



UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRONIK DAN KOMPUTER

BERR 2243
DATABASE & CLOUD SYSTEM

SEM 2 2024/2025

EXERCISE : WEEK 3

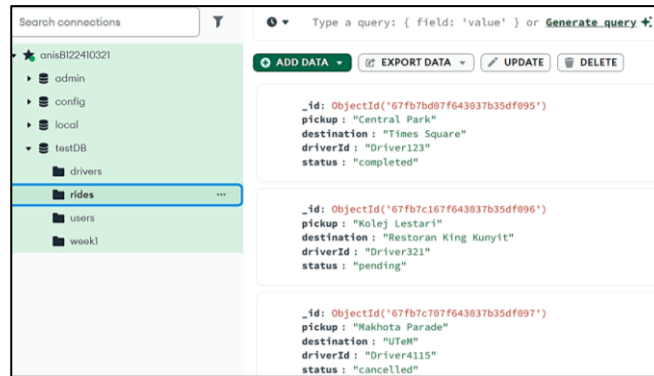
No	Name	Matrix No	Photo
1	NURUL ANIS HAFIFZA BINTI AMRAN	B122410321	
2	NUR AIN HIDAYAH BINTI ABDUL RAHIM	B122410323	

NURUL ANIS HAFIFZA BINTI AMRAN (B122410321)

1. Complete Code :

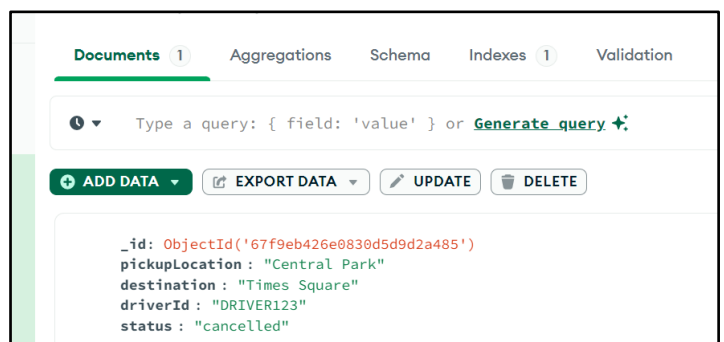
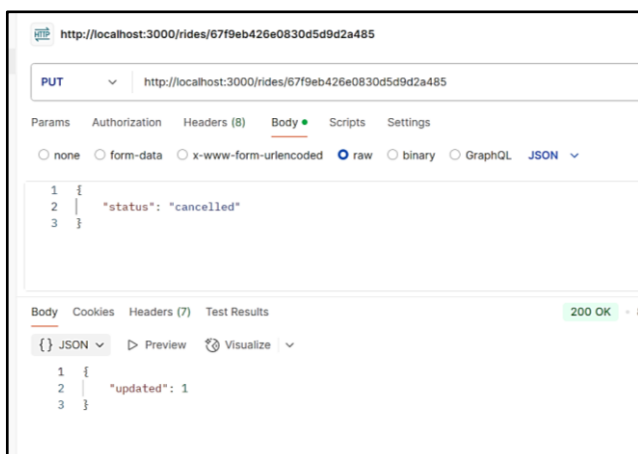
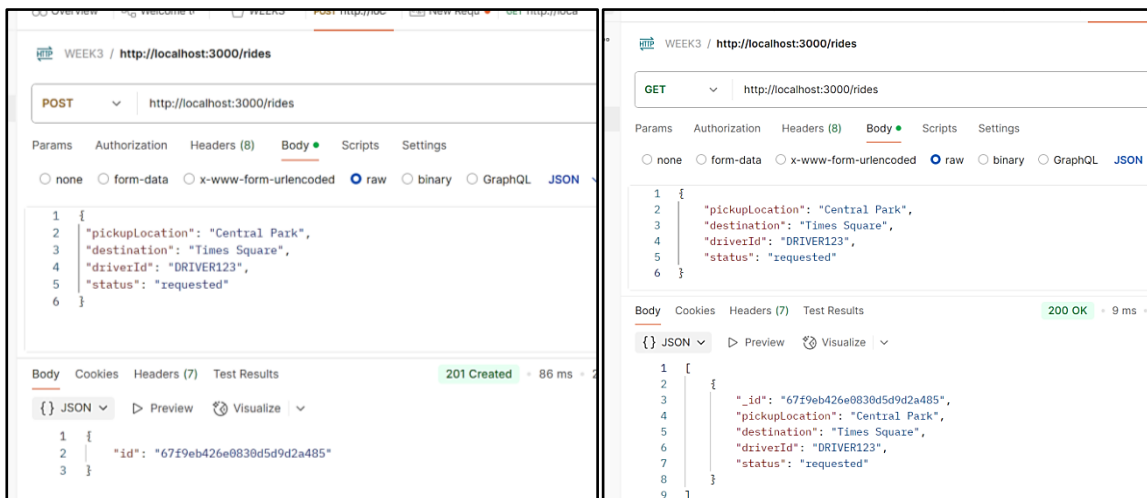
GitHub Repository: <https://github.com/anishafifza>

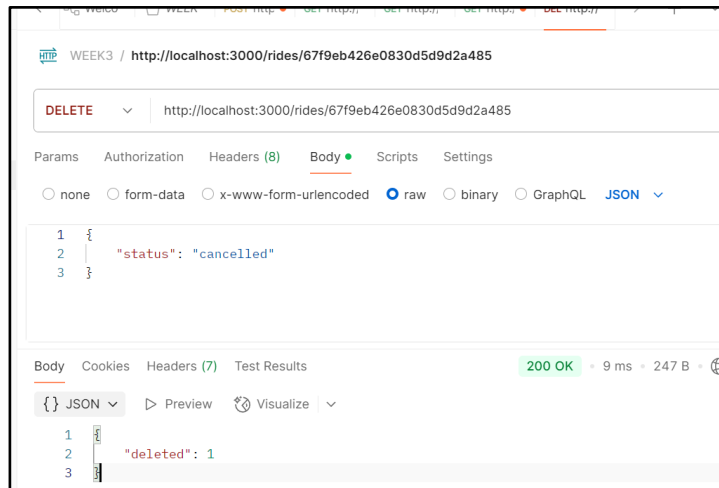
2. MongoDB compass Output :



3. Screenshots of:

➤ Postman requests/responses.

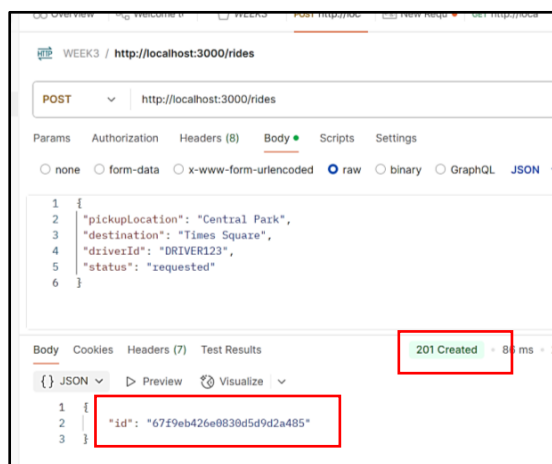




Exercise Questions

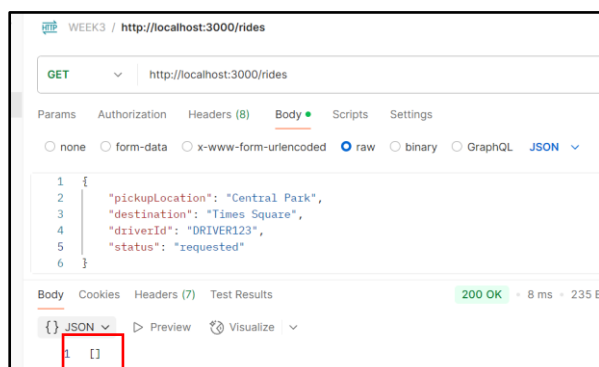
Answer 1:

- 201 created
- Response body :



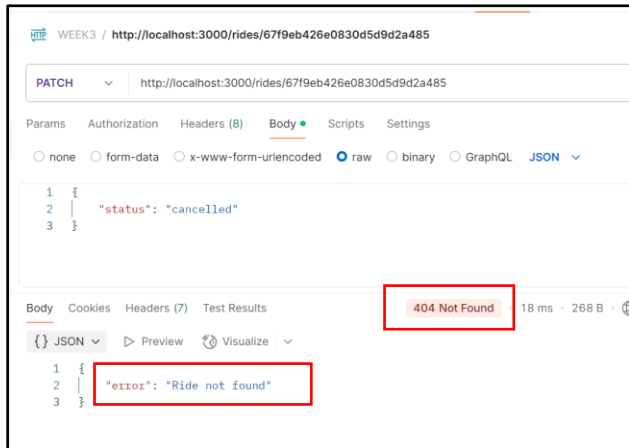
Answer 2:

- Rides collection is empty : when a collection is empty. It's a standard way to indicate that the request was successful but there are no items



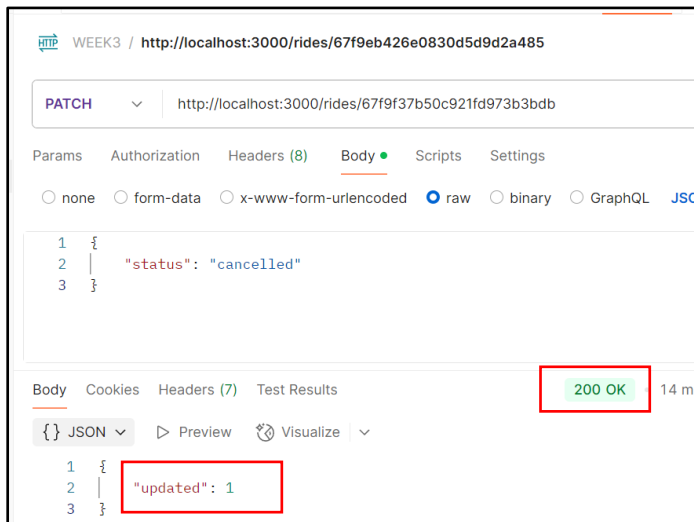
Answer 3:

- Catch the error when requesting PATCH :



-To update a non-existent ride ID, the status code is returned : **404 Not Found**

- Value of updated in the response if the update succeeds :



- API differentiates between a successful deletion and a failed one :

If `deletedCount === 1` → Successful → 200 OK

If `deletedCount === 0` → Not found → 404 Not Found

Answer 4 :

- Users Endpoints :
CRUD operations for users account

```
// GET /rides = Fetch all rides
app.get('/users', async (req, res) => {
  try {
    const rides = await db.collection('users').find().toArray();
    res.status(200).json(users);
  } catch (err) {
```

```

    res.status(500).json({ error: "Failed to fetch user" });
  }
});

// POST /rides - Create a new user
app.post('/users', async (req, res) => {
  try {
    const result = await db.collection('users').insertOne(req.body);
    res.status(201).json({ id: result.insertedId })
  } catch (err) {
    res.status(400).json({ error: "Invalid user data" });
  }
});

// PATCH/users/:id - Cancel a user status
app.patch('/users/:id', async (req, res) => {
  try {
    const result = await db.collection('users').updateOne(
      {
        _id: new ObjectId(req.params.id) },
      { $set: { status: req.body.status } }
    );

    if (result.modifiedCount === 0) {
      return res.status(404).json({ error: "User not found" });
    }

    res.status(200).json({ updated: result.modifiedCount });
  } catch (err) {
    res.status(400).json({ error: "Invalid user ID or data" });
  }
});

// DELETE /users/:id - Cancel a user
app.delete('/users/:id', async (req, res) => {
  try {
    const result = await db.collection('users').deleteOne(
      { _id: new ObjectId(req.params.id) }
    );

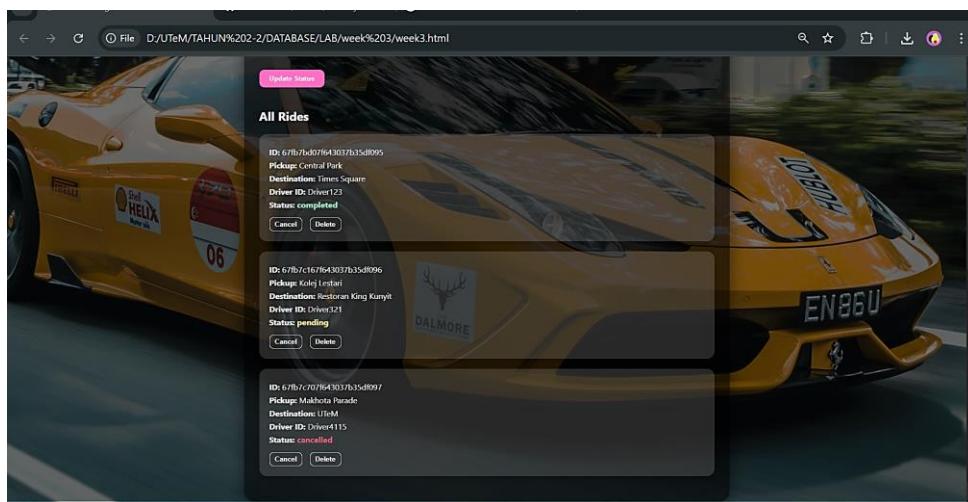
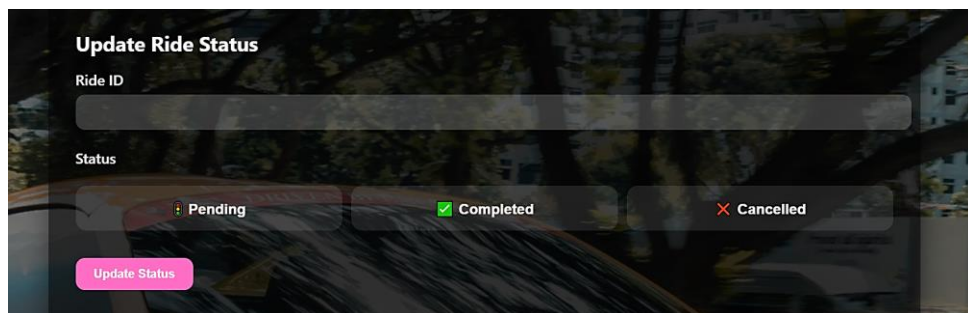
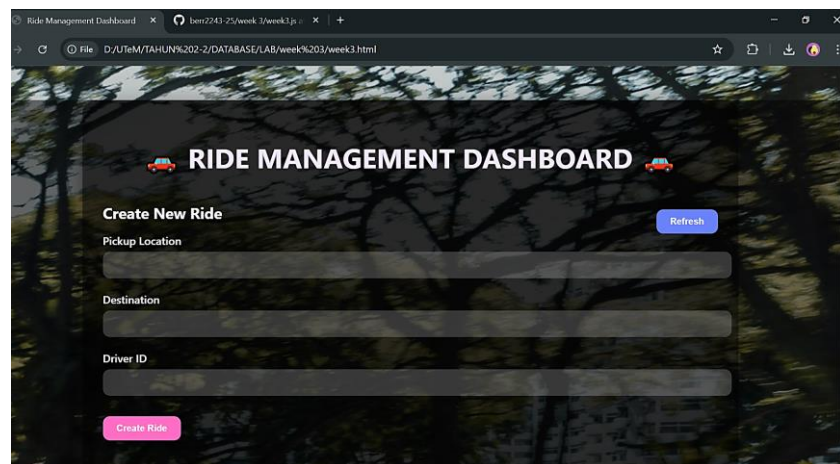
    if (result.deletedCount === 0) {
      return res.status(404).json({ error: "User not found" });
    }

    res.status(200).json({ deleted: result.deletedCount });
  } catch (err) {
    res.status(400).json({ error: "Invalid user ID" });
  }
});

```

Answer 5 :

-Upload the Postman JSON to any AI tools, and generate a simple HTML and JS Dashboard



Nur ain hidayah binti abdul rahim(b122410323)

Submission Requirements

1. GitHub Repository with:

- o Complete code.

Github link: <https://github.com/b122410323/berr2243-25.git>

```
JS week3.js X week3.html JS week2.js JS index.js
week 3 > JS week3.js > app.post('/rides') callback
1 // Install Express Library from the terminal
2 // function node is run to code "node index.js"
3
4 const express = require('express'); // Node.js framework to build API's web app
5 const { MongoClient, ObjectId } = require('mongodb');
6 const port = 3000; // create port and server
7
8 const app = express();
9 app.use(express.json());
10
11 const cors = require('cors');
12 app.use(cors());
13
14 let db;
15
16 async function connectToMongoDB() { // line 13 sampai 30 connect to mongoDB
17   const uri = "mongodb://localhost:27017/"
18   const client = new MongoClient(uri);
19
20   try {
21     await client.connect();
22     console.log("Connected to MongoDB !");
23
24     db = client.db("testDB");
25
26   } catch (err) {
27     console.error("Error:", err);
28   }
29 }
30 connectToMongoDB();
31
32 app.listen(port, () => {
33   console.log(`Server running on http://localhost:${port}`);
34 });
35
36 // GET /rides = Fetch all rides
37
38 app.get('/rides', async (req, res) => { // send a GET req to /rides, this fx is triggered
39   try {
```

```

40     const rides = await db.collection('rides').find().toArray(); // Find all doc in rides collection & convert the result into an array
41     res.status(200).json(rides); // sends the ride back to client as JSON with status 200 OK
42
43   } catch (err) {
44     res.status(500).json({ error: "Failed to fetch rides" }); // If error happens ( like db issues), it sends a 500 internal server error
45   }
46 });
47
48 // POST /rides - Create a new ride
49
50 app.post('/rides', async (req, res) => { // Hnadles POST req to create a new ride
51   try {
52     const result = await db.collection('rides').insertOne(req.body); // Inserts data from the (req.body) into the ride collection
53     res.status(201).json({ id: result.insertedId }) // send back ID of the newly created ride
54   } catch (err) {
55     res.status(400).json({ error: "Invalid ride data" }); // If something goes wrong ( missing / bad data ) it returns a 400 Bad req
56   }
57 });
58
59 // PATCH /rides/:id - Update ride status
60
61 app.patch('/rides/:id', async (req, res) => { // Handles PATCH req to update a ride by its ID
62   try {
63     const result = await db.collection('rides').updateOne( // looks for a ride with the matching ID & updates the status field
64       {
65         _id: new ObjectId(req.params.id),
66         $set: { status: req.body.status }
67       },
68     );
69
70     if (result.modifiedCount === 0) { // if no ride was updated, it returns 404 Not Found
71       return res.status(404).json({ error: "Ride not found" });
72     }
73     res.status(200).json({ updated: result.modifiedCount }); // if successful it returns how many rides were updated (usually 1)
74   } catch (err) { // Catches errors like an invalid ID format / bad req data
75
76
77

```

```

78   // Handle invalid ID format or DB errors
79   res.status(400).json({ error: "Invalid ride ID or data" });
80 }
81 });
82
83 // DELETE /rides/:id - Cancel a ride
84
85 app.delete('/rides/:id', async (req, res) => { // Handles DELETE req to remove a ride by ID
86   try {
87     const result = await db.collection('rides').deleteOne( // Deletes the ride with the matching ID from the db
88       { _id: new ObjectId(req.params.id) }
89     );
90
91     if (result.deletedCount === 0) { // if nothing was deleted, the ride probably didn't exist - return 404
92       return res.status(404).json({ error: "Ride not found" });
93     }
94     res.status(200).json({ deleted: result.deletedCount }); // on succes, res with hoe many rides were deleted
95   } catch (err) { // catch & return 400 Bad req for invalid IDs / other errors
96     res.status(400).json({ error: "Invalid ride ID" });
97   }
98 }
99 });
100

```

Github

GitHub repository view for `berr2243-25` (branch: `week 3`).

Files:

- node_modules
- week 3
 - week3 exercise.postman_collection...
 - week3.html
 - week3.js
- week2
 - .gitignore
 - README.md
 - index.js

Commit History:

Name	Last commit message	Last commit date
...		
week3 exercise.postman_collection.json	initial setup	38 minutes ago
week3.html	initial setup	38 minutes ago
week3.js	initial setup	38 minutes ago

2. Postman Collection:

- Export and include the collection file.

3. Exercise Report:

- Screenshots of Postman requests/responses.

Answer:

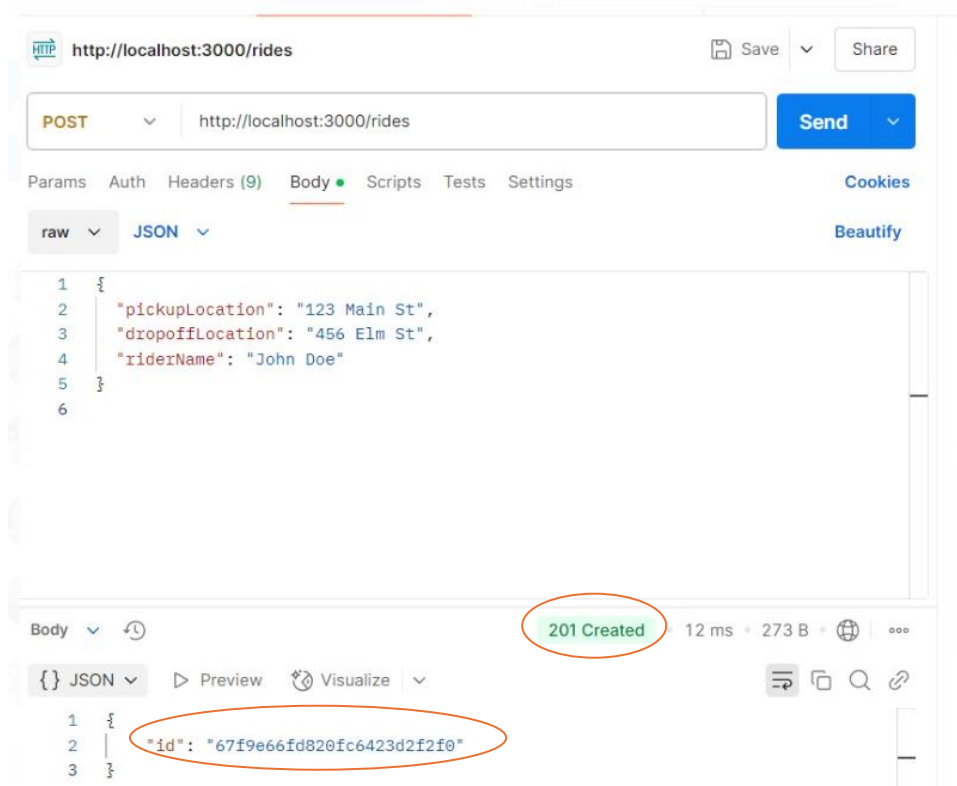
1. POST Request:

- What HTTP status code is returned when a ride is created successfully?

-201 created

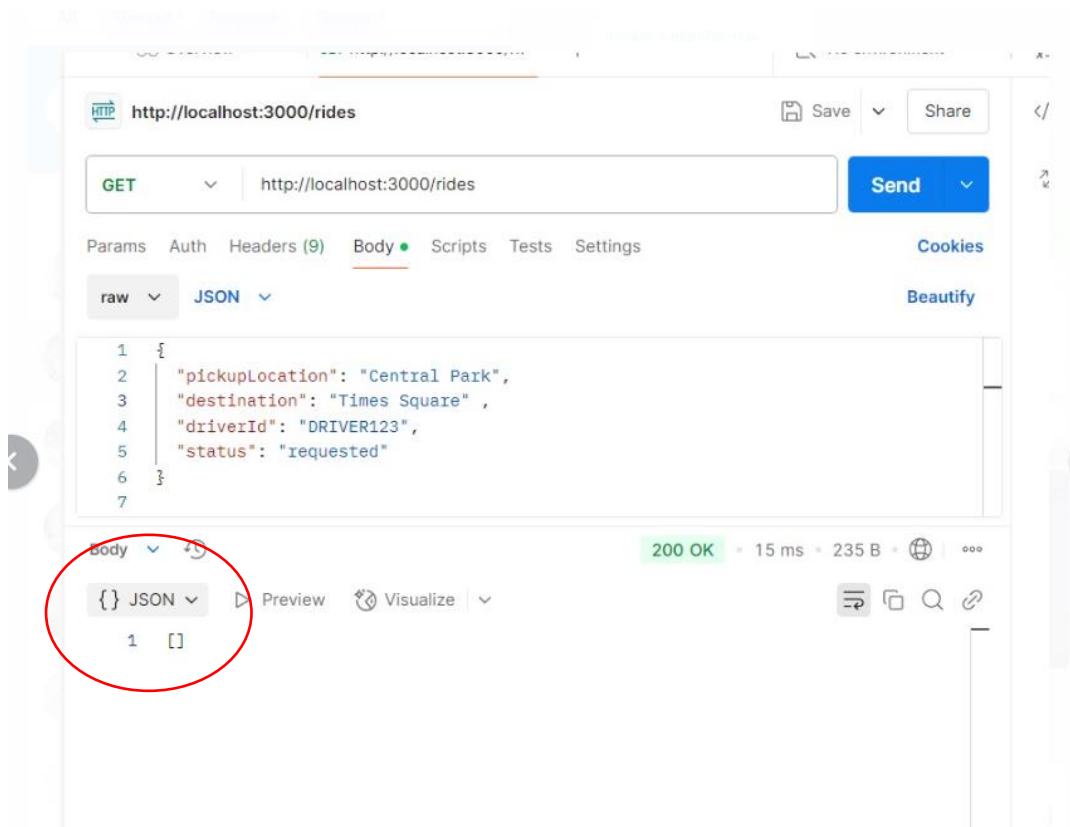
- What is the structure of the response body?

-`"id": "67f9e66fd820fc6423d2f2f0"`



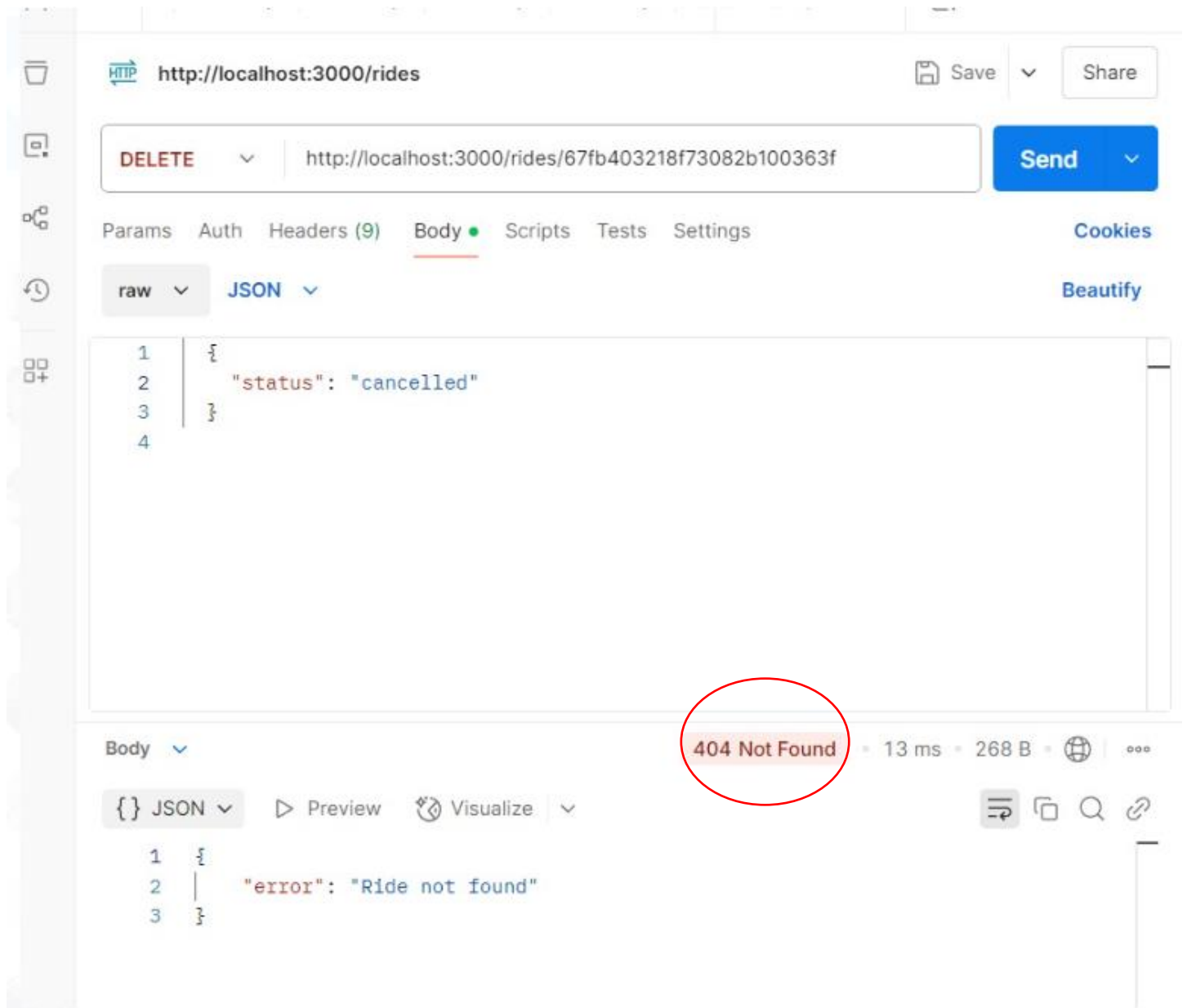
2. GET Request:

- What happens if the rides collection is empty?
- What data type is returned in the response (array/object)?



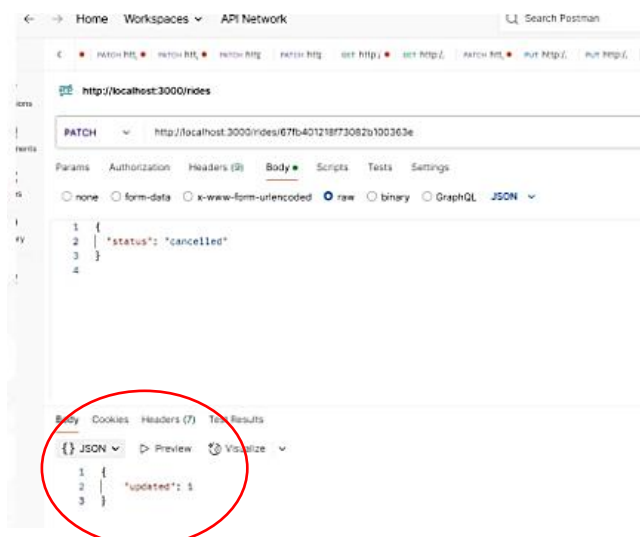
3. Fix PATCH and DELETE Error:

- Catch the error when requesting PATCH or DELETE API, then try to fix the issue reported.
- If you try to update a non-existent ride ID, what status code is returned?
-404 not found



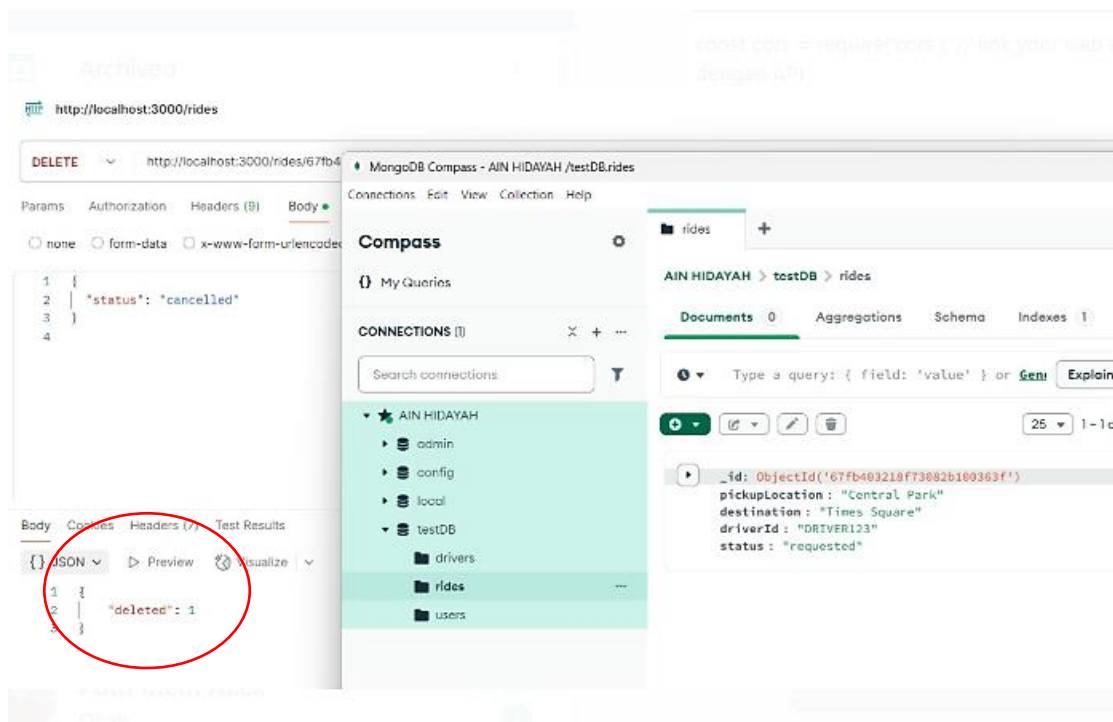
o What is the value of updated in the response if the update succeeds?

-if the output in postman exist "updated":1 that means update succes



- How does the API differentiate between a successful deletion and a failed one?

-successful deleted will be like this



4. Users Endpoints:

- Based on the exercise above, create the endpoints to handle the CRUD operations for users account

5. FrontEnd:

- Upload the Postman JSON to any AI tools, and generate a simple HTML and JS Dashboard for you

```

7
8 const app = express();
9 app.use(express.json());
10
11 const cors = require('cors');
12 app.use(cors());
13
14 let db;
15
16 async function connectToMongoDB() { // line 13 sampai 30 connect to mongoDB
17   const uri = "mongodb://localhost:27017/"
18   const client = new MongoClient(uri);
19
20   try {
21     await client.connect();
22     console.log("Connected to MongoDB !");
23
24     db = client.db("testDB");
25
26   } catch (err) {
27     console.error("Error:", err);
28   }
29 }
30 connectToMongoDB();

```

HTML Dashboard

Delma 2.0

Apps

Introducing ChatGPT | OpenAI

Dashboard Gah Gah

Ride Dashboard

Simple Ride Dashboard

C:/Users/User/OneDrive/Desktop/database/1AB1/week9/01/week3.html

All Bookmarks

Ride Dashboard

ID	Pickup	Destination	Driver	Status	Delete
67fb4cedbda723ad4ee0a7ea	central park	times square	DRIVER123	requested	Delete

Tambah Ride Baru

Pickup Location

Destination

Driver ID

Status

Harbor

Output in mongoDB

mpass - AIN HIDAYAH / TEST/US/NOES

it View Collection Help

rides

AIN HIDAYAH > testDB > rides

Documents 0 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or Generate query

ADD DATA

EXPORT DATA

UPDATE

DELETE

```
_id: ObjectId('67fb4cedbda723ad4ee0a7ea')
pickupLocation: "central park"
destination: "times square"
driverId: "DRIVER123"
status: "requested"
```

AIN HIDAYAH

in

fig

il

DB

trivers

ides

rsers