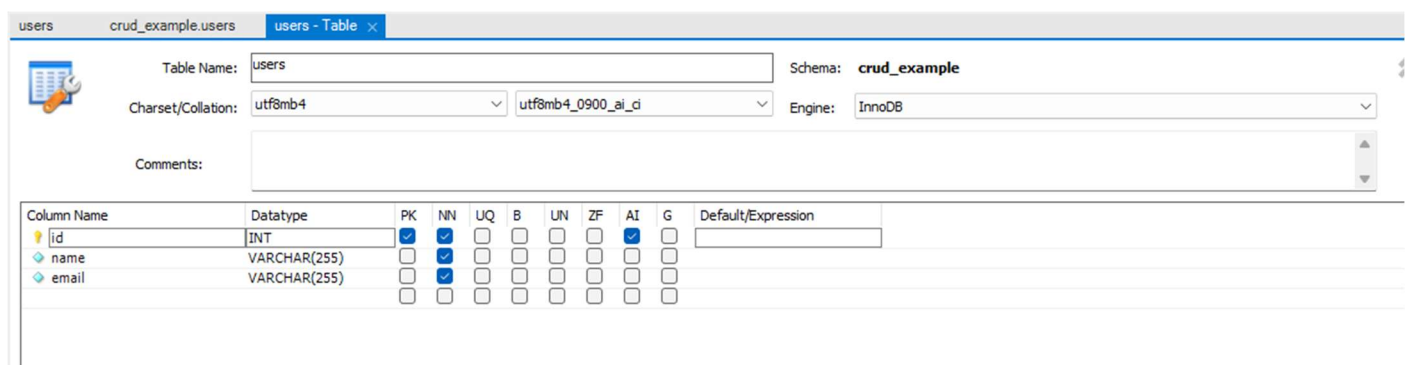
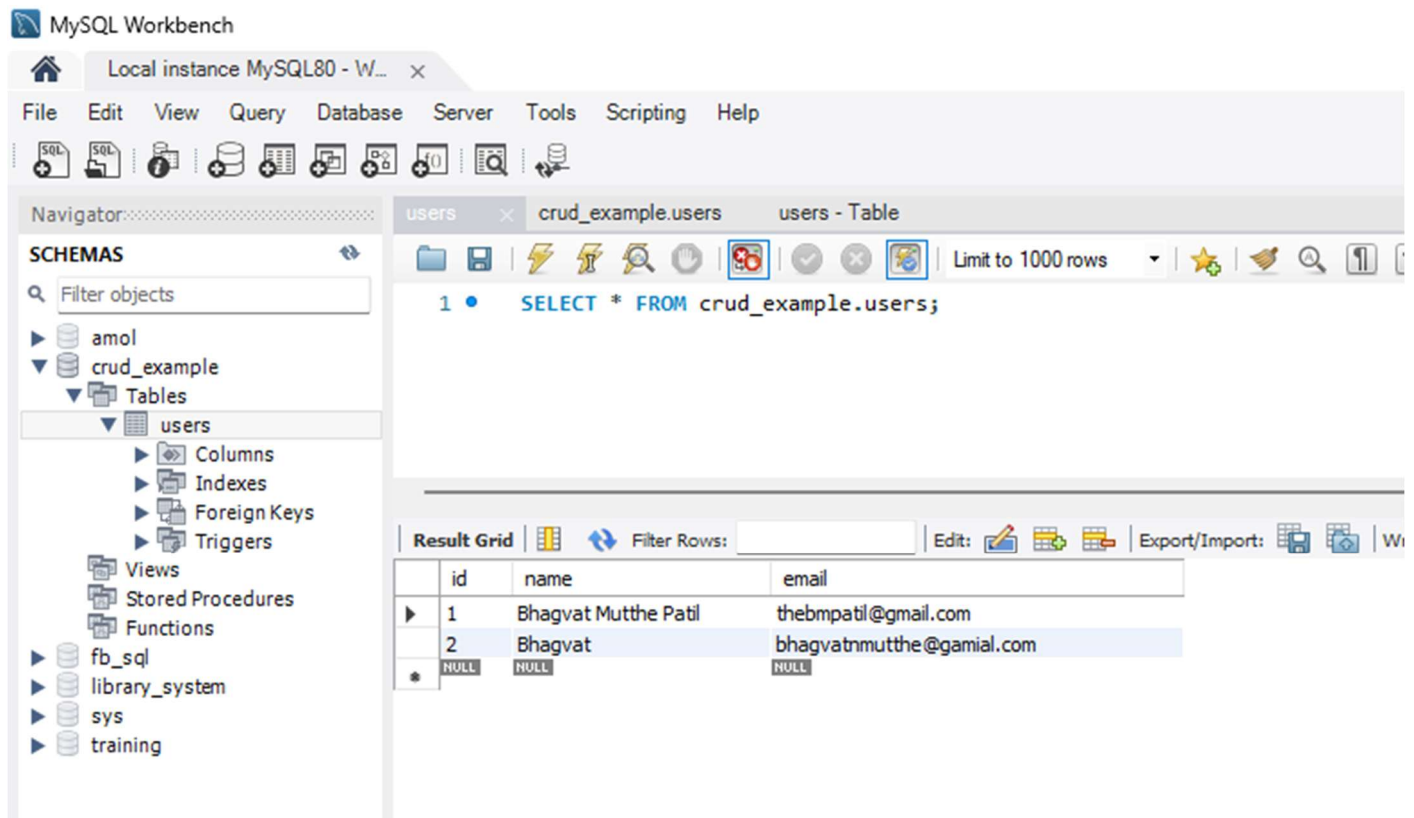


# DB Connection task Using MYSQL and JS

Steeps:

Prereq { Node.js , MYSQL installed)

- 1) Create a folder for project
- 2) Open with VS Code -> open terminal
- 3) Type npm init -y
- 4) Then create below files
- 5) Open terminal again execute command : npm i cors express mysql2 body-parser



# CRUD Operations

User added successfully

## Users List

ID	Name	Email	Actions
1	Bhagvat Mutthe Patil	thebmpatil@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	Bhagvat	bhagvatnmutthe@gamial.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	Ankush Pawar	ankussh@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
4	Ajay Paalhal	ajay@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

# CRUD Oper

127.0.0.1:5500 says

Enter new name:

## Users List

ID	Name	Email	Actions
1	Bhagvat Mutthe Patil	thebmpatil@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	Bhagvat	bhagvatnmutthe@gamial.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	Ankush Pawar	ankussh@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
4	Ajay Paalhal	ajay@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

# CRUD Operations

User updated successfully

## Users List

ID	Name	Email	Actions
1	Bhagvat Mutthe Patil	thebmpatil@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	Bhagvat Patil	bhagvatnmutthe@gamial.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	Ankush Pawar	ankussh@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
4	Ajay Paalhal	ajay@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

127.0.0.1:5500 says

Are you sure you want to delete this user?

## CRUD Oper

## Users List

ID	Name	Email	Actions
1	Bhagvat Mutthe Patil	thebmpatil@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	Bhagvat Patil	bhagvatnmutthe@gamial.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	Ankush Pawar	ankussh@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
4	Ajay Paalhal	ajay@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

# CRUD Operations

User deleted successfully

## Users List

ID	Name	Email	Actions
1	Bhagvat Mutthe Patil	thebmpatil@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	Ankush Pawar	ankussh@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
4	Ajay Paalhal	ajay@gmail.com	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Navigator: crud\_example.users users - Table

**SCHEMAS**

Filter objects

- amol
- crud\_example
  - Tables
    - users
      - Columns
      - Indexes
      - Foreign Keys
      - Triggers
    - Views
    - Stored Procedures
    - Functions
  - fb\_sql
  - library\_system
  - sys
  - training

1 • `SELECT * FROM crud_example.users;`

Limit to 1000 rows

Result Grid

	id	name	email
▶	1	Bhagvat Mutthe Patil	thebmpatil@gmail.com
	3	Ankush Pawar	ankussh@gmail.com
	4	Ajay Paalhal	ajay@gmail.com
*	NULL	NULL	NULL

Filter Rows: Edit: Export/Import: Wr

```
mysql> show tables;
```

```
| Tables_in_crud_example |
```

```
| users |
```

```
1 row in set (0.00 sec)
```

```
mysql> desc users;
```

```
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id    | int       | NO   | PRI | NULL    | auto_increment |
| name  | varchar(255) | NO   |     | NULL    |               |
| email | varchar(255) | NO   |     | NULL    |               |
+-----+-----+-----+-----+-----+
```

```
3 rows in set (0.00 sec)
```

```
mysql> show create table users;
```

```
| Table | Create Table
```

```
| users | CREATE TABLE `users` (
```

```
  `id` int NOT NULL AUTO_INCREMENT,
```

```
  `name` varchar(255) NOT NULL,
```

```
  `email` varchar(255) NOT NULL,
```

```
  PRIMARY KEY (`id`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
```

```
1 row in set (0.00 sec)
```

```
mysql>
```

```
mysql> select *from users;
```

```
+-----+-----+-----+
| id | name                | email                |
+-----+-----+-----+
| 1  | Bhagvat Mutthe Patil | thebmpatil@gmail.com |
| 3  | Ankush Pawar        | ankussh@gmail.com   |
| 4  | Ajay Paalhal        | ajay@gmail.com       |
+-----+-----+-----+
```

```
3 rows in set (0.00 sec)
```

```
const express = require("express");
const mysql = require("mysql2");
const bodyParser = require("body-parser");
const cors = require("cors");

const app = express();
const port = 3000;

// MySQL Database connection
const db = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "0787",
  database: "crud_example",
  port: "3308",
});

db.connect((err) => {
  if (err) throw err;
  console.log("Connected to the database");
});

// Middleware
app.use(cors());
app.use(bodyParser.json());

// Root route
app.get("/", (req, res) => {
  res.send("Welcome to the CRUD API!");
});

// Routes for CRUD operations

// Create
app.post("/add-user", (req, res) => {
  const { name, email } = req.body;
  const query = "INSERT INTO users (name, email) VALUES (?, ?)";
  db.query(query, [name, email], (err, result) => {
    if (err) throw err;
    res.send("User added successfully");
  });
});

// Read
app.get("/users", (req, res) => {
  const query = "SELECT * FROM users";
  db.query(query, (err, result) => {
    if (err) throw err;
    res.json(result);
  });
});
```

```

// Update
app.put("/update-user/:id", (req, res) => {
  const { id } = req.params;
  const { name, email } = req.body;
  const query = "UPDATE users SET name = ?, email = ? WHERE id = ?";
  db.query(query, [name, email, id], (err, result) => {
    if (err) throw err;
    res.send("User updated successfully");
  });
});

// Delete
app.delete("/delete-user/:id", (req, res) => {
  const { id } = req.params;
  const query = "DELETE FROM users WHERE id = ?";
  db.query(query, [id], (err, result) => {
    if (err) throw err;
    res.send("User deleted successfully");
  });
});

// Start the server
app.listen(port, () => {
  console.log(`Server running at http://localhost:${port}`);
});

```

Script.js

```

const apiUrl = "http://localhost:3000";

// Show message
function showMessage(message, type = "success") {
  const messageDiv = document.getElementById("message");
  messageDiv.textContent = message;
  messageDiv.className = `message ${type}`;
  setTimeout(() => {
    messageDiv.textContent = "";
    messageDiv.className = "message";
  }, 3000); // Hide the message after 3 seconds
}

// Fetch all users
function fetchUsers() {
  fetch(`${apiUrl}/users`)
    .then((response) => response.json())
    .then((data) => {
      const usersTable = document.querySelector("#usersTable tbody");
      usersTable.innerHTML = "";
      data.forEach((user) => {

```

BHAGVAT MUTTHE

```

const row = document.createElement("tr");
row.innerHTML = `
    <td>${user.id}</td>
    <td>${user.name}</td>
    <td>${user.email}</td>
    <td>
        <button class="edit" onclick="editUser(${user.id}, '${user.name}',
'${user.email}')">Edit</button>
        <button class="delete"
onclick="deleteUser(${user.id})">Delete</button>
    </td>
`;
usersTable.appendChild(row);
});
});
}

// Add user
document.getElementById("addUserForm").addEventListener("submit", (e) => {
    e.preventDefault();
    const name = document.getElementById("name").value;
    const email = document.getElementById("email").value;

    fetch(`${apiUrl}/add-user`, {
        method: "POST",
        headers: {
            "Content-Type": "application/json",
        },
        body: JSON.stringify({ name, email }),
    }).then((response) => {
        if (response.ok) {
            fetchUsers();
            document.getElementById("addUserForm").reset();
            showMessage("User added successfully");
        } else {
            showMessage("Error adding user", "error");
        }
    });
});

// Edit user
function editUser(id, name, email) {
    const newName = prompt("Enter new name:", name);
    const newEmail = prompt("Enter new email:", email);

    if (newName && newEmail) {
        fetch(`${apiUrl}/update-user/${id}`, {
            method: "PUT",
            headers: {
                "Content-Type": "application/json",
            },
            body: JSON.stringify({ name: newName, email: newEmail }),
        }).then((response) => {

```



```

        if (response.ok) {
            fetchUsers();
            showMessage("User updated successfully");
        } else {
            showMessage("Error updating user", "error");
        }
    });
}
}

// Delete user
function deleteUser(id) {
    if (confirm("Are you sure you want to delete this user?")) {
        fetch(`${apiUrl}/delete-user/${id}`, {
            method: "DELETE",
        }).then((response) => {
            if (response.ok) {
                fetchUsers();
                showMessage("User deleted successfully");
            } else {
                showMessage("Error deleting user", "error");
            }
        });
    }
}

// Initialize
fetchUsers();

```

index.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>CRUD Operations</title>
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <div class="container">
      <h1>CRUD Operations</h1>

      <!-- Message Display -->
      <div id="message" class="message"></div>

      <!-- Add User Form -->
      <form id="addUserForm">
        <input type="text" id="name" placeholder="Name" required />
        <input type="email" id="email" placeholder="Email" required />

```

**BHAGVAT MUTTHE**

```

        <button type="submit">Add User</button>
    </form>

    <hr />

    <!-- Users List -->
    <h2>Users List</h2>
    <table id="usersTable">
        <thead>
            <tr>
                <th>ID</th>
                <th>Name</th>
                <th>Email</th>
                <th>Actions</th>
            </tr>
        </thead>
        <tbody></tbody>
    </table>
</div>

<script src="script.js"></script>
</body>
</html>

```

Style.css

```

body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    padding: 20px;
}

.container {
    max-width: 800px;
    margin: 0 auto;
    background: #fff;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

form input,
form button {
    padding: 10px;
    margin: 5px;
}

button {
    cursor: pointer;
}

table {

```

BHAGVAT MUTTHE

```

width: 100%;
margin-top: 20px;
border-collapse: collapse;
}

table,
th,
td {
border: 1px solid #ddd;
}

th,
td {
padding: 10px;
text-align: left;
}

button.edit,
button.delete {
background-color: #4caf50;
color: white;
border: none;
padding: 5px 10px;
cursor: pointer;
}

button.delete {
background-color: #f44336;
}

.message {
padding: 10px;
margin-bottom: 15px;
border-radius: 5px;
font-weight: bold;
display: none;
}

.message.success {
background-color: #28a745;
color: white;
}

.message.error {
background-color: #dc3545;
color: white;
}

#message {
display: block;
}

```