

Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	28 MAY 2025
Team ID	LTVIP2025TMID55905
Project Name	BookNest: Where stories nestle
Maximum Marks	4 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/>



Book Nest Technology Stack

Comprehensive Technology Architecture & Implementation Plan



FRONTEND

React.js
UI Framework

v18.2

TypeScript
Type Safety

v5.0

Tailwind CSS
Styling Framework

v3.3

Next.js
React Framework

v14



BACKEND

Node.js
Runtime Environment

v20 LTS

Express.js
Web Framework

v4.18

Prisma ORM
Database ORM

v5.0

JWT
Authentication

Latest



DATABASE

PostgreSQL
Primary Database

v15

Redis
Caching & Sessions

v7.0

Elasticsearch
Search Engine

v8.8



CLOUD & HOSTING

AWS EC2
Server Hosting

Latest

AWS RDS
Database Hosting

Latest

CloudFront CDN
Content Delivery

Latest

S3
File Storage

Latest



DEVOPS

Docker
Containerization

Latest

GitHub Actions
CI/CD Pipeline

Latest

Nginx
Reverse Proxy

v1.24



TESTING

Jest
Unit Testing

v29

Cypress
E2E Testing

v13

Artillery
Performance Testing

Latest



ANALYTICS

Google Analytics
User Analytics

GAA

Sentry
Error Monitoring

Latest

New Relic
Performance Monitoring

Latest



SECURITY

bcrypt
Password Hashing

Latest

Helmet.js
Security Headers

Latest

Rate Limiting
API Protection

Express

TECHNOLOGY	CATEGORY	PURPOSE	JUSTIFICATION	COST
React.js + TypeScript	Frontend	User Interface Development	Industry standard, large community, excellent for complex UIs with type safety	Free
Node.js + Express	Backend	Server-side Development	JavaScript ecosystem consistency, high performance, extensive package ecosystem	Free
PostgreSQL	Database	Data Storage & Management	ACID compliance, excellent performance, JSON support, strong community	Free
AWS Cloud Services	Infrastructure	Hosting & Scaling	Industry leader, comprehensive services, excellent scaling capabilities	Pay-as-you-go
Redis	Caching	Performance Optimization	In-memory speed, session management, reduces database load	Free
Elasticsearch	Search	Advanced Search Features	Full-text search, faceted search, excellent for book catalog searching	Licensing
Docker + CI/CD	DevOps	Deployment & Automation	Consistent environments, automated deployments, scalability	Free

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application Web UI	HTML, CSS,ReactJS+Vite/Bootstrap, CSS etc.
2.	Application Logic-1	Logic for a process in the application	JavaScript.
3.	Database	Data Type, Configurations etc.	MongoDB, Mongoose.
4.	File Storage	File storage requirements	MongoDB Cluster storage.
5.	External API-1	Purpose of External API used in the application	
6.	External API-2	Purpose of External API used in the application	

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Frontend uses React (via Vite), Tailwind CSS, Bootstrap for UI components, Axios for HTTP requests. Backend is built using Node.js with Express.	React, Vite, Tailwind CSS, Bootstrap, Axios, Node.js, Express.js
2.	Security Implementations	Passwords are encrypted using bcrypt. CORS is implemented for secure cross-origin communication. Input validations prevent injection attacks.	bcrypt, CORS, express-validator, Helmet (optional)
3.	Scalable Architecture	Follows a modular architecture separating frontend, backend, and database (3-tier). Can be containerized using Docker for scaling.	Node.js Microservices (optional),
4.	Availability	Application can be deployed on cloud platforms (e.g., Heroku, Render, AWS) with horizontal scaling. Load balancers can be used if demand increases.	Cloud platforms (Render, AWS, etc.), Nginx (optional)

5.	Performance	Efficient API calls with Axios, caching static content using CDN. MongoDB handles highvolume reads/writes efficiently.	Axios, MongoDB, CDN (e.g., Cloudflare), Compression
----	-------------	--	---

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>