LAB-3

* STEP 1: - mkdir bookmart
* STEP 2: -cd bookmart
* STEP 3: -npm init –y
* STEP 4: -npm install express body-parser
* STEP 5: -Create a new file named as index.js (in bookmart folder).
* STEP 6: -Copy the below code to index.js.

const express = require('express');

const bodyParser = require('body-parser');

const app = express();

const port = 3000;

// In-memory array to store book data

let books = [

  { id: 1, title: 'The Great Gatsby', author: 'F. Scott Fitzgerald', price: 15.99 },

  { id: 2, title: 'To Kill a Mockingbird', author: 'Harper Lee', price: 12.99 },

  // Add more books as needed

];

app.use(bodyParser.json());

// Welcome message for the root URL

app.get('/', (req, res) => {

  res.send('Welcome to the Book Cart API!');

});

// Get all books

app.get('/books', (req, res) => {

  res.json(books);

});

// Get a specific book by ID

app.get('/books/:id', (req, res) => {

  const bookId = parseInt(req.params.id);

  const book = books.find(b => b.id === bookId);

  if (!book) {

    res.status(404).json({ error: 'Book not found' });

  } else {

    res.json(book);

  }

});

// Add a new book

app.post('/books', (req, res) => {

  const newBook = req.body;

  newBook.id = books.length + 1;

  books.push(newBook);

  res.status(201).json(newBook);

});

// Update a book by ID

app.put('/books/:id', (req, res) => {

  const bookId = parseInt(req.params.id);

  const updatedBook = req.body;

  const index = books.findIndex(b => b.id === bookId);

  if (index === -1) {

    res.status(404).json({ error: 'Book not found' });

  } else {

    books[index] = { ...books[index], ...updatedBook };

    res.json(books[index]);

  }

});

// Delete a book by ID

app.delete('/books/:id', (req, res) => {

  const bookId = parseInt(req.params.id);

  books = books.filter(b => b.id !== bookId);

  res.json({ message: 'Book deleted successfully' });

});

app.listen(port, () => {

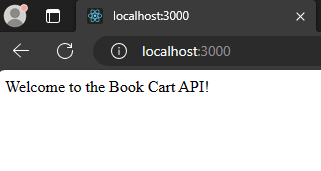
  console.log(`Server is running at http://localhost:${port}`);

});

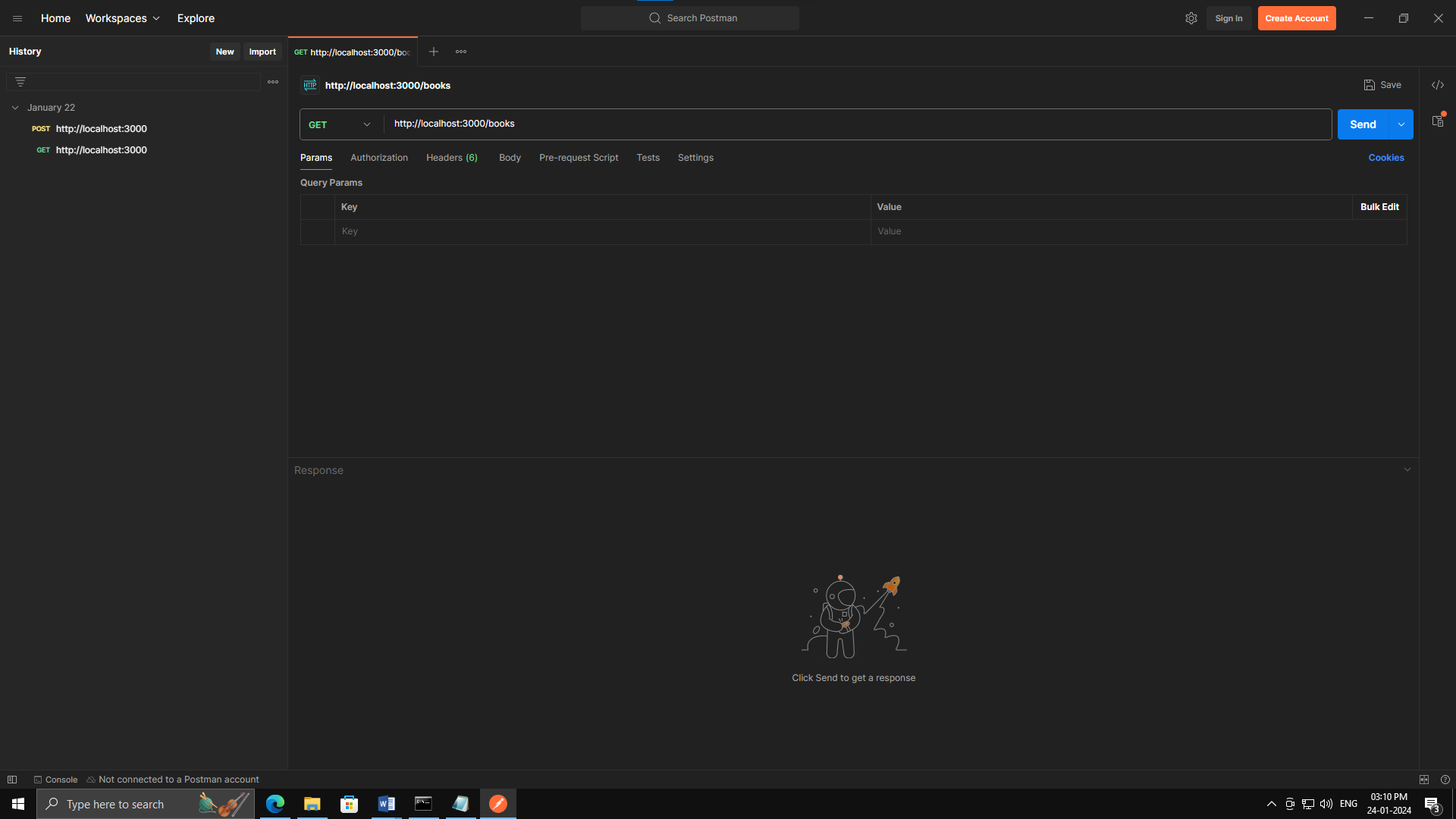
* STEP 7: - node index.js.



* STEP 8: -enter the link in browser.



* STEP 9: - open postman application and enter the link https:localhost:3000



* GET Request (Fetch All Books):

Open Postman.

Set the request type to GET.

Enter the URL: http://localhost:3000/books.

Click the "Send" button.

This should retrieve all books from your Node.js application.

* GET Request (Fetch a Specific Book):

Set the request type to GET.

Enter the URL: http://localhost:3000/books/1 (replace "1" with the ID of an existing book).

Click the "Send" button.

This should retrieve the book with the specified ID.

* POST Request (Add a New Book):

Set the request type to POST

Enter the URL: http://localhost:3000/books.

Go to the "Body" tab, select raw, and enter the JSON for a new book:

json

Copy code

{

"title": "New Book",

"author": "New Author",

"price": 19.99

}

Click the "Send" button.

This should add a new book to your Node.js application.

* PUT Request (Update a Book):

Set the request type to PUT.

Enter the URL: http://localhost:3000/books/1 (replace "1" with the ID of an existing book).

Go to the "Body" tab, select raw, and enter the JSON for the updated book:

json

Copy code

{

"title": "Updated Book Title",

"author": "Updated Author",

"price": 24.99

}

Click the "Send" button.

This should update the book with the specified ID in your Node.js application.

* DELETE Request (Delete a Book):

Set the request type to DELETE.

Enter the URL: http://localhost:3000/books/1 (replace "1" with the ID of an existing book).

Click the "Send" button.

This should delete the book with the specified ID from your Node.js application.