

DEVSHELF SLEEPYBOIS

Library-Management

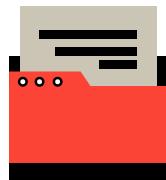
Learning

Our Journey

Features

Applications

Future Plans



Learning

The Mission

Focus on learning tools and Software for Web Development to contribute to real world solutions

The Vision

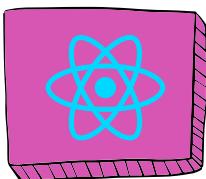
Distribute work among teammates for various fields, gaining knowledge in it, and work together as a whole to bring the project to life

What we've accomplished so far

Working on this project, we gained vast knowledge of following applications such as:

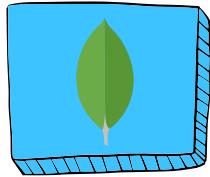
- React:
- Tailwind:
- Node JS & Express:
- MongoDB:

TECH STACK



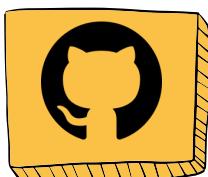
01. REACT

For Frontend:
Tailwind, Bootstrap
framework and
material-UI
components



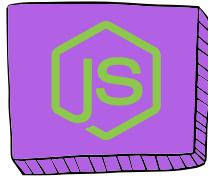
02. MongoDB

For DataBase:
Storing books in
library and
user information



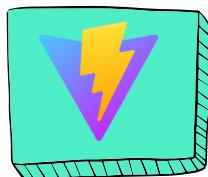
03. GitHub

For Updating Project
resources regularly
and Collaborating



04. Express & Node JS

For Backend:
Adding functionality
to Frontend and
connecting it with
the DataBase



05. VITE

For quickly
initializing REACT
websites and
updating resources

Our Journey

The origin of the river

Our journey began unexpectedly, driven by curiosity and a shared passion for technology. Dividing responsibilities amongst ourselves, Varshith, Yash took charge of frontend development, Prajwal handled backend tasks, while Suhas managed the database. We embarked on this journey by exploring resources such as YouTube, various websites, and even sought guidance from ChatGPT.

Shaping

Guided by our mentor, whose insights were invaluable, we delved into numerous project videos to deepen our understanding. Integrating diverse elements into a cohesive project emerged as a pivotal challenge. It was during this process that we realized the significance of writing immaculate code adorned with explanatory comments—a practice that accelerated comprehension and troubleshooting, contrasting sharply with the time-consuming deciphering of others' code.

Merging with the sea

Through perseverance and resilience, navigating through highs and lows, we triumphed in bringing our vision to life—a testament to our dedication and collaborative spirit.

Features

- Dark Mode:

The screenshot shows the homepage of the IIT DHARWAD website. At the top, there is a navigation bar with links for Home, All Books, Branch, a search bar, and user icons. The main content area features a large image of a modern library building with a blue and white facade and a glass entrance. Below the image, the text "Knowledge Resource and Data Center" is displayed in a large, bold, blue font. To the left of the text, there is a brief description of the library's collection and its mission to build up the library collection up to Ph.D. level. The overall theme of the page is dark, with white text on a black background.

The screenshot shows a code editor window with a dark theme. The file is named "Navbar.jsx". The code implements a useEffect hook to change the theme based on a stored preference. It adds or removes a 'dark' class from the body element and stores the theme in local storage. The code is as follows:

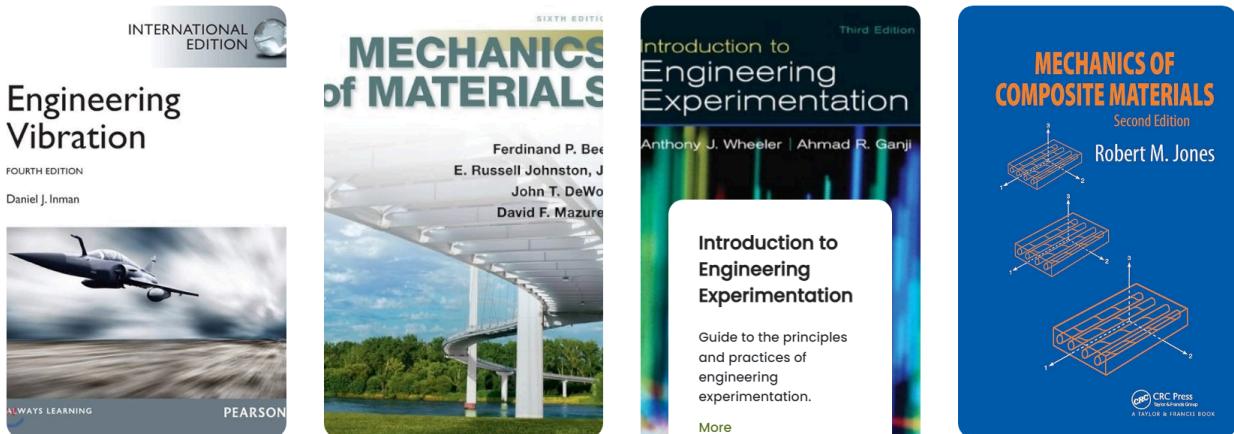
```
useEffect(() => {
  if (theme === 'dark') {
    element.classList.add('dark');
    localStorage.setItem('theme', 'dark');
    document.body.classList.add('dark');
  } else {
    element.classList.remove('dark');
    localStorage.setItem('theme', 'light');
    document.body.classList.remove('dark');
  }
}, [theme]);
```

snappify.com

We have incorporated a **Dark Mode** feature, allowing users to effortlessly switch between themes according to their preference or mood. This ensures a customizable and comfortable browsing experience at any time.

- Recommended Books:

Recommended Books



```
Recommended

const [book, setBook] = useState([])
useEffect(() => {
  const getBook = async() => {
    try {
      const res = await axios.get("http://localhost:4001/book");
      const data = res.data.filter((data) => data.department === me);
      setBook(data);
    } catch (error) {
      console.log("error:", error);
    }
  };
  getBook();
}, []);


```

```
Recommended

let me = "Computer Science"
if(authUser){
  const user = JSON.parse(localStorage.getItem("Users"));
  const email = user.email;
  const branch = email.slice(0, 2);
  if (branch === "cs" || branch === "CS") {
    me = "Computer Science";
  } else if (branch === "EE" || branch === "ee") {
    me = "Electrical Engineering";
  } else if (branch === "me" || branch === "ME") {
    me = "Mechanical Engineering";
  } else if (branch === "ce" || branch === "CE") {
    me = "Civil Engineering";
  } else if (branch === "ch" || branch === "Ch") {
    me = "Chemical Engineering";
  } else if (branch === "EP" || branch === "ep") {
    me = "Engineering Physics";
  } else {
    me = "Computer Science";
  }
}


```

The Home page shows **Recommended books** based on user's Branch.

By default it shows Computer Science books, but when logged in, it takes first two letters from user's email and uses it to show the respected department books

- Login and Sign-Up:

IT DHARWAD

Home All Books Branch Search Login

Knowledge Resource and Data Center

The collection consists of nearly 4750+ books. These are arranged subject wise. It has also adopted international standards (International Standards Organization). At present, the main motto is to build up the library so that it has information resources up to Ph.D. level and advanced texts and reference materials.

Recommended Books

```
const {
  register,
  handleSubmit,
  useForm,
  useState,
} = useForm();

const onSubmit = async (data) => {
  const userInfo = {
    email: data.email,
    password: data.password,
  };
  await axios
    .post('http://localhost:4001/user/login', userInfo)
    .then((res) => {
      console.log(res.data);
      if (res.data) {
        toast.success('Login Successful');
        document.getItemById('my_modal_3').close();
      }
    })
    .catch((error) => {
      if (error.response) {
        console.log(error);
        toast.error('Error: ' + error.response.data.message);
        setTimeout(() => {}, 1000);
      }
    });
};

setTimeout(() => {
  window.location.reload();
  localStorage.setItem('Users', JSON.stringify(res.data.user));
}, 800);
};
```

snapppy.com

Clicking on the **Login** button opens a box modal where user provides credentials to log into their account

On submitting, if the user exists in database then these credentials are stored in **LocalStorage** for easy access

Sign-Up
These Credentials will be used for Login

Email ID:

Password:

SignUp

```
const {
  register,
  handleSubmit,
  useForm,
  useState,
} = useForm();
const navigate = useNavigate();

const onSubmit = async (data) => {
  const userInfo = {
    email: data.email,
    password: data.password,
  };
  await axios
    .post('http://localhost:4001/user/signup', userInfo)
    .then((res) => {
      console.log(res.data);
      if (res.data) {
        toast.success('Password set Successfully');
        localStorage.setItem('Users', JSON.stringify(res.data.user));
        setTimeout(() => {
          navigate('/');
        }, 500);
      }
    })
    .catch((error) => {
      if (error.response) {
        console.log(error);
        toast.error('Error: ' + error.response.data.message);
      }
    });
};

setTimeout(() => {
  window.location.reload();
  localStorage.setItem('Users', JSON.stringify(res.data.user));
}, 800);
};
```

snapppy.com

When clicked on the **Sign-Up** button, the page redirects to "/signup" route where User inputs his Email id and sets a new Password

By default, Email ids are stored in database with Password field kept empty. Signing up stored new updated Password in Database in hashed format to 8 rounds(using BcryptJs), hence enhancing security

- **User Information Page:**

```


<Navbar />
<div className="flex pt-16 flex-col lg:flex-row p-5 lg:space-x-8 lg:pt-12">
  <div className="flex flex-col w-full lg:w-1/2 mb-8 lg:mb-0">
    {user && <UserProfile profilePic={picture} username={user.username} email={user.email} />}
  </div>
  <div className="w-full">
    <BookList title="Books Borrowed" books={books} onIssueDateChange={handleIssueDateChange} />
  </div>
</div>
</div>;


```

Welcome to your **Personalized Profile** dashboard. Here, you can seamlessly view and manage your account details. Your profile section prominently displays your profile picture, username, email address, and Books Borrowed.

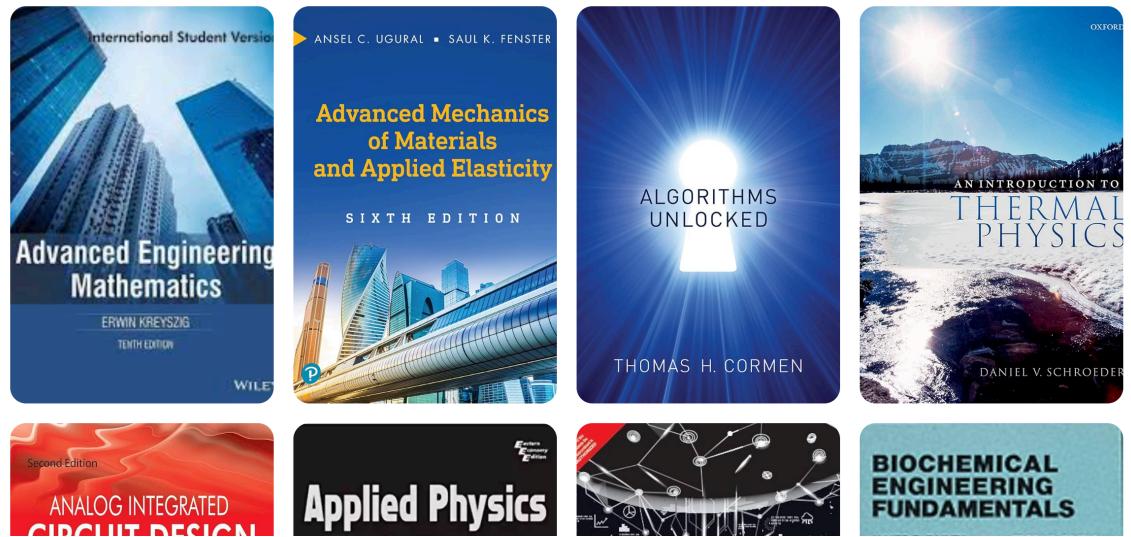
Whether you prefer compact screens or expansive displays, we've got you covered.

- **All Books Page:**

All Books

Sort by: Alphabetical Ascending

Books per page: 40

Department:
All DepartmentsGenre:
All Genres

AllBooks.jsx

```
<div className="book-grid-container">
  {sortedBooks.slice((currentPage - 1) * booksPerPage, currentPage * booksPerPage).map((item) => (
    <Bookcard item={item} key={item._id} />
  )));
</div>;
```

snappify.com



AllBooks.jsx

```
<label>
  Sort by:
  <select
    value={sortOrder}
    onChange={(e) => setSortOrder(e.target.value)}
    className='bg-[#ffffff] dark:bg-gray-800 text-gray-800 dark:text-white border rounded-md px-3 py-2 focus:outline-none focus:border-blue-500'>
    <option value="count-asc">Count Ascending</option>
    <option value="count-desc">Count Descending</option>
    <option value="alpha-asc">Alphabetical Ascending</option>
    <option value="alpha-desc">Alphabetical Descending</option>
  </select>
</label>;
```

snappify.com



AllBooks.jsx

```
<label>
  Department:
  <select
    className="dark:bg-gray-800"
    value={selectedDepartment}
    onChange={(e) => setSelectedDepartment(e.target.value)}
  >
    <option value="">All Departments</option>
    {uniqueDepartments.map((department) => (
      <option key={department._id} value={department}>
        {department}
      </option>
    )))
  </select>
</label>;
```

snappify.com

Displaying books with the ability to:

- **Sort** - by count (available books) and Alphabetical order
- **Filter** - by Branch and Genre
- **Search**- by title and Author
- **Book Limit**- choose the Number of books to be displayed per page

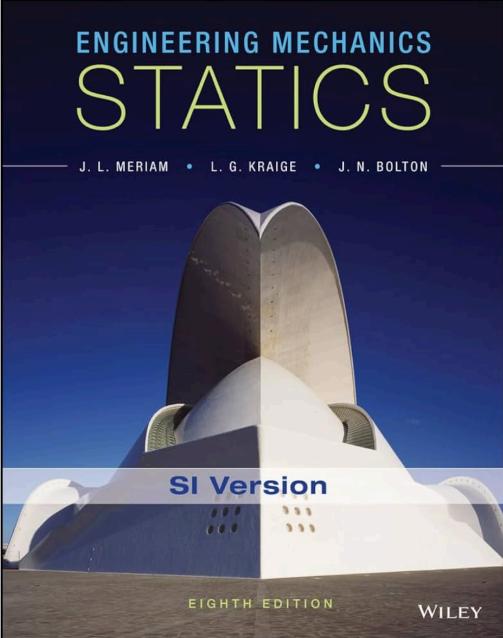
• Book Detail Page:

IIT DHARWAD

Home All Books Branch ▾

Search





**Engineering Mechanics:
Statics**

Author: J. L. Meriam, L. G. Kraige
Genre: Statics
Comprehensive guide to the principles of statics in engineering mechanics.

Borrow

Return

I'm Interested

```
useEffect(() => {
  const getBook = async () => {
    try {
      console.log('Fetching book with ID:', _id);
      const response = await axios.get(`http://localhost:4001/book/${_id}`);
      console.log('Book data:', response.data);
      setBook(response.data);
    } catch (error) {
      console.error('Error fetching book:', error);
      setError('Failed to fetch book data. Please try again later.');
    } finally {
      setLoading(false);
    }
  };

  getBook();
}, [_id]);
```

snappify.com

The **Book Detail** View provides a comprehensive overview of each book, making it easier for users to engage with the content and make informed decisions about their reading choices

Borrow

The **handleBorrow** function takes the user from localstorage, then updates the database of the user with the book id and date of borrowing, also changes the count of the book.

```
const handleBorrow = async () => {
  try {
    const user = JSON.parse(localStorage.getItem('Users'));
    const userIdRaw = user._id;
    const userInfo = { userId: userIdRaw };
    console.log('this is id data:', userInfo);
    await axios
      .post(`http://localhost:4001/user/borrow/${_.id}`, userInfo)
      .then((res) => {
        console.log(res.data);
        if (res.data) {
          toast.success('Book borrowed Successfully');
        }
      })
      .catch((error) => {
        if (error.response) {
          console.log(error);
          toast.error('Error: ' + error.response.data.message);
        }
      });
  } catch (error) {
    console.error('Error borrowing book:', error);
    toast.error('Failed to borrow book. Please try again later.');
  }
};
```

snappyf.com

Return

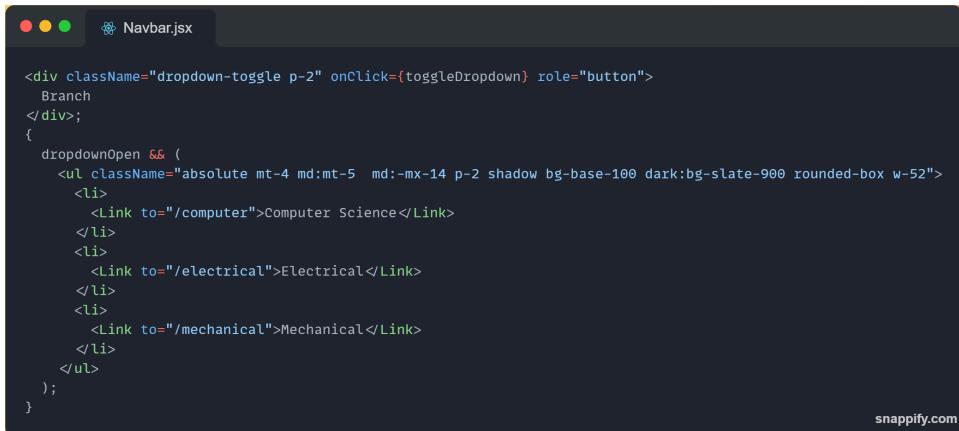
The **handleReturn** function returns the book if the book is already borrowed and updates the database with the new count in books database, and blank data in the respective book data in the users database

```
const handleReturn = async () => {
  try {
    const user = JSON.parse(localStorage.getItem('Users'));
    const userIdRaw = user._id;
    const userInfo = { userId: userIdRaw };
    console.log('this is id data:', userInfo);
    await axios
      .post(`http://localhost:4001/user/returnBook/${_.id}`, userInfo)
      .then((res) => {
        console.log(res.data);
        if (res.data) {
          toast.success('Book returned Successfully');
        }
      })
      .catch((error) => {
        if (error.response) {
          console.log(error);
          toast.error('Error: ' + error.response.data.message);
        }
      });
  } catch (error) {
    console.error('Error returning book:', error);
    toast.error('Failed to return book. Please try again later.');
  }
};
```

snappyf.com

• Branch Drop-Down

A screenshot of a web browser window displaying a library management system. The URL bar shows 'localhost:5173/computer'. The page has a header with 'IIT DHARWAD' and navigation links for 'Home', 'All Books', 'Branch', and a search bar. A dropdown menu under 'Branch' lists 'Computer Science', 'Electrical', and 'Mechanical'. Below the header, there are several book cards. One card for 'ARTIFICIAL INTELLIGENCE' by Stuart Russell and Peter Norvig is visible. Another card for 'ALGORITHMS UNLOCKED' by Thomas H. Cormen is shown with a bright blue light effect. A third card for 'Introduction to the Theory of COMPUTATION' by Michael Sipser is partially visible. To the right, there's a section for 'THE CLASSIC WORK' featuring 'The Art of Computer Programming' by Donald E. Knuth, Volume 3, Second Edition. At the bottom, other book cards for 'Design Patterns' and 'Database System Concepts' are visible.



```


Branch

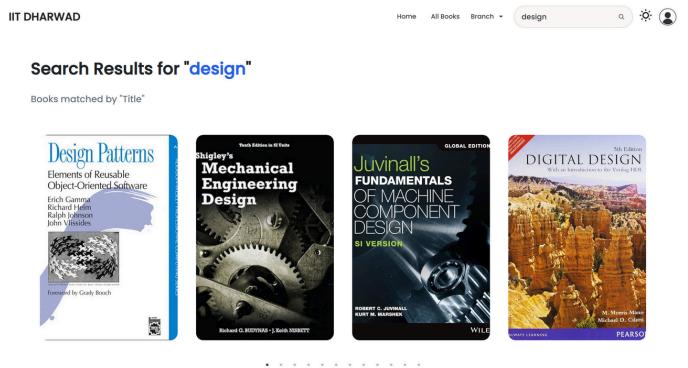

{
  dropdownOpen && (
    <ul className="absolute mt-4 md:mt-5 md:-mx-14 p-2 shadow bg-base-100 dark:bg-slate-900 rounded-box w-52">
      <li>
        <Link to="/computer">Computer Science</Link>
      </li>
      <li>
        <Link to="/electrical">Electrical</Link>
      </li>
      <li>
        <Link to="/mechanical">Mechanical</Link>
      </li>
    </ul>
  );
}

```

snappyf.com

For user-friendly navigation, we've integrated a branch-specific dropdown directly into the header. This feature allows users to effortlessly browse books sorted by their desired branch.

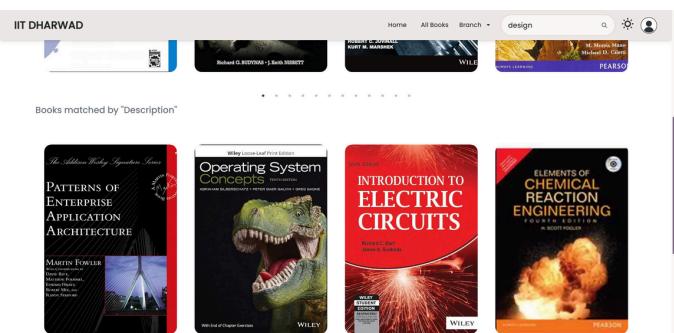
• Search Feature:



Search bar is functional where adding a query in it redirect to the **Search results page**

This page has search results shown in following order:

1. Title
2. Description
3. Author



```

const { keyword } = useParams();
const [books, setBooks] = useState([]);
const [loading, setLoading] = useState(true);
const [error, setError] = useState(null);

useEffect(() => {
  const fetchSearchResults = async () => {
    try {
      const response = await axios.get(`http://localhost:4001/book/search/${keyword}`);
      setBooks(response.data);
      setError(null);
      console.error('Error fetching search results. Please try again later.');
    } catch (error) {
      setError('Error fetching search results. Please try again later.');
      console.error('Error fetching search results:', error);
    } finally {
      setLoading(false);
    }
  };

  if (keyword) {
    fetchSearchResults();
  } else {
    setLoading(false);
  }
}, [keyword]);

```

snappyf.com

Admin controls

We have integrated robust administrative controls that encompass the following functionalities:

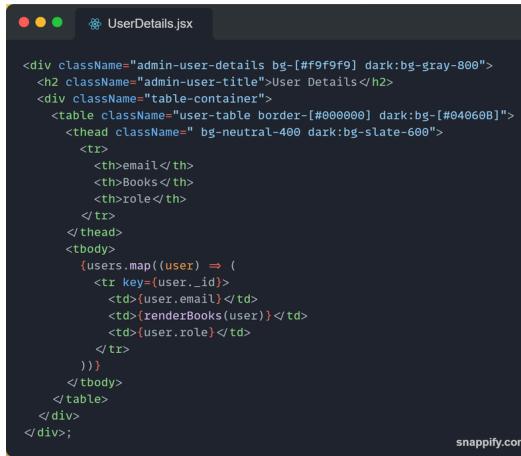
email	Books	role
cs23bt075@iitdh.ac.in	book1: 66921l9ec427627cd02dd2f0 book1Date: Sat Jul 13 2024	user
prajwalkoppad30@gmail.com		user
123@gmail.com	book1: 66921l9ec427627cd02dd2df book1Date: Mon Jul 15 2024 book2: 66921l9ec427627cd02dd2de book2Date: Mon Jul 15 2024	user
me23bt009@iitdh.ac.in	book1: 66921l9ec427627cd02dd2ee book1Date: Mon Jul 15 2024 book2: 66921l9ec427627cd02dd2f9 book2Date: Mon Jul 15 2024	user
EE23BT035@iitdh.ac.in	book1: 66921l9ec427627cd02dd2fa book1Date: Mon Jul 15 2024 book2: 66921l9ec427627cd02dd2fa book2Date: Mon Jul 15 2024 18:32:28 GMT+0530 (India Standard Time) book3: 66921l9ec427627cd02dd2fa book3Date: Mon Jul 15 2024 18:51:47 GMT+0530 (India Standard Time)	user
kamatham.varshith@gmail.com		admin



Admin.jsx

```
<aside className="sidebar bg-[#482958] dark:bg-[#2B1834]">
  <div className="sidebar-header">Admin Dashboard</div>
  <nav className="sidebar-nav">
    <ul>
      <li className={activeSection === 'userDetails' ? 'active' : ''} onClick={() => setActiveSection('userDetails')}>
        User Details
      </li>
      <li className={activeSection === 'addBook' ? 'active' : ''} onClick={() => setActiveSection('addBook')}>
        Add Books
      </li>
      <li className={activeSection === 'bookList' ? 'active' : ''} onClick={() => setActiveSection('bookList')}>
        Book List
      </li>
    </ul>
  </nav>
</aside>;
```

snappyf.com



UserDetails.jsx

```
<div className="admin-user-details bg-[#f9f9f9] dark:bg-gray-800">
  <h2 className="admin-user-title">User Details</h2>
  <div className="table-container">
    <table className="user-table border-[#000000] dark:bg-[#04060B]">
      <thead className="bg-neutral-400 dark:bg-slate-600">
        <tr>
          <th>Email</th>
          <th>Books</th>
          <th>Role</th>
        </tr>
      </thead>
      <tbody>
        {users.map((user) => (
          <tr key={user._id}>
            <td>{user.email}</td>
            <td>{renderBooks(user)}</td>
            <td>{user.role}</td>
          </tr>
        ))}
      </tbody>
    </table>
  </div>
</div>;
```

snappyf.com

Admin page which is accessible only by clicking on the profile icon when the user is an admin. If the user is not an admin, then it will redirect to home page if brute forced the path in URL.

On the user details page, we feature a comprehensive table listing all users, presenting each user's username and email ID in a structured format, ensuring a seamless and organized viewing experience.

User Details

Add Books

Book List

Add Book

Title

Description

Author

Genre

Department

Count

Image URL

[Upload JSON/CSV](#)

```
 ① ② ③ AddBookForm.jsx

const handleAddBook = async (e) => {
  e.preventDefault();

  // Validate fields
  if (!title || !description || !author || !genre || !department || count === 0) {
    toast.error('Please fill in all fields.');
    return;
  }

  try {
    const response = await axios.post('http://localhost:4001/admin/books', {
      title,
      description,
      author,
      genre,
      department,
      count,
      image,
    });
    console.log('Book added', response.data);
    onBookAdded(response.data); // Notify parent component about the new book
    toast.success('Book added successfully!');
  } catch (error) {
    console.error('Error adding book', error);
    toast.error('Failed to add book. Please try again later.');
  }
};

// Reset form fields
setTitle('');
setDescription('');
setAuthor('');
setGenre('');
setDepartment('');
setCount(0);
setImage('');
setFileData(null); // Clear file data
setImageUrlPreview(''); // Clear image preview
};
```

snappyf.com

Admins can input detailed information, such as titles, authors, genres, and descriptions, count, image URL, ensuring each book entry is thorough and precise.

Admin Dashboard

User Details

Add Books

Book List

Book List

Title	Author	Genre	Department	Count	Actions	
Artificial Intelligence: A Modern Approach	Stuart Russell, Peter Norvig	Artificial Intelligence	Computer Science	11	Edit	Delete
Algorithms Unlocked	Thomas H. Cormen	Algorithms	Computer Science	6	Edit	Delete
Introduction to the Theory of Computation	Michael Sipser	Computational Theory	Computer Science	13	Edit	Delete
The Art of Computer Programming	Donald E. Knuth	Algorithms	Computer Science	10	Edit	Delete
Design Patterns: Elements of Reusable Object-Oriented Software	Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides	Software Engineering	Computer Science	6	Edit	Delete
Structure and Interpretation of Computer Programs	Harold Abelson, Gerald Jay Sussman	Programming	Computer Science	10	Edit	Delete
Database System Concepts	Abraham Silberschatz, Henry F. Korth, S. Sudarshan	Database Systems	Computer Science	12	Edit	Delete
Compilers: Principles, Techniques, and Tools	Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman	Compilers	Computer Science	13	Edit	Delete
Mechanical Engineering Design	J. E. Shigley, Charles R. Mischke, Richard G. Budynas	Design	Mechanical Engineering	9	Edit	Delete
Materials Science and Engineering: An Introduction	William D. Callister Jr.	Materials Science	Mechanical Engineering	10	Edit	Delete

```


## className="book-list-title">Book List

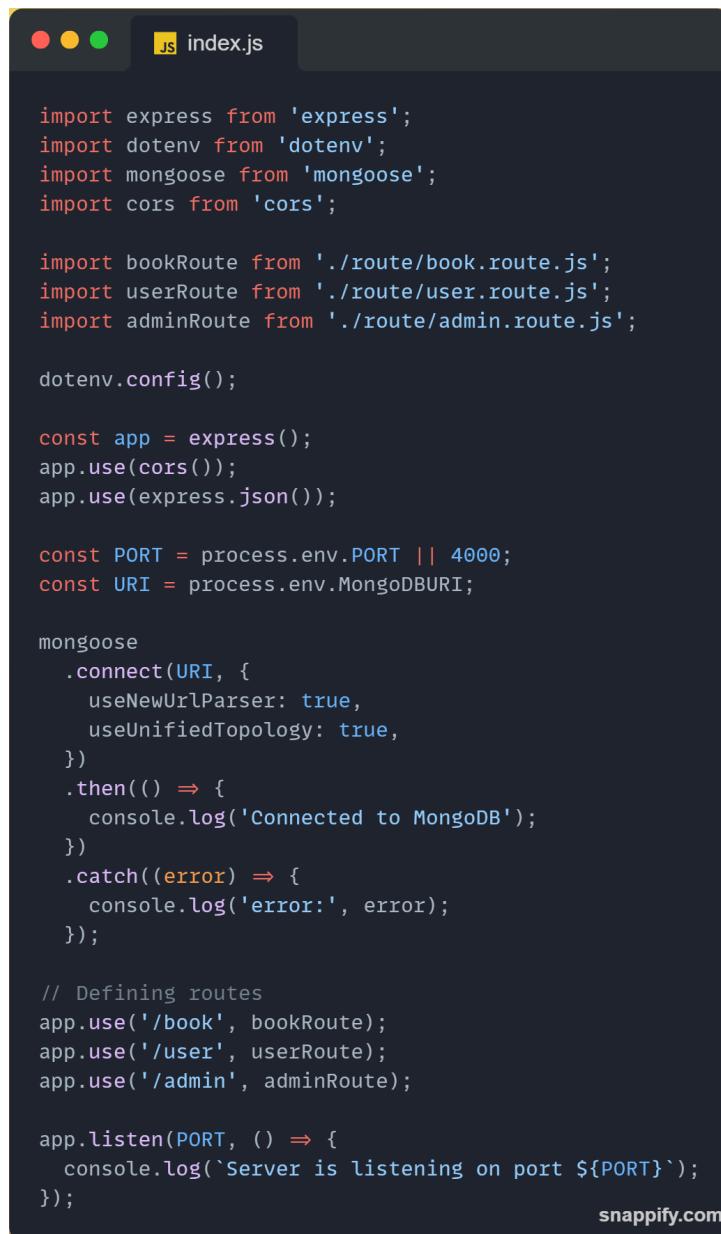


| Title | Author | Genre | Department | Count | Actions |  |
|-------|--------|-------|------------|-------|---------|--|
|-------|--------|-------|------------|-------|---------|--|


```

Made a wrong entry or Book to be removed? We got you covered

Backend



```
index.js

import express from 'express';
import dotenv from 'dotenv';
import mongoose from 'mongoose';
import cors from 'cors';

import bookRoute from './route/book.route.js';
import userRoute from './route/user.route.js';
import adminRoute from './route/admin.route.js';

dotenv.config();

const app = express();
app.use(cors());
app.use(express.json());

const PORT = process.env.PORT || 4000;
const URI = process.env.MongoDBURI;

mongoose
  .connect(URI, {
    useNewUrlParser: true,
    useUnifiedTopology: true,
  })
  .then(() => {
    console.log('Connected to MongoDB');
  })
  .catch((error) => {
    console.log('error:', error);
  });

// Defining routes
app.use('/book', bookRoute);
app.use('/user', userRoute);
app.use('/admin', adminRoute);

app.listen(PORT, () => {
  console.log(`Server is listening on port ${PORT}`);
});
```

snappify.com

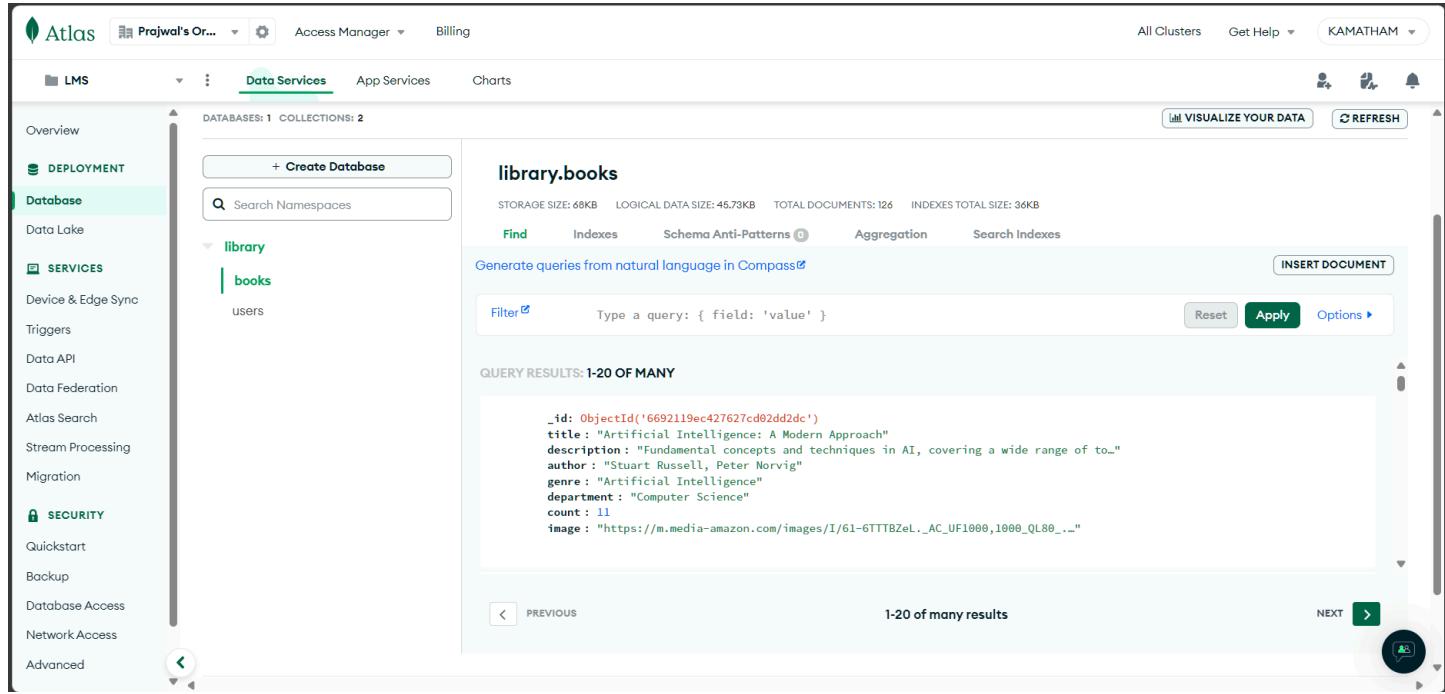
```
PS C:\Users\kamat\Desktop\DevShelf\lms\backend> npm start

> lms@1.0.0 start
> nodemon index.js

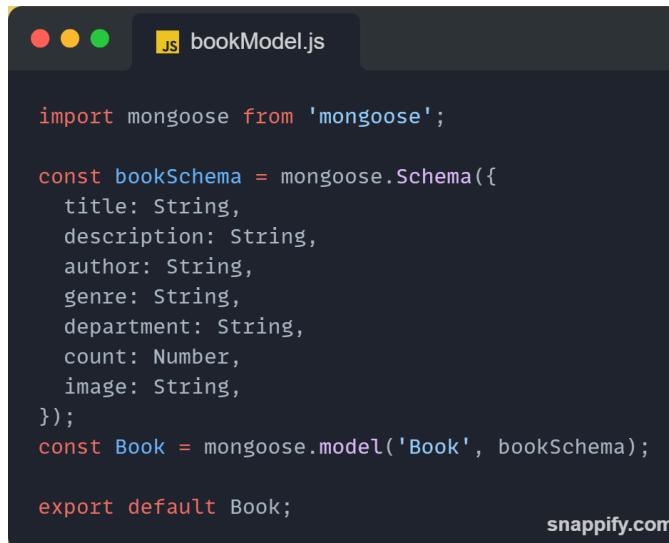
[nodemon] 3.1.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): ***!
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node index.js`
(node:38968) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:38968) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server is listening on port 4001
Connected to MongoDB
```

• Database:

○ Books:



The screenshot shows the MongoDB Atlas Data Services interface. On the left sidebar, under the "Database" section, there is a "library" database which contains a "books" collection. The main panel displays the "library.books" collection details, including storage size (68KB), logical data size (45.79KB), total documents (126), and index size (36KB). Below this, there are tabs for Find, Indexes, Schema Anti-Patterns, Aggregation, and Search Indexes. A search bar at the top right says "Generate queries from natural language in Compass". A query result table shows the first document: _id: ObjectId('6692119ec427627cd02dd2dc'), title: "Artificial Intelligence: A Modern Approach", description: "Fundamental concepts and techniques in AI, covering a wide range of topics.", author: "Stuart Russell, Peter Norvig", genre: "Artificial Intelligence", department: "Computer Science", count: 11, image: "https://m.media-amazon.com/images/I/61-6TTTBZeL._AC_UF1000,1000_QL80_.jpg".



```
import mongoose from 'mongoose';

const bookSchema = mongoose.Schema({
  title: String,
  description: String,
  author: String,
  genre: String,
  department: String,
  count: Number,
  image: String,
});
const Book = mongoose.model('Book', bookSchema);

export default Book;
```

snappify.com

○ Users:

Atlas Prajwal's Or... Access Manager Billing All Clusters Get Help KAMATHAM

LMS Data Services App Services Charts

Overview Search Namespaces STORAGE SIZE: 36KB LOGICAL DATA SIZE: 1.93KB TOTAL DOCUMENTS: 7 INDEXES TOTAL SIZE: 72KB

DEPLOYMENT Database SERVICES

Data Lake Device & Edge Sync Triggers Data API Data Federation Atlas Search Stream Processing Migration SECURITY

Quickstart Backup Database Access Network Access Advanced

Find Indexes Schema Anti-Patterns Aggregation Search Indexes INSERT DOCUMENT

library books users

Generate queries from natural language in Compass Filter Type a query: { field: 'value' } Reset Apply Options

`_id: ObjectId('6692cab2e5110c353cd85ec2')
email: "EE23BT035@iitdh.ac.in"
book1: "6692119ec427627cd02dd2fa"
book1Date: "Mon Jul 15 2024"
book2: "6692119ec427627cd02dd2fa"
book2Date: "Mon Jul 15 2024 18:32:28 GMT+0530 (India Standard Time)"
book3: "6692119ec427627cd02dd2fa"
book3Date: "Mon Jul 15 2024 18:51:47 GMT+0530 (India Standard Time)"
password: "$2a$08$wytwRX02Uk3k/0u8TPImGevBk7UyB5ZgUpP.DCuS4tQnjK7avoloW"
role: "user"
updatedAt: 2024-07-15T14:09:59.728+00:00`

System Status: All Good ©2024 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

A small circular icon with a speech bubble and a checkmark is located in the bottom right corner.

```
● ○ ● JS user.model.js

import mongoose from 'mongoose';

const userSchema = new mongoose.Schema(
{
  email: {
    type: String,
    required: true,
    unique: true,
    trim: true,
    validate: {
      validator: function (value) {
        // Custom validation logic
        return value.endsWith('iitdh.ac.in');
      },
      message: 'Email must have the suffix "iitdh.ac.in"',
    },
  },
  username: {
    type: String,
    required: false,
  },
  password: {
    type: String,
    required: true,
  },
  role: {
    type: String,
    default: 'user',
    enum: ['user', 'admin'],
  },
  book1: {
    type: String,
    default: '',
    required: false,
  },
  book2: {
    type: String,
    default: '',
    required: false,
  },
  book3: {
    type: String,
    default: '',
    required: false,
  },
  book1Date: {
    type: String,
    default: '',
    required: false,
  },
  book2Date: {
    type: String,
    default: '',
    required: false,
  },
  book3Date: {
    type: String,
    default: '',
    required: false,
  },
},
{
  timestamps: true,
}
);

const User = mongoose.model('User', userSchema);
export default User;
```

snappify.com

Challenges Faced

Backend Challenges

1. **Database Design and Management:**
 - Designing a scalable and efficient schema for storing books, users.
 - Handling relationships between different collections (e.g., users and books).
 - Implementing indexing and optimizing queries for performance.
2. **Authentication and Authorization:**
 - Securing endpoints with JWT to ensure that only authenticated users can access certain routes.
 - Implementing role-based access control to restrict certain actions to admin users.
 - Managing token expiration and refresh mechanisms to maintain user sessions securely.
3. **Error Handling:**
 - Creating comprehensive error handling to manage database connection errors, invalid input, or unauthorized access.
 - Providing meaningful error messages to the client side.

Frontend Challenges

1. **State Management:**
 - Managing complex state, especially when dealing with forms, filters, and paginated lists of books.
2. **Form Handling and Validation:**
 - Creating forms for adding books, registering users, and handling borrow/return requests.
 - Validating form inputs and providing user-friendly error messages.
 - Handling file uploads for adding books via JSON or CSV files and ensuring the fields are correct before submitting.
3. **Authentication Flows:**
 - Implementing secure login, registration, and password reset flows.
 - Storing JWT securely (e.g., in HTTP-only cookies or local storage) and using it to authenticate API requests.
 - Redirecting users based on their authentication status and roles.

Integration Challenges

API Communication:

- Ensuring reliable communication between the frontend and backend.
- Handling CORS issues when making API requests from the React frontend to the Node.js backend.

Security Challenges

1. **Securing Sensitive Data:**
 - Encrypting sensitive data (e.g., passwords) while storing.
2. **Token Security:**
 - Protecting JWTs from being stolen or tampered with.

- Ensuring tokens are stored securely and using proper mechanisms to invalidate them when necessary (e.g., on logout).

User Experience Challenges

Intuitive UI/UX:

- Designing a user-friendly interface that makes it easy for users to navigate and perform actions like searching for books, borrowing, and returning them.
- Ensuring the application is responsive and works well on different devices.

Future Plans

Always scope for improvement 😊

As we look ahead, we recognize the continuous scope for enhancement and expansion of our project. Here are our planned improvements:

- Enhance the Admin System:
 - Implement advanced admin controls for better management, such as approval workflows for returns.
- More Security Features:
 - Adding enhanced security features like using security algorithms and encrypting with protocols.
 - Adding security features to handle DDoS attacks
- Develop a Recommendation System:
 - **Initial Considerations:** The team initially explored using the count of available book copies to assess popularity. However, this approach faced limitations, particularly when only a few copies were available or when books were frequently unavailable when needed.
 - **Diverse User Interests:** Branch-specific recommendations were reconsidered due to users' tendencies to explore topics beyond their primary field of study.
 - **Focus on Popularity and Engagement:** Subsequently, the focus shifted to leveraging popularity metrics and user engagement data to recommend books that were trending or highly rated.
 - **Exploring Collaborative Filtering:** The team contemplated implementing collaborative filtering techniques based on user interactions and preferences. This approach, however, necessitated substantial user data and ratings, which were initially lacking.
 - **Average Rating as an Initial Strategy:** To address the absence of initial ratings data, the mentor proposed using an average rating as a starting point. This method would evolve as more ratings were gathered, thereby improving recommendation accuracy over time.
 - **Tracking User Activity:** Recognizing the importance of user interaction history, the decision was made to track activities such as book views and borrowings. This data would facilitate more personalized recommendations based on individual preferences.

- **Implementation Challenges:** Despite the conceptual soundness, implementing the recommendation system posed challenges, particularly in terms of integrating machine learning components and managing the acquisition of necessary knowledge from instructional videos.
- **Future Development:** The team remains committed to refining and enhancing the recommendation system beyond the completion of the current project (DEVSHELF).
- Implement an "I'm Interested" Button:
 - Enable users to express interest in books, facilitating proactive engagement.
- Introduce Rating and Comments:
 - Allow users to rate and provide feedback on books, fostering community interaction and improving recommendations.
- Integrate "Ask a Librarian" Feature:
 - Implement a feature where users can seek assistance from librarians regarding book recommendations or inquiries.
- Launch Newsletters and Magazine Subscriptions:
 - Offer regular updates and curated content through newsletters and subscriptions to enrich user experience and engagement.
- Deployment:
 - Deploying the website to a domain and making it accessible without downloading the GitHub repo.

Team

Kamatham Varshith

Yash Halbhavi

Prajwal Koppad

Suhas Reddypally