SCHEDULE B – STATEMENT OF WORKS

Statement of Work –Backend Development

Overview Form

Parties:

1. **Mrs. Seema Rani**, Resident of Suratgaria Bazar, Sirsa – 125055, India (Service Receiver)
2. **Mr. Abhishek Arora**, Resident of Near Jain Hospital, Ward no 8, Raman Mandi, Ramsara – 151301, India (Supplier)

CONTACT DETAILS AND KEY TERMS:

Agreement	This SOW is entered as part of MSA agreed between the parties.
Correspondence address	**Service Receiver:** Suratgaria Bazar, Sirsa – 125055, India **Supplier:** Near Jain Hospital, Ward no 8, Raman Mandi, Ramsara – 151301, India
Email (and for service of notices and communication)	**Service Receiver:** cagmonga0826@gmail.com **Supplier:** abhishekarora5437@gmail.com
Mobile Numbers	**Service Receiver:** +91 9813860916 **Supplier:** +91 9464647327
Project Commencement Date	12th July 2025
Project Target Completion Date	7th August 2025

1. **SERVICES:**
   a. **Development of backend using Express JS and Node JS.**
   b. **RESTful API development for interactions with frontend and Office 365.**
   c. **Schema design, Indexing, optimizations, and integration of PostgreSQL database with backend.**

2. **FEES:**
   a. **Total fees based on the services as detailed in Statement of work –**
      i. Four Thousand Five Hundred Only (₹4,500)
      ii. Currency – INR

IN WITNESS WHEREOF, the Parties have executed this Statement of Work dated 28th June 2025.

For the Service Receiver

Name: Mrs. Seema Rani

Date: June, 2025

For The Supplier

Abhishek Arora

Name: Mr. Abhishek Arora

Date: 2 July , 2025

Article 1 : Statement of Work

1. **Scope of Service:** The Service Receiver requires a web portal to present her company to the public and to manage client contracts, either personally or through her staff. The Supplier, Mr. Abhishek Arora, will develop the web portal for the Service Receiver. The Supplier shall use Node JS and Express JS to develop the backend of the web portal, and he shall use PostgreSQL as a database for the web portal.

   The requirements include development of business logic, authentication and authorization logic, RESTful API development for frontend and office 365, Database schema creation, Database indexing, Database optimizations and integration with business logic.

2. **Part of Master Supply of Service Agreement (MSA):** This Statement of Work is considered as part of MSA and is governed by the clauses of MSA and Non-Disclosure Agreement (NDA) signed between parties. The meaning of any word or expression shall be as defined in the Master Service Agreement (MSA), unless otherwise specified in this Statement of Work (SOW).

3. **Service and Deliverables:**

   3.1 Service Deliverables:
   - Database Schema, Integration, and Optimization files.
   - Source code of backend under version control (Git).

   3.2 Deliverables will be considered as delivered once Service Receiver approves via the designated email thread.

   3.3 Backend Development:
   - **Authentication and Authorization - Clients:** The requirements for authentication and authorization shall include the following:
     - When a client registers through the website, they shall be asked to provide their mobile number and email address and CAPTCHA. Once submitted, a unique OTP shall be sent to both the mobile number and email address.
     - After the user enters both OTPs received separately, the system shall verify them. Upon successful verification, an authentication token shall be generated. The user will then be prompted to enter basic details such as name, contact information, address, etc.
     - Upon successful registration, the user shall be allowed to create a password of their choice, entered twice to ensure accuracy. There must be password strength requirements such as (Minimum 8 characters, one Capital, One lowercase Letter, One number and one symbol). Once the password is successfully created, the user shall be redirected to the login page.
     - The user can then log in using their email address as the user ID and the password created during the registration process and CAPTCHA.
     - Once the user logs in, the system shall verify the credentials (user ID and password) and direct the user to the dashboard.
     - The authentication token shall be generated with consideration for role-based access controls specific to the user.

Abhishek Arora

    - Clients can change their password by visiting the password change page through their profile after logging in, or by clicking "Forgot Password" on the login page. OTP verification is required for both the registered email address and mobile number.
- **Authentication and Authorization – Employees:** The requirements for authentication and authorization shall include the following:
    - Employees cannot register themselves through the web portal. Employee details and password creation shall be handled by the admin.
    - Once employees have their user ID and password, they can go to the login page and enter their credentials along with the CAPTCHA. When employees log in for the first time, OTP verification is required, similar to the client login process.
    - For subsequent logins, OTP verification is not required. Process for subsequent logins shall be same as client login.
    - The authentication token functionality will remain the same as in the client login process.
    - Employees do not have the option to change their password themselves, they must contact the admin to have their password changed.
- **Authentication and Authorization – Admin:** The requirements for authentication and authorization shall include the following:
    - Supplier Shall provide the default user id and password for admin.
    - Once admin receive the default login details, admin must reset these using OTP verification as the client password change process after logging for first time. If admin does not reset the password after first login, then admin must click forgot password option during second login (default login details can only be used once). Forgot password shall be same as client forgot password process.
    - The admin can then log in using their email address as the user ID and the password created during the above process and CAPTCHA.
    - Once the admin logs in, the system shall verify the credentials (user ID and password) and direct the user to the dashboard.
    - The authentication token shall be generated.
    - Admin can change their password by visiting the password change page through their profile after logging in, or by clicking "Forgot Password" on the login page. OTP verification is required for both the registered email address and mobile number along with CAPTCHA.

- **Business Logic:** The Business logic is divided into four parts as follows:
    - General features
    - Client Specific features
    - Employee Specific features
    - Admin Specific features

1. **General Features:**
    - **Profile Page -** Every user shall have a unique profile page containing various details such as name, contact information, address, PAN,

Abhishek Arora

Aadhaar, profile photo, and more (Shall allow custom fields). Additionally, the user shall have option to change profile details using password except contact details (Email and Mobile number) which can only be changed after OTP Verification. The user can view their own activity log, including logins, file views, comment actions, and more.

- **Dashboard –** Every user shall have a unique Dashboard that shows various counters and five most recent contacts related to that user, and it shall allow user to filter and sort contracts on the basis of Date entered (Single Date or Range), Period(Term) of contract, Contract Value, Contract Status or any custom field. Charts showing contract distribution by type, value, status, or any custom field.
- **List of Contracts Page –** The user shall be able to view all assigned contracts and shall have the ability to filter and sort them based on the date entered (single date or range), contract period (term), contract value, contract status, or any custom field. It shall also allow to sort in ascending or descending sorting.
- **Contract Details Page –** User shall be able to view all details about the contract like name of the parties, Commencement Date, Termination Date, Contract Value, Linked Contracts (A Contract can be linked to any number of other contracts), Contract Status(Draft, In Review, Reviewed, Pending for Client Review, Client Reviewed, Final Reviewed – Sign Pending, Final Reviewed – Signed, Modified – in Review , Modified – Reviewed, Modified – Pending for Client Review, Modified – Client Reviewed, Modified – Sign Pending, Modified – Signed), Term of the Contract, Brief Summary of Contract, Custom Fields, All Comments, Deliverables, Comments History, Edit History, Option to View/Edit contract in Word/Excel/PDF format either in Office Online or Desktop Office Applications (Detailed out in "Statement of Work – Microsoft Office 365 Integration"). Client Details shall be hidden unless user has the permission to view the details. User can compare two versions of same contract and comparison tool shall automatically highlight the differences. It shall also have a search bar to search in agreement file.
- **Global Search Bar:** Universal search across contracts, clients, employees using fizzy matching as per role based access controls (powered via frontend logic and API).

2. **Client Specific Features:**

- **Profile Page –** In addition to general features, clients shall have the option to change their password, as detailed in the Authentication and Authorization section above. The system shall also display the edit history of changes made in the profile section.
- **Dashboard –** In addition to general features, the client shall be able to add more than five contracts to the dashboard, up to a maximum of twenty.
- **List of Contracts Page –** In addition to general features, the client shall be able to sort contracts by the date of the latest comment added

Abhishek Arora

and by the number of comments. And it shall provide an option to view or download the contract as a PDF. Client can also request a new contract via a special "contract request form" from the company. The form shall go the dashboard of admin or designated employee as per Role-Based Access Controls. Not all status is visible to client, Contract is only visible to client when assigned to client.

- **Contract Detail Page -** In Addition to general features, The client shall only be able to see the comments assigned to him and history of assigned comments from the date of assignment to the date of revocation of assignment, not edit history of the contract. Client cannot edit any contract.

  Additionally, It displays the contract PDF and allows the client to add comments to any line within the document. Each comment shall include the following details:

  - Timestamp (date and time of the comment).
  - Client name (user ID).
  - Comment description.
  - Option to mark the comment as resolved.
  - Options to edit or delete the comment.
- **Comments -** Client can add comments and it is automatically assigned to employees to whom the contract was assigned as per Role Based Access Controls. Client cannot determine to whom the comment should be assigned. Comments can be replied in threaded format for better discussion.

3. Employee Specific Features:

- **Profile Page -** In addition to general features, employees shall have the option to request a password change, as detailed in the Authentication and Authorization section above. The system shall also display the edit history of changes made to the profile section. The employee cannot change contact details by themselves, employee must make a request to admin for change of contact details.
- **Dashboard -** In addition to general features, employees shall have access to a counter that displays pending tasks—such as responding to client, employee, or admin comments, adding, modifying, or deleting contracts—related to various agreements. It also includes link to task management portal page, which allows employees to view and manage assigned tasks (e.g., pending comments, contract edits).
- **Comments –** Employees can add comments and assign them to other employees, administrators, or clients related to a given contract, based on Role-Based Access Controls. Comments can be replied in threaded format for better discussion. Employees can also add private notes to contracts visible only to employees/admins as per Role Based Access Controls.
- **List of Contracts Page –** In addition to general features, employees shall have the option to filter contracts by individual clients. They may

Abhishek Arora

also add or delete contracts if assigned by the administrator, in accordance with Role-Based Access Controls. Client Request for new contracts shall be visible here, as per Role-Based Access Controls.

- **Contract Detail Page –** In addition to general features, employees can assign comments to clients or other employees, based on Role-Based Access Controls.
- **User Management Page –** Shall be as per "Statement of Work – Role-Based Access Control (RBAC)".

4. Admin Specific Features:

- **Profile Page –** In addition to general features, the administrator shall have the option to change their password, as detailed in the Authentication and Authorization section above. The system shall also display the edit history of changes made to the profile section.
- **Dashboard –** In addition to general features, it shall display a counter indicating the total number of contracts the company holds. It will also include multiple counters categorizing contracts by client, employee, status (open/in progress), and status (closed—either delivered or cancelled). Additionally, it shall feature lists of the five most recent contracts, sorted by date, contract value, and contract duration.
- **List of Contracts Page –** In addition to general features, the administrator can filter contracts by individual clients and employees. Client Request for new contracts shall be visible here.
- **Contract Detail Page –** In addition to general features, the administrator can add comments and assign them to anyone.
- **New Contact Page –** The administrator shall have access to create new contracts by adding various details and linking Word or Excel files through Microsoft 365 integration.
- **User Management Page –** Shall be as per "Statement of Work – Role-Based Access Control (RBAC)".

- **Database:**
  - The database schema shall be designed strictly in accordance with the defined business logic and functional requirements, ensuring logical data modeling and referential integrity.
  - All database queries shall be optimized for performance, including the use of efficient joins, indexing strategies, and query execution plans to minimize latency and resource consumption.
  - The database shall implement comprehensive indexing mechanisms (e.g., primary, composite, and full-text indexes) to support high-performance data retrieval and scalability under production workloads.

  The database shall implement comprehensive indexing mechanisms (e.g., primary, composite, and full-text indexes) to support high-performance data retrieval and scalability under production workloads.

3.4 User Guides - Standard user guides will be provided covering core functionality.

Abhishek Arora

3.5 Version Control – Within seven (7) calendar days from the execution date of this Agreement, the Service Receiver shall establish a secure and private GitHub repository designated exclusively for the storage and management of all project-related digital assets, including but not limited to backend source code, technical documentation, and user manuals. The Service Receiver shall provision access to the Supplier with appropriate permissions to facilitate collaboration and contribution.

- All aforementioned assets shall be version-controlled and maintained solely within the designated GitHub repository. The storage, duplication, or transmission of any project-related files or intellectual property on alternative platforms, repositories, local machines, or third-party systems—whether cloud-based or on-premises—shall constitute a material breach of the Non-Disclosure Agreement (NDA) and may result in immediate legal recourse.
- The repository shall adhere to semantic versioning standards (e.g., v1.0.0), and comprehensive changelogs shall be maintained to document all modifications, enhancements, and bug fixes. The Service Receiver shall unilaterally define and enforce the branching model (e.g., Git Flow, trunk-based development) and commit message conventions (e.g., Conventional Commits) at the time of repository initialization. These governance protocols shall be deemed final, binding, and non-negotiable for the duration of the engagement.

3.6 Disaster Recovery and Backup

- The Supplier shall implement and maintain a disaster recovery and backup plan throughout the duration of the project. This plan shall include the following:
  - **Repository-Based Backup:** All project files—including source code, database schemas, database and project configuration files, and documentation—must be committed and pushed to the GitHub repository designated and owned by the Service Receiver. No files shall be stored on any other platform, device, or repository unless repository is initialized and its communicated to Supplier.
  - **Daily Commit Requirement:** The Supplier shall commit and push all changes to the GitHub repository at the end of each working day. This ensures that the latest version is always recoverable from the repository.
  - **No Local Storage:** Storing any project-related files on local machines, personal cloud storage, or third-party platforms is strictly prohibited and shall constitute a material breach of the Non-Disclosure Agreement (NDA).
  - **Recovery Protocol:** In the event of accidental deletion, corruption, or system failure, the Supplier shall restore the latest working version from the GitHub repository within 24 hours and notify the Service Receiver immediately.
  - **Verification:** The Service Receiver reserves the right to periodically verify the integrity and completeness of the repository and may request a demonstration of recovery procedures.
  - **Final Backup:** Upon project completion or early termination, the Supplier shall ensure that the final version of all deliverables is fully committed and pushed to the GitHub repository. A written confirmation of this action shall be submitted via the designated email thread.

3.7 In-tool walkthroughs - The Supplier shall implement embedded, context-sensitive walkthrough guides within the application interface to facilitate end-user onboarding and operational efficiency. These interactive guides shall be designed to provide real-time, role-specific instructional support, covering the core functional domains of the system, including but not limited to: Capture, Review and Approval, and Analytics workflows.

- The walkthroughs shall serve as a self-service enablement mechanism, offering procedural overviews, task-specific instructions, and resolution pathways for frequently encountered user queries. The implementation shall ensure that the guides are seamlessly integrated into the user interface, accessible without external dependencies, and maintained in alignment with system updates to preserve instructional accuracy and relevance.
- Failure to provide or maintain such walkthroughs in accordance with the agreed specifications shall be deemed as substandard work under the clause 13 of the MSA.

3.8 New features and enhancements:

- The Supplier shall, where necessary, implement additional features or enhancements required to ensure the seamless execution and continuity of business logic workflows. Such features may also be introduced in accordance with the Role-Based Access Controls (RBAC) Implementation Statement of Work, the Microsoft 365 Integration Statement of Work, or the UI/UX Design and Frontend Development Statement of Work, subject to prior written approval from the Service Receiver via the designated project email thread.
- All such additions shall be delivered at no additional cost to the Service Receiver, provided they fall within the functional intent and architectural boundaries of the aforementioned Statements of Work. The Service Receiver shall retain sole and exclusive authority to determine whether a specific feature, page, or UI element is within the contractual scope. Such determinations shall be final, binding, and non-negotiable, and shall not be subject to dispute, escalation, or renegotiation.

3.9 Bugs/Fixed Post Delivery Period:

Pursuant to Clause 25 of the Master Services Agreement (MSA), the Supplier shall provide comprehensive post-development support for a period of one hundred and twenty (120) calendar days commencing from the date of final delivery acceptance. During this support window, the Service Receiver shall be entitled to report any software defects, malfunctions, or performance anomalies (collectively, “Issues”) via the GitHub repository designated for the project.

- Each Issue shall be formally logged as a GitHub Issue entry and shall include sufficient detail to enable reproducibility and diagnosis. The Supplier shall be obligated to remediate:
  - Critical Issues (i.e., those resulting in system unavailability, data loss, or major functional breakdowns) within twenty-four (24) hours of logging, and
  - Non-critical Issues within forty-eight (48) hours of logging.
- Upon resolution, the Supplier shall notify the Service Receiver by posting a comment on the corresponding GitHub Issue and concurrently via the designated email thread. An Issue shall be deemed officially resolved only upon:

Abhishek Arora

  - Written confirmation of resolution by the Service Receiver via the designated email thread, and
  - Formal closure of the GitHub Issue marked as "Resolved" by Service Receiver.

3.10 Database Schema Approval:

- The Supplier shall have the technical discretion to design the database schema in accordance with the functional, performance, and scalability requirements specified in the project documentation. This includes, but is not limited to, the definition of entities, relationships, data types, indexing strategies, normalization levels, and referential integrity constraints.
- However, prior to the commencement of any backend development activities, the complete database schema must be formally submitted for review and approved in writing by the Service Receiver via the designated project email thread. This approval shall serve as a mandatory precondition for initiating any backend implementation, integration, or data-layer development work.
- The Service Receiver's approval shall be considered final, binding, and non-negotiable, and any backend development initiated without such approval shall be deemed a material breach of the Agreement.

3.11 Testing and Q/A:

- The Supplier shall be responsible for the development and maintenance of comprehensive test cases that provide full functional coverage of all features, modules, and workflows defined within the scope of the project. These test cases shall include, but are not limited to, unit tests, stress/load testing, integration tests, and end-to-end tests, and shall be traceable to specific functional requirements.
- Furthermore, all backend source code shall be developed in strict adherence to the OWASP Top 10 Security Guidelines, including but not limited to:
  - Input validation and sanitization to prevent injection attacks and data corruption,
  - Secure authentication and token management, including proper handling of session tokens and access credentials,
  - Mitigation of common web application vulnerabilities, such as Cross-Site Scripting (XSS), SQL Injection (SQLi), Cross-Site Request Forgery (CSRF), and Insecure Deserialization.
- The Supplier shall implement appropriate security controls, conduct regular code reviews, and utilize automated static analysis tools to ensure compliance. Any deviation from these standards shall be considered a material non-conformance and subject to remediation at no additional cost to the Service Receiver.

3.12 Session Management:

- Each user session shall automatically expire after 20 minutes of inactivity.
- After expiry user shall be logged out and redirected to the login screen with a message that the session has ended due to inactivity.
- Login sessions shall be secured using authentication tokens (e.g., JWT or session ID's).

Abhishek Arora

- Each token shall have a fixed expiry time of 15 minutes, after which the user must log in again.
- Users shall not remain logged in permanently – tokens must expire even if browser is left open.
- When user logged out, their session must be fully ended and the token invalidated immediate.
- Users must be allowed to login from only one device at a time unless approved by the Admin on case by case basis.
- The admin shall have the ability to log out any user from the system manually in case of misuse, mistake, or emergency.
- If a user's access is changed or revoked, their session shall end immediately.
- Each login and logout shall be recorded, including user ID, login time, IP address, and device/browser used.
- These records must be available to the admin through the admin panel for the time as decided by the admin.
- If the user session involves Microsoft 365 (e.g., One drive or SharePoint upload), those sessions must also expire along with the main session.
- Access tokens granted via Microsoft 365 integration must be safely stored an revoked upon log out.
- If the session expires, logout, or token handling is found to be faulty or missing, It will be treated as substandard work and must be corrected within 72 hours.
- Repeated failure to maintain secure session handling may be treated as a material breach of contact.

3.13 Data Encryption Standards:

- The Supplier shall ensure that all sensitive data fields—specifically including, but not limited to, user passwords, contact information, and other Personally Identifiable Information (PII)—are stored using industry-standard cryptographic techniques in compliance with applicable data protection regulations and best practices.
  - Passwords shall be hashed using a strong, adaptive, one-way hashing algorithm, specifically bcrypt with a configurable cost factor to ensure resistance against brute-force and rainbow table attacks.
  - All PII and critical data stored in the database (e.g., email addresses, phone numbers, identification numbers) shall be encrypted using Advanced Encryption Standard (AES) with a 256-bit key length (AES-256) or an equivalent cryptographic algorithm that meets or exceeds current NIST standards.
- Encryption keys must be securely managed using a Key Management System (KMS) or equivalent secure storage mechanism, and must not be hardcoded or stored in plaintext within the application codebase or configuration files.
- Failure to comply with these encryption standards shall constitute a material breach of the Agreement.

Abhishek Arora

3.14 Exporting and Reporting:

- The application shall provide authorized administrative users, as defined by the Role-Based Access Controls (RBAC) framework, with the ability to export structured data sets—including but not limited to contracts, user directories, access logs, and audit trails—into machine-readable formats such as Microsoft Excel (.xlsx) and Comma-Separated Values (.csv).
- In addition, the system shall support the generation of basic analytical reports through the administrative interface, including but not limited to:
  - Monthly aggregation reports (e.g., contracts created per month),
  - Status-based distribution reports (e.g., by contract or user status),
  - Employee-specific activity reports, filtered and rendered in accordance with RBAC permissions.
- All exported data and generated reports must adhere to the system's data access policies, ensuring that users can only retrieve information for which they are explicitly authorized. The reporting module shall be designed for extensibility and must support future enhancements without requiring architectural refactoring.

4. Project Timelines:

This project will have a timeline of 3.5 weeks i.e., completed by 7th August 2025 but delivered along with Microsoft 365 and Role based Access control deliverables by 13th September 2025.

5. Documentation Deliverables: The following project and technical documentation will be provided by Supplier during the project:

- Project Plan.
- ERD (Entity Relationship Diagram) illustrating all tables, relationships, primary keys, and foreign keys.
- RESTful API Swagger spec Documentation and Guide on how to use API for each endpoint.
- Configuration specifications.
- Application architecture.
- User Guides.
- In-File Documentation.
- Database schema documentation.
- Database integration specifications.
- Dummy data for verification of functionalities.
- List of all open-source libraries used, with license type and confirmation of their suitability for commercial use.

6. Specific Instructions:

6.1 The Supplier is strictly prohibited from incorporating any software, library, framework, or third-party dependency that is governed by a restrictive license—defined as any license that imposes limitations on commercial use, redistribution, or modification. Only components licensed under permissive terms (e.g., MIT, Apache 2.0, BSD), which require attribution without imposing commercial restrictions, are permitted for use in the project.

Abhishek Arora

6.2 The Supplier shall design, develop, and deploy a custom centralized logging service to capture and persist all system-level and application-level errors, as well as user and API activity. This logging service shall record, at a minimum, the following metadata for each API-level request: timestamp, user identifier (user ID), endpoint accessed, request method, response status code, and error stack trace (if applicable).

The logging infrastructure must support real-time log aggregation, be scalable, and allow for future integration with external monitoring or SIEM (Security Information and Event Management) tools. Logs must be securely stored, tamper-proof, and retained in accordance with the Service Receiver's data retention policies.

6.3 The application shall be architected such that users assigned the Administrator role under the Role-Based Access Control (RBAC) framework are granted unrestricted access to all features, modules, configurations, and datasets within the web portal. This includes, but is not limited to, access to user management, system settings, audit logs, reporting tools, and module-level controls.

No functional or data-level restrictions shall be imposed on administrative users unless explicitly defined and approved in writing by the Service Receiver.

6.4 Supplier shall prepare a detailed analysis statement of all service providers for implementation of OTP functionality that should include details about service providers, pricing, limits, various plans available. And Once Service Receiver approves in writing via designated email thread from Service Provider then only Supplier integrate that service into web portal.

6.5 The Supplier shall implement a robust rate-limiting mechanism for all One-Time Password (OTP) requests initiated for email and mobile number verification. Each user shall be restricted to a maximum of five (5) OTP requests per hour, per verification channel (email and mobile), to prevent abuse and ensure system integrity.

In addition, the system shall incorporate progressive back-off algorithms to dynamically increase the delay between subsequent OTP requests upon detection of repeated or suspicious activity. This mechanism shall be designed to mitigate brute-force attempts, automated abuse, and denial-of-service vectors, while maintaining usability for legitimate users.

All OTP request attempts, throttling events, and abuse patterns shall be logged and made available for administrative review and audit. All API endpoints shall implement rate limiting (e.g., via middleware) to protect against brute force attacks and abuse. Specific thresholds will be agreed upon in writing via the Designated email thread (100 requests/minute/user by default).

6.6 All backend API endpoints shall be engineered and optimized to meet defined performance benchmarks. Specifically, 90% of all API responses—measured over any rolling 24-hour period—must be returned within 300 milliseconds (ms) under normal operating conditions. Performance metrics shall be continuously monitored using automated observability tools, and any sustained deviation from this threshold shall be treated as a performance degradation event subject to remediation.

Abhishek Arora

6.7 Upon completion of backend development and database design and configuration, the Service Receiver shall conduct User Acceptance Testing (UAT). The backend shall not be deemed approved or production-ready until the Service Receiver issues written confirmation via the project email thread.

6.8 The backend infrastructure shall be architected to support a minimum of 1,000 concurrent client sessions without degradation in performance, availability, or data integrity. Additionally, the system must be capable of storing, indexing, and retrieving at least 10,000 contract documents, with efficient query performance and minimal latency.

The Supplier shall ensure that the backend is horizontally scalable and stress-tested under simulated load conditions to validate compliance with these requirements. Failure to meet these scalability thresholds shall constitute a material deficiency in system design.

IN WITNESS WHEREOF, the Parties have executed this Master Service Agreement dated 28th June 2025.

For the Service Receiver	**For The Supplier**
	Abhishek Arora
Name: Mrs. Seema Rani	**Name: Mr. Abhishek Arora**
Date: June, 2025	**Date:** 2 July , 2025