**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 26-06-2025 |
| Team ID | Single |
| Project Name | Freelancing application mern |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The Freelance application mern platform is designed with a scalable 3-tier architecture that includes the presentation layer (frontend), business logic (backend), and data storage layer. The solution ensures performance, security, and ease of integration with third-party APIs (e.g., Stripe for payments).

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

| **S.No** | **Component** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | User Interface | Web-based interface for clients and freelancers | HTML, CSS, JavaScript / React Js etc. |
|  | Application Logic-1 | Job posting, application, dashboard, messaging | Node.js, Express.js |
|  | Application Logic-2 | |  | | --- | |  |  |  | | --- | | Admin panel, job tracking, notifications | | React js, Node js |
|  | Database | Stores user data, jobs, applications, messages | MongoDB |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Frontend frameworks | React.js, Node.js, BootStrap, Tailwind CSS |
|  | Scalable Architecture | 3-tier architecture with RESTful APIs | Microservices |

**References**

Core Technologies

1. React.js
   * Official Documentation: <https://reactjs.org/docs/getting-started.html>  
     *Usage: Frontend component architecture, state management (Context/Hooks).*
2. Node.js & Express.js
   * Node.js Best Practices: <https://github.com/goldbergyoni/nodebestpractices>  
     *Usage: RESTful API design, middleware implementation, error handling.*
3. MongoDB
   * Data Modeling Guide: <https://www.mongodb.com/docs/manual/core/data-modeling-introduction/>  
     *Usage: Schema design for users, jobs, and transactions.*

Authentication & Security

1. JSON Web Tokens (JWT)
   * RFC 7519 Standard: <https://tools.ietf.org/html/rfc7519>  
     *Usage: Secure user sessions, role-based access control.*
2. OWASP Security Guidelines
   * API Security Cheat Sheet: <https://cheatsheetseries.owasp.org/cheatsheets/REST_Security_Cheat_Sheet.html>  
     *Usage: Input validation, rate limiting, anti-CSRF measures.*

Architecture & Diagrams

1. Technical Architecture Diagrams
   * Medium Article: [How to Draw Useful Technical Architecture Diagrams](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)  
     *Usage: Designing DFDs and system overviews (cited in your request).*
2. Mermaid.js
   * Documentation: <https://mermaid-js.github.io/mermaid/>  
     *Usage: Code-based flowchart generation for documentation.*