**PROJECT REPORT FILE.**

**ONLINE MEDICAL MANAGEMENT SYSTEM.**

**NAME:ABHAY GUPTA**

**UNIVERSITY ID:1911985068**

**DEPARTMENT:CSE**

**BATCH:2019 VIRTUSA(INTEGRATED B.E-M.E)**

**PROJECT TITLE:ONLINE MEDICAL MANAGEMENT SYSTEM.**

**SUBMITTED TO:MR.MANANJAY DUBEY AND MR.MUKESH.**

**SUBMISSION DATE:05-MAY-2022**

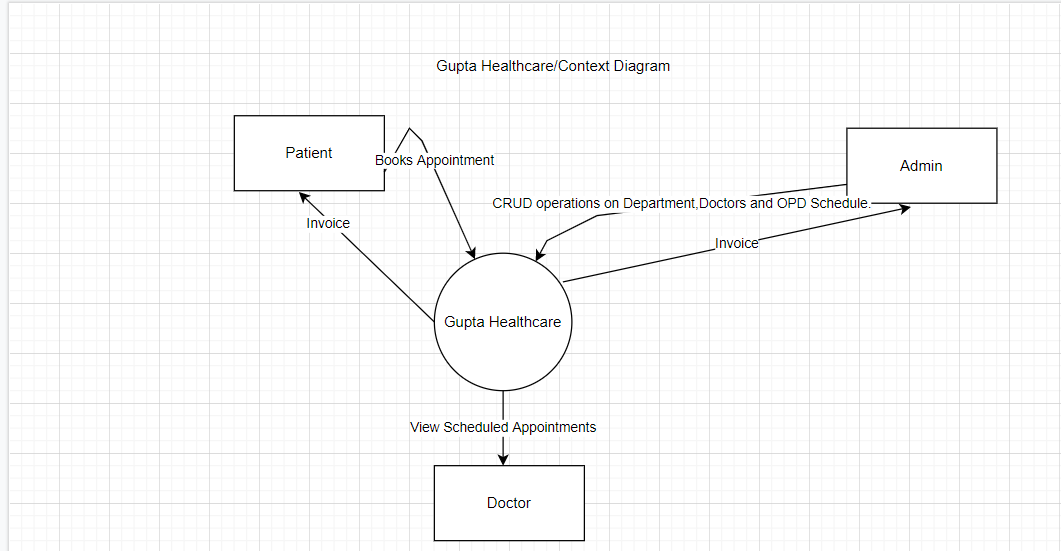
**GitHub Link Frontend:** **https://github.com/AbhayGupta-cloud/MedicalManagementFrontend**

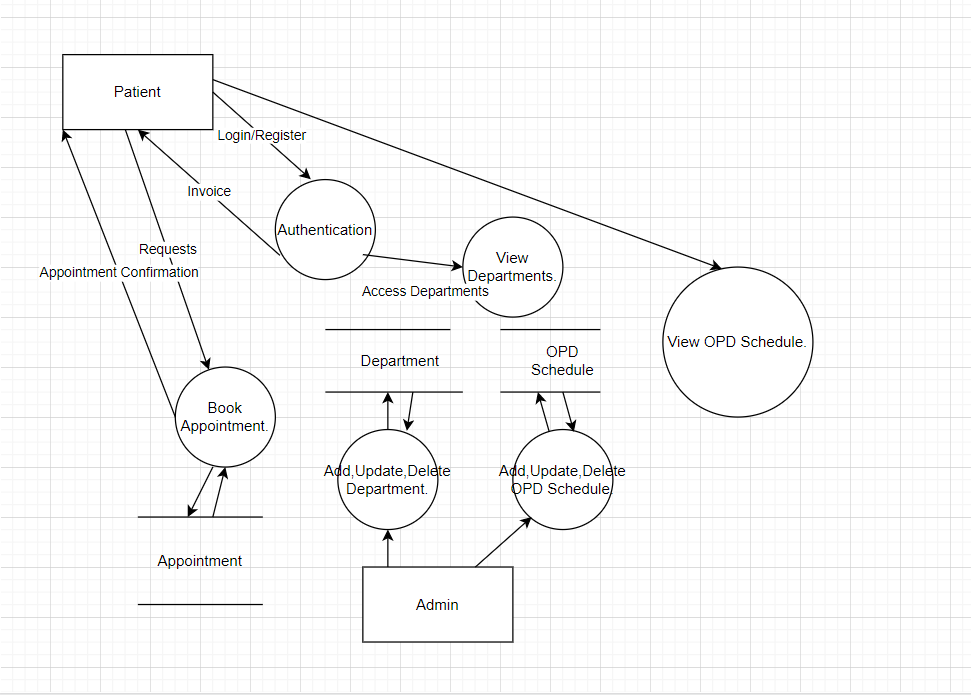
**GitHub Link Backend:** **https://github.com/AbhayGupta-cloud/MedicalManagementBackend**

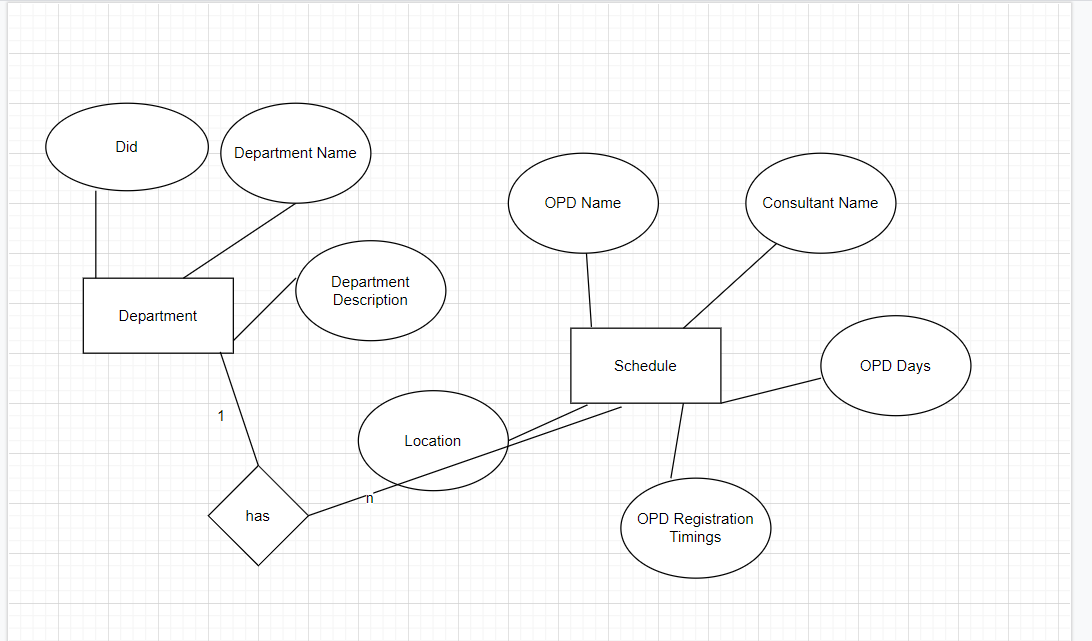
Purpose:

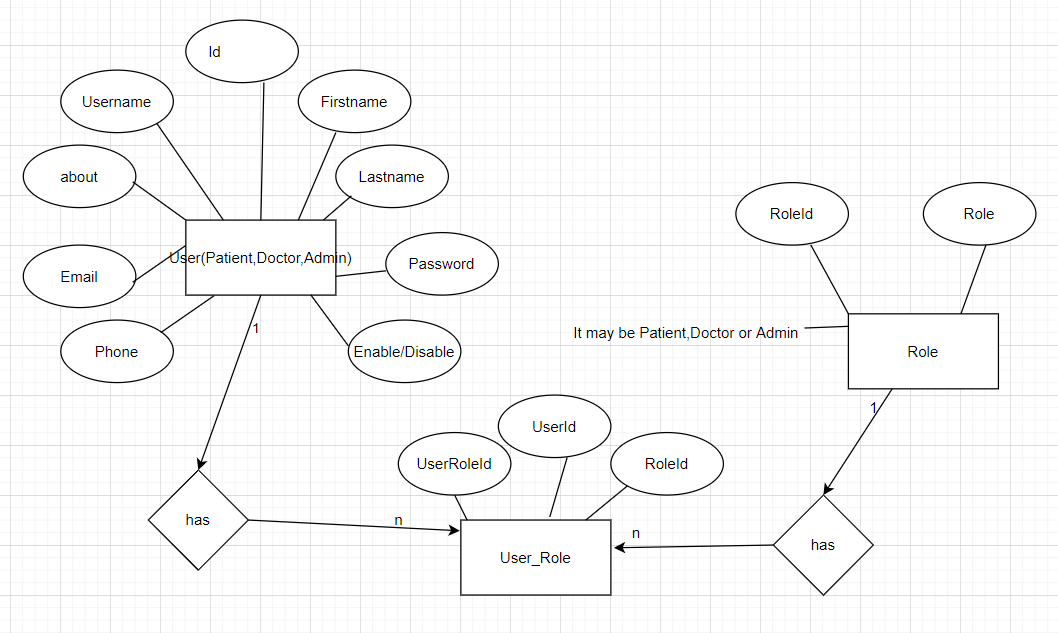
It’s purpose is to provide transparent information to the patients regarding departments in the hospital,different OPD Schedule with OPD Name,Consultant Name,OPD Registration Timings,OPD Name etc.So,that there is no miscommunication and confusion to any patient regarding any department or OPD and patient and consultant can focus directly on treatment of the disease by saving their precious time.

Data Flow Diagrams:





ER Diagrams: 

Future Scope:

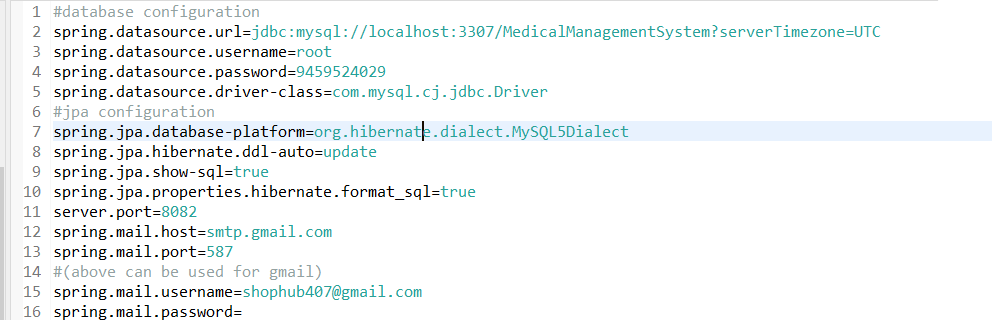
As the convergence of IT and medical services has increasingly enabled patients to access medical services in the course of their daily lives without time or space limitations, use of biosensors and related forms of medical technology has become a preferred means of delivering healthcare services.

The rapid increase in use of cellular phones in recent years has encouraged the development of significantly advanced health applications. In what has become known as cellular phone-based medical informatics (CPBMI) for healthcare, cellular phone-based personal healthcare tools aimed at prevention, diagnosis, and treatment of diseases are used to enable communication between healthcare providers and patients. Despite their proliferation, few studies have examined the user-friendliness or effectiveness of CPBMI tools or their commercialization potential. It is particularly essential to investigate patient and provider experiences with CPBMI to assist in the effective linking of mobile health and health management services. Before applying a newly developed system to medical practice, evidence-based examination of the practicality and clinical effectiveness of the system and cost–benefit analysis of its potential for commercialization should be conducted. To contribute to the fulfillment of these research needs, this study systematically reviewed the results of previous studies that had investigated the use of CPBMI to identify its current status within the medical field, advantages and disadvantages, practicality, clinical effectiveness, costs, and cost-saving potential.

TOPIC Discriptions(Online Medical Management System).

Today's web based technology offers many online services in almost every field. Almost everything can be done online reducing the amount of tasks, cost, and efforts to a greater extent using cloud computing online storage also can be managed. The paper describes about an idea of such a web based platform that make many medical/hospital procedures online using Web, networking, Cloud and android programming technology that can be very important in implementing the functionality of online medical management. This will help in management of patients, managing the schedules of the doctors, maintaining the records of patients which can be accessed throughout hospital. Storing, managing, communicating, analysing and updating the patient details online. Thus by implementing this web based application using Spring Boot,Angular and MYSQL and creating customized application using android application programming we can manage many tasks that are usually time consuming and inconvenient.

**Application.properties configuration file.**



Technology Used With Details :

Spring Boot: Spring Boot **helps developers create applications that just run**. Specifically, it lets you create standalone applications that run on their own, without relying on an external web server, by embedding a web server such as Tomcat or Netty into your app during the initialization process. 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.

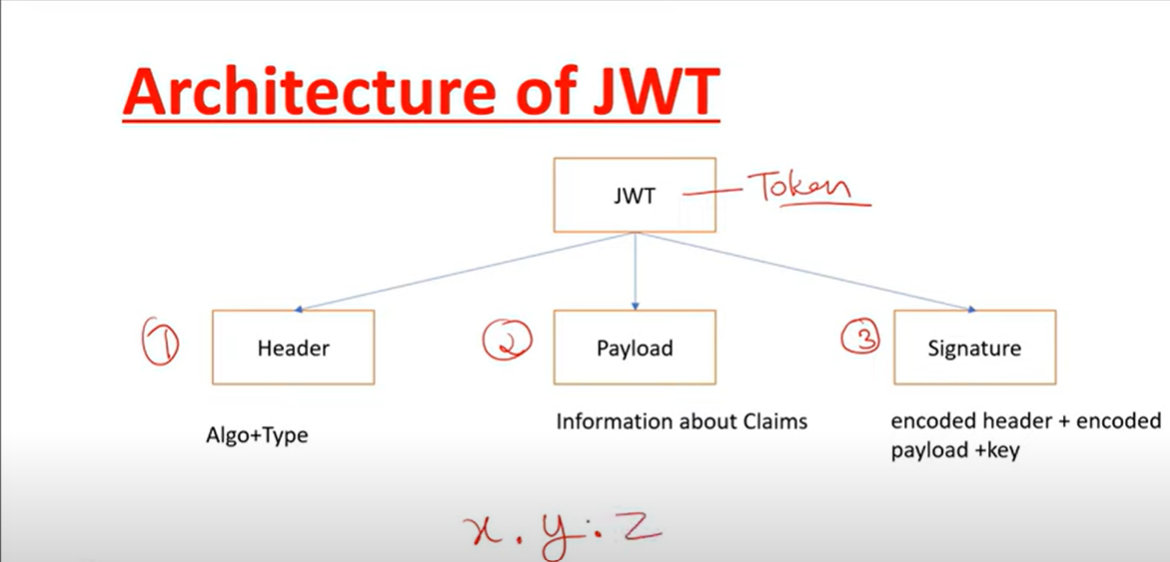
Spring Boot automatically scans all the components included in the project by using **@ComponentScan** annotation. Handling dependency management is a difficult task for big projects. Spring Boot resolves this problem by providing a set of dependencies for developers convenience.

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.

JWT Token: **JSON Web Token or JWT**, as it is more commonly called, is an open Internet standard (RFC 7519) for securely transmitting trusted information between parties in a compact way. The tokens contain claims that are encoded as a JSON object and are digitally signed using a private secret or a public key/private key pair. They are self-contained and verifiable as they are digitally signed. JWT’s can be signed and/or encrypted. The signed tokens verify the integrity of the claims contained in the token, while the encrypted ones hide the claims from other parties.

JWT’s can also be used for the exchange of information though they more commonly used for authorization as they offer a lot of advantages over session management using in-memory random tokens. The biggest of them being the enabling the delegation of authentication logic to a third-party server like **AuthO** etc.



A JWT token is divided into 3 parts namely – header, payload, and signature in the format of

[Header].[Payload].[Signature]

* **Header** − The Header of a JWT token contains the list cryptographic operations that are applied to the JWT. This can be the signing technique, metadata information about the content-type and so on. The header is presented as a JSON object which is encoded to a base64URL. An example of a valid JWT header would be

{ "alg": "HS256", "typ": "JWT" }

Here, “**alg**” gives us information about the type of algorithm used and “typ gives us the type of the information.

* **Payload** − The payload part of JWT contains the actual data to be transferred using the token. This part is also known as the “claims” part of the JWT token. The claims can be of three types – registered, public and private.
* The registered claims are the ones which are recommended but not mandatory claims such as iss(issuer), sub(subject), aud(audience) and others.
* Public claims are those that are defined by those using the JWTs.
* Private claims or custom claims are user-defined claims created for the purpose of sharing the information between the concerned parties.

Example of a payload object could be.

{ "sub": "12345", "name": "Johnny Hill", "admin": false }

The payload object, like the header object is base64Url encoded as well and this string forms the second part of the JWT.

* **Signature**− The signature part of the JWT is used for the verification that the message wasn’t changed along the way. If the tokens are signed with private key, it also verifies that the sender is who it says it is. It is created using the encoded header, encoded payload, a secret and the algorithm specified in the header. An example of a signature would be.

HMACSHA256( base64UrlEncode(header) + "." + base64UrlEncode(payload), secret)

If we put the header, payload and signature we get a token as given below.

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6I

kpvaG4gRG9lIiwiYWRtaW4iOmZhbHNlfQ.gWDlJdpCTIHVYKkJSfAVNUn0ZkAjMxskDDm-5Fhe

WJ7xXgW8k5CllcGk4C9qPrfa1GdqfBrbX\_1x1E39JY8BYLobAfAg1fs\_Ky8Z7U1oCl6HL63yJq\_

wVNBHp49hWzg3-ERxkqiuTv0tIuDOasIdZ5FtBdtIP5LM9Oc1tsuMXQXCGR8GqGf1Hl2qv8MCyn

NZJuVdJKO\_L3WGBJouaTpK1u2SEleVFGI2HFvrX\_jS2ySzDxoO9KjbydK0LNv\_zOI7kWv-gAmA

j-v0mHdJrLbxD7LcZJEGRScCSyITzo6Z59\_jG\_97oNLFgBKJbh12nvvPibHpUYWmZuHkoGvuy5RLUA

Now, this token can be used in the Authorization header using the Bearer schema as.

**Authorization − Bearer** <token>

The use of JWT token for authorization is the most common of its applications. The token is usually generated in the server and sent to the client where it is stored in the session storage or local storage. To access a protected resource the client would send the JWT in the header as given above. We will see the JWT implementation in Spring Security in the section below.

JPA(JAVA Persistence API): Spring Boot JPA is a Java specification for managing relational data in Java applications. It **allows us to access and persist data between Java object/ class and relational database**. JPA follows Object-Relation Mapping (ORM). It is a set of interfaces. The Java Persistence API (JPA) is a specification of Java. It is used to persist data between Java object and relational database. JPA acts as a bridge between object-oriented domain models and relational database systems.

As JPA is just a specification, it doesn't perform any operation by itself. It requires an implementation. So, ORM tools like Hibernate, TopLink and iBatis implements JPA specifications for data persistence.

The **javax.persistence** package contains the JPA classes and interfaces.

The first version of Java Persistenece API, JPA 1.0 was released in 2006 as a part of EJB 3.0 specification.

Following are the other development versions released under JPA specification: -

**00:00/24:36**

* JPA 2.0 - This version was released in the last of 2009. Following are the important features of this version: -
  + It supports validation.
  + It expands the functionality of object-relational mapping.
  + It shares the object of cache support.
* JPA 2.1 - The JPA 2.1 was released in 2013 with the following features: -
  + It allows fetching of objects.
  + It provides support for criteria update/delete.
  + It generates schema.
* JPA 2.2 - The JPA 2.2 was released as a development of maintainenece in 2017. Some of its important feature are: -
  + It supports Java 8 Date and Time.
  + It provides @Repeatable annotation that can be used when we want to apply the same annotations to a declaration or type use.
  + It allows JPA annotation to be used in meta-annotations.
  + It provides an ability to stream a query result.

Hibernate: Hibernate **reduces lines of code by maintaining object-table mapping itself and returns result to application in form of Java objects**. It relieves programmer from manual handling of persistent data, hence reducing the development time and maintenance cost. Hibernate is a Java framework that simplifies the development of Java application to interact with the database. It is an open source, lightweight, ORM (Object Relational Mapping) tool. Hibernate implements the specifications of JPA (Java Persistence API) for data persistence.

ORM Tool

An ORM tool simplifies the data creation, data manipulation and data access. It is a programming technique that maps the object to the data stored in the database.



The ORM tool internally uses the JDBC API to interact with the database.

46.8M

1K

Features of Java - Javatpoint

Advantages of Hibernate Framework

Following are the advantages of hibernate framework:

1) Open Source and Lightweight

Hibernate framework is open source under the LGPL license and lightweight.

2) Fast Performance

The performance of hibernate framework is fast because cache is internally used in hibernate framework. There are two types of cache in hibernate framework first level cache and second level cache. First level cache is enabled by default.

3) Database Independent Query

HQL (Hibernate Query Language) is the object-oriented version of SQL. It generates the database independent queries. So you don't need to write database specific queries. Before Hibernate, if database is changed for the project, we need to change the SQL query as well that leads to the maintenance problem.

4) Automatic Table Creation

Hibernate framework provides the facility to create the tables of the database automatically. So there is no need to create tables in the database manually.

5) Simplifies Complex Join

Fetching data from multiple tables is easy in hibernate framework.

6) Provides Query Statistics and Database Status

Hibernate supports Query cache and provide statistics about query and database status.

Spring Security: Spring Security is a framework that **enables a programmer to impose security restrictions to Spring-framework–based Web applications through JEE components**. In short, it is a library that can be used, extended to customize as per the programmer's needs. Spring Security is a framework which provides various security features like: authentication, authorization to create secure Java Enterprise Applications.

It is a sub-project of Spring framework which was started in 2003 by Ben Alex. Later on, in 2004, It was released under the Apache License as Spring Security 2.0.0.

It overcomes all the problems that come during creating non spring security applications and manage new server environment for the application.

This framework targets two major areas of application are authentication and authorization. Authentication is the process of knowing and identifying the user that wants to access.

## Advantages of using Spring Security.

Spring Security has numerous advantages. Some of that are given below.

* Comprehensive support for authentication and authorization.
* Protection against common tasks
* Servlet API integration
* Integration with Spring MVC
* Portability
* CSRF protection
* Java Configuration support

MYSQL: MySQL is a relational database management system based on SQL – Structured Query Language. The application is used for a wide range of purposes, including **data warehousing, e-commerce, and logging applications**. The most common use for mySQL however, is for the purpose of a web database.

It is very important to understand the database before learning MySQL. A database is an application that stores the organized collection of records. It can be accessed and manage by the user very easily. It allows us to organize data into tables, rows, columns, and indexes to find the relevant information very quickly. Each database contains distinct [API](https://www.javatpoint.com/api-full-form) for performing database operations such as creating, managing, accessing, and searching the data it stores. Today, many databases available like MySQL, Sybase, [Oracle](https://www.javatpoint.com/what-is-oracle), [MongoDB](https://www.javatpoint.com/mongodb-tutorial), [PostgreSQL](https://www.javatpoint.com/postgresql-tutorial), [SQL Server](https://www.javatpoint.com/sql-server-tutorial), etc. In this section, we are going to focus on MySQL mainly. MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with [PHP](https://www.javatpoint.com/php-tutorial) scripts for creating powerful and dynamic server-side or web-based enterprise applications.

It is developed, marketed, and supported by **MySQL AB, a Swedish company**, and written in [C programming language](https://www.javatpoint.com/c-programming-language-tutorial) and [C++ programming language](https://www.javatpoint.com/cpp-tutorial). The official pronunciation of MySQL is not the My Sequel; it is ***My Ess Que Ell****. However, you can pronounce it in your way.* Many small and big companies use MySQL. MySQL supports many Operating Systems like [Windows](https://www.javatpoint.com/windows), [Linux](https://www.javatpoint.com/linux-tutorial), MacOS, etc. with C, C++, and [Java languages](https://www.javatpoint.com/java-tutorial).

MySQL is a [Relational Database Management System](https://www.javatpoint.com/what-is-rdbms) (RDBMS) software that provides many things, which are as follows:

* It allows us to implement database operations on tables, rows, columns, and indexes.
* It defines the database relationship in the form of tables (collection of rows and columns), also known as relations.
* It provides the Referential Integrity between rows or columns of various tables.
* It allows us to updates the table indexes automatically.
* It uses many SQL queries and combines useful information from multiple tables for the end-users.

How MySQL Works?

MySQL follows the working of Client-Server Architecture. This model is designed for the end-users called clients to access the resources from a central computer known as a server using network services. Here, the clients make requests through a graphical user interface (GUI), and the server will give the desired output as soon as the instructions are matched. The process of MySQL environment is the same as the client-server model.



The core of the MySQL database is the MySQL Server. This server is available as a separate program and responsible for handling all the database instructions, statements, or commands. The working of MySQL database with MySQL Server are as follows:

1. MySQL creates a database that allows you to build many tables to store and manipulate data and defining the relationship between each table.
2. Clients make requests through the GUI screen or command prompt by using specific SQL expressions on MySQL.
3. Finally, the server application will respond with the requested expressions and produce the desired result on the client-side.

A client can use any MySQL [GUI](https://www.javatpoint.com/gui-full-form). But, it is making sure that your GUI should be lighter and user-friendly to make your data management activities faster and easier. Some of the most widely used MySQL GUIs are MySQL Workbench, SequelPro, DBVisualizer, and the Navicat DB Admin Tool. Some GUIs are commercial, while some are free with limited functionality, and some are only compatible with MacOS. Thus, you can choose the GUI according to your needs.

Reasons for popularity

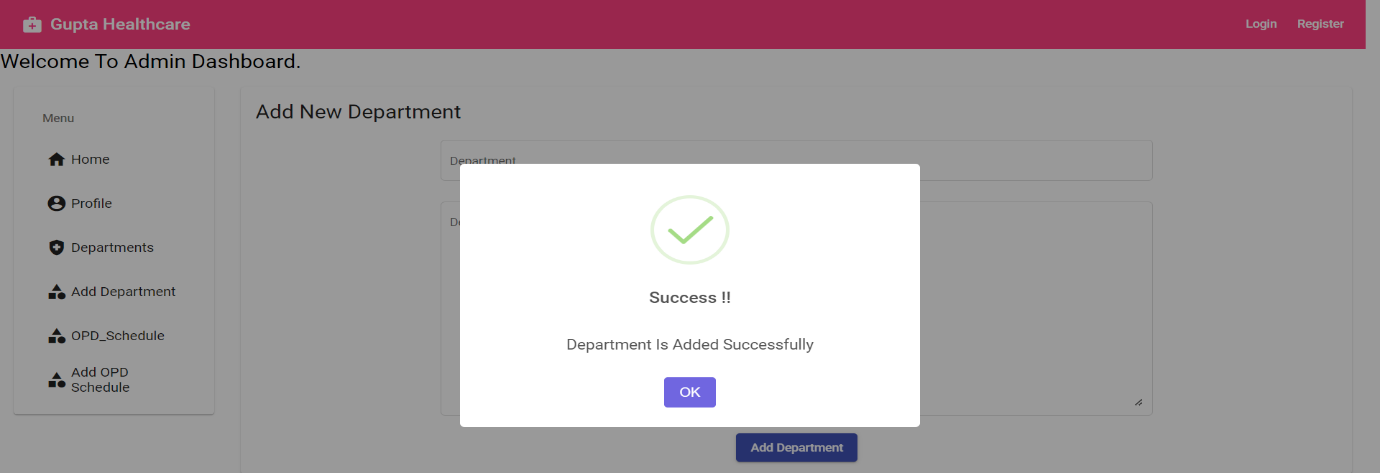
MySQL is becoming so popular because of these following reasons:

* MySQL is an open-source database, so you don't have to pay a single penny to use it.
* MySQL is a very powerful program that can handle a large set of functionality of the most expensive and powerful database packages.
* MySQL is customizable because it is an open-source database, and the open-source GPL license facilitates programmers to modify the SQL software according to their own specific environment.
* MySQL is quicker than other databases, so it can work well even with the large data set.
* MySQL supports many operating systems with many languages like PHP, PERL, C, C++, JAVA, etc.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL is very friendly with PHP, the most popular language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

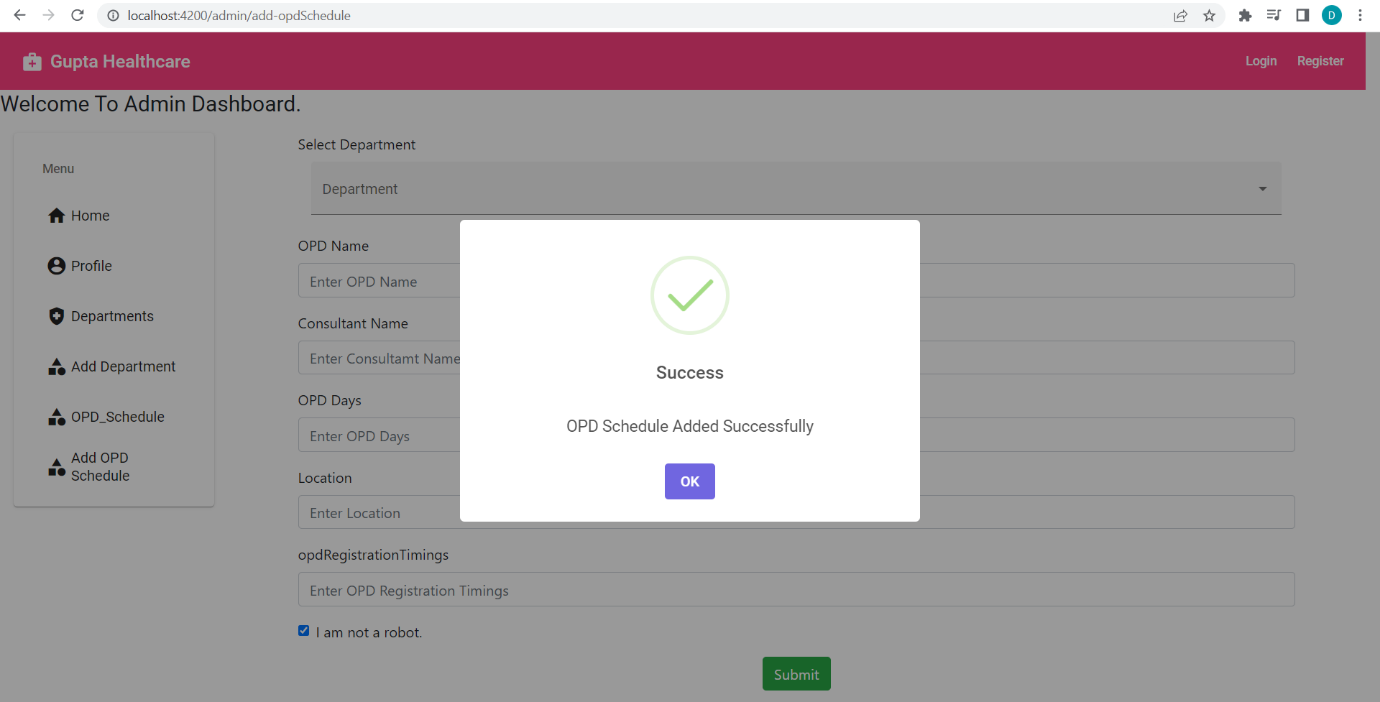
Angular: Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your applications.

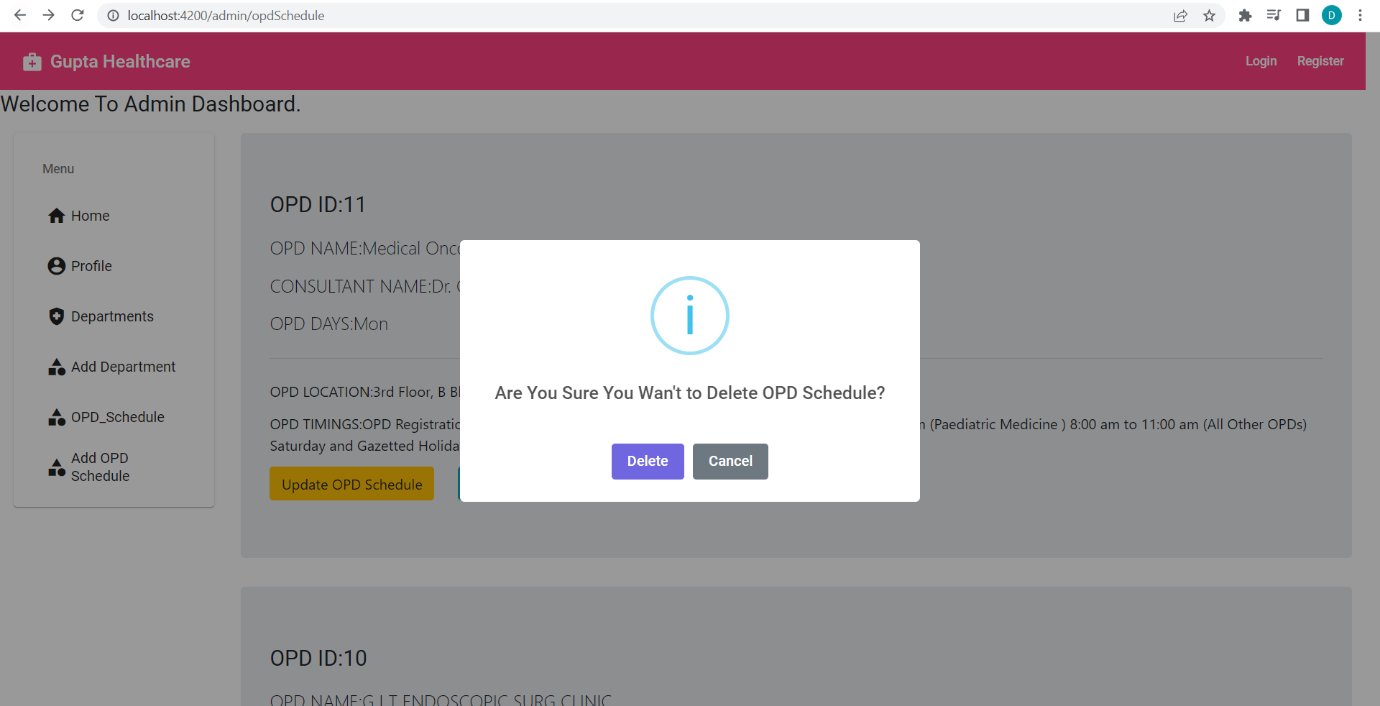
The architecture of an Angular application relies on certain fundamental concepts. The basic building blocks of the Angular framework are Angular components that are organized into *NgModules*. NgModules collect related code into functional sets; an Angular application is defined by a set of NgModules. An application always has at least a *root module* that enables bootstrapping, and typically has many more *feature modules*.

Screenshots Of My Project:

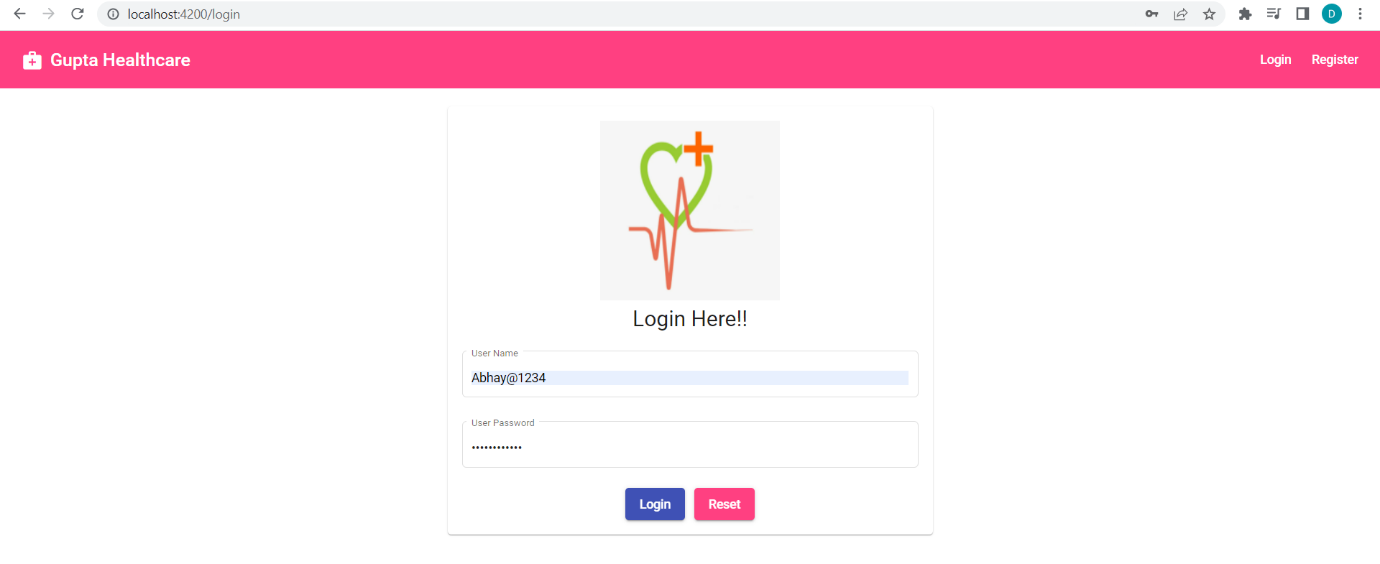
Add New Department(Admin Module):

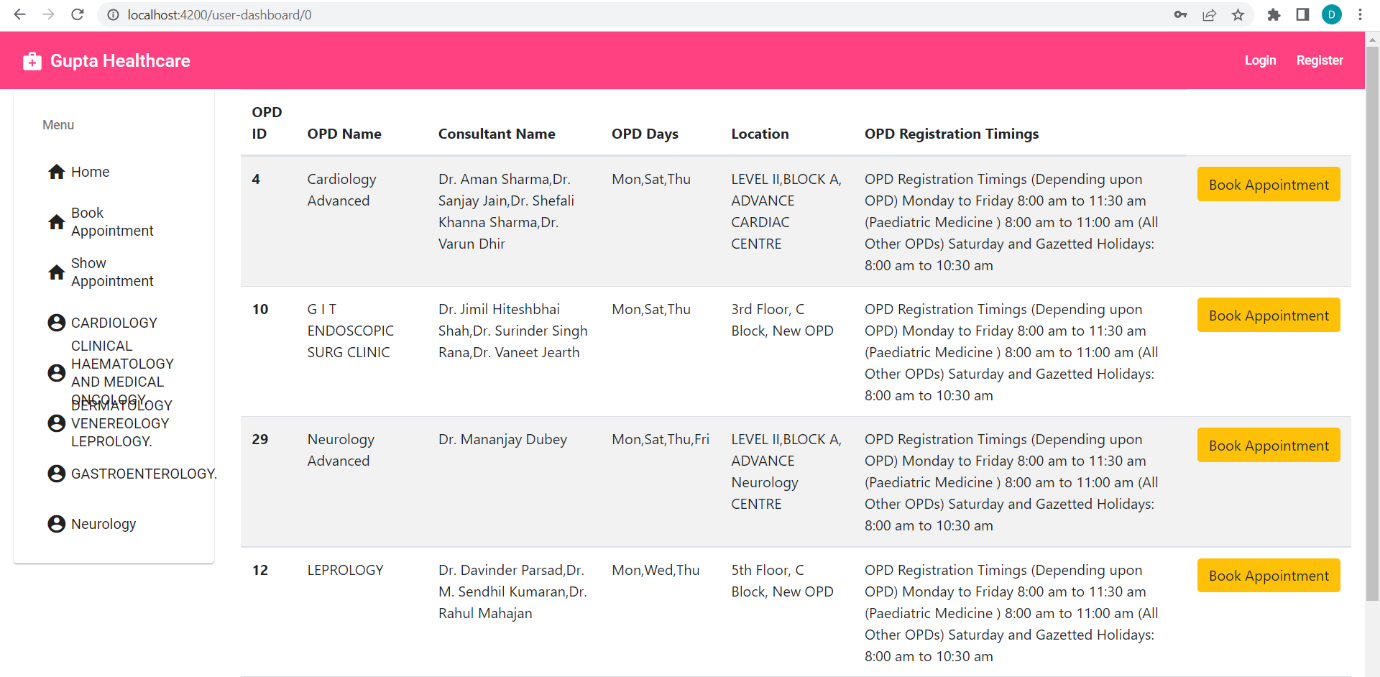
Add OPD Schedule(Admin Module):



Delete OPD Schedule(Admin Module):

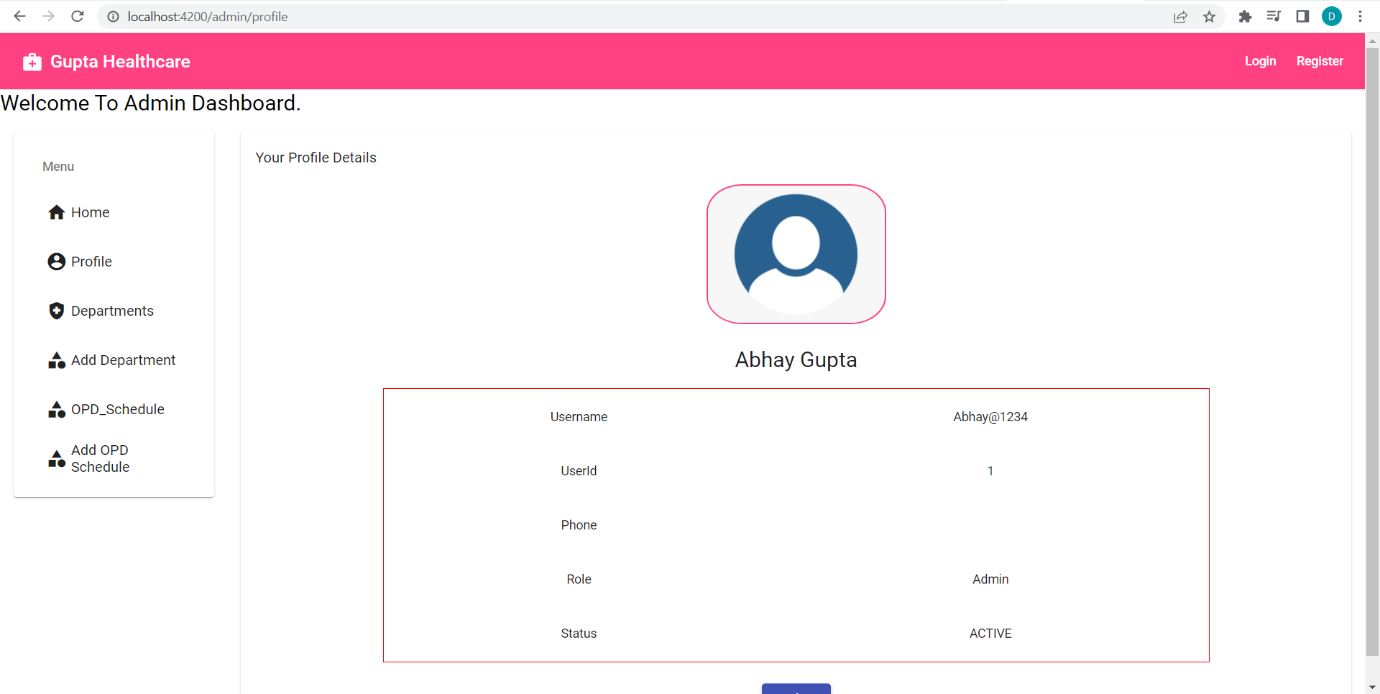
Login Page:

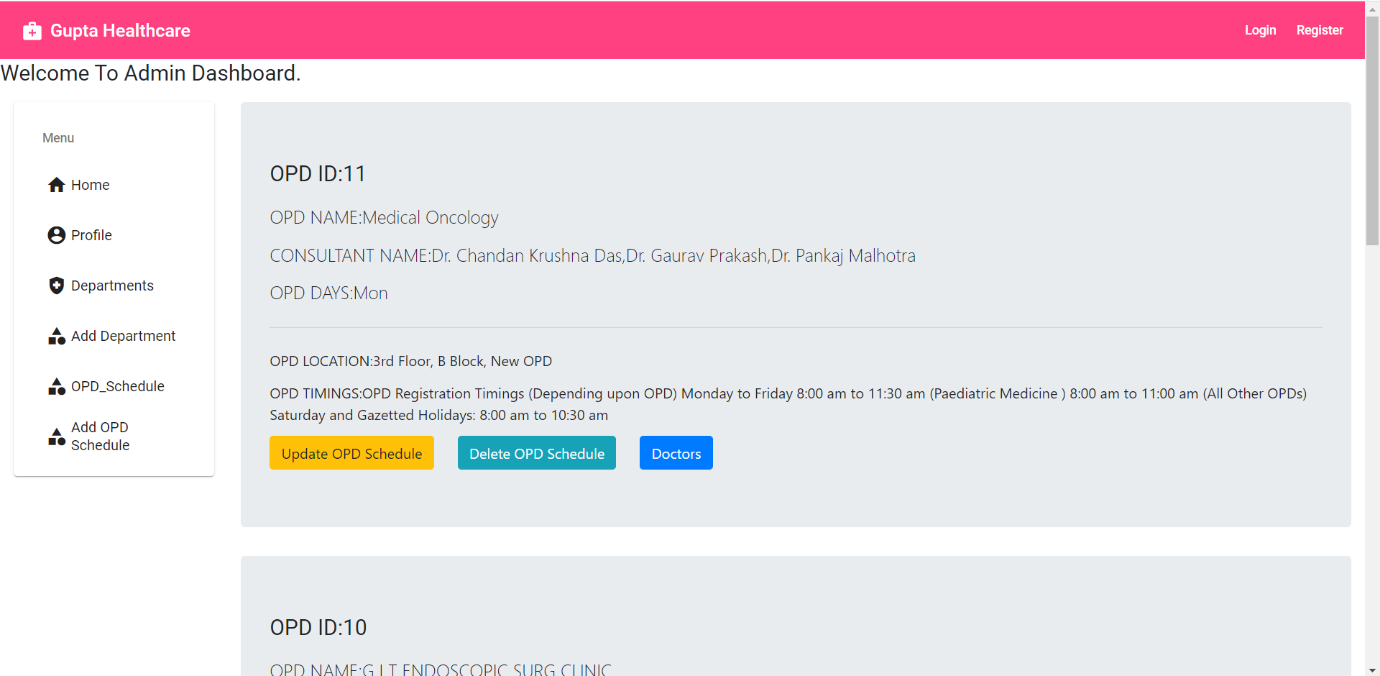


Normal User Dashboard:

Book Appointment:



Profile Details:

View OPD Schedule

Thank You Letter:

Email Id:abhaygupta.cse19@chitkarauniversity.edu.in

Date:05-May-2022

To,

Mr.Mananjay Dubey(MAPIT),Mr.Mukesh(MAPIT) and Chitkara University(Himachal Pradesh) Faculty.

Subject: Thank You for Being a Part of My Project Online Medical Management System.

Respected Mr.Mananjay Dubey,Mr.Mukesh and Chitkara University Faculty.

I would like to thank you for being a part of the Online Medical Management System Project. As the project has been successfully completed on 05-May-2022, I would like to extend my sincere gratitude towards you. It was a pleasure to work with you and you helped me like a mentor for the completion of this project.

I must say that your ideas were terrific and you have a great imagination. Thank you, once again, for helping me on this project. I look forward to working with you again.

Sincerely,

Abhay Gupta

University Id:1911985068