KARNATAK UNIVERSITY

DHARWAD

Janata Shikshana Samiti's

SHRI MANJUNATHESHWARA INSTITUTE OF UG & PG STUDIES, VIDYAGIRI, DHARWAD - 580 004

A

PROJECT REPORT ON

"Child Vaccination"

BACHELOR OF COMPUTER SCIENCE (BSc (CS))

OF

KARNATAK UNIVERSITY, DHARWAD

PROJECT GUIDED BY:

Prof. Narasimha Bhat

Submitted by:

Kavya Hegde	Vinoda Patil
BSc (CS) V Semester	BSc (CS) V Semester
20M10132	20M10186

DEPARTMENT OF COMPUTER SCIENCE 2022-23

Janata Shikshana Samiti's

Shri Manjunatheshwara Institute of UG & PG Studies, Vidyagiri, Dharwad. 580004

CERTIFICATE

This is to certify that Ms. Kavya Hegde and Ms. Vinoda Patil has satisfactorily completed Project Work entitled "CHILD VACCINATION" for the partial fulfillment of BSc (CS) prescribed by Karnatak University Dharwad during the academic year 2022-2023.

Prof. Narasimha Bhat	Prof. Vivek. M. Laxmeshwar	Dr. Ajith Prasad
[Project Guide]	[Head of The Department]	[Principal]

Examination Register No:	Examiners
1. 20M10132, Kavya Hegde	1. ....
2. 20M10186, Vinoda Patil	2. ....

ACKNOWLEDGEMENT

The successful presentation of this project is acknowledgements of the immense support extended by JSS SMI UG & PG STUDIES, DHARWAD, which has provided us opportunity to fulfill the most cherished & desired way to reach our goal.

We would like to express our heartfelt thanks to our President Shri Shri Vishwaprasanna Teertha Swamiji of Sri Pejavarmath, Udupi and Parama Pujya Dr. D. Veerendra Heggadeji, the Chairman of Janata Shikshana Samiti & Dharmadhikari of Shree Kshetra Dharmasthala.

*We would like to express our sincere and hearty thanks of gratitude to our beloved Secretary of JSS & Principal Dr. Ajith Prasad and our Head of the Computer Science Department Prof.Vivek.M. Laxmeshwar who gave us the golden opportunity to do this wonderful project on the topic "**Child Vaccination**", which also helped us in doing a lot of research and exposed to lot of new information that would help us in our mere future.*

We would also take this opportunity to offer our sincere gratitude to our Project guide Prof. Narasimha Bhat for his excellent support throughout the development of this project and providing the necessary information on our demand at all the times.

Kavya Hegde & Vinoda Patil

DECLARATION

We, ***Kavya. B. Hegde & Vinoda. M. Patil*** *students of Fifth Semester BSc (CS), Department of Computer Science, JSS SHREE MANJUNATHESHWARA INSTITUTE OF UG & PG STUDIES, VIDYAGIRI, DHARWAD affiliated to Karnatak University, Dharwad hereby declare that the Project entitled* ***"Child Vaccination"*** *submitted in partial fulfillment of the course requirement for the award of degree in Bachelor of Computer Science, Karnatak University, Dharwad during the academic year 2022-2023. We have not submitted the matter embodied to any other University or Institution for the award of any other degree.*

Date: ***Kavya. B. Hegde***

Place: Dharwad ***Vinoda. M. Patil***

CONTENTS

THE PROJECT SYNOPSIS

INTRODUCTION:

Right from birth till the age of 12 years the child will be vulnerable to a host of diseases, viruses and bacteria at such tender age. The body of the child may not have the required immune system to protect them from the diseases so make sure that children have access to proper healthcare and immunization against diseases. this system helps parents book vaccination appointment for their children with just few clicks.

1.OBJECTIVE OF THE PROJECT:

Young children are at increased risk for infectious diseases because their immune systems have not yet built up the necessary defenses to fight serious infections and diseases. Making sure that children have access to proper healthcare and immunization against diseases that can be prevented by vaccines, is a huge challenge that is being faced by developing countries like ours.

This highlights the importance and need of having a better, smarter system in place, to improve the situations. This application provides a system to provide information, store records and help parents schedule vaccination appointments for their children.

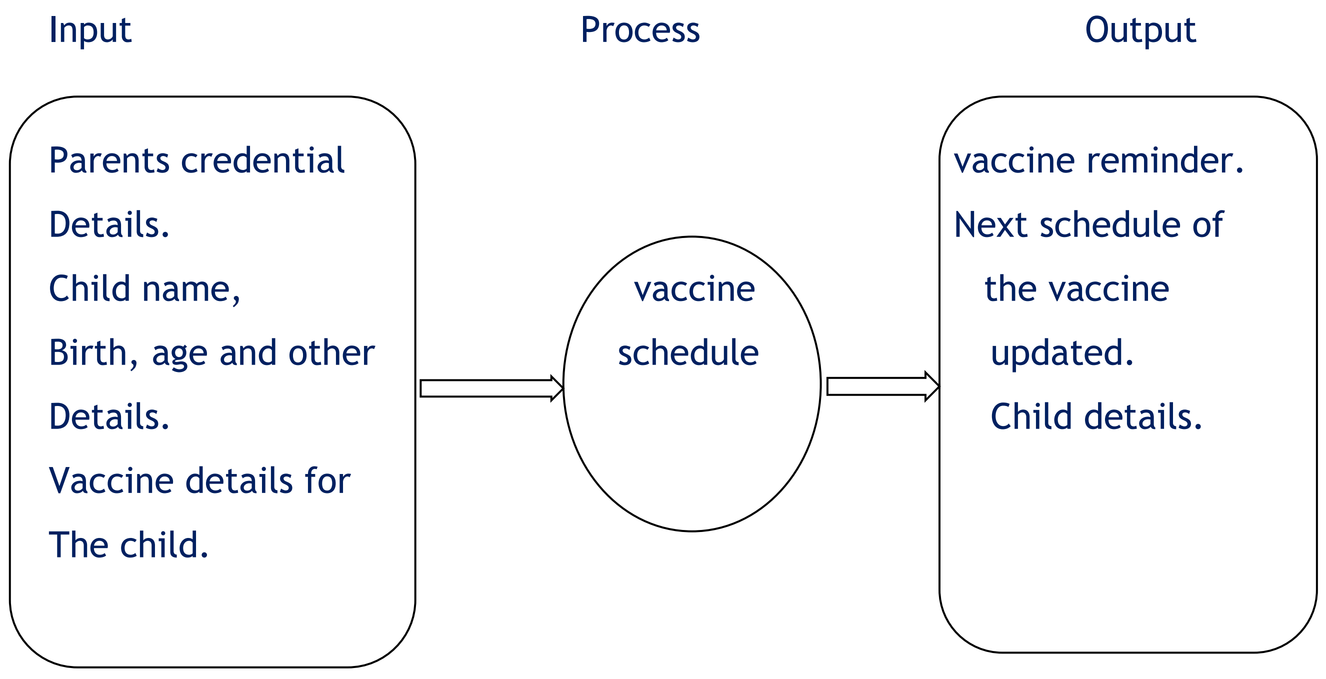
2.INPUT OF THE PROJECT:

- The main input to the project is that the user login to the website “child vaccination” by entering the credential details and then the system helps to book appointments for their child’s vaccination easily.
- Stock of the vaccine can be added.
- Parents can view their updates and also their reminders for appointments.

3. OUTPUT OF THE PROJECT:

- This application provides a system to provide information store a record.
- Help a parent's schedule vaccination appointments for their children.
- Admin will manage the child and vaccination report approval of the appointment.
- Hospital will update the status of the vaccination applied for the child. with this application, a lot of man hours can be saved.

4.Process logic:



System Requirement Specification

Hardware Requirements

SL.NO	HARDWARE	DESCRIPTICON
1	PROCESSOR	11th Gen Intel®Core™
2	RAM	8.00GB and above
3	HARD DISK	250GB of available hard disk space
4	PROCESSOR SPEED	3.00GHz

Software Requirements

SL.NO.	SOFTWARE	DESCRIPTION
1	SERVER SIDE SCRIPT	PHP
2	OPERATING SYSTEM	Windows 11
3	DATABASE	MySQL
4	FRONT END	HTML, CSS, JavaScript, Bootstrap, Sublime
5	WEB SERVER	XAMPP

ANALYSIS

a) System requirements:

System specification forms the foundation on which the architecture, design, and implementation of a software is built. Documents containing system specifications are critical because major requirements as a result of not having a specification document on hand. System specification documents can thus be defined as the requirements documentation that formally specifies the system-level requirements of a software application.

System specification documents most predominantly contain information on basic website requirements which include:

- performance levels
- reliability
- quality
- interfaces
- security and privacy
- constraints and limitations
- functional capabilities
- data structures and elements

b) Existing System:

- In the present system all the works are non-computerized.
- Several types of Immunization cards are in use like BMC provided immunization card.
- It consumes more time and also needs more human efforts.

c) Proposed Systems:

- The proposed system is a web-based application that reduces the paper work.
- The system helps to book appointments for their child's vaccination easily.
- Parents can view their updates and also their reminders for appointments.
- Secure access control and prevents misuse by unauthorized users.
- The proposed system consists of web-based GUI and provides easy access to its user.
- To create user friendly interface to operate for parents.
- Fully validated to ensure correct and error free data entry.

Modules:

- **ADMIN:**

- **Login: The admin can login to the system using a username and password.**
- **View Hospitals**
  - **Accept/Reject: They can view all the hospital details. Also, they can accept or reject the hospital's application.**
- **View Child Data: The admin can view all the child data**
- **Appointment Details: All the appointment details can be viewed by the admin.**
- **Export/Save Data: All the appointment data can be saved or exported by the admin.**

❖ <u>HOSPITAL:</u>

- Register: The hospital will need to register first and request approval from the admin.
- Login: They can log in to the system only after it has been approved by the admin.
- View Appointments
    - Filter by Date: The hospital can view appointments by date after applying a filter to sort.
    - Update Status: They can update the status of appointments.

❖ <u>PARENT:</u>

- Register: Parents will need to register first to log in with their basic details.
- Login: They can log in using their username and password.
- My profile: Also, they can make necessary changes to their profile details.
- Change Password: They can easily change their old password to the new one.
- Manage Child data: Parents can manage and view their child’s data, and also, they can add, update and delete these data.
- Book Appointment: They can further book an appointment for their child’s vaccination.
- View Appointments
    - Details & Status: Parents can view appointment details and their status.
- My Reminder: They can also view the appointment reminder.

Limitations:

- Parents will need to upload their child's data correctly, otherwise, it will be led to the wrong output.
- To keep the database up to date regular update is needed.
- There is no particular booking of the system for child vaccines

Advantages:

- **It is easy to maintain.**
- **It's user-friendly.**
- **The system helps to book appointments for their child's vaccination easily.**
- **Parents can view their updates and also their reminders for appointments.**

Future Enhancement of The Project:

- ➢ In future Android application can be developed.
- ➢ Home schedule of vaccine can be added.
- ➢ It provides the convenient way of communication to parents through chatbots.
- ➢ Multiple hospital option can be added.

FRAMEWORK

A. What is XAMPP?

XAMPP is an abbreviation where *X stands for Cross-Platform, A stands for Apache, M stands for* ***MYSQL****, and the Ps stand for PHP and Perl*, respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command-line executables along with modules such as Apache server, MariaDB, PHP, and Perl.

XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies, Perl is a programming language used for web development, PHP is a backend scripting language, and MariaDB is the most vividly used database developed by MySQL.

B. Hypertext Pre-Processor (PHP):

PHP stands for Hypertext Preprocessor created by **Rasmus Lerdorf**, it is a server-side scripting language that powers some of the most popular websites in the world, including WordPress and Facebook. However, PHP alone isn't enough in order to build dynamic web sites. To use PHP on a web site, we need a server that can process PHP scripts. XAMPP server allows developers to test PHP scripts locally, this makes it an invaluable piece of your local development environment.

Additionally, dynamic websites are dependent on stored information that can be added and updated easily, this is the main difference between a dynamic application and a static application. However, PHP doesn't provide a simple, efficient way to store data. This is where a relational database management system like MySQL comes into play.

Syntax:
```
<?PHP
        //PHP CODE
    ?>
```

C.JAVASCRIPT:

JavaScript is a dynamic computer programming language. It is most commonly used as part of Web browsers, whose implementations allow client-side **scripts** to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed.

D.HTML:

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages. HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999. Though HTML 4.01 version is widely used but currently we are having HTML-5 version which is an extension to HTML 4.01, and this version was published in 2012.

Syntax:
```
<! DOCTYPE html>
<html>
        <head>
        <title>This is a Title </title>
        </head>
        <body>
<p>Hello World! </p>
        </body>
</html>
```

<! DOCTYPE>

This tag defines the document type and HTML version.

<HTML>

This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.

<HEAD>

This tag represents the document's header which can keep other HTML tags like <title>, <link> etc.

<TITLE>

The <title> tag is used inside the <head> tag to mention the document title.

<BODY>This tag represents the document's body which keeps other HTML tags like<h1>, <div>, <p> etc.

HTML Elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img/> and <input/> introduce content into the page directly. Others such as<p>.... </p>, surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

E. Cascading Style Sheet (CSS):

Cascading Style Sheets was invented by **HakonWium Lie** on October 7, 1994 and maintained through a group of people within the W3C called the CSS Working Group

CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. There are three types CSS mainly

- External CSS
- Internal CSS
- Inline CSS

a. **External** style sheets are separate files full of CSS instructions (with the file extension .CSS). When any web page includes an external style sheet, its look and feel will be controlled by this CSS file. This is how you change a whole website at once. And that's perfect if you want to keep up with the latest fashion in web pages without rewriting every page.

b. **Internal** styles are placed at the top of each web page document, before any of the content is listed. This is the next best thing to external, because they're easy to find, yet allow you to 'override' an external style sheet -- for that special page that wants to be a nonconformist.

c. **Inline** styles are placed right where you need them, next to the text or graphic you wish to decorate. You can insert inline styles anywhere in the middle of your HTML code, giving you real freedom to specify each web page element. On the other hand, this can make maintaining web pages a real chore.

Advantages of CSS

- **CSS saves time** − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So, less code means faster download times.
- **Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- **Superior styles to HTML** − CSS have a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- **Multiple Device Compatibility** − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.

- **Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. So, it's a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.
- **Offline Browsing** − CSS can store web applications locally with the help of an offline cache. Using of this, we can view offline websites. The cache also ensures faster loading and better overall performance of the website.
- **Platform Independence** − The Script offer consistent platform independence and can support latest browsers as well.

F. MYSQL:

SQL is a Relational Database Management System (RDBMS) that runs exclusively under the Windows operating system. One benefit of using Windows exclusively is that we can send and receive E-mail messages based on SQL "events" and we can also let the operating system handle login security. The data base is an organized collection of data. A database management system (DBMS) such as Access, FileMaker Pro, Oracle or a flexible manner. It includes facilities to add, modify or delete data from the SQL Server provides we with the software tools we need to organize that data in database, ask questions (or queries)
about the data stored in the database and produce reports summarizing selected contents.

Database Connection

```
<?PHP
$servername ="localhost";
$username = "username";
$password = "password";
```

```
$dbname = "database";
// Create connection
$conn =mysqli_connect ($servername, $username, $password, $dbname);
// Check connection
if (! $conn)
{
die ("Connection failed: “. mysqli_connect_error ());

}    ?>
```

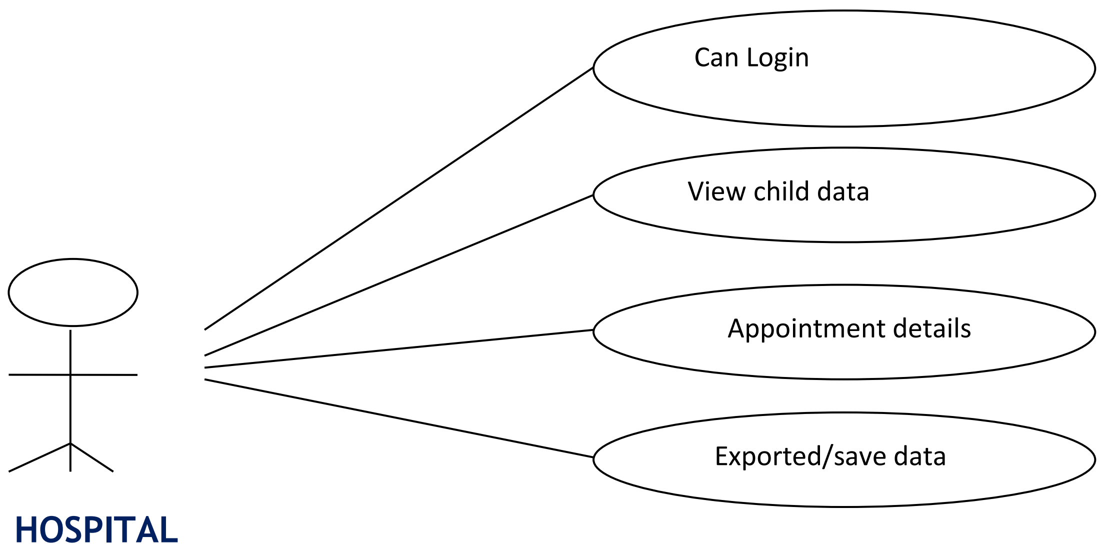
a) FEATURES OF SQL:

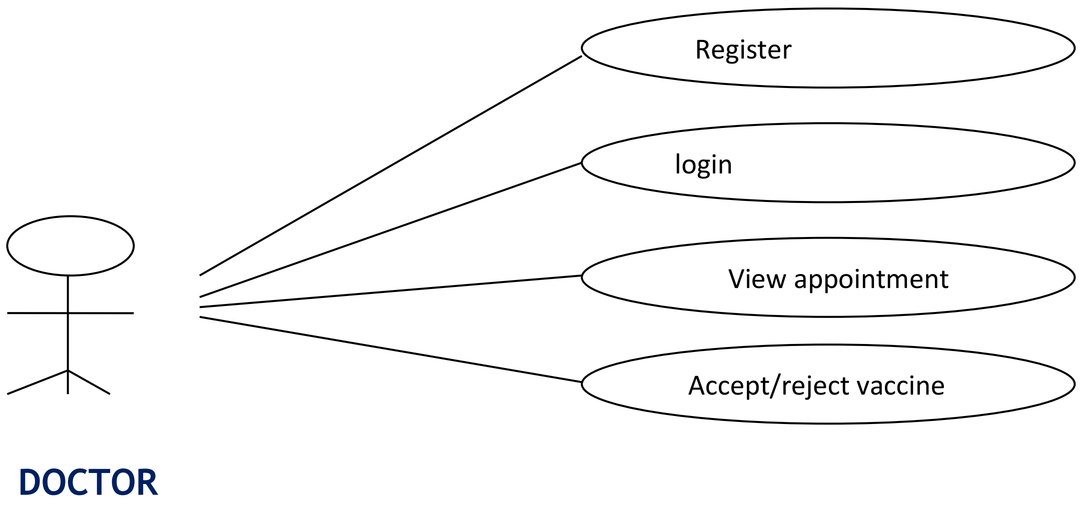
1. It is simple English like language and uses simple commands such as SELECT, CREATE, DROP etc.
2. It is not having condition loops, variables and most of the commands are single line commands.
3. To implement application logics, SQL has got extension language popularly called as PL/SQL (Procedural language of sql).
4. The entire SQL has been divided into 4 major categories.
   a. Data Manipulation Language.
   b. Data Definition Language.
   c. Transaction Control language.
   d. Data Control Language.

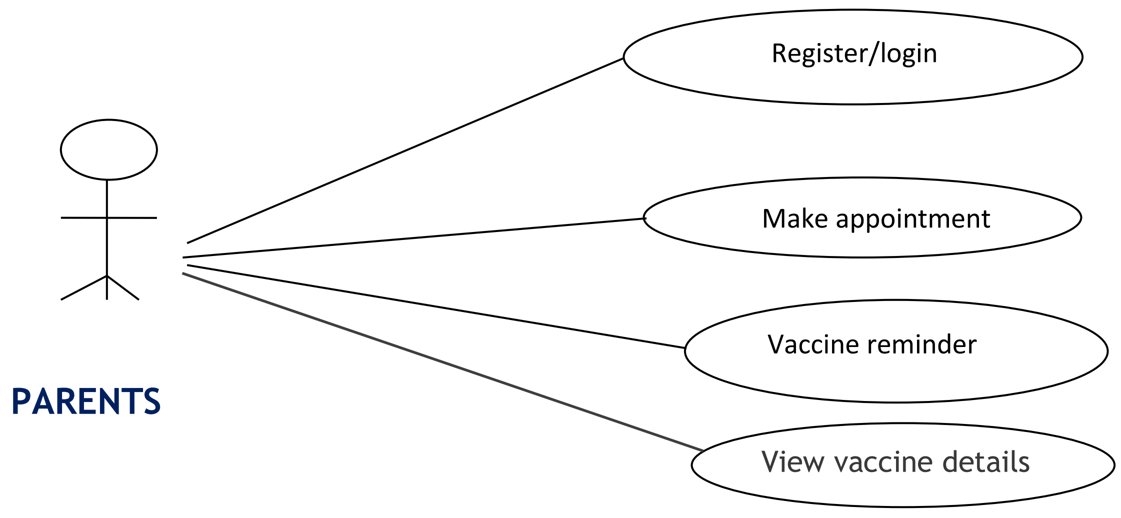
Use Case Diagrams

USE CASES DESCRIPTION:

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. Use case is a list of steps, typically defining interactions between a role (known in UML as an actor) and a system, to achieve a goal. The actor can be a human or an external system.




Register
login
View appointment
Accept/reject vaccine
DOCTOR


Register/login
Make appointment
Vaccine reminder
View vaccine details
PARENTS

Software Requirement Specification (SRS)

AIM:

PREPARE A SOFTWARE REQUIREMENT SPECIFICATION (SRS) DOCUMENT FOR A GIVEN PROBLEM STATEMENT.

PROBLEM STATEMENT:

THIS PROJECT IS TO DEVELOP A WEB APPLICATION FOR THE HOSPITAL. IT DESCRIBE THE FUNCTIONS AND TASKS THAT THE SYSTEM CAN PERFORM. THIS IS USED TO DESCRIBE THE SCOPE OF THE PROJECT AND TO PLAN FOR THE SYSTEM DESIGN AND IMPLEMENTATION. IT KEEPS DETAILS AND HISTORY OF A CHILD AND VACCINATION DETAILS. IT GENERATES THE VACCINE REMAINDER FOR THE CHILDS SALES REPORT. THE SOFTWARE HANDLES MOST NECESSARY DETAILS.

OBJECTIVES:

TO UNDERSTAND DIFFERENT SECTIONS OF SOFTWARE REQUIREMENT SPECIFICATION (SRS).

- TO UNDERSTAND FUNCTIONAL REQUIREMENTS OF THE SYSTEM.
- TO UNDERSTAND PERFORMANCE REQUIREMENTS OF THE SYSTEM.
- TO APPLY DESIGN CONSTRAINTS AND APPROPRIATE VALIDATION ON THE WEB APPLICATION.

THEORY:

SOFTWARE REQUIREMENT SPECIFICATION (SRS) DOCUMENT USUALLY CONTAINS A CHILD VACCINATION DETAILS BY UNDERSTANDING THE PARENTS CREDENTIAL DETAILS. IT DESCRIBES THE THE NON-FUNCTIONS GOALS AND TASKS THAT THE SYSTEM CAN PERFORM.THIS IS USED TO DESCRIBE THE SCOPE OF THE PROJECT AND TO PLAN FOR THE SYSTEMS DESIGN AND IMPLENTATION.

Introduction

Purpose:

The software is for automation of child vaccine details. The software includes maintaining child details along with its vaccination details providing and maintaining all kinds of vaccine from birth to 12 years of child.

Scope of development project:

The proposed software is the child vaccination system the system will be used to get the information from the parents and then storing that data for future usage the current system in use is paper based system it is too slow and cannot provide updated child vaccination within reasonable time period the intention of the system is to reduce the work. requirement statement in this document is both functional and non-functional.

References

- Books
- Websites

Overall Description:

Product Functions

The child vaccination projects help parents to maintain their child information in the form of data in website. this website helps to maintain their child information and also get regular vaccination records and reminders for their child to get vaccinated on time.

The main purpose of this project is to reduce the manual work this web application is capable of **maintaining child information, along with type of vaccine and concerned doctor as well.**

Operating Environment:

The product will be operating in windows environment. The basic input devices required are keyboard, mouse and output devices are monitor, printer etc.

Development Environment:

Software Configuration:

This web application uses,

Operating system: windows 11

Language: HTML, CSS, Javascript (Front end)

Database: MYSQL (Back end)

Server-Side Script: PHP

Web server: Xamp Server

Hardware Configuration:

Processor: 11th Gen Intel(R) Core (TM)

Processor Speed: 3.00GHz

RAM: 8.00GB

Hard Disk: 20GB free space

Data Requirement

Parents details, child details:

The inputs consist of the query to the database and output consists of the solutions for the query. In this project the inputs will be the queries as fired by the parents for vaccine for their children. Keeping an accurate database of parents as well as children information.

External Interface Requirements

The purpose of this section is to identify and document interfaces and interaction of the website with external entities in detail.

The website provides good graphical interface for the owners/managers who can operate on the system, performing the required task such as create, update, viewing the details of the plant sales/availability/sold. The website should provide payment option in online as well as offline mode.

Functional requirements

Functional requirements are the following:

1. User will be able to add the profile of children based on name, age father’s name, date of birth, height, weight, profile picture,etc.
2. For each child, the user will add the record of the medical history of a child.
3. The user will able to add the vaccination record of each child.
4. It will give an alert message if the vaccination date is near or has passed.
5. There should be a facility to upload the images of prescriptions against illness records.

System Features:

The Hospital should be providing the surety that their data is secure. This is possible by providing:

- Parents authentication and validation using their unique parents id.
- Proper monitoring by the admin which includes updating data.
- Proper vaccine details are maintained for every children.

Other Non-functional Requirements

Performance Requirement

The proposed website that we are going to develop will be used as the Chief performance system by particular hospital which interacts with parents regarding to their child vaccine.

Therefore, it is expected that the hospital would perform functionally all the requirements that are required by their parents.

- The performance of the system should be fast accurate.

This system must support many people at a time the user interface screen shall response within time. The system must confirm to the Microsoft accessibility.

Safety Requirements

- Website will use secured database.
- Normal users can just read information but they cannot edit or modify except their personal and some other information.
- System will have different types of users and every user has access constraints.
- Proper user authentication should be provided.

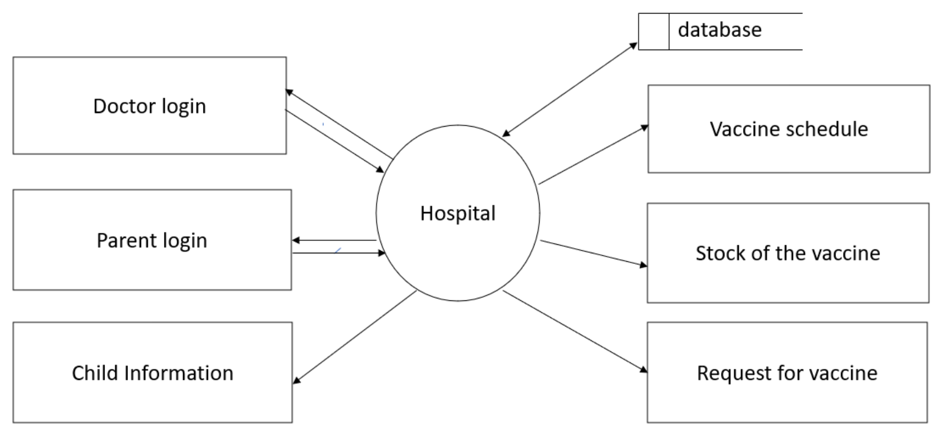
Conclusion:

This SRS document is used to give details regarding child vaccination details in this all functional and non- functional requirement are specified in order to get clear cut idea to develop a project.

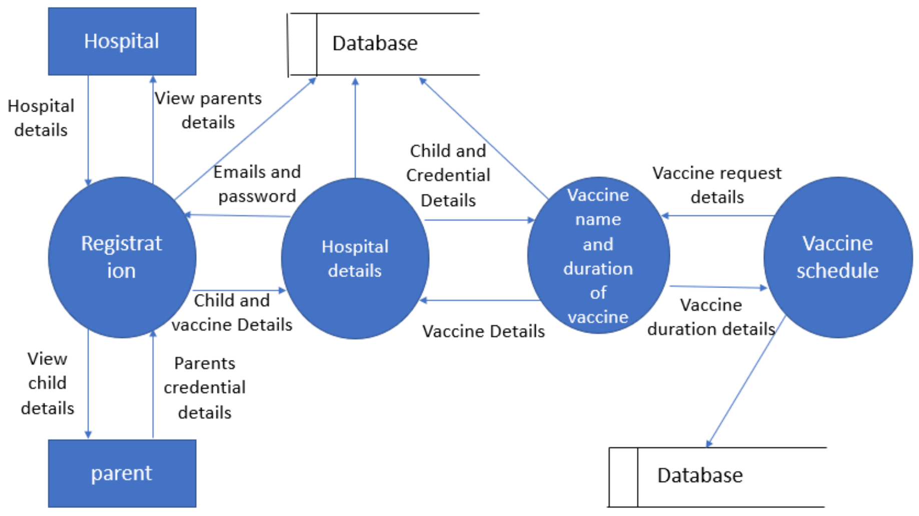
DATA FLOW DIAGRAM

A data flow diagram (DFD) is a significant modeling technique for analyzing and constructing information processes. Data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). On a DFD, data items flow from an external data source or an internal data store to an Internal data store or an external data sink, via an internal process. A DFD provides no information about the timing or ordering of processes, or about whether processes will operate in sequence or in parallel. It is therefore quite different from a flowchart.

<u>Level Zero DFD:</u>



Level One DFD:



Entity Relationship Diagram (ER DIAGRAM)

An entity-relationship diagram is a data modelling technique that creates a graphical representation of the entities, and the relationships between entities, within an information system. An entity-relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship modelling is a database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called entity-relationship diagrams, ER diagrams, or ERDs.

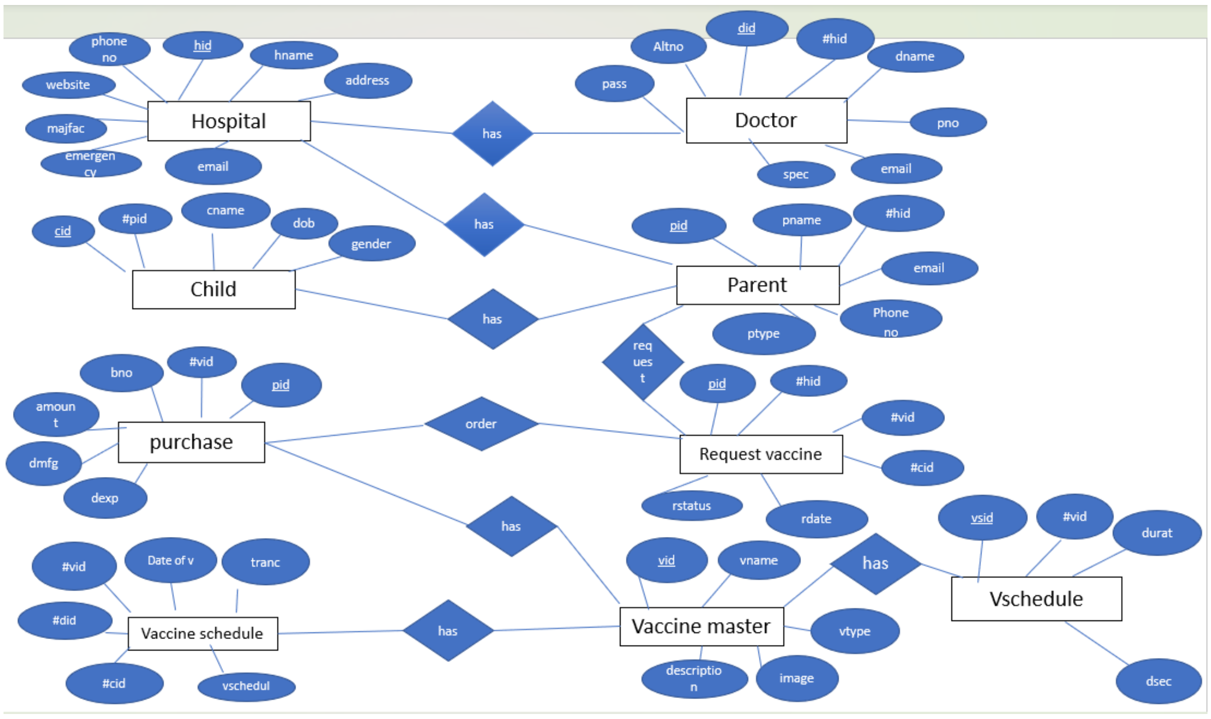
The three main components of an E-R Diagram are:

- The entity is a person, object, place or event for which data is collected. For example, if you consider the information system for a business, entities would include not only customers, but the customer's address, and orders as well. The entity is represented by a rectangle and labelled with a singular noun.
- The relationship is the interaction between the entities. In the example above, the customer places an order, so the word "places" defines the relationship between that instance of a customer and the order or orders that they place. A relationship may be represented by a diamond

shape, or more simply, by the line connecting the entities. In either case, verbs are used to label the relationships.

- The cardinality defines the relationship between the entities in terms of numbers. An entity may be optional: for example, a sales representative could have no customers or could have one or many customers; or mandatory: for example, there must be at least one product listed in an order. There are several different types of cardinality notations; crow's foot notation, used here, is a common one. In crow's foot notation, a single bar indicates one, a double bar indicates one and only one (for example, a single instance of a product can only be stored in one warehouse), a circle indicates zero, and a crow's foot indicates many. The three main cardinal relationships are: one-to-one, expressed as 1:1; one-to-many, expressed as 1: N; and many-to-many, expressed as M: N.

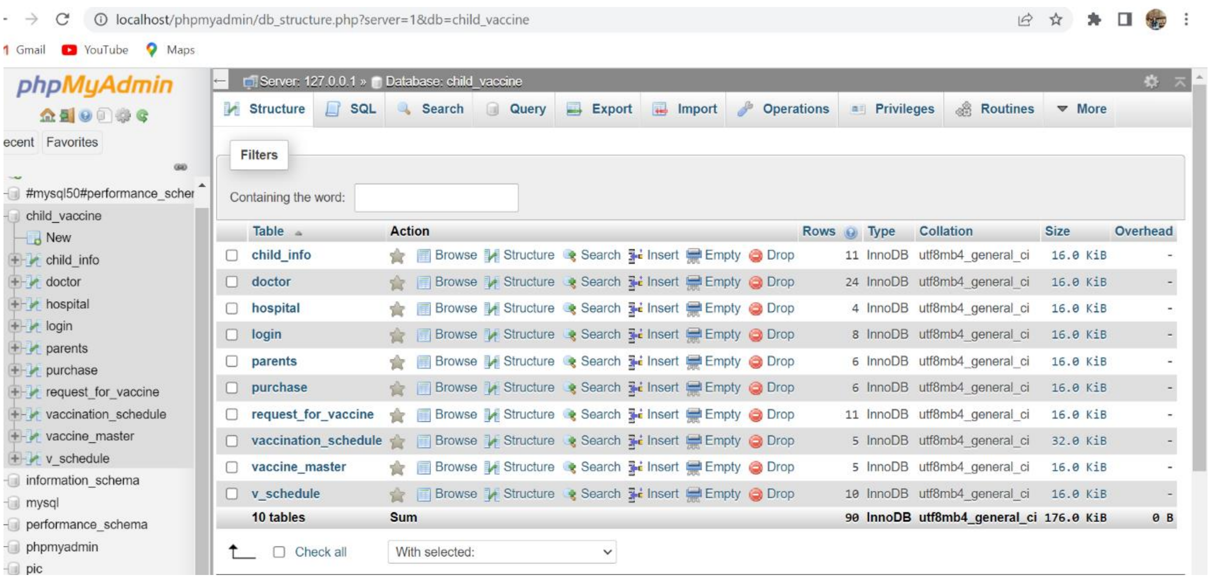
E-R Diagram for Child Vaccinations System:



SYSTEM DEVELOPMENT

The development phase focuses on how, that is, during development a software engineer attempts to define how data are to be structured, how function is to be implemented within a software architecture, how procedural details are to be implemented, how interfaces are to be characterized and how testing will be performed.

TABLES USED IN DATABASE:

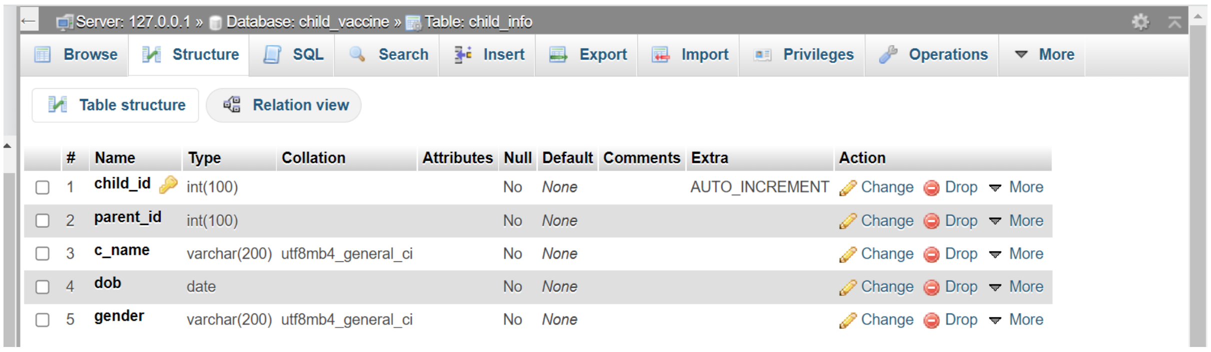


localhost/phpmyadmin/db_structure.php?server=1&db=child_vaccine

Server: 127.0.0.1 » Database: child_vaccine

	Table	Action	Rows	Type	Collation	Size	Overhead
☐	child_info	Browse Structure Search Insert Empty Drop	11	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	doctor	Browse Structure Search Insert Empty Drop	24	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	hospital	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	login	Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	parents	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	purchase	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	request_for_vaccine	Browse Structure Search Insert Empty Drop	11	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	vaccination_schedule	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	vaccine_master	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	v_schedule	Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_general_ci	16.0 KiB	-
	10 tables	Sum	90	InnoDB	utf8mb4_general_ci	176.0 KiB	0 B

1.child information table



Server: 127.0.0.1 » Database: child_vaccine » Table: child_info

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	child_id	int(100)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	parent_id	int(100)			No	*None*			Change Drop More
☐	3	c_name	varchar(200)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	4	dob	date			No	*None*			Change Drop More
☐	5	gender	varchar(200)	utf8mb4_general_ci		No	*None*			Change Drop More

2.Doctor Table

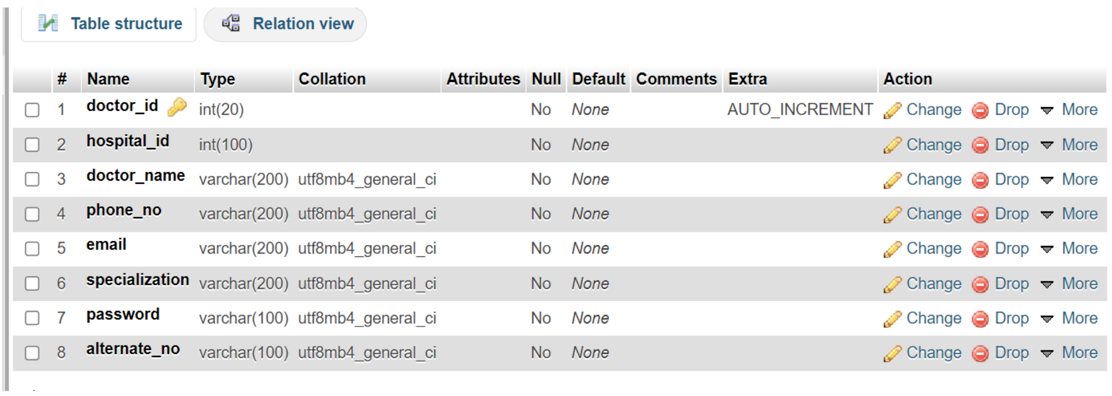


Table structure | Relation view

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	doctor_id	int(20)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	hospital_id	int(100)			No	*None*			Change Drop More
☐	3	doctor_name	varchar(200)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	4	phone_no	varchar(200)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	5	email	varchar(200)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	6	specialization	varchar(200)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	7	password	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	8	alternate_no	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More

3.Hospital Table

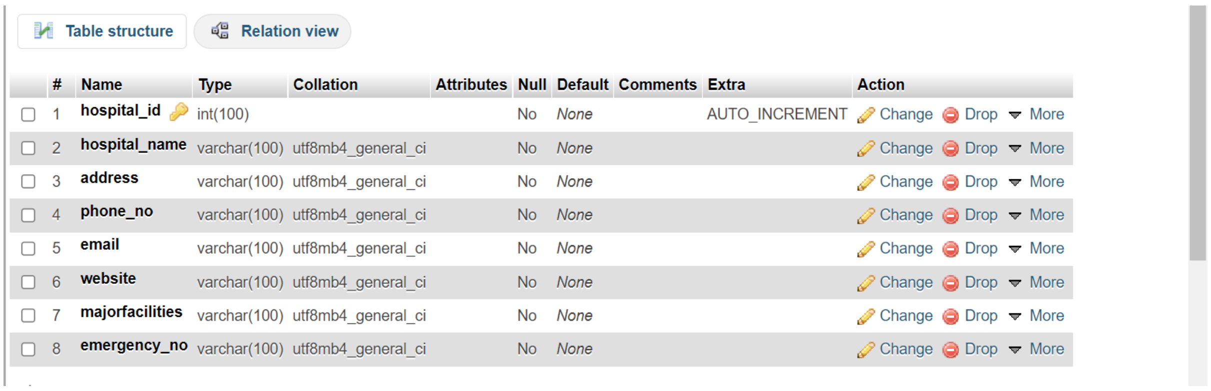


Table structure | Relation view

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	hospital_id	int(100)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	hospital_name	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	3	address	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	4	phone_no	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	5	email	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	6	website	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	7	majorfacilities	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More
☐	8	emergency_no	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More

4.Purchase Table

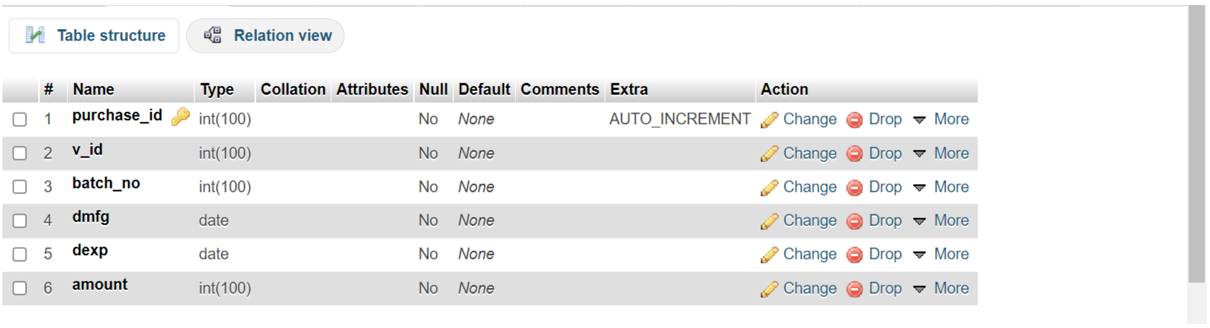


Table structure | Relation view

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	purchase_id	int(100)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	v_id	int(100)			No	*None*			Change Drop More
☐	3	batch_no	int(100)			No	*None*			Change Drop More
☐	4	dmfg	date			No	*None*			Change Drop More
☐	5	dexp	date			No	*None*			Change Drop More
☐	6	amount	int(100)			No	*None*			Change Drop More

5.Request Vaccine Table

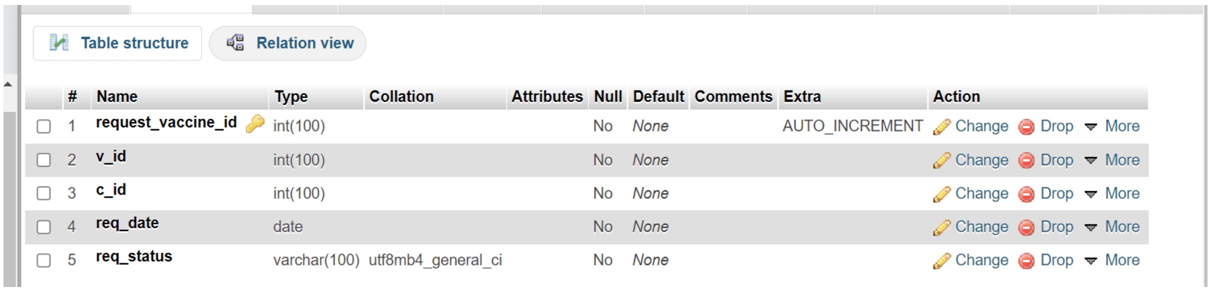
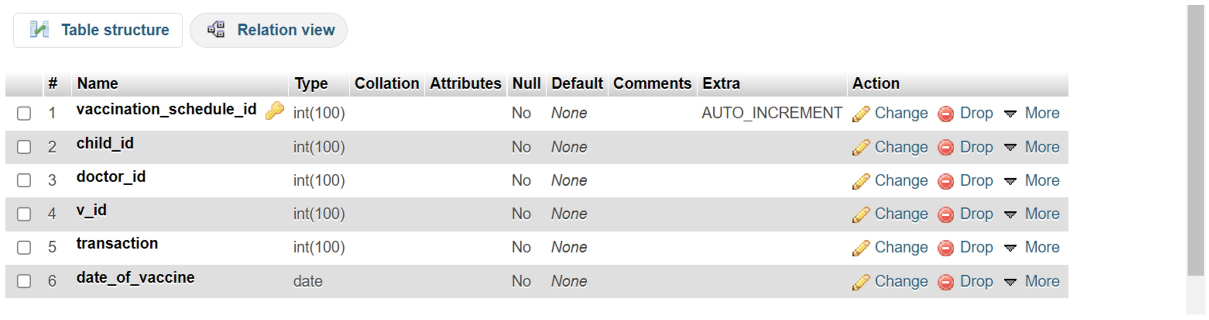


Table structure | Relation view

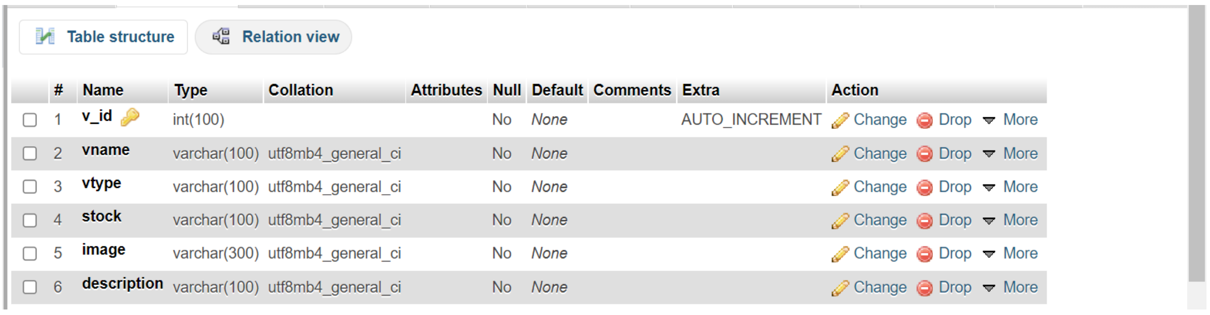
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	request_vaccine_id	int(100)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	v_id	int(100)			No	*None*			Change Drop More
☐	3	c_id	int(100)			No	*None*			Change Drop More
☐	4	req_date	date			No	*None*			Change Drop More
☐	5	req_status	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop More

6.Vaccination schedule Table



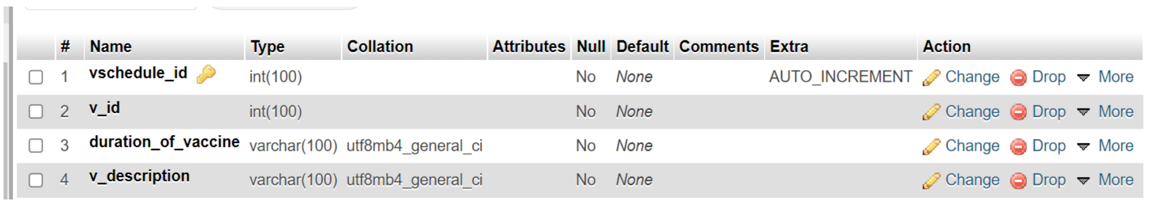
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	vaccination_schedule_id	int(100)			No	None		AUTO_INCREMENT	Change Drop More
2	child_id	int(100)			No	None			Change Drop More
3	doctor_id	int(100)			No	None			Change Drop More
4	v_id	int(100)			No	None			Change Drop More
5	transaction	int(100)			No	None			Change Drop More
6	date_of_vaccine	date			No	None			Change Drop More

7.vaccine Master Table



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	v_id	int(100)			No	None		AUTO_INCREMENT	Change Drop More
2	vname	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
3	vtype	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
4	stock	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
5	image	varchar(300)	utf8mb4_general_ci		No	None			Change Drop More
6	description	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More

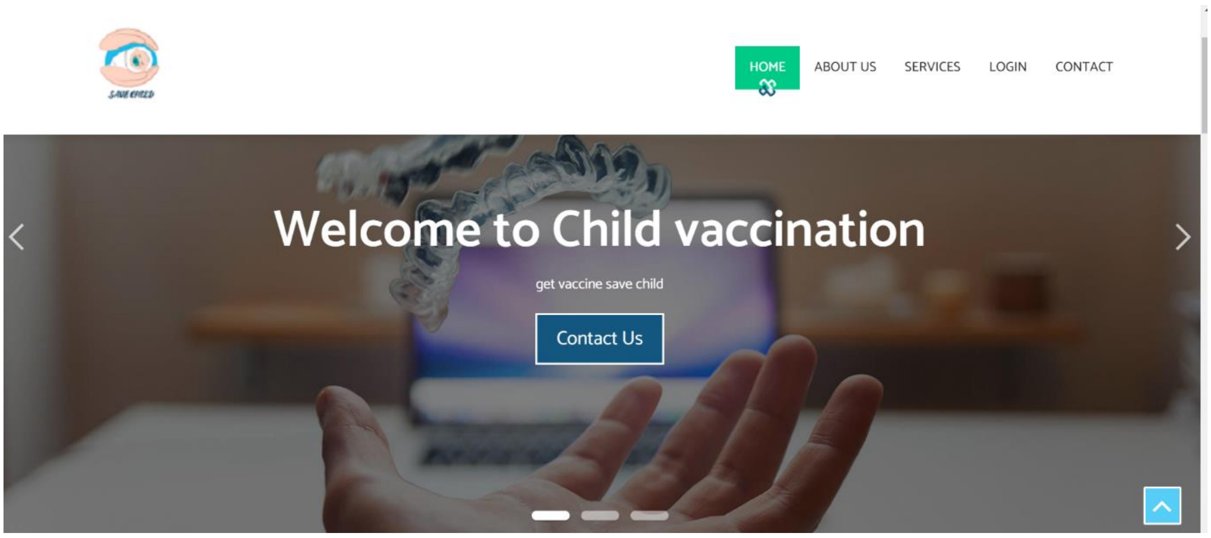
8.Vaccine Schedule Table

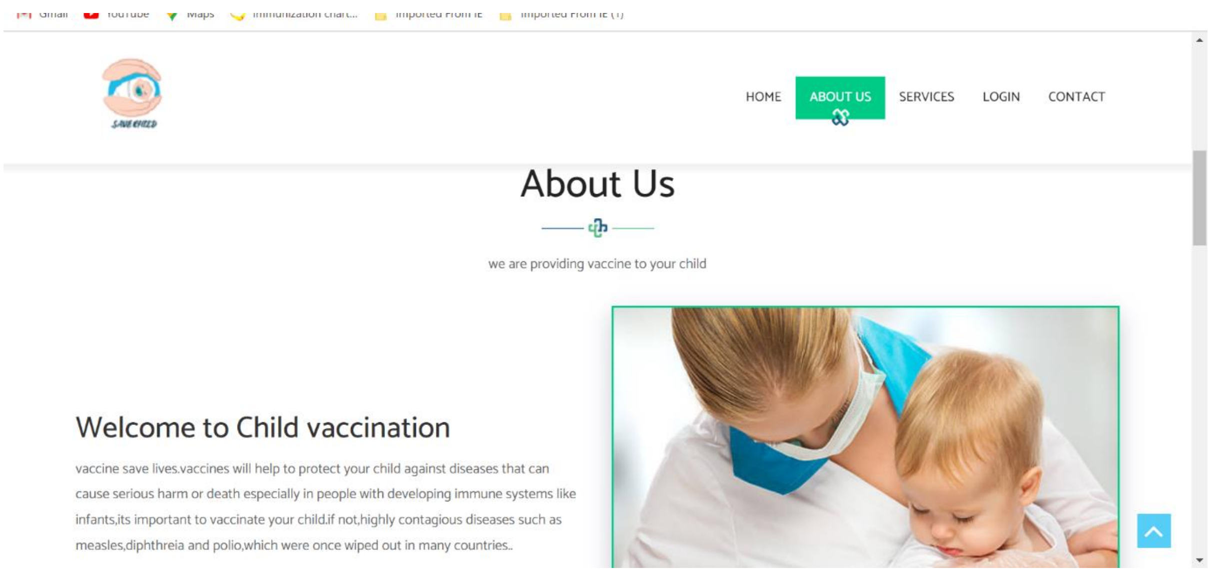


#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	vschedule_id	int(100)			No	None		AUTO_INCREMENT	Change Drop More
2	v_id	int(100)			No	None			Change Drop More
3	duration_of_vaccine	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
4	v_description	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More

SCREENSHOTS AND SOURCECODE

HOME PAGE:





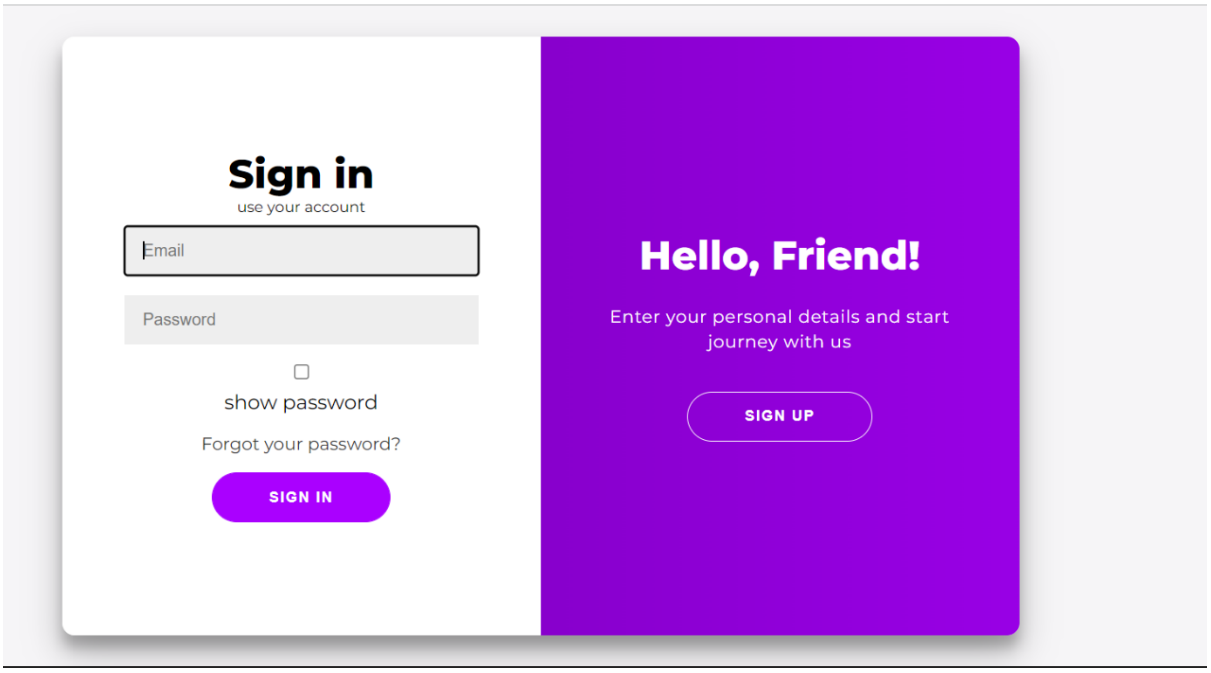
About Us

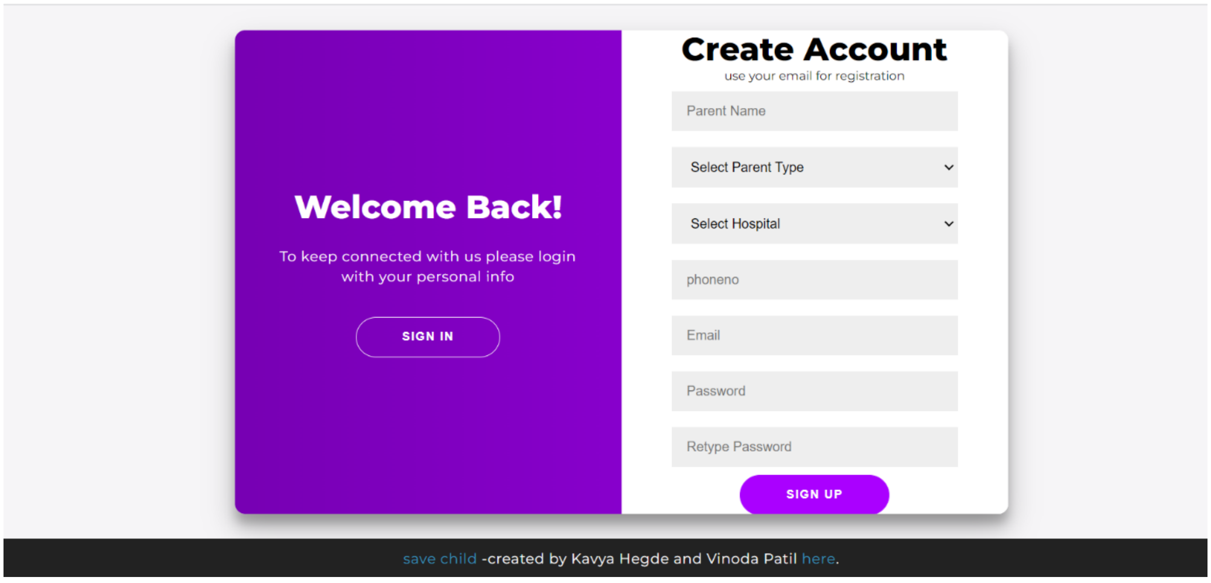
we are providing vaccine to your child

Welcome to Child vaccination

vaccine save lives.vaccines will help to protect your child against diseases that can cause serious harm or death especially in people with developing immune systems like infants,its important to vaccinate your child.if not,highly contagious diseases such as measles,diphthreia and polio,which were once wiped out in many countries..

Admin and parent login page:




Welcome Back!
To keep connected with us please login with your personal info
SIGN IN
Create Account
use your email for registration
Parent Name
Select Parent Type
Select Hospital
phoneno
Email
Password
Retype Password
SIGN UP
save child -created by Kavya Hegde and Vinoda Patil here.

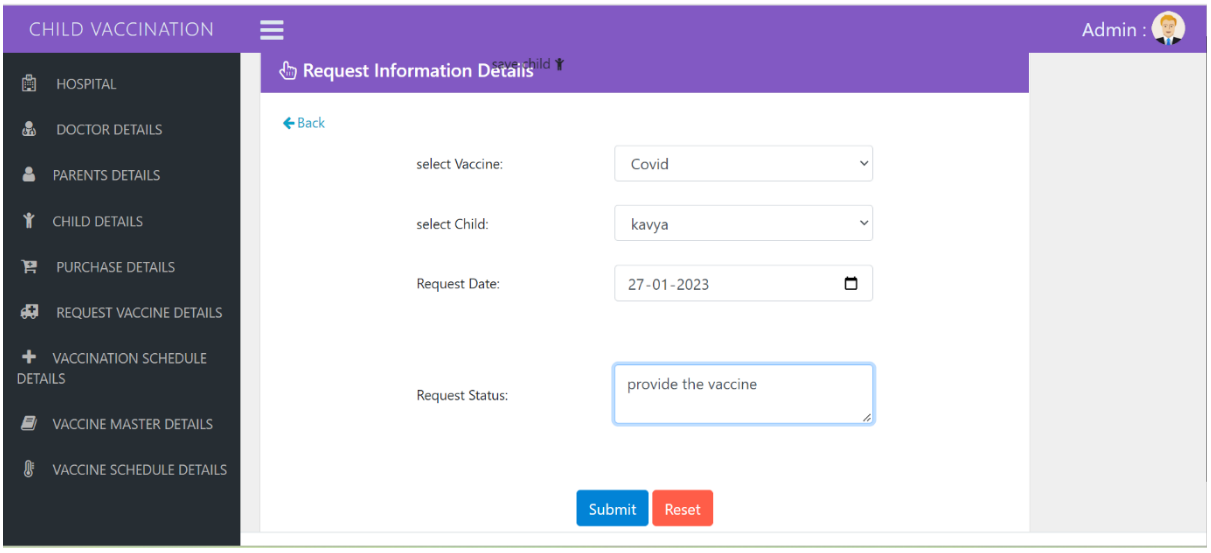
```
<div class="container" id="container">
<div class="form-container sign-up-container">
<form action="reg_insert.php" method="post">
<h1>Create Account</h1>
<span>use your email for registration</span>
<input type="text" placeholder="Parent Name" name="parent name"/>
<select name="p_type">
```

```
<option>Select Parent Type</option>
<option>Father</option>
<option>Mother</option>
<option>Other</option>
</select>
<select name="hospital_id" id="hospital_id" class="form-control">
<option>Select Hospital</option>
<?php
include('dbconnect/dbconnect.php');
$sql="select * from hospital";
$res=mysqli_query($conn,$sql);
while($row=mysqli_fetch_array($res))
{
?>
<option value="<?php echo $row['hospital_id'];?>"><?php echo $row['hospital_name'];?></option>
<?php
}
?>
</select>
<input type="number" name="phone_no" id="phone_no" placeholder="phoneno" class="form-control required">
<input type="email" name="email" id="email" placeholder="Email" class="form-control required"/>
<input type="password" placeholder="Password" name="password" id="myInput4" class="form-control required"/>
<input type="password" placeholder="Retype Password” name="repassword" id='myInput3'/>
<button>Sign Up</button>
```

```html
</form>
</div>
<div class="form-container sign-in-container">
<form action="logcheck.php" method="post">
<h1>Sign in</h1>
<span>use your account</span>
<input type="Email" name="username" required placeholder="Email" />
<input type="password" name="password" required placeholder="Password"
id="myInput1" />
<input type="checkbox" onclick="myFunction ()"/>show password
<a href="#">Forgot your password? </a>
<button>Sign In</button>
</form>
</div>
<div class="overlay-container">
<div class="overlay">
<div class="overlay-panel overlay-left">
<h1>Welcome Back! </h1>
<p>To keep connected with us please login with your personal info</p>
<button class="ghost" id="signIn">Sign In</button>
</div>
<div class="overlay-panel overlay-right">
<h1>Hello, Friend! </h1>
<p>Enter your personal details and start journey with us</p>
<button class="ghost" id="signUp">Sign Up</button>
</div></div></div></div>
<footer>
<p>
```

```
<i class="fa fa-heart"></i>
<a target="_blank" href="https://florin-pop.com">save child<i class="fa fa-fw fa-child"></i></a>
-created by Kavya Hegde and Vinoda Patil
<a target="_blank" href="https://www.florin-pop.com/blog/2019/03/double-slider-sign-in-up-form/">here</a>.
</p>
</footer>
```

Insertion For Request Vaccine Table:

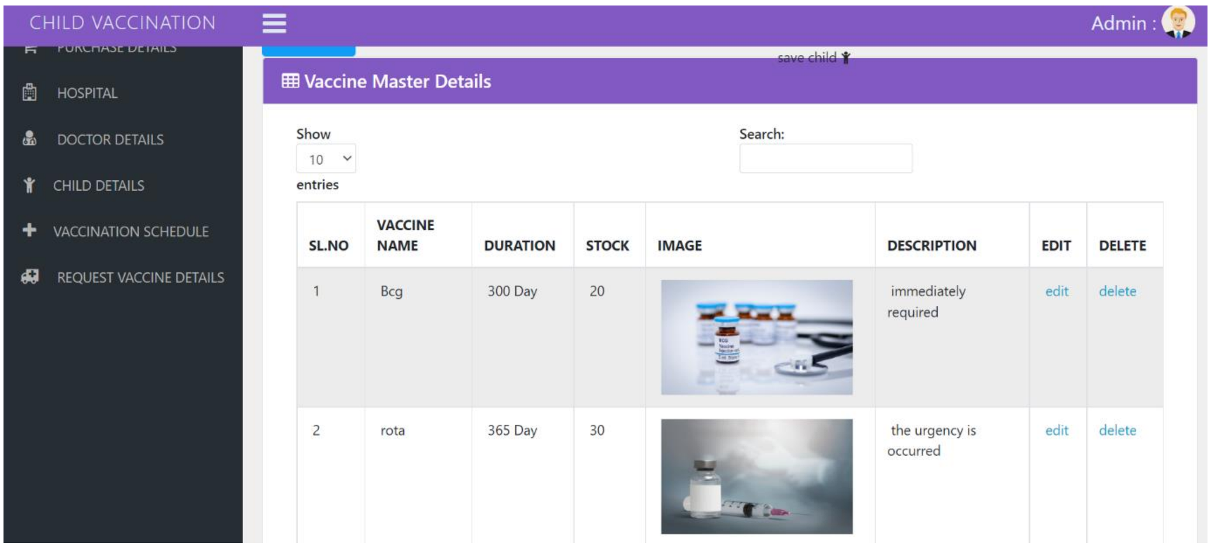


select Vaccine:	Covid
select Child:	kavya
Request Date:	27-01-2023
Request Status:	provide the vaccine

```
<?php
include('../dbconnect/dbconnect.php');
//$request_vaccine_id =$_POST ['request_vaccine_id’];
$v_id=$_POST['v_id'];
$c_id=$_POST['c_id'];
$req_date=$_POST['req_date'];
$req_status=$_POST['req_status'];
```

```
$sql="insert into request_for_vaccine values(null,'$v_id','$c_id','$req_date','$req_status')";
mysqli_query($conn,$sql);
?>
<script language="javascript1.2">
alert ("values are inserted");
document. Location="Request_vaccine_view.php";
</script>
```

View For Vaccine Master Table:

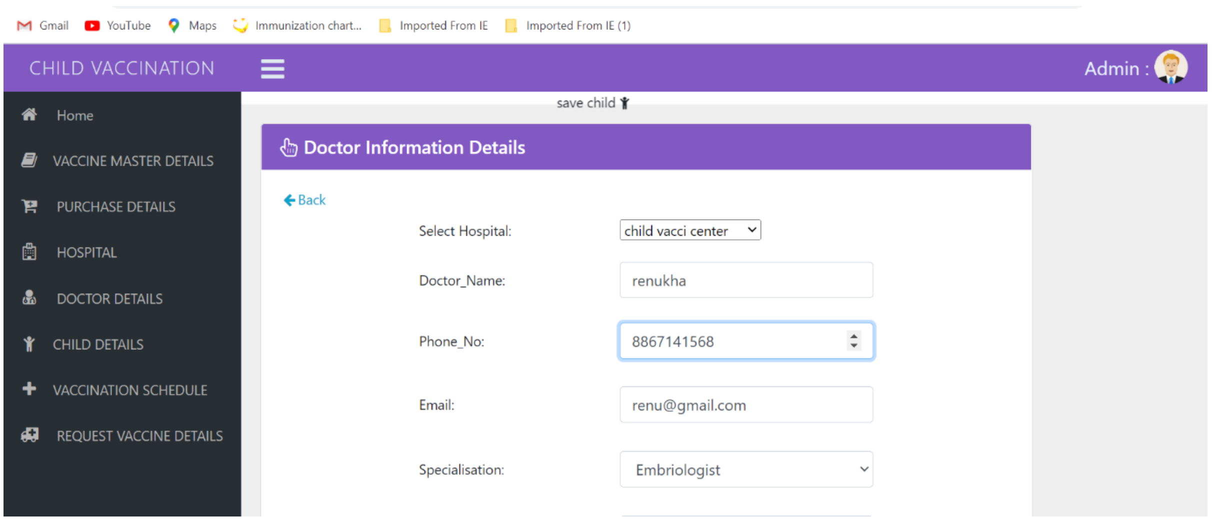


SL.NO	VACCINE NAME	DURATION	STOCK	IMAGE	DESCRIPTION	EDIT	DELETE
1	Bcg	300 Day	20		immediately required	edit	delete
2	rota	365 Day	30		the urgency is occurred	edit	delete

```
<!DOCTYPE html>
<html lang="en">
<?php include('metadata.php'); ?>
<body class="adminbody">
 <div id="main">
<! -- top bar navigation -->
<?php include('header.php'); ?>
 <! -- End Navigation -->
<?php include ('sidebar. Php');?>
```

```
  <! -- End Sidebar -->
  <div class="content-page">
   <! -- Start content -->
 <div class="content">
  <div class="container-fluid">
  <div class="row">
  <div class="col-xl-12">
  <div class="breadcrumb-holder">
   <h1 class="main-title float-left"> Vaccine Master Details</h1>
    <! -- <ol class="breadcrumb float-right">
    <li class="breadcrumb-item">Home</li>
    <li class="breadcrumb-item active">Data Tables</li>
    </ol> -->
    <div class="clearfix"></div>
     </div>
    </div>
     </div>
     <! -- end row -->
     <hr>
<a href="Vaccine_market.php" class="btn btn-primary">ADD NEW</a>
<div class="row">
 <div class="col-xs-12 col-sm-12 col-md-12 col-lg-12 col-xl-12">
 <div class="card mb-3">
  <div class="card-header">
 <h3><i class="fa fa-table"></i> Vaccine Master Details </h3>
   </div>
   <div class="card-body">
    <div class="table-responsive">
   <table id="example1" class="table table-bordered table-hover display">
    <thead>
      <tr>
    <th>SL.NO</th>
    <th>VACCINE NAME </th>
    <th>VACCINE TYPE </th>
    <th>STOCK</th>
    <th>IMAGE</th>
    <th>DESCRIPTION</th>
    <th>EDIT </th>
    <th>DELETE</th>
  </tr>
</thead>
  <tbody>
 <?php
include('../dbconnect/dbconnect.php');
$sl=1;
$sql="select * from vaccine_master ";
$res=mysqli_query($conn,$sql);
while($row=mysqli_fetch_array($res))
{
?>
  <tr>
 <td>&nbsp;<?php echo $sl++;?></td>
 <td>&nbsp;<?php echo $row['vname'];?></td>
 <td>&nbsp;<?php echo $row['vtype'];?></td>
 <td>&nbsp;<?php echo $row['stock'];?></td>
 <td>&nbsp;<?php echo $row['image'];?></td>
  <td>&nbsp;<?php echo $row['description'];?></td>
 <td>&nbsp;<a href="Vaccine_market_edit.php?v_id=<?php echo
$row['v_id'];?>">edit</a></td>
    <td><a href="Vaccine_market_delete.php?v_id=<?php echo
$row['v_id'];?>"onclick="return confirm('are u sure want to
delete')">delete</a></td>
  </tr>
  <?php
}
?>
</tbody>
</table>
</div>
 </div>
  </div><! -- end card-->
   </div>
   </div>
  <! -- END container-fluid -->
  </div>
  <! -- END content -->
 </div>
 <! -- END content-page -->
  <?php include('footer.php'); ?>
</div>
<! -- END main -->
<?php include('script.php'); ?>
</body>
</html>
```

Edit for Hospital Table:



Doctor Information Details

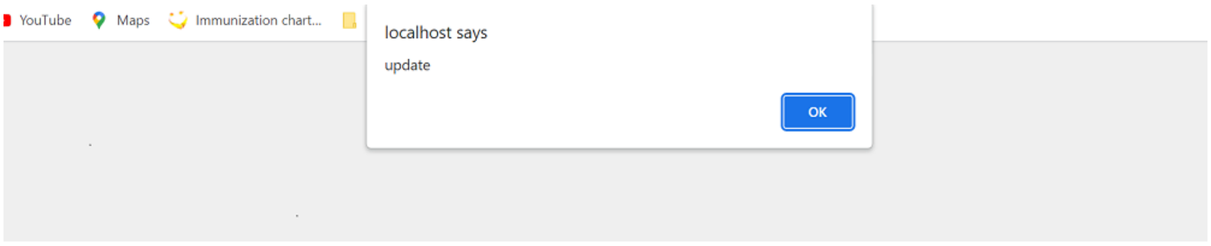
Select Hospital:	child vacci center
Doctor_Name:	renukha
Phone_No:	8867141568
Email:	renu@gmail.com
Specialisation:	Embriologist

```
<!DOCTYPE html>
<html lang="en">
<head>
  <?php include("metadata.php"); ?>
  <?php include('header.php'); ?>
  <?php include('sidebar.php'); ?>
</head>
<body class="adminbody">
  <! -- End Navigation -->
 <!-- Left Sidebar -->
 <!-- End Sidebar -->
<div class="content-page">
  <!-- Start content -->
  <div class="content">
 <div class="container-fluid">
  <div class="row">
  <div class="col-xl-12">
  <div class="breadcrumb-holder">
  <! -- <h1 class="main-title float-left">Form validation</h1>
  <ol class="breadcrumb float-right">
  <li class="breadcrumb-item">Home</li>
  <li class="breadcrumb-item active">Form validation</li>
   </ol> -->
  <div class="clearfix"></div>
   </div>
   </div>
    </div>
    <! -- end row -->
  <div class="row">
    <div class="col-xs-20 col-sm-20 col-md-10 col-lg-10 col-xl-10">
   <div class="card mb-3">
   <div class="card-header">
   <h3><i class="fa fa-hand-pointer-o"></i> Hospital Information  Details</h3>
    </div>
   <div class="card-body">
  <!--------------------------------------------->
<?php
include('val.php');
?>
<?php
include('../dbconnect/dbconnect.php');
$hospital_id =$_REQUEST['hospital_id'];
$sql="select * from hospital where hospital_id='$hospital_id'";
$res=mysqli_query($conn,$sql);
$row=mysqli_fetch_array($res);
?>
<a href="hospital_view.php" ><i class="fa fa-fw fa-arrow-left"></i>Back</a>
<form name="formID" id="formID" method="post" action="Hospital_update.php">
 <input type="hidden" value="<?php echo $row['hospital_id'];?>" name="hospital_id">
<table width="513" height="661" border="0" align="center">
 <tr  >
 <td colspan="2" align="center"> Hospital Register form </td>
  </tr>
  <tr>
 <td width="225">Hospital_name:</td>
 <td width="272"><input type="text" name="hospital_name" class="form-control
validate [required, custom[onlyLetter]]" value="<?php echo
$row['hospital_name'];?>"></td>
  </tr>
  <tr>
   <td>Address:</td>
 <td><textarea name="address" class="form-control validate[required]"
id="address"><?php echo $row['address'];?> </textarea></td>
```

```
    </tr>

    <tr>

    <td>Phone_No:</td>

   <td><input type="number" name="phone_no" class="form-control
validate[required,custom[mobile]]" value="<?php echo $row['phone_no'];?>"></td>

   </tr>

   <tr>

   <td>Email:</td>

  <td><input type="email" name="email" class="form-control
validate[required,custom[email]]" value="<?php echo $row['email'];?>"></td>

 </tr>

  <tr>

<td>Website:</td><td><input type="text"  name="website" class="form-control
validate[required,custom[onlyLetter]]" value="<?php echo
$row['website'];?>"></td>

  </tr>

  <tr>

   <td>Majorfacilities:</td>

  <td><select name="majorfacilities" id="majorfacilities" class="form-control
validate[required]">

  <option value="Day_care_services">Day_care_services</option>

  <option value="Clinical_laboratary">Clinical_laboratory</option>

   </select></td>

    </tr>

    <tr>

    <td>Emergency_No:</td>

    <td><input type="number" name="emergency_no" class="form-control
validate[required,custom[mobile]]" value="<?php echo
$row['emergency_no'];?>"></td>

    </tr>

    <tr>

    <td height="56" colspan="2" align="center">

   <input type="submit" name="Submit" value="Submit" class="btn btn-primary">

<input type="Reset" name="Reset" value="Reset" class="btn btn-danger"></td>
```

```
    </tr>
  </table>
  </form>
   </div>
   </div><! -- end card-->
   </div>
   </div>
   </div><!-- end card-->
    </div>
    </div>
     </div>
    <!-- END container-fluid -->
   </div>
    <! -- END content -->
     </div>
     <!-- END content-page -->
   <?php include('footer.php'); ?>
   </div>
<!-- END main -->
<?php include('script.php'); ?>
<?php include('val.php'); ?>
</body>
</html>
```

Update For Hospital Table:

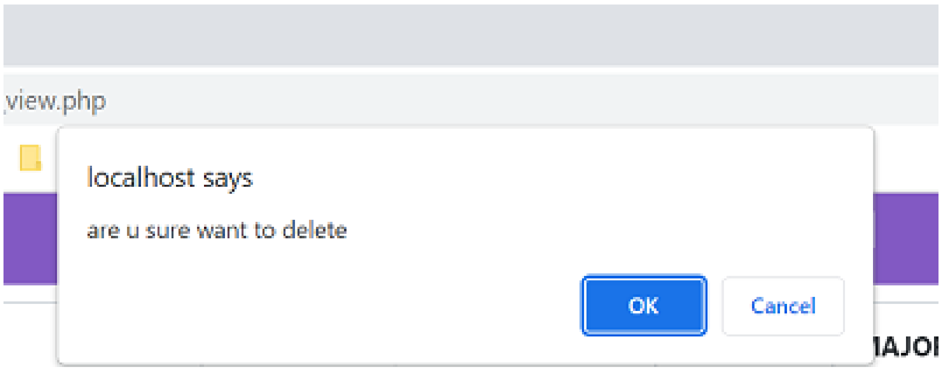


localhost says

update

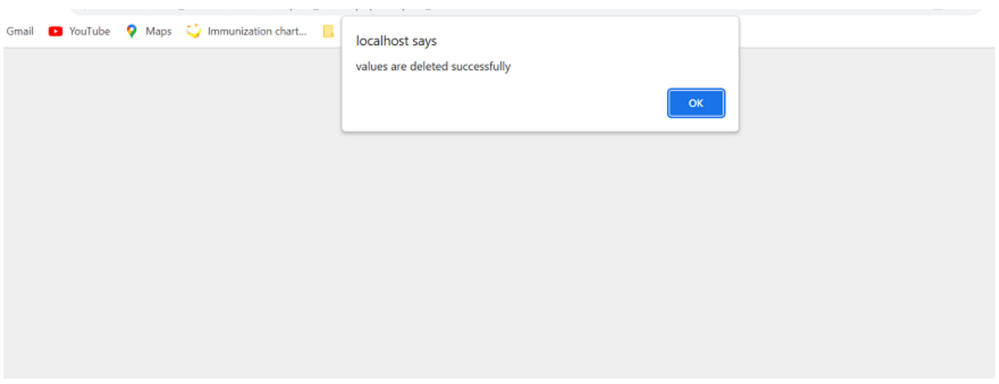
```
<?php
include('../dbconnect/dbconnect.php');
$hospital_id=$_POST['hospital_id'];
$hospital_name=$_POST['hospital_name'];
$address=$_POST['address'];
$phone_no=$_POST['phone_no'];
$email=$_POST['email'];
$website=$_POST['website'];
$majorfacilities=$_POST['majorfacilities'];
$emergency_no=$_POST['emergency_no'];
$sql="update hospital set
hospital_name='$hospital_name',address='$address',phone_no='$phone_no',email='$
email',website='$website',majorfacilities='$majorfacilities',emergency_no='$emerge
ncy_no' where hospital_id='$hospital_id'";
mysqli_query($conn,$sql);
?>
<script language="javascript1.2">
alert('update');
document. Location="hospital_view.php";
</script>
```

Delete For Hospital Table:



localhost says

are u sure want to delete



localhost says

values are deleted successfully

```
<?php
include('../dbconnect/dbconnect.php');
$hospital_id=$_REQUEST['hospital_id'];
$sql="delete from hospital where hospital_id=$hospital_id";
mysqli_query ($conn, $sql);
 ?>
<script>
alert ("values are deleted successfully");
document.location="hospital_view.php";
</script>
```

SYSTEM TESTING AND RESULTING

INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. Testing is done to see if the entire feature provided by the module are performing satisfactorily and to ensure that the process of testing is as realistic as possible. Testing part is done to major phases via unit level and module level testing. The idea of testing the software in a phased manner is to identify and isolate the bugs for any easy correction. Both phases are over and results meet with the user needs.

Test Approaches:

Black box Testing:

Black box testing is done to find:

- Incorrect or missing functions.
- Interface Errors
- Errors in external database access
- Performance error.
- Initialization and termination error.

White box Testing:

White box testing is done to find out:

- Check whether all independent paths within a module have been exercised at least once
- Exercise all logical decision on their true and false sides
- Execute all loops at their boundaries and within their bounds.
- Exercise the internal data structure to ensure their validity.
- Ensure whether all the possible validity checks and validity lookups have been provided to validate data entry.

Testing Strategies:

Levels of testing:

$$\begin{tikzcd}
\text{Client Needs} \arrow[r, leftrightarrow] \arrow[d] & \text{Acceptance Testing} \arrow[d] \\
\text{Requirements} \arrow[r, leftrightarrow] \arrow[d] & \text{System Testing} \arrow[d] \\
\text{Design} \arrow[r, leftrightarrow] \arrow[d] & \text{Integration Testing} \arrow[d] \\
\text{Coding} \arrow[r, leftrightarrow] & \text{Unit Testing}
\end{tikzcd}$$

The different testing is carried out on the reflects the effectiveness and Efficiency of different phases of software development these test help to uncover the error in the corresponding phase.

- **Unit testing** is carried out to check the coding errors and program logic.
- **Integration testing** which also known as module testing uncovers the system design errors.
- **Performance testing** is performed to determine how fast some aspect of a system performs under a particular workload.
- **User Acceptance testing** is done to test whether the product developed needs the client needs and acceptable to him.

Unit Testing:

Unit testing involves, checking all the modules in the system individually against the specification produced during the design of the module and for their performance. Unit testing also involves code produced in the coding phase and hence the internal logic of the program. Each module is tested for different test cases design to check each specific combination of conditions handled by the program. Error handlers are included in each module for each event trap and handled the errors.

Unit testing is done by inputting proper input s in each page and checking whether the data is in correct format as used to backend. Each Http Request is also checked whether each method is working properly to fulfill the requirement.

Integration testing:

Integration testing takes as its input modules that have been checked out by unit testing, groups them in larger aggregates, applies tests defined in an Integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items.

System Testing:

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called *assemblages*) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

Security Testing:

This attempts to verify whether the protection mechanism is built into the system. In this testing the authentication of the users is checked and authorized users are allowed to access the database.

Performance Testing:

Performance testing can serve different purpose. It can demonstrate that the system meets the performance criteria. It can compare two systems to find which performs better, or it can measure what parts of the system or workload cause the system to perform badly. In the diagnostic case, software engineers use tools such as profilers to measure what parts of a device or Software contributes most to the poor performance.

Acceptance testing

Acceptance testing can mean one of two things:

1. A smoke test is used as an acceptance test prior to introducing a new build to the main testing process, i.e. before integration or regression.
2. Acceptance testing performed by the customer, often in their lab environment on their own hardware, is known as user acceptance testing (UAT). Acceptance testing may be performed as part of the hand-off process between any two phases of development.

Test Cases and Results:

Test case:

Test Case	Input	Expected O/P	Actual O/P	Result
1	Valid Username and Password	It should display respective page according to user type.	Respective Home is displayed .	Passed
2	Invalid Username and Password	It should give appropriate error message saying "Enter proper User-n and Password"	Error message Displayed	Passed
3	Add/Update/ delete Member details	Add/Update/delete action is taken.	Add/Update/delete Member done successfully	Passed
4	User enters valid Username and password.	Respective Home is displayed	Respective Home is displayed .	Passed

CONCLUSION:

Software is said to have attained its objective only when it meets all the requirements of the user, further the user himself is the person to judge the success of the system.

Every attempt has been made to ensure that the system is fully functional and works effectively and efficiently. The system has been tested with simple data to cover all possible options and checked for all outputs. Since the system is flexible and modular, further modifications of this package can be easily incorporated.

Future Enhancement of The Project:

- In future Android application can be developed.
- Home schedule of vaccine can be added.
- It provides the convenient way of communication to parents through chatbots.
- Multiple hospital option can be added.

BIBLIOGRAPHY:

References:

1. Fundamentals of database system-B. Navathe

2. Ian Summerville “Software Engineering” Pearson Education Ltd 6th edition2004

3. Ali Bahrami Object Oriented Systems Development, McGraw hill,1999

4. Elias M Awad, System Analysis and Design, Golgotia 1995

5. HTML How to Program by Rohit Khurana.

Websites:

1. http://w3schools.com
2. www.tutorialspoint.com
3. https://learn.coderslang.com/0022-how-to-compare-strings-in-javascript/

4. https://www.w3schools.com/php/php_ajax_database.asp
5. https://www.programiz.com/javascript/examples/string-comparison
6. https://www.w3schools.com/jsref/jsref_length_string.asp

7. https://developer.chrome.com/docs/devtools/console/javascript/

8. https://code.visualstudio.com/

9. https://www.tutorialspoint.com/what-is-md5-in-information-security#:~:text=MD5%20stands%20for%20message%2Ddigest,that%20it%20is%20sent%20to.