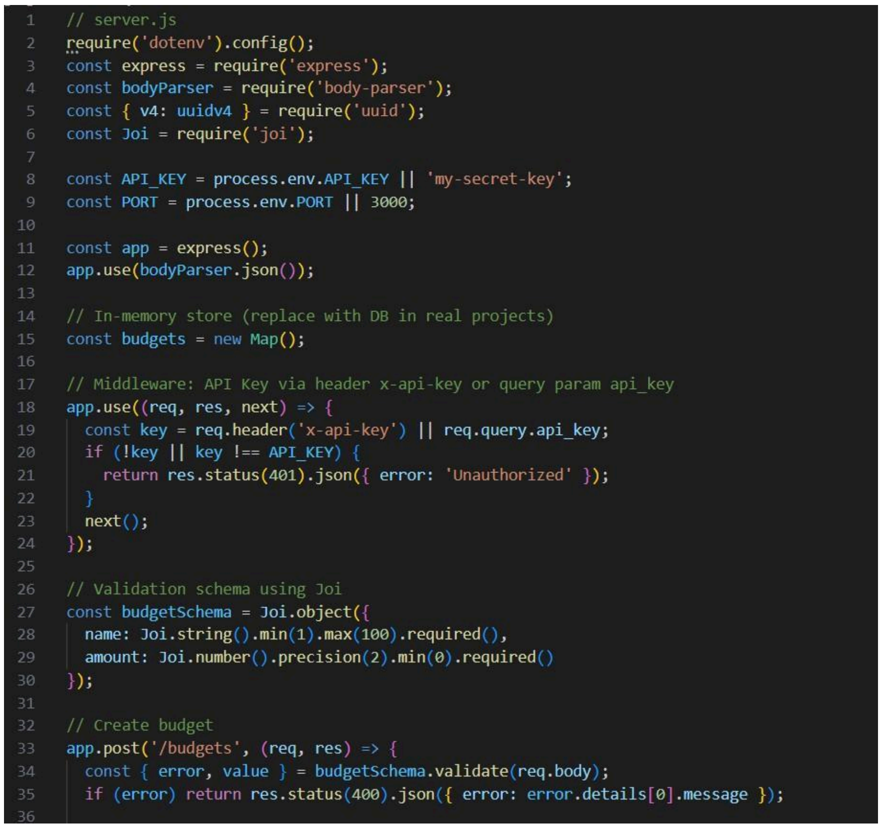
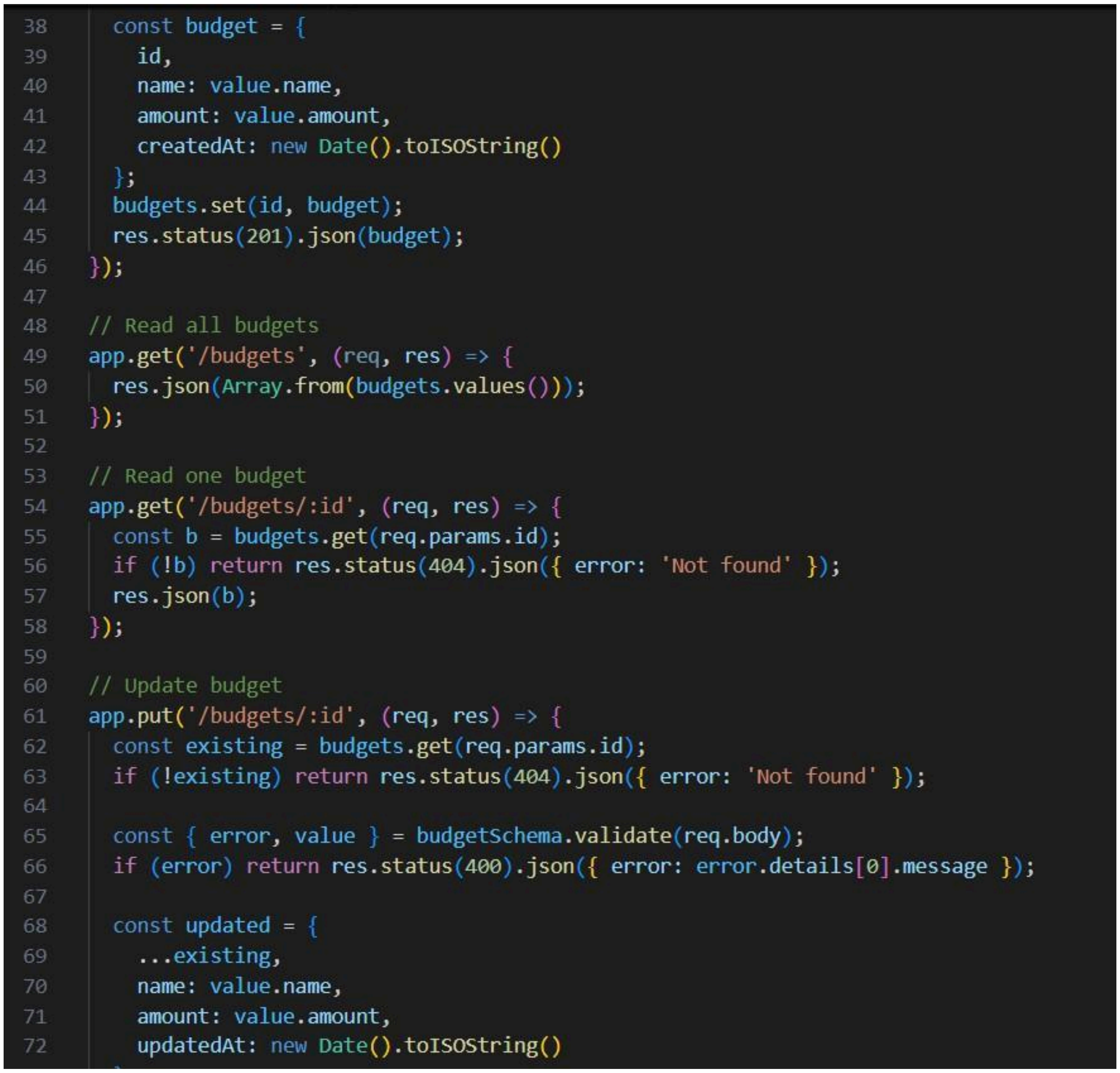
EXPERIMENT 7

Code:

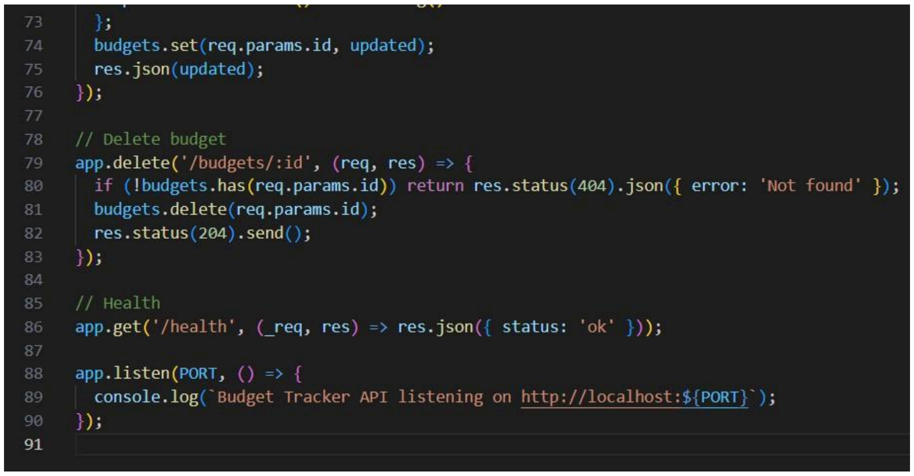
1) server.js



```javascript
// server.js
require('dotenv').config();
const express = require('express');
const bodyParser = require('body-parser');
const { v4: uuidv4 } = require('uuid');
const Joi = require('joi');

const API_KEY = process.env.API_KEY || 'my-secret-key';
const PORT = process.env.PORT || 3000;

const app = express();
app.use(bodyParser.json());

// In-memory store (replace with DB in real projects)
const budgets = new Map();

// Middleware: API Key via header x-api-key or query param api_key
app.use((req, res, next) => {
  const key = req.header('x-api-key') || req.query.api_key;
  if (!key || key !== API_KEY) {
    return res.status(401).json({ error: 'Unauthorized' });
  }
  next();
});

// Validation schema using Joi
const budgetSchema = Joi.object({
  name: Joi.string().min(1).max(100).required(),
  amount: Joi.number().precision(2).min(0).required()
});

// Create budget
app.post('/budgets', (req, res) => {
  const { error, value } = budgetSchema.validate(req.body);
  if (error) return res.status(400).json({ error: error.details[0].message });

```

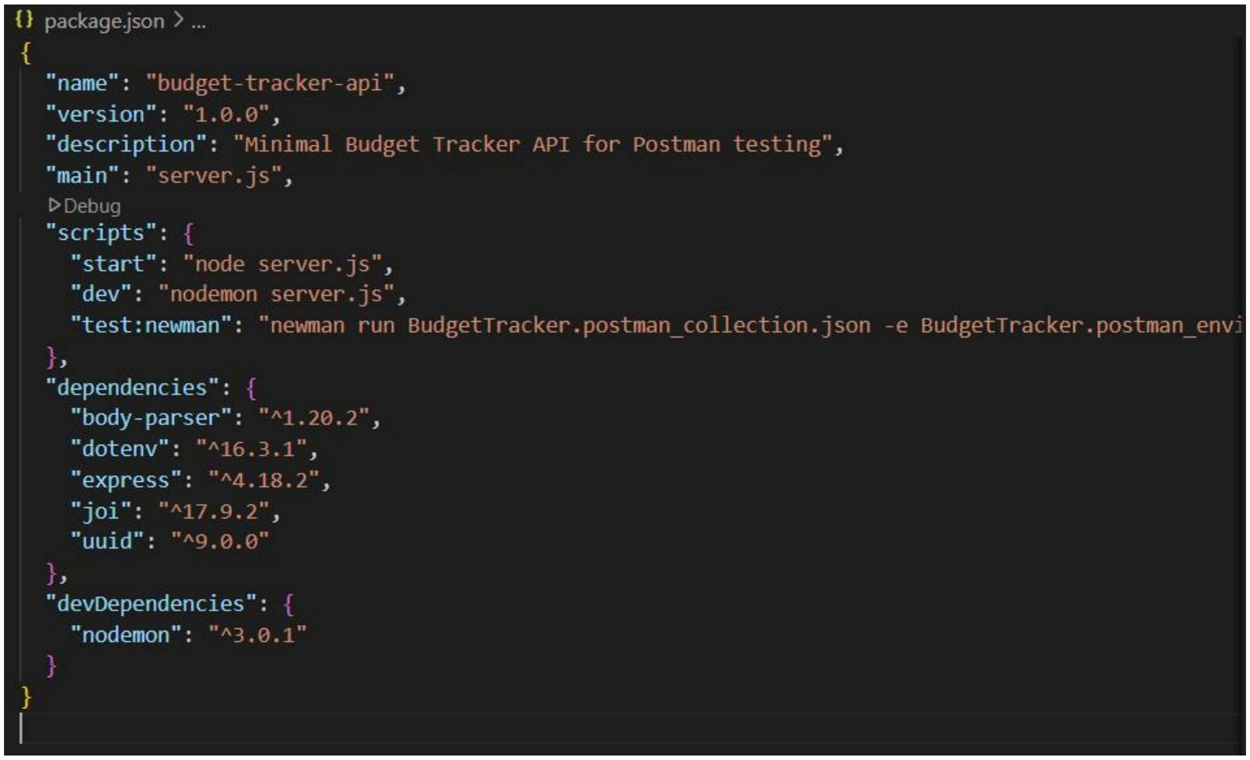


```
  const budget = {
    id,
    name: value.name,
    amount: value.amount,
    createdAt: new Date().toISOString()
  };
  budgets.set(id, budget);
  res.status(201).json(budget);
});

// Read all budgets
app.get('/budgets', (req, res) => {
  res.json(Array.from(budgets.values()));
});

// Read one budget
app.get('/budgets/:id', (req, res) => {
  const b = budgets.get(req.params.id);
  if (!b) return res.status(404).json({ error: 'Not found' });
  res.json(b);
});

// Update budget
app.put('/budgets/:id', (req, res) => {
  const existing = budgets.get(req.params.id);
  if (!existing) return res.status(404).json({ error: 'Not found' });

  const { error, value } = budgetSchema.validate(req.body);
  if (error) return res.status(400).json({ error: error.details[0].message });

  const updated = {
    ...existing,
    name: value.name,
    amount: value.amount,
    updatedAt: new Date().toISOString()
```



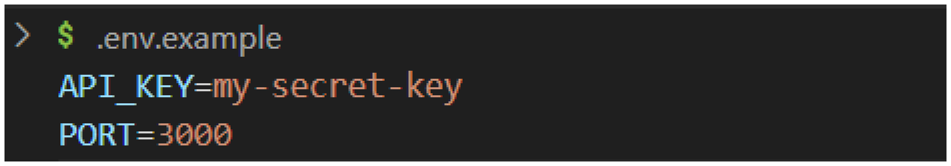
```
  };
  budgets.set(req.params.id, updated);
  res.json(updated);
});

// Delete budget
app.delete('/budgets/:id', (req, res) => {
  if (!budgets.has(req.params.id)) return res.status(404).json({ error: 'Not found' });
  budgets.delete(req.params.id);
  res.status(204).send();
});

// Health
app.get('/health', (_req, res) => res.json({ status: 'ok' }));

app.listen(PORT, () => {
  console.log(`Budget Tracker API listening on http://localhost:${PORT}`);
});
```

2) package.json



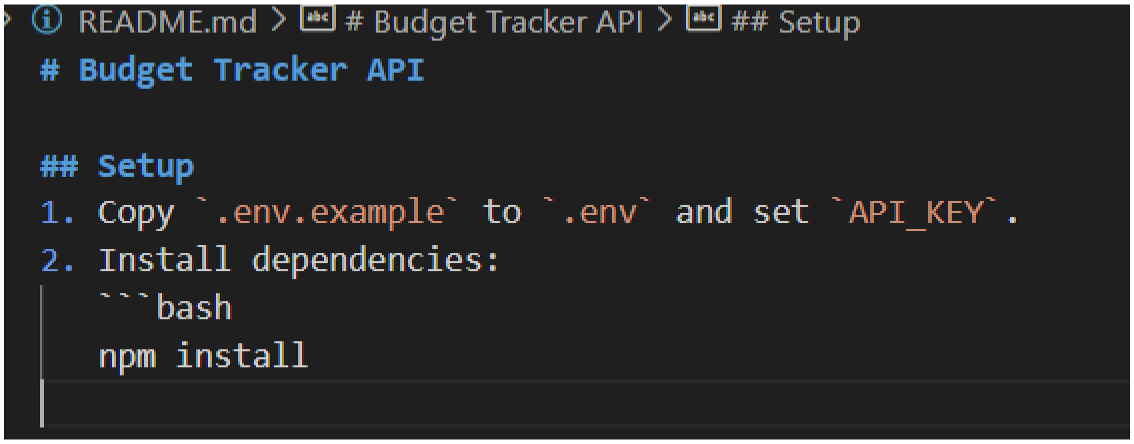
```
{} package.json > ...
{
  "name": "budget-tracker-api",
  "version": "1.0.0",
  "description": "Minimal Budget Tracker API for Postman testing",
  "main": "server.js",
  ▷Debug
  "scripts": {
    "start": "node server.js",
    "dev": "nodemon server.js",
    "test:newman": "newman run BudgetTracker.postman_collection.json -e BudgetTracker.postman_envi
  },
  "dependencies": {
    "body-parser": "^1.20.2",
    "dotenv": "^16.3.1",
    "express": "^4.18.2",
    "joi": "^17.9.2",
    "uuid": "^9.0.0"
  },
  "devDependencies": {
    "nodemon": "^3.0.1"
  }
}
```

3) .env

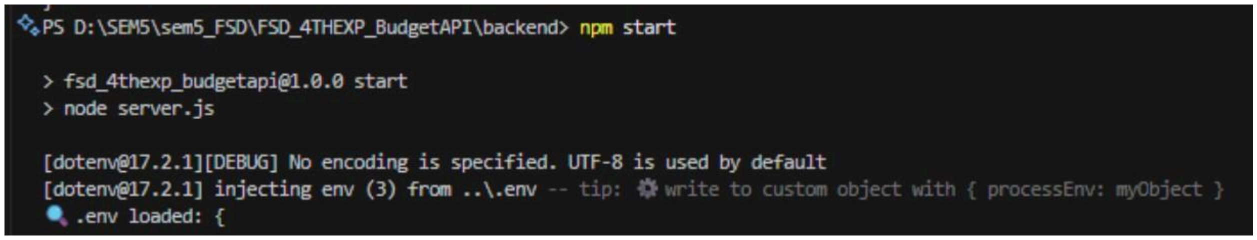


```
> $ .env.example
API_KEY=my-secret-key
PORT=3000
```

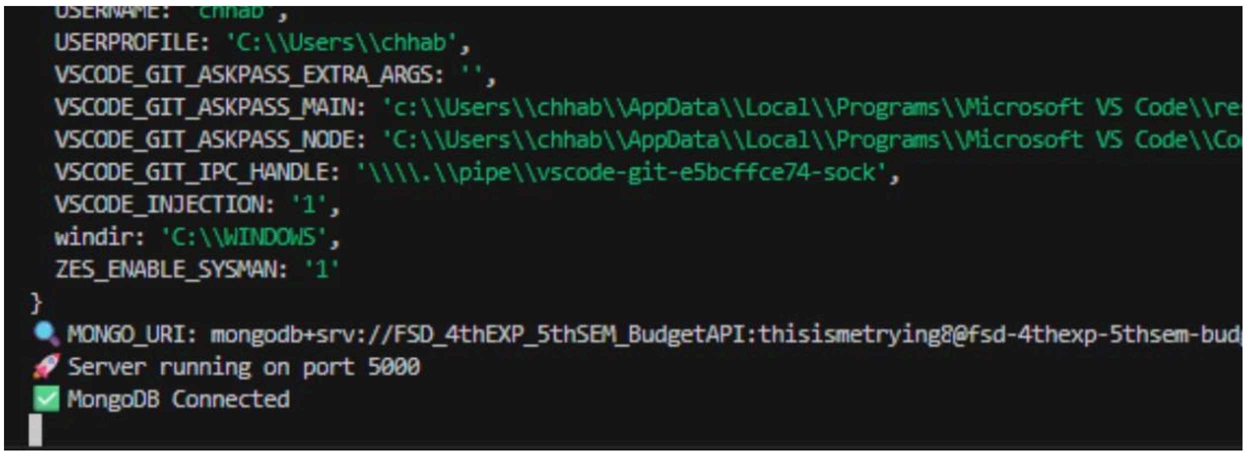
4) README.md



~~~
> ⓘ README.md > # Budget Tracker API > ## Setup
# Budget Tracker API

## Setup
1. Copy `.env.example` to `.env` and set `API_KEY`.
2. Install dependencies:
   ```bash
   npm install
~~~

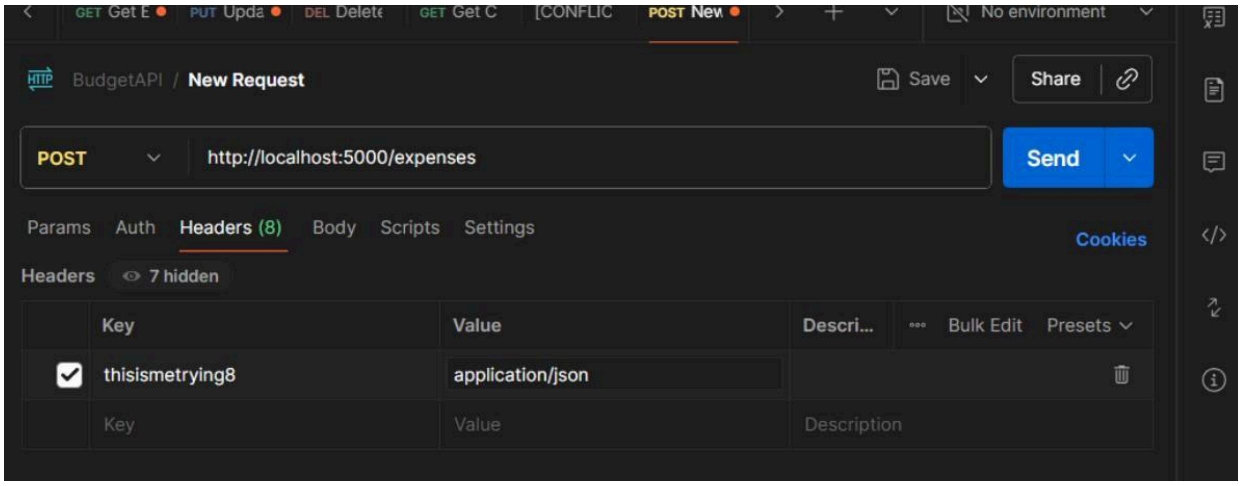


```
PS D:\SEM5\sem5_FSD\FSD_4THEXP_BudgetAPI\backend> npm start

  > fsd_4thexp_budgetapi@1.0.0 start
  > node server.js

  [dotenv@17.2.1][DEBUG] No encoding is specified. UTF-8 is used by default
  [dotenv@17.2.1] injecting env (3) from ..\.env -- tip: ⚙ write to custom object with { processEnv: myObject }
  🔍 .env loaded: {
```



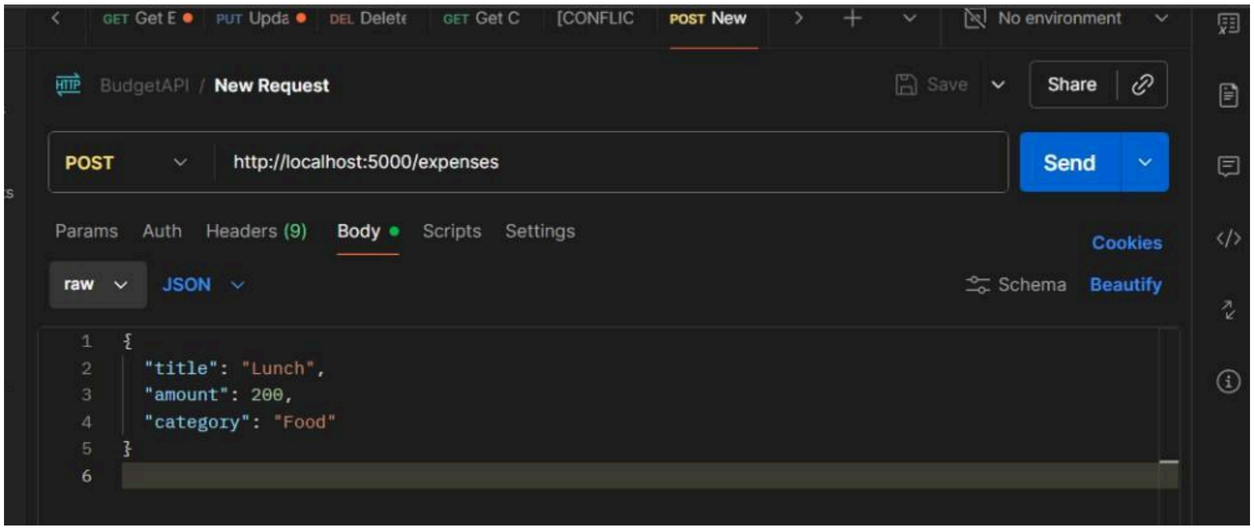
```
    USERNAME: 'chhab',
    USERPROFILE: 'C:\\Users\\chhab',
    VSCODE_GIT_ASKPASS_EXTRA_ARGS: '',
    VSCODE_GIT_ASKPASS_MAIN: 'c:\\Users\\chhab\\AppData\\Local\\Programs\\Microsoft VS Code\\re
    VSCODE_GIT_ASKPASS_NODE: 'C:\\Users\\chhab\\AppData\\Local\\Programs\\Microsoft VS Code\\Co
    VSCODE_GIT_IPC_HANDLE: '\\\\.\\pipe\\vscode-git-e5bcffce74-sock',
    VSCODE_INJECTION: '1',
    windir: 'C:\\WINDOWS',
    ZES_ENABLE_SYSMAN: '1'
  }
  🔍 MONGO_URI: mongodb+srv://FSD_4thEXP_5thSEM_BudgetAPI:thisismetrying8@fsd-4thexp-5thsem-bud
  🚀 Server running on port 5000
  ✅ MongoDB Connected
```

Implementation Steps with Postman



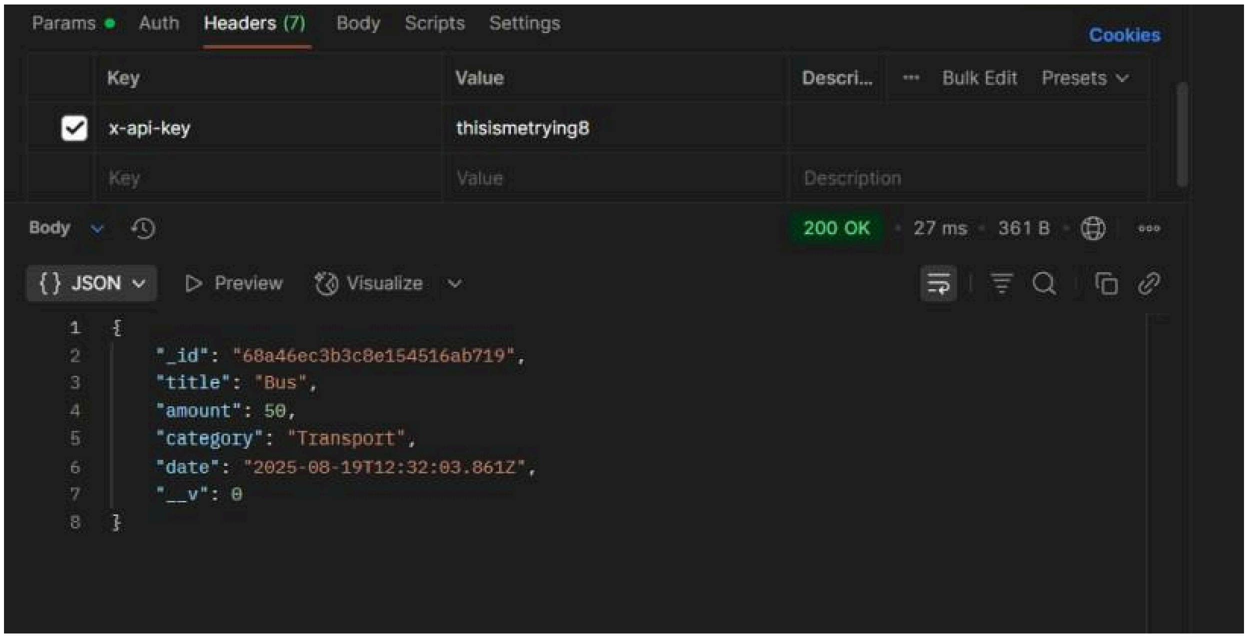
POST http://localhost:5000/expenses

Key	Value	Descri...
✓ thisismetrying8	application/json	
Key	Value	Description



POST http://localhost:5000/expenses

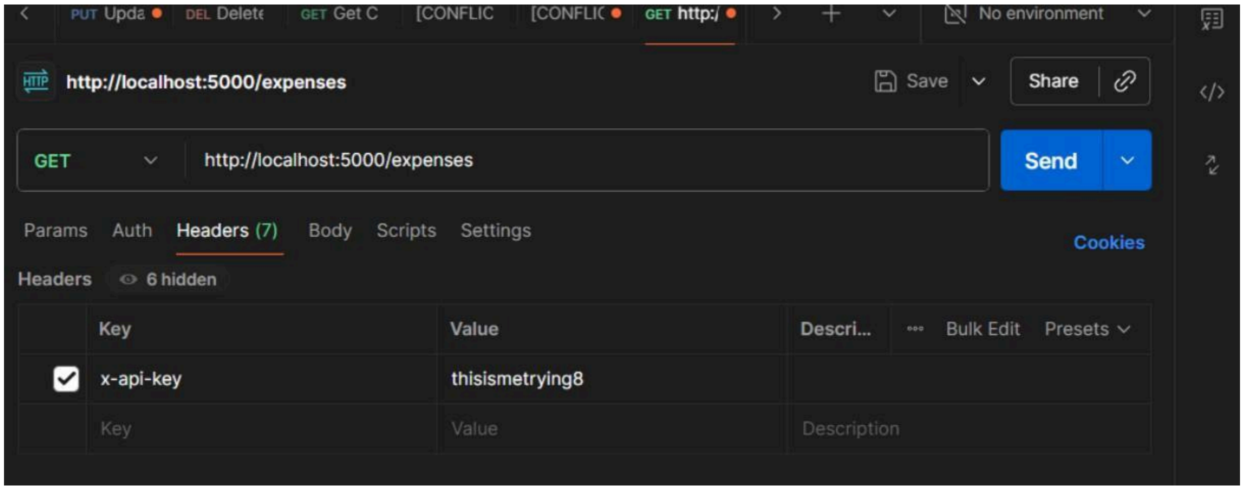
```
{
  "title": "Lunch",
  "amount": 200,
  "category": "Food"
}
```


Params Auth Headers (7) Body Scripts Settings Cookies

	Key	Value	Description
✓	x-api-key	thisismetrying8	

Body 200 OK 27 ms 361 B JSON Preview Visualize

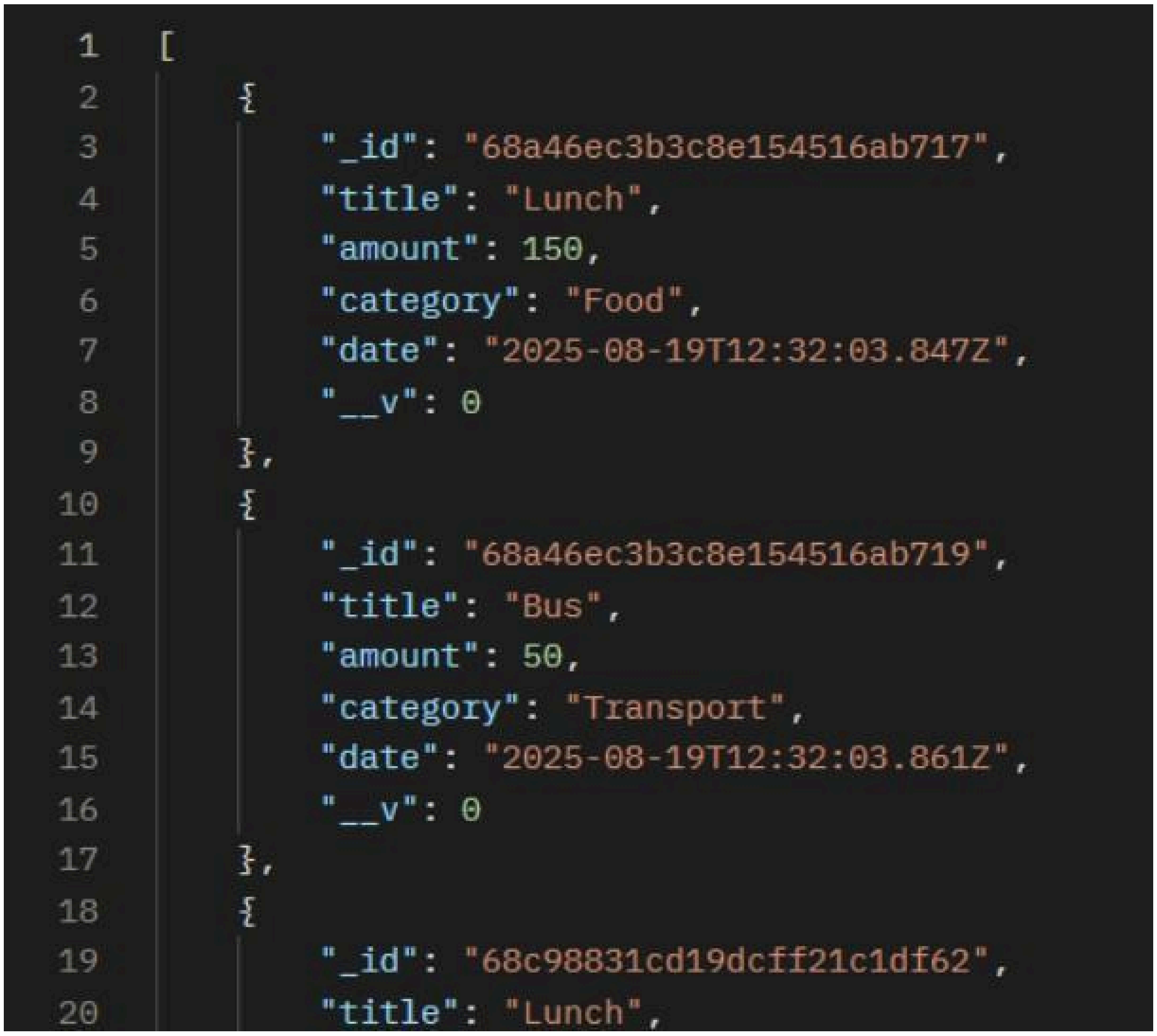
```
{
    "_id": "68a46ec3b3c8e154516ab719",
    "title": "Bus",
    "amount": 50,
    "category": "Transport",
    "date": "2025-08-19T12:32:03.861Z",
    "__v": 0
}
```


PUT Upda DEL Delete GET Get C [CONFLIC [CONFLIC GET http:/ No environment
Save Share
Send
Params Auth Headers (7) Body Scripts Settings Cookies
Headers 6 hidden

http://localhost:5000/expenses

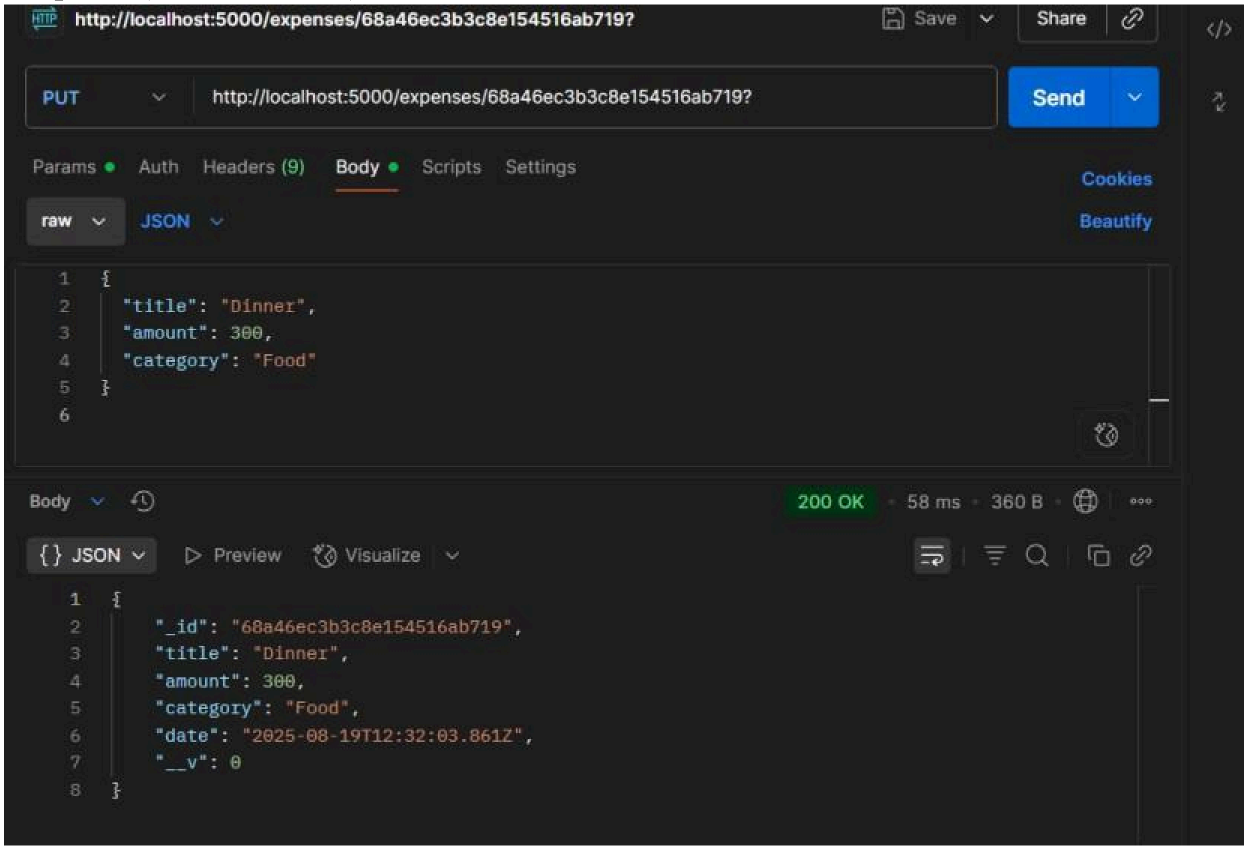
GET http://localhost:5000/expenses

	Key	Value	Description
✓	x-api-key	thisismetrying8	



```
[
    {
        "_id": "68a46ec3b3c8e154516ab717",
        "title": "Lunch",
        "amount": 150,
        "category": "Food",
        "date": "2025-08-19T12:32:03.847Z",
        "__v": 0
    },
    {
        "_id": "68a46ec3b3c8e154516ab719",
        "title": "Bus",
        "amount": 50,
        "category": "Transport",
        "date": "2025-08-19T12:32:03.861Z",
        "__v": 0
    },
    {
        "_id": "68c98831cd19dcff21c1df62",
        "title": "Lunch",
```

1. Update (PUT / PATCH)



http://localhost:5000/expenses/68a46ec3b3c8e154516ab719?

PUT http://localhost:5000/expenses/68a46ec3b3c8e154516ab719?

Params Auth Headers (9) Body Scripts Settings

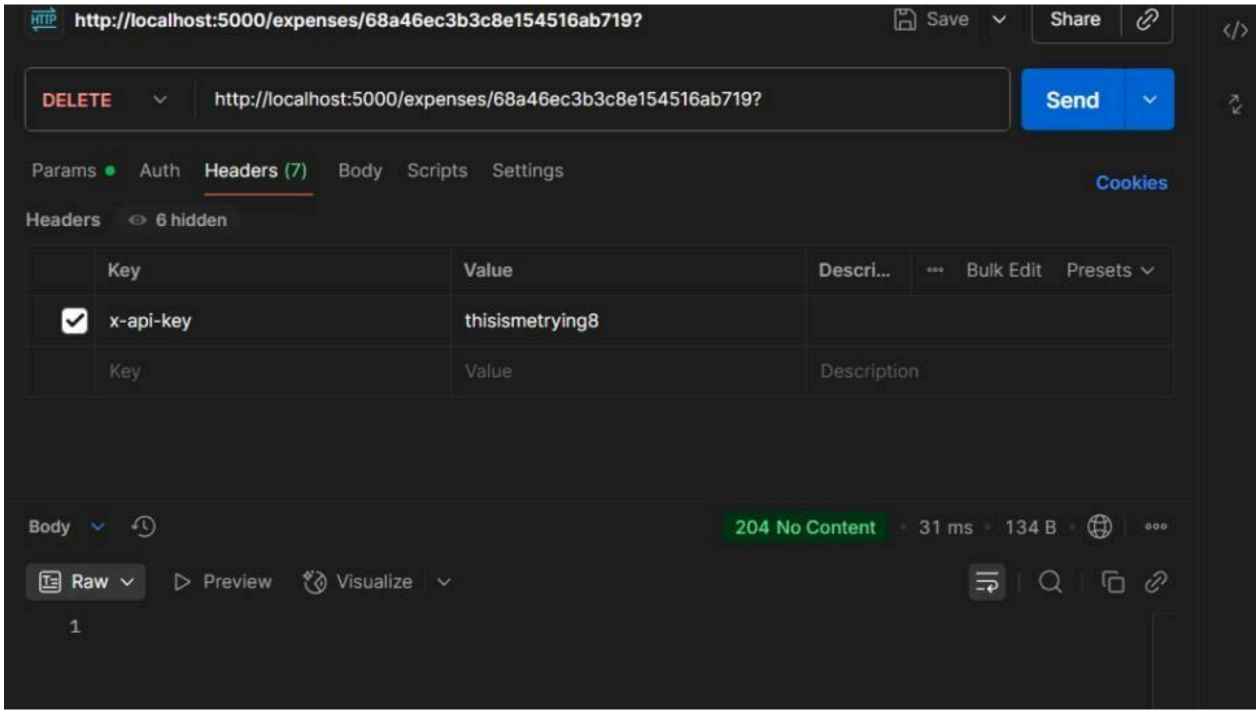
raw JSON

```
{
  "title": "Dinner",
  "amount": 300,
  "category": "Food"
}
```

Body 200 OK · 58 ms · 360 B

```
{
    "_id": "68a46ec3b3c8e154516ab719",
    "title": "Dinner",
    "amount": 300,
    "category": "Food",
    "date": "2025-08-19T12:32:03.861Z",
    "__v": 0
}
```

2. Delete



http://localhost:5000/expenses/68a46ec3b3c8e154516ab719?

DELETE http://localhost:5000/expenses/68a46ec3b3c8e154516ab719?

Params Auth Headers (7) Body Scripts Settings

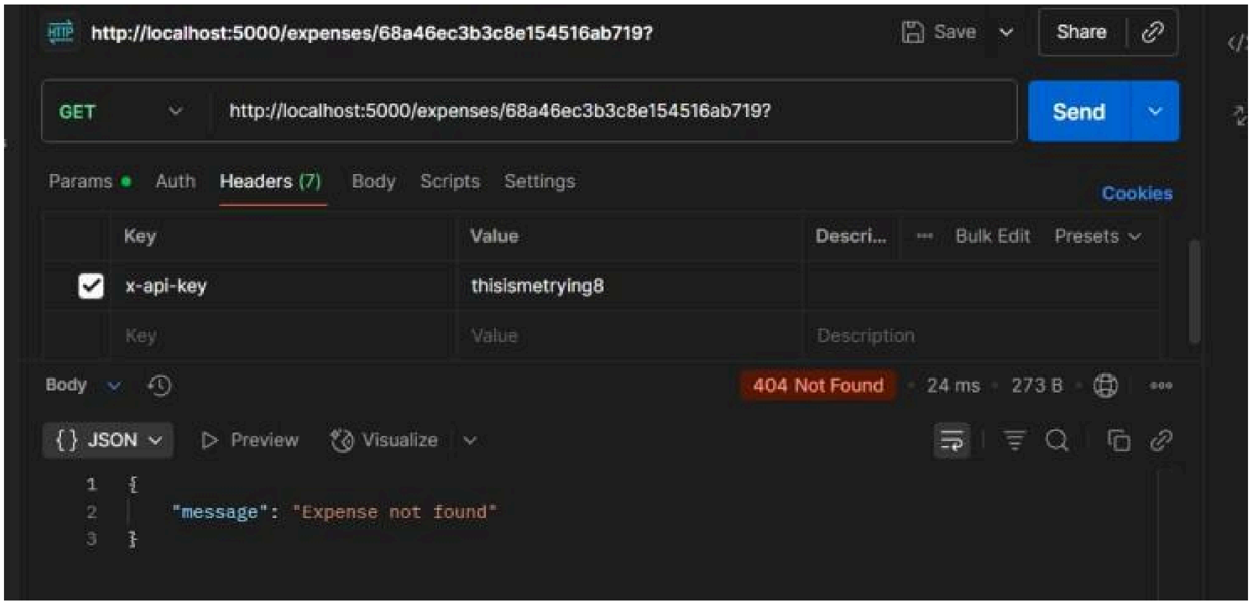
Headers 6 hidden

	Key	Value	Description
✓	x-api-key	thisismetrying8	

Body 204 No Content · 31 ms · 134 B

Raw

```
1
```



http://localhost:5000/expenses/68a46ec3b3c8e154516ab719?

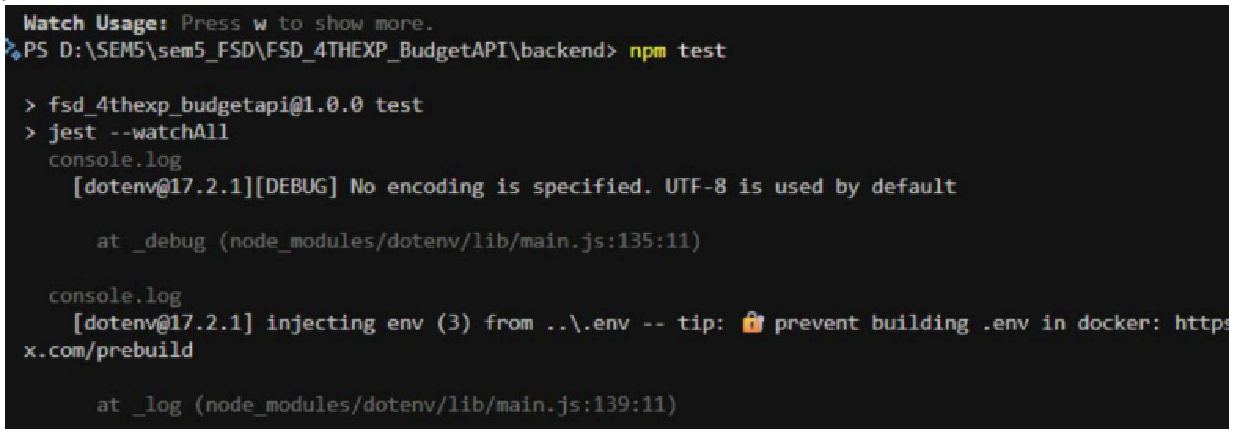
GET http://localhost:5000/expenses/68a46ec3b3c8e154516ab719?

Key	Value	Descri...
✓ x-api-key	thisismetrying8	
Key	Value	Description

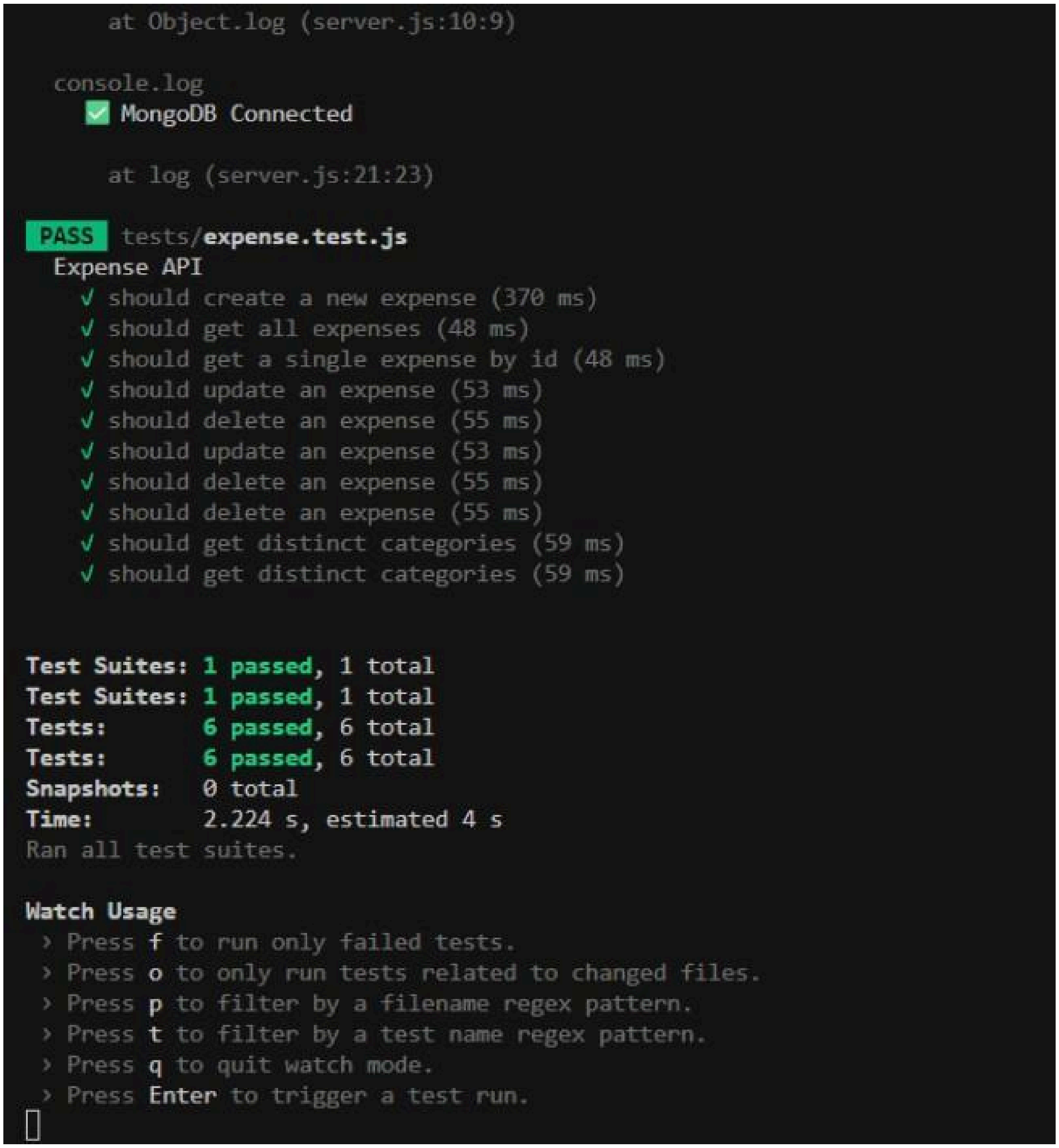
Body 404 Not Found · 24 ms · 273 B

```
{
    "message": "Expense not found"
}
```

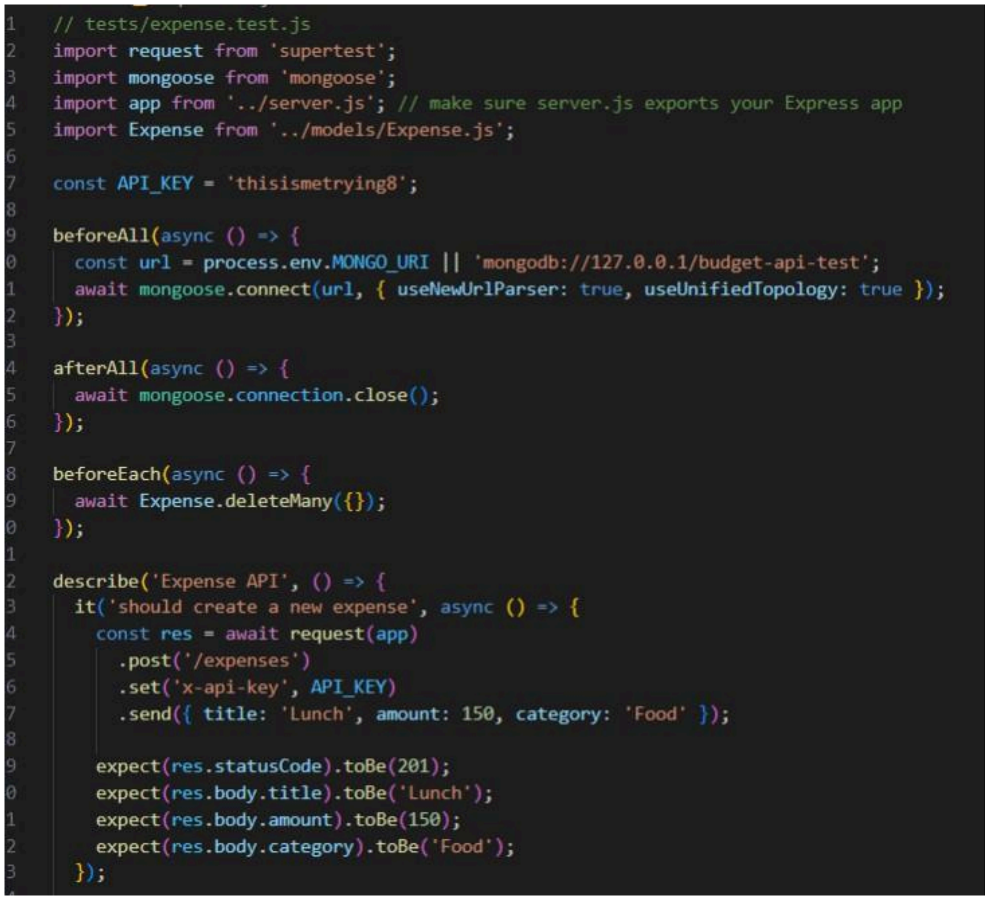
30% extra:(automated testing)



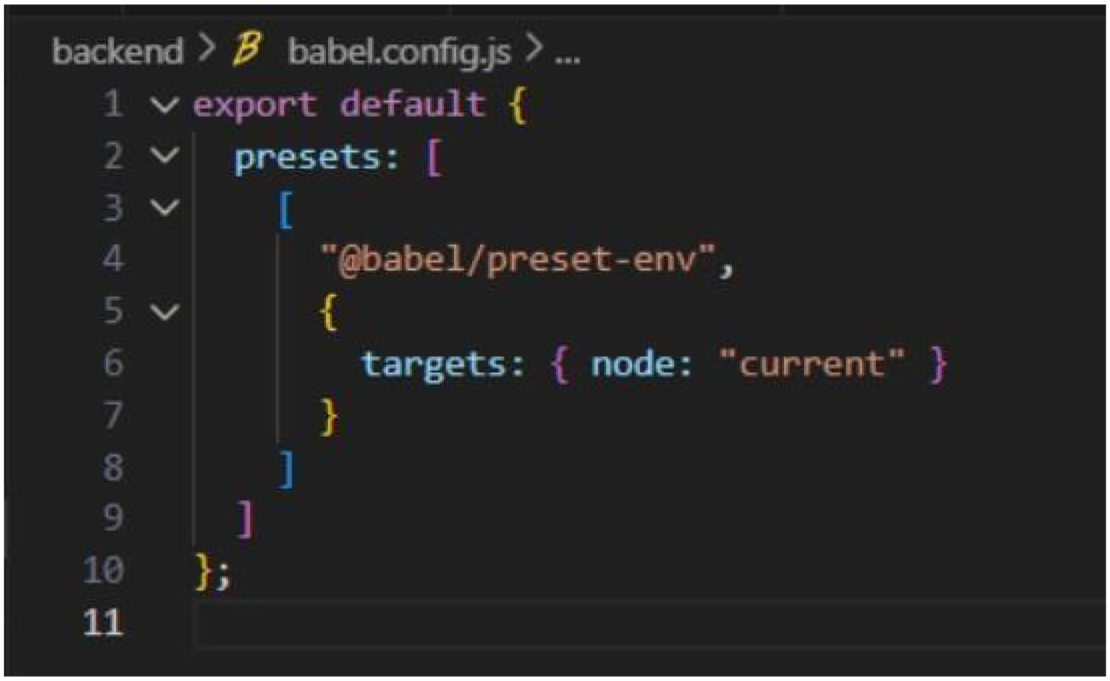
```
Watch Usage: Press w to show more.
PS D:\SEM5\sem5_FSD\FSD_4THEXP_BudgetAPI\backend> npm test

> fsd_4thexp_budgetapi@1.0.0 test
> jest --watchAll
  console.log
    [dotenv@17.2.1][DEBUG] No encoding is specified. UTF-8 is used by default

      at _debug (node_modules/dotenv/lib/main.js:135:11)

  console.log
    [dotenv@17.2.1] injecting env (3) from ..\.env -- tip: 🔐 prevent building .env in docker: https
x.com/prebuild

      at _log (node_modules/dotenv/lib/main.js:139:11)
```



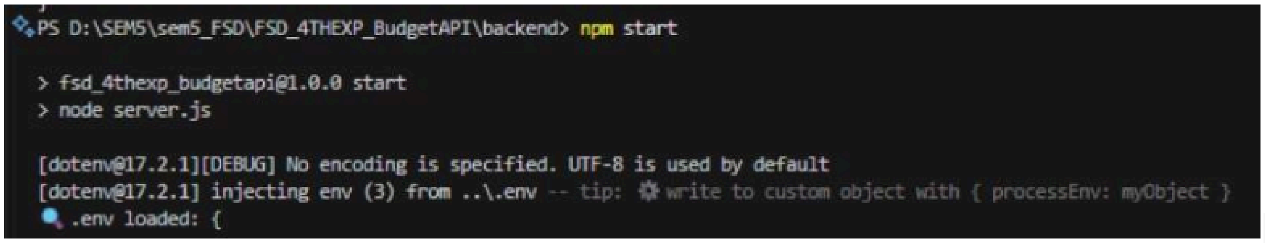
```
      at Object.log (server.js:10:9)

    console.log
      ✅ MongoDB Connected

      at log (server.js:21:23)

 PASS  tests/expense.test.js
  Expense API
    √ should create a new expense (370 ms)
    √ should get all expenses (48 ms)
    √ should get a single expense by id (48 ms)
    √ should update an expense (53 ms)
    √ should delete an expense (55 ms)
    √ should update an expense (53 ms)
    √ should delete an expense (55 ms)
    √ should delete an expense (55 ms)
    √ should get distinct categories (59 ms)
    √ should get distinct categories (59 ms)


Test Suites: 1 passed, 1 total
Test Suites: 1 passed, 1 total
Tests:       6 passed, 6 total
Tests:       6 passed, 6 total
Snapshots:   0 total
Time:        2.224 s, estimated 4 s
Ran all test suites.

Watch Usage
 › Press f to run only failed tests.
 › Press o to only run tests related to changed files.
 › Press p to filter by a filename regex pattern.
 › Press t to filter by a test name regex pattern.
 › Press q to quit watch mode.
 › Press Enter to trigger a test run.
▯
```



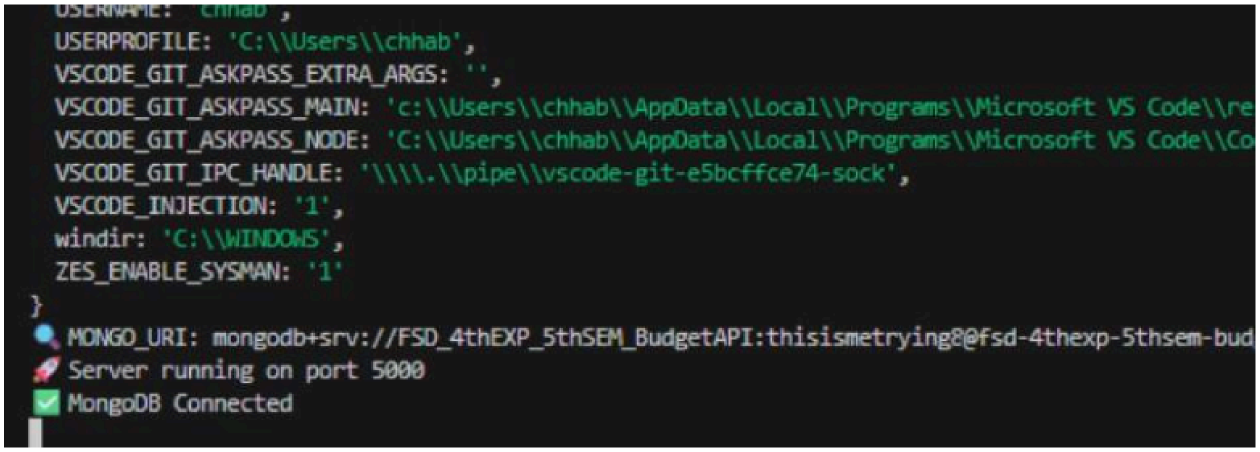
```
// tests/expense.test.js
import request from 'supertest';
import mongoose from 'mongoose';
import app from '../server.js'; // make sure server.js exports your Express app
import Expense from '../models/Expense.js';

const API_KEY = 'thisismetrying8';

beforeAll(async () => {
  const url = process.env.MONGO_URI || 'mongodb://127.0.0.1/budget-api-test';
  await mongoose.connect(url, { useNewUrlParser: true, useUnifiedTopology: true });
});

afterAll(async () => {
  await mongoose.connection.close();
});

beforeEach(async () => {
  await Expense.deleteMany({});
});

describe('Expense API', () => {
  it('should create a new expense', async () => {
    const res = await request(app)
      .post('/expenses')
      .set('x-api-key', API_KEY)
      .send({ title: 'Lunch', amount: 150, category: 'Food' });

    expect(res.statusCode).toBe(201);
    expect(res.body.title).toBe('Lunch');
    expect(res.body.amount).toBe(150);
    expect(res.body.category).toBe('Food');
  });
```



```
backend > babel.config.js > ...
export default {
  presets: [
    [
      "@babel/preset-env",
      {
        targets: { node: "current" }
      }
    ]
  ]
};

```



```
PS D:\SEM5\sem5_FSD\FSD_4THEXP_BudgetAPI\backend> npm start

 > fsd_4thexp_budgetapi@1.0.0 start
 > node server.js

 [dotenv@17.2.1][DEBUG] No encoding is specified. UTF-8 is used by default
 [dotenv@17.2.1] injecting env (3) from ..\.env -- tip: ⚙ write to custom object with { processEnv: myObject }
 🔍 .env loaded: {
```



```
   USERNAME: 'chhab',
   USERPROFILE: 'C:\\Users\\chhab',
   VSCODE_GIT_ASKPASS_EXTRA_ARGS: '',
   VSCODE_GIT_ASKPASS_MAIN: 'c:\\Users\\chhab\\AppData\\Local\\Programs\\Microsoft VS Code\\re
   VSCODE_GIT_ASKPASS_NODE: 'C:\\Users\\chhab\\AppData\\Local\\Programs\\Microsoft VS Code\\Co
   VSCODE_GIT_IPC_HANDLE: '\\\\.\\pipe\\vscode-git-e5bcffce74-sock',
   VSCODE_INJECTION: '1',
   windir: 'C:\\WINDOWS',
   ZES_ENABLE_SYSMAN: '1'
 }
 🔍 MONGO_URI: mongodb+srv://FSD_4thEXP_5thSEM_BudgetAPI:thisismetrying8@fsd-4thexp-5thsem-bud
 🚀 Server running on port 5000
 ✅ MongoDB Connected
```