

Requirement 2

e-Invoice JSON preparation, upload, receipt of IRN and error handling

The requirement you're referring to involves demonstrating the process of handling e-invoices under the GST framework, which includes preparing JSON files, uploading them, receiving the Invoice Reference Number (IRN), and managing any errors that may occur during this process. Here's an overview of each step:

1. e-Invoice JSON Preparation:

- The e-invoice is prepared in a specified JSON format per the GSTN schema.
- This involves capturing all the mandatory and optional fields like supplier details, recipient details, invoice items, HSN codes, tax amounts, etc.
- The JSON file should be validated against the schema to ensure it meets all the necessary requirements.

2. Uploading the JSON File:

- Once the e-Invoice JSON is prepared, it is uploaded to the Invoice Registration Portal (IRP) through APIs or directly via the portal.
- The system should ensure that the JSON is properly structured and meets all the validation rules set by the IRP.

3. Receipt of IRN (Invoice Reference Number):

- Upon successful upload and validation by the IRP, the system generates a unique Invoice Reference Number (IRN) for the e-Invoice.
- The IRP also provides a digitally signed JSON of the e-Invoice and a QR code that can be used to verify the authenticity of the invoice.

4. Error Handling:

- If there are any issues with the JSON file (e.g., missing mandatory fields, incorrect data formats), the IRP will return an error response.
- The system should be capable of handling these errors, notifying the user, and allowing for corrections to be made.
- After corrections, the JSON needs to be re-uploaded until it passes the validation and an IRN is generated.

This process ensures that the e-Invoice is accurately generated, validated, and accepted by the GST system, and it helps maintain compliance with GST regulations.

Error Handling

https://docs.google.com/document/d/1l8c5vjIIN84wn9247usX_ejdIVkCmy8j4VDt5HOJps/edit?usp=drive_link

Requirement	Scenario Title	Scenario Explanation	Sub Scenarios
e-Invoice JSON Preparation	Ensuring Compliance with GSTN JSON Schema	Demonstrate the process of capturing all mandatory and optional fields required by the GSTN, such as supplier details, recipient details, invoice items, HSN codes, and tax amounts. The scenario should show how the system validates the JSON file against the GSTN schema, ensuring that all fields are correctly formatted and no required data is missing.	<ol style="list-style-type: none"> 1. Validating Mandatory Fields 2. Validating Optional Fields 3. Schema Compliance Check
	Customization of JSON for Different Business Models	Showcase how the e-Invoice system can be customized to accommodate different business models (e.g., retail, manufacturing) while still adhering to the GSTN's JSON schema. The scenario should include the addition of optional fields specific to the business needs, ensuring the flexibility of the system without compromising on compliance.	<ol style="list-style-type: none"> 1. Adding Custom Fields for Retail 2. Customizing JSON for Manufacturing 3. Validating Custom Fields
	Automated Data Entry and Validation	Highlight how the system automates the entry of invoice data into the JSON format, reducing manual errors. The scenario should show how the system automatically validates the data against the schema and flags any discrepancies for correction before the file is finalized. This ensures a streamlined and error-free JSON preparation process.	<ol style="list-style-type: none"> 1. Data Entry Automation 2. Automated Validation 3. Handling Discrepancies

	Multi-Language Support for Invoice Preparation	Demonstrate how the system can prepare e-Invoice JSON files in multiple languages to cater to businesses operating in different linguistic regions. The scenario should show how the system handles translations and ensures that the JSON file remains compliant with the GSTN schema, regardless of the language used.	<ol style="list-style-type: none"> 1. Preparing JSON in Multiple Languages 2. Translation Accuracy Check 3. Schema Validation
	Handling Multi-Currency Invoices	Showcase the system's ability to prepare JSON files for invoices issued in different currencies. The scenario should highlight how the system converts the amounts to INR for GST compliance while retaining the original currency details in the JSON file. This is particularly important for businesses involved in international trade.	<ol style="list-style-type: none"> 1. Currency Conversion to INR 2. Retaining Original Currency Details 3. Validating Multi-Currency JSON
	Compliance with Specific GST Notifications	Demonstrate how the e-Invoice JSON preparation aligns with the latest GST notifications, such as mandatory 6-digit HSN codes or specific requirements for government transactions. This scenario ensures that the system is updated and compliant with the latest regulatory changes.	<ol style="list-style-type: none"> 1. Implementing HSN Code Requirements 2. Handling Government Agency Invoices 3. Ensuring Latest Compliance
	Preparation of Export Invoices	Show how the system prepares JSON files for export invoices, ensuring that all export-specific details (e.g., LUT/Bond details) are accurately captured in compliance with GST rules. This scenario is critical for businesses involved in international trade.	<ol style="list-style-type: none"> 1. Capturing LUT/Bond Details 2. Ensuring Export Compliance 3. Validating Export-Specific Fields

	Handling Reverse Charge Mechanism	Illustrate how the system prepares invoices under the reverse charge mechanism and ensures that the JSON reflects this accurately according to GST requirements. This is vital for businesses dealing with reverse charge supplies.	<ol style="list-style-type: none"> 1. Identifying Reverse Charge Transactions 2. JSON Preparation for Reverse Charge 3. Validation Process
Uploading the JSON File	Seamless API Integration for JSON Upload	Demonstrate the process of uploading the prepared JSON file to the Invoice Registration Portal (IRP) via an API. The scenario should show how the system ensures that the JSON is properly structured and meets all the validation rules set by the IRP before submission. This step is critical for preventing upload failures and ensuring that the data is accurately transmitted to the IRP.	<ol style="list-style-type: none"> 1. API Connection Setup 2. JSON Structure Validation 3. Successful Upload Confirmation
	Direct Portal Upload with Real-Time Feedback	In this scenario, the JSON file is uploaded directly to the IRP via the web portal. The demo should highlight how the system provides real-time feedback during the upload process, immediately notifying the user if the file is not compliant with the IRP's requirements. This feature is essential for allowing quick corrections and resubmission, minimizing downtime and delays.	<ol style="list-style-type: none"> 1. Portal Upload Process 2. Real-Time Error Feedback 3. Corrective Actions and Resubmission
	Bulk Uploads of Multiple JSON Files	This scenario covers the process of uploading multiple JSON files in bulk. The demo should show how the system handles the validation and upload of each file individually, flagging any errors and processing successful files. This capability is crucial for businesses that generate a high	<ol style="list-style-type: none"> 1. Initiating Bulk Upload 2. Error Identification in Bulk 3. Processing Successful Files

		volume of invoices and need to ensure that all files are uploaded efficiently.	
	Handling Interrupted Upload Sessions	This scenario demonstrates how the system manages interrupted upload sessions due to network or system failures. The demo should show how the system resumes the upload from where it left off, ensuring that the JSON file is fully uploaded without data loss or duplication. This feature is critical for maintaining the integrity of the data during upload.	<ol style="list-style-type: none"> 1. Detecting Upload Interruptions 2. Resuming Interrupted Uploads 3. Ensuring Data Integrity
	User Role-Based Access for Uploads	Illustrate how the system enforces role-based access control during the JSON upload process. The scenario should show how different users with varying levels of permissions can access the upload function, ensuring that only authorized personnel can submit e-Invoice JSON files to the IRP. This helps maintain security and accountability within the organization.	<ol style="list-style-type: none"> 1. Defining User Roles 2. Role-Based Access Control 3. Upload Authorization Process
	Time-Bound Invoice Reporting	Highlight how the system handles the requirement for time-bound reporting of invoices, such as the 30-day reporting limit for high-turnover businesses. This scenario is crucial for ensuring timely compliance and avoiding penalties.	<ol style="list-style-type: none"> 1. Tracking Reporting Deadlines 2. Handling Late Submissions 3. System Alerts for Timely Reporting

Receipt of IRN (Invoice Reference Number)	Automatic Generation of IRN and QR Code	Demonstrate the system's capability to automatically generate an Invoice Reference Number (IRN) and a QR code upon successful upload and validation of the JSON by the IRP. The scenario should show how the system retrieves the digitally signed JSON and QR code from the IRP and associates it with the original invoice in the system. This ensures the authenticity and traceability of the e-Invoice.	<ol style="list-style-type: none"> 1. IRN Generation Process 2. QR Code Retrieval 3. Associating IRN with Invoice
	Verification and Archival of IRN and Signed JSON	Show how the system verifies the authenticity of the IRN and digitally signed JSON received from the IRP. The scenario should include the archival of these documents for future reference and compliance audits. This process is essential for maintaining accurate records and ensuring that all e-Invoices can be verified at any time.	<ol style="list-style-type: none"> 1. Verifying Digital Signature 2. Archiving e-Invoice Records 3. Retrieval for Audit
	IRN Retrieval in Case of System Failure	Illustrate how the system handles scenarios where there is a delay or failure in receiving the IRN due to network or system issues. The demo should show the system's capability to automatically retry the retrieval of the IRN from the IRP and how it notifies the user once the IRN is successfully obtained. This ensures that no invoice is left unprocessed due to technical issues.	<ol style="list-style-type: none"> 1. Detecting IRN Retrieval Failure 2. Automatic Retry Mechanism 3. User Notification on Success
	Notifications for Successful IRN Generation	Highlight the system's capability to send notifications (e.g., email, SMS) to relevant stakeholders once an IRN is successfully generated. The scenario should demonstrate how these notifications ensure that the appropriate team members are promptly informed, reducing the time to process the invoice further.	<ol style="list-style-type: none"> 1. Setting Up Notifications 2. Triggering Notifications on IRN Generation 3. Ensuring Timely Alerts

	Archiving and Retrieval of Signed e-Invoice	Demonstrate how the system archives the digitally signed JSON and IRN for future reference. The scenario should include a demonstration of the retrieval process, showing how the system allows users to access and verify archived e-Invoices easily. This is important for compliance and audit purposes.	<ol style="list-style-type: none"> 1. Digital Archiving Process 2. Easy Retrieval of Archived Invoices 3. Ensuring Data Integrity
	Auto-Population into GSTR-1 from IRN	Demonstrate how the system automatically populates the GSTR-1 form with the data from the IRN-generated invoices, ensuring seamless integration and compliance. This scenario is essential for simplifying return filing and ensuring accuracy.	<ol style="list-style-type: none"> 1. Auto-Population of GSTR-1 2. Reconciliation of Auto-Populated Data 3. Manual Adjustments if Necessary
Error Handling	Handling Missing or Incorrect Data in JSON	Demonstrate how the system responds when the JSON file submitted to the IRP contains missing or incorrect data, such as an invalid GSTIN or a missing mandatory field. The scenario should show how the system identifies the specific errors, provides detailed error messages, and guides the user in correcting the data. Once the errors are corrected, the JSON is revalidated and resubmitted until it passes the IRP's checks.	<ol style="list-style-type: none"> 1. Error Detection in JSON 2. Detailed Error Messaging 3. Correcting and Resubmitting JSON

	Real-Time Error Notifications and Resolution	Highlight the system's ability to provide real-time notifications to the user when an error is encountered during the JSON upload process. The scenario should show how the system immediately informs the user of the issue, explains the nature of the error, and offers step-by-step guidance to resolve it. This proactive error handling is essential for ensuring quick resolution and preventing delays in obtaining the IRN.	<ol style="list-style-type: none"> 1. Real-Time Error Detection 2. Immediate User Notification 3. Guided Error Resolution
	Automated Re-Submission After Error Correction	Show how the system automates the re-submission of the JSON file after the user has corrected any errors identified by the IRP. The scenario should include the validation process to ensure that the corrected JSON now meets all the IRP's requirements, followed by a successful upload and receipt of the IRN. This feature is critical for streamlining the error correction process and minimizing the effort required from the user.	<ol style="list-style-type: none"> 1. Correcting Errors in JSON 2. Automated Re-Validation 3. Successful Re-Submission
	Detailed Audit Trail for Error Handling	Show how the system maintains a detailed audit trail of all errors encountered during the JSON upload process. The scenario should demonstrate how users can review past errors, the corrective actions taken, and the final successful upload. This feature is crucial for ensuring transparency and accountability in the error-handling process.	<ol style="list-style-type: none"> 1. Logging Errors 2. Tracking Corrective Actions 3. Reviewing Successful Uploads

	Automated Error Reporting and Analytics	Demonstrate how the system generates automated reports and analytics based on the errors encountered during JSON preparation and upload. The scenario should show how these reports help businesses identify common issues and take proactive measures to prevent them in the future. This feature is essential for continuous improvement and reducing error rates.	<ol style="list-style-type: none"> 1. Error Reporting Setup 2. Generating Analytics 3. Implementing Proactive Measures
	Scenario for Managing Errors in Bulk Uploads	This scenario demonstrates how the system handles errors when multiple JSON files are uploaded in bulk. The demo should show how the system identifies errors in specific files, processes the correct files, and allows for the correction and re-upload of the erroneous files without affecting the entire batch. This capability is vital for businesses that deal with high volumes of invoices and need efficient error management.	<ol style="list-style-type: none"> 1. Bulk Error Identification 2. Isolating Error Files 3. Correcting and Re-Uploading Specific Files
	Handling Auto-Cancellation of Invoices	Illustrate how the system manages auto-cancellation of invoices that were mistakenly uploaded or require corrections, including the process for re-issuing and updating GSTR-1. This scenario addresses the challenges of managing mistakes in a regulated environment.	<ol style="list-style-type: none"> 1. Detecting and Handling Auto-Cancellations 2. Re-Issuing Corrected Invoices 3. Updating GSTR-1 with Correct Data
	Managing API Failures During IRN Generation	Show how the system handles API failures during IRN generation, including retry mechanisms and user notifications. This scenario ensures that technical issues do not disrupt compliance.	<ol style="list-style-type: none"> 1. Detecting API Failures 2. Implementing Retry Mechanisms 3. Notifying Users of

			Success or Failure
--	--	--	--------------------

1. Sub Scenario Title: Validating Mandatory Fields

- **Main Requirement Name:** e-Invoice JSON Preparation
- **Main Scenario:** Ensuring Compliance with GSTN JSON Schema
- **Sub Scenario Explanation:** This scenario involves ensuring that all mandatory fields required by the GSTN schema are correctly filled out in the e-Invoice JSON. These fields include critical information such as the supplier's GSTIN, invoice date, and total invoice value. The system should automatically validate these fields during the JSON preparation process to prevent submission errors.
- **Example:** A supplier prepares an invoice but forgets to enter the recipient's GSTIN. The system immediately flags this omission and prevents the JSON from being finalized until the GSTIN is correctly added.
- **Priority:** High

2. Sub Scenario Title: Validating Optional Fields

- **Main Requirement Name:** e-Invoice JSON Preparation
- **Main Scenario:** Ensuring Compliance with GSTN JSON Schema
- **Sub Scenario Explanation:** Although optional fields are not required for basic compliance, they can provide additional valuable information. This scenario demonstrates how the system handles optional fields like the recipient's email address or item-specific discounts. The system should allow users to input these fields without disrupting the mandatory compliance checks.
- **Example:** A business includes a discount on a product in the invoice. The system allows the user to enter this data and ensures it is correctly formatted within the JSON, even though it's optional.
- **Priority:** Medium

3. Sub Scenario Title: Schema Compliance Check

- **Main Requirement Name:** e-Invoice JSON Preparation
 - **Main Scenario:** Ensuring Compliance with GSTN JSON Schema
 - **Sub Scenario Explanation:** This scenario highlights the final compliance check where the system validates the entire JSON file against the GSTN schema to ensure that the format, structure, and data types adhere to the required standards. This check helps prevent rejections during the upload process.
 - **Example:** After filling out all necessary fields, the system runs a schema validation and identifies that the HSN code entered does not match the required number of digits. The user is prompted to correct this before the JSON can be finalized.
 - **Priority:** High
-

4. Sub Scenario Title: Adding Custom Fields for Retail

- **Main Requirement Name:** e-Invoice JSON Preparation
 - **Main Scenario:** Customization of JSON for Different Business Models
 - **Sub Scenario Explanation:** This scenario shows how the system can be customized to include additional fields specific to the retail industry, such as loyalty points or promotional discounts. The system should ensure these fields are correctly formatted in the JSON while maintaining compliance with the mandatory fields.
 - **Example:** A retail business includes a "Loyalty Points Earned" field in the invoice. The system incorporates this field into the JSON, ensuring that it does not interfere with compliance checks.
 - **Priority:** Medium
-

5. Sub Scenario Title: Customizing JSON for Manufacturing

- **Main Requirement Name:** e-Invoice JSON Preparation
 - **Main Scenario:** Customization of JSON for Different Business Models
 - **Sub Scenario Explanation:** In this scenario, the system customizes the JSON to include manufacturing-specific data, such as batch numbers or production dates. The system should seamlessly integrate these fields without disrupting the overall schema compliance.
 - **Example:** A manufacturing company includes batch numbers for each product. The system ensures that this data is correctly included in the JSON without causing validation errors.
 - **Priority:** Medium
-

6. Sub Scenario Title: Validating Custom Fields

- **Main Requirement Name:** e-Invoice JSON Preparation
 - **Main Scenario:** Customization of JSON for Different Business Models
 - **Sub Scenario Explanation:** This scenario covers the validation of custom fields added to the JSON. The system should check these fields to ensure they do not conflict with the schema's required structure and that they are correctly formatted.
 - **Example:** After adding custom fields for promotional discounts, the system validates the JSON to ensure the data types and field lengths meet the GSTN requirements.
 - **Priority:** Medium
-

7. Sub Scenario Title: Data Entry Automation

- **Main Requirement Name:** e-Invoice JSON Preparation
 - **Main Scenario:** Automated Data Entry and Validation
 - **Sub Scenario Explanation:** This scenario demonstrates how the system automates the entry of invoice data into the JSON format, reducing manual errors. The system should pull data from integrated ERP systems or databases and automatically populate the JSON fields.
 - **Example:** A company's ERP system generates an invoice, and the system automatically transfers all relevant data into the JSON format, ready for validation and submission.
 - **Priority:** High
-

8. Sub Scenario Title: Automated Validation

- **Main Requirement Name:** e-Invoice JSON Preparation
 - **Main Scenario:** Automated Data Entry and Validation
 - **Sub Scenario Explanation:** This scenario covers the automated validation process where the system checks all fields in the JSON for compliance with the GSTN schema. The system should flag any discrepancies or errors before finalizing the JSON.
 - **Example:** The system detects that the invoice date is in the wrong format (MM/DD/YYYY instead of DD/MM/YYYY) and automatically corrects it or prompts the user to make the correction.
 - **Priority:** High
-

9. Sub Scenario Title: Handling Discrepancies

- **Main Requirement Name:** e-Invoice JSON Preparation
- **Main Scenario:** Automated Data Entry and Validation

- **Sub Scenario Explanation:** This scenario highlights how the system handles discrepancies during the validation process. The system should provide clear error messages and guide users in resolving any issues.
 - **Example:** The system identifies that the GST rate applied to an item does not match the standard rate for that category. It prompts the user to select the correct rate before finalizing the JSON.
 - **Priority: High**
-

10. Sub Scenario Title: Preparing JSON in Multiple Languages

- **Main Requirement Name:** e-Invoice JSON Preparation
 - **Main Scenario:** Multi-Language Support for Invoice Preparation
 - **Sub Scenario Explanation:** This scenario demonstrates how the system supports the preparation of e-Invoice JSON files in different languages, accommodating businesses operating in various linguistic regions. The system should ensure that language-specific fields are correctly formatted and comply with the GSTN schema.
 - **Example:** A business operating in both English and Hindi markets prepares an invoice in Hindi. The system generates the JSON in Hindi, ensuring all fields are accurately translated and compliant.
 - **Priority: Medium**
-

11. Sub Scenario Title: Translation Accuracy Check

- **Main Requirement Name:** e-Invoice JSON Preparation
 - **Main Scenario:** Multi-Language Support for Invoice Preparation
 - **Sub Scenario Explanation:** The scenario covers how the system checks the accuracy of translations in the JSON file to prevent errors or misunderstandings during submission. The system should verify that all translations are correct and consistent with GST requirements.
 - **Example:** The system translates the item description from English to Tamil. It checks that the translation is accurate and that the JSON structure remains valid.
 - **Priority: Low**
-

12. Sub Scenario Title: Schema Validation

- **Main Requirement Name:** e-Invoice JSON Preparation
- **Main Scenario:** Multi-Language Support for Invoice Preparation
- **Sub Scenario Explanation:** This scenario ensures that even after translating the JSON into another language, the file remains compliant with the GSTN schema. The system

should run a final validation check to ensure there are no errors caused by language-specific issues.

- **Example:** After preparing the JSON in a regional language, the system validates it to ensure that numeric fields like GSTIN and HSN codes are unaffected by the language change.
 - **Priority: Medium**
-

13. Sub Scenario Title: Currency Conversion to INR

- **Main Requirement Name:** e-Invoice JSON Preparation
 - **Main Scenario:** Handling Multi-Currency Invoices
 - **Sub Scenario Explanation:** This scenario focuses on how the system handles multi-currency transactions by converting foreign currencies to INR for GST compliance. The system should automatically apply the correct exchange rate and ensure that the converted values are accurately reflected in the JSON.
 - **Example:** A company issues an invoice in USD. The system converts the total invoice value to INR using the current exchange rate and includes this information in the JSON.
 - **Priority: High**
-

14. Sub Scenario Title: Retaining Original Currency Details

- **Main Requirement Name:** e-Invoice JSON Preparation
 - **Main Scenario:** Handling Multi-Currency Invoices
 - **Sub Scenario Explanation:** This scenario demonstrates how the system retains the original currency details within the JSON file while still ensuring compliance with INR reporting requirements. The system should include both the original currency amount and the INR equivalent.
 - **Example:** The JSON includes fields for both the USD invoice value and its INR equivalent, ensuring that all currency details are accurately reported.
 - **Priority: Medium**
-

15. Sub Scenario Title: Validating Multi-Currency JSON

- **Main Requirement Name:** e-Invoice JSON Preparation
- **Main Scenario:** Handling Multi-Currency Invoices
- **Sub Scenario Explanation:** The scenario involves validating the multi-currency fields in the JSON to ensure they are correctly formatted and comply with GSTN requirements. The system should ensure that all currency conversions are accurate and that the JSON structure supports multiple currencies.

- **Example:** The system checks that the exchange rate used is accurate and that both the original and converted amounts are correctly represented in the JSON.
 - **Priority: High**
-

16. Sub Scenario Title: Implementing HSN Code Requirements

- **Main Requirement Name:** e-Invoice JSON Preparation
 - **Main Scenario:** Compliance with Specific GST Notifications
 - **Sub Scenario Explanation:** This scenario covers how the system handles updates to GST regulations, such as mandatory 6-digit HSN codes. The system should ensure that the JSON includes the correct HSN code length and format as per the latest GST notifications.
 - **Example:** The system identifies that an HSN code entered with only 4 digits needs to be updated to the 6-digit format and prompts the user to make the necessary correction.
 - **Priority: High**
-

17. Sub Scenario Title: Handling Government Agency Invoices

- **Main Requirement Name:** e-Invoice JSON Preparation
 - **Main Scenario:** Compliance with Specific GST Notifications
 - **Sub Scenario Explanation:** This scenario demonstrates how the system prepares invoices for transactions involving government agencies, ensuring compliance with any specific requirements or notifications applicable to such transactions.
 - **Example:** An invoice issued to a government agency requires additional fields, such as the agency's unique identifier. The system automatically includes these fields in the JSON.
 - **Priority: Medium**
-

18. Sub Scenario Title: Ensuring Latest Compliance

- **Main Requirement Name:** e-Invoice JSON Preparation
- **Main Scenario:** Compliance with Specific GST Notifications
- **Sub Scenario Explanation:** The scenario involves regularly updating the system to comply with the latest GST notifications and ensuring that all JSON files generated adhere to these updates. The system should prompt users when changes occur and guide them through any necessary adjustments.
- **Example:** The system detects a recent update requiring specific disclosures on invoices over a certain value and ensures that this information is included.
- **Priority: High**

19. Sub Scenario Title: Seamless API Integration for JSON Upload

- **Main Requirement Name:** Uploading the JSON File
 - **Main Scenario:** Ensuring Proper JSON Upload
 - **Sub Scenario Explanation:** Demonstrate the process of uploading the prepared JSON file to the Invoice Registration Portal (IRP) via an API. The scenario should show how the system ensures that the JSON is properly structured and meets all the validation rules set by the IRP before submission. This step is critical for preventing upload failures and ensuring that the data is accurately transmitted to the IRP.
 - **Example:** The system uses an API to connect with the IRP, uploading the JSON file and confirming its successful submission with an immediate response from the portal.
 - **Priority:** High
-

20. Sub Scenario Title: Direct Portal Upload with Real-Time Feedback

- **Main Requirement Name:** Uploading the JSON File
 - **Main Scenario:** Ensuring Proper JSON Upload
 - **Sub Scenario Explanation:** In this scenario, the JSON file is uploaded directly to the IRP via the web portal. The demo should highlight how the system provides real-time feedback during the upload process, immediately notifying the user if the file is not compliant with the IRP's requirements. This feature is essential for allowing quick corrections and resubmission, minimizing downtime and delays.
 - **Example:** A user uploads the JSON file through the IRP's portal and receives immediate feedback indicating that a mandatory field is missing, allowing for instant correction and resubmission.
 - **Priority:** High
-

21. Sub Scenario Title: Bulk Uploads of Multiple JSON Files

- **Main Requirement Name:** Uploading the JSON File
- **Main Scenario:** Ensuring Proper JSON Upload
- **Sub Scenario Explanation:** This scenario covers the process of uploading multiple JSON files in bulk. The demo should show how the system handles the validation and upload of each file individually, flagging any errors and processing successful files. This capability is crucial for businesses that generate a high volume of invoices and need to ensure that all files are uploaded efficiently.
- **Example:** A company uploads 50 invoices in bulk. The system processes each JSON file, successfully uploading those that pass validation while flagging the ones that need correction.
- **Priority:** High

22. Sub Scenario Title: Handling Interrupted Upload Sessions

- **Main Requirement Name:** Uploading the JSON File
 - **Main Scenario:** Ensuring Proper JSON Upload
 - **Sub Scenario Explanation:** This scenario demonstrates how the system manages interrupted upload sessions due to network or system failures. The demo should show how the system resumes the upload from where it left off, ensuring that the JSON file is fully uploaded without data loss or duplication. This feature is critical for maintaining the integrity of the data during upload.
 - **Example:** A network outage occurs during the upload process. The system saves the progress and resumes the upload once the connection is restored, ensuring that no data is lost.
 - **Priority:** Medium
-

23. Sub Scenario Title: User Role-Based Access for Uploads

- **Main Requirement Name:** Uploading the JSON File
 - **Main Scenario:** Ensuring Proper JSON Upload
 - **Sub Scenario Explanation:** Illustrate how the system enforces role-based access control during the JSON upload process. The scenario should show how different users with varying levels of permissions can access the upload function, ensuring that only authorized personnel can submit e-Invoice JSON files to the IRP. This helps maintain security and accountability within the organization.
 - **Example:** Only users with administrative privileges can upload JSON files to the IRP, while others can view the upload status but cannot initiate the upload.
 - **Priority:** Medium
-

24. Sub Scenario Title: Time-Bound Invoice Reporting

- **Main Requirement Name:** Uploading the JSON File
 - **Main Scenario:** Ensuring Proper JSON Upload
 - **Sub Scenario Explanation:** Highlight how the system handles the requirement for time-bound reporting of invoices, such as the 30-day reporting limit for high-turnover businesses. This scenario is crucial for ensuring timely compliance and avoiding penalties.
 - **Example:** The system tracks the reporting deadlines for each invoice and alerts users if an invoice is nearing the 30-day reporting limit, ensuring it is uploaded on time.
 - **Priority:** High
-

25. Sub Scenario Title: Automatic Generation of IRN and QR Code

- **Main Requirement Name:** Receipt of IRN (Invoice Reference Number)
 - **Main Scenario:** Ensuring Accurate IRN Generation
 - **Sub Scenario Explanation:** Demonstrate the system's capability to automatically generate an Invoice Reference Number (IRN) and a QR code upon successful upload and validation of the JSON by the IRP. The scenario should show how the system retrieves the digitally signed JSON and QR code from the IRP and associates it with the original invoice in the system. This ensures the authenticity and traceability of the e-Invoice.
 - **Example:** After successfully uploading the JSON file, the system receives an IRN and QR code from the IRP, which are then automatically appended to the invoice for verification purposes.
 - **Priority:** High
-

26. Sub Scenario Title: Verification and Archival of IRN and Signed JSON

- **Main Requirement Name:** Receipt of IRN (Invoice Reference Number)
 - **Main Scenario:** Ensuring Accurate IRN Generation
 - **Sub Scenario Explanation:** Show how the system verifies the authenticity of the IRN and digitally signed JSON received from the IRP. The scenario should include the archival of these documents for future reference and compliance audits. This process is essential for maintaining accurate records and ensuring that all e-Invoices can be verified at any time.
 - **Example:** The system stores the digitally signed JSON and IRN in a secure archive, allowing users to retrieve and verify the document during audits or regulatory checks.
 - **Priority:** High
-

27. Sub Scenario Title: IRN Retrieval in Case of System Failure

- **Main Requirement Name:** Receipt of IRN (Invoice Reference Number)
- **Main Scenario:** Ensuring Accurate IRN Generation
- **Sub Scenario Explanation:** Illustrate how the system handles scenarios where there is a delay or failure in receiving the IRN due to network or system issues. The demo should show the system's capability to automatically retry the retrieval of the IRN from the IRP and how it notifies the user once the IRN is successfully obtained. This ensures that no invoice is left unprocessed due to technical issues.
- **Example:** A network failure prevents immediate retrieval of the IRN. The system automatically retries and successfully obtains the IRN once the connection is restored, notifying the user of the successful retrieval.
- **Priority:** High

28. Sub Scenario Title: Notifications for Successful IRN Generation

- **Main Requirement Name:** Receipt of IRN (Invoice Reference Number)
- **Main Scenario:** Ensuring Accurate IRN Generation
- **Sub Scenario Explanation:** Highlight the system's capability to send notifications (e.g., email, SMS) to relevant stakeholders once an IRN is successfully generated. The scenario should demonstrate how these notifications ensure that the appropriate team members are promptly informed, reducing the time to process the invoice further.
- **Example:** After an IRN is generated, the system sends an email notification to the finance team, confirming that the invoice is now registered and ready for further processing.
- **Priority:** Medium

29. Sub Scenario Title: Archiving and Retrieval of Signed e-Invoice

- **Main Requirement Name:** Receipt of IRN (Invoice Reference Number)
- **Main Scenario:** Ensuring Accurate IRN Generation
- **Sub Scenario Explanation:** Demonstrate how the system archives the digitally signed JSON and IRN for future reference. The scenario should include a demonstration of the retrieval process, showing how the system allows users to access and verify archived e-Invoices easily. This is important for compliance and audit purposes.
- **Example:** The system archives the signed e-Invoice, and users can search for and retrieve the invoice using the IRN or invoice number during an audit.
- **Priority:** High

30. Sub Scenario Title: Auto-Population into GSTR-1 from IRN

- **Main Requirement Name:** Receipt of IRN (Invoice Reference Number)
 - **Main Scenario:** Ensuring Accurate IRN Generation
 - **Sub Scenario Explanation:** Demonstrate how the system automatically populates the GSTR-1 form with the data from the IRN-generated invoices, ensuring seamless integration and compliance. This scenario is essential for simplifying return filing and ensuring accuracy.
 - **Example:** After receiving the IRN, the system automatically transfers the invoice data to the GSTR-1 form, ready for filing, ensuring that all the necessary information is accurately reported.
 - **Priority:** High
-

31. Sub Scenario Title: Handling Missing or Incorrect Data in JSON

- **Main Requirement Name:** Error Handling
 - **Main Scenario:** Identifying and Correcting Errors in JSON
 - **Sub Scenario Explanation:** Demonstrate how the system responds when the JSON file submitted to the IRP contains missing or incorrect data, such as an invalid GSTIN or a missing mandatory field. The scenario should show how the system identifies the specific errors, provides detailed error messages, and guides the user in correcting the data. Once the errors are corrected, the JSON is revalidated and resubmitted until it passes the IRP's checks.
 - **Example:** The system identifies that the recipient's GSTIN is missing in the JSON file. It flags the error, prevents submission, and guides the user to input the correct GSTIN before resubmitting.
 - **Priority:** High
-

32. Sub Scenario Title: Real-Time Error Notifications and Resolution

- **Main Requirement Name:** Error Handling
 - **Main Scenario:** Identifying and Correcting Errors in JSON
 - **Sub Scenario Explanation:** Highlight the system's ability to provide real-time notifications to the user when an error is encountered during the JSON upload process. The scenario should show how the system immediately informs the user of the issue, explains the nature of the error, and offers step-by-step guidance to resolve it. This proactive error handling is essential for ensuring quick resolution and preventing delays in obtaining the IRN.
 - **Example:** During JSON upload, the system detects an incorrect HSN code and instantly alerts the user via a pop-up message, detailing the error and providing steps to correct it.
 - **Priority:** High
-

33. Sub Scenario Title: Automated Re-Submission After Error Correction

- **Main Requirement Name:** Error Handling
- **Main Scenario:** Identifying and Correcting Errors in JSON
- **Sub Scenario Explanation:** Show how the system automates the re-submission of the JSON file after the user has corrected any errors identified by the IRP. The scenario should include the validation process to ensure that the corrected JSON now meets all the IRP's requirements, followed by a successful upload and receipt of the IRN. This feature is critical for streamlining the error correction process and minimizing the effort required from the user.

- **Example:** After correcting a GSTIN error, the system automatically re-validates the JSON and resubmits it to the IRP, successfully obtaining an IRN without requiring further user intervention.
 - **Priority: High**
-

34. Sub Scenario Title: Detailed Audit Trail for Error Handling

- **Main Requirement Name:** Error Handling
 - **Main Scenario:** Identifying and Correcting Errors in JSON
 - **Sub Scenario Explanation:** Show how the system maintains a detailed audit trail of all errors encountered during the JSON upload process. The scenario should demonstrate how users can review past errors, the corrective actions taken, and the final successful upload. This feature is crucial for ensuring transparency and accountability in the error-handling process.
 - **Example:** The system logs each error encountered during the JSON submission, recording the time, user, and specific issue, allowing auditors to review the history of corrections and submissions.
 - **Priority: Medium**
-

35. Sub Scenario Title: Automated Error Reporting and Analytics

- **Main Requirement Name:** Error Handling
 - **Main Scenario:** Identifying and Correcting Errors in JSON
 - **Sub Scenario Explanation:** Demonstrate how the system generates automated reports and analytics based on the errors encountered during JSON preparation and upload. The scenario should show how these reports help businesses identify common issues and take proactive measures to prevent them in the future. This feature is essential for continuous improvement and reducing error rates.
 - **Example:** The system generates a monthly report highlighting the most common errors encountered in JSON submissions, allowing the company to focus on training or process improvements to reduce these errors.
 - **Priority: Medium**
-

36. Sub Scenario Title: Scenario for Managing Errors in Bulk Uploads

- **Main Requirement Name:** Error Handling
- **Main Scenario:** Identifying and Correcting Errors in JSON
- **Sub Scenario Explanation:** This scenario demonstrates how the system handles errors when multiple JSON files are uploaded in bulk. The demo should show how the system identifies errors in specific files, processes the correct files, and allows for the correction

and re-upload of the erroneous files without affecting the entire batch. This capability is vital for businesses that deal with high volumes of invoices and need efficient error management.

- **Example:** The system processes a bulk upload of 100 invoices, identifying 10 that contain errors. These 10 files are flagged and separated, allowing the remaining 90 to be successfully uploaded while the errors are addressed separately.
 - **Priority: High**
-

37. Sub Scenario Title: Handling Auto-Cancellation of Invoices

- **Main Requirement Name:** Error Handling
 - **Main Scenario:** Managing Unexpected Issues Post-Submission
 - **Sub Scenario Explanation:** Illustrate how the system manages auto-cancellation of invoices that were mistakenly uploaded or require corrections, including the process for re-issuing and updating GSTR-1. This scenario addresses the challenges of managing mistakes in a regulated environment.
 - **Example:** An invoice is mistakenly uploaded and auto-canceled due to incorrect data. The system allows for quick re-issuance of the corrected invoice and automatically updates GSTR-1 with the new details.
 - **Priority: High**
-

38. Sub Scenario Title: Managing API Failures During IRN Generation

- **Main Requirement Name:** Error Handling
 - **Main Scenario:** Managing Unexpected Issues Post-Submission
 - **Sub Scenario Explanation:** Show how the system handles API failures during IRN generation, including retry mechanisms and user notifications. This scenario ensures that technical issues do not disrupt compliance.
 - **Example:** An API failure occurs while generating the IRN. The system automatically retries the request and, if successful, notifies the user. If the retry fails, the system alerts the user to intervene manually.
 - **Priority: High**
-

Sample Data

Serial Number	Sub Scenario Title	Sample Data Preparation
1	Validating Mandatory Fields	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "InvoiceDate": "2024-08-17", "TotalInvoiceValue": "50000.00", "RecipientGSTIN": ""}</p> <p>Note: Missing RecipientGSTIN</p>
2	Validating Optional Fields	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "InvoiceDate": "2024-08-17", "Discount": "10%"}</p> <p>Note: Optional field "Discount" added correctly</p>
3	Schema Compliance Check	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "HSNCode": "12345"}</p> <p>Note: Incorrect HSN code length (5 digits instead of 6)</p>
4	Adding Custom Fields for Retail	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "InvoiceDate": "2024-08-17", "LoyaltyPointsEarned": "100"}</p> <p>Note: Retail-specific field "LoyaltyPointsEarned" added</p>
5	Customizing JSON for Manufacturing	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "InvoiceDate": "2024-08-17", "BatchNumber": "BATCH123"}</p> <p>Note: Manufacturing-specific field "BatchNumber" added</p>

6	Validating Custom Fields	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "InvoiceDate": "2024-08-17", "DiscountCode": "DISC10"}</p> <p>Note: Validating custom field "DiscountCode"</p>
7	Data Entry Automation	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "InvoiceDate": "2024-08-17", "TotalInvoiceValue": "50000.00"}</p> <p>Note: Data auto-populated from ERP</p>
8	Automated Validation	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "InvoiceDate": "08/17/2024"}</p> <p>Note: Incorrect date format detected</p>
9	Handling Discrepancies	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "ItemRate": "18%"}</p> <p>Note: Discrepancy in GST rate, prompting correction</p>
10	Preparing JSON in Multiple Languages	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "InvoiceDate": "2024-08-17", "ItemDescription": "वस्त्र"}</p> <p>Note: JSON prepared in Hindi</p>
11	Translation Accuracy Check	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "ItemDescription": "பருத்தி"}</p> <p>Note: Item description translated to Tamil, accuracy checked</p>

12	Schema Validation	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "ItemDescription": "సూక్ష్మ వస్తువులు"}</p> <p>Note: JSON validated after translation to Telugu</p>
13	Currency Conversion to INR	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "InvoiceValueUSD": "1000.00", "InvoiceValueINR": "74000.00"}</p> <p>Note: Currency conversion from USD to INR applied</p>
14	Retaining Original Currency Details	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "InvoiceValueUSD": "1000.00", "InvoiceValueINR": "74000.00"}</p> <p>Note: Original and converted currency values included</p>
15	Validating Multi-Currency JSON	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "InvoiceValueUSD": "1000.00", "InvoiceValueINR": "74000.00"}</p> <p>Note: Validating currency conversion details</p>
16	Implementing HSN Code Requirements	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "HSNCode": "123456"}</p> <p>Note: Correct HSN code length implemented</p>
17	Handling Government Agency Invoices	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "GovAgencyCode": "GOV123"}</p> <p>Note: Government agency-specific field "GovAgencyCode" added</p>

18	Ensuring Latest Compliance	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "InvoiceValue": "100000.00", "SpecialDisclosure": "Yes"}</p> <p>Note: Latest compliance updates applied</p>
19	Seamless API Integration for JSON Upload	<p>Data: {"APIStatus": "Connected", "UploadStatus": "Successful"}</p> <p>Note: JSON file successfully uploaded via API</p>
20	Direct Portal Upload with Real-Time Feedback	<p>Data: {"PortalUploadStatus": "Error", "MissingField": "RecipientGSTIN"}</p> <p>Note: Real-time feedback indicates missing field</p>
21	Bulk Uploads of Multiple JSON Files	<p>Data: {"TotalFiles": "50", "SuccessfulUploads": "45", "FailedUploads": "5"}</p> <p>Note: Bulk upload results with 5 failed files</p>
22	Handling Interrupted Upload Sessions	<p>Data: {"UploadStatus": "Interrupted", "ResumptionStatus": "Successful"}</p> <p>Note: Upload resumed and completed after interruption</p>
23	User Role-Based Access for Uploads	<p>Data: {"UserRole": "Admin", "UploadAccess": "Granted"}</p> <p>Note: Only admin has upload access</p>
24	Time-Bound Invoice Reporting	<p>Data: {"InvoiceDate": "2024-07-17", "ReportDeadline": "2024-08-16", "AlertStatus": "Issued"}</p>

		Note: System alerts nearing reporting deadline
25	Automatic Generation of IRN and QR Code	Data: {"IRN": "IRN1234567890", "QRStatus": "Generated"} Note: IRN and QR code generated successfully
26	Verification and Archival of IRN and Signed JSON	Data: {"IRN": "IRN1234567890", "ArchiveStatus": "Completed"} Note: IRN and signed JSON archived for future reference
27	IRN Retrieval in Case of System Failure	Data: {"IRN": "IRN1234567890", "RetryCount": "2", "FinalStatus": "Successful"} Note: IRN retrieval successful after retries
28	Notifications for Successful IRN Generation	Data: {"IRN": "IRN1234567890", "NotificationStatus": "Sent", "Recipient": "FinanceTeam"} Note: Notification sent after successful IRN generation
29	Archiving and Retrieval of Signed e-Invoice	Data: {"IRN": "IRN1234567890", "ArchiveStatus": "Completed", "RetrievalStatus": "Successful"} Note: Signed e-Invoice archived and retrieved successfully
30	Auto-Population into GSTR-1 from IRN	Data: {"IRN": "IRN1234567890", "GSTR1Status": "Auto-Populated"} Note: GSTR-1 auto-populated from IRN

31	Handling Missing or Incorrect Data in JSON	<p>Data: {"SupplierGSTIN": "27AAEPM1234A1ZZ", "RecipientGSTIN": "", "Error": "MissingRecipientGSTIN"}</p> <p>Note: Error flagged for missing recipient GSTIN</p>
32	Real-Time Error Notifications and Resolution	<p>Data: {"HSNCode": "12345", "Error": "InvalidHSNLength", "ResolutionPrompt": "CorrectHSN"}</p> <p>Note: Real-time notification for incorrect HSN code</p>
33	Automated Re-Submission After Error Correction	<p>Data: {"ErrorStatus": "Corrected", "ResubmissionStatus": "Successful"}</p> <p>Note: JSON resubmitted after error correction and successfully processed</p>
34	Detailed Audit Trail for Error Handling	<p>Data: {"ErrorID": "E12345", "CorrectionDate": "2024-08-18", "CorrectionStatus": "Resolved"}</p> <p>Note: Detailed audit trail for error correction</p>
35	Automated Error Reporting and Analytics	<p>Data: {"ErrorType": "MissingGSTIN", "Frequency": "5", "LastOccurrence": "2024-08-17"}</p> <p>Note: Automated report highlighting common errors</p>
36	Scenario for Managing Errors in Bulk Uploads	<p>Data: {"TotalFiles": "100", "FailedFiles": "10", "ErrorType": "InvalidHSN"}</p> <p>Note: Errors identified in bulk uploads, specific files flagged</p>

37	Handling Auto-Cancellation of Invoices	Data: {"InvoiceID": "INV123456", "CancellationStatus": "Auto-Cancelled", "ReissueStatus": "Completed"} Note: Auto-cancelled invoice reissued successfully
38	Managing API Failures During IRN Generation	Data: {"IRN": "IRN1234567890", "APIStatus": "Failed", "RetryStatus": "Successful"} Note: API failure handled, IRN generated after retries