

# ASSIGNMENT

**Name:A.Sai Varshith**

**Hall ticket number:2303a51831**

**Batch-04**

**Date:-19-01-2025**

## 1. Implementing CRUD Operations

You implement Create, Read, Update, and Delete operations by calling specific methods on your Sequelize model objects within your controller logic.

- **Create:** Use the `.create()` method to insert a new row into the database.
- **Read:** Use `.findAll()` to retrieve all records or `.findByPk(id)` to find a specific entry by its primary key.
- **Update:** Use the `.update()` method, which requires a `where` clause to specify which record to modify.
- **Delete:** Use the `.destroy()` method with a `where` clause to remove records from the database.

## 2. Data Validation and Constraints

Sequelize handles data integrity at both the application level and the database level:

- **Model Validation:** These are defined in the model schema (e.g., `isEmail: true, len: [2, 20]`) and are checked before the query is sent to the database.
- **Constraints:** These are database-level rules such as `allowNull: false, unique: true`, or foreign key restrictions that prevent invalid data from being stored.

## 3. Defining Relationships

Relationships are defined in your model configuration files to link different tables together.

- **One-to-Many:** For a platform where one course has many enrollments, you use `Course.hasMany(Enrollment)` and `Enrollment.belongsTo(Course)`.
- **Many-to-Many:** If a student can join many courses and a course has many students, you use `User.belongsToMany(Course, { through: 'Enrollments' })`.

## 4. Role of Sequelize Methods

These built-in methods act as an abstraction layer, translating JavaScript code into SQL queries:

Method	Role	SQL Equivalent
<code>create</code>	Inserts a new record.	<code>INSERT INTO</code>

Method	Role	SQL Equivalent
<code>findAll</code>	Retrieves all records.	<code>SELECT *</code>
<code>findById</code>	Retrieves one record by ID.	<code>SELECT ... WHERE id = x</code>
<code>update</code>	Modifies existing records.	<code>UPDATE ... SET</code>
<code>destroy</code>	Removes records.	<code>DELETE FROM</code>

## 5. Handling Database Errors Gracefully

To prevent your server from crashing during database issues, you should use a structured error-handling approach:

- **Async/Await:** Use `try/catch` blocks or an `asyncHandler` middleware (as seen in your `app.js`) to catch rejected promises.
- **Middleware:** Pass errors to a global `errorHandler` (like the one imported in your `app.js`) to send a clean JSON response back to the client.
- **Specific Errors:** Check for specific Sequelize error types, such as `SequelizeValidationError` or `SequelizeUniqueConstraintError`, to provide helpful feedback to the user.

## Images

The screenshot shows a Windows desktop environment with a Visual Studio Code (VS Code) instance open. The terminal window at the bottom left displays the following PowerShell commands:

```

PS D:\FULLSTACK\2026\20260119> Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
PS D:\FULLSTACK\2026\20260119> npm init -y
Wrote to D:\FULLSTACK\2026\20260119\package.json

```

The code editor window shows a package.json file with the following content:

```

{
  "name": "20260119",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" & exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "type": "commonjs",
  "dependencies": {
    "express": "^5.2.1",
    "react": "^19.2.3"
  },
  "devDependencies": {
    "node-sass": "^3.1.11"
  }
}

```

A floating AI interface on the right side of the screen says "Ask about your code" and "Ask about your code". It includes a message "AI responses may be inaccurate." and a link "Generate Agent Instructions to onboard AI onto your codebase." The status bar at the bottom right shows system information like battery level, signal strength, and the date/time.

```
C:\Users\91939\OneDrive\Desktop\Fullstack2026>npm init express
Need to install the following packages:
create-express@1.0.4
Ok to proceed? (y) y

npm warn deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm warn deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm warn deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm warn deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm warn deprecated formidable@1.2.6: Please upgrade to latest, formidable@v2 or formidable@v3! Check these notes: https://bit.ly/2ZEQIau
npm warn deprecated babel-preset-es2015@6.24.1: 🎉 Thanks for using Babel: we recommend using babel-preset-env now: please read https://babeljs.io/env to update!
npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm warn deprecated superagent@2.3.0: Please upgrade to superagent v10.2.2+, see release notes at https://github.com/forwardemail/superagent/releases/tag/v10.2.2 - maintenance is supported by Forward Email @ https://forwardemail.net
npm warn deprecated core-js@2.6.12: core-js@<3.23.3 is no longer maintained and not recommended for usage due to the number of issues. Because of the V8 engine whims, feature detection in old core-js versions could cause a slowdown up to 100x even if nothing is polyfilled. Some versions have web compatibility issues. Please, upgrade your dependencies to the actual version of core-js.

> fullstack2026@1.0.0 npx
> create-express
```

```
C:\Users\91939\OneDrive\Desktop\Fullstack2026>
Ln 1 Col 1 Spaces: 2 HTI
> fullstack2026@1.0.0 npx
> create-express
```

```
C:\Users\91939\OneDrive\Desktop\Fullstack2026>npm install express
added 65 packages, and audited 66 packages in 4s

22 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

```
C:\Users\91939\OneDrive\Desktop\Fullstack2026>
run npm fund for details
found 0 vulnerabilities
C:\Users\91939\OneDrive\Desktop\Fullstack2026>npm i -D nodemon
added 27 packages, and audited 93 packages in 2s

26 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

```
C:\Users\91939\OneDrive\Desktop\Fullstack2026>
`Node.js v24.13.0
PS D:\FULLSTACK 2026\20260119> {
  >>   "name": "Sai Varshith"
  >>   "Gmail:asv@123.gmail.com"
  >> }
```

The image shows two instances of Microsoft Visual Studio Code running on a Windows desktop. Both instances have their title bars and panes visible.

**Top Window (Left):**

- Terminal:** Displays PowerShell history with commands like `Set-ExecutionPolicy RemoteSigned -Scope CurrentUser` and `npm init -y`.
- Code Editor:** Shows the `package.json` file content:

```
{
  "name": "20260119",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "type": "commonjs",
  "dependencies": {
    "express": "^5.2.1",
    "react": "^19.2.3"
  },
  "devDependencies": {
    "node-sass": "^3.1.11"
  }
}
```

**Bottom Window (Right):**

- Terminal:** Displays PowerShell history with the same commands.
- Code Editor:** Shows the `package-lock.json` file content, which is a large JSON object listing dependencies and their resolved versions, hashes, and other metadata.

Both windows include standard VS Code UI elements like tabs, status bars, and floating code exploration panes.

The screenshot shows the VS Code interface with the package.json file open in the center editor. The code content is as follows:

```
1 {
2   "name": "20260119",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" & exit 1"
8   },
9   "keywords": [],
10  "author": "",
11  "license": "ISC",
12  "type": "commonjs",
13  "dependencies": {
14    "express": "^5.2.1",
15    "react": "^19.2.3"
16  },
17  "devDependencies": {
18    "nodemon": "^3.1.11"
19  }
20}
```

The sidebar on the left shows the project structure with files like .env and package-lock.json. A floating panel on the right says "Ask about your code".

The screenshot shows the VS Code interface with the package.json file open in the center editor. The code content is identical to the one above. Below the editor, a terminal window is open with the following command history:

```
PS D:\FULLSTACK 2026\20260119> Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
PS D:\FULLSTACK 2026\20260119> npm init -y
Wrote to D:\FULLSTACK 2026\20260119\package.json:
```

A floating panel on the right says "Ask about your code".

The screenshot displays two instances of the Visual Studio Code (VS Code) interface side-by-side. Both instances have the title bar "Q\_20260119".

**Top Instance (Left):**

- Explorer:** Shows a folder named "20260119" containing "node\_modules", ".env", ".se", "package-lock.json", and "package.json".
- Editor:** The "package-lock.json" tab is active, showing a large JSON object representing the dependency tree. It includes fields like "name", "version", "description", "main", "scripts", "keywords", "author", "license", and "type". Numerous dependencies are listed under "dependencies" and "devDependencies", each with its own version number and hash.
- Bottom Bar:** Includes icons for File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar. Status bar: Line 1, Col 1 / Spaces: 2 / UTF-8 / LF / JSON / ENG / IN / 19-01-2026.

**Bottom Instance (Right):**

- Explorer:** Shows a folder named "20260119" containing "node\_modules", ".env", ".se", "package-lock.json", and "package.json".
- Editor:** The "package.json" tab is active, showing a smaller JSON object with basic metadata:

```
{  "name": "20260119",  "version": "1.0.0",  "description": "",  "main": "index.js",  "scripts": {    "test": "echo \"Error: no test specified\" & exit 1"  },  "keywords": [],  "author": "",  "license": "ISC",  "type": "commonjs"}
```
- Bottom Bar:** Includes icons for File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar. Status bar: Line 1, Col 1 / Spaces: 2 / UTF-8 / LF / JSON / ENG / IN / 19-01-2026.

The screenshot shows the Visual Studio Code interface with a dark theme. In the Explorer sidebar, there is a folder named '20260119' containing files like '.env', '.se', 'package-lock.json', and 'package.json'. The 'package.json' file is selected and shown in the main editor area. The code content is as follows:

```
1  {
2    "name": "20260119",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "Debug": {
7      "scripts": {
8        "test": "echo \\\"Error: no test specified\\\" & exit 1"
9      },
10     "keywords": [],
11     "author": "",
12     "license": "ISC",
13     "type": "commonjs",
14     "dependencies": {
15       "express": "^5.2.1",
16       "react": "^19.2.3"
17     },
18     "devDependencies": {
19       "nodemon": "^3.1.11"
20     }
21 }
```

A floating window titled 'Ask about your code' is overlaid on the editor. It contains the text 'Explore and understand your code' and 'Ask' and 'Auto' dropdown menus. The status bar at the bottom right of the floating window shows 'package.json'.