EXPERIMENT 4

REST API Design with MongoDB + Mongoose Integration

CODE:

awarenessRoutes.js | db.js | Footprint.js | Waterlog.js | Recyclable.js | Donation.js

routes > awarenessRoutes.js > ...

```
1   const express = require('express');
2   const router = express.Router();
3
4   // Sample sustainability awareness tips
5   const tips = [
6     { id: 1, tip: "Switch to reusable bottles instead of single-use plastic." },
7     { id: 2, tip: "Turn off lights and appliances when not in use to save energy." },
8     { id: 3, tip: "Donate old clothes instead of throwing them away." },
9     { id: 4, tip: "Walk or cycle for short distances instead of using vehicles." },
10  ];
11
12  // GET all tips
13  router.get('/', (req, res) => {
14    res.json(tips);
15  });
16
17  // GET one tip by ID
18  router.get('/:id', (req, res) => {
19    const tip = tips.find(t => t.id === parseInt(req.params.id));
20    if (!tip) {
21      return res.status(404).json({ message: "Tip not found" });
22    }
23    res.json(tip);
24  });
25
26  module.exports = router;
27
```

ootprintController.js | footprintRoutes.js | recyclablesController.js | recyclablesRoute

routes > donationsRoutes.js > ...

```
1   const express = require('express');
2   const router = express.Router();
3   const Donation = require('../models/donationModel');
4
5   // POST a new donation
6   router.post('/', async (req, res, next) => {
7     try {
8       const { item, quantity, donorName } = req.body;
9
10      const donation = new Donation({ item, quantity, donorName });
11      const savedDonation = await donation.save();  // saves to MongoDB
12
13      res.status(201).json({
14        message: 'Donation saved successfully!',
15        donation: savedDonation
```

footprintController.js | footprintRoutes.js | recyclablesController.js | recyclablesRoutes.js | seed.js | donationsI

routes > footprintRoutes.js > ...

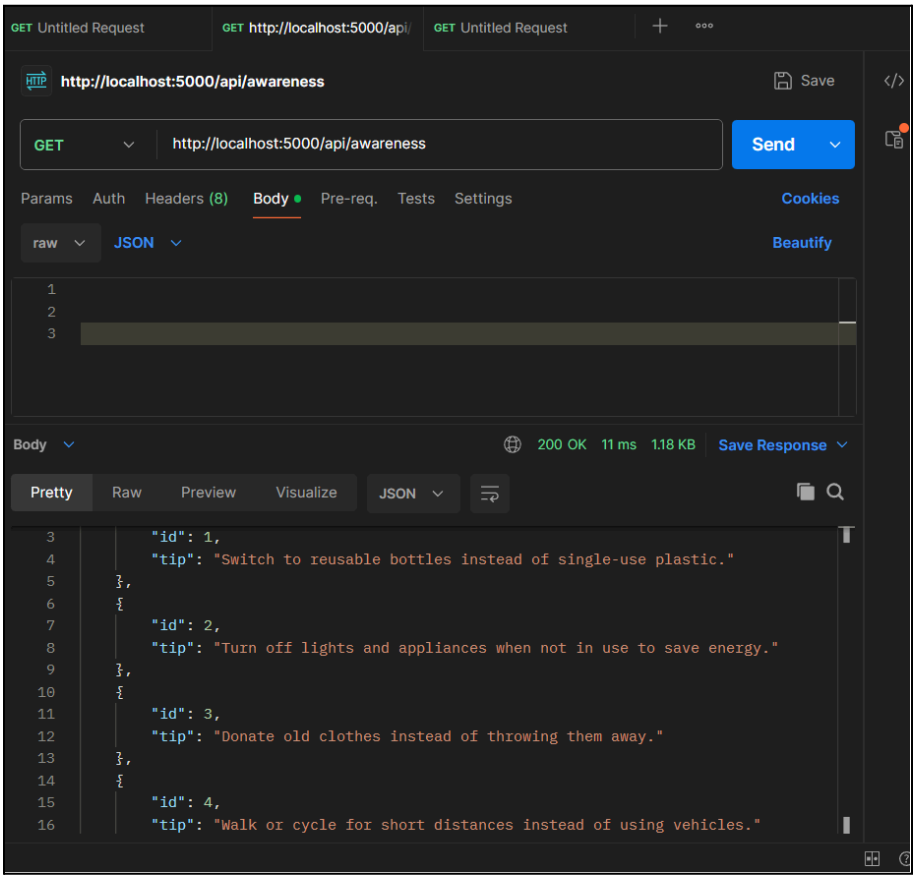
```
5
6   router.post('/',
7     body('transportMode').isString().notEmpty(),
8     body('distanceKm').isNumeric().optional({ nullable: true }),
9     async (req, res, next) => {
10      const errors = validationResult(req);
11      if (!errors.isEmpty()) return res.status(400).json({ errors: errors.array() });
12      next();
13    },
14    createFootprint
15  );
16
17  router.get('/', getFootprints);
18  router.get('/summary', getSummary);
19
20  module.exports = router;
21
```

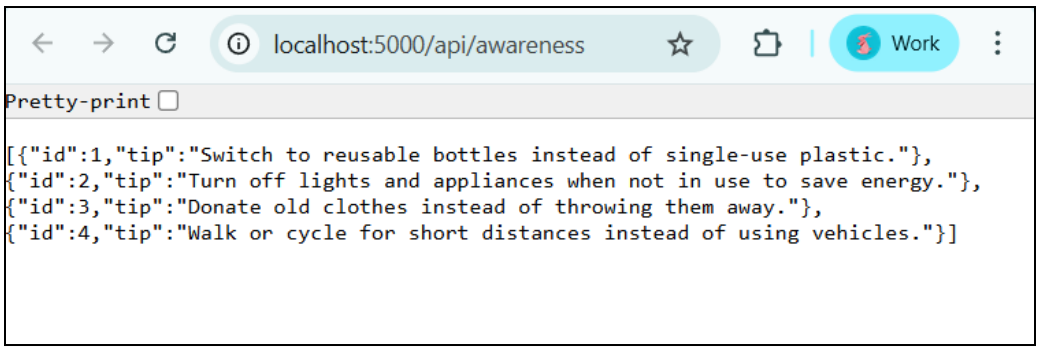
footprintController.js | footprintRoutes.js | recyclablesController.js | recyclablesRoutes.js | seed.js | donationsRoutes.js

routes > recyclablesRoutes.js > ...

```
1   const express = require('express');
2   const { createRecyclable, getRecyclables, updateRecyclable, deleteRecyclable } = require('../controllers/
    recyclablesController');
3   const router = express.Router();
4
5   router.post('/', createRecyclable);
6   router.get('/', getRecyclables);
7   router.put('/:id', updateRecyclable);
8   router.delete('/:id', deleteRecyclable);
9
10  module.exports = router;
11
```

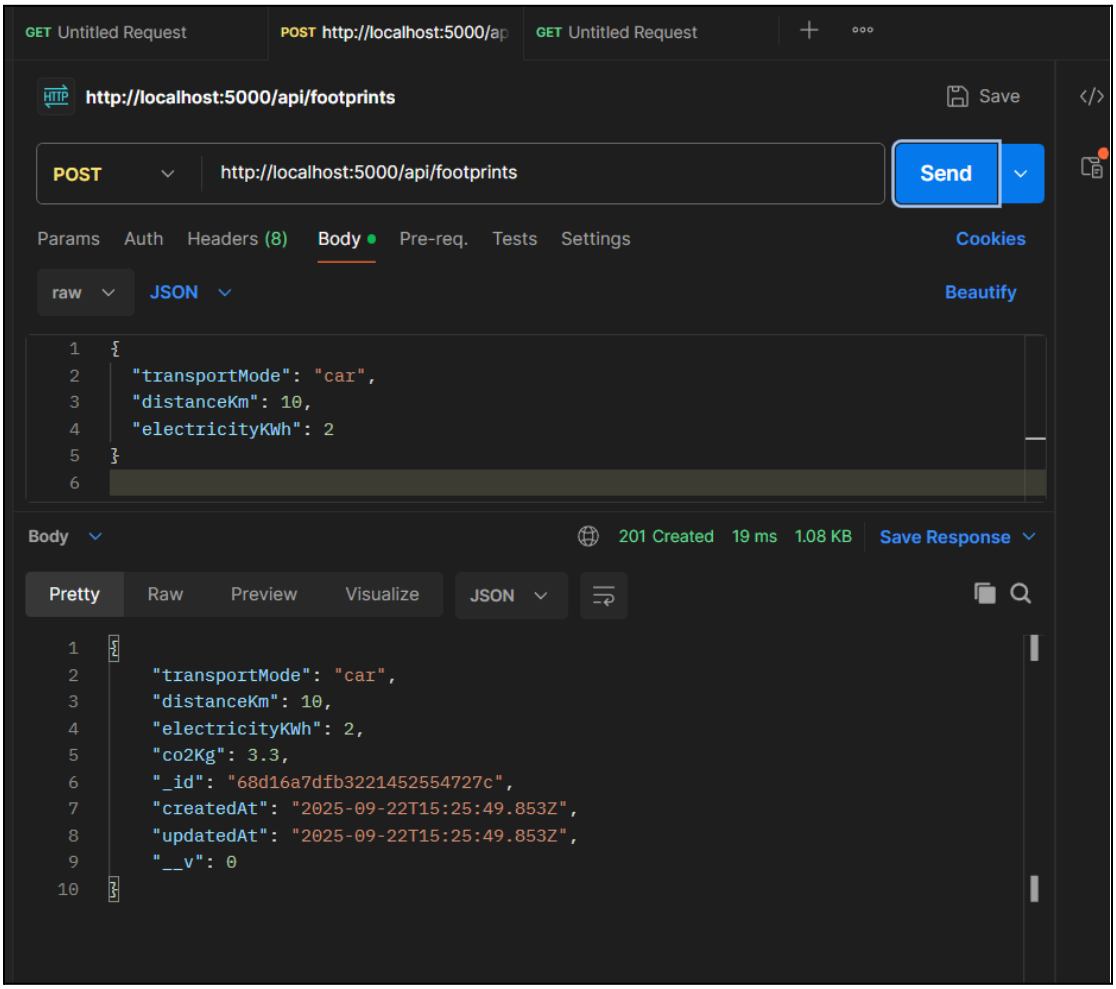
OUTPUT:

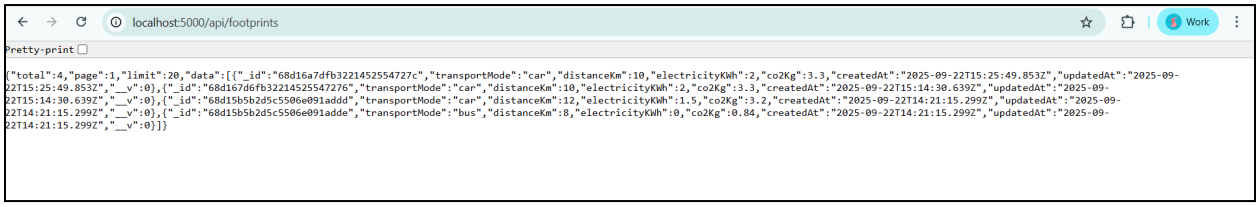




Awareness Tips (GET)

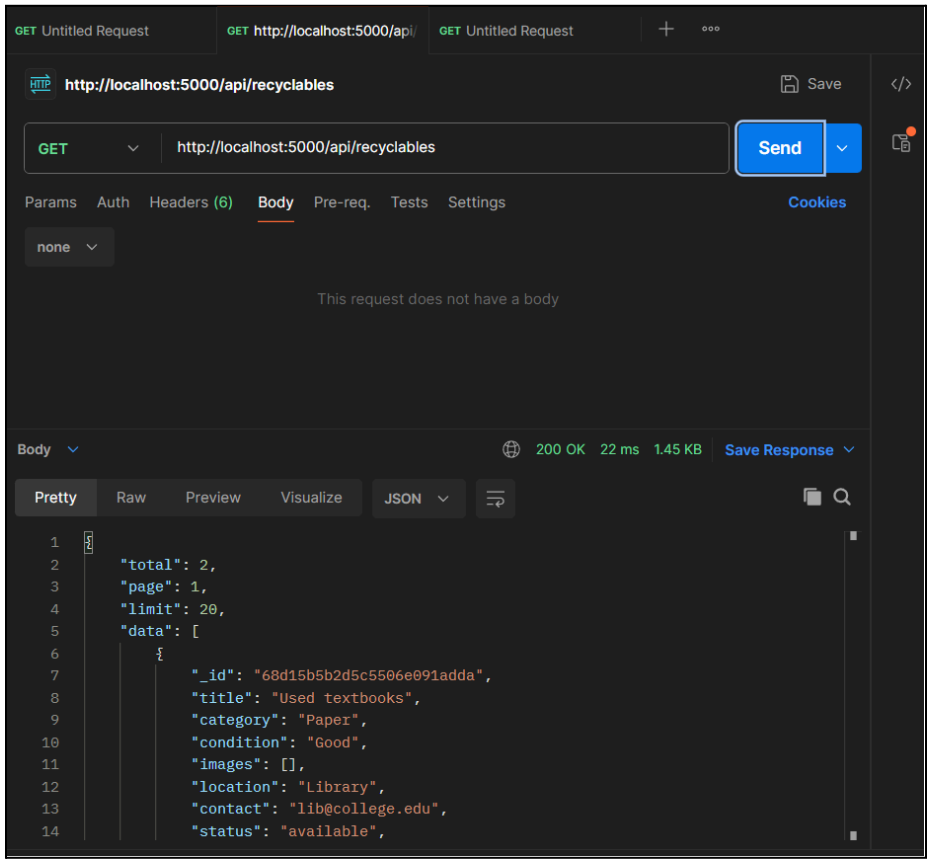
2



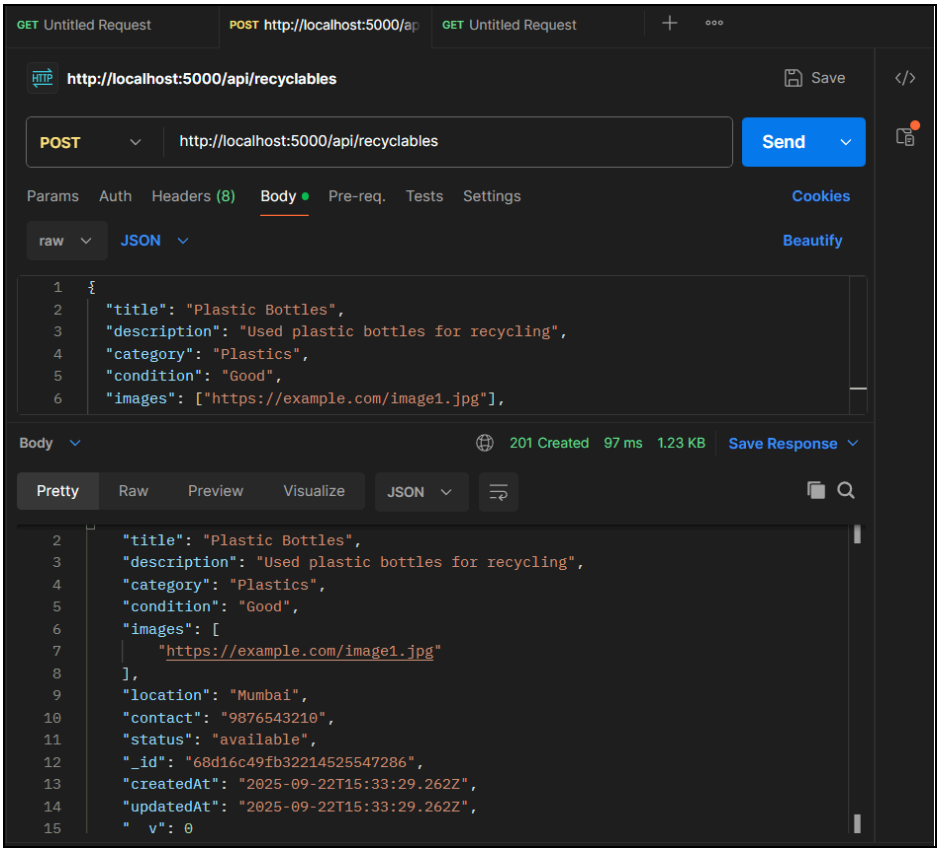


Carbon Footprint (POST)

3

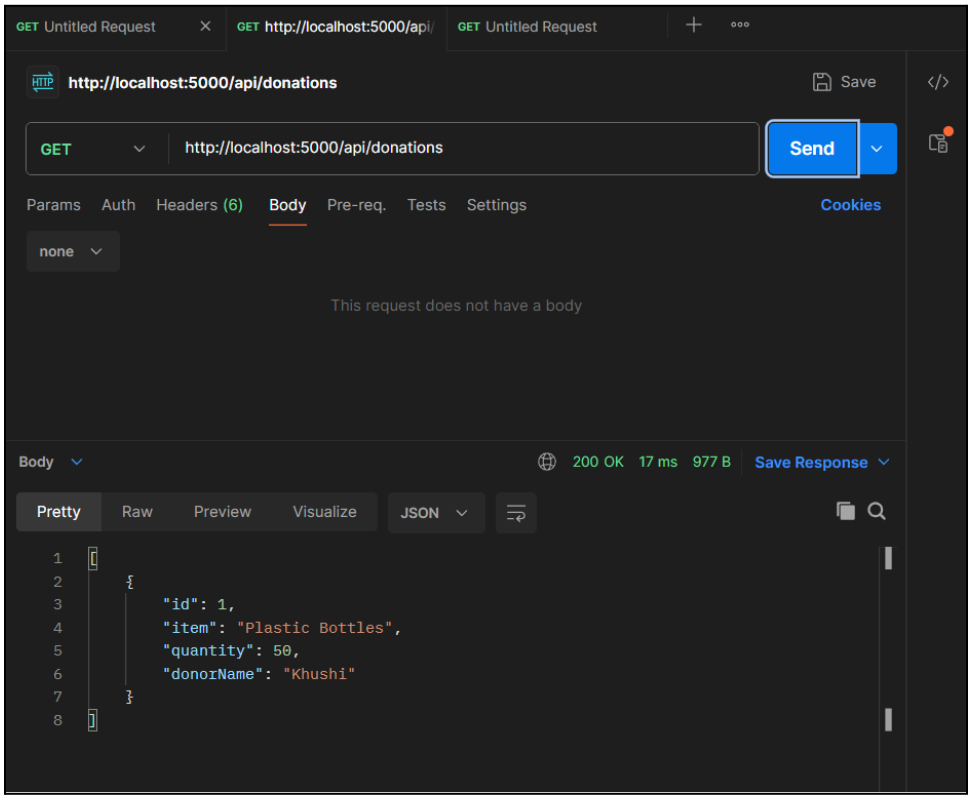


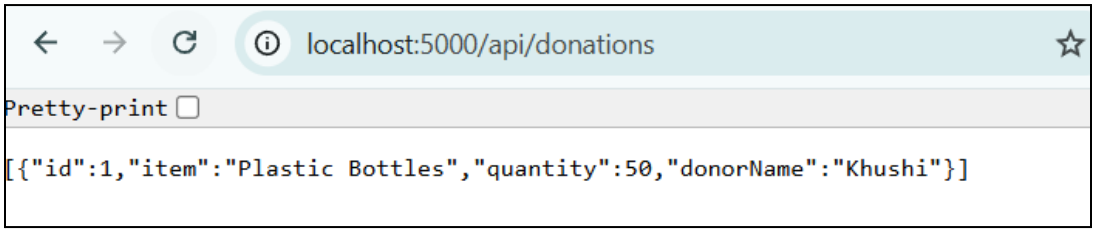
Recyclables (GET)

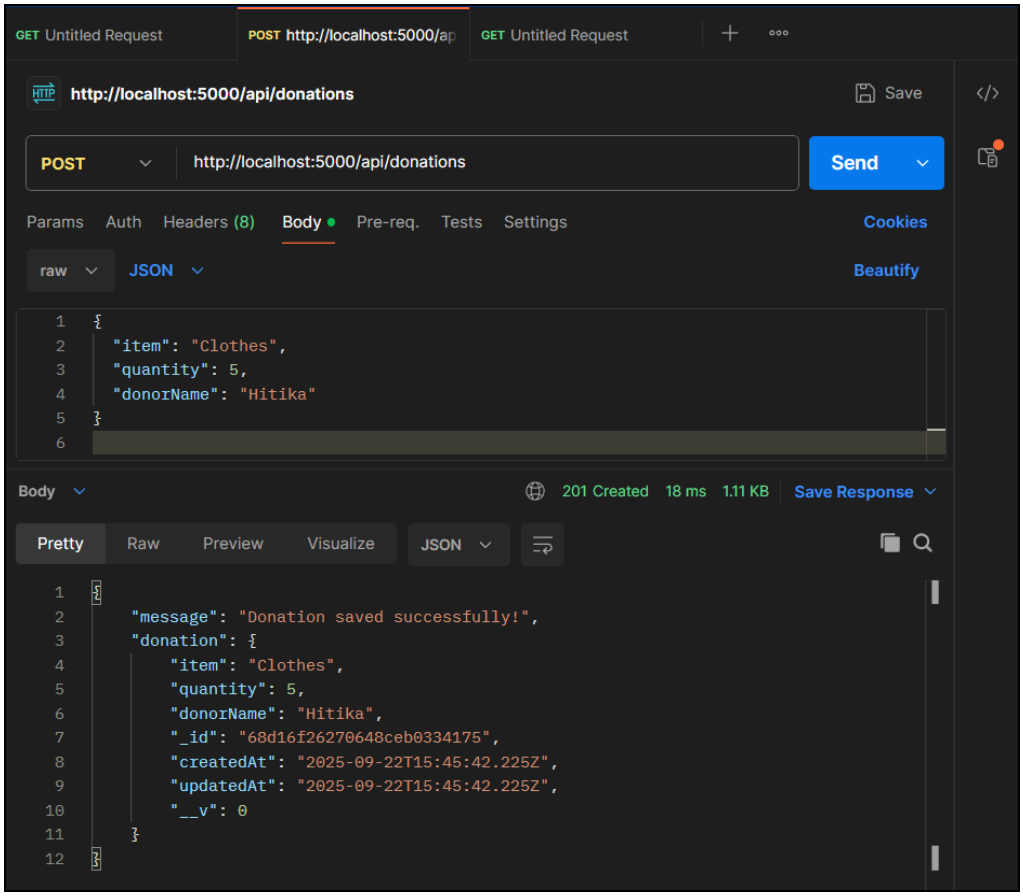


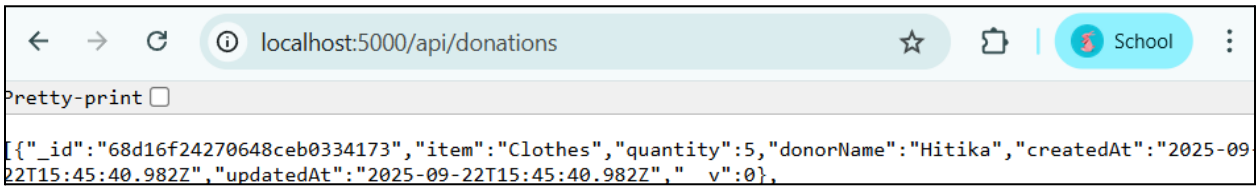
Recyclables (POST)

4


GET Untitled Request
GET http://localhost:5000/api/
GET Untitled Request
http://localhost:5000/api/donations
Save
GET
http://localhost:5000/api/donations
Send
Params
Auth
Headers (6)
Body
Pre-req.
Tests
Settings
Cookies
none
This request does not have a body
Body
200 OK
17 ms
977 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
[
{
"id": 1,
"item": "Plastic Bottles",
"quantity": 50,
"donorName": "Khushi"
}
]


localhost:5000/api/donations
Pretty-print
[{"id":1,"item":"Plastic Bottles","quantity":50,"donorName":"Khushi"}]





Donations (GET + POST)