ID: 258744

Book-E-Sale

A SUMMER INTERNSHIP REPORT

Submitted by

Devarshi Pradipbhai Trivedi

190470107070

In partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

In

Computer Engineering

V.V.P. ENGINEERING COLLEGE, Rajkot

Gujarat Technological University, Ahmedabad

July 2022

GUJARAT TECHNOLOGICAL UNIVERSITY

CERTIFICATE FOR COMPLETION OF ALL ACTIVITIES AT ONLINE PROJECT PORTAL

B.E. SEMESTER VII, ACADEMIC YEAR 2021-2022

Date of certificate generation : 19 November 2022 (11:53:55)

This is to certify that, ***Trivedi Devarshi Pradipbhai*** (Enrolment Number - 190470107070) working on project entitled with ***Book-E-Sale*** from ***Computer Engineering*** department of ***VYAVASAYI VIDYA PRATISHTHANS SANCH. COLLEGE OF ENGINEERING, RAJKOT*** had submitted following details at online project portal.

Internship Project Report	**Completed**

Name of Student : Trivedi Devarshi Pradipbhai

Name of Guide : Miss. Shivangi Kiritbhai Bakori

Signature of Student : ____________________

*Signature of Guide : ____________________

Disclaimer :
This is a computer generated copy and does not indicate that your data has been evaluated. This is the receipt that GTU has received a copy of the data that you have uploaded and submitted as your project work.

*Guide has to sign the certificate, Only if all above activities has been Completed.

ID: 258744

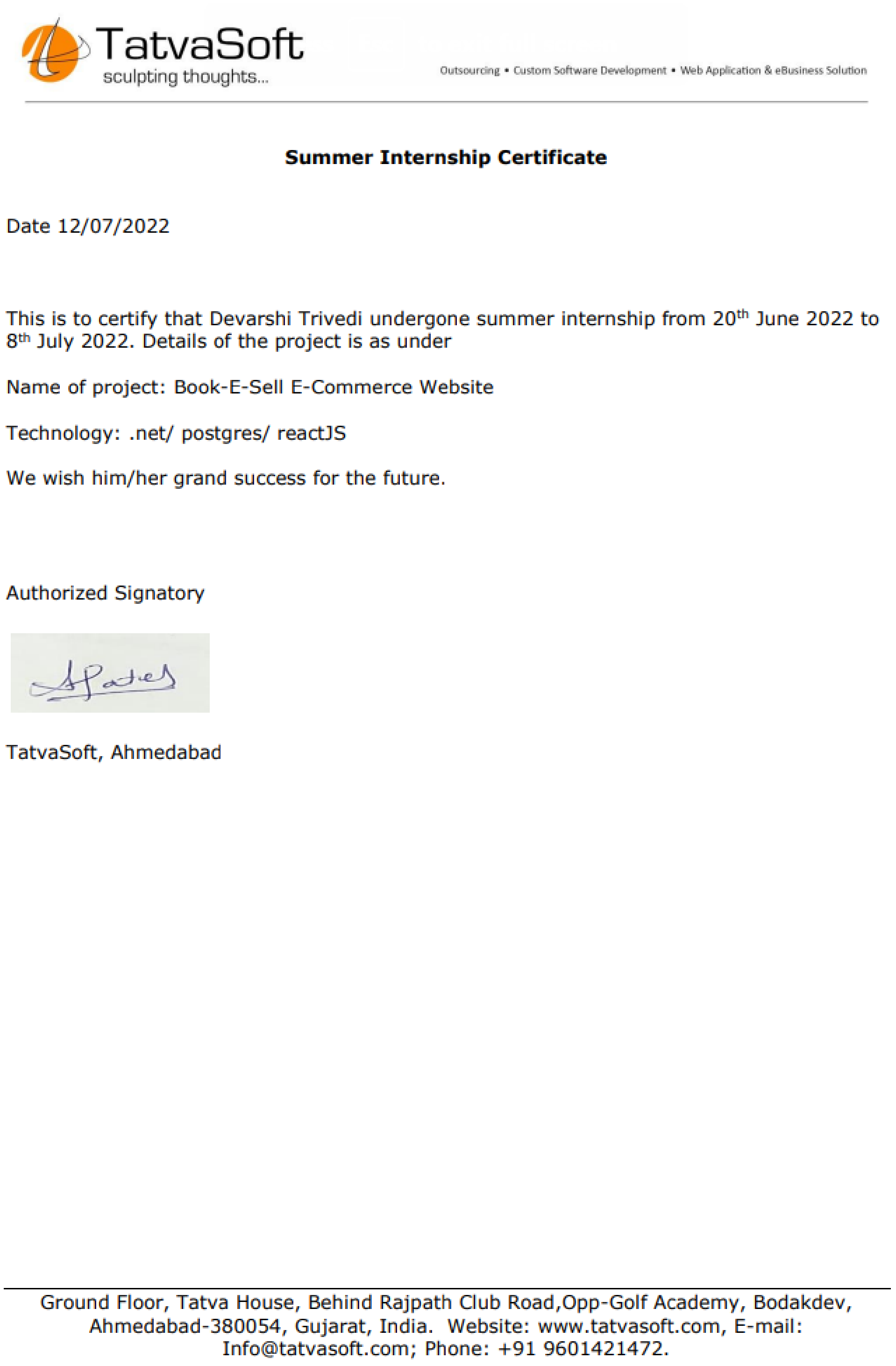
Vyavsayi Vidhya Pratishthan Engineering College

Kalavad Road Virda Vajadi, Rajkot, Gujarat 360005

CERTIFICATE

This is to certify that the internship report submitted along with the project entitled **Book-E-Sale** has been carried out by **Devarshi Pradipbhai Trivedi 190470107070** under my guidance in partial fulfilment for the degree of Bachelor of Engineering in **Computer Engineering**, 7th Semester of Gujarat Technological University, Ahmadabad during the academic year 2022-23.

Miss. Shivangi Kiritbhai Bakori	Dr. Tejas Patalia
Internal Guide	Head of the Department



TatvaSoft
sculpting thoughts...

Outsourcing • Custom Software Development • Web Application & eBusiness Solution

Summer Internship Certificate

Date 12/07/2022

This is to certify that Devarshi Trivedi undergone summer internship from 20th June 2022 to 8th July 2022. Details of the project is as under

Name of project: Book-E-Sell E-Commerce Website

Technology: .net/ postgres/ reactJS

We wish him/her grand success for the future.

Authorized Signatory

[illegible]

TatvaSoft, Ahmedabad

Ground Floor, Tatva House, Behind Rajpath Club Road,Opp-Golf Academy, Bodakdev, Ahmedabad-380054, Gujarat, India. Website: www.tatvasoft.com, E-mail: Info@tatvasoft.com; Phone: +91 9601421472.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion task would be incomplete without the mention of the people who made it possible, whose constant guidance, support and encouragement crown all the efforts with the success.

Our sincere thanks to Principal Dr. JAYESH DESHKAR Sir, H.O.D. of Computer Engineering Dr. TEJAS PATALIA Sir for having consented to be the guide and for their valuable guidance and support during the preparation of this internship.

Also, Miss. Shivangi Kiritbhai Bakori of CE Department helped us to work out on the software side of our project. Last but not the least, my industry mentor Ankit Khunt and Sweety Patel helped a lot for this project. My sincere dedication and keen desire to learn something new helped us to achieve success in the project.

I would also like to thank to GOD, Our Family and Friends who have been a constant source of inspiration.

Thank You

Devarshi P. Trivedi

ABSTRACT

Book-E-Sale is an E-commerce website for book purchasing and selling. It contains basic functionalities such as Login, Registration, Book Search, Book listing, Add to Cart, and Logout. It has buyer, seller and admin module. User can become seller or buyer. Website help people to find the book they need. People can sell their non-used books online. It follows C2C Business model.

List Of Figures

Figure No.	Title	Page No.
Fig 2.4	Basic TypeScript Program-1	10
Fig 2.5	Basic TypeScript Program Output-1	10
Fig 2.6	Basic TypeScript Program-2	11
Fig 2.7	Basic TypeScript Program Output-2	11
Fig 2.8	React with TypeScript Program	12
Fig 2.9	React Formik Program Output	13
Fig 2.10	ASP.NET Middleware code	14
Fig 2.11	ASP.NET Middleware Code Output	14
Fig 2.12	PostgreSQL with pgAdmin	15
Fig 2.13	CRUD APIS for User	16
Fig 2.14	Read Operation for User	16
Fig 2.15	Read with id fetch particular user’s details	17

Fig 2.16	Update Operation for User	17
Fig 2.17	Delete Operation for User	18
Fig 2.18	Login and Register APIs	19
Fig 2.19	Unsuccessfully Login	19
Fig 2.20	Successful Login	20
Fig 2.21	Registration of User	20
Fig 2.22	CRUD Operations for Book table	21
Fig 2.23	Global Search Backend Implementation	21
Fig 2.24	Read Operation for Book	22
Fig 2.25	Add Operation for Book	22
Fig 2.26	Update Operation for Book	23
Fig 2.27	Delete Operation for Book	23
Fig 2.28	CRUD Operations on Category	24
Fig 2.29	Read Operation on Category	24
Fig 2.30	Fetching particular category	25
Fig 2.31	Add Operation on Category	25
Fig 2.32	Delete Operation on Category	26

Fig 2.33	CRUD Operations on Cart	27
Fig 2.34	Read Operation on Cart	27
Fig 2.35	Add Operation on Cart	28
Fig 2.36	Update Operation on Cart	28
Fig 2.37	Delete Operation on Cart	29
Fig 2.38	CRUD Operations on Publisher	30
Fig 2.39	Read Operation on Publisher	30
Fig 2.40	Fetching Publisher through id	31
Fig 2.41	Add Operation on Publisher	31
Fig 2.42	Update Operation on Publisher	32
Fig 2.43	Delete Operation on Publisher	32
Fig 2.44	Registration Frontend Module	33
Fig 2.45	Successful Registration	33
Fig 2.46	Incorrect User Details	34
Fig 2.47	Successful Login	34
Fig 2.48	Book Listing Module	35

Fig 2.49	Admin Module	35
Fig 2.50	Book-Listing Page	36
Fig 2.51	Seller Module	36
Fig 2.52	Global Search Implementation	37
Fig 2.53	Website Header Search	37
Fig 2.54	Successful adding in Cart	38

List Of Tables

Figure No.	**Title**	**Page No.**
Table 2.1	Week-1 Activities Table	8
Table 2.2	Week-2 Activities Table	8
Table 2.3	Week-3 Activities Table	9

Table of Contents

Sr. No	**Chapter No**	**Topic Name**		**Page No.**
1		Title Page		I
2		College Certificate		II
3		Internship Certificate		III
4		Acknowledgement		IV
5		Abstract		V
6		List of Figures		VI
7		List of Tables		X
8		Table of Contents		XI
9	**Chapter: 1**	**Introduction**		1
10		1.1	Objective of Internship	1
11		1.2	About the company	2
12		1.3	About Technology	3
13		1.4	Project Information	4
14	**Chapter: 2**	**Roles and Responsibilities during internship**		6
15		2.1	Problems Given	6
16		2.2	My Roles and Responsibilities	7
17		2.3	Daily Tasks and Activities	8
18	**Chapter: 3**	**Skills Learned**		39
19		3.1	About the Skills	39
20		3.2	How do I learn the skills	43
21	**Chapter: 4**	**Overall Experience**		45

22		4.1	Technical Experience	45
23		4.2	Personal Experience	46
24	**Chapter: 5**	**Conclusion**		47
25		5.1	Conclusion	47
26	**Chapter: 6**	**Future Scope**		48
27		6.1	Future Scope	48
28	**Chapter: 7**	**Bibliography**		49
29		7.1	Bibliography	49

CHAPTER-1: INTRODUCTION

1.1 OBJECTIVE OF INTERNSHIP

- Internships are generally thought of to be reserved for college students looking togain experience in a particular field. However, a wide array of people can benefitfrom Training Internships to receive real-world experience and develop their skills.

- An objective for this position should emphasize the skills you already possess inthe area and your interest in learning more.

- Internships are utilized in many different career fields, including architecture, engineering, healthcare, economics, advertising, and many more.

- Some internship is used to allow individuals to perform scientific research whileothers are specifically designed to allow people to gain first-hand experience working.

- Utilizing internships is a great way to build your resume and develop skills that canbe emphasized in your resume for future jobs.

1.2 ABOUT THE COMPANY

TatvaSoft is Ahmedabad, India, based Consumer Custom Software Development company delivering splendid business IT Solutions and relatedservices to customers across the globe. Their development services are led bytheir dedicated and passionate team to provide best industry practices combined with technical expertise and business domain knowledge to drivedigital transformation. Our proficiency in understanding business challenges and professional competence allows them to create a better experience for their customers.

They excel in delivering the best-suited solution as per the custom needs, be itsmall start-ups in their ideation phase or mid-size businesses focusing on growth or large enterprises actively optimizing processes across varied industries like Fintech & Insurance, Oil & Gas, Mining, Education, Retail & Ecommerce, Energy and Utilities, Logistics & Distribution, Healthcare, Travel & Hospitality, Media & Entertainment, Public Sector.

TatvaSoft design and implement advanced custom software solutions and mobile apps to simplify your business problems. Focussing on the latest technologies, agile methodologies & DevOps, they offer cost-effective digitalsolutions for you to innovate and optimize your business performance.

1.3 ABOUT TECHNOLOGY

This Project is a Small E-Book selling website using ReactJS,TypeScript, ASP.net, and PostgreSQL. It is a website to buy and sell books. It has various functionalities like login, registration, Addition of book to cart, etc…

1.4 PROJECT INFORMATION:

I’m working on Book-E-Sale System as an intern. E-Book Store System is a simple project similar to a shopping cart or eCommerce website but is only for book shopping and selling books. Through a web browser the customers need to login or register later can search for a book by its title or author, later can add it to the shopping cart, and finally purchase the books. And the sellers can add books for selling.

In this project, we work on some modules like:

For Admin:

A. Dashboard – For the admin dashboard, you will be able to all the basic access in the whole system. Such as a summary of products, orders, and categories.

B. Manage Books– The admin has access to the books management information system. He can add, update, and delete the books.

C. Manage Categories – The page where the admin can add, edit, and delete category information.

D. Manage Orders – As the main function of the admin, the admin can accept or reject the order from the customers on a case-to-case basis and the list of customer orders is listed.

E. Manage User– The admin can manage the user’s account. Admin can add, update and Block users in the system.

For Buyer:

A. Login Page – Customer enters their website credentials on this page to gain access to log in.

B. Register Page– The page where new customer created their login credentials for the website.

C. Home Page– When a customer visits the website, this is the system’s default page. This page shows the books for sale in the store, or by entering a keyword in the search box above the books.

D. Book View Page – The page on which the product’s specific information is shown, as well as the page on which the customer adds the product to his or her cart.

E. Cart List Page– The page that lists the items that customers have chosen. This is the page where the customer can complete the order checkout process.

F. Order Page – The page that lists the customer’s orders.

For Seller:

A. Login Page –Seller enters their website credentials on this page to gain access to log in.

B. Register Page– The page where new seller created their login credentials for the website.

C. Manage Books– The seller has access to the books management information system. They can add, update, and delete the books or prices on their own.

CHAPTER-2: ROLES AND RESPONSIBILITIES DURING INTERNSHIP

2.1 PROBLEMS GIVEN

- E-Book Store is a specific requirement of the client that integrates the buying and selling services specifically to their customers.
- The details regarding all users, and books can also be maintained as their information is very helpful and sometimes becomes a critical requirement.
- Allows the user to get registered from their places and transact for the required product.
- To overcome these problems, we develop “Book-E-Sale”.

2.2 MY ROLES AND RESPONSIBILITIES

- My company role is as an intern for 15 days.
- My responsibilities are as follows

  A. To design frontend, backend APIs, and integration with Database in an eCommerce website project named as Bookstore

  B. To design the backend APIs for login and registration of the user.

  C. To design the backend APIs for CRUD operation on the website’s various functionalities like cart, user, book, and category.

  D. To test APIs using Postman.

  E. To design frontend modules for Login, Register, Book-listing, and View-Cart page.

  F. To provide functionalities for searching and sorting the books.

  G. To provide various functionalities according to the type of user as

    - Buyers can only buy books.
    - Sellers can buy and update the quantity and category of books and can add new books.
    - Admin can create, update, and delete users.

2.3 DAILY TASK AND ACTIVITIES

2.3.1 Weekly Overview of Internship

TABLE 2.1 Week-1 Activities Table

	DATE	DAY	NAME OF THE TOPIC/ MODULE COMPLETED
1st WEEK	20/06/2022	Monday	Introduction of TypeScript
	21/06/2022	Tuesday	Introduction to React with TypeScript
	22/06/2022	Wednesday	Introduction to React Fromik
	23/06/2022	Thursday	Introduction to ASP.NET Core Web API
	24/06/2022	Friday	Introduction to PostgreSQL

TABLE 2.2 Week-2 Activities Table

	DATE	DAY	NAME OF THE TOPIC/ MODULE COMPLETED
2nd WEEK	27/06/2022	Monday	Designing APIs for CRUD operations on User
	28/06/2022	Tuesday	Designing APIs for Login and Register
	29/06/2022	Wednesday	Designing APIs for CRUD operations on Book and Book Category
	30/06/2022	Thursday	Designing APIs for CRUD operations on Cart
	01/07/2022	Friday	Designing APIs for CRUD operations on Publisher

TABLE 2.3 Week-3 Activities Table

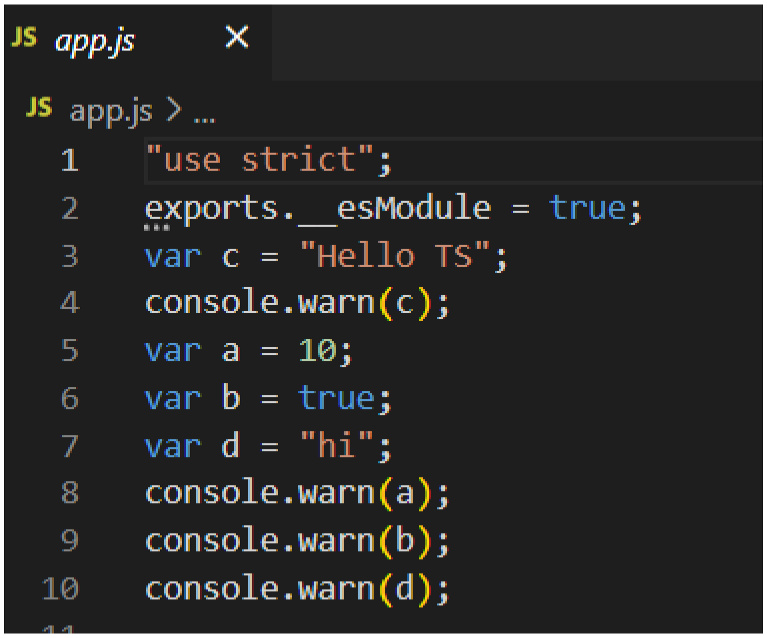
	DATE	**DAY**	**NAME OF THE TOPIC/ MODULE COMPLETED**
3rd WEEK	04/07/2022	Monday	Designing frontend for Register and integrate it with backend
	05/07/2022	Tuesday	Designing frontend for Login and integrate it with backend
	06/07/2022	Wednesday	Designing frontend for Book-Listing and integrate it with backend
	07/07/2022	Thursday	Designing Global Search for Books
	08/07/2022	Friday	Designing frontend for Cart and integrate it with backend

2.3.2 Detailed Overview of Internship

Day-1: Introduction to TypeScript

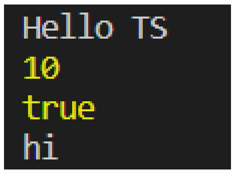
Basics of TypeScript

On Day 1 we have to learn about the basics of typescript, arrays, classes, interface, and objects.



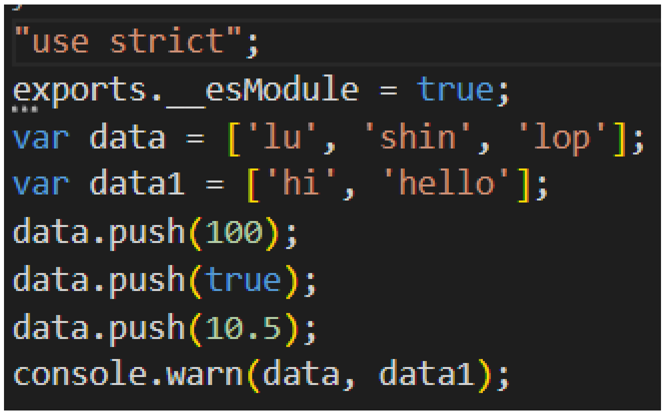
```
"use strict";
exports.__esModule = true;
var c = "Hello TS";
console.warn(c);
var a = 10;
var b = true;
var d = "hi";
console.warn(a);
console.warn(b);
console.warn(d);
```

Fig 2.4 Basic TypeScript Program-1



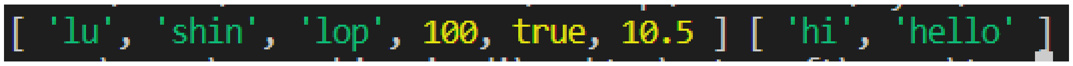
```
Hello TS
10
true
hi
```

Fig 2.5 Basic TypeScript Program Output-1



```
"use strict";
exports.__esModule = true;
var data = ['lu', 'shin', 'lop'];
var data1 = ['hi', 'hello'];
data.push(100);
data.push(true);
data.push(10.5);
console.warn(data, data1);
```

Fig 2.6 Basic TypeScript Program-2



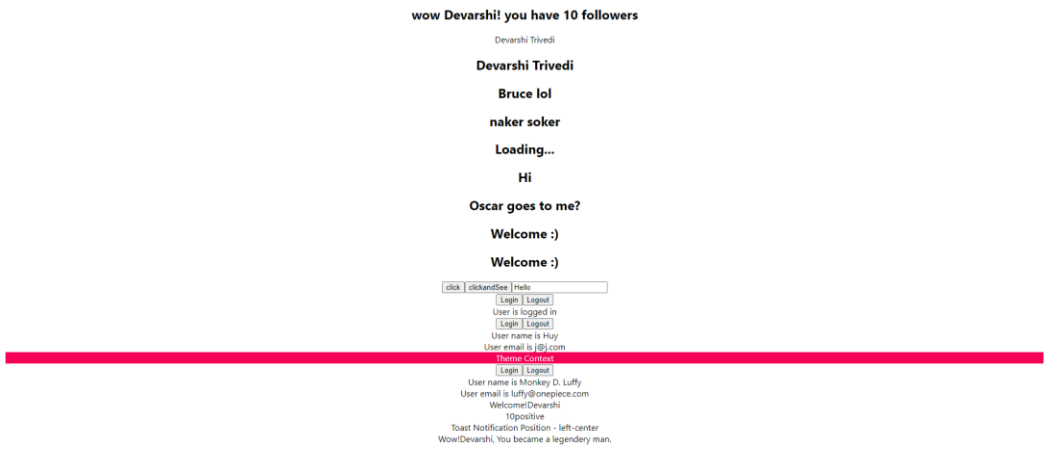
```
[ 'lu', 'shin', 'lop', 100, true, 10.5 ] [ 'hi', 'hello' ]
```

Fig 2.7 Basic TypeScript Program Output-2

Day-2: Introduction to React with TypeScript

On Day 2 we learned about how to use react with the use of typescript. We learn components, States, and Context.

Implement various task such as login using components, login and logout using states.

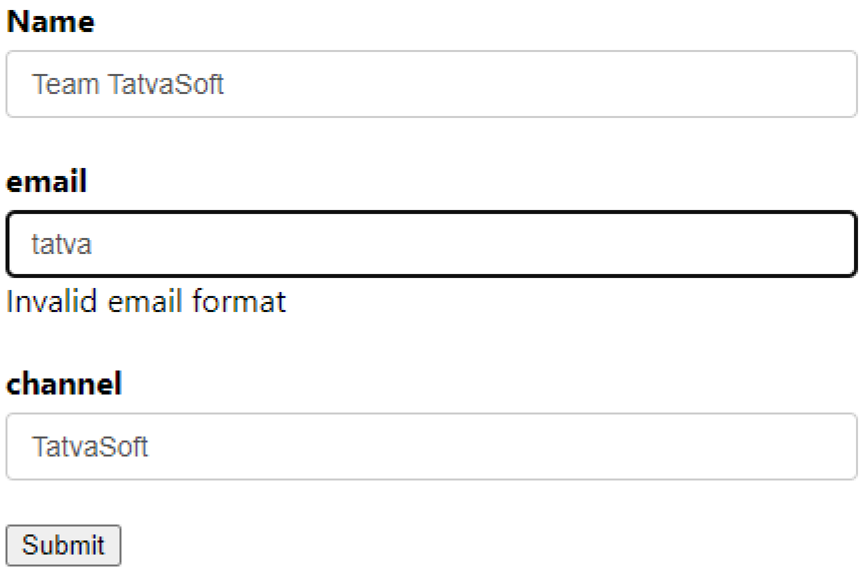


```
wow Devarshi! you have 10 followers
Devarshi Trivedi
Devarshi Trivedi
Bruce lol
naker soker
Loading...
Hi
Oscar goes to me?
Welcome :)
Welcome :)
click clickandSee Hello
Login Logout
User is logged in
Login Logout
User name is Huy
User email is j@j.com
Theme Context
Login Logout
User name is Monkey D. Luffy
User email is luffy@onepiece.com
Welcome!Devarshi
10positive
Toast Notification Position - left-center
Wow!Devarshi, You became a legendery man.
```

Fig 2.8 React with TypeScript Program

Day-3: Introduction to React Formik

We learn about React library for form checking. We learn about formic form's component and also learn about yup library for form checking.



Name

Team TatvaSoft

email

tatva

Invalid email format

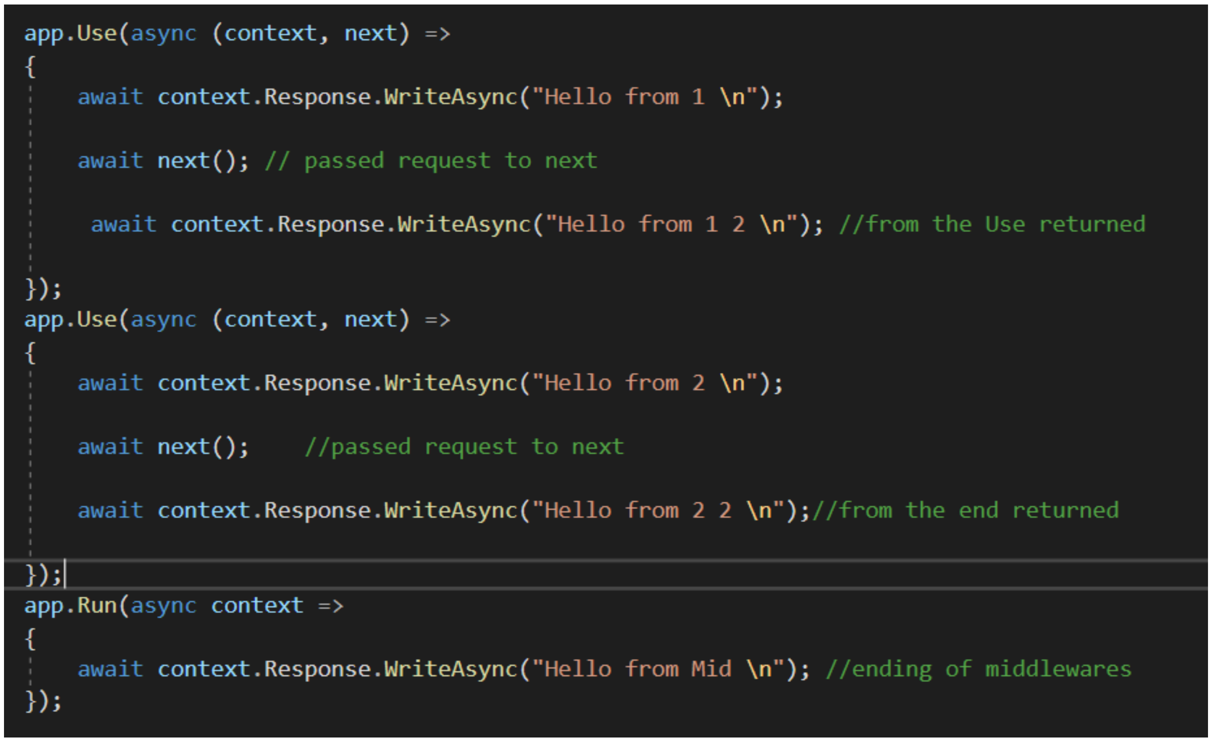
channel

TatvaSoft

Fig 2.9 React Formik Program Output

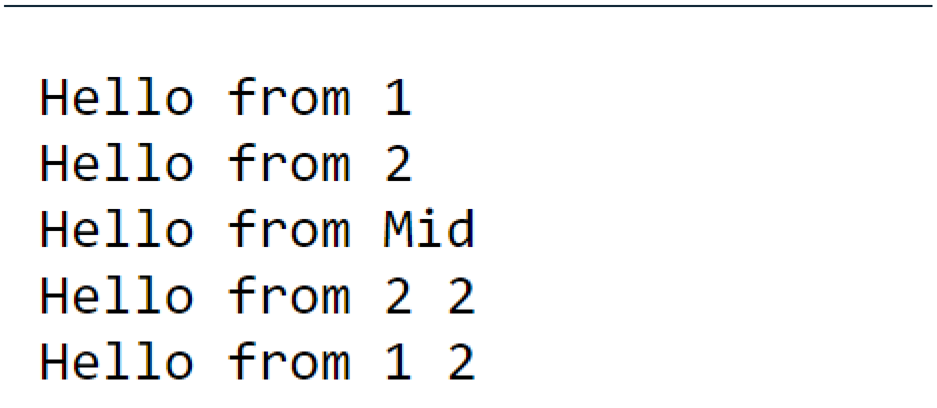
Day-4: Introduction to ASP.NET Core Web API

On day 4 we learn about how to make API, learn the concepts of middleware and build basics API.



```
app.Use(async (context, next) =>
{
    await context.Response.WriteAsync("Hello from 1 \n");

    await next(); // passed request to next

     await context.Response.WriteAsync("Hello from 1 2 \n"); //from the Use returned

});
app.Use(async (context, next) =>
{
    await context.Response.WriteAsync("Hello from 2 \n");

    await next();    //passed request to next

    await context.Response.WriteAsync("Hello from 2 2 \n");//from the end returned

});
app.Run(async context =>
{
    await context.Response.WriteAsync("Hello from Mid \n"); //ending of middlewares
});
```

Fig 2.10 ASP.NET Middleware code

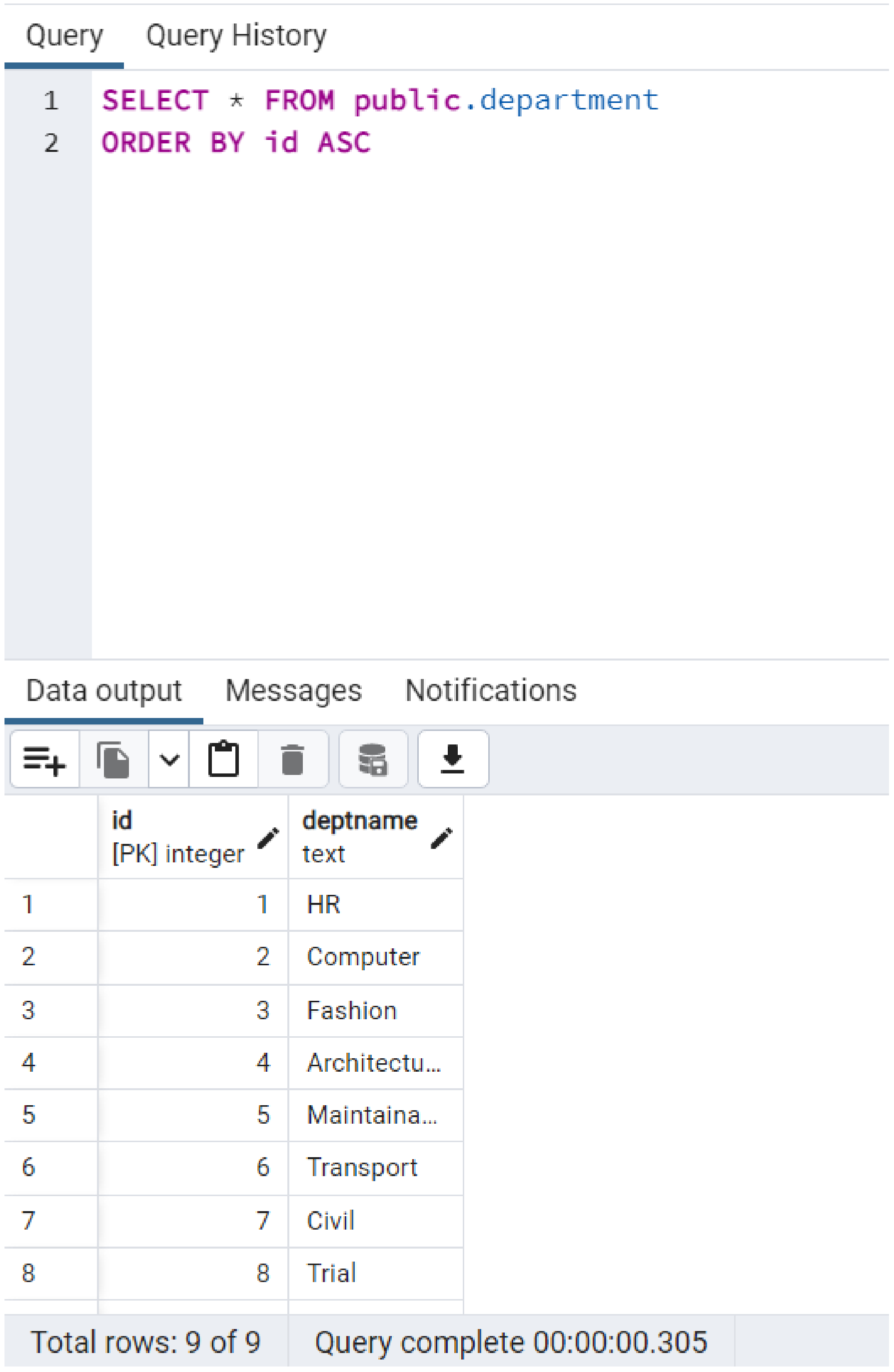


```
Hello from 1
Hello from 2
Hello from Mid
Hello from 2 2
Hello from 1 2
```

Fig 2.11 ASP.NET Middleware Code Output

Day-5: Introduction to PostgreSQL

We learn about creating a Database and writing queries using pgAdmin.



```
SELECT * FROM public.department
ORDER BY id ASC
```

	id [PK] integer	deptname text
1	1	HR
2	2	Computer
3	3	Fashion
4	4	Architectu...
5	5	Maintaina...
6	6	Transport
7	7	Civil
8	8	Trial

Total rows: 9 of 9   Query complete 00:00:00.305

Fig 2.12 PostgreSQL with pgAdmin

Day 6: Designing APIs for CRUD operations on User

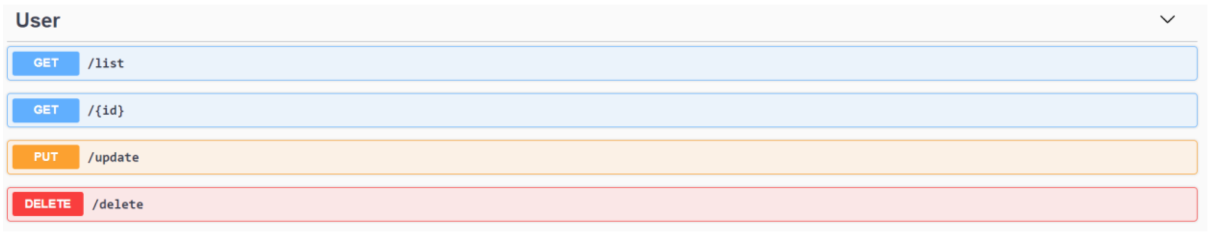
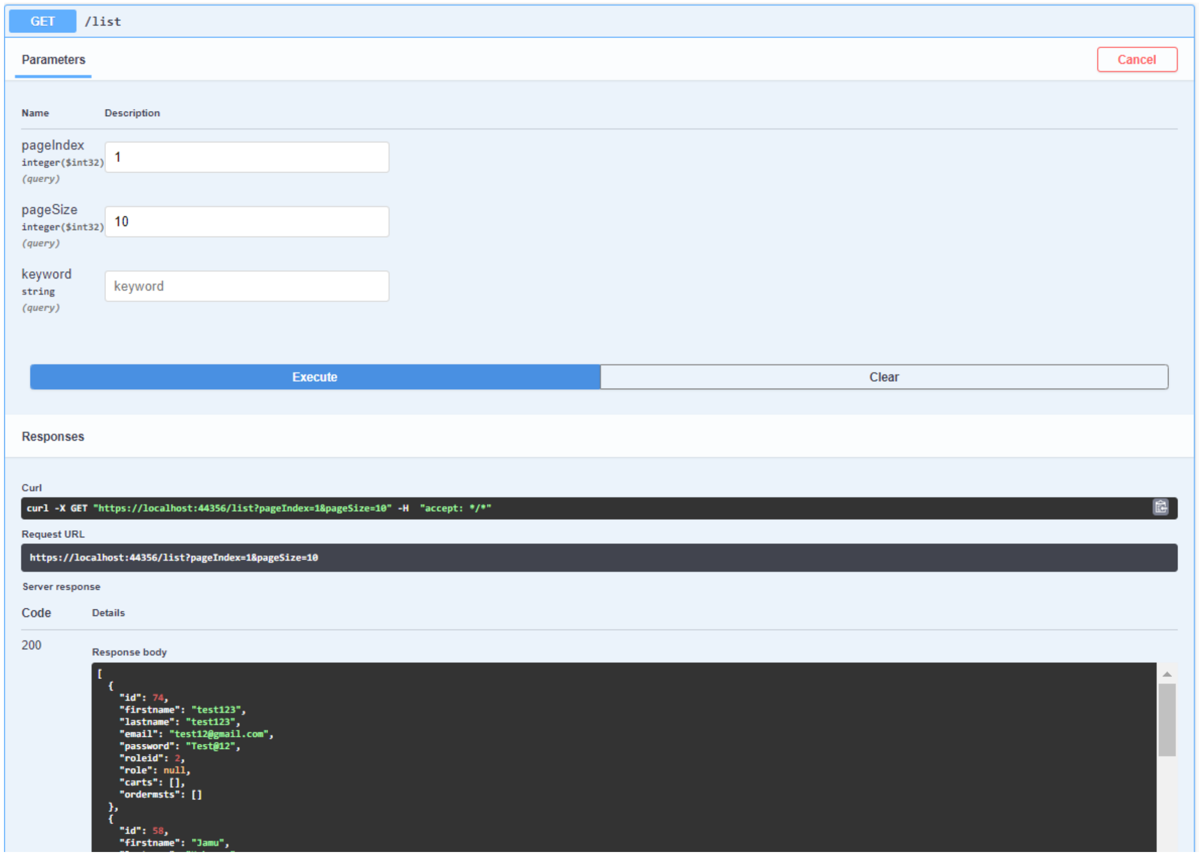


Fig 2.13 CRUD APIS for User

APIs working: -

List API fetches all the details of registered users.



pageIndex integer($int32) (query): 1

pageSize integer($int32) (query): 10

keyword string (query): keyword

Curl

```
curl -X GET "https://localhost:44356/list?pageIndex=1&pageSize=10" -H  "accept: */*"
```

Request URL

```
https://localhost:44356/list?pageIndex=1&pageSize=10
```

Server response

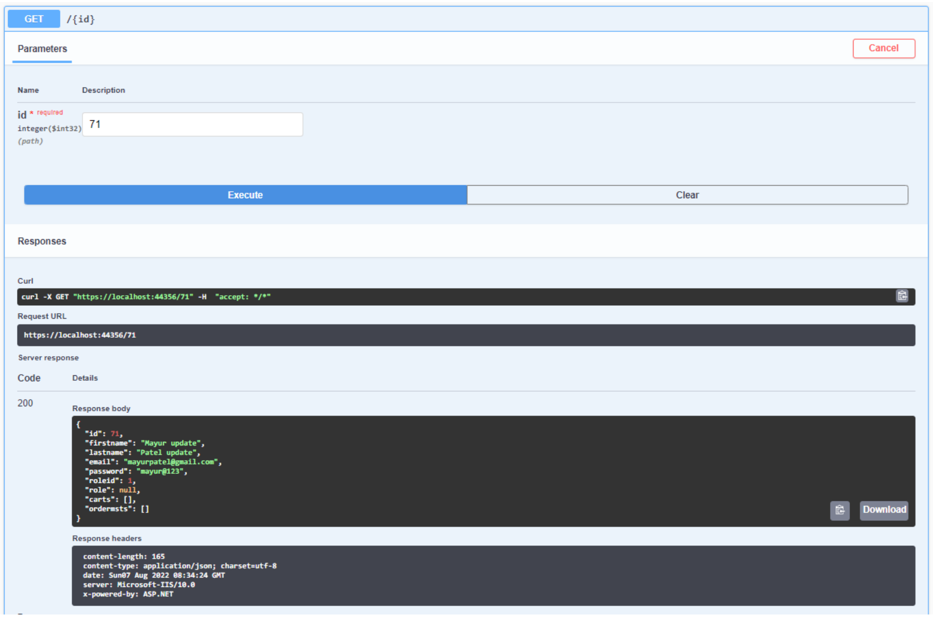
Code: 200

Details: Response body

```
[
  {
    "id": 74,
    "firstname": "test123",
    "lastname": "test123",
    "email": "test12@gmail.com",
    "password": "Test@12",
    "roleid": 2,
    "role": null,
    "carts": [],
    "ordermsts": []
  },
  {
    "id": 58,
    "firstname": "Jamu",
```

Fig 2.14 Read Operation for User

This API fetches particular user’s detail.



`/{id}`

Name	Description
id * required integer($int32) (path)	71

Responses

Curl

```
curl -X GET "https://localhost:44356/71" -H  "accept: */*"
```

Request URL

```
https://localhost:44356/71
```

Server response

Code	Details
200	Response body

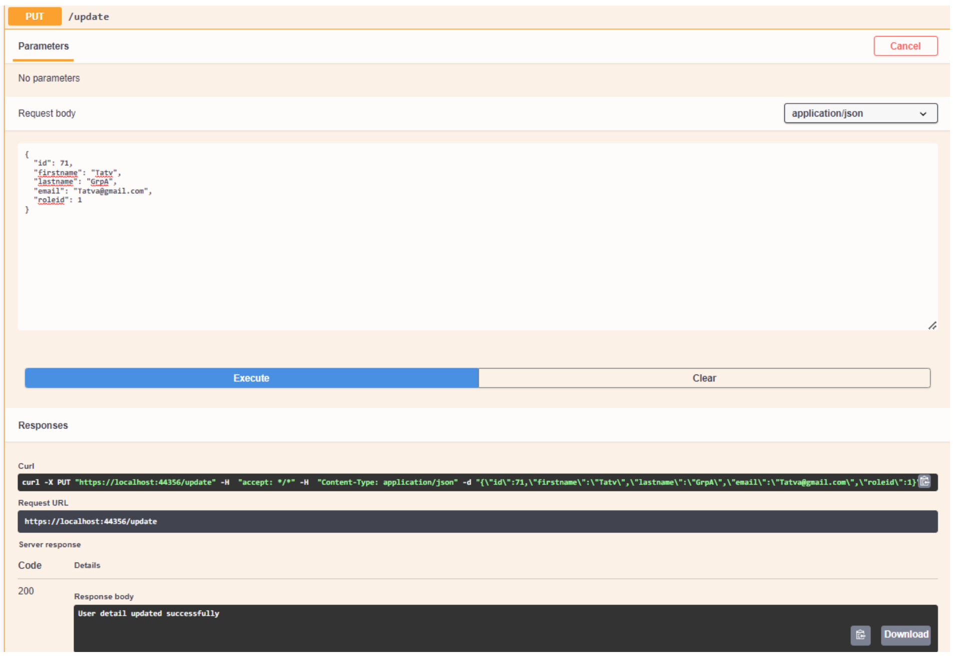
```
{
  "id": 71,
  "firstname": "Mayur update",
  "lastname": "Patel update",
  "email": "mayurpatel@gmail.com",
  "password": "mayur@123",
  "roleid": 1,
  "role": null,
  "carts": [],
  "ordermsts": []
}
```

Response headers

```
content-length: 165
content-type: application/json; charset=utf-8
date: Sun07 Aug 2022 08:34:24 GMT
server: Microsoft-IIS/10.0
x-powered-by: ASP.NET
```

Fig 2.15 Read with id fetch particular user’s details

This API updates existing user which can be used in update profile module in frontend.



`/update`

No parameters

Request body: application/json

```
{
  "id": 71,
  "firstname": "Tatv",
  "lastname": "GrpA",
  "email": "Tatva@gmail.com",
  "roleid": 1
}
```

Responses

Curl

```
curl -X PUT "https://localhost:44356/update" -H  "accept: */*" -H  "Content-Type: application/json" -d "{\"id\":71,\"firstname\":\"Tatv\",\"lastname\":\"GrpA\",\"email\":\"Tatva@gmail.com\",\"roleid\":1}
```

Request URL

```
https://localhost:44356/update
```

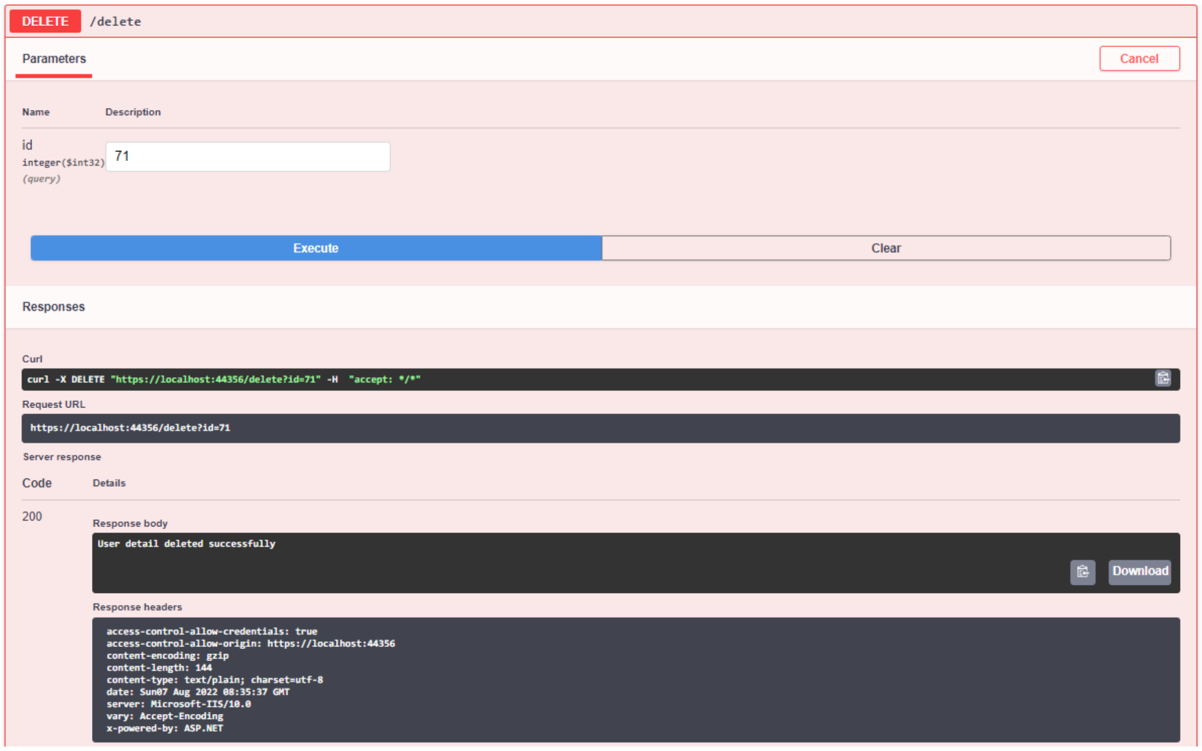
Server response

Code	Details
200	Response body

```
User detail updated successfully
```

Fig 2.16 Update Operation for User

This API delete existing user which can be used in delete account module in frontend.



Curl

```
curl -X DELETE "https://localhost:44356/delete?id=71" -H  "accept: */*"
```

Request URL

```
https://localhost:44356/delete?id=71
```

Server response

Code Details

200

Response body

```
User detail deleted successfully
```

Response headers

```
access-control-allow-credentials: true
access-control-allow-origin: https://localhost:44356
content-encoding: gzip
content-length: 144
content-type: text/plain; charset=utf-8
date: Sun07 Aug 2022 08:35:37 GMT
server: Microsoft-IIS/10.0
vary: Accept-Encoding
x-powered-by: ASP.NET
```

Fig 2.17 Delete Operation for User

Day-7: Designing APIs for Login and Register

If the user register API enter the data of the user into the database, and while the user login API has to check whether the credential is valid or not and based upon that it shows success or failure.

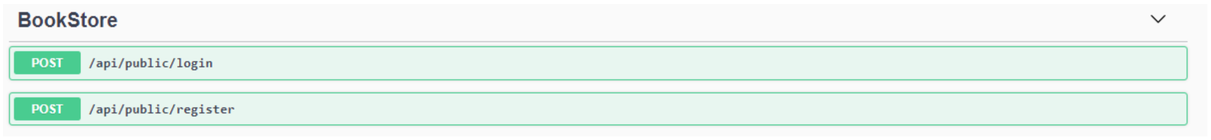
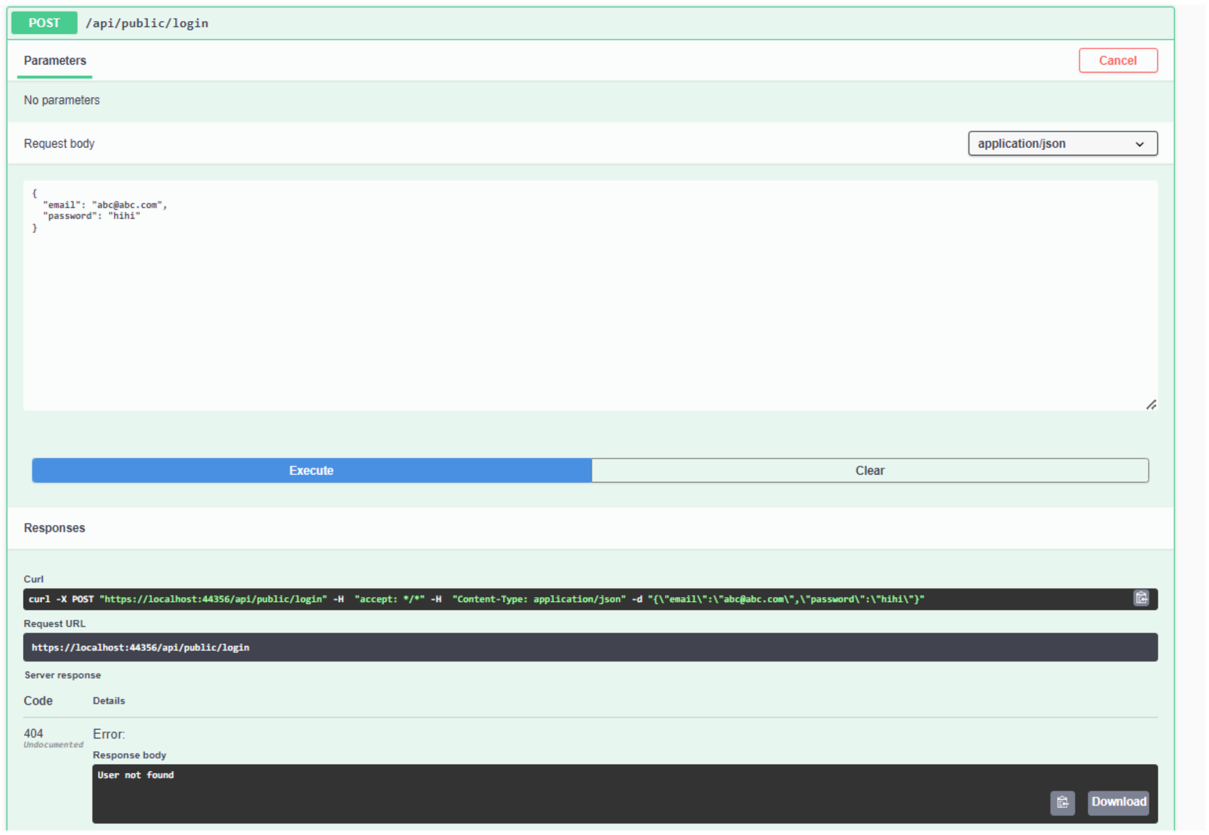


Fig 2.18 Login and Register APIs

APIs working: -



```
{
  "email": "abc@abc.com",
  "password": "hihi"
}
```

Curl

```
curl -X POST "https://localhost:44356/api/public/login" -H  "accept: */*" -H  "Content-Type: application/json" -d "{\"email\":\"abc@abc.com\",\"password\":\"hihi\"}"
```

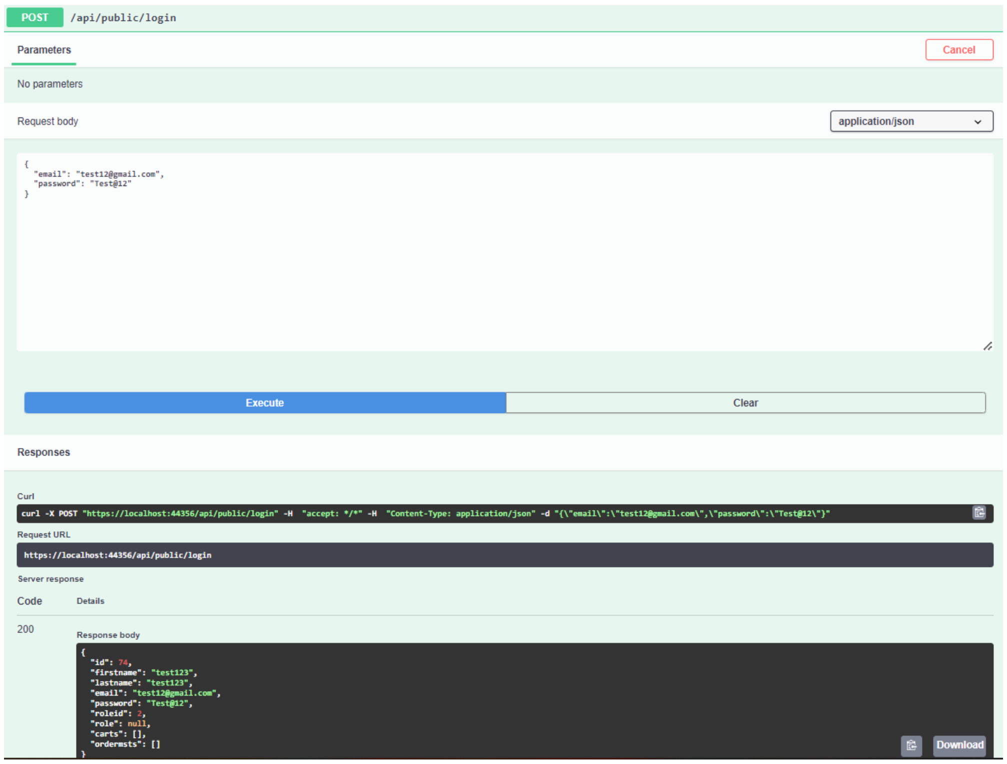
Request URL

```
https://localhost:44356/api/public/login
```

Server response

Code	Details
404 Undocumented	Error: Response body `User not found`

Fig 2.19 Unsuccessfully Login



```
{
  "email": "test12@gmail.com",
  "password": "Test@12"
}
```

Curl

```
curl -X POST "https://localhost:44356/api/public/login" -H  "accept: */*" -H  "Content-Type: application/json" -d "{\"email\":\"test12@gmail.com\",\"password\":\"Test@12\"}"
```

Request URL

```
https://localhost:44356/api/public/login
```

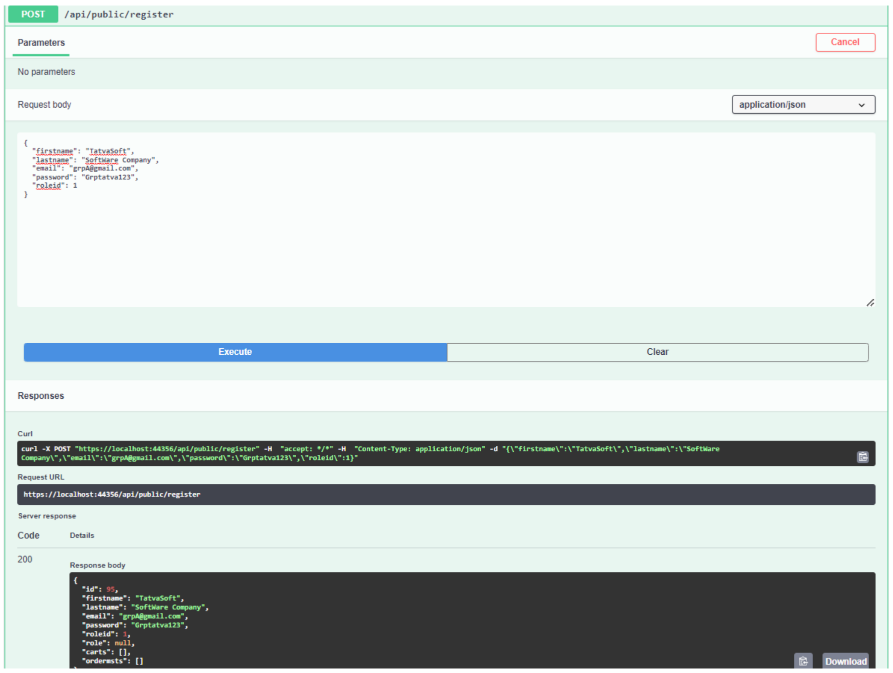
Server response

Code Details

200 Response body

```
{
  "id": 74,
  "firstname": "test123",
  "lastname": "test123",
  "email": "test12@gmail.com",
  "password": "Test@12",
  "roleid": 2,
  "role": null,
  "carts": [],
  "ordermsts": []
}
```

Fig 2.20 Successful Login



```
{
  "firstname": "TatvaSoft",
  "lastname": "SoftWare Company",
  "email": "grpA@gmail.com",
  "password": "Grptatva123",
  "roleid": 1
}
```

Curl

```
curl -X POST "https://localhost:44356/api/public/register" -H  "accept: */*" -H  "Content-Type: application/json" -d "{\"firstname\":\"TatvaSoft\",\"lastname\":\"SoftWare
Company\",\"email\":\"grpA@gmail.com\",\"password\":\"Grptatva123\",\"roleid\":1}"
```

Request URL

```
https://localhost:44356/api/public/register
```

Server response

Code Details

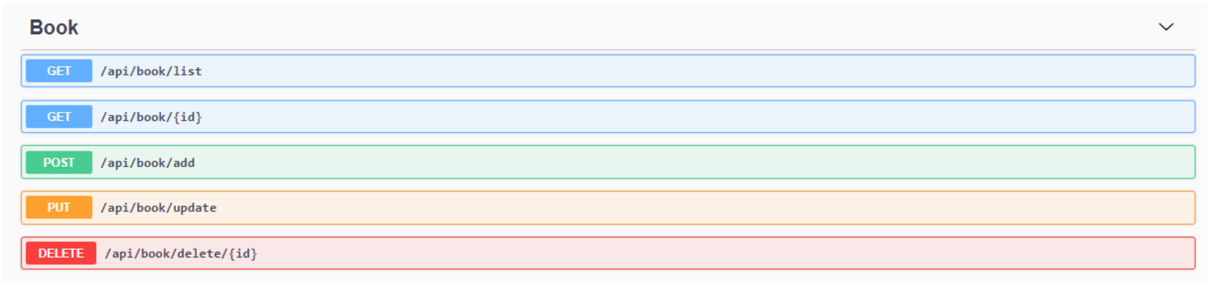
200 Response body

```
{
  "id": 95,
  "firstname": "TatvaSoft",
  "lastname": "SoftWare Company",
  "email": "grpA@gmail.com",
  "password": "Grptatva123",
  "roleid": 1,
  "role": null,
  "carts": [],
  "ordermsts": []
```

Fig 2.21 Registration of User

Day 8: Designing APIs for CRUD operations on Book Category

The main task is to design 5 APIs for CRUD operations on Book and Category table.

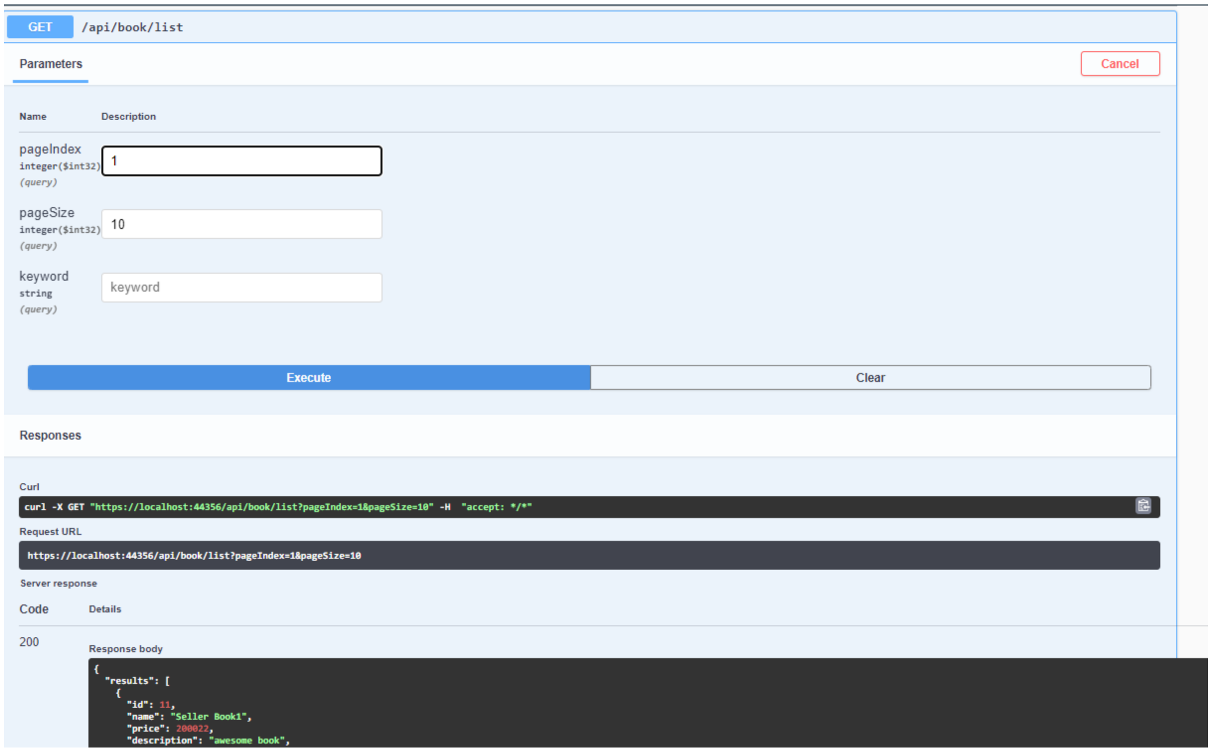


```
/api/book/list
/api/book/{id}
/api/book/add
/api/book/update
/api/book/delete/{id}
```

Fig 2.22 CRUD Operations for Book table

APIs Working: -

API fetch book according to the keyword. It is the backend for the global search module.



/api/book/list

Name	Description
pageIndex integer($int32) (query)	1
pageSize integer($int32) (query)	10
keyword string (query)	keyword

Curl

```
curl -X GET "https://localhost:44356/api/book/list?pageIndex=1&pageSize=10" -H  "accept: */*"
```

Request URL

```
https://localhost:44356/api/book/list?pageIndex=1&pageSize=10
```

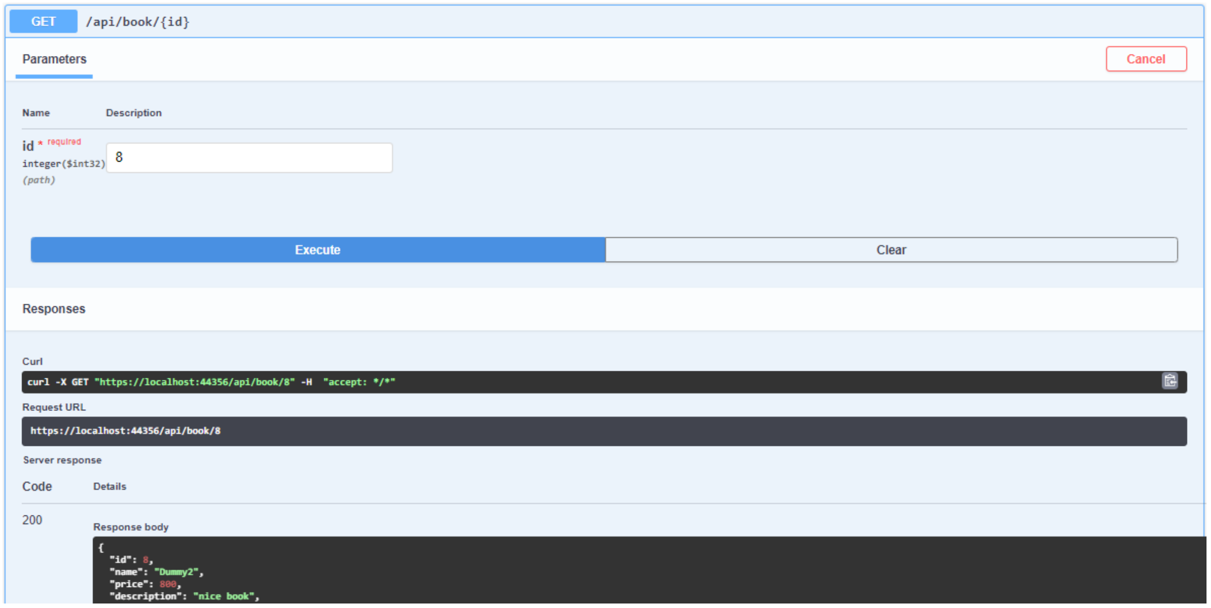
Server response

Code	Details
200	Response body

```
{
  "results": [
    {
      "id": 11,
      "name": "Seller Book1",
      "price": 200022,
      "description": "awesome book",
```

Fig 2.23 Global Search Backend Implementation

API fetch particular book detail according to the id of the book.



`/api/book/{id}`

Name	Description
id * required integer($int32) (path)	8

Curl

```
curl -X GET "https://localhost:44356/api/book/8" -H  "accept: */*"
```

Request URL

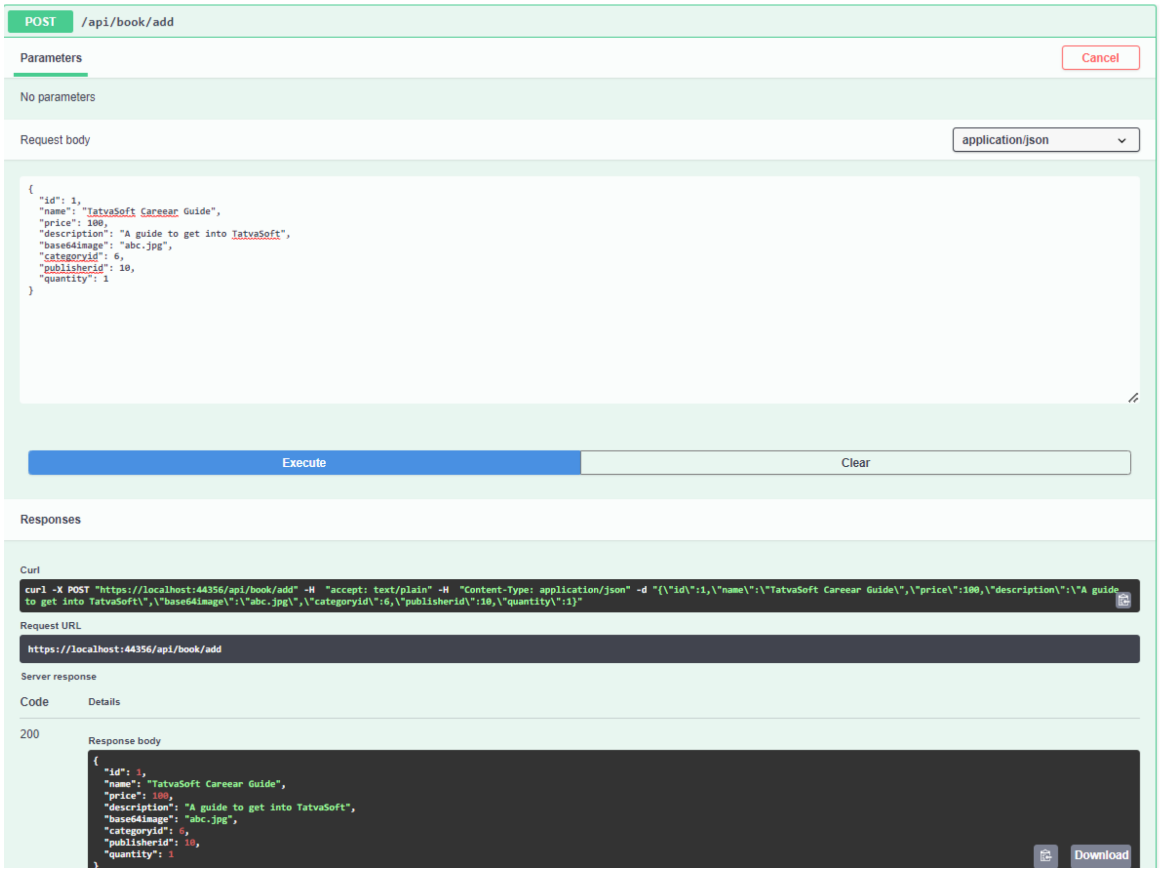
```
https://localhost:44356/api/book/8
```

Server response

Code	Details
200	Response body

```
{
  "id": 8,
  "name": "Dummy2",
  "price": 800,
  "description": "nice book",
```

Fig 2.24 Read Operation for Book



`/api/book/add`

No parameters

Request body application/json

```
{
  "id": 1,
  "name": "TatvaSoft Careear Guide",
  "price": 100,
  "description": "A guide to get into TatvaSoft",
  "base64image": "abc.jpg",
  "categoryid": 6,
  "publisherid": 10,
  "quantity": 1
}
```

Curl

```
curl -X POST "https://localhost:44356/api/book/add" -H  "accept: text/plain" -H  "Content-Type: application/json" -d "{\"id\":1,\"name\":\"TatvaSoft Careear Guide\",\"price\":100,\"description\":\"A guide to get into TatvaSoft\",\"base64image\":\"abc.jpg\",\"categoryid\":6,\"publisherid\":10,\"quantity\":1}"
```

Request URL

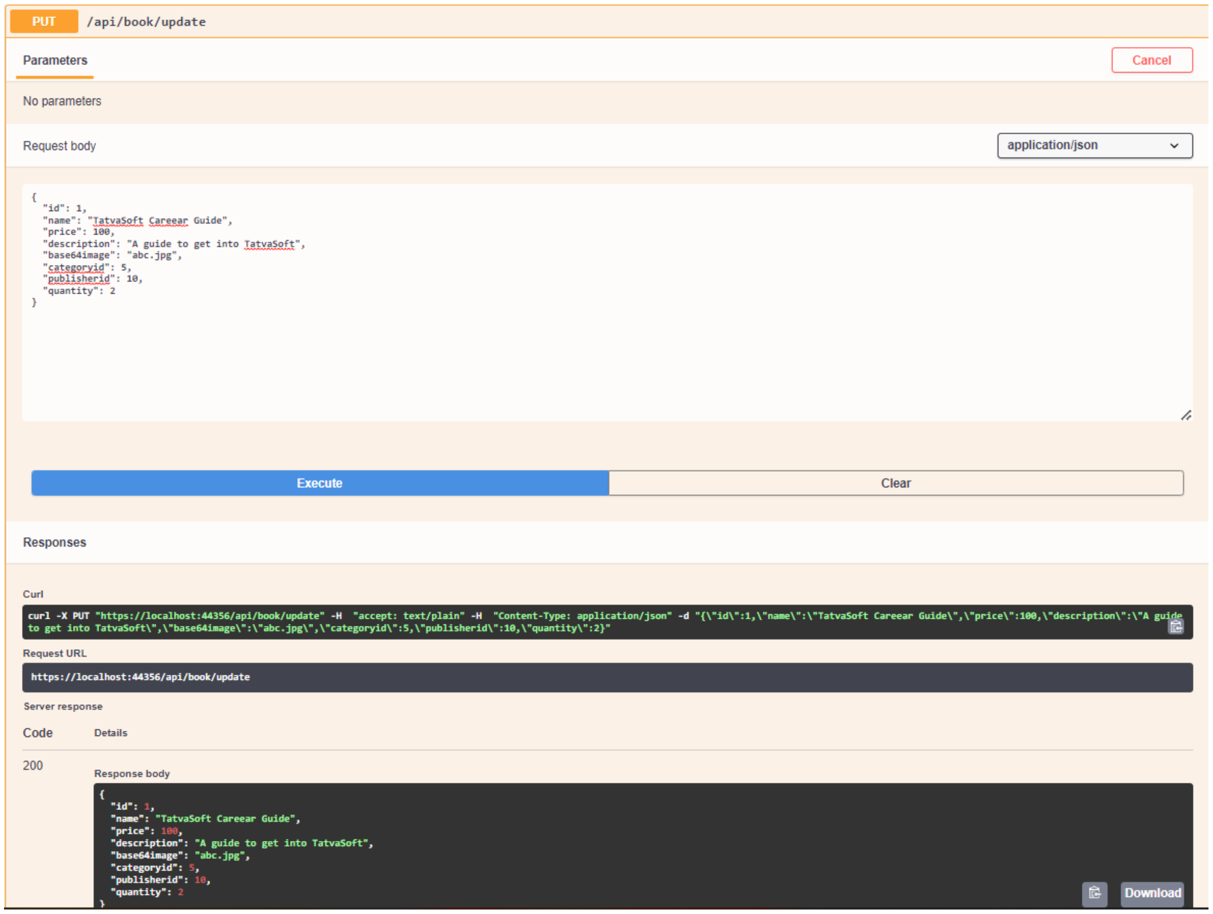
```
https://localhost:44356/api/book/add
```

Server response

Code	Details
200	Response body

```
{
  "id": 1,
  "name": "TatvaSoft Careear Guide",
  "price": 100,
  "description": "A guide to get into TatvaSoft",
  "base64image": "abc.jpg",
  "categoryid": 6,
  "publisherid": 10,
  "quantity": 1
}
```

Fig 2.25 Add Operation for Book



No parameters

Request body

```
{
  "id": 1,
  "name": "TatvaSoft Careear Guide",
  "price": 100,
  "description": "A guide to get into TatvaSoft",
  "base64image": "abc.jpg",
  "categoryid": 5,
  "publisherid": 10,
  "quantity": 2
}
```

Responses

Curl

```
curl -X PUT "https://localhost:44356/api/book/update" -H  "accept: text/plain" -H  "Content-Type: application/json" -d "{\"id\":1,\"name\":\"TatvaSoft Careear Guide\",\"price\":100,\"description\":\"A guide to get into TatvaSoft\",\"base64image\":\"abc.jpg\",\"categoryid\":5,\"publisherid\":10,\"quantity\":2}"
```

Request URL

```
https://localhost:44356/api/book/update
```

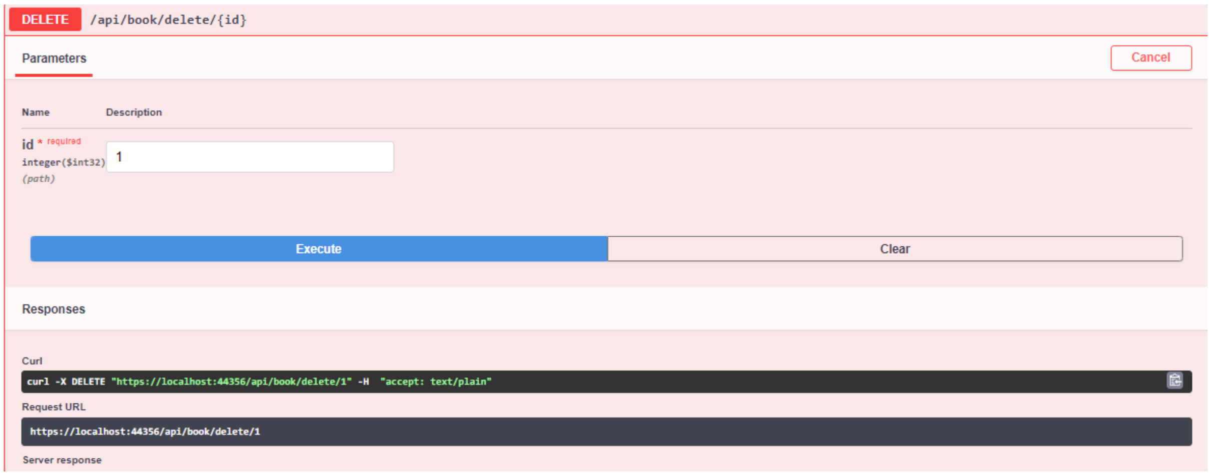
Server response

Code Details

200 Response body

```
{
  "id": 1,
  "name": "TatvaSoft Careear Guide",
  "price": 100,
  "description": "A guide to get into TatvaSoft",
  "base64image": "abc.jpg",
  "categoryid": 5,
  "publisherid": 10,
  "quantity": 2
}
```

Fig 2.26 Update Operation for Book



Name	Description
id * required integer($int32) (path)	1

Responses

Curl

```
curl -X DELETE "https://localhost:44356/api/book/delete/1" -H  "accept: text/plain"
```

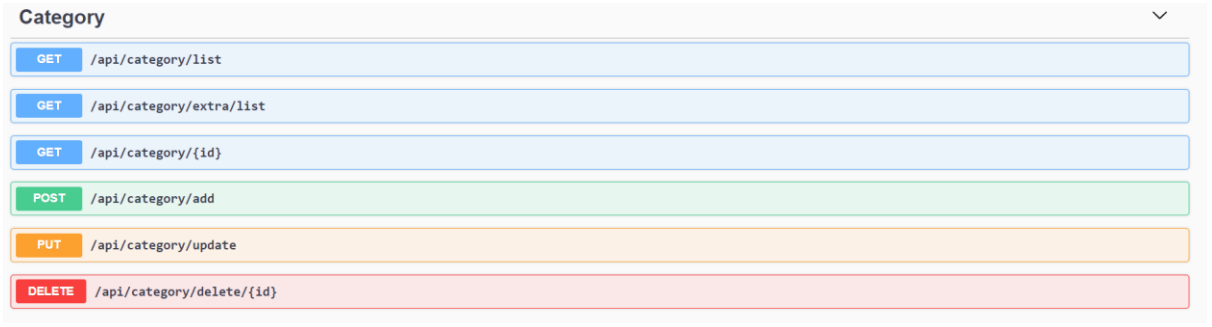
Request URL

```
https://localhost:44356/api/book/delete/1
```

Server response

Fig 2.27 Delete Operation for Book

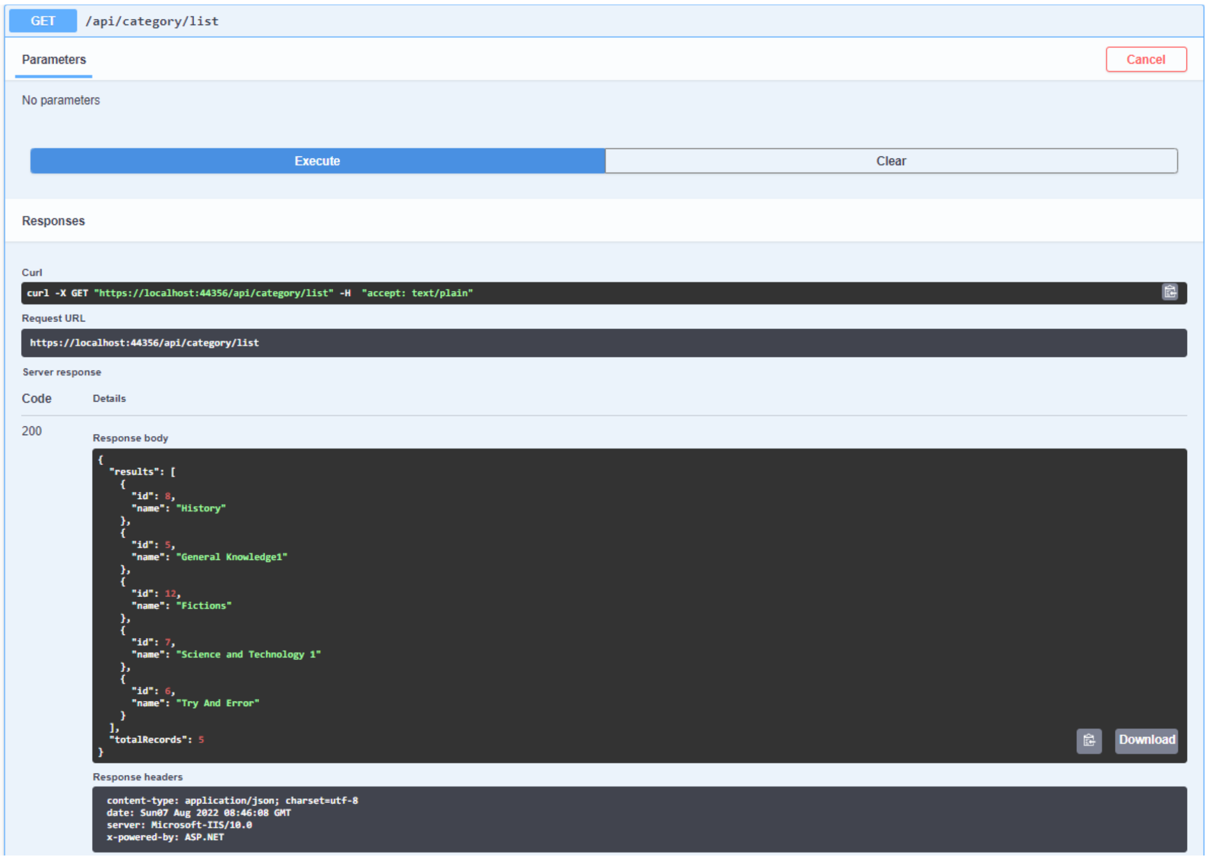
CRUD Operations for Category are shown below: -


Category

```
GET     /api/category/list
GET     /api/category/extra/list
GET     /api/category/{id}
POST    /api/category/add
PUT     /api/category/update
DELETE  /api/category/delete/{id}
```

Fig 2.28 CRUD Operations on Category

APIs Working: -


```
GET /api/category/list
```

Curl

```
curl -X GET "https://localhost:44356/api/category/list" -H  "accept: text/plain"
```

Request URL

```
https://localhost:44356/api/category/list
```

Server response

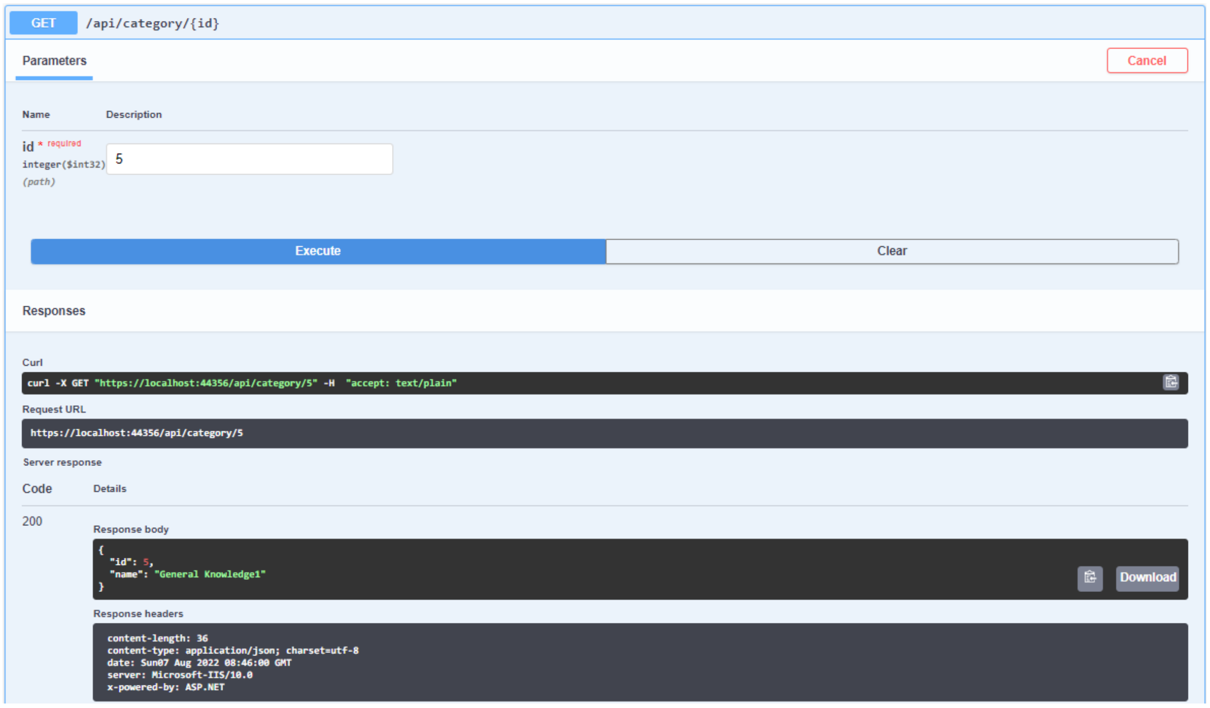
Code	Details
200	Response body

```
{
  "results": [
    {
      "id": 8,
      "name": "History"
    },
    {
      "id": 5,
      "name": "General Knowledge1"
    },
    {
      "id": 12,
      "name": "Fictions"
    },
    {
      "id": 7,
      "name": "Science and Technology 1"
    },
    {
      "id": 6,
      "name": "Try And Error"
    }
  ],
  "totalRecords": 5
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Sun07 Aug 2022 08:46:08 GMT
server: Microsoft-IIS/10.0
x-powered-by: ASP.NET
```

Fig 2.29 Read Operation on Category



GET /api/category/{id}

Parameters

Name: id * required, integer($int32) (path)

Value: 5

Responses

Curl

```
curl -X GET "https://localhost:44356/api/category/5" -H  "accept: text/plain"
```

Request URL

```
https://localhost:44356/api/category/5
```

Server response

Code: 200

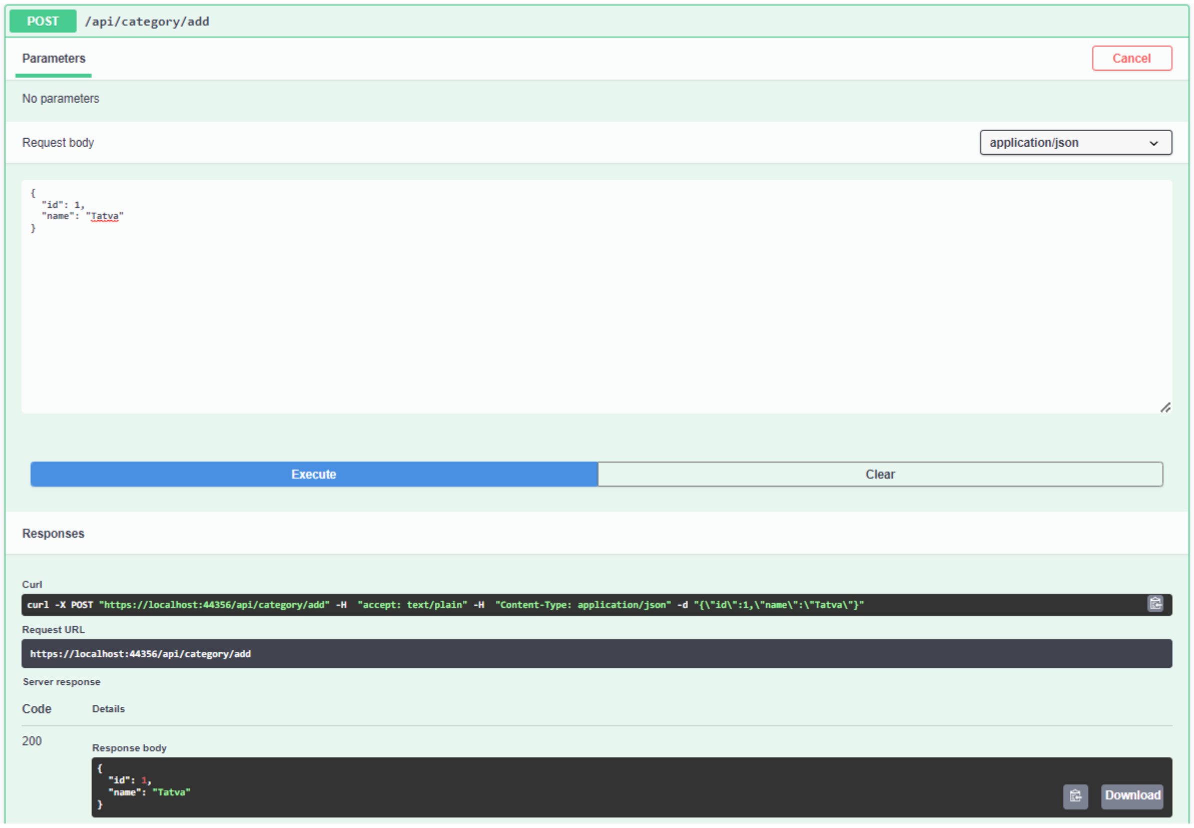
Response body

```
{
  "id": 5,
  "name": "General Knowledge1"
}
```

Response headers

```
content-length: 36
content-type: application/json; charset=utf-8
date: Sun07 Aug 2022 08:46:00 GMT
server: Microsoft-IIS/10.0
x-powered-by: ASP.NET
```

Fig 2.30 Fetching particular category



POST /api/category/add

Parameters

No parameters

Request body: application/json

```
{
  "id": 1,
  "name": "Tatva"
}
```

Responses

Curl

```
curl -X POST "https://localhost:44356/api/category/add" -H  "accept: text/plain" -H  "Content-Type: application/json" -d "{\"id\":1,\"name\":\"Tatva\"}"
```

Request URL

```
https://localhost:44356/api/category/add
```

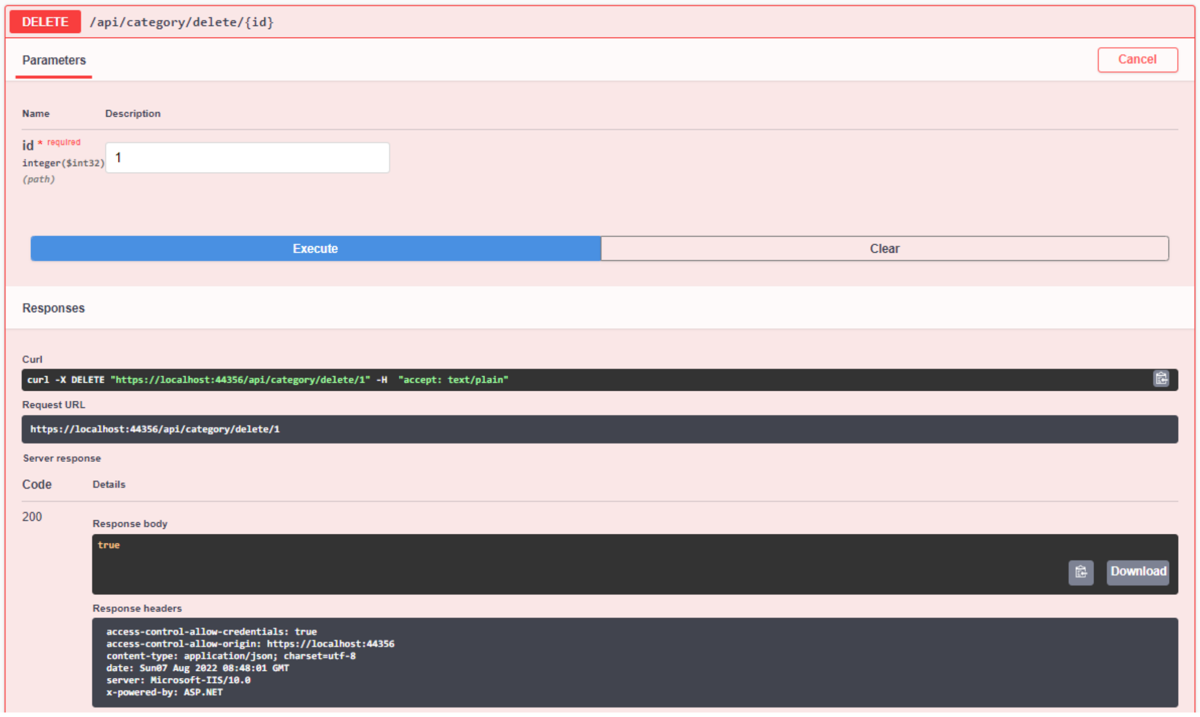
Server response

Code: 200

Response body

```
{
  "id": 1,
  "name": "Tatva"
}
```

Fig 2.31 Add Operation on Category



```
curl -X DELETE "https://localhost:44356/api/category/delete/1" -H  "accept: text/plain"
```

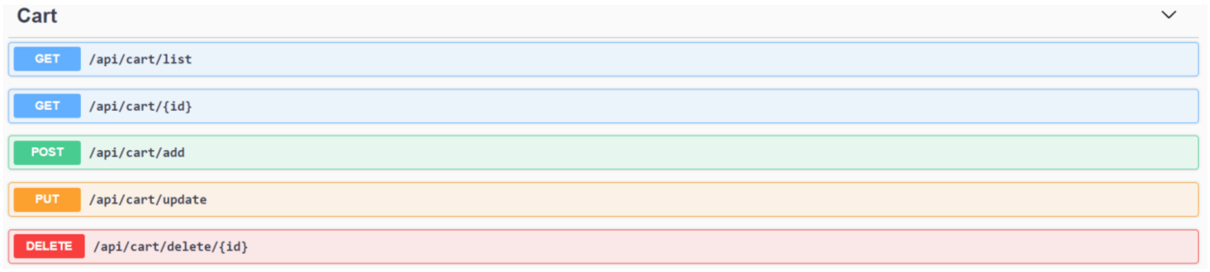
```
https://localhost:44356/api/category/delete/1
```

```
true
```

```
access-control-allow-credentials: true
access-control-allow-origin: https://localhost:44356
content-type: application/json; charset=utf-8
date: Sun07 Aug 2022 08:48:01 GMT
server: Microsoft-IIS/10.0
x-powered-by: ASP.NET
```

Fig 2.32 Delete Operation on Category

Day 9: Designing APIs for CRUD operations on Cart

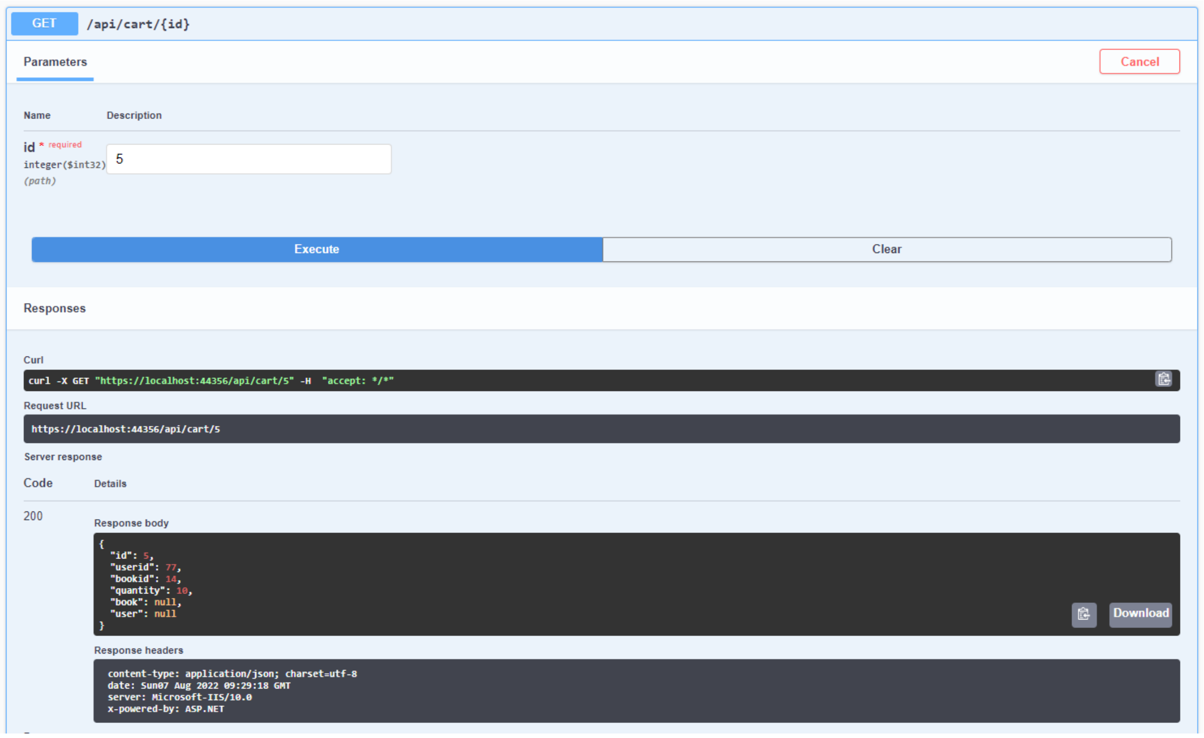


Cart

```
GET     /api/cart/list
GET     /api/cart/{id}
POST    /api/cart/add
PUT     /api/cart/update
DELETE  /api/cart/delete/{id}
```

Fig 2.33 CRUD Operations on Cart

APIs Working: -



GET /api/cart/{id}

id * required
integer($int32)
(path)
5

Curl

```
curl -X GET "https://localhost:44356/api/cart/5" -H  "accept: */*"
```

Request URL

```
https://localhost:44356/api/cart/5
```

Server response

Code Details

200

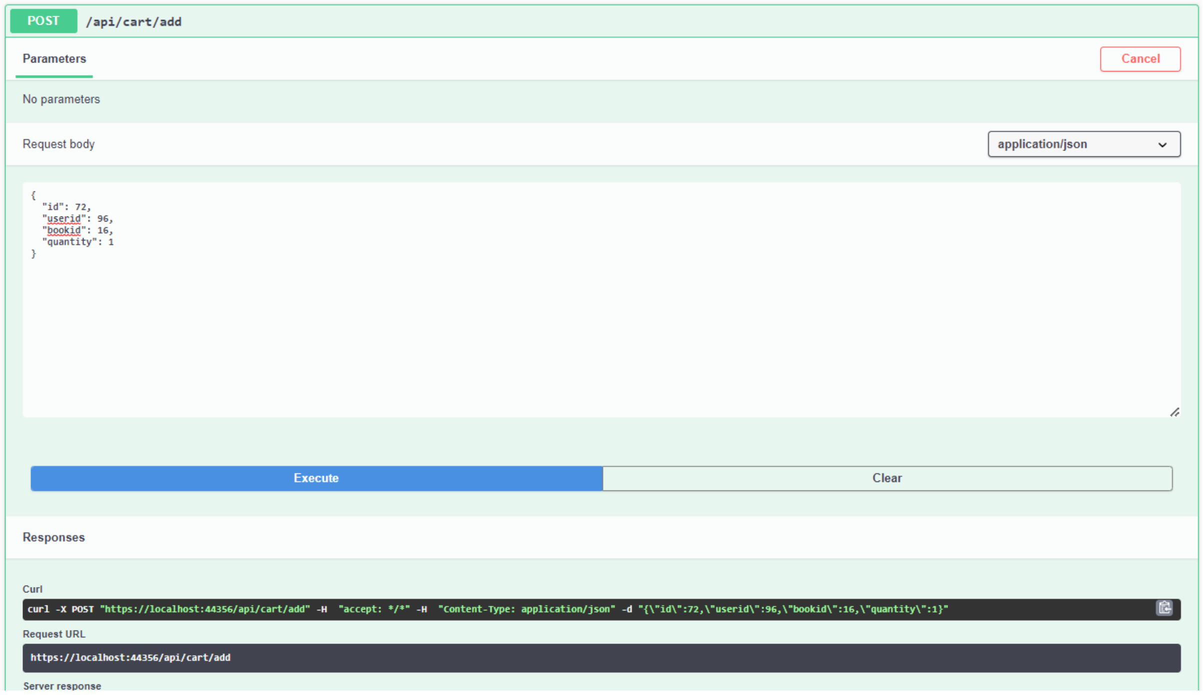
Response body

```
{
  "id": 5,
  "userid": 77,
  "bookid": 14,
  "quantity": 10,
  "book": null,
  "user": null
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Sun07 Aug 2022 09:29:18 GMT
server: Microsoft-IIS/10.0
x-powered-by: ASP.NET
```

Fig 2.34 Read Operation on Cart



No parameters

Request body: application/json

```
{
  "id": 72,
  "userid": 96,
  "bookid": 16,
  "quantity": 1
}
```

Responses

Curl

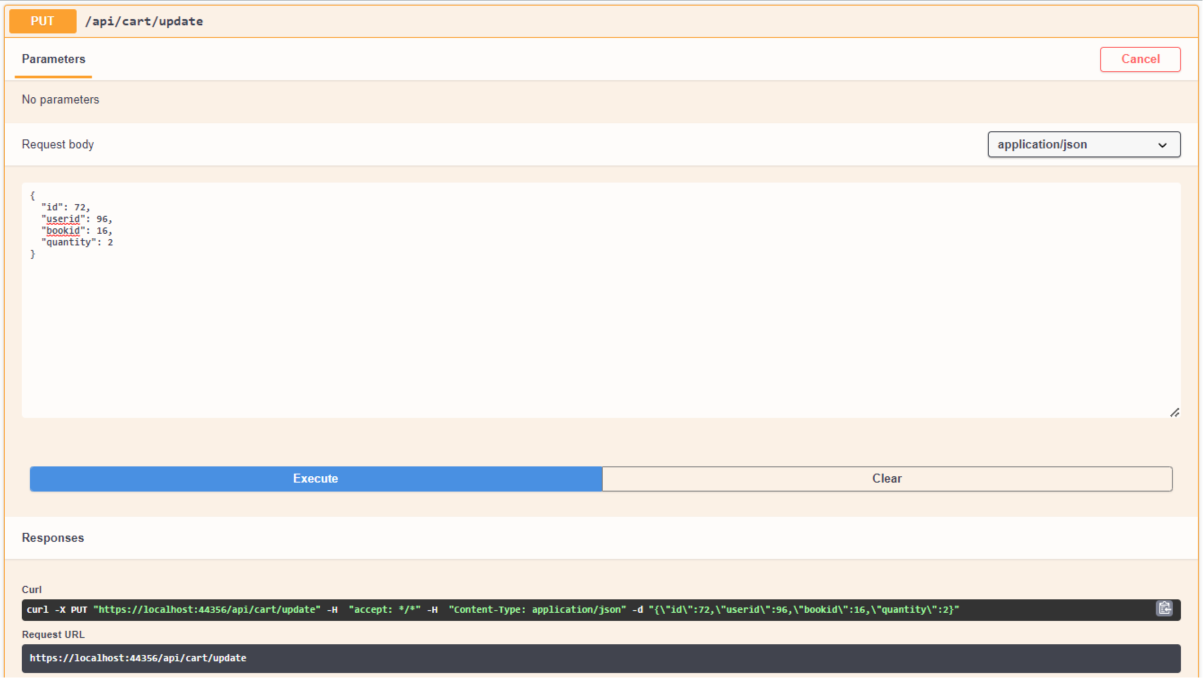
```
curl -X POST "https://localhost:44356/api/cart/add" -H  "accept: */*" -H  "Content-Type: application/json" -d "{\"id\":72,\"userid\":96,\"bookid\":16,\"quantity\":1}"
```

Request URL

```
https://localhost:44356/api/cart/add
```

Server response

Fig 2.35 Add Operation on Cart



No parameters

Request body: application/json

```
{
  "id": 72,
  "userid": 96,
  "bookid": 16,
  "quantity": 2
}
```

Responses

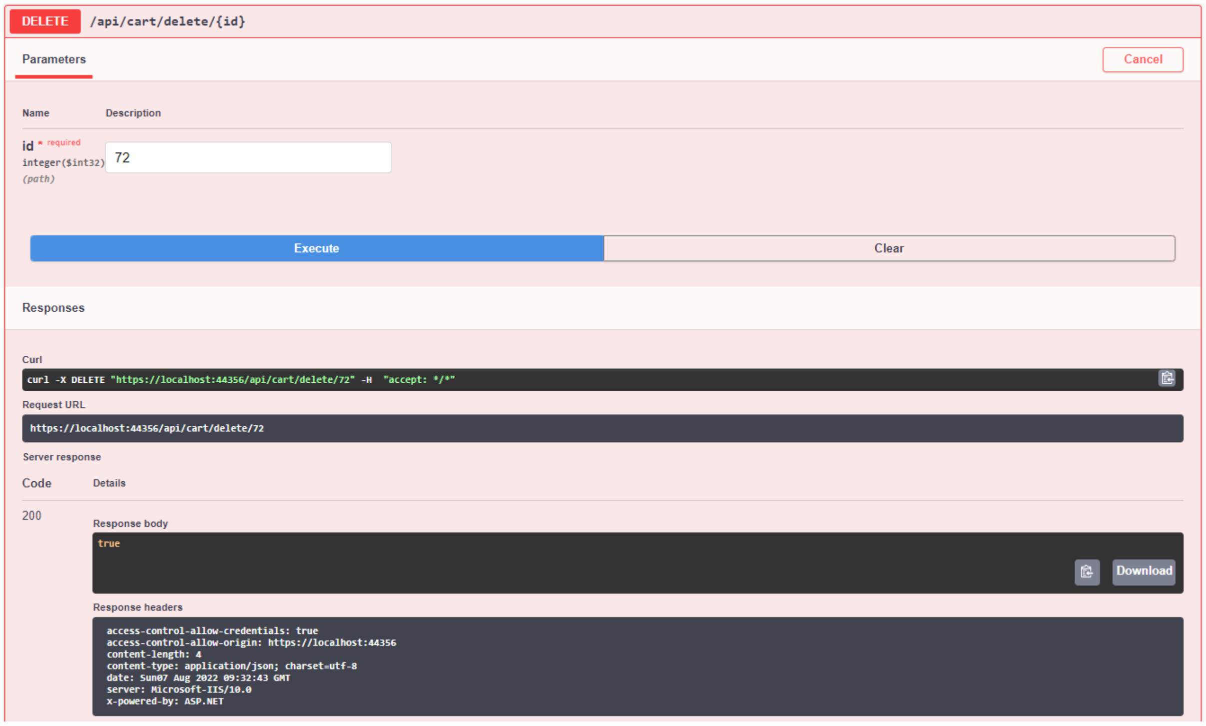
Curl

```
curl -X PUT "https://localhost:44356/api/cart/update" -H  "accept: */*" -H  "Content-Type: application/json" -d "{\"id\":72,\"userid\":96,\"bookid\":16,\"quantity\":2}"
```

Request URL

```
https://localhost:44356/api/cart/update
```

Fig 2.36 Update Operation on Cart



```
/api/cart/delete/{id}
```

id * required
integer($int32)
(path)
72

Curl

```
curl -X DELETE "https://localhost:44356/api/cart/delete/72" -H  "accept: */*"
```

Request URL

```
https://localhost:44356/api/cart/delete/72
```

Server response

Code: 200

Response body

```
true
```

Response headers

```
access-control-allow-credentials: true
access-control-allow-origin: https://localhost:44356
content-length: 4
content-type: application/json; charset=utf-8
date: Sun07 Aug 2022 09:32:43 GMT
server: Microsoft-IIS/10.0
x-powered-by: ASP.NET
```

Fig 2.37 Delete Operation on Cart

Day 10: Designing APIs for CRUD operations on Publisher

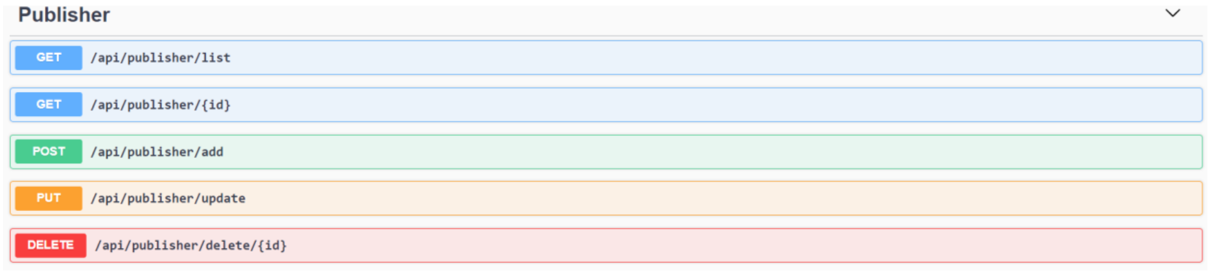
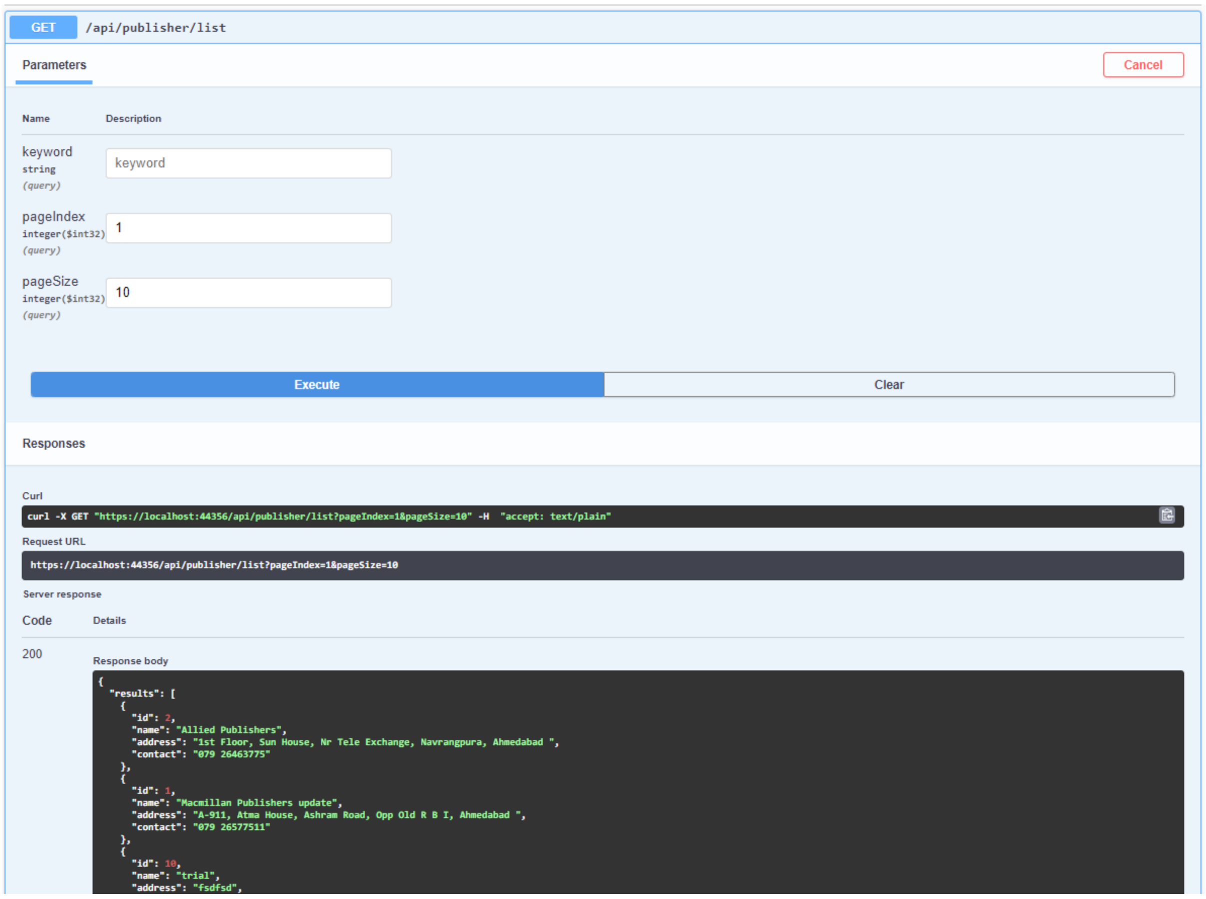


Fig 2.38 CRUD Operations on Publisher

APIs Working: -



Name	Description
keyword string (query)	keyword
pageIndex integer($int32) (query)	1
pageSize integer($int32) (query)	10

Curl

```
curl -X GET "https://localhost:44356/api/publisher/list?pageIndex=1&pageSize=10" -H  "accept: text/plain"
```

Request URL

```
https://localhost:44356/api/publisher/list?pageIndex=1&pageSize=10
```

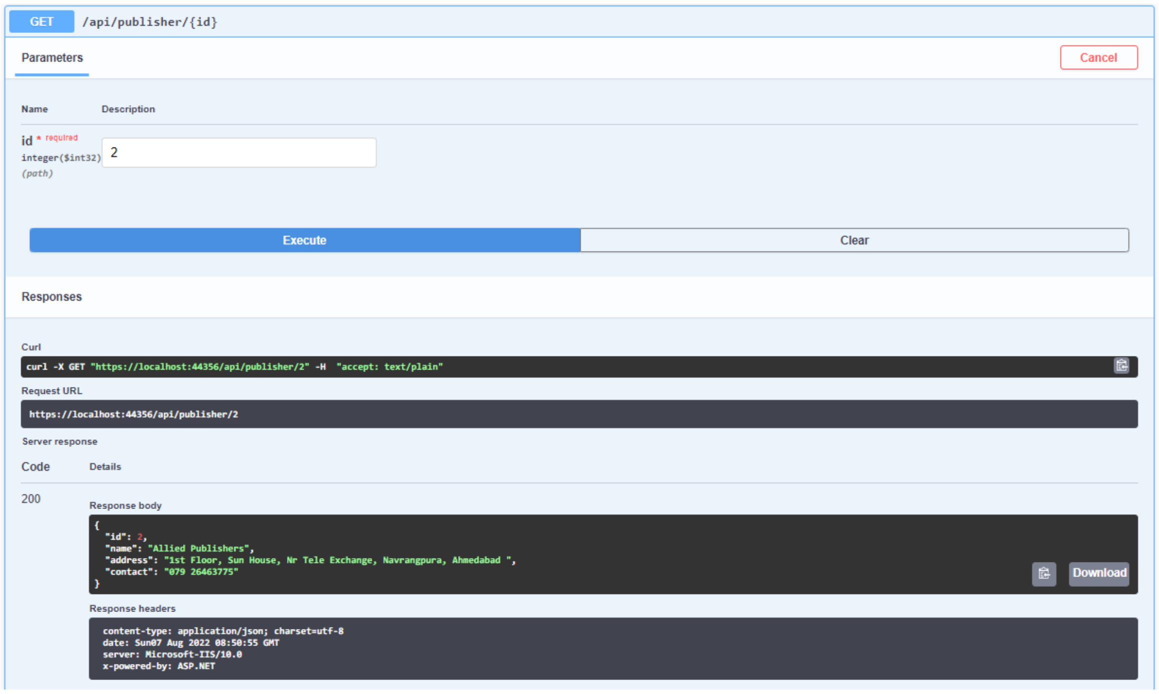
Server response

Code Details

200 Response body

```
{
  "results": [
    {
      "id": 2,
      "name": "Allied Publishers",
      "address": "1st Floor, Sun House, Nr Tele Exchange, Navrangpura, Ahmedabad ",
      "contact": "079 26463775"
    },
    {
      "id": 1,
      "name": "Macmillan Publishers update",
      "address": "A-911, Atma House, Ashram Road, Opp Old R B I, Ahmedabad ",
      "contact": "079 26577511"
    },
    {
      "id": 10,
      "name": "trial",
      "address": "fsdfsd",
```

Fig 2.39 Read Operation on Publisher



/api/publisher/{id}

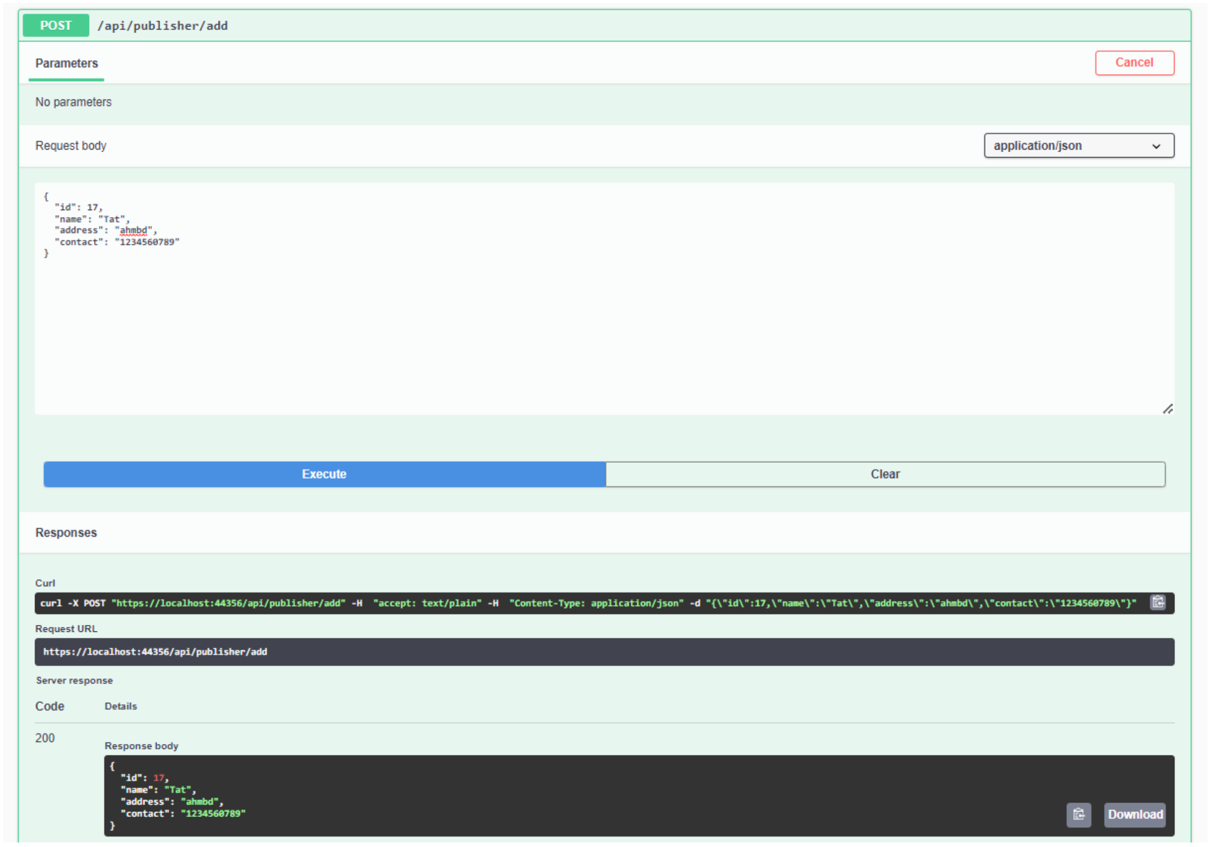
id * required
integer($int32)
(path)
2

Curl
```
curl -X GET "https://localhost:44356/api/publisher/2" -H  "accept: text/plain"
```
Request URL
```
https://localhost:44356/api/publisher/2
```
Server response

Code: 200

Response body
```
{
  "id": 2,
  "name": "Allied Publishers",
  "address": "1st Floor, Sun House, Nr Tele Exchange, Navrangpura, Ahmedabad ",
  "contact": "079 26463775"
}
```
Response headers
```
content-type: application/json; charset=utf-8
date: Sun07 Aug 2022 08:50:55 GMT
server: Microsoft-IIS/10.0
x-powered-by: ASP.NET
```

Fig 2.40 Fetching Publisher through id



/api/publisher/add

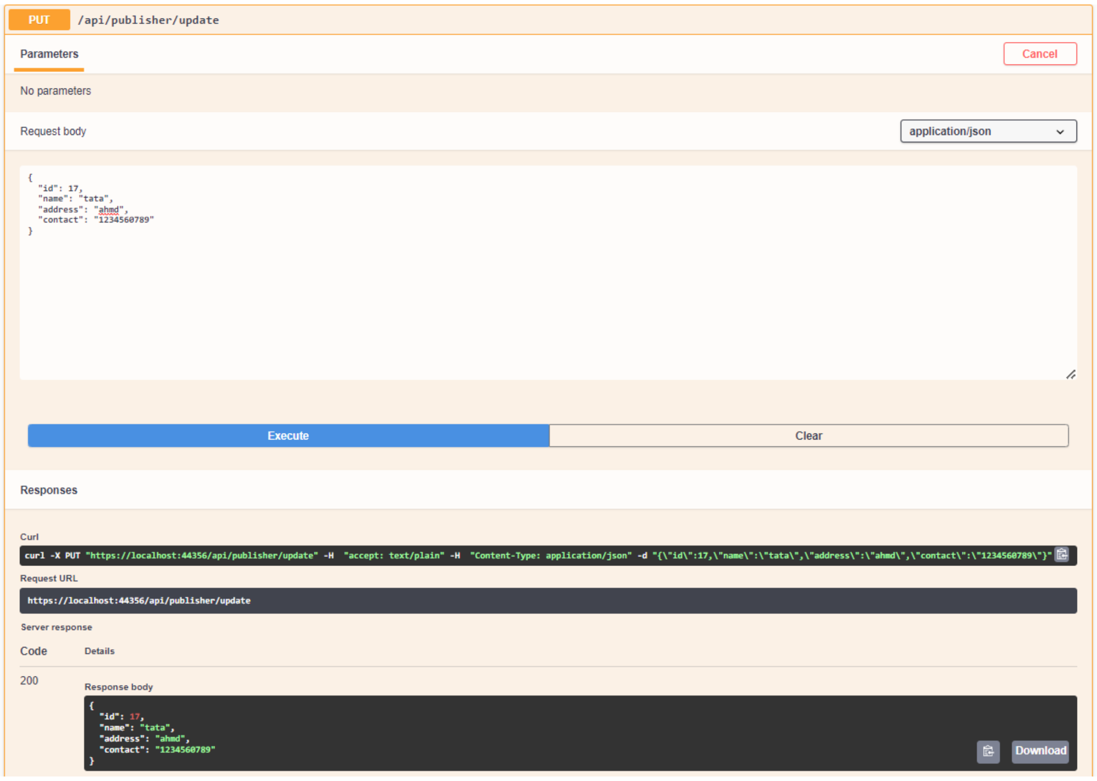
No parameters

Request body: application/json
```
{
  "id": 17,
  "name": "Tat",
  "address": "ahmbd",
  "contact": "1234560789"
}
```
Curl
```
curl -X POST "https://localhost:44356/api/publisher/add" -H  "accept: text/plain" -H  "Content-Type: application/json" -d "{\"id\":17,\"name\":\"Tat\",\"address\":\"ahmbd\",\"contact\":\"1234560789\"}"
```
Request URL
```
https://localhost:44356/api/publisher/add
```
Server response

Code: 200

Response body
```
{
  "id": 17,
  "name": "Tat",
  "address": "ahmbd",
  "contact": "1234560789"
}
```

Fig 2.41 Add Operation on Publisher



```
{
  "id": 17,
  "name": "tata",
  "address": "ahmd",
  "contact": "1234560789"
}
```

Curl

```
curl -X PUT "https://localhost:44356/api/publisher/update" -H  "accept: text/plain" -H  "Content-Type: application/json" -d "{\"id\":17,\"name\":\"tata\",\"address\":\"ahmd\",\"contact\":\"1234560789\"}"
```

Request URL

```
https://localhost:44356/api/publisher/update
```

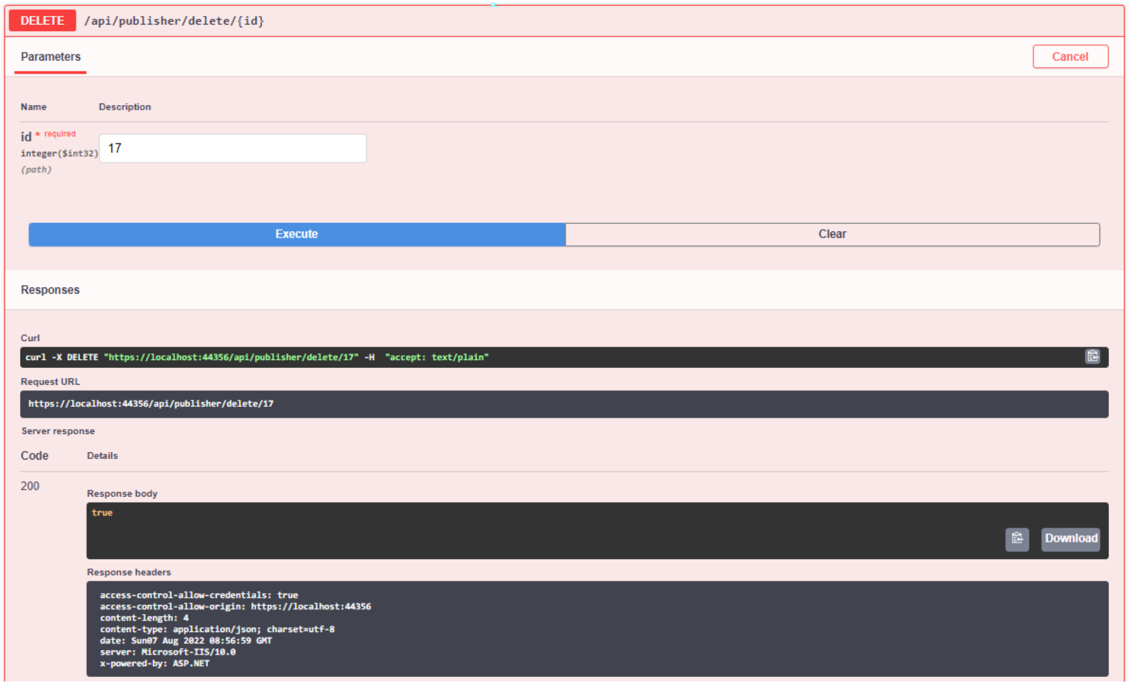
Server response

Code Details

200 Response body

```
{
  "id": 17,
  "name": "tata",
  "address": "ahmd",
  "contact": "1234560789"
}
```

Fig 2.42 Update Operation on Publisher



Name	Description
id * required integer($int32) (path)	17

Curl

```
curl -X DELETE "https://localhost:44356/api/publisher/delete/17" -H  "accept: text/plain"
```

Request URL

```
https://localhost:44356/api/publisher/delete/17
```

Server response

Code Details

200 Response body

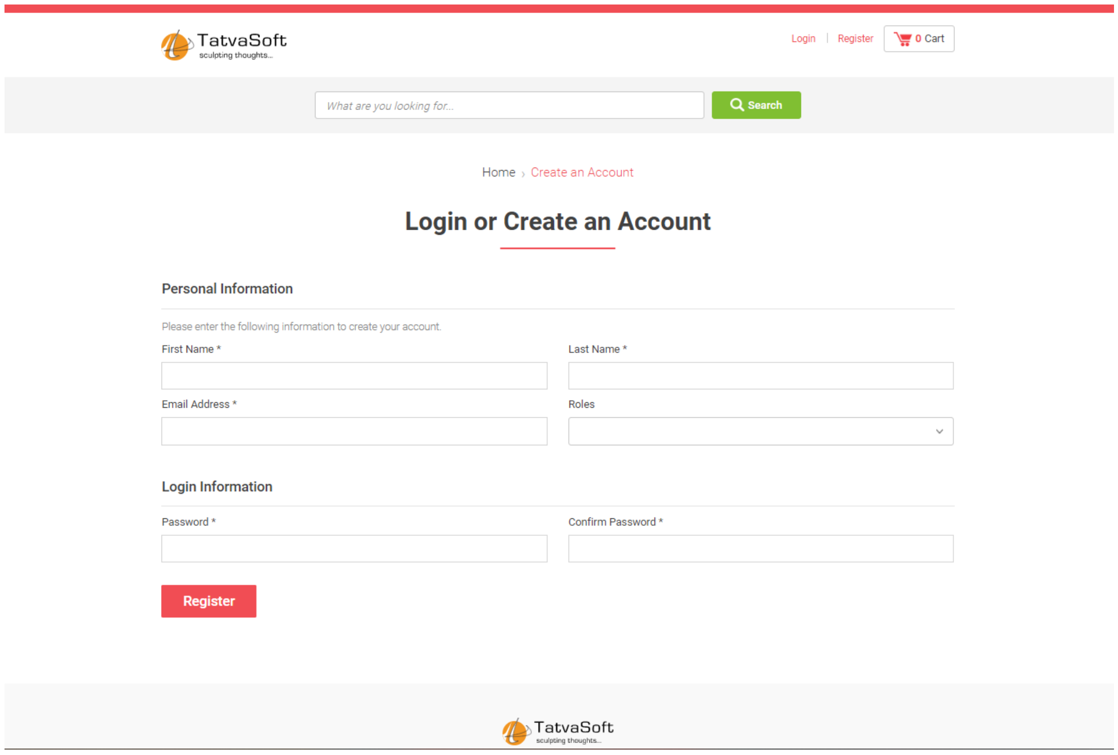
```
true
```

Response headers

```
access-control-allow-credentials: true
access-control-allow-origin: https://localhost:44356
content-length: 4
content-type: application/json; charset=utf-8
date: Sun07 Aug 2022 08:56:59 GMT
server: Microsoft-IIS/10.0
x-powered-by: ASP.NET
```

Fig 2.43 Delete Operation on Publisher

Day 11: Designing frontend for Register and integrate it with backend



Login or Create an Account

Personal Information

Please enter the following information to create your account.

First Name *

Last Name *

Email Address *

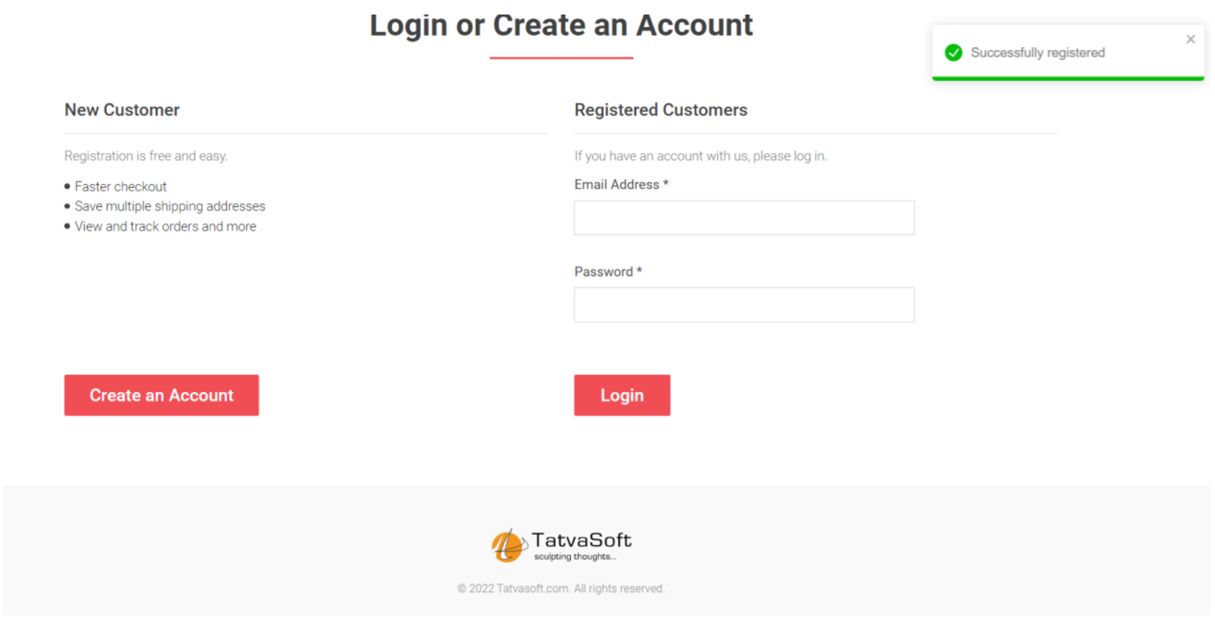
Roles

Login Information

Password *

Confirm Password *

Fig 2.44 Registration Frontend Module


Login or Create an Account

Successfully registered

New Customer

Registration is free and easy.

- Faster checkout
- Save multiple shipping addresses
- View and track orders and more

Registered Customers

If you have an account with us, please log in.

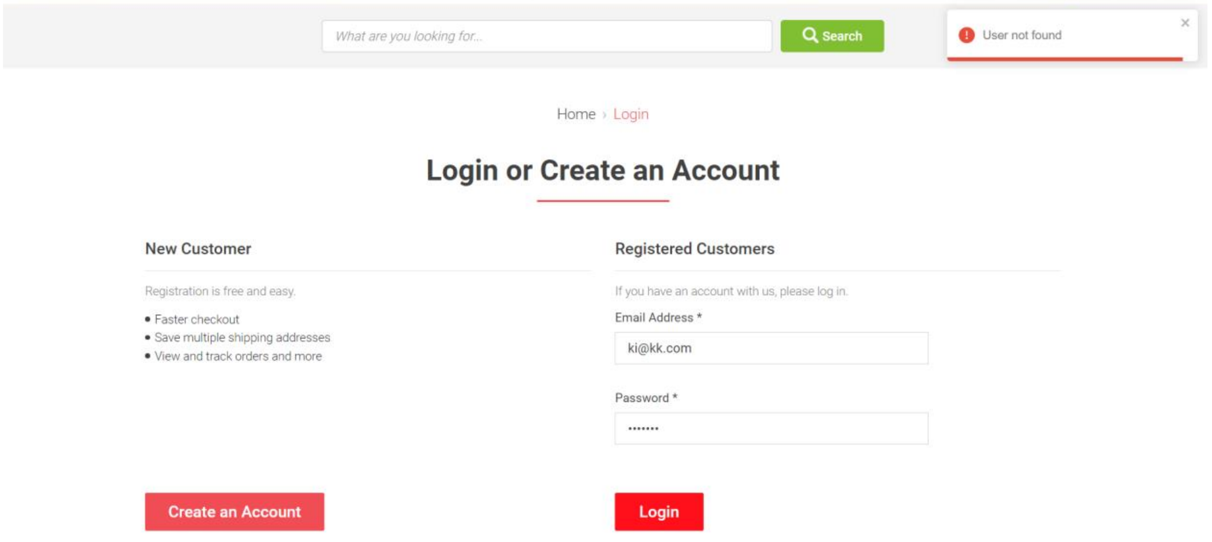
Email Address *

Password *

© 2022 Tatvasoft.com. All rights reserved.

Fig 2.45 Successful Registration

Day 12: Designing frontend for Login and integrate it with backend



Login or Create an Account

New Customer

Registration is free and easy.

- Faster checkout
- Save multiple shipping addresses
- View and track orders and more

Registered Customers

If you have an account with us, please log in.

Email Address *
ki@kk.com

Password *
•••••••

Fig 2.46 Incorrect User Details

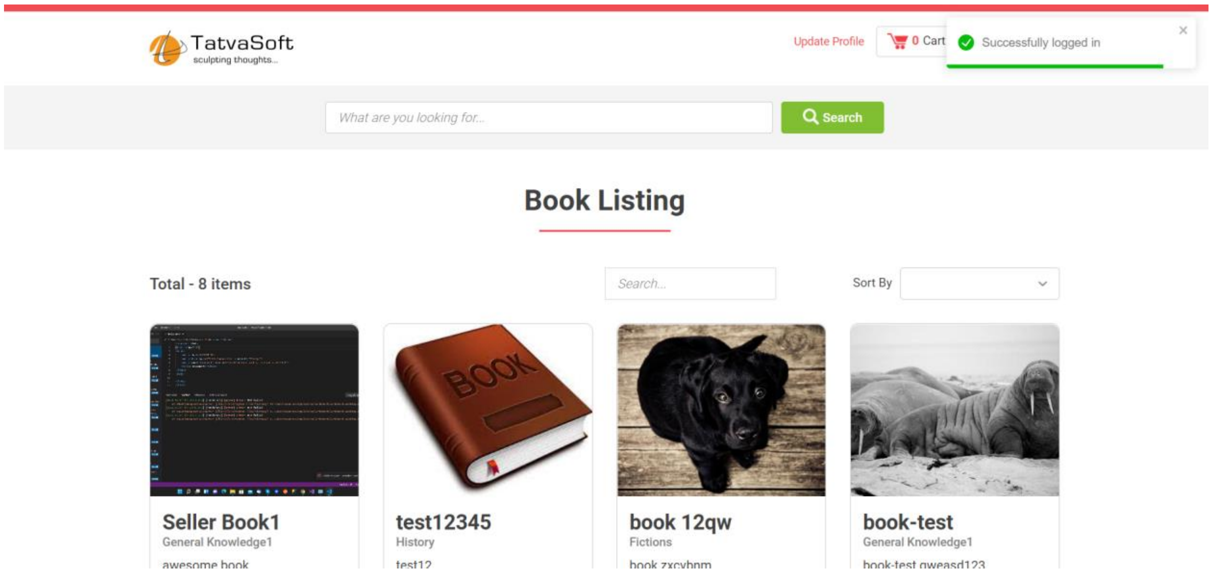


Fig 2.47 Successful Login

Day 13: Designing frontend for Book-Listing and integrate it with backend

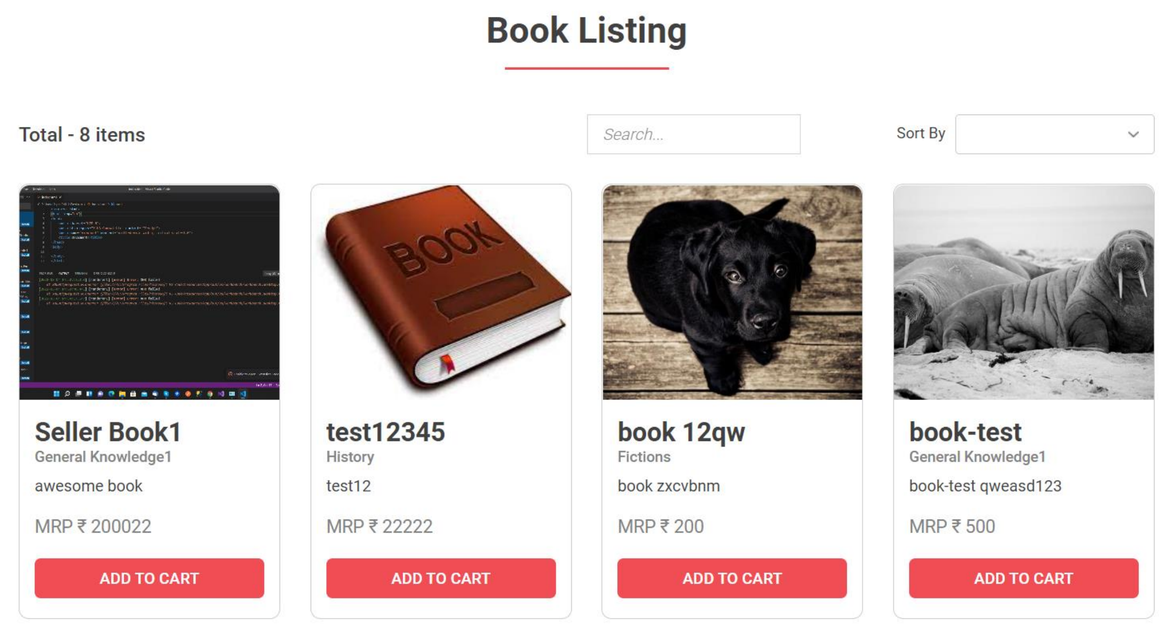


Fig 2.48 Book Listing Module

There are 3 modules on the Website. Admin, Buyer, and Seller. Admin has various functionalities like Users, Categories, Books, and Update Profile as shown in Fig-2.48.

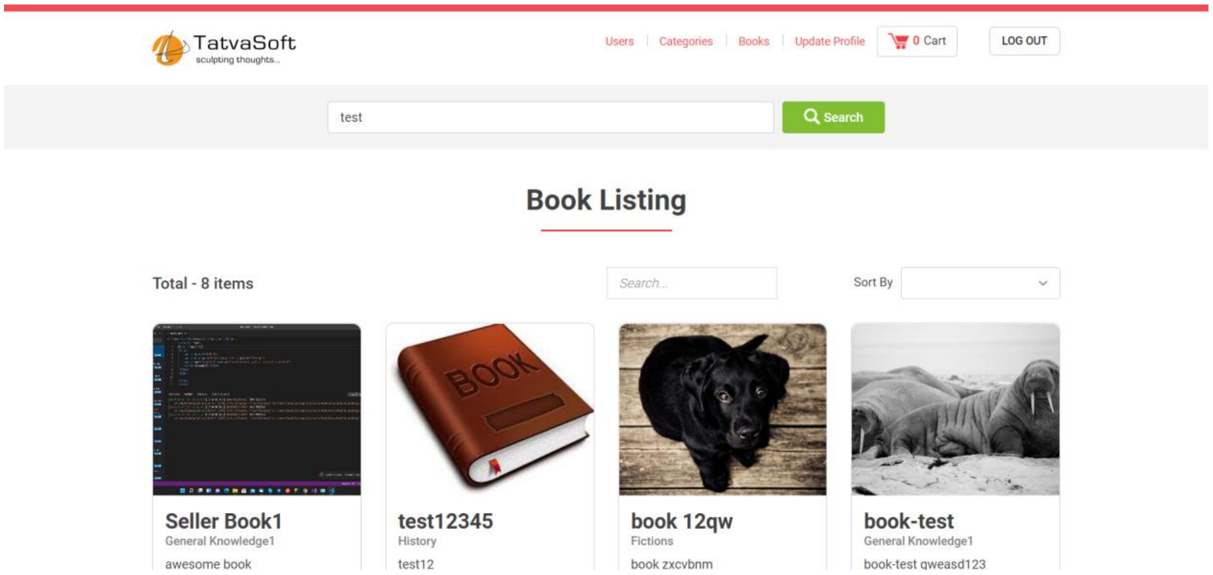
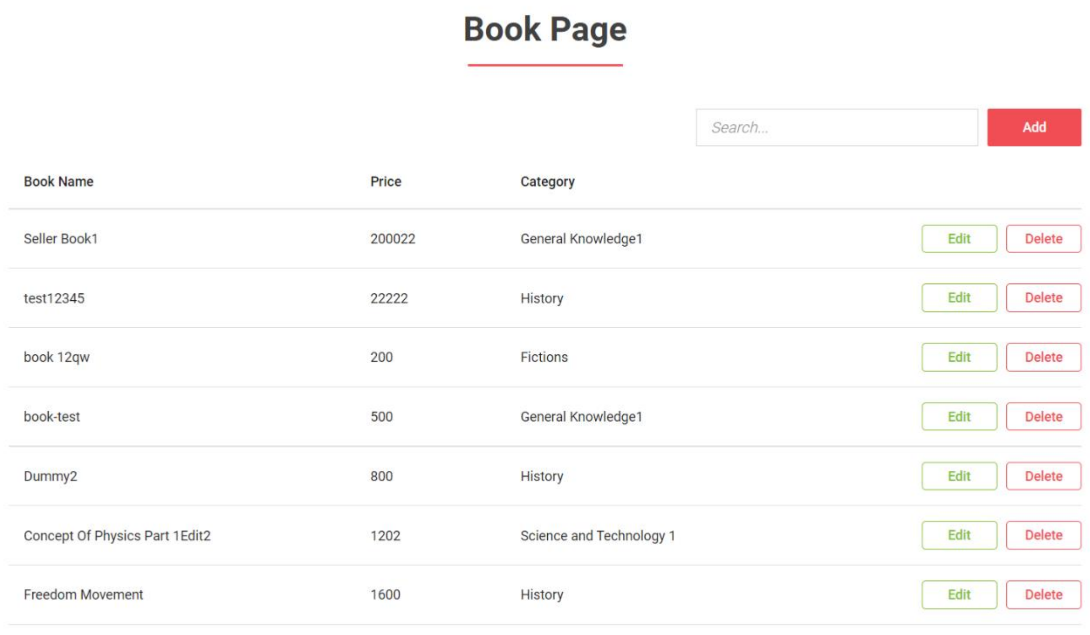


Fig 2.49 Admin Module


Book Page

Search... Add

Book Name	Price	Category	
Seller Book1	200022	General Knowledge1	Edit Delete
test12345	22222	History	Edit Delete
book 12qw	200	Fictions	Edit Delete
book-test	500	General Knowledge1	Edit Delete
Dummy2	800	History	Edit Delete
Concept Of Physics Part 1Edit2	1202	Science and Technology 1	Edit Delete
Freedom Movement	1600	History	Edit Delete

Fig 2.50 Book-Listing Page

The seller can Buy or Sell books. Seller Module is shown in Fig 2.51

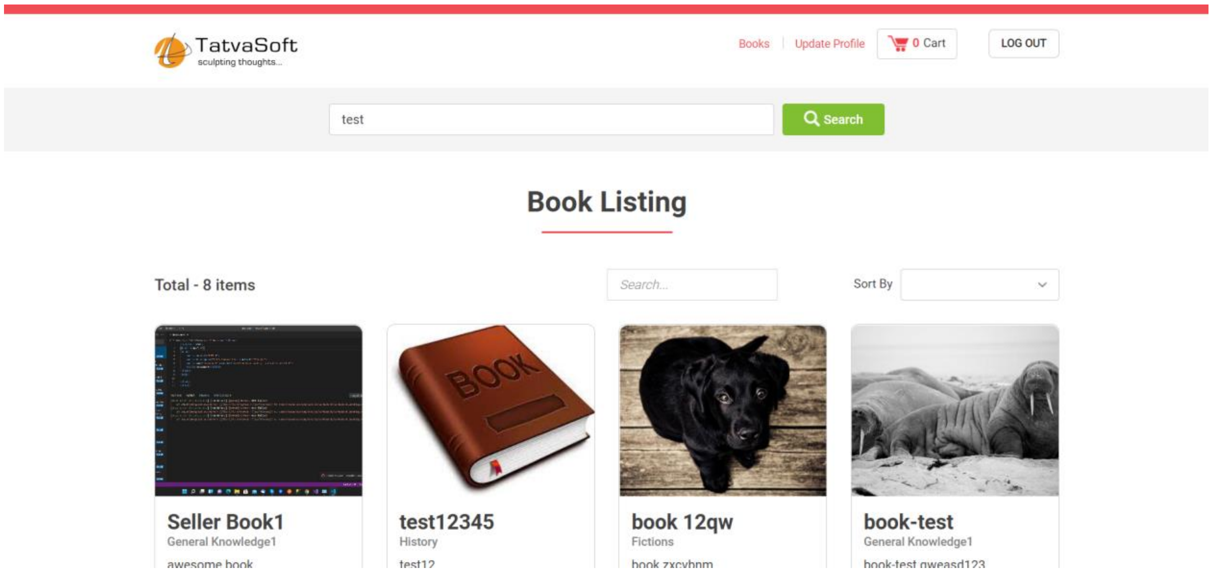


Fig 2.51 Seller Module

Day 14: Designing Global Search for Books

Users can Search based on his/her own interests.

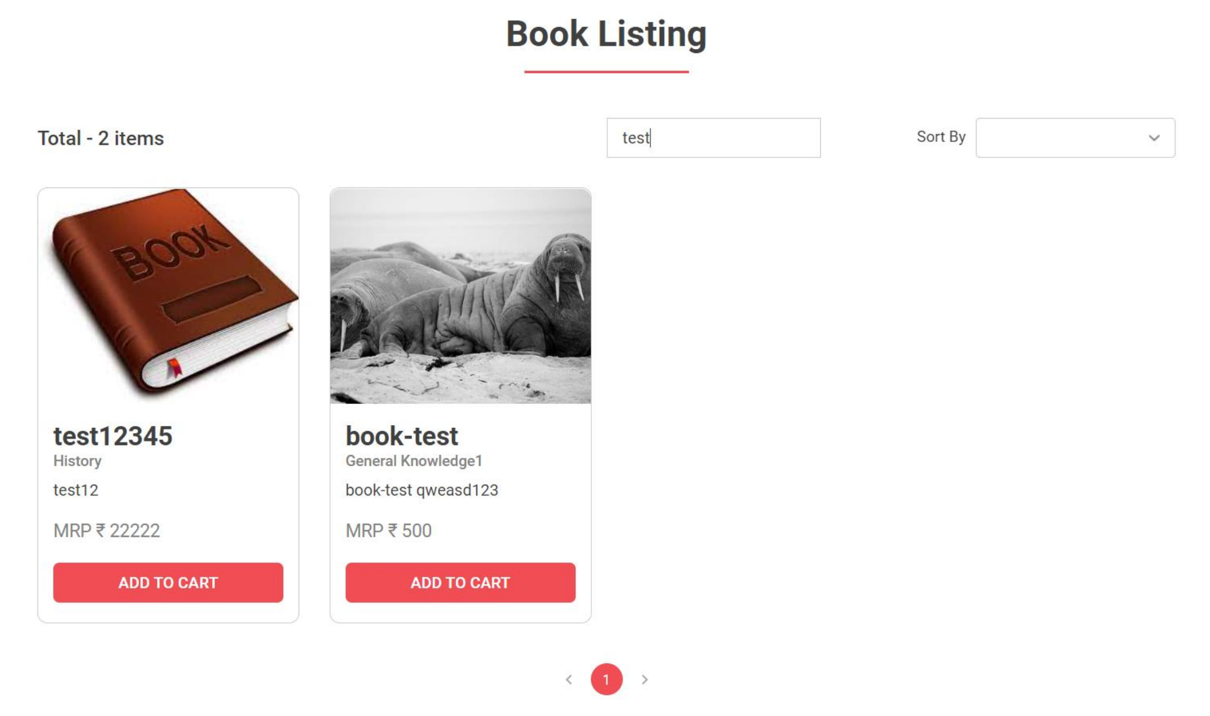


Fig 2.52 Global Search Implementation

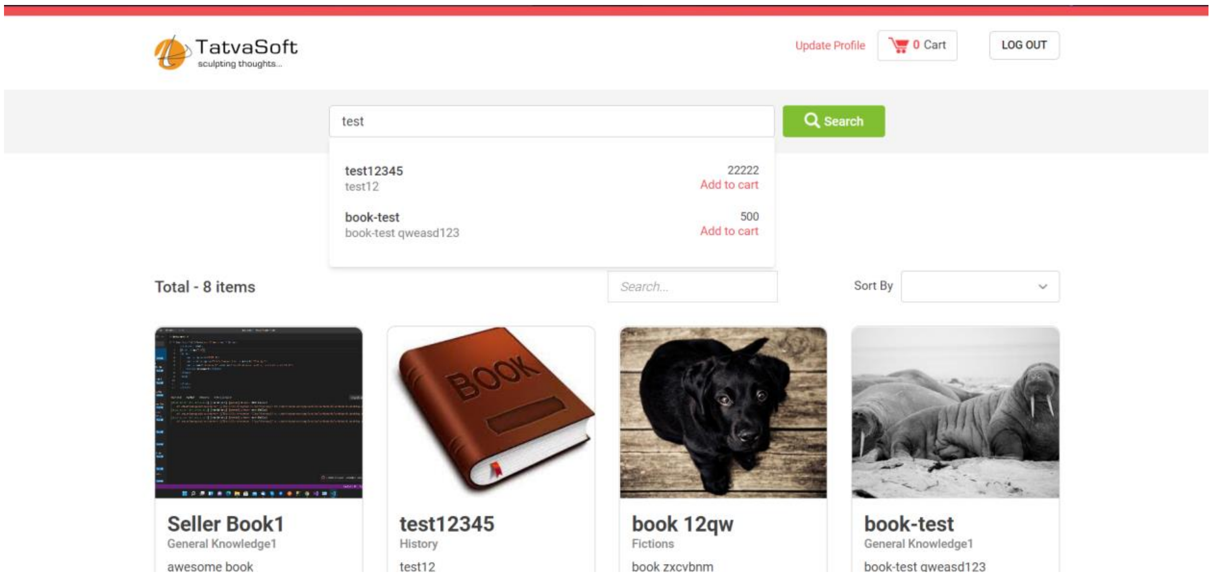
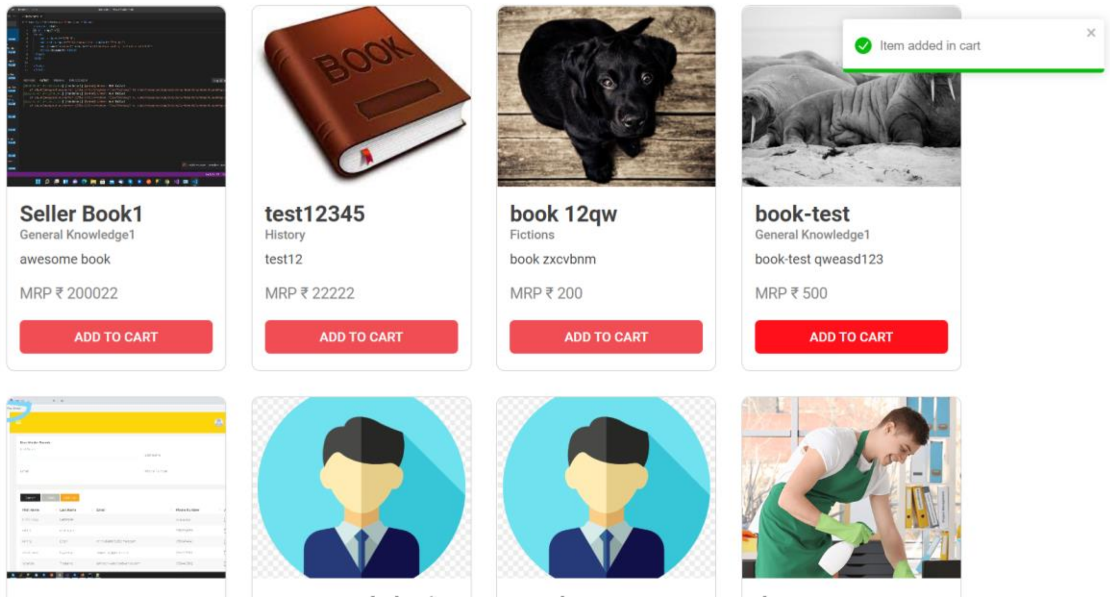


Fig 2.53 Website Header Search

Day-15: Designing frontend for Cart and integrate it with backend



Item added in cart

Seller Book1
General Knowledge1
awesome book
MRP ₹ 200022

test12345
History
test12
MRP ₹ 22222

book 12qw
Fictions
book zxcvbnm
MRP ₹ 200

book-test
General Knowledge1
book-test qweasd123
MRP ₹ 500

Fig 2.54 Successful adding in Cart

CHAPTER-3: SKILL LEARNED

3.1 ABOUT THE SKILLS

In the internship we learn the basics of .net, React, PostgreSQL and postman.

Backend Technologies: -

1. ASP.net: -

- ASP.net is a framework of C#.
- C-Sharp is one of the most widely used languages for creating system backends. It's because of its incredible features, such as Windows server automation. Apart from that, it's fantastic because it runs codes quite quickly. It can also be used to create CLI applications and game creation.
- Some key features of C# include cross-platform compatibility, garbage data, and values collection, and object-oriented programming.
- Some widely used C# frameworks for backend development are ASP.NET Core and .NET MVC.

In ASP.net we work on MVC Architecture.

ASP.NET Core is an open-source and free framework that follows in the footsteps of ASP.NET, a widely-used backend created in partnership with the .NET Foundation. ASP.NET Core is a modular framework that can run across the entirety of the .NET Framework across Windows and .NET Core.

The most vital skill as a .NET developer includes the knowledge of ASP.NET MVC (Model View Controller).

It permits you to manage the control of every function of the application. MVC refers to the web framework allowing you to create an efficient web application with ease of control, high security, and robust technology.

It is the software providing access to client-side technologies allowing the developer to build software swiftly and efficiently.

With the help of ASP.net, one can build a better, scalable, and faster web application that can gain popularity among clients.

In the professional field, ASP.NET MVC has replaced other frameworks.

2. PostgresSQL:-

- PostgreSQL, also known as Postgres, is a free and open-source relational database management system emphasizing extensibility and technical standards compliance.
- PostgreSQL is a powerful, open source object-relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads.
- It uses Structured Query Language (SQL) for accessing the data in the tables of the database, and hence it is also called Postgres.

3. POSTMAN:-

- Postman is an Application Programming Interface (API) testing tool.
- Used Entity Framework with Code First approach and worked on Code First Migrations along with Fluent API.
- Integrated functionality to generate Postman import files from the RAML specs to ease API endpoint testing.
- Performed API Testing using Rest Client and SOAP UI.

- Utilized Postman to make API requests, SQLite to store users, and TODO items.
- Worked on jQuery that communicates with Web API for Ajax functionality.

Frontend Technologies:-

1. Typescript:-

- TypeScript is a free, open-source programming language developed and maintained by Microsoft. It is a strict superset of JavaScript.
- TypeScript is a statically compiled programming language for writing clear and concise JavaScript code. It’s fulfilling the same purpose as JavaScript and can be used for both client-side and server-side applications. In addition, the libraries of JavaScript are also compatible with TypeScript.
- TypeScript is a programming language that expands on traditional JavaScript. It sprinkles some new syntax on top of JS to support static typing. Static typing is an extremely useful feature that allows engineers to catch many runtime bugs.
- TypeScript is a programming language that supports both dynamic and static typing. It provides classes, visibility scopes, namespaces, inheritance, unions, interfaces, and many other features. Also, it offers comments, variables, statements, expressions, modules, and functions.
- In TypeScript, types are optional, and any JavaScript file is a valid TypeScript file. While the compiler will notify you if any of your initial files have type issues, it will still provide a JavaScript file that works. TypeScript will stand up to expectations, and it’s simple to improve your skills over time.
- TypeScript is compiled to JavaScript, it is suitable for both the frontend and backend of app development.
- Besides, JavaScript is a preferred programming language for the frontend of web pages and apps. As a result, TypeScript may be used for the same reason, but it also works well on the server-side for complicated and large-scale enterprise projects.

2. React with Typescript:-

- React is the most popular JavaScript front-end framework in use today.
- React is a JavaScript library for creating user interfaces.
- It allows us to write HTML code directly in our React project. Using TypeScript with React provides better IntelliSense, code completion for JSX.
- Default TypeScript support for common libraries.

3.2 HOW DO I LEARN THE SKILLS

I learn the basics skill of frontend and backend technologies using YouTube. I mention all links which I use in that by topics.

Reactjs:-

1. **Typescript**

I learn typescript using the YouTube platform. YouTube Channel Code Step By Step provides a typescript Hindi tutorial. First, I learn the core concept of type scripting and also remember. For practice, I perform all tasks which are included in the video. For more practice, our mentor allocates task which is relevant to that concept. It is really helpful to me in learning and understanding typescript fundamentals.

2. **Reactjs with Typescripts :**

I learn typescript with react using the YouTube channel Codevolution.In this React TypeScript for beginners series, I learn to use TypeScript with React by building a few components of varying complexity. With static type checking, he says about potential bugs as I am typing the code, rather than heading to the browser and figuring it out at runtime. TypeScript with React also provides a way to describe the shape of an object hence providing better documentation and autocomplete. Typescript even makes maintenance and refactoring of large code bases much easier.

Some of the videos provided by the company which is about watching and understanding advanced level features of it.

ASP.NET:-

3. ASP.NET Core Web API

I Learn ASP.net via the YouTube channel WebGentle. ASP.NET Core Web API is the latest and most powerful framework for the development of RESTful Web API. This Asp.Net Core Web API is open source and supported by Microsoft. RESTful Web APIs are the most essential part of any modern world application. These RESTful Web APIs help us to extend our application on multiple platforms like Web apps, Android apps, iOS apps, etc. In this, I learn ASP.NET Core Web API step by step from very basics to advanced level concepts. Web API from ASP.NET Core is the same as the one from ASP.NET Core MVC. The Web API offers a simple communication way based on Representational State Transfer (REST). With REST, HTTP verbs such as GET, POST, PUT, and DELETE are used.

Some of the videos provided by the company which is about watching and understanding advanced level features of it.

4.PostgreSQL

I learn PostgreSQL using a YouTube channel known as Programming Gugu. It has Basic PostgreSQL Tutorial. First, I learn how to query the data from a single table using basic data selection techniques such as selecting columns, sorting result sets, and filtering rows. Then, I learn about creating databases, and tables. Also, I learn how to retrieve insert updates and delete data from the database. And also condition base data retrieve and group result data.

Some of the videos provided by the company which is about watching and understanding advanced level features of it.

CHAPTER-4: OVERALL EXPERIENCE

4.1 TECHNICAL EXPERIENCE

- How to identify the problems and find the solution of problems.
- Understand the different type of error and find a different way to handle the error.
- Learn how the APIs work in backend and how the integrate backend, frontend and database with each other.
- Learn concept of middleware.
- Learn industries code standard and practice.

4.2 PERSONAL EXPERIENCE

During this project,

- I worked in a team so I learned how to work with team members, understand the team member’s points and help the team members.
- Learned how to maintain the project summary in industries.
- How to maintain the code on GitHub so other team members can use the code as a collaborator.
- Learned reactjs with typescript which is a new concept.

CHAPTER-5: CONCLUSION

5.1 CONCLUSION

The E-Book store project is designed using .net, react with typescript, and PostgreSQL database. E-Book Store System is a simple project similar to a shopping cart or an eCommerce website but is only for book shopping. Through a web browser customers need to log in or register and after registering customers can search for a book by its title or author, add it to the shopping cart, and finally purchase the book. Here I provide functions based on the type of user. There are three types of users buyer, seller, and admin. Each has its own functionalities like the user can purchase a book, a seller can add, delete, and update the book based on the requirement, and the admin has all rights to handle the application.

CHAPTER-6: FUTURE SCOPE

6.1 FUTURE SCOPE

In the future, we are planning to add some additional functionalities like

- Our priority is for users to add, update and remove items from their carts.
- Second, for the admin panel, the admin can add, update and remove users and also book and their categories.
- Third for the seller panel seller can update, add and remove their book.
- Fourth is user can Place an order with cash on delivery or online payment.
- Last for making the website live 24X7.

CHAPTER-7: BIBLIOGRAPHY

7.1 BIBLIOGRAPHY

1. https://reactjs.org/docs/getting-started.html

2. https://www.typescriptlang.org/docs/

3. https://www.typescriptlang.org/docs/handbook/react.html

4. https://docs.microsoft.com/en-us/aspnet/core/web-api/?view=aspnetcore-5.0

5. https://www.postgresql.org/docs/

1. Reactjs:
   https://www.youtube.com/watch?v=QmwZwSiOic0
2. Reactjs Component Structure:
   https://www.youtube.com/watch?v=9VIiLJL0H4Y
3. what is type script and why it is used over Javascript / Typescript basic:
   https://www.youtube.com/watch?v=ucZRzCYQRQo&list=PL8p2I9GklV46OtmTnYKPa2Mp9sPLWpRA-&index=2
4. Reactjs with Typescripts:
   https://www.youtube.com/watch?v=TiSGujM22OI&list=PLC3y8rFHvwi1AXijGTKM0BKtHzVC-LSK
5. useEffect Hook:
   https://www.youtube.com/watch?v=0ZJgIjIuY7U&list=PLZlA0Gpn_vH8EtggFGERCwMY5u5hOjf-h&index=2
6. useMemo Hook:
   https://www.youtube.com/watch?v=THL1OPn72vo&list=PLZlA0Gpn_vH8EtggFGERCwMY5u5hOjf-h&index=3
7. Formik:
   https://www.youtube.com/watch?v=a94FOvaBomQ&list=PLC3y8rFHvwiPmFbtzEWjESkqBVDbdgGu
8. Asp.Net Core Web API Tutorial | Asp.Net Core 5 Web API Tutorial:
   https://www.youtube.com/playlist?list=PLaFzfwmPR7_IPzBR4AI0eoojmIdTFJmHs
9. PostgreSQL:
   https://www.youtube.com/playlist?list=PLk1kxccoEnNEtwGZW-3KAcAlhI_Guwh8x

Dupli Checker

PLAGIARISM SCAN REPORT

5%
Plagiarised

95%
Unique

Date	2022-11-18
Words	989
Characters	6225

Content Checked For Plagiarism

ABSTRACTBook-E-Sale is an E-commerce website for book purchasing and selling. It contains basic functionalities such as Login, Registration, Book Search, Book listing, Add to Cart, and Logout. It has buyer, seller and admin module. User can become seller or buyer. Website help people to find the book they need. People can sell their non-used books online. It follows C2C Business model.Book-E-Sale is an E-commerce website for book purchasing and selling. It contains basic functionalities such as Login, Registration, Book Search, Book listing, Add to Cart, and Logout. It has buyer, seller and admin module. User can become seller or buyer. Website help people to find the book they need. People can sell their non-used books online. It follows C2C Business model.1.2
1.3 ABOUT TECHNOLOGYThis Project is a Small E-Book selling website using ReactJS, TypeScript, ASP.net, and PostgreSQL. It is a website to buy and sell books. It has various functionalities like login, registration, Addition of book to cart, etc...
I'm working on Book-E-Sale System as an intern. E-Book Store System is a simple project similar to a shopping cart or eCommerce website but is only for book shopping and selling books. Through a web browser the customers need to login or register later can search for a book by its title or author, later can add it to the shopping cart, and finally purchase the books. And the sellers can add books for selling.
In this project, we work on some modules like:
For Admin:
A. Dashboard – For the admin dashboard, you will be able to all the basic access in the whole system. Such as a summary of products, orders, and categories. B. Manage Books– The admin has access to the books management information system. He can add, update, and delete the books.
C. Manage Categories – The page where the admin can add, edit, and delete category information. D. Manage Orders – As the main function of the admin, the admin can accept or reject the order from the customers on a case-to-case basis and the list of customer orders is listed.
E. Manage User– The admin can manage the user's account. **Admin can add, update and Block users in the system.**
For Buyer:
A. Login Page – Customer enters their website credentials on this page to gain access to log in.
B. Register Page– The page where new customer created their login credentials for the website.
C. Home Page– When a customer visits the website, this is the system's default page.
This page shows the books for sale in the store, or by entering a keyword in the search box above the books.
D. Book View Page – The page on which the product's specific information is shown, as well as the page on which the customer adds the product to his or her cart.
E. Cart List Page– The page that lists the items that customers have chosen.
This is the page where the customer can complete the order checkout process.
F. Order Page – The page that lists the customer's orders.
For Seller:
A. Login Page –Seller enters their website credentials on this page to gain access to log in.
B. Register Page– The page where new seller created their login credentials for the website.

C. Manage Books– The seller has access to the books management information system. They can add, update, and delete

the books or prices on their own.
• E-Book Store is a specific requirement of the client that integrates the buying and selling services specifically to their customers.
• The details regarding all users, and books can also be maintained as their information is very helpful and sometimes becomes a critical requirement.
• Allows the user to get registered from their places and transact for the required product.
• To overcome these problems, we develop "Book-E-Sale".
1 TECHNICAL EXPERIENCE
• How to identify the problems and find the solution of problems.
• Understand the different type of error and find a different way to handle the error.
• Learn how the APIs work in backend and how the integrate backend, frontend and database with each other.
• Learn concept of middleware.
• Learn industries code standard and practice.

4.2 PERSONAL EXPERIENCE
During this project,
• I worked in a team so I learned how to work with team members, understand the team member's points and help the team members.
• Learned how to maintain the project summary in industries.
• How to maintain the code on GitHub so other team members can use the code as a collaborator.
• Learned reactjs with typescript which is a new concept.

CHAPTER-5: CONCLUSION
5.1 CONCLUSION
The E-Book store project is designed using .net, react with typescript, and PostgreSQL database. E-Book Store System is a simple project similar to a shopping cart or an eCommerce website but is only for book shopping. Through a web browser customers need to log in or register and after registering customers can search for a book by its title or author, add it to the shopping cart, and finally purchase the book. Here I provide functions based on the type of user. There are three types of users buyer, seller, and admin. Each has its own functionalities like the user can purchase a book, a seller can add, delete, and update the book based on the requirement, and the admin has all rights to handle the application.
CHAPTER-6: FUTURE SCOPE
6.1 FUTURE SCOPE
In the future, we are planning to add some additional functionalities like
• Our priority is for users to add, update and remove items from their carts.
• Second, for the admin panel, the admin can add, update and remove users and also book and their categories.
• Third for the seller panel seller can update, add and remove their book.
• Fourth is user can Place an order with cash on delivery or online payment.
• Last for making the website live 24X7.

Matched Source

Similarity 8%
Title:Attendance Management System in JavaScript with Source ...

https://itsourcecode.com/humix/video/d8a341b0848e12edd1546dd10423d384c116312e542c39ef196642af1e80f6e0

Similarity 7%
Title:Online Book Store Project in Python Django
WebThis page shows the books for sale in the store, or by entering a keyword in the search box above the books. Book View Page – The page on which the product's specific information ...
https://projectworlds.in/online-book-store-project-in-python-django/

Similarity 7%

Title:Synopsis_final22.doc - SYNOPSIS OF BOOKSWAP 1 Abstract ...

Synopsis_final22.doc - SYNOPSIS OF BOOKSWAP 1 Abstract ...https://www.coursehero.com › filehttps://www.coursehero.com › fileThis is the pagewhere the customer can complete the order checkout process. My Order Page – The page that lists the customer's orders.

https://www.coursehero.com/file/148119855/Synopsis-final22doc/