LEGO repository

Deployment instructions can be find in(readme.docx)

More details on my GitHub:

Front-end: https://github.com/D18130495/LEGO\_repository-React\_client

Back-end: https://github.com/D18130495/LEGO\_repository-React\_server

Briefly project description:

This is a website for querying Lego sets information. In this project, there are three main modules:

Search Lego information: Use to search Lego sets information.

Manage Lego sets information: Use to manage Lego sets theme and release year, manage the details of Lego sets

Manage user: Use to manage the user information.

## Technology Selection:

Front-end: React, Ant Design, Ajax(Axios).

Back-end: Node.js, Express, MongoDB.

Model definition for database:

This is use to store the theme and year, if parentId equal to 0, mean this is theme, if parentId not equal to 0, mean this is year.

Category: name: {type: String, required: true}// the name of the category

parentId: {type: String, required: true, default: '0'} // the parent ID of the category

This is use to store set information

Set: categoryId: {type: String, required: true} // category ID, year of the set, such as(2021, 2020, 2019)

  pCategoryId: {type: String, required: true}// parent category ID, such as(Architecture, Batman™, Harry Potter™)

  name: {type: String, required: true} // name of set

  price: {type: Number, required: true} // price of set

  desc: {type: String} // short desc of set

  imgs: {type: Array, default: []} // address of images

  detail: {type: String} // detail of the set

This is use to store user information

User: username: {type: String, required: true}, // user name

  password: {type: String, required: true}, // user password

  phone: String, // user phone number

  email: String, // user email

  create\_time: {type: Number, default: Date.now}, // create time

  menus: Array // user can visit which function

API definition to receive data:

/login: use for user login (POST)

/manage/category/list: use for get category list (GET)

/manage/category/add: use for add category list (POST)

/manage/category/update: use for update category list, can be theme or year (POST)

/manage/category/year: use for get year list (GET)

/manage/category/year/delete: use for delete year (GET)

/manage/set/add: use for add set information (POST)

/manage/set/update: use for update set information (POST)

/manage/set/delete: use for delete set information (POST)

/manage/set/list: use to get set list (GET)

/manage/set/search: use to get set list by pagination (GET)

/manage/user/list: use for get user list (GET)

/manage/user/add: use for add user (POST)

/manage/user/update: use for update user (POST)

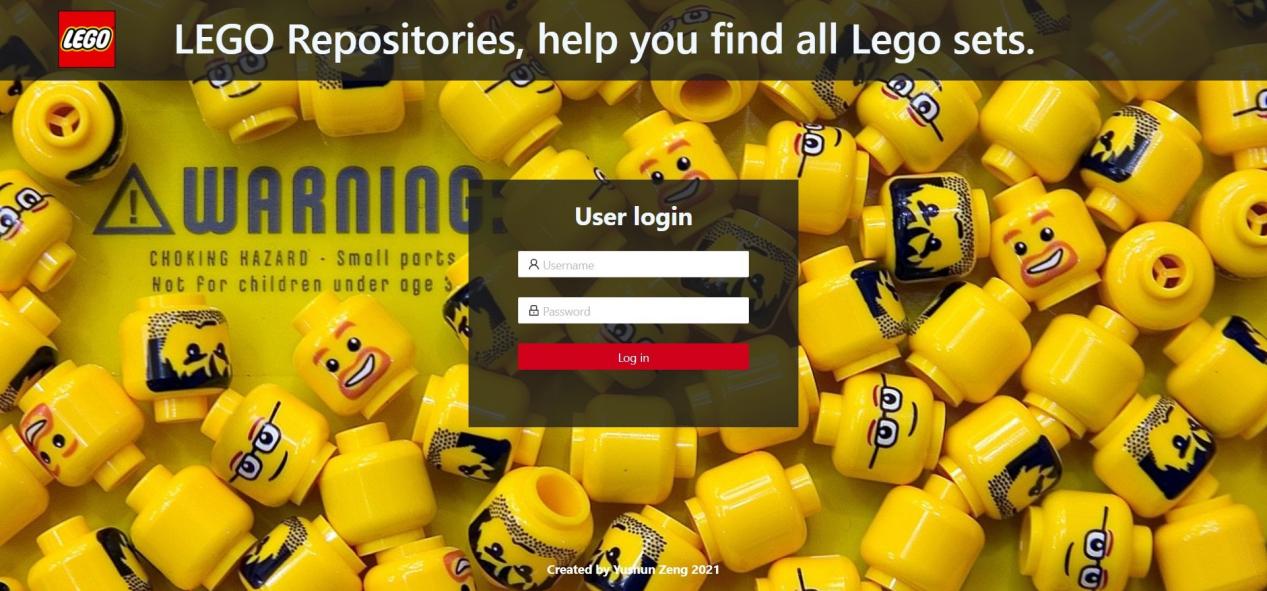
/manage/user/delete: use for delete user (POST)

/manage/img/upload: use for upload images (POST)

/manage/img/delete: use for delete images (POST)

Login page:

The login page provides user verification function, input verification.



Show login page

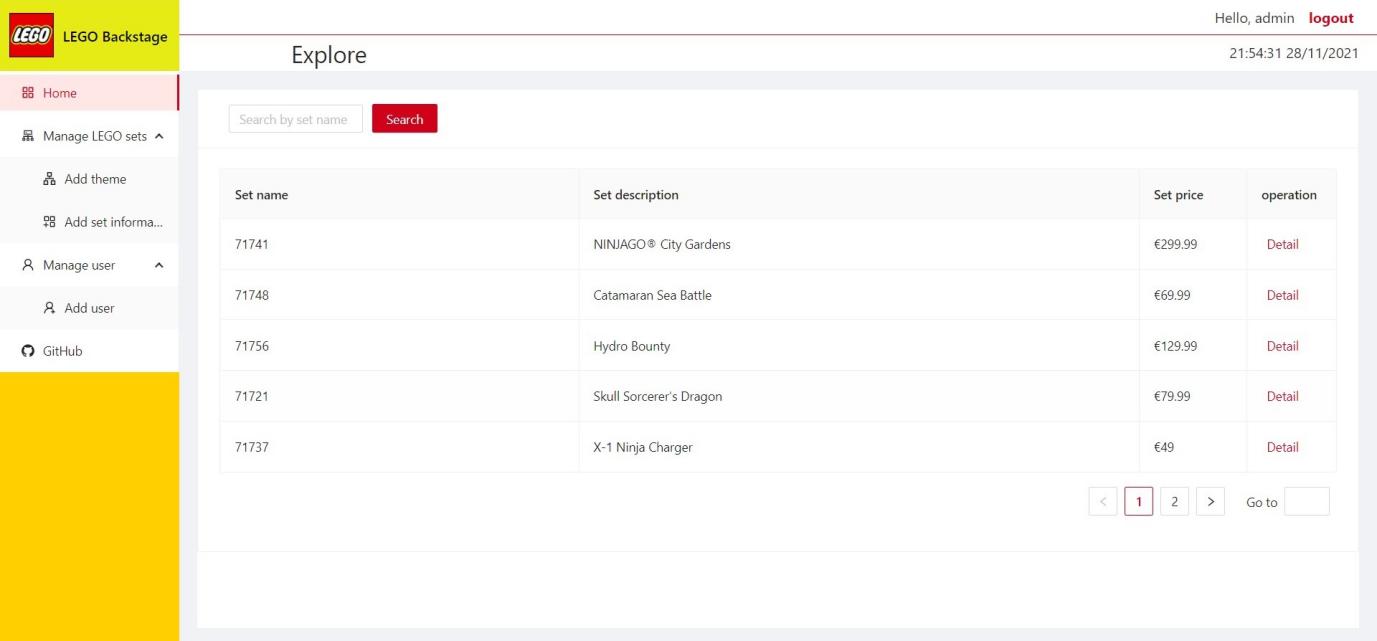
Overall layout:

This is the overall layout of my project.

On the left is the navigation bar, click to jump to the corresponding function.

Above is the header part to display the currently logged in user and the current time, the user can log out through the button in the upper right corner

The middle part is the main display area for different functions.

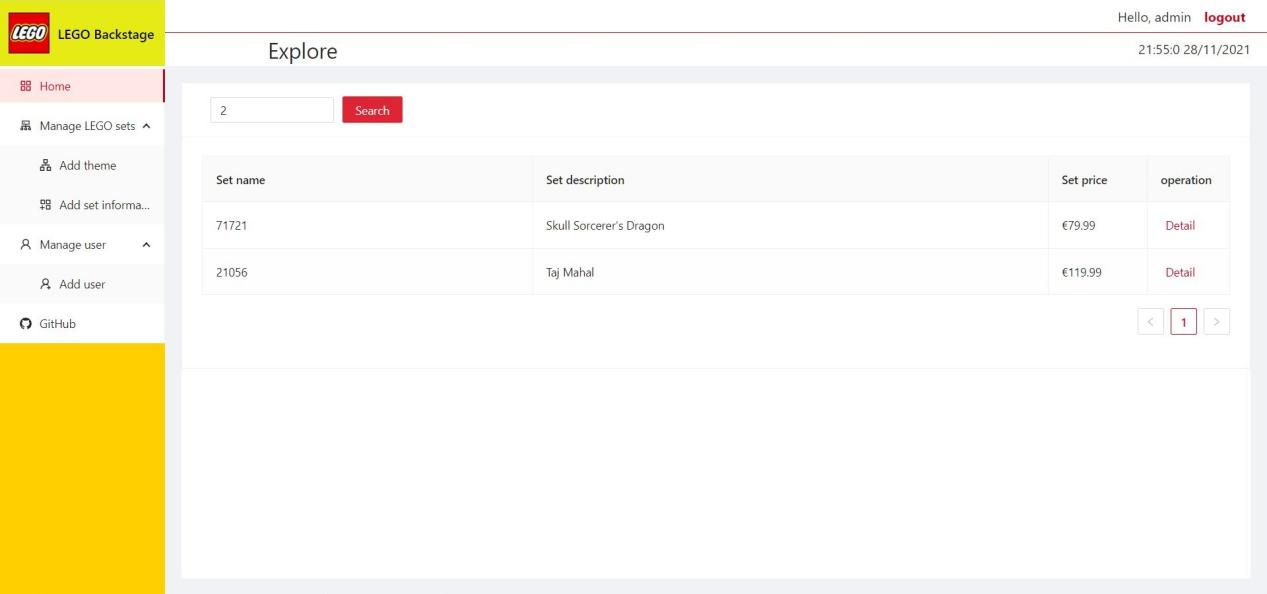


Show overall looking of the website

Detailed description of different module functions:

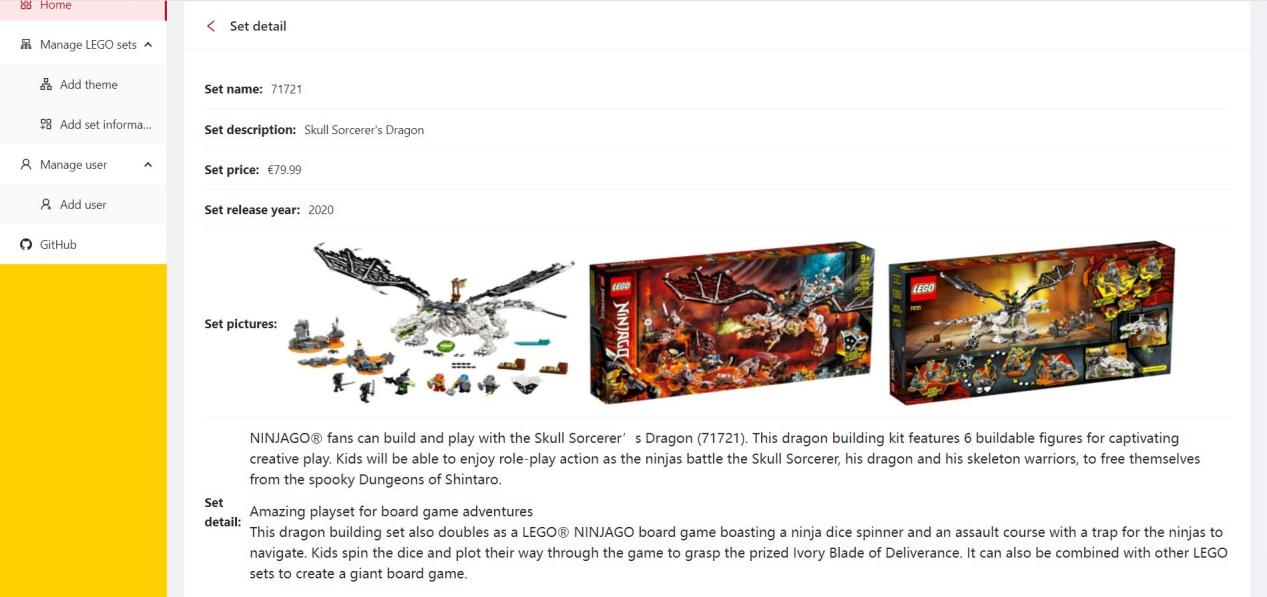
1. Home:

Home part provide user search and display set detail function. The user can query the Lego set by entering the number corresponding to the set in the search box, you can enter all or half of the number. This search function provides a fuzzy query function to help users find Lego sets better.



Show fuzzy query

When you click the detail button on the back side, the detailed information of this set will be displayed.



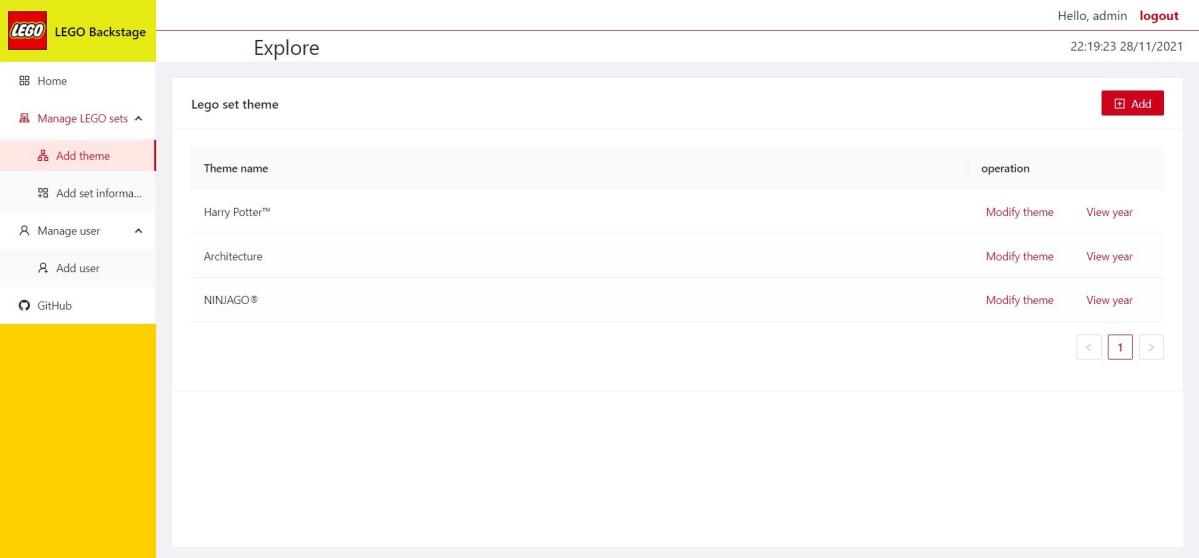
Show detail page

1. Manage LEGO sets:

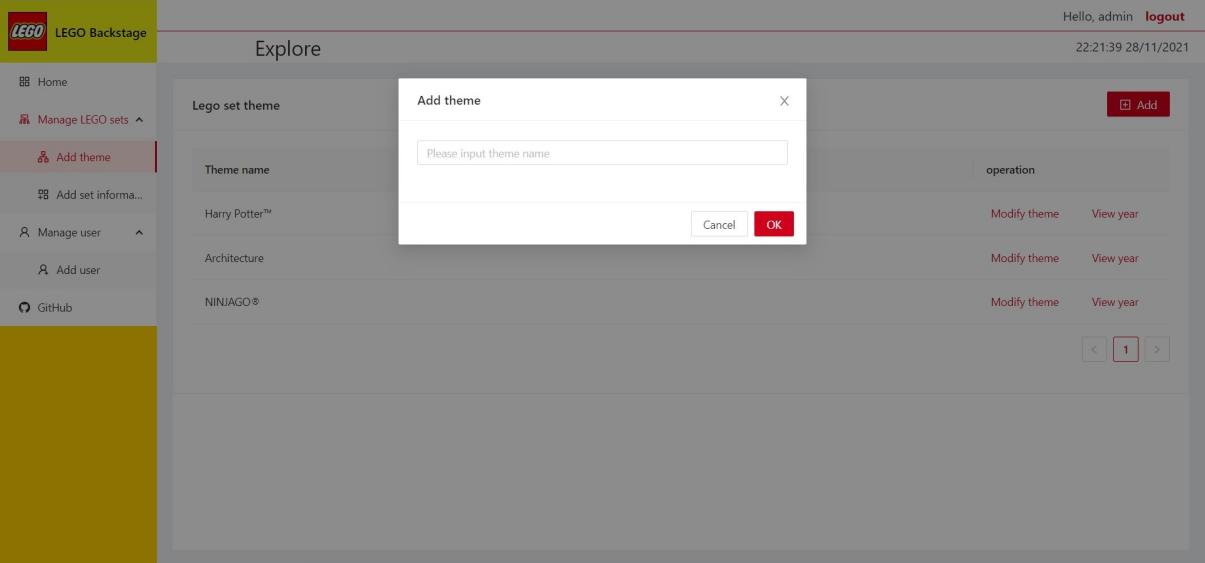
Two functions are mainly involved in this module.

The first function is to add the theme and year of the set. This is used to add detailed information about the set later.

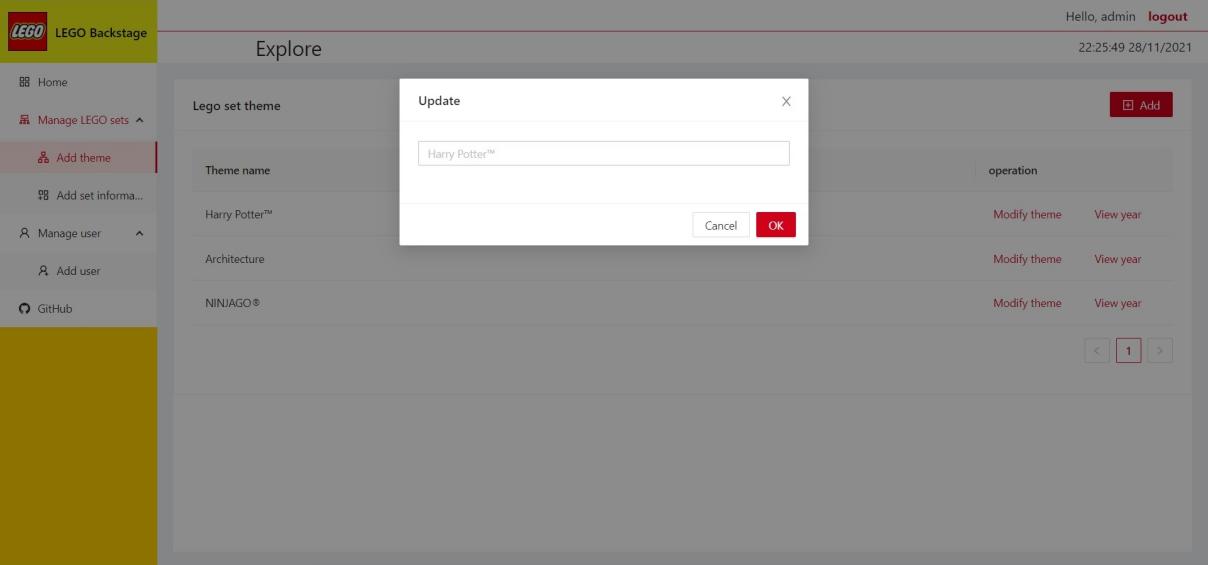
The second function is to add specific information about the set.



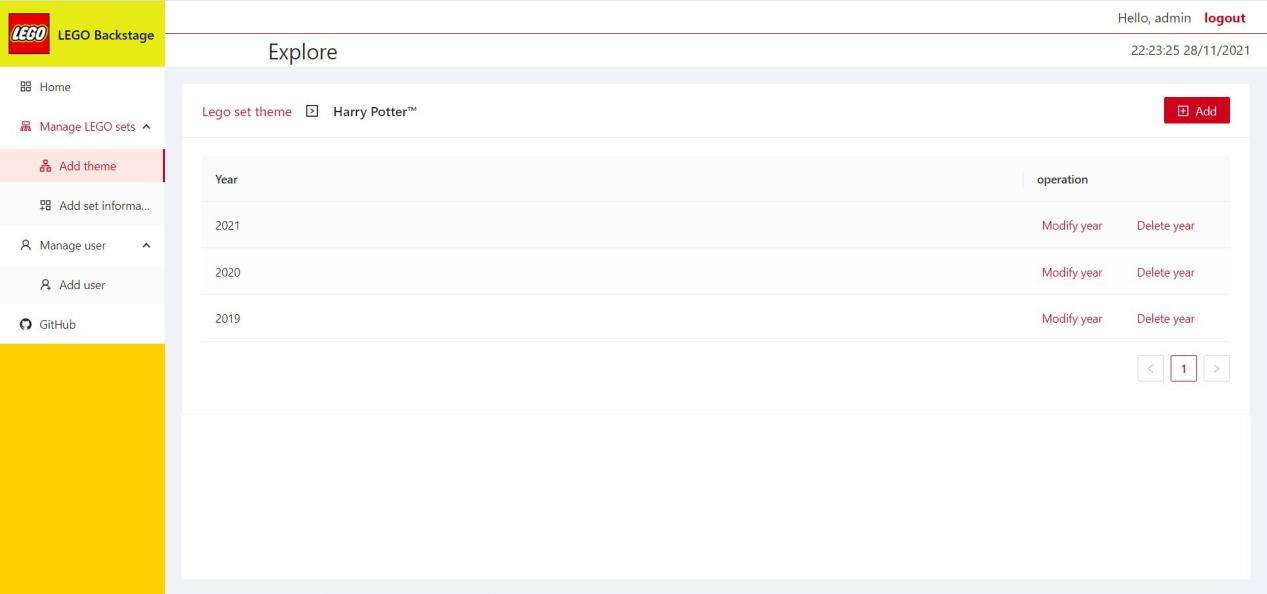
Show overall page of add theme



Show add theme table



Show update theme table

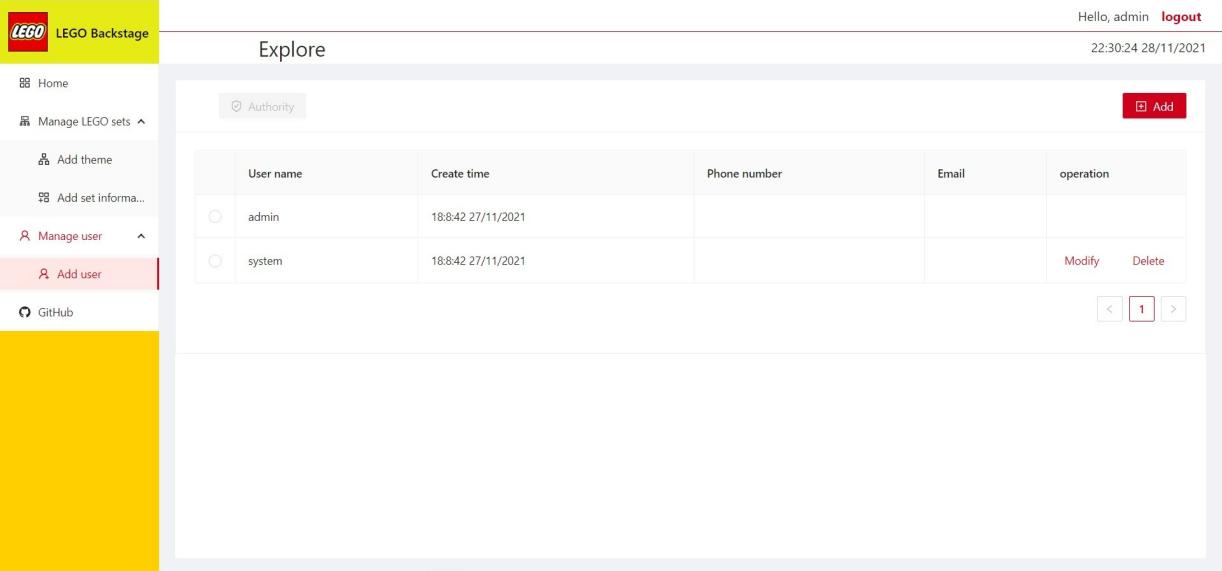


Show add year overall

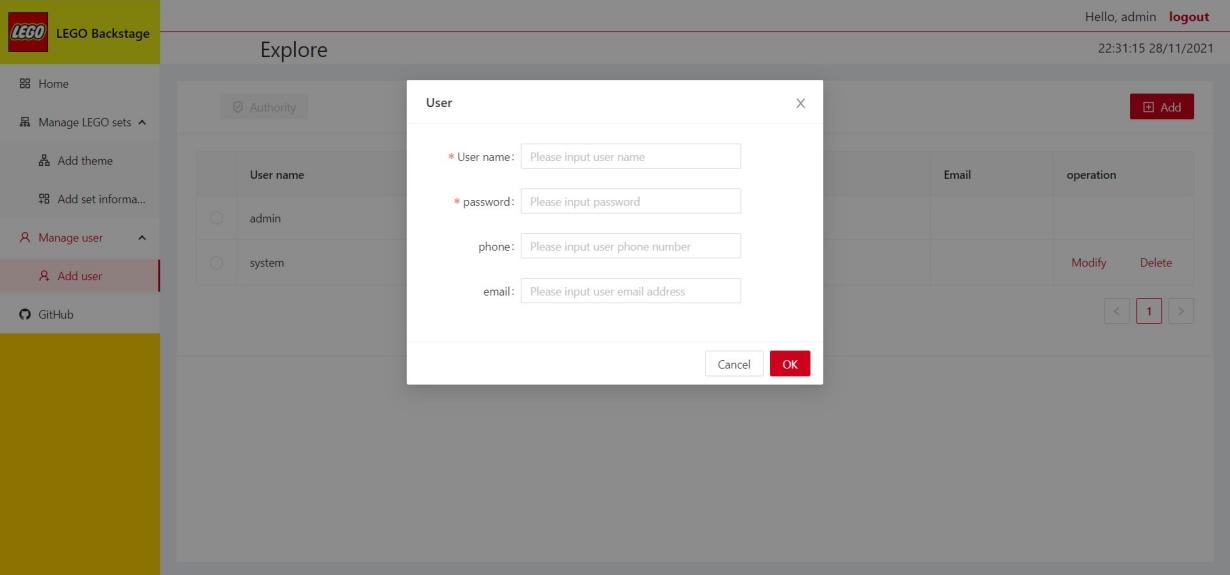
The add and update function is same as above.

1. Manage user:

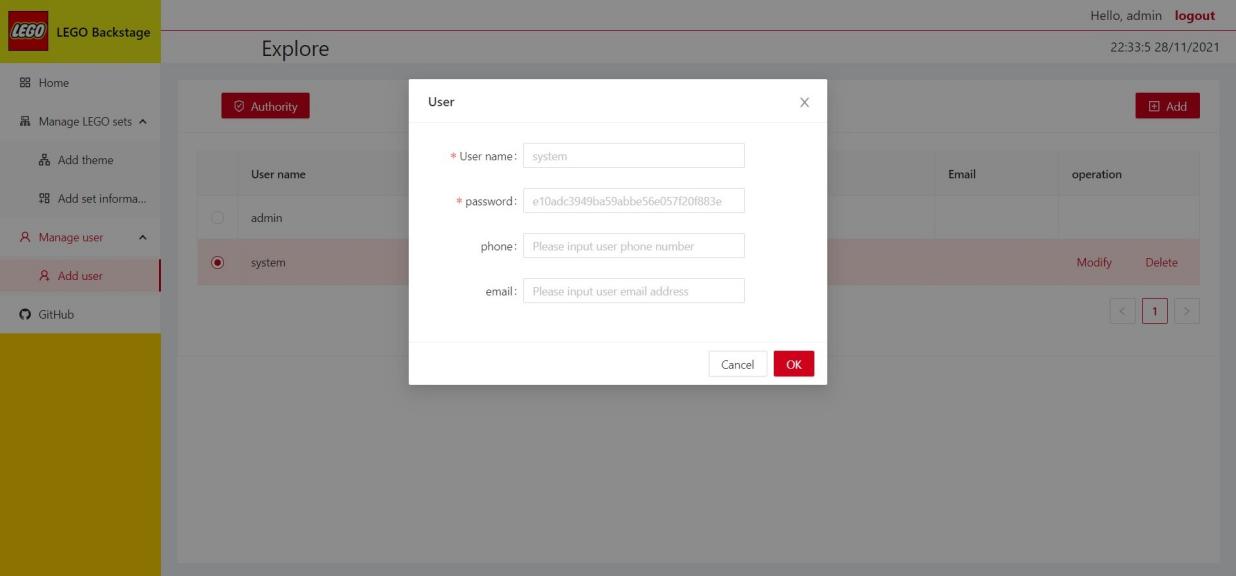
This part mainly implements the functions of adding users, modifying users, deleting users and authorization.



Show overall manage user page



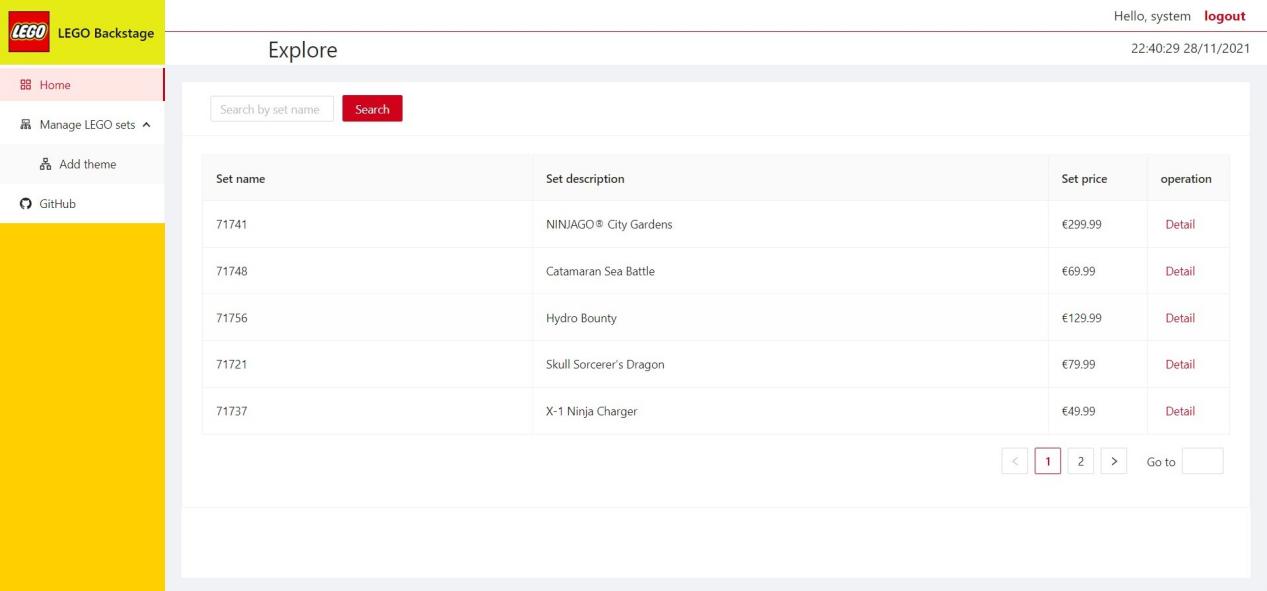
Show add user table



Show update user table

Authorization in the manage user function is use to make log in user can visit some function or not.

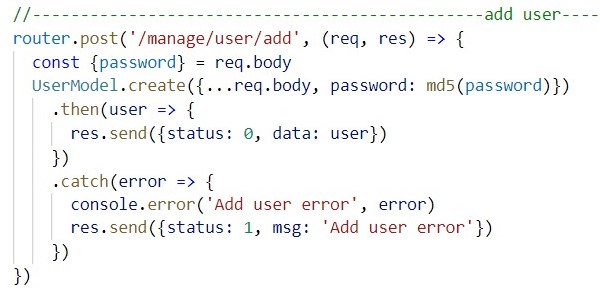
The above user has all auth, so he can visit all the function, the follow picture will show the user who do not have all auth, and can only visit partial function.



Show user without the auth, he can only visit partial function.

The part related to the rubric:

Authentication: In my project, I used MD5 to encryption the password.



And when user logged in the website, this user information will be store in the localStorage, unless you log out, the user will always remain logged in.



Personal and Public areas:

In my website, manage user provide manage user permissions, and manage user information function.

CRUD operations:

Home page provide search function(Select).

Manage LEGO sets page provide add set function(Insert), modify set function(Update) and delete set function(Delete).

Profile management:

In the manage set and manage user function, you can manage set information and user information.

Advanced Search and Filter mechanism:

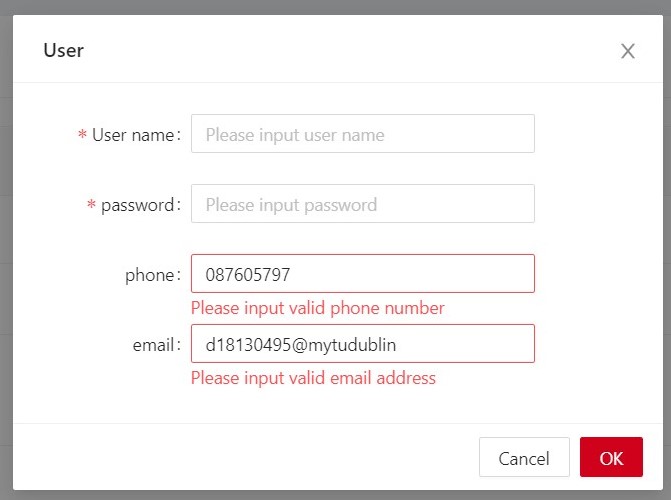
In the home page, search function provide fuzzy query function.

Validation of inputs:

All the input form in this project provide validation function.

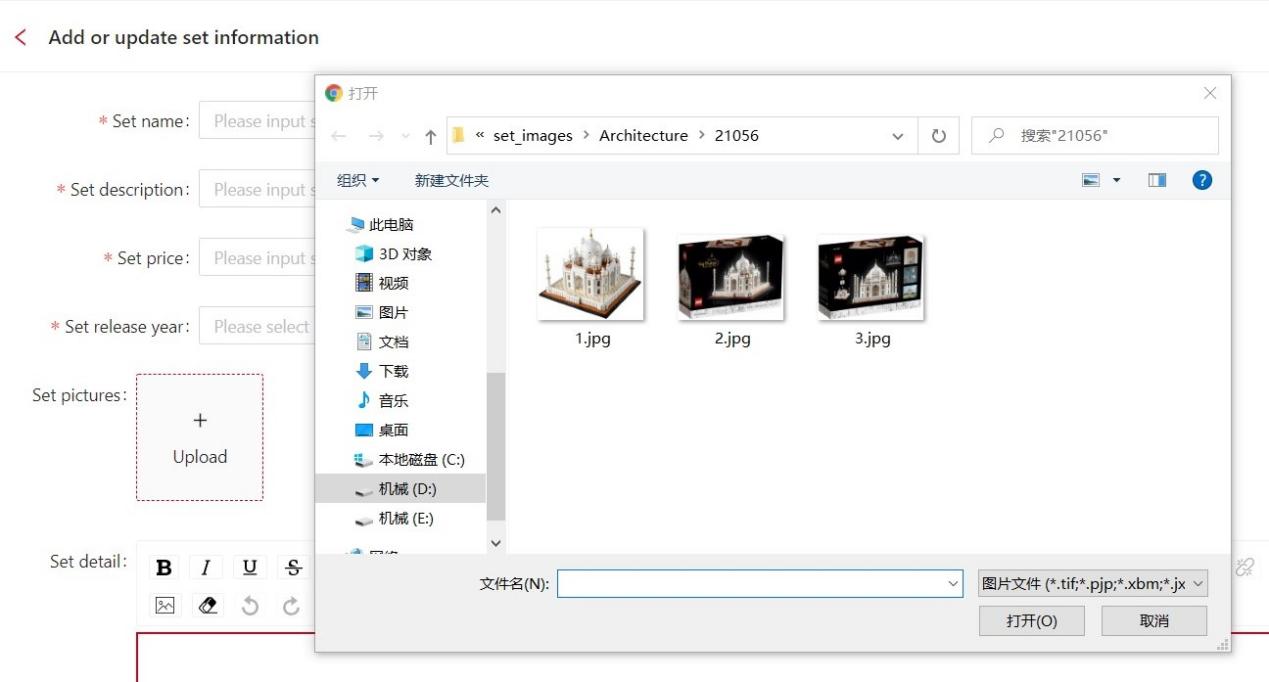
This is add user information table.

The phone number must be 10 digital and email address must be valid email address.



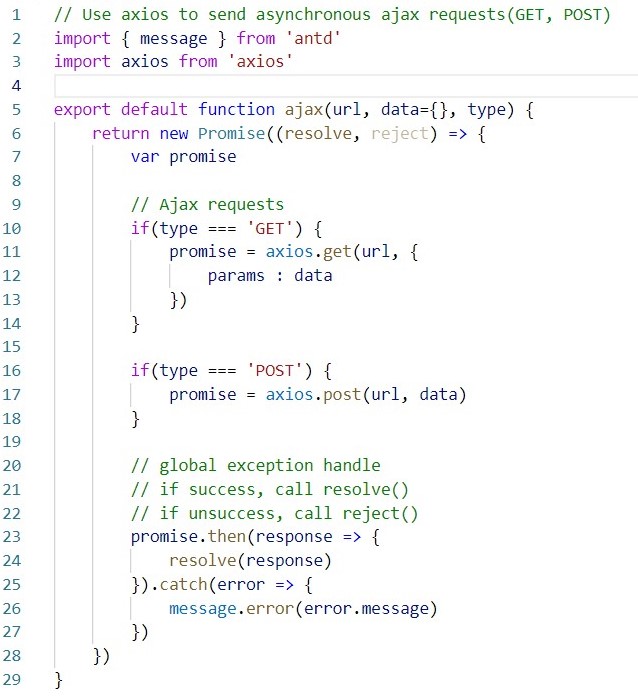
Upload mechanism:

In the add set information page, I provide add set image function.



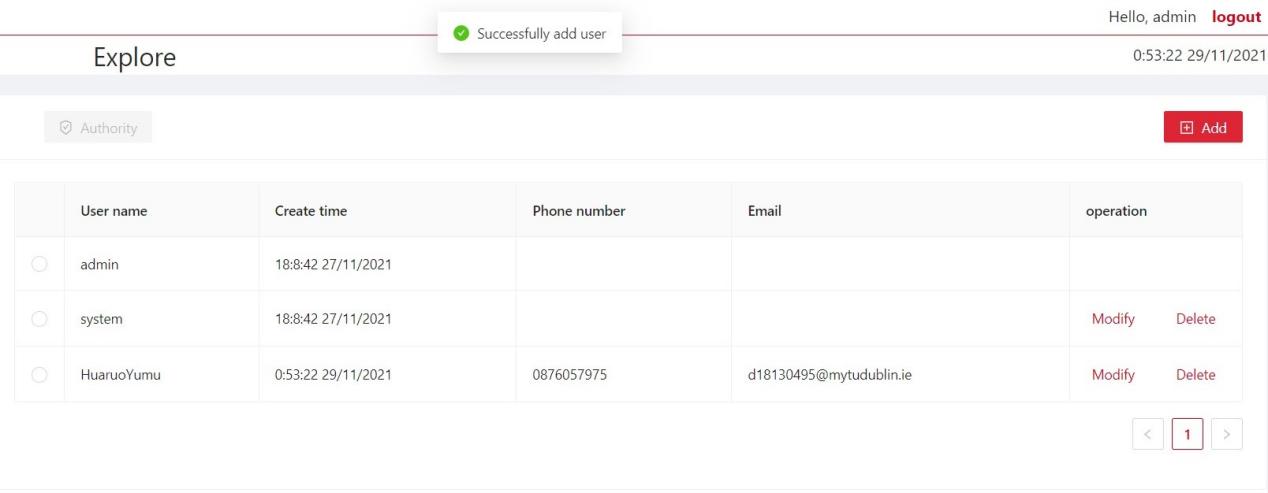
Asynchronous communication:

In my project, I use Axios to send Ajax request.



Communication with the User:

In my project, when you do some operation, whatever you success or fail, the message will display on the screen, and all the input table will have validation.



jQuery and Bootstrap:

In my project, I use Ant Design which is React UI components repository to inside of jQuery and Bootstrap, using a dedicated react component library will be more suitable for the website, more beautiful and more easy to use.

MVC design pattern:

View: The entire front-end is the view part of this project.

Controller: The Controller is in the back-end, use to get the request from front-end, and send respond back to the front-end.

In the back-end, index.js will be the controller.

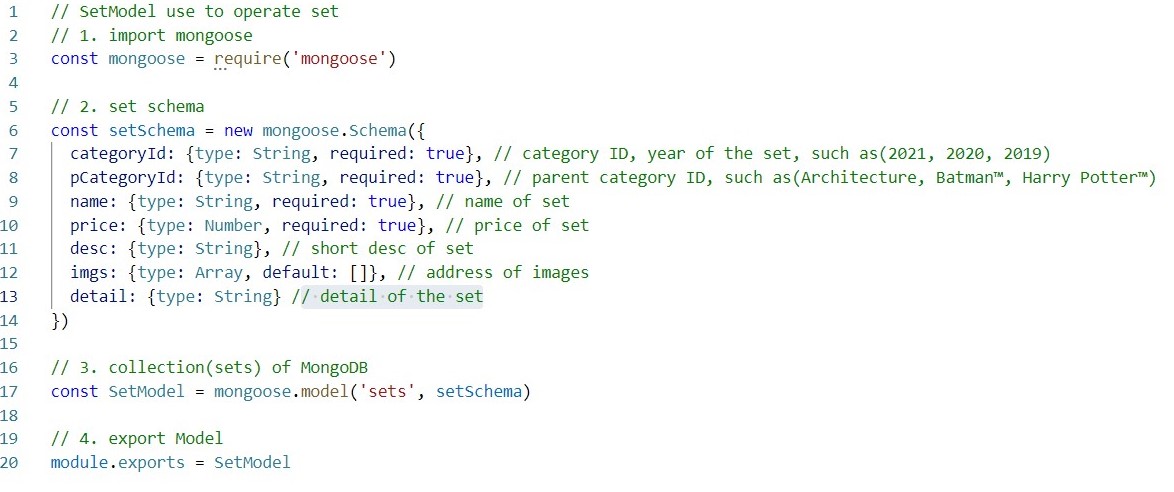


Module:

Module is use to define the format of the data to save to the database.

There are three modules in my project.

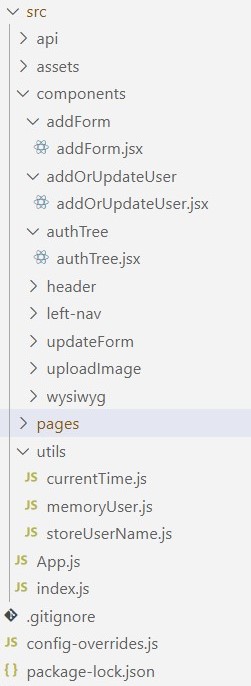




DRY principle:

In my project, I separate some commonly used components into separate components or tool classes.

These components may be used once or multiple times.



Prepared Statements (to avoid SQL injection):

When the project try to connection the DB and get some data, all the query are carry out by mongoose, so all the query request are encapsulated by mongoose and no SQL injection will happen.

Deployment instructions (readme.txt):

The deployment instruction can be find in the readme.docx, because I try to use some image to better show the deployment process.