

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

| | |
|---------------|--|
| Date | 19 June 2025 |
| Team ID | LTVIP2025TMID56879 |
| Project Name | FreelanceFinder: Discovering Opportunities, Unlocking Potential |
| Maximum Marks | 4 Marks |

Technical Architecture:

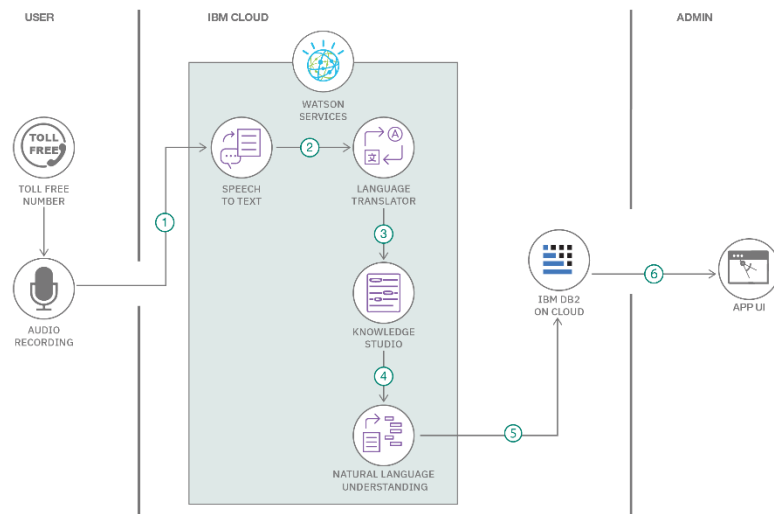
The SB Works application is a full-stack freelancing platform built using the **MERN stack** (MongoDB, Express.js, React.js, Node.js) and hosted in the cloud. It enables seamless interaction between freelancers and clients, featuring registration, bidding, real-time chat, project tracking, and admin oversight.

Key Architecture Elements Include:

- **Web-based user interface** (React.js)
- **RESTful APIs** for business logic (Node.js + Express.js)
- **NoSQL cloud-based database** (MongoDB Atlas)
- **Cloud deployment** (e.g. Vercel for frontend, Render/Heroku for backend)
- **JWT-based authentication** for secure sessions
- **Real-time chat** using Socket.io
- **Admin panel** to manage and resolve disputes
- **Optional third-party integrations** for email verification and file uploads (e.g., SendGrid, Cloudinary)

Example: Order processing during pandemics for offline mode

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



Guidelines:

Include all the processes (As an application logic / Technology Block)
 Provide infrastructural demarcation (Local / Cloud)
 Indicate external interfaces (third party API's etc.)
 Indicate Data Storage components / services
 Indicate interface to machine learning models (if applicable)

Table-1: Components & Technologies

| Component | Description |
|-------------------------|---|
| User Interface | - Web UI using React.js, HTML, CSS, and Bootstrap |
| Application Logic | - Backend with Node.js and Express.js handling registration, bidding, chat, admin |
| Real-time Communication | - Implemented using Socket.io for freelancer-client chat |
| Database | - MongoDB Atlas used for storing users, projects, applications |
| File Storage | - Cloudinary/AWS S3 used for managing project and profile media |
| Authentication | - JWT used for secure login, bcrypt for password hashing |
| Email Services | - SendGrid/Nodemailer used for email verification and communication |

| Component | Description |
|-----------------|---|
| Social Login | - Google OAuth for third-party authentication (optional) |
| Admin Dashboard | - AdminBro or custom panel for oversight and dispute resolution |
| Infrastructure | - Frontend on Vercel, backend on Render/Heroku, CI/CD enabled |

Table-2: Application Characteristics:

| S.No | Characteristics | Description | Technology |
|------|--------------------------|---|---|
| 1. | Open-Source Frameworks | All stack components are open-source | React.js, Node.js, Express.js, MongoDB |
| 2. | Security Implementations | JWT Auth, password hashing, validation, admin access controls | bcrypt.js, JWT, Helmet, CORS, OWASP |
| 3. | Scalable Architecture | Modular MVC architecture with REST APIs and NoSQL DB for horizontal scaling | MERN stack, RESTful APIs, Docker (optional) |
| 4. | Availability | Deployed to cloud with high uptime, monitored logs | Vercel, Render, UptimeRobot |
| 5. | Performance | Optimized queries, async operations, static asset delivery, CDN-ready | MongoDB indexing, lazy loading, caching |

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>