# COVID19 Testing Management System

## Project Report

## Index



## Acknowledgement


I would like to express my deepest appreciation to all those who provided me the possibility to complete this project. A special gratitude I give to our final year project manager, Dr. Alok and Ravindera Sir , whose contribution in stimulating suggestions and encouragement, helped me to coordinate my project.

Furthermore I would also like to acknowledge with much appreciation the crucial role of the staff, who gave the permission to use all required equipment and the necessary materials to complete project. Last but not least, many thanks go to the head of the project, **Dr. Alok Mathur** whose have invested his full effort in guiding the team in achieving the goal. I have to appreciate the guidance given by other supervisor as well as the panels especially in our project presentation that has improved our presentation skills thanks to their comment and advices.


---

# Abstract

Nowadays, **COVID19 Testing Management System** is one of the most essential tools that are mostly used in Testing Lab; it is mostly used to manage COVID19 medical lab related activities.

In this project we tried to develop a computerized and web based COVID19 Testing management system. Our main intention is to allow this application to be used in most retailing COVID19 lab, where a small point of customization will be required to each COVID19 lab in the implementation period. This system is designed to overcome all challenges related to the management of diagnostic that were used to be handled locally and manually.

The system is an online COVID19 lab manager application that brings up various COVID19 test working online. Using this system, it will help us to records all transaction made at the daily tests; recognize all customers, employees, etc. It will manage all activities around the COVID19 lab that increases productivity and maximize profit, it will also minimizing the risk of getting loss because all transactions are recorded to the system.

# Introduction

COVID19 Testing Management System is web based technology which brings up various diagnosis works online. Here patients are first allowed to register on the website and provide personal, test information. Once registered with their address and contact details, the patients may now see a variety of tests conducted by the lab. The patient will select the required test and book appointment after that lab center send a lab boy at registered address to collect a sample. After successful sample collection patient can track their test history using the name, order and registered mobile number. The system allows admin to attach a copy of the report into the system and automatically shown on user side so user can downloads report.

In COVID19 Testing Management System we use PHP and MySQL database. It has three modules i.e.

1. **Admin**
2. **User (Patient)**

## Admin Module

Admin is the super user of the website who can manage everything on the website. Admin can log in through the login page

- **Dashboard:** In this section, the admin can see all detail in brief like the total, assigned and the sample collected and completed tests.
- **Phlebotomist:** In this section, the admin can manage Phlebotomist (add, update, delete).
- **Testing**: In this section, the admin can manage all the tests like assign the test to Phlebotomist and update the history.

- **Report:** In this section, the admin can generate two types of report. One is between dates reports and another one is by search. Admin can search the report by order number, name and mobile number.
- **Notification:** In this section, the admin will get a notification for every new test request (notification bell).
- Admin can also update his profile, change the password and recover the password.

### User (Patient) Module

- User can visit the application through a URL.
- **Testing:** This section divided into two parts. One is for new user and another one is for registered user. New user (First-time user) needs to provide personal and testing Information. A registered user only needs to provide test information; their personal information will be fetched from the database.
- **Report:** In this section, Users can search their test report using order number, name and registered mobile number.
- **Dashboard:** In this section, the User can see the in which State of how many tests are done.

## Purpose

The main purpose of COVID19 Testing management system to provide a platform where patients can book the test online and get their COVID19 test done at home. With the help of this project we are bringing the use of technology in the field of medical diagnosis where patients can avail all the diagnosis facilities at their door steps. Another purpose for developing this application is to generate the report automatically.

## Scope

Today also we have to go to the COVID19 Test Lab center, wait in the queue to get our COVID19 test done. As Technology is growing rapidly we are also moving to a technical world where everything we want to be online. So with the help of this project we are bringing the use of technology in the field of medical diagnosis where patients can avail all the diagnosis facilities at their door steps. This project makes the diagnosis process easy and reduces the burden of patients. At a same time its help the diagnostic center to track all their patients details with their test reports. This access friendly software provides quick and effective services which helps the diagnostic center to increase their sales and profit.

**Advantages:**

- The system allows automate diagnosis system.
- Allows for faster service.
- Allows increased sales and profits for diagnostic labs.

- Easy, user friendly GUI.
- Validation of data will be ensure only accurate valid and complete data stored in the database.
- Easy retrieval or data will be made possible by finding techniques.
- Report generation will help made it easy to analyze the performance.

**Disadvantages:**

- It reduces employment as the human efforts are being automated by this system.

## Requirement Specification

**Hardware Configuration:**

**Client Side:**

| | |
|---|---|
| **RAM** | 512 MB |
| | |
| **Hard disk** | 10 GB |
| | |
| **Processor** | 1.0 GHz |
| | |

**Server side:**

| | |
|---|---|
| **RAM** | **1 GB** |
| **Hard disk** | **20 GB** |
| **Processor** | **2.0 GHz** |

### Software Requirement:

**Client Side:**

| | |
|---|---|
| **Web Browser** | Google Chrome or any compatible browser |
| **Operating System** | Windows or any equivalent OS |

**Server Side:**

| | |
|---|---|
| **Web Server** | APACHE |
| **Server side Language** | PHP5.6 or above version |
| **Database Server** | MYSQL |
| **Web Browser** | Google Chrome or any compatible browser |
| **Operating System** | Windows or any equivalent OS |

## APACHE

The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.

The Apache HTTP Server ("httpd") was launched in 1995 and it has been the most popular web server on the Internet since April 1996. It has celebrated its 20th birthday as a project in February 2015.

## PHP

- PHP stands for PHP: Hypertext Preprocessor.
- PHP is a server-side scripting language, like ASP.
- PHP scripts are executed on the server.
- PHP supports many databases (MYSQL, Informix, Oracle, Sybase, Solid, Generic ODBC, etc.).
- PHP is an open source software.
- PHP is free to download and use.

## MYSQL

- MYSQL is a database server
- MYSQL is ideal for both small and large applications
- MYSQL supports standard SQL
- MYSQL compiles on a number of platforms
- MYSQL is free to download and use

- How to access MySQL:
- **http://localhost/phpmyadmin**

**Analysis and Design**

**Analysis:**

Today also we have to go to the diagnostic center, wait in the queue to get our COVID19 test done. As Technology is growing rapidly we are also moving to a technical world where everything we want to be online. So with the help of this project we are bringing the use of technology in the field of medical diagnosis where patients can avail all the diagnosis facilities at their door steps. This project makes the diagnosis process easy and reduces the burden of patients. At a same time its help the diagnostic center to track all their patients details with their test reports.

**Disadvantage of present system:**

- **Not user friendly:** The present system not user friendly because data is not stored in structure and proper format.
- **Manual Control:** All report calculation is done manually so there is a chance of error.
- **Lots of paper work:** Visitors maintain in the register so lots of paper require storing details.
- **Time consuming**

## Design Introduction:

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization. Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

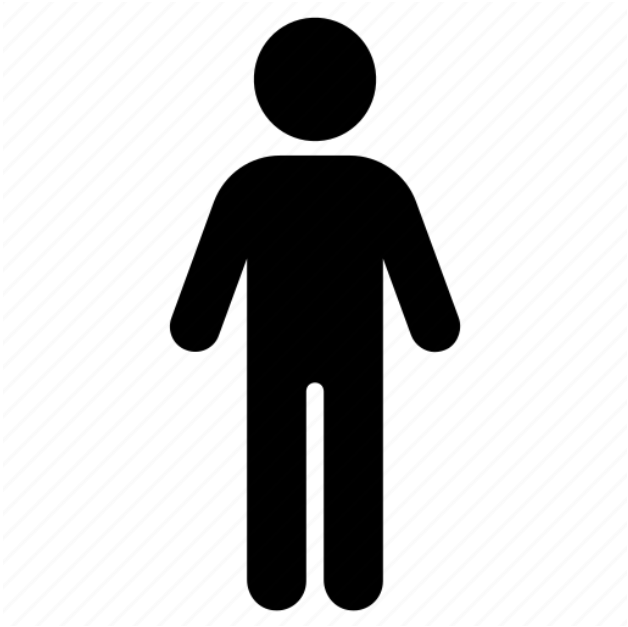
The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data

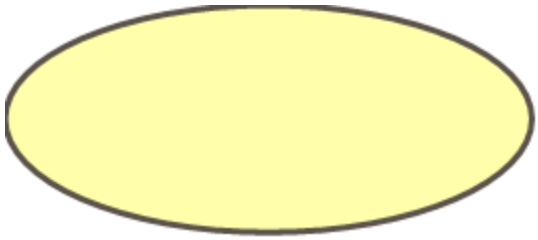
**UML Diagrams:**

Actor:

A coherent set of roles that users of use cases play when interacting with the use `cases.



Use case: A description of sequence of actions, including variants, that a system performs that yields an observable result of value of an actor.



UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

## USECASE DIAGRAMS:

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what's called an actor.

Use case diagram can be useful for getting an overall view of the system and clarifying that can do and more importantly what they can't do.

Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

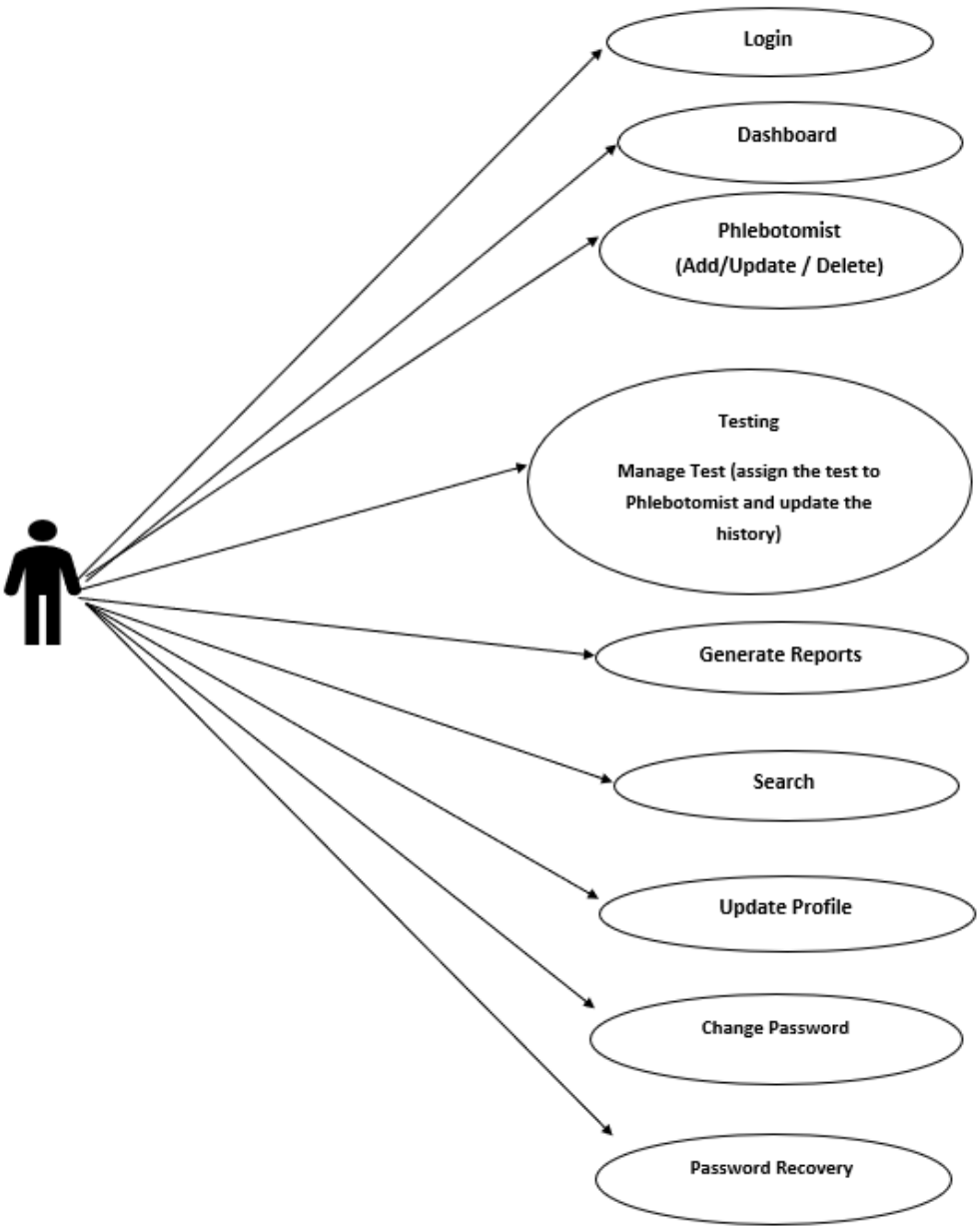
- The purpose is to show the interactions between the use case and actor.
- To represent the system requirements from user's perspective.

- An actor could be the end-user of the system or an external system.

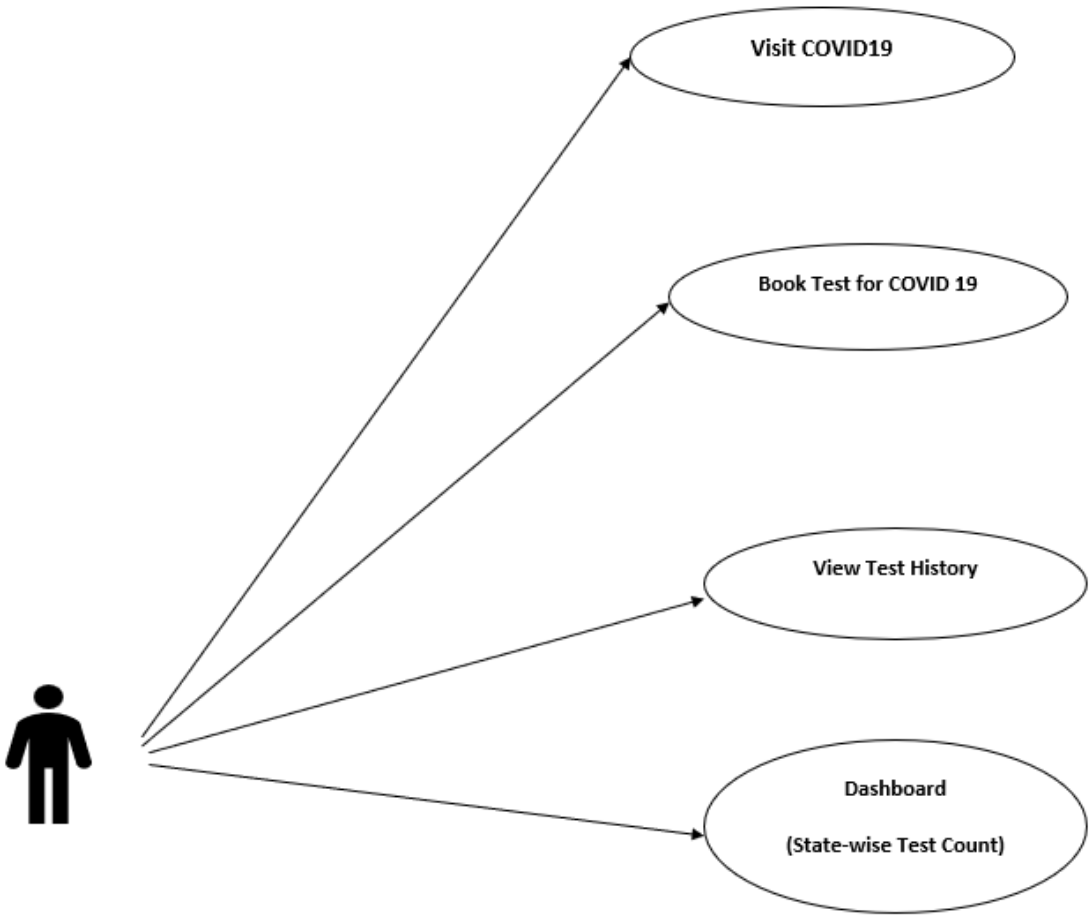
**USECASE DIAGRAM:** A Use case is a description of set of sequence of actions. Graphically it is rendered as an ellipse with solid line including only its name. Use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationship. It is an association between the use cases and actors. An actor represents a real-world object. Primary Actor – Sender, Secondary Actor Receiver.

**Use Case Diagrams:**

**Admin**

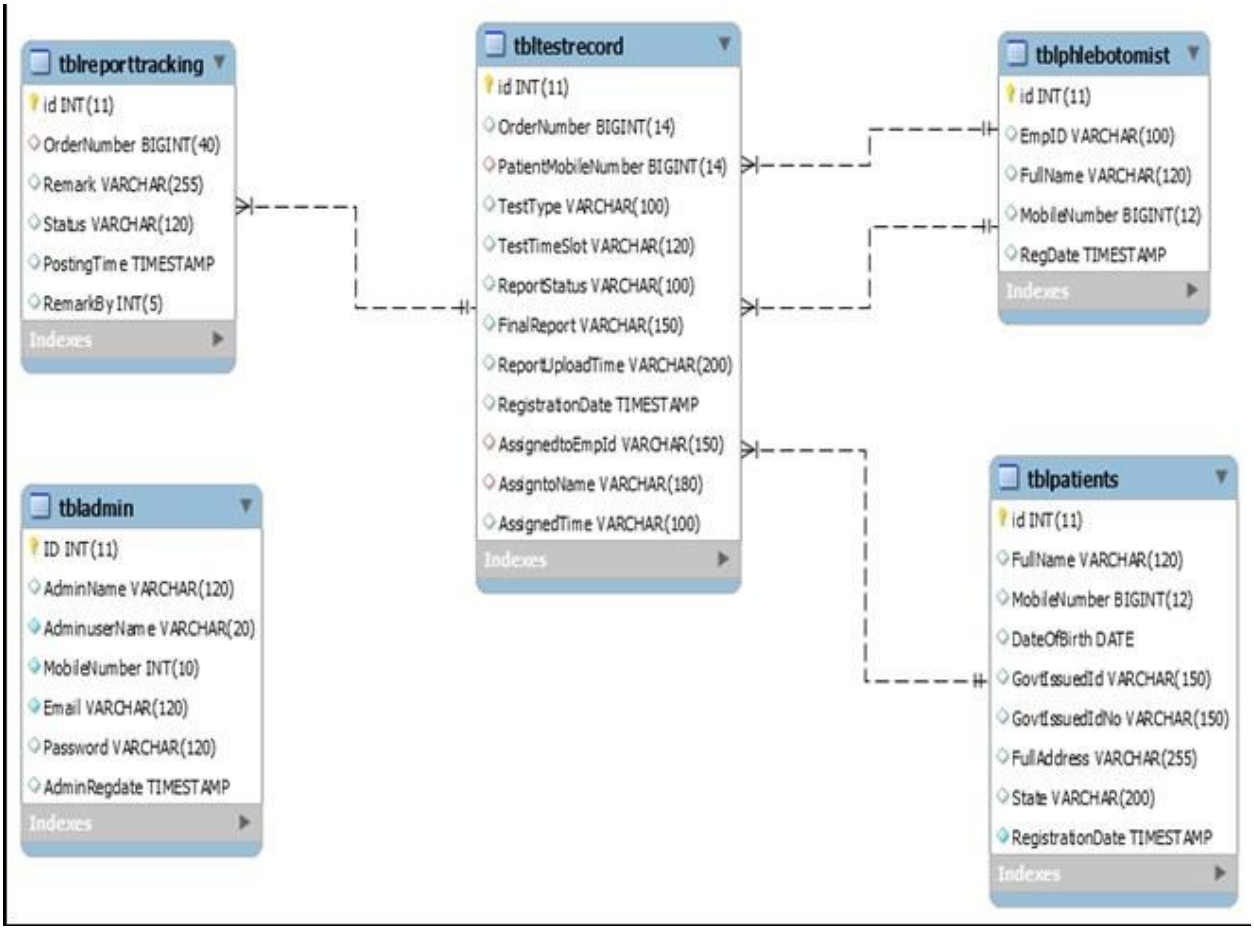



**Users (Patient)**




## Class Diagram:

A description of set of objects that share the same attributes operations, relationships, and semantics




## ER Diagram:

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:

- It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
- It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
- In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.

## ER Notation

There is no standard for representing data objects in ER diagrams. Each modeling methodology uses its own notation. The original notation used by Chen is widely used in academics texts and journals but

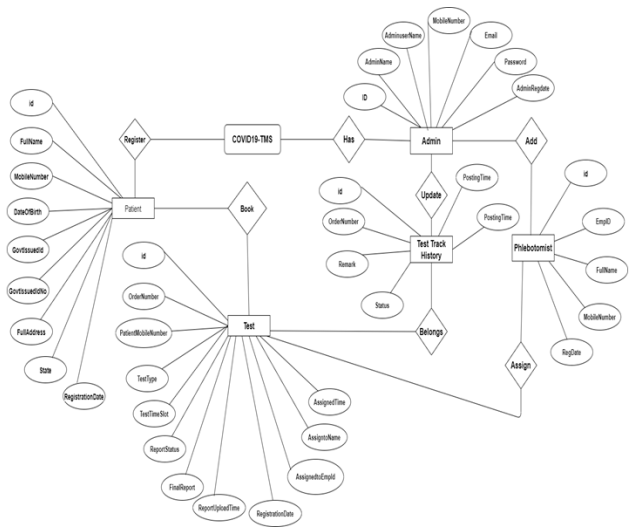
rarely seen in either CASE tools or publications by non-academics. Today, there are a number of notations used; among the more common are Bachman, crow's foot, and IDEFIX.

All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection. The notation used in this document is from Martin. The symbols used for the basic ER constructs are:

- **Entities** are represented by labeled rectangles. The label is the name of the entity. Entity names should be singular nouns.
- **Relationships** are represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs
- **Attributes**, when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined. Attribute names should be singular nouns.
- **Cardinality** of many is represented by a line ending in a crow's foot. If the crow's foot is omitted, the cardinality is one.

**Existence** is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity that is optional.

### ER Diagram




## MySQL Data Tables

**Admin Table:**(Table name is tbladmin)

This store admin personal and login details.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|---|---|---|---|---|---|---|---|---|
| 1 | ID | int(11) | | | No | None | | AUTO_INCREMENT |
| 2 | AdminName | varchar(120) | latin1_swedish_ci | | Yes | NULL | | |
| 3 | AdminuserName | varchar(20) | latin1_swedish_ci | | No | None | | |
| 4 | MobileNumber | int(10) | | | No | None | | |
| 5 | Email | varchar(120) | latin1_swedish_ci | | No | None | | |
| 6 | Password | varchar(120) | latin1_swedish_ci | | Yes | NULL | | |
| 7 | AdminRegdate | timestamp | | | Yes | current_timestamp() | | |

Indexes

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | ID | 0 | A | No | |

## tblpatients

This table store the data of patient personal Information.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|---|---|---|---|---|---|---|---|---|
| 1 | id | int(11) | | | No | None | | AUTO_INCREMENT |
| 2 | FullName | varchar(120) | latin1_swedish_ci | | Yes | NULL | | |
| 3 | MobileNumber | bigint(12) | | | Yes | NULL | | |
| 4 | DateOfBirth | date | | | Yes | NULL | | |
| 5 | GovtIssuedId | varchar(150) | latin1_swedish_ci | | Yes | NULL | | |
| 6 | GovtIssuedIdNo | varchar(150) | latin1_swedish_ci | | Yes | NULL | | |
| 7 | FullAddress | varchar(255) | latin1_swedish_ci | | Yes | NULL | | |
| 8 | State | varchar(200) | latin1_swedish_ci | | Yes | NULL | | |
| 9 | RegistrationDate | timestamp | | | No | current_timestamp() | | |

Indexes

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 5 | A | No | |
| MobileNumber | BTREE | No | No | MobileNumber | 5 | A | Yes | |

## tbltestrecord

This table stores the patient test record details.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|---|---|---|---|---|---|---|---|---|
| 1 | id | int(11) | | | No | None | | AUTO_INCREMENT |
| 2 | OrderNumber | bigint(14) | | | Yes | NULL | | |
| 3 | PatientMobileNumber | bigint(14) | | | Yes | NULL | | |
| 4 | TestType | varchar(100) | latin1_swedish_ci | | Yes | NULL | | |
| 5 | TestTimeSlot | varchar(120) | latin1_swedish_ci | | Yes | NULL | | |
| 6 | ReportStatus | varchar(100) | latin1_swedish_ci | | Yes | NULL | | |
| 7 | FinalReport | varchar(150) | latin1_swedish_ci | | Yes | NULL | | |
| 8 | ReportUploadTime | varchar(200) | latin1_swedish_ci | | Yes | NULL | | |
| 9 | RegistrationDate | timestamp | | | Yes | current_timestamp() | | |
| 10 | AssignedtoEmpId | varchar(150) | latin1_swedish_ci | | Yes | NULL | | |
| 11 | AssigntoName | varchar(180) | latin1_swedish_ci | | Yes | NULL | | |
| 12 | AssignedTime | varchar(100) | latin1_swedish_ci | | Yes | NULL | | |

Indexes

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 6 | A | No | |
| OrderNumber | BTREE | No | No | OrderNumber | 6 | A | Yes | |

## tblreporttracking

This table stores the tracking  details of tests.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|---|---|---|---|---|---|---|---|---|
| 1 | id | int(11) | | | No | None | | AUTO_INCREMENT |
| 2 | OrderNumber | bigint(40) | | | Yes | NULL | | |
| 3 | Remark | varchar(255) | latin1_swedish_ci | | Yes | NULL | | |
| 4 | Status | varchar(120) | latin1_swedish_ci | | Yes | NULL | | |
| 5 | PostingTime | timestamp | | | Yes | current_timestamp() | | |
| 6 | RemarkBy | int(5) | | | Yes | NULL | | |

Indexes

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 12 | A | No | |

## tblphlebotomist

This table stores the phlebotomist information.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra |
|---|---|---|---|---|---|---|---|---|
| 1 | id | int(11) | | | No | None | | AUTO_INCREMENT |
| 2 | EmpID | varchar(100) | latin1_swedish_ci | | Yes | NULL | | |
| 3 | FullName | varchar(120) | latin1_swedish_ci | | Yes | NULL | | |
| 4 | MobileNumber | bigint(12) | | | Yes | NULL | | |
| 5 | RegDate | timestamp | | | Yes | current_timestamp() | | |

Indexes

| Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|---|---|---|---|---|---|---|---|---|
| PRIMARY | BTREE | Yes | No | id | 3 | A | No | |
| EmpID | BTREE | No | No | EmpID | 3 | A | Yes | |
| FullName | BTREE | No | No | FullName | 3 | A | Yes | |

# Implementation and System Testing

After all phase have been perfectly done, the system will be implemented to the server and the system can be used.

## System Testing

The goal of the system testing process was to determine all faults in our project .The program was subjected to a set of test inputs and many explanations were made and based on these explanations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing

1. Unit testing

2. Integration testing

# UNIT TESTING

Unit testing is commenced when a unit has been created and effectively reviewed .In order to test a single module we need to provide a complete environment i.e. besides the section we would require

- The procedures belonging to other units that the unit under test calls
- Non local data structures that module accesses
- A procedure to call the functions of the unit under test with appropriate parameters

## 1. Test for the admin module

- **Testing admin login form-**This form is used for log in of administrator of the system. In this form we enter the username and password if both are correct administration page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask the details.
- **Report Generation:** admin can generate report from the main database.

# INTEGRATION TESTING

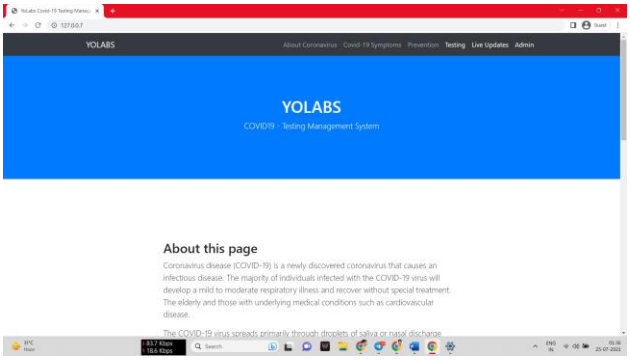
In the Integration testing we test various combination of the project module by providing the input.

The primary objective is to test the module interfaces in order to confirm that no errors are occurring when one module invokes the other module.

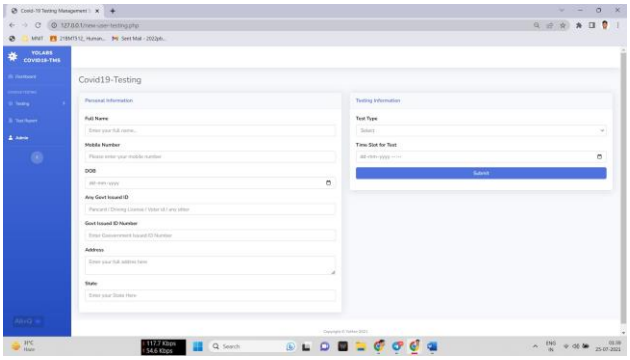
# Evaluation

**Project URL: http://localhost/**

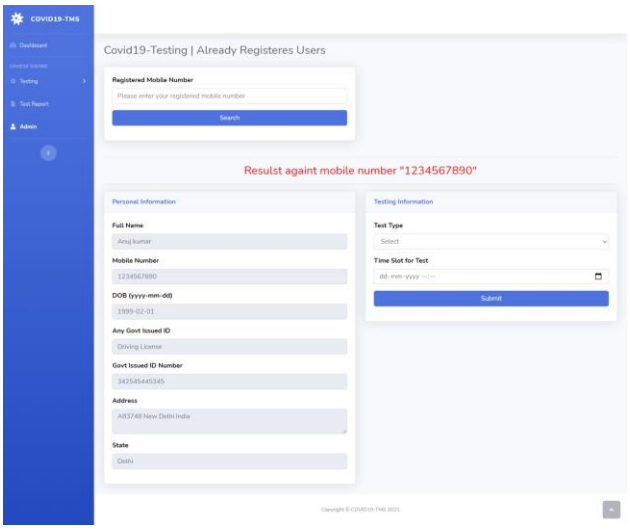
**Home Page**



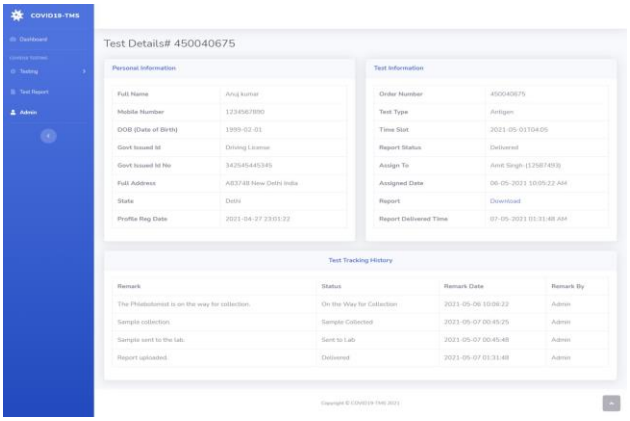

## New User (Patient) Test Booking



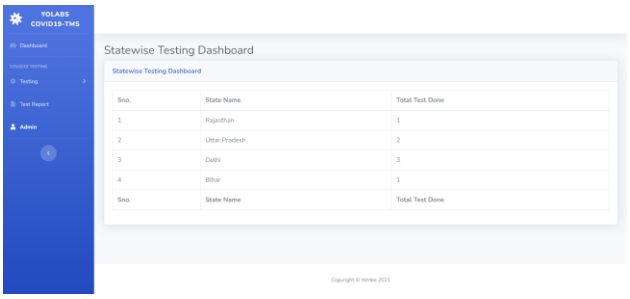

## Already Registered User (Patient) Test Booking



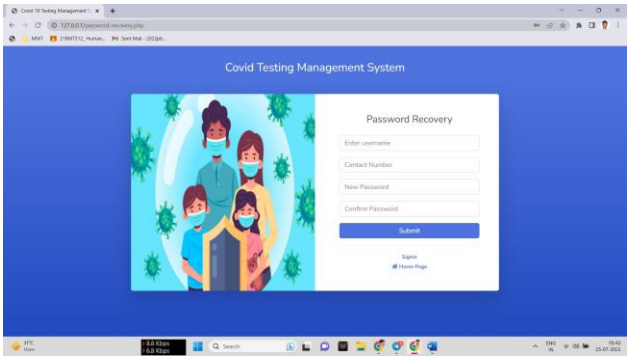
## User (Patient) Test Details



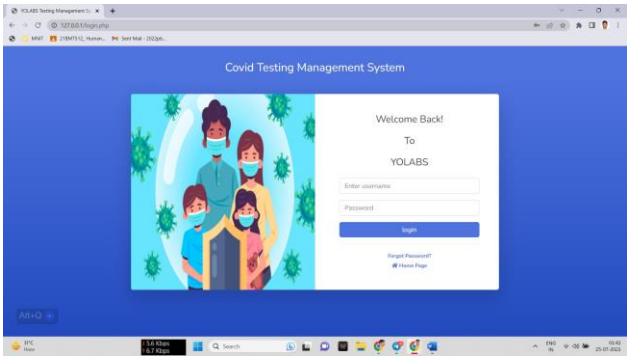

**State Wise Dashboard**



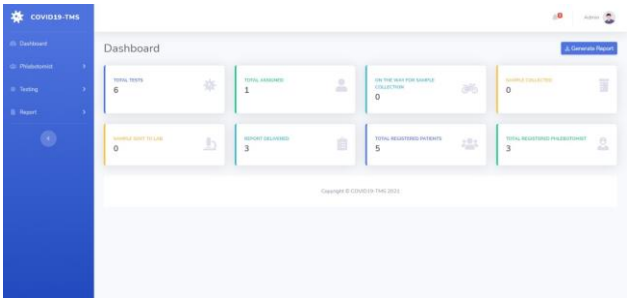

## Admin Login



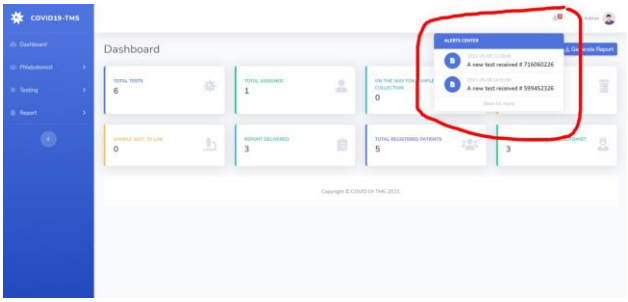

## Admin Password Recovery



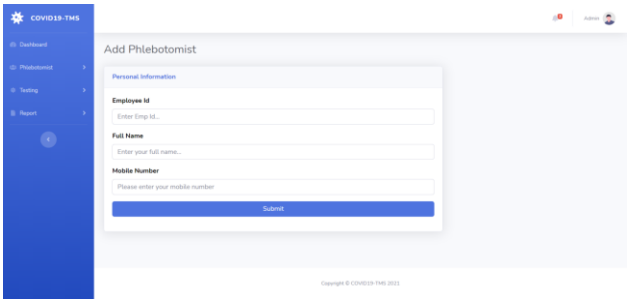
## Admin Dashboard



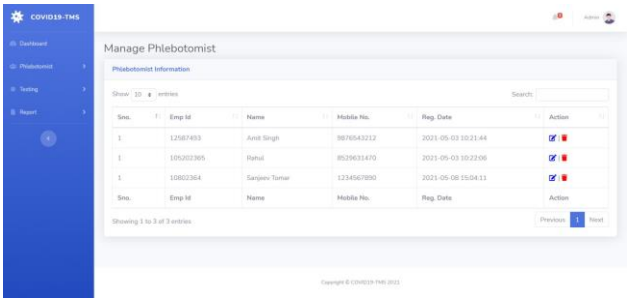

## Admin Notification



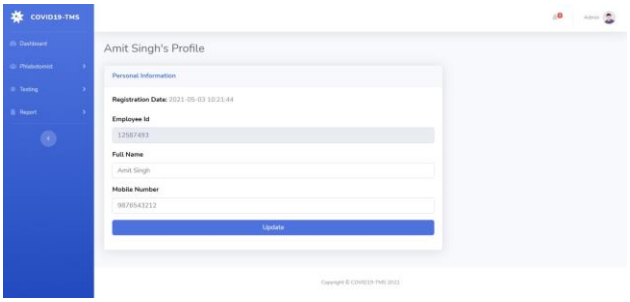

## Add Phlebotomist



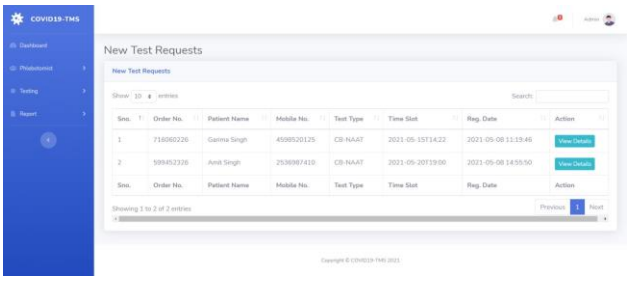

## Manage Phlebotomist



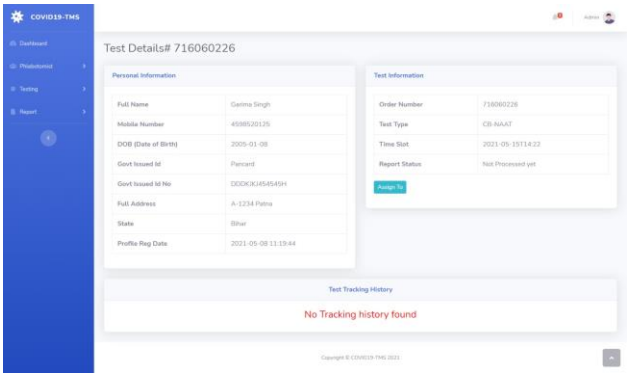
## Edit/Update Phlebotomist Information



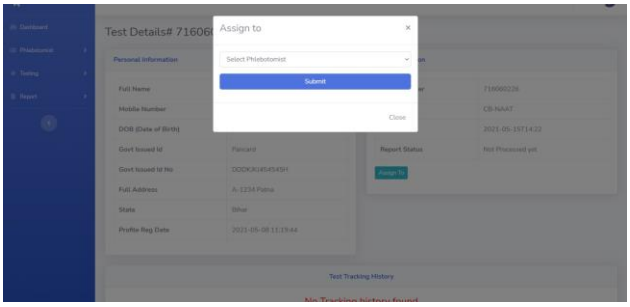

**New/Assigned/On the way for collection/Sample Collected /Sent to Lab / Delivered / All Tests**



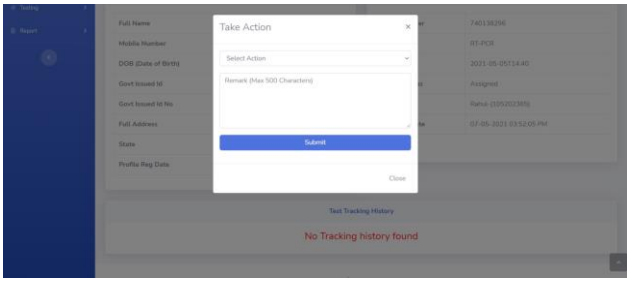

## Test Details-1



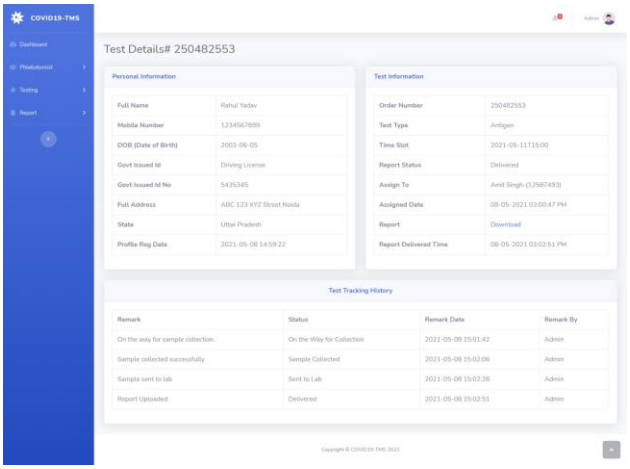

**Assigned to**



## Take Action

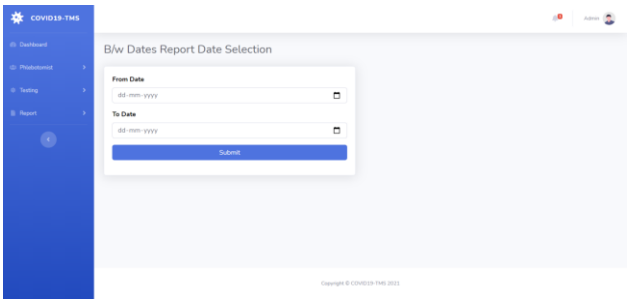



**Test Details Admin**

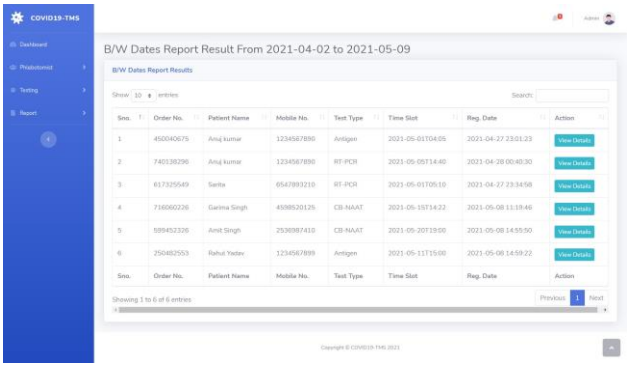



**Reports**

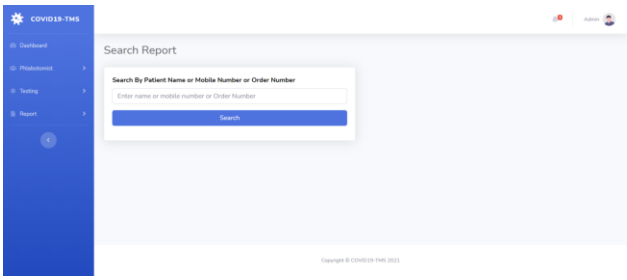
**B/w Dates Report Date Selection**



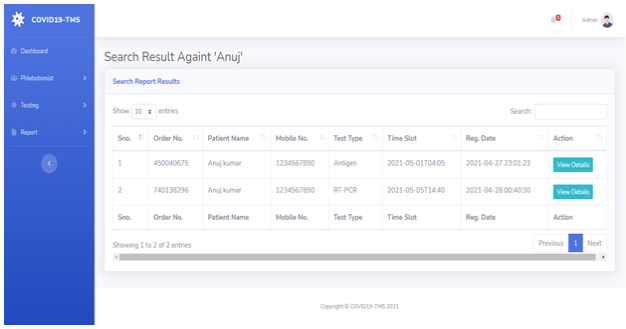

**B/w Dates Test Result**



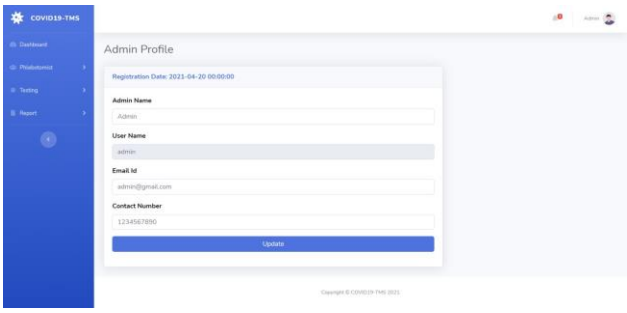

## Search Report



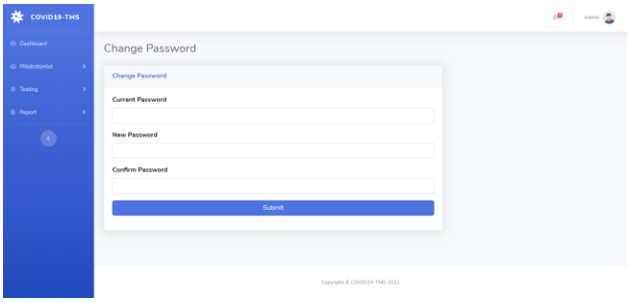

## Search Report Result



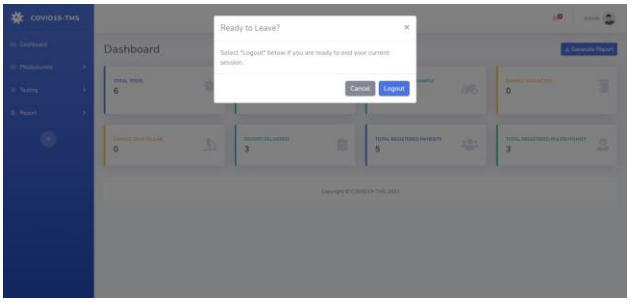

## Admin Profile



### Admin Change Password




### Admin Logout




## Conclusion

COVID19 Testing Management System is very much graceful and lively. Patients have to register to the portal by giving their details and then they can take appointment through online with minimal effort. The Phlebotomist comes to patient address to collect the sample. Once test is done and test report is generated patient can download the report by logged in to the portal. This system can be implemented in diagnostic labs and clinics.

- Automation of the entire system improves the productivity.
- It provides a friendly graphical user interface which proves to be better when compared to the existing system.
- It gives appropriate access to the authorized users depending on their permissions.
- It effectively overcomes the delay in communications.
- Updating of information becomes so easier.
- System security, data security and reliability are the striking features.
- The System has adequate scope for modification in future if it is necessary.

## References

### For PHP


- https://www.w3schools.com/php/default.asp

- https://www.sitepoint.com/php/
- https://www.php.net/

### For MySQL

- https://www.mysql.com/
- http://www.mysqltutorial.org

### For XAMPP

- https://www.apachefriends.org/download.html