

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

Date	25-06-2025
Team ID	LTVIP2025TMID53185
Project Name	BookNest: Where Stories Nestle
Maximum Marks	4 Marks

### Technical Architecture:

#### Overview

BookNest is a digital platform where users can write, share, and discover stories. The design phase includes architectural decisions, technology selection, and identification of system characteristics to ensure scalability, performance, and maintainability.

**BookNest: Where Stories Nestle** is designed with a scalable 3-tier architecture—like a well-organized library with separate floors for different functions:

1. **Presentation Tier (Frontend)** – The user-facing layer built with React.js, like the front desk of a library guiding readers to their stories.
2. **Application Tier (Backend/API)** – Node.js with Express.js processes logic and requests, much like librarians handling book searches and checkouts.
3. **Data Tier (Database & Storage)** – PostgreSQL/MongoDB and AWS S3 store and retrieve data, acting as the archive room where all stories and media are safely shelved.

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

Component	Technology	Purpose
Frontend	React.js	Build dynamic, responsive user interfaces
Backend Framework	Node.js + Express.js	REST API development and server-side logic
Database	MongoDB	NoSQL database for storing users, books, reviews, etc.
Authentication	JWT, bcrypt.js	Secure user login and access control
Hosting (Frontend)	Vercel / Netlify	Hosting and deployment of frontend
Hosting (Backend & API)	Render / Railway	Hosting backend services and Node.js APIs
State Management	React Context / Redux	Manage app-wide states such as user login
Styling	Tailwind CSS / CSS3	Styling and responsive design
Form Handling	React Hook Form	Validate and manage user input efficiently
API Communication	Axios / Fetch API	Client-server communication
Version Control	Git + GitHub	Source code management and collaboration
Testing	Jest / React Testing Library	Unit and component testing
CI/CD	GitHub Actions	Automate testing and deployment

**Table-2: Application Characteristics:**

Characteristic	Implementation Strategy
User Registration & Login	JWT-based authentication with bcrypt password hashing
Role-Based Access Control	Admin vs. User privileges managed on frontend and backend
Book Posting & Reading	MongoDB collections for storing and querying book metadata and content
Review and Rating System	Nested document structure in MongoDB with user ID references
Search and Filter	MongoDB Atlas Search + client-side filters using React
Responsive UI	Mobile-first design with Tailwind CSS
Real-time Features	Socket.io (for live chat/discussion)
Error Handling	Custom middleware in Express for API error response
Scalability	Stateless API, database sharding (MongoDB Atlas), microservices-ready

**References:**

**MongoDB Documentation** – <https://www.mongodb.com/docs/>

**React.js Official Docs** – <https://reactjs.org/>

**Node.js Docs** – <https://nodejs.org/en/docs/>

**Express.js Guide** – <https://expressjs.com/>

**Tailwind CSS** – <https://tailwindcss.com/>

**JWT Authentication** – <https://jwt.io/introduction>

**GitHub Actions CI/CD** – <https://docs.github.com/en/actions>

**BookNest Concept & Design Notes** – Project Requirements Document (internal)