

COVID PREVENTOCITY

Project Report Submitted By

SREYA M

Reg. No.: AJC20MCA-2069

In Partial fulfillment for the Award of the Degree of

**MASTER OF COMPUTER APPLICATIONS(2 Year)
(MCA)**

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY



**AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2021-2022

DEPARTMENT OF COMPUTER APPLICATIONS
AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY



CERTIFICATE

This is to certify that the Project report, “**COVID PREVENTOCITY**” is the bonafide work of **SREYA M (Reg.No:AJC20MCA-2069)** in partial fulfillment of the requirements for the award of the Degree of Regular Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2021-22.

Mr.T.J Jobin
Internal Guide

Ms. Grace Joseph
Coordinator

Rev.Fr.Dr.Rubin Thottupurathu Jose
Head of the Department

External Examiner

DECLARATION

I hereby declare that the project report “**COVID PREVENTOCITY**” is a bonafided work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications (2 Year) from APJ Abdul Kalam Technological University, during the academic year 2021- 2022.

Date: 22/02/2022

SREYA M

KANJIRAPPALLY

Reg. No: AJC20MCA-2069

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal Dr. Lillykutty Jacob for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping me. I extend my whole hearted thanks to the project coordinator **Ms. Grace Joseph** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide, **Mr. T. J. Jobin** for his inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

SREYA M

ABSTRACT

Coronavirus disease (COVID-19) is an infectious disease caused by the SARS-CoV-2 virus. Most people who fall sick with COVID-19 will experience mild to moderate symptoms and recover without special treatment. However, some will become seriously ill and require medical attention. Covid Preventocity is an online web application for managing covid-19 pandemic and its related works in Panchayath level. In this system there are mainly Five modules: Admin, Panchayath, Hospital, Vaccine and User . Admin can view and manage the panchayath , hospital, vaccine and user. Admin will add, update and disable all this . Hospitals can provide details about the covid positives patients along with their details. The panchayath can able to view the details of the positives and add ward belongs to the patients into containment zone. Panchayath will able to know the positives and their list of contacts. Panchayath can view the list of death in a ward due to the covid pandemic. Asha workers are added by the panchayath and Asha workers will update the status of covid patients and their contacts. Panchayath can able to add the available vaccines in panchayath and also add new vaccines in panchayath .User can view the containment zone in particular panchayath. User can register for vaccine and download vaccine certificate. Covid test registration , view covid 19 test result and download test result through this site.Covid test result is mailed to user by hospital. Also user can take appointment for physician or for counselling.

CONTENT

Sl. No	Topic	Page No
1	INTRODUCTION	1
1.1	PROJECT OVERVIEW	2
1.2	PROJECT SPECIFICATION	2
2	SYSTEM STUDY	3
2.1	INTRODUCTION	4
2.2	EXISTING SYSTEM	5
2.3	DRAWBACKS OF EXISTING SYSTEM	5
2.4	PROPOSED SYSTEM	5
2.5	ADVANTAGES OF PROPOSED SYSTEM	6
3	REQUIREMENT ANALYSIS	7
3.1	FEASIBILITY STUDY	8
3.1.1	ECONOMICAL FEASIBILITY	8
3.1.2	TECHNICAL FEASIBILITY	9
3.1.3	BEHAVIORAL FEASIBILITY	9
3.2	SYSTEM SPECIFICATION	10
3.2.1	HARDWARE SPECIFICATION	10
3.2.2	SOFTWARE SPECIFICATION	10
3.3	SOFTWARE DESCRIPTION	10
3.3.1	PHP	10
3.3.2	MYSQL	11
4	SYSTEM DESIGN	13
4.1	INTRODUCTION	14
4.2	UML DIAGRAM	14
4.2.1	USE CASE DIAGRAM	15
4.2.2	SEQUENCE DIAGRAM	18
4.2.3	STATE CHART DIAGRAM	20
4.2.4	ACTIVITY DIAGRAM	21
4.2.5	CLASS DIAGRAM	22
4.2.6	OBJECT DIAGRAM	23
4.2.7	COMPONENT DIAGRAM	24
4.2.8	DEPLOYMENT DIAGRAM	24

4.5	USER INTERFACE DESIGN	25
4.6	DATA BASE DESIGN	28
5	SYSTEM TESTING	36
5.1	INTRODUCTION	37
5.2	TEST PLAN	38
5.2.1	UNIT TESTING	38
5.2.2	INTEGRATION TESTING	39
5.2.3	VALIDATION TESTING	39
5.2.4	USER ACCEPTANCE TASTING	40
6	IMPLEMENTATION	41
6.1	INTRODUCTION	42
6.2	IMPLEMENTATION PROCEDURE	42
6.2.1	USER TRAINING	43
6.2.2	TRAINING ON APPLICATION SOFTWARE	43
6.2.3	SYSTEM MAINTENANCE	43
7	CONCLUSION & FUTURE SCOPE	44
7.1	CONCLUSION	45
7.2	FUTURE SCOPE	45
8	BIBLIOGRAPHY	46
9	APPENDIX	49
9.1	SAMPLE CODE	50
9.2	SCREEN SHOTS	61

List of Abbreviation

IDE	-	Integrated Development Environment
HTML	-	Hyper Text Markup Language.
CSS	-	Cascading Style Sheet
SQL	-	Structured Query Language
UML	-	Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1.PROJECT OVERVIEW

Covid Preventocity system is to manage the work related to the covid pandemic. The user can covid vaccine ,covid test and download certificate or results through this. User also can able to view containment zone and number of positives. Panchayath can view containment zone and details of containment zone and also vaccinated details in panchayath from different hospital. Hospital will send the covid positive details from covid testing and vaccinated details.

1.2.PROJECT SPECIFICATION

The proposed system is a website in which user can view the containment zone, vaccine availability in hospitals , book for vaccine and download vaccine certificate . User can take covid test and download result. Positives will upload to panchayath by hospitals. Panchayath add ward as containment zone and view details. Panchayath can view vaccinated details and allot vaccines for hospitals.

The system includes 4 modules. They are:

Admin Module

Admin must have a login into this system. He has the overall control of the system. Admin can add or update panchayath details. Admin can View all the registered users and also manage all his data. Admin can view registered hospitals and details.

Panchayath Module

Admin will register panchayath and panchayath will manage the containment zones, view the positives and manage vaccine. Panchayath can view the details of containment zones. Panchayath can view vaccines and allot vaccines to hospitals. Panchayath can view positives from different hospitals. Asha worker will add the contacts of the positives. Panchayath can view vaccinated status and covid tests in graphically and report of containment zone and positives are generated.

Hospital Module

Hospital will register themselves and they are get verified. Hospital can view and schedule vaccines. They can request for vaccines to panchayath. They will result update covid test result and test result is mailed to the user. Positive result is updated to panchayath. Vaccinated status and details is also updated to panchayath. Hospital also schedule for doctor appointment.

User Module

User will register themselves and user can view containment zone and details. User can view available vaccines in different hospitals and book for vaccine. They can also download the vaccine certificate. The covid test registration and its result download can obtain through this application. They can make online payment. User can view and take appointment for counselling or physicians.

CHAPTER 2

SYSTEM STUDY

2.1. INTRODUCTION

Information gathering, analysis, problem-solving, and recommendation-making are all steps in the process of system analysis. It's a process of problem-solving that demands regular communication between system designers and users. System analysis or study is a vital element in every system development process. In-depth analysis and exploration of the system are performed. The system analyst examines the functioning of the current system in great detail. The inputs to the system are noted, and the system is viewed as a whole. The outputs of the organization can be connected to the numerous processes. The goal of system study is to define the problem, find the relevant and crucial factors, evaluate and combine the many components, and choose the best or, at the very least, most acceptable course of action.

It is necessary to thoroughly examine the process using a variety of methods, including questionnaires and interviews. To reach a conclusion, it is necessary to carefully examine the data gathered by various sources. An understanding of the system's operation serves as the conclusion. What we refer to is the existing system. Currently, problem areas have been identified following a thorough analysis of the present system. The designer now assumes a problem-solving role and works to find solutions to the problems the company is currently facing. The solutions are replaced with proposals. The best option is then selected after an analytical comparison of the proposal and the existing system. The proposal is presented to the user with the option to accept or reject it. On user request, the proposal is assessed and appropriate revisions are made. As soon as the user is content with the suggestion, this loop breaks.

In order to do additional research on the system, preliminary investigation involves acquiring and interpreting data. A problem-solving process known as preliminary research necessitates close coordination between system users and developers. Many feasibility studies are conducted by it. From these studies, it is possible to determine an approximate estimate of the system activities, from which it is possible to make a decision regarding the methods to use for an efficient system study and analysis.

2.2 EXISTING SYSTEM

Existing system is not an automated system. Panchayath will be able to know the containment and positives in manually. Covid test registration is done manually. It includes lots of paper work.

It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure. Using the new system panchayath can manage ward and containment zone. Panchayath can manage vaccines. User can book for vaccine and download certificate. Covid test result can also be downloaded from this system

2.3 DRAWBACKS OF EXISTING SYSTEM

- 2.3.1 No proper online management of system
- 2.3.2 Human effort is needed.
- 2.3.3 It is difficult to maintain important information in papers.
- 2.3.4 More manual hours need to generate required reports.

2.4 PROPOSED SYSTEM

The proposed system is defined to meet all the disadvantages of the existing system. It is necessary to have a system that is more user friendly and user attractive; it will help our authorities to manage the pandemic in an easy way. In our proposed system, there is an admin who can view panchayath, hospitals and users. Users of this proposed system are admin, Panchayath, Hospitals, Asha worker and User. It allows panchayath to know the containment zone, positives from hospitals and vaccines availability and vaccinated status in the panchayath. User can book vaccine and download certificate, also download covid test result. User can view the containment zone and details. Hospital can schedule vaccine and upload vaccine status. Upload the covid test result and mail it. Hospital also upload positives details to panchayath. Asha worker will upload the contacts of positives. Panchayath can view the vaccinated and covid test status in graphically. Report of containment zone and positive details are generated. The software application which avoids more manual hours that need to spend in record keeping and generating reports.

This application keeps the data in a centralized way which is available to all the users simultaneously. It is very easy to manage historical data in database. No specific training is required for the distributors to use this application. It is very easy to record the information of covid cases and containment zones in the databases.

2.5 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources, and the system will work in almost all configurations. It has got following features:

- Reports of the containment zone can be easily viewed and downloaded by Panchayath.
- User can easily download the covid test report and covid vaccine certificate.
- A graphic representation of the rate of Covid vaccination and test taking in a panchayath is available.
- The user will receive the results of the COVID test through email from the hospitals.
- The system is user friendly.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

An assessment of the project's viability is made to see if, given the amount of time, energy, and resources put in it, it will ultimately achieve the goals of the organization. With the help of a feasibility study, the project's creator can forecast the project's usefulness and prospective future. An assessment of a system proposal's viability, which takes into account its influence on the organization, capacity to satisfy user needs, and efficient use of resources, forms the basis of a feasibility study. Therefore, before a new application is authorized for development, a feasibility review is typically carried out.

The document outlines the project's viability and contains a number of factors that were carefully taken into account throughout this project's feasibility study, including its technical, economic, and operational viabilities. It has the following characteristics: -

3.1.1 Economical Feasibility

Cost and benefit analyses are required to support the emerging system. criteria to make sure that focus is placed on the project that will yield the best results the earliest. The price that would be involved in developing a new system is one of the variables.

The following are a few of the significant fiscal queries raised during the initial investigation:

- The expense of conducting a thorough system examination
- The price of the hardware and software.
- The advantages in the form of lower costs or fewer expensive mistakes.

The proposed system is being built as part of a project work; there are no manual costs associated with it. Additionally, the fact that all the resources are currently at hand indicates that the system is commercially viable for development.

The Covid Preventocity project's costs were broken down into three categories: system costs, development costs, and hosting costs. The project was developed at a cheap cost, based on all the estimates. Because open source software was used exclusively during development.

3.1.2 Technical Feasibility

Initially, the system needs to be assessed from a technological standpoint. Based on an outline design of the system requirement in terms of input, output, programs, and procedures, the assessment of this feasibility must be made. After identifying a general system, the study must next recommend the kind of equipment, necessary procedures for constructing the system, and ways to operate the system once it has been designed. Technical issues raised during the investigation are:

- Is the suggested technology compatible with the current one?
- Can the system improve if it is developed?

The project should be designed in such a way that the required performance and functionality are met while still adhering to the limits. The system may still be used even though the technology may become outdated after a while because a newer version of the same software still works with an earlier version. Therefore, this project only has a few limitations. The system was created using PHP for the front end and a MySQL server for the back end; it is technically feasible to complete the project. The system was created using PHP for the front end and a MySQL server for the back end; it is technically feasible to complete the project. Additionally, the system's Intel i3 core processor, 8GB of RAM, and 1TB of hard drive provided good performance.

3.1.3 Behavioral Feasibility

The following inquiries are part of the suggested system:

- Is there sufficient support for users?
- Will anyone be harmed by the proposed system?

The project would be advantageous because, when created and implemented, it would achieve the goals. The project is deemed to be behaviorally feasible after carefully weighing all behavioral factors.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - Intel core i3

RAM - 8 GB

Hard disk - 1 TB

3.2.2 Software Specification

Front End - HTML, CSS

Backend - MYSQL

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, AJAX, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

In addition to being used as a general-purpose programming language, PHP is a server-side scripting language created for web development. Currently, PHP is present on 2.1 million web servers and more than 244 million websites. The reference implementation of PHP, which was first developed by Rasmus Lerdorf in 1995, is now created by the PHP group. The recursive acronym PHP:HypertextPreprocessor has replaced the original meaning of PHP, which was personal Home page. A PHP processor module on a web server interprets PHP code to produce the final web page. Instead of contacting an external file to handle data, PHP commands can be directly included into an HTML source document. The GNU General Public License (GPL) is incompatible with it due to restrictions on the use of the word PHP, and it has grown to incorporate a command-line interface capability. For free, PHP may be installed on the majority of web servers as well as a standalone shell on practically all platforms and operating systems.

3.3.2 MySQL

Oracle Corporation created, distributed, and provided support for MySQL, the most well-known Open Source SQL database management system. The most recent details regarding MySQL software are available on the MySQL website.

- **MySQL is a database management system.**

A systematic collection of data is called a database. The huge volumes of information in a business network or a simple shopping list might all be examples. A database management system, like MySQL, is required to add, access, and process data kept in a computer database. Database management systems are essential to computing, whether they are used as stand-alone tools or as components of other programs, because computers are excellent at processing vast volumes of data.

- **MySQL databases are relational.**

In a relational database, the data is kept in separate tables rather than being gathered into one enormous warehouse. The database structures are stored in physical files with high performance. Databases, tables, views, rows, and columns are just a few examples of the components that make up the logical model's adaptable programming environment. One-to-one, one-to-many, unique, mandatory or optional, and "pointers" between different tables are just a few examples of the rules you may create to control the relationships between different data fields. Since the database upholds these principles, it is unlikely that your application will ever come across inconsistent, duplicate, orphan, out-of-date, or missing data. Structured Query Language (SQL) is what MySQL stands for. The most popular standard language for accessing databases is SQL. You might either enter SQL directly (for example, to generate reports), combine SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax, depending on your programming environment. The SQL language is defined by the ANSI/ISO SQL Standard. The SQL standard, which has been evolving since 1986, includes many variations. The terms "SQL92," "SQL: 1999," and "SQL: 2003" are used in this document to refer to the 1992, 1999, and 2003 editions of the standard, respectively. The most recent version of the SQL Standard is always meant when we refer to "the SQL standard."

- **MySQL software is Open Source.**

Anybody can use and alter software that is open source, according to the term. The MySQL software is accessible online and can be used by anyone for free. The source code is available for you to examine and modify as you see fit. The GPL (GNU General Public License), which governs how the MySQL software is used, specifies what you are allowed to do and are not allowed to do with it in various circumstances. If you dislike the GPL or require MySQL code to be used in a for-profit application, you can buy a commercially licenced version from us. The MySQL Licensing Overview can provide more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

You ought to give it a shot if that's what you're searching for. With minimal to no maintenance, MySQL Server may function smoothly on a laptop or desktop with your other software, web servers, and other devices. You can change the settings to utilise all the memory, CPU power, and I/O capacity if you dedicate an entire machine to MySQL.

- **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that includes a multi-threaded SQL server that supports several backends, a number of different client programs and libraries, administration tools, and a broad variety of application programming interfaces (APIs). We also offer MySQL Server as an integrated multi-threaded library that you can link into your program to get a smaller, faster, easier-to-manage standalone product.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

For each technical system or product, design is the initial step in the development phase. Design involves creativity. Optimal system design is essential for success. The term "design" is defined as "the act of using multiple techniques and concepts" in order to sufficiently describe a process or a system to permit its physical implementation. It may be described as the process of using a number of various approaches and concepts to fully describe a device, method, or system so that it can be physically implemented. Independent of the development paradigm utilized, software design forms the technical foundation of the software engineering process. The system design generates the architectural detail required to build a system or product. As with any systematic technique, this software underwent the best design phase possible, fine-tuning all efficiency, performance, and accuracy levels. During the design stage, a user-oriented document is transformed into a document for programmers or database employees. The two stages of system design development are logical design and physical design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

Unified Modeling Language is known as UML. Compared to other popular programming languages like C++, Java, COBOL, etc., UML is unique. A visual language called UML is used to create software blueprints. A general-purpose visual modelling language for software system visualisation, specification, construction, and documentation is what UML is known as. UML is not just used to represent software systems, despite the fact that this is its most common application. It is also used to model systems that are not software-based. For instance, the manufacturing facility's process flow, etc. Although UML is not a programming language, tools can be used to generate code using UML diagrams in a variety of languages. The analysis and design of objects-oriented systems are directly related to UML. UML has been standardised to the point where it is now an

OMG standard. A comprehensive UML diagram that depicts a system is made up of all the elements and relationships. The most crucial aspect of the entire procedure is the UML diagram's aesthetic impact. It is completed by using all the additional components. These eight diagrams are included in UML.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Activity diagram
- Statechart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a visual representation of how a system's components interact. Use cases are an approach in system analysis that help to find, explain, and arrange system needs. The term "system" in this context refers to something being created or run, such as a website for mail-order product sales and services. In UML (Unified Representing Language), a common notation for modelling actual items and systems, use case diagrams are used.

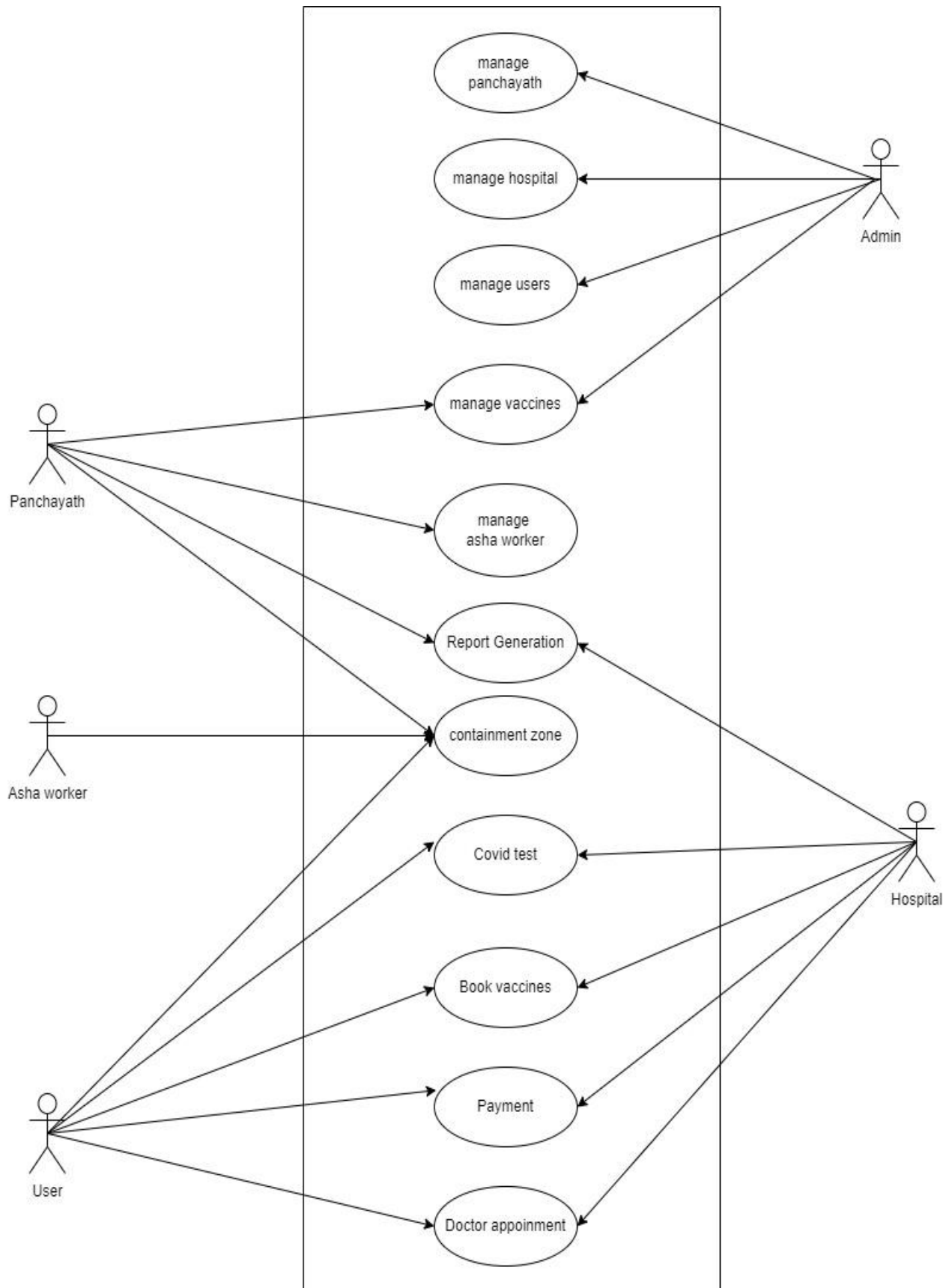
The planning of general requirements, the validation of a hardware design, the testing and debugging of a software product in development, the creation of an online help reference, or the completion of a job focused on customer support are all examples of system objectives. For instance, use cases in a product sales context would involve item ordering, catalogue updating, payment processing, and customer interactions. There are four parts to a use case diagram.

- The boundary, which isolates the system of interest from its surroundings.
- The performers, who are typically system participants identified by the roles they play.

- The actors within and around the system play the roles specified by the use cases.
- The connections and interactions between the actors and use cases.

Use case diagrams are created to depict a system's functional requirements. To create an effective use case diagram after identifying the aforementioned things, we must adhere to the following rules.

- A use case's naming is very significant.
- Identify actors with an appropriate name.
- Clearly depict links and dependencies in the diagram.
- Keep in mind that the diagram's primary function is to indicate the needs; do not attempt to include all possible relationships.
- When necessary, take notes to help you remember some crucial details.



4.2.2 SEQUENCE DIAGRAM

A sequence diagram only represents the sequential order in which objects interact with one another, or the order in which these interactions occur. The terms event diagrams and event scenarios can also be used to describe a sequence diagram. Sequence diagrams show the actions that the components of a system take. To define and comprehend requirements for both new and current systems, businesspeople and software developers frequently use these diagrams.

Sequence Diagram Notations –

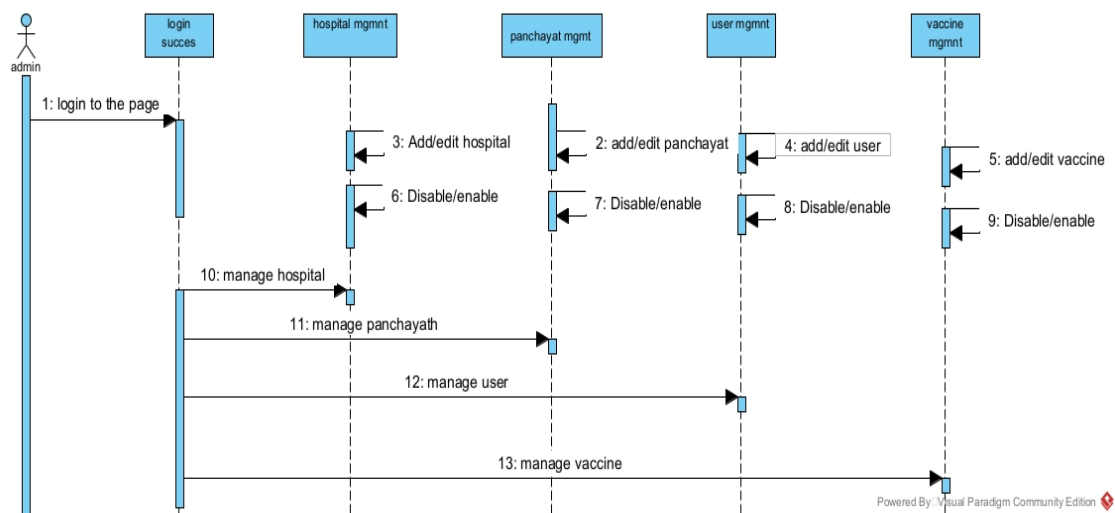
- i. **Actors** – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.
- ii. **Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram
- iii. **Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.
Messages can be broadly classified into the following categories:
 - **Synchronous messages**
 - **Asynchronous Messages**
 - **Create message**
 - **Delete Message**
 - **Self-Message**

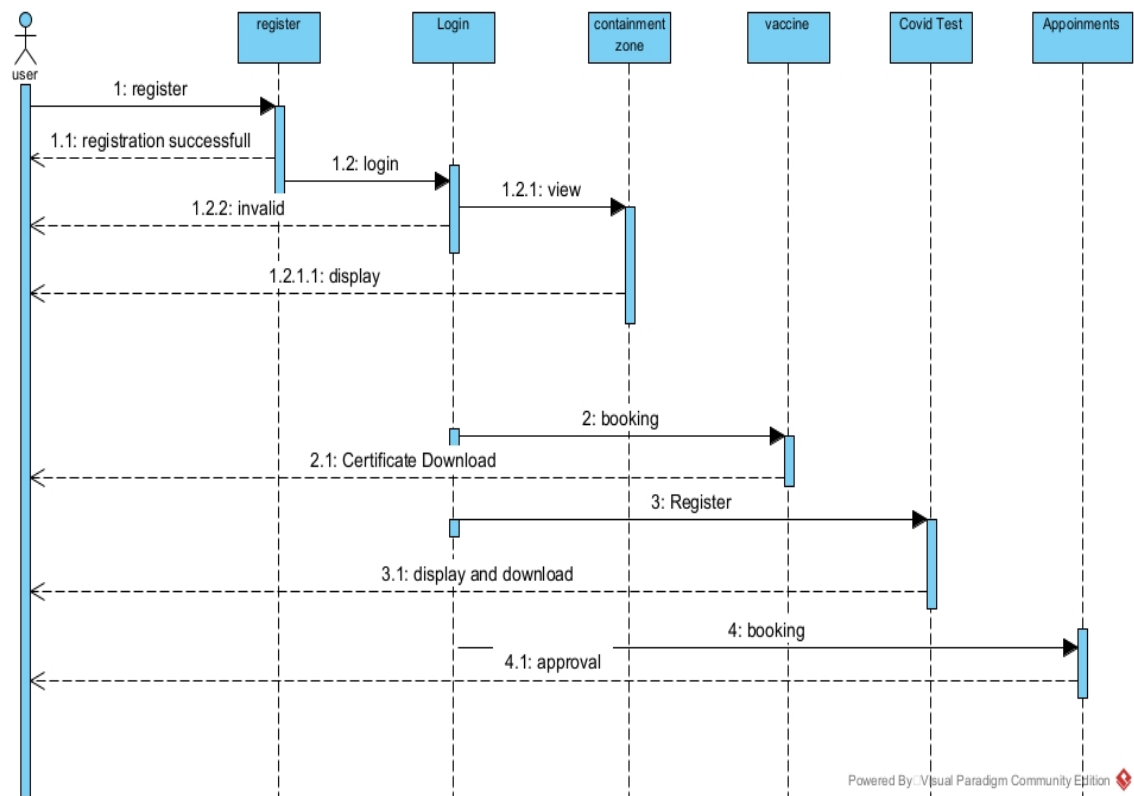
- **Reply Message**
- **Found Message**
- **Lost Message**

iv. Guards – In the UML, we utilize guards to model circumstances. When we need to limit the flow of messages under the guise of a condition being met, we use them. Software engineers rely on guards to inform them of the restrictions imposed by a system or specific process.

Uses of sequence diagrams –

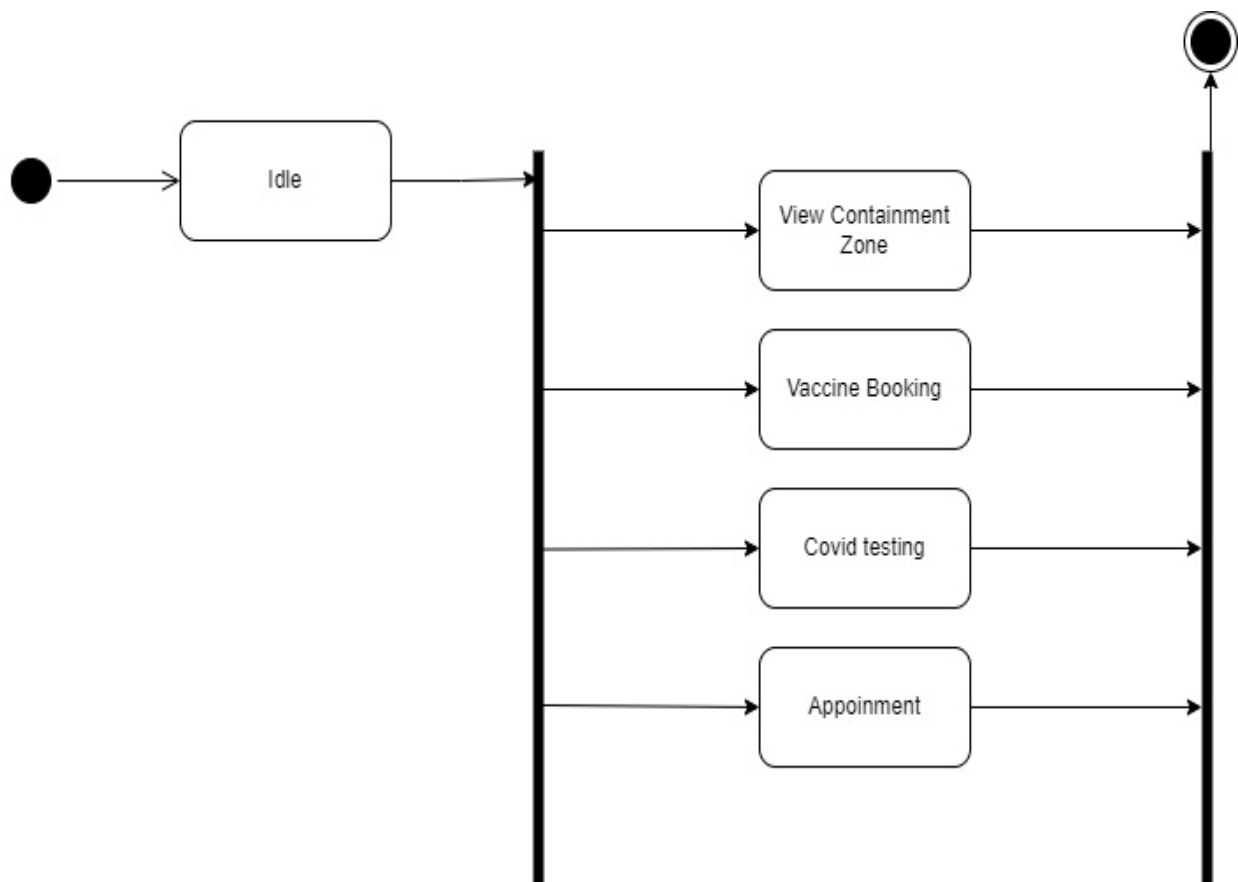
- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualize how messages and tasks move between objects or components in a system.





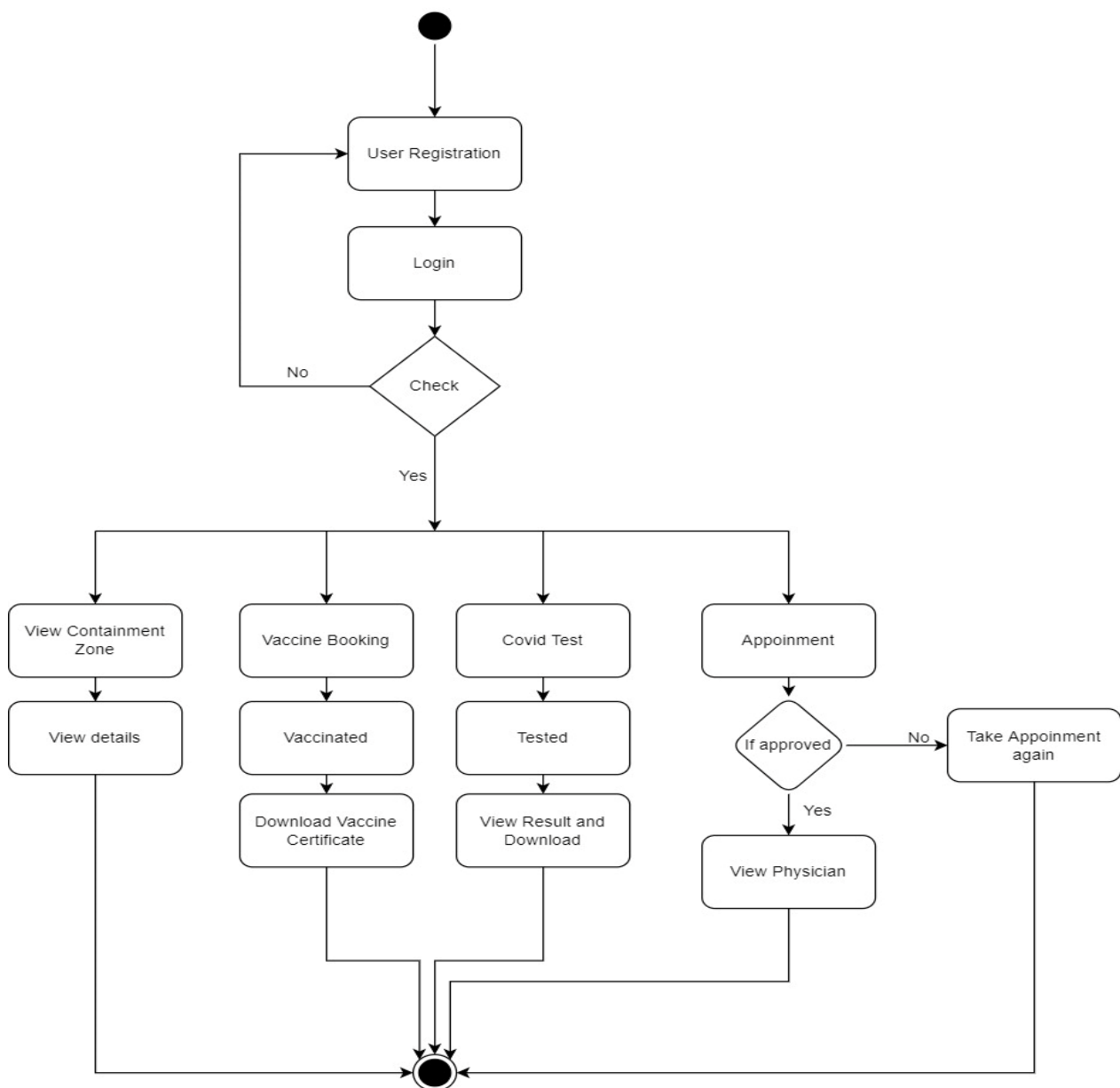
4.2.3 STATE DIAGRAM

A software system's behaviour is depicted using state chart diagrams. A class, a subsystem, a package, or even a complete system's behaviour can be modelled using UML state machine diagrams. The phrase "state chart" or "state transition graphic" may also be used. The interactions or communication that take place between external entities and a system can be efficiently modelled using statechart diagrams. To model the event-based system, use these diagrams. With the use of an event, one can control an object's state. Diagrams of state charts are used in applications to describe different states of an item.



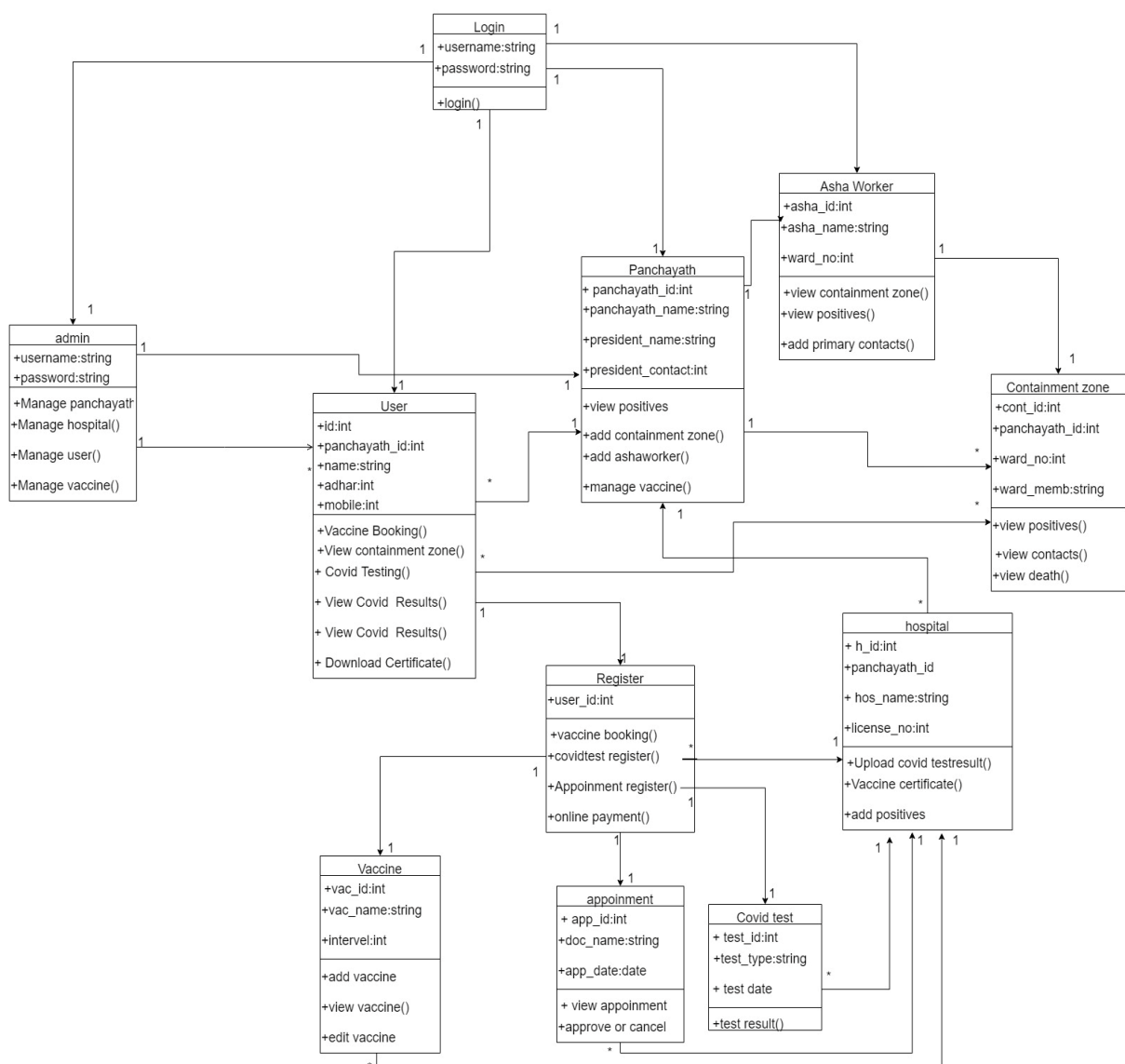
4.2.4 ACTIVITY DIAGRAM

Activity diagrams show how multiple levels of abstraction of activities are coordinated to produce a service. Typically, an event must be accomplished by some operations, especially when the operation is meant to accomplish several different things that call for coordination. Another common requirement is how the events in a single use case relate to one another, especially in use cases where activities may overlap and require coordination. It can also be used to illustrate how a set of related use cases interact together to reflect business workflows.



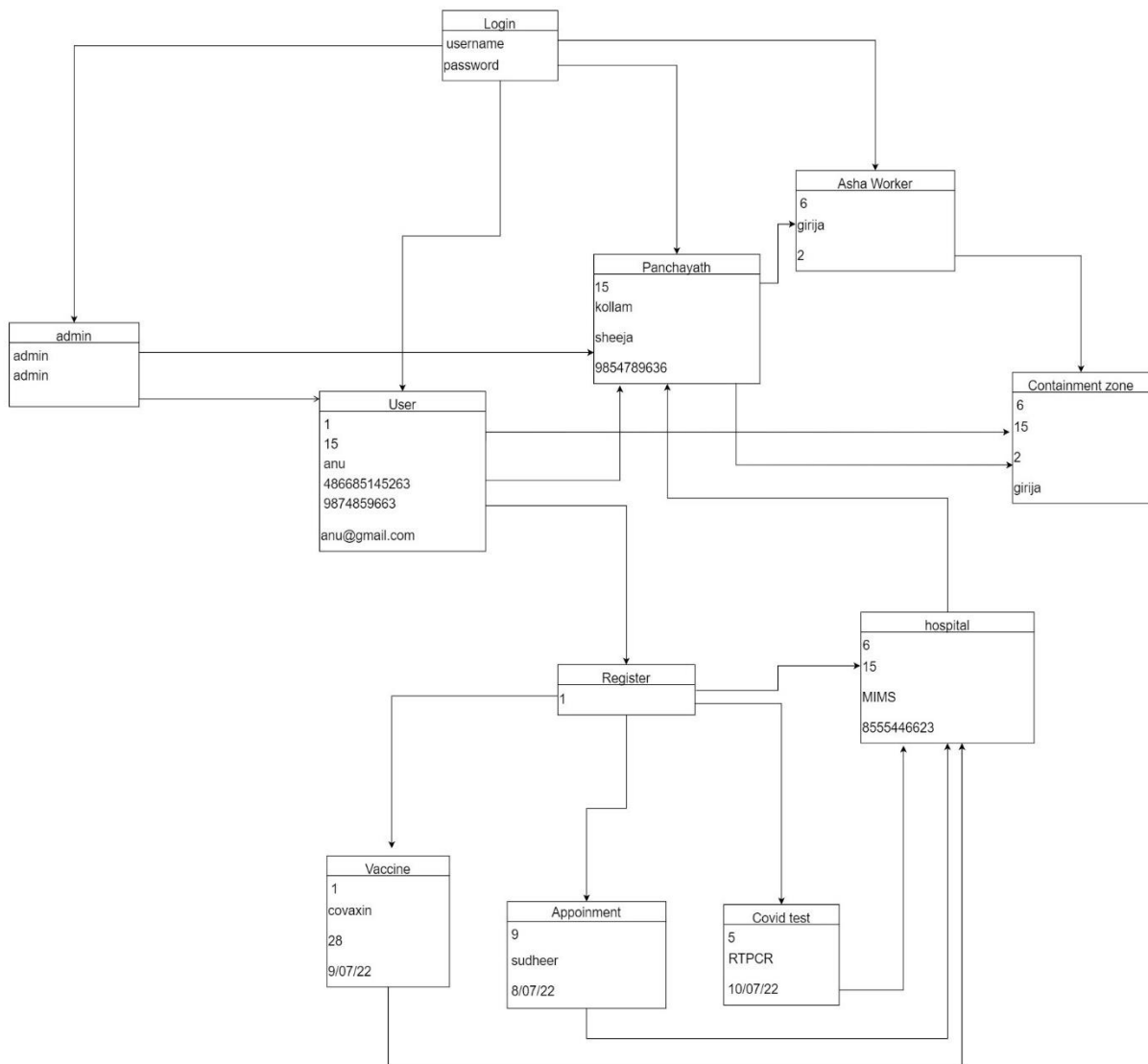
4.2.5 CLASS DIAGRAM

A class diagram is a static diagram. A static view of an application is represented by it. Class diagrams are employed not only for representing, describing, and documenting various parts of a system, but also for writing the software application's executable code. A class diagram explains a class's characteristics, methods, and restrictions on the system. The only UML diagrams that can be directly transferred to object-oriented languages are the class diagrams, which are extensively used in the modelling of object-oriented systems. Classes, interfaces, affiliations, collaborations, and constraints are all displayed in a class diagram. Alternatively, it is called a structural diagram.



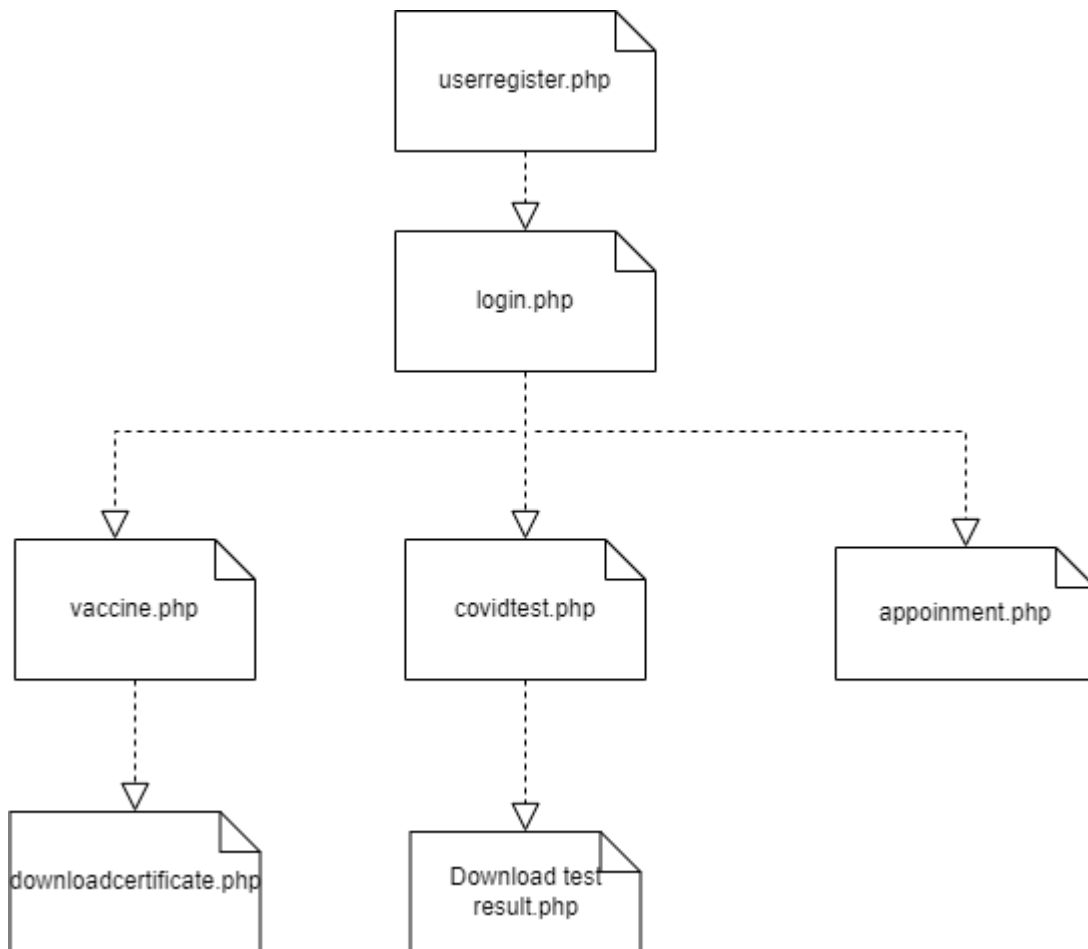
4.2.6 OBJECT DIAGRAM

Since object diagrams are formed from class diagrams, they are dependent class diagrams. An example of a class diagram is shown by an object diagram. Class diagrams and object diagrams both use the same fundamental ideas. The static view of a system is likewise represented by object diagrams, but this static view represents a snapshot of the system at a specific time. Using object diagrams, you can illustrate a group of things and their connections.



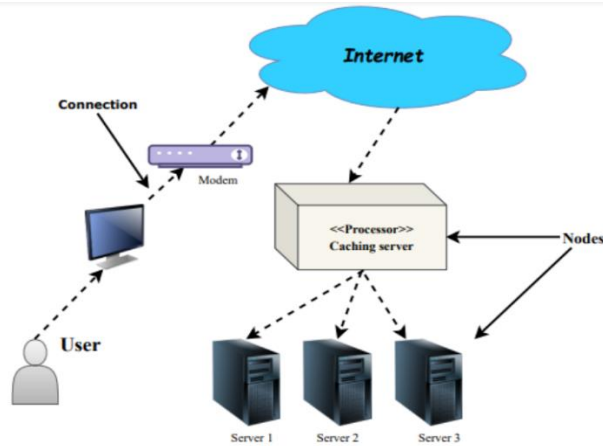
4.2.7 COMPONENT DIAGRAM

A component diagram, often called a UML component diagram, shows how the physical parts of a system are wired up and organised. To model implementation specifics and ensure that all necessary functionalities of the system are covered by planned development, component diagrams are frequently created.



4.2.8. DEPLOYMENT DIAGRAM

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system



4.5 USER INTERFACE DESIGN

4.5.1-INPUT DESIGN

Form Name : User Login



LOGIN

User Name:

PassWord:

☐ Remember me [Forgot Password](#)

[Don't You Have an Account...??](#) [Sign Up Now](#)

Form Name : Admin Home

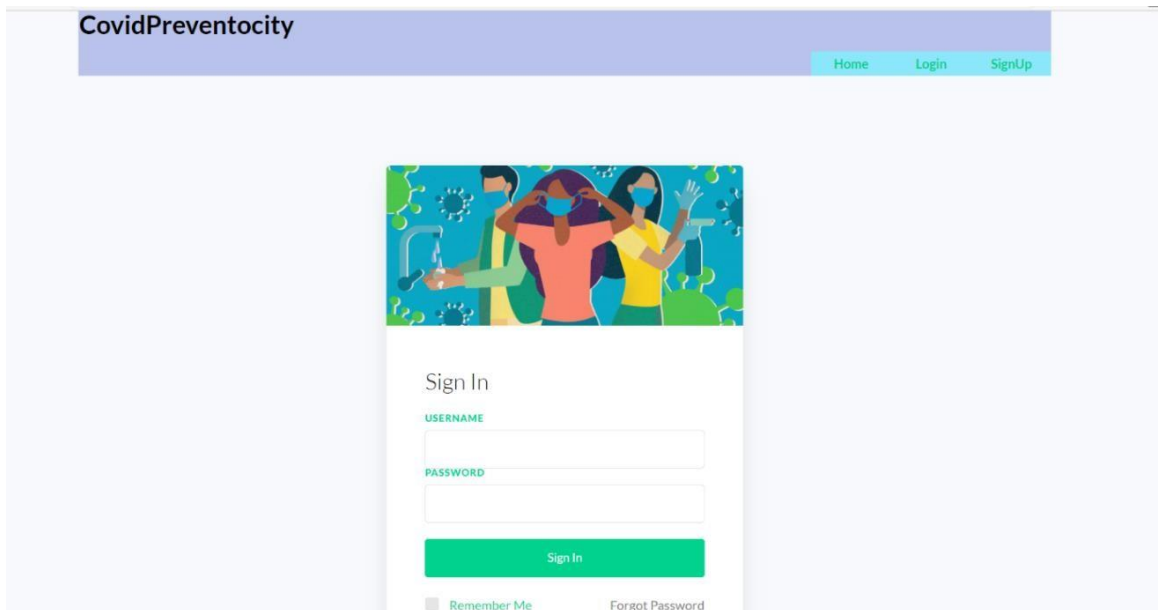
CoviD Preventocity														
Admin Home		Logout												
<div>View Users</div> <div>Manage Asha Worker</div> <div>View Containment Zones</div> <div>View wards</div> <div>Manage Panchayath</div> <div>Manage Hospital</div>	<div>Kerala</div> <table><thead><tr><th>Total cases</th><th>Deaths</th></tr></thead><tbody><tr><td>51.6L</td><td>41,124</td></tr><tr><td>+4,995</td><td>+4.995</td></tr></tbody></table> <div>India</div> <table><thead><tr><th>Total cases</th><th>Deaths</th></tr></thead><tbody><tr><td>3.46Cr</td><td>4.71L</td></tr><tr><td>+8,603</td><td>+415</td></tr></tbody></table>		Total cases	Deaths	51.6L	41,124	+4,995	+4.995	Total cases	Deaths	3.46Cr	4.71L	+8,603	+415
Total cases	Deaths													
51.6L	41,124													
+4,995	+4.995													
Total cases	Deaths													
3.46Cr	4.71L													
+8,603	+415													

Form Name : Panchayath Home

CoviD Preventocity														
Panchayath Home		Logout												
<div>Wards</div> <div>Containment Zone</div> <div>Vaccine</div> <div>Positive results</div> <div>Deaths</div> <div>Hospitals</div> <div>Orders and Notification</div>	<div>Kerala</div> <table><thead><tr><th>Total cases</th><th>Deaths</th></tr></thead><tbody><tr><td>51.6L</td><td>41,124</td></tr><tr><td>+4,995</td><td>+4.995</td></tr></tbody></table> <div>India</div> <table><thead><tr><th>Total cases</th><th>Deaths</th></tr></thead><tbody><tr><td>3.46Cr</td><td>4.71L</td></tr><tr><td>+8,603</td><td>+415</td></tr></tbody></table>		Total cases	Deaths	51.6L	41,124	+4,995	+4.995	Total cases	Deaths	3.46Cr	4.71L	+8,603	+415
Total cases	Deaths													
51.6L	41,124													
+4,995	+4.995													
Total cases	Deaths													
3.46Cr	4.71L													
+8,603	+415													

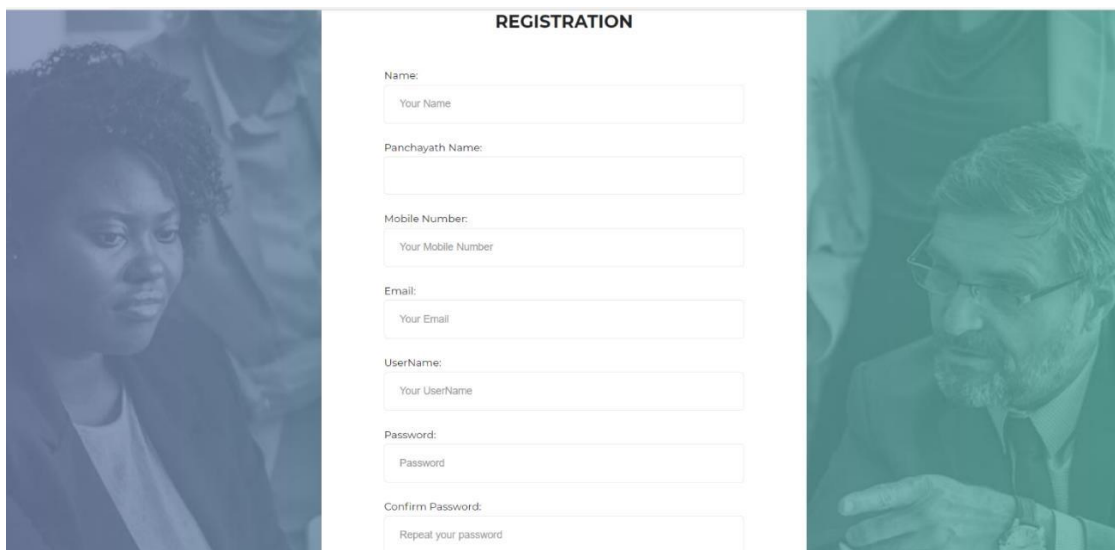
4.5.2 OUTPUT DESIGN

User Login



The image shows a web application interface for "CovidPreventocity". At the top, there is a purple header bar with the site name "CovidPreventocity" on the left and navigation links "Home", "Login", and "SignUp" on the right. Below the header, there is a large illustration of three people wearing face masks and gloves, with a virus particle nearby. In the center, there is a white "Sign In" form. The form has two input fields labeled "USERNAME" and "PASSWORD", a green "Sign In" button, and two links at the bottom: "Remember Me" and "Forgot Password".

User Registration



The image shows a web application interface for "REGISTRATION". The form is titled "REGISTRATION" and is set against a background of two people, one on the left and one on the right. The form fields are as follows:

- Name:
- Panchayath Name:
- Mobile Number:
- Email:
- UserName:
- Password:
- Confirm Password:

4.6. DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

4.6.1 Relational Database Management System (RDBMS)

A relational model represents the database as a set of relations. Every relation looks like a table or file of records with values. In the formal relational model language, a row is known as a tuple, a column heading as an attribute, and the table is known as a relation. There are many tables in a relational database, each with a unique name. Each row in a table stands for a collection of related values.

Relations, Domains & Attributes

A relation is a table. Tuples are the name for the rows in a table. An ordered group of n elements is a tuple. Attributes are referred to as columns. Every table in the database has relationships already established between them. This guarantees the integrity of both referential and entity relationships. A group of atomic values make up a domain D . Choosing a data type from which the domain's data values are derived is a typical way to define a domain. To make it easier to understand the values of the domain, it is also helpful to give it a name. Each value in a relation is atomic and cannot be broken down.

Relationships

- Key is used to establish the associations between tables. The Primary Key and Foreign Key are the two principal keys of utmost significance. These keys enable the establishment of Referential Integrity Relationships and Entity Integrity Relationships.
- No Primary Key may have null values, according to Entity Integrity.
- No Primary Key may have null values, according to the principle of referential integrity.
- Referential Integrity Each unique Foreign Key value must have a corresponding Primary Key value inside the same domain. The Super Key and Candidate Keys are other keys.

4.6.2 Normalization

To minimize the impact of future changes on data structures, data are put together in the simplest possible way. Normalization is the formal method of arranging data structures in ways that reduce duplication and support integrity. The process of normalization involves dividing superfluous fields and dispersing a huge table into several smaller ones. Additionally, it is employed to prevent insertion, deletion, and update abnormalities. Two notions, keys and relationships, are used in the standard style of data modelling. A table row can only be uniquely identified by a key. Keys come in two varieties: primary keys and foreign keys. When identifying records from the same table, a primary key is an element—or a set of elements—in the table. A column in a table known as a foreign key is used to uniquely identify records from other tables. Up to the third normal form, all tables have been normalized.

As the name suggests, it means returning things to their regular state. The application developer wants to establish a data structure through normalisation that allows users to readily correlate names with the data and that has the data logically sorted into the right tables and columns. Data redundancy, which places a heavy demand on the computer resources, is prevented by normalization, which eliminates repetitive groups at data. These include:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.

- ✓ Choose the proper name for the data.

First Normal Form

According to the First Normal Form, each attribute in a tuple must have a single value from the attribute's domain and its domain must only include atomic values. The 1NF forbids "relations within relations" or "relations as attribute values within tuples," in other words. By 1NF, only single atomic or indivisible values are allowed for attribute values. The first step is to convert the data to First Normal Form. To address this issue, data that is of a comparable type in each table can be transferred into different tables. According to the project's requirements, a Primary Key or Foreign Key is assigned to each table. According to the project's requirements, a Primary Key or Foreign Key is assigned to each table. For each non-atomic property or nested relation, new relations are created in this process. Repeated data groupings were removed as a result. A relation is considered to be in first normal form if and only if it complies with the main key-only requirement.

Second Normal Form

No non-key attribute should, in accordance with Second Normal Form, be functionally dependent on a portion of the primary key for relations when the main key has several attributes. For each partial key and its dependent attributes, we decompose and set up a new relation in this manner. Maintain the relationship between the original primary key and any properties that are entirely dependent on it. This procedure aids in removing information that is only dependent on a portion of the key. A relationship is said to be in second normal form if and only if all of the requirements for first normal form are met for the primary key and all of the relationship's non-primary key properties are completely dependent on the primary key alone.

Third Normal Form

A non-key attribute of a Relation should not be functionally determined by another non-key attribute or by a collection of non-key attributes, according to the Third Normal Form. In other words, the primary key shouldn't be transitively dependent on anything. In this, we break down the relation into its component parts and set up the non-key attributes that functionally determine the other non-key attributes. This step is taken to remove anything not fully dependent on the Primary Key. The non-key characteristics of a relation should also not be dependent on other non-key characteristics because a relation is only considered to be in third normal form if it is in second normal form.

TABLE DESIGN**Table No : 01****Table Name : tbl_login****Primary Key : l_id****Table Description : To store login information**

Field	Datatype	Constraint	Description
l_id	Varchar(20)	Primary key	unique id to table entry
username/email	Varchar(20)		username
password	Varchar(20)		password
type	Varchar(20)		type of users
status	int(11)		active or not

Table No : 02**Table Name : tbl_register****Primary Key : r_id****Foreign Key :pa_id, wa_id****Table Description : To store Registration information**

Field	Datatype	Constraint	Description
r_id	Varchar(20)	Primary key	unique id to table entry
pa_id	Varchar (20)	foreign key	panchayath table id
ward_id	Varchar(20)	foreign key	ward table id
name	varchar(20)		name
adhar	Varchar(20)		adhar
mobilenno	bigint		mobile number
email	varchar(20)		email

Table No : 03

Table Name : tbl_panchayath

Primary Key : pa_id

Table Description : To store panchayath information

Field	Datatype	Constraints	Description
pa_id	Varchar(20)	Primary Key	unique id to table entry
pa_name	Varchar(20)		panchayath name
no_wards	int(11)		number of wards
pa_president_name	Varchar(20)		panchayath president name
pa_president_number	Varchar(20)		panchayath president mobile number
pa_president_email	Varchar(20)		president email

Table No : 04

Table Name : tbl_ward

Primary Key : wa_id

Foreign Key :pa_id

Table Description : To store ward information

Field	Datatype	Constraints	Description
wa-id	Varchar(20)	Primary Key	unique id to table entry
pa_id	Varchar(20)	foreign key	panchayath table id
wa_no	Varchar(20)		ward number
ward_member_name	Varchar(20)		ward member name
ward_member_numb	Varchar(20)		member mobile no.
ward_member_email	Varchar(20)		member email

cont_status	varchar(10)		containment status active or not
-------------	-------------	--	-------------------------------------

Table No : 05

Table Name : tbl_containment_zone

Primary Key : zone_id

Foreign Key : ward_id

Table Description : To store containment zone information

Field	Datatype	Constraints	Description
zone_id	Varchar(20)	Primary key	unique id to table entry
ward_id	Varchar(20)	foreign key	ward table id
cont_status	int(11)		containment zone status
cont_declare_date	date		containment zone declare date
cont_end_date	date		zone end date

Table No : 06

Table Name : tbl_cont_details

Primary Key : det_id

Foreign Key : zone_id, pan_id, wa_id

Table Description : To store containment zone detail information

Field	Datatype	Constraints	Description
det_id	Varchar(20)	Primary key	unique id to table entry
zone_id	Varchar(20)	foreign key	containment zone table id
pan_id	Varchar(20)	foreign key	panchayath table id
wa_id	Varchar(20)	foreign key	ward table id
no_of_pstvs	int(11)		number of positives in the wards
no_of_contacts	int(11)		number of contacts with pstvs in a day
no_of_deaths	int(11)		number of deaths in a day

date	date		date
------	------	--	------

Table No : 07

Table Name : tbl_list_of_pstvs

Primary Key : li_pstv_id

Foreign Key :det_id

Table Description : To store postives information

Field	Datatype	Constraints	Description
li_pstv_id	Varchar(20)	primary key	unique id to table entry
det_id	Varchar(20)	foreign key	conatmnt zone detail table id
p_person_name	Varchar(20)		positive person name
p_person_dob	date		positive person date of birth
p_person_add	Varchar(20)		positive person house name
no_of_contacts	int(11)		number of contacts with this positive
pstv_date	date		positive date
pstv_status	int(11)		status of positive

Table No : 08

Table Name : tbl_contact_details

Primary Key : cont_id

Foreign Key :li_pstv_id

Table Description : To store primary contact information

Field	datatype	constraints	Description
cont_id	varchar(20)	primary key	unique id to table entry
li_pstv_id	varchar(20)	foreign key	list of pstv table id
c_per_name	varchar(20)		contact person name
c_per_add	varchar(20)		contact person house name
con_status	varchar(20)		status aftr test

Table No : 09

Table Name : tbl_list_of_death

Primary Key : li_death_id

Foreign Key :zone_id

Table Description : To store list of death information

Field	Datatype	Constraints	Description
li_death_id	Varchar(20)	Primary key	unique id to table entry
zone_id	Varchar(20)	foreign key	containnt zone table id
d_person_name	Varchar(20)		death person name
d_person_dob	date		death person date of birth
d_person_add	Varchar(20)		address

Table No : 10

Table Name : tbl_ashaworker

Primary Key : as_id

Foreign Key :wa_id,pa_id

Table Description : To store asha worker information

Field	Datatype	Constraints	Description
as_id	Varchar(20)	primary key	unique id to table entry
wa_id	Varchar(20)	foreign key	ward table id
pa_id	Varchar(20)	foreign key	panchayath table id
as_name	Varchar(20)		asha worker name
as_dob	date		asha worker date of birth
as_mobile	numeric		mobile number
as_email	Varchar(20)		email
as_status	int(11)		active or not

Table No : 11

Table Name : tbl_hospital

Primary Key : h_id

Foreign Key :pa_id,wa_id,vac_id

Table Description : To store hospital information

Field	Datatype	Constraints	Description
h_id	varchar(20)	primary key	unique id to table entry
pa_id	varchar(20)	foreignkey	panchayath table id
wa_id	varchar(20)	foreignkey	ward table id
vac_id	varchar(20)	foreignkey	vaccine table id

h_name	varchar(20)		hospital name
h_place	varchar(20)		hospital place
h_vaccine_status	int(11)		availability

Table No : 12

Table Name : tbl_vaccine

Primary Key : vac_id

Foreign Key :

Table Description : To store vaccine information

Field	Datatype	Constraints	Description
vac_id	varchar(20)	primary key	unique id for table entry
vac_name	int(11)		name of vaccine
no_of_dose	int(11)		number of dose needed
interval	int(11)		interval period

Table No : 13

Table Name : tbl_consultreg

Primary Key : cons_id

Foreign Key :log_id,sch_id

Table Description : To store registration for consulting information

Field	Datatype	Constraint	Description
cons_id	int(11)	Primary key	unique id to table entry
log_id	int(11)	foreign key	login table id
sch_id	int(11)	foreign key	doctorschedule table id
conscat	Varchar(11)		doctor category
name	Varchar(20)		name
age	int(11)		age
con_type	varchar(10)		consultation type

mobile	bigint		mobile number
status	varchar(10)		status of registration
consult	consult(15)		status of consultation
codate	date		date of consultation

Table No : 14

Table Name : tbl_covpat

Primary Key : cpat_id

Foreign Key : panch_id, hos_id

Table Description : To store covid patients information in hospital

Field	Datatype	Constraint	Description
cpat_id	int(11)	Primary key	unique id to table entry
panch_id	int(11)	foreign key	panchayath table id
hos_id	int(11)	foreign key	hospital table id
name	varchar(20)		patient name
age	int(11)		patient age
mobile	bigint		patient mobile number
status	varchar(10)		patience discharge status

Table No : 15

Table Name : tbl_cvdtest

Primary Key : ctest_id

Foreign Key : panch_id, ward_id, hos_id, log_id

Table Description : To store covid test information

Field	Datatype	Constraint	Description
ctest_id	int(11)	Primary key	unique id to table entry
panch_id	int(11)	foreign key	panchayath table id
ward_id	int(11)	foreign key	ward table id
hos_id	int(11)	foreign key	hospital table id

log_id	int(11)	foreign key	login table id
type	int(11)	foreign key	testtype table id
tpname	varchar(20)		name
age	int(11)		age
adhar	bigint		adhar number
ctmobile	bigint		mobile number
ctemail	varchar(30)		email
tdate	date		date
status	varchar(10)		status
result	varchar(10)		result
estatus	varchar(10)		email status
pstatus	varchar(10)		payment status

Table No : 16

Table Name : tbl_docsch

Primary Key : sch_id

Foreign Key :doc_id,slot_id

Table Description : To store doctor schedule information

Field	Datatype	Constraint	Description
sch_id	int(11)	Primary key	unique id to table entry
doc_id	int(11)	foreign key	doctor table id
slot_id	int(11)	foreign key	time slot table id
date	date		date
tkns	int(11)		token
status	int(11)		status

Table No : 17

Table Name : tbl_doctor

Primary Key : doc_id

Foreign Key :cat_id,h_id

Table Description : To store doctor information

Field	Datatype	Constraint	Description
doc_id	int(11)	Primary key	unique id to table entry
cat_id	int(11)	foreign key	doc category table id
h_id	int(11)	foreign key	hospital table id
doc_name	varchar(20)		doctor name
doc_spec	varchar(20)		specialization
tkn	int(11)		token
status	varchar(8)		status

Table No : 18

Table Name : tbl_doc_cat

Primary Key : cat_id

Foreign Key :

Table Description : To store doctor category information

Field	Datatype	Constraint	Description
cat_id	int(11)	Primary key	unique id to table entry
ctype	varchar(20)		category type

Table No : 19

Table Name : tbl_hosvacc

Primary Key : hvac_id

Foreign Key :hos_id,vac_id

Table Description : To store hospital vaccine information

Field	Datatype	Constraint	Description
hvac_id	int(11)	Primary key	unique id to table entry
hos_id	int(11)	foreign key	hospital table id
vac_id	int(11)	foreign key	vaccine table id
num_vacc	int(11)		number of vaccine

Table No : 20

Table Name : tbl_hosvaccsch

Primary Key : hvs_id

Foreign Key :hvac_id,tset_id

Table Description : To store hospital vaccine schedule information

Field	Datatype	Constraint	Description
hvs_id	int(11)	Primary key	unique id to table entry
hvac_id	int(11)	foreign key	hosvacc table id
tset_id	int(11)	foreign key	time table id
num	int(11)		number
date	date		date
status	int(11)		status

Table No : 21

Table Name : tbl_panchvacc

Primary Key : pvacc_id

Foreign Key :vac_id,panch_id

Table Description : To store panchayath vaccine information

Field	Datatype	Constraint	Description
pvacc_id	int(11)	Primary key	unique id to table entry
vac_id	int(11)	foreign key	vaccine table id
panch_id	int(11)	foreign key	panchayath table id
avail	int(11)		availability

Table No : 22

Table Name : tbl_payment

Primary Key : pamnt_id

Foreign Key :log_id,cvd_id

Table Description : To store payment information

Field	Datatype	Constraint	Description
pamnt_id	int(11)	Primary key	unique id to table entry
log_id	int(11)	foreign key	login table id
cvd_id	int(11)	foreign key	covidtest table id
pamount	int(11)		amunt
pdate	datetime		date
pstatus	varchar(10)		status

Table No : 23

Table Name : tbl_posthos

Primary Key : pt_id

Foreign Key : panch_id, h_id

Table Description : To store positive result information from hospital

Field	Datatype	Constraint	Description
pt_id	int(11)	Primary key	unique id to table entry
panch_id	int(11)	foreign key	panchayath table id
h_id	int(11)	foreign key	hospital table id
ward_num	int(11)		ward number
pt_name	varchar(20)		patient name
pt_mob	bigint(20)		patient mobile
cdate	date		date

Table No : 24

Table Name : tbl_usvacc

Primary Key : uvc_id

Foreign Key : user_id, hvs_id, panch_id

Table Description : To store user vaccine registration information

Field	Datatype	Constraint	Description
uvc_id	int(11)	Primary key	unique id to table entry
user_id	int(11)	foreign key	register table id
hvs_id	int(11)	foreign key	hosvaccsch table id
panch_id	int(11)	foreign key	panchayath table id
uvname	varchar(20)		name
uvage	int(11)		age
uvadhar	bigint(20)		adhar
vdose	int(11)		dose
vdate	date		date

vstatus	varchar(15)		status
---------	-------------	--	--------

Table No : 25

Table Name : tbl_vaccreq

Primary Key : req_id

Foreign Key :h_id,panch_id,vac_id

Table Description : To store vaccine request information from hospital

Field	Datatype	Constraint	Description
req_id	int(11)	Primary key	unique id to table entry
h_id	int(11)	foreign key	hospital table id
panch_id	int(11)	foreign key	panchayth table id
vac_id	int(11)	foreign key	vaccine table id
vac_num	int(11)		number of vaccine
vdate	date		date
status	varchar(20)		status

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software testing is the process of carefully controlling the execution of software in order to determine whether it behaves as intended. The words verification and validation are frequently used in conjunction with software testing. Validation is the process of examining or evaluating a product, including software, to determine whether it complies with all relevant specifications. One type of verification, software testing, uses methods including reviews, analyses, inspections, and walkthroughs as well. Verifying that what has been specified matches what the user truly desired is the process of validation.

Static and dynamic analysis are additional practices that are frequently related to software testing. Software's source code is examined via static analysis, which collects metrics and looks for errors without actually running the code. In order to give data like execution traces, timing profiles, and test coverage information, dynamic analysis examines how software behaves while it is running.

A series of activities known as testing can be organized in advance and carried out in a methodical manner. Beginning with the module level, testing progresses to the integration of the full computer-based system. Numerous rules can be utilised as testing objectives, and testing would not be complete without them because they are crucial to the system's success. They are:

A program is tested by being run with the goal of identifying any errors.

- A test case with a high likelihood of detecting an unknown fault qualifies as a good test case.
- A test that finds an error that has not yet been found is successful.

If a test is successfully carried out in accordance with the aforementioned aims, it will reveal software bugs. Additionally, testing shows that the software functions seem to be functioning in accordance with the specifications and that the performance requirements seem to have been satisfied.

There are three ways to test program.

- For correctness

- For implementation efficiency
- For computational complexity

Testing for correctness is meant to ensure that a program performs exactly as it was intended to. This is much harder than it might initially seem, especially for big programs.

5.2 TEST PLAN

A test plan suggests a number of required steps that need be taken in order to complete various testing methodologies. The activity that is to be taken is outlined in the test plan. A computer program, its documentation, and associated data structures are all created by software developers. It is always the responsibility of the software developers to test each of the program's separate components to make sure it fulfils the purpose for which it was intended. In order to solve the inherent issues with allowing the builder evaluate what they have developed, there is an independent test group (ITG). Testing's precise goals should be laid forth in quantifiable language. The mean time to failure, the cost to find and fix faults, the frequency of occurrence or density of residual defects, and the number of test work hours needed for each regression test should all be included in the test plan.

The levels of testing include:

- ❖ Unit testing
- ❖ Integration Testing
- ❖ Data validation Testing
- ❖ Output Testing

5.2.1 Unit Testing

The smallest unit of software design—the software component or module—is the focus of unit testing, which concentrates verification work. Important control pathways are examined in order to find faults inside the module's border using the component level design description as a guide. The scope set for unit testing and the relative complexity of tests. Unit testing can be carried out simultaneously for numerous components and is white-box focused. Information flow into and out of the program unit under test is monitored by the modular interface to ensure appropriate operation. To make sure that temporary data is kept in its original format during all phases of an algorithm's execution, the local data structure is inspected. To confirm that each statement in a

module has been executed at least once, boundary conditions are evaluated. Finally, each path for managing errors is examined. Before starting any other test, tests of data flow over a module interface are necessary. All other tests are irrelevant if data cannot enter and depart the system properly. An important duty during the unit test is the selective examination of execution pathways. Error handling pathways must be set up to cleanly reroute or stop work when an error does occur. Error conditions must be anticipated in excellent design. Boundary testing comes after unit testing. The limits of software regularly cause failures.

In the Sell-Soft System, unit testing was carried out by treating each module as a distinct entity and subjecting them to a variety of test inputs. The internal logic of the modules had some issues, which were fixed. Each module is tested and run separately after coding. To guarantee that every module functions properly and produces the desired outcome, all extraneous code was deleted.

5.2.2 Integration Testing

The methodical procedure of building the program's structure and performing tests simultaneously to detect interface problems is known as integration testing. The objective is to build a program structure that has been predetermined by design using unit-tested components. The program as a whole is put to the test. The vastness of the entire program makes it challenging to isolate the reasons, making correction challenging. Once these mistakes are fixed, new ones come into being, and the process keeps going in an apparently infinite loop. All the modules were merged into the System after unit testing was completed to check for any interface consistency issues. Additionally, variations in program structures were eliminated, and a special program structure developed.

5.2.3 Validation Testing or System Testing

Testing has reached its conclusion. With all forms, code, modules, and class modules present, the full system was tested in this. System tests or Black Box testing are two terms commonly used to describe this type of testing.

The focus of the black box testing strategy is the functional requirements of the software. In other words, black box testing enables the software engineer to create sets of input scenarios that will fully exercise all functional requirements for a program.

Erroneous or missing functions, interface problems, issues with data structures or external data access, performance problems, initialization problems, and termination problems are all things

that black box testing looks for in the code.

5.2.4 Output Testing or User Acceptance Testing

User approval of the system under consideration is tested; in this case, it must meet the needs of the company. When developing, the program should stay in touch with the user and perspective system to make modifications as needed. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The aforementioned testing is carried out using a variety of test data. The preparation of test data is essential to the system testing process. The system under investigation is then put to the test using the prepared test data. Errors in the system are once again found during testing, fixed using the methods described above, and logged for use in the future.

5.2.5.Automation Testing

Software and other computer goods are tested automatically to make sure they abide by tight guidelines. In essence, it's a test to ensure that the hardware or software performs exactly as intended. It checks for errors, flaws, and any other problems that might occur throughout the creation of the product. Any time of day can be used to do automation testing. It looks at the software using scripted sequences. It then summarizes what was discovered, and this data can be compared to results from earlier test runs.

Benefits of Automation Testing

Detailed reporting capabilities - Test cases for different scenarios are carefully built for automation testing. These planned sequences can cover a lot of ground and produce in-depth reports that are simply impossible for a human to produce.

Improved bug detection - Finding bugs and other flaws in a product is one of the key reasons to test it. This procedure can be made simpler with automation testing. Additionally, it can examine a greater test coverage than perhaps people can.

- Simplifies testing - Most SaaS and IT organizations routinely include testing in their daily operations. The key is to keep things as basic as you can. Automation has a lot of advantages. The test scripts can be reused when automating test tools.
- Quickens the testing procedure - Machines and automated technology operate more quickly

than people. This is why we employ them, along with increased accuracy. Your software development cycles are subsequently shortened by this.

- Lessens the requirement for human supervision - Tests may be conducted at any time of day, including overnight. Additionally, this can lessen the possibility of human error when it is carried out mechanically.

Selenium Testing

An open-source program called Selenium automates web browsers. It offers a single interface via which you can create test scripts in a variety of computer languages, including Ruby, Java, NodeJS, PHP, Perl, Python, and C#. Web application testing for cross-browser compatibility is automated using the Selenium testing tool. Whether they are responsive, progressive, or standard, it is utilized to assure high-quality web apps. Selenium is a free software program.

Test cases for a Login Page

Project Name: Covid Preventocity					
Login Test Case					
Test Case ID: Fun_1			Test Designed By: Sreya M		
Test Priority (Low/Medium/High): High			Test Designed Date:17-07-2022		
Module Name: Login Screen			Test Executed By: Mr. T J Jobin		
Test Title: Verify login with valid username and password			Test Execution Date: 18-07-2022		
Description: Test the Login Page					
Pre-Condition: User has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigation to Login Page		Login Page should be display ed	Login page displayed	Pass
2	Provide Valid username	Username : anu123	User should d be able to Login	User Logged in and navigated to User Dashboard	Pass
3	Provide Valid Password	Password:anu123			
4	Click on Sign In button				
5	Provide Invalid username or password	Username:abin123 Password: abin12345	User should not be able to Login	Message for enter valid email id or password displayed	Pass
6	Provide Null username or Password	Username : null Password: null			
7	Click on Sign In button				

Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database.

Code Package:

```
package mca;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class test{
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver","C:\\\\Users\\hp\\Downloads\\chromedriver_win32\\c
hromedriver.exe" );
        WebDriver driver=new ChromeDriver();
        driver.get("http://localhost/CovidPt/login.php");
        driver.findElement(By.id("username")).sendKeys("anu123");
        driver.findElement(By.id("passwd")).sendKeys("anu123");
        driver.findElement(By.id("submit")).click();
        String actualUrl="http://localhost/CovidPt/userhome.php";
        String expectedUrl= driver.getCurrentUrl();
        if(actualUrl.equalsIgnoreCase(expectedUrl)) {
            System.out.println("Test passed");
        } else {
            System.out.println("Test failed");
        }
    }
}
```



```
1 package mca;
2
3
4 import org.openqa.selenium.By;
5
6 public class test{
7     public static void main(String[] args) {
8         System.setProperty("webdriver.chrome.driver", "C:\\Users\\hp\\Downloads\\chromedriver_win32\\chromedriver.exe" );
9         WebDriver driver=new ChromeDriver();
10        driver.get("http://localhost/CovidPt/login.php");
11        driver.findElement(By.id("username")).sendKeys("anu123");
12        driver.findElement(By.id("passwd")).sendKeys("anu123");
13        driver.findElement(By.id("submit")).click();
14        String actualUrl="http://localhost/CovidPt/userhome.php";
15        String expectedUrl= driver.getCurrentUrl();
16        if(actualUrl.equalsIgnoreCase(expectedUrl)) {
17            System.out.println("Test passed");
18        } else {
19            System.out.println("Test failed");
20        }
21    }
22 }
23
24 }
25
26
27
```

```
Starting ChromeDriver 103.0.5060.53 (a171181edd74ff1cf2150f36ffa3b0dae40b17f-refs/branch-heads/5060@{#853}) on port 57420
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul 18, 2022 12:12:59 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected upstream dialect: W3C
Jul 18, 2022 12:12:59 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 103
Test passed
```

Test cases for User Registration

Project Name: Covid Preventocity					
Updation Test Case					
Test Case ID: registration			Test Designed By: Sreya M		
Test Priority (Low/Medium/High): High			Test Designed Date: 17-07-2022		
Module Name: Registration			Test Executed By: Mr. T J Jobin		
Test Title: User Registration Details			Test Execution Date: 18-07-2022		
Description: Register to system and Registration is completed then login , if someerror occurs, test will fail					
Pre-Condition: User has valid user name and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/Fail)
1	Navigation to Register Page		Register Page should be	Registrtrion page displayed	Pass
2	Provide Valid Registration details	User Name: Sreevi dhya	User should be able to Register	User registrionCompleted after go to the login page	Pass
3					
4	Click on Login button				
5	Provide profile details	Input profile details	User will be redirected to Login page	Use will be redirected to Login page	Pass
7	Click on register button				
8	Provide invalid information	Input invalid profile details.	User will be stay in register page	User will be stay on that page showing error message	Pass
9	Click on register button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database.					

Code Package:

```
package mca;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class register{
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver","C:\\Users\\hp\\Downloads\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.get("http://localhost/CovidPt/reguser.php");
        driver.findElement(By.id("name")).sendKeys("Sreevidhya");
        driver.findElement(By.id("panch")).sendKeys("Kollam");
        driver.findElement(By.id("adhar")).sendKeys("969874563241");
        driver.findElement(By.id("mobile")).sendKeys("9645857441");
        driver.findElement(By.id("emailid")).sendKeys("sreevid@gmail.com");
        driver.findElement(By.id("username")).sendKeys("Sree123");
        driver.findElement(By.id("passwd")).sendKeys("Sree123@");
        driver.findElement(By.id("cpasswd")).sendKeys("Sree123@");
        driver.findElement(By.id("submit")).click();
        String actualUrl="http://localhost/CovidPt/login.php?e=1";
        String expectedUrl= driver.getCurrentUrl();
        if(actualUrl.equalsIgnoreCase(expectedUrl)) {
            System.out.println("Test passed");
        } else {
            System.out.println("Test failed");
        }
    }
}
```



```
exp.java test.java register.java ×
1 package mca;
2
3 import org.openqa.selenium.By;
4
5 public class register{
6     public static void main(String[] args) {
7         System.setProperty("webdriver.chrome.driver", "C:\\Users\\hp\\Downloads\\chromedriver_win32\\chromedriver.exe" );
8         WebDriver driver=new ChromeDriver();
9         driver.get("http://localhost/CovidPt/reguser.php");
10        driver.findElement(By.id("name")).sendKeys("Sreevidhya");
11        driver.findElement(By.id("panch")).sendKeys("Kollam");
12        driver.findElement(By.id("adhar")).sendKeys("969874563241");
13        driver.findElement(By.id("mobile")).sendKeys("9645857441");
14        driver.findElement(By.id("emailid")).sendKeys("sreevid@gmail.com");
15        driver.findElement(By.id("username")).sendKeys("Sree123");
16        driver.findElement(By.id("passwd")).sendKeys("Sree123@");
17        driver.findElement(By.id("cpasswd")).sendKeys("Sree123@");
18        driver.findElement(By.id("submit")).click();
19        String actualUrl="http://localhost/CovidPt/login.php?e=1";
20        String expectedUrl= driver.getCurrentUrl();
21        if(actualUrl.equalsIgnoreCase(expectedUrl)) {
22            System.out.println("Test passed");
23        } else {
24            System.out.println("Test failed");
25        }
26    }
27
28 }
29 }
30 }
31
32
```

```
Problems @ Javadoc Declaration Console ×
<terminated> register [Java Application] C:\Users\hp\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (18-Jul-2022, 12:23:57)
Starting ChromeDriver 103.0.5060.53 (a1711811edd74ff1cf2150f36ffa3b0dae40b17f-refs/branch-heads/5060@{#853}) on port 53877
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul 18, 2022 12:24:01 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected upstream dialect: W3C
Jul 18, 2022 12:24:02 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 103
Test passed
```

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

The step of the project known as implementation is where the conceptual design is translated into a functional system. The key to developing a successful new system is gaining the users' confidence that it will perform as intended, be dependable, and effective. Its primary concerns are user education and documentation.

Normally, conversion happens either at the same time that the user is being trained or after. Implementing a new system design simply refers to putting it into action, which is the process of turning a newly revised system design into an operational one.

The user department now bears the most of the workload, faces the most disruption, and has the biggest influence on the current system. If the implementation is not well thought out or managed, confusion and mayhem may result.

Implementation encompasses all of the steps used to switch from the old system to the new one. The new system can be completely different, replace an existing human or automated system, or simply be improved. The correct implementation of a dependable system that meets organisational requirements is required. System use or system implementation is the act of putting a built-in system into practice. This comprises all the processes involved in switching from the old to the new system. Only after extensive testing and if it is determined that the system is operating in accordance with the standards can it be put into use. The system personnel assess the system's viability. The amount of work needed to implement the three crucial components of education and training, system testing, and changeover will depend on how complicated the system being implemented is. This will also determine how much work is needed for system analysis and design.

The implementation state involves the following tasks:

- ☐ Careful planning.
- ☐ Investigation of system and constraints.
- ☐ Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Software implementation refers to the complete installation of the package in its intended environment, as well as to the system's functionality and satisfaction of its intended applications. The software development project is frequently commissioned by someone who will not be using it. People first have doubts about the software, but it's important to

make sure that resistance doesn't grow because one needs to make sure that:

- ☐ The active user must be aware of the benefits of using the new system.
- ☐ Their confidence in the software is built up.
- ☐ Proper guidance is imparted to the user so that he is comfortable in using the application.

Before examining the system, the user must be aware that the server software needs to be running on the server in order to access the results. Inactivity of the server object on the server will prevent the real procedure from occurring.

6.2.1 User Training

The purpose of user training is to get the user ready to assess and modify the system. The participants must have faith in their roles in the new system in order to achieve the goal and gain the benefits anticipated from a computer-based system. Training is becoming more and more important as systems get more complex. User training teaches the user how to perform critical operations such as entering data, handling error warnings, querying the database, calling up routines to generate reports, and more.

6.2.2 Training on the Application Software

Before being taught how to utilize the new application software, the user must first get basic instruction in computer literacy. This will cover the basic concepts of using the new system, such as how the screens operate, what type of help is shown on them, what kinds of mistakes people make when entering data, how each entry is validated, and how to update the data that was entered. The knowledge required by a certain user or group to use the system or a particular system component should therefore be included in the program's training sessions on the application. This training may alter depending on the user group and organisational level.

6.2.3 System Maintenance

The mystery of system development is maintenance. When a software product is in the maintenance stage of its lifecycle, it is actively working. A system should be properly maintained after it has been effectively implemented. An essential part of the software development life cycle is system maintenance. In order for a system to be flexible to changes in the system environment, maintenance is required. Of course, software maintenance involves much more than just "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The current system working technology is manual and there is no usage of commonly used technologies like internet, digital money. The proposed system introduces facility for easy access of covid informations. This proposed system allows user to view containment zone details in each panchayath, take vaccine and covid tests. User can also able to download the vaccine certificate and covid test report. Panchayath can able to view the covid test details and positives from test report from each hospitals. Also panchayath can view the vaccines status in particular panchayath.

7.2 FUTURE SCOPE

- Users can access the special corona benefits provided by Panchayath
- Can add government health centers in this system.
- Data security can be enhanced.
- Provide status of ICU and bed vacancy in hospitals.
- User can provide complaints and feedback

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”, 2009.
- Roger S Pressman, “*Software Engineering*”, 1994.
- Pankaj Jalote, “*Software engineering: a precise approach*”, 2006.
- James lee and Brent ware Addison, “Open source web development with LAMP”, 2003
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES:

- www.w3schools.com
- www.jquery.com
- <http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf>
- www.agilemodeling.com/artifacts/useCaseDiagram.html

CHAPTER 9

APPENDIX

9.1 Sample Code

log.php

```

<?php
session_start();
include("connection.php");
if (isset($_SESSION['userid'])) {
    unset($_SESSION['userid']);
    session_destroy();
    if (headers_sent()) {
        die('<script type="text/javascript">window.location.href="login.php?e=1"</script>');
    } else {
        header("location:login.php?e=1");
        die();
    }
} else {
    $conn = mysqli_connect($servername, $username, $password);
    mysqli_select_db($conn, $db);
    $username = $_POST["username"];
    $passwd = $_POST["passwd"];
    $sql2 = "select * from tbl_login where username='" . $username . "' AND passwd='" . $passwd . "'";
    $result = mysqli_query($conn, $sql2);
    if ($result) {
        if ($row = mysqli_fetch_array($result)) {
            $_SESSION['id'] = $row['login_id'];
            if ($row[3] == "1") {
                ?>
                <script type="text/Javascript">
                    window.location.href="adminpanel.php"</script>
                <?php
            } else if ($row[3] == "2"&& $row[4] == "1") {
                ?>
                <script type="text/Javascript">
                    window.location.href="hospitalhome.php"</script>
                <?php
            } else if ($row[3] == "3"&& $row[4] == "1") {
                ?>
                <script type="text/Javascript">
                    window.location.href="userhome.php"</script>
                <?php
            } else if ($row[3] == "4"&& $row[4] == "1") {
                ?>
                <script type="text/Javascript">
                    window.location.href="panchhome.php"</script>
                <?php
            } else {
                echo "Invalid Username and Password";
            }
        }
    }
}
}

```

```
}?>
```

hostreemail.php

```
<?php
//Import PHPMailer classes into the global namespace
//These must be at the top of your script, not inside a function
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;
$email=$_GET['email'];
$id=$_GET['id'];
require 'src\Exception.php';
require 'src/PHPMailer.php';
require 'src/SMTP.php';
include('connection.php');
mysqli_query($conn,"update `tbl_cvdtest` set estatus='Sent' where ctest_id='$id'");
$sql=mysqli_query($conn,"SELECT * FROM `tbl_cvdtest` WHERE `ctemail`='$email'");
$rows=mysqli_fetch_array($sql);
$resl=$rows['result'];
$name=$rows['tpname'];

//Create an instance; passing `true` enables exceptions
$mail = new PHPMailer(true);

try {
    //Server settings
    $mail->SMTPDebug = SMTP::DEBUG_SERVER;           //Enable verbose debug output
    $mail->isSMTP();                               //Send using SMTP
    $mail->Host      = 'smtp.gmail.com';             //Set the SMTP server to send through
    $mail->SMTPAuth  = true;                         //Enable SMTP authentication
    $mail->Username   = 'covidpreventocity@gmail.com'; //SMTP username
    $mail->Password   = 'gkvfphkmqozsdund';          //SMTP password
    $mail->SMTPSecure = 'tls';                       //Enable implicit TLS encryption
    $mail->Port       = 587;                         //TCP port to connect to; use 587 if you have set `SMTPSecure =
    PHPMailer::ENCRYPTION_STARTTLS`

    //Recipients
    $mail->setFrom('covidpreventocity@gmail.com', 'Covid Preventocity');
    $mail->addAddress($email); //Add a recipient
    // $mail->addAddress('ellen@example.com'); //Name is optional
    // $mail->addReplyTo('info@example.com', 'Information');
    // $mail->addCC('cc@example.com');
    // $mail->addBCC('bcc@example.com');

    //Attachments
    /// $mail->addAttachment('/var/tmp/file.tar.gz'); //Add attachments
    // $mail->addAttachment('/tmp/image.jpg', 'new.jpg'); //Optional name

    //Content
    $mail->isHTML(true); //Set email format to HTML
    $mail->Subject = 'Your Covid Test Result';
    $mail->Body    = 'Hy, '.$name.' Your Covid 19 test result is <br>Result : '.$resl;
```

```

$mail->AltBody = 'Thanks For your Interest';

$mail->send();
echo "<script>alert('Message has been Sent');window.location.href='hospitalhome.php';</script>";

} catch (Exception $e) {
    echo $email;
    echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";
}

```

cvdgraph.php

```

<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<?php
include("connection.php");

$chkresults = mysqli_query($conn,"SELECT `tdate`, COUNT(*) AS cvd_count FROM `tbl_cvdtest`
GROUP BY tdate")
?>
<h2>Covid Test </h2>
<script type="text/javascript">

    google.charts.load('current', {'packages':['Line']});

    google.charts.setOnLoadCallback(drawChart);

    function drawChart() {

        var data = google.visualization.arrayToDataTable([

            ['tdate','tdate'],

            <?php
            while($row=mysqli_fetch_array($chkresults)){

                echo "[".$row["tdate"].",".$row["cvd_count"]."],"";

            }

            ?>

        ]);

        var options = {

            chart: {

                title: "

            },


```

```

bars: 'vertical',

vAxis: {format: 'decimal'},

height: 300,

colors: ['#d95f02']

};

var chart = new google.charts.Line(document.getElementById('line-chart-location'));
chart.draw(data, google.charts.Line.convertOptions(options));

}

</script>

```

paymentgateway.php

```

<?php
    $apiKey="rzp_test_uXSVdcc4SvCdVl";
    include 'connection.php';
    session_start();
    if(isset($_SESSION['id']))
    {
    ?>
    <!DOCTYPE html>
    <html>
        <title>Payment Gateway</title>
    <head>

    </head>
    <div class="anim" id="anim"></div>
    <?php
        $u=$_SESSION['id'];
        $query4 ="SELECT * FROM tbl_register where login_id='$u'";
        $res4 = mysqli_query($conn,$query4);
        $r4=mysqli_fetch_array($res4);
        $l=$_GET['uid'];
        $query="select * from tbl_cvdtest";
        $query5 ="SELECT tbl_cvdtest.tpname,tbl_cvdtest.ctemail,tbl_testtype.tamount
        FROM tbl_cvdtest JOIN tbl_testtype on tbl_cvdtest.type=tbl_testtype.tety_id AND
tbl_cvdtest.ctest_id='$l'";
        $res5 = mysqli_query($conn,$query5);
        $r5=mysqli_fetch_array($res5);

    ?>
    <section>
        <div class="product">
            <div class="description">
                <h3>The AmountPay By: <?php echo $r5['tpname'];?> </h3>
                <h3>The Amount : <?php echo $r5['tamount'];?> </h3>
            </div>

```

```

</div>

<form action="payaction.php?id=<?php echo $l;?>" method="post">
<script
  src="https://checkout.razorpay.com/v1/checkout.js"
  data-key="<?php echo $apiKey; ?>" // Enter the Test API Key ID generated from Dashboard
  → Settings → API Keys
  data-amount="<?php echo $r5['tamount'] * 100;?>" // Amount is in currency subunits. Hence,
  29935 refers to 29935 paise or ₹299.35.
  data-currency="INR"// You can accept international payments by changing the currency
  code. Contact our Support Team to enable International for your account
  data-order_id="<?php rand(100000, 999999);?>"// Replace with the order_id generated by
  you in the backend.
  data-buttontext="Procced to Pay"
  data-name="Covid Preventocity"
  data-description="VBA Payment"
  data-image=""
  data-prefill.name=""
  data-prefill.email="<?php echo $r5['ctemail'];?>"
  data-theme.color="darkblue"
></script>
<input type="hidden" name="bkid" value="<?php echo $l; ?>">
<input type="hidden" custom="Hidden Element" name="hidden">
</form>

<?php
} else {
  if (headers_sent()) {
    die('<script type="text/javascript">window.location.href="login.php?e=1"</script>');
  } else {
    header("location:login.php?e=1");
    die();
  }
}
?>

```

contpdf.php

```

<?php
session_start();

include('connection.php');
include('pdf_table.php');
if(isset($_POST['generatepdf']))
{
  // $eid=$_POST['eid'];
  $pid=$_POST['rid'];

  $result = mysqli_query($conn," SELECT
tbl_containmentzone.zone_id,
tbl_containmentzone.cont_status,
tbl_containmentzone.cont_declaredate,tbl_contdet.* ,
tbl_wards.ward_id,
tbl_wards.ward_mem_name,tbl_wards.ward_mem_num,

```

```
tbl_wards.ward_num,tbl_panchayath.panch_name
FROM tbl_wards
JOIN tbl_panchayath
ON tbl_wards.panch_id = tbl_panchayath.panch_id AND tbl_wards.panch_id='$pid'
JOIN tbl_containmentzone
ON tbl_containmentzone.ward_id = tbl_wards.ward_id
JOIN tbl_contdet
ON tbl_containmentzone.zone_id=tbl_contdet.zone_id") or die(mysql_error($con));
$pdf = new PDF_MC_TABLE();
$pdf->AddPage();

$pdf->SetFont('Arial', 'B', 15);
$pdf->Cell(176, 5, 'ContainmentZone Details', 0, 0, 'C');
$pdf->Ln();
$pdf->Ln();
$pdf->Ln();

$pdf->SetFont('Arial', "", 10);

$pdf->SetWidths(Array(35,30,30,30,30));

$pdf->SetLineHeight(5);

$pdf->SetFont('Arial', 'B', 10);

$pdf->Cell(35,5, "Panchayath Name", 1, 0);
$pdf->Cell(30,5, "Ward Number", 1, 0);
$pdf->Cell(30,5, "No of Positives", 1, 0);
$pdf->Cell(30,5, "No of Contacts", 1, 0);
$pdf->Cell(30,5, "No of Deaths", 1, 0);

$pdf->Ln();

$pdf->SetFont('Arial', "", 10);

foreach($result as $row) {
    $pdf->Row(Array(
        $row['panch_name'],
        $row['ward_num'],
        $row['no_of_pstvs'],

        $row['no_of_contacts'],
        $row['no_of_deaths'],

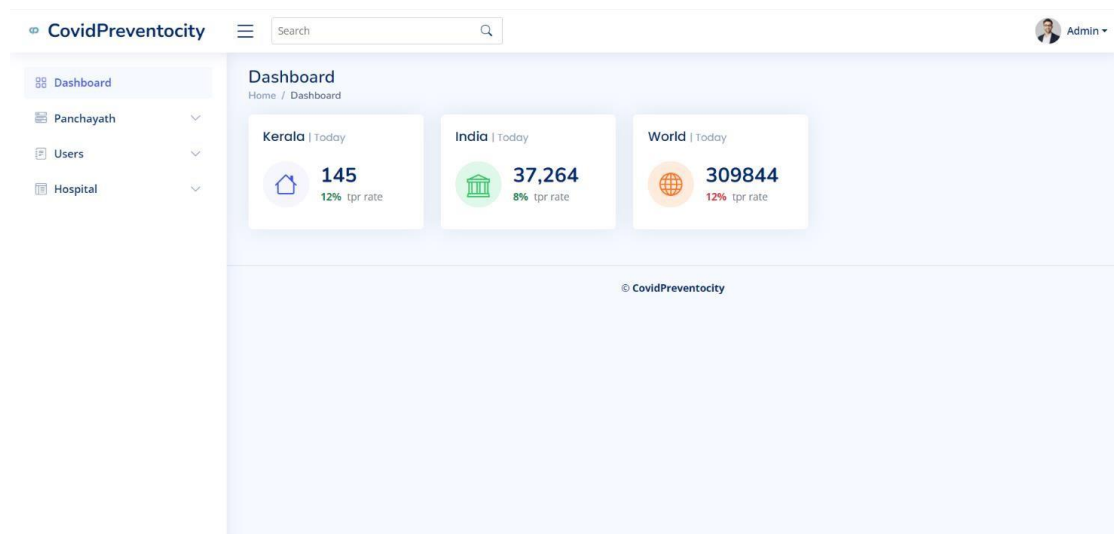
    ));
}
$pdf->Output();
?>
```

9.2 Screen Shots

Home page



Admin Home



User Containment zone view

CovidPreventocity

[Home](#) [Vaccine](#) [Appointments](#) [Services](#) [CovidTest](#) [Anu](#) [LOGOUT](#)

Containment Zones

Select Panchayath Name:

[Search](#) [Reset](#)

Ward Number	Panchayath Name	Containment Status	Declare Date	Number of Positives
2	Kollam	Active	2022-07-07	2
1	Kollam	Active	2022-07-07	2

User Vaccine Booking

CovidPreventocity

[Home](#) [Vaccine](#) [Appointments](#) [Services](#) [CovidTest](#) [Anu](#) [LOGOUT](#)

Vaccine

Your Name

Anu

Age

23

Adhar Number

985674126354

Dose: 1

[Register](#) [Reset](#)

Doctor Appointment Status

CovidPreventocity

Home Vaccine Appointments Services CovidTest Anu LOGOUT

Appointments

Name	Age	Doctor	Date	Appointment Status	Consult Status
Anu	23	Anu Mathew	2022-07-07	Accepted	Consult Completed
Riya	20	Sudheer	2022-07-08	Accepted	Consult Completed
Reena	20	Sudheer	2022-07-08	Pending	Pending

Covid Preventocity

MCA
Amal Jyothi
Kanjirappally

Phone: 123333
Email: info@example.com

Hospital Vaccine View

Covid Preventocity

Home Vaccine Available Vaccine Booking Vaccine Send Request Vaccine Scheduled MIMS LOGOUT

Vaccine

Vaccine Name	Number of Vials	Available Dose Count		
Covishield	41	410	Request For Vaccine	Schedule Vaccine
Covaxin	25	250	Request For Vaccine	Schedule Vaccine

Covid Preventocity

Upload Positive Result By Hospitals

Covid Preventocity
Home Appointments Covid Patients Doctor Services MIMS LOGOUT

Result Uploaded To Panchayath

Name	Panchayath	Contact Number
Girija	Kollam	7485963625
Soja	Kollam	7485963625
Vismaya	Kollam	9685741425
Teena	Kollam	8974563652
Silja	Pinarayi	7788996655
Sreya	Kollam	9685741425
Sangeetha	Kollam	7896541236

Hospital Covid test Page

Covid Preventocity
Home Appointments Covid Patients Doctor Services MIMS LOGOUT

Name	Age	Adhar	Panchayath	Email	Contact Number	Status	Result	Payment Status	
Girija	20	748596362514	Kollam	loo@gmail.com	7485963625	Tested	Positive	Paid	<button>Sent Email</button>
Soja	20	748596362514	Kollam	soja@gmail.com	7485963625	Tested	Positive	Paid	<button>Sent Email</button>
Huy	55	857496362514	Kollam	sd@gmail.com	9874563214	Tested	Negative	Paid	
Sreya	20	748596362544	Kollam	sreyabharath99@gmail.com	9685741425	Tested	Positive	Paid	<button>Sent Email</button>
Silja	22	859674142536	Pinarayi	fff@gmail.com	7788996655	Tested	Pending	Paid	<button>Result Upload</button>
Sangeetha	20	748596362541	Kollam	sangeethasebastian384@gmail.co	7896541236	Tested	Positive	Paid	<button>Sent Email</button>

Hospital Doctor Schedule

Covid Preventocity [Home](#) [Appointments](#) [Covid Patients](#) [Doctor](#) [Services](#) [MIMS](#) [LOGOUT](#)

Select Date

[Search](#) [Reset](#)

Name	Specialization	Departments	
Sudheer	MBBS	Physician	Schedule
Anu Mathew	MD	Psychologist	Schedule
Jiya Mathew	MBBS	Psychologist	Schedule

[↑](#)

Panchayath Containment zone Page

localhost/CovidPt/viewward.php

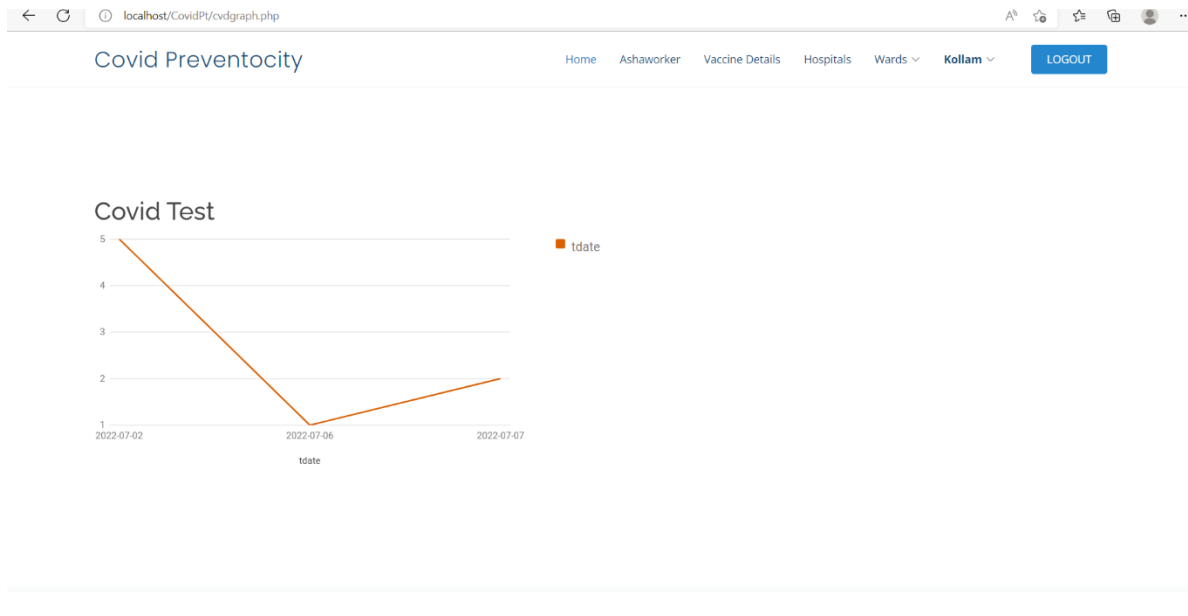
Covid Preventocity [Home](#) [Ashaworker](#) [Vaccine Details](#) [Hospitals](#) [Wards](#) [Kollam](#) [LOGOUT](#)

[Ward Report](#)

WARDS

Ward Number	Member Name	Member Number	Member Email	
1	lintu	9087654323	lintu@gmail.com	Containment Zone
2	Eran	9078563412	eran@gmail.com	Containment Zone
3	yuva	90090090090	yuva@gmail.com	Add to Containment
4	Jisha	9876564567	jisha@gmail.com	Add to Containment
5	Jeena	9685741452	jeena@gmail.com	Add to Containment
6	Sreeya	9685741452	sreeya@gmail.com	Add to Containment

Covid Test Status Graphically Represenatation



Panchayath Search Positives from Hospital by Date

The screenshot shows the 'Covid Preventocity' application interface. The browser address bar indicates the URL is `localhost/CovidPt/panchhospstvshear.php?from_date=2022-07-02&to_date=2022-07-05&eid=1`. The application header includes navigation links: Home, Ashaworker, Vaccine Details, Hospitals, Wards, Kollam, and a LOGOUT button. The main content area displays the 'SEARCH RESULT MIMS HOSPITAL' section. It includes a search form with 'From Date' (02-07-2022) and 'To Date' (05-07-2022) fields, and a 'Click to Search' button. Below the search form, a table displays the search results.

Ward Number	Positive Name	Contact number	Date
1	Girija	7485963625	2022-07-02
1	Soja	7485963625	2022-07-03
1	Sreya	9685741425	2022-07-02

Covid test payment

localhost/CovidPt/paymentgateway.php?uid=12

Test Mode

Covid Preventocity

VBA Payment

₹ 500

English

Country

+91

Phone Number

9874563241

Email

anu@gmail.com

This payment is secured by Razorpay.

PROCEED

localhost/CovidPt/bill.php?id=12

Covid Preventocity

covidpreventocity@gmail.com
Covid Preventocity
+989 345 6789

Invoice
Order Date : 19-7-2022

Covid Test

Name	Testtype	Amount	
Anu	RTPCR	500	
		Grand Total	500

Payment Status: Paid
Payment Mode: Online Payment
Payment Date:2019-07-22 10:09:13

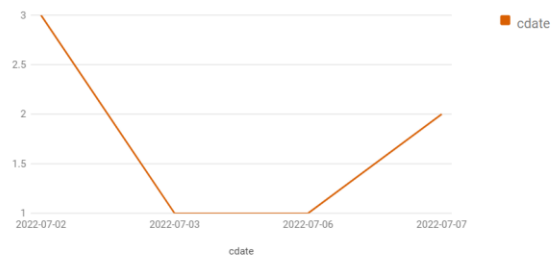
Thank You for Visiting...Click Here to go Home

Positive rate in panchayath

Covid Preventocity

[Home](#)[Ashaworker](#)[Vaccine Details](#)[Hospitals](#)[Wards](#)[Kollam](#)[LOGOUT](#)

Positive Rate



Vaccination certificate



Ministry of Health & Family Welfare
Government of India

Certificate for COVID-19 Vaccination
Issued in India by Ministry of Health & Family Welfare, Govt. of India

Beneficiary Details

Name : Anu

Age : 23

Adhar : 447788665588

Vaccination Details

Vaccine Name : Covishield

Vaccine Dose : 1

Vaccinated Date : 2022-07-07

Vaccinated Place : MIMS

Covid Test Result**Covid Test Result****Patient Information**

Name : Girija

Age : 20

Mobile Number : 7485963625

Specimen Information

Collected at : MIMS

Specimen Collected : Nasopharyngeal Swab

Test Type:	RTPCR
Result :	Positive

INTERPRETATION :

Negative : RNA specific to SARS-CoV-2 not detected

Positive : RNA specific to SARS-CoV-2 detected

Invalid : Invalid results are obtained if the internal controls do not amplify.

SPECIFICITY :

This test detects the 2019 novel coronavirus strain (SARS - CoV- 2) . It was shown by direct testing or in silico analysis to not cross-react with a large number of other bacteria and viruses.

Authorized by:



Dr. Anisha Asokan M.D.(Microbiology)
HOD Molecular Biology