

## B12-A11\_category-19

Requirement Explanation Video Link : [Watch Now](#)

🚩 We Will Notify Requirement update here if we made any

---

**Dear Candidates,**

We are pleased to inform you that you have successfully passed the first round of the selection process!! 🎉

Your application and skills have impressed us, and we are excited to move forward with you in the next stages. We are excited to have you explore this opportunity with us. This project is designed to assess your skills, creativity, and problem-solving abilities. It will help us understand how you approach challenges and your ability to deliver high-quality solutions.

---

### Project Name: BookCourier – Library-to-Home Delivery System

#### Project Overview & Discussion

BookCourier is a library delivery management system where users can request book pickup or delivery from their nearby libraries. The system helps students, researchers, and readers borrow and return books without physically visiting the library.

---

#### Ensure the Following things to get 100% mark ( No mark will contain)

- Include at least 20 meaningful commits on the client side & 12 meaningful commits on the server side with descriptive messages.

- Include a README file with the project name, purpose, live URL, key features, and any npm packages you have used.
- Secure Firebase configuration keys using environment variables.
- Secure your MongoDB credentials using the environment variable.
- Create a design that encourages recruiters. Color contrast should please the eye & ensure that the website has proper alignment, space, and the website does not express gobindo design.
- If we found your project similar to any project of your module / conceptual / assignment. You will get 0 and may miss the chance of any upcoming reward.

## Deployment Guideline

If your Deployment is not okay you will get 0 and may miss the chance of our upcoming rewards.

- Ensure that your server is working perfectly on production and not throwing any **CORS / 404 / 504** Errors.
  - Ensure that your Live Link is working perfectly and that it is not showing errors on Landing in your system.
  -  ensure that the page doesn't throw any error on reloading from any routes.
  -  Add your domain for authorization to Firebase if you use **Netlify / surge / Vercel**
  -  Logged in User must not redirect to Login on reloading any private route
- 

## Layout & Page Structure

### Navbar

- Logo + website name
- Links: Home, Books, Request Delivery, Dashboard, Login/Register
- Show user profile image when logged in
- Mobile responsive hamburger menu
- Theme toggle (light/dark)

### Footer

- Quick links
- Contact details
- Social icons (with new X logo, not old Twitter icon)
- Copyright text

## Main Layout

Pages will have a traditional layout where it starts with a navbar and ends with a footer. In between the main content will be placed.

## Dashboard Layout

The dashboard will have a sidebar and main content section. If possible make the sidebar collapsible. Make sure the dashboard is responsive on all devices

---

## Authentication System

### Login

1. Implement email password login and 1 other social login
2. Email verification not required for logging in

### Registration

1. The registration form will have Email, Password, Name, and image fields
2. Upon registration make sure the profile picture properly updates
3. The password must contain validation for it to be a strong password
4. If possible make the image field and “File input” not a text input field (Optional)

## Home Page

The homepage will have a banner with at-least 3 sliders. Each slider will contain images of books, their title and short description and a link button to All books page

5+1 Bonus

### Latest Books Section

This section will show the last 4-6 books added to the database by library owners

### Coverage Section

This section shows a map that shows all the available cities where books can be delivered

### Why Choose Book Courier

This section explains why people should buy books from BookCourier and not somewhere else. Design it however you want.

## Other Requirements

- At least **1 animated section**
- 2 extra well-designed sections

## All Books Jodi Time Pai Tahole Pagenation Add korbo

This page shows all the books that are added into the database by the librarians. Show the books in a card layout. Clicking each card will open book details page

### Book Details

In this page all the details of a book will be shown. And “Order Now” button. Clicking this button will open a modal where user can fill up some information such as

- Name (readonly)
- Email (readonly)
- Phone Number
- Address
- Place Order button
- Status -Payment Status

Clicking the place order button will save the form information into the database and close the modal automatically. When a book is ordered it's status will be “pending” and payment status will be “unpaid”

---

## User Dashboard

### My Orders

In this page users will be able to see all the books they ordered and their status in a ~~table~~.

Each row will show the book title, order date and “Cancel” and “Pay Now” button initially

If an order status is “pending” they will be able to cancel the order by clicking the “Cancel Button”. Otherwise the cancel button will be hidden.

If an order is pending, show them a “Pay now” button, clicking this button will take them to the payment page for that order. After successfully paying for the order, the order's payment status will be “paid” and the Pay Now button will be hidden

Clicking the Cancel button will change the status of the website to “cancelled” and all buttons will be hidden.

## **My Profile**

This page shows the currently logged in user's profile information and a form to update the user profile name and image

## **Invoices**

This page will show all the payments the user have made with the Payment ID, Amount and Date, Optionally you can show the Book name if you want

# **Librarian Dashboard**

## **Add Book**

In this page a librarian will be able to add a book by filling up a form. The form must contain book name, book image, book author, status (dropdown), price fields and a submit button. You can add any other field you want.

Status field will be a dropdown that have 2 values "published", "unpublished" if the book is unpublished then it will not be visible in the all books page

## **My Books**

This page shows all the books the librarian added in a table format. Each row will show the book name, image and edit button. A book can't be deleted after being added, but can be unpublished. Clicking Edit button will open another page where the book can be edited

## **Orders**

In this page a librarian will see all the orders for the books he added in a table, and the librarian can cancel an order.

Also from this page librarians will be able to change the status of an order from “pending” to “shipped” and from “shipped” to “delivered”. It can be a dropdown menu or something else you can think of.

## Admin Dashboard

### All Users

Show all the users who logged-in/registered in the website. For each user there will be a “Make Librarian” and “Make Admin” button

### Manage Books

Show all the books that all the librarians added, Admin can publish / unpublish books and also delete a book. Deleting a book will also delete all the orders of that book

### My Profile

Same as User Dashboard’s My Profile page

### UI Design Requirements:

- **Unique Design:** First, decide what kind of website you want to make. Then, search online or check out websites like ThemeForest to get ideas for the design. But remember, your website idea shouldn’t be similar to any projects you’ve done before or to any examples in our modules or conceptual sessions.  
★ You can also look for free resources on [blogs](#) to help with your website.
1. Keep the main heading style (font, size, color) consistent across all sections.
  2. Keep paragraph spacing balanced and text easily readable.
  3. Maintain uniform image sizes and spacing.
  4. Use the same button style as on the home page.
  5. Ensure good spacing and proper alignment.

6. Navbar, Keep the heading/logo same style and size as on the home page.
7. Use a grid layout with equal image sizes.
8. Keep all cards equal height and width (especially in services, projects, or products section)
9. Use the new X logo instead of the old Twitter bird to match the latest rebrand
10. Responsiveness: Make it responsive for all devices, including mobile, tablet, and desktop views.

### **Dashboard UI Requirements:**

- Responsive design for mobile and tablet screens
- Consistent color theme
- Full-width dashboard
- Charts and graphs for quick data visualization
- User profile section

### Resources:

- <https://uiverse.io/>
- <https://devmeetsdevs.com/>
- <https://bootcamp.uxdesign.cc/free-images-and-resources-collection-for-website-c77f2fc46ce5>
- [https://themeforest.net/?srsItid=AfmBQopTj6PNz51iuV2YJXUtBP8nt19\\_zT5LG2dToAjlHQqzNCzreqnQ](https://themeforest.net/?srsItid=AfmBQopTj6PNz51iuV2YJXUtBP8nt19_zT5LG2dToAjlHQqzNCzreqnQ)
- <https://codecanyon.net/?srsItid=AfmBQooRoUfeK7IOROpchCuA4hPVj5P9WRmtDQJ9K0E6Yhf4VTrHhXKt>

---

## **Challenge Requirement Guideline**

**Book wishlist:** Allow users to add a book to wishlist from the book details page and users wishlisted book will be shown in the My Wishlist page on user dashboard

**Review/Rating:** If a user ordered a book he/she can give the book a rating or review on the book details page, so that other users can see the ratings

**Search & Sort:** In the all books page user can search a book by its name and sort the books by its price.

**JWT Token:** For protected routes use firebase token verification. So only authenticated users can access those routes from the server.

## Optional Requirements

- Use tanstack-query for data fetching so that data can be cached and revalidated.
  - Instead of loading spinners, use skeleton loader for a better look.
  - Implement dark and light mode using tailwindcss
  - Alternative CSS frameworks (like **ShadCN**, **Chakra UI**, **Material React**)
- 

## What to Submit

Admin Email:

Admin Password:

Live Site Link :

Github Repository (**server**):

Github Repository (**client**):