

**Requirement Gathering and Analysis Phase  
Technology Stack (Architecture & Stack)**

Date	22 July 2024
Team ID	SWTID1720760336
Project Name	Bookstore- Where Stories Nestle
Maximum Marks	

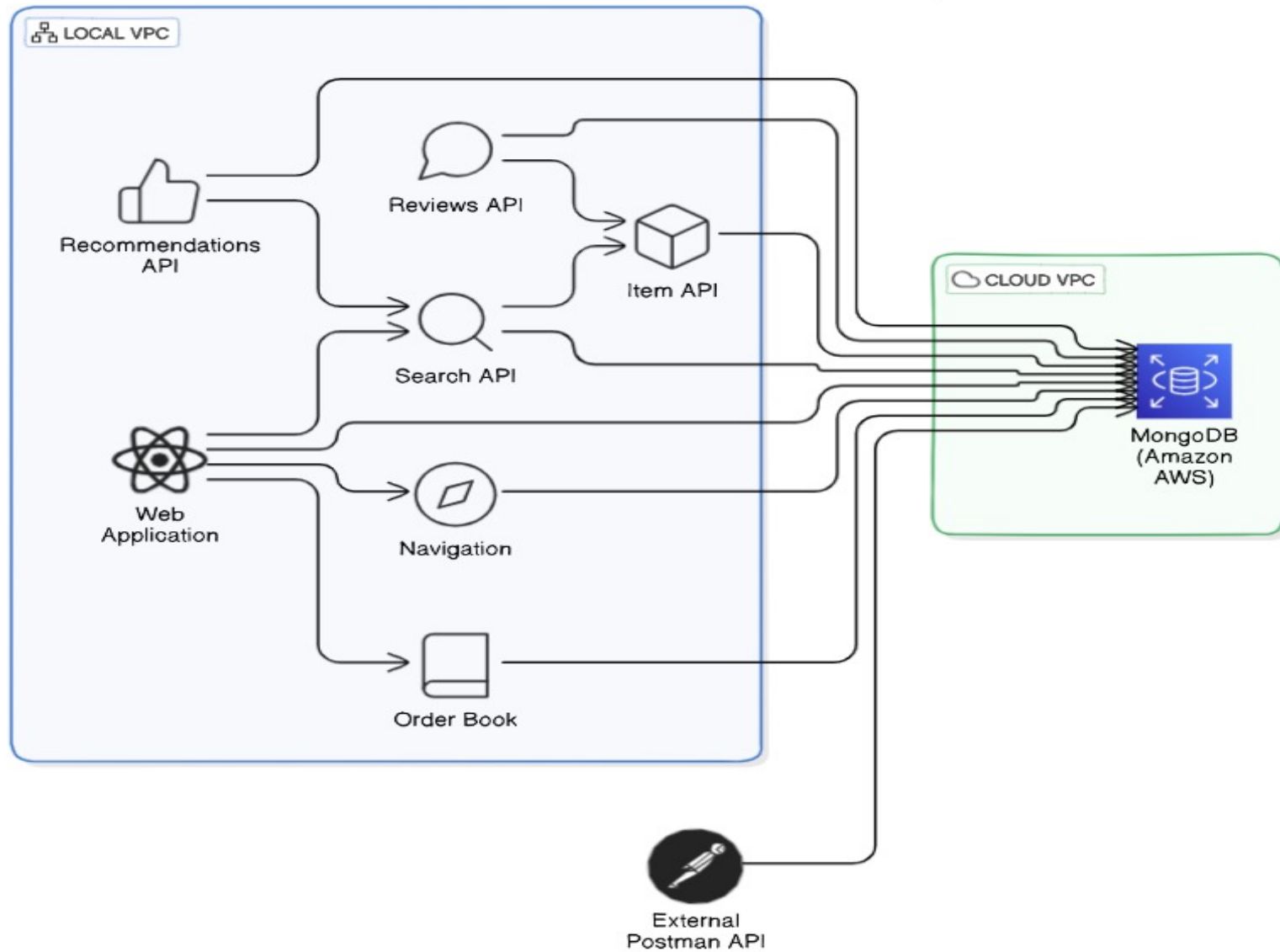
**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

**Example: Order processing during pandemics for offline mode**

**Reference:** <https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/>

## Where Stories Nestle: Bookstore Project



**Table-1 : Components & Technologies:**

S.No	Component	Description	Technology
1.	User Interface	How users interact with the application through web or mobile platforms.	HTML, CSS, JavaScript / React.js
2.	Application Logic-1	Handles specific processes within the application, enabling backend operations.	Node.js
3.	Application Logic-2	Manages routing and server-side logic to ensure smooth data flow and communication.	Express.js
4.	Database	Stores and manages data types and configurations for the application.	MongoDB
5.	Cloud Database	Provides scalable database services hosted in the cloud.	MongoDB Atlas
6.	File Storage	Handles storage needs for files, both locally and in the cloud.	Local Filesystem, Cloud Storage
7.	External API-1	Purpose of External API used in the application	Postman
8.	Infrastructure (Server / Cloud)	Deploys the application either locally or on the cloud for scalability and management	Local, Cloud Foundry

**Table-2: Application Characteristics:**

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilizes open-source technologies to build the application with flexibility and community support.	MERN Stack (MongoDB, Express.js, React.js, Node.js)
2.	Availability	Ensures the application is highly available through load balancing and distributed server architecture.	Load Balancers, Distributed Servers
3.	Performance	Enhances performance using caching mechanisms and content delivery networks , with Express.js optimizing backend performance.	Caching, Express.js

**References:**

<https://c4model.com/>

<https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/>

<https://www.ibm.com/cloud/architecture>

<https://aws.amazon.com/architecture>

<https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d>