

You Can **collect data from an HTML form** and **insert it into a MySQL table** using Express.js. Below is a complete beginner-friendly guide to do just that.

□ **Goal: Submit Form → Store Data in MySQL**

□ **Prerequisites**

Make sure you have these installed:

- □ Node.js
 - □ MySQL Server running
 - □ MySQL database and a table created
-

□ **1. Install Required Packages**

Inside your project folder:

```
npm install express mysql body-parser
```

□ **2. Create MySQL Table Example**

In your MySQL database, run:

```
CREATE DATABASE testdb;
```

```
USE testdb;
```

```
CREATE TABLE users (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(100),  
  email VARCHAR(100)  
);
```

□ 3. Create HTML Form (views/form.html):-

```
<!DOCTYPE html>
<html>
<head>
  <title>Add User</title>
</head>
<body>
  <h2>Add User</h2>
  <form action="/add-user" method="POST">
    <label>Name:</label><br>
    <input type="text" name="name" required><br><br>
    <label>Email:</label><br>
    <input type="email" name="email" required><br><br>
    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

□ 4. Update app.js to Handle Form and MySQL

```
const express = require('express');
const mysql = require('mysql');
const bodyParser = require('body-parser');
const path = require('path');

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

// MySQL connection
const db = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: '', // Replace with your MySQL password
  database: 'testdb' // Make sure this matches your DB
});

db.connect((err) => {
  if (err) throw err;
  console.log('Connected to MySQL');
});

// Middleware
app.use(bodyParser.urlencoded({ extended: false }));
app.use(express.static(path.join(__dirname, 'public')));

// Route to show the form
app.get('/', (req, res) => {
  res.sendFile(path.join(__dirname, 'views', 'form.html'));
});

// Route to handle form submission
app.post('/add-user', (req, res) => {
  const { name, email } = req.body;
```

```
const sql = 'INSERT INTO users (name, email) VALUES (?, ?)';
db.query(sql, [name, email], (err, result) => {
  if (err) throw err;
  console.log('User added:', result.insertId);
  res.send(`<h2>User added successfully!</h2><a href="/">Go Back</a>`);
});
});

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

▶ 5. Run Your App

node app.js

Visit: <http://localhost:3000>

Fill out the form and submit — the data will be saved to your MySQL `users` table!

Want to Check the Data?

In MySQL:

```
SELECT * FROM users;
```

Done!

Let me know if you want:

- To validate the form inputs
- Redirect after submission
- Use `async/await` with `mysql2` instead of callbacks