

Assignment

Learner Details

- **Name:** Afreen Ahmed
 - **Enrollment Number:** SU625MR003
 - **Batch / Class:** June 2025 MERN
 - **Assignment:** CRUD Operations using Postman
 - **Date of Submission:** 13/08/2025
-

Problem Solving Activity

1. Program Statement

Perform CRUD (Create, Read, Update, Delete) operations on ATM data using Postman as the API testing tool. The application will be built using Node.js with Express.js and will accept and return data in JSON format.

2. Algorithm

- Start the Node.js server using Express.js.
 - Enable JSON parsing in Express to handle incoming JSON data from Postman.
 - Create routes for:
 - **GET "/"** → Display ATM home page message.
 - **GET "/contactUs"** → Display contact information.
 - **POST "/sendAtm"** → Create ATM data from request body.
 - **PUT "/updateAtm/:id"** → Update ATM data based on the given ID.
 - **DELETE "/deletedAtm/:id"** → Delete ATM data by ID.
 - Start server and listen on the specified port.
 - Test each route in Postman by sending appropriate HTTP requests with JSON payloads where needed.
-

3. Pseudocode

BEGIN

IMPORT express module

CREATE app object from express

ENABLE express.json() middleware to parse JSON input

LISTEN on port 4000

PRINT "Server running"

DEFINE route GET "/"

SEND "ATM home page message"

DEFINE route GET "/contactUs"

SEND "ATM contact info message"

DEFINE route POST "/sendAtm"

EXTRACT data from request body

SEND "ATM data received" along with JSON data

DEFINE route PUT "/updateAtm/:id"

EXTRACT id from request parameters

EXTRACT updated data from request body

SEND "ATM data updated" message with new data

DEFINE route DELETE "/deletedAtm/:id"

EXTRACT id from request parameters



SEND "ATM data deleted" message

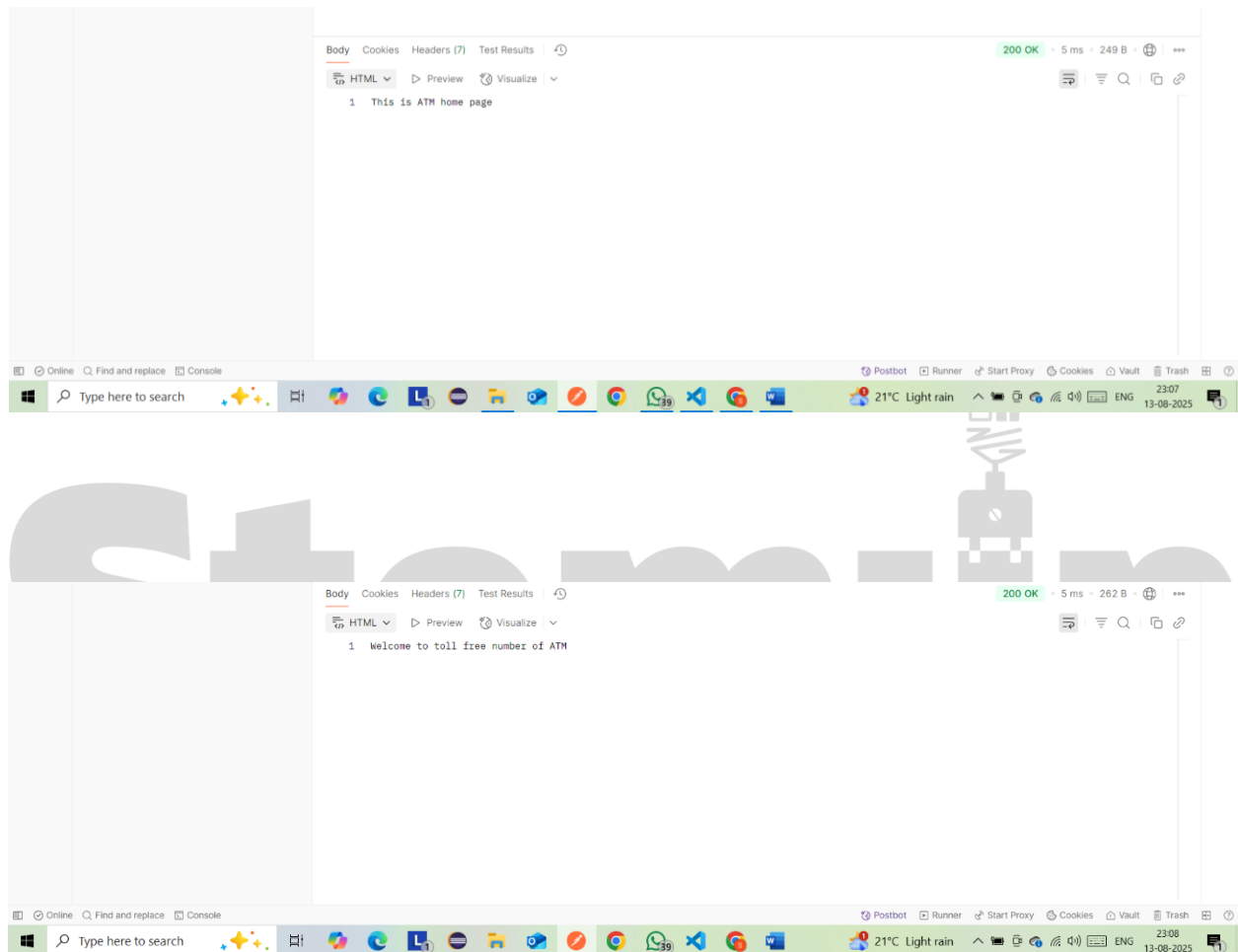
END

4. Program Code

```
{} package.json  JS httpserver.js  JS exp.js  X  JS middleware.js

JS exp.js > ...
1  var express = require('express')
2  var app = express();
3
4  //change html data to json data explicitly
5  app.use(express.json());
6
7  app.listen(4000, () =>{
8    console.log('Request Running on port 4000 server running');
9  })
10
11 app.get("/",(req,res)=>{
12   res.send('This is ATM home page');
13 })
14
15 app.get("/contactUs",(req,res)=>{
16   res.send('Welcome to toll free number of ATM');
17 })
18
19 app.post("/sendAtm",(req,res)=>{
20   res.send(`Sending ATM data ${JSON.stringify(req.body)}`);
21 })
22
23 app.put("/updateAtm/:id",(req,res)=>{
24   res.send(`Updating ATM data ${JSON.stringify(req.body)}`);
25 })
26
27 app.delete("/deletedAtm/:id",(req,res)=>{
28   res.send(`Deleting ATM data ${req.params.id}`);
29 })
```

6. Screenshots of Output



POST http://localhost:4000/sendAtm

Params Authorization Headers (9) Body Scripts Settings

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

```
1 { "atmId": 101,  
2   "location": "Mumbai"  
3 }
```

200 OK · 4 ms · 278 B

HTML Preview Visualize

1 Sending ATM data {"atmId":101,"location":"Mumbai"}

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

Type here to search 21°C Mostly cloudy 23:08 13-08-2025

PUT http://localhost:4000/updateAtm/id

Params Authorization Headers (9) Body Scripts Settings

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

```
1 { "atmId": 101,  
2   "location": "Delhi"  
3 }
```

200 OK · 4 ms · 279 B

HTML Preview Visualize

1 Updating ATM data {"atmId":101,"location":"Delhi"}

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

Type here to search 21°C Mostly cloudy 23:11 13-08-2025

7. Observation / Reflection

Setting up routes in Express was easy, but I had to enable `express.json()` for JSON data. Postman testing helped me check each method. I learned to name routes clearly and use `req.params` and `req.body` properly. Next time, I'll add error handling and connect it to MongoDB for real data.

