SB Works – Design Documentation

**Frontend:**

* **Framework & Libraries:**  
  Built with **React.js**, leveraging **React Router** for SPA (Single Page Application) navigation and **Axios** for API calls.
* **Styling & Components:**  
  UI is crafted using a combination of **Bootstrap** for responsiveness and **Material UI (MUI)** for a modern and consistent design language.
* **Features:**
  + Role-based dashboards (Freelancer, Client, Admin)
  + Reusable card components for jobs, profiles, reviews, etc.
  + Form validation using libraries like **Formik** and **Yup**
  + Responsive design compatible with desktop and mobile screens.

**2. Backend:**

* **Runtime Environment:**  
  **Node.js** with **Express.js** handles routing, server-side logic, and business workflows.
* **RESTful API Design:**
  + Follows REST principles for clear and predictable endpoint design.
  + Routes are modularly structured based on functionality (/auth, /jobs, /users, /messages).
* **Error Handling & Logging:**
  + Centralized error handler with meaningful HTTP status codes.
  + Integration of **Winston** or **Morgan** for request logging.

**3. Database:**

* **Type:**  
  **MongoDB (NoSQL)** for flexibility and fast development.
* **Collections include:**
  + Users: stores freelancer/client data, roles, ratings
  + Projects: job details, requirements, status, deadlines
  + Messages: chat logs, timestamps, sender/receiver IDs
  + Reviews: client feedback and freelancer ratings

**4. Communication Layer:**

* **Frontend-Backend Interaction:**
  + **Axios** is used on the frontend to send and receive data through secure REST APIs.
* **Real-time Messaging (Planned for future):**
  + Integration of **Socket.io** to enable real-time updates and chat communication between freelancers and clients.
* **Notifications:**
  + Toast notifications for success/error states.
  + Planned integration of email notifications (via Nodemailer) for job status, messages, and verification.

**5. Security:**

* **Authentication & Authorization:**
  + **JWT (JSON Web Tokens)** for secure and stateless authentication.
  + Middleware to protect sensitive routes and ensure proper role-based access.
* **Data Protection:**
  + HTTPS enforced using SSL certificates.
  + User password hashing with **bcrypt**.
* **Input Validation:**
  + Sanitization and validation on both frontend and backend to prevent injection attacks.

**6. UI/UX Design Principles:**

* **Minimalist and Task-Focused Design:**
  + Clean interface with soft colors and readable fonts.
  + Action-based dashboard where users can view, apply, or post gigs easily.
* **Role-Based Views:**
  + **Freelancers:** Skill profile, project applications, earnings, reviews.
  + **Clients:** Post job, track applicants, manage ongoing work.
  + **Admin (optional):** Platform moderation, user flagging, reports.
* **Mobile Responsiveness:**
  + All views are responsive and optimized for touch-based navigation.

**Future Scalability Considerations:**

* **Microservices Readiness:**  
  Current modular structure allows future transition to microservices architecture.
* **CI/CD Pipeline Support:**  
  Can be integrated with platforms like GitHub Actions or Jenkins for continuous deployment.
* **Cloud Hosting Support:**  
  Compatible with platforms like Vercel/Netlify (frontend) and Render/Heroku/AWS (backend).