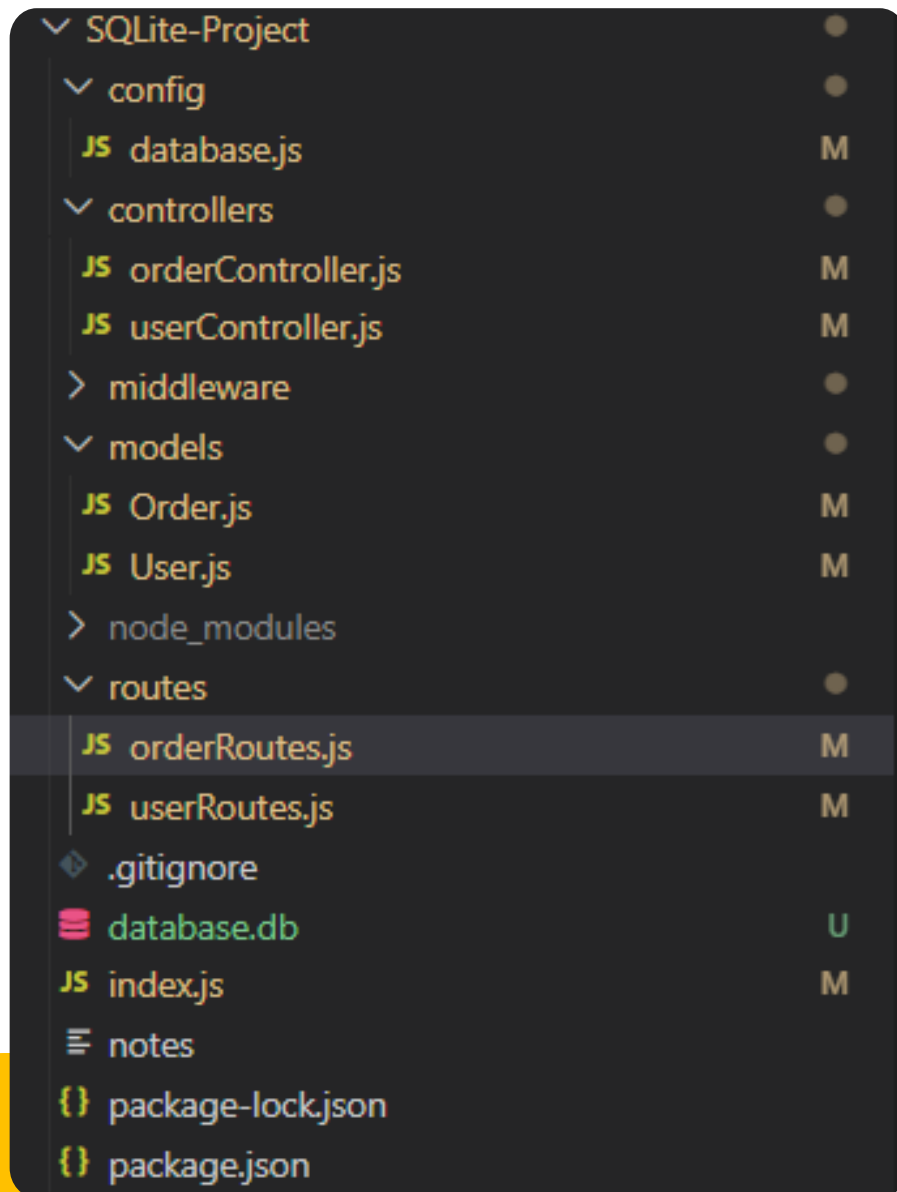# API Development

Safiul Alam

Project

- RESTful API using Node.js, designed to manage users and orders.

- The API allows clients to perform Create, Read, Update, and Delete (CRUD) operations on users and orders, effectively handling various business logic scenarios.

Key Features

User Management

Order Management

Data Validation and Constraints
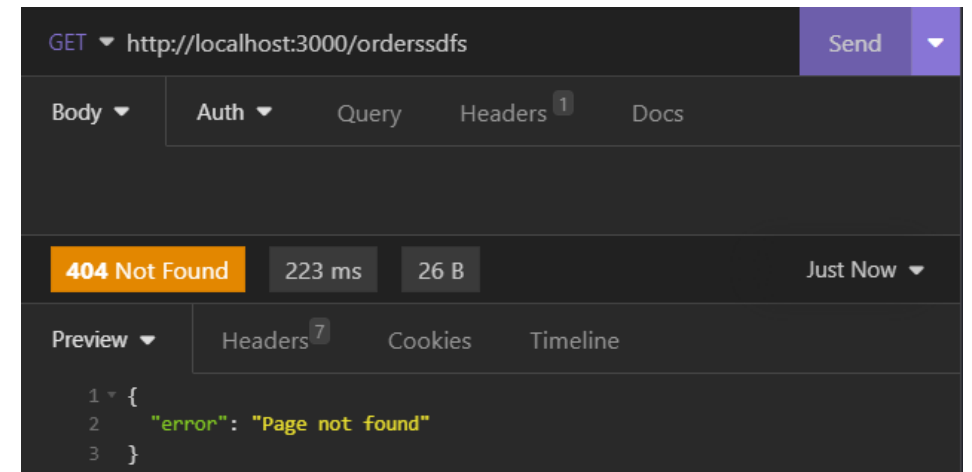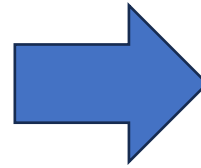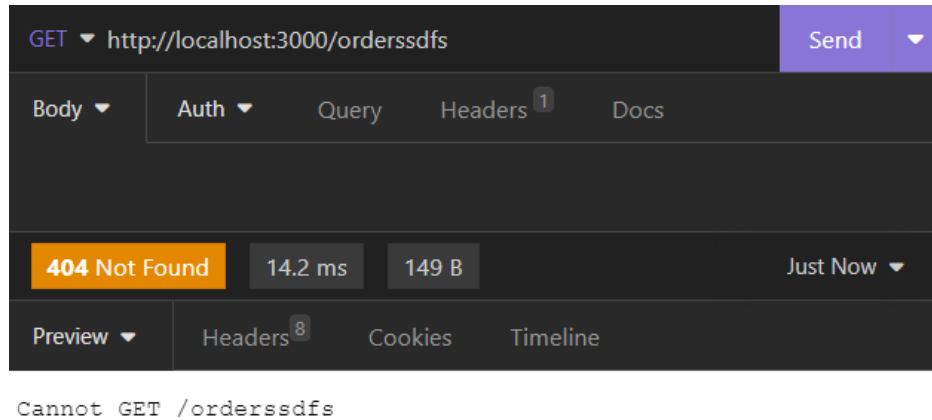
Relational Data Handling

Error Handling

Technology

- **Node.js**
- **Express.js.**
- **Sequelize**
- **SQLite**
- **Nodemon**

Error Handling

Challenges

Error handling

```
module.exports = (err, req, res, next)
    => {
        console.error(err.stack);
        res.status(500).json({ error: err.
    message });
    };
```

Relationship between multiple tables tables

```
17    app.use((req, res, next) => {
18        res.status(404).json({ error: "Page not found" });
19    });
```

Challenges

- **Relationship between multiple tables tables .**

```
37    User.hasMany(Order, { foreignKey: "userId" });
38    Order.belongsTo(User, { foreignKey: "userId" });
```

```
1 ▾ [
2 ▾    {
3        "id": 1,
4        "item": "Mouse",
5        "quantity": 1,
6        "price": 54,
7        "createdAt": "2023-11-29T19:21:34.314Z",
8        "updatedAt": "2023-11-29T19:21:34.314Z",
9        "userId": 3,
10 ▾     "user": {
11          "id": 3,
12          "firstName": "Cristiano",
13          "lastName": "Ye",
14          "createdAt": "2023-11-29T19:21:29.640Z",
15          "updatedAt": "2023-11-29T19:21:29.640Z"
16        }
17     },
```

Timeline

- Folder Structure = 2h

- **User** & **Orders** Models including constrains, indexes and relationships = 4h

- Logic (GET, POST, PUT) Searching multiple info etc.. = 6h

- Relationships between **Orders** and **Users** tables = 1h

- Error Handlers = 1hr

- Final Edit = 3h

TOTAL = **16h**