CRUD Operations in MERN

---

1. CREATE (Insert Data)

Backend (Node + Express + MongoDB)

```
// POST - Create Task
app.post("/tasks", async (req, res) => {
  try {
    const newTask = new TaskModel(req.body);
    await newTask.save();
    res.json(newTask);
  } catch (error) {
    res.status(500).json(error);
  }
});
```

Frontend (React)

```
const addTask = async () => {
  await fetch("http://localhost:5000/tasks", {
    method: "POST",
    headers: {
      "Content-Type": "application/json"
    },
    body: JSON.stringify({ taskName: task })
  });
};
```

Adds new data to MongoDB

---

2. READ (Fetch Data)

Backend

```
// GET - Fetch All Tasks
app.get("/tasks", async (req, res) => {
  const tasks = await TaskModel.find();
  res.json(tasks);
});
```

Frontend

```
useEffect(() => {
  fetch("http://localhost:5000/tasks")
    .then(res => res.json())
    .then(data => setTasks(data));
}, []);
```

Retrieves data from database

---

3. UPDATE (Modify Data)

Backend

```
// PUT - Update Task
app.put("/tasks/:id", async (req, res) => {
  await TaskModel.findByIdAndUpdate(
    req.params.id,
    req.body
  );
  res.json({ message: "Updated" });
});
```

Frontend

```
const updateTask = async (id) => {
  await fetch(`http://localhost:5000/tasks/${id}`, {
    method: "PUT",
    headers: {
      "Content-Type": "application/json"
    },
    body: JSON.stringify({ status: "Completed" })
  });
};
```

Updates existing data

---

4. DELETE (Remove Data)

Backend

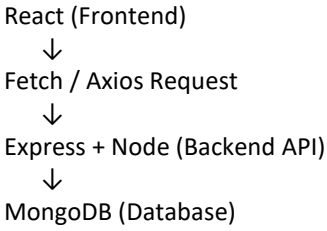
```
// DELETE - Remove Task
app.delete("/tasks/:id", async (req, res) => {
  await TaskModel.findByIdAndDelete(req.params.id);
  res.json({ message: "Deleted" });
});
```

Frontend

```
const deleteTask = async (id) => {
  await fetch(`http://localhost:5000/tasks/${id}`, {
    method: "DELETE"
  });
};
```

Removes data from database

Basic MERN CRUD Flow


React (Frontend)
↓
Fetch / Axios Request
↓
Express + Node (Backend API)
↓
MongoDB (Database)