Day 3 - API Integration and Data Migration

API Overview:

API (Application Programming Interface) allows different systems to communicate and exchange data. On Day 3, the focus is on integrating external APIs into your application and migrating data efficiently.

Provided APIs:

List of APIs provided for integration during Day 3.

Insert API Documentation Here:

Here, you will document all the APIs provided for your integration, including endpoints, request parameters, responses, authentication methods, etc.

Using External Data Sources (Optional):

Incorporating external data sources like databases or third-party APIs is optional but can enhance the functionality of your application. Ensure the integration is smooth and data is fetched accurately.

Steps for Day 3:

1. Understand the Provided API:

- Review API documentation.
- Identify the endpoints, required parameters, and response formats.
- Determine the type of authentication (if any) and set it up.

2. Validate and Adjust Your Schema:

- Analyze your existing data schema.
- . Modify your schema (e.g., database structure or API models) to match the API responses.
- Ensure data consistency and integrity across your system.

3. Data Migration Options:

- Data Import/Export: Choose between CSV, JSON, or XML formats for transferring data.
- . Bulk Migration: For large datasets, ensure batch processing is handled efficiently.
- · Real-time Migration: Consider using webhooks or API callbacks for real-time data migration.

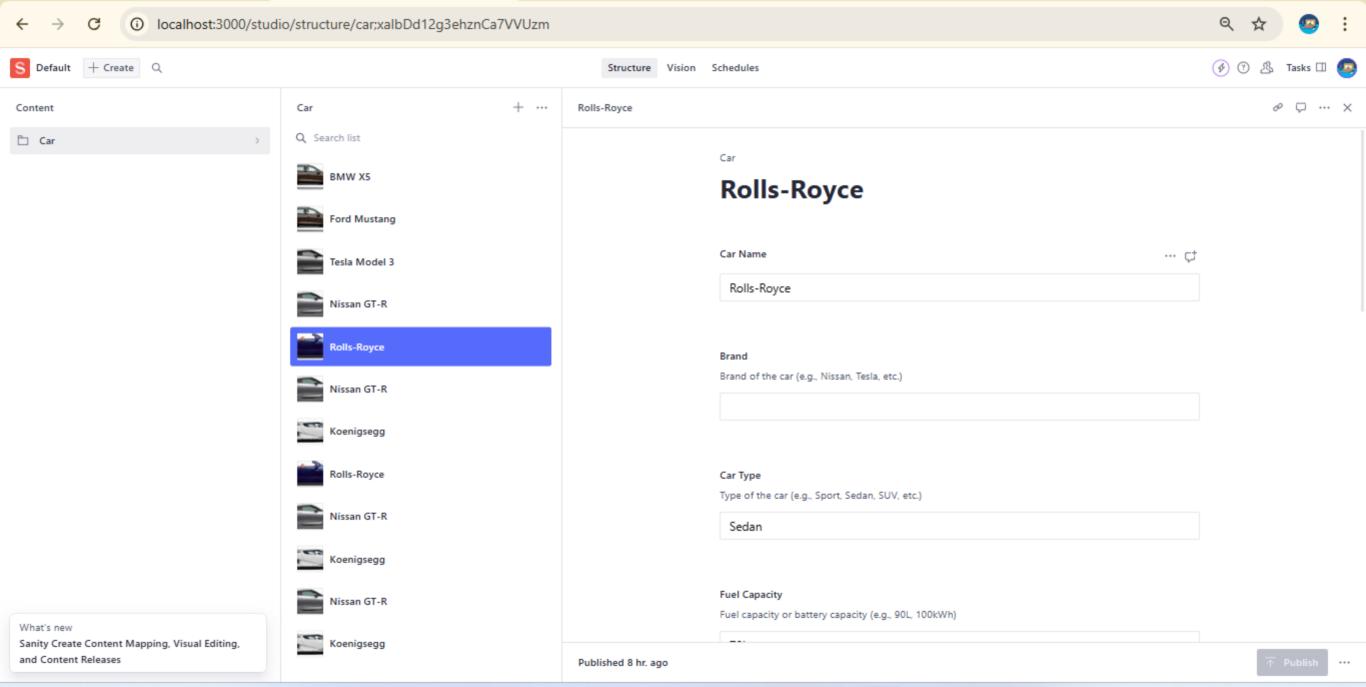
4. API Integration in Next.js:

- Install necessary packages (like Axios, fetch, etc.) for API calls.
- Use getServerSideProps or getStaticProps for server-side fetching.
- Handle asynchronous data fetching in React components.
- . Map API data to your component states and render them dynamically.

Error Handling Tips:

- · Always check for error responses (4xx or 5xx status codes).
- · Implement try-catch blocks in async functions to handle unexpected failures.
- · Provide clear error messages to users and log errors for debugging.

```
15 env.ts A X
sanity > 15 env.ts > ...
  1
       export const apiVersion =
         process.env.NEXT PUBLIC SANITY API VERSION | '2025-01-18'
       export const dataset = assertValue(
         process.env.NEXT PUBLIC SANITY DATASET,
         'Missing environment variable: NEXT PUBLIC SANITY DATASET'
       export const projectId = assertValue(
         process.env.NEXT PUBLIC SANITY PROJECT ID,
         'Missing environment variable: NEXT PUBLIC SANITY PROJECT ID'
       function assertValue<T>(v: T | undefined, errorMessage: string): T {
         if (v === undefined) {
           throw new Error(errorMessage)
         return v
```



Page 65 Carl

