Learner Assignment Submission Format

Learner Details

- **Name:H O Akash**
- **Enrollment Number:SU625MR**
- **Batch / Class:MERN Stack**
- **Assignment:Express JS**
- **Date of Submission:20/08/2025**

Problem Solving Activity 1.1

1. Program Statement

Create a RESTful API using Node.js and Express.js that allows users to create, read, and update contact information. The contact information should be stored in a JSON file named cont.json.

---

2. Algorithm

1. Initialize the Express.js app and set up the port.
2. Use the express.json() middleware to parse JSON requests.
3. Define a route for creating new contact information using the PUT /cont endpoint.
4. In the PUT /cont endpoint:
   - Extract the name, number, address, and email from the request body.
   - Validate that all fields are present.
   - Create a new contact object and add it to the submission array.
   - Convert the new contact object to JSON and append it to the cont.json file.
5. Define a route for retrieving all contact information using the GET /cont endpoint.
6. In the GET /cont endpoint:
   - Read the cont.json file and parse its contents as JSON.
   - Return the parsed JSON data in the response.

3. Pseudocode

INITIALIZE Express.js app

SET port = process.env.PORT || 5000

USE express.json() middleware

DEFINE route PUT /cont

  EXTRACT name, number, address, email from request body

  VALIDATE that all fields are present

  CREATE new contact object

  APPEND new contact object to cont.json file

  RETURN new contact object in response

DEFINE route GET /cont

  READ cont.json file

  PARSE file contents as JSON

  RETURN parsed JSON data in response

LISTEN on port

4. Program Code

```
var express=require('express');

const app=express();

const fs=require('fs');

var environ=require('dotenv').config();
```

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

app.use(express.json())

// app.post('/cont',(req,res)=>{

//     const{name,number,address,email}=req.body;

//     if(!name||!number||!address||!email){

//         return res.json({"error":"not found"})

//     }

//     let submission=[];

//     const newData={name,number,address,email};

//     submission.push(newData);

//     const jsonData=JSON.stringify(newData,null,2);


//     fs.writeFile('cont.json',jsonData,(err)=>{

//          if(err){

//             return res.json({"error":"not found"})

//          }

//          return res.json(jsonData);


//     })

// })

app.put('/cont',(req,res)=>{

    const{name,number,address,email}=req.body;

    if(!name||!number||!address||!email){

        return res.json({"error":"not found"})

    }

    let submission=[];
```

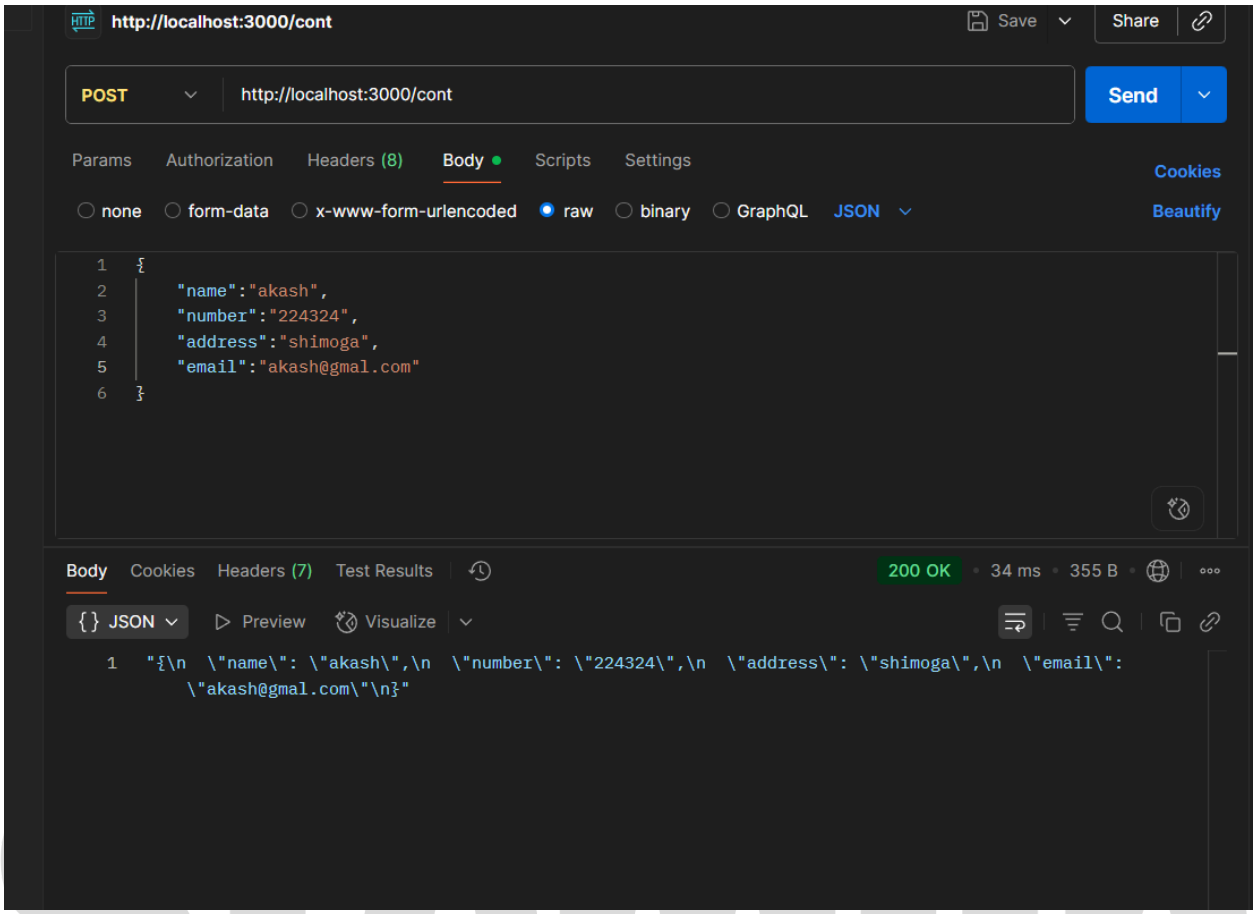
```
    const newData={name,number,address,email};
    submission.push(newData);
    const jsonData=JSON.stringify(newData,null,2);

    fs.appendFile('cont.json',jsonData,(err)=>{
         if(err){
            return res.json({"error":"not found"})
         }
         return res.json(jsonData);

    })
})
app.get('/cont', (req, res) => {

   fs.readFile('cont.json','utf-8',(err,data)=>{
         if(err){
            console.log('error');

         }
         const nw=JSON.parse(data)
          res.send(nw);;

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

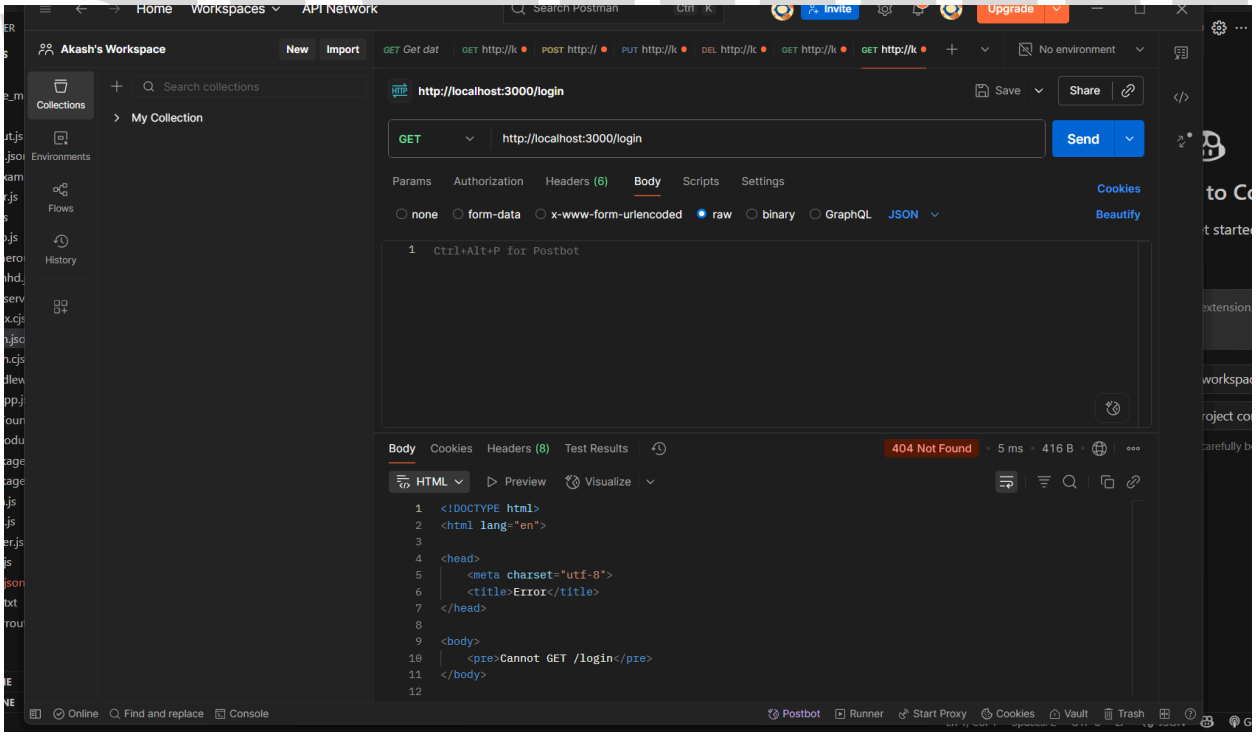
```
})
```

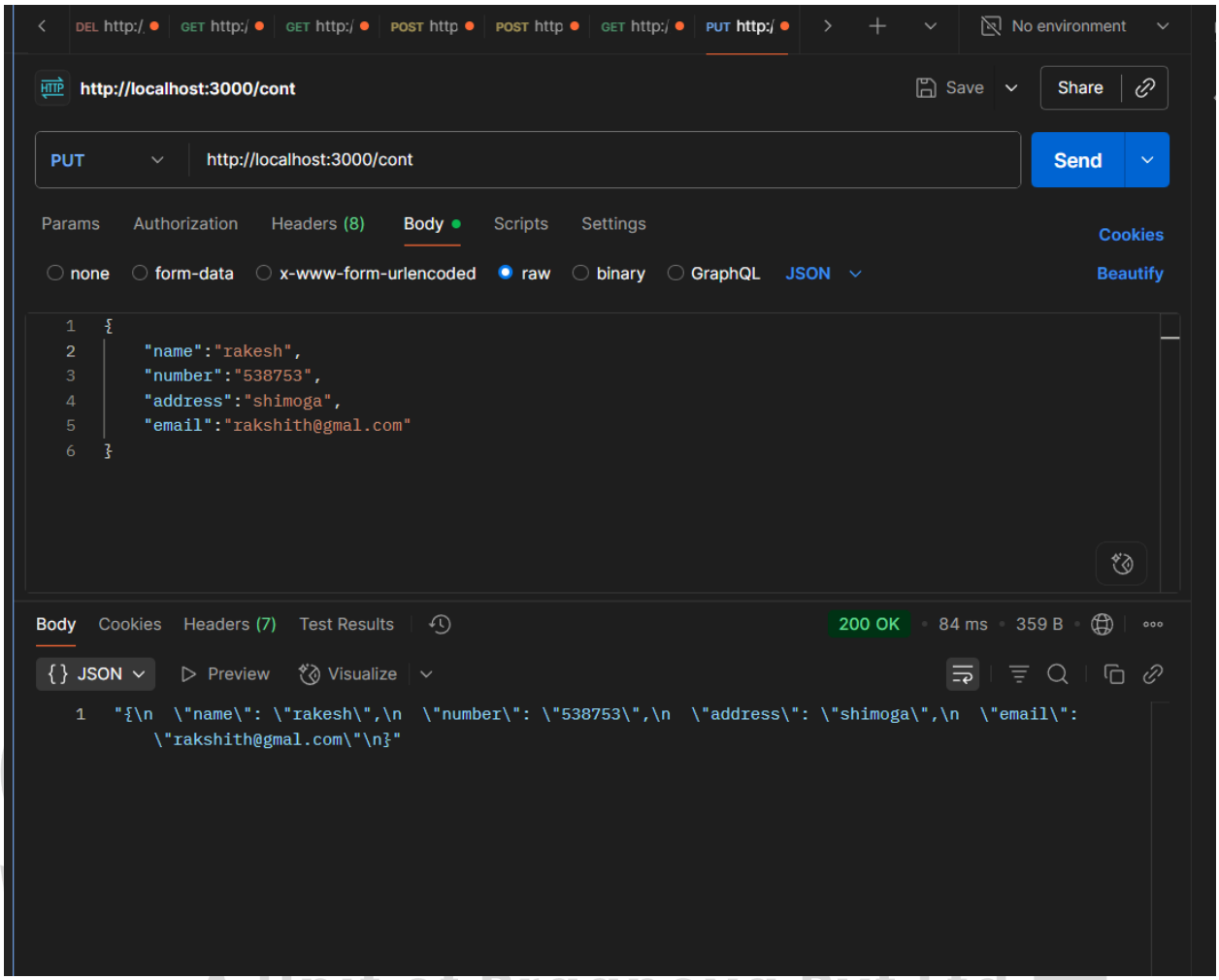
---

5. Screenshots of Output

```
{
  "name": "akash",
  "number": "224324",
  "address": "shimoga",
  "email": "akash@gmal.com"
}{
  "name": "rakshith",
  "number": "538753",
  "address": "shimoga",
  "email": "rakshith@gmal.com"
}{
  "name": "rakesh",
  "number": "538753",
  "address": "shimoga",
  "email": "rakshith@gmal.com"
}
```


http://localhost:3000/cont
Save
Share
POST
http://localhost:3000/cont
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"name":"akash",
"number":"224324",
"address":"shimoga",
"email":"akash@gmal.com"
}
Body
Cookies
Headers (7)
Test Results
200 OK
34 ms
355 B
JSON
Preview
Visualize
"{\n \"name\": \"akash\",\n \"number\": \"224324\",\n \"address\": \"shimoga\",\n \"email\": \"akash@gmal.com\"\n}"


Home
Workspaces
API Network
Search Postman
Invite
Upgrade
Akash's Workspace
New
Import
Collections
Environments
Flows
History
My Collection
No environment
http://localhost:3000/login
Save
Share
GET
http://localhost:3000/login
Send
Params
Authorization
Headers (6)
Body
Scripts
Settings
Cookies
Beautify
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Ctrl+Alt+P for Postbot
Body
Cookies
Headers (8)
Test Results
404 Not Found
5 ms
416 B
HTML
Preview
Visualize
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Error</title>
</head>
<body>
<pre>Cannot GET /login</pre>
</body>
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Vault
Trash



6. Observation / Reflection

The code provided is a good start, but there are a few issues and areas for improvement:

- The submission array is not necessary and can be removed.

- The fs.appendFile method is used to append new contact objects to the cont.json file, but this will result in invalid JSON if multiple contacts are added. Instead, consider reading the existing JSON data, parsing it, adding the new contact to the array, and then writing the updated JSON data back to the file.

- The GET /cont endpoint assumes that the cont.json file exists and contains valid JSON data. Consider adding error handling to handle cases where the file does not exist or the data is invalid.