2022

HEALTH CARE EXPRESS

R D PRADEEKAR
AJAY RAWAL
C SAI SUBRAMANYAM
SHREYA C N
SHWETHA C
SYEDA HEENA KAUSAR

valtech_

HEALTH CARE EXPRESS

Table of contents

Chapter 1: Introduction 1.1 Abstract 1.2 Introduction 1.3 Problem Statement 1.4 Objective 1.5 Scope of the Project 1.6 Modules 1.7 Admin Module 1.8 Doctor Module 1.9 Staff Module 1.10 Ambulance Module	1
Chapter 2: Design 2.1 Use Case Diagram 2.2 Class Diagrams 2.3 Sequence Diagrams 2.4 E-R Diagram	5
Chapter 3: Analysis 3.1 Proposed System 3.2 Feasibility Study 3.2.1 Economic Feasibility 3.2.2 Technical Feasibility 3.2.3 Operational Feasibility 3.3 Software Specification 3.4 H2 Database 3.5 Java- Spring Boot	10

Chapter 4: Sample Screenshots 4.1 Login Page 4.2 Main Façade 4.3 Departments 4.4 Adding New Employee Doctor 4.5 Doctor Module 4.6 Operation Details 4.7 Create Operation Details 4.8 Bed Allotment 4.9 Payment Details 4.10 Birth Report 4.11 Operation Report	22

CHAPTER 1
INTRODUCTION

1.1 ABSTRACT

Hospital Management System is an organized computerized system designed and programmed to deal with day-to-day operations and management of the hospital activities. The program can look after inpatients, outpatients, records, database treatments, status illness, billings in the pharmacy and labs. It also maintains hospital information such as ward id, doctors in charge and department administering. The major problem for the patient nowadays to get report after consultation, many hospital managing reports in their system but it's not available to the patient when he / she is outside. In this project we are going to provide the extra facility to store the report in the database and make available from anywhere in the world.

1.2 INTRODUCTION

The Best Medical Service of Bangalore City. Check All Feature Welcome to our new website. We encourage you to find out more about the values and vision that guides us, the services, and programs we offer, the indicators that ensure our continued accountability and transparency, and the news and events that matter most to you. It is the place where you will find information on career opportunities with HMH and be able to access general healthcare information to keep you well informed. India Hospital has a proud tradition of care. For Last so many years, India Hospital has cared for the people of Bangalore, surrounding communities, and beyond, being a significant player within the district. Residing in the heart of a beautiful cottage country on the shores of the Kaveri River, we offer healthcare not only to our community residents but to the many visitors who frequent our beautiful area year-round. To provide each patient with the world-class care, exceptional service, and compassion we would want for our loved ones and To Lead the evolution of healthcare to enable every member of the communities we serve to enjoy a better, healthier life. The mission of Health Care Express is to provide quality health services and facilities for the community, to promote wellness, to relieve suffering, and to restore health as swiftly, safely, and humanely as it can be done, consistent with the best service we can give at the highest value for all concerned.

1.3 PROBLEM DEFINITION

Whenever any person is critically needs to be taken to the hospital through ambulance. While travelling through ambulance the major problem is the doctor's unavailablity to attend the patient or monitor situation of the patient. And due to this majority of death happens in the ambulances. The system will facilitate the doctors to monitor the health and condition of the patient during the travel between patients' place and hospital. The system should be facilitated such that it should continuously send data of patient to the server from which the doctor can monitor the health condition of the patient. An android app which will make doctors and hospital staff to monitor the condition. The life of the person can be saved who are travelling through the ambulance.

1.4 OBJECTIVE

- Recording the information about the patient that come.
- The information can be viewed by the doctor.
- The instructions will be given by the doctor to the staff(ambulance) So that the staff can treat the patient accordingly.
- To provide quality health services and facilities for the community, to promote wellness, to relieve suffering, and to restore health as swiftly, safely, and humanely as it can be done.

1.5 SCOPE OF THE PROJECT

All this work is done by the staff and the doctor will be able to view the details of the patient and give the instructions to the staff in the ambulance. So that the staff can treat the patient accordingly and update about the situation of the patient to the doctor all the time. This reduces the criticality of the patient which helps to restore health swiftly and safely.

1.6 MODULES

The entire project mainly consists of 4 modules, which are

- ❖ Admin module
- ❖ Doctor module
- ❖ Staff module

1.4 ADMIN MODULE

- manage department of hospitals, user, doctor, nurse, account.
- watch appointment of doctors
- watch transaction reports of patient payment
- Bed, ward, cabin status
- watch medicine status of hospital stock
- watch operation report

1.3 DOCTOR MODULE

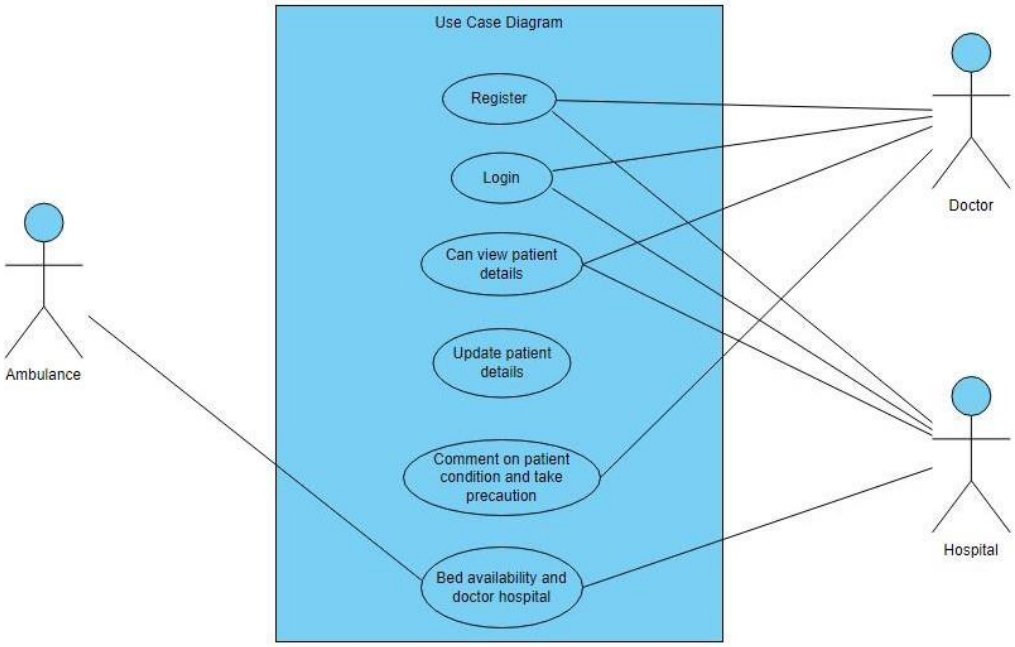
- View appointment list
- View doctor list
- View operation history
- Manage own profile
- Provide medication for patients

1.4 STAFF MODULE

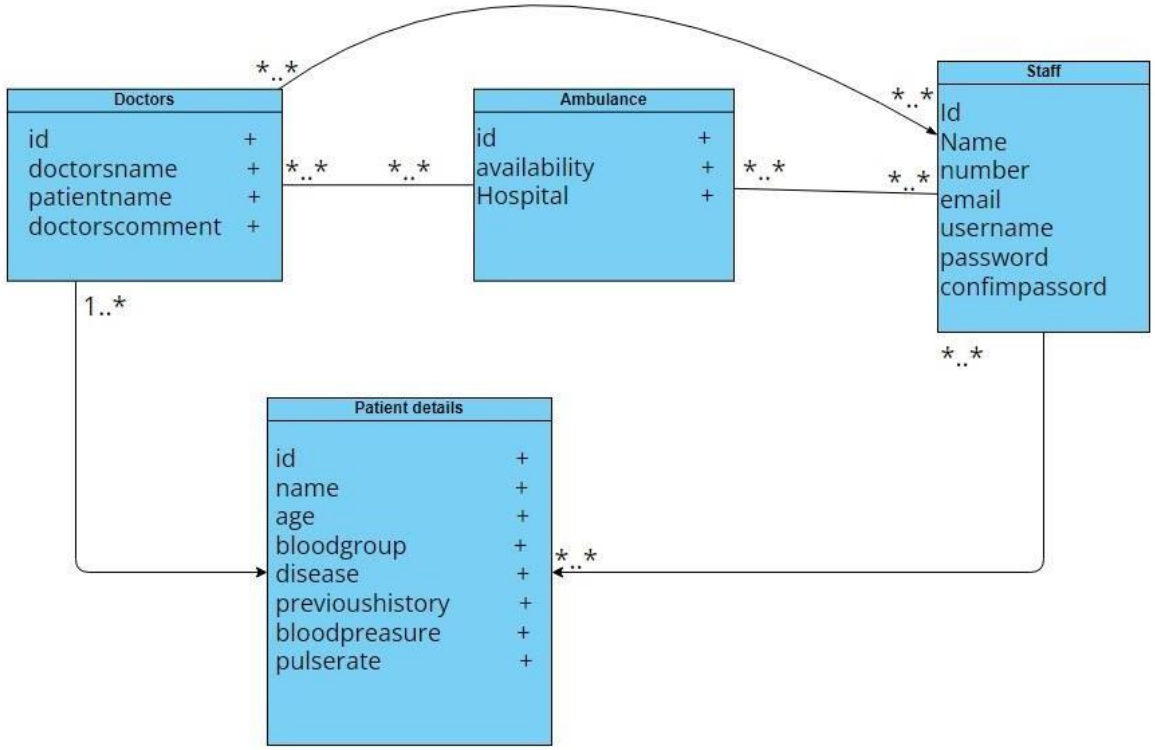
- Manage patient.
- Create, manage appointment with patient
- Issue for operation of patients and creates operation report
- Manage own profile
- Allot bed, ward, cabin for patients

CHAPTER 2
DESIGN

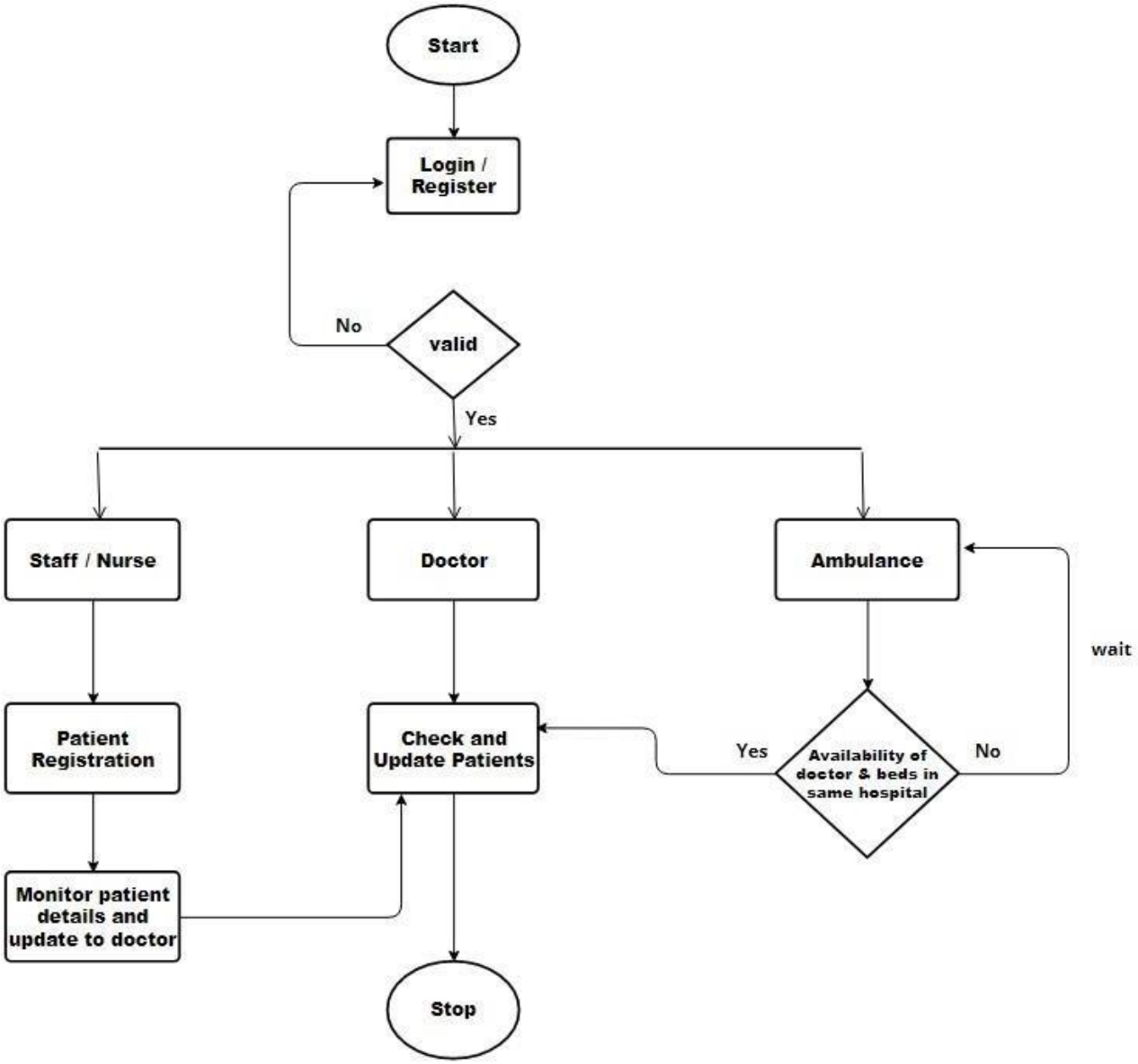
2.1 USE CASE DIAGRAM



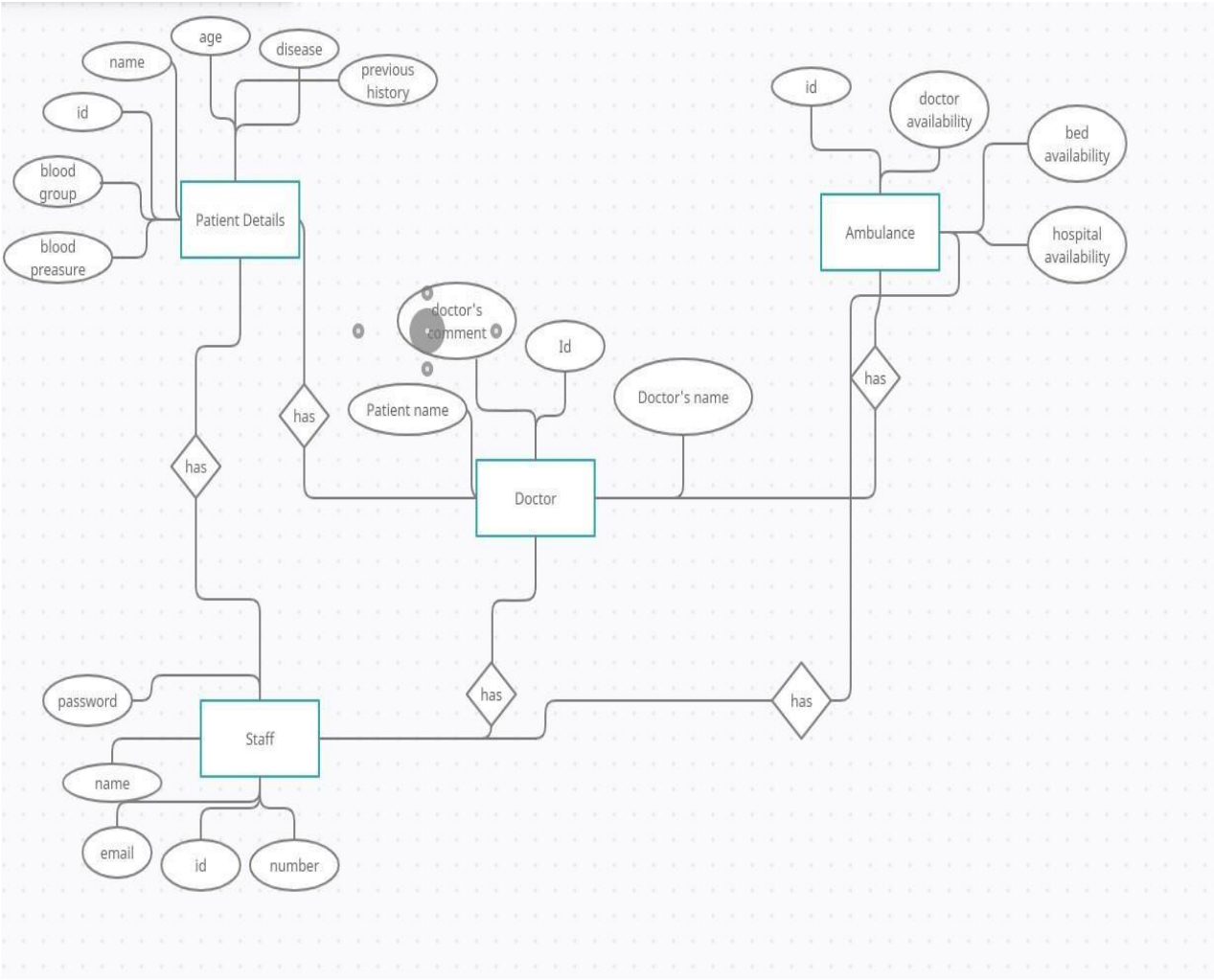
2.2 CLASS DIAGRAM



2.3 SEQUENCE DIAGRAM



2.4 E-R DIAGRAM



CHAPTER 3
ANALYSIS

3.1 PROPOSED SYSTEM

The Hospital Management System is designed for any hospital to replace their existing manual paper-based system. The new system is to control the information of patients. Room availability, staff and operating room schedules and patient invoices. These services are to be provided in an efficient, cost-effective manner, with the goal of reducing the time and resources currently required for such tasks.

3.2 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

3.2.1 ECONOMIC FEASABILITY

This study is carried out to check the economic impact will have on the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products have to be purchased.

3.2.2 TECHNICAL FEASABILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes for the implementing this system.

3.2.3 OPERATIONAL FEASABILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.3 SOFTWARE SPECIFICATION

3.3.1 HTML:

HTML or Hypertext Markup Language is the standard markup language used to create web pages. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>). HTML tags most come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example <image>. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). Though not always necessary, it is best practice to append a slash to tags which are not paired with a closing tag. The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language. HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms.

3.3.2 CASCADING STYLE SHEETS (CSS):

It is a style sheet language used for describing the look and formatting of a document written in a markup language. While most often used to style web pages and interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG and XUL. CSS is a cornerstone specification of the web and almost all web pages use CSS style sheets to describe their presentation.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colors, and fonts.[1] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content.

CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However, if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.

3.4 H2 DATABASE

H2 is an embedded, open-source, and in-memory database. It is a relational database management system written in Java. It is a client/server application. It stores data in memory, not persist the data on disk. Here we will be discussing how can we configure and perform some basic operations in Spring Boot using H2 Database.

CONNECTIVITY

STEP 1: To use the H2 database in the spring boot application we have to add the following dependency in the pom.xml file:

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

<scope>runtime</scope>

</dependency>

STEP 2: Write some properties in the application.properties file

H2 Database

spring.h2.console.enabled=true

spring.datasource.url=jdbc:h2:mem:dcbapp

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=password

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

Let’s understand what are these properties are by opening the H2 Database console.

H2 Console: By default, the console view of the H2 database is disabled. Before accessing the H2 database, we must enable it by using the following property.

spring.h2.console.enabled=true

Once we have enabled the H2 console, now we can access the H2 console in the browser by invoking the URL http://localhost:8082/h2-console.

3.5 JAVA

- **Object Oriented** − In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
- **Platform Independent** − Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.
- **Simple** − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.
- **Secure** − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- **Architecture-neutral** − Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.
- **Portable** − Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.

WHAT CAN SPRING BOOT DO?

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.

WHY SPRING BOOT?

- 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

HOW DOES IT WORK?

Spring Boot automatically configures your application based on the dependencies you have added to the project by using **@EnableAutoConfiguration** annotation. For example, if MySQL database is on your class path, but you have not configured any database connection, then Spring Boot auto-configures an in-memory database.

The entry point of the spring boot application is the class contains **@SpringBootApplication** annotation and the main method.

Spring Boot automatically scans all the components included in the project by using **@ComponentScan** annotation.

SPRING BOOT STARTERS

Handling dependency management is a difficult task for big projects. Spring Boot resolves this problem by providing a set of dependencies for developers convenience.

For example, if you want to use Spring and JPA for database access, it is sufficient if you include spring-boot-starter-data-jpa dependency in your project.

Note that all Spring Boot starters follow the same naming pattern spring-boot-starter- *, where * indicates that it is a type of the application.

EXAMPLES

Look at the following Spring Boot starters explained below for a better understanding

Spring Boot Starter Actuator dependency is used to monitor and manage your application. Its code is shown below −

```
<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-actuator</artifactId>

</dependency>
```

Spring Boot Starter Security dependency is used for Spring Security. Its code is shown below –

```
<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-security</artifactId>

</dependency>
```

Spring Boot Starter web dependency is used to write a Rest Endpoints. Its code is shown below −

```
<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>
```

Spring Boot Starter Thyme Leaf dependency is used to create a web application. Its code is shown below −

```
<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-thymeleaf</artifactId>

</dependency>
```

Spring Boot Starter Test dependency is used for writing Test cases. Its code is shown below −

```
<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

</dependency>
```

AUTO CONFIGURATION

Spring Boot Auto Configuration automatically configures your Spring application based on the JAR dependencies you added in the project. For example, if MySQL database is on your class path, but you have not configured any database connection, then Spring Boot auto configures an in-memory database.

For this purpose, you need to add **@EnableAutoConfiguration** annotation or **@SpringBootApplication** annotation to your main class file. Then, your Spring Boot application will be automatically configured.

Observe the following code for a better understanding −

```
import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.EnableAutoConfiguration;

@EnableAutoConfiguration

public class DemoApplication {

public static void main(String[] args) {

SpringApplication.run(DemoApplication.class, args);

}

}
```

SPRINGBOOT APPLICATION

The entry point of the Spring Boot Application is the class contains **@SpringBootApplication** annotation. This class should have the main method to run the Spring Boot application. **@SpringBootApplication** annotation includes Auto- Configuration, Component Scan, and Spring Boot Configuration.

If you added **@SpringBootApplication** annotation to the class, you do not need to add the **@EnableAutoConfiguration**, **@ComponentScan** and **@SpringBootConfiguration** annotation. The **@SpringBootApplication** annotation includes all other annotations.

Observe the following code for a better understanding −

```
import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class DemoApplication {

public static void main(String[] args) {

SpringApplication.run(DemoApplication.class, args);

}

}
```

COMPONENT SCAN

Spring Boot application scans all the beans and package declarations when the application initializes. You need to add the **@ComponentScan** annotation for your class file to scan your components added in your project.

Observe the following code for a better understanding −

```
import org.springframework.boot.SpringApplication;

import org.springframework.context.annotation.ComponentScan;

@ComponentScan

public class DemoApplication {

public static void main(String[] args) {

SpringApplication.run(DemoApplication.class, args);

}

}
```

CHAPTER 4

SNAPSHOTS

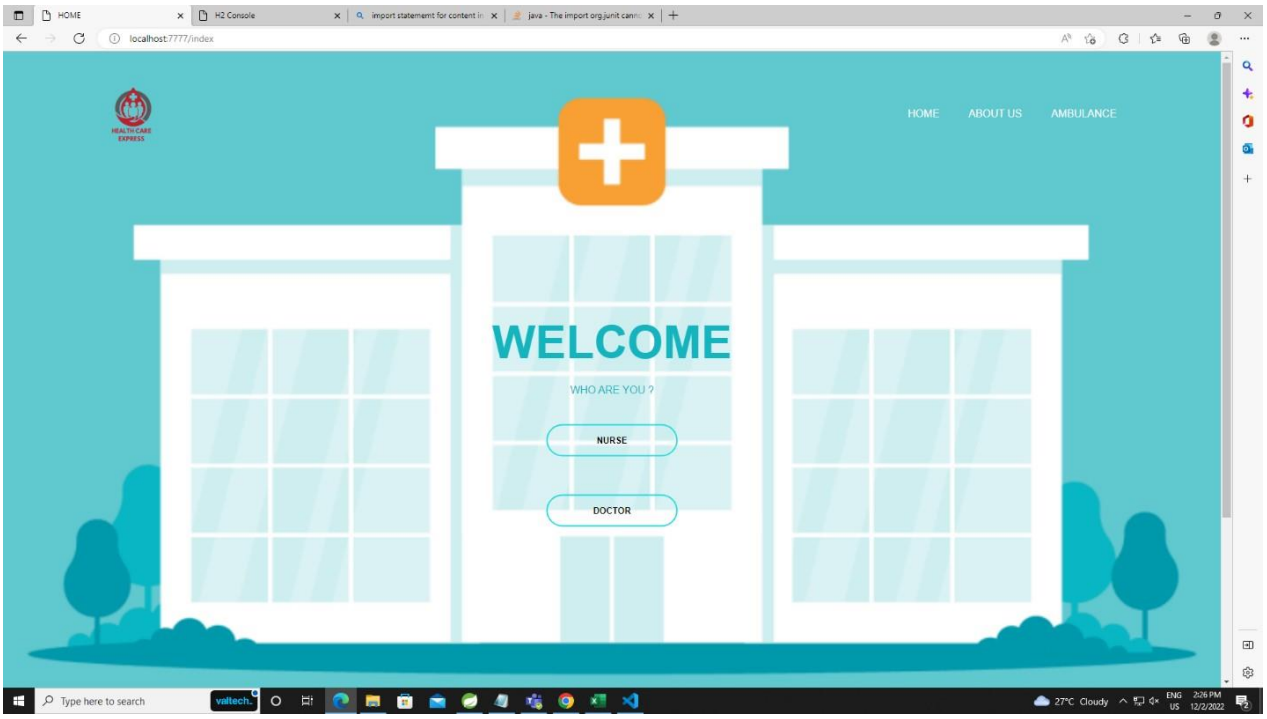


FIG 1: INDEX PAGE

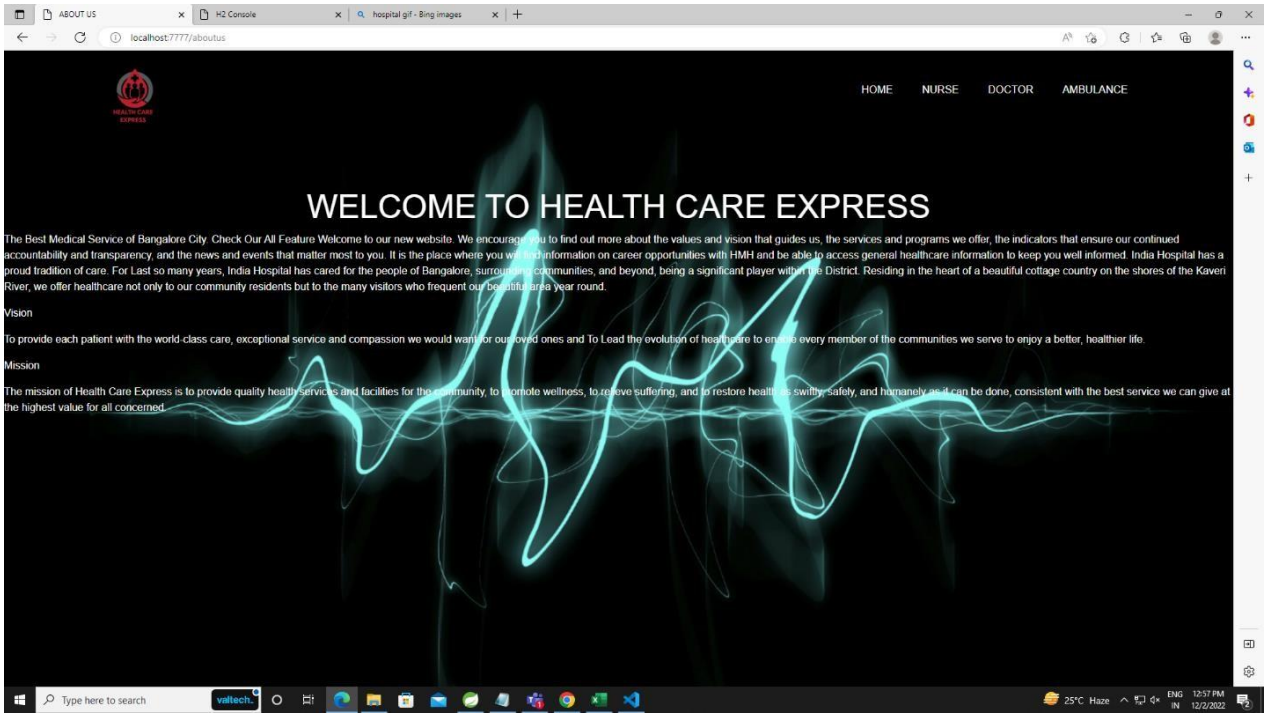


FIG 2: ABOUTUS PAGE

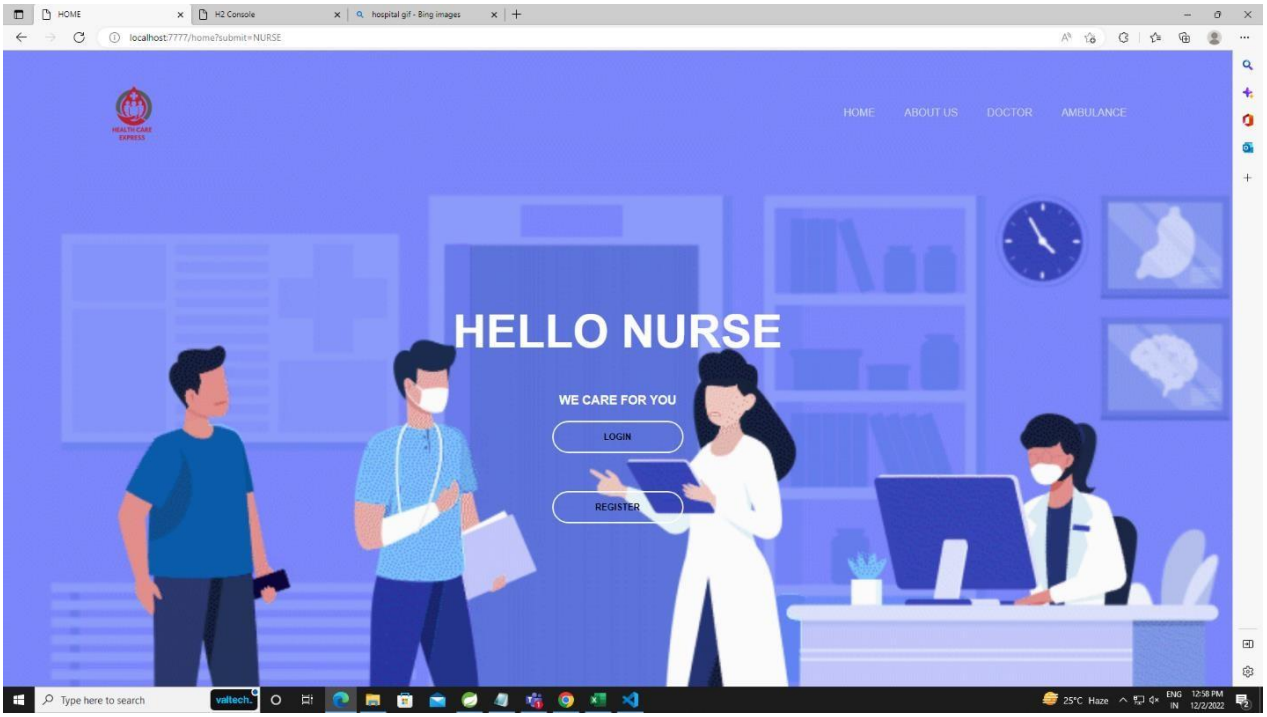


FIG 3:HOME PAGE

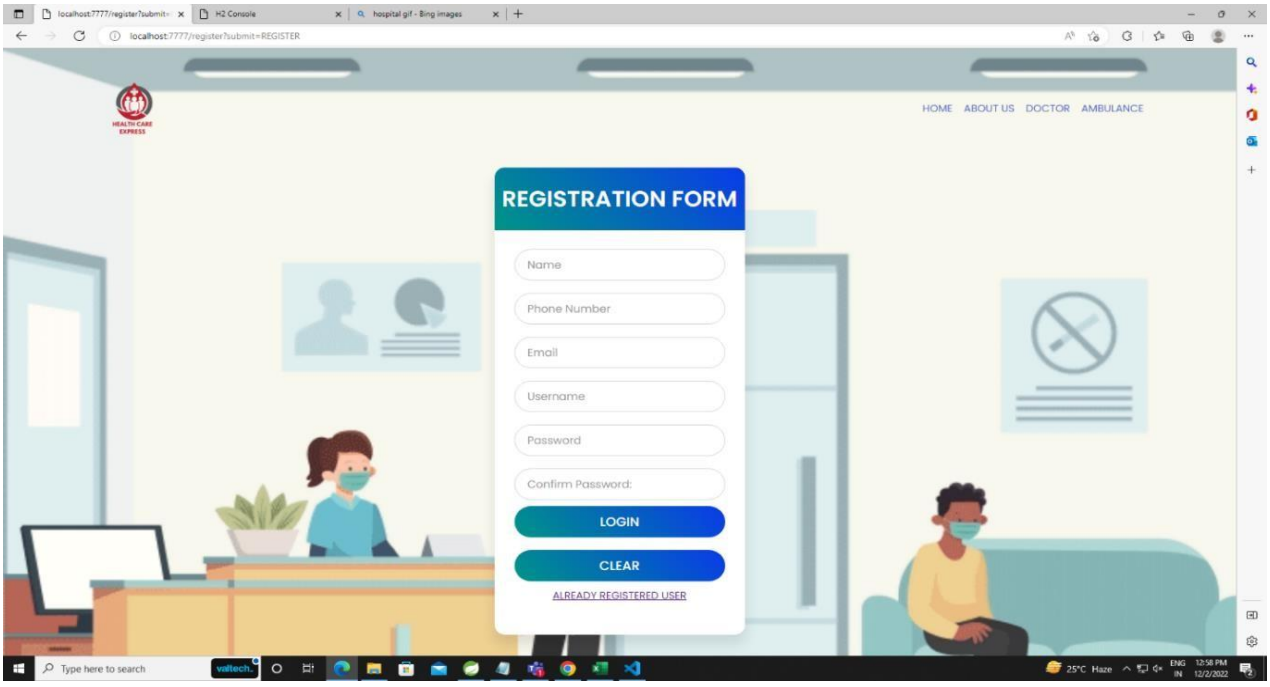


FIG 4:REGISTRATION PAGE

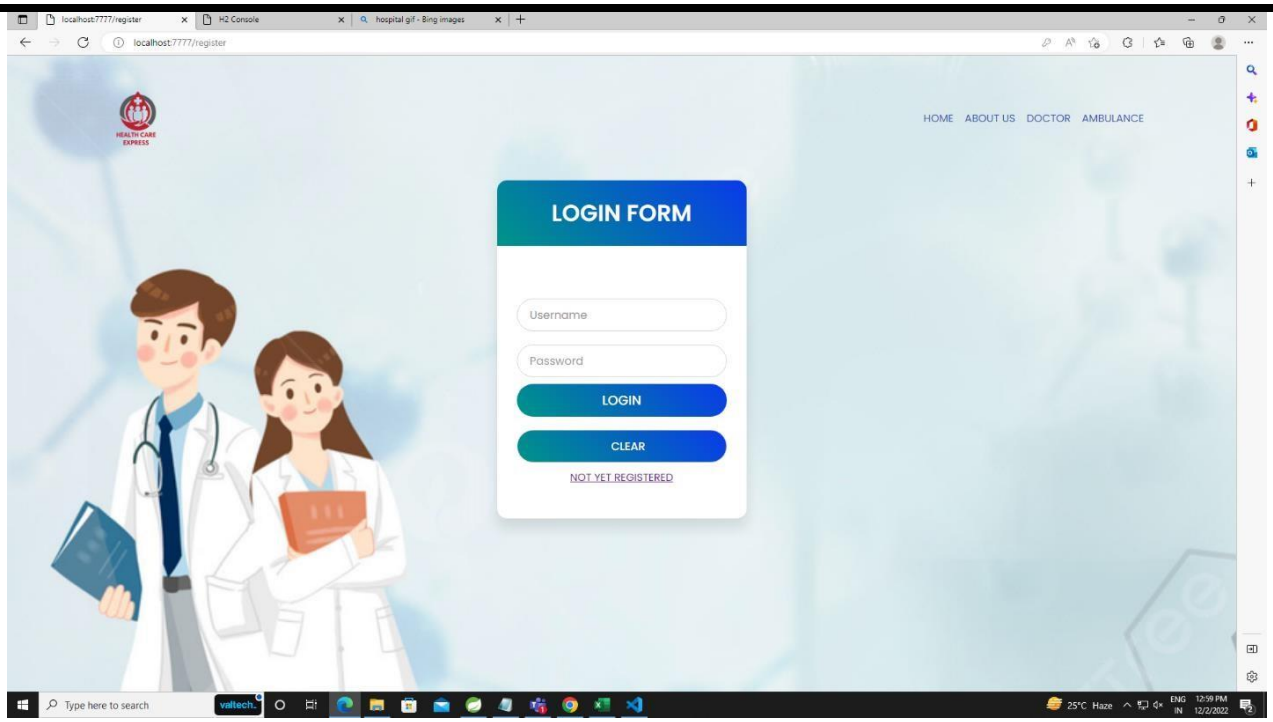


FIG 5:LOGIN PAGE

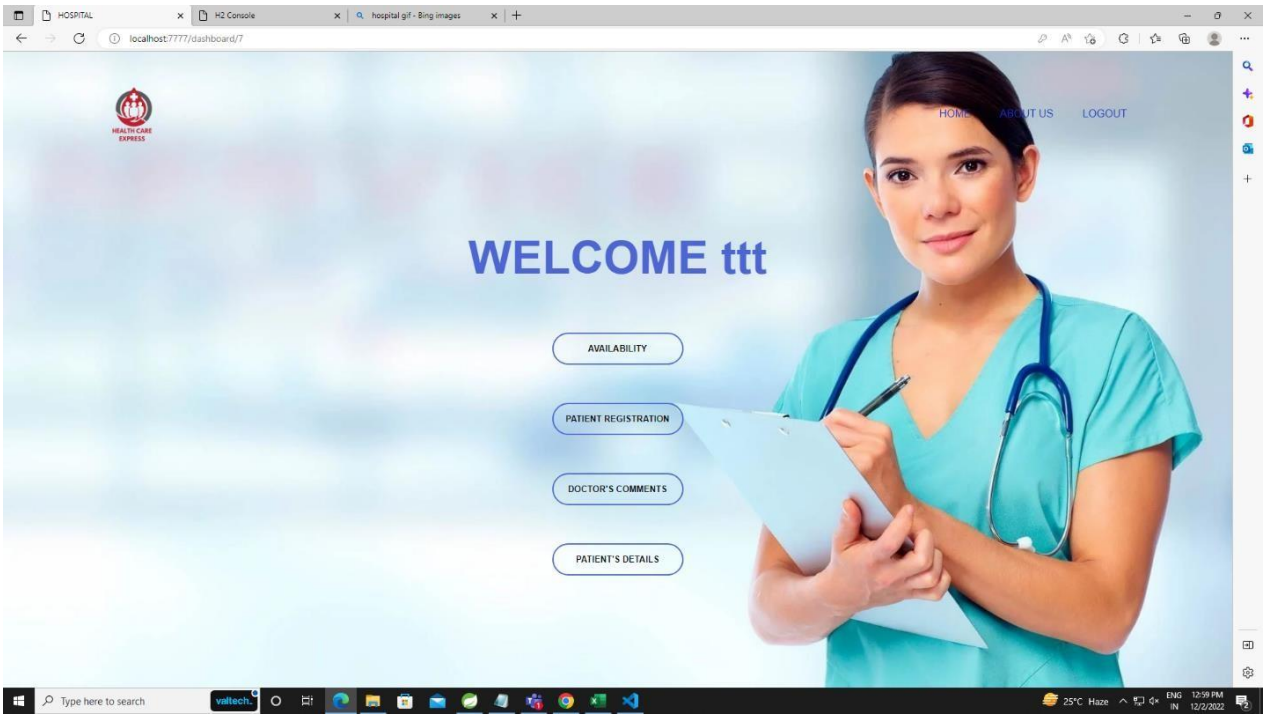


FIG 6:DASHBOARD PAGE

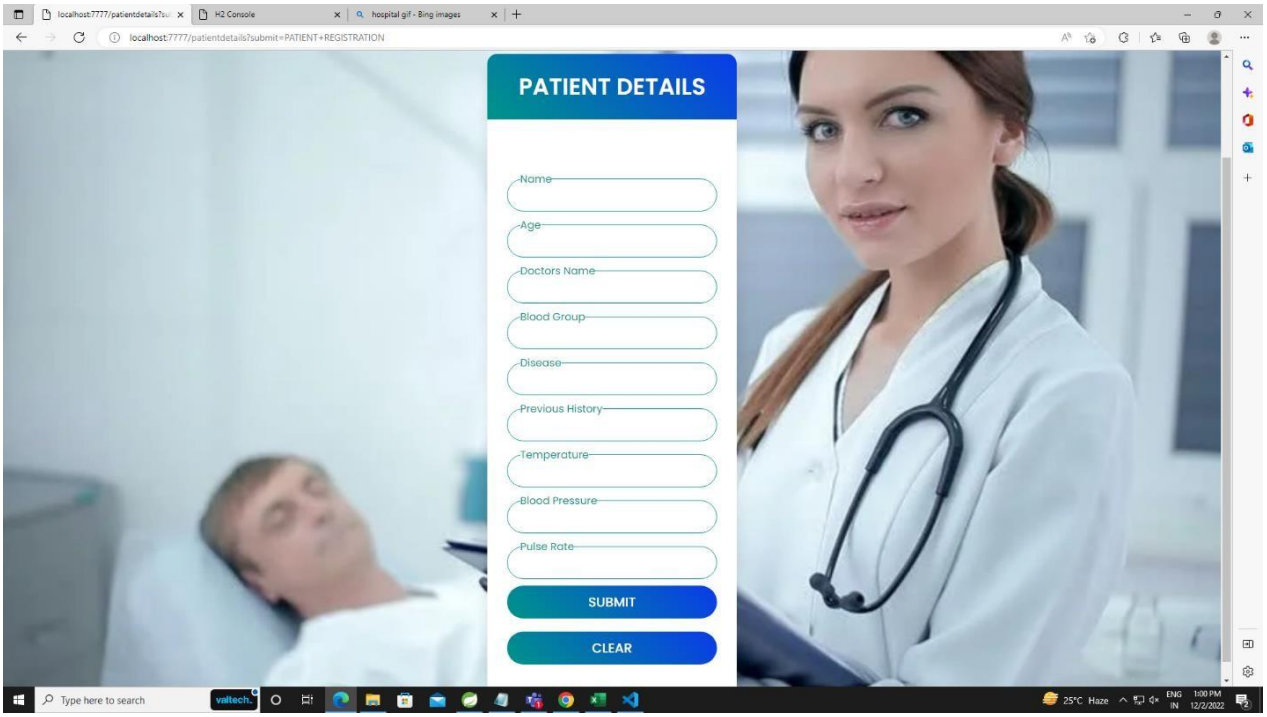


FIG 7: PATIENT DETAILS PAGE

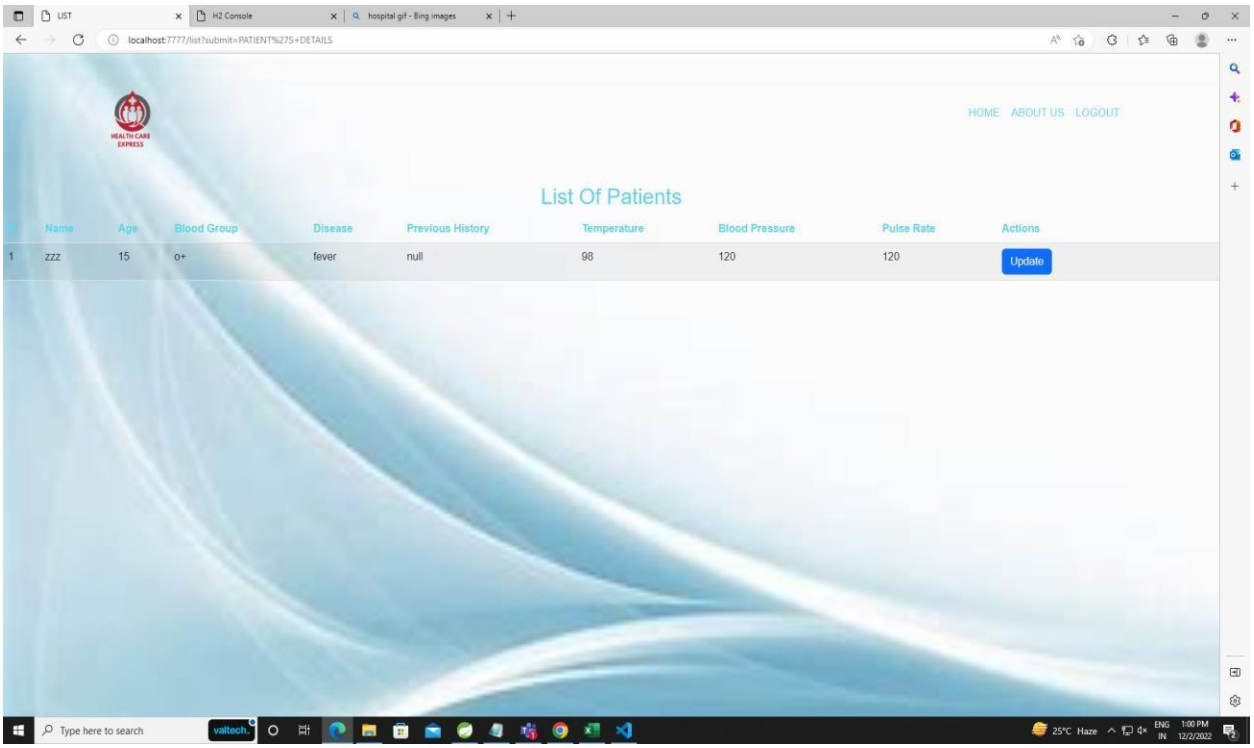


FIG 8: LIST OF PATIENTS PAGE

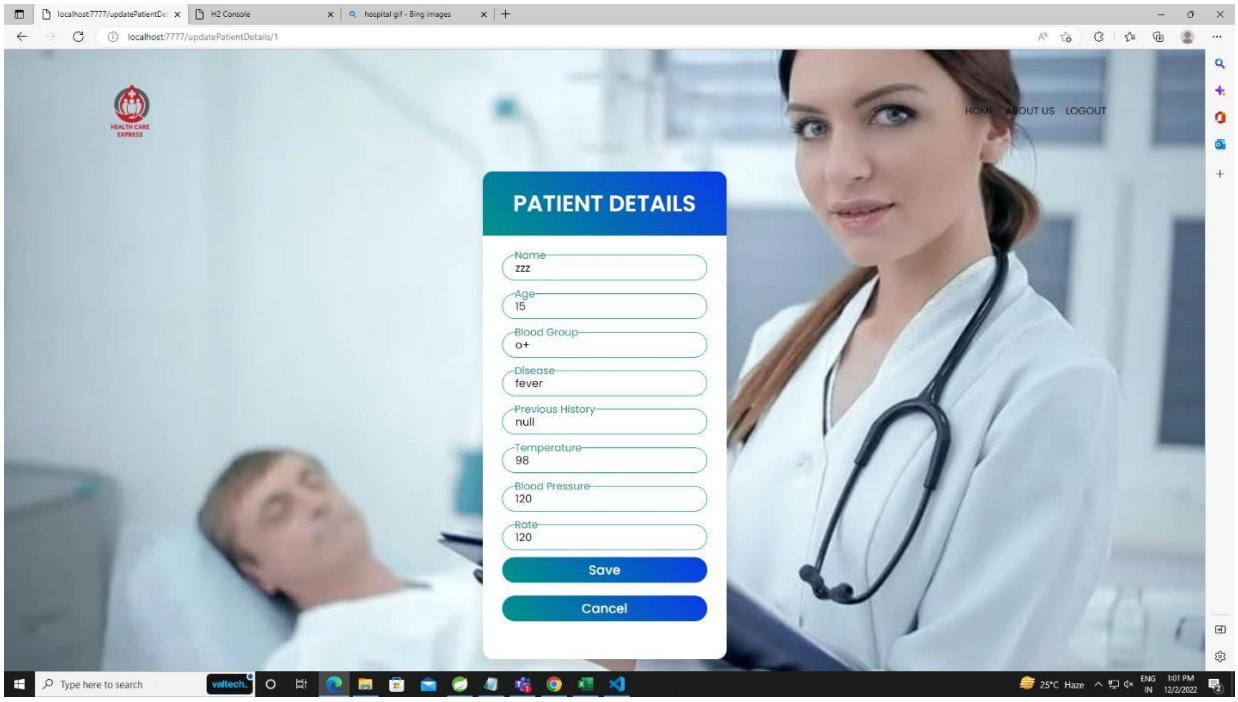


FIG 9: UPDATE PATIENT DETAILS PAGE

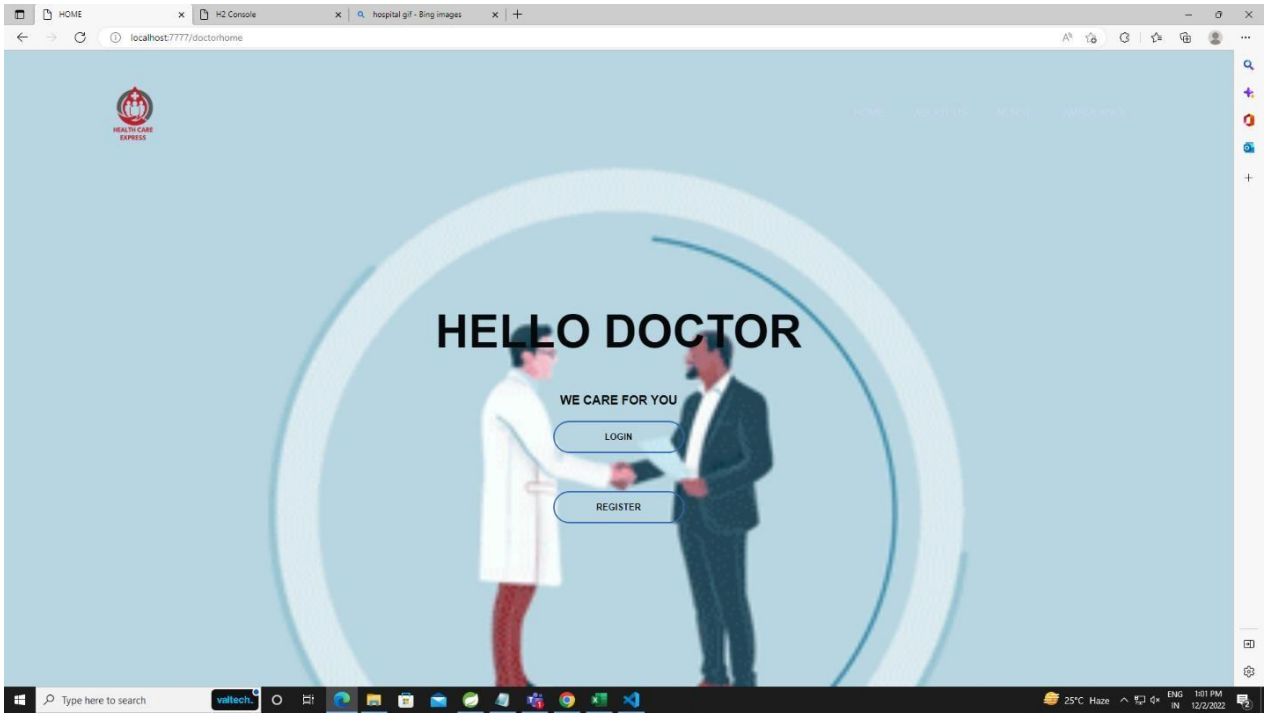


FIG 10: DOCTOR HOME PAGE

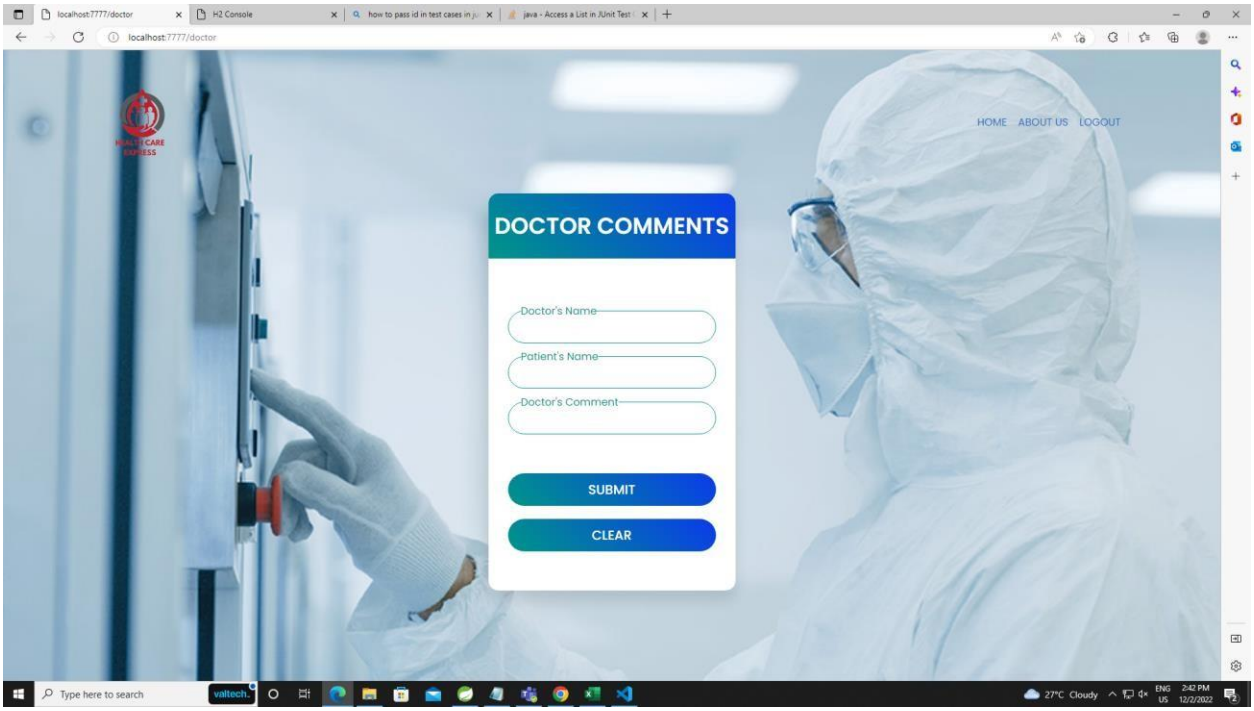


FIG 11: DOCTOR'S COMMENT ON PATIENT PAGE

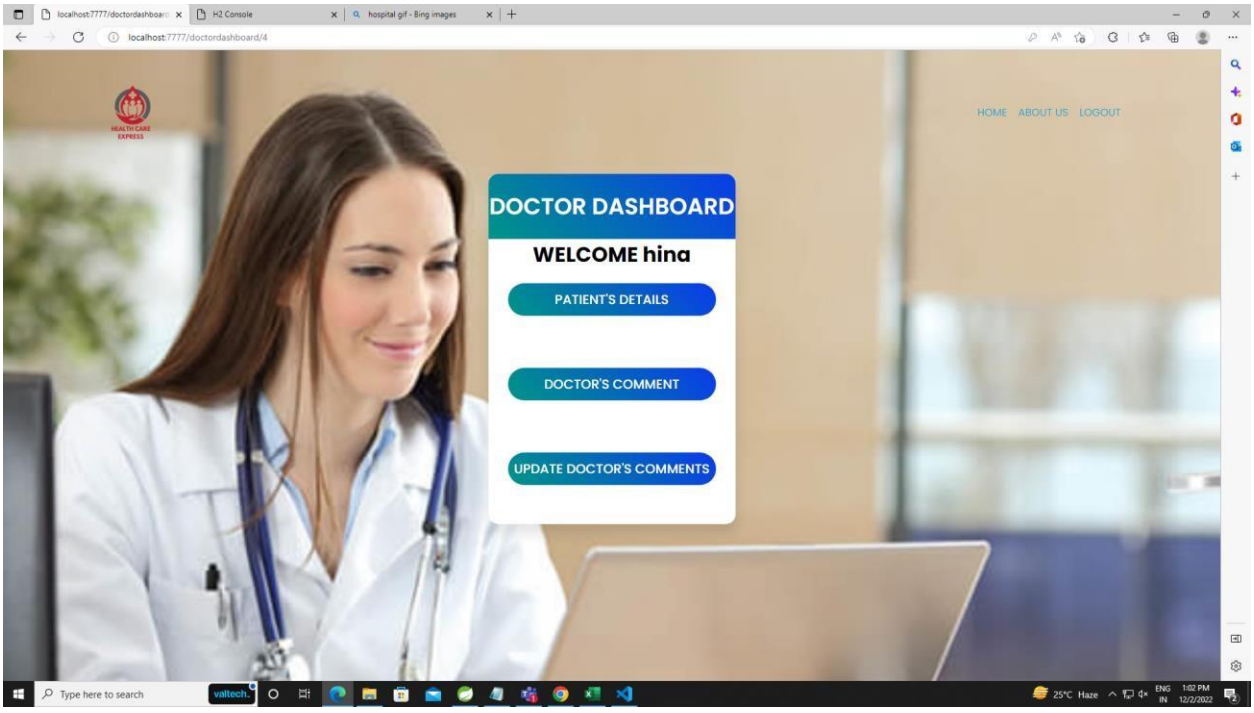


FIG 12: DOCTOR DASHBOARD PAGE

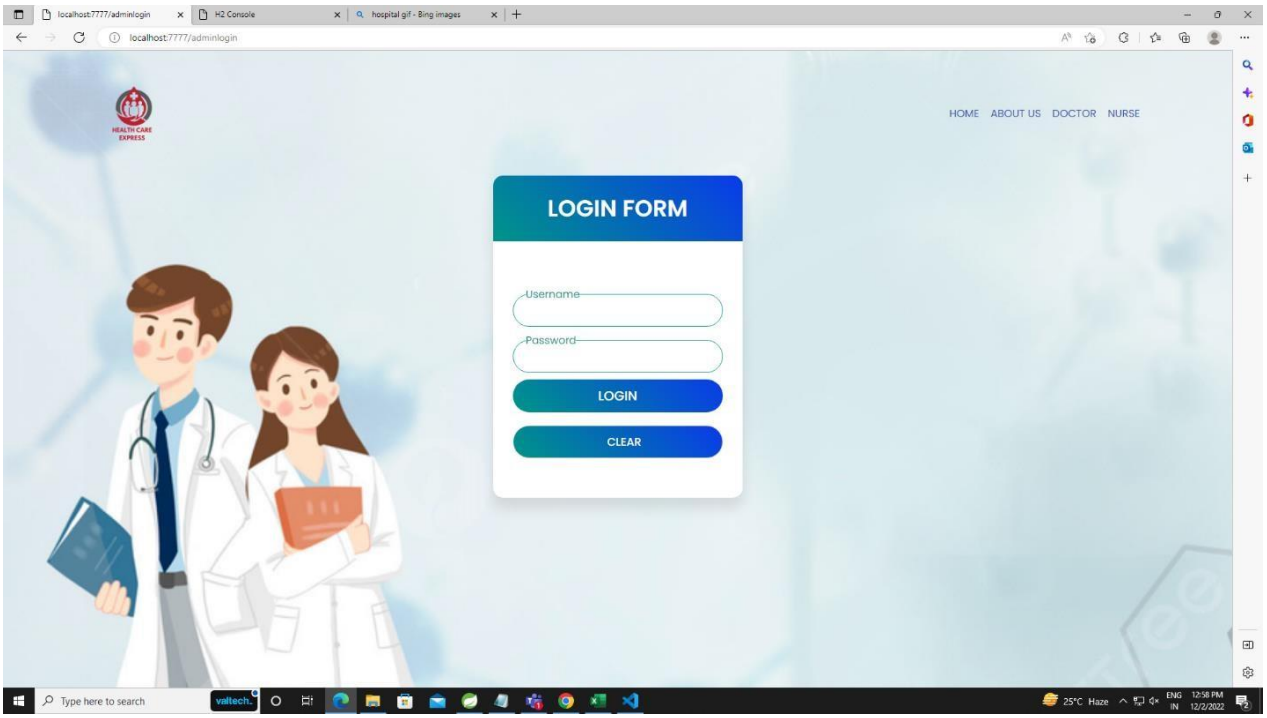


FIG 13:ADMIN LOGIN PAGE

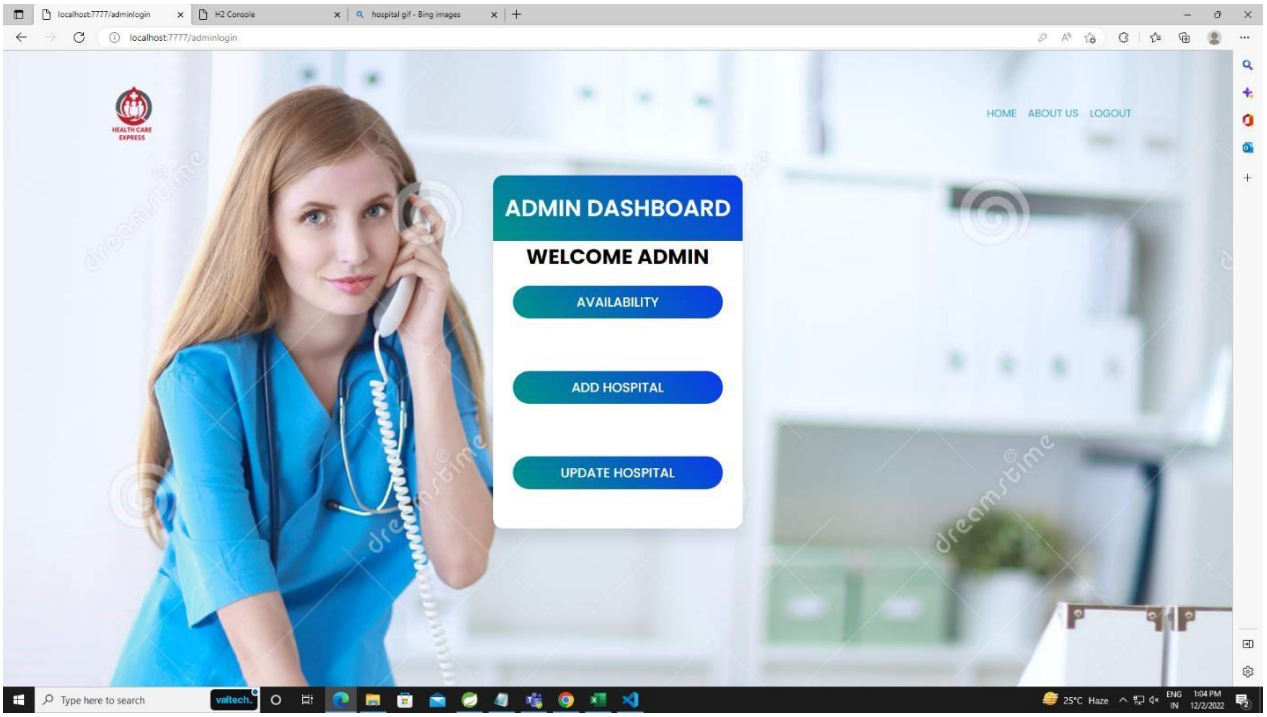


FIG 14: ADMIN DASHBOARD PAGE

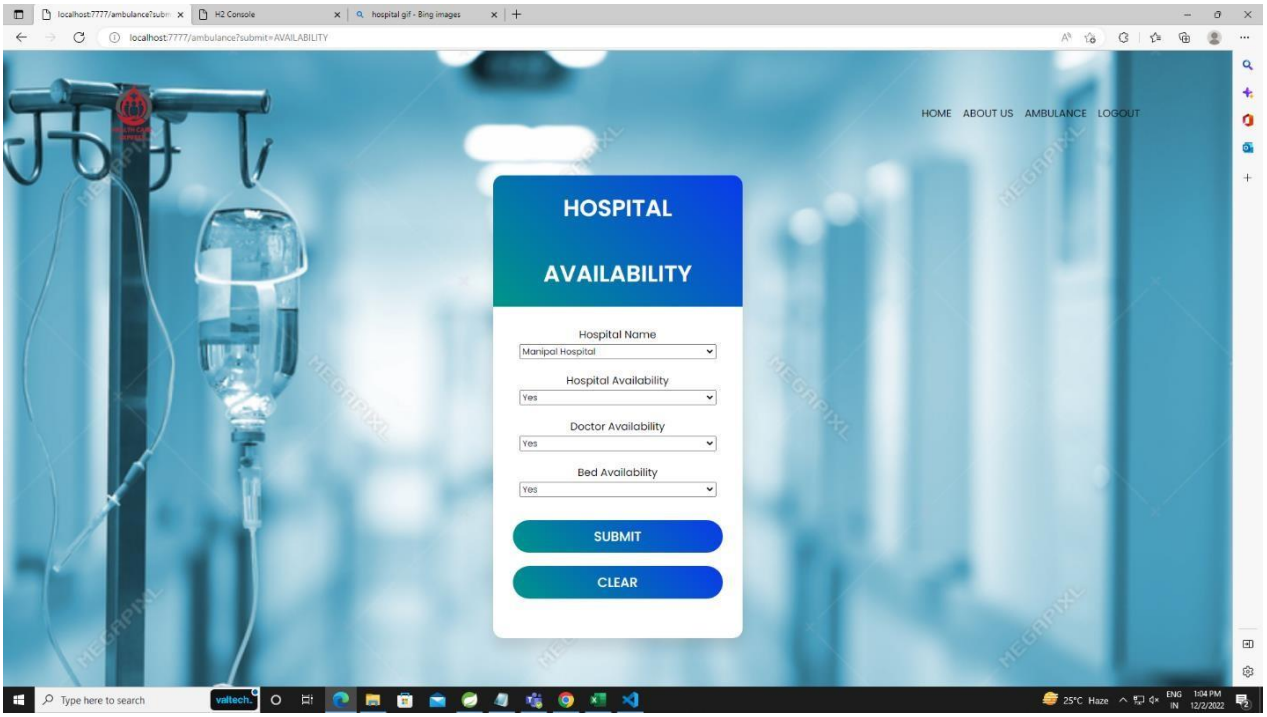


FIG 15: ADMIN LIST PAGE

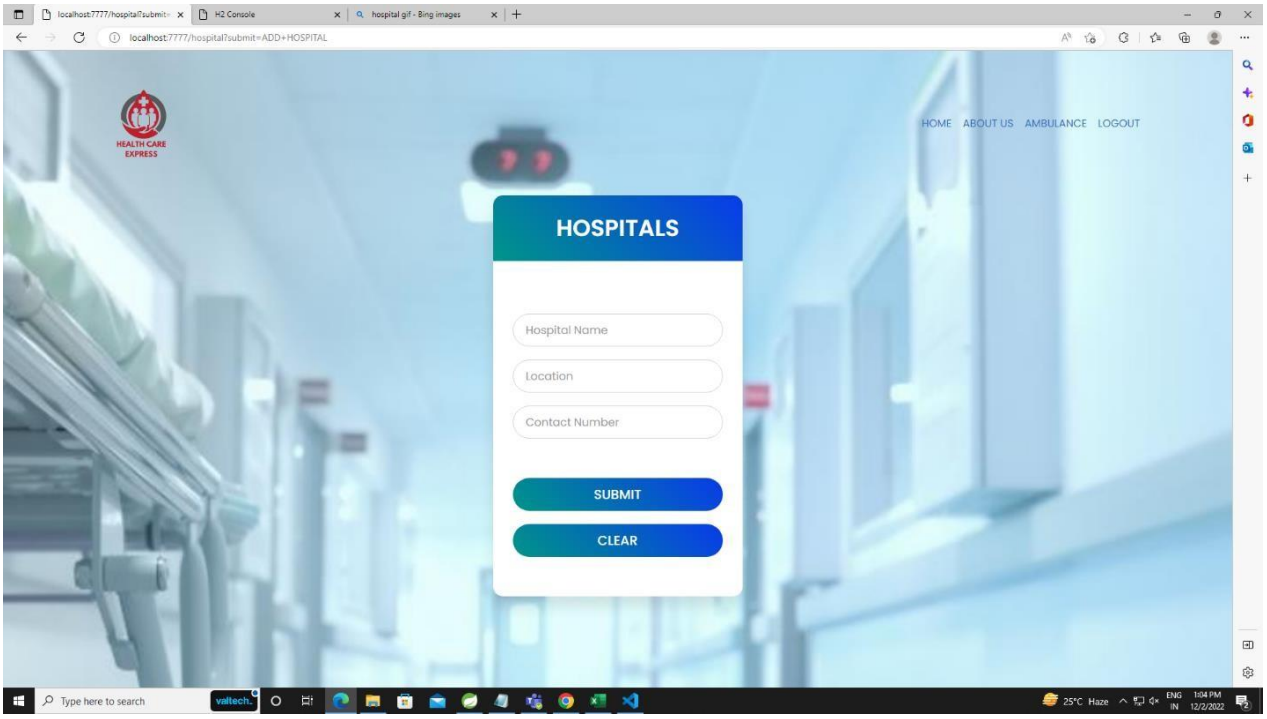


FIG 16:ADMIN UPDATE PAGE