Scaylar Technologies

Challenge: Full-Stack Book Catalog App with PostgreSQL and NextAuth.js (Associate Software Engineers)

This task will assess your skills in building a Full-stack application using Next.js, TypeScript, PostgreSQL, Prisma ORM, and NextAuth.js for user authentication.

In this assignment, you are expected to build a book catalog app that allows users to view, add, and delete books, as well as implement user authentication via email/password or Google sign-up using Next Auth.

You are encouraged to seek help and use any resources you feel are necessary to complete the task. Our goal is to assess your approach to building scalable applications, code quality, and problem-solving skills. If you pass this round, we will review your code in the next interview and ask follow-up questions based on your implementation.

Deadline: 4-5 days

Tech Stack

• Frontend	Next.js (App Router) + TS
• Backend	Next.js API Routes
• Authentication	NextAuth.js (Email/Password, Google)
• Database	PostgreSQL (hosted via Neon/Supabase/ElephantSQL)
• ORM	Prisma
• Deployment	Vercel

BACKEND ROUTES

Methods	Routes	Description
GET	/api/books	Fetch All Books
POST	/api/books	Add a new book
DELETE	/api/books	Delete a book

FRONTEND PAGES (APP ROUTER)

Create pages under `/app` for the following:

Home Page (`/`)

- Fetch and display all books (either **SSR** or **CSR**).
- Use cards, a table, or any other layout for displaying (Tailwind CSS for styling).

Add Book Page (`/add`)

- Form with inputs for `title`, `author`, and `genre`.
- Validate inputs before submission.
- Upon successful submission, redirect to the home page or show a confirmation message.

Delete Book

- Each book entry should have a **delete** button that triggers the `DELETE /api/books/:id` route.

Authentication Pages

- **Login Page (**`/auth/signin`**)** for logging in using **email/password** or **Google**.
- **Sign-Up Page** (optional, can be done through `/auth/signin` using **CredentialsProvider**).
- Show user details and logout functionality when logged in.

Styling

- Use **Tailwind CSS** or another CSS framework.
- Ensure the app is at least 50-70% **responsive** for any one mobile device (and desktop views).
- Style the **login/signup** form and book list page.

Deployment

1. **Push code to GitHub**.
2. **Deploy the app to Vercel**:
   - Set environment variables in **Vercel** (e.g., `DATABASE_URL`, `GOOGLE_CLIENT_ID`, `GOOGLE_CLIENT_SECRET`, `NEXTAUTH_SECRET`).
3. Ensure the app is working correctly on Vercel and that authentication and database integration are functioning.

Deliverables

- GitHub repo link.
- Live deployed link (Vercel).
- **README** with:
  - Setup instructions for running locally.
  - Deployment link.
  - Description of features implemented.
  - **Authentication flow** details.
  - Steps for creating a `.env.local` file.

Evaluation Criteria

– Backend API (Next.js + Prisma)	25%
– Authentication (NextAuth.js)	30%
– TypeScript Usage	15%
– Code Structure & Organization	15%
– UI Design & UX	10%
– Deployment	5%