

## AA BOOK STORE

### Installation

- Node.js
- npm (Node Package Manager)
- MongoDB

### Running the Application

**Start the development server: BACKEND(mern-server)**

**In VS code terminal:**

D:\AA book store>cd mern-server

D:\AA book store>npm start

Pinged your deployment. You successfully connected to MongoDB!

**Start the development server: FRONTEND(mern-client)**

**In VS code terminal:**

D:\AA book store>cd mern-client

D:\AA book store>npm run dev

Local: <http://localhost:5173/>

After press ctrl+mouse on above link

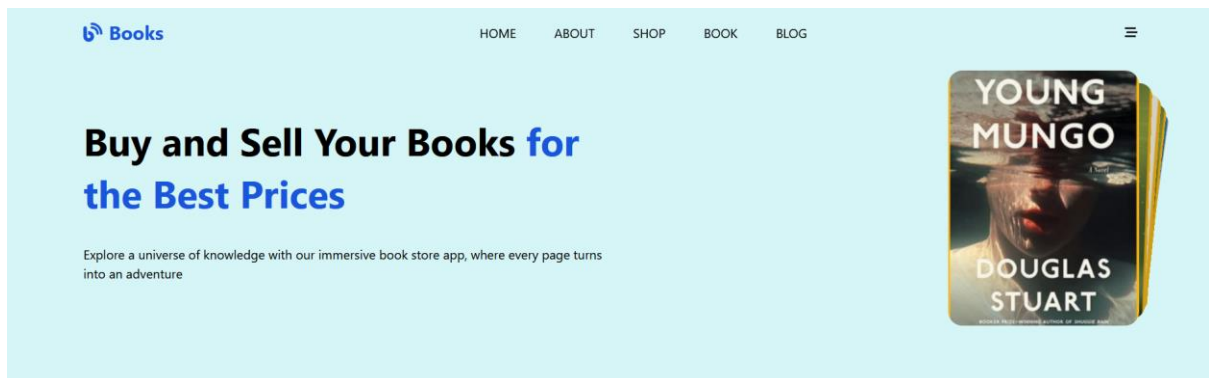
Navigated to **[http://localhost:5173\(Vite\)](http://localhost:5173(Vite))**

Then **<http://localhost:5173/sign-up->** navigated to Sign-up

After Sign-up Navigated to login page.

After login user can able to view Home page.

**Landed to HOME Page:**

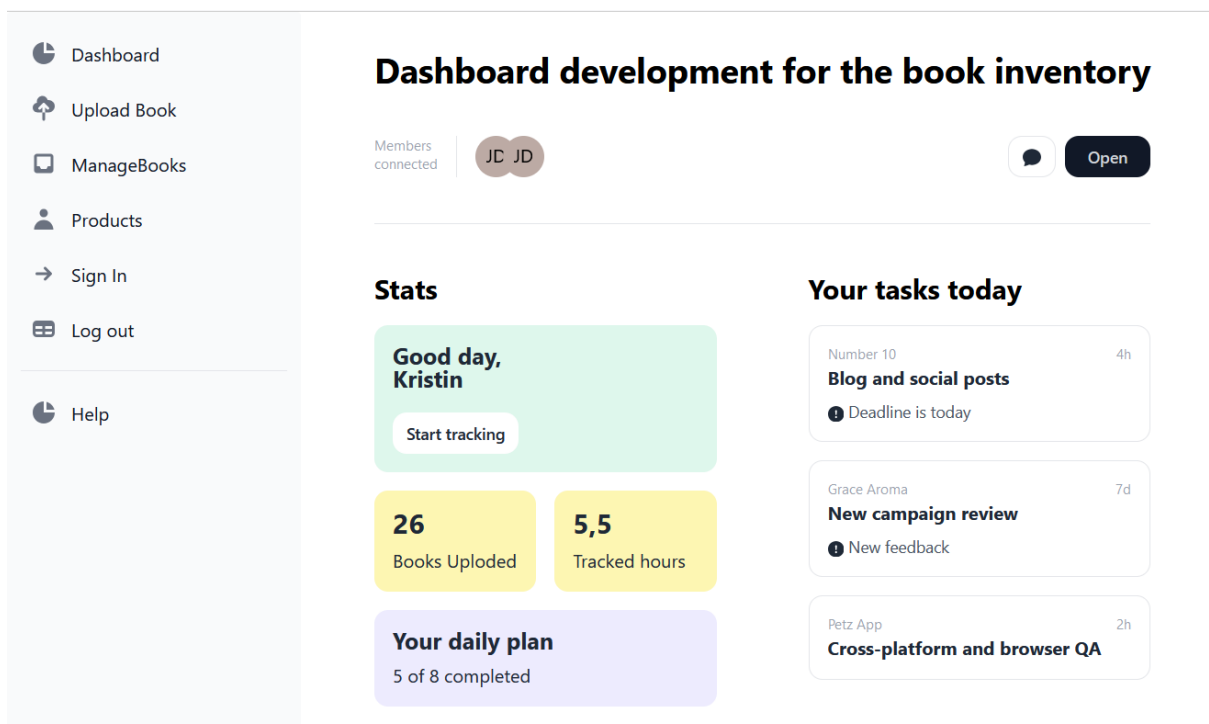


**Home Page:** After Login user will be landed to Home page.

**Shop:** When user clicks on Shop page they can able to see the books that are uploaded by admin.

**Book:** In Book page it will be redirected to dashboard page where admin can able to edit ,manage, upload, and delete books.

In Home Page: when admin click on Book page navigated to **Dashboard page:**

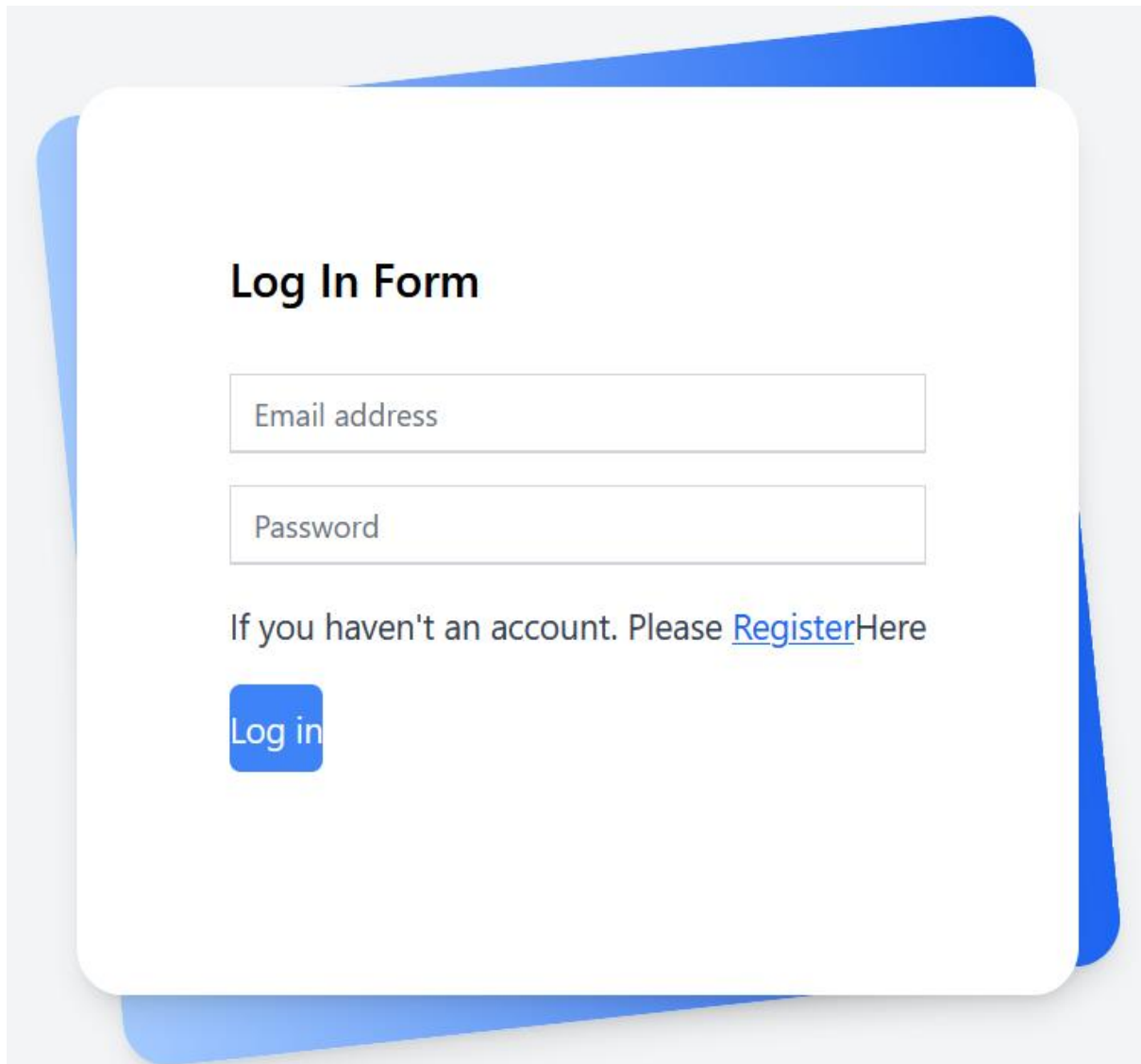


**Upload Book:** Admin can able to Upload books.

**Manage Books:** Admin can able to Manage the inventory such as he can able to view the upload books, edit books, delete books.

## FRONTEND CODE:

### LOGIN PAGE:



```
import React, { useContext,useState } from 'react'
import { Link, useLocation, useNavigate } from 'react-router-dom';
import { AuthContext } from '../contexts/AuthProvider';
```

```
const Login = () => {
  const {login} = useContext(AuthContext);
  const [error, setError] = useState("");
```

```
  const location = useLocation();
  const navigate = useNavigate();
  const from = location.state?.from?.pathname || "/";
```

```
  const handleLogin = (event) => {
```

```

event.preventDefault();
const form = event.target;
const email = form.email.value;
const password = form.password.value;

login(email, password).then((userCredential) => {
  const user = userCredential.user;
  alert("Login Successfull!")
  navigate(from, {replace: true})
})

.catch((error) => {
  const errorCode = error.code;
  const errorMessage = error.message;
  setError(errorMessage)
});

}
return (
  <div className="min-h-screen bg-gray-100 py-6 flex flex-col justify-center sm:py-12">
    <div className="relative py-3 sm:max-w-xl sm:mx-auto">
      <div
        className="absolute inset-0 bg-gradient-to-r from-blue-300 to-blue-600 shadow-lg transform -skew-
        y-6 sm:skew-y-0 sm:-rotate-6 sm:rounded-3xl">
      </div>
      <div className="relative px-4 py-10 bg-white shadow-lg sm:rounded-3xl sm:p-20">
        <div className="max-w-md mx-auto">
          <div>
            <h1 className="text-2xl font-semibold">Log In Form</h1>
          </div>
          <div className="divide-y divide-gray-200">
            <form onSubmit ={ handleLogin } className="py-8 text-base leading-6 space-y-4 text-gray-700
            sm:text-lg sm:leading-7">
              <div className="relative">
                <input id="email" name="email" type="text" className="peer h-10 w-full border-b-2 border-gray-
                300 text-gray-900 focus:outline-none focus:borer-rose-600" placeholder="Email address" />
              </div>
              <div className="relative">
                <input id="password" name="password" type="password" className="peer h-10 w-full border-b-2
                border-gray-300 text-gray-900 focus:outline-none focus:borer-rose-600" placeholder="Password" />
              </div>
              <p className='text-red-600 text-base'>
                {error ? "Email or password is not valid" : ""}
              </p>
              <p> If you haven't an account. Please <Link to="/sign-up" className='text-blue-600
              underline'>Register</Link>Here</p>
              <div className="relative">
                <button className="bg-blue-500 text-white rounded-md px-6x py-2">Log in</button>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>
)

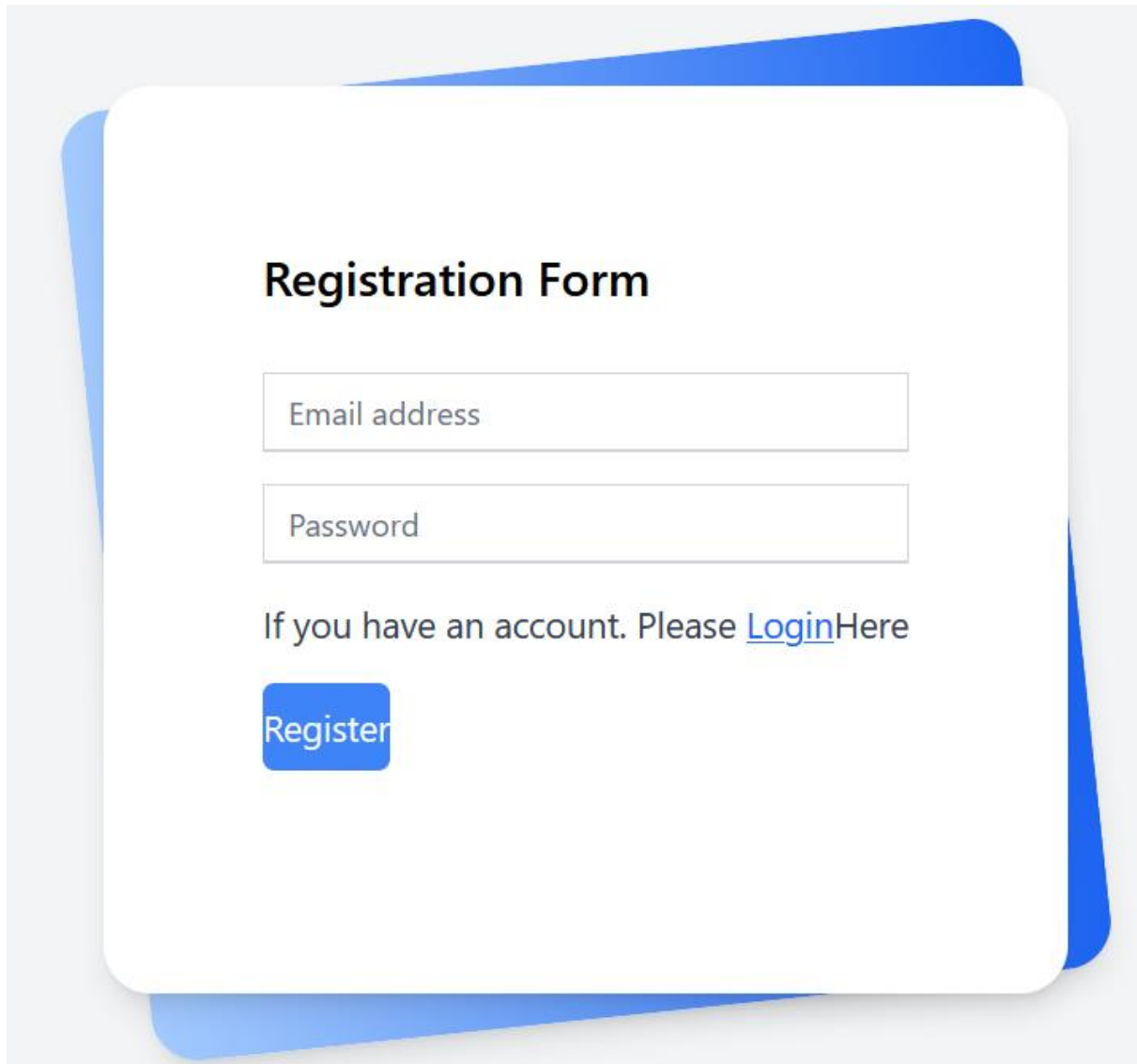
```

```
</div>  
</div>
```

```
)  
}
```

```
export default Login
```

## REGISTER PAGE:

A registration form UI with a white background and rounded corners, set against a light gray background with blue decorative shapes. The form contains a title "Registration Form", two input fields labeled "Email address" and "Password", a text prompt "If you have an account. Please [Login](#)Here", and a blue "Register" button.

```
import React, { useContext,useState } from 'react'  
import { Link, useLocation, useNavigate } from 'react-router-dom';  
import { AuthContext } from '../contexts/AuthProvider';
```

```
const Signup = () => {  
  const { createUser } = useContext(AuthContext);  
  const [error, setError] = useState("error");  
  const location = useLocation();  
  const navigate = useNavigate();
```

```
const from = location.state?.from?.pathname || "/login";
```

```
const handleSignup = (event) => {  
  event.preventDefault();  
  const form = event.target;
```

```
  const email = form.email.value;  
  const password = form.password.value;
```

```
  createUser(email,password).then((userCredential) => {  
    const user = userCredential.user;  
    alert("Registered Successfully");  
    navigate(from, {replace: true})  
  })
```

```
  .catch((error) => {  
    const errorCode = error.code;  
    const errorMessage = error.message;  
    setError(errorMessage)
```

```
  // ..  
})
```

```
}  
return (
```

```
<div className="min-h-screen bg-gray-100 py-6 flex flex-col justify-center sm:py-12">
```

```
<div className="relative py-3 sm:max-w-xl sm:mx-auto">
```

```
<div
```

```
  className="absolute inset-0 bg-gradient-to-r from-blue-300 to-blue-600 shadow-lg transform -skew-y-6 sm:skew-y-0 sm:-rotate-6 sm:rounded-3xl">
```

```
</div>
```

```
<div className="relative px-4 py-10 bg-white shadow-lg sm:rounded-3xl sm:p-20">
```

```
<div className="max-w-md mx-auto">
```

```
<div>
```

```
<h1 className="text-2xl font-semibold">Registration Form</h1>
```

```
</div>
```

```
<div className="divide-y divide-gray-200">
```

```
<form onSubmit={handleSignup} className="py-8 text-base leading-6 space-y-4 text-gray-700 sm:text-lg sm:leading-7">
```

```
  {/ * <div className="relative">
```

```
<input id="username" name="username" type="text" className="peer h-10 w-full border-b-2 border-gray-300 text-gray-900 focus:outline-none focus:borer-rose-600 placeholder="Username" />
```

```
</div> */}
```

```
<div className="relative">
```

```
<input id="email" name="email" type="text" className="peer h-10 w-full border-b-2 border-gray-300 text-gray-900 focus:outline-none focus:borer-rose-600 placeholder="Email address" />
```

```

</div>
<div className="relative">
<input id="password" name="password" type="password" className="peer h-10 w-full border-b-2
border-gray-300 text-gray-900 focus:outline-none focus:borer-rose-600" placeholder="Password" />

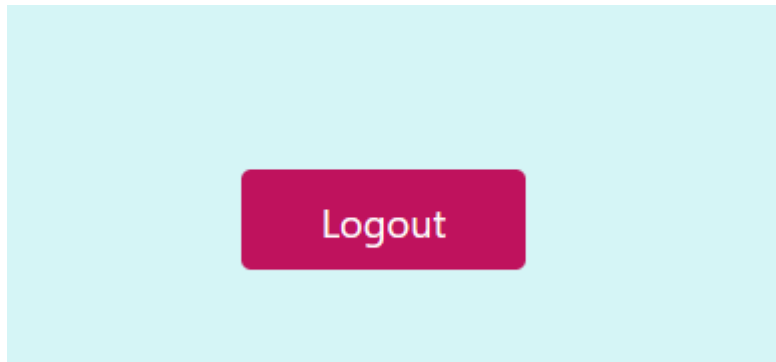
</div>
{/* <div className="relative">
<input id="username" name="username" type="username" className="peer h-10 w-full border-b-2
border-gray-300 text-gray-900 focus:outline-none focus:borer-rose-600" placeholder="Username" />

</div> */}
<p> If you have an account. Please <Link to="/login" className='text-blue-600
underline'>Login</Link>Here</p>
<div className="relative">
<button className="bg-blue-500 text-white rounded-md px-6x py-2">Register</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
)
}

```

export default Signup

## LOGOUT CODE:



```

import React, { useContext } from 'react'
import { AuthContext } from '../contexts/AuthProvider'
import { useLocation, useNavigate } from 'react-router-dom';

const Logout = () => {
const { logOut } = useContext(AuthContext);
const location = useLocation();
const navigate = useNavigate();
const from = location.state?.from?.pathname || "/";
const handleLogout = () => {
logOut().then(() => {
alert('Sign-out successfully!!');
navigate(from, {replace: true})

```

```

}).catch((error) => {

})

}
return (
<div className='h-screen bg-teal-100 flex items-center justify-center'>
<button className='bg-pink-700 px-8 py-2 text-white rounded' onClick={handleLogout}
>Logout</button>
</div>
)
}export default Logout

```

## Upload Books:

Dashboard

Upload Book

ManageBooks

Products

→ Sign In

Log out

Help

### Upload A Book!

Book Title

Book Name

Author Name

Author Name

Book Image URL

Image URL

Book Category

Fiction

Book Description

Book Description

Book PDF Link

Paste Book PDF URL here

Upload book

```

import React, { useState } from 'react'
import { Button, Checkbox, Label, Select, TextInput, Textarea } from 'flowbite-react';

```

```

const UploadBook = () => {
const bookCategories = [
"Fiction",
"Non-fiction",
"Mystery",
"Programming",
"Science fiction",
"Fantasy",
"Horror",
"Autobiography",
"History",
"Self-help",
"Business",
"Poetry",
"Children's books",
"Travel",
"Science",
"Art and design",
"Darma",
];

```



```

const [selectedBookCategory, setSelectedBookCategory] = useState(
bookCategories[0]
);

const handleChangeSelectedValue = (event) => {
console.log(event.target.value);
setSelectedBookCategory(event.target.value);
};
// handle book submission

const handleSubmit = (event) => {
event.preventDefault();
const form = event.target;

const bookTitle = form.bookTitle.value;
const authorName = form.authorName.value;
const imageURL = form.imageURL.value;
const category = form.categoryName.value;
const bookDescription = form.bookDescription.value;
const bookPDFURL = form.bookPDFURL.value;

const bookObj = {
bookTitle,
authorName,
imageURL,
category,
bookDescription,
bookPDFURL,
};
// console.log(dataObj)
// send data to database
fetch("http://localhost:5000/upload-book", {
method: "POST",

headers: {
"Content-type": "application/json",
},

body: JSON.stringify(bookObj),
})
.then((res) => res.json())
.then((data) => {
// console.log(data);
alert("Book updated successfully!!!!");
form.reset();
});
});

return (
<div className='px-4 my-12'>
<h2 className='mb-8 text-3xl font-bold'>Upload A Book!</h2>
<form className="flex lg:w-[1180px] flex-col flex-wrap gap-4" onSubmit={handleSubmit}>

{/* first row */}
<div className='flex gap-8'>

```

```
{/* book name */}
<div className='lg:w-1/2'>
  <div className="mb-2 block">
    <Label
      htmlFor="bookTitle"
      value="Book Title"
    />
  </div>
  <TextInput
    id="bookTitle"
    placeholder="Book Name"
    required
    type="text"
    name='bookTitle'
    className='w-full'
  />
</div>
```

```
{/* author name */}
<div className='lg:w-1/2'>
  <div className="mb-2 block">
    <Label
      htmlFor="authorName"
      value="Author Name"
    />
  </div>
  <TextInput
    id="authorName"
    placeholder="Author Name"
    required
    type="text"
    name='authorName'
    className='w-full'
  />
</div>
```

```
</div>
```

```
{/* 2nd Row */}
<div className='flex gap-8'>
  {/* book url */}
  <div className='lg:w-1/2'>
    <div className="mb-2 block">
      <Label
        htmlFor="imageUrl"
        value="Book Image URL"
      />
    </div>
    <TextInput
      id="imageUrl"
      placeholder="Image URL"
      required
      type="text"
      name='imageUrl'
    />
  </div>
  <div>
```

```

className='w-full'
/>
</div>

{/* book category */}
<div className='lg:w-1/2'>
<div className="mb-2 block">
<Label
htmlFor="inputState"
value="Book Category"
/>
</div>
<Select
id="inputState"
name="categoryName"
className="w-full rounded"
value={selectedBookCategory}
onChange={handleChangeSelectedValue}
>
{bookCategories.map((option) => (
<option key={option} value={option}>
{option}
</option>
))}
</Select>
</div>

</div>

{/* full width div for book description */}
<div>
<div className="mb-2 block">
<Label
htmlFor="bookDescription"
value="Book Description"
/>
</div>
<Textarea
id="bookDescription"
placeholder="Book Description"
required
type="text"
name='bookDescription'
className='w-full'
rows={4}
/>
</div>

{/* book pdf url */}
<div>
<div className="mb-2 block">
<Label
htmlFor="bookPDFURL"
value="Book PDF Link"
/>

```

```

</div>
<TextInput
id="bookPDFURL"
placeholder="Paste Book PDF URL here"
required
type="text"
name='bookPDFURL'
className='w-full'
/>
</div>

{/* Submit btn */}
<Button type="submit" className='mt-5'>
Upload book
</Button>

</form>
</div>
)
}

```

```
export default UploadBook
```

## ManageBooks:

Manager Your Books Inventory!						
NO.	BOOK NAME	AUTHOR NAME	CATEGORY	PRICE	EDIT OR MANAGE	
1	Check & Mate	Ali Hazelwood 2	Fantasy	\$10.99	<a href="#">Edit</a>	<a href="#">Delete</a>
2	A Curse for True Love	Stephanie Garber	Fiction	\$10.99	<a href="#">Edit</a>	<a href="#">Delete</a>
3	Happy Place	Emily HenryGoodreads	Romance	\$10.99	<a href="#">Edit</a>	<a href="#">Delete</a>
4	Meet Me at the Lake	Carley Fortune	Romance	\$10.99	<a href="#">Edit</a>	<a href="#">Delete</a>
5	The Seven Year Slip	Ashley Poston	Romance	\$10.99	<a href="#">Edit</a>	<a href="#">Delete</a>
6	The Covenant of Water12	Abraham Verghese	History	\$10.99	<a href="#">Edit</a>	<a href="#">Delete</a>
7	Light Bringer	Pierce Brown	Science Fiction	\$10.99	<a href="#">Edit</a>	<a href="#">Delete</a>
8	None of This Is True	Lisa Jewel	Horror	\$10.99	<a href="#">Edit</a>	<a href="#">Delete</a>

```

import { Table } from 'flowbite-react'
import React, { useEffect, useState } from 'react'
import { Pagination } from 'flowbite-react';
import { Link } from 'react-router-dom';

```

```

const ManageBooks = () => {
const [allBooks, setAllBooks] = useState([]);
useEffect(() => {
fetch(`http://localhost:5000/all-books`)
.then((res) => res.json())
.then((data) => {
// console.log(data);

```

```

setAllBooks(data);
});
}, []);

// delete a books
const handleDelete = (id) => {
  console.log(id);
  fetch(`http://localhost:5000/book/${id}`, {
    method: "DELETE",
  })
  .then(res => res.json())
  .then(data => {
    alert("Book is deleted Successfully!")
    // console.log(data);
    // setAllBooks(data);
  });
};

// pagination
const [currentPage, setCurrentPage] = useState(1);
const onPageChange = () => setCurrentPage(page);

return (
  <div className='px-4 my-12'>
    <h2 className='mb-8 text-3xl font-bold'>Manager Your Books Inventory!</h2>

    { /* table */

    <Table className='lg:w-[1180px]'>
      <Table.Head>
        <Table.HeadCell>
          No.
        </Table.HeadCell>
        <Table.HeadCell>
          Book name
        </Table.HeadCell>
        <Table.HeadCell>
          Author Name
        </Table.HeadCell>
        <Table.HeadCell>
          Category
        </Table.HeadCell>
        <Table.HeadCell>
          Price
        </Table.HeadCell>
        <Table.HeadCell>
          Edit or Manage
        </Table.HeadCell>
      </Table.Head>

      {
        allBooks.map((book, index) => <Table.Body className="divide-y" key={book._id}>
          <Table.Row className="bg-white dark:border-gray-700 dark:bg-gray-800">
            <Table.Cell className="whitespace-nowrap font-medium text-gray-900 dark:text-white">
              {index + 1}

```

```

</Table.Cell>
<Table.Cell className="whitespace-nowrap font-medium text-gray-900 dark:text-white">
{book.bookTitle}
</Table.Cell>
<Table.Cell>
{book.authorName}
</Table.Cell>
<Table.Cell>
{book.category}
</Table.Cell>
<Table.Cell>
$10.99

</Table.Cell>

<Table.Cell>
<Link
className="font-medium text-cyan-600 hover:underline dark:text-cyan-500 mr-5"
to={` /admin/dashboard/edit-books/${book._id}`}`
>
Edit
</Link>
<button className="bg-red-600 px-4 py-1 font-semibold text-white rounded-sm hover:bg-sky-600"
onClick={() => handleDelete(book._id)}>Delete</button>

</Table.Cell>
</Table.Row>
</Table.Body>)
}
</Table>

{/* pagination */}
<div className="flex items-center justify-center text-center mt-8">
<Pagination
currentPage={1}
layout="pagination"
nextLabel="Go forward"
onPageChange={page => { setCurrentPage(page) }}
previousLabel="Go back"
showIcons
totalPages={10}
/>
</div>
</div>
)
}
export default ManageBooks

```

## Edit Books:

### Update the Book Data!

Book Title	Author Name
<input type="text" value="Check &amp; Mate"/>	<input type="text" value="Ali Hazelwood 2"/>
Book Image URL	Book Category
<input type="text" value="https://i.gr-assets.com/images/S/compressed.photo.goodreads.com/books/1678378842/"/>	<input type="text" value="Fiction"/>
Book Description	
<div>Author Ali Hazelwood made her name in the book business by writing smart love stories for discerning adult readers. So, it's an encouraging development that her first venture into the young adult aisles is proving equally popular. Check &amp; Mate follows reluctant chess genius Mallory Greenleaf as she deploys gambits and strategies in the game of life. Smart kids need love too, you know.young adult aisles is proving equally popular. Check &amp; Mate follows reluctant chess genius Mallory Greenleaf as she deploys gambits and strategies in the game of life. Smart kids need love too, you know.</div>	
<input type="button" value="Update book"/>	

```
import React, { useState } from 'react'
import { Button, Checkbox, Label, Select, TextInput, Textarea } from 'flowbite-react';
import { useLoaderData, useParams } from 'react-router-dom';

const EditBooks = () => {
  const { id } = useParams();
  const { bookTitle, authorName, imageURL, category, bookDescription, bookPDFURL } =
    useLoaderData();
  // console.log(bookTitle)

  const bookCategories = [
    "Fiction",
    "Non-fiction",
    "Mystery",
    "Programming",
    "Science fiction",
    "Fantasy",
    "Horror",
    "Autobiography",
    "History",
    "Self-help",
    "Business",
    "Poetry",
    "Children's books",
    "Travel",
    "Science",
    "Art and design",
    "Darma",
  ];

  const [selectedBookCategory, setSelectedBookCategory] = useState(
    bookCategories[0]
```

```

);

const handleChangeSelectedValue = (event) => {
  console.log(event.target.value);
  setSelectedBookCategory(event.target.value);
};

const handleUpdate = (event) => {
  event.preventDefault();
  const form = event.target;

  const bookTitle = form.bookTitle.value;
  const authorName = form.authorName.value;
  const imageURL = form.imageURL.value;
  const category = form.categoryName.value;
  const bookDescription = form.bookDescription.value;
  // const bookPDFURL = form.bookPDFURL.value;

  const bookObj = {
    bookTitle,
    authorName,
    imageURL,
    category,
    bookDescription,
    // bookPDFURL,
  };
  // console.log(bookObj)

  // update the book object
  fetch(`http://localhost:5000/book/${id}`, {
    method: "PATCH",

    headers: {
      "Content-type": "application/json",
    },

    body: JSON.stringify(bookObj),
  }).then((res) => res.json())
  .then((data) => {
    // console.log(data);
    alert("Book is updated successfully!!!!");
    form.reset();
  });
};

return (
  <div className='px-4 my-12'>
    <h2 className='mb-8 text-3xl font-bold'>Update the Book Data!</h2>
    <form className="flex lg:w-[1180px] flex-col flex-wrap gap-4" onSubmit={handleUpdate}>

      {/* first row */}
      <div className='flex gap-8'>

        {/* book name */}
        <div className='lg:w-1/2'>

```



```
<div className="mb-2 block">
  <Label
    htmlFor="bookTitle"
    value="Book Title"
  />
</div>
<TextInput
  id="bookTitle"
  placeholder="Book Name"
  required
  type="text"
  name='bookTitle'
  className='w-full'
  defaultValue={ bookTitle }
/>
</div>
```

```
{/* author name */}
<div className='lg:w-1/2'>
  <div className="mb-2 block">
    <Label
      htmlFor="authorName"
      value="Author Name"
    />
  </div>
  <TextInput
    id="authorName"
    placeholder="Author Name"
    required
    type="text"
    name='authorName'
    className='w-full'
    defaultValue={ authorName }
  />
</div>
```

```
</div>
```

```
{/* 2nd Row */}
<div className='flex gap-8'>
  {/* book url */}
  <div className='lg:w-1/2'>
    <div className="mb-2 block">
      <Label
        htmlFor="imageUrl"
        value="Book Image URL"
      />
    </div>
    <TextInput
      id="imageUrl"
      placeholder="Image URL"
      required
      type="text"
      name='imageUrl'
      className='w-full'
    />
  </div>
</div>
```

```

defaultValue={ imageURL }
/>
</div>

{/* book category */}
<div className='lg:w-1/2'>
<div className="mb-2 block">
<Label
htmlFor="inputState"
value="Book Category"
/>
</div>
<Select
id="inputState"
name="categoryName"
className="w-full rounded"
value={ selectedBookCategory }
onChange={ handleChangeSelectedValue }
>
{bookCategories.map((option) => (
<option key={ option } value={ option }>
{ option }
</option>
))}
</Select>
</div>

</div>

{/* full width div for book description */}
<div>
<div className="mb-2 block">
<Label
htmlFor="bookDescription"
value="Book Description"
/>
</div>
<Textarea
id="bookDescription"
placeholder="Book Description"
required
type="text"
name='bookDescription'
className='w-full'
rows={ 4 }
defaultValue={ bookDescription }
/>
</div>

{/* book pdf url
<div>
<div className="mb-2 block">
<Label
htmlFor="bookPDFURL"
value="Book PDF Link"

```

```

/>
</div>
<TextInput
id="bookPDFURL"
placeholder="Paste Book PDF URL here"
required
type="text"
name='bookPDFURL'
className='w-full'
defaultValue={ bookPDFURL }
/>
</div> */}

{/* Submit btn */}
<Button type="submit" className='mt-5'>
Update book
</Button>

</form>
</div>
)
}

```

```
export default EditBooks
```

## BACKENDCODE:

### Index.js (Connection code)

```

const express = require('express')
const app = express()
const port = process.env.PORT || 5000
const cors = require('cors');

//middleware
app.use(cors());
app.use(express.json());

app.get('/', (req, res) => {
res.send('Hello World!')
})

//mongodb configuration

const { MongoClient, ServerApiVersion, ObjectId } = require('mongodb');
//
// const uri = "mongodb+srv://book-store-
app:archana123@cluster0.o4ivwuk.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0
";
const uri = "mongodb+srv://book-store-
app:archana123@cluster0.o4ivwuk.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0
"

```

```

// Create a MongoClient with a MongoClientOptions object to set the Stable API version
const client = new MongoClient(uri, {
  serverApi: {
    version: ServerApiVersion.v1,
    strict: true,
    deprecationErrors: true,
  }
});

async function run() {
  try {
    // Connect the client to the server (optional starting in v4.7)
    await client.connect();

    //create a collection of documents
    const bookCollections = client.db("BookInventory").collection("books");

    // insert a book to the db : post method
    app.post("/upload-book", async(req, res) => {
      const data = req.body;
      const result = await bookCollections.insertOne(data);
      res.send(result);
    })

    // get all books from database
    // app.get("/all-books" , async(req, res) =>{
    //   const books = bookCollections.find();
    //   const result = await books.toArray();
    //   res.send(result);
    // })

    // update a book data : patch or update methods
    app.patch("/book/:id", async(req, res) =>{
      const id = req.params.id;
      // console.log(id);
      const updateBookData = req.body;
      const filter = { _id: new ObjectId(id) };
      const options = { upsert: true };

      const updateDoc = {
        $set: {
          ...updateBookData
        }
      }

      // update
      const result = await bookCollections.updateOne(filter, updateDoc, options );
      res.send(result);
    })

    // delete a book data
    app.delete("/book/:id", async(req, res) => {

```

```

const id = req.params.id;
const filter = {_id: new ObjectId(id)};
const result = await bookCollections.deleteOne(filter);
res.send(result);

})

//find by category
app.get("/all-books", async(req, res)=>{
let query = {};
if(req.query?.category){
query = {category: req.query.category}

}
const result = await bookCollections.find(query).toArray();
res.send(result);
})

// to get single book data
app.get("/book/:id", async(req, res) => {
const id = req.params.id;
const filter = {_id: new ObjectId(id)};
const result = await bookCollections.findOne(filter);
res.send(result);
})

// Send a ping to confirm a successful connection
await client.db("admin").command({ ping: 1 });
console.log("Pinged your deployment. You successfully connected to MongoDB!");
} finally {
// Ensures that the client will close when you finish/error
// await client.close();
}
}
run().catch(console.dir);

app.listen(port, () => {
console.log(`Example app listening on port ${port}`)
})

```

