Intelligent Form Generator Using Expert Systems

KGH Piumali[#1], Budditha Hettige[2]

[1]Department of Computer Science, Faculty of Computing, General Sir John Kotelawala Defence University, Sri Lanka
[2] Department of Computer Engineering, General Sir John Kotelawala Defence University, Sri Lanka
[#] <36-se-0011@kdu.ac.lk>

DOI: 10.29322/IJSRP.13.01.2023.p13313
http://dx.doi.org/10.29322/IJSRP.13.01.2023.p13313

Paper Received Date: 18th November 2022
Paper Acceptance Date: 27th December 2022
Paper Publication Date: 6th January 2023

***Abstraction* -** Forms play an integral part in modern living. People fill and handle forms to accomplish various tasks in day today life. There are various developed technologies related to form handling. However, in some instances, these existing form-filling, and handling technologies are not sufficient to provide the best solutions with clear guidance for the users. The main objective of this research paper is to design, and development of a web-based forms generation solution named "Interllib Forms" that can be considered as an expert in filling forms. This proposed solution will be helpful to reduce uncertainties faced by applicants during form filling. This system was built as an expert system builder tool. Also, Interllib forms are built as a platform with the facility to create new form components, each with modules that represent the main functionalities of an expert system builder tool. Thus, "Add Questions", and" Add Results" modules were built to provide a knowledge acquisition functionality. The "Map and Submit" module was built to map each "Questions and Results" and build rules using a specific calculation. API keys and Links module was built to generate a unique API key for each form component which is embedded in all the Questions and Results. Also, a specific logic was used for inferencing mapped rules relevant to the applicant's queries. This acts as an intelligent feature. The "Interllib Forms" -web application is tested with user support and can be concluded that 70.8% of end users have positive feedback regarding the acceptable level. And 53.3% of knowledge experts have moderate feedback regarding the logic used in Question and Results mapping which will be subjected to be modified with an advanced algorithm in the future.

***Keywords* -** *Validation, Uncertainty, Form handling, Expert System.*

1. Introduction

People fill out and submit various forms for different purposes in their day today life. When we consider web interfaces, there are some standard controllers like Text boxes, Combo boxes, Radio Buttons, etc. Most of these controllers allow only one option from selected groups. Also, Controllers like list boxes have multiple-choice options. Some forms are simple as they contain very straightforward textboxes and controls. However, some of the form controls are more difficult for the user to correctly understand and fill. And when we are concerned about the validation levels of the forms, the Form level, Field level are the major levels in validating a form. Also, validating the field using any JavaScript code or a control program is the most common mechanism. Further, these, existing Form filling technologies are not sufficient to provide the best solutions and clear guidelines for the users in some situations. Note that, there is no proper mechanism to guide and give the most appropriate suggestions to the applicant to decide what he/she should select as the input value for the form control. Thus, the applicant may come with some uncertainties when filling out the form considering the exact input value for a given form control. This is a real scenario, where applicants might face uncertainties when filling out an application form. real The Electronic Green Card Program which is officially known as "the diversity immigrant Visa Program" is also referred to as the Green Card Program.

Up to 55,000 Green Cards will be granted annually to individuals born in nations with historically low rates of immigration to the United States. Denial Statistics of Green Card Applications show that the U.S. Citizenship and Immigration Services (USCIS) denies tens of thousands of applications for green cards each year. According to records, the USCIS received and rejected 81,169 of the 757,206 petitions for foreign relations (Form I-130) during the fiscal year 2021. Further, USCIS received 288,668 family-based applications to change status (Form I-485) during that time and rejected 44,181 of them. In actuality, more than 16% of applications were rejected by USCIS. The 11% of applications that USCIS usually rejects are not included in this. Applications for green cards based on family are those when the applicant is related to a citizen or lawful permanent resident of the United States.

So, one of the critical reasons behind this increased rate of Denial Statistics is the mistakes that the applicants do on the online green card application form.

According to the above facts, this research proposes the design and development of a web-based solution named "Interllib Forms" to be an expert in filling forms. This will be helpful to reduce uncertainties faced by applicants during form filling. The API key which is generated from the "Interllib-Forms" can be integrated into a web application form. The grid similar to the 2D Matrix structure will act as a knowledge base builder tool. This grid similar to a 2D Matrix structure will be an aid for users to map data entered by themselves. Mapping data will lead to creating relationships between data. So, these mapped relationships will be stored inside the database. So, the database including the logical calculations will play the role of a knowledgebase. Also, specific mathematical and logical calculations are performed next to check for matching relationships or rules for the user queries or inputs and redirect to the user interface as decisions, suggestions, or justifications.

This system will provide many benefits. "Interllib Forms", An expert system-based solution will reduce the uncertainties faced when filling a form due to not understanding the questions that the form asks the applicant. Also, the forms built embedding the "specific API Key" will guide the user to select the best options by giving explanations for each selection is done and inputting details.

"Interllib Forms "-A web-based expert system solution will be hosted online and freely accessible by any web developers and used as a generalized framework to build application forms by integrating the code snippet or the API Key that is generated by the "Interllib Forms". Also, any knowledge expert/engineer even can create mapping rules specific to a certain domain. And he/she can direct the generated code snippet or the API key for a web developer. So that the web developer can integrate that to create the application form for the client's requirements. So that, an applicant with or without an IT background can experience the functionality like explanations for their answers in an application form. So, the users of "Interllib Forms" are Web developers, applicants, Knowledge Engineers, and Knowledge experts.

The rest of the paper is organized as follows. Section 2 critically evaluates the existing form-handling technologies. Also, the later part of this section presents the technologies related to the existing web-based expert systems. Section 3 presents the design and the implementation of the system under the main modules. Section 4 presents the results gained from the testing and evaluation of the system. Section 5 discusses the security measures that were taken during and after the implementation of the system. Finally, Section 6 concludes the paper with a note on future works.

2. Related Works

The literature review covers two subsections. The comparison of 26 existing systems is covered in the first section. And this is under related work on form handling technologies (input validating technologies). Also, Under the second subsection, Related work on web expert systems is described.

Related works on form handling and validating technologies.

Various studies have been conducted on input validation technologies. This input validation can be categorized as server-side and client-side input validation. These studies have used various methods to achieve their aims and objectives.

Related work on Client-side input validating technologies.

FLAX(Saxena and Hanna, 2010) employs dynamic approaches like dynamic taint analysis to discover client-side vulnerabilities. This study suggests using taint-enhanced black-box fuzzing, a dynamic analytic technique, to find vulnerabilities in web applications. The study(Aljawarneh, Alkhateeb and Al Maghayreh, 2010) presents a new validation service based on semantic web technologies. Even if the input validation modules are circumvented, this helps to prevent security vulnerabilities at the application level and secure the online system. The paper(Levis, Helfert and Brady, 2008) describes a technique for determining whether a client-side input validation function complies with a policy. They also used dynamic slicing to retrieve input validation routines from websites automatically as part of their investigation.

XForms(Honkala and Vuorimaa, 2005) is a new graphical interface standard that focuses on input validation rather than integrated scripts. XForms extends the standard HTML form with validation and adaptation. The validation can be performed using XML Schema. The paper(Alkhalaf, Bultan and Gallegos, 2012) presents an analysis technique for checking if a client-side input validation function conforms to a given policy. The input validation policies are expressed using two regular expressions. Dynamic slicing is used to analyze to automatically extract the input validation functions from web applications and uses automata-based string analysis to analyze the extracted functions. Pex (Taneja et al., 2010) is used to verify that user input validation mechanisms in .NET web applications are accurate. The first category these functions fall under is the kind of input they validate. Next, a subset of the functions in the same class is used to test each validation function.

Client-side strategies (Kirda et al., 2006),(Vogt et al., 2007) aim to stop the leakage of sensitive data by thwarting attempts to communicate the data to other servers. These programs address XSS attack symptoms (such as a cookie-stealing script). A

form model for Portable Form Files (PFF)(Web, 2002) that is unrelated to the (X) HTML language. The user of this form has the option to lock and encrypt the form's data before sending it to a web server. A client-side encryption scheme has been suggested by Hassinen and Mussalo(Hassinen and Mussalo, 2005) to safeguard user trust, data integrity, and secrecy. They use a client encryption key to encrypt data inputs before submitting the contents of an (X) HTML Form.PowerForms(Zaremba and Smoleński, 2000) is an HTML form add-on that enables complex form field interdependencies and pure declarative specification of input formats. PowerForms' language is built on regular expressions that are placed in HTML and then converted into a mix of conventional HTML and JavaScript.

Related work on Server-side input validating technologies

XSS-GUARD (Bisht and Venkatakrishnan, 2008) is a new framework that is intended to provide a server-side preventive mechanism for XSS assaults. XSS-GUARD protects against cross-site scripting attacks by dynamically learning the set of scripts that a web application plans to produce in response to any request of HTML. IPAAS (Sunkari and Rao, 2014) is a novel technique for preventing the exploitation of XSS and SQL injection vulnerabilities based on automated data type detection of input parameters. IPAAS automatically and transparently augments otherwise insecure web application development environments with input validators that result in significant and tangible security improvements for real systems. The study (Xie and Aiken, 2006) presents a static analysis algorithm for detecting security vulnerabilities in PHP. The architecture of the algorithm enables the handling of dynamic features unique to scripting languages such as dynamic typing and code inclusion, which have not been adequately addressed by previous techniques.

MiTV(Taneja et al., 2010) is a novel approach, that applies Multiple-implementation Testing for Validators. To generate test inputs that can detect different behaviors among validators of the same type, this approach synthesizes test drivers for each pair of validators of the same type and use a test generator for structural testing to generate test inputs for the synthesized test drivers. The study (Shar and Tan, 2012) proposes a set of static code attributes that represent the characteristics of these routines for predicting the two most common web application vulnerabilities—SQL injection and cross-site scripting. The first open-source program for statically identifying XSS flaws in PHP 4 code is Pixy(Jovanovic, Kruegel and Kirda, 2006). The Pixy prototype can be used to identify other taint-style issues, such as SQL injection and command injection, even though it was created to identify XSS flaws.For validating incorrect user inputs and identifying malicious scripts, Scott and Sharp(Aljawarneh and Alkhateeb, 2009b) developed a gateway model, which is an application-level firewall on a server (e.g. SQL injection attack and cross-site scripting attack).

Huang and colleagues(Aljawarneh and Alkhateeb, 2009b) employed behavior tracking to find dangerous content before it reached users as an alternative method of self-protection. They create WAVES (Web application security assessment system), which uses behavior stimulation to provoke bad behavior in the components that are being watched. A type of qualifier-based analyzer for PHP is called WebSSARI (Xie and Aiken, 2006). It searches for instances when user-controlled values enter functions that call for trustworthy input using a conventional intraprocedural tainting analysis (sensitive functions). Three user-written "prelude" files are used in the study to offer information. A PHP taint mode was proposed by Nguyen-Tuong(Xie and Aiken, 2006). When a query involves user input fragments, it uses a set of heuristics to determine whether it is safe to utilize the query. And Offutt and Xu (Offutt and Xu, 2004) made testing for SQL injection attacks a focal area.

The set of permissible SQL instructions for each data access is modeled using finite state automata by Wasserman and Su (Saxena et al., 2010)using the static analysis technique. For validating incorrect user inputs and identifying malicious scripts, Scott and Sharp (Scott and Sharp,2003)developed a gateway model, which is an application-level firewall on a server (e.g. SQL injection attack The strategy(Balzarotti et al., 2008) is to examine the sanitization. This indicates that they used a combination of static and dynamic analytic approaches to find flaws in the sanitization processes that a criminal could exploit. There are numerous methods for identifying applications vulnerable to XSS injection attacks that rely on static analysis techniques (Livshits and Lam, 2005)(Xie and Aiken, 2006). These tools are typically made for developers to utilize when creating new code. These methods are restricted to locating sinks and sources and determining if input validation is applied to each flow between them.

In MiMosa(Balzarotti et al., no date) methodology, the vulnerabilities are referred to as multi-module vulnerabilities. It stimulates the extended state of an application to find vulnerabilities that cross modules. In particular, (Pietraszek and Berghe, 2006) make a passing mention in their discussion of the possibility that exact XSS attack detection may be achieved by imposing syntactical constraints on corrupted input. This method shows how this concept has been realized in a real-world application for XSS attack detection. In (Minamide, 2005), replacement operations are modeled using multi-track DFAs, also referred to as transducers, in conjunction with a string analysis method based on grammar. The resulting tool has proven useful in identifying PHP application vulnerabilities. In (Ashcraft and Engler, 2002), It is described that, When user-supplied integer and pointer values are used without enough validation, operating system code vulnerabilities can be found via tainting analysis.

Related works on both client-side approach and server-side input validation approach.

AdapForms(Bohøj, Bouvin, and Gammelmark, 2011) performs server-side validation while delivering continuous feedback to the client. AdapForm is a general solution for handling forms that allow developers to designate accepted types of input, reuse existing templates (e.g., sub-forms for postal addresses), use complex validation rules, etc. Also, Perturbation-based Interactive UIV Testing (PIUIVT) (Li et al., 2010) is a method for improving the quality of a web-application UIV by testing it for robustness against invalid inputs. Regular expressions are mainly used in Perturbation-based Interactive UIV Testing (PIUIVT). ViewPoints(Alkhalaf et al., 2012) is based on the idea that web application developers often perform duplicate input validation on the client and server sides, and that the validation performed on one side can be used as a specification for the validation performed on the other. ViewPoints may detect a huge number of inconsistencies in their input validation functions automatically. On the client side, NoTamper(Bisht et al., 2010) leverages dynamic symbolic execution to find limitations on HTML form inputs. The server-side input validation is then tested using these limitations to produce input values.

The study(Alkhalaf, Aydin and Bultan, 2014) presents an automated differential repair technique for input validation and sanitization functions. The differential repair can be used within an application to repair client and server-side code concerning each other, or across applications to strengthen the validation and sanitization. Semantic Data Validation (SDV)(Aljawarneh, Alkhateeb and Al Maghayreh, 2010) is a service that uses semantic web technologies to safeguard the web system even when input validation modules are disregarded and to prevent security vulnerabilities at the application level. By testing a web application's UIV for robustness against erroneous inputs, a technique called perturbation-based interactive UIV testing can be used to raise its quality.

In PIUIVT, the adaptability of proxy-based UIV testing and the automation of crawler-based UIV testing are combined. A web application's client-side data is examined by PIUIVT to obtain input-field data that helps with the automatic generation of test inputs for semantics and the generation of valid inputs. In PIUIVT, each input field has a regular expression attached to it that establishes the input field's valid-input restrictions. After that, PIUIVT perturbs the regular expression and creates invalid test inputs using the regular expressions that have been perturbed. PIUIVT, like crawler-based UIV testing systems, allows testers to manually alter the automatically generated test inputs. To automatically determine if a test passed or failed, PIUIVT compares the structural similarities of the original HTML page, the response page for invalid input, and the response page for correct input (which are automatically generated based on the regular expressions that define valid inputs).
The technologies in existing input validation tools and approaches are tabulated in Table 1.

Table 1. Technologies in existing input validation tools and approaches

Tools and Approaches	**Client-side**	**Server-side**	**Regular Expressions**	**Semantic Web Technologies**	**(Abstract Trees) /Context Free**	**Xml/xHTML**	**Static Analysis**	**Dynamic Analysis**	**String Analysis**	**Deterministic Finite Automata**
FLAX (Saxena and Hanna, 2010)	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗
(Aljawarneh and Alkhateeb, 2009)	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗
PIUIVT (Li *et al.*, 2010)	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗
The study (Saxena and Hanna, 2010)	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗
The study (Aljawarneh and Alkhateeb, 2009)	✓	✗	✓	✗	✗	✗	✗	✓	✓	✓
XSS-GUARD (Bisht and Venkatakrishnan, 2008)	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
The study(Sunkari and Rao, 2014)	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗
The study (Xie and Aiken, 2006)	✗	✓	✓	✗	✗	✗	✓	✗	✗	✗
MiTV (Taneja *et al.*, 2010)	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗

Pixy (Jovanovic, Kruegel and Kirda, 2006)	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗
Adapform (Bohøj, Bouvin and Gammelmark, 2011)	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
NoTamper (Bisht *et al.*, 2010)	✓	✓	✗	✗	✗	✗	✓	✓	✗	✗
ViewPoints (Alkhalaf *et al.*, 2012)	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓
Powerforms (Zaremba and Smoleński, 2000)	✓	✗	✓	✗	✗	✓	✗	✗	✗	✓

Related work on Web-based Expert Systems.

The study's(Wangphanich, 2011) objective was to develop a fundamental Web-Based Expert System for assessing suppliers for a renowned automotive company that uses JIT(Wangphanich, 2011). The main methodology was building a web-based expert system and gaining knowledge through expert interviews and knowledge presentations using IF-THEN logic. As a result, a tree diagram-based rule-based decision system was developed to assess six crucial factors: management, engineering, safety, quality, and cost. Then, a basic Web-Based Expert System was built using two key computer software. PHP web pages that support interactive information exchange between a user and a server, as well as between web pages and a database, have been made using Adobe Dreamweaver. A MySQL database is created using phpMyAdmin, and the resultant Expert System has been tested using Apache.

An expert system has made it possible to transmit the medical experience to a computer system as a result of improvements in technology during this time. PHP was used as the web programming language and MySQL as the database management system to create the expert system(Hogo, 2009). By simply connecting to the internet, users of the web-based expert system can diagnose children's illnesses at any time and from any location. The certainty factor approach is the one that is employed. Based on the patient's observations, the diagnosis will be made by checking off symptoms on the consultation page. The inputs will be treated under production standards founded on science and the competence of the expert, in this case, a doctor as a medical professional. This web expert system describes many stages of building the RCSES system(Hogo, 2009), such as knowledge gathering and selection, ontology, and knowledge representations in XML format. ASP.net was used to create XML-rule-based knowledge sources and inference processes. For entering the ontology, knowledge base, and inferencing, it designed an interactive interface. It enables you to use, modify, update, and broaden the current knowledge base quickly. Civil service law specialists verified the information, and the proposed RCSES was examined, checked, and verified by a wide range of technical users as well as the team of developers.

When compared with the existing form handling tools and technologies," Interllib Forms" has the functionalities to provide explanations and suggest solutions for applicants' queries. "Interllib forms" takes the structure of the "Expert System Builder tool". But, It is flexible and customizable when compared to existing web expert systems.

3. Design and Implementation

The entire design of the "Interllib Forms" – web-based application is a combination of the user interfaces for major modules like "Add Questions"," Add Results/Answers"," Map & Submit", and" API Keys & Links". These modules represent the main components of an expert system builder like Knowledge acquisition, knowledge base, and Inference Engine. A simplified overview of the system's overall design process is shown in Fig. 2.

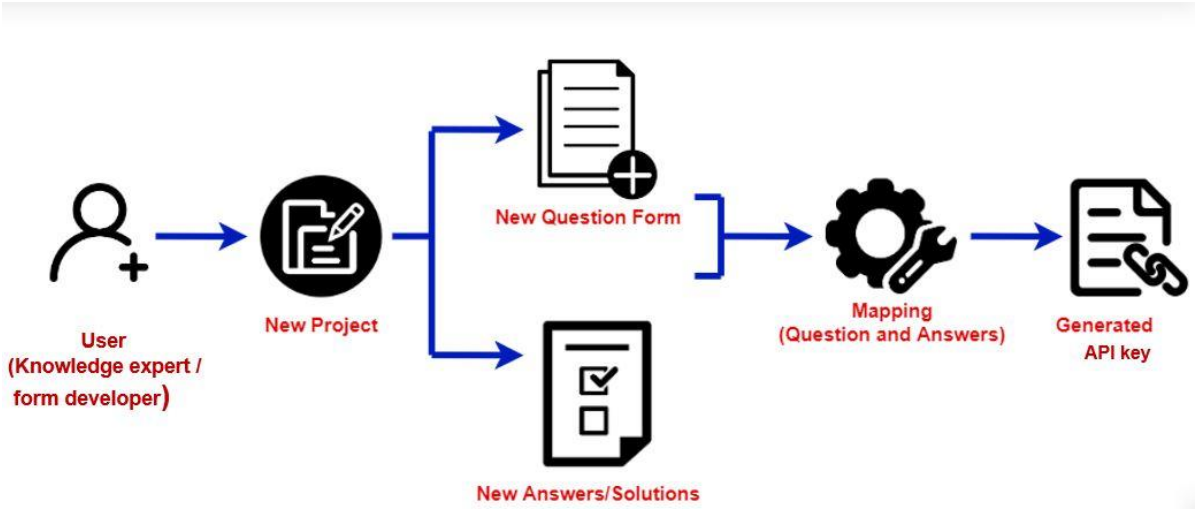


Fig. 2. Overview of the System.

The knowledge experts or knowledge engineers of a particular domain are the direct users of the "Interllib Forms". So, they can log in to the system and create a new project. This new project is also referred to as a "new form". This project has the space for adding the set of Questions and intended Results. These " questions" are the set questions that are displayed in a popped-up window(Which is discussed in the following sections). Also," intended results" are the data that the user used as input for particular questions in an application form. This process is similar to knowledge acquisition in an expert system. Questions and Results are the knowledge that is added to the knowledge base.

After the knowledge acquisition, the user can move to the most critical phase. That is to map both sets of questions and intended results/answers that the user had already added when creating a new project. "Interllib Forms" web-based applications can be used by any knowledge expert to add explanation/suggestion or guidance functionality to the application form that he/she builds; hence the user experience should be very high. To achieve this goal, this web-based utility is presented in a user-friendly way. Instead of creating "IF ELSE Rules" for creating rules, this system uses an interactive and user-friendly grid similar to a 2D Matrix (Fig. 9) for mapping questions and results. Here a color code is given for each mapping. And a weight-assigning mechanism is introduced for calculation purposes. (Similar to generating IF else rules). Also, a specific interface is built, so that any developer can integrate the functionality of the "Interllib forms" by just copying and pasting the specific API key (that is unique per each new form component created by Interllib-Forms) in their application form building environment. This key is also given as an embedded code snippet.

The implementation was carried out using two main subcategories. The first subcategory discusses the implementation of the system according to the MVC architecture and the next discusses how the expert system concepts (user interfaces, inference rules, and Knowledge base) are acquired in the implementation. Also, the implementation of the Interllib-forms is discussed under subtopics as the implementation of the web page for a dashboard, creating a new form to add questions, answers, or results, mapping both questions and Results, and Generating a unique key for the form created.

Tools and Technologies used.

The "Interllib Forms" application is developed using the Laravel framework. Laravel is an open-source PHP-based application framework. Programmers utilize it voluntarily and it is fully documented. It makes use of the Model-View-Controller (MVC) design, which makes the code clear, comprehensible, and organized and makes website building incredibly simple. When a large group of people collaborates on an application, this style of design is also advantageous. Since Laravel was created using Symfony's components, both web frameworks share the aforementioned functionalities.

And MySQL database was used as the database for "Interllib Forms". Also, a specific library called "Laravel Livewire" was used. This is a framework that makes it simple to create contemporary, reactive, dynamic interfaces. Xampp was used as the local server to test the "Interllib forms" using desktop and laptop PCs before posting the website to the main server.

Following the MVC architecture, the model classes were built to communicate with the database. Also, Laravel Migrations were used to create tables in the database. So, all the modifications at the structural level were done to each table, and created in the database.

Also, for each modal class, controllers and routes were built. Since we have to give some dynamic and live updating functionalities to the system, we use a special package called "livewire". So all the controllers are inside the "livewire" package.

3.1 The expert System concept is acquired during implementation.

Implementation of the User-Interfaces.

The dashboard of "Interllib forms" was built, so that a user (knowledge expert of a particular domain) can see all the previous projects when he is logging into the system. So, this gives the facility to create new projects(also referred to as a "new form") as well as to view, delete and configure the previously created forms. This new project creation is done using four steps.

The first in the stepper is: Fill Form Details. This interface is built to add the name and description of the project/form.

The second step in the stepper is: **Add Questions**

This interface is built to add questions. This interface acts as the role of a knowledge acquisition module. So, this is built so that any knowledge expert can provide facts in a particular domain (in this scenario "facts" are the "Questions").

The third step in the stepper is: **Add Results.**

This interface also acts the role of a knowledge acquisition module. So, this is also built-in order that any knowledge expert can provide facts in a particular domain (in this scenario "facts" are the "Results").

For example: If the "form component" or the project is built as a guide for selecting an option from a combo box, then the same set of options in the combo box can be used as the set of "Results" for the "Add Result" module.

Each "Add Questions" and "Add Results" module were built, so that the user can add or delete the facts added.

The fourth step in the stepper is **Map & Submit**.

After adding the results, will direct to the Question-and-Answer mapping grid (Fig: 7)

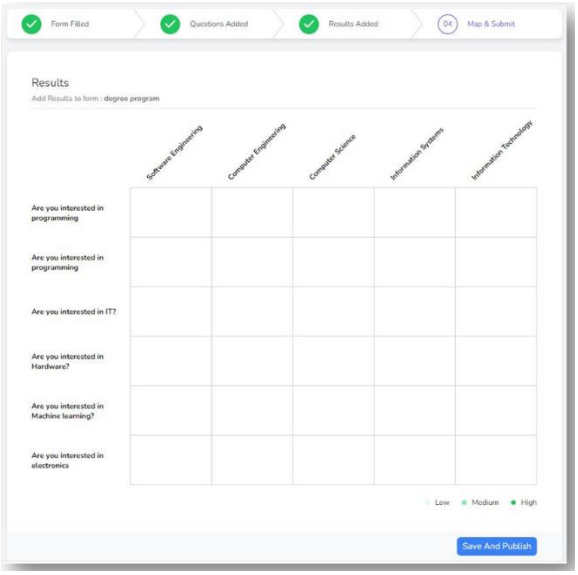


Fig. 7. Map & Submit Interface

Fig. 8. Mapped rules interface

3.1.1. Overview of the Question and Results mapping:

When compared with the architecture of an expert system builder, "Question and Results mapping "plays the role of the knowledge base. Here, the knowledge base is in an empty state at the beginning. The Questions and Results are mapped to create rules by assigning weights/values. So, the "Questions and Results mapping" section deals with a logical weight calculation related to knowledge base handling.

This interface deals with Questions and Results. Here, both Questions and Results that the user gives are looped using a foreach loop and displayed. A specific function called "generatedTableRows() is written to generate this grid. According to the table,

'X- axis ➔ Results

'Y- axis' ➔Questions.

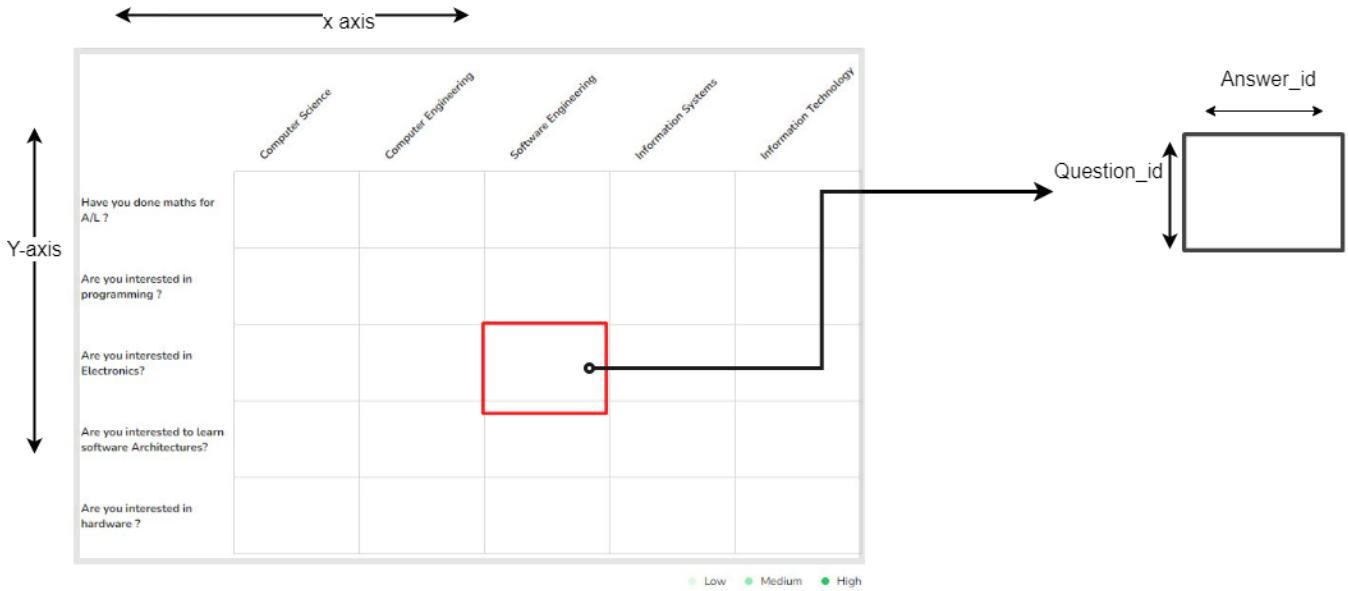


Fig. 9. Structure Similar to the "2D matrix"

Next, a function called "setResult() " is written to create and set rules for the generated table. Here the rules are set by clicking each block according to the colors that display per each click.

User's input value(per each mouse click) is set to be represented in a color code. And a weight/value is given to the particular color as below.

1st click ➔pass value '1'

2nd click ➔ pass value '2'

3rd click ➔ pass value '3'

4th click ➔ value resets.

By default, each block contains a "0" weight. For the first mouse click on the block, the weight is updated as " 1" and given color as 'bg-green-100'. Here, for each click on the same block, it is checked whether the question id and answer id are the same as the previous click Also, for the second mouse click on the same block, updates the weight as "2"..Then the color is set to 'bg-green-300'. And for the 3rd click, the weight is assigned to 3, and the color is set to 'bg-green-500'. Then for the 4th click, the value of the weight will be set to '0' (Table 2, Fig. 10).

Table 2. Colour Codes used

Color Code	Weight/Value
bg-green-100	1
bg-green-300	2
bg-green-500	3

But, the user's input values are not stored in the database directly per each click. First, the user's response (mapped rule) is set to be stored in a 2D array temporarily. Next, these user responses (mapped rules) are stored in the database, soon after clicking the Save and publish button.

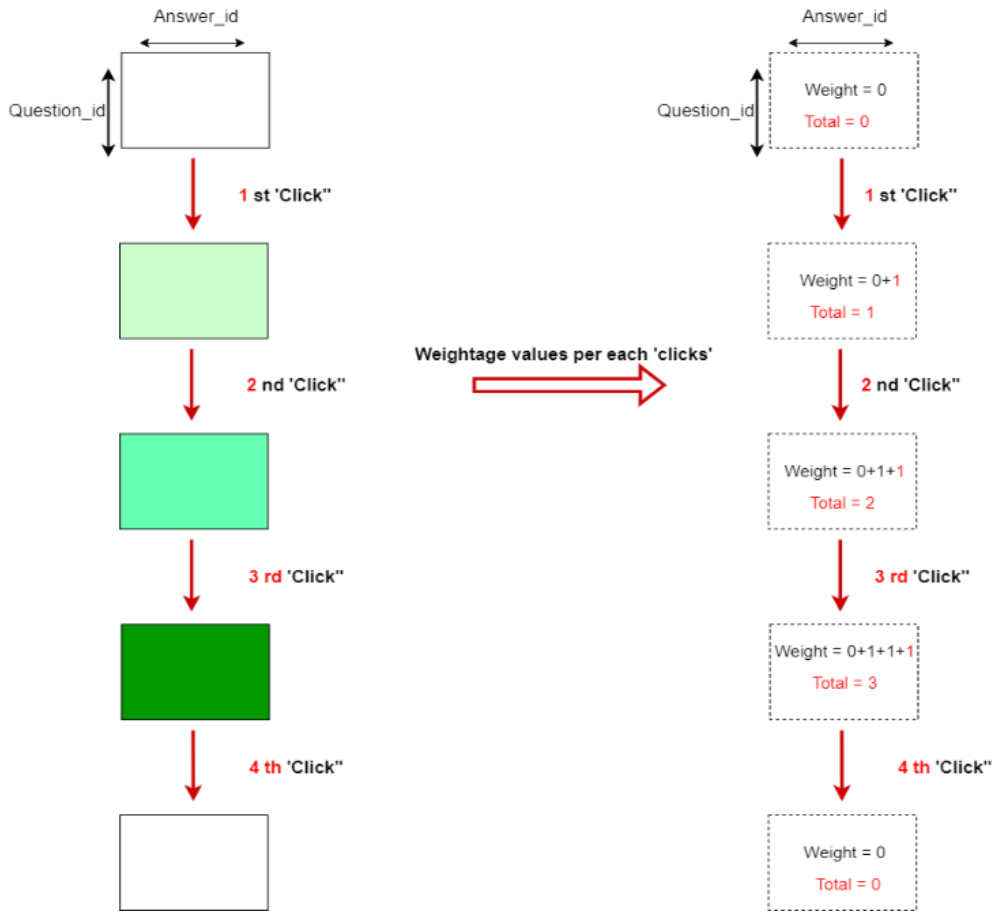


Fig. 10. Color code given per each mouse click

The interface Fig. 11 is to generate the specific API key (Secret Key) value, related to the form we built. The user is given the facility to either copy the specific key directly or an iFrame Code snippet which is embedded with the specific key of the form. Here, the form id is taken as the Secret key.

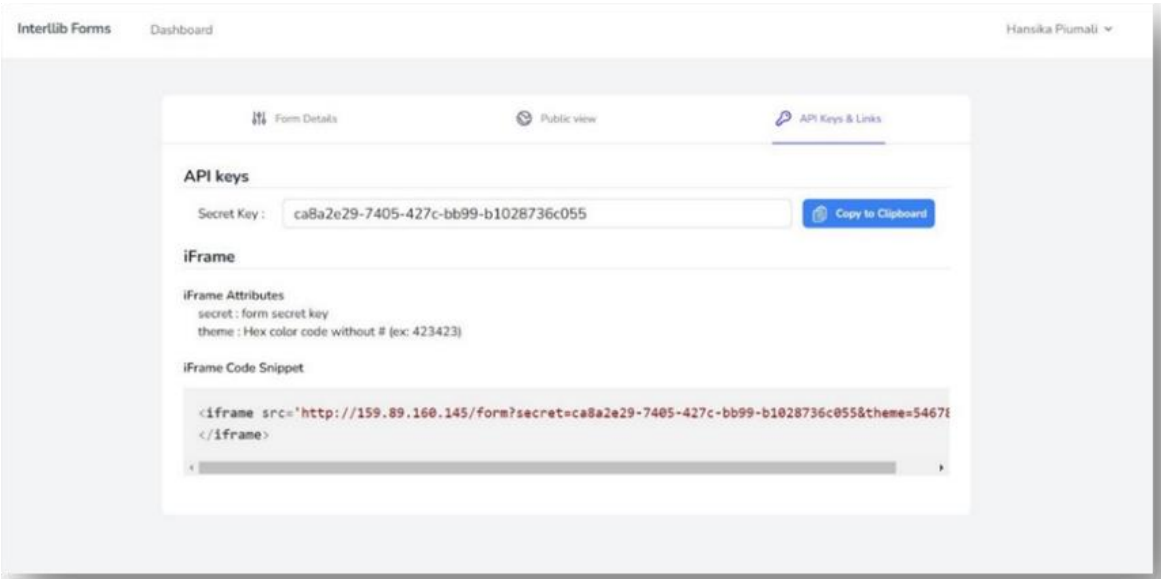


Fig. 11. API key and Links interface

The web form developer can copy and paste the code snippet into an appropriate place of the application. Finally, the applicant can view an iframe view containing the set of questions (already added by the knowledge expert in "Interllib forms") with "Yes" and "No" responses. So, when the applicant faces any uncertainty while filling out the particular application form, he /she can get guidance from the iframe view. This iframe view can be placed near the form control to be generated as a popup window for a button click. Here the applicant's "Yes" and "No" responses are taken as "user queries". Out of the queries, the," Yes" responses are stored inside an array temporarily. When the applicant clicks the "Get A Solution" button, "getASolution" function is called. This function is created to compare the applicant's queries with mapped rules and extract the most appropriate mapped rule as the suggestion or the solution (Fig. 12).

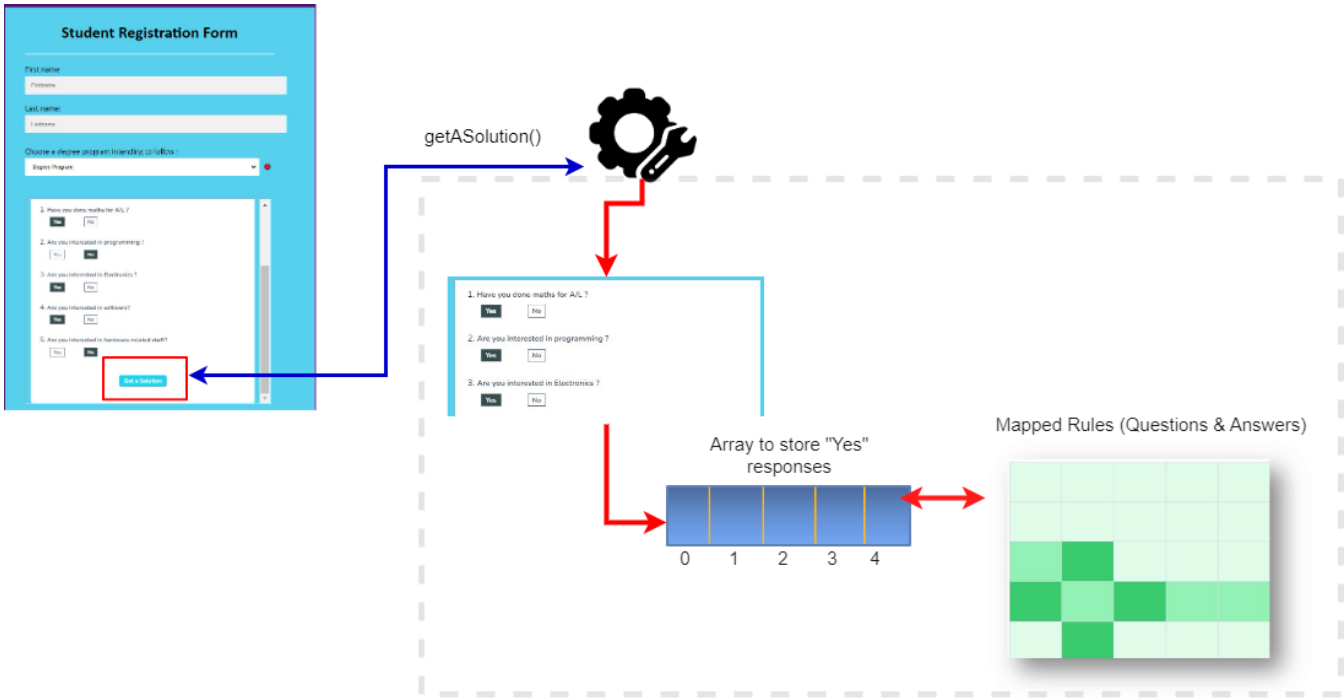


Fig. 12. Mechanism behind the retrieving the exact solution/suggestion

3.1.2. The Calculation and the logic behind the Retrieval of the exact solution/suggestion

A specific weight/value-assigning mechanism is used in Question-and-Answer mapping (as described previously). Also, a specific Calculation Fig. 13 is applied to retrieve the exact Answer from mapped rules for user's queries. Here, the "Result" that gains the maximum weight using the specific calculation will be retrieved as the exact solution for the applicant's queries.

Fig. 13. Calculation behind the retrieval of exact solution/suggestion

If the user's response is as in Fig. 14:

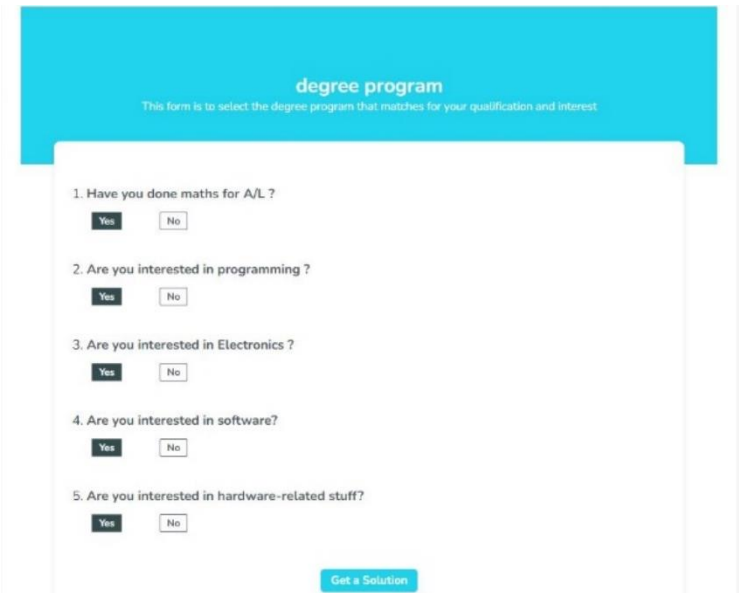


Fig. 14. Applicant's response solution/suggestion

Here's how the logic is applied:

No:	Question	All the Results	Weight Added	Total
1	Have you done maths for A/L?	Computer Science	1	1
		Computer Engineering	1	1
		Software Engineering	1	1
		Information System	1	1
		Information Technology	1	1

Fig. 15. Weight Calculation for the first question click

No :	Question	All the Results	Weight Added	Total
2	Are you interested in programming?	Computer Science	1+1	2
		Computer Engineering	1+1	2
		Software Engineering	1+1	2
		Information System	1+1	2
		Information Technology	1+1	2

Fig. 16. Weight Calculation for the second question

No :	Question	All the Results	Weight Added	Total
3	Are you interested in Electronics?	Computer Science	1+1+2	4
		Computer Engineering	1+1+3	5
		Software Engineering	1+1+1	3
		Information System	1+1+1	3
		Information Technology	1+1+1	3

Fig. 17. Weight Calculation for the third que

No :	Question	All the Results	Weight Added	Total
5	Are you interested in hardware related stuff?	Computer Science	1+1+2+3+1	8
		Computer Engineering	1+1+3+2+3	10
		Software Engineering	1+1+1+3+1	7
		Information System	1+1+1+2+1	6
		Information Technology	1+1+1+2+1	6

Fig. 18. Weight Calculation for the fourth question

No :	Question	All the Results	Weight Added	Total
4	Are you interested in software?	Computer Science	1+1+2+3	7
		Computer Engineering	1+1+3+2	7
		Software Engineering	1+1+1+3	6
		Information System	1+1+1+2	5
		Information Technology	1+1+1+2	5

Fig. 19. Weight Calculation for the fifth question

Exact Result/	maximum weight calculated
Computer Engineering	10

Fig. 20. Maximum weighted answer

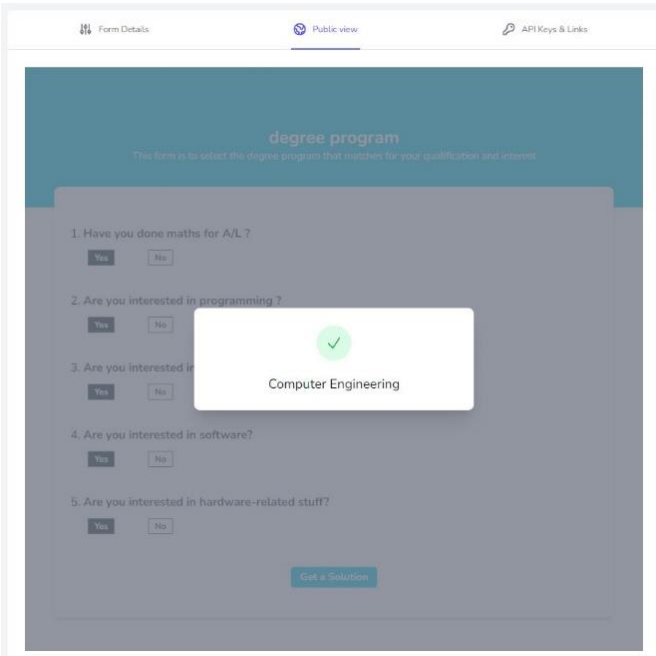


Fig. 21. Maximum weighted Result that displays in the interface

4. Testing and Evaluation

Here, the techniques utilized for testing the system before, during, and after implementation will be considered. The three major testing methods used are Unit testing, integration testing, and system testing.

Unit Testing

Unit testing includes features that are critical to the performance of the unit under test. Thus, the developers can make changes to source code without being concerned about how those changes would affect how other components or the program as a whole function. So, Unit testing for the "Interllib Forms" was done for each test scenario using a set of test cases (Table 3).

Table 3. Unit testing done for each test cases

	Test Case	Results
Unit Testing	Login and Authentication	• User credentials will be authenticated and provide access. Unless an error message will display.
	User Registration	• The system will allow registration if all the required details are entered in the correct format. • Email verification is done when registering to the system.
	Form Creation	• A new form without a form name and description is not allowed to move to the next step. • In the "Add Question" phase user needs to add at least one question to move to the next step. • In the "Adding Results" phase user need to add at least one answer to move to the next step.

Integration Testing

The larger components of the "Interllib Forms" are examined using integration testing after all the units have been confirmed to be running in the most effective and error-free manner possible (Table 4).

Table 4. Integration testing done for each test cases

	Test Case	Results
Integration Testing	User interface testing	The linking of the system was flawless. The system's buttons and performance were all up to standard.
	User scenarios testing	It ensures that all possible outcomes are functioning as expected.
	Data flow testing	Data flow was error-free and functioning properly.

System Testing and Acceptance testing

During system testing, the overall behavior of the system is evaluated. The system is tested under the scope of the project or product. It comprises tests based on risk and requirement definitions, as well as business processes, high-level descriptions, or Use cases of the system interactions with system resources as well as operating systems. This testing strategy is performed by specialists or independent testers. So, both functional, as well as non-functional requirements, should be taken into consideration. This System testing can be introduced as the last test to ensure that the system is up to the level that it meets the specified requirements and serves them. So, in this project, the system testing was carried out by evaluating the end users' feedback regarding both functional and non-function requirements. The end users were the applicants, Software Engineers, Web developers, and Knowledge experts in different domains.

Evaluating both functional and non-functional requirements from end-user perspectives

As mentioned previously, the system testing strategy is usually carried out to evaluate the functional and non-functional requirements of the end product of the project and whether the end product is acceptable or not. So, an evaluation was done by participating actual end-users; Applicants, Software Engineers /web developers, and Knowledge experts. The evaluation was carried out by distributing four different Google forms with relevant questions.

Applicants

The evaluation was done by participating 20 participants. First, the applicants were given a general form to register for a certain degree program. This form did not contain the user guidance or explanation functionality provided by the "Interllib Forms ". Next, the applicants are instructed to fill out a new form containing the same questions, but the user guidance or explanation functionality is integrated. Also, the applicants are instructed to use the specific functionality by clicking the small button close to the form control in an uncertain situation. Further, a brief description was given of "Uncertainties that anyone can face when filling application forms in real life ".

Next, the applicants were given a google form -1 to evaluate the feedback and experience they gained by filling out both forms.

From the responses given by the applicants, It is concluded that 80% of the applicants have found it easy to fill out the second form which is integrated by the iframe view.

The results gained regarding the uncertainty faced by users in both forms are shown in Fig. 23, and Fig. 24.

So, from the above analysis, after integrating this user guidance functionality, the uncertainty experienced by the applicants has been reduced to 0%.

Next, the guidance provided in uncertain situations is evaluated. The results show that 80% applicants (who filled the form -2) are very satisfied with the guidance they gained.

The final question was to evaluate the "Acceptability "of the user-guidance functionality which is the product of "Interllib forms" -a web application.

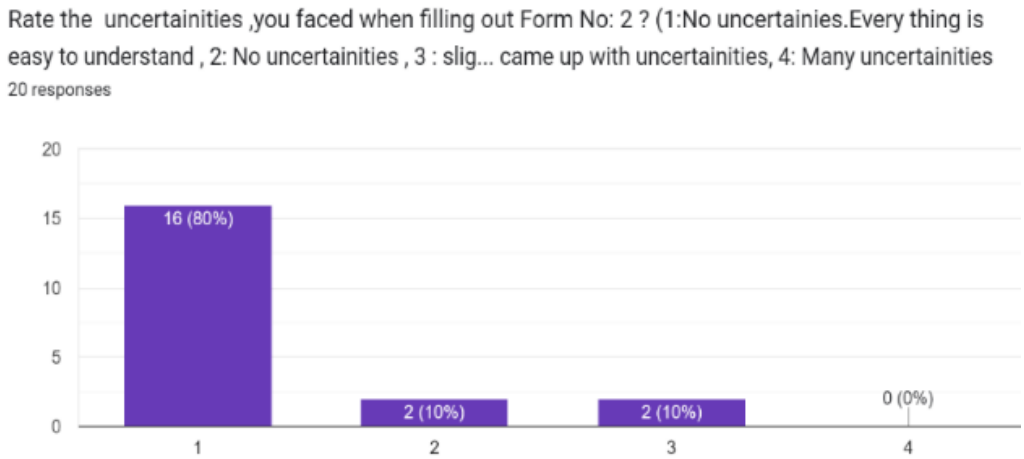


Fig. 23. Uncertainties faced with the "user guidance functionality"

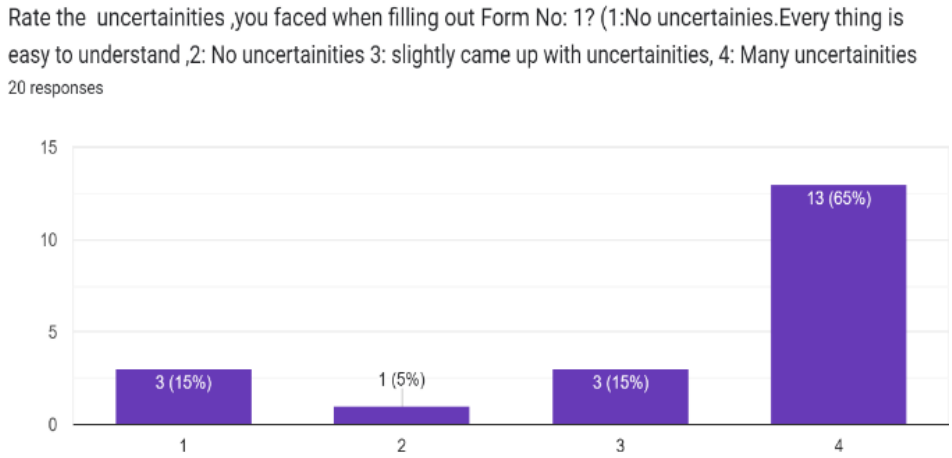


Fig. 24. Uncertainties faced with the "user guidance functionality"

So, when comparing the results obtained, 75% of the applicants have given positive feedback on the "Acceptability "of the end product of "Interllib forms" (Fig. 25).

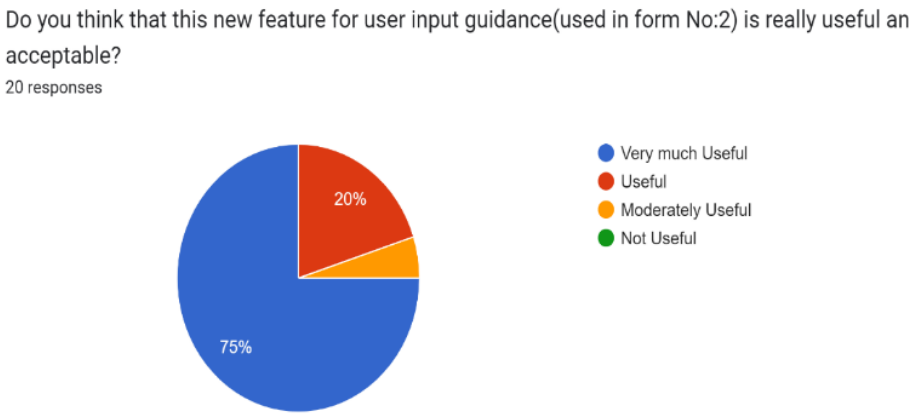


Fig. 25. Acceptability level (Applicants' perspective)

So, From the applicants' perspective, it is revealed that the user-guidance functionality, which is the final product of "Interllib forms" is at an acceptable level.

Knowledge Experts – In any domain

The knowledge experts are the end users who directly interact with the "Interllib Forms" web application. So, an evaluation was done by distributing a google form among 15 participants who are experts in different domains. Out of them, 53.3% are from an IT background. 46.7% are from the health, education, business, and engineering sectors. The feedback on both functional and non-functional requirements of "Interllib-Forms" are evaluated in Fig. 26.

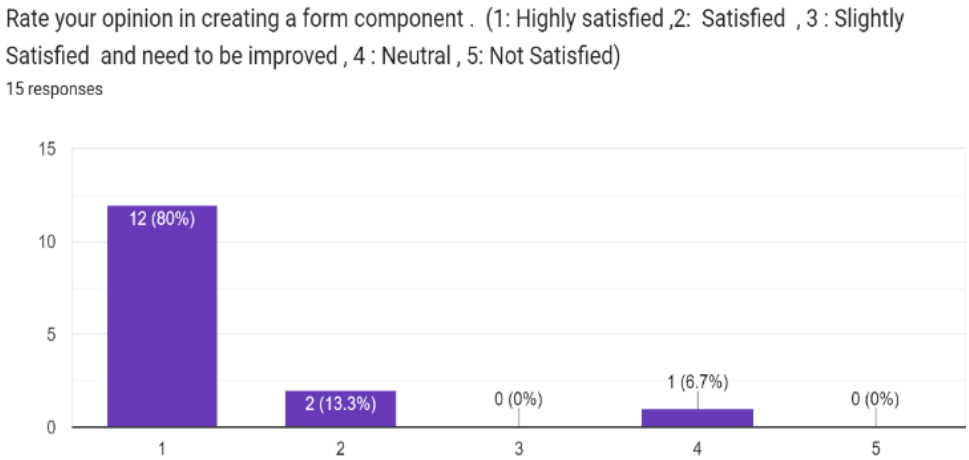


Fig. 26. Responses to the second question

The majority of the knowledge experts (80%) show the highest satisfaction regarding the steps followed in the form component in generating the iframe view.

Next, the user experience level regarding the User interfaces design for "Questions and Results" mapping is evaluated. The results depict that,66.7% of knowledge experts are highly satisfied. Collectively (66.7% + 20%) of them have positive feedback on interface design while 6.7% have neutral feedback. But none of them are not satisfied with the user interface design. So, from the majority of the positive responses (66.7% + 20% = 86.7%), is concluded that the User Interface design of the web application is up to standard level.

"Interllib forms" use a specific logical and mathematical calculation to map both questions and answers. So, the feedback of the knowledge experts is evaluated regarding the calculation used.

From the responses,50% of the knowledge experts say that the logic used in mapping is at a satisfactory level. And 28.6% of the remaining say that the logic is slightly satisfied and needs to be improved. This percentage is higher in value when compared to the percentage of "Highly satisfied". Since the majority of the knowledge experts (21.4% +50% = 71.4%) have positive feedback, It can be concluded that the logic of "Questions and Answer mapping is at an acceptable level.

Next, the evaluation was done to analyze the level of the understandability of the mapping logic. The question and responses are given in Fig. 27.

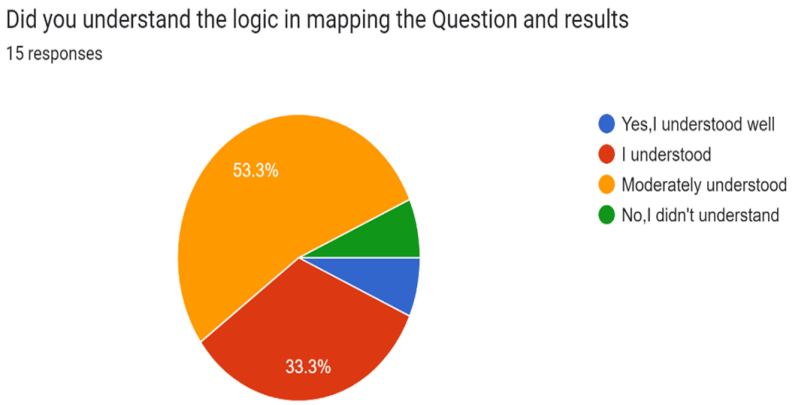


Fig. 27. Level of understanding the logic behind question-and-answer mapping

From the responses obtained, It is revealed that the majority (53.3%) of the knowledge experts have a moderate understanding of logic applied in "Question and Answer "mapping. Also, 6.7% of the rest have not understood the specific logic behind "Question and Answer Mapping".

Software Engineers and Web developers

The Software Engineers or the web developers are the end users who use the specific code snippet generated from the "Interllib Forms "platform. Thus, the evaluation was done by participating 15 software engineers and web developers who work in reputed software companies in Sri Lanka. Out of the participants, 80% of them have used this specific code snippet in various form-building environments that used frameworks like, React, Angular, etc.

Since they were able to use the code snippet in different form building environments, the scalability and flexibility functionality were evaluated. The question and responses are given in Fig. 28.

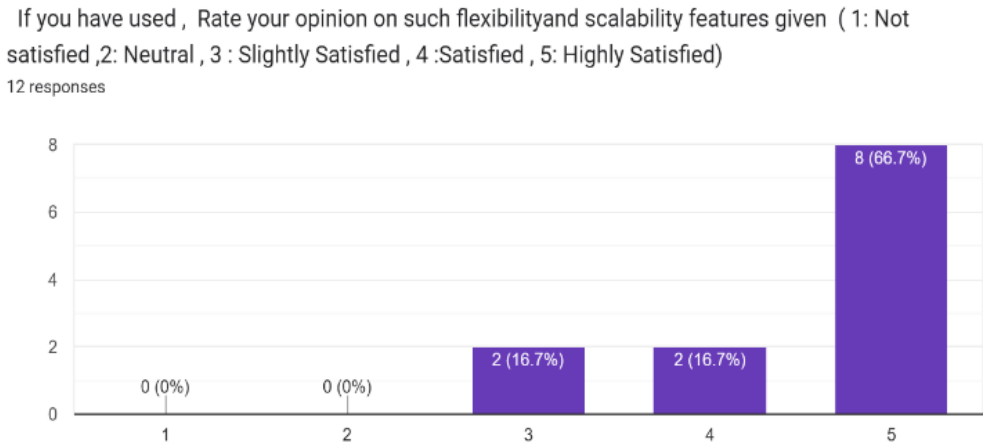


Fig. 28. Evaluation of scalability and flexibility functionality

From the responses, it is concluded that the majority (66.7% + 16.7% = 83.4%) of the responses have positive feedback regarding the non-functional requirements, Scalability, and flexibility.

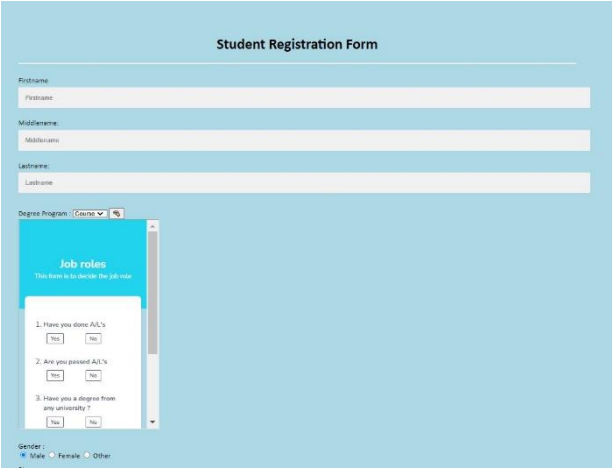
Also, out of the software engineers and developers, only 7.7% have faced issues in using the code snippet and displaying the iframe view in their forms. The issues were "consuming 10-20 seconds to display the iframe view ". But the particular issue seems to be inconsiderable since 92.3% have not met any issues mentioned in the question. Also, the severity of the issue seems to be low as such issues may depend on the strength of the internet connection.

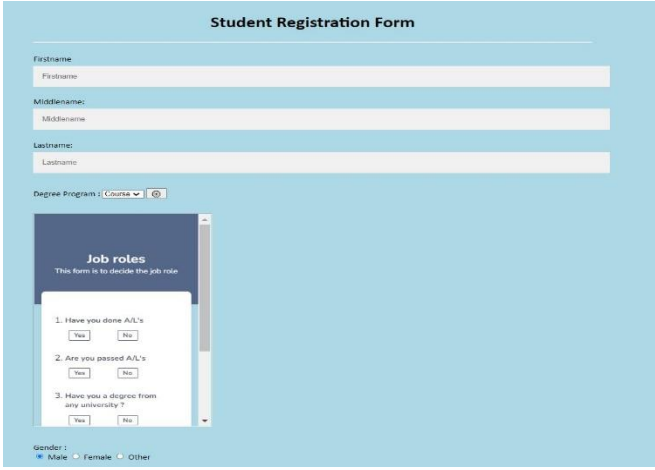
Finally, the feedback on customizability was evaluated. The majority (60% + 33.3% = 93.3%) of the responses have a positive impression of the customizability functionality (Figure 89, Figure 90).

Finally, it is concluded that the majority(70.8%) of the applicants, knowledge experts and software engineers who participated in the evaluation have a positive approach regarding both functional and non-functional requirements provided by the "Interllib forms" web application. Also, a few recommendations and suggestions were given to be improved this system.

Evaluating the customizability feature of the system

The evaluation of the scalability of the system was done by embedding the unique key or the code snippet into various application development environments. So, it is platform independent. Even we can embed this unique key inside a div





element or a button. So, it was able to render and display a fully functional iframe view of the form components. Even, we were able to embed and view the iframe view in a local environment. Also, the theme color of the iframe view can be customized (Fig. 29, Fig. 30).

Security Measures

When we come to the security concerns related to the system, several measures were taken for web security vulnerabilities like SQL injections, and cross-site scripting. “Interllib forms” is implemented using the Laravel framework. Thus, Laravel offers a simple way to safeguard the “Interllib Forms” application against cross-site request forgeries. Since the system does not handle sensitive data like personal information, encryption measures were not taken up to the current level (Because every data that is taken as input are being exposed to the public). And, the system is not in the production environment yet. Thus, to check and test the functionality of the iframe, a few of the associate security features were disabled temporarily. But, when it comes to the production environment, we propose to limit the iframe access per domains of the client’s websites.

5. Conclusion and Further works

While mar Fig. 29. With color 1 ightforward, some are more chal Fig. 30. With color 2 terpret and fill. Thus, uncertainties due to not understanding the questions that the application form contains. Thus, “Interllib Forms”, An expert system-based solution will reduce the uncertainties faced when filling out a form. From the evaluation done targeting the end users of the “Interllib Forms”, it was concluded that 70.8% of end users have positive feedback regarding the acceptable level. And 53.3% of knowledge experts have moderate feedback regarding the Logic used in Question and Results mapping Thus, providing a user manual and developing the logic that we used in “Question & Answer mapping: and “retrieving the exact solution” will be focused on future works. Further, the system can be improved, so that any applicant can get guidance using a chatbot to fill out the application form.

References

Aljawarneh, S., Alkhateeb, F. and Al Maghayreh, E. (2010) ‘A semantic data validation service for web applications’, Journal of Theoretical and Applied Electronic Commerce Research, 5(1), pp. 39–55. doi: 10.4067/S0718-18762010000100005.

Alkhalaf, M. et al. (2012) ‘ViewPoints: differential string analysis for discovering client- and server-side input validation inconsistencies’, p. 56. doi: 10.1145/2338965.2336760.

Alkhalaf, M., Aydin, A. and Bultan, T. (2014) ‘Semantic differential repair for input validation and sanitization’, 2014

International Symposium on Software Testing and Analysis, ISSTA 2014 - Proceedings, pp. 225–236. doi: 10.1145/2610384.2610401.

Alkhalaf, M., Bultan, T. and Gallegos, J. L. (2012) 'Verifying client-side input validation functions using string analysis', Proceedings - International Conference on Software Engineering, pp. 947–957. doi: 10.1109/ICSE.2012.6227124.

Ashcraft, K. and Engler, D. (2002) 'Using programmer-written compiler extensions to catch security holes', Proceedings - IEEE Symposium on Security and Privacy, 2002-Janua, pp. 143–159. doi: 10.1109/SECPRI.2002.1004368.

Balzarotti, D. et al. (2008) 'Saner: Composing static and dynamic analysis to validate sanitization in web applications', Proceedings - IEEE Symposium on Security and Privacy, pp. 387–401. doi: 10.1109/SP.2008.22.

Balzarotti, D. et al. (no date) 'Multi-Module Vulnerability Analysis of Web-based Applications', pp. 25–35.

Bohøj, M., Bouvin, N. O. and Gammelmark, H. (2011) 'AdapForms: A framework for creating and validating adaptive forms', Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6757 LNCS, pp. 105–120. doi: 10.1007/978-3-642-22233-7_8.

Hassinen, M. and Mussalo, P. (2005) 'Client controlled security for web applications', Proceedings - Conference on Local Computer Networks, LCN, 2005, pp. 810–816. doi: 10.1109/LCN.2005.38.

Honkala, M. and Vuorimaa, P. (2005) 'Secure Web forms with client-side signatures', Lecture Notes in Computer Science, 3579(December 2005), pp. 340–351. doi: 10.1007/11531371_46.

Kirda, E. et al. (2006) 'Noxes: A client-side solution for mitigating cross-site scripting attacks', Proceedings of the ACM Symposium on Applied Computing, 1, pp. 330–337.

Levis, M., Helfert, M. and Brady, M. (2008) 'Form input

validation: An empirical study on irish corporate websites', ICEIS 2008 - Proceedings of the 10th International Conference on Enterprise Information Systems, HCI, pp. 148–153. doi: 10.5220/0001681601480153.

Li, N. et al. (2010) 'Perturbation-based user-input-validation testing of web applications', Journal of Systems and Software, 83(11), pp. 2263–2274. doi: 10.1016/j.jss.2010.07.007.

Livshits, V. B. and Lam, M. S. (2005) 'Finding Security Errors in Java Programs with Static Analysis', Proc. Usenix Security Symposium, pp. 271–286.

Minamide, Y. (2005) 'Static approximation of dynamically generated Web pages', p. 432. doi: 10.1145/1060745.1060809.

Offutt, J. and Xu, W. (2004) 'Generating test cases for web services using data perturbation', ACM SIGSOFT Software Engineering Notes, 29(5), pp. 1–10. doi: 10.1145/1022494.1022529.

Pietraszek, T. and Berghe, C. Vanden (2006) '185. Context-Sensitive String Evaluation', pp. 124–145.

Saxena, P. et al. (2010) 'A symbolic execution framework for JavaScript', Proceedings - IEEE Symposium on Security and Privacy, pp. 513–528. doi: 10.1109/SP.2010.38.

Saxena, P. and Hanna, S. (2010) 'FLAX : Systematic Discovery of Client-side Validation Vulnerabilities in Rich Web Applications', 17th Annual Network Distributed System Security SymposiumNDSS, p. 17.

Scott, D. and Sharp, R. (2003) 'Specifying and enforcing application-level web security policies', IEEE Transactions on Knowledge and Data Engineering, 15(4), pp. 771–783. doi: 10.1109/TKDE.2003.1208998.

Taneja, K. et al. (2010) 'MiTV: Multiple-implementation testing of user-input validators for web applications', ASE'10 - Proceedings of the IEEE/ACM International Conference on Automated Software Engineering, pp. 131–134. doi: 10.1145/1858996.1859019.

Vogt, P. et al. (2007) 'Cross-Site Scripting Prevention with Dynamic Data Tainting and Static Analysis', Work, 42(13), pp. 4188–4190. doi: 10.1021/cm801305f.

Wassermann, G. and Su, Z. (2008) 'Static detection of cross-site scripting vulnerabilities', Proceedings - International Conference on Software Engineering, pp. 171–180. doi: 10.1145/1368088.1368112.

Web, O. (2002) 'ON-THE-FLY WEB CONTENT INTEGRITY', 45(11), pp. 33–37.

Xie, Y. and Aiken, A. (2006) 'Static detection of security vulnerabilities in scripting languages', 15th USENIX Security Symposium, pp. 179–192.

Zaremba, L. S. and Smoleński, W. H. (2000) 'PowerForms: Declarative Client-Side Form Field Validation', Annals of Operations Research, 97(1–4), pp. 131–141. doi: 10.1023/A.

Clemente, D. (2022) Is digitalocean good for website hosting?, WebsiteBuilderInsider.com. Available at: https://www.websitebuilderinsider.com/is-digitalocean-good-for-website-hosting/ (Accessed: December 14, 2022).

Levanduski, M. (2022) Digitalocean hosting: Dev-friendly but do they get the basics right?, Digital.com. Digital.com. Available at: https://digital.com/best-web-hosting/digitalocean/ (Accessed: December 14, 2022).

Smith, B. (2022) Digitalocean Review, WebsiteSetup. Available at: https://websitesetup.org/hosting-reviews/digitalocean/ (Accessed: December 14, 2022).

Managed cloud hosting platform (no date) DigitalOcean. Available at: https://www.digitalocean.com/solutions/managed-cloud-hosting (Accessed: December 14, 2022).

How is digitalocean for hosting a high-traffic website? (no date) Quora. Available at: https://www.quora.com/How-is-DigitalOcean-for-hosting-a-high-traffic-website (Accessed: December 14, 2022).

Taylor, D. (2022) Digitalocean vs AWS – difference between them, Guru99. Available at: https://www.guru99.com/digitalocean-vs-aws.html (Accessed: December 14, 2022).

The PHP framework for web artisans (no date) Laravel. Available at: https://laravel.com/docs/9.x#why-laravel (Accessed: December 14, 2022).

The PHP framework for web artisans (no date) Laravel. Available at: https://laravel.com/docs/4.2/security (Accessed: December 14, 2022).