**Create Student Tables**

CREATE TABLE student (

id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(50),

emailid VARCHAR(50),

mobileno VARCHAR(10),

address VARCHAR(50)

)

Connect with MySQL database from NodeJs

const express = require('express');

const mysql = require('mysql2');

const app = express();

const connection = mysql.createConnection({

  host: 'localhost',

  user: 'root',

  password: 'Admin#123',

  database: 'testdb'

});

connection.connect((err) => {

  if (err) {

    console.log('Connection Error : '+err);

    throw err;

  } else {

    console.log('Connected to database');

  }

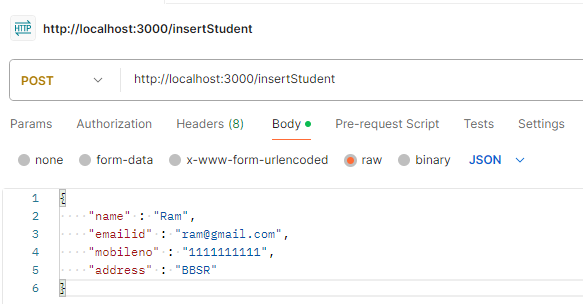
});

app.listen(3000, () => {

  console.log(`Server is running at http://localhost:3000`);

});

**Inserting Records**

****

const express = require('express');

const mysql = require('mysql2');

const app = express();

app.use(express.json());

const connection = mysql.createConnection({

  host: 'localhost',

  user: 'root',

  password: 'Admin#123',

  database: 'testdb'

});

connection.connect((err) => {

  if (err) {

    console.log('Connection Error : '+err);

    throw err;

  } else {

    console.log('Connected to database');

  }

});

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

  const { name, emailid, mobileno, address } = req.body;

  const sql = 'INSERT INTO student (name, emailid, mobileno, address) VALUES (?, ?, ?)';

  connection.query(sql, [name, emailid, mobileno, address], (error, result) => {

      if (error) {

          console.error('Error inserting data:', error);

          res.status(500);

          res.send('Error inserting data into student');

      } else {

          console.error('Error inserting data:', result);

          res.status(200);

          res.send('Product inserted successfully');

      }

  });

});

app.listen(3000, () => {

  console.log(`Server is running at http://localhost:3000`);

});

**Retrieving Records**

const express = require('express');

const mysql = require('mysql2');

const app = express();

app.use(express.json());

const connection = mysql.createConnection({

  host: 'localhost',

  user: 'root',

  password: 'Admin#123',

  database: 'testdb'

});

connection.connect((err) => {

  if (err) {

    console.log('Connection Error : '+err);

    throw err;

  } else {

    console.log('Connected to database');

  }

});

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

  const sql = 'SELECT \* FROM student';

  connection.query(sql, (err, results) => {

      if (err) {

          res.status(500);

          res.send('Error in fetching data');

      }

      res.send(results);

  });

});

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

  const { name, emailid, mobileno, address } = req.body;

  const sql = 'INSERT INTO student (name, emailid, mobileno, address) VALUES (?, ?, ?)';

  connection.query(sql, [name, emailid, mobileno, address], (error, result) => {

      if (error) {

          console.error('Error inserting data:', error);

          res.status(500);

          res.send('Error inserting data into student');

      } else {

          console.error('Error inserting data:', result);

          res.status(200);

          res.send('Product inserted successfully');

      }

  });

});

app.listen(3000, () => {

  console.log(`Server is running at http://localhost:3000`);

});

**Fetch Single Record**

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

  const id = req.params.id;

  const sql = 'SELECT \* FROM student WHERE id=?';

  connection.query(sql, [id], (err, results) => {

      if (err) {

          res.status(500);

          res.send('Error in fetching data');

      }

      res.send(results);

  });

});

**Update Record**

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

  const { id, name, emailid, mobileno, address } = req.body;

  const sql = 'UPDATE student SET name=?, emailid=?, mobileno=?, address=? WHERE id=?';

  connection.query(sql, [name, emailid, mobileno, address, id], (error, result) => {

      if (error) {

          res.status(500);

          res.send('Error in updating data');

      } else {

          res.status(200);

          res.send('Product updated successfully');

      }

  });

});

**Delete Record**

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

  const id = req.params.id;

  const sql = 'DELETE FROM student WHERE id=?';

  connection.query(sql, [id], (error, result) => {

      if (error) {

          res.status(500);

          res.send('Error deleting data into student');

      } else {

          if(result.affectedRows == 0){

            res.status(200);

            res.send('Record not found');

          } else {

            res.status(200);

            res.send('Record deleted successfully');

          }

      }

  });

});