**PORTAL DEVELOPMENT PHASE SUMMARY**

The portal will be a one of its kind assessment and monitoring platform. There will be multiple modules which will have to be developed to cater all the functional requirements. Here is the list of modules to be developed. The Development phase:

1. Development of an assessment module: This includes a question bank module (a module to help create a database of questions with different response types), a form builder module, data collection module (web interface to respond to questions and upload relevant files).
2. Development of a validation module: This includes an interface to accept or reject response to a question along with remarks from one or more validators

Detailed modules are here:

|  |  |  |
| --- | --- | --- |
| **S.no** | **Module** | **Functionality** |
| 1 | Data collection modules | 1. This module should be in ODK XFormsspecification compatible. 2. Users can see a list of assessments for them to be filled along with the deadline 3. Feature to choose a language of the form 4. Identify and implement a workflow to get supporting documents and remarks associated with each response. 5. User can input the response to each question 6. Data from previous assessment year(s) associated to the question will be visible to the user while filling in the response 7. User can download a template for the supporting document available with each question 8. Option to bulk upload the responses and corresponding supporting documents using a template. 9. Ability to auto save the form as draft in the background. 10. Show progress of form completion in percentage and absolute value, in each group and overall 11. Show number of questions in each group 12. Ability to navigate between different groups 13. User is informed on unsuccessful validation of response of each question 14. At any point in time while filling the responses a user can download the PDF with all the questions and responses submitted as a preview |
| 2 | Assessment Module -Question bank | Question bank module (a database of questions shared across various users with appropriate permission) for users with specific roles to add questions with various response types:   * Select One * Select Many * Short Text * Long Text * Number (Integer) * Decimal (Float) * Date * Time * Date and time * Point (on a map) [with geofencing support that the user can optionally turn on] * Line (on a map) [with geofencing support that the user can optionally turn on] * Area (on a map) [with geofencing support that the user can optionally turn on] * Photo (With a preference to set the maximum size [MiB] and a list of supported formats) * Audio (With a preference to set the maximum size [MiB] and a list of supported formats) * Video (With a preference to set the maximum size [MiB] and a list of supported formats) * Note (no response, just a note) * Barcode / QR Code * Acknowledgement (text + a checkbox) * Rating * Question Matrix (response filled in a cell of a table) * Ranking: Ordered from a list of options * Calculated field (to be auto filled based on a custom logic and variable from some response) * Hidden (type=” hidden”, value set while building the form) * File Upload * Range * Any other response type if needed will be incorporated during the development phase   Question bank module should have a feature to group questions in a collection. Examples of collection could be economy, demographics, finance etc. Collection within the question bank can be shared with various users with read / write access to view / add questions. Every question in the bank will have a unique question owner (a type of user) associated with it. |
| 3 | Assessment Module - Form Builder | A form builder tool to create an assessment from the question bank or adding the question directly to the form. The form builder should have the following functionalities:   1. Feature to group a set of questions and set a name to the group 2. Drag and drop UI to order the questions in the form and move them across groups 3. Feature to copy a question 4. Save the form as a draft 5. User can edit a draft or a published form 6. Custom logic to skip one or more questions or a group 7. Custom validation condition for each question. Validations can be set through a UI or added as a script. 8. User can preview the form before publishing it 9. User can invite people through emails to fill the assessment / form 10. Users can set a form layout style. Up to 5 custom form layout designs will be designed and developed along with integrating them in the data collection module. 11. Users can customize the name of the data point (or column name) for each question or group. 12. Feature to add media (images, audio, or video) along with the text to a question. 13. Feature to disable a question (“<input disabled>”) 14. Feature to add metadata to the form like deadline, title, description etc. 15. Feature to add metadata to every question. This includes description (string), data sources (string), reference period (string) and a template for the supporting documents (attachment as a spreadsheet and / or document). 16. Create and implement an efficient workflow for the submission of the supporting documents for each question. 17. Form creator can grant permissions (view / edit forms, view / add / edit / validate / delete submissions) to specific users 18. Feature to connect two or more forms to autofill data in the child form based on a calculated value from a parent form 19. Feature to automatically post submissions to an API endpoint   This assessment module should also have   1. Data preview functionality for submissions which consist of :  * Data preview in a tabular format with edit, show,delete functionalities for every row in the table * Preview all media submitted as a response to the assessment * Preview all GIS data submitted as a response on a map  1. Form summary to show total submission and number of submissions each day. 2. Feature to add multiple language translations for each string in the form with compatible web font for each language   This Assessment Module should be developed in ODK, which is an open-source tool |
| 4 | Admin Data Validation (JS form Validation, Admin role, Validator role and Question role) | This module is part of a three-layer response validation:  a. First Layer: Within the frontend (JavaScript Form Validation) as given in the custom validation logic in the Form Builder Module  b. Second Layer: Validation using a data pipeline that will be triggered on each submission (as given in Data Processing requirement in the table below)  c. Third Layer: Data validation by a human by accepting /rejecting responses after reading them. The feature in this layer is given in this section.   1. The validator module will support multiple rounds of validation 2. The validator module will have three types of users: admin, validator and question owner. 3. An admin user can create a job by assigning a subset of responses or collections (as per the question bank, for example all the economy questions to one validator) to one or more validator user(s). The same job will be maintained across different rounds of validation.      1. A validator user will see a list of all the jobs assigned along with other metadata for each job of rounds of validation completed, statistics of validation (questions in “in progress”, “approved”, “not approved” and “reverted with comment” state) etc. 2. A validator user can view all the questions within a job. The user can also filter the questions associated with tags, collections, and owners. 3. Once the validator clicks on a question, all responses for that question will be made available along with options to “approved”, “not approved” and “revert with comment” (with an input box) for every response. By default, all the responses will be in “in progress” state. 4. An option to select multiple responses and assign the three statuses above will be given to the validator user. 5. Validator can view all the responses to a question from the previous cycle of the assessment in the past years. 6. Supporting documents to any response should be shown side by side either in a split view or something like ease validation. 7. Once the validator user has done validating, the user can forward the responses to the “question owners”. 8. A “question owner” will see the list of all the jobs but will only be able to view the questions owned by the user within each job. 9. A “question owner” can overwrite the status of each response with a log maintained and visible for all the changes. 10. Once the validation is completed by all the question owners, all the questions (irrespective of the user who reviewed) corresponding to the responses with status “revert with comment” will be sent to the respondent for resubmission. 11. Once the responses are resubmitted, validation through the same job can restart from ‘b’. Appropriate statuses can be assigned to each job (eg: “pending”, “ongoing”, “completed”) to inform the validator and question owner. 12. A notification dropdown can be used to inform a user about new jobs, update in the status of the job etc. 13. For all responses with status “revert with comment”, the user who filled the assessment will be informed through a notification on the platform, email or SMS and a resubmission for that specific question will be requested 14. A user who filled the assessment will be able to see all “revert with comment” responses and resubmit a fresh one. Old responses will be visible to the user during resubmission and a log for all the submissions will be saved. 15. Users can download responses along with the supporting documents in bulk in Excel, CSV or any similar format. |
| 5 | UI/UX requirement | The UI is expected to be modern with KPIs such as time to fill the assessment (ms/question), time to add a question to an assessment, time to read through all the questions in the assessment etc. |
| 6 | Role Based access control requirement | 1. All components including assessment, validation, API endpoints, dashboards, visualization or data downloads will be exposed to one or more users though assigned roles. 2. These roles can be specific to each module like a role to create assessment, a role to view assessment, a role to submit response to an assessment. Such roles shall be defined in all the components mentioned in the above point. |
| 7 | API Requirement | Two set of API to be developed with access control:   1. Data Sharing: To facilitate egress of processed and machine-readable data from portal7. API should be India Urban Data Exchange (IUDX) and API Setu compatible. b. 2. Operations: These APIs should control and facilitate the operations of the overall AMPLIFI 2.0 platform. This includes but is not limited to an endpoint to trigger a data pipeline, an endpoint exposing data in the operations dashboard etc. More endpoints will be identified during the design phase of AMPLIFI 2.0 |