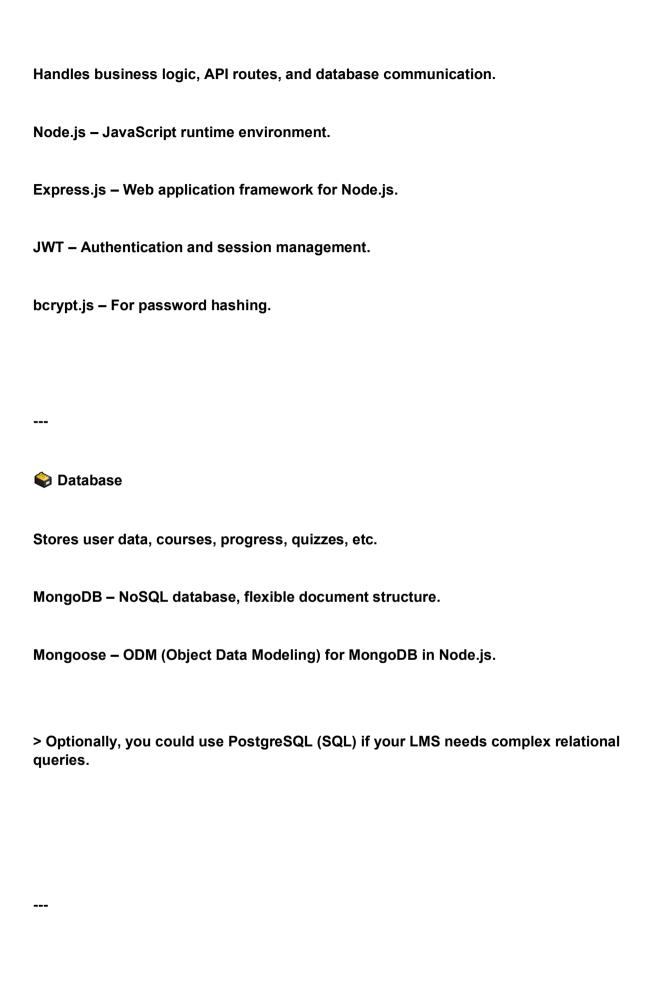
## Project Design Phase-II Technology Stack (Architecture & Stack)

Date	26-05-2025
Team ID	
Project Name	LEARNHUB
Maximum Marks	4 Marks

Maximum Marks	4 Marks
	ck for a Learning Management System (LMS)
like LearnHub typically includes a combin hosting, and other supporting technologic you can use:	ation of frontend, backend, database, es. Here's a modern and scalable tech stack
This is what users interact with.	
React.js – Fast, component-based UI.	
Redux / Context API – State management.	
Tailwind CSS / Bootstrap / Material UI – St	yling and UI components.
Vite – For faster development and build to	oling (instead of CRA).
React Router – Client-side routing.	

Backend (Server-Side)



○ Cloud & Hosting
Where the app lives and runs.
Render / Vercel / Netlify – Hosting for frontend (React).
Render / Railway / Heroku / Cyclic – Hosting backend Node.js/Express API.
MongoDB Atlas – Cloud MongoDB database.
<b></b>
For course videos, images, documents.
Cloudinary / AWS S3 – Image/video storage and optimization.
<b></b>
Email & Notification Services
For user communication.
Nodemailer – Sending transactional emails (e.g., registration, reset password).
Firebase Cloud Messaging (FCM) or OneSignal – Push notifications.

Admin Dashboard (Optional)
For managing users, courses, enrollments.
React Admin or Custom Admin Panel – Built with the same frontend stack.
Chart.js / Recharts – Data visualization for stats.
✓ Others / Dev Tools
Git & GitHub – Version control.
Postman – API testing.
Dotenv – Manage environment variables.
ESLint / Prettier – Code quality and formatting.
Testing (Optional)

**Jest – Unit testing for React components.** 

Supertest + Mocha/Chai - API endpoint testing.

---

**Summary Diagram:** 

Frontend: React + Vite + Tailwind

I

API: Node.js + Express + JWT + Mongoose

**Database: MongoDB Atlas** 

1

**Media: Cloudinary** 

ı

Hosting: Vercel (frontend) + Render/Railway (backend)

Let me know if you want the stack tailored for mobile apps, multilingual support, payment integration, or microservices architecture.

Table-1 : Components & Technologies:

S.N	Component	Description	Technology
0			
1.	User Interface	Web and mobile-friendly interface for patients and providers	HTML, CSS, JavaScript / React Js etc.
2.	Application Logic-1	Appointment booking, calendar management, reminders	Node.js, Express.js
3.	Application Logic-2		React js, Node js
		Admin panel, provider management, reporting	
4.	Database	Stores user profiles, appointments, provider datas	MongoDB

**Table-2: Application Characteristics:** 

S.N o	Characteristics	Description	Technology
5.	Open-Source Frameworks	Frontend frameworks	React.js, Node.js, BootStrap, Tailwind CSS
6.	ecture	3-tier architecture with RESTful APIs	Microservices

## References:

**React.js Documentation** 

**Node js Best Practice** 

**JSON Web Server Referance** 

 $\frac{https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d}{}$