

KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

(APPROVED BY AICTE, NEW DELHI)

Department of Computer science and Engineering



Project Report

On

“ Hospital Management System “

Submitted in the partial fulfilment for the award of the degree of

Bachelor of Engineering

In

Computer Science and Engineering

Submitted by

Vishwanath

2GI18CS188

Under the Guidance of

Ass.Prof. Sudha V S

(KLS GOGTE INSTITUTE OF TECHNOLOGY)

2021 – 2022

KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

(APPROVED BY AICTE, NEW DELHI)

Department of Computer Science



CERTIFICATE

Certified that the project entitled **"Hospital Management System"** carried out by **Mr. Vishwanath USN. 2GI18CS188** student of KLS Gogte Institute of Technology, Belagavi, can be considered as a bonafide work in partial fulfilment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi during the year 2021-2022. It is certified that all corrections/suggestions indicated have been incorporated in the report. The project report has been approved as it satisfies the academic requirements prescribed for the said Degree.

Guide

Co-Guide

HOD

Principal

Final Viva-Voce

	Name of the examiners	Date of Viva -voce	Signature
1.			
2.			

Declaration by the student

I, **Vishwanath** hereby declare that the project report entitled “ **Hospital Management System** ” submitted by me to KLS Gogte Institute of Technology, Belagavi, in partial fulfilment of the Degree of Bachelor of Engineering in Computer Science and Engineering is a record of the project carried out under the guidance of **Ass.Prof. Sudha V S**. This report is for the academic purpose. I further declare that the report has not been submitted and will not be submitted, either in part or full, to any other institution and University for the award of any diploma or degree.

Place: Belgaum

Name of the student: Vishwanath

Date: 16/06/2022

USN:2GI18CS188

Signature of the student

Acknowledgement

I take this opportunity to express my gratitude to all those people who have been instrumental in making this project successful. I feel honoured to place warm salutation to K.L.S. GOGTE INSTITUTE OF TECHNOLOGY, Belagavi, which gave the opportunity to pursue our BE at this esteemed institute.

I express my profound gratitude to our Principal, **Dr Jayant Kittur** for his warm support throughout the B.E. program, express my sincere gratitude to our Professor and Head, Computer Science and Engineering **Dr. V. S. Rajpurohit** for his constant encouragement and support throughout our project. It is our privilege to express my sincere regards to our project guide **Ass.Prof. Sudha V S** for her valuable inputs, able guidance, encouragements, wholehearted cooperation for the implementation of our project.

I take this opportunity to thank our professors, non-teaching staff, technical staff who have directly or indirectly helped us in our project. I pay my respect to our parents, family members and friends for their support and encouragement throughout our project.

VISHWANATH

ABSTRACT

Our project help to manage the hospital properly. Our project is for the hospital management and the workers. Our project uses the following azure services:

- It uses Static Web App for hosting the application.
- It uses Azure functions for computations.
- It uses Azure Database for MySQL to store the regular data.
- It uses Key Vault to store sensitive data.
- It uses Azure Blob Storage to store images.

Our project is trying to solve the problems faced by the hospital management/workers to manage the hospital properly. We are solving this problem by developing a system that is user friendly and that can be used by newbies.

Table Of Contents:

Chapter no.		Contents	Page no.
		Declaration	i
		Acknowledgement	ii
		Abstract	iii
		Table of Contents	iv
		Table of figures	vii
1.		Introduction	1 to 2
	1.1	Problem Statement	2
	1.2	Objectives	2
	1.3	Project Description	2
	1.4	Opportunities and Complications	2
2.		Literature Survey	3 to 8
3.		Requirements specifications	9 to 10
	3.1	Functional Requirement	9

	3.2	Non-Functional Requirement	10
4.		Design Details	11 to 12
	4.1	Use case diagram	11
	4.2	Sequence diagram	12
5.		Technology Used	13 to 17
	5.1	Programming Language (Front-End): React	13 to 14
	5.2	Programming Language (Back-End): Spring-Boot	15 to 17
	5.3	Restful API's	18 to 19
	5.4	Platform: Microsoft Azure	20 to 22
	5.5	Development IDE: VS Code	23 to 24
	5.6	Operating System: Windows/Linux/MacOS	25
	5.7	Fundamental Subjects/Concepts Used in the Project	26 to 27
6.		Implementation	28 to 30
	6.1	Sample source code	28 to 30
7.		Results and Discussion	31 to 38
	7.1	Output Screenshot	31 to 37

	7.2	Live project link	38
		Conclusions and Future scope	39
		Conclusion	39
		Future Scope	39
		References	40

List Of Figures

Fig. No	Figure	Page no.
4.1	Use case diagram	11
4.2	Sequence diagram	12
5.1	React logo	13
5.2	Spring-boot logo	15
5.3	Rest-API	18
5.4	Microsoft Azure logo	20
5.5	VS Code logo	23
5.6	Windows, MacOS and Linux logo	25
5.7	HTML , CSS , JS	26
5.8	Java and spring logo	26
5.9	H2 database	27
7.1	Welcome logo screen	31
7.2	Welcome screen	31
7.3	Login screen	32

7.4	Login screen when wrong password entered	32
7.5	Dashboard screen	33
7.6	Add new patient profile screen	33
7.7	Patient added successfully screen	34
7.8	Add new doctor profile screen	34
7.9	Doctor added successfully screen	35
7.10	Schedule an appointment screen	35
7.11	Appointment scheduled successfully screen	36
7.12	Generate bill screen	36
7.13	Invoice in PDF format screen	37
7.14	Future scope	39

Chapter 1

Introduction

Hospital management system is a computer system that helps manage the information related to health care and aids in the job completion of health care providers effectively. They manage the data related to all departments of healthcare such as, Clinical. Financial. Laboratory.

Microsoft Azure, formerly known as Windows Azure, is Microsoft's public cloud computing platform. It provides a range of cloud services, including compute, analytics, storage and networking. Users can pick and choose from these services to develop and scale new applications, or run existing applications in the public cloud.

The Azure platform aims to help businesses manage challenges and meet their organizational goals. It offers tools that support all industries -- including e-commerce, finance and a variety of Fortune 500 companies -- and is compatible with open source technologies. This provides users with the flexibility to use their preferred tools and technologies. In addition, Azure offers 4 different forms of cloud computing: infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS) and serverless.

Microsoft charges for Azure on a pay-as-you-go basis, meaning subscribers receive a bill each month that only charges them for the specific resources they have used.

Once customers subscribe to Azure, they have access to all the services included in the Azure portal. Subscribers can use these services to create cloud-based resources, such as virtual machines (VM) and databases.

Problem Statement:

Develop a hospital management system with the help of Microsoft azure services.

Objectives:

- Design a system for better patient care.
- Reduce hospital operating costs.
- Provide MIS (Management Information System) report on demand to management for better decision making.
- Better co-ordination among the different departments.
- Provide top management a single point of control.

Project Description:

Our project help to manage the hospital properly. Our project is for the hospital management and the workers. Our project uses the following azure services:

- It uses Static Web App for hosting the application.
- It uses Azure functions for computations.
- It uses Azure Database for MySQL to store the regular data.
- It uses Key Vault to store sensitive data.
- It uses Azure Blob Storage to store images. Our project is trying to solve the problems faced by the hospital management/workers to manage the hospital properly. We are solving this problem by developing a system that is user friendly and that can be used by newbies. Our project functionalities are mapped to the problem statement.

Opportunities and Complications:

This project will help in the management of hospitals. Already many solutions are there to solve the problem but they are not user friendly, they are complex and harder to understand. Our project is user friendly which will help the hospital management/workers to use the system properly even if they are a newbie and take appropriate decisions.

Chapter 2

Literature Survey

Cloud Computing and Healthcare

The healthcare industry has experienced a much needed technological overhaul in recent years, which has been ushered in by developments in healthcare information technology (HIT) and regulatory reform. As a result, technology aimed at building collaboration and coordination in this sphere, once limited, is now seeing a welcomed boom. Leading the charge in the modernization of healthcare-related technological tools is cloud technology. This is because the industry is on a trajectory that calls for a shift toward an information centric and consumer-driven care delivery model.

In response, cloud computing has emerged as the go-to solution, because it provides the infrastructure needed for healthcare systems to leverage more advanced technological resources; thereby, yielding the following benefits, among a host of others: on-demand access to massive data storage capabilities—which is especially pertinent as large data sets like electronic health records (EHR) are managed. Enhanced collaboration opportunities for such data are also enabled, as cloud solutions diminish limitations traditionally experienced due to geographical restrictions or departmental silos; such real-time sharing of data can add tremendously to optimized and evidentiary patient care delivery.

There's also the flexibility to scale up or down according to organizational needs and the ability to streamline workflows and operational facets. Lastly, but certainly not least, there are cost saving advantages. With these benefits in mind, it's no wonder that a recent Healthcare Information and Management Systems Society (HIMSS) Analytics' survey of cloud computing adoption in healthcare provider organizations found that 83% of IT executives report they are currently using cloud services.

Along with the increased adoption of technologies in this realm comes the parallel need for stringent requirements for security, confidentiality and accessibility. While these advancements are meant to improve quality of care and patient safety, concerns of security cannot be ignored.

Hence, cloud solutions must be able to meet stringent governmental and industry regulations and standards made to safeguard such highly sensitive data; and in turn the privacy of all associated individuals.

Microsoft Azure has emerged as a trusted solution to address present challenges. In fact, according to the HIMSS Analytics survey, Azure was most frequently cited as the cloud solution for health organizational demands.

HMS came into the picture of hospital management as early as 1960 and have ever since been evolving and synchronizing with the technologies while modernizing healthcare facilities. In today's world, the management of healthcare starts from the hands of the patients through their mobile phones and facilitates the needs of the patient.

Why is HMS important for a hospital?

HMS was introduced to solve the complications coming from managing all the paper works of every patient associated with the various departments of hospitalization with confidentiality. HMS provides the ability to manage all the paperwork in one place, reducing the work of staff in arranging and analyzing the paperwork of the patients. HMS does many works like:

- Maintain the medical records of the patient
- Maintain the contact details of the patient
- Keep track of the appointment dates
- Tracking the bill payments.

The advantages of HMS can be pinpointed to the following:

- Time-saving Technology
- Improved Efficiency by avoiding human errors
- Reduces scope for Error
- Data security and correct data retrieval made possible
- Cost effective and easily manageable
- Easy access to patient data with correct patient history
- Improved patient care made possible
- Reduces the work of documentation

Features of Hospital Management System:

Appointment Management

For hospitals having their own site, appointment widgets will be integrated onto the site. Patients visiting the hospital's website can book online appointments with ease.

Billing Management

Integrated Billing with treatments, Lab and Radiology. Alerts will be sent on Discount Authorization. Automatic due capture, Option to bill before and after consultation.

Prescription Management

Manage commonly and recently used medicines. Option to show medicines available in the pharmacy. SMS prescriptions to Patients.

Discharge Summary

Template based Discharge Summary. ICD10 integration. Option to prevent discharge summary till IP bill is closed.

Operation Theatre Management

Automatic notification can be sent to customers on test results. Lab notifications like email, SMS of the test reports sent from the Automated Lab notification module.

Pharmacy Management

Comprehensive Pharmacy Management handles stock, Prescription Integration, Ward Request, Stock Management, Stock Moment and intelligent reports.

Lab Management

Comprehensive Lab Management handles complete order management, Custom Reports, Smart Notifications, Credit Settlement, detailed MIS Reports, Analytics and App for Phlebotomist.

Master Information Systems

Lets you access entire MIS data from your palm.

Manage Multiple Locations

Any number of branches can be added and managed using a single account.

Benefits of Hospital Management System:

Easy Patient data retrieval:

HMS makes it possible to access all the data related to a patient via a system by the means of a few simple clicks. Information like patient history, current illness, doctors involved, tests reports taken, billing information and many more can be made visible to the user. These data will help to connect the dots about the patient, like specific diagnosis, related treatment, and medication.

The Electronic Medical Record (EMR) or Electronic Health Record (EHR):

This electronic-based medical record system can be viewed as a patient's health chart. It retrieves information based on the patient's name or medical record number or the physician's record number.

Increased Data security:

The patient data can be kept a hundred percent safe by using HMS in your hospital. It can be made accessible by only a limited amount of authorized personnel. With HMS, all the data is stored on a server or cloud and kept safe by just securing the login information safe.

Improve Visibility and Transparency:

Hospital Management System (HMS) improves the visibility and transparency in the complete management process and in all records.

Streamline Accurate Reporting:

It helps in streamlining the accurate reporting with the help of updated and accurate records.

Improved Quality Control:

Hospital Management System improves the quality control on the products and services of the hospital.

Improved Management Visibility:

It also improves the management visibility of hospital, all information, and data regarding the patient, doctor and medicine could be seen by any department easily.

Ease to Access System Facilities:

Hospital Management System makes it easy to get access to the management system facilities for the authorized users and keep it safe from unauthorized users.

Cost Effective:

HMS not only saves time in the hospital but also is cost-effective in decreasing the number of people working on the system of manual entry of data and paperwork. The implementation of His will decrease the human intervention into the system thereby avoiding human-caused errors.

Chapter 3

Requirements Specifications:

Functional Requirements

Hospitals have various types of functional requirements from patient registration, to scheduling consultations, to report generation. HMS helps in several functional requirements of a hospital, a few of which are as follows.

- **Undertaking various registration**

HMS is able to facilitate various registration needs in a hospital. The system not only records all relevant information about the patient but also assigns a specific ID that uniquely identifies the individuals throughout the patient lifecycle in the hospital. This essentially means the operators in all subsequent departments, say diagnostics, IPD registration, physiotherapy or rehabilitation care, pharmacy, etc need not again record patient data but retrieve the information from the unique ID. This helps in uniquely integrating the entire treatment lifecycle, health data, and information pertaining to an individual automatically, thereby building transparency, reducing redundant and cyclical activities, infusing efficiency, and saving time and money by automating processes.

- **Generating various reports**

Hospitals need to generate various reports which include those that are related to patients and those that are pertaining to administration. By integrating all information flow and data recording through HMS, the generation of various reports becomes rather easy. HMS, therefore, helps record relevant information and relays them as per need, helping both treatments of patients and administration of hospitals.

- **Database**

HMS helps in maintaining the database of patient information and health data in a transparent, secure, and accurate way. It makes recording and storage of the information rather easy. The platform facilitates almost instantaneous retrieval of data when required.

Non-functional requirements

Every hospital has some non-functional requirements, such as security, performance, maintenance, reliability, and so on. These are essential aspects of the entire healthcare delivery ecosystem, as they are not limited to one or few functions but are overarching requirements. HMS is able to cater to these non-functional requirements as well.

- **Security**

HMS oversees humongous volumes of data generation, information exchange, storage, and analysis at every level of hospital functioning. As it is based on Cloud and other advanced digital technologies, it offers strong, multi-layered security to all data exchanges, and thereby protects the system from misuse or loss of information. The HMS platforms usually comply with the most stringent data security and privacy policies set in a country. As HMS is hosted on Cloud-based servers which are located away from the premises, it remains protected from cyberattacks on hospital systems. Overall HMS promotes transparency, protects the confidentiality, prevents data theft, and offers a safe and secure ecosystem for hospitals operations to continue.

- **Performance**

By streamlining and integrating multiple processes, HMS infuses much speed, agility, and efficiency into the system. The platform has specially designed modules for various functions such as OPD management, IPD management, Cath Lab and diagnostics management, emergency care response, billing and payments, and operations. It has the ability to offer role-based control to users to allow them the use of one part of the function or multiple functions and help them monitor and track every activity necessary for healthcare delivery. Due to such intra-operability and flexible properties, HMS boosts the performance and capabilities of a healthcare facility in treating patients.

- **Maintainability**

Good HMS technologies are easy to maintain. They are usually SaaS-based platforms that can be upgraded and improved upon remotely without hindering the daily activities of hospitals. Its maintenance activities can be pre-scheduled in parts considering when the activity is lowest, and it doesn't take long to upgrade.

- **Reliability**

Being software as a service, HMS is highly resilient to any technology disruptions, downtime, or crashes experienced by other technology systems. It has a certain capacity to work offline. It is highly secure from a data safety point of view. Furthermore, good HMS has a highly instinctive and intelligent user interface which makes them convenient to use.

Chapter 4

Design Details

Use Case Diagram

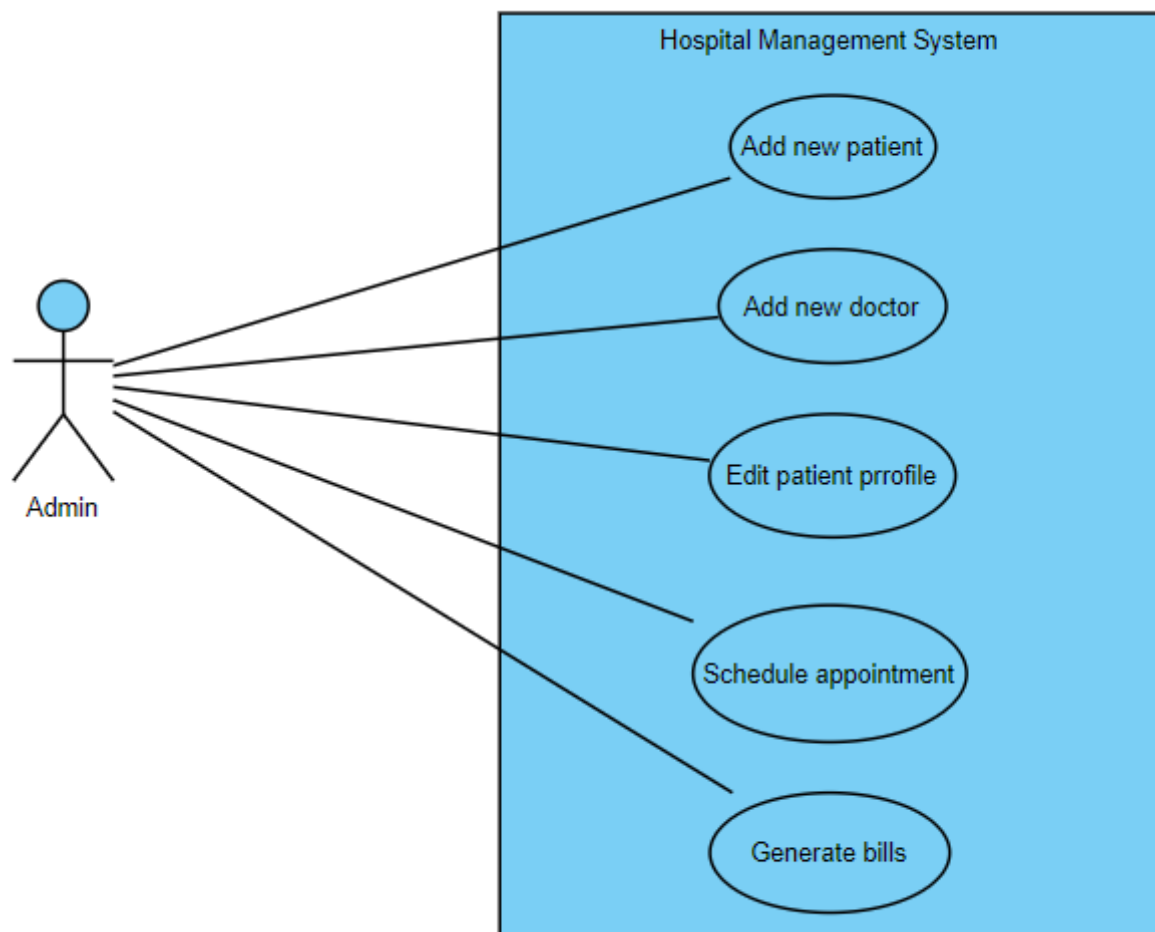


Figure 4.1: Use case diagram

Sequence Diagram

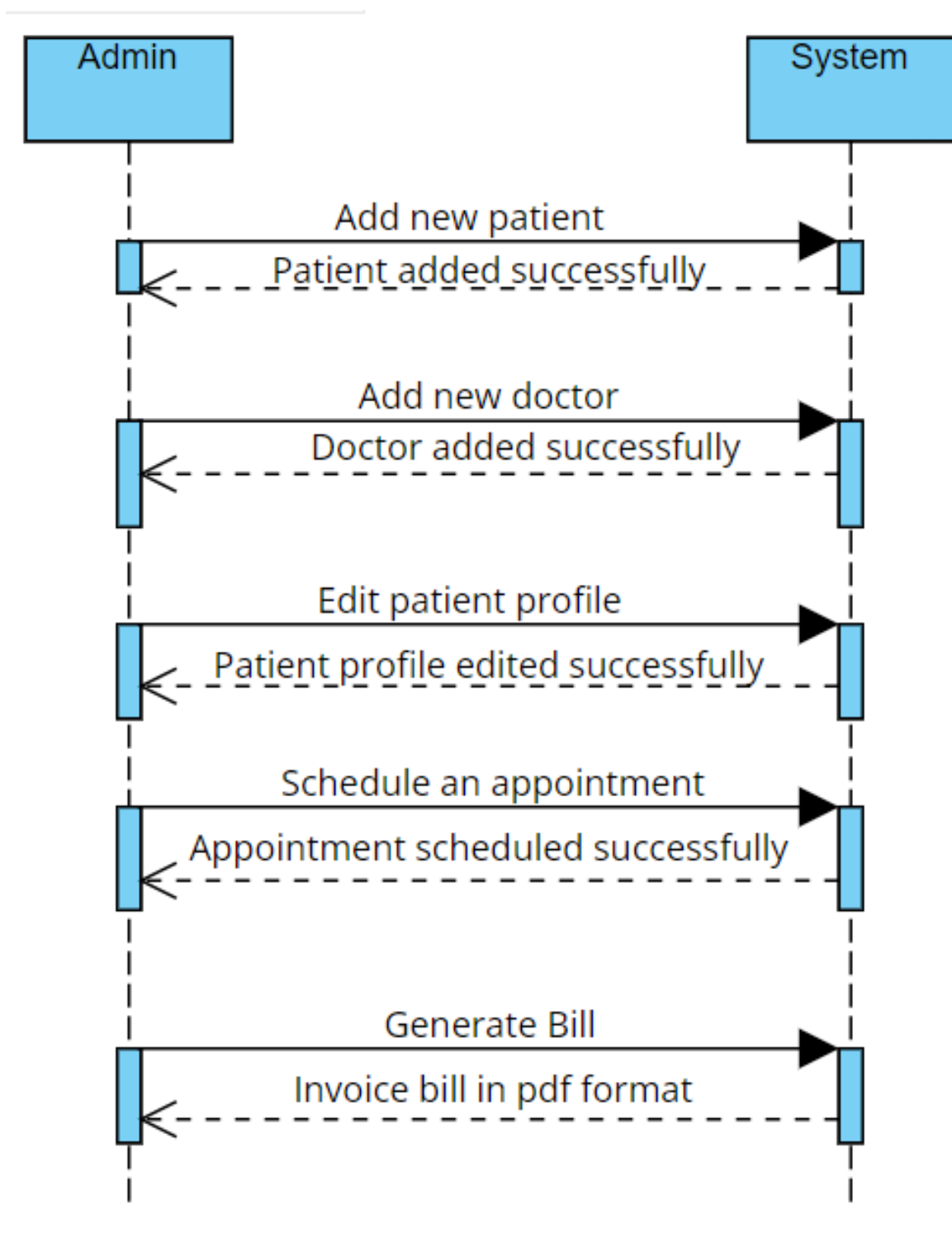


Figure 4.2 : Sequence diagram

Chapter 5

Technology Used

Programming Language (Front-End): React

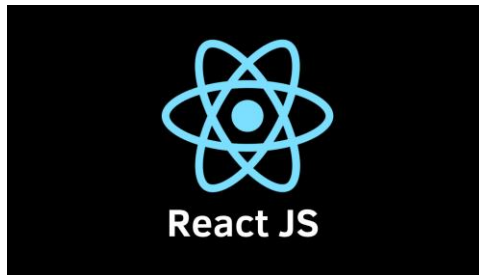


Figure 5.1 : React logo

About

React is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta and a community of individual developers and companies. React can be used as a base in the development of single-page, mobile, or server-rendered applications with frameworks like Next.js. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

Basic usage

The following is a rudimentary example of React usage in HTML with JSX and JavaScript.

```
1  import React from "react";
2
3  const Greeting = () => {
4    return (
5      <div className="hello_world">
6        <h1> Hello, world! </h1>
7      </div>
8    );
9  };
10
11 export default Greeting;
```

The Greeting function is a React component that displays the famous introductory "Hello, world".

When displayed in a web browser, the result will be a rendering of:

```
<div class="hello_world">  
  <h1>Hello, world!</h1>  
</div>
```

How does it work:

While building client-side apps, a team of Facebook developers realized that the DOM is slow (The Document Object Model (DOM) is an application programming interface (API) for HTML and XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated.). So, to make it faster, React implements a virtual DOM that is basically a DOM tree representation in JavaScript. So when it needs to read or write to the DOM, it will use the virtual representation of it. Then the virtual DOM will try to find the most efficient way to update the browser's DOM.

Unlike browser DOM elements, React elements are plain objects and are cheap to create. React DOM takes care of updating the DOM to match the React elements. The reason for this is that JavaScript is very fast and it's worth keeping a DOM tree in it to speed up its manipulation.

Although React was conceived to be used in the browser, because of its design it can also be used in the server with Node.js.

Programming Language (Back-End): Spring-Boot



Figure 5.2 : Spring-boot logo

About

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.

What is Micro Service?

Micro Service is an architecture that allows the developers to develop and deploy services independently. Each service running has its own process and this achieves the lightweight model to support business applications.

Advantages

Micro services offers the following advantages to its developers –

- Easy deployment
- Simple scalability
- Compatible with Containers
- Minimum configuration
- Lesser production time

What is Spring Boot?

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.

Advantages

Spring Boot offers the following advantages to its developers –

- Easy to understand and develop spring applications
- Increases productivity
- Reduces the development time

Spring Boot is designed with the following goals –

- To avoid complex XML configuration in Spring
- To develop a production ready Spring applications in an easier way
- To reduce the development time and run the application independently
- Offer an easier way of getting started with the application

Why Spring Boot?

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

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 classpath, 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.

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.

Restful API's



Figure 5.3 : Rest-API

About:

Representational state transfer (REST) is a software architectural style that was created to guide the design and development of the architecture for the World Wide Web. REST defines a set of constraints for how the architecture of an Internet-scale distributed hypermedia system, such as the Web, should behave. The REST architectural style emphasizes the scalability of interactions between components, uniform interfaces, independent deployment of components, and the creation of a layered architecture to facilitate caching components to reduce user-perceived latency, enforce security, and encapsulate legacy systems.

REST has been employed throughout the software industry and is a widely accepted set of guidelines for creating stateless, reliable web APIs. A web API that obeys the REST constraints is informally described as RESTful. RESTful web APIs are typically loosely based on HTTP methods to access resources via URL-encoded parameters and the use of JSON or XML to transmit data.

"Web resources" were first defined on the World Wide Web as documents or files identified by their URLs. Today, the definition is much more generic and abstract and includes every thing, entity, or action that can be identified, named, addressed, handled, or performed in any way on the Web. In a RESTful Web service, requests made to a resource's URI elicit a response with a payload formatted in HTML, XML, JSON, or some other format. For example, the response can confirm that the resource state has been changed. The response can also include hypertext links to related resources. The most common protocol for these requests and responses is HTTP. It provides operations (HTTP methods) such as GET, POST, PUT, and DELETE. By using a stateless protocol and standard operations, RESTful systems aim for fast performance, reliability, and the ability to grow by reusing components that can be managed and updated without affecting the system as a whole, even while it is running.

The goal of REST is to increase performance, scalability, simplicity, modifiability, visibility, portability, and reliability.

In HTTP there are five methods that are commonly used in a REST-based Architecture i.e., POST, GET, PUT, PATCH, and DELETE. These correspond to create, read, update, and delete (or CRUD) operations respectively. There are other methods which are less frequently used like OPTIONS and HEAD.

GET: The HTTP GET method is used to read (or retrieve) a representation of a resource. In the safe path, GET returns a representation in XML or JSON and an HTTP response code of 200 (OK). In an error case, it most often returns a 404 (NOT FOUND) or 400 (BAD REQUEST).

POST: The POST verb is most often utilized to create new resources. In particular, it's used to create subordinate resources. That is, subordinate to some other (e.g. parent) resource. On successful creation, return HTTP status 201, returning a Location header with a link to the newly-created resource with the 201 HTTP status.

NOTE: POST is neither safe nor idempotent.

PUT: It is used for updating the capabilities. However, PUT can also be used to create a resource in the case where the resource ID is chosen by the client instead of by the server. In other words, if the PUT is to a URI that contains the value of a non-existent resource ID. On successful update, return 200 (or 204 if not returning any content in the body) from a PUT. If using PUT for create, return HTTP status 201 on successful creation. PUT is not safe operation but it's idempotent.

PATCH: It is used to modify capabilities. The PATCH request only needs to contain the changes to the resource, not the complete resource. This resembles PUT, but the body contains a set of instructions describing how a resource currently residing on the server should be modified to produce a new version. This means that the PATCH body should not just be a modified part of the resource, but in some kind of patch language like JSON Patch or XML Patch. PATCH is neither safe nor idempotent.

DELETE: It is used to delete a resource identified by a URI. On successful deletion, return HTTP status 200 (OK) along with a response body.

Platform: Microsoft Azure



Figure 5.4 : Microsoft Azure logo

What is Azure?

At its core, Azure is a public cloud computing platform—with solutions including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) that can be used for services such as analytics, virtual computing, storage, networking, and much more. It can be used to replace or supplement your on-premise servers.

Facts about Azure.

Microsoft Azure – IaaS, PaaS and SaaS

- Flexible – Move compute resources up and down as needed
- Open – Supports almost any OS, language, tool, or framework
- Reliable – 99.95% availability SLA and 24x7 tech support
- Global – Data housed in geo-synchronous data centers
- Economical – Only pay for what you use

Azure is a fast, flexible, and affordable platform, and its pricing and capabilities make it the best public cloud offering on the market. Now let's take a look at how to put it to work for you.

1. Enhance and Implement Backup and Disaster Recovery

Azure is a backup and disaster recovery dream tool. Why? Because of its flexibility, advanced site recovery, and built-in integration.

As a cloud-based solution, Azure is innately flexible – it can back up your data in almost any language, on any OS, and from any location. Plus, you define the frequency and extent of your backup schedule (daily, weekly, monthly, etc.).

Tape backup has a time and place, but it has limited abilities as a stand-alone backup and disaster recovery solution. Azure site recovery can enhance your tape backup with offsite replication, minimal onsite maintenance, up to ninety-nine years of data retention, minimal or no capital investment, and minimal operational costs. Azure backup stores three copies of your data in three different locations in the data center, and then another three copies in a remote Azure data center, so you never have to worry about losing data.

2. Host and Develop Web and Mobile Apps

Whether you're looking for a platform for hosting, developing, or managing a web or mobile app, Azure makes those apps autonomous and adaptive with patch management, AutoScale, and integration for on-premise apps.

With Automatic patch management for your virtual machines, you can spend less time managing your infrastructure and focus on improving your apps. Azure also comes with continuous deployment support, which allows you to streamline ongoing code updates.

AutoScale is a feature built into Azure Web Apps that adjusts your resources automatically based on customer web traffic so you have the resources you need when traffic is high, and save money when you're not in peak times.

3. Distribute and Supplement Active Directory

Azure can integrate with your Active Directory to supplement your identity and access capabilities—this gives your DNS a global reach, centralized management, and robust security.

With Azure, you can globally distribute an Active Directory environment that is direct connect enabled. No other cloud provider has the ability to extend the reach of your domain controller and consolidate AD management like Azure.

If you have multiple locations or use on-premise apps or cloud apps like Microsoft 365, Active Directory integration with Azure will be the central tool for managing and maintaining access to all of these tools.

Azure also enables you to utilize multi-factor authentication, adding a new layer of security to your data and applications with zero hassle for your users. You can also easily implement single sign-on for Windows, Mac, Android, and iOS cloud apps.

4. Innovate with IoT Industry Solutions

The scalability, flexibility, and security of Microsoft Azure makes it the perfect resource for companies moving toward IoT solutions. You can connect your devices to the cloud using solutions that integrate with your existing infrastructure and start collecting new data about your company.

Within the Azure IoT Hub, you can monitor and manage billions of devices and gain insights to help you make better business decisions, improve customer experiences, reduce complexity, lower costs, and speed up development.

The enhanced security of Azure is a huge asset for IoT solutions, which traditionally have security gaps that hackers can take advantage of. Other benefits include remote monitoring and predictive maintenance and analytics.

Development IDE: VS Code



Figure 5.5 : VS Code logo

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

VS Code offers an array of features for efficient, conformant coding.

Source control: One of the core inbuilt features of VS Code is source control. A dedicated tab in the menu bar grants users access to the version control setting. In addition, users can view changes made on a running project. However, leveraging this source control option necessitates linking the VS code to requisite version control systems, e.g., Perforce, Apache, Git, and more. Users can create repositories, push and pull requests directly from the VS code program.

Vast programming language: The VS code supports numerous programming languages, including Python, C++, Go, JavaScript and Node.js. Aside from basic features like code folding, syntax highlighting and matching brackets. Other features may differ depending on the programming language. VS Code also offers IntelliSense for TypeScript, CSS, HTML, etc., and debugging for Node.js. Users can leverage the freely available extension for additional language support, themes, and debuggers. In addition, using the language server protocol, users can add code linter and perform static code analysis.

IntelliSense Code completion: Virtual Studio Code's IntelliSense code completion can give programmers hints on the best possible function parameter, syntax, or variable. In addition, it can sense coding errors and offer suggestions to correct them.

Other features include Snippets, code refactoring, and embedded Git. Users can customize the user experience by installing extensions, using keyboard shortcuts, changing themes, and more.

Operating System: Windows/Linux/MacOS



Figure 5.6 : Windows, MacOS and Linux logo

- This project works on Windows OS.
- This project works on Mac OS.
- This project works on Linux.
- This project works on any OS that supports browser.

Fundamental Subjects/Concepts Used in the Project



Figure 5.7 : HTML , CSS , JS

Web Development – For making the front end using React Framework.



Figure 5.8 : Java and spring logo

Java Programming – For making the backend using Spring Boot Framework.



Figure 5.9 : H2 database

Database Management System - For storing the data in Spring Boot H2 database.

Chapter 6

Implementation:

Sample Source code:

```
1  import React from 'react';
2  import axios from 'axios';
3
4  function WelcomeTitleDiv() {
5    axios({
6      method: 'get',
7      url: 'https://kls-college-project-hospital.herokuapp.com/StartServer',
8      data: {
9        }
10   })
11   .then((response) => {
12     }, (error) => {
13     });
14   return <div className="WelcomeTitleDiv">
15     <div className='WelcomeTitleDivContent'>KLS HOSPITAL MANAGEMENT SYSTEM</div>
16   </div>;
17 }
18
19 export default WelcomeTitleDiv;
```

```

1  import React from 'react'
2  import CheckBoxIcon from '@mui/icons-material/CheckBox';
3
4  function DoctorAddedSuccessDiv() {
5    return (
6      <div className='PatientAddedSuccessDiv'>
7        <div className='PatientAddedSuccess'>
8          <div className='PatientAddedSuccessBottom'>Doctor details added successfully.</div>
9          <div className='PatientAddedSuccessTop'><CheckBoxIcon className="checkBoxIcon"/></div>
10         </div>
11       </div>
12     )
13   }
14
15   export default DoctorAddedSuccessDiv

```

```

1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import App from './App';
4
5
6  window.name = "nnnn";
7
8  ReactDOM.render(
9    <React.StrictMode>
10     <App />
11   </React.StrictMode>,
12   document.getElementById('root')
13 );

```

```

1  import axios from 'axios';
2  import { Button, TextField ,Input} from '@mui/material';
3  import { useState } from 'react';
4
5
6
7  import * as React from 'react';
8  import Paper from '@mui/material/Paper';
9  import Table from '@mui/material/Table';
10 import TableBody from '@mui/material/TableBody';
11 import TableCell from '@mui/material/TableCell';
12 import TableContainer from '@mui/material/TableContainer';
13 import TableHead from '@mui/material/TableHead';
14 import TablePagination from '@mui/material/TablePagination';
15 import TableRow from '@mui/material/TableRow';
16
17 function ThirdSideBarElementDiv() {
18
19     return <div className='ViewDoctorDiv'>
20         <div className='ViewDoctorInnerDiv'>
21             <div className='ViewDoctorInnerDivTitle'>
22                 GENERATE BILLS
23             </div>
24             <div className='ViewDoctorInnerDivTop'>
25
26             </div>
27             <div className='DoctorInnerDivBottom'>
28
29             </div>
30         </div>
31
32     </div>;
33 }
34
35 export default ThirdSideBarElementDiv;

```

Chapter 7

Results and discussions:

Output (Screenshot)



Figure 7.1: Welcome logo screen



Figure 7.2: Welcome screen

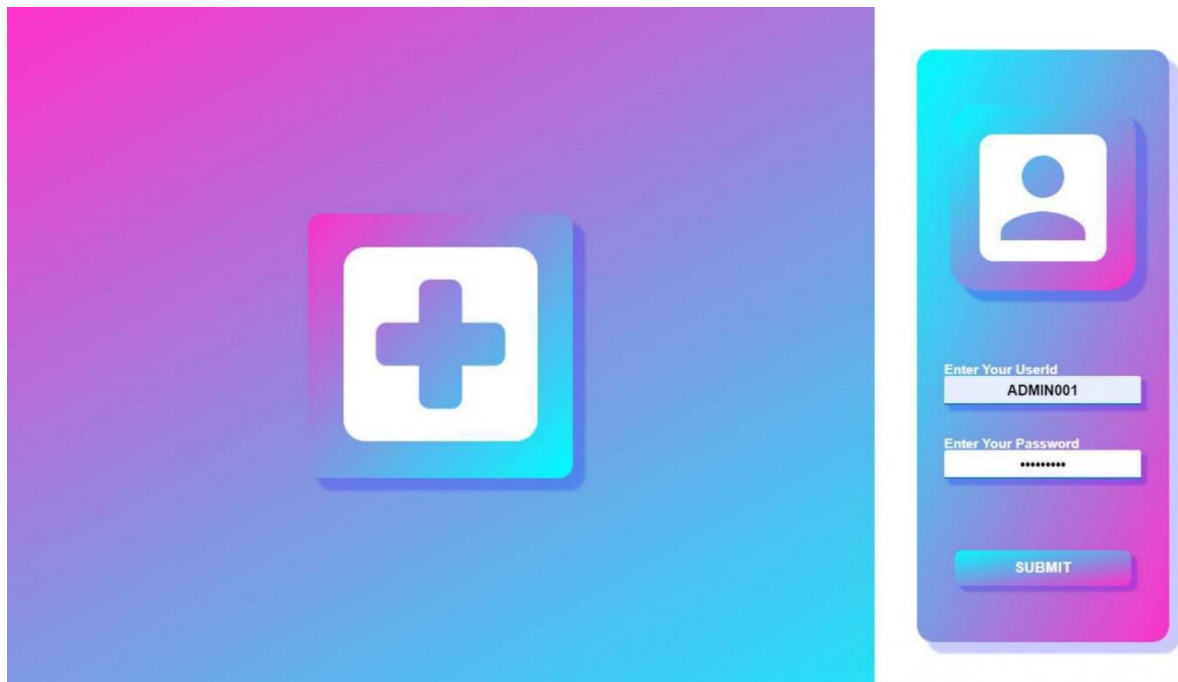


Figure 7.3 : Login screen

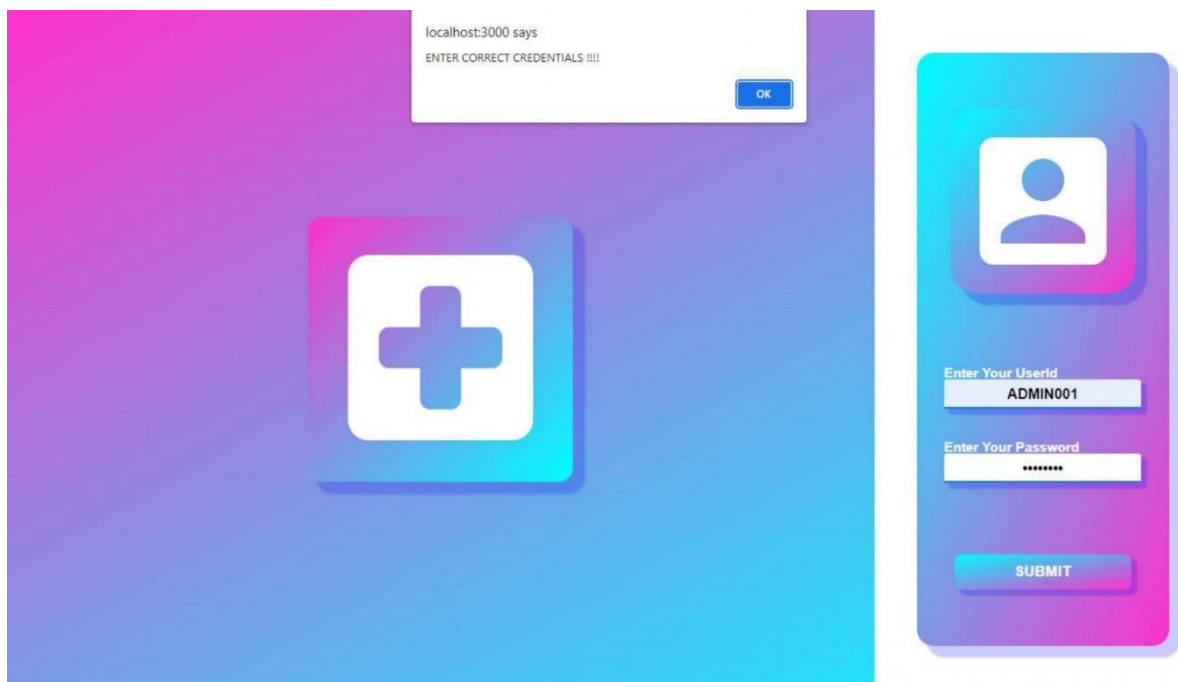


Figure 7.4 : Login screen when wrong password entered



Figure 7.5 : Dashboard screen

The "Add new patient profile" screen features a header with a plus icon, the title "KLS HOSPITAL MANAGEMENT SYSTEM", and a user profile for "Vishwanath Admin". A left sidebar contains navigation links: Home, Add Doctor, Schedule Appt., Add Patient, Edit Patient, and Generate Bill. The main content area displays a form titled "ENTER PATIENT DETAILS" with fields for Id (KLS_PT003), Email (SUMEET@GMAIL.COM), Name (SUMEET), Address (Tilakwadi Belagavi), Gender (MALE), Age (22), and Contact (8765456787). A profile picture placeholder shows a man with glasses. A "Choose File" button is next to the profile picture, and a "SUBMIT" button is at the bottom.

Figure 7.6 : Add new patient profile screen



The screenshot displays the 'KLS HOSPITAL MANAGEMENT SYSTEM' interface. The top header includes a home icon, the system name, and the user 'Vishwanath Admin'. A left sidebar lists navigation options: Home, Add Doctor, Schedule Appt., Add Patient, Edit Patient, and Generate Bill. The main content area features a large green checkmark and the text 'Patient details added successfully.' To the right, a 'Corona Covid-19 Tracker' widget shows 'LIVE' status, the date '28-04-2022', and case counts: World (408805), India (5399), and Karnataka (85).

Figure 7.7 : Patient added successfully screen



The screenshot displays the 'KLS HOSPITAL MANAGEMENT SYSTEM' interface with the 'Add Doctor' option selected in the sidebar. The main content area is titled 'ENTER DOCTOR DETAILS' and contains a form with the following fields: Id (KLS_PT005), Contact (9878766765), Name (Dr. Raman Pawar), Email (Ramanpawar@gmail.com), Specialization (Heart Specialist), Qualification (Phd), and Experience (20). A 'SUBMIT' button is at the bottom. A preview image of a doctor is shown with the text 'Choose File doctor-png-1....-preview.png'. To the right, the 'Corona Covid-19 Tracker' widget shows 'LIVE' status, the date '10-07-2022', and case counts: World (760934), India (18840), and Karnataka (1037).

Figure 7.8 : Add new doctor profile screen

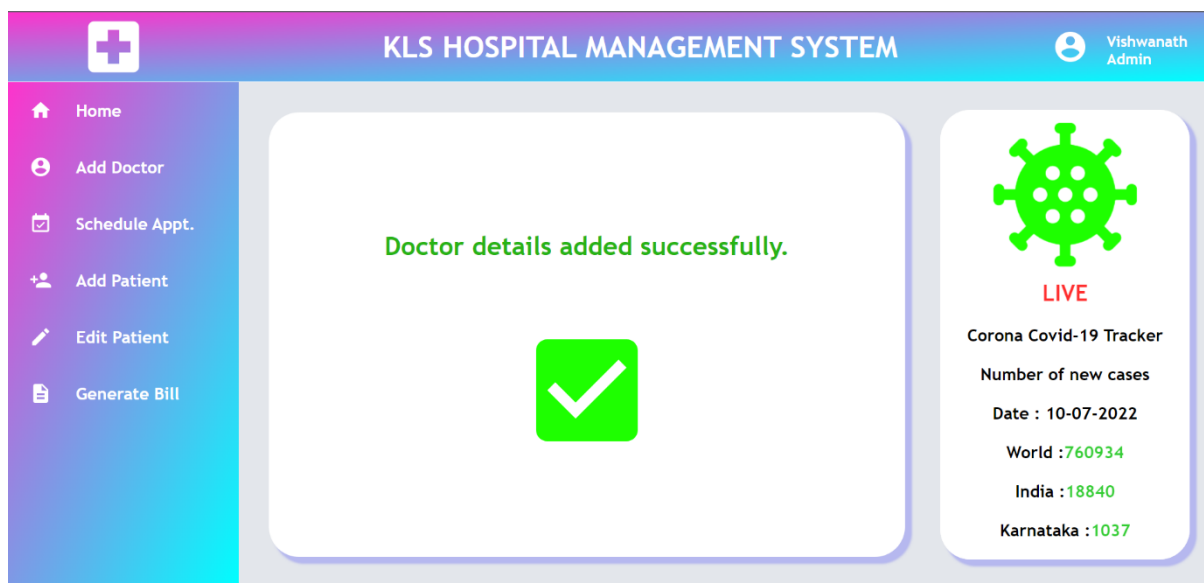


Figure 7.9 : Doctor added successfully screen



Figure 7.10 : Schedule an appointment screen

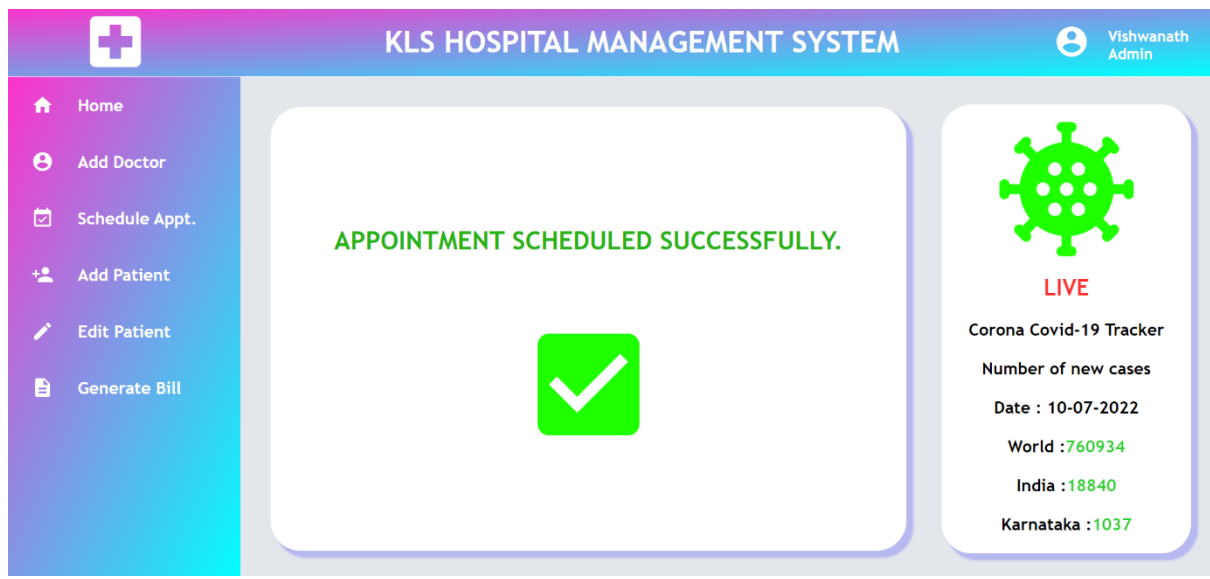


Figure 7.11 : Appointment scheduled successfully screen

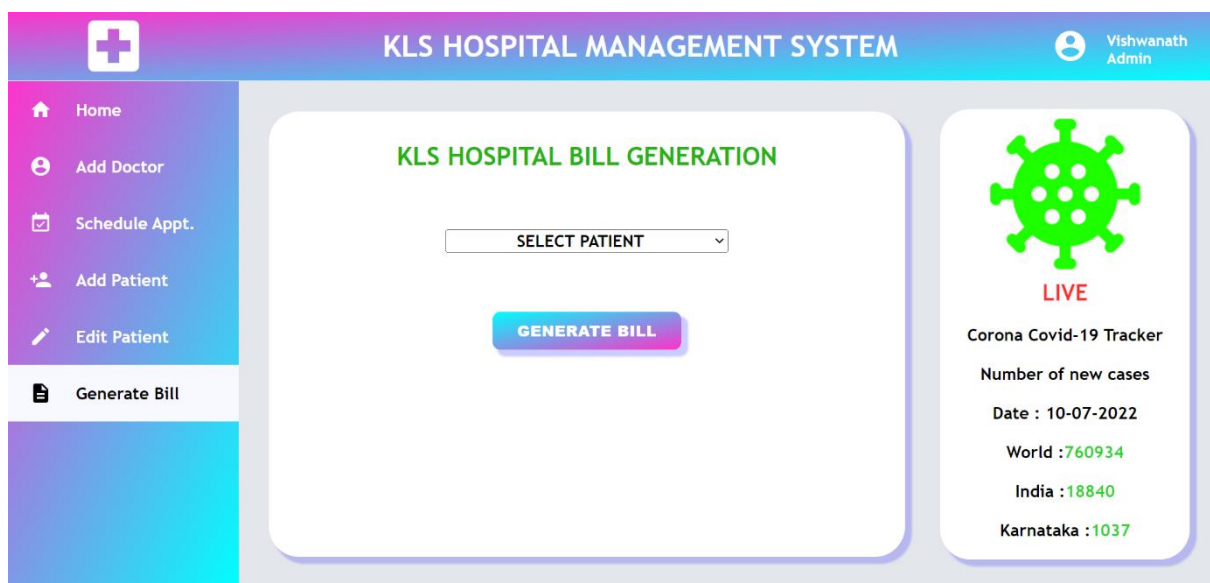


Figure 7.12 : Generate bill screen

KLS HOSPITAL MANAGEMENT SYSTEM

INVOICE

Patient Name : vishwanath samaji

Contact No. : 9019199891

Email ID : vishwanath987samaji@gmail.com

Address : 18 / 2 new sainik nagar laxmi tek camp belgaum

No. of Schedule Appointment	Cost Per Appointment
2	500
Total	1000

Figure 7.13 : Invoice in PDF format screen

Live Project Link :

Link : <https://agreeable-island-07c260410.1.azurestaticapps.net/>

Conclusion and Future Scope:

Conclusion:

In this project, we have successfully developed a hospital management system using react, spring boot and Microsoft azure platform. This hospital management system will help hospitals to manage hospitals in a proper way.

Future Scope:

Adding IOT Device for calculating Heart beats and predicting the health of a patient using the number of heart beats.

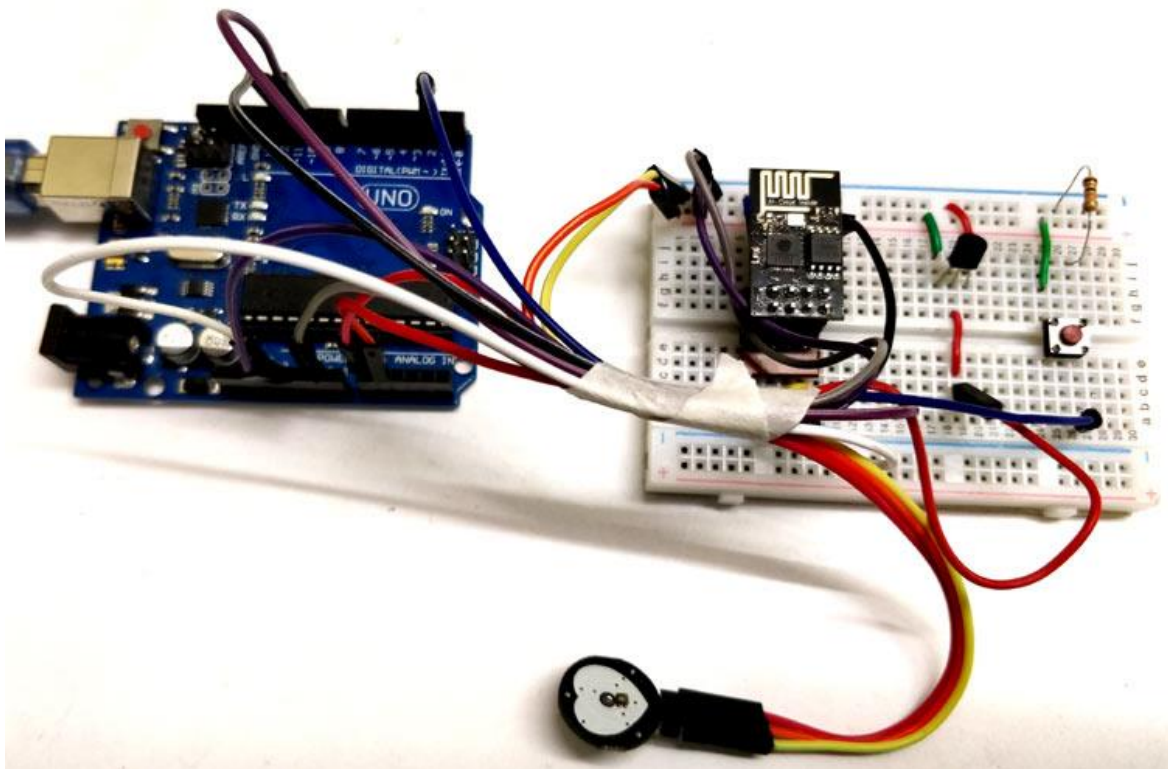


Figure 7.14 : Future scope

References:

- ❖ <https://futurereadytalent.in/>
- ❖ <https://spring.io/projects/spring-boot>
- ❖ <https://reactjs.org/docs/getting-started.html>