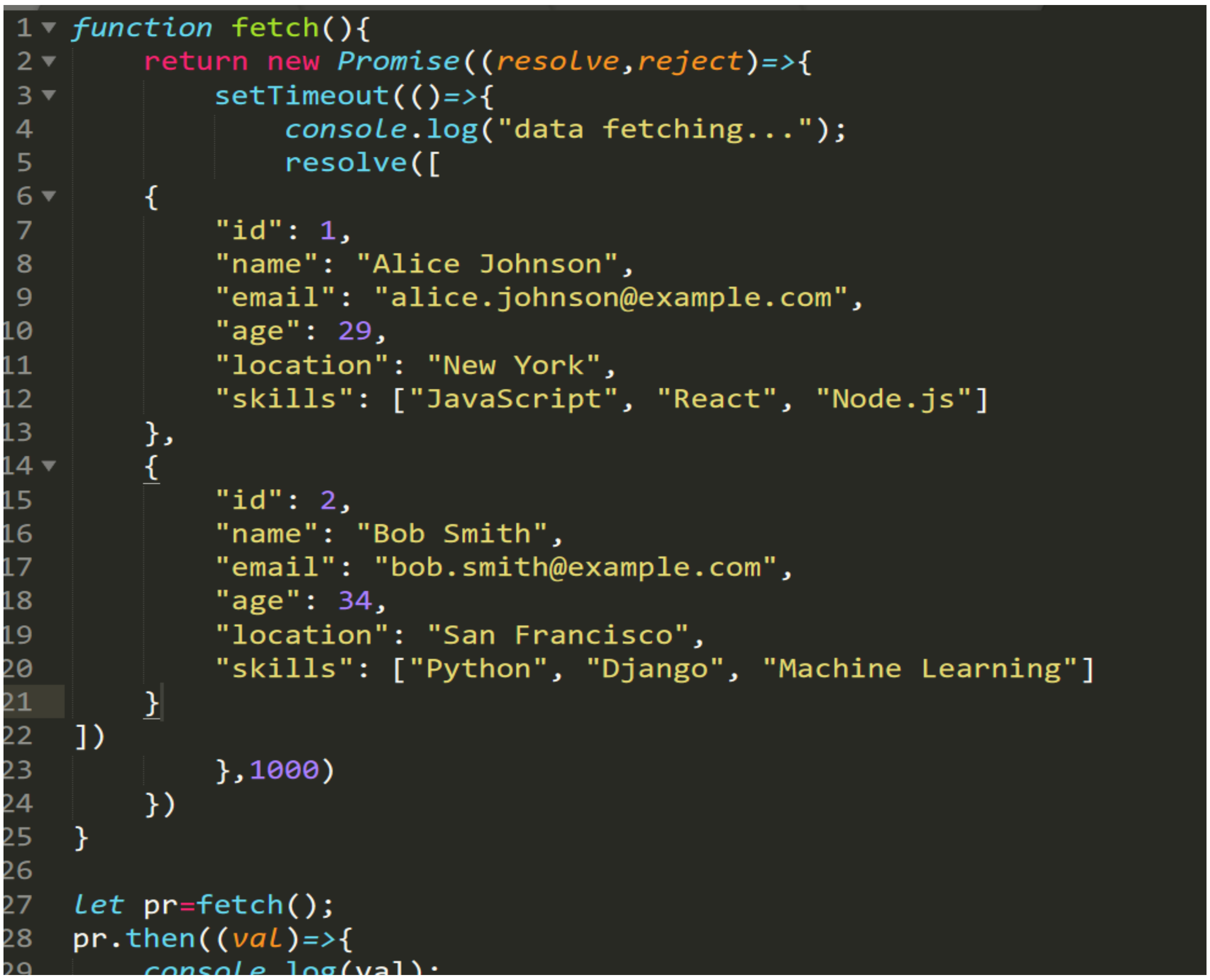
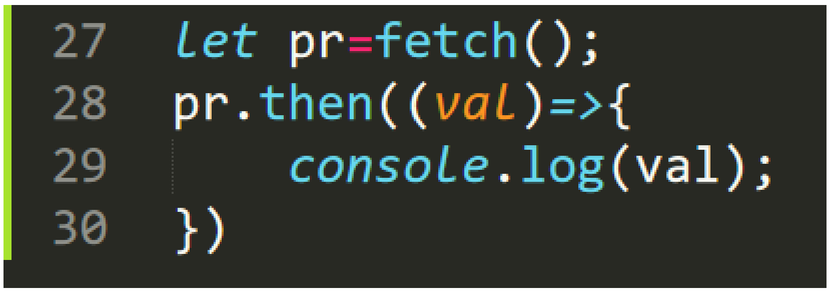
JAVASCRIPT

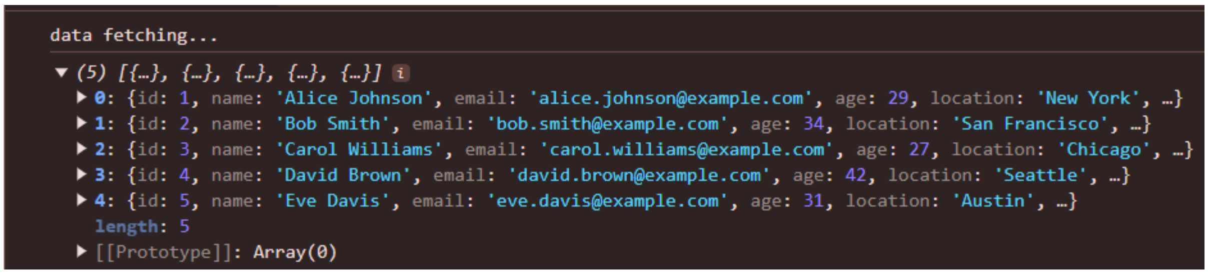
CRUD :



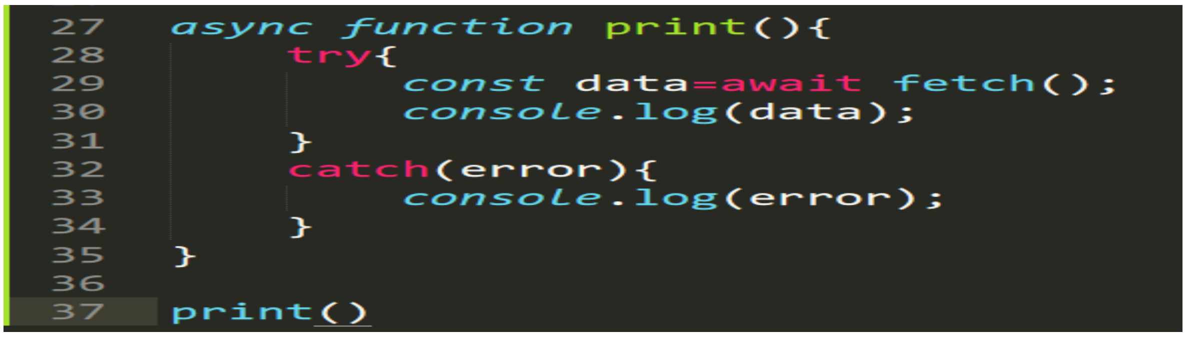
```
function fetch(){
    return new Promise((resolve,reject)=>{
        setTimeout(()=>{
            console.log("data fetching...");
            resolve([
    {
        "id": 1,
        "name": "Alice Johnson",
        "email": "alice.johnson@example.com",
        "age": 29,
        "location": "New York",
        "skills": ["JavaScript", "React", "Node.js"]
    },
    {
        "id": 2,
        "name": "Bob Smith",
        "email": "bob.smith@example.com",
        "age": 34,
        "location": "San Francisco",
        "skills": ["Python", "Django", "Machine Learning"]
    }
])
        },1000)
    })
}

let pr=fetch();
pr.then((val)=>{
```



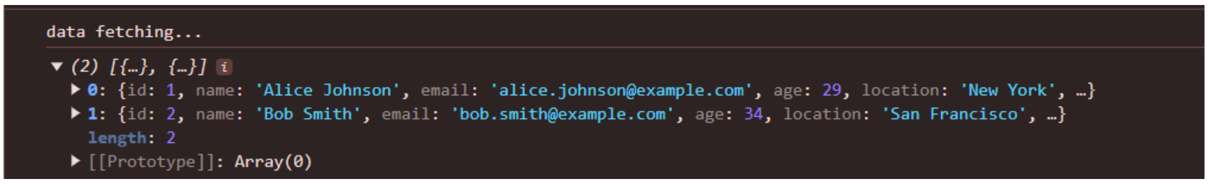
```
let pr=fetch();
pr.then((val)=>{
    console.log(val);
})
```



```
data fetching...
▼ (5) [{…}, {…}, {…}, {…}, {…}]
  ▶ 0: {id: 1, name: 'Alice Johnson', email: 'alice.johnson@example.com', age: 29, location: 'New York', …}
  ▶ 1: {id: 2, name: 'Bob Smith', email: 'bob.smith@example.com', age: 34, location: 'San Francisco', …}
  ▶ 2: {id: 3, name: 'Carol Williams', email: 'carol.williams@example.com', age: 27, location: 'Chicago', …}
  ▶ 3: {id: 4, name: 'David Brown', email: 'david.brown@example.com', age: 42, location: 'Seattle', …}
  ▶ 4: {id: 5, name: 'Eve Davis', email: 'eve.davis@example.com', age: 31, location: 'Austin', …}
    length: 5
  ▶ [[Prototype]]: Array(0)
```

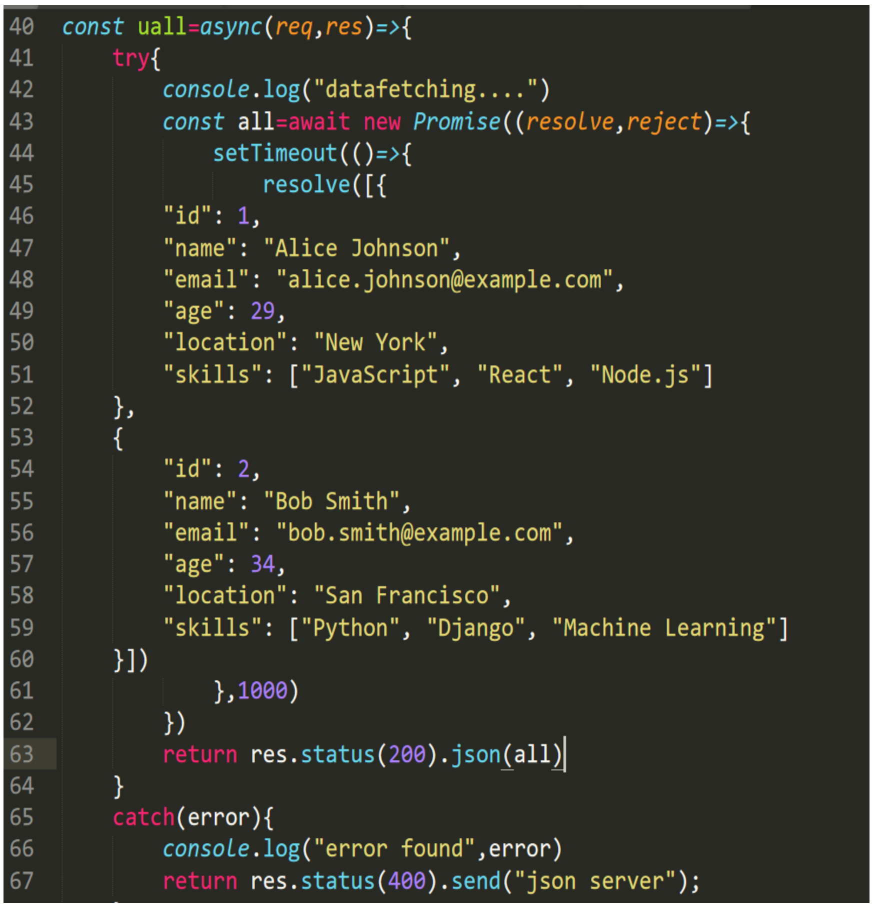


```
async function print(){
        try{
            const data=await fetch();
            console.log(data);
        }
        catch(error){
            console.log(error);
        }
}

print()
```

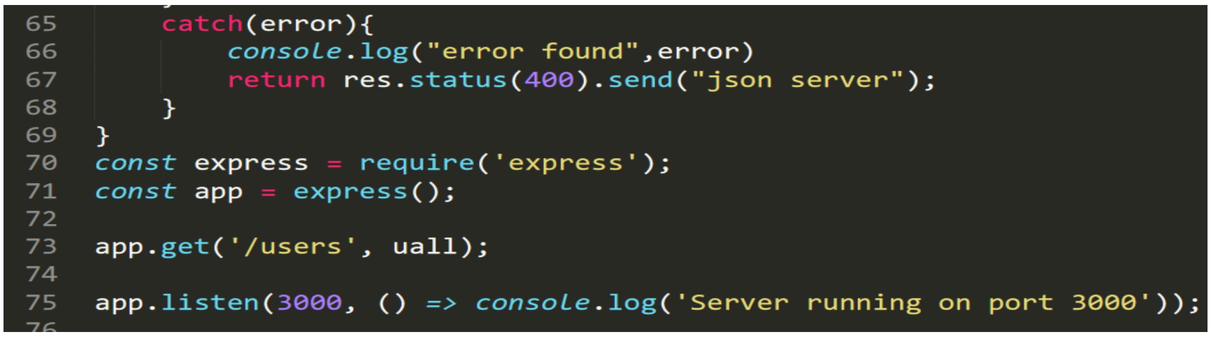


```
data fetching...
▼ (2) [{…}, {…}] i
  ▶ 0: {id: 1, name: 'Alice Johnson', email: 'alice.johnson@example.com', age: 29, location: 'New York', …}
  ▶ 1: {id: 2, name: 'Bob Smith', email: 'bob.smith@example.com', age: 34, location: 'San Francisco', …}
    length: 2
  ▶ [[Prototype]]: Array(0)
```

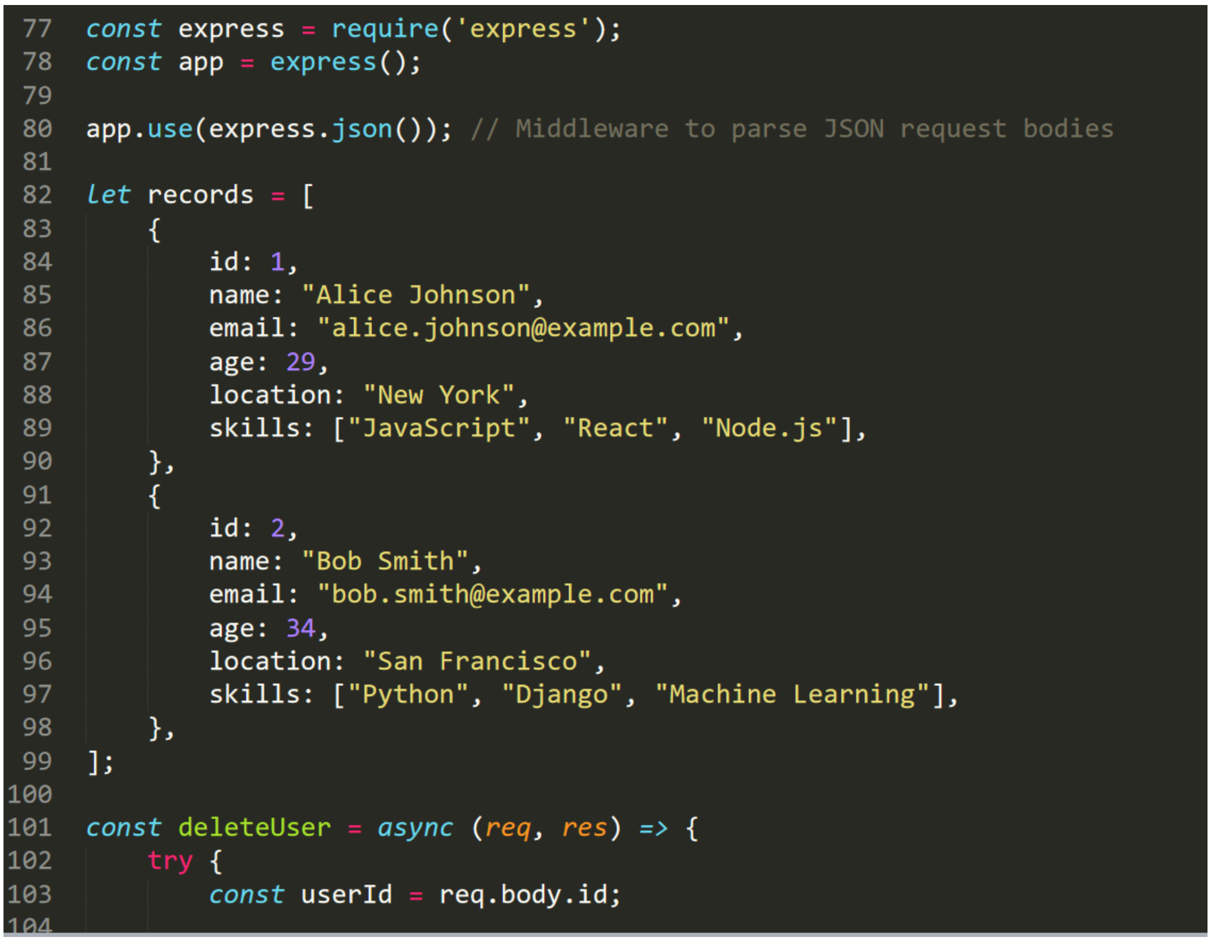
```
const uall = async (req, res) => {

  try {

    console.log("Fetching all users...");

    const all = await user.find(); // Add a breakpoint or log here

    console.log("Fetched users:", all);

    return res.status(200).json(all);

  } catch (error) {

    console.error("Error in uall:", error);

    return res.status(400).send("Server error...");

  }

};
```



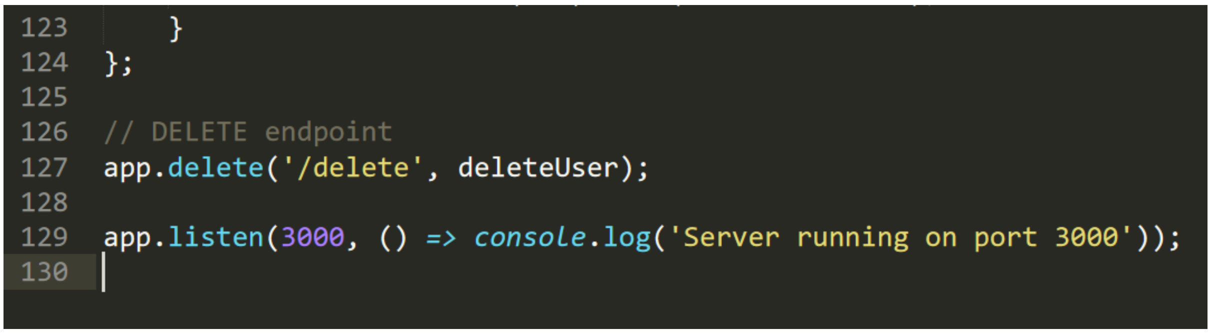
```
const uall=async(req,res)=>{
    try{
        console.log("datafetching....")
        const all=await new Promise((resolve,reject)=>{
            setTimeout(()=>{
                resolve([{
        "id": 1,
        "name": "Alice Johnson",
        "email": "alice.johnson@example.com",
        "age": 29,
        "location": "New York",
        "skills": ["JavaScript", "React", "Node.js"]
    },
    {
        "id": 2,
        "name": "Bob Smith",
        "email": "bob.smith@example.com",
        "age": 34,
        "location": "San Francisco",
        "skills": ["Python", "Django", "Machine Learning"]
    }])
            },1000)
        })
        return res.status(200).json(all)
    }
    catch(error){
        console.log("error found",error)
        return res.status(400).send("json server");
```



```
    catch(error){
        console.log("error found",error)
        return res.status(400).send("json server");
    }
}
const express = require('express');
const app = express();

app.get('/users', uall);

app.listen(3000, () => console.log('Server running on port 3000'));
```



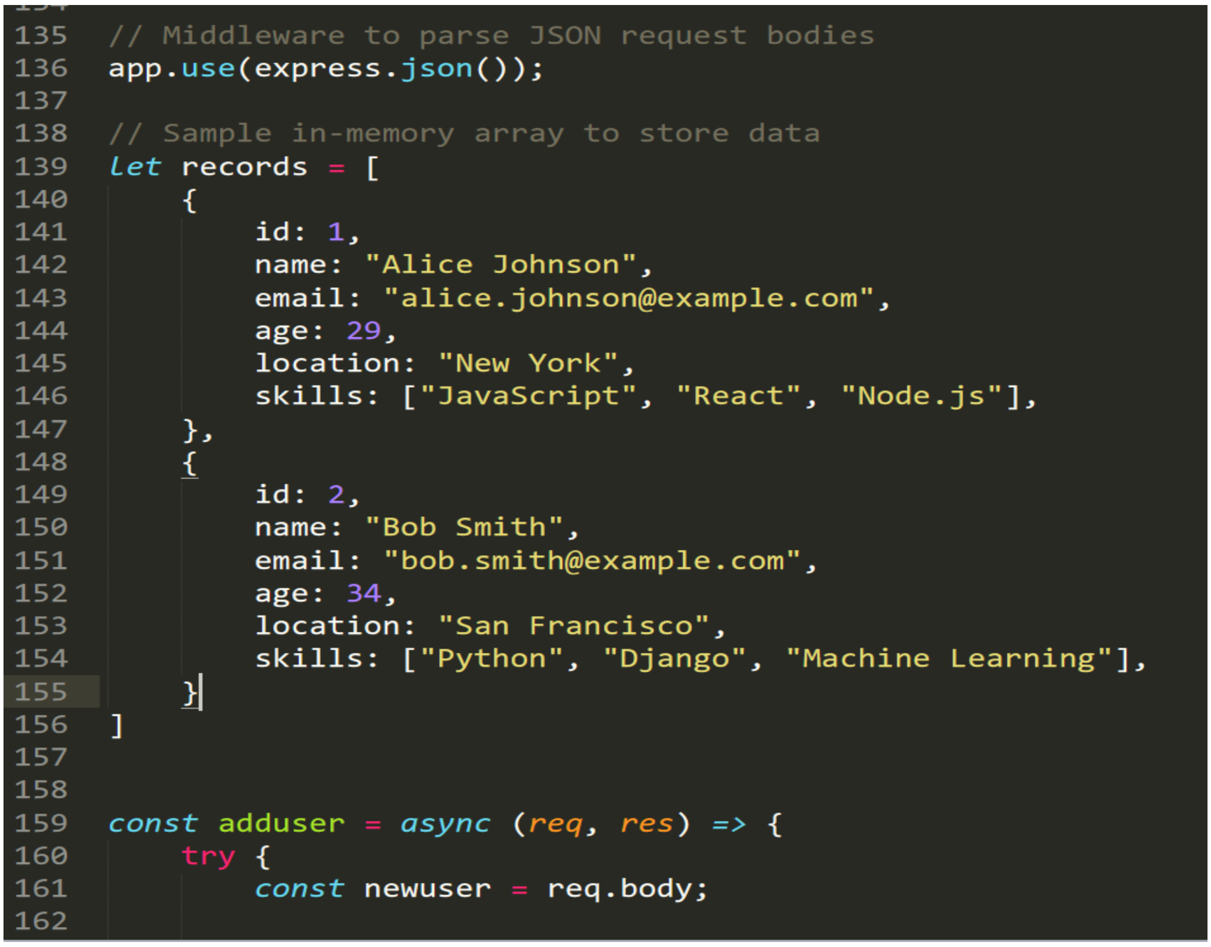
```javascript
const express = require('express');
const app = express();

app.use(express.json()); // Middleware to parse JSON request bodies

let records = [
    {
        id: 1,
        name: "Alice Johnson",
        email: "alice.johnson@example.com",
        age: 29,
        location: "New York",
        skills: ["JavaScript", "React", "Node.js"],
    },
    {
        id: 2,
        name: "Bob Smith",
        email: "bob.smith@example.com",
        age: 34,
        location: "San Francisco",
        skills: ["Python", "Django", "Machine Learning"],
    },
];

const deleteUser = async (req, res) => {
    try {
        const userId = req.body.id;
```



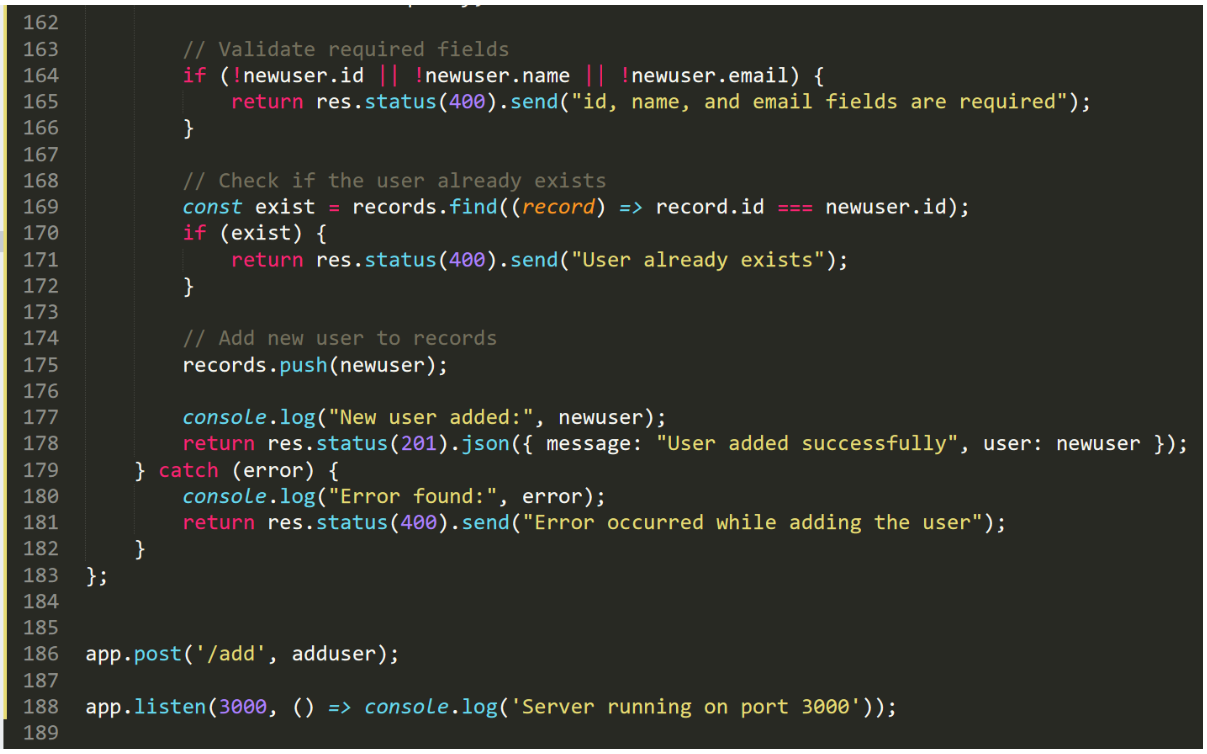
```javascript
    }
};

// DELETE endpoint
app.delete('/delete', deleteUser);

app.listen(3000, () => console.log('Server running on port 3000'));

```

```
const deleteUser = async (req, res) => {

  try {

    const userId = req.body.id;



    if (!userId) {

      return res.status(400).send("User ID is required.");

    }
```

```
        // Find and delete the user
        const result = await User.findOneAndDelete({ id: userId });

        if (!result) {
            return res.status(404).send("User not found.");
        }

        console.log(`User with ID ${userId} deleted.`);
        return res.status(200).json({ message: "User deleted successfully", user: result });
    } catch (error) {
        console.error("Error found:", error);
        return res.status(500).send("Server error.");
    }
};

// DELETE Route
app.delete('/delete', deleteUser);

app.listen(3000, () => console.log('Server running on port 3000'));
```

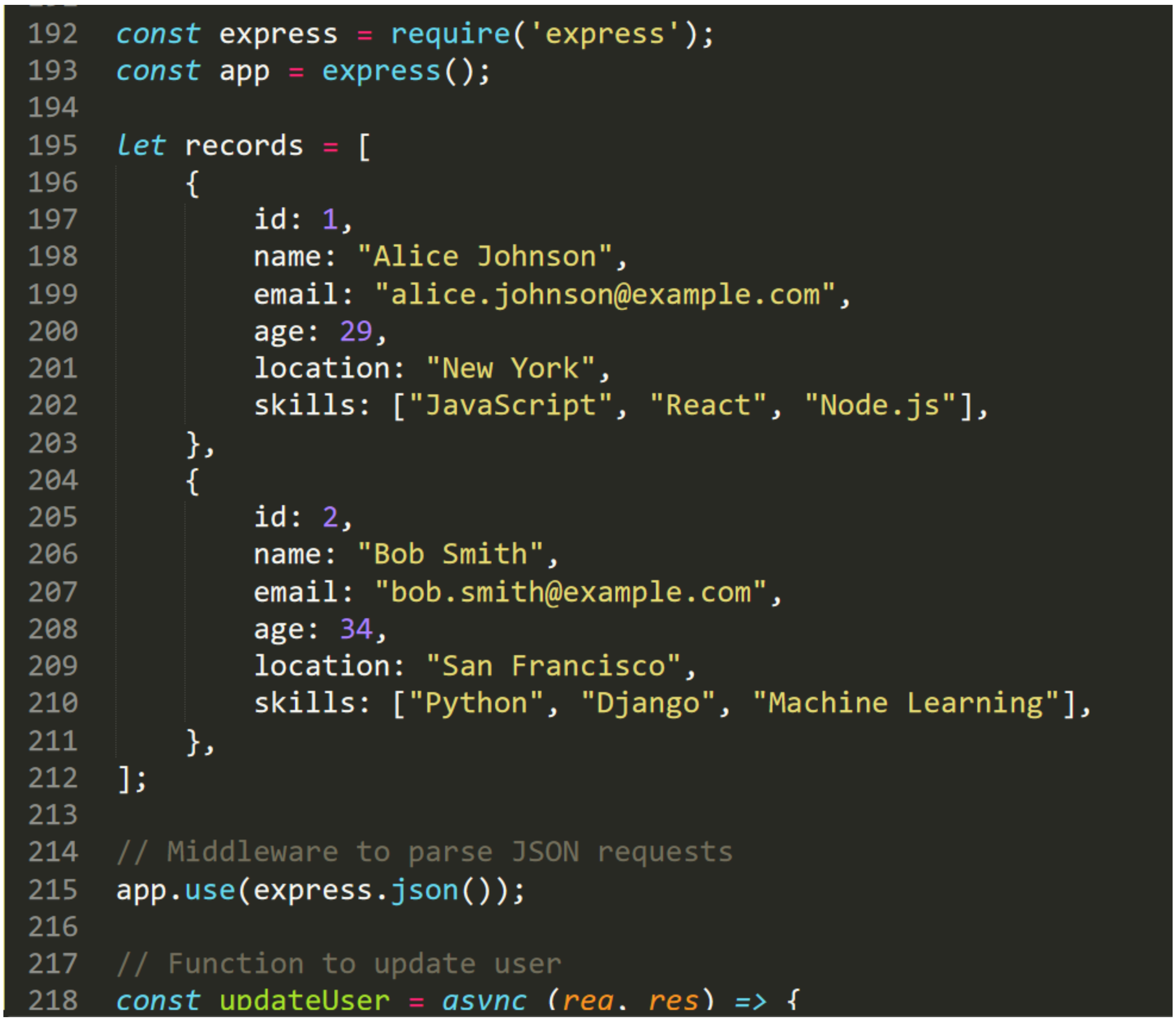


```javascript
// Middleware to parse JSON request bodies
app.use(express.json());

// Sample in-memory array to store data
let records = [
    {
        id: 1,
        name: "Alice Johnson",
        email: "alice.johnson@example.com",
        age: 29,
        location: "New York",
        skills: ["JavaScript", "React", "Node.js"],
    },
    {
        id: 2,
        name: "Bob Smith",
        email: "bob.smith@example.com",
        age: 34,
        location: "San Francisco",
        skills: ["Python", "Django", "Machine Learning"],
    }
]


const adduser = async (req, res) => {
    try {
        const newuser = req.body;

```

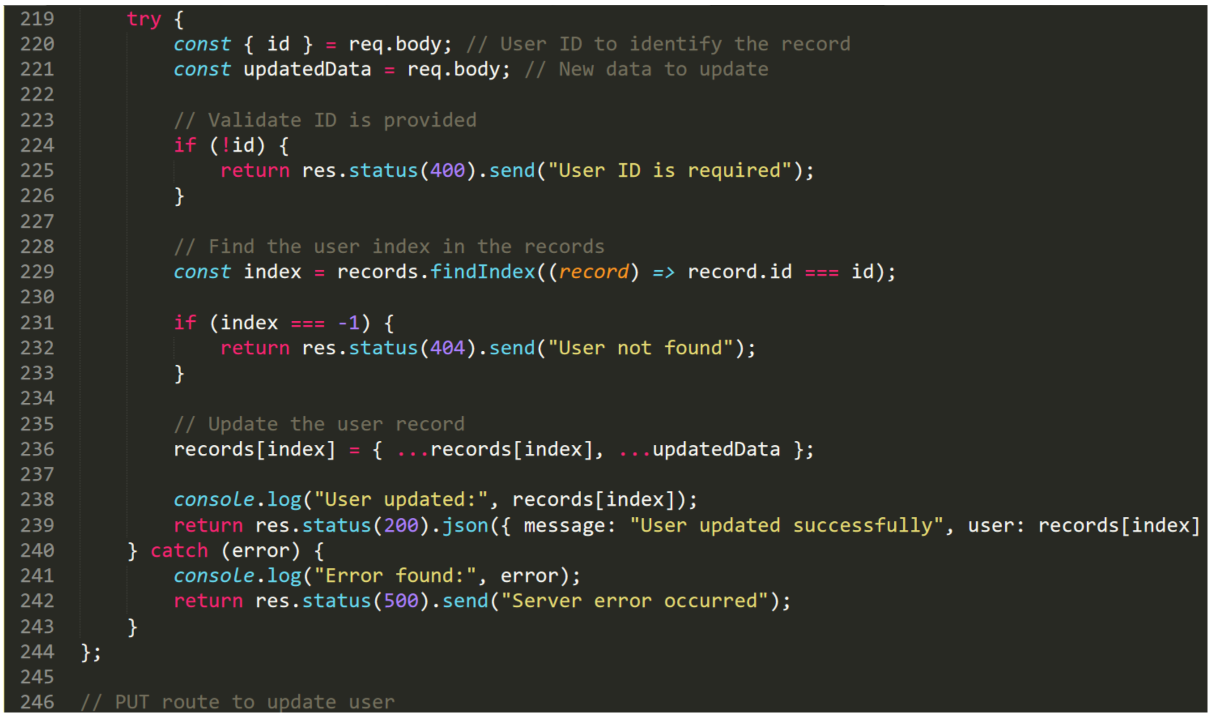


```javascript

        // Validate required fields
        if (!newuser.id || !newuser.name || !newuser.email) {
          return res.status(400).send("id, name, and email fields are required");
        }

        // Check if the user already exists
        const exist = records.find((record) => record.id === newuser.id);
        if (exist) {
          return res.status(400).send("User already exists");
        }

        // Add new user to records
        records.push(newuser);

        console.log("New user added:", newuser);
        return res.status(201).json({ message: "User added successfully", user: newuser });
    } catch (error) {
        console.log("Error found:", error);
        return res.status(400).send("Error occurred while adding the user");
    }
};


app.post('/add', adduser);

app.listen(3000, () => console.log('Server running on port 3000'));

```

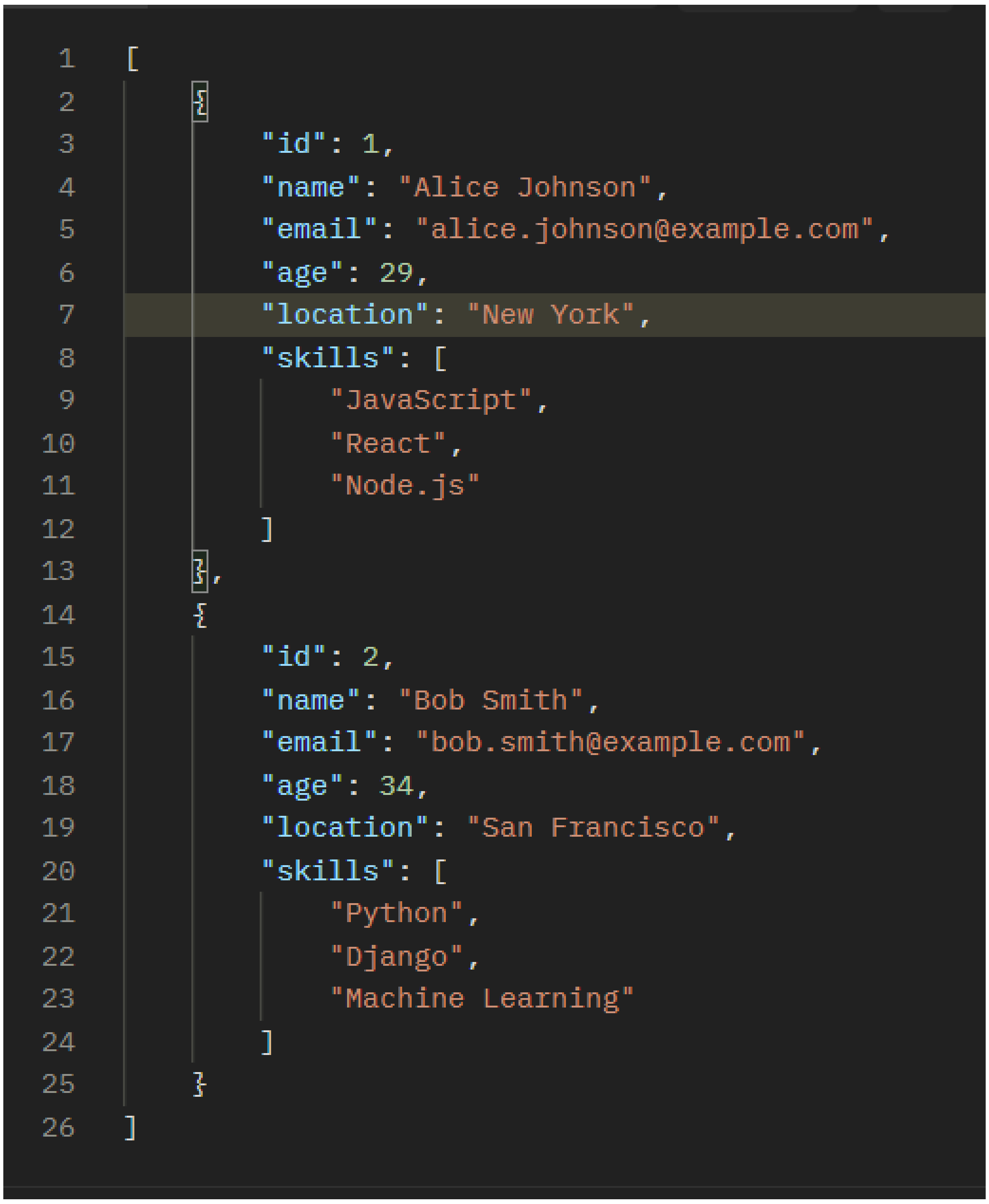
```
const adduser = async (req, res) => {
  try {
    const newUser = req.body;

    // Validate required fields
    if (!newUser.id || !newUser.name || !newUser.email) {
      return res.status(400).send('id, name, and email fields are required');
    }

    // Check if the user already exists
    const exist = await User.findOne({ id: newUser.id });
    if (exist) {
      return res.status(400).send('User already exists');
    }

    // Save new user to the database
    const user = new User(newUser);
    await user.save();

    console.log('New user added:', newUser);
    return res.status(201).json({ message: 'User added successfully', user: newUser });
  } catch (error) {
    console.log('Error found:', error);
    return res.status(500).send('Error occurred while adding the user');
  }
};
```



```javascript
const express = require('express');
const app = express();

let records = [
    {
        id: 1,
        name: "Alice Johnson",
        email: "alice.johnson@example.com",
        age: 29,
        location: "New York",
        skills: ["JavaScript", "React", "Node.js"],
    },
    {
        id: 2,
        name: "Bob Smith",
        email: "bob.smith@example.com",
        age: 34,
        location: "San Francisco",
        skills: ["Python", "Django", "Machine Learning"],
    },
];

// Middleware to parse JSON requests
app.use(express.json());

// Function to update user
const updateUser = async (req, res) => {
```



```javascript
    try {
        const { id } = req.body; // User ID to identify the record
        const updatedData = req.body; // New data to update

        // Validate ID is provided
        if (!id) {
            return res.status(400).send("User ID is required");
        }

        // Find the user index in the records
        const index = records.findIndex((record) => record.id === id);

        if (index === -1) {
            return res.status(404).send("User not found");
        }

        // Update the user record
        records[index] = { ...records[index], ...updatedData };

        console.log("User updated:", records[index]);
        return res.status(200).json({ message: "User updated successfully", user: records[index]
    } catch (error) {
        console.log("Error found:", error);
        return res.status(500).send("Server error occurred");
    }
};

// PUT route to update user
const updatedUser = await User.findOneAndUpdate(
      { id }, // Filter to find the user
      updateFields, // Fields to update
      { new: true } // Return the updated user
    );

    if (!updatedUser) {
      return res.status(404).send("User not found.");
    }

    return res.status(200).json({
      message: "User updated successfully.",
      user: updatedUser,
    });
  } catch (error) {
    console.error("Error updating user:", error);
    return res.status(500).send("Server error.");
  }
};
```



```json
[
    {
        "id": 1,
        "name": "Alice Johnson",
        "email": "alice.johnson@example.com",
        "age": 29,
        "location": "New York",
        "skills": [
            "JavaScript",
            "React",
            "Node.js"
        ]
    },
    {
        "id": 2,
        "name": "Bob Smith",
        "email": "bob.smith@example.com",
        "age": 34,
        "location": "San Francisco",
        "skills": [
            "Python",
            "Django",
            "Machine Learning"
        ]
    }
]
```