Assignment

Learner Details

- **Name: Nikhil k**
- **Enrollment Number: SU625MR004**
- **Batch / Class: June 2025 MERN**
- **Assignment: Contact Form Assignment**
- **Date of Submission: 20/08/2025**

Problem Solving Activity

1. Program Statement

Create a Contact Form backend that accepts form submissions (via POST request) and saves the data to a JSON file. Also, provide a GET endpoint to fetch all stored contacts. Test the API using **POSTMAN**.

---

2. Algorithm

- Start the program and import required modules (express, fs, dotenv).
- Initialize an Express app and configure it to accept JSON data.
- Define a POST route /contact to handle new contact form submissions:
- Extract name, email, phonenumber, education, address from request body.
- Validate inputs: if any field is missing, return an error response.
- Create a new contact object with the form data and current date.
- Save the data into contact.json file.
- Send a success response with saved data.
- Define a GET route /contact to fetch stored contacts:
- Read data from contact.json.
- Parse JSON and send it as a response.
- If file not found or invalid, return an error response.
- Start the server on the defined port.

- Test the API using **POSTMAN** (send POST and GET requests).

---

3. Pseudocode

```
BEGIN
  IMPORT express, fs, dotenv
  INITIALIZE app
  SET port from environment or 5000
  ENABLE JSON parsing in app

  DEFINE POST "/contact":
    EXTRACT name, email, phonenumber, education, address FROM req.body
    IF any field missing:
        RETURN error response
    ELSE:
        CREATE newdata object with all fields + current date
        CONVERT newdata to JSON string
        WRITE JSON data to "contact.json"
        RETURN success response with saved data

  DEFINE GET "/contact":
    READ "contact.json"
    IF error:
        RETURN error response
    ELSE:
        PARSE JSON and RETURN data

  START server on port
```

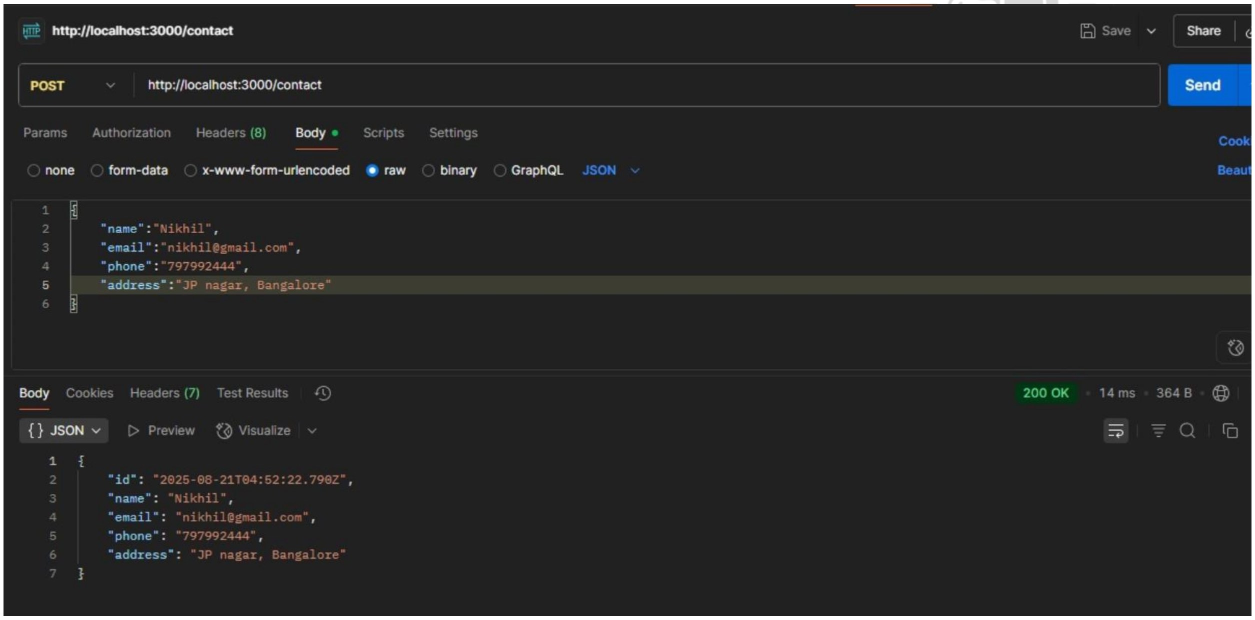
END

4. Program Code

```
var express = require("express");
const app = express();

const fs = require("fs");

require("dotenv").config();

const port = process.env.PORT || 5000;

app.use(express.json());

app.post("/contact", (req, res) => {
  const { name, email, phonenumber, education, address } = req.body;

  if (!name || !email || !phonenumber || !education || !address) {
    return res.json({ error: "data not found" });
  }

  let submission = [];
  const newdata = {
    name,
    email,
    phonenumber,
    education,
    address,
    date: new Date(),
  };

  submission.push(newdata);
  const jsonData = JSON.stringify(newdata, null, 2);

  fs.writeFile("contact.json", jsonData, (err) => {
    if (err) {
      return res.json({ error: "Not found" });
    }
    return res.json(jsonData);
  });
});

app.listen(port, () => {
  console.log("server started");
});
```

```
var express = require("express");
const app = express();

const fs = require("fs");
require("dotenv").config();

const port = process.env.PORT || 5000;

app.use(express.json());

app.get("/contact", (req, res) => {
  fs.readFile("contact.json", "utf8", (err, data) => {
    if (err) {
      return res.json({ error: "File not found" });
    }
    try {
      const contacts = JSON.parse(data);
      return res.json(contacts);
    } catch (e) {
      return res.json({ error: "Invalid JSON data" });
    }
  });
});

app.listen(port, () => {
  console.log("server started");
});
```

5. Screenshots of Output

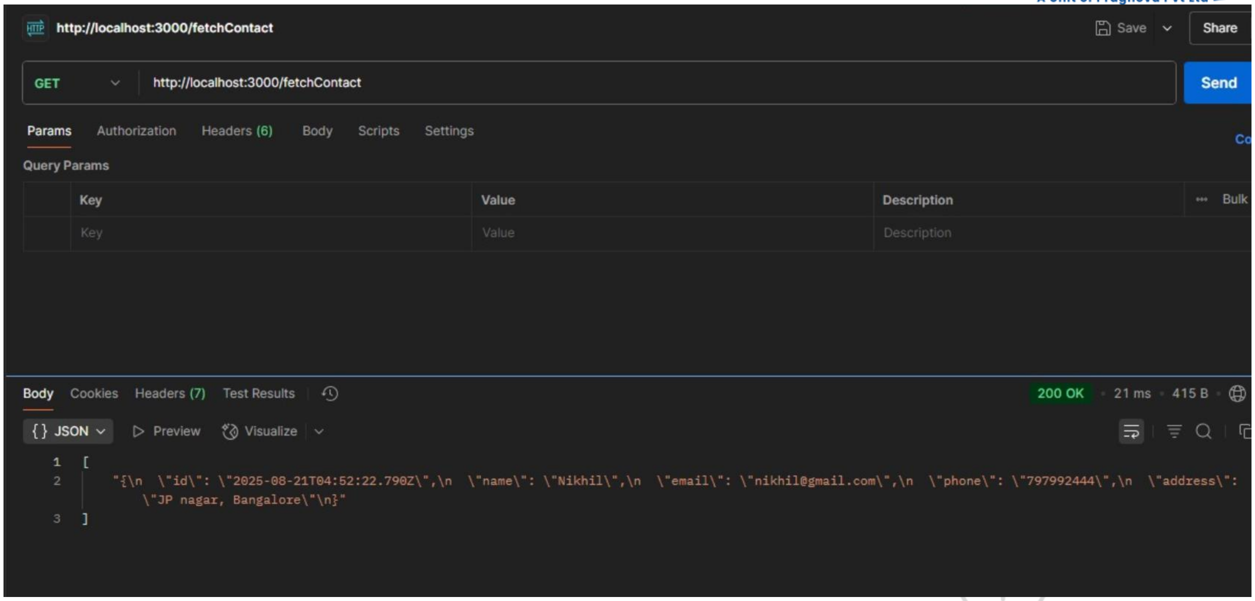


POST http://localhost:3000/contact

```
{
    "name":"Nikhil",
    "email":"nikhil@gmail.com",
    "phone":"797992444",
    "address":"JP nagar, Bangalore"
}
```

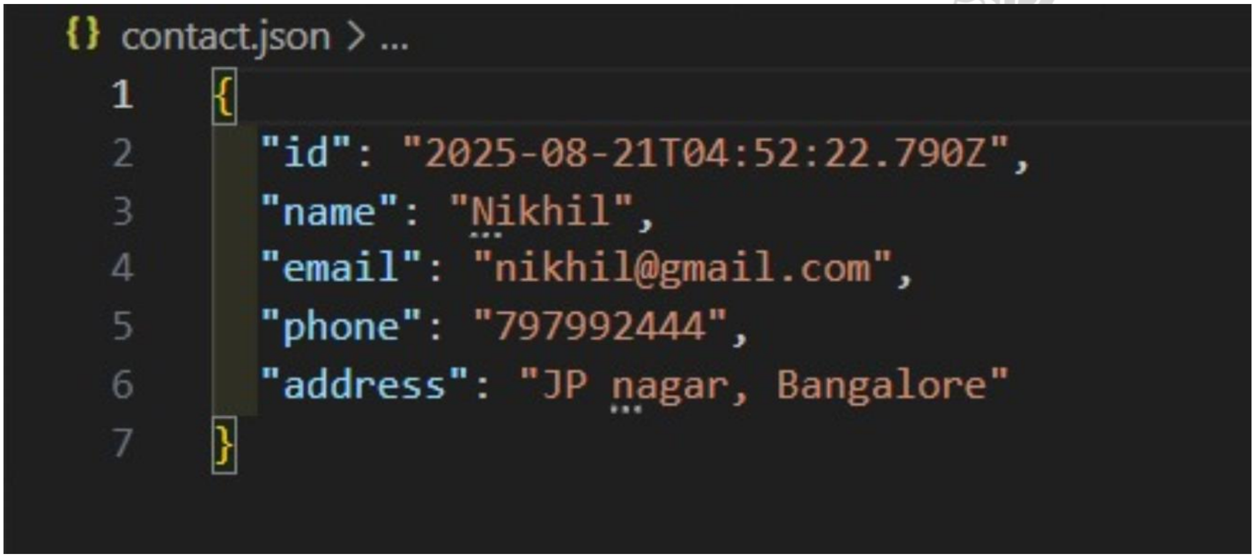
200 OK · 14 ms · 364 B

```
{
    "id": "2025-08-21T04:52:22.790Z",
    "name": "Nikhil",
    "email": "nikhil@gmail.com",
    "phone": "797992444",
    "address": "JP nagar, Bangalore"
}
```



GET http://localhost:3000/fetchContact

```
[
    "{\n  \"id\": \"2025-08-21T04:52:22.790Z\",\n  \"name\": \"Nikhil\",\n  \"email\": \"nikhil@gmail.com\",\n  \"phone\": \"797992444\",\n  \"address\":
      \"JP nagar, Bangalore\"\n}"
]
```



```json
{
  "id": "2025-08-21T04:52:22.790Z",
  "name": "Nikhil",
  "email": "nikhil@gmail.com",
  "phone": "797992444",
  "address": "JP nagar, Bangalore"
}
```

6. Observation / Reflection

I faced issues with JSON handling and testing in Postman but learned how to create REST APIs using Express. Next time, I would add validations, support multiple entries, and use MongoDB for storage.