ST. TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM

AFFILIATED TO MAHATMA GANDHI UNIVERSITY

MEDICO

PROJECT REPORT

In partial fulfilment of the requirements for the award of the degree of

BACHELOR OF SCIENCE IN COMPUTER APPLICATIONS

[TRIPLE MAIN]

Submitted By:

DILPRIYA P.B (SB22CA012)

AND

NEHA MARIA MELVIN(SB22CA028)

III B.Sc. Computer Applications [Triple Main]

Under the guidance of

NITHYA A B

Department of Computer Applications

2022 – 2025

ST. TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM

AFFILIATED TO MAHATMA GANDHI UNIVERSITY

THE FEAR OF THE LORD IS THE BEGINNING OF WISDOM
ST. TERESA'S COLLEGE
(AUTONOMOUS)
ESTD 1925

CERTIFICATE

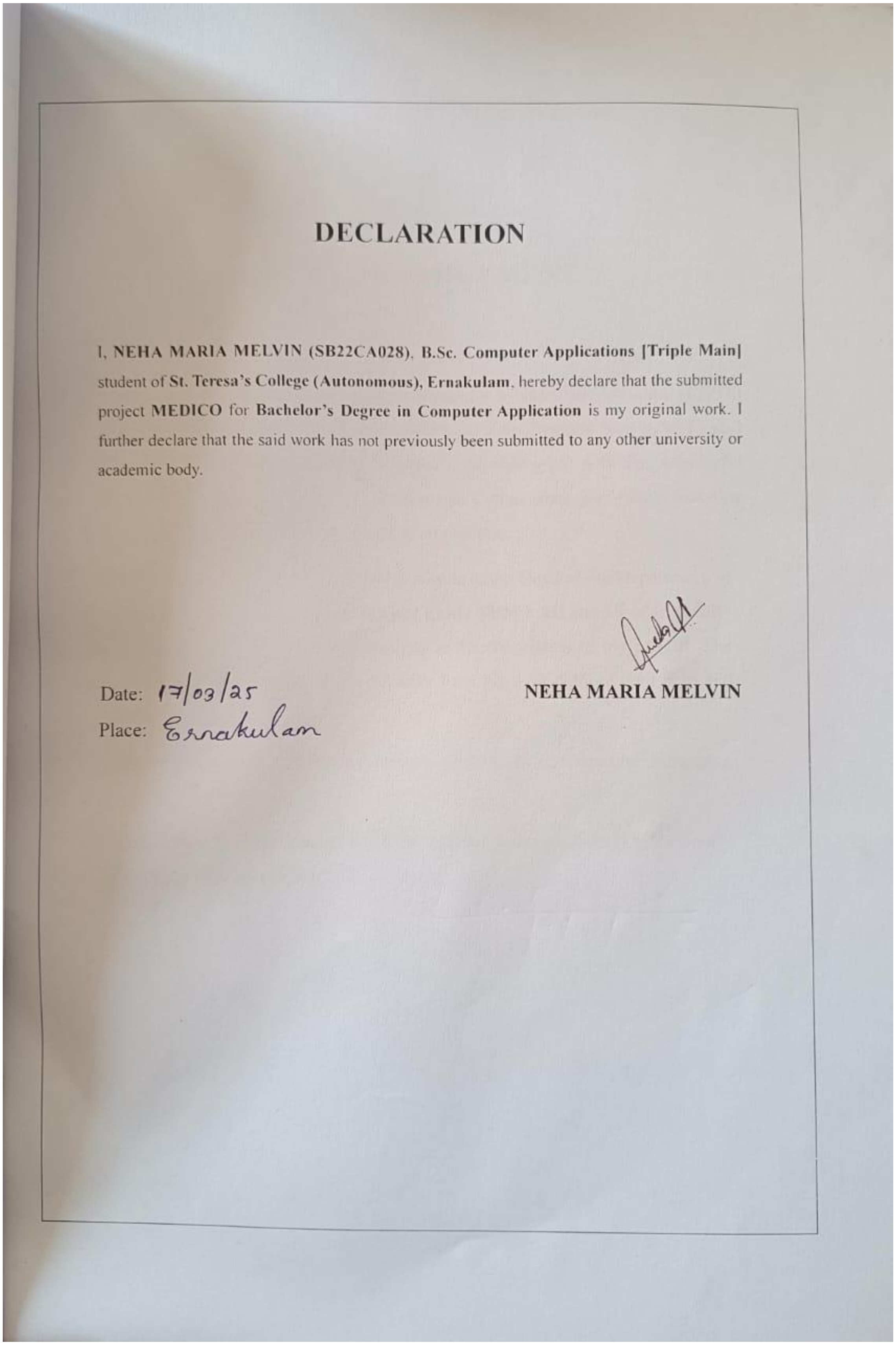
This is to certify that the project report entitled **"MEDICO"** is a bonafide record of the work done by **NEHA MARIA MELVIN**(SB22CA028) during the year 2022– 2025 and submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Applications (Triple Main) under Mahatma Gandhi University, Kottayam.

Shueba E
Head of the Department 17/03/2025

14/3/25
Internal Examiner

Date:

Roseena 17/3/'25
External Examiner
Dr. Roseena.mol.N.A -



DECLARATION

I, **NEHA MARIA MELVIN (SB22CA028)**, **B.Sc. Computer Applications [Triple Main]** student of **St. Teresa's College (Autonomous), Ernakulam**, hereby declare that the submitted project **MEDICO** for **Bachelor's Degree in Computer Application** is my original work. I further declare that the said work has not previously been submitted to any other university or academic body.

Date: 17/03/25

Place: Ernakulam

NEHA MARIA MELVIN

ACKNOWLEDGEMENT

I would like to convey my heartfelt gratitude to Rev. Sr. Nilima (CSST) Manager, Director Rev. Sr. Tessa (CSST), Rev. Sr. Francis Ann (CSST) and Principal Dr. Alphonsa Vijaya Joseph for providing me with this wonderful opportunity to work on a project with the topic Comparative Analysis of Machine Learning Algorithms and Interface Implementation.

I would like to express my profound gratitude to the Head of the Department of Computer Applications and my project guide Nithya AB and all other faculty of the department for their contributions to the completion of my project. The completion of the project would not have been possible without their help and insights.

I would also like to Thank my project guide at LCC Computer Education Institute.

Finally, I take this opportunity to Thank all them who has directly or indirectly helped me with my project.

INDEX

1. INTRODUCTION

1.1 ABOUT THE PROJECT

The demand for prompt and effective medical services has made access to healthcare an essential part of daily life. Traditional healthcare systems feature long wait periods and manual handling of medical records, which presents difficulties, particularly for people who need medical care frequently.

Users can browse several hospitals on this website, look through the departments, and locate the appropriate doctors. Additionally, they can easily schedule a consultation time with their selected physician. Prescriptions, test results, lab reports, and other medical documents can be uploaded by the consulted physician and viewed by the patient on their profile

The increasing frequency of many illnesses has made access to healthcare services crucial. In today's fast-paced world, people look for quick and easy ways to manage their health issues. This change has prompted the creation of systems that streamline healthcare procedures, increasing the accessibility and usability of medical services.

1.2 OBJECTIVE OF THE PROJECT

1. Automate Patient Registration: Make it possible for online form submissions to automatically register patients and generate login credentials.

2. Simplify Appointment Scheduling: Permit patients to schedule appointments with doctors from the location of their choice, based on availability.

3. Assure Medical Record Management: Make it easier to upload, store, and retrieve x-ray reports, lab findings, and medical data.

4. Enable Online Prescriptions: Following consultations, give patients online prescriptions that they can obtain via alerts.

5. Provide Doctor recommendations: Make it simple for patients to use the system to obtain recommendations or switch doctors.

6. Secure Online Payments: Using a variety of payment options, incorporate a secure payment gateway for consultation fees

7. Increase Patient Convenience: By including features like automated alerts and an intuitive dashboard, you may increase patient convenience.

2. SYSTEM ANALYSIS

2.1 INTRODUCTION

System Analysis is a detailed study of the various operations performed by the system and their relationship within the modules of the system. This phase involves the study of the parent system and identification of the system objectives . The main objective of this phase involves gathering necessary information and using the structured tool for analysis. This includes designing the system. In this project, the requirements are studied in detail and information are collected and documented.

2.2 EXISTING SYSTEM

□ In current scenario , you have to wait in line to take appointment for the doctors and wait for your turn to meet with them and discuss about your health problems. As you have to provide your information and other reports many times at different places such as the reception, medicine store which is again a burden of carrying documents.

□ It’s just as easy to pick up an infection in a waiting room. Imagine waiting for over an hour in a packed waiting room where other ill people are coughing and sneezing. As hard as you try, you may end up walking out of the urgent care center with a cold, flu, or other contagious illness.

2.3 PROPOSED SYSTEM

- Patients have to fill out an online form by which ID and password are created and sent to their email and they must log in to the patient panel upon accepting data. Through this panel, patients can select the doctors and have appointments with them on their own time.

- Soon after the appointment, patients will receive a message in their inbox with all of their reports and prescription medications. There is no need for cash because the necessary fees have been paid with their account, debit card, or credit card through a secure payment channel.

  Patients are empowered to take an active role in their care by being able to monitor their lab results, treatment plans, and medical records.

  Online records lower the possibility of mistakes or unnecessary testing by facilitating smooth data sharing and communication between departments or specialists.

2.4 SYSTEM SPECIFICATION

After the analyst has collected all requires information regarding the software to be developed, and has removed all completeness, inconsistence,and anomalies from specification, he starts to systematically organize the requirements the form of an SRS document. The software developers refer to the SRS document to make sure that they developed exactly what the customer requires. The SRS document helps the maintenance engineers to understand the functionality of the new system.

2.5 OPERATING SYSTEM

Windows 10 professional integrates the strengths of windows 2008 professional such as standards-based security, manageability, and reliability, with the best business features of windows 98 and windows Millennium Edition, such as plug and play, simplified user interface, and innovative support services. This combination creates the best desktop operating system for business.

It is more users friendly and a stable operating system equipped with much more added features. The operating system supports new technologies such as digital video disks, multiple monitors etc. along with plug and play and multi display features. It has a graphical user interface operating environment. Faster computing, easy access to remote information and control remote computers are some added features. Following are the common features of Windows 10.

Faster computing, easy access to remote information and control remote computers.
Built-in networking and messaging facility.

- Easier to set up, add or remove.
- Increase system security and control

2.6 LANGUAGE OR SOFTWARE PACKAGE

PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high- level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language. Python is a high-level, interpreted, interactive and object- oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python is a MUST for students and working professionals to become a great Software Engineer especially when they are working in Web Development Domain. I will list down some of the key advantages of learning Python. Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, and UNIX shell and other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

Python is often described as a "batteries included" language due to its comprehensive standard library. Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.The Python 2 language, i.e. Python 2.7.x, was officially discontinued on 1 January 2020 (first planned for 2015) after which security patches and other improvements will not be released for it. With Python 2's end- of-life, only Python 3.5.x and later are supported. Python interpreters are available for many operating systems. A global community of programmers develops and maintains Python, an open source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and Python development.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one and preferably only one obvious way to do it" design philosophy. Alex Martello, a Fellow at the Python Software foundation and Python book author, writer that “To describe something as ‘clever’ is not considered a compliment in the python culture.” Python’s developers strive to avoid premature optimization, and reject patches to non-critical part of the Python reference implementation that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move timecritical functions to extension modules written in languages such as C, or use Pippy, a just-intime compiler. Python is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

DJANGO

Django is a Python based free and open-source web framework, which follows the modeltemplate-view (MTV) architectural pattern. Django's primary goal is to ease the creation of complex, database driven websites. The framework emphasizes reusability and pluggability of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Despite having its own nomenclature, such as naming the callable objects generating the HTTP responses views, the core Django framework can be seen as an MVC architecture. It consists of an object-relational mapper (ORM) that mediates between data models (defined as Python classes) and a relational database (Model), a system for processing HTTP requests numberwith a web templating system (View), and a regular-expression-based URL dispatcher (Controller).

Also included in the core framework are:

- A lightweight and standalone web server for development and testing.
- A form serialization and validation system that can translate between HTML forms and values suitable for storage in the database.
- A template system that utilizes the concept of inheritance borrowed from object- oriented programming.
- A caching framework that can use any of several cache methods
- Support for middleware classes that can intervene at various stages of request processing and carry out custom functions
- An internal dispatcher system that allows components of an application to communicate events to each other via pre-defined signal.
- An internationalization system, including translations of Django's own components into a variety of languages
- A system for extending the capabilities of the template engine

- An interface to Python's built-in unit test framework
- Django REST framework is a powerful and flexible toolkit for building Web APIs

HYPERTEXT MARK-UP LANGUAGE (HTML)

It is the standard mark-up language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets.

Tags such as <imp /> and <input /> directly introduce content into the page. Other tags such as

<p> surround and provide information about document text and may include other tags as subelements. Browsers do not display the HTML tags but use them to interpret the content of the page.HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content.

MYSQL

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

MySQL is an important component of an open-source enterprise stack called WAMP. WAMP is a web development platform that uses Linux as the operating system, Apache as the web server, and MySQL as the relational database management system and PHP as the objectoriented scripting language. (Sometimes Perl or Python is used instead of PHP.)

Originally conceived by the Swedish company MySQL AB, MySQL was acquired by Sun Microsystems in 2008 and then by Oracle when it bought Sun in 2010. Developers can use MySQL under the GNU General Public License (GPL), but enterprises must obtain a commercial license from Oracle. Today, MySQL is the RDBMS behind many of the top websites in the world and countless corporate and consumer-facing web-based applications, including Facebook, Twitter and YouTube. MySQL is the database management system, or a database server

2.7 HARDWARE AND SOFTWARE SPECIFICATION

HARDWARE SPECIFICATION

Machine	Intel Dual Core
Mother Board	Intel 945 Chipset
Memory	4 GB
Hard Disk	500 GB
Monitor	18.5” LED Monitor
Keyboard	USB/3
Mouse	USB/3

SOFTWARE SPECIFICATION

Operating System	Windows
Web Technologies	Python, Django, HTML, CSS
Database	MySQL
Web Browser	Google Chrome/Mozilla Firefox

O Technology : PYTHON O

Framework : Django

O Web Technologies: Html,

JavaScript, CSS O IDE :

Notepad++ O Web Server : Django

Server O Database : MySQL O

Operating System: Windows

3. SYSTEM DESIGN

3.1 INTRODUCTION

System design is an interactive process through which requirements are transmitted to a "blueprint" for constructing the software initial; the blueprint depicts a holistic view of software that's design is represented at a high-level abstraction. A level that can be directly traced to specific data, functional and behavioral requirements. As design interaction occur subsequent refinement led to design representation at a much lower level of abstraction. System design is a creative art of inventing and developing input, databases, offline files, method, and procedures, for processing data to get meaningful full output that satisfy the organization objectives. Through the design phase consideration to the human factor, that is inputs to the users will have on the system.

Some of the main factors that have to be noted using the design of the system are:

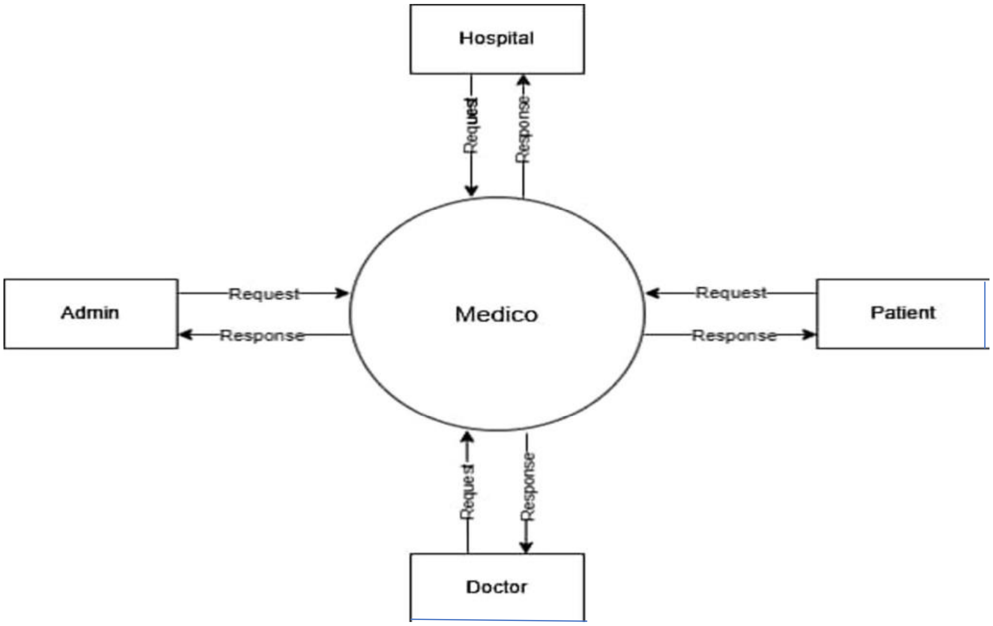
- Practicality: System must be capable of being operated over a long period of time and must have ease of use.
- Efficiency: Make better use of available resources. Efficiency involves accuracy, timeliness, and comprehensive system output.
- Cost: Aim of minimum cost and better results.
- Security: Ensure physical security of data.

3.2 DATA FLOW DIAGRAM

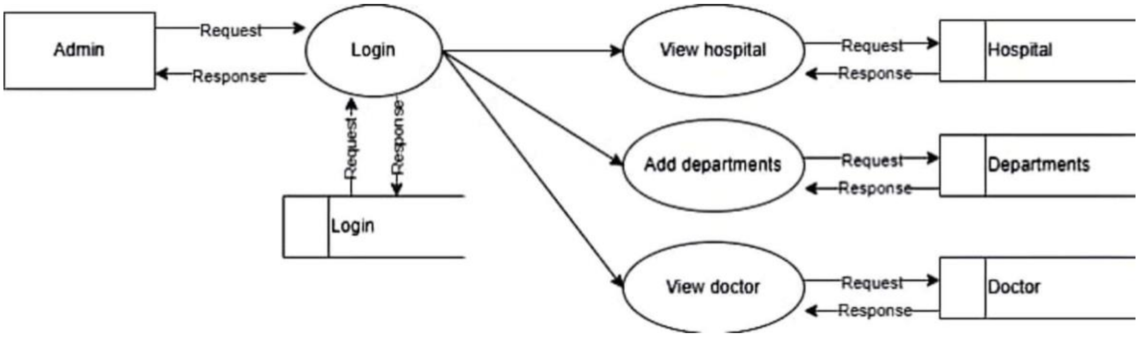
Data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. A DED shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored.

DFD is a designing tool used in the top-down approach to system Design. This context level DFD is next "exploded ", to produce a Level 1 DFD that shows some of the detail of the system being modelled. The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job and shows the flow of data between the various parts of the system.

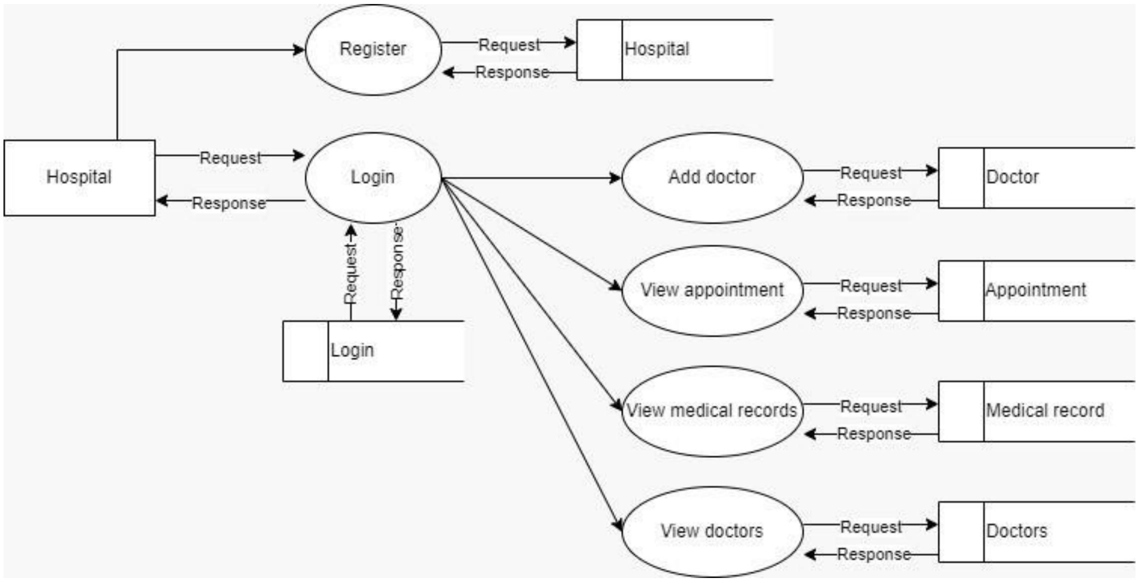
LEVEL 0



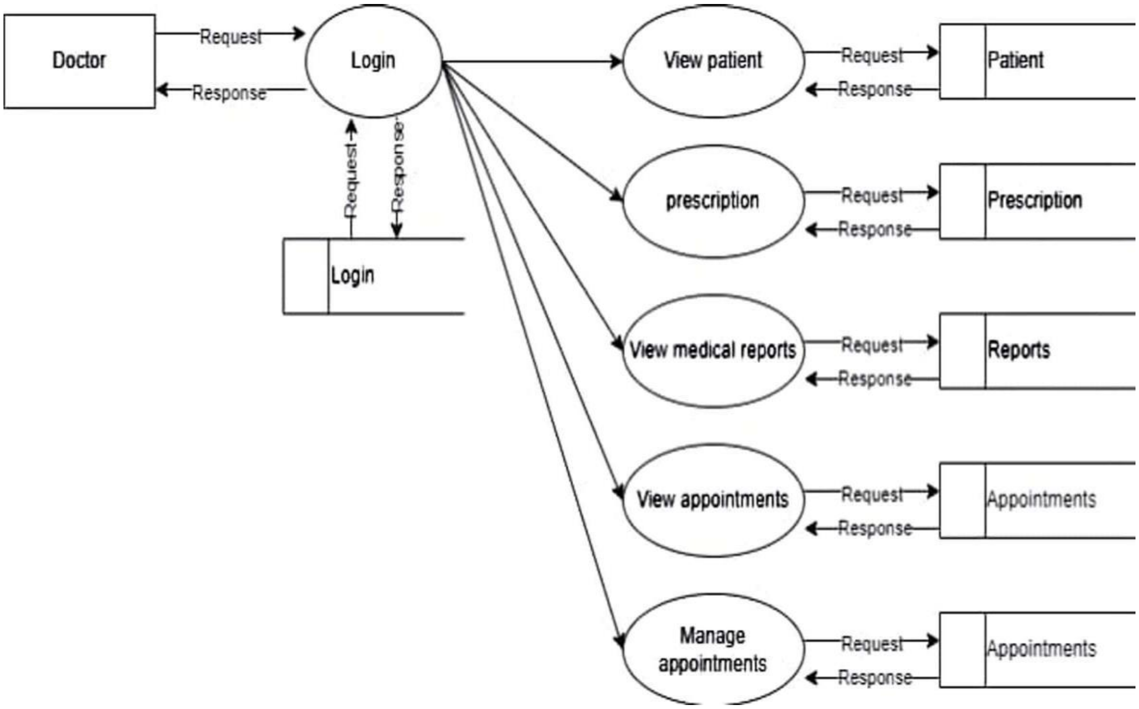
LEVEL 1 .1 ADMIN



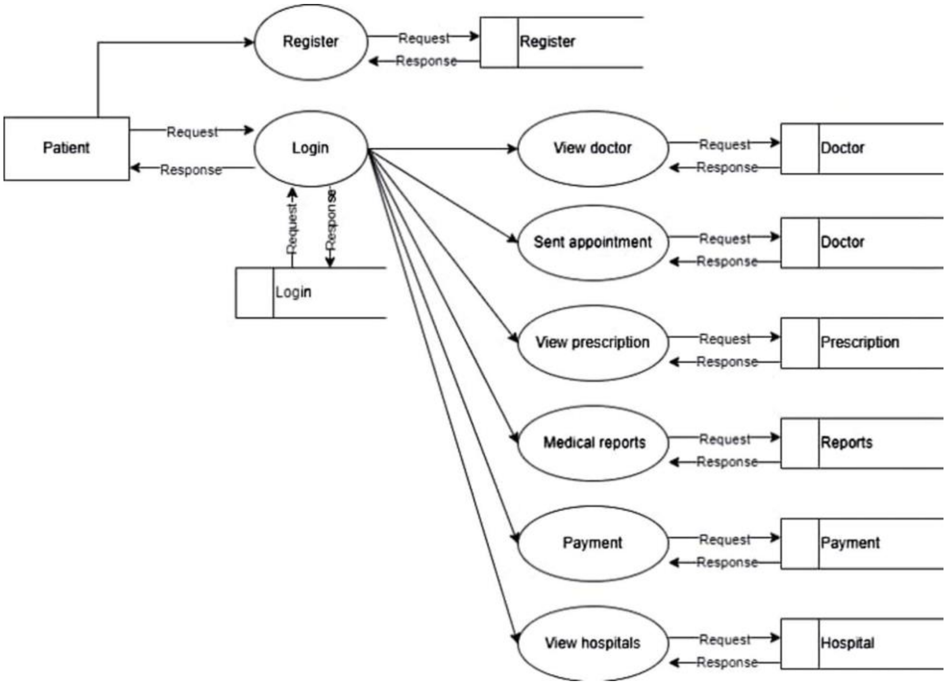
LEVEL 1.2 HOSPITAL



LEVEL 1.3 DOCTOR



LEVEL 1.4 PATIENT



3.3 DATABASE DESIGN

Database design, A most important part of the system design phase. In a database environment, data available are used by several users instead of each program managing its own data, authorized users share data across application with the database software managing the data as an entity. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive, and flexible for the users. The general theme behind a database is to integrate all information. Database design is recognized as a standard ofmanagement information system and is available virtually for every computer system.

TABLE DESIGN

TABLE 1: LOGIN

Field	Datatype	Constraint
username	Char	Null
loginid	Int	Primarykey
password	Varchar	Null
usertype	Varchar	Null

TABLE 2: HOSPITALREG

Field	Datatype	Constraint
hospitalid	Int	Primarykey
hospital_log	Int	Foreignkey
hospital_name	Char	Null

Phone	Int	Null
Email	Varchar	Null
hospital_address	Int	Null
Pincode	Int	Null
district	Varchar	Null
Image	Image	Null
license	file	null

TABLE 3:DEPARTMENT

Field	Datatype	Constraint
department_name	Char	Primarykey

TABLE 4 : DOCTOR REG

Field	Datatype	Constraint
doctor_log	int	foreignkey
hospital	char	foreignkey
department	varchar	null
fname	char	null
lname	char	null
phone	char	null
fee	char	null
email	varchar	null
doctor_address	varchar	null

image	image	null
time_to	time	null
highest_quali	char	null
time_from	time	null
quali_certi	file	null

TABLE 5:PATIENT REG

Field	Datatype	Constraint
Patient log	Int	foreignkey
fname	Char	null
lname	Char	null
Phone	Char	null
email	Varchar	null
patient_address	Varchar	null
gender	Char	null
Age	Varchar	null
weight	char	null

TABLE 6:APPOINTMENT

Field	Datatype	Constraint
patient	Int	foreignkey
doctor	Int	foreignkey
app_date	Varchar	null
booked_on	Varchar	null
app_time	Char	null
app_status	Char	null

paid_on	Date	null
paytype	char	null

TABLE 7:MEDICAL RECORD

Field	Datatype	Constraint
appointment	varchar	Foreignkey
date_of_issue	Int	null
remarks	Char	null
symptoms	Varchar	null
next_app	varchar	null

TABLE 8: Medicie_details

Field	Datatype	Constraint
medical_record	varchar	Foreignkey
m_name	Char	Null
m_dosage	char	Null
m_quantity	char	Null
m_directions	char	Null
m_days	char	null

TABLE 9: Test_details

Field	Datatype	Constraint
medical_record	Varchar	Foreignkey
test_name	Char	Null
test_upload	char	null

4. SYSTEM DEVELOPMENT

4.1 INTRODUCTION

Modular programming is a software design technique that emphasizes separating the functionality of a program into independent, interchangeable modules, such that each contains everything necessary to execute only one aspect of the desired functionality. Conceptually, modules represent a separation of concerns, and improve maintainability by enforcing logical boundaries between components.

4.2 PROCESS DESCRIPTION

MODULES

1.Patient Form and Login Module

2.Online Appointment Module

3.Medicine Module

4.Referral Module

5.Payment Module

6.Online Reports Module

1. Patient Form and Login Module

For the first time visitors, they have to just enter their basic details and can enter their dashboard. System will take care of creating their new profile. For existing patients, they will have to enter their id and password sent to their email earlier. This module will like virtual office from where all activity can be performed.

2. Online Appointment Module

Through this module, patients can select doctors and have discussion regarding their health problems. Patients will able to get their availability time or choose from the available ones and start their diagnosis immediately.

3. Medicine Module

This module will provide details of medicines which should be taken by the patients. It will also include the limit up to which these medicines should be taken and date to have meet again with doctors.

4. Referral Module

Referral module will allow patients to change their doctors. For this process, patients have to click on the doctor name whose request will be made available to particular doctor inbox and provide their meeting time.

5. Payment Module

Patients will have to make their payment online to take services using this system. Each doctor will have some fees which need to pay before their appointment session begins.

6. Online Report Module

Patients can get their lab results and health reports through this section which is available under each patient homepage. When documents are available under this module, a special notification symbol appears above it which helps to notify their patients. Thus, it provides relieve to patients for carrying these from here and there.

5. SYSTEM TESTING AND IMPLEMENTATION

5.1 TESTING

The term software testing is defined as to find for the errors in the application that might lead to fault or failure of the whole application. There are testing conditions that the system must pass to says that it is tested and working properly. The quality and reliability are also attained by going through the process of testing.

5.2 UNIT TESTING

Unit testing is a level of a software testing where individual units/components of software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing. Each module's individual components are tested separately using test cases for functionality, validation, and error handling. For example, login credentials are tested for correctness, appointment booking is validated, and payments are verified for successful transactions.

5.3 INTEGRATION TESTING

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of the level of testing is to expose faults in the interaction between integrated units. The purpose of this level of testing is to expose faults in the intersection between integrated units. The drivers and test stubs are used to assist in Integration Testing. Modules are tested together to ensure smooth data flow and interaction. For instance, when a patient books an appointment, it should reflect in the doctor's schedule, and payment completion should update the transaction history and reports module.

5.4 SYSTEMS TESTING

System testing is a level of software testing where complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. The entire system is tested for performance, security, and compatibility under real-world conditions. Load testing ensures multiple users can access the system simultaneously, while security testing verifies encrypted storage and restricted access to medical records

5.5 ACCEPTANCE TESTING

Acceptance testing is performed to ensure that the functional, behavioral, and performance requirements of the software are met IEEE defines acceptance testing as a 'formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.'

During acceptance testing, the software is tested and evaluated by a group of users either at the developer's site or user's site. This enables the users to site. The enables the users to test the software themselves and analyses whether it is meeting their requirements. To perform acceptance testing, a predetermined set of data is given to the software as input. It is important to know the expected output before performing acceptance testing so that outputs produced by the software as a result of testing can be compared with them. Based on the results of tests decide whether to accept or reject the software is correct and is accepted; otherwise, it is rejected. End-users (patients and doctors) test the system to ensure it meets their needs. Reallife scenarios such as booking an appointment, making payments, accessing reports, and consulting multiple doctors are performed to confirm usability and efficiency before deployment.

.

5.6 SECURITY

The protection of computer-based resources that includes hardware, software, data, procedures and against unauthorized use or natural.

- Integrity
- Privacy
- Disaster is known as system security.
- System security can be divided into four related issues.
- Confidentiality
- Security

Data security is the protection of data from loss, disclosure, modification and destruction.

System Integrity refers to the power functioning of hardware and programs, appropriate physical security, and safety against external threats such as eavesdropping and writes tapping

5.6.1Scope for Future Enhancement

- Before seeing a doctor, incorporate an AI-powered chatbot to offer initial health evaluations.
  AI can assist in assigning patients to the appropriate specialist and triaging them according to urgency.
- Improve the technology to enable high-quality voice and video communications so that patients and doctors can communicate more effectively.
  Provide real-time chat assistance so that patients and doctors can communicate instantly.
  Create an iOS and Android mobile application to improve accessibility.
- Turn on push alerts for prescription changes and appointment reminders Collaborate with internet pharmacies to enable platform users to order medications directly.

Permit prescription drugs to be delivered to homes.

- Provide multilingual support to serve a varied patient population. For patients who are old or visually challenged, incorporate voice-assisted navigation.
- Include online therapy sessions with licensed psychologists as part of the expanded offerings.

  Introduce programs for stress management and meditation.

SYSTEM MAINTENANCE

Maintenance means restoring something to its original conditions. Enhancement means adding, modifying the code to support the changes in the user specification. System maintenance conforms the system to its original requirements and enhancement adds to system capability by incorporating new requirements.

Thus, maintenance changes the existing system, enhancement adds features to the existing system, and development replaces the existing system. It is an important part of system development that includes the activities which corrects errors in system design and implementation, updates the documents, and tests the data.

Maintenance Types

System maintenance can be classified into four types –

- Corrective Maintenance
- Adaptive Maintenance
- Perfective Maintenance
- Preventive Maintenance

Corrective Maintenance

Corrective Maintenance deals with the repair of faults or defects found in day- today system functions. A defect can result due to errors in software design, logic and coding. Design errors occur when changes made to the software are incorrect, incomplete, wrongly communicated, or the change request is misunderstood. Logical errors result from invalid tests and conclusions, incorrect implementation of design specifications, faulty logic flow, or incomplete implementation of design specifications, faulty logic flow, or incomplete test of data. All these errors, referred to as residual errors, prevent the software from confirming to its agreed specifications. Note that the need for corrective maintenance is usually initiated by big reports drawn by the users.

Adaptive Maintenance

Adaptive Maintenance is the implementation of changes in a part of the system, which has been affected by a change that occurred in some other part of the system. Adaptive Maintenance consists of adapting software to changes in the environment such as the hardware or the operating system. The term environment in this context refers to the conditions and the influences which act (from outside) on the system. For example, business rules, work patterns and government policies have a significant impact on the software system

Perfective Maintenance

Perfective Maintenance mainly deals with implementing new or changed user requirements. Perfective Maintenance involves making functional enhancements to the system in addition to the activities to increase the system's performance even when the changes have not been suggested by faults. This includes enhancing both the function and efficiency of the code and changing the functionalities of the system as per the users' changing needs.

Preventive Maintenance

Preventive Maintenance involves performing activities to prevent the occurrence of errors. It tends to reduce the software complexity thereby improving program understand ability and increasing software maintainability. It comprises documentation updating, code optimization and code restructuring. Documentation updating involves modifying the documents affected by the changes in order to correspond to the present state of the system. Code optimization involves modifying the programs for faster execution or efficient use of storage space. Code restructuring involves transforming the program structure for reducing the complexity in source code and making it easier to understand.

6.CONCLUSION

By providing a smooth, digital healthcare solution, the Medico-The Online medical project successfully tackles the difficulties related to conventional in-person medical consultations. By allowing patients to schedule appointments, obtain prescriptions, and view medical reports from the comfort of their homes, this technology improves accessibility, convenience, and efficiency. The technology greatly enhances the patient experience by removing lengthy wait times, reducing in-person encounters, and incorporating secure payment channels.

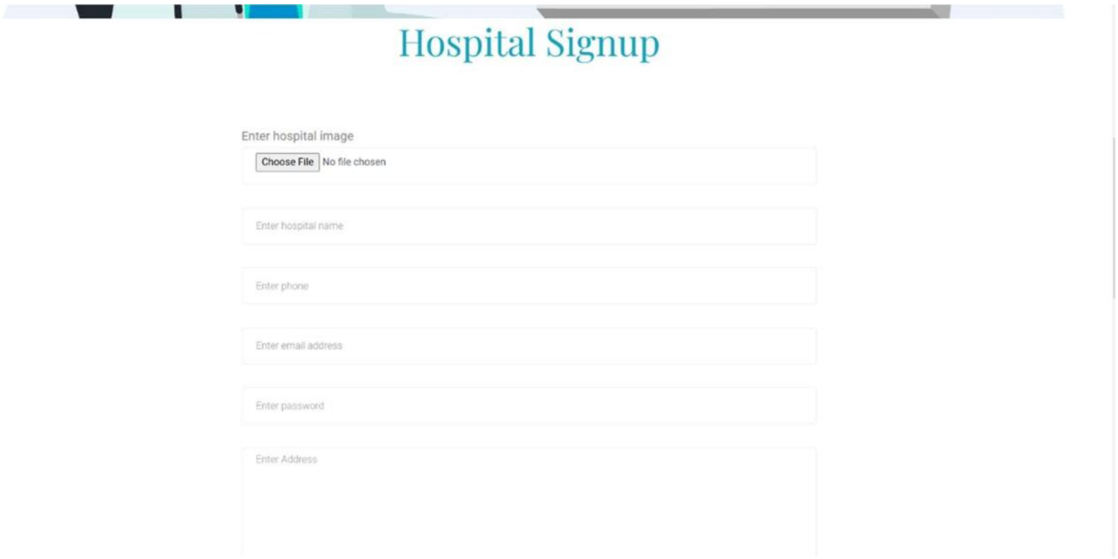
This system guarantees a seamless and efficient healthcare process through the deployment of essential modules like patient login, online appointments, medication tracking, referrals, and report management. Contemporary technologies like Python, Django, and MySQL guarantee system scalability, security, and dependability.

7. APPENDIX

7.1 Input and Output Screen

Input Design

Signup-hospital



Hospital Signup

Enter hospital image

Enter hospital name

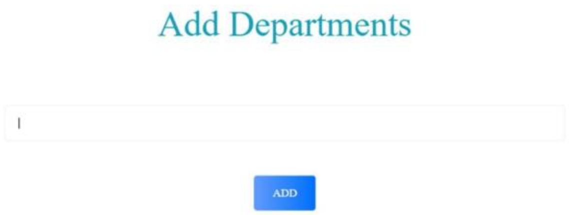
Enter phone

Enter email address

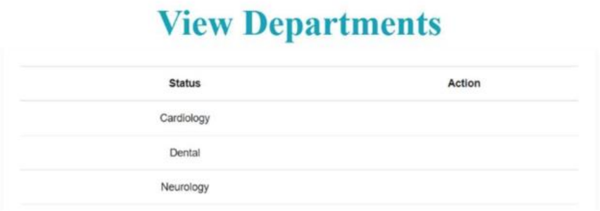
Enter password

Enter Address

Add depts



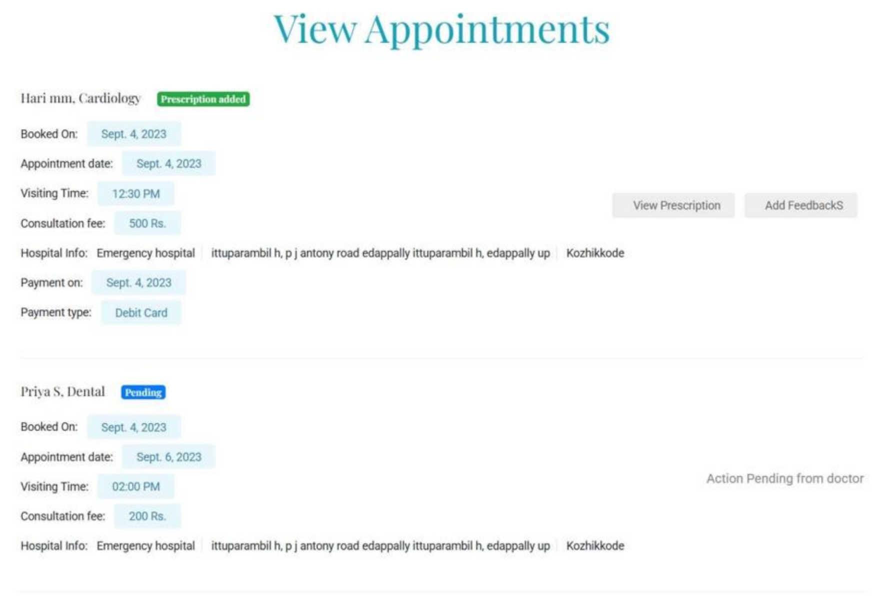
Add Departments



View Departments

Status	Action
Cardiology	
Dental	
Neurology	

Appointments



View Appointments

Hari mm, Cardiology **Prescription added**

Booked On: Sept. 4, 2023

Appointment date: Sept. 4, 2023

Visiting Time: 12:30 PM

Consultation fee: 500 Rs.

Hospital Info: Emergency hospital | ittuparambil h, p j antony road edappally ittuparambil h, edappally up | Kozhikkode

Payment on: Sept. 4, 2023

Payment type: Debit Card

Priya S, Dental **Pending**

Booked On: Sept. 4, 2023

Appointment date: Sept. 6, 2023

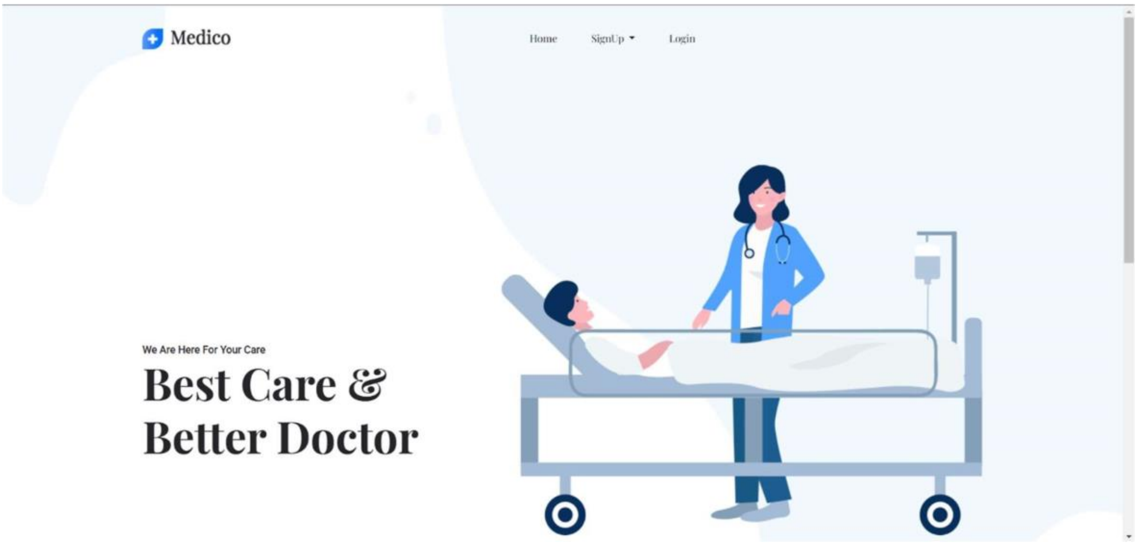
Visiting Time: 02:00 PM

Consultation fee: 200 Rs.

Hospital Info: Emergency hospital | ittuparambil h, p j antony road edappally ittuparambil h, edappally up | Kozhikkode

Action Pending from doctor

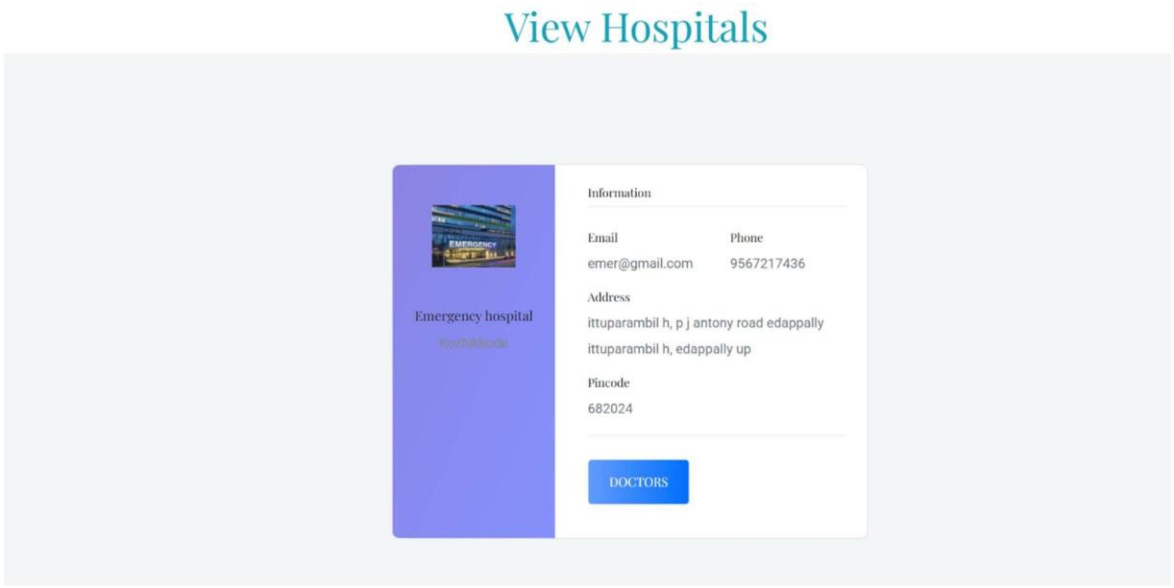
Output Screen Homepage:



We Are Here For Your Care

Best Care & Better Doctor

Hospital view:



View Hospitals

Emergency hospital

Kozhikkode

Information

Email

emer@gmail.com

Phone

9567217436

Address

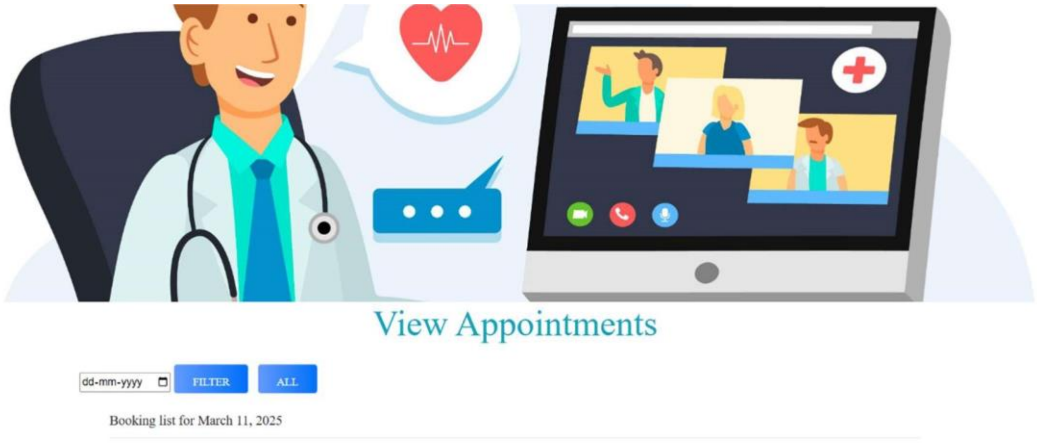
ittuparambil h, p j antony road edappally

ittuparambil h, edappally up

Pincode

682024

Appointment view:



View Appointments

Booking list for March 11, 2025

7.2 Sample code

```
user login def
users_login(request):
if request.POST:
    email=request.POST['email']
password=request.POST['password']
user=authenticate(username=email,password=password)
    if a user is not None:          if
user.user_type=='admin':
        msg=messages.success(request,'Welcome to admin
dashboard')          return redirect('/admin-dashboard')          elif
user.user_type=='hospital':
        msg=messages.success(request,'Welcome to hospital
dashboard')          userregid=user
hospitalid=HospitalReg.objects.get(hospital_log=userregid).id
request.session['hid']=hospitalid          return redirect('/hospital-
dashboard')          elif user.user_type=='patient':
        msg=messages.success(request,'Welcome to patient
dashboard')          patientregid=user
patientid=PatientReg.objects.get(patient_log=patientregid).id
request.session['pid']=patientid          print(request.session['pid'])
return redirect('/patient-dashboard')
      elif user.user_type=='doctor':
            msg=messages.success(request,'Welcome to doctor
dashboard')            doctorregid=user
doctorid=DoctorReg.objects.get(doctor_log=doctorregid).id
request.session['did']=doctorid            return redirect('/doctor-
dashboard')      else:
        msg=messages.success(request,'Invalid Login
again')        return redirect('/users-login')    return
render(request,'users_login.html')
```

Hospital signup

```
def
hospital_signup(request):
if request.POST:
    name=request.POST['name']
phone=request.POST['phone']
image=request.FILES['image']
email=request.POST['email']
password=request.POST['password']
district=request.POST['district']
address=request.POST['address']
pin=request.POST['pincode']
licence=request.FILES['licence']      if
Login.objects.filter(username=email).exists():
print('exists,,,,,,')
```

```
            msg=messages.success(request,'Already Taken')

            return         redirect('/')

else:


h_login=Login.objects.create_user(user_type='hospital',view_password=password,use
rname=email,password=password,is_active=0)

            h_login.save()
hadd=HospitalReg.objects.create(hospital_log=h_login,hospital_name=name,image=i
mage,phone=phone,


email=email,district=district,hospital_address=address,pincode
=pin,

                                        licence=licence)            hadd.save()

msg=messages.success(request,'Hospital added sucessfully, Wait for approval')

            return redirect('/')     return

render(request,'hospital_signup.html') Adding Doctor def

hospital_adddoctor(request):

departments=Department.objects.all().order_by('department_name')

hospital=request.session['hid']     print(hospital)     if request.POST:

        fname=request.POST['fname']

lname=request.POST['lname']

phone=request.POST['phone']         fee=request.POST['fee']

image=request.FILES['image']

email=request.POST['email']
```

```
password=request.POST['password']

hqualification=request.POST['hqualification']

qcertificate=request.FILES['qcertificate']

address=request.POST['address']

department=request.POST['department']

tfrom=request.POST['tfrom']         tto=request.POST['tto']

if Login.objects.filter(username=email).exists():

        print('exists,,,,,,')

        msg=messages.success(request,'Already

Taken')            return redirect('/hospital-adddoctor')

else:


d_login=Login.objects.create_user(user_type='doctor',view_password=password,user
name=email,password=password)

        d_login.save()

dadd=DoctorReg.objects.create(doctor_log=d_login,hospital_id=hospital,department_

id=department,fname=fname,fee=fee,lname=lname,image=image,phone=phone,


email=email,highest_quali=hqualification,quali_certi=qcertific
ate,doctor_address=address,time_from=tfrom,

                                time_to=tto)            dadd.save()

msg=messages.success(request,'Doctor added sucessfully')

return redirect('/hospital-viewdoctors')     return

render(request,'hospital/hospital_adddoctor.html',{"departments"
:departments}) Appointment def

doctor_viewappointments(request):

   doctor=request.session['did']     print(doctor)

doctor=DoctorReg.objects.get(id=doctor)     today = date.today()

print(today,'datemmmmz')

appointments=Appointment.objects.filter(doctor=doctor,app_date=today)

ddate=today     if request.POST:

      ddate=request.POST['ddate']

appointments=Appointment.objects.filter(doctor=doctor,app_date=ddate)

   return
render(request,'doctor/doctor_viewappointments.html',{'appointments':appointments,'app_
dat e':ddate,'today':today})
```

8. BIBLIOGRAPHY

REFERENCES

1.https://www.reddit.com/r/learnpython/comments/1breznt/reference_projects_to_learn_python_efficiently/?rdt=34035

2.https://www.w3schools.com/references/index.php

3.https://www.pluralsight.com/paths/javascript-core-language

4.https://dev.mysql.com/doc/refman/8.4/en/

5. https://docs.djangoproject.com/en/5.1/intro/tutorial01/

6.Python Cookbook-avid Beazley, Brian Jeffrey-2020