**ELAAJ A COVID PORTAL**

**A PROJECT REPORT**

**Submitted By**

**SHALINI TOMAR**

**University Roll No 1900290149088**

**Submitted in partial fulfillment of the**

**Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATION**

**Under the Supervision of**

**Mr. Ankit Verma**

### ASSISTANT PROFESSOR



**Submitted to**

**Department Of Computer Application**

**DR. A. P. J ABDUL KALAM TECHNICAL UNIVERSITY**

**Lucknow**

**August 2021**

**DECLARATION**

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to substantial extent has been accepted for the award of any degree or diploma of the university or another institute of higher learning except where due acknowledgment has been made in the text.

**Name: Shalini Tomar**

**Roll No.: 1900290149088**

**Signature** **:** 

**CERTIFICATE**

Certified that **Shalini Tomar (University Roll No 1900290149088)** have carried out the project work having **Elaaj A Covid Portal** for Master of Computer Application from Dr. A.P.J. Abdul Kalam Technical University (AKTU**)** (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself and the contents of the project report do not form the basis for the award ofany other degree to the candidate or to anybody else from this or any other University/Institution.

**Shalini Tomar**

**University Roll No**

**1900290149088**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

**Mr. Ankit Verma**

**Assistant Professor**

**Department of Computer Application**

**KIET Group of Institutions, Ghaziabad**

**Signature of Internal Examiner Signature of External Examiner**

**Dr. Ajay Srivastava**

**Head, Department of Computer Application**

**ABSTRACT**

In the given situation of the current ongoing COVID-19 pandemic, people are suffering from lack of hospital Beds, Oxygen Support, Medicines and other required resources to fight COVID-19, due to spike in daily reported infected cases. Thus, gathering genuine information about resources that help to fight against COVID-19 is becoming a difficult task.

“Elaaj - The COVID Portal” is an online platform which aims to provide one-stop digital healthcare solution to all the issues possible that can be faced in the course of treatment to cure COVID-19.

The main purpose of our portal is to provide genuine required information about the disease and tackling the issue of management of the available resources such as beds, ICUs, Oxygen Cylinders, Plasma Donors and Vaccines, so that the person in need gets the required resource without panicking or worrying about where to get it from

**ACKNOWLEDGEMENT**

It is my pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced our thinking, behavior and acts during the course of study.

We are thankful to **Mr. Ankit Verma** for her support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

We would also like to thank our **H.O.D Mr. Ajay Kumar Srivastava** for his valuable suggestions which helps us lot in completion of this project.

Lastly, we would like to thank the almighty and our parents for their moral support and friends with whom we shared our day-to-day experience and received lots of suggestions that improved our quality of work.

**Shalini Tomar**

**Univ. Roll.No. 1900290149088**

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
|  | Page No. |
|  |  |
| Certificate | ii |
| Abstract | iii |
| Acknowledgement | iv |
| List of Figures and Tables | vii |
|  |  |
| **CHAPTER 1: INTRODUCTION** | **1-3** |
| 1.1 Project Scope | 1 |
| 1.2 Hardware Used in Project | 1 |
| 1.3 Software Used in Project | 2 |
| 1.4 Project Schedule | 3 |
|  |  |
| **CHAPTER 2: LITERATURE REVIEW** | **4-8** |
| 2.1 IntelliJ | 4 |
| 2.2 Spring Boot | 4 |
| 2.3 Architecture of Android | 4 |
| 2.4 The Android Market | 4 |
| 2.5 Obtaining the required tools | 4 |
| 2.2.1 Developer Ergonomics | 4 |
| 2.2.2 Built in Developer Tools | 4 |
| 2.5.3 Creating new Project | 4 |
| 2.3 MySQL Workbench | 7 |
| 2.3.1 Design | 7 |
| 2.3.2 Develop | 7 |
| 2.3.3 Administer | 7 |
| 2.3.4 Visual Performance Dashboard | 8 |
| 2.3.5 Database Migration | 8 |
| **CHAPTER 3: TECHNICAL FEASIBILITY** | **9-11** |
| 3.1 Feasibility Study | 9 |
| 3.2 Technical Feasibility | 9 |
| 3.3 Technology Description | 10 |
| 3.4 Technology Used in the Project | 11 |
|  |  |
| **CHAPTER 4: BACKEND DESIGN** | **12-21** |
| 4.1 Data Dictionary | 12 |
| 4.2 ER Diagram | 12 |
| 4.3 Use Case Diagram | 15 |
| 4.4 Data Flow Diagram | 17 |
| 4.5 Database Design | 18 |
| **CHAPTER 5: TESTING** | **22-24** |
| 5.1 Introduction | 22 |
| 5.2 Types of Testing | 22 |
|  |  |
| **CHAPTER 6: SCREEN SHOTS** | **25-31** |
| **CHAPTER 7: LIMITATIONS** | **32** |
| 7.1 Limitations | 32 |
| 7.2 Future Scope | 32 |
| 7.3 Future Enhancement | 32 |
|  |  |
| **CHAPTER 8: CONCLUSION** | **33** |
|  |  |
| **CHAPTER 9: BIBLIOGRAPHY** | **34** |
|  |  |

**LIST OF FIGURES AND TABLES**

|  |  |
| --- | --- |
| 4.2.1 ER Diagram | 14 |
| 4.3.1 Use Case Diagram for covid portal | 15 |
| 4.3.2 Sequence Diagram | 16 |
| 4.4.1 Admin DFD | 17 |
| 4.4.2 Representative DFD | 17 |
| 4.5.1 Data Model | 18 |
| 4.5.2 Appointment | 18 |
| 4.5.3 Ambulance | 19 |
| 4.5.4 Areas | 19 |
| 4.5.5 Users | 20 |
| 4.5.6 Suppliers | 20 |
| 4.5.7 Hospitals | 21 |
| 6.1 Home Page | 25 |
| 6.2 User Login Page | 27 |
| 6.3 Ambulance Details | 27 |
| 6.4 Supplier’s Details | 28 |
| 6.5 Vaccination Details | 28 |
| 6.6 Representative Homepage | 29 |
| 6.7 Hospital Add/Remove | 29 |
| 6.8 Admin Homepage | 30 |
| 6.9Admin Add/Block Representative | 30 |
| 6.10 User Sign Up | 31 |

**CHAPTER-1**

### INTRODUCTION

### 

In the given situation of the current ongoing COVID-19 pandemic, people are suffering from lack of hospital Beds, Oxygen Support, Medicines and other required resources to fight COVID-19, due to spike in daily reported infected cases. Thus, gathering genuine information about resources that help to fight against COVID-19 is becoming a difficult task.

“Elaaj - The COVID Portal” is an online platform which aims to provide one-stop digital healthcare solution to all the issues possible that can be faced in the course of treatment to cure COVID-19.

The main purpose of our portal is to provide genuine required information about the disease and tackling the issue of management of the available resources such as beds, ICUs, Oxygen Cylinders, Plasma Donors and Vaccines, so that the person in need gets the required resource without panicking or worrying about where to get it from.

· Providing information about COVID-19 symptoms, precautions and treatment procedure.

· Providing information about hospitals in the vicinity with beds and ICU available for patients who need hospitalization on the basis of severity.

· To provide information about vendors providing Oxygen Cylinders and Concentrators if the patient requires oxygen support while at Home Quarantine.

· Providing patients with an Appointment Management Portal to find and fix appointments with the doctors treating COVID-19 in the vicinity.

· Providing a portal to register and schedule an appointment for COVID-19 Vaccination at the vaccination centers in the vicinity.

· Providing a one-stop digital healthcare solution with a multilingual and an easy to navigate UI making it accessible to everyone, including those who lack education or aren’t well technically skilled.

**1.1 Project Scope**

Project scope is the part of project planning that involves determining and documenting a list of specific project goals, deliverables, features, functions, tasks, deadlines, and ultimately costs. In other words, it is what needs to be achieved and the work that must be done to deliver a project.

**1.2 Hardware Used In Project:**

* **Windows 10**
* **500 GB HDD**
* **RAM 8**

**1.3 Software Used in Project**

* **IntelliJ**
* **MySQL**
* **JAVA**

**Database:** MySQL

**Operating System:** Windows7/8/8.1/10

Scope

Followings are the key features that has User, Support Analyst and Admin:

User

1. User Registration and Login

2. User Authentication

3. Search for Hospitals, Ambulance, Supplier's data

4. Apply for COVID-19 test and vaccination

5. View application status

6. View site FAQs and COVID-19 Alerts

Representative

1.Representative login and authentication

2.View previously added Data

3.Add new Data (Hospitals, Ambulances, Suppliers details)

4.Edit new Data

5.Delete new Data

Admin

1.Login and Authentication

2.View Representative’s list

3.Add new Representative

4.Block Representative

5.Delete Data (wrong Hospitals, Ambulances, Suppliers details)

2.Assumptions

Not Applicable.

3.Dependencies

1.  Representative’s account can be created by only Admins.

4.Risks

If there is any failure in applying for vaccination or applying for test then system should able to handle those risks.

If a user has to search Hospital or Ambulance data urgently then system should perform accurately and firmly.

**1.4 Project Schedule**

In order to develop our schedule, we first need to define the activities, sequence them in the right order, estimate the resources needed, and estimate the time it will take to complete the tasks.

**Defining Activities**

The activity definition process is a further breakdown of the work package elements of the

WBS. It documents the specific activities needed to fulfill the deliverables detailed in the WBS. These activities are not the deliverables themselves but the individual units of work that must be completed to fulfill the deliverables. Activity definition uses everything we already know about the project to divide the work into activities that can be estimated. You might want to look at all the lessons learned from similar projects your company has done to get a good idea of what you need to do on the current one**.**

**How to create project schedules**

1. Define your project goals. ...
2. Identify all stakeholders. ...
3. Determine your final deadline. ...
4. List each step or task. ...
5. Assign a team member responsible for each task.
6. Work backward to set due dates for each task.

**CHAPTER- 2**

**LITERATURE REVIEW**

2.1 INTELLIJ

IntelliJ IDEA is an **Integrated Development Environment (IDE)** for JVM languages designed to maximize developer productivity. It does the routine and repetitive tasks for you by providing clever code completion, static code analysis, and refactoring, and lets you focus on the bright side of software development, making it not only productive but also an enjoyable experience.

IntelliJ IDEA analyzes your code, looking for connections between symbols across all project files and languages. Using this information it provides in depth coding assistance, quick navigation, clever analysis, and of course, refactoring.

 It is available in two editions: the Community Edition which is licensed by Apache 2.0, and a commercial edition known as the Ultimate Edition. Both of them can be used for creating software which can be sold. What makes IntelliJ IDEA so different from its counterparts is its ease of use, flexibility and its solid design.

IntelliJ IDEA was developed by JetBrains, formerly known as IntelliJ. It was first released in 2001, and it boasted features like advanced code navigation and the ability to refactor codes, which made it very popular. It even received the distinction of being voted the best programming tool based on Java in 2010, sidelining established tools like NetBeans, Eclipse and JDeveloper. The open-source development environment for Android released by Google in 2014 is also based on IntelliJ IDEA. The IDE supports many other programming languages such as Python, Lua and Scala.

The biggest reason it is regarded as one of the best programming tools based on Java is its assistance features, which makes it easy to use and makes the programs created by it very well designed. It also has advanced error checking features which allows faster and easier error checking.

* BENEFITS

IDE stands for Integrated Development Environment. It is a combination of multiple tools, which make software development process easier, robust and less error-prone. It has following benefits over plain text editor −

* Integration with useful tools like compiler, debugger, version control system, build tools, various frameworks, application profilers and so on.
* Supports code navigation, code completion, code refactoring and code generation features which boosts development process.
* Supports unit testing, integration testing and code coverage via plug-ins.
* Provides rich set of plug-ins to enhance **IDE** functionality further.
* FEATURES OF INTELLIJ

IntelliJ IDEA has some top productive Java code completion features. Its predictive algorithm can accurately assume what a coder is attempting to type, and completes it for him, even if he doesn’t know the exact name of a particular class, member or any other resource.

### Deep Insight

IntelliJ IDEA really understands and has a **deep insight** into your code, as well as the context of the coder, which makes it so unique among other Java IDEs.

* **Smart code completion** − It supports context-based code completion. It gives a list of the most relevant symbols applicable in the current context.
* **Chain code completion** − It is an advanced code completion feature which lists applicable symbols accessible via methods or getters in the current context.
* **Static member's completion** − It allows you to use static methods or constants and automatically adds the required import statements to avoid compilation error.
* **Detecting duplicates** − It finds the duplicate code fragments on the fly and gives notification/suggestion about it to user.
* **Inspections and quick-fixes** − Whenever IntelliJ detects that you're about to make a mistake, a little light bulb notification pops up on same line. Clicking it shows the suggestion list.

2.2 SPRING BOOT

Spring Boot is an open source Java-based framework used to create a micro Service. It is developed by Pivotal Team and is used to build stand-alone and production ready spring applications. This chapter will give you an introduction to Spring Boot and familiarizes you with its basic concepts.

Spring Boot provides a good platform for Java developers to develop a stand-alone and production-grade spring application that you can **just run**. You can get started with minimum configurations without the need for an entire Spring configuration setup.

You can choose Spring Boot because of the features and benefits it offers as given here −

* It provides a flexible way to configure Java Beans, XML configurations, and Database Transactions.
* It provides a powerful batch processing and manages REST endpoints.
* In Spring Boot, everything is auto configured; no manual configurations are needed.
* It offers annotation-based spring application
* Eases dependency management
* It includes Embedded Servlet Container.



Spring Boot is a combination of **Spring Framework** and **Embedded HTTP Servers**.

## 2.2.1 Developer Ergonomics

IntelliJ IDEA is designed around the coding principle that developers should be allowed to write codes with as less distraction as possible. That is why in this case, the editor is the only thing visible on the screen, with dedicated shortcuts for all other coding-unrelated functions.

* **Editor-centric environment** − Quick pop-ups help in checking additional information without leaving the current context.
* **Shortcuts for everything** − IntelliJ IDEA has keyboard shortcuts for nearly everything, including rapid selection and switching between tool windows and many more.
* **Inline debugger** − Inline debugger allows you to debug application in IDE itself. It makes the development and debugging process seamless.

## 2.2.2 Built-in Developer Tools

To help the developers organize their workflow, IntelliJ IDEA offers them an amazing toolset, which comprises of a de-compiler, Docker support, bytecode viewer, FTP and many other tools −

* **Version control** − IntelliJ supports most of the popular version control system like Git, Subversion, Mercurial, CVS, Perforce, and TFS.
* **Build tools** − IntelliJ supports Java and other build tools like Maven, Gradle, Ant, Gant, SBT, NPM, Webpack, Grunt, and Gulp.
* **Test runner and code coverage** − IntelliJ IDEA lets you perform unit testing with ease. The IDE includes test runners and coverage tools for major test frameworks, including JUnit, TestNG, Spock, Cucumber, Scala Test, spec2, and Karma.
* **De-compiler** − IntelliJ comes with a built-in de-compiler for Java classes. When you want to take a look inside a library that you do not have the source code for, you can do it without using any third-party plug-ins.
* **Terminal** − IntelliJ provides built-in terminal. Depending on your platform, you can work with the command line prompt, like PowerShell or Bash.
* **Database tools** − IntelliJ provides database tools, which allow you to connect to live databases; run queries; browse and update data; and even manage your schemas in a visual interface from IDE itself.
* **Application server** − IntelliJ supports major application servers: Tomcat, JBoss, WebSphere, WebLogic, Glassfish, and many others. You can deploy your artifacts onto application servers and debug the deployed applications in IDE itself.
* **Docker support** − Via a separate plug-in, IntelliJ provides a dedicated tool window that lets you connect to locally running Docker machines.

2.3 MySQL Workbench

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more. MySQL Workbench is available on Windows, Linux and Mac OS X.

## 2.3.1 Design

MySQL Workbench enables a DBA, developer, or data architect to visually design, model, generate, and manage databases. It includes everything a data modeler needs for creating complex ER models, forward and reverse engineering, and also delivers key features for performing difficult change management and documentation tasks that normally require much time and effort.

## Develop

MySQL Workbench delivers visual tools for creating, executing, and optimizing SQL queries. The SQL Editor provides color syntax highlighting, auto-complete, reuse of SQL snippets, and execution history of SQL. The Database Connections Panel enables developers to easily manage standard database connections, including MySQL Fabric. The Object Browser provides instant access to database schema and objects.

## 2.3.3 Administer

MySQL Workbench provides a visual console to easily administer MySQL environments and gain better visibility into databases. Developers and DBAs can use the visual tools for configuring servers, administering users, performing backup and recovery, inspecting audit data, and viewing database health.

## 2.3.4 Visual Performance Dashboard

MySQL Workbench provides a suite of tools to improve the performance of MySQL applications. DBAs can quickly view key performance indicators using the Performance Dashboard. Performance Reports provide easy identification and access to IO hotspots, high cost SQL statements, and more. Plus, with 1 click, developers can see where to optimize their query with the improved and easy to use Visual Explain Plan.

## Database Migration

MySQL Workbench now provides a complete, easy to use solution for migrating Microsoft SQL Server, Microsoft Access, Sybase ASE, PostreSQL, and other RDBMS tables, objects and data to MySQL. Developers and DBAs can quickly and easily convert existing applications to run on MySQL both on Windows and other platforms. Migration also supports migrating from earlier versions of MySQL to the latest releases.

**CHAPTER-3**

**3.1 FEASIBILITY STUDY**

Feasibility is a measure of how beneficial the development of the information system will be to an organization. This is done by investigating the existing system in the area under investigation or general ideas about a new system. It is a test of a system proposal according to its workability, impact on the organization, ability to meet user needs and effective use of resources.

A prefeasibility study seeks to validate a project and to expose organizational spots or problems that may condemn it before it sees the light of day, using a limited number of analytical tools. A prefeasibility study does not concentrate on the day-to-day operations; in other words, it does not investigate the specifics of tasks or the team member composition. This is something that the feasibility study does. From this perspective, a feasibility study is very pragmatic; once we have defined the project in approximate terms using the five frames of analysis, we can then examine how this is going to translate into daily operations, most evident during the transformation phase.

As an example, let us take the case of the replacement of the Champlain Bridge that connects the island of Montréal to its south shore and the United States and which is nearing the end of its useful life. The 4.2-billion-dollar (C$) project will inevitably generate a perimeter of intense noise (machinery, etc.) and dust in an urban area that is quite populated. Natural ecosystems are threatened. A thorough environmental feasibility study is required. A feasibility analyst could linger on whether the construction is feasible, taking into account dust and displacements of rare animal and plant species as well as maximum noise levels. Of course, that would not be enough in this case: other aspects such as financial and technical concerns would also have to be addressed.

### 3.2 TECHNICAL FEASIBILITY

A technical feasibility study assesses the details of how you intend to deliver a product or service to customers. Think materials, labor, transportation, where your business will be located, and the technology that will be necessary to bring all this together.

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn’t want to try to put Star Trek’s transporters in their building—currently, this project is not technically feasible.

Technology Feasibility and Cost Analysis is performed to determine the potential economic viability of a process or technology, and helps to identify which technologies have the greatest likelihood of economic success.

A technical feasibility study can provide relevant context to the different aspects of your project and serve as a great planning tool by providing an overhead view of how your project can evolve during the course of its development, troubleshooting and tracking the progress of your project from concept to reality.

3.3 **TECHNOLOGY DESCRIPTION**

**Technology** ("science of craft", from [Greek](https://en.wikipedia.org/wiki/Ancient_Greek) τέχνη, *techne*, "art, skill, cunning of hand"; and λογία, [*-logi*](https://en.wiktionary.org/wiki/-logia)[*a*)](https://en.wikipedia.org/wiki/Technology#cite_note-Liddell_1980-2) is the sum of [techniques](https://en.wikipedia.org/wiki/Art_techniques_and_materials)[, skills](https://en.wikipedia.org/wiki/Skill)[, methods,](https://en.wikipedia.org/wiki/Scientific_method) and [processes](https://en.wikipedia.org/wiki/Business_process) used in the production of [goods](https://en.wikipedia.org/wiki/Good_(economics)) or [services](https://en.wikipedia.org/wiki/Service_(economics)) or in the accomplishment of objectives, such as [scientific investigation](https://en.wikipedia.org/wiki/Scientific_investigation). Technology can be the [knowledge](https://en.wikipedia.org/wiki/Knowledge) of techniques, processes, and the like, or it can be embedded in [machines](https://en.wikipedia.org/wiki/Machines) to allow for operation without detailed knowledge of their workings. [Systems](https://en.wikipedia.org/wiki/System) (e.g. machines) applying technology by taking an [input,](https://en.wikipedia.org/wiki/Input/output) changing it according to the system's use, and then producing an [outcome](https://en.wikipedia.org/wiki/Input/output) are referred to as **technology systems** or **technological systems**.

The simplest form of technology is the development and use of basic [tools.](https://en.wikipedia.org/wiki/Tool)

The [prehistoric](https://en.wikipedia.org/wiki/Prehistory) invention of shaped stone tools followed by the discovery of [how to control fire](https://en.wikipedia.org/wiki/Control_of_fire_by_early_humans) increased sources of food. The later [Neolithic Revolution](https://en.wikipedia.org/wiki/Neolithic_Revolution) extended this, and quadrupled the sustenance available from a territory. The invention of the [wheel](https://en.wikipedia.org/wiki/Wheel) helped humans to travel in and control their environment.

Developments in historic times, including the [printing press,](https://en.wikipedia.org/wiki/Printing_press) the [telephone,](https://en.wikipedia.org/wiki/Telephone) and the [Internet,](https://en.wikipedia.org/wiki/Internet) have lessened physical barriers to [communication](https://en.wikipedia.org/wiki/Communication) and allowed humans to interact freely on a global scale.

Technology has many effects. It has helped develop more advanced [economies](https://en.wikipedia.org/wiki/Economy) (including today's [global economy)](https://en.wikipedia.org/wiki/Economic_globalization) and has allowed the rise of a [leisure class.](https://en.wikipedia.org/wiki/Conspicuous_leisure) Many technological processes produce unwanted by-products known as [pollution](https://en.wikipedia.org/wiki/Pollution) and deplete natural resources to the detriment of Earth's [environment.](https://en.wikipedia.org/wiki/Natural_environment) Innovations have always influenced the [values](https://en.wikipedia.org/wiki/Value_(personal_and_cultural)) of a society and raised new questions in the [ethics of technology.](https://en.wikipedia.org/wiki/Ethics_of_technology) Examples include the rise of the notion of [efficiency](https://en.wikipedia.org/wiki/Efficiency) in terms of human [productivity,](https://en.wikipedia.org/wiki/Productivity) and the challenges of [bioethics.](https://en.wikipedia.org/wiki/Bioethics)

3.4 **TECHNOLOGY USED IN PROJECT**

* CSS
* JAVA
* MySQL

**Server:** Tomcat8

**Database:** MySQL

**Operating System:** Windows7/8/8.1/10

**CHAPTER-4**

**BACKEND DESIGN**

4.1 DATA DICTIONARY

A data dictionary is a collection of the names, definitions, and attributes for data elements and models. The data in a data dictionary is the metadata about the database. These elements are then used as part of a database, research project, or information system.

Data dictionaries are used to provide detailed information about the contents of a dataset or database, such as the names of measured variables, their data types or formats, and text descriptions. A data dictionary provides a concise guide to understanding and using the data.

There are two types of data dictionaries: active and passive. An active data dictionary is tied to a specific database which makes data transference a challenge, but it updates automatically with the data management system.

A data dictionary in Software Engineering means a file or a set of files that includes a database's metadata (hold records about other objects in the database), like data ownership, relationships of the data to another object, and some other data.

## 4.2 ER DIAGRAMS

An E-R model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business. It does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (*entities*) that are connected by lines (*relationships*) which express the associations and dependencies between entities. An ER model can also be expressed in a verbal form, for example: *one building may be divided into zero or more apartments, but one apartment can only be located in one building.*

Entities may be characterized not only by relationships, but also by additional properties (*attributes*), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity–relationship models.

An ER model is typically implemented as a [database.](https://en.wikipedia.org/wiki/Database) In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a [relational database](https://en.wikipedia.org/wiki/Relational_database) a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity.

There is a tradition for ER/data models to be built at two or three levels of abstraction. Note that the conceptual-logical-physical hierarchy below is used in other kinds of specification, and is different from the [three schema approach](https://en.wikipedia.org/wiki/Three_schema_approach) to [software engineering](https://en.wikipedia.org/wiki/Software_engineering).

HOSPITALS

agency

APPOINTMENT

USES

HAS

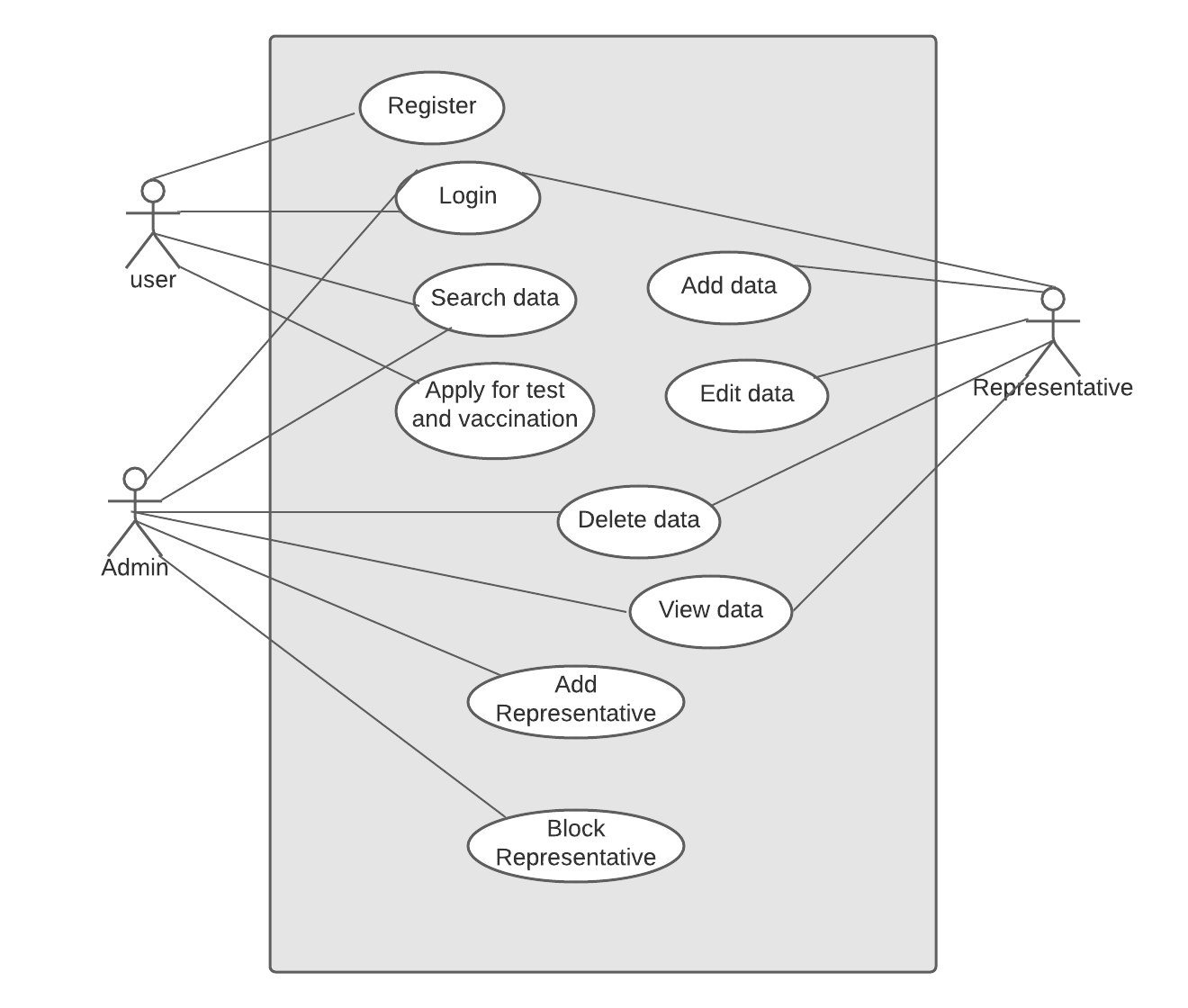
AMBULANCE

SUPPLIERS

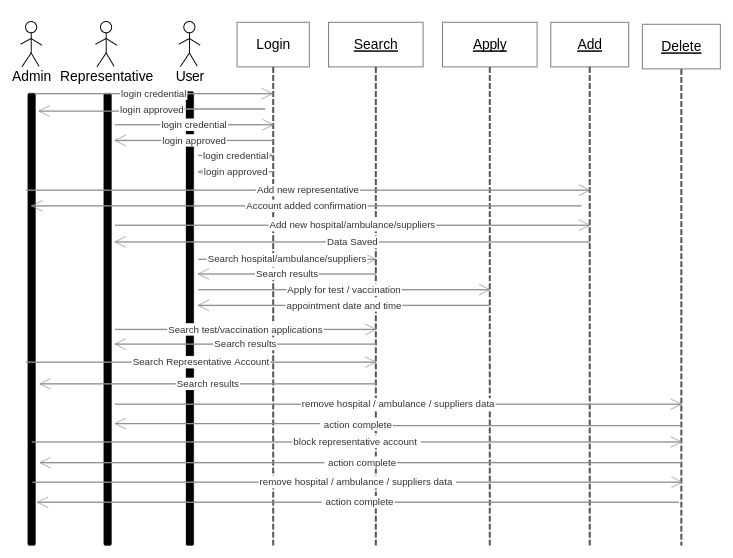
AREAS

AREAS

**Fig-4.2.1: ER DIAGRAM**

**4.3Use Case Diagram:** 

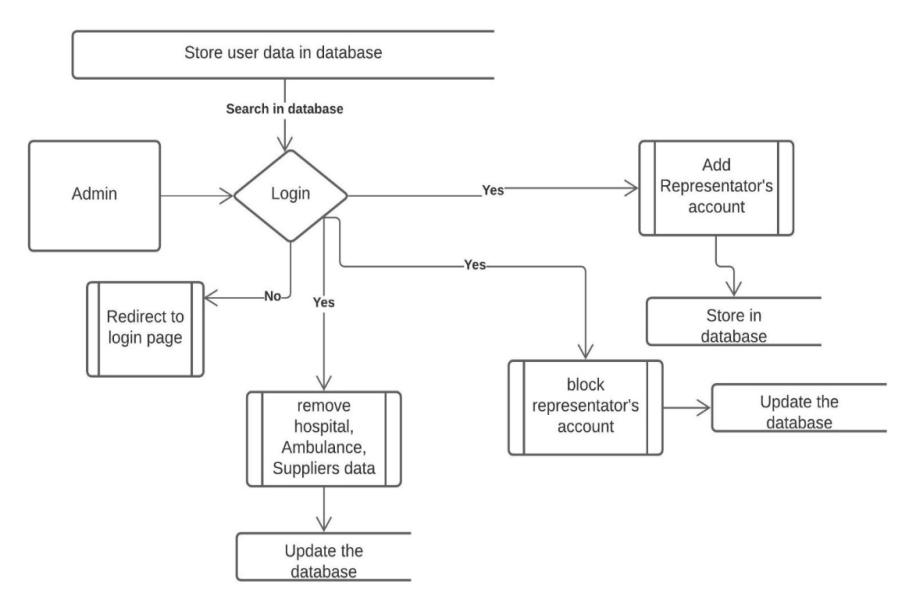
**Fig-4.3.1: Use Case Diagram for covid portal**



**Fig-4.3.2: Sequence Diagram**

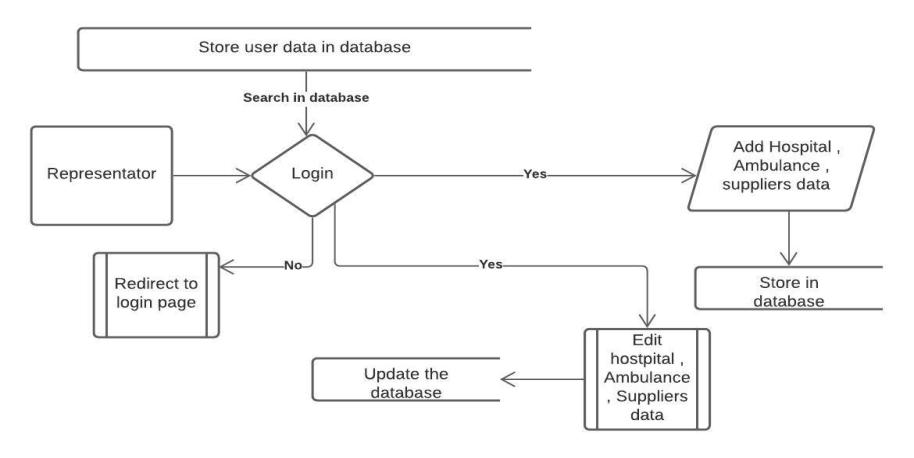
**4.4 Data Flow Diagrams:**

Admin:



**Fig-4.4.1: Admin Dataflow Diagram**

Representative:



**Fig-4.4.2: Representative’s Dataflow Diagram**

### 4.5 Database Design:

### 

Fig4.5.1: Data Model

**Tables:**

**4.5.2** Appointment

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Length | Nulls |
| id | int | 10 | PRIMARY, NOT NULL |
| user\_ id | int | 10 | NULL |
| hospital\_id | int | 10 | NULL |
| cause | varchar | 20 | NULL |
| status | varchar | 20 | NULL |
| appointment\_time | timestamp |  | NULL |

4.5.3 Ambulance

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Length | Nulls |
| id | int | 10 | PRIMARY, NOT NULL |
| pincode | int | 6 | NULL |
| agency | varchar | 50 | NULL |
| phone | Varchar[u] | 12 | NOT NULL, UNIQUE |

4.5.4 Areas

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Length | Nulls |
| street | varchar | 80 | NULL |
| pincode | Varchar[P] | 6 | PRIMARY, NOT NULL |
| city | Varchar[u] | 15 | UNIQUE, NOT NULL |
| state | varchar | 15 | NULL |

4.5.5 Users

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Length | Nulls |
| user\_id | int | 10 | NOT NULL, AUTO INCREMENT |
| User\_name | varchar | 30 | PRIMARY, NOT NULL |
| Password | varchar | 20 | NULL |
| type | varchar | 10 | NULL |
| address | varchar | 80 | NULL |
| phone | varchar | 12 | NULL |
| status | varchar | 10 | NULL |

4.5.6 Suppliers

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Length | Nulls |
| id | int | 10 | PRIMARY, NOT NULL |
| name | Varchar | 50 | NULL |
| address | Varchar | 80 | NULL |
| pincode | int | 6 | NULL |
| phone | varchar | 12 | UNIQUE, NOT NULL |
| availability | int | 10 | NULL |

4.5.7 Hospitals

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Length | Nulls |
| id | int | 10 | NOT NULL, AUTO INCREMENT |
| name | varchar | 50 | NULL |
| phone | varchar | 12 | NOT NULL, UNIQUE |
| pincode | int | 6 | NULL |
| address | varchar | 80 | NULL |
| beds | int | 10 | NULL |
| doctor | int | 10 | NULL |
| ventilators | int | 10 | NULL |
| Last\_update | timestamp |  | NULL |

**CHAPTER-5**

**TESTING**

5.1 INTRODUCTION

Testing is the integral part of any System Development Life Cycle insufficient and interested application tends to crash and result in loss of economic and manpower investment besides user’s dissatisfaction and downfall of reputation.

“Software Testing can be looked upon as one among much process, an organization performs, and that provides the last opportunity to correct any flaws in the developed system. Software Testing includes selecting test data that have more probability of giving errors.” The first step in System testing is to develop the plan that all aspect of system. Complements, Correctness, Reliability and Maintainability.

Software is to be tested for the best quality assurance, an assurance that system meets the specification and requirement for its intended use and performance System Testing is the most useful practical process of executing the program with the implicit intention of finding errors that makes the program fail.

5.2Types of Testing

Black Box (Functional) Testing:

Testing against specification of system or components. Study it by examining its inputs and related outputs. Key is to devise inputs that have a higher likelihood of causing outputs that reveal the presence of defects. Use experience and knowledge of domain to identify such test cases. Failing this a systematic approach may be necessary. Equivalence partitioning is where the input to a program falls into a number of classes, e.g. positive numbers vs. negative numbers. Programs normally behave the same way for each member of a class. Partitions exist for both input and output. Partitions may be discrete or overlap. Invalid data (i.e. outside the normal partitions) is one or more partitions that should be tested. Internal System design is not considered in this type of testing. Tests are based on requirements and functionality.

This type of test case design method focuses on the functional requirements of the software, ignoring the control structure of the program. Black box testing attempts to find errors in the following categories:

* Incorrect or missing functions.
* Interface errors.
* Errors in data structures or external database access.
* Performance errors.
* Initialization and termination errors.

White Box (Structural) Testing:

Testing based on knowledge of structure of component (e.g. by looking at source code). Advantage is that structure of code can be used to find out how many test case need to be performed. Knowledge of the algorithm (examination of the code) can be used to identify the equivalence partitions. Path testing is where the tester aims to exercise every independent execution path through the component. All conditional statements tested for both true and false cases. If a unit has no control statements, there will be up to 2n possible paths through it. This demonstrates that it is much easier to test small program units than large ones. Flow graphs are a pictorial representation of the paths of control through a program (ignoring assignments, procedure calls and I/O statements). Use flow graph to design test cases that execute each path. Static tools may be used to make this easier in programs that have a complex branching structure. Tools support. Dynamic program analyzers instrument a program with additional code. Typically this will count how many times each statement is executed. At end print out report showing which statements have and have not been executed. Problems with flow graph derived testing:

* Data complexity could not take into account.
* We cannot test all paths in combination.
* In really only possible at unit and module testing stages because beyond that complexity is too high.

This testing is based on knowledge of the internal logic of an application’s code. Also known as a Glass Box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

Unit Testing:

Unit testing concentrates on each unit of the software as implemented in the code. This is done to check syntax and logical errors in programs. At this stage, the test focuses on each module individually, assuring that it functions properly as a unit. In our case, we used extensive white-box testing at the unit testing stage.

A developer and his team typically do the unit testing do the unit testing is done in parallel with coding; it includes testing each function and procedure.

## Incremental Integration Testing:

Bottom-up approach for testing i.e. continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately done by programmers or by testers.

## Integration Testing:

Testing of integration modules to verify combined functionality after integration. Module are typically code modules, individual applications, client and server and distributed systems.

## Functional Testing:

This type of testing ignores the internal parts and focus on the output is as per requirement or not. Black -box type testing geared to functionality requirements of an application.

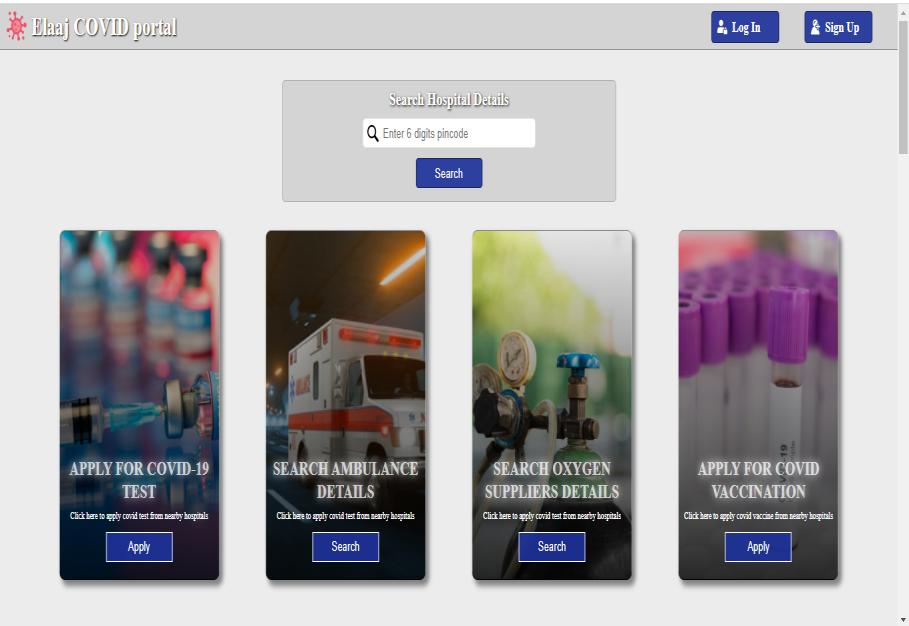
## System Testing:

Entire system is tested as per the requirements. Black box type test that is based on overall requirement specifications covers all combined parts of a system.

**CHAPTER-6**

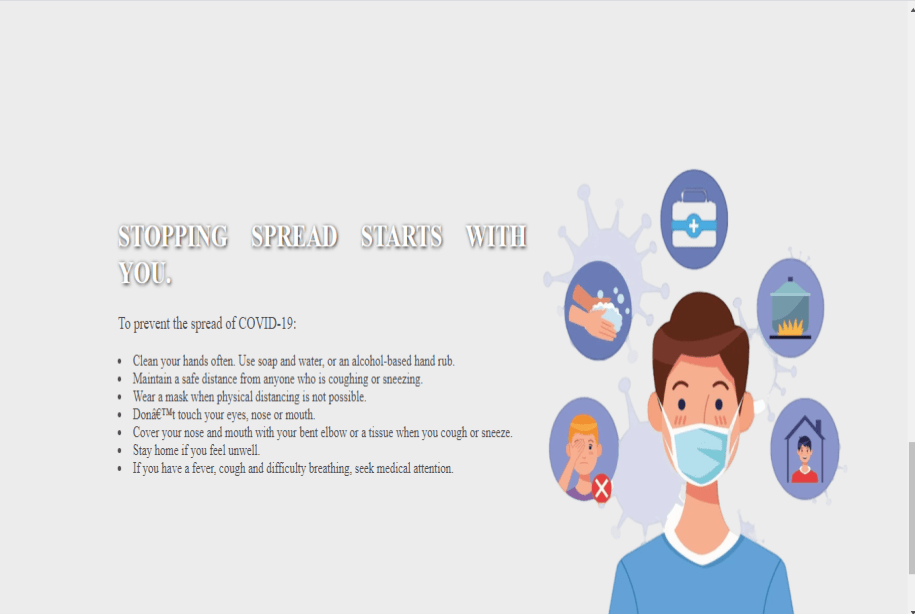
**SCREEN SHOTS**

**Home Page:**

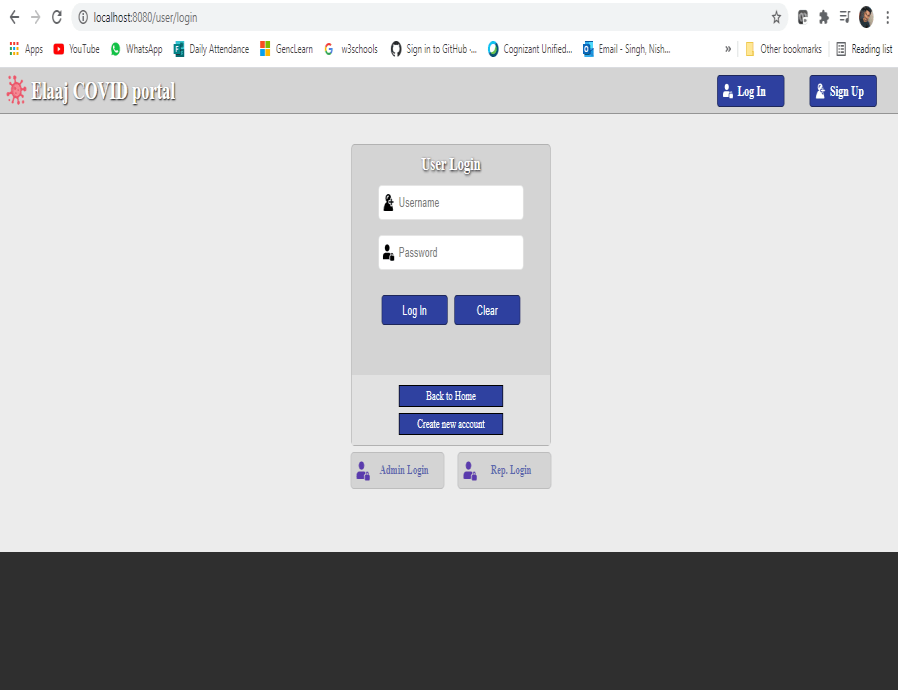
****

****

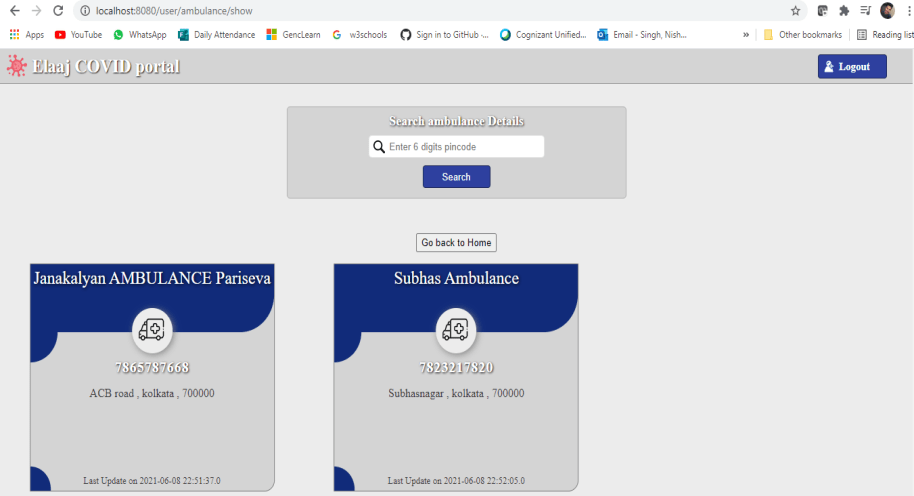
****

****

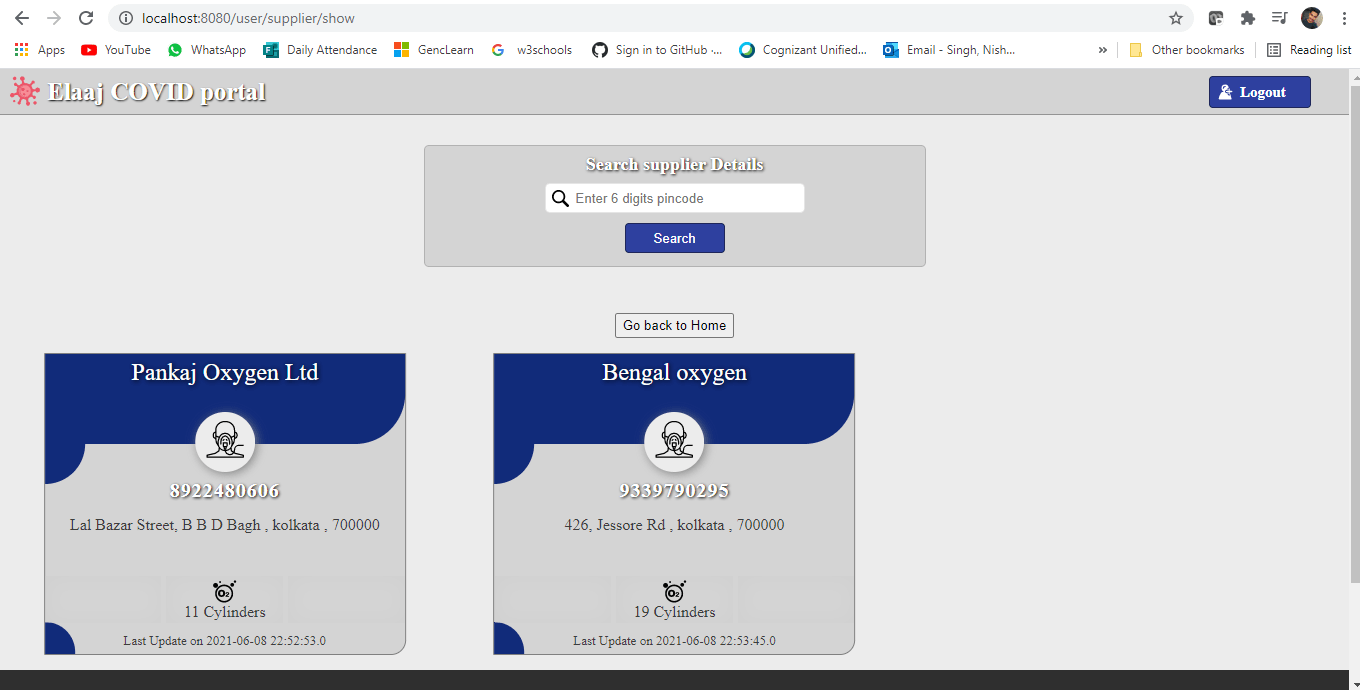
**User Login Page:**

****

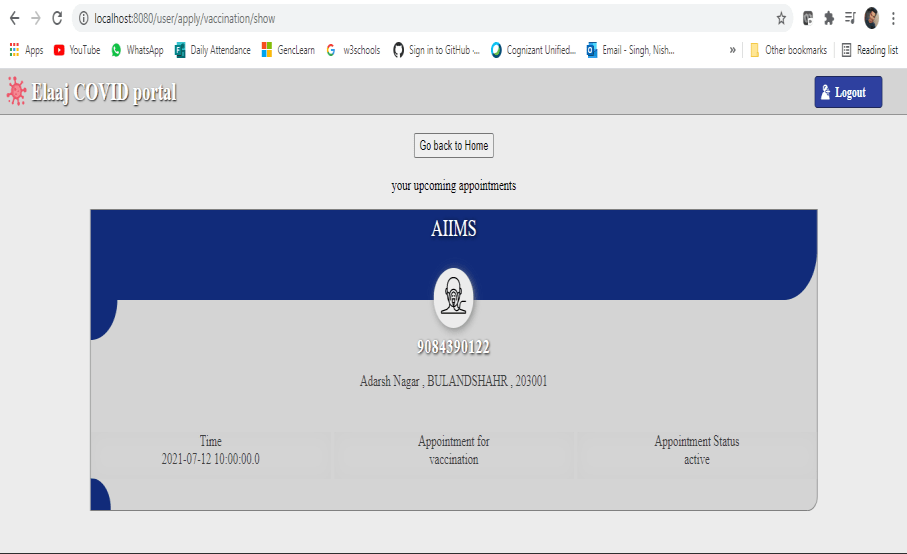
**Ambulance Details:**

****

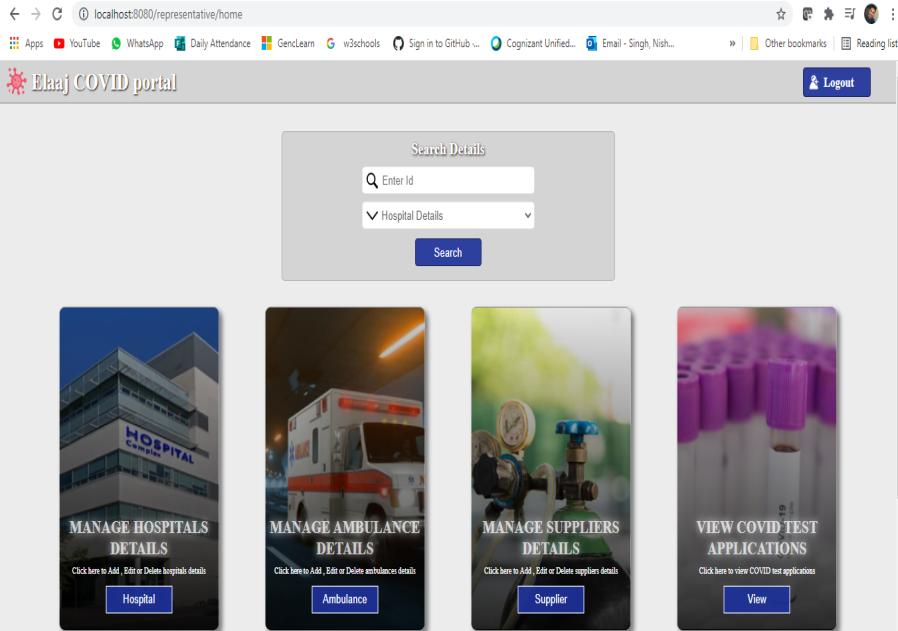
**Supplier’s Details:**

****

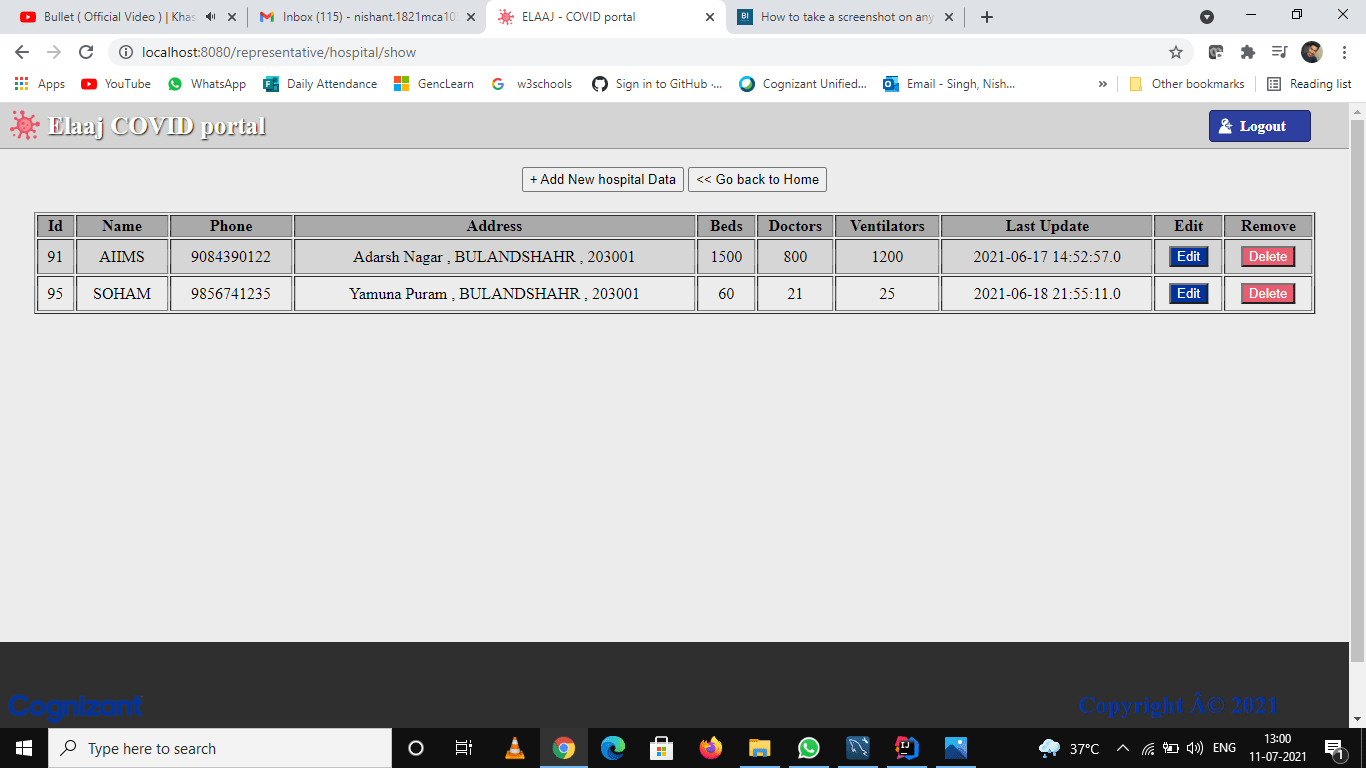
**Vaccination Details:**

****

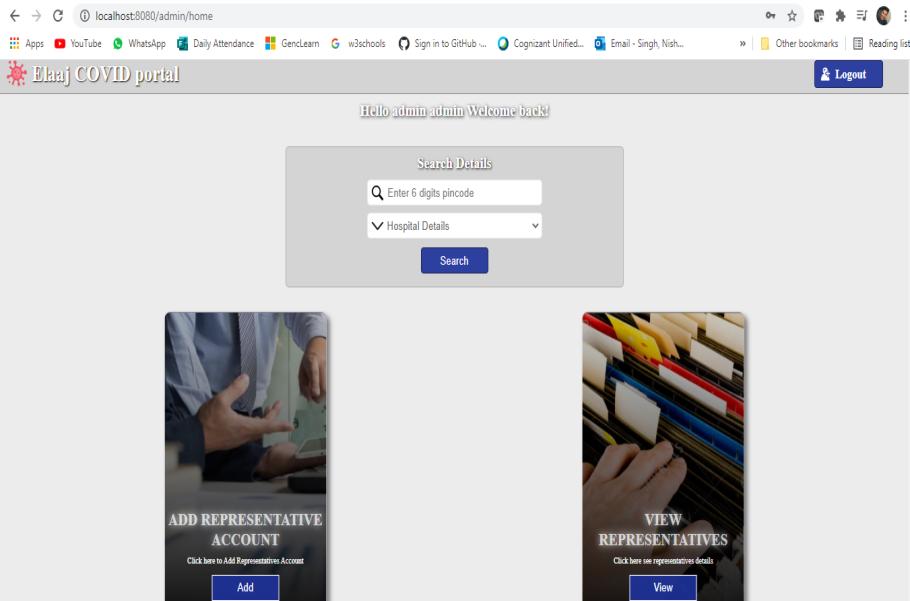
**Representative Homepage:**

****

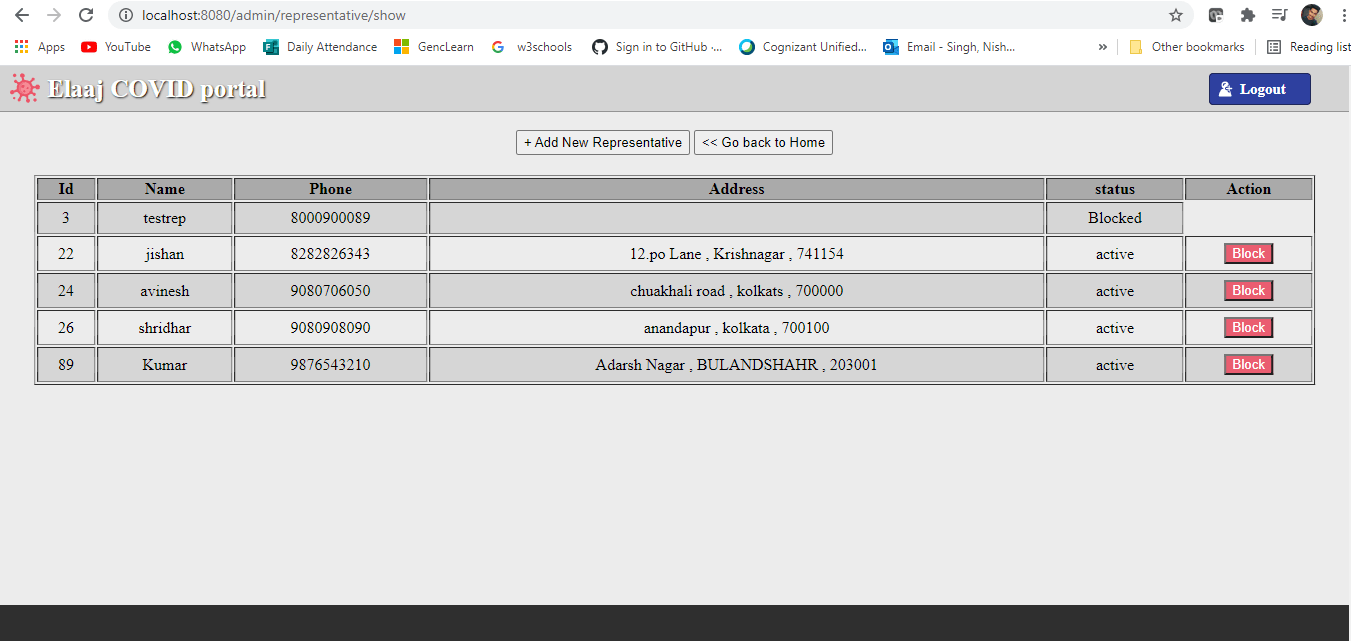
**Hospital Add/Remove:**

****

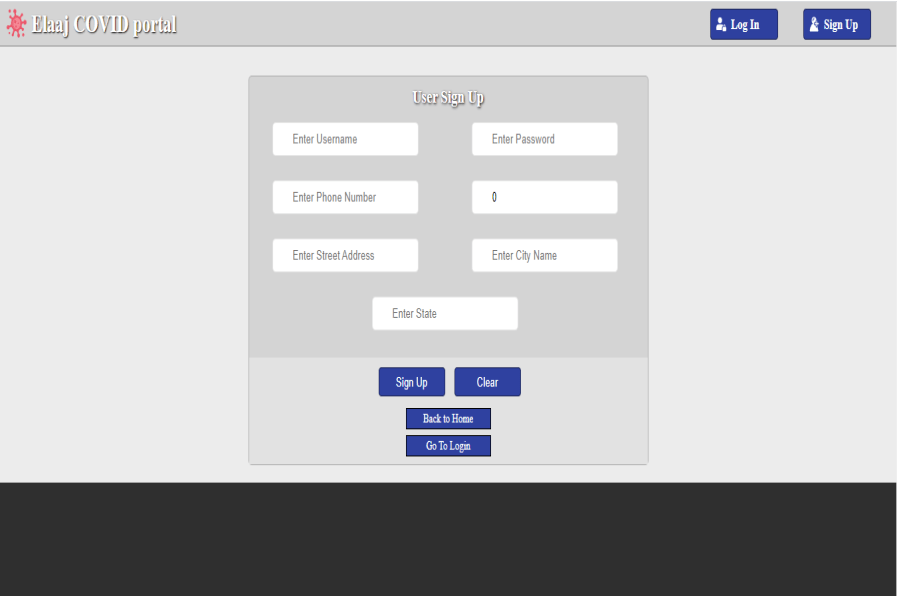
**Admin Homepage:**

****

**Admin Add/Block Representative:**

****

**User Sign Up:**

****

**CHAPTER-7**

**LIMITATIONS**

**7.1 LIMITATIONS**

The limitations of the study are those characteristics of design or methodology that impacted or influenced the application or interpretation of the results of your study. They are the constraints on generalizability and utility of findings that are the result of the ways in which you chose to design the study and/or the method used to establish internal and external validity.

Every project in this world has some limitations, similarly our project also has some of them:

* Internet based, without internet user cannot access its tools.
* Server Dependent
* User Details must be correct

**7.2 FUTURE SCOPE**

The future scope of the project are as follows:

* Providing better User Experience
* Additional features like more states to be added.
* Increase more beds and hospitals

**7.3 FUTURE ENHANCEMENT**

1.Prediction Model

India has not reached the peak yet, once it reaches the peak the prediction model can be built to show that how much time it will take to get things back to the normal.

2.Sentiment Analysis

India has never experience such pandemic in last 100 years so what do people think about this pandemic, lockdown, government approach/policies etc. can be studied to have sentiment insight of this pandemic.

**CHAPTER-8**

**CONCLUSION**

● India is now 6th largest confirmed cases of COVID-19 in the world.

● India has not reached the peak yet so as of now we cannot predict the approximate number of confirmed cases in India.

● Recovery rate of India is also showing exponential behavior same as confirmed cases. But if cases increase beyond certain point then thing can go out of control which will affect the recovery rate.

● Gender information of most of the patients is nor released by the government but whatever data is available shows number of infected males is more than that of females. This is may be due to more exposure/contact of males with outdoor world.

● Same as gender, age information is unavailable for most of the patients, but whatever data is available shows age and cases are normally distributed and 21-40 is the age bin which has been infected more.

● This analysis showed that pandemic like this affects economy the most. Whichever the country’s GDP source is, it gets targeted the most. Slowing down the economy then unemployment, job losses and then this chain reaction continues.

● India was in lockdown for more than 2 moths but still situation did not get any better. This may be due to weak administration or the violation of lockdown by citizens.

**CHAPTER 9**

**BIBLIOGRAPHY AND REFERENCES**

1. Herbert Schildt (JAVA- The Complete Reference)

11th Edition

**Publisher –** McGraw Hill Education

2. Craig Walls (Spring in Action)

5th Edition

**Publisher –** Manning

3. Craig Walls (Spring Boot in Action)

**Publisher –** Manning

4. John Carnell (Spring Microservices in Action)

**Publisher –** Manning

5. Arnold, K., Gosling, J., & Holmes, D. (2005). *The Java programming language*. Addison Wesley Professional.

6. Poo, D., Kiong, D., & Ashok, S. (2007). *Object-oriented programming and Java*. Springer Science & Business Media.

7. Zhang, P. (2017). *Practical Guide for Oracle SQL, T-SQL and MySQL*. CRC Press.

8. Halpin, T., & Morgan, T. (2010). *Information modeling and relational databases*. Morgan Kaufmann.

9. Gasston, P. (2011). *The book of CSS3: A Developer's Guide to the Future of Web Design*

10. Gans, M., Hodges, T., & Wilson, G. (2020). *JavaScript for Data Science*. Chapman and Hall/CRC.

11. Bertolino, A. (2007, May). Software testing research: Achievements, challenges, dreams. In *Future of Software Engineering (FOSE'07)* (pp. 85-103). IEEE.

12. Cook, S., & Daniels, J. (1994). *Designing object systems* (Vol. 135). Englewood Cliffs, NJ: Prentice Hall.13.

14. Mermoud, A., & Lejeune, T. (2010). Partial shadings on PV arrays: by-pass diode benefits analysis. In *25th European Photovoltaic Solar Energy Conference*.

15. David, M. (2013). *HTML5: designing rich Internet applications*. Routledge.