JIMMA INSTITUTE OF TECHNOLOGY

Faculty of Computing and Informatics

Department of Software Engineering

CBTP-PHASE IV: - Intervention and Evaluation

On

Web-based Infant Immunization System

Submission date: January 26,2024 G.C
Jimma, Ethiopia

Group members

	NAME	ID
1.	LALISA BULA	RU2359/13
2.	EYUEL DEBREWORK	RU0210/13
3.	HERMELA HAILEGIORGIS	RU2124/13
4.	ABDI MUDESIR	RU0019/13
5.	TSION BEYENE	RU2909/13
6.	KEDIR SEID	RU0602/13
7.	NAOL FISSEHA	RU1931/13
8.	SITRA JEMAL	RU0013/13
9.	HUMED ABAHABA	RU0358/13
10.	ELDANA KIBRU	RU3003/13

Advisors **Sign**

Mr. Alemisa _____

Ms. Meron _____

Acknowledgment

First, we would like to thank JIT Campus especially the Community Based Training Program (CBTP) coordinators and Faculty of Computing and Informatics for preparing a chance for us to do this CBTP project which allows us to address the community problem and solve the problem by applying the knowledge we have in a practical way.

Next, our special thanks goes to our project advisors Mr.Alemisa and Ms. Meron for their valuable support, great deal of advice, encouragement and offering us their support on our project. And we would like to thank all the group members who participated is this challenging project and put all their effort to do the best of us.

The last but not the least we would like to thank our honorable teachers who give as their time to support us.

Table of Content

List of figures

List of table

1. Introduction

This documentation is about testing our web based infant immunization system. This document provides a comprehensive overview of our testing process for both the graphical user interface (GUI) and functionality of the software. Our team has developed a detailed checklist that covers all aspects of the testing process, ensuring that the software performs optimally and meets all requirements.

As software becomes more critical to daily operations, it is essential to thoroughly test it before release. Effective testing helps to identify any potential issues, ensuring that the software works as intended, is free from defects, and meets the needs of its users.

In this documentation, you will find a step-by-step guide to our testing process, including the testing environment, test cases, and methodology. Our team follows a structured approach to testing, using a checklist to ensure that we cover all aspects of the software. We start by testing the GUI to ensure that it is user-friendly and easy to navigate, then move on to functionality testing, where we test each feature and function of the software.

By following our comprehensive testing checklist, we can identify and address any potential issues, providing our users with a seamless and satisfying experience.

We use bootstrap which is a popular php web framework that allows developers to build complex and scalable web applications quickly. Once the development of a bootstrap web app is complete, it needs to be deployed on a server to make it accessible to users. There are several deployment methods available for bootstrap-based web apps, including traditional server deployment, cloud-based deployment, and container-based deployment. The deployment process involves configuring the server environment, installing dependencies, setting up the database, and deploying the code. The choice of deployment method depends on the specific needs of the application and the resources available for deployment.

1.2. PURPOSE

The purpose of this document is to provide a detailed outline of our testing process for both the graphical user interface (GUI) and functionality of the software. And perform testing on our web based infant immunization system.

1.3. SCOPE AND LIMITATION

The scope of the software testing document is focused solely on the graphical user interface (GUI) and functionality of the software. The document outlines the testing process, test cases, and methodology used to thoroughly test these areas of the software. The testing document does not cover other aspects of the software, such as performance testing or security testing. By focusing on the GUI and functionality, the testing document aims to ensure that the software is user-friendly, easy to navigate, and functions as intended, providing a seamless and satisfying experience for its users.

1.4. METHODOLOGY

The methodology used for testing the GUI and functionality of the database in the software testing document involves using SQL Developer. SQL Developer is an integrated development environment that provides tools for database development and management.

To test the GUI, the testing team will use manual testing methods to navigate through the software, identifying any issues with the GUI such as poor design, layout, or navigation. The team will use a checklist to ensure all GUI elements are thoroughly tested.

For functionality testing of the database, SQL Developer will be used to perform database checkpoint testing. This will involve testing the database for default check, custom check, and runtime record check. The testing team will create test cases to ensure that the database is performing optimally and meets all requirements.

The testing team will also use SQL Developer to create automated tests for the database functionality. This will involve writing SQL queries and scripts to test the database and ensure that it is functioning as intended.

the methodology for testing the GUI and functionality of the database in the software testing document involves a combination of manual and automated testing using SQL Developer. This approach ensures that the software is thoroughly tested and meets all requirements

Tools Used In The Implementation Phase

Software Tools Used	Hardware tool used
Php for back end	Server for data storage
Html for front end	Computer or cellphone for accessing the web
Css for front end	Network for connection
Javascript for responsiveness	

1.5. OBJECTIVE OF STUDY

1.5.1. General Objective

The main objective of this project is to test functionality and operational of our web-based infant immunization system.

1.5.2. Specific objective

These are the specific objective identified in order to implement the general objectives

- ✓ Prepare test cases for GUI checkpoint for a single property
- ✓ Prepare test cases for GUI checkpoint for single object/window
- ✓ Prepare test cases for GUI checkpoint for multiple object
- ✓ Prepare test cases for Database checkpoint for Default check
- ✓ Prepare test cases for Database checkpoint for custom check
- ✓ Prepare test cases for Database checkpoint for runtime record check

2. TESTING

2.1. Operational Testing

2.2. GUI checkpoint for a single property

In this testing we test object that have no event handling on them

a. Login page

Fig 2.1 login page

Table 2.1 Username field single pint test case

Test Cases for Username field	Yes	No	Remark
Form components			
Is the object enabled?	Y		
Is the object editable?	Y		
Does the field show warning for incorrect user input?	Y		
Are default values provided in fields, where appropriate?		N	
Is the cursor changed when the user tries in appropriate operation on the username field	Y		
Does the field give warning for the user if it reaches maximum input trial	Y		
Does the field have maximum amount of input characters		N	
Does the field support lower- and upper-case differences	Y		
Does it give hint?	Y		

Table 2.2Password field single pint testcase

Test Cases for Password filed	Yeas	No	Remark
Form components			
Is the object enabled?	**Y**		
Is the object editable?	**Y**		
Is the object encrypted?	**Y**		
Does the field show warning for incorrect user input?	**Y**		
Are default values provided in fields, where appropriate?		**N**	
Is the cursor changed when the user tries in appropriate operation on the username field?	**Y**		
Does the field give warning for the user if it reaches maximum input trial?	**Y**		
Does the field have maximum amount of input characters?		**N**	
Does the field support lower- and upper-case differences?	**Y**		
Does it give hint?	**Y**		

Table 2.3 Remark single pint test case

Test Cases for Remark	**Yes**	**No**	**Remark**
Form components (including form title, form field, form text font)			
Is the element enabled?	Y		
Are default values provided in fields, where appropriate?		N	

b. Set new Password Page

Fig 2.3 Set new Password page

Table 2.5 New Password Field Single Point Check

Test Cases for new password	Yeas	No	Remark
Form components			
Is the object enabled?	Y		
Is the object editable?	Y		
Is the object encrypted?	Y		
Does the field show warning for incorrect user input?	Y		
Are default values provided in fields, where appropriate?		N	
Is the cursor changed when the user tries in appropriate operation on the username field	Y		
Does the field give warning for the user if it reaches maximum input trial	Y		
Does the field have maximum amount of input characters		N	
Does the field support lower- and upper-case differences	Y		
Does it give hint?	Y		

Table 2.6 Confirm Password Field Single Point Check

Test Cases for confirm password	Yes	No	Remark
Form components			
Is the object enabled?	Y		
Is the object editable?	Y		
Is the object encrypted?	Y		
Does the field show warning for incorrect user input?	Y		
Are default values provided in fields, where appropriate?		N	
Is the cursor changed when the user tries in appropriate operation on the username field	Y		
Does the field give warning for the user if it reaches maximum input trial	Y		
Does the field have maximum amount of input characters		N	
Does the field support lower- and upper-case differences	Y		
Does it give hint?	Y		

C. Profile page

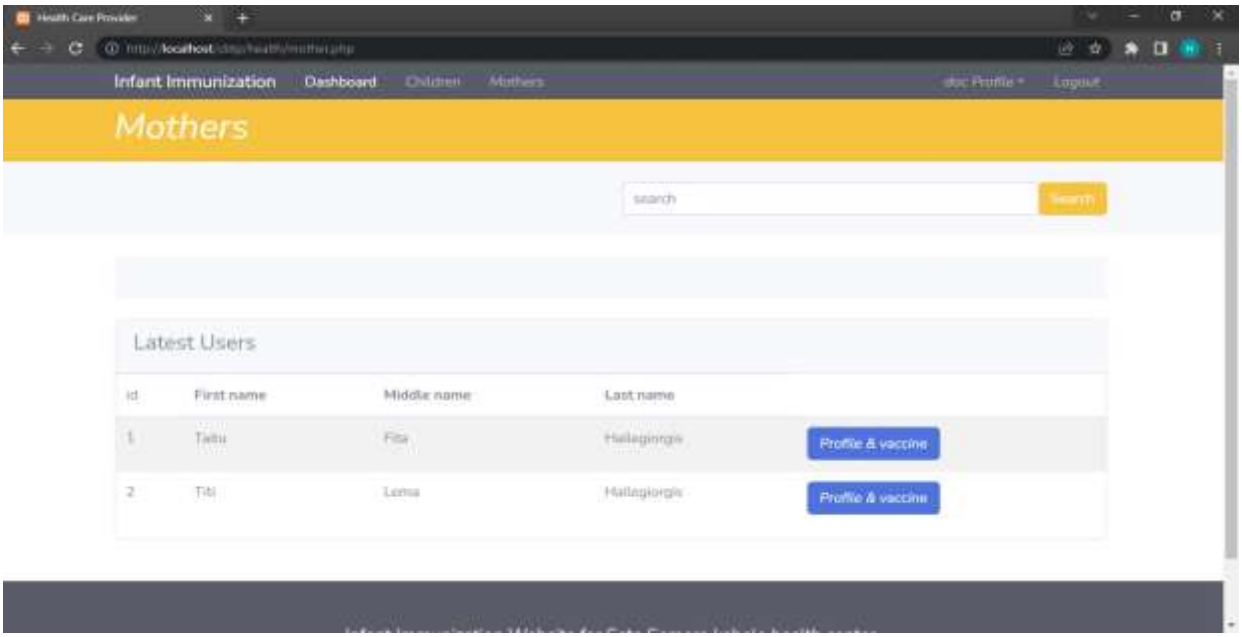

Fig 2.4 Profile page

Table 2.7 Drop down box single point check

Test Cases for add profile selection drop down	Yes	No	Remark
Is the element enabled?	Y		
Does the drop-down button work properly?	Y		
Does the button is visible for any user new for profile page	Y		
Does the specified operations list properly on the list		N	
Does the drop-down box cause disorder in the layout of the page when clicked?		N	
Does the spelling for the button text is correct or not.	Y		

D. Admin Dashboard

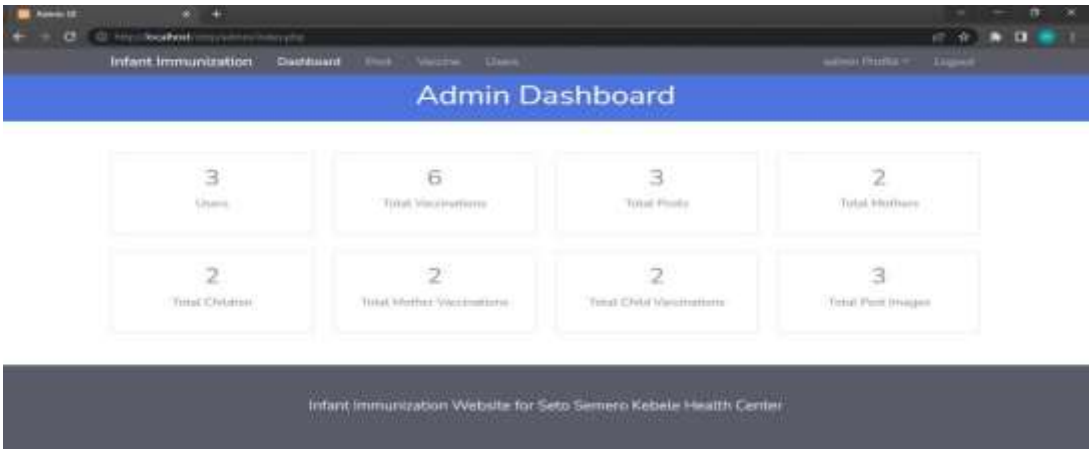

Fig 2.5 Admin Dashboard single point check

Table 2. 8 Menu single point check

Test Cases for Menu bar	Yes	No	Remark
Form components			
Is the menu bar being visible and clearly distinguishable from the rest of the interface?	Y		
Is each menu item being clickable and leads to the correct page or submenu?	Y		
Is the menu bar adjusting to different screen sizes and resolutions, and that all menu items remain accessible?	Y		
Is the menu bar can be accessed and used with assistive technologies such as screen readers and keyboard navigation?		N	
Are the menu items are logically organized and easily understandable to users?	Y		
Is the menu bar including language options, test that it accurately translates menu items and provides a consistent experience across languages?		N	

Table 2. 9 Authentication and Authoritarian Table single point check

Test Cases for Authentication and Authoritarian Table	Yes	No	Remark
Form components			
the table is displaying the correct number of rows and columns specified	Y		
Are the column headers are displayed in the correct order and contain the expected text?	Y		
Is the table allowing sorting of columns in ascending or descending order	Y		
Is the table supports searching or filtering based on one or more criteria		N	
Is the table correctly displays data with different data types, such as text, numbers, and dates?		N	
Does the table have a button or link that allows the admin to add new users and that this function works correctly	Y		
Is the table correctly handles user interaction, such as selecting rows or cells, and copying and pasting data?	Y		
Does table is responsive and adjusts its size and layout appropriately based on the device screen size?	Y		
Does the table support accessibility features such as keyboard navigation and screen readers for users with disabilities?	Y		
Does the table have a button or link that allows the admin to add new users and that this function works correctly	Y		
Does the table allow the admin to edit object information such as name, date of creation etc.…	Y		

Test Cases for Authentication and Authoritarian Table	Yeas	No	Remark
Form components			
the table is displaying the correct number of rows and columns specified	Y		
Are the column headers are displayed in the correct order and contain the expected text?	Y		
Is the table allowing sorting of columns in ascending or descending order	Y		
Are the table supports searching or filtering based on one or more criteria		N	
Is the table correctly displays data with different data types, such as text, numbers, and dates?		N	

Is the table correctly displays data with different formatting, such as currency or percentages?		N	
Is the table correctly handles user interaction, such as selecting rows or cells, and copying and pasting data?	Y		
Does table is responsive and adjusts its size and layout appropriately based on the device screen size?	Y		
Does the table support accessibility features such as keyboard navigation and screen readers for users with disabilities?	Y		

Table 2.10 Authentication and Authorization

E. Health Professional Dashboard

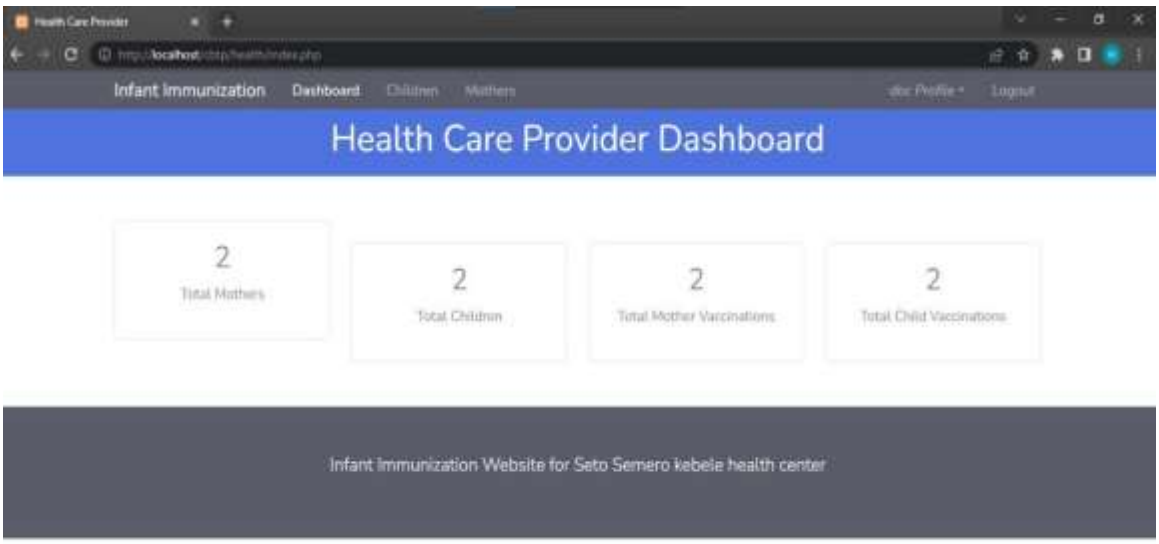

Fig 2.6 Health Professional Dashbord

F. Registrar Dashboard

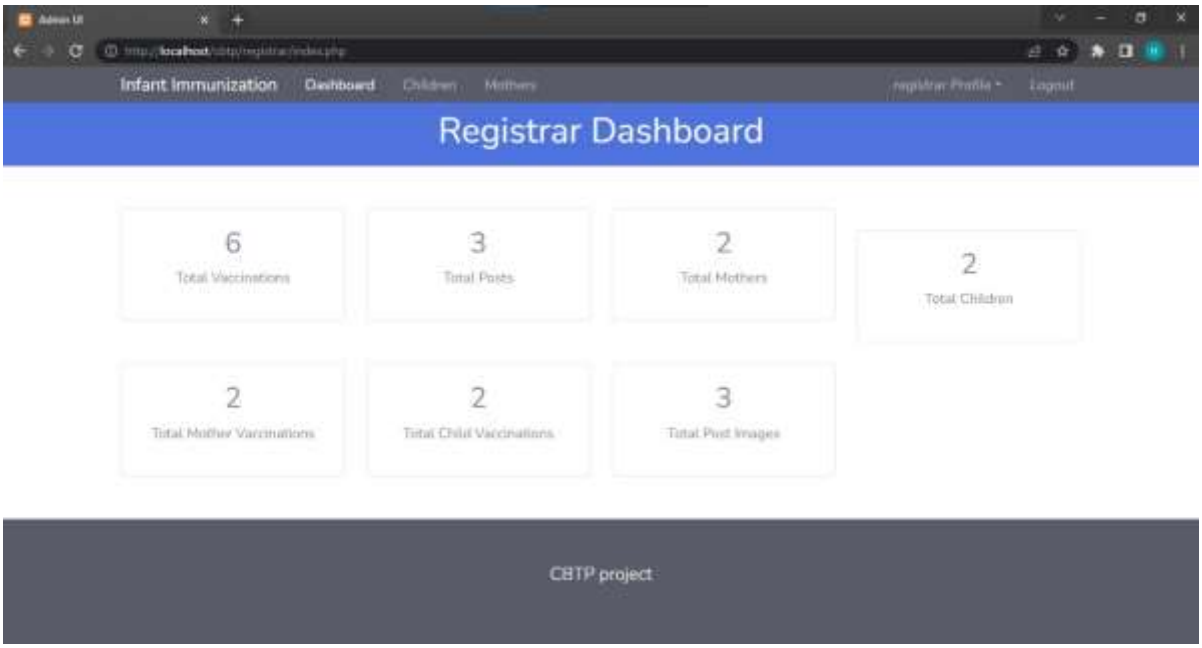

Fig 2.7 Registrar dashboard

G. Vaccine Dashboard

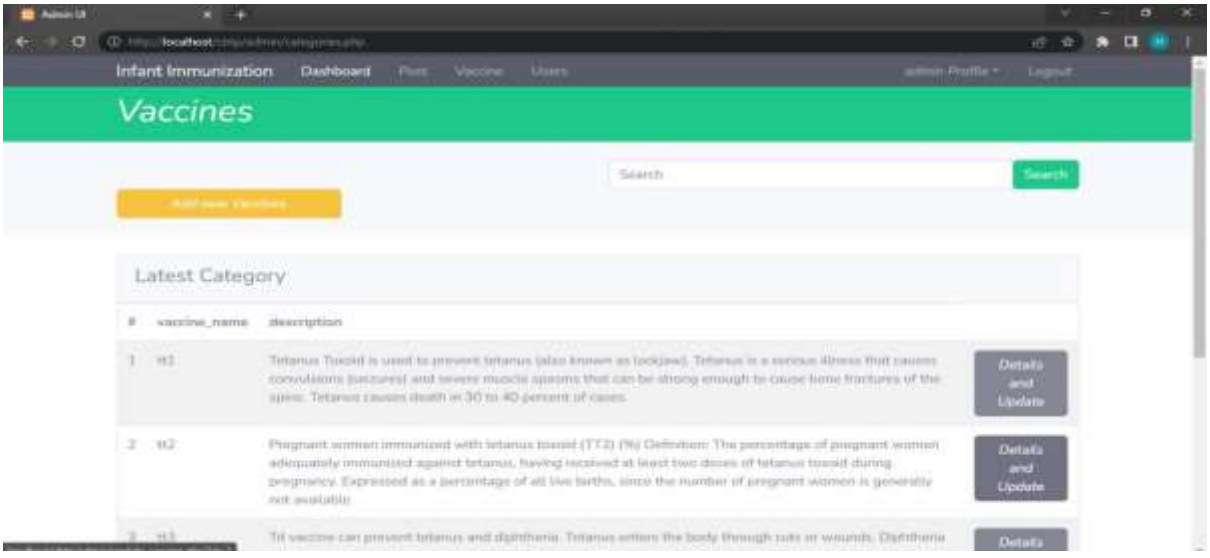

Fig 2.8 Vaccine dashboard

Table 2.11 vaccine list Table single point check

Test Cases for vaccine list Table	Yeas	No	Remark
Form components			
the table is displaying the correct number of rows and columns specified	Y		
Are the column headers are displayed in the correct order and contain the expected text?	Y		
Is the table allowing sorting of columns in ascending or descending order	Y		
Is the table supports searching or filtering based on one or more criteria		N	
Is the table correctly displays data with different data types, such as text, numbers, and dates?		N	
Does the table has a button or link that allows the admin to add new vaccine types	Y		
Is the table correctly handles user interaction, such as selecting rows or cells, and copying and pasting data?	Y		
Does table is responsive and adjusts its size and layout appropriately based on the device screen size?	Y		
Does the table support accessibility features such as keyboard navigation and screen readers for users with disabilities?	Y		

Table 2.12 vaccine list Table single point check

Test Cases for vaccine search Bar	Yes	No	Remark
Is the object enabled?	Y		
Is the object editable?	Y		
Is the object visible to users on first sight experience	Y		
Does the field show warning for incorrect user input?	Y		
Does the search field display frequently asked search histories while hovering on it?	Y		
Does it have auto-spell checker if the user can’t know the full vaccine name	Y		
Does the field give warning for the user if it reaches maximum input trial	Y		
Does the field have maximum amount of input characters		N	

Does the field support lower- and upper-case differences	Y		
Does it give hint?	Y		

H. User list

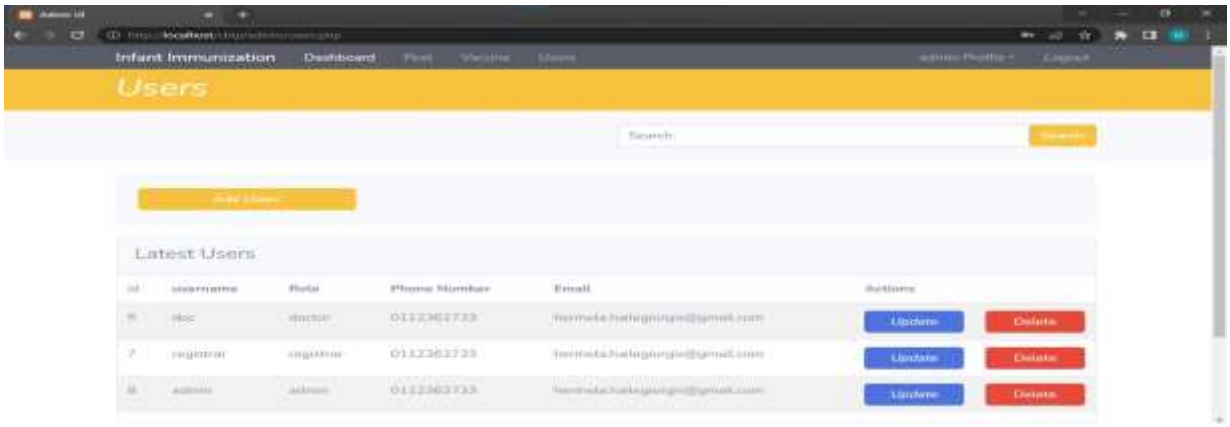

Fig 2.9 users list page

Table 2.13 vaccine list Table single point check

Test Cases for user list Table	Yes	No	Remark
Form components			
the table is displaying the correct number of rows and columns specified	**Y**		
Is the column headers are displayed in the correct order and contain the expected text?	**Y**		
Is the table allowing sorting of columns in ascending or descending order	**Y**		
Is the table supports searching or filtering based on one or more criteria		**N**	
Is the table correctly displays data with different data types, such as text, numbers, and dates?		**N**	
Does the table has a selector to select whole or single row data	**Y**		
Is the table correctly handles user interaction, such as selecting rows or cells, and copying and pasting data?	**Y**		
Does table is responsive and adjusts its size and layout appropriately based on the device screen size?	**Y**		

Does the table support accessibility features such as keyboard navigation and screen readers for users with disabilities?	**Y**		
Does the table allows the admin to edit object information such as name, date of creation etc.…	Y		

Table 2.14 list search bar single point check

Test Cases for user list search Bar	**Yeas**	**No**	**Remark**
Search bar components			
Is the object enabled?	**Y**		
Is the object editable?	**Y**		
Is the object visible to users on first sight experience	**Y**		
Does the field show warning for incorrect user input?	**Y**		
Does the search field displays frequently asked user name	**Y**		
Does it have auto-spell checker if the user can't know the full vaccine name	**Y**		
Does the field give warning for the user if it reaches maximum input trial	**Y**		
Does the field have maximum amount of input characters		**N**	
Does the field support lower- and upper-case differences	**Y**		
Does it give hint?	**Y**		

Table 2.15 Add user button list search bar single point check

Test Cases for Add user button	Yes	No	Remark
Is the element enabled?	**Y**		
Does the add new user when it is clicked	**Y**		
Does the button is visible for any user new for user list page	**Y**		
Does the button is enabled/clickable or not by clicking on the button.		**N**	
Does values added by the user on click on the Add button saved correctly in the database.	**Y**		
Does the spelling for the button text is correct or not.	**Y**		

Table 2.16 go to user button list search bar single point check

Test Cases for add profile Go button	**Yeas**	**No**	**Remark**
Is the element enabled?	**Y**		
Does the add profile operation work when it is clicked	**Y**		
Does the button is visible for any user new for profile page	**Y**		
Does the button is enabled/clickable or not by clicking on the button.		**N**	
Does values deleted by the user on click on the Go button deleted from the table correctly database.	**Y**		
Does the spelling for the button text is correct or not.	**Y**		

2.2.1. GUI checkpoint for a single object/window

In this testing we test objects that we handle events

Table 2.17 Main window list search bar single point check

Test Cases for main window	Yeas	No	Remark
Is the object/window being visible on the screen and does not overlap with any other elements?	**Y**		
Does object/window responsiveness to different screen sizes, orientations, and resolutions?	**Y**		
Is the object/window is located in the correct position on the screen as per the design specification?	**Y**		
Is the object/window function expected action correctly when it clicked, hovered and scrolled?	**Y**		
Is the object/window, including the ability to move to different parts of the object/window, scroll up and down, and use navigation controls?	**Y**		
All labels for the object/window are visible and accurate?	**Y**		
Are the tooltips that appear when hovering over any icons or buttons on the object/window to ensure that they are accurate and informative?		**N**	

All error messages that appear on the object/window to ensure that they are clear, informative, and help the user resolve any issues?	**Y**		
all keyboard commands, screen reader support, and other assistive technologies function correctly.	**Y**		

2.1.3. GUI checkpoint for multiple object

Table 2.18 checkpoint for multiple object search bar single point check

Test Cases checkpoint for multiple object	Yeas	No	Remark
Are the objects being visible to each?	**Y**		
Are the objects being in the correct position and size?	**Y**		
Are the objects being the correct color and font?	**Y**		
Are states of each object enabled and disabled correctly?	**Y**		
Are the objects respond correctly for user input?	**Y**		
Each objects functions correctly when they are clicked, hovers and etc. ...?	**Y**		
Are each objects behave as expected and produce the correct output or perform the correct action?	**Y**		
Is the data displayed in each object, such as text in a label or value in a dropdown?		**N**	
Are the objects display the correct data and update as expected?	**Y**		
Is the tooltip text displayed for each object?		**N**	
Is there keyboard shortcuts associated with each object?		**N**	
Is the error handling of each object, such as displaying an error message for invalid input.	**Y**		
Is the object hand error such as displaying an error message for invalid input?		**N**	
Are the objects handle errors correctly and provide helpful error messages?		**N**	
Are the objects have the correct internationalization support and work correctly with different languages and locales?		**N**	

2.2. Functionality

2.2.1 Database checkpoint for Default check

Table 2.19 Database checkpoint for Default check

Test Cases for Database checkpoint for Default check	Yeas	No	Remark
Is default database checkpoint interval is set to the correct value?	**Y**		
Is the database checkpoint process runs at the specified interval?	**Y**		
Is the database have ability to manually trigger a database checkpoint process and completes successfully?	**Y**		
Is the database checkpoint process frees up any unused disk space?		**N**	
Is the database checkpoint process under high system load and confirm that it continues to operate as expected?		**N**	
Is the database checkpoint process being compatible with the current version of the database software?	**Y**		
Is the database restore the data from a previous checkpoint and confirm that all data is intact?		**N**	
Is the database checkpoint process is properly configured to match the specific database requirements?	**Y**		
the database checkpoint process does not negatively impact the performance of other processes running on the same system?		**N**	

2.2.2 Database Checkpoint for Custom Check

2.2.3. Database checkpoint for runtime record check

Table 2.20 Database checkpoint for custom check

Test Cases for Database checkpoint for customs check	Yes	No	Remark
Is the custom check is configured with the correct parameters and criteria?	**Y**		
Is the custom check by intentionally inserting incorrect data into the database that violates the criteria set for the check?	**Y**		
Is the custom check identifies the incorrect data and generates the appropriate error or warning messages?	**Y**		
Is the custom check logs the errors or warnings generated during the check process?	**Y**		

3. DEVELOPMENT ENVIRONMENTS

There are several development tools and methods available for developing php web apps. Some of the popular ones include:

1. Integrated Development Environments (IDEs): IDEs such as Visual Studio Code, and Sublime Text provide a range of features such as code completion, debugging, and syntax highlighting that make it easier to develop Django web apps.

2. Version Control Systems (VCS): VCS such as Git and SVN allow developers to track changes made to the code and collaborate with other team members.

3. Testing Frameworks: Testing frameworks such as pytest and unittest help developers ensure that their bootsrap web apps are functioning correctly.

4. Agile Development Methodology: Agile development methodology emphasizes iterative development, continuous feedback, and collaboration between developers and stakeholders.

Overall, choosing the right development tools and methods can significantly improve the efficiency and quality of php web app development.

4. CONCLUSION

Our software testing of the web based infant immunization app has been a thorough and comprehensive process. We have conducted testing on various aspects of the application, including its functionality, usability, security, and performance. Our testing has helped to identify and address a range of issues and bugs, ensuring that the app is reliable, user-friendly, and secure.

In addition to testing the web app itself, we have also conducted GUI checkpoint testing for single and multiple objects and windows. For single properties, we have verified that the property is displayed correctly and that it meets the specified requirements. For single objects/windows, we have tested the object's functionality, usability, and accessibility to ensure that it works as intended and provides a seamless user experience. For multiple objects, we have verified that all objects are displayed correctly, enabled and disabled correctly, function correctly, have the correct keyboard shortcuts and accessibility features, and have the correct error handling and focus behavior.

We have also conducted database checkpoint testing, including default, custom, and runtime record checks. With default checks, we have verified that the database table is created correctly and that it contains the expected data. With custom checks, we have tested the data's accuracy and consistency, ensuring that it meets the specified requirements. With runtime record checks, we have tested the application's ability to handle and retrieve data in real-time, ensuring that it works as intended and that it can retrieve and display data correctly.

Overall, our testing efforts have significantly improved the quality of the web app and provided a more satisfying experience for its users. Our comprehensive approach to testing has helped to identify and address issues and bugs, ensuring that the app is reliable, secure, and user-friendly.