📘 BookNest Project Documentation

---

1. Project Overview

BookNest is a modern, scalable web application for **renting** and **purchasing** books, offering a streamlined interface for both users and administrators.

Key Objectives:

- **User-centric:** Seamless book discovery, rent/buy processes, and account management.
- **Admin-focused:** Robust control over inventory, users, and order workflows.
- **Technical:** Secure architecture, high performance, and future expansion.

Tech Stack:

- **Frontend:** React.js, HTML5, CSS3
- **Backend:** Node.js, Express.js
- **Database:** MongoDB
- **Authentication:** JWT
- **Infrastructure:** AWS (EC2, S3, MongoDB Atlas)
- **CI/CD:** GitHub Actions

---

2. Functional Specifications

2.1 User Module

- **Registration & Authentication** (Email/password, JWT tokens)

- **Browse/Search Catalog** (filters by title, author, genre, availability)
- **Rent/Buy Workflow** (rent duration selection, cart handling)
- **Dashboard** (active orders, due dates, history)
- **Profile Management** (user info, password updates)

2.2 Admin Module

- **Admin Authentication**
- **Book Management** (Add/Edit/Delete, metadata and stock control)
- **User Management** (list, view activities, deactivate accounts)
- **Order Reporting** (daily/weekly metrics, analyze rentals vs purchases)

2.3 Inventory Management & Order Lifecycle

- **Book Metadata:** title, author, ISBN, genre, cover image, inventory
- **Statuses:** Available / Rented / Sold
- **Order Path:** Order → Payment → Confirmation → (If Rent) Due Date → Return → Completed

---

3. System Architecture & Diagrams

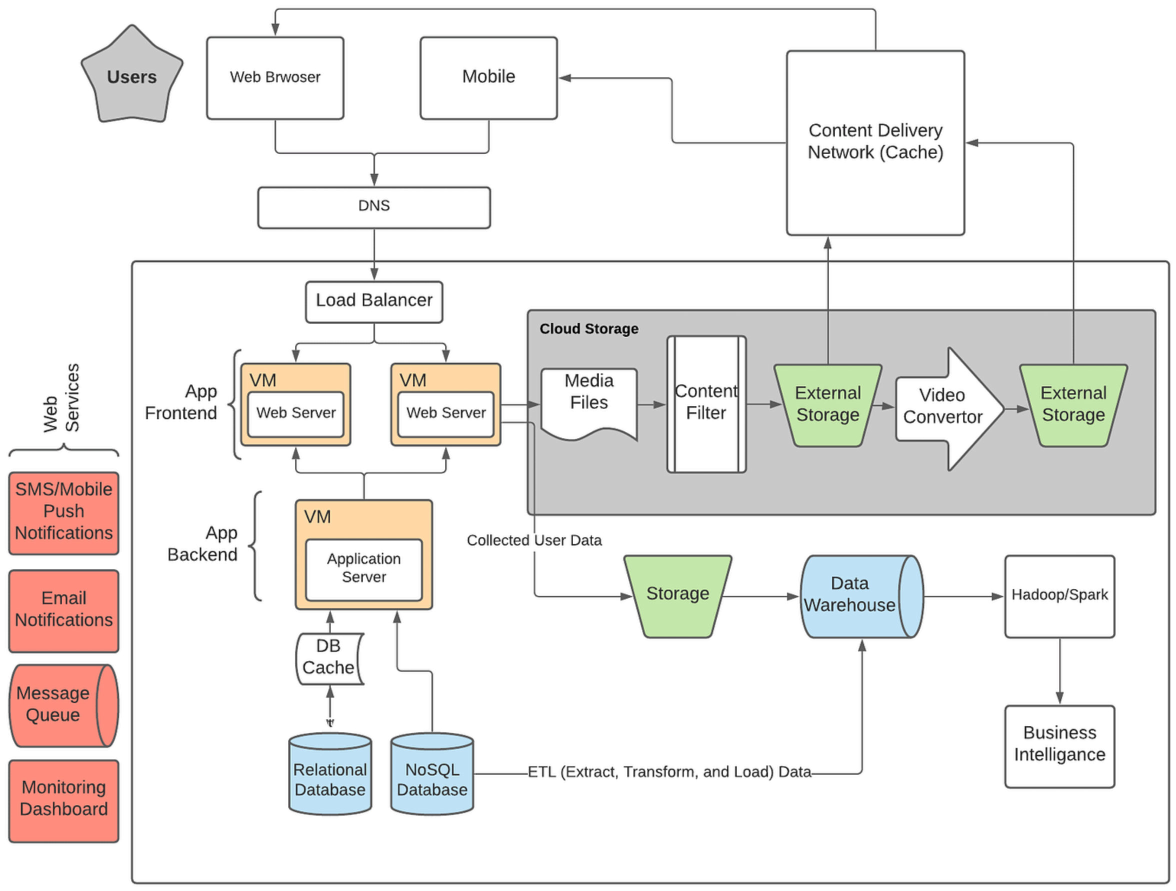
3.1 Architecture Overview

Place this diagram near the top of the section:

Insert Image 1: Web-app architecture (e.g., image0)

This depicts:

- Client (React)
- API server (Express)
- MongoDB backend
- Payment gateway integration
- Cron scheduler for due-date reminders
  Adapted with simplification from standard web architecture references en.wikipedia.org+1medium.com+1reddit.comradixweb.com+13medium.com+13creately.com+13.



3.2 Layered Layer-by-Layer Breakdown

1. **Presentation Layer** – React components, UI logic

2. **Business Layer** – API endpoints, services, validation en.wikipedia.org+2integrio.net+2existek.com+2
3. **Data Access Layer** – MongoDB queries, Mongoose
4. **Database Layer** – MongoDB Atlas clusters

3.3 Entity-Relationship Diagram

Use a drawing to illustrate:

Entities:

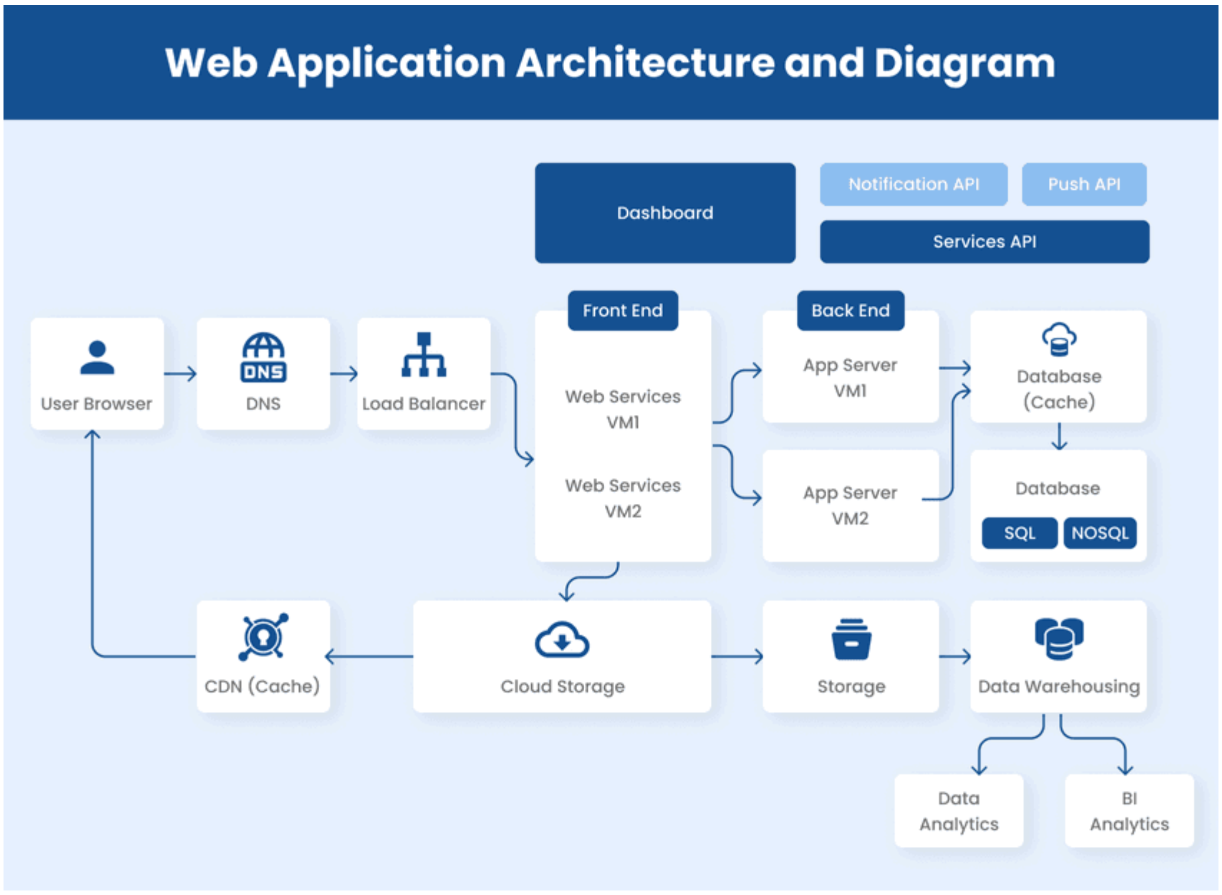
- **User** (userId, name, email, passwordHash, role, timestamps)
- **Book** (bookId, title, author, ISBN, genre, coverURL, totalCopies, availableCopies)
- **Order** (orderId, userId, bookId, type, duration, status, paymentStatus, dueDate, timestamps)

Connect `User → Order → Book`.

3.4 Architecture Alternatives Comparison

- **Monolithic**: Simpler, easy deployment, limited scalability integrio.netreddit.com+3asperbrothers.com+3bacancytechnology.com+3existek.com+6softkraft.co+6radixweb.com+6
- **Microservices**: Scalable, independent development; complexity overhead stackify.com+13integrio.net+13radixweb.com+13
- **Serverless**: Cost-effective, scalable; debugging and vendor lock-in concerns researchgate.net+2softkraft.co+2reddit.com+2

Recommendation: Start monolithic; modularize critical services as needed.



4. Workflow & Process Flows

4.1 User Workflow Diagram

[Insert a flowchart: Browse → Detail → Order → Payment → Confirmation → (Rent) due date/crons → Return]

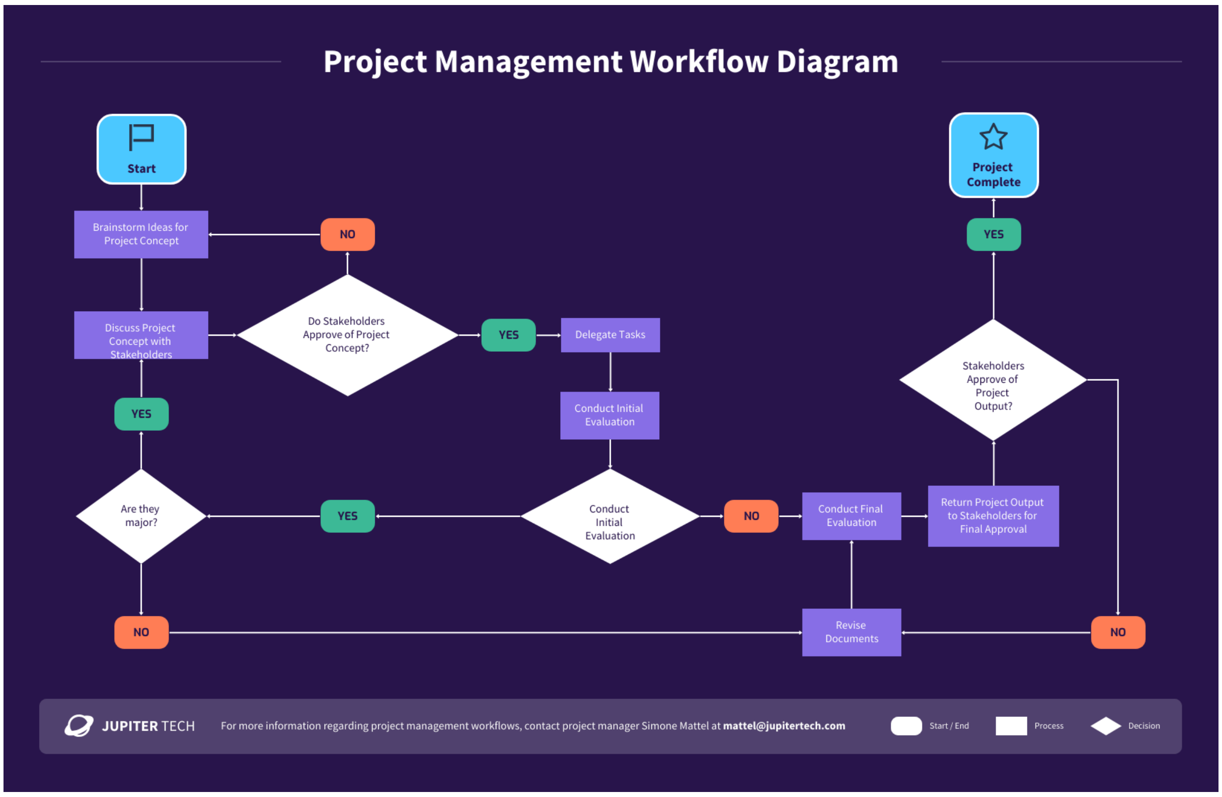
4.2 Admin Workflow Process

Flowchart: Admin login → Inventory → New book → Order view → Return processing → Reports

4.3 Component Interaction Overview

- Load Balancer → Express API → MongoDB

- Payment via gateway + webhook
- Cron scheduler for reminders/returns


Project Management Workflow Diagram
Start
Brainstorm Ideas for Project Concept
NO
Discuss Project Concept with Stakeholders
Do Stakeholders Approve of Project Concept?
YES
Delegate Tasks
Conduct Initial Evaluation
YES
Are they major?
YES
Conduct Initial Evaluation
NO
Conduct Final Evaluation
Return Project Output to Stakeholders for Final Approval
Stakeholders Approve of Project Output?
YES
Project Complete
NO
Revise Documents
NO
JUPITER TECH
Start / End
Process
Decision

For more information regarding project management workflows, contact project manager Simone Mattel at **mattel@jupitertech.com**

---

5. API Specification

Endpoint	Method	Description	Access
`/api/auth/register`	POST	Create a new user	Public
`/api/auth/login`	POST	Authenticate and return JWT	Public

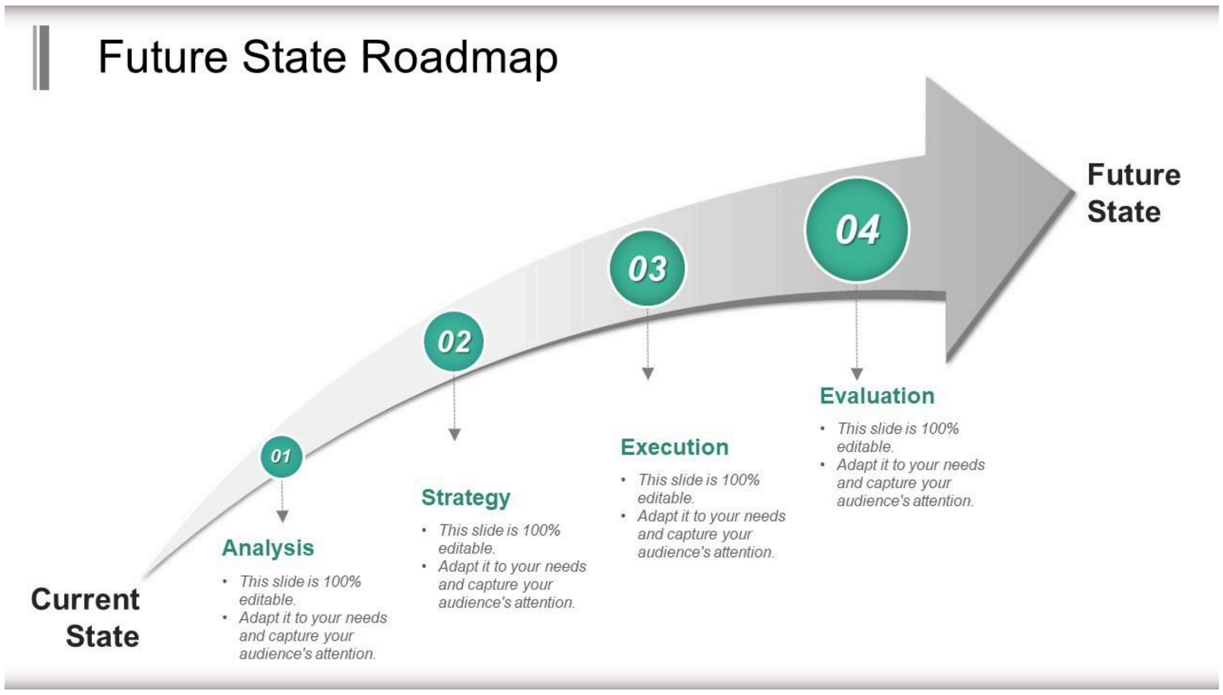
`/api/books`	GET	List all books	Auth
`/api/books/:id`	GET	Book details	Auth
`/api/books`	POST	Add a book	Admin
`/api/books/:id`	PUT	Update book	Admin
`/api/books/:id`	DELETE	Delete book	Admin
`/api/orders`	POST	Place a new order	Auth
`/api/orders/user/:id`	GET	Get user orders	Auth/Admin
`/api/orders`	GET	List all orders	Admin

Headers: All require `Authorization: Bearer <JWT>`

---

6. Future Roadmap & Enhancements

- **Payment Integrations** (Stripe, Razorpay with webhooks)
- **Mobile App** in React Native
- **Recommendation Engine** using collaborative filtering
- **Ratings & Reviews** feature
- **Notifications** via Email/SMS (SendGrid, Twilio)
- **Analytics Dashboard** (top genres, revenues, overdue count)
- **In-app Chat Support**


Future State Roadmap
Future State
04
03
02
01
Evaluation
• This slide is 100% editable.
• Adapt it to your needs and capture your audience's attention.
Execution
• This slide is 100% editable.
• Adapt it to your needs and capture your audience's attention.
Strategy
• This slide is 100% editable.
• Adapt it to your needs and capture your audience's attention.
Analysis
• This slide is 100% editable.
• Adapt it to your needs and capture your audience's attention.
Current State

---

7. Tools & Diagramming Tips

- Use **draw.io / diagrams.net** (free, integrates with Drive) velvetech.com+4peerbits.com+4asperbrothers.com+4medium.com+2asperbrothers.com+2reddit.com+2en.wikipedia.org+1reddit.com+1
- Try **Mermaid** for text-based diagrams en.wikipedia.org+5en.wikipedia.org+5creately.com+5

- Follow **C4 model** and **Clean Architecture** for structure clarity reddit.com+3en.wikipedia.org+3learn.microsoft.com+3
- Alternatives: **Visio**, **Lucidchart**, **yEd**, **PlantUML** reddit.com+1en.wikipedia.org+1
-