

Advanced Web Technologies

Library management



Submitted by: Ambika Shrestha

Submitted to :Thomas Broussard

Description

Create a REST API that should be able to manage books from a library.

The following actions should be possible:

- Create a book entry,
- Get all the book entries,
- Modify a book entry,
- Delete a book entry,
- Search all the books thanks to some criteria

System Used

System	Version
Visual Studio Code	Version: 1.53.0
Operating System	Windows 8.1

Book data structure:

Name	Data type
ID	Integer
ISBN	Character varying
Title	Character varying
Author	Character varying
Publication Date	Character varying
Created At	timestamp
Updated At	timestamp

Sn	Actions	React URL	Node Js URL	Parameters
1	Create	http://localhost:3000/api/create	http://127.0.0.1:3000/api/create	<pre>{ "message": "Your book with the title Lord of the rings has been cr eated successfully ", "book": {</pre>

				<pre> "book_id": 27, "title": "Lord of the rings", "author": "J. R. R. Tooklin", "p_date": "2/15/1937", "isbn": "7675789", "updatedAt": "2021-02-10T15:36:37.709Z", "createdAt": "2021-02-10T15:36:37.709Z" } } </pre>
2	Get all List	http://localhost:3000/api/books	http://127.0.0.1:3000/api/books	<pre> [{ "title": "web update ", "author": "Ambika 2", "p_date": "2/10/2021", "isbn": "789456123", "book_id": 20, "createdAt": "2021-02-07T14:44:53.951Z", "updatedAt": "2021-02-10T15:02:01.676Z" }, { "title": "Adding books", "author": "Alisha Shrestha", "p_date": "5/1/2005", "isbn": "772534896", "book_id": 26, "createdAt": "2021-02-10T15:04:52.974Z", "updatedAt": "2021-02-10T15:04:52.974Z" }, { "title": "Lord of the rings", "author": "J. R. R. Tooklin", "p_date": "2/15/1937", "isbn": "7675789", "book_id": 27, "createdAt": "2021-02-10T15:36:37.709Z", </pre>

				<pre> "updatedAt": "2021-02-10T15:36:37.709Z" }] </pre>
3	Modify	http://localhost:3000/api/books/\${this.id}	http://127.0.0.1:3000/api/books/27	<pre> { "message": "Book updated successfully", "data": { "title": "Lord of the ring: Fellowship", "author": "J. R. R. Tooklin", "p_date": "2/15/1937", "isbn": "888888888" } } </pre>
4	Delete	http://localhost:3000/api/books/\${id}	http://127.0.0.1:3000/api/books/22	<pre> { "message": "Book successfully deleted" } </pre>
5	Search	http://localhost:3000/api/search	http://127.0.0.1:3000/api/search/ <pre> { "name": "Lord" } </pre>	<pre> [{ "title": "Lord of the ring: Fellowship", "author": "J. R. R. Tooklin", "p_date": "2/15/1937", "isbn": "888888888", "book_id": 27, "createdAt": "2021-02-10T15:36:37.709Z", "updatedAt": "2021-02-10T15:40:32.287Z" }] </pre>
6	Get by Id	http://localhost:3000/api/getbyid/\${id}	http://127.0.0.1:3000/api/getbyid/27	<pre> { "title": "Lord of the ring: Fellowship", "author": "J. R. R. Tooklin", "p_date": "2/15/1937", "isbn": "888888888", "book_id": 27, </pre>

				<pre>"createdAt": "2021-02-10T15:36:37.709Z", "updatedAt": "2021-02-10T15:40:32.287Z" }</pre>
--	--	--	--	---

Database

Name	PostgreSQL
------	------------

pgAdmin

FileObjectToolsHelp

Browser

public

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Procedures

Sequences

Tables (1)

Books

Columns (7)

title

author

p_date

isbn

book_id

createdAt

updatedAt

Constraints

DashboardPropertiesSQLStatisticsDependenciesDependentsbookstore/postgres@library_management *

bookstore/postgres@library_management

Query EditorQuery History

1select * from "Books"

Data OutputExplainMessagesNotifications

	title	author	p_date	isbn	book_id	createdAt
	character varying (255)	character varying (255)	character varying (255)	character varying (255)	[PK] integer	timestamp with time zone
1	Two Souls	Dinnese White	9/12/2000	254789555	18	2021-02-07 15:27:12.409+01
2	The life of old man	old man	12/12/1997	2547856	22	2021-02-07 19:29:43.71+01
3	Web Technology	Ambika Shrestha	2/10/2021	789456123	25	2021-02-10 16:00:33.115+01
4	web update	Ambika 2	2/10/2021	789456123	20	2021-02-07 15:44:53.951+01
5	Adding books	Alisha Shrestha	5/1/2005	772534896	26	2021-02-10 16:04:52.974+01

pgAdmin

FileObjectToolsHelp

Browser

public

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Procedures

Sequences

Tables (1)

Books

Columns (7)

title

author

p_date

isbn

book_id

createdAt

updatedAt

Constraints

DashboardPropertiesSQLStatisticsDependenciesDependentsbookstore/postgres@library_management *

bookstore/postgres@library_management

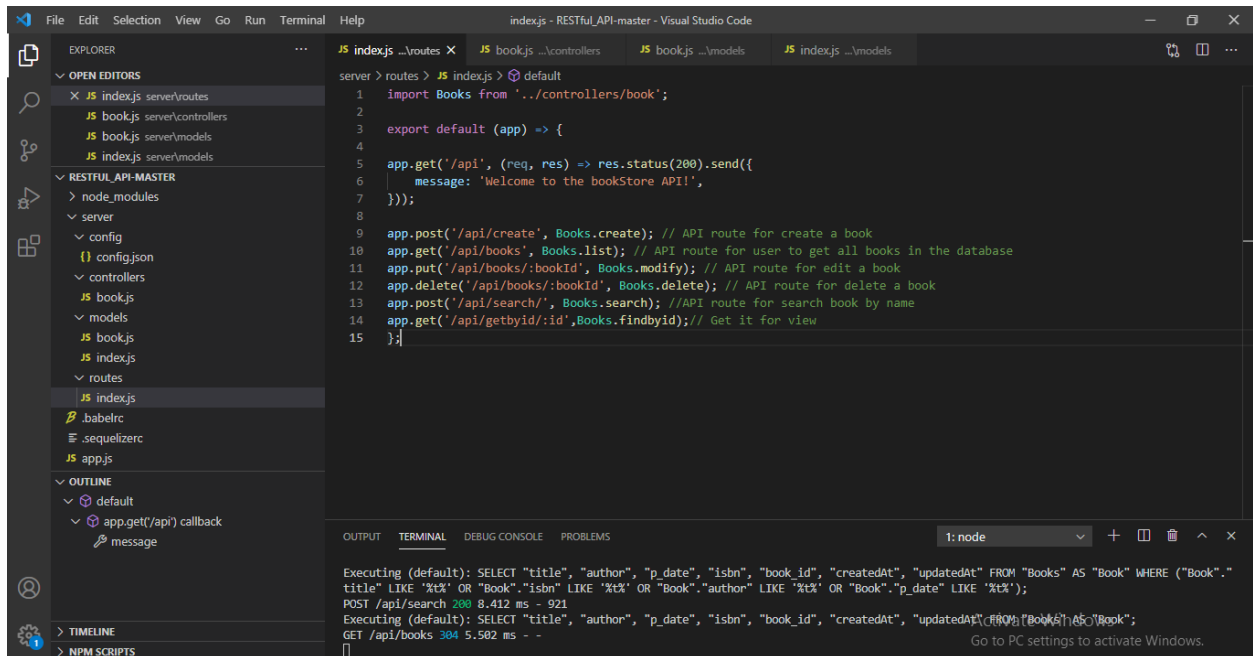
Query EditorQuery History

1select * from "Books"

Data OutputExplainMessagesNotifications

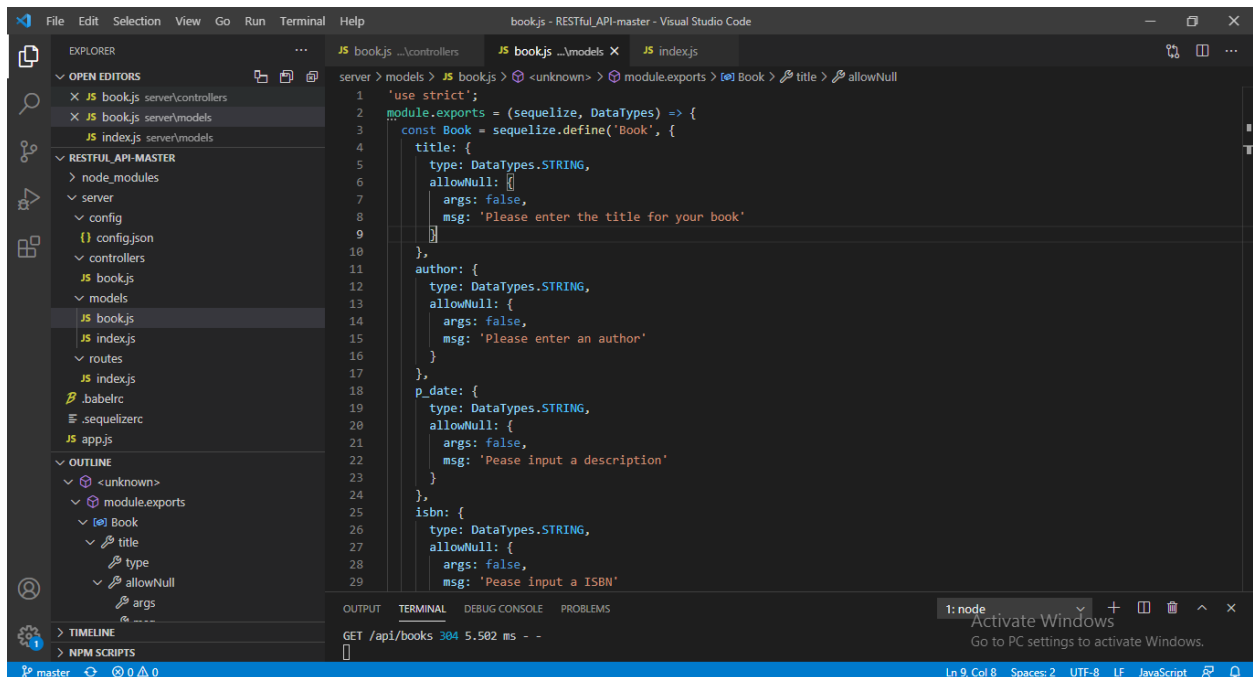
	title	author	p_date	isbn	book_id	createdAt
	character varying (255)	character varying (255)	character varying (255)	character varying (255)	[PK] integer	timestamp with time zone
1	Two Souls	Dinnese White	9/12/2000	254789555	18	2021-02-07 15:27:12.409+01
2	Web Technology	Ambika Shrestha	2/10/2021	789456123	25	2021-02-10 16:00:33.115+01
4	Adding books	Alisha Shrestha	5/1/2005	772534896	26	2021-02-10 16:04:52.974+01
5	Lord of the Ring: Fellowship	J. R. R. Tooklin	2/15/1937	888888888	27	2021-02-10 16:36:37.709+01

Screenshots backend



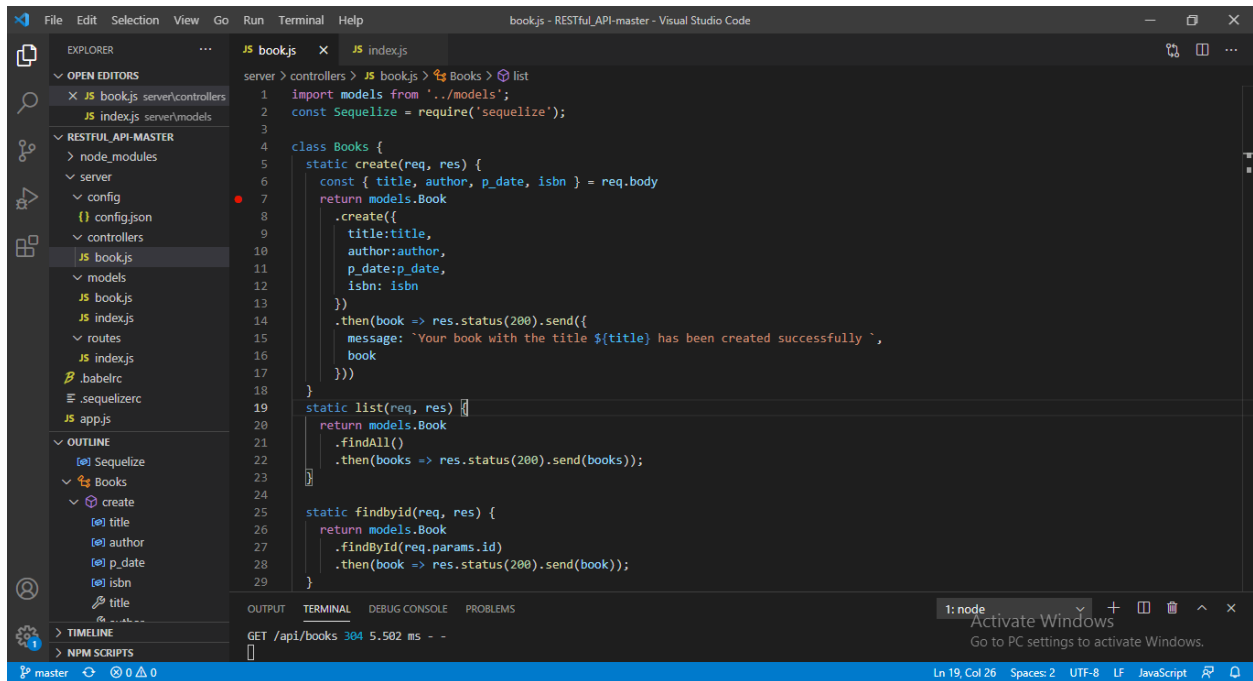
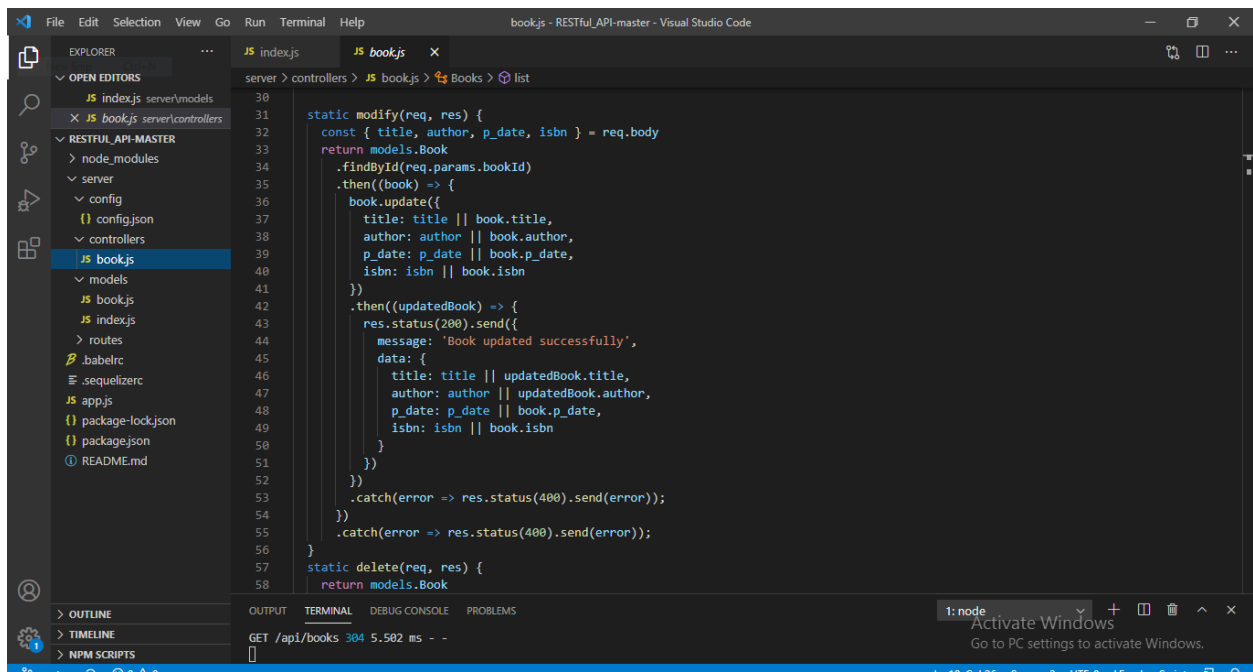
```
server > routes > JS indexjs > default
1 import Books from '../controllers/book';
2
3 export default (app) => {
4
5   app.get('/api', (req, res) => res.status(200).send({
6     message: 'Welcome to the bookStore API!',
7   }));
8
9   app.post('/api/create', Books.create); // API route for create a book
10  app.get('/api/books', Books.list); // API route for user to get all books in the database
11  app.put('/api/books/:bookId', Books.modify); // API route for edit a book
12  app.delete('/api/books/:bookId', Books.delete); // API route for delete a book
13  app.post('/api/search/', Books.search); //API route for search book by name
14  app.get('/api/getbyid/:id', Books.findbyid); // Get it for view
15 }
```

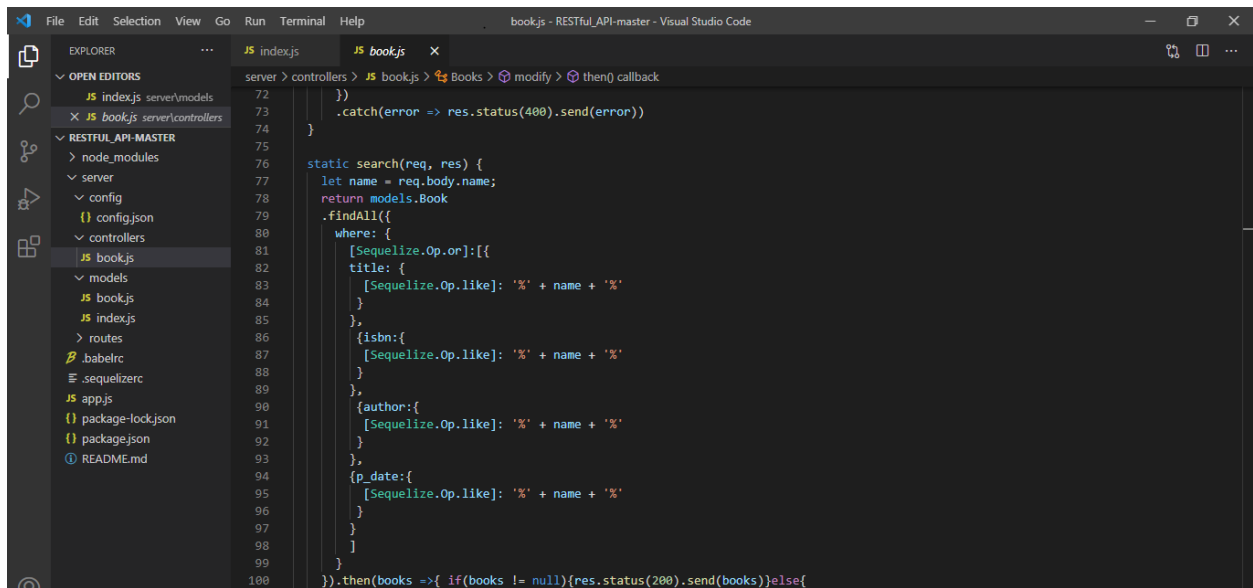
Executing (default): SELECT "title", "author", "p_date", "isbn", "book_id", "createdAt", "updatedAt" FROM "Books" AS "Book" WHERE ("Book"."title" LIKE "%%" OR "Book"."isbn" LIKE "%%" OR "Book"."author" LIKE "%%" OR "Book"."p_date" LIKE "%%");
POST /api/search 200 8.412 ms - 921
Executing (default): SELECT "title", "author", "p_date", "isbn", "book_id", "createdAt", "updatedAt" FROM "Books" AS "Book";
GET /api/books 304 5.582 ms - -



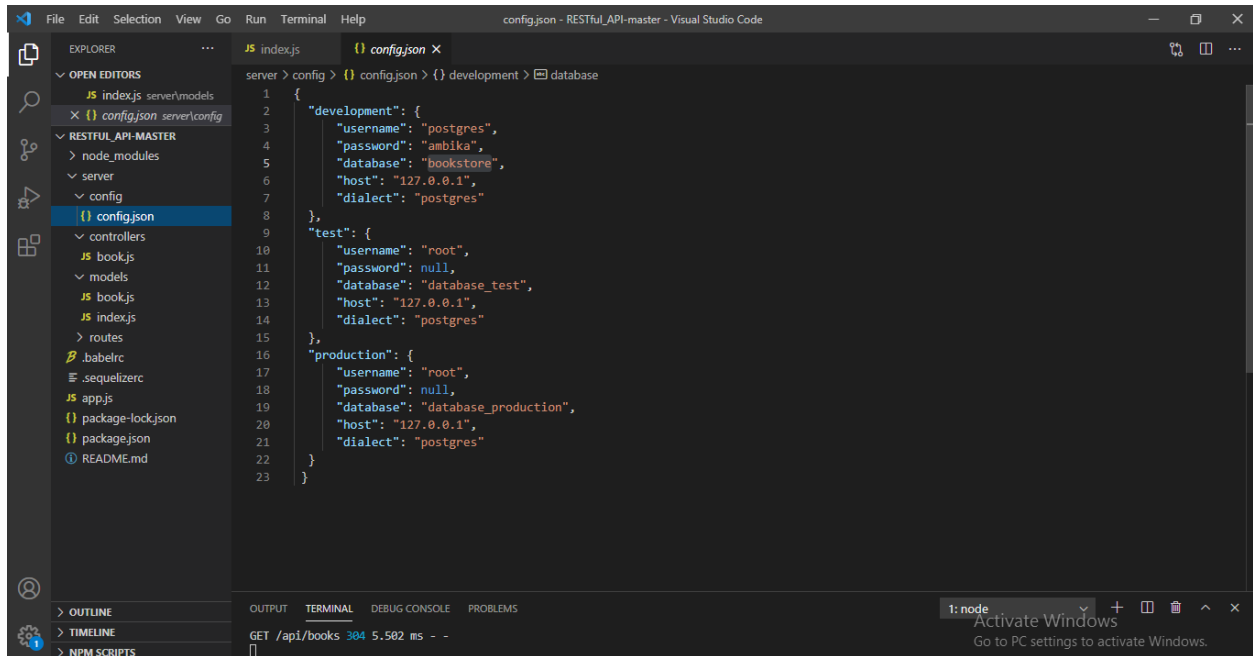
```
server > models > JS bookjs > <unknown> > module.exports > Book > title > allowNull
1 'use strict';
2 module.exports = (sequelize, DataTypes) => {
3   const Book = sequelize.define('Book', {
4     title: {
5       type: DataTypes.STRING,
6       allowNull: {
7         args: false,
8         msg: 'Please enter the title for your book'
9       }
10    },
11    author: {
12      type: DataTypes.STRING,
13      allowNull: {
14        args: false,
15        msg: 'Please enter an author'
16      }
17    },
18    p_date: {
19      type: DataTypes.STRING,
20      allowNull: {
21        args: false,
22        msg: 'Pease input a description'
23      }
24    },
25    isbn: {
26      type: DataTypes.STRING,
27      allowNull: {
28        args: false,
29        msg: 'Pease input a ISBN'
30      }
31    }
32  }, {
33    sequelize: sequelize,
34    tableName: 'Books'
35  });
36  return Book;
37 }
```

GET /api/books 304 5.582 ms - -





```
server > controllers > JS bookjs > Books > modify > then() callback
72  })
73  .catch(error => res.status(400).send(error))
74  }
75
76  static search(req, res) {
77    let name = req.body.name;
78    return models.Book
79    .findAll({
80      where: {
81        [Sequelize.Op.or]: [{
82          title: {
83            [Sequelize.Op.like]: '%' + name + '%'
84          }
85        },
86        {isbn: {
87          [Sequelize.Op.like]: '%' + name + '%'
88        }
89      },
90      {author: {
91        [Sequelize.Op.like]: '%' + name + '%'
92      }
93    },
94    {p_date: {
95      [Sequelize.Op.like]: '%' + name + '%'
96    }
97  }
98  ]
99  }
100  }).then(books => { if(books != null){res.status(200).send(books)}else{
```



```
server > config > {} configjson > {} development > database
1  {
2    "development": {
3      "username": "postgres",
4      "password": "ambika",
5      "database": "bookstore",
6      "host": "127.0.0.1",
7      "dialect": "postgres"
8    },
9    "test": {
10     "username": "root",
11     "password": null,
12     "database": "database_test",
13     "host": "127.0.0.1",
14     "dialect": "postgres"
15   },
16   "production": {
17     "username": "root",
18     "password": null,
19     "database": "database_production",
20     "host": "127.0.0.1",
21     "dialect": "postgres"
22   }
23 }
```

1: node
Activate Windows
Go to PC settings to activate Windows.

Screenshots frontend

1. Create

Library

Add Books

Add new books

Title

Adding books

Author

Alisha Shrestha

Publication Date

5/1/2005

ISBN

772534896

Submit

2. Update

Library

Add Books

Search..

Update books

Title

web update

Author

Ambika 2

Publication Date

2/10/2021

ISBN

789456123

Update

Activate Windows
Go to PC settings to activate Windows.

3. Getting list

Library		Add Books				Search..	
ID	Book Name	ISBN	Author	Publication Date	Action		
20	et	rwwt	rtt	re	Delete	Update	
19	qq	ta	go to	2/15/1	Delete	Update	
21	tt	7675789	Ambiakkkkk	2/15/19941	Delete	Update	
23	Harry porter	205415151	J K Rolling	2/15/1994	Delete	Update	
24	Jojo's Adventure	78542358	Jojo grandpa	2/15/1997	Delete	Update	
18	Two Souls	254789555	Dinnese White	9/12/2000	Delete	Update	
22	The life of old man	2547856	old man	12/12/1997	Delete	Update	

4. Search by ISBN

Library		Add Books				25	
ID	Book Name	ISBN	Author	Publication Date	Action		
18	Two Souls	254789555	Dinnese White	9/12/2000	Delete	Update	
22	The life of old man	2547856	old man	12/12/1997	Delete	Update	
26	Adding books	772534896	Alisha Shrestha	5/1/2005	Delete	Update	

5. Search by title (book name)

Library

Add Books

Two |

ID	Book Name	ISBN	Author	Publication Date	Action
18	Two Souls	254789555	Dinnese White	9/12/2000	<div>DeleteUpdate</div>

Library

Add Books

11

ID	Book Name	ISBN	Author	Publication Date	Action
18	Two Souls	254789555	Dinnese White	9/12/2000	<div>DeleteUpdate</div>
22	The life of old man	2547856	old man	12/12/1997	<div>DeleteUpdate</div>
25	Web Technology	789456123	Ambika Shetha	2/10/2021	<div>DeleteUpdate</div>

6. Search by Author.

Library

Add Books

old man|

ID	Book Name	ISBN	Author	Publication Date	Action
22	The life of old man	2547856	old man	12/12/1997	<div>DeleteUpdate</div>

7. Delete

Library

Add Books

Search..

ID	Book Name	ISBN	Author	Publication Date	Action
18	Two Souls	254789555	Dinnese White	9/12/2000	<div>DeleteUpdate</div>
22	The life of old man	2547856	old man	12/12/1997	<div>DeleteUpdate</div>
25	Web Technology	789456123	Ambika Shestha	2/10/2021	<div>DeleteUpdate</div>
20	web update	789456123	Ambika 2	2/10/2021	<div>DeleteUpdate</div>
26	Adding books	772534896	Alisha Shrestha	5/1/2005	<div>DeleteUpdate</div>

Library

Add Books

Search..

ID	Book Name	ISBN	Author	Publication Date	Action
18	Two Souls	254789555	Dinnese White	9/12/2000	<div>DeleteUpdate</div>
25	Web Technology	789456123	Ambika Shestha	2/10/2021	<div>DeleteUpdate</div>
20	web update	789456123	Ambika 2	2/10/2021	<div>DeleteUpdate</div>
26	Adding books	772534896	Alisha Shrestha	5/1/2005	<div>DeleteUpdate</div>
27	Lord of the ring: Fellowship	888888888	J. R. R. Tooklin	2/15/1937	<div>DeleteUpdate</div>

Errors Faced

SyntaxError: Unexpected end of JSON input
React-router v4 this.props.history.push(...) not working
Failed form propTypes: You provided a `value` prop to a form field without an `onChange` handler
Uncaught ReferenceError: Change Handler is not defined
onChangeHandler' is not defined no-undef
Error: is not defined react/jsx-no-undef
ReferenceError: sequelize is not defined
TypeError: Cannot read property 'create' of undefined
Cannot find module 'moment'
Invalid value Error: while executing Find Query

Link

<https://github.com/Ambika-shrestha/LibraryFrontend-React>

<https://github.com/Ambika-shrestha/RestApiLibrary>