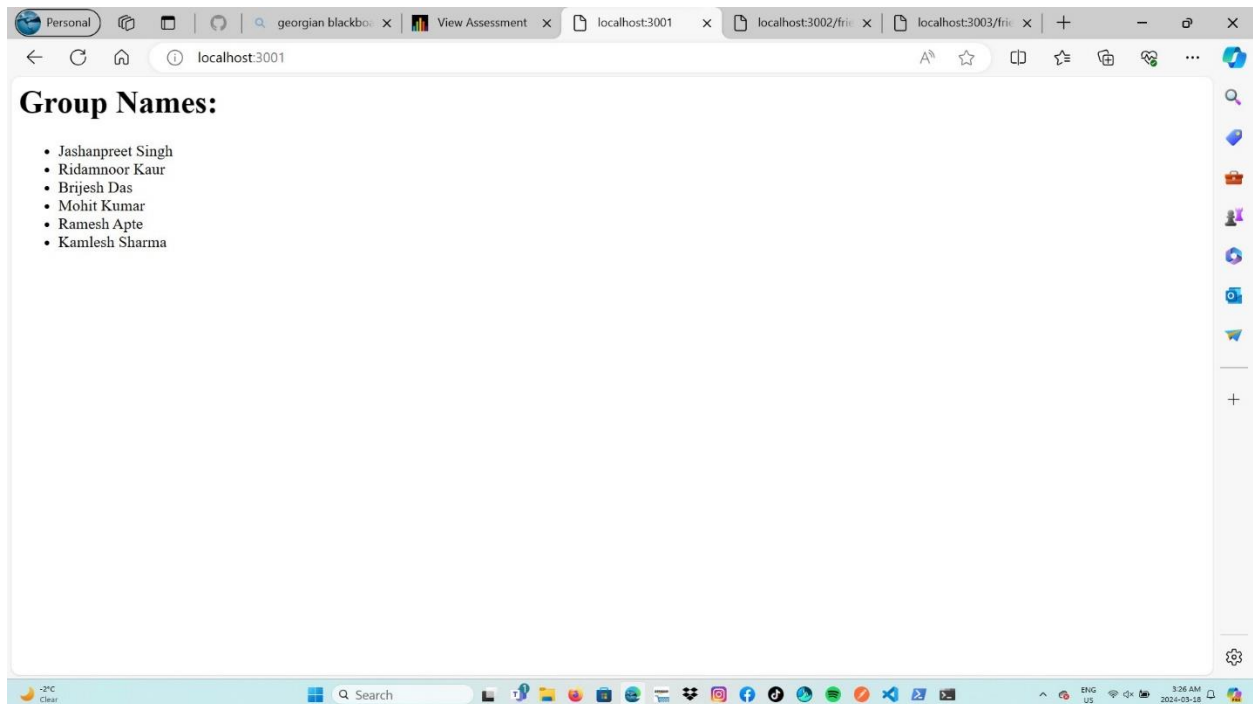
File2.js

```javascript
const express = require('express');
const app = express();
const port = 3002;
const jsonData = require('./data/data.json');

app.get('/friends', (req, res) => {
  res.json(jsonData);
});

app.listen(port, () => {
  console.log(`Server running at http://localhost:${port}`);
});
```

File3.js

```javascript
const express = require('express');
const app = express();
const port = 3003;
const fs = require('fs');

let friends = require('./data/data.json').friends;

app.use(express.json());

// Read operation - Get all friends
app.get('/friends', (req, res) => {
  res.json(friends);
});

// Create operation - Add new friend
app.post('/friends', (req, res) => {
  const newFriend = req.body;
  friends.push(newFriend);
  saveData();
  res.status(201).json(newFriend);
});

// Update operation - Update an existing friend
// Complexity is 3 Everything is cool!
app.put('/friends/:id', (req, res) => {
  const id = parseInt(req.params.id);
  const updatedFriend = req.body;
  friends = friends.map(friend => (friend.id === id ? updatedFriend : friend));
  saveData();
  res.json(updatedFriend);
```

```javascript
24    app.put('/friends/:id', (req, res) => {
25      const id = parseInt(req.params.id);
26      const updatedFriend = req.body;
27      friends = friends.map(friend => (friend.id === id ? updatedFriend : friend));
28      saveData();
29      res.json(updatedFriend);
30    });
31
32    // Delete operation - Delete an existing friend
33    app.delete('/friends/:id', (req, res) => {
34      const id = parseInt(req.params.id);
35      friends = friends.filter(friend => friend.id !== id);
36      saveData();
37      res.sendStatus(204);
38    });
39
      Complexity is 3 Everything is cool!
40    function saveData() {
41      fs.writeFile('./data/data.json', JSON.stringify({ friends }), 'utf8', err => {
42        if (err) {
43          console.log('Error writing file:', err);
44        }
45      });
46    }
47
48    app.listen(port, () => {
49      console.log(`Server running at http://localhost:${port}`);
50    });
51
```


Group Names:

- Jashanpreet Singh
- Ridamnoor Kaur
- Brijesh Das
- Mohit Kumar
- Ramesh Apte
- Kamlesh Sharma

```json
{
    "friends": [
        {
            "id": 1,
            "name": "Jashanpreet Singh"
        },
        {
            "id": 2,
            "name": "Ridamnoor Kaur"
        },
        {
            "id": 3,
            "name": "Brijesh Das"
        },
        {
            "id": 4,
            "name": "Mohit Kumar"
        },
        {
            "id": 5,
            "name": "Ramesh Apte"
        },
        {
            "id": 6,
            "name": "Kamlesh Sharma"
        }
    ]
}
```

```json
[
    {
        "id": 1,
        "name": "Jashanpreet Singh"
    },
    {
        "id": 2,
        "name": "Ridamnoor Kaur"
    },
    {
        "id": 3,
        "name": "Brijesh Das"
    },
    {
        "id": 4,
        "name": "Mohit Kumar"
    },
    {
        "id": 5,
        "name": "Ramesh Apte"
    },
    {
        "id": 6,
        "name": "Kamlesh Sharma"
    }
]
```

Screenshot 1 (POST request):

Home Workspaces ⌄ API Network ⌄   Search Postman   Invite   Upgrade

Learn from experts and join the Postman community at POST/CON 24. Register by March 26 to save 30%.   Learn more ↗

My Workspace   New   Import

Collections
Environments
History

Overview   Getting started   POST http://localhost:3003/:   +   No environment ⌄

HTTP http://localhost:3003/friends   Save ⌄

POST ⌄   http://localhost:3003/friends   Send ⌄

Params Authorization Headers (8) Body • Pre-request Script Tests Settings   Cookies

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON ⌄   Beautify

```
1  {
2    "id": 7,
3    "name": "New Friend"
4  }
5
```

Body Cookies Headers (7) Test Results   201 Created 25 ms 268 B   Save as example •••

Pretty Raw Preview Visualize JSON ⌄

```
1  {
2      "id": 7,
3      "name": "New Friend"
4  }
```

My first collection
First folder inside collection
GET
POST
GET
Second folder inside collection
GET
GET

Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

Create Collection

Screenshot 2 (PUT request):

Home Workspaces ⌄ API Network ⌄   Search Postman   Invite   Upgrade

Learn from experts and join the Postman community at POST/CON 24. Register by March 26 to save 30%.   Learn more ↗

My Workspace   New   Import

Collections
Environments
History

Overview   Getting started   PUT http://localhost:3003/fr   +   No environment ⌄

HTTP http://localhost:3003/friends/7   Save ⌄

PUT ⌄   http://localhost:3003/friends/7   Send ⌄

Params Authorization Headers (8) Body • Pre-request Script Tests Settings   Cookies

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON ⌄   Beautify

```
1  {
2    "id": 7,
3    "name": "Updated Friend"
4  }
5
```

Body Cookies Headers (7) Test Results   200 OK 5 ms 267 B   Save as example •••

Pretty Raw Preview Visualize JSON ⌄

```
1  {
2      "id": 7,
3      "name": "Updated Friend"
4  }
```

My first collection
First folder inside collection
GET
POST
GET
Second folder inside collection
GET
GET

Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

Create Collection

My Workspace   New   Import

Collections

Environments

History

My first collection
  First folder inside collection
    GET
    POST
    GET
  Second folder inside collection
    GET
    GET

Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

Create Collection

Overview   Getting started   DEL http://localhost:3003/fr ●   +

No environment

http://localhost:3003/friends/7   Save   ✏   💬

DELETE   http://localhost:3003/friends/7   Send

Params   Authorization   Headers (8)   Body ●   Pre-request Script   Tests   Settings   Cookies

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL   JSON ∨   Beautify

```
1  {
2  ··"id":·7,
3  ··"name":·"Updated·Friend"
4  }
5
```

Body   Cookies   Headers (5)   Test Results   204 No Content 7 ms 175 B   Save as example

Pretty   Raw   Preview   Visualize   Text ∨

```
1
```

My Workspace   New   Import

Collections

Environments

History

New Collection
  DEL http://localhost:3003/friends/7
  PUT http://localhost:3003/friends/7
  POST http://localhost:3003/friends
  GET http://localhost:3003/friends

Overview   Getting sta   GET http://loca   POST http://loc   PUT http://loca   DEL http://loca   +

No environment

New Collection / **http://localhost:3003/friends**   Save   ✏   💬

GET   http://localhost:3003/friends   Send

Params   Authorization   Headers (6)   Body   Pre-request Script   Tests   Settings   Cookies

Query Params

Key	Value	Description	••• Bulk Edit
Key	Value	Description	

Body   Cookies   Headers (7)   Test Results   200 OK 6 ms 429 B   Save as example

Pretty   Raw   Preview   Visualize   JSON ∨

```
1  [
2      {
3          "id": 1,
4          "name": "Jashanpreet Singh"
5      },
6      {
7          "id": 2,
8          "name": "Ridamnoor Kaur"
9      }
```