SRS - Data Extraction & Visualization of Form-Like Structured Documents

Darshan Satra, Nikhil Sharma, Param Shendekar, Vishal Salgond

1st October, 2021

Contents

1 Introduction

1.1 Product Overview

Various enterprises have massive amounts of scanned forms that needs to be processed every month and added to the enterprise database.The data is also used for generating insights for the organization. All this needs a lot of time and effort in manual data entry into the computer system and then further analysis. This involves lot of manual repetitive tasks which take up a lot of manpower. In enterprises the data that is present in the form is mostly printed text apart from the signature, this gives us a window to utilize the advancements in the area of OCR, natural language processing, etc to deliver a product that will automate the manual, cumbersome and error-prone process and generate insights with the data that we have received

Using text extraction technique, machine learning and data visualization we intend to provide a solution which automates the manual data entry by extracting information from the uploaded forms and store it in the database of the organization. In addition, we will also be using sentiment analysis on relevant fields and using data analysis to produce meaningful insights for the organization and visualize it using data visualization techniques.

2 Specific Requirements

2.1 External Interface Requirements

2.1.1 User Interfaces

- Landing page will provide information about the platform.
- Register page for creating new account will be provided.
- User can use the Login Page to log into his/her existing account.
- New Project creation page will display previous projects as well as feature to add new project.
- Project creation page will ask for the project title and the empty form.
- Value type for each field will be asked.
- Dashboard Page with visual analysis of data.
- Page to add any amount of forms to provide insights about the extracted data.

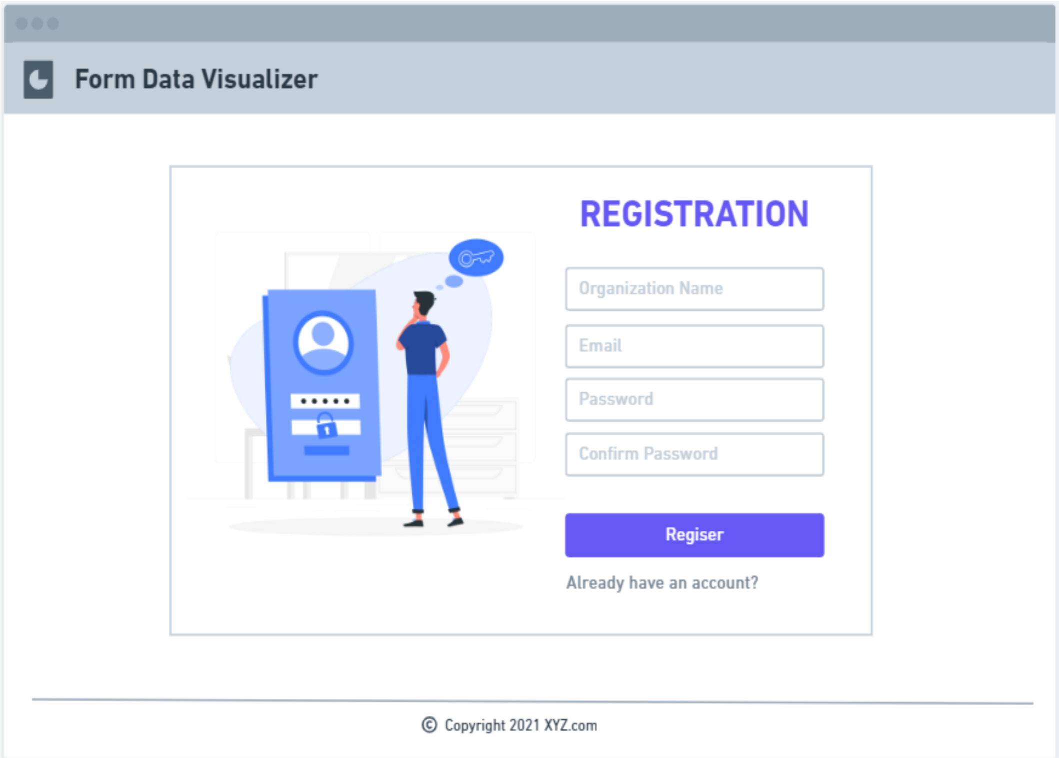


Figure 1: Registration Page

When visiting the website for the first time, a user will have to first of all register himself/herself on the website. This function will be facilitated by the Register Page of the website. User details like Organization Name, Email, and Password will be taken as input from the user, and stored into the database collection. These details will be unique to that particular user and will be used to streamline further process.

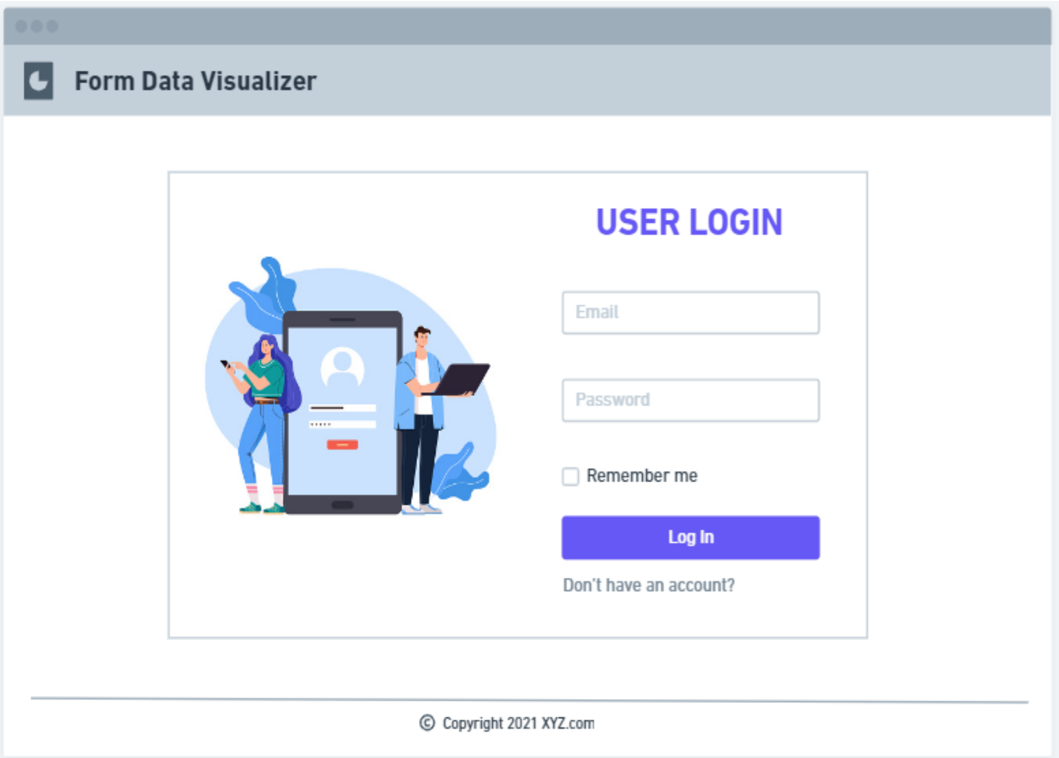


Figure 2: Login Page

Once registered with the system, or an existing user returning to use the platform will have to Log In to the platform by providing his credentials which he/she used while registrations. After authenticating the user, they will then be given access to the rest of the system.

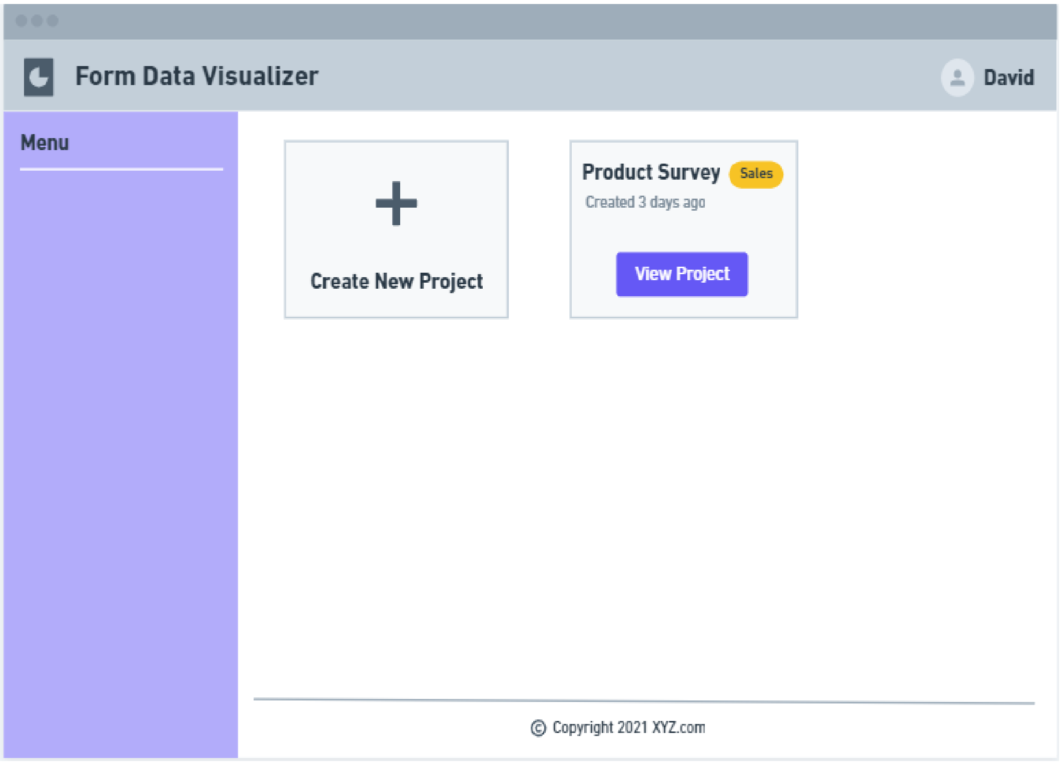


Figure 3: Project List

Once the user logs in, they can view all the projects that they have created. Each project will have a different type of empty form given while creation of the project. User will be free to visit a previous project or create a new one.

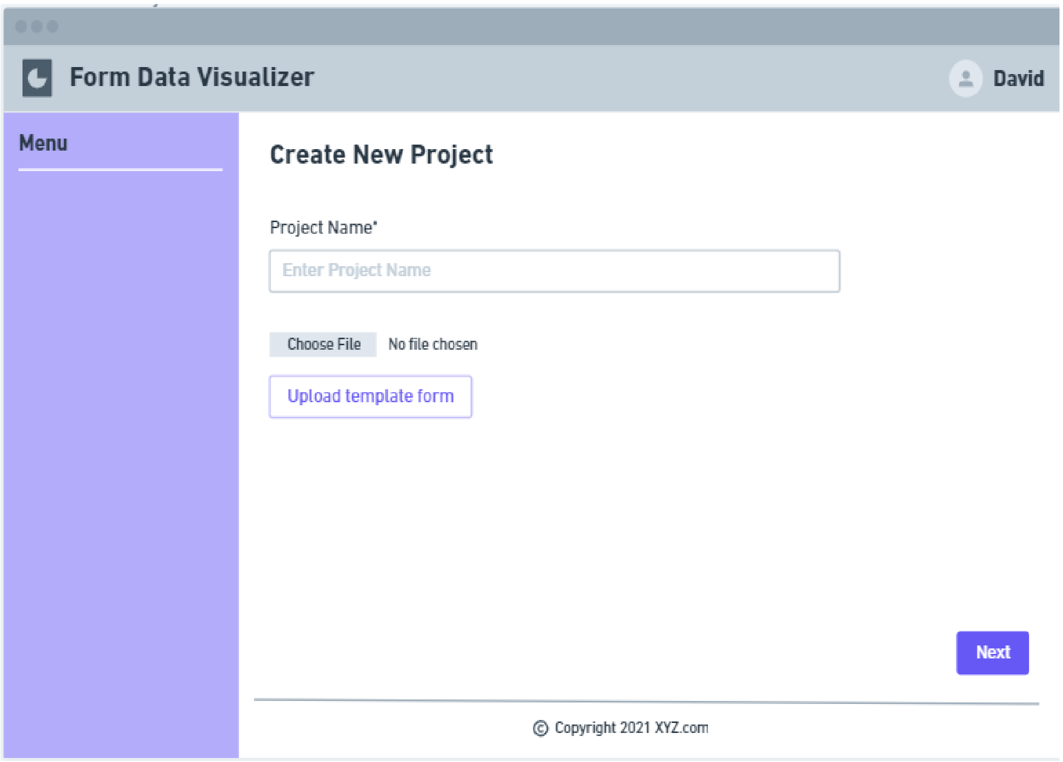
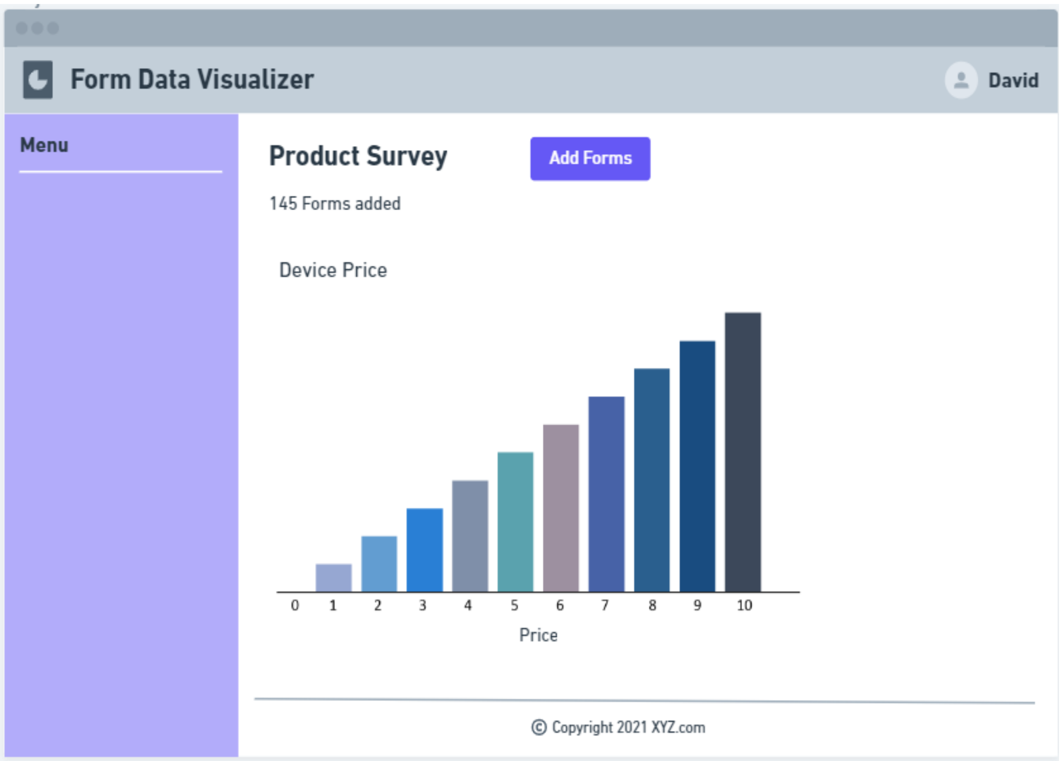


Figure 4: New Project Creation Page

When the user will be logged in to his/her account, they will be presented with the New Project Creation Page after having selected the 'Create New Project' in the previous page. On this page, a Project Name will be asked for and a blank template of the desired form will be taken from the user.



145 Forms added

Figure 5: Insights & Visualization Page

Once data is extracted from bulk documents uploaded by the user, after the processing is done, we will then display the insights generated via the extracted data to the user based on their selected fields. The visualization will depend on what kind of field is it, and the kind of responses it has received.

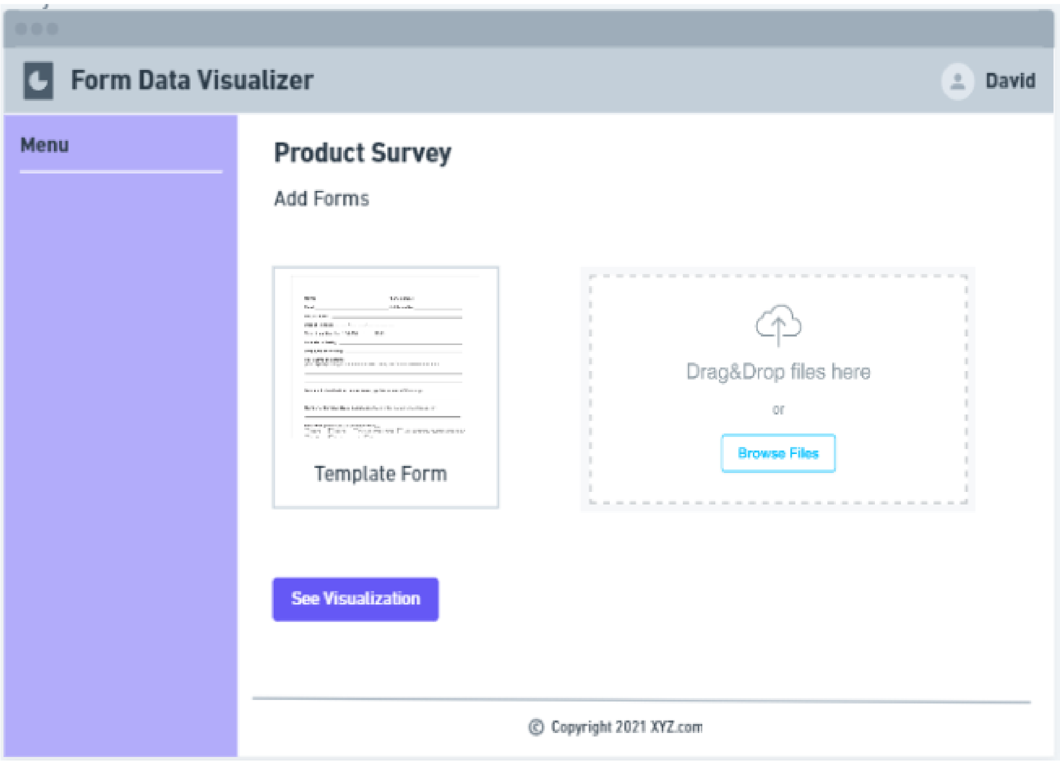


Figure 6: Add Forms Page

This page will provide the user to add the subsequent forms the user wants to analyze. The user can pick the forms from the file picker or can just drag and drop the forms into the drop box. On submitting, the user will be able to see the required analysis in the dashboard.

2.1.2 Hardware Interfaces

Each user must have access to the standard browser. To access the website user should have access to the internet to make http calls.

2.1.3 Software Interfaces

The following software components shall be used and be compatible with the application:

- Node v14
- Python3 and pip
- MongoDB v5
- Editor for executing React.JS web-app and Django server

2.1.4 Communication Interfaces

- REST API shall be used to communicate with the client over HTTP/HTTPS protocol.
- Data shall be stored in the form of JSON objects.

2.2 Software Product Features

1. Account Creation and Modification
   (a) Description:
      i. Users will be able to register and subsequently log him/herself in, and fill in details about their organization for which they're trying to perform the tasks for.
      ii. Profile page will also act as a settings page, wherein users will be able to modify their personal details like name, password, email-id, etc.

   (b) Stimulus/Responses:
      i. User details will be asked while logging in to the application, in case of no registered account, the system will respond by redirecting to the Registration page.
      ii. On proper registration and logging in, a new user will be automatically redirected to create a project.
      iii. After successful modification on the profile page, the system will now display the new details and the same will be modified in the database.

   (c) Functional Requirements:

   - i. The system shall do input sanitation checks, both while logging in and registering and hence ensure no malicious code/text is injected into the application.
   - ii. In case of any errors while doing so, a pop-up message will be displayed conveying the fault to the user, and what is actually required.

2. Project Creation

   (a) Description:

   - i. User once logged in will now be able to use the platform to its full potential.
   - ii. The first step of this process would be to create a Project, which will basically pertain to a particular single form template.
   - iii. Later, the blank template form will be uploaded, post which the user will be prompted to convey the data type of each field identified and the fields on which they want visualization and insights.

   (b) Stimulus/Responses:

   - i. User will be prompted to name the project, and add a blank template of the form in JPEG or PNG format.
   - ii. The blank form will be analyzed and fields will be identified.
   - iii. These fields will then be presented to the user along with options to choose it's datatype from pre-defined data types.

   (c) Functional Requirements:

   - i. Users should be able to create a project and add a template form for the project.
   - ii. The system shall recognize all the fields in the template form.
   - iii. Once the fields are recognized, user will be prompted with a list containing all the fields and the user is expected to select a type for each of the fields.

3. Analyze Forms

   (a) Description:

   - i. Once the user has created a project, they can input forms in bulk to get the required analysis.
   - ii. User would need a project for to achieve this feature, inside a project, an option will be given to add forms.
   - iii. Once the forms have been added, the system will extract the data out of those forms and store the data for further analysis.

   (b) Stimulus/Responses:

        i. User will be asked to enter the forms as images. The user can add forms in bulk or one by one. The system will start processing the forms as soon as the user uploads the forms and clicks on next.
        ii. After successfully uploading the forms of correct format, the user will be shown the number of forms processing and processed. The data will then be used in the analytics section.

    (c) Functional Requirements:
        i. The user should be able to submit forms for a specific project, without following any complex process.
        ii. Once the forms are submitted, the system should extract the text and provide valuable insights.
        iii. The form should have the same format as the blank form uploaded, if it is not the user will be shown an error regarding the same.
        iv. If the form is not of the correct type or the size is too large the user will be shown an appropriate error.

4. Form Data Visualization and Analytics

    (a) Description:
        i. User - once the project has been created and the forms have been submitted, should be able to see the visualization of the data from those submitted form.
        ii. User can view various type of visualization for different type of data.
        iii. The visualization will be mostly shown using graphs for better insight gain.

    (b) Stimulus/Responses:
        i. The system shall search the database to find out all the fields where the visualizations can be shown.
        ii. For each of the fields recognized in the first step, the system shall show analytics and visualization for those fields.

    (c) Functional Requirements:
        i. The visualization will be appropriate to the kind of data it is.
        ii. The visualizations will be responsive to various screen sizes. Data should be in format that will be necessary to feed to the Visualization module.

2.3 Software System Attributes

2.3.1 Reliability

- User data should be safely stored in the database
- The data should be backed up and restored in case of server failure
- The system should not break when processing large amount of data

2.3.2 Availability

The platform will be highly available and ready to use for all kinds of users at all the times of the day. There will be no hindrance in performance most of the times, and updates would contribute to minimum down time of the platform.

2.3.3 Security

- Passwords should be encrypted.
- All input data will be validated and complete mediation will be present.
- All sensitive strings should be stored in environment variables and not be visible in source code
- All sensitive data will be accessed only through proper authorization and authentication
- Data of one company should not be visible to another company on the platform.
- Data of the company should not be shared with third party platforms.
- The option of hard deletion of data should be available to the companies and it should erase all data on our servers.

2.3.4 Maintainability

- Modular development will be followed.
- Error & Bug fixing will be supported for the platform to run seamlessly.
- Capability enhancement & Adaptiveness to new technology as per requirements will take place.

2.3.5 Portability

The website should be compatible with all the modern browsers and most of the legacy browsers.

2.3.6 Performance

- User will be allowed to add 1000 forms per projects.
- Size of each form should be less than 4 MB.
- Batch size of 100 will be supported for uploading forms.

2.4 Database Requirements

- The database shall hold integer, varchar and datetime values.
- Data Validation should be performed while storing user data.
- Completed transactions should be committed into the database while failed/unfinished transactions should be rolled back.

All the data shall be stored in the database as collections, namely:

1. Users:
    - Every user will be stored as a JSON object in a document.
    - A unique ID will be generated automatically for each user object which will be used to refer their projects.
    - Every user object shall have the following attributes stored as key-value pairs.
        - Id: This field will be used to uniquely identify each user
        - First Name: Each user will have their first name.
        - Last Name: Each user will have their last name.
        - Email Id: This field will store the email id of the user as text.
        - Password: This field will store the password in an encrypted format so that it remains secure. The system will use a modified version of text to hash passwords.
        - Project List: Each user will be associated with certain number of projects that the user has created. This field will store all those projects.
2. Form Blueprint:
    - Every formtype will have a blueprint associated with it.
    - A unique ID will be generated automatically for each blueprint object.
    - Every form blueprint object shall have the following attributes stored as key-value pairs.
        - Id: This field will be used to uniquely identify each blueprint
        - Fields: An array of (key : string, type: Enum (Boolean/Alphaneumeric/Number), sentiment analysis: Boolean)

    - Project Id: unique identifier of project

3. Forms Data:
    - The extracted data from the forms will be stored here
    - A unique ID will be generated automatically for each user object which will be used to reference their projects.
    - Every user object shall have the following attributes stored as key-value pairs.
        - Id: This field will be used to uniquely identify each user
        - Fields: An array of (key, number value, Boolean value, alpha value, sentiment score)
        - Project Id: unique identifier of project Id
        - Blueprint Id: unique identifier of blueprint
4. Project:
    - When a user creates a project, the system store that project.
    - A unique ID will be generated automatically for each project object which will be used to uniquely refer this project.
        - Every project object shall have the following attributes stored as key-value pairs.
        - Id: This field will be used to uniquely identify each user
        - Name: Each user will have their first name.
        - User Id: unique identifier.
        - Form List: List of form data id.
        - Form Blueprint: unique identifier of blueprint id .