

Assignment

Learner Details

- **Name:** Afreen Ahmed
 - **Enrollment Number:** SU625MR003
 - **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);
```



Stemup
A Unit of Pragnova Pvt Ltd

```
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;
```

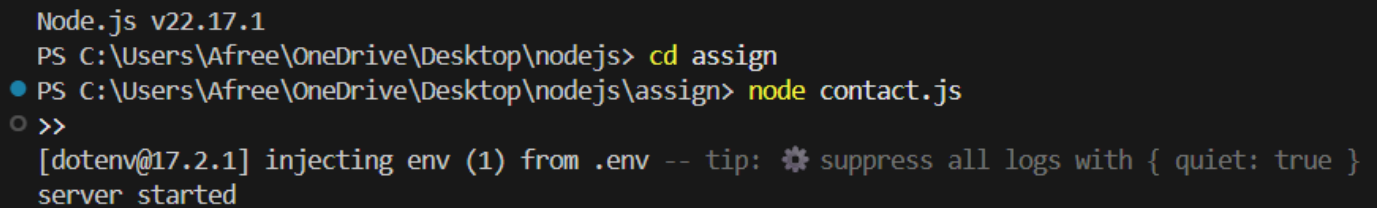
```
app.use(express.json());
```

```
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



```
Node.js v22.17.1  
PS C:\Users\Afree\OneDrive\Desktop\nodejs> cd assign  
PS C:\Users\Afree\OneDrive\Desktop\nodejs\assign> node contact.js  
>>  
[dotenv@17.2.1] injecting env (1) from .env -- tip: ⚙ suppress all logs with { quiet: true }  
server started
```

Home Workspaces API Network Search Postman Ctrl K

Afreen Ahmed's Workspace New Import

POST http://localhost:3000/ GET http://localhost:3000/ PUT http://localhost:3000/ POST http://localhost:4000/ GET http://localhost:4000/ No environment

http://localhost:4000/contact Save Share

POST http://localhost:4000/contact Send

Params Authorization Headers (8) Body Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☒ JSON Beautify

```

1 {
2   "name": "afreen",
3   "email": "afreen@gmail.com",
4   "phoneNumber": 1234567899,
5   "education": "BE",
6   "address": "Tiptur"
7 }

```

Body Cookies Headers (7) Test Results 200 OK 60 ms 431 B

JSON Preview Visualize

```

1 {
  "name": "afreen",
  "email": "afreen@gmail.com",
  "phoneNumber": 1234567899,
  "education": "BE",
  "address": "Tiptur",
  "date": "2025-08-21T03:22:31.818Z"
}

```

Online Find and replace Console Postbot Runner Start Proxy Cookies Vault Trash

view Go Run Terminal Help

... .env fsandexpress JS server.js JS contact.js {} contact.json X .env assign {} login.js

assign > {} contact.json > ...

```

1 {
2   "name": "afreen",
3   "email": "afreen@gmail.com",
4   "phoneNumber": 1234567890,
5   "education": "BE",
6   "address": "Tiptur",
7   "date": "2025-08-21T03:22:31.818Z"
8 }

```

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.

