**Customer Management API Documentation**

**Overview**

This API provides endpoints to manage customer data including creation, retrieval, update, and deletion of customer records in a SQL database.

**Base URL**

http://localhost:3000

Endpoints

**1. Create a new customer**

**URL**: **/api/customer**

**Method**: **POST**

**Request Body**: JSON

{

"firstName": "John",

"lastName": "Doe",

"email": "john.doe@example.com",

"mobileNumber": "+1234567890",

"address": "123 Main Street, City",

"pincode": "12345"

}

**Response**:

Success: 200 OK

{

"success": true,

"message": "Data inserted successfully"

}

Failure: 400 Bad Request or 500 Internal Server Error with error message

**2. Fetch all customers**

**URL**: **/api/customer**

**Method**: **GET**

**Response**:

Success: 200 OK with an array of customer objects

{

"success": true,

"customers": [

{

"Customer\_ID": 1,

"First\_Name": "John",

"Last\_Name": "Doe",

"Email": "john.doe@example.com",

"Mobile\_number": "+1234567890",

"Address": "123 Main Street, City",

"Pincode": "12345"

},

// Other customers...

]

}

Failure: 500 Internal Server Error with error message

**3. Update an existing customer**

**URL**: **/api/customer/:id**

**Method**: **PUT**

**URL Parameters**: **id** (Customer ID)

**Request Body**: JSON

{

"firstName": "Updated John",

"lastName": "Updated Doe",

"email": "updated.john.doe@example.com",

"mobileNumber": "+1234567890",

"address": "Updated Address",

"pincode": "54321"

}

**Response:**

Success: 200 OK

{

"success": true,

"message": "Data updated successfully"

}

Failure: 400 Bad Request or 500 Internal Server Error with error message

**4. Delete a customer**

**URL**: **/api/customer/:id**

**Method**: **DELETE**

**URL Parameters**: **id** (Customer ID)

**Response**:

Success: 200 OK

{

"success": true,

"message": "Data deleted successfully"

}

Failure: 500 Internal Server Error with error message

**Error Responses**

400 Bad Request: Missing or invalid request parameters

500 Internal Server Error: Server-side error occurred, check the error message for details

**Instruction:**

For now insert the data using postman app or any other app

**App.js**

const express = require('express');

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

const sql = require('mssql');

const app = express();

const port = process.env.PORT || 3000;

// Middleware for parsing JSON body

app.use(bodyParser.json());

// Configuration for SQL Server connection

const config = {

  user: 'root',

  password: '',

  server: 'localhost',

  database: 'smartconnect',

  options: {

    encrypt: true // If you're using Azure SQL Database

  }

};

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

  try {

    const { firstName, lastName, email, mobileNumber, address, pincode } = req.body;

    await sql.connect(config);

    const result = await sql.query(`EXEC InsertCustomer @FirstName='${firstName}', @LastName='${lastName}', @Email='${email}', @MobileNumber='${mobileNumber}', @Address='${address}', @Pincode='${pincode}'`);

    await sql.close();

    res.status(200).json({ message: 'Customer details inserted successfully.' });

  } catch (error) {

    console.error(error);

    res.status(200).json({ error: 'Failed to insert customer details.' });

  }

});

// GET method to fetch all customers

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

  const query = 'SELECT \* FROM CustomerDetails';

  db.query(query, (err, result) => {

    if (err) {

      console.error('Error fetching data:', err);

      return res.status(500).json({ success: false, message: 'Failed to fetch data from database' });

    }

    res.status(200).json({ success: true, customers: result });

  });

});

// PUT method to update a customer by ID

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

  const customerId = req.params.id;

  const { firstName, lastName, email, mobileNumber, address, pincode } = req.body;

  const query = 'UPDATE CustomerDetails SET First\_Name = ?, Last\_Name = ?, Email = ?, Mobile\_number = ?, Address = ?, Pincode = ? WHERE Customer\_ID = ?';

  db.query(query, [firstName, lastName, email, mobileNumber, address, pincode, customerId], (err, result) => {

    if (err) {

      console.error('Error updating data:', err);

      return res.status(500).json({ success: false, message: 'Failed to update data in database' });

    }

    console.log('Data updated successfully');

    res.status(200).json({ success: true, message: 'Data updated successfully' });

  });

});

// DELETE method to delete a customer by ID

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

  const customerId = req.params.id;

  const query = 'DELETE FROM CustomerDetails WHERE Customer\_ID = ?';

  db.query(query, [customerId], (err, result) => {

    if (err) {

      console.error('Error deleting data:', err);

      return res.status(500).json({ success: false, message: 'Failed to delete data from database' });

    }

    console.log('Data deleted successfully');

    res.status(200).json({ success: true, message: 'Data deleted successfully' });

  });

});

// Start the server

app.listen(port, () => {

  console.log(`Server is running on port ${port}`);

});