

## **DAY : 3**

### **API INTEGRATION & DATA MIGRATION**

#### Overview

On Day 2, the focus is on API integration and data migration to ensure the marketplace project functions efficiently. These steps establish smooth communication between the frontend and backend while securely transferring data from existing systems to the new platform.

#### Objectives

Integrate APIs for seamless data exchange between frontend and backend.

Migrate existing data to the new platform without errors.

#### API Integration

##### 1. Backend Setup:

Define API endpoints for key functionalities:

User authentication (login, signup, password reset).

Product management (add, update, delete, list products).

Order processing (create, update, retrieve orders).

Implement these endpoints using [Express.js/Next.js API routes].

Test the endpoints using Postman or Thunder Client to ensure reliability.

## 2. Frontend Integration:

Use Axios or Fetch to connect frontend components to the backend.

Handle API responses effectively:

Show loading states while fetching data.

Display error messages for failed requests.

Update the UI dynamically based on API responses.

## 3. Security:

Implement token-based authentication (e.g., JWT).

Use HTTPS for secure communication.

## Data Migration

### 1. Planning the Migration:

List all data entities to be migrated (e.g., Users, Products, Orders).

Map the fields in the old database to the new schema to ensure compatibility.

## 2. Data Extraction:

Export data from the old database in a standard format like CSV or JSON.

Use database-specific tools or scripts to perform the extraction.

## 3. Data Transformation:

Clean and preprocess data to match the structure of the new database.

Handle missing or duplicate data during this step.

## 4. Data Loading:

Import the processed data into the new database using migration scripts or tools.

Run validation tests to confirm the data has been imported correctly.

## Notes

Error Handling: Implement logs to capture errors during API calls or data migration.

Testing: Verify all API endpoints with sample data. Check for data consistency after migration.

Security: Use encryption during data transfer and secure sensitive information like passwords.

## Conclusion

API integration ensures smooth interactions between the user interface and server, while proper data migration guarantees a solid start for the marketplace. Both are critical steps for building a scalable and user-friendly platform.