SIT314 – Week 7 Technical Tasks 7A & 7B Report

# Task 7A: Testing CRUD RESTful APIs using Postman and JSON Server

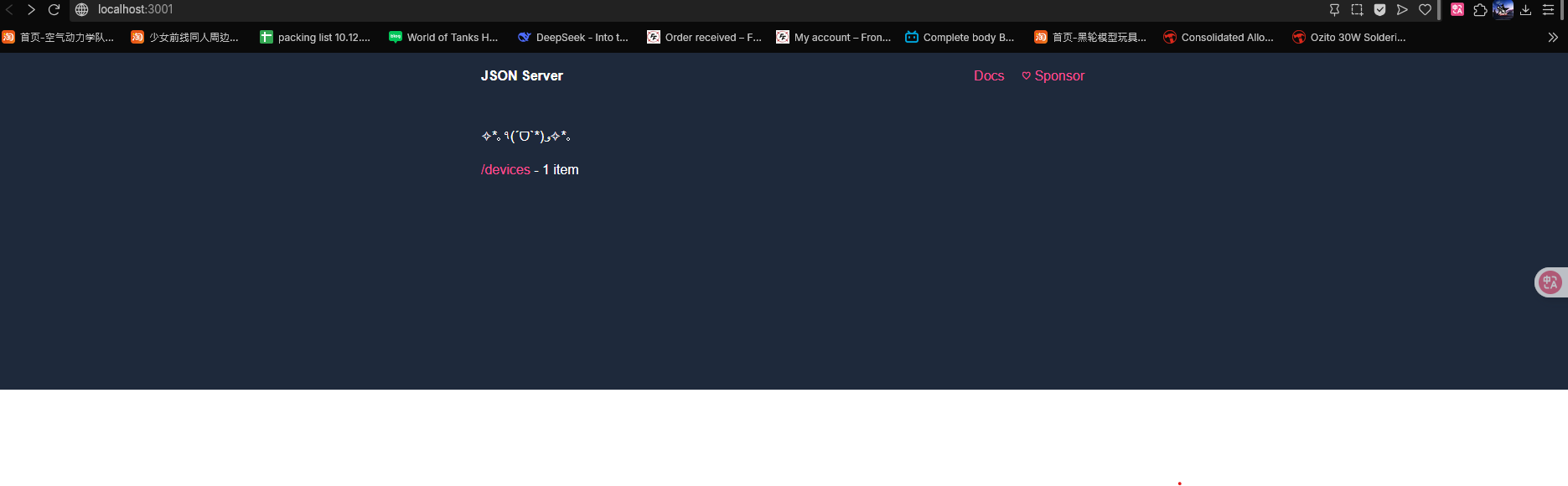
## Objective

The aim of this task was to set up a mock RESTful web API using json-server, populate it with sensor data in a local data.json file, and then test CRUD operations (Create, Read, Update, Delete) using Postman.

## Steps Completed

### Step 1: JSON Server Setup

A mock database was created using json-server and a sample data.json file with devices and sensor readings. The server was launched using the command:  
  
json-server data.json --port 3001

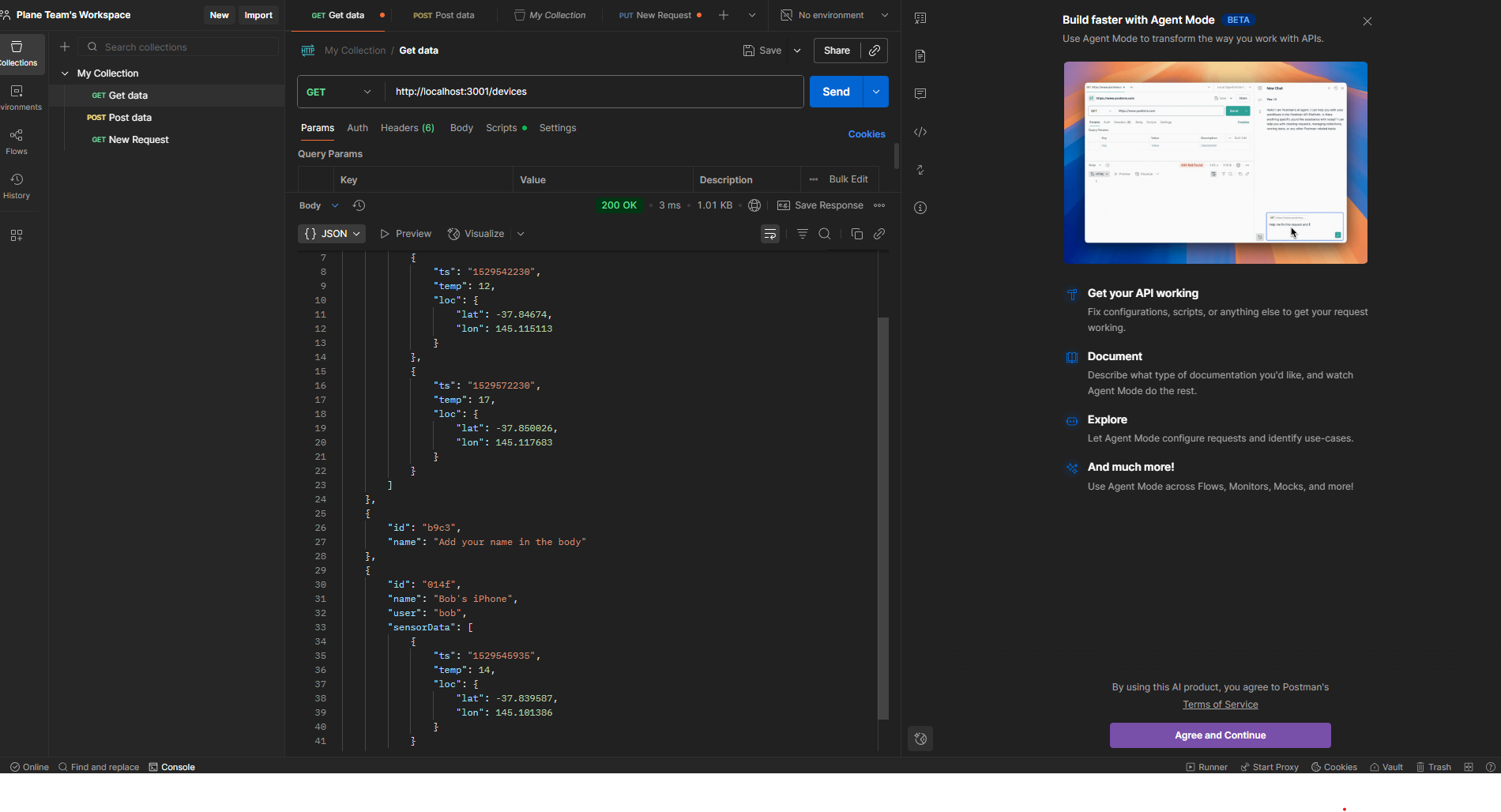
Screenshot:  
 Description: This screenshot shows the landing page of the json-server running on http://localhost:3001. The route /devices is available and indicates one item is currently stored in the database.

### Step 2: GET All Devices

### Screenshot:

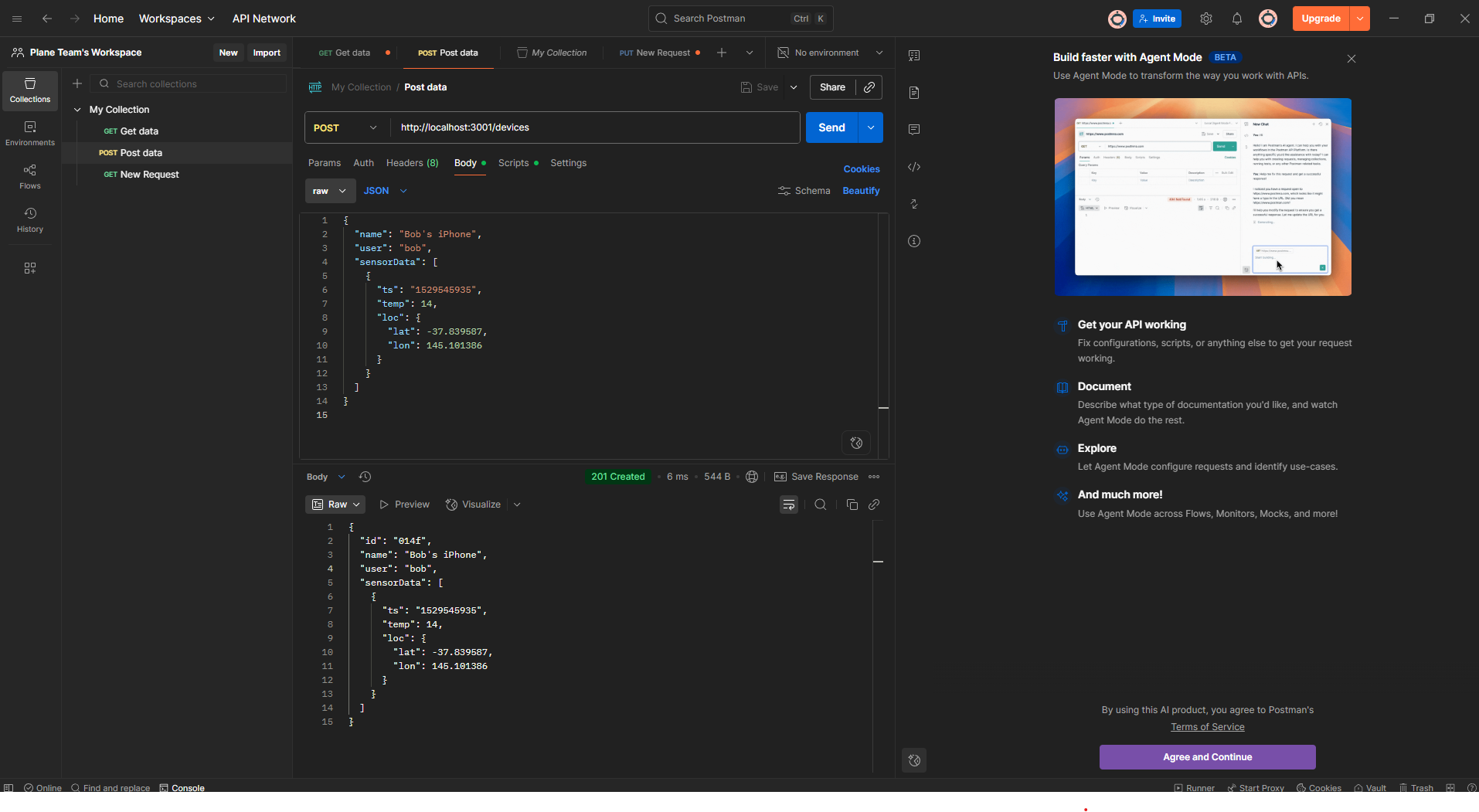
Description: This screenshot shows the Postman interface after sending a GET request. The response includes multiple sensor device entries, each containing timestamp, temperature, and GPS location data.

### Step 3: GET Single Device by ID

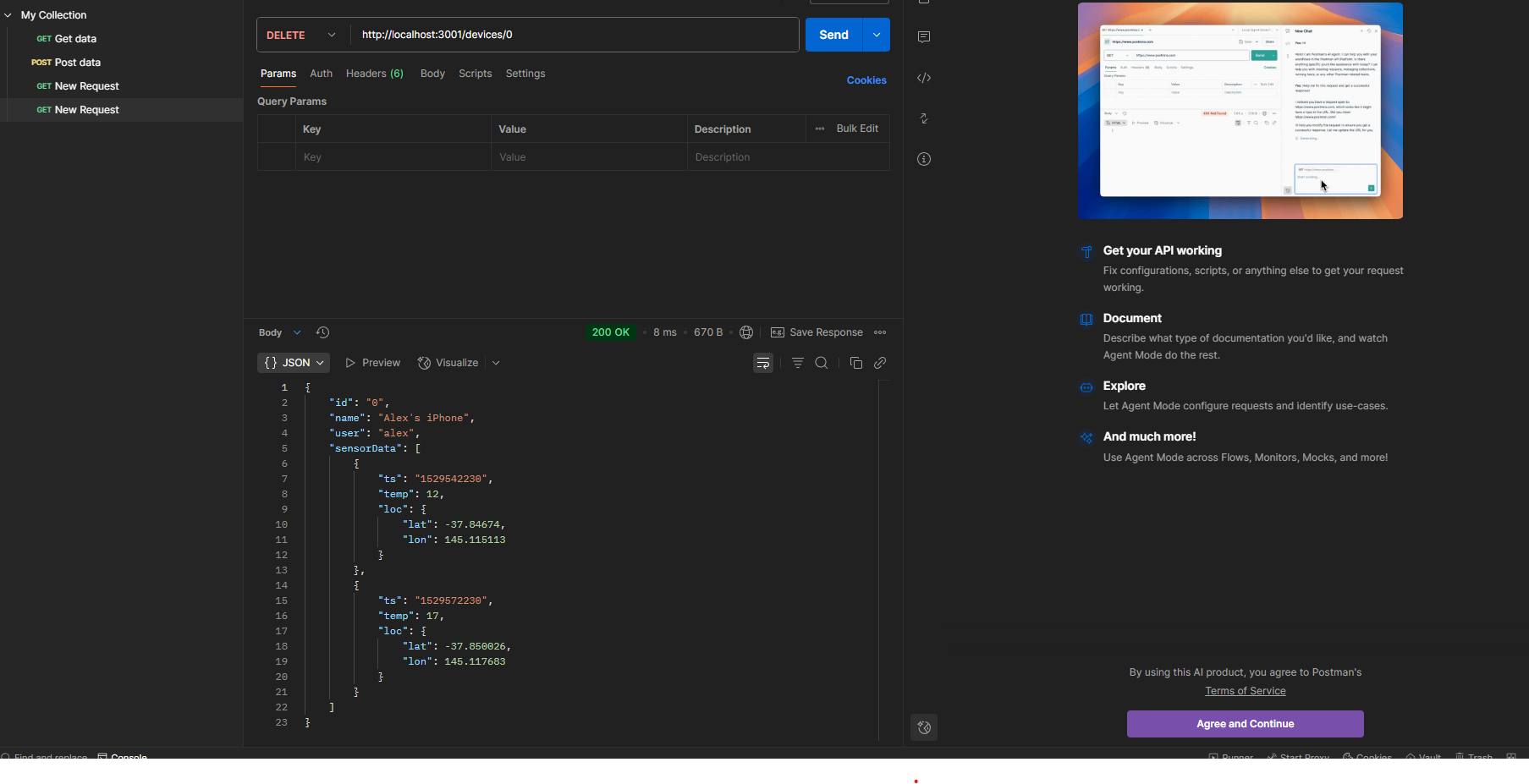
Screenshot:  
 Description: This request targets device ID 0. The response contains the full sensor history for Alex’s iPhone, including two sets of sensor readings.

### Step 4: POST New Device Entry

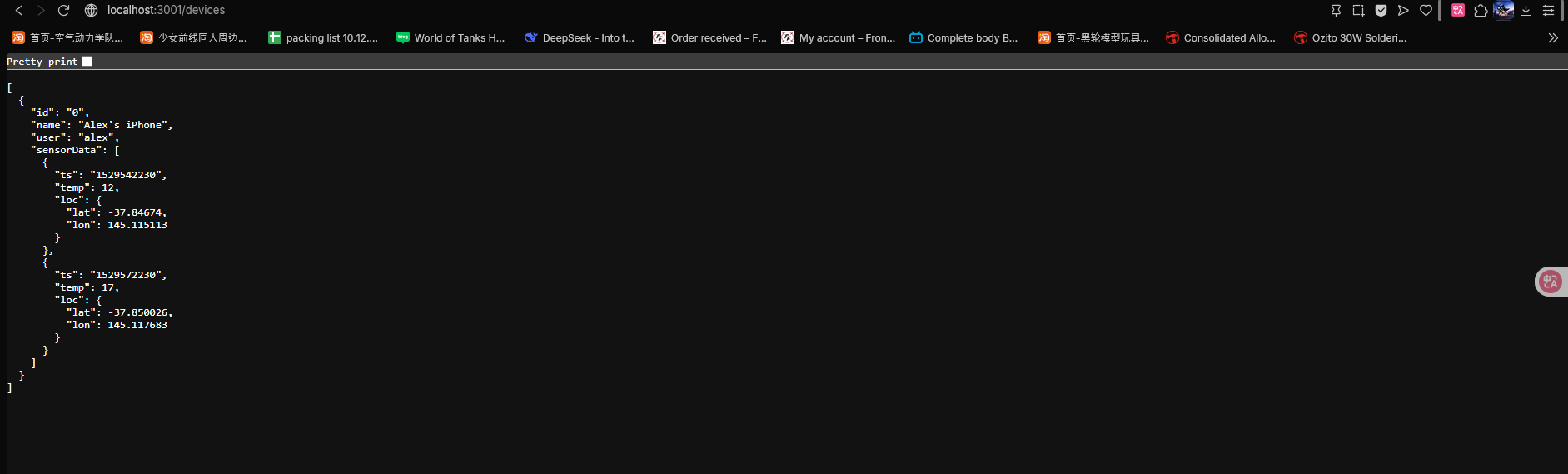
Screenshot:

  
 Description: This screenshot shows a successful POST request. A new device ("Bob's iPhone") with corresponding sensor data was added to the devices array, confirmed by the returned 201 Created response.

### Step 5: DELETE Device by ID

Screenshot:  
 Description: The response confirms a successful deletion of the device with ID 0. The 200 OK status shows the operation was completed correctly.

### Step 6: Visualize All Devices Again

Screenshot:  
 Description: The updated mock database is viewed in the browser. It reflects all previously made changes, including the removal of one device and addition of another.

## Summary

Successfully set up and ran a local mock REST API using json-server.  
  
Performed all four CRUD operations:  
- GET (read all and read by ID)  
- POST (create new device entry)  
- DELETE (remove a device entry)  
  
Used Postman for testing each API operation.  
Collected screenshots as evidence of each step.

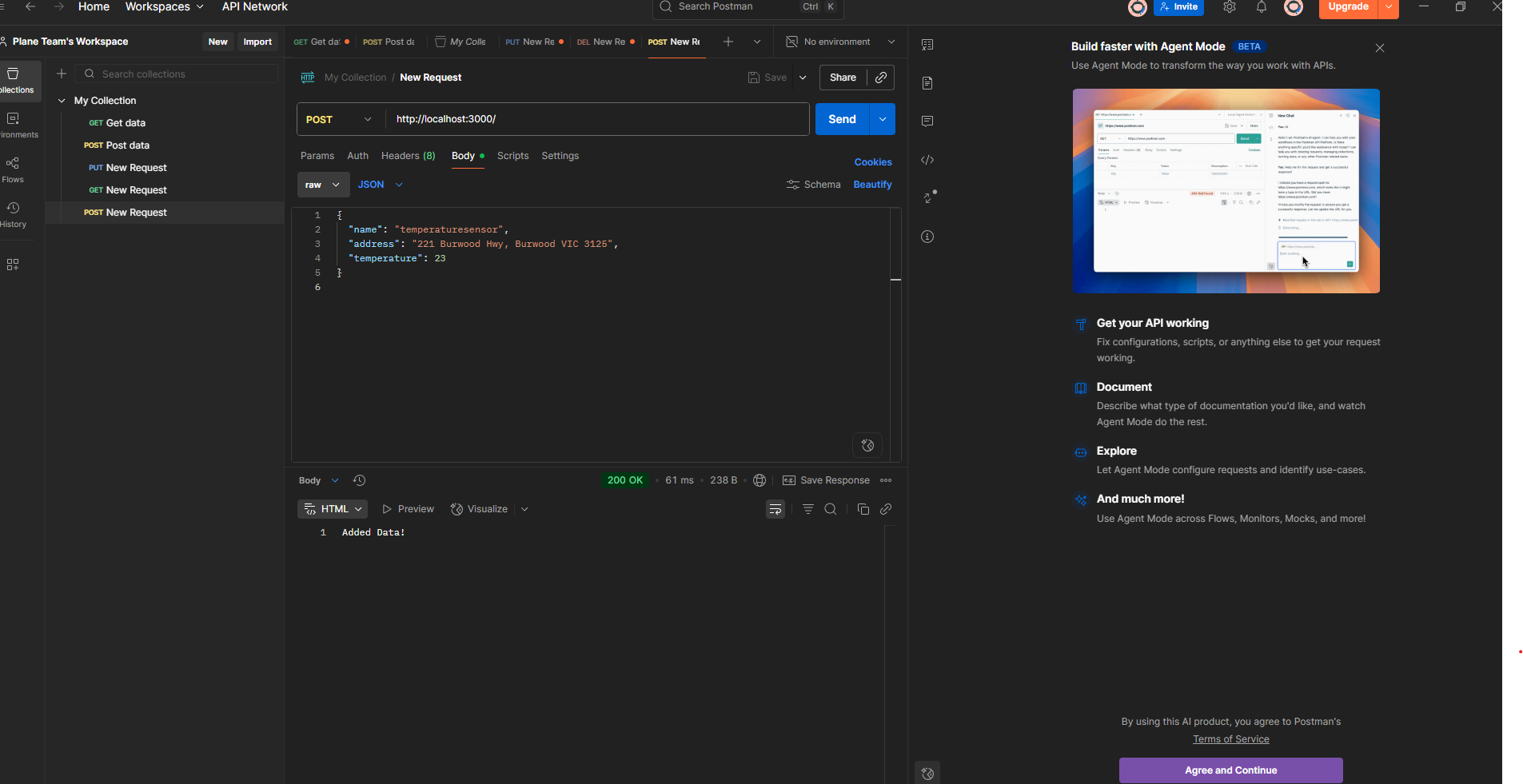
# Task 7B: Testing RESTful API with Express.js and MongoDB

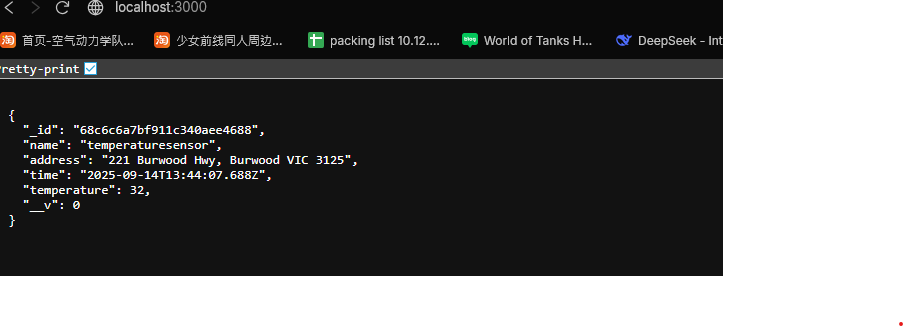
## Objective

The goal of this task was to create a backend API using Express.js and MongoDB to store temperature sensor data and interact with it via POST and GET requests. Postman was used for sending HTTP requests, while MongoDB Compass was used to validate inserted records in the database.

## Steps Completed

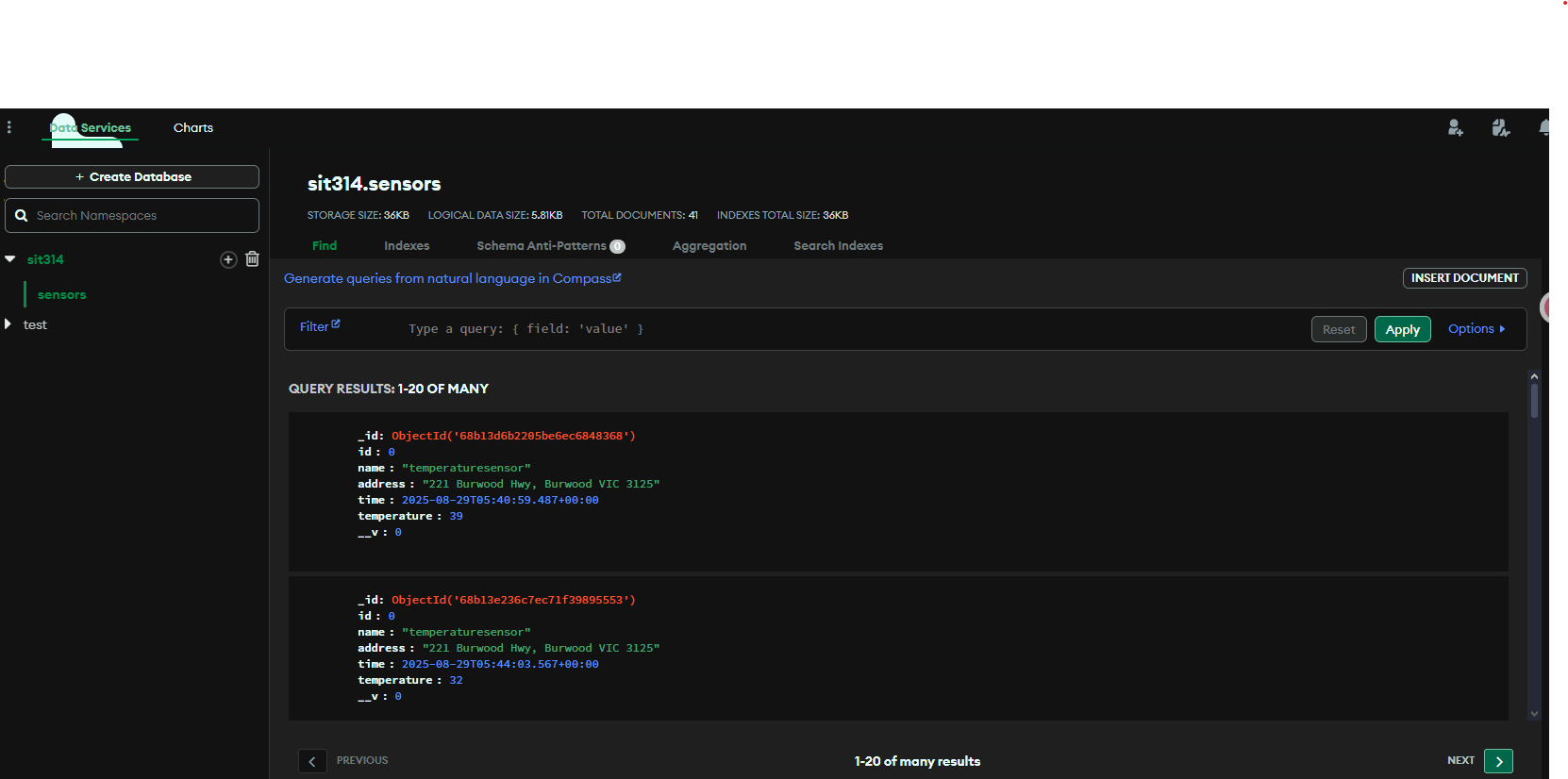
### Step 1: POST New Sensor Data via Postman Screenshot:

  
 Description: This screenshot shows a successful POST request made via Postman. A JSON payload containing a temperature sensor ("temperaturesensor") at an address in Burwood was sent to the server, and the response returned a 200 OK status along with a confirmation message: "Added Data!"

  
Step 2: GET a Single Record by ID

**Description:** This screenshot shows the result of querying MongoDB for a specific document by ObjectID. The correct sensor details (name, address, temperature, time) were returned in the response.

### Step 3: Validate Sensor Data in MongoDB Compass

Screenshot:  
 Description: The screenshot displays two successfully inserted documents inside the sit314.sensors collection. Each record has a temperature reading (32°C and 39°C respectively), a name ("temperaturesensor"), and a location ("221 Burwood Hwy, Burwood VIC 3125"). This validates that the server correctly processed and stored the POSTed data.

## Summary

- Successfully configured an Express.js backend that interacts with MongoDB.  
- Tested the backend using Postman to send POST requests containing sensor data.  
- Verified data persistence using MongoDB Compass, confirming document creation in the correct database and collection.  
- This task demonstrates the ability to perform data insertion into a NoSQL database through a RESTful API and verify backend integration using Postman and Compass.