Technical Specifications

The offered software system should be scalable to support unlimited users, including all stakeholders of NTPC, its subsidiaries, and JVs.

The offered solution to meet the following minimum capabilities:

1. **Authentication System Specifications:**
2. User Types and Authentication Requirements  
   The application will be accessible by both internal and external users. The authentication system must be flexible enough to support diverse authentication mechanisms for different user types.
3. Pluggable Authentication Support  
   The system should support a pluggable authentication architecture that can integrate with various authentication systems, including but not limited to:
   * LDAP (Lightweight Directory Access Protocol)
   * Database-based authentication
   * X.509 certificates
   * SPNEGO (Simple and Protected GSS-API Negotiation Mechanism)
   * JAAS (Java Authentication and Authorization Service)
   * JWT (JSON Web Tokens)
4. Support for Multiple Authentication Protocols  
   The system must support multiple authentication protocols, ensuring compatibility with widely adopted standards such as:
   * SAML v1 and v2 (Security Assertion Markup Language)
   * WS-Federation
   * OAuth2 (Open Authorization)
   * OpenID Connect
   * REST-based authentication methods
   * CAS (Central Authentication Service)
5. Multifactor Authentication Support  
   The system should support multifactor authentication (MFA) through a variety of methods, which can include:
   * Mobile-based authenticator apps (e.g., Google Authenticator)
   * The application shall necessarily support Microsoft MFA as it is already in use at NTPC.
   * Email-based verification
   * SMS-based verification
6. Real-Time Monitoring and Metrics  
   The authentication system should provide capabilities for real-time monitoring, tracking system behavior, and generating statistics and performance metrics to ensure effective operation and proactive issue resolution.
7. Attribute Release and User Consent  
   The system should allow for the release of user attributes in compliance with regulatory and organizational policies. Additionally, it should support obtaining and managing user consent as needed.
8. Built-In Support for User Management  
   The system should provide built-in functionality for managing user accounts, including password management, notifications, terms of use management, and the ability to support impersonation when required for administrative purposes.
9. Delegated Authentication Support  
   The authentication system should allow for delegation to external identity providers, supporting integrations with widely used solutions such as:
   * ADFS (Active Directory Federation Services)
   * Google/Gmail authentication
   * OIDC (OpenID Connect) providers like PING, Okta, etc.
10. Audit and Log Management  
    The system should offer centralized management of audit logs and user activity logs, with the ability to publish this data to downstream systems such as SIEM or XDR for further analysis or integration.
11. Client Application Registration and Authentication Policies  
    The system should allow for managing and registering client applications and services. It should provide the ability to define and enforce specific authentication policies tailored to different application and service requirements.
12. **Search Schema Design:**
13. **Flexible Schema Support:** The system should support a flexible, schema-less design that allows indexing of a variety of data types without requiring predefined structures.
14. **Configurable Tokenization:** The system should offer configurable options for text processing, such as tokenization, to enable customization of how text data is broken down and indexed for search.
15. **Dynamic Field Addition:** The system should allow for the dynamic addition of new fields for indexing, so that new data types can be indexed and made searchable easily, based on field names.
16. **Content Analysis Flexibility:** The system should provide the ability to analyze and categorize content in multiple ways, depending on user requirements, ensuring flexibility for diverse use cases.
17. **External Configuration Support:** The system should allow for external configuration of resources such as stopword lists, synonym lists, and protected word lists. These configurations should support text processing techniques such as word splitting, regular expressions, and stemming to improve search relevancy and efficiency.
18. **REST-Based Query Interface Specifications:**
19. **RESTful Query Interface**  
    The application must provide a RESTful interface to query the indexed data. This interface should follow standard REST conventions and allow users to interact with the system via simple HTTP requests.
20. **Sorting Capabilities**  
    The query interface must support sorting of results by multiple fields, including complex functions for numeric fields. This allows users to customize the sorting order based on their requirements.
21. **Handling Missing Values**  
    The system should be configurable to treat documents with missing values by ensuring that they are sorted last in the result set. This ensures that incomplete records do not interfere with the primary results.
22. **Relevance-Based Results**  
    The application should provide relevant results based on the indexed data, ensuring that the search results are ranked and displayed in order of relevance, considering factors like content, metadata, and user query context.
23. **Advanced Query Parsing**  
    The query parser should support a variety of search types to accommodate different search requirements. This includes, but is not limited to, boolean queries, phrase searches, term-based searches, numeric filters, and fielded searches (where specific fields are targeted for querying).
24. **Custom Query Building**  
    The query interface should allow users to build custom queries using logical operators (AND, OR, NOT) and to structure queries using parentheses to nest expressions, enabling more complex and precise searches.
25. **Multiple Scoring and Similarity Models**  
    The system should support multiple scoring or similarity models for ranking search results, providing flexibility in fine-tuning the relevance of results based on different algorithms or criteria. This allows easy adjustment of how search results are prioritized.
26. **Near Real-Time Search (NRT)**  
    The application should support Near Real-Time (NRT) search, meaning that any additions, updates, or deletions of documents should be reflected in the search results almost immediately. This ensures that the system remains up-to-date with minimal latency.
27. **Searcher Grouping and Data Aggregation Features:**
28. Data Grouping:
    * The search system should allow for grouping and organizing search results in multiple ways. Users should be able to group data based on one or more fields and customize the groupings to meet their specific analysis needs.
    * Grouping should be dynamic, allowing for the grouping of data at the time of search and throughout the analysis process.
29. Grouping by Terms:
    * The system should support grouping search results by terms or values from any data field, such as keywords, tags, or other specific attributes (e.g., category, product type).
    * Users should be able to define the terms or values they want to group by, and the system should efficiently categorize the results based on these terms.
30. Grouping by Query:
    * The system should allow for grouping data based on specific queries or search conditions that have been applied. This could include grouping based on search filters, specific text-based queries, or logical conditions such as "AND", "OR", or "NOT".
    * Users should be able to combine multiple search criteria to group results based on their custom queries.
31. Grouping by Range:
    * The system should support grouping search results by numeric or continuous ranges (e.g., price ranges, age ranges, or any measurable data).
    * Users should be able to define the range intervals (e.g., "0-100", "101-200", etc.) for grouping and the system should organize the data accordingly.
32. Grouping by Date:
    * The system should allow grouping search results by date or time intervals (e.g., day, week, month, quarter, year).
    * Users should be able to define custom date ranges for grouping, or the system can automatically provide pre-configured options (e.g., "last week", "this month", "last year").
    * This feature is particularly useful for tracking trends and patterns over time.
33. Pivot Table-Based Grouping:
    * The system should provide functionality to create pivot tables from search results, allowing users to perform multidimensional data analysis.
    * Pivot tables should allow users to "drag and drop" data fields to dynamically organize and aggregate search results.
    * Users should be able to summarize data based on categories or fields (e.g., by product, by region, by time) and use different aggregation methods (e.g., sum, count, average).
34. Aggregation of Grouped Data:
    * For each group, the system should allow data aggregation (e.g., sum, average, count, min, max) to summarize and analyze the grouped information.
    * Aggregation should be flexible, with the ability to define custom aggregation rules based on user needs.
    * The system should handle large data volumes efficiently and ensure accurate aggregation in real-time or near real-time.
35. Multi-Field Grouping:
    * The search system should allow grouping of results based on multiple fields simultaneously (e.g., grouping by both "Category" and "Date").
    * This functionality should enable cross-analysis of different data attributes, providing users with a deeper understanding of relationships between different variables in the data.
36. Custom Grouping Rules:
    * The system should allow users to define custom grouping rules, so they can tailor the grouping process to specific use cases or business requirements.
    * Custom rules could involve grouping based on data thresholds, ranges, or complex business logic defined by the user.
37. Interactive Grouping Interface:
    * The system should provide an easy-to-use interface for defining, adjusting, and viewing groupings. This interface should be interactive, allowing users to drag and drop fields, select criteria, and visualize the grouped data in real-time.
    * Grouped data should be displayed in a clear, organized manner, such as in tables, graphs, or charts, depending on user preferences.
38. Real-Time or Near Real-Time Grouping:
    * The system should allow for the grouping and aggregation of data in real-time or near real-time, ensuring that users can work with the most up-to-date information without significant delays.
    * As new data is added or updated, groupings and aggregations should reflect these changes immediately.
39. Export and Reporting:
    * The grouped and aggregated data should be exportable in various formats (e.g., CSV, Excel, PDF, or image formats) for reporting and further analysis.
    * The system should support the creation of reports based on grouped data, and these reports should be customizable according to the user's preferences.
40. **Auto-Complete and Auto-Suggest Features**
41. Auto-Complete and Auto-Suggest Functionality:
    * The system should provide real-time auto-complete and auto-suggest capabilities as users type their queries.
    * The auto-complete feature should display a list of suggested terms or phrases that match the user's input, helping users select or complete their queries quickly.
    * The system should offer suggestions based on popular search terms, previous queries, or frequently used terms in the data set.
    * The suggestions should be dynamically updated as the user continues typing, providing the most relevant options at every stage of their input.
42. Spelling Suggestions:
    * The system should automatically detect and provide spelling suggestions to help users correct any misspelled terms or phrases in their query.
    * When a user enters a word that is not in the system's dictionary or matches no results, the system should offer alternative spellings or terms based on proximity or known patterns.
    * The system should be able to recognize common typing errors, such as misplaced characters or missing letters, and suggest the correct term(s).
    * Spelling suggestions should be offered as clickable alternatives, allowing users to easily choose the intended term.
43. Configurable Hit Highlighting:
    * The system should support configurable hit highlighting to emphasize where search terms appear in the results.
    * Matches within the results (whether in document text, titles, or other fields) should be highlighted in a manner that helps users quickly locate relevant information.
    * The highlight configuration should be customizable, allowing users or administrators to adjust the color, style (bold, italic, underline), and prominence of the highlighted terms.
    * Hit highlighting should apply not only to the main query but also to any suggestions, filters, or facets presented in the system.
44. **Searcher Mathematical Operations Inside the Query**

Support for Complex Mathematical Analysis:

* The system should support advanced mathematical operations within search queries, allowing users to perform calculations or analysis on result sets as part of their search queries.
* The system should enable data aggregation, such as sums, averages, and weighted values, directly in the search or query processing layer, enabling advanced data analysis and business intelligence on the retrieved data.
* The capability should allow for combining search results with mathematical operations (e.g., filtering, grouping) in a seamless and efficient manner.
* The system should support the use of common mathematical functions such as addition, subtraction, multiplication, division, exponential, logarithmic, and trigonometric functions as part of query processing.

Support for Statistical Computations:

* The system should include a statistical processing component capable of performing the following calculations on result sets or data groups:
  + Min: Compute the minimum value of a specified numeric field or result set.
  + Max: Compute the maximum value of a specified numeric field or result set.
  + Sum: Compute the total sum of a specified numeric field across a set of results.
  + Mean (Average): Compute the mean or average value of a numeric field across the search results.
  + Distinct Count: Compute the count of unique values of a specific field across a dataset (i.e., deduplicating data).
  + Median: Calculate the median value of a given numeric field (if applicable).
  + Standard Deviation and Variance: Calculate statistical dispersion (spread) of numeric data.
* The system should support aggregation operations that allow for grouping search results based on specific fields (e.g., group by date, category, or region) and applying the statistical functions within those groups.

1. **Search Inside Attachments**

The following file types should be supported for full content search:

1. HyperText Markup Language
2. XML and derived formats
3. Microsoft Office document formats
4. OpenDocument Format
5. iWorks document formats
6. Portable Document Format
7. Rich Text Format
8. Compression and packaging formats
9. Text formats
10. Audio formats
11. Image formats
12. Video formats
13. Mail formats
14. CAD formats
15. Font formats
16. Scientific formats
17. Crypto formats
18. Database formats
19. Natural Language Processing
20. Image and Video object recognition
21. **Search Performance Reporting**
22. Extensive reporting and control web-based interfaces should be offered
23. The web-interface should allow for insight into search cores, shards etc
24. Should allow to debug data and queries with web-based interfaces for analysis, index data exploration
25. Allow Instant review of key performance metrics in the web-browser
26. **Business Intelligence and Reporting to users and management**
27. The application should offer a report designer to allow DBA to create rport templates.
28. The report should be able to fetch the data from many data sources, including flat files, MySQL databases, XML, and web services.
29. The reporting engine should be able to create Joint Data Sets after combining data from different data sources and use them to create texual [Tables/Pivot Charts]and visual reports[pie charts, bar charts etc].
30. The reporting engine should allow DBA to modify the data using Scripts and implement custom logic, communicate with other objects, and access data within an application.
31. The engines Charting engine should offer a number of types of charts and styles, headers, footers, page breaks and margins.
32. The application should offer customized reports as per requirements of business and these may include per employee reports, group of employee reports or complete department or organization reports.
33. All the reports should be available for viewing of the web and export of these reports into multiple formats as XLSX, PDF, DOCX, PPTX etc
34. The application should allow to schedule these reports for automatic execution and circulation via email as per defined schedule and as per formats defined in parameters. These reports can be per user or per organization or as required by the management/Business.
35. The scheduler should comply with standard linux cron expressions for defined schedules.
36. **Email notifications:**

I.OutGoing email notifications

1. For all correspondences as well as actions taken - there should be email notifications.
2. The notifications for each individual should be created complying with the role-based access control privileges that are defined in the application for the concerned individual.
3. The email notifications should be relayed by the SMTP Service.
4. The message-ids of the email email relayed - should be also collected and stored against the action in the concerned fields/database tables.
5. The attachments that are enclosed by the users - should not have any size upper limit. The user can attach any number of attachments of any size in any record.
6. At the time of email relays to the concerned users - whether the attachments need to be sent as an attached file or as a link should be decided as per permissions configured.
7. If the total size of the attachments enclosed is above SMTP relay limit - the attachment should be converted to the hyperlinks.
8. The hyperlinks can also be for anonymous download of attachments or should allow download via authenticated session as per their role based access control permissions.
9. All record view as well as attachment view/download actions should be recorded in the audit-trail.
10. The correspondence letter that is generated should also be converted into PDF file and its MD5 hash signature should be also pushed along with the outgoing email notifications. MD5 hash should also be sent for all enclosed attachments in the concerned action in the outgoing email notifications.
11. The application should also allow for verification of the MD5 hash for all file types enclosed.

II.Incoming email notifications

1. The application should be able to download and process the emails - whether new emails or reply to the notifications sent or system generated emails as bounce or error messages / meeting requests or confirmations as sent by the other email servers.
2. The complete original email should be also available in .eml format against the concerned action that is created based on the received emails.
3. If the incoming emails also include people other than authorized by the protocols - such actions should also be recorded in UI. The critical emails details as Message-Id, thread-Id should be also captured in the application.
4. **M365 Email Processing Specifications**

The **email processing feature** should enable seamless integration with external email platforms, such as **M365** (Microsoft 365), ensuring that all relevant project-related emails are automatically fetched and recorded for correspondence tracking. This feature should adhere to the following specifications:

#### 1. **Automatic Email Fetching and Recording**:

* The system should be capable of **automatically fetching and recording emails** that are **sent** or **received** by authorized users involved in the project.
* **Emails to or from internal users** within the organization should be **excluded** from the processing to ensure that only relevant external communications are captured.
* The captured emails should be **automatically categorized and tagged** based on the project and the user’s involvement.

#### 2. **Integration with M365 Platform**:

* If permission is granted, the system should be able to **integrate with M365** (or similar platforms) to fetch email data, using standard protocols or APIs available for email processing.
* The system should be **compatible with M365 email services**, such as **Outlook** or Exchange, allowing it to retrieve email messages in real time or on a scheduled basis.
* The email data should be processed and imported in a structured manner, ensuring that key metadata such as **sender**, **recipient**, **date/time**, **subject**, and **attachments** are captured.

#### 3. **Capture All Relevant Correspondences**:

* The main objective of this integration is to ensure that **100% of all relevant correspondences** (external communications) are captured in the application.
* The system should ensure that **no emails** from internal organizational communications are processed or stored unless explicitly specified by the configuration settings.
* The integration should automatically handle **new, incoming emails**, as well as **sent emails** from the users' accounts.
* The system should also **capture email attachments** (if any) along with the email content, ensuring that all relevant information is available for later reference.

#### 4. **Permissions Management**:

* **Permission-based access** should be in place to ensure that only authorized users can enable or disable the email fetching functionality.
* Permissions should be configurable to allow or restrict fetching for specific users based on their roles or involvement in the project.
* The system should allow for flexible user management, ensuring that only relevant project participants’ emails are fetched, and **internal organization emails** (non-project-specific) are excluded.

#### 5. **Data Synchronization and Timeliness**:

* The email processing integration should allow for **real-time or near-real-time synchronization** of emails.
* The system should be capable of fetching new emails periodically, ensuring that no critical project communication is missed.
* The system should provide **timely updates** for any new correspondences or responses related to the ongoing project.

1. **Correspondence Reports required**
2. incoming register
3. outgoing register
4. overdue correspondences register
5. User activity report
6. Any other report as required by the user/business users.

Leverage **Large Language Models (LLMs)** to automatically categorize and classify incoming correspondence based on content, enabling efficient routing, prioritization, and streamlined management of communications.

1. **Backup & Replication:**

The vendor shall configure the backup and replication mechanisms to ensure that there is no data loss or theft. NTPC will provide the necessary hardware for backup operations. During disaster recovery exercises or drills, the vendor shall extend full support to NTPC to ensure seamless recovery and continuity.

1. **Cyber Security Requirements:** 
   1. **Application and Server Security:**
      1. Secure by design and Secure by default principles shall be used to design and develop the applications.
      2. Source code shall be audited for any vulnerabilities during the development.
      3. Server/OS/Middleware/API/Web-Services or any other such components shall also be configured securely to make the complete eco-system secure.
      4. Components in scope of NTPC shall be intimated for any misconfiguration/vulnerability to NTPC by the vendor.
   2. **Security Audit:**
      1. The application/s/API/Web-services shall undergo an annual audit by a CERT-In empanelled auditor, with any identified vulnerabilities to be resolved by the vendor within 30 days. The final audit report shall be shared with NTPC. Critical vulnerabilities identified during the audit must be resolved immediately.
      2. The application shall be audited for all the application vulnerabilities as per international standards such as OWASP, relevant ISO 27002 controls etc.
   3. **Data Security and privacy protection**:
      1. All the data stored, in transit or under processing shall be suitably encrypted. Reasonable security protections shall be taken to ensure that the Data is secure against any threats.
      2. The Data privacy shall also be ensured by masking the data as required and any PII’s shall be secured in such a way that compliance with DPDP Act’2023, DPDP rules and any future amendments, is always maintained during the contract period.
   4. The data shared with LLM and LLM integration with Application/Server shall also be secured against any kind of adversarial AI prompts, malicious prompt injection, LLM always poisoning etc. (ISO 42001 controls may be referred).
   5. **Response to Security Incidents/Vulnerability Reporting**:
      1. In the event of security incidents, breaches, or defacement, the vendor shall promptly deploy qualified personnel to diagnose and help NTPC resolve the issue without delay.
      2. On reporting any vulnerabilities by NTPC or any third party, the same shall be immediately investigated and relevant patches/workarounds shall be implemented and intimated to NTPC.

**Infrastructure Requirements**

**Software Requirements (In the scope of Bidder):**

* The preferred approach is to develop the web-based enterprise application using open-source operating systems and databases. However, if licensed products (Operating System or Database) are used, the bidder must provide the necessary licenses in the name of NTPC Ltd. for a minimum duration of 5 years, with no additional costs payable.

**Hardware & Storage Requirements (In the scope of NTPC):**

NTPC will provide Virtual Machines (VMs) for the configuration of the following components:

* Authentication Server
* Outgoing Email Cluster
* Database Server
* Application Server
* Storage Media/Server

The bidder is required to specify the number of VMs and their respective configurations (including CPU, RAM, and storage requirements) as part of the bid submission. Any additional hardware or infrastructure requirements, if applicable, should also be clearly outlined in the bid.

**Networking & Public IP Requirements (In the scope of NTPC):**

The bidder is required to provide a detailed specification of the networking infrastructure and public IP requirements at the time of bidding. This should include:

* Number and type of network connections
* Public IP addresses required for each component
* Network bandwidth and performance specifications
* Any specific networking configurations or protocols required for the successful deployment and integration of the system