/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* WEB322 – Assignment 05

\*

\* I declare that this assignment is my own work in accordance with Seneca's

\* Academic Integrity Policy:

\*

\* https://www.senecacollege.ca/about/policies/academic-integrity-policy.html

\*

\* Name: \_\_\_Thi Kieu Trinh Vu\_\_\_\_\_\_\_\_\_ Student ID: \_\_\_122630221\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_22nd March 2024\_\_\_\_\_\_\_\_\_\_\_\_

\*

\* Published URL: \_https://happy-outerwear-tuna.cyclic.app/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Reflection

Part 1:

I set up my database on <https://neon.tech/> and I got my information in .env file:

A screenshot of a computer program

Description automatically generated

Then I modify the file legoSets.js for set up the Sequelize and connect to the database:

A screen shot of a computer

Description automatically generated

I got an error message “ConnectionError [SequelizeConnectionError]: connection is insecure (try using sslmode=require)” that why I fixed it by adding the dialectOptions to ensure that Sequelize will use SSL for the database connection.

After that, I created two models as Theme and Set and disable the createdAt and updatedAt fields.

A screen shot of a computer program

Description automatically generated

I also create an association between the two:



Part 2: Refactoring Existing Code to use Sequelize.

I deleted the parts that I no longer need and refactored some functionality that I already have.

I used the Set model to resolve the returned Promise, I also put the reject in the case that no set was found. And the findAll() function in Sequelize is used to retrieve all sets based on the database. Although the first time, I used the include: [Theme] as the instruction but I got an error with the alias, then I found out that I should change it into model: Theme and it worked!

A screen shot of a computer program

Description automatically generated

Part 3: Adding, Editing, and Deleting

I used async function because it involves asynchronous operations, such as interacting with a database. For example, with the addSet(setData), it awaits the asynchronous operation of creating a new set in the database using Set.create(setData). By marking it as async, it allows us to use await inside the function to wait for the database operation to complete before proceeding. It’s the same with the others in the code that I provided below:

A screen shot of a computer program

Description automatically generated

Then I exported the modules in the legoSet.js



Here is the file 500.ejs which is created base on the file 404.ejs  
A screenshot of a computer program

Description automatically generated

There is the file addSet.ejs which is created based on the requirements in the instruction:

A screenshot of a computer

Description automatically generated

It’s the same with the editSet.ejs:  
A screenshot of a computer program

Description automatically generated

For the server.js, I made some changes in it:

I created the route for adding the new set, editing a set and deleting a set based on the requirements in the instruction and send the error message when it catches an error. For example, asynchronously retrieves set data and themes, then renders the "editSet" view with the retrieved data or asynchronously updates set data in the database, then redirects to "/lego/sets".

A screen shot of a computer program

Description automatically generated

I did the testes and it passed with the adding, editting and deleting functions 😊 it also can push into cyclic.