
 Marwadi University Marwadi Chandarana Group 	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Advanced Web Technology	Aim: Create an Endpoint t perform CRUD operation of student using mysql	
Task-3	Date:- 30-03-2025	Enrollment No:- 92200133013

CODE :

Index.js

```
require('dotenv').config();
const app = require('./src/app');
const db = require('./src/models');

const PORT = process.env.PORT || 5000;

db.sequelize.authenticate()
  .then(() => {
    console.log("✔ Database connected!");
    app.listen(PORT, () => console.log(` Server running on http://localhost:${PORT}`));
  })
  .catch(err => console.error("✗ Database connection error:", err));
```

Db.config.js

```
require('dotenv').config();

module.exports = {
  HOST: process.env.DB_HOST,
  USER: process.env.DB_USER,
  PASSWORD: process.env.DB_PASSWORD,
  DB: process.env.DB_NAME,
  DIALECT: process.env.DB_DIALECT
};
```

Db.helper.js

```
const { Sequelize } = require('sequelize');
const dbConfig = require('./db.config');

const sequelize = new Sequelize(dbConfig.DB, dbConfig.USER, dbConfig.PASSWORD, {
  host: dbConfig.HOST,
  dialect: dbConfig.DIALECT
});


sequelize.authenticate()
  .then(() => console.log("✔ Database connected successfully!"))
  .catch(err => console.error("✗ Database connection failed:", err));

module.exports = sequelize;
```

Student.controller.js

```
const studentService = require('../services/student.service');

exports.createStudent = async (req, res) => {
  try {
    const student = await studentService.createStudent(req.body);
    res.status(201).json(student);
  } catch (error) {
    res.status(400).json({ error: error.message });
  }
}
```

 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Advanced Web Technology	Aim: Create an Endpoint t perform CRUD operation of student using mysql	
Task-3	Date:- 30-03-2025	Enrollment No:- 92200133013

```
};

exports.getAllStudents = async (req, res) => {
  try {
    const students = await studentService.getAllStudents();
    res.status(200).json(students);
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
};

exports.getStudentById = async (req, res) => {
  try {
    const student = await studentService.getStudentById(req.params.id);
    student ? res.status(200).json(student) : res.status(404).json({ message: "Student not found" });
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
};

exports.updateStudent = async (req, res) => {
  try {
    const result = await studentService.updateStudent(req.params.id, req.body);
    result[0] ? res.status(200).json({ message: "Student updated successfully" }) : res.status(404).json({ message: "Student not found" });
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
};

exports.deleteStudent = async (req, res) => {
  try {
    const result = await studentService.deleteStudent(req.params.id);
    result ? res.status(200).json({ message: "Student deleted successfully" }) : res.status(404).json({ message: "Student not found" });
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
};
```

Model.Index.js

```
const sequelize = require('../config/db.helper');
const Student = require('../student.model');

const db = { sequelize, Student };



sequelize.sync()
  .then(() => console.log("✔ Tables synchronized!"))
  .catch(err => console.error("✗ Sync error:", err));

module.exports = db;
```

Student.model.js

```
const { DataTypes } = require('sequelize');
const sequelize = require('../config/db.helper');

const Student = sequelize.define('students', {
  id: { type: DataTypes.INTEGER, autoIncrement: true, primaryKey: true },
  name: { type: DataTypes.STRING, allowNull: false },
  email: { type: DataTypes.STRING, allowNull: false, unique: true },
```

 Marwadi University Marwadi Chandarana Group 	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Advanced Web Technology	Aim: Create an Endpoint t perform CRUD operation of student using mysql	
Task-3	Date:- 30-03-2025	Enrollment No:- 92200133013

```
age: { type: DataTypes.INTEGER }
}, { timestamps: false });
```

```
module.exports = Student;
```

Student.routes.js

```
const express = require('express');
const studentController = require('../controllers/student.controller');
```

```
const router = express.Router();
```

```
router.post('/', studentController.createStudent);
router.get('/', studentController.getAllStudents);
router.get('/:id', studentController.getStudentById);
router.put('/:id', studentController.updateStudent);
router.delete('/:id', studentController.deleteStudent);
```

```
module.exports = router;
```

Student.services.js

```
const { Student } = require('../models');
```

```
const createStudent = async (data) => await Student.create(data);
const getAllStudents = async () => await Student.findAll();
const getStudentById = async (id) => await Student.findByPk(id);
const updateStudent = async (id, data) => await Student.update(data, { where: { id } });
const deleteStudent = async (id) => await Student.destroy({ where: { id } });
```

```
module.exports = { createStudent, getAllStudents, getStudentById, updateStudent, deleteStudent };
```

App.js

```
const express = require('express');
const bodyParser = require('body-parser');
const cors = require('cors');
const studentRoutes = require('./routes/student.routes');
```

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


```
app.use('/students', studentRoutes);
```

```
app.get('/', (req, res) => res.send(' Student Management API is running!'));
```

```
module.exports = app;
```

.env

```
DB_HOST=localhost
DB_USER=root
DB_PASSWORD=
DB_NAME=mu_2425_test
DB_DIALECT=mysql
PORT=5000
```

 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Advanced Web Technology	Aim: Create an Endpoint t perform CRUD operation of student using mysql	
Task-3	Date:- 30-03-2025	Enrollment No:- 92200133013

OUTPUT :

Insert the student

POST

http://localhost:5000/students/

Send

Status: 201 Created Size: 64 Bytes Time: 25 ms

Response Headers 7 Cookies Results Docs

```

1 {
2   "id": 1,
3   "name": "Prashant",
4   "email": "prashant@gmail.com",
5   "age": 21
6 }

```

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```

1 {
2   "id": 1,
3   "name": "Prashant",
4   "email": "prashant@gmail.com",
5   "age": 21
6 }

```

Get all the student

GET

http://localhost:5000/students/

Send

Status: 200 OK Size: 125 Bytes Time: 7 ms

Response Headers 7 Cookies Results Docs

```

1 [
2   {
3     "id": 1,
4     "name": "Prashant",
5     "email": "prashant@gmail.com",
6     "age": 21
7   },
8   {
9     "id": 2,
10    "name": "Aryan",
11    "email": "aryan@gmail.com",
12    "age": 22
13   }
14 ]

```

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```

1 {
2   "id": 2,
3   "name": "Aryan",
4   "email": "aryan@gmail.com",
5   "age": 22
6 }

```

Edit the student

PUT

http://localhost:5000/students/2

Send

Status: 200 OK Size: 42 Bytes Time: 10 ms

Response Headers 7 Cookies Results Docs

```

1 {
2   "message": "Student updated successfully"
3 }

```

Query Headers 2 Auth Body 1 Tests Pre Run


JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```

1 {
2   "id": 2,
3   "name": "Aryan",
4   "email": "aryan@yahoo.com",
5   "age": 22
6 }

```

 Marwadi University Marwadi Chandarana Group	Marwadi University Faculty of Technology Department of Information and Communication Technology	
Subject: Advanced Web Technology	Aim: Create an Endpoint to perform CRUD operation of student using mysql	
Task-3	Date:- 30-03-2025	Enrollment No:- 92200133013

Delete the student

DELETE
⌵
http://localhost:5000/students/2
Send

Status: 200 OK
Size: 42 Bytes
Time: 11 ms

Query
Headers 2
Auth
Body 1
Tests
Pre Run

JSON
XML
Text
Form
Form-encode
GraphQL
Binary

Response
Headers 7
Cookies
Results
Docs

```

1 {
2   "message": "student deleted successfully"
3 }

```

DATABASE

Server: 127.0.0.1 » Database: mu_2425_test » Table: students

Browse
Structure
SQL
Search
Insert
Export
Import
Privileges

✓ Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

SELECT * FROM `students`

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 ⌵ | Filter rows: Search this table

Extra options

⬅️ ➡️

	name	email	age	id
<input type="checkbox"/> Edit Copy Delete	Prashant	prashant@gmail.com	21	1