API Integration Report - Day 3

Project: LMS (Learning Management System)
Date: 20-1-2025
Report By: Sumbul Mohammad Saleem

1. Overview

Today's focus was on integrating APIs to fetch dynamic data from **Sanity CMS** and setting up **authentication** using Firebase Auth / NextAuth.js. We ensured seamless data flow between the frontend and backend, improving performance and reliability.

2. APIs Integrated

Sanity CMS API

✅ **Fetched Course Data:**

- Retrieved course details (title, description, images, duration, price).
- Applied filtering to fetch only relevant courses.
- Used GROQ queries to optimize data retrieval.

✅ **Menu Items Fetching:**

- Implemented dynamic fetching of menu items for display.
- Added category-based filtering.

✅ **Structured Queries for Optimization:**

- Limited fields in API requests to improve response time.
- Implemented SWR (stale-while-revalidate) for caching API responses.

Authentication API (Firebase Auth / NextAuth.js)

✅ **User Authentication Setup:**

- Integrated **Google Sign-In** for seamless login.
- Implemented email/password authentication.

✅ **Token Management & Persistence:**

- Configured JWT tokens for session management.
- Stored authentication tokens securely using HTTP-only cookies.

✅ **User Role-Based Access Control:**

- Implemented conditional UI rendering based on user roles (admin/student).

3. Challenges Faced & Solutions

◆ Data Fetching Optimization

Issue: Initial API queries were returning excessive data, affecting load time.
Solution: Optimized GROQ queries to request only essential fields. Added caching via **SWR**.

◆ Authentication Token Persistence

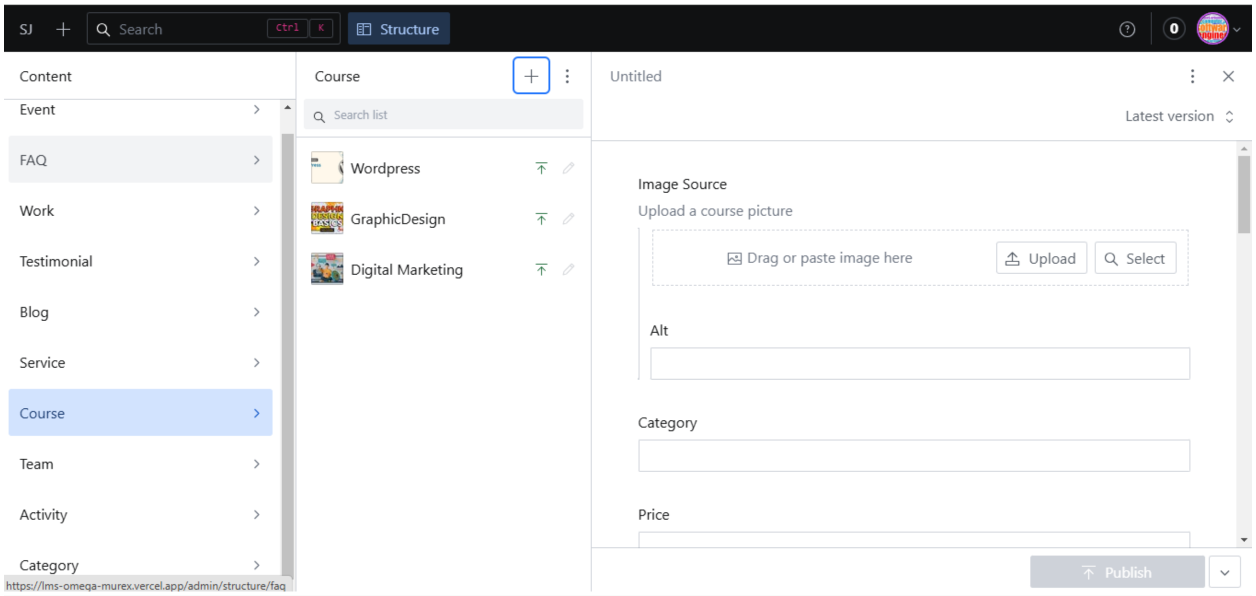
Issue: Tokens were not persisting across sessions, causing unexpected logouts.
Solution: Configured **NextAuth.js session handling** and stored tokens securely in cookies.

◆ Handling API Rate Limits

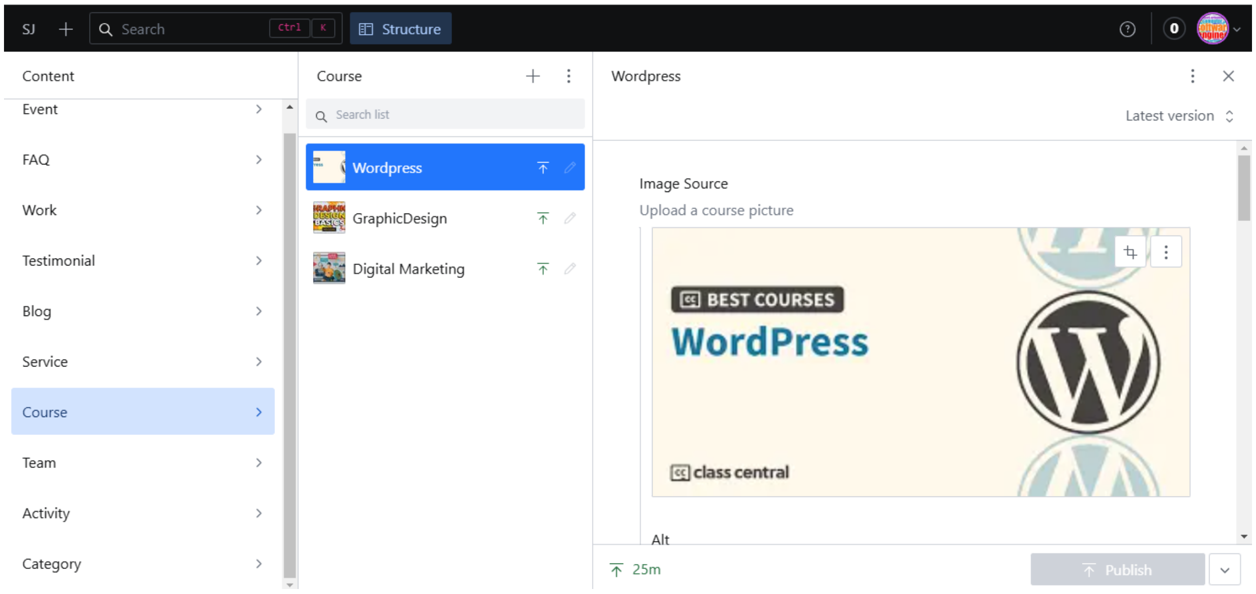
Issue: Sanity API requests were exceeding free-tier limits during testing.
Solution: Batched requests and optimized queries to reduce redundant calls.

4. Next Steps

- **Integrate API for Order and Reservation Management**
  - Develop endpoints to create, read, update, and delete orders/reservations.
  - Secure endpoints with proper authentication & authorization.
- **Improve API Error Handling & Validation**
  - Implement centralized error handling for API failures.
  - Add validation checks for API responses.
- **Enhance Performance & Security**
  - Implement caching strategies for frequently accessed data.
  - Optimize API calls for real-time data updates.



SJ
Search
Ctrl K
Structure
0
Content
Event
FAQ
Work
Testimonial
Blog
Service
Course
Team
Activity
Category
https://lms-omega-murex.vercel.app/admin/structure/faq
Course
Search list
Wordpress
GraphicDesign
Digital Marketing
Untitled
Latest version
Image Source
Upload a course picture
Drag or paste image here
Upload
Select
Alt
Category
Price
Publish



SJ
Search
Ctrl K
Structure
0
Content
Event
FAQ
Work
Testimonial
Blog
Service
Course
Team
Activity
Category
Course
Search list
Wordpress
GraphicDesign
Digital Marketing
Wordpress
Latest version
Image Source
Upload a course picture
BEST COURSES
WordPress
class central
Alt
25m
Publish