# Software Requirements Specification

for

# Beyond QWERTY: Form Filling's Vernacular Voyage with Voice Versatility

**Version 1.0 approved**

**Prepared by**

**Riddhima Bansal**

**04.01.2025**

# Table of Contents

# Beyond QWERTY - AI-Powered Voice-Driven Form Filling

## 1. Introduction

### 1.1. Purpose

This document outlines the Software Requirements Specification (SRS) for **Beyond QWERTY - AI-Powered Voice-Driven Form Filling**. The primary goal of this project is to revolutionize the form-filling process by enabling users to complete forms using voice recognition and artificial intelligence. This innovative solution aims to enhance efficiency, accessibility, and user experience, particularly for frontline workers (FLWs) and individuals in various industries who require simplified and faster form filling. By leveraging **Azure OpenAI**, speech-to-text technologies, and real-time translation, the system provides a hands-free, time-efficient, and multilingual solution for completing web-based forms.

The document will outline the **functional** and **non-functional requirements**, as well as the overall **architecture** of the system. It will provide details on the key functionalities such as voice input processing, form auto-fill, real-time validation, multilingual support, and error handling. Additionally, this document will define the system's performance, scalability, security measures, and compliance standards to ensure it meets industry standards and user needs.

### 1.2. Document Convention

- **Actors**: Highlight all the actors the system has, where all actors in the system are written in bold and non-italic manner <**User**>.
- **Screens/Interfaces**: Highlight all the screens in the system and all screens or user interfaces are written in *italic* and in blue color.<Registration>.
- **Hardware and Software Interfaces**: Mentioned in regular text, that is non-italic manner.
- **Attributes**: Basic attributes of actors or objects are inherited from **user** actor and described in regular text.

### 1.3. Intended Audience and Reading Suggestions

The intended audience for this document includes **developers, AI researchers, business analysts, product managers, system administrators, accessibility experts, enterprise clients, and end-users such as frontline workers** who will interact with the **Beyond QWERTY - AI-Powered Voice-Driven Form Filling** system.

It is recommended that readers start with the **general descriptions** and **system overview** sections to understand the project's purpose, scope, and objectives before exploring the **specific functional and non-functional requirements**. This structured approach ensures that both technical and non-technical stakeholders can effectively grasp the system's background, architecture, and intended functionalities.

The web-based **Beyond QWERTY** solution is designed to simplify and expedite processes such as **healthcare form filling**, **bank account creation**, **digital identity applications**, and other similar workflows that traditionally rely on manual data entry.

## 1.4. Product Scope

The web-based **Beyond QWERTY - AI-Powered Voice-Driven Form Filling** system aims to enhance accessibility and efficiency in form-filling tasks by leveraging AI and voice recognition. It eliminates the need for traditional text input, making the process faster and more user-friendly. Through the use of AI-driven suggestions, voice navigation, and real-time feedback, this system is set to improve productivity, reduce human errors, and eliminate language barriers, providing a highly adaptive solution for diverse use cases.

**Benefits Include:**

- Multilingual support enables users to fill forms in multiple languages with real-time translation.
- Hands-free operation allows users to complete forms through voice commands, improving accessibility for individuals with disabilities.
- Faster data entry speeds up form-filling processes, reducing time spent on repetitive tasks.
- Context-aware AI suggestions predict and auto-fill fields based on user history and contextual input.
- Error reduction minimizes human errors through AI-powered real-time validation and correction.
- Seamless integration makes it compatible with various industries, including healthcare, banking, and government services.
- A user-friendly interface simplifies navigation and interaction, even for non-technical users.
- Secure and compliant, ensuring data privacy and adherence to regulations like GDPR and HIPAA.
- Streamlined form completion, reducing effort for users.
- Greater accessibility for individuals with disabilities or language barriers.
- Real-time validation to minimize errors and improve data accuracy.
- Automated data entry, reducing manual workload.
- Flexible integration with various platforms and industries.

## 1.5. References

- *IEEE Standard for Software Requirements Specifications*.

# 2. Overall Description

## 2.1 Product Perspective

The system is an independent, self-contained solution designed to enhance the form-filling process by leveraging voice recognition and AI. It provides a hands-free, efficient method for completing web-based forms, bypassing traditional text-based input methods. This system incorporates voice command processing, AI-powered auto-filling, real-time validation, and multilingual support, offering an innovative approach to tasks such as job applications, bank account openings, and digital identity creation.

Unlike traditional form systems that rely on manual data entry, this system eliminates the need for typing, significantly reducing human error and time spent on repetitive tasks. This system is not part of any existing form management program but replaces the outdated, manual methods of form filling, providing increased accessibility, productivity, and accuracy for both users and organizations. It offers a more secure, streamlined, and adaptive solution to form completion, making it ideal for sectors like healthcare, finance, and government.

## 2.2 Product Functions

- **Voice-based Form Filling**: Users complete forms via voice input, with speech converted into text.
- **Multilingual Support**: Real-time translation for users speaking different languages.
- **AI-driven Auto-Fill**: AI suggests or auto-fills form fields based on context and past inputs.
- **Error Detection & Validation**: Flags incorrect or incomplete form entries in real-time.
- **Data Security**: Secure storage and transmission of user data, compliant with regulations.
- **Voice Command Navigation**: Navigate forms using voice commands like "Next" and "Submit".
- **Custom Voice Commands**: Custom commands for frequent actions like "Apply for a loan".
- **Progress Tracking**: Tracks and notifies users of their form completion progress.
- **Real-Time Feedback**: Provides voice feedback to confirm entered data.
- **Offline Mode**: Allows form filling without internet, syncing once online.
- **Integration**: Integrates with external systems like CRM, banking, and government services.
- **Role-Based Access**: Ensures only authorized users can access sensitive data.

## 2.3 User Classes and Characteristics

The target users of the Automatic Form Filling System include:-

- **Frontline Workers (FLWs):** Primary users who need quick and efficient form-filling. They require a hands-free, voice-driven system to reduce workload and enhance productivity. Users may have limited technical skills and require an intuitive interface.

- **Business Professionals:** Employees in banking, healthcare, and government sectors who handle high volumes of forms. They need multilingual support, AI-powered suggestions, and integration with existing systems to streamline operations.

- **Regulatory Bodies & Compliance Officers:** Ensure adherence to GDPR, HIPAA, and other data protection standards. Require access to audit logs, security controls, and data encryption features.

- **IT Administrators:** Responsible for system maintenance, user access management, and troubleshooting. Require full control over system settings, data storage, and integration with enterprise solutions.

- **End Consumers:** Individuals using voice-driven forms for personal applications (e.g., bank account openings, government services). Require a simple, accessible, and error-free experience.

- **Researchers & Data Analysts:** Analysts studying user interaction patterns and AI efficiency. Require access to analytics, reporting tools, and system logs to refine AI models and improve accuracy.

## 2.4 Operating Environment

- **Processor:** Intel Core i5 or higher / AMD Ryzen 5 or higher
- **RAM:** 8 GB or higher (16 GB recommended for AI processing)
- **Storage:** 256 GB SSD or higher (HDD with at least 100 GB free space)
- **Operating System:** Windows 10/11, macOS 11+, or Linux (Ubuntu 20.04+)
- **Web Browser:** Chrome, Firefox, Edge, Safari (latest versions)
- **Internet Connection:** Stable broadband connection (minimum 10 Mbps)
- **Additional Requirements:** Microphone for voice input, Azure Cloud access for AI services

## 2.5 Design and Implementation Constraints

- **Voice Input Dependency:** Requires a stable internet connection and a functional microphone for accurate speech recognition.
- **Multilingual Support Limitations:** Some languages or dialects may have reduced accuracy in recognition and translation.
- **Form Complexity:** Dynamic or highly customized forms may require manual adjustments despite AI-powered auto-filling.
- **Regulatory Compliance:** Must adhere to data protection laws (e.g., GDPR, HIPAA), affecting data storage and processing.
- **Offline Limitations:** Core functionalities may work offline, but AI processing and real-time translation require internet access.
- **Integration Constraints:** Compatibility with third-party services depends on API availability and authorization policies.

## 2.6 User Documentation

- **Testing Video:** A demonstration of the system's functionality and workflows.
- **README.md:** Overview of the project, installation steps, and usage instructions.
- **LICENSE:** Information on project licensing.
- **User Manual:** Guidelines on using voice commands, navigating forms, and submitting data.

## 2.7 Assumptions and Dependencies

**Assumptions:**

- The system assumes users have a working microphone for voice input.
- A stable internet connection is required for real-time processing and form submission.
- Users will follow the provided voice command structure for optimal accuracy.

**Dependencies:**

- The system relies on **Azure Speech-to-Text API** for voice recognition.
- **Azure Translator API** is required for multilingual processing.
- The backend depends on **Django (Python)** with **PostgreSQL or MongoDB** for data storage.
- **Docker** is used for containerization, making deployment and scaling easier.
- **Kubernetes** will be used for orchestration, ensuring efficient scaling and management of containers.

# 3. External Interface Requirements

## 3.1 User Interfaces

The voice-driven form-filling system will have an intuitive user interface (UI) designed to optimize accessibility and productivity. The system will use a voice-first design, where voice commands are the primary method of interaction. However, it will also offer fallback text-based inputs when voice is not available. The key interfaces include:

- **Login Screen:** For user authentication.
- **Dashboard:** A central hub where users can access different features and modules based on roles (e.g., healthcare, banking, government).
- **Voice Command Input Screen:** Where users can input voice commands to fill out forms.
- **Form Navigation Screen:** Allows users to navigate between form fields using voice commands (e.g., "next field", "back", "submit").
- **Multilingual Input Screen:** For users to select and enter data in their preferred language with real-time translations.
- **Reports Screen:** Displays reports and analytics based on the forms filled out and submitted.

## 3.2 Hardware Interfaces

The system will be compatible with various devices with adequate processing capabilities. The hardware required includes:

- **Client Devices (End Users):**
  - ✓ Desktop/Laptop: Core i3 processor or higher, 4 GB RAM, 100 GB HDD.
  - ✓ Mobile Devices: Android/iOS devices with microphone support.
- **Server:**
  - ✓ Dual-core processor, 8 GB RAM, 1 TB HDD.
- **Network:**
  - ✓ Reliable internet connection (for voice recognition, translation services, and form submission).

## 3.3 Software Interfaces

The system will interface with multiple software tools and services to provide a seamless experience:

- **Operating System:** Windows 10 or higher, Linux (Ubuntu, CentOS).
- **Web Browser:** Chrome, Firefox, IE11+, Safari.
- **Database:** PostgreSQL or MongoDB (for storing form data and user profiles).
- **Voice Processing:** Azure Speech-to-Text API.
- **Translation Service:** Azure Translator API.
- **AI Automation:** OpenAI GPT models (for form auto-filling).
- **Development Tools:** Docker, Kubernetes, Python (Django), React.js.

## 3.4 Communications Interfaces

The system will communicate over the internet using standard protocols for data transfer and encryption:

- **Protocols:** HTTP/HTTPS for secure web communication.
- **Data Encryption:** SSL/TLS encryption to secure data between the client and server, including voice data and personal information.
- **Real-Time Communication:** WebSocket or REST APIs for real-time updates and interactions.

# 4. System Features

## 4.1 Registration

### 4.1.1 Description and Priority

This feature allows new users to register for the platform by entering their personal details and setting up their username and password. It's a high-priority feature, as users need to register before using the system.

### 4.1.2 Stimulus/Response Sequences

- **Input**: User enters their personal details (name, email, password).
- **Output**: A confirmation dialog will appear, showing whether the registration was successful or not.
- **Processing**: The system checks the provided data against the database for uniqueness (username/email) and validity (password strength).
- **Error**: If there's an error (e.g., duplicate username or weak password), an error message is displayed prompting the user to try again.

### 4.1.3 Functional Requirements

- **Input**: User provides personal information and selects a username and password.
- **Output**: Registration details are saved to the database.

- **Processing**: Validation checks are applied to ensure valid and unique inputs.
- **Error**: Validation errors (e.g., invalid email, weak password) are displayed immediately.

## 4.2 User Login/Logout

### 4.2.1 Description and Priority

Users must log in with their credentials to access their profile and interact with the system. The logout functionality is also essential to securely exit the system.

### 4.2.2 Stimulus/Response Sequences

- **Input**: User enters login credentials (username and password).
- **Output**: The system redirects the user to the profile page upon successful login.
- **Processing**: The system verifies the credentials against the database.
- **Error**: If the credentials are incorrect, the system prompts the user to retry.

### 4.2.3 Functional Requirements

- **Input**: Username and password.
- **Output**: Successful login takes the user to their profile page.
- **Processing**: Credential verification.
- **Error**: Error message shown for incorrect credentials.

## 4.3 Admin Panel

### 4.3.1 Description and Priority

Admins can manage the users, view data, and handle operations like modifying. This feature is critical for managing the system.

### 4.3.2 Stimulus/Response Sequences

- **Input**: Admin adds users, updates bookings, or checks data.
- **Output**: Confirmation dialog for successful actions or a display of the requested data.
- **Processing**: The system updates the database based on the admin's actions.
- **Error**: If there’s a failure in the process (e.g., database issues), an error message will appear.

### 4.3.3 Functional Requirements

- **Input**: Admin enters data for rooms, bookings, users, etc.
- **Output**: Successful updates and data retrievals.
- **Processing**: Admin operations are executed on the backend.
- **Error**: Errors related to failed actions are displayed to the admin.

## 4.4 Voice Command Processing

### 4.4.1 Description and Priority

Users will interact with the system using voice commands to navigate forms, input data, and receive assistance. This feature is crucial for a hands-free user experience.

### 4.4.2 Stimulus/Response Sequences

- **Input**: User speaks a voice command (e.g., "Enter my name").

- **Output**: The system transcribes the speech to text and auto-fills the corresponding field.
- **Processing**: The system uses a speech-to-text API to convert speech to text and processes the command.
- **Error**: If the system doesn't understand the voice input, it prompts the user to repeat or rephrase their command.

### 4.4.3 Functional Requirements

- **Input**: Voice commands.
- **Output**: Voice transcriptions and corresponding field auto-fill actions.
- **Processing**: Speech-to-text conversion and action execution.
- **Error**: Feedback for unrecognized commands.

## 4.5 Voice Input and Language Processing

### 4.5.1 Description and Priority

This feature processes user voice input in various languages, enabling the system to recognize commands and data across different languages. It ensures accessibility and usability for diverse users.

### 4.5.2 Stimulus/Response Sequences

- **Input**: User speaks in a given language.
- **Output**: The system recognizes the voice and processes the input accordingly.
- **Processing**: Voice recognition and language processing.
- **Error**: If the language or input is not recognized, the system prompts the user to retry or provides an alternative input method.

### 4.5.3 Functional Requirements

- **Input:** Voice input in multiple supported languages.
- **Output:** Transcribed text and corresponding data entry or command execution.
- **Processing:** Speech-to-text conversion, language detection, and action execution.
- **Error:** System provides feedback for unrecognized input, including prompts for retries or alternative input methods.

## 4.6 Form Auto-Fill

### 4.6.1 Description and Priority

This feature automatically fills in forms based on user inputs, streamlining the form submission process. It is essential for enhancing the user experience and saving time.

### 4.6.2 Stimulus/Response Sequences

- **Input**: User provides voice input for personal details (e.g., "My name is John Doe").
- **Output**: The system auto-fills the form fields with the provided information.
- **Processing**: The system recognizes the data from voice input and fills the corresponding form fields.
- **Error**: If the system misinterprets the input, it prompts the user to correct it.

### 4.6.3 Functional Requirements

- **Input**: Voice input (e.g., name, address, email).
- **Output**: Automatically populated form fields.
- **Processing**: Data mapping to corresponding form fields.
- **Error**: Incorrectly filled fields prompt user feedback.

## 4.7 AI-Powered Auto-Filling

### 4.7.1 Description and Priority

This feature uses AI to intelligently predict and fill form fields based on user patterns and previous inputs, reducing the need for manual entry. It is highly important for improving form-filling efficiency.

### 4.7.2 Stimulus/Response Sequences

- **Input**: User interacts with previous forms or fields.
- **Output**: The system suggests and auto-fills relevant data for the current form.
- **Processing**: AI algorithms analyze past user behavior to suggest data.
- **Error**: If the suggestion is incorrect, the user can manually edit the field.

### 4.7.3 Functional Requirements

- **Input**: User's past data and current form context.
- **Output**: Pre-filled fields based on AI predictions.
- **Processing**: AI analysis for auto-filling fields.
- **Error**: Option for users to manually adjust data.

## 4.8 Real-Time Validation

### 4.8.1 Description and Priority

Ensures that form data is validated in real-time as the user fills it out, preventing errors during form submission. This is critical for maintaining data integrity.

### 4.8.2 Stimulus/Response Sequences

- **Input**: User enters data into form fields.
- **Output**: The system immediately validates the data and provides feedback (e.g., error messages for invalid inputs).
- **Processing**: The system checks the validity of the entered data (e.g., email format, required fields).
- **Error**: Invalid inputs prompt error messages next to the respective fields.

### 4.8.3 Functional Requirements

- **Input**: Data entered into form fields.
- **Output**: Real-time feedback (validation messages).
- **Processing**: Validation checks for field correctness.
- **Error**: Error messages for incorrect or missing data.

## 4.9 Voice Assistance for Navigation

### 4.9.1 Description and Priority

Provides users with voice-guided navigation through forms, helping them understand the steps and complete fields effectively. This feature enhances accessibility.

### 4.9.2 Stimulus/Response Sequences

- **Input**: User requests voice assistance (e.g., “Help me fill out the address field”).
- **Output**: The system guides the user through the process, offering instructions or tips.
- **Processing**: The system recognizes the request and provides the relevant voice guidance.
- **Error**: If the system doesn’t understand the request, it asks the user to clarify.

### 4.9.3 Functional Requirements

- **Input**: Voice commands requesting assistance.
- **Output**: Voice instructions or help.
- **Processing**: Processing user requests for help.
- **Error**: Clarifying questions if the system doesn’t understand the request.

## 4.10 Form Review and Edit

### 4.10.1 Description and Priority

After filling out a form, users can review and edit their data before submitting it. This ensures that users can correct any mistakes before final submission.

### 4.10.2 Stimulus/Response Sequences

- **Input**: User requests to review and edit the form.
- **Output**: The system displays the filled form, allowing the user to make changes.
- **Processing**: The system allows data modifications before final submission.
- **Error**: If the user tries to submit incomplete or invalid data, an error message is displayed.

### 4.10.3 Functional Requirements

- **Input**: User makes edits to filled fields.
- **Output**: Updated data on the form.
- **Processing**: Data modifications before submission.
- **Error**: Error feedback if invalid data is submitted.

## 4.11 Error Handling

### 4.11.1 Description and Priority

This feature ensures the system gracefully handles any errors, providing users with clear error messages and instructions to resolve issues.

### 4.11.2 Stimulus/Response Sequences

- **Input**: User encounters an error (e.g., incorrect data, failed submission).
- **Output**: A descriptive error message is shown to the user.
- **Processing**: The system detects errors and provides feedback.

- **Error**: If the system itself encounters an error, it displays a general error message and logs the issue for further inspection.

### 4.11.3 Functional Requirements

- **Input**: User actions that trigger errors.
- **Output**: Error message explaining the issue and corrective actions.
- **Processing**: Error detection and feedback generation.
- **Error**: General system errors if something goes wrong on the server side.

## 4.12 User Profile Management

### 4.12.1 Description and Priority

Users can view and update their profile information. This is important for ensuring that user data remains accurate and up-to-date.

### 4.12.2 Stimulus/Response Sequences

- **Input**: User requests to edit profile details.
- **Output**: The system displays the editable profile form.
- **Processing**: The system allows the user to make changes to their profile and save them.
- **Error**: If changes are invalid or incomplete, the system displays an error message.

### 4.12.3 Functional Requirements

- **Input**: User submits updated profile information.
- **Output**: Profile details are updated in the database.
- **Processing**: Updates are validated and saved.
- **Error**: Invalid updates or incomplete fields trigger error feedback.

## 4.13 Analytics and Reporting

### 4.13.1 Description and Priority

This feature provides insights into form submissions, error rates, and user activity. It's essential for tracking system performance and improving user experience.

### 4.13.2 Stimulus/Response Sequences

- **Input**: Admin requests reports (e.g., error statistics, user activity).
- **Output**: A report is generated and displayed or exported.
- **Processing**: Data is pulled from the system and processed into a report.
- **Error**: If data cannot be processed, an error message is displayed.

### 4.13.3 Functional Requirements

- **Input**: Request for a specific report type.
- **Output**: Displayed or downloadable report.
- **Processing**: Data extraction and report generation.
- **Error**: Failure to generate reports displays an error message.

## 4.14 Integration with External Data Sources

### 4.14.1 Description and Priority

This feature allows the system to pull or push data to external sources like CRM systems or third-party services.

### 4.14.2 Stimulus/Response Sequences

- **Input**: Data exchange requests between the system and an external service.
- **Output**: Data fetched or sent to external sources.
- **Processing**: The system interfaces with external services through APIs.
- **Error**: If there's a problem with the integration, an error message is displayed.

### 4.14.3 Functional Requirements

- **Input**: Data exchange requests.
- **Output**: Data received from or sent to external systems.
- **Processing**: API-based communication.
- **Error**: Error handling for integration failures.

## 4.15 Workflow Automation and Optimization

### 4.15.1 Description and Priority

Automates repetitive tasks to streamline processes like form validation, submission, and user notifications, improving system efficiency.

### 4.15.2 Stimulus/Response Sequences

- **Input**: User submits a form or requests a process.
- **Output**: Automated workflows trigger corresponding actions.
- **Processing**: Pre-defined workflows are executed automatically.
- **Error**: Automation failure prompts an error message.

### 4.15.3 Functional Requirements

- **Input**: Form submission or triggered workflow.
- **Output**: Automated system actions (e.g., data validation, email notification).
- **Processing**: Workflow automation and optimization.
- **Error**: Errors in workflow execution are reported to the user.

## 4.16 Multilingual Voice Input & Real-Time Translation

### 4.16.1 Description and Priority

Supports multiple languages for voice input and real-time translation, enabling users to interact with the system in their preferred language.

### 4.16.2 Stimulus/Response Sequences

- **Input**: User speaks in any supported language.
- **Output**: The system translates the input to the desired language and processes it accordingly.
- **Processing**: The system uses voice-to-text and language translation APIs.

- **Error**: Unsupported languages trigger an error message.

#### 4.16.3 Functional Requirements

- **Input**: Voice input in any supported language.
- **Output**: Translated and processed data.
- **Processing**: Translation and text processing.
- **Error**: Errors for unsupported languages or failed translations.

### 4.17 Data Submission

#### 4.17.1 Description and Priority

This feature enables users to submit their filled forms to the system for processing. It's crucial for completing the form-filling process.

#### 4.17.2 Stimulus/Response Sequences

- **Input**: User confirms the form submission.
- **Output**: The system sends the data to the server for storage and processing.
- **Processing**: The system validates and processes the submitted data.
- **Error**: Submission failure due to invalid data or server issues prompts an error message.

#### 4.17.3 Functional Requirements

- **Input**: Form data ready for submission.
- **Output**: Form data successfully submitted.
- **Processing**: Data validation and submission.
- **Error**: Error messages for failed submissions.

## 5. Other Nonfunctional Requirements

### 5.1.1 Performance Requirements

For the voice-driven form-filling system, performance is critical to ensure smooth and efficient user experience. Although the system's core functionality is voice-driven, it still needs to be responsive and capable of handling various types of requests concurrently.

- **Response Time**: The system should respond to user voice input and actions within 3 seconds to maintain user engagement and usability.

- **Throughput**: The system should handle up to 500 concurrent users without significant degradation in performance.

- **Scalability**: The system should be scalable to accommodate future growth, such as adding more users, processing more complex forms, or handling multiple languages.

- **Reliability**: The system must be reliable, ensuring that users can fill forms without interruptions. Web pages should load quickly, and all features should be functional during usage.

- **Web Pages Load Time**: Pages and form processing should complete within a few seconds to ensure user satisfaction.

### 5.1.2 Safety Requirements

The safety of the data processed by the system is a critical concern, particularly as sensitive information is collected from users, such as personal details, medical history, financial information, and more. Proper database management is crucial to ensure information security and consistency.

- **Data Encryption**: All sensitive data, including voice input and form responses, must be encrypted both in transit (using HTTPS) and at rest (using AES encryption) to prevent unauthorized access.
- **Backup**: The system must perform daily backups of user data and form submissions to prevent data loss. Regular backups will help recover data in case of unexpected failures.

### 5.1.3 Security Requirements

Security is fundamental to prevent unauthorized access to the system and safeguard user data. Only authenticated and authorized users should be able to interact with the system.

- **User Authentication**: Secure login mechanisms, such as username-password authentication or two-factor authentication (2FA), should be implemented. Role-based access control (RBAC) should restrict access to sensitive areas based on user roles (e.g., admin, guest, etc.).
- **Data Access Control**: Access to user-specific data, such as personal information, should be limited to authorized users only (e.g., only admins can access all user data, while users can only access their own data).
- **Audit Trails**: All user actions (e.g., login attempts, form submissions, edits, deletions) should be logged for auditing purposes. This will allow tracking of any suspicious or unauthorized activities.

### 5.1.4 Software Quality Attributes

The system must maintain high-quality standards to ensure that it performs efficiently, remains reliable, and is easy to maintain and update.

- **Usability**: The system should be user-friendly, allowing even non-technical users to quickly understand how to fill forms using voice input. The user interface should be intuitive, and voice commands should be easy to follow.
- **Reliability**: The system should be highly available, with an uptime of 99.9% or better. Regular testing and monitoring should ensure that the system is free from critical bugs and downtime.
- **Maintainability**: The system should be modular and easily maintainable. Updates to the system should be possible without significant downtime, and new features (such as adding support for more languages) should be easy to integrate.

### 5.1.5 Business Rules

- **Cancellation Policy**: Users must be able to cancel a form submission request before the form is finalized. Once submitted, the request cannot be undone without administrative intervention.
- **Form Completion**: Users are required to fill out all mandatory fields in the form before submission. If any required fields are missing or incorrect, the system should prompt the user to complete them.

# 6. Legal and Ethical Considerations

## 6.1 Compliance with Local Laws

The system must comply with applicable data protection laws in India, such as the **Personal Data Protection Bill** (PDPB) 2019, which governs the processing of personal data, and other relevant regulations regarding privacy and data security. Any personal or sensitive data entered by users must be handled in compliance with these laws.

The Personal Data Protection Bill mandates that personal data should only be processed with the consent of the individual, and the system must ensure that user data is stored securely and handled responsibly. Additionally, the system should include provisions for users to access, correct, and delete their personal data in accordance with the rights granted by the PDPB.

## 6.2 Ethical Considerations

The system must handle guest and user data ethically. Users should be informed about how their data will be used, and explicit consent should be obtained before collecting any personal information. Data should be stored securely, and the system should respect users' right to privacy by implementing strict data access policies.

# 7. Other Requirements

Not Applicable.

# Appendix A: Glossary

## Acronyms and Abbreviations:

1. **AI** - Artificial Intelligence
2. **API** - Application Programming Interface
3. **HTTPS** - Hypertext Transfer Protocol Secure
4. **AI-Powered Auto-Filling** - Intelligent system that auto-completes fields in the form based on the user's past behavior and preferences.
5. **SSL** - Secure Sockets Layer (for encrypted communication)
6. **AES** - Advanced Encryption Standard (encryption standard for data security)
7. **2FA** - Two-Factor Authentication (additional layer of security)
8. **RBAC** - Role-Based Access Control (user permissions based on roles)
9. **PDPB-** Personal Data Protection Bill
10. **RAM** - Random Access Memory (temporary storage for quick data retrieval)
11. **SMS** - Short Message Service (text messaging service)