

Hospital Management System

A Project Report

Submitted in the partial fulfillment of the requirements for Technical Proficiency – I (20TS3101K)

Bachelor of Technology In

Department of Artificial Intelligence & Data Science

By

|  |  |
| --- | --- |
| NAME | ID No. |
| GULLA ADITHYA | 2000080039 |

under the supervision of Ms. D. Deepika

Department of Artificial Intelligence & Data Science K L E F, Green Fields,

Vaddeswaram- 522502, Guntur(Dist), Andhra Pradesh, India.

November,2022

# DECLARATION

The project report entitled “Hospital Management System” is a record bonafide work of GULLA ADITHYA – 2000080039 submitted in partial fulfillment for the award of B.Tech in Artificial Intelligence and Data Science to the KL University. The Results embodied in this report have not been copied from any other contracts/University/Institute.

|  |  |
| --- | --- |
| GULLA ADITHYA | 2000080039 |



# CERTIFICATE

This is to certify that the project-based laboratory report entitled “HOSPITAL MANAGEMENT SYSTEM” submitted by **GULLA ADITHYA** bearing Regd. No. **2000080039** to the **Department of Artificial intelligence and Data Science, KL University** in partial fulfillment of the requirements for the completion of a skilling based Laboratory in **“TECHNICAL PROFECIENCY- 1”** course in 3rd B Tech 1st Semester, is a bonafide record of the work carried out by us under the supervision of **Ms. D. Deepika** during the academic year 2022 – 2023.

PROJECT SUPERVISOR HEAD OF THE DEPARTMENT Dr. TIRAPATHI REDDY

# ACKNOWLEDGEMENTS

We would like to express our gratitude towards Ms. Deepika project supervisor and our honorable HOD Dr. TIRAPATHI REDDY BURRAMUKKU of KONERU LAKSHMAIAH EDUCATION FOUNDATION for their support in accomplishment of our project on “HOSPITAL MANAGEMENT SYSTEM”.

We express our thanks to Project Supervisor Ms. Deepika, for his Continuous support. and encouragement in completion of our project. We thank all teaching faculty of Department of Artificial Intelligence and Data Science, whose suggestions during reviews helped us in accomplishment of our project.

I would like to extend my deep appreciation to all my group members, without their support and coordination we would not have been able to complete this project

# ABSTRACT

The goal of the project "Hospital Management System " is to computerize hospital front desk administration in order to provide a software solution that is user-friendly, simple, quick, and cost-effective. It deals with compiling patient data, such as identity information, etc. In the past, it had been carried out manually. The main functions of the system are to register, store, and retrieve patient and doctor information as needed, as well as to meaningfully govern these facts. Patient information and identity information are fed into the system, and these facts are presented on the screen. A username and countersign will be used to log into the hospital management system. An associate degree administrator or secretary can access it. They alone will contribute information to the data. It will be easy to obtain the data. Information processing is quick thanks to the information's strong privacy protections.

# INDEX

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **PAGE NO** |
| 1 | Introduction | 6 |
| 2 | Modules Explanation Theoretically | 7 |
| 3 | Implementation Concept or Subject Knowledge | 8 |
| 4 | ER Diagram | 9 |
| 5 | Project Flow Diagram | 10 |
| 6 | Database Information | 11 |
| 7 | Implementation Using Flask | 12-29 |
| 8 | Implementation Using Django | 30-38 |
| 9 | Output Screenshots | 39-44 |
| 10 | Conclusion | 45 |

**INTRODUCTION**

The project Hospital Management system involves patient registration, saving personal information in the system, as well as automated lab and pharmacy ordering. Each patient's unique ID may be shown by the software package, which also automatically stores each patient's and each employee's tiny print. It contains a facility for understanding the status of each region through groundwork. Users will use the ID to search for a doctor's convenience as well as patient information. A username and countersign will be used to log into the hospital management system. An associate degree administrator or secretary can access it. They alone will contribute information to the data. It will be easy to obtain the data. The user interface is simple. Information processing is quick thanks to the information's strong privacy protections. The robust, adaptable, and user-friendly Hospital Management System was designed to provide hospitals every potential benefit.

# MODULES EXPLANATION THEORETICALLY

## Admin:

The entity in charge of making the survey is called Admin. The administrator can check the responses from each student who completed the survey and add new questions as needed. so that the administrator can examine the information to comprehend and assist the student in comprehending their degree of subject understanding.

## Student:

The student portion will allow students to participate in taking quizzes for subjects which can help them in self-assessment and also lets the instructor/teacher get an idea of updates regarding the course without a need to constantly meet them physically for the college welfare such as selecting class representative, college general secretary etc. they are also allowed to participate in the college survey and decision-making system for the college.

# IMPLEMENTATION CONCEPT

## Django

The model-template-views (MTV) architectural pattern is used by the Python-based web framework known as Django . The Django Software Foundation (DSF), an independent organisation founded in the US as a non-profit, is responsible for maintaining Django.

Django's main objective is to make it simpler to create intricate, database-driven websites. Python is utilised extensively, including for settings, files, and data models. The framework emphasises reusability and "pluggability" of components, less code, low coupling, rapid development, and the idea of don't repeat yourself. Additionally, Django offers a customizable administrative creation, read, update, and delete interface that is dynamically built by introspection.

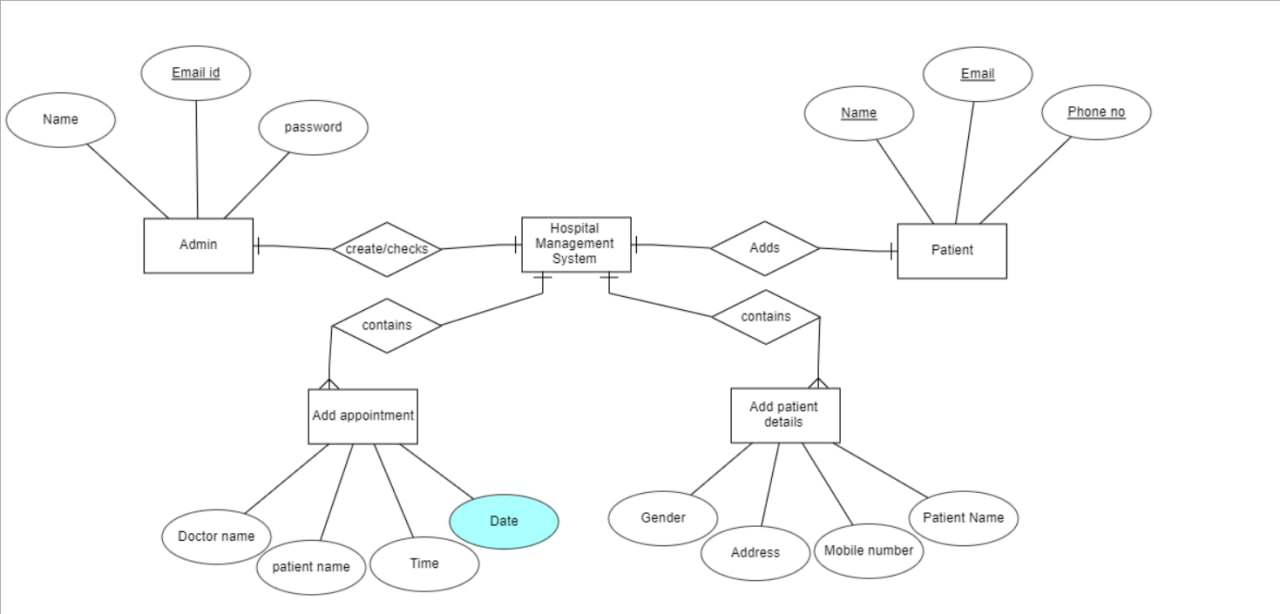
Several well-known websites use Django, including Nextdoor, Clubhouse, Mozilla, Disqus, Bitbucket, and Instagram .

## Flask

Python-based Flask is a microweb framework. Due to the fact that it doesn't require any specific tools or libraries, it is categorized as a microframework. It lacks any components where pre-existing third-party libraries already provide common functions, such as a database abstraction layer, form validation, or other components. However, Flask allows for extensions that may be used to add application functionalities just like they were built into the core of Flask. There are extensions for object-relational mappers, form validation, upload handling, several open authentication protocols, and several utilities associated with popular frameworks.

The Flask framework is used by applications like Pinterest and LinkedIn.

**ER Diagram**



**PROJECT FLOW DIAGRAM**

Diagram

Description automatically generated

**DATABASE INFORMATION**

# Data Base

We use Sqllite data base to store the data of the patients give to us while applying for a appointment**.**

A screenshot of a computer

Description automatically generated with low confidence

# Doctors Details

Here also we use the Sql-lite data base for storing the data.

Table

Description automatically generated

# IMPLEMENTATION USING

# DJANGO

**HTML FILES**

## Navigation.html

<!DOCTYPE html>

<html lang="en">

<head>

{% load static %}

<title>Hospital Management System</title>

<link rel="icon" href="images/favicon.ico">

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<script src="https://unpkg.com/ionicons@5.0.0/dist/ionicons.js"></script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>

<link rel = stylesheet href = "//cdn.datatables.net/1.10.19/css/jquery.dataTables.min.css">

<script src="//cdn.datatables.net/1.10.19/js/jquery.dataTables.min.js"></script>

<script>

$(document).ready( function () {

$('#myTable').DataTable();

} );

</script>

<style>

\* {

box-sizing: border-box;

}

#myInput {

background-image: url('/css/searchicon.png');

background-position: 10px 10px;

background-repeat: no-repeat;

width: 20%;

font-size: 16px;

padding: 12px 20px 12px 40px;

border: 1px solid #ddd;

margin-bottom: 12px;

}

#myTable {

border-collapse: collapse;

width: 100%;

border: 1px solid #ddd;

font-size: 18px;

}

#myTable th {

text-align: left;

padding: 12px;

background-color: green;

color: white;

}

#myTable td {

text-align: left;

padding: 12px;

background-color: white;

}

#myTable tr {

border-bottom: 1px solid #ddd;

}

#myTable tr.header, #myTable tr:hover {

background-color: #f1f1f1;

}

.button {

background-color: orange; /\* Green \*/

border: none;

color: white;

padding: 10px;

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 14px;

margin: 1px 1px;

cursor: pointer;

}

.button1 {border-radius: 15px;}

/\* Make the image fully responsive \*/

.carousel-inner img {

width: 100%;

height: 100%;

}

</style>

<style>

.nav-link:hover{

color: lightblue !important;

font-weight:bold;

background-color : #931B86;

border-radius:10px

}

<link rel="stylesheet" href="https://pro.fontawesome.com/releases/v5.10.0/css/all.css" integrity="sha384-AYmEC3Yw5cVb3ZcuHtOA93w35dYTsvhLPVnYs9eStHfGJvOvKxVfELGroGkvsg+p" crossorigin="anonymous"/>

</head>

<body style="background-color: #f0f1f2;">

<nav class="navbar navbar-expand-sm" style = "background-image : url({% static 'images/banner.svg' %}) ; border-bottom:1px solid red">

<a class="navbar-brand pl-5" href="#" style="color:white ; font-weight : bold">Hospital Management System</a>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="collapsibleNavbar">

<ul class="navbar-nav pl-5 text-center">

<li class="nav-item">

<a class="nav-link" href="{% url 'index' %}" style="color:white"><ion-icon style="margin-right:2px" name="home-sharp"></ion-icon> Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'about' %}" style="color:white"><ion-icon style="margin-right:2px" name="people"></ion-icon> About</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'contact' %}" style="color:white"><ion-icon style="margin-right:2px" name="people"></ion-icon> Book Appointment</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'login' %}" style="color:white"><i class="fas fa-sign-in-alt" style="margin-right:2px"></i> Admin Login</a>

</li>

</ul>

</div>

</nav>

{% block body %}

{% endblock %}

## admin.html

<!DOCTYPE html>

<html lang="en">

<head>

{% load static %}

<title>Hospital Management System</title>

<link rel="icon" href="images/favicon.ico">

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<script src="https://unpkg.com/ionicons@5.0.0/dist/ionicons.js"></script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>

<link rel = stylesheet href = "//cdn.datatables.net/1.10.19/css/jquery.dataTables.min.css">

<script src="//cdn.datatables.net/1.10.19/js/jquery.dataTables.min.js"></script>

<script>

$(document).ready( function () {

$('#myTable').DataTable();

} );

</script>

<body>

<br><br><br>

<div class="market-updates">

<div class="col-md-4 market-update-gd">

<div class="market-update-block clr-block-1">

<div class="col-md-8 market-update-left">

<h3>{{dc}}</h3>

<h4><a href="{% url 'view\_doctor' %}" style="color : white">Total Doctor</a></h4>

</div>

<div class="col-md-4 market-update-right">

<i class="fa fa-user-md"></i>

</div>

<div class="clearfix"> </div>

</div>

</div>

<div class="col-md-4 market-update-gd">

<div class="market-update-block clr-block-2">

<div class="col-md-8 market-update-left">

<h3>{{pc}}</h3>

<h4><a href="{% url 'view\_patient' %}" style="color : white;">Total Patient</a></h4>

</div>

<div class="col-md-4 market-update-right">

<i class="fa fa-user-o"></i>

</div>

<div class="clearfix"> </div>

</a>

</li>

<li class="nav-item">

<a href="/showall" class="nav-link">

<p>

Check All Results

</p>

</a>

</li>

<li class="nav-item">

<a href="/logout" class="nav-link">

<p>

Logout

</p>

</a>

</li>

{% endblock %}

## index.html

{% extends 'navigation.html' %}

{% load static %}

{% block body %}

<head>

<style media="screen">

.jumbotron {

width : 100;

height : 500px;

background-image: url('{% static "images/h3.jpg" %}');

background-size: cover;

background-repeat: no-repeat;

}

.jumbotron h5,

h3 {

text-align: center;

}

.alert {

margin: 1px;

</style>

</head>

<br>

<br>

<div class="jumbotron" style="margin-bottom: 0px;margin-top: 0px;">

<br><br>

<h5>HOSPITAL MANAGEMENT SYSTEM</h5>

<br><br><br><br>

</div>

{% endblock%

## login.html

{% extends 'navigation.html' %}

{% load static %}

{% block body %}

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<title>Admin Login </title>

<style type="text/css">

body {

color: #aa082e;

background-color: #b6bde7;

font-family: 'Roboto', sans-serif;

}

a:link {

text-decoration: none;

}

.note {

text-align: center;

height: 80px;

background: -webkit-linear-gradient(left, #0072ff, #8811c5);

color: #fff;

font-weight: bold;

line-height: 80px;

}

.form-content {

padding: 4%;

border: 1px solid #ced4da;

margin-bottom: 1%;

text-align : center;

}

.form-control {

border-radius: 1.5rem;

text-align : center;

}

.btnSubmit {

border: none;

border-radius: 1.5rem;

padding: 1%;

width: 30%;

cursor: pointer;

background: #0062cc;

color: #fff;

}

</style>

</head>

<body>

{% if error == "no" %}

<script>

alert('Logged In Successfully');

window.location=('{% url 'admin\_home' %}');

</script>

{% endif %}

{% if error == "yes" %}

<script>

alert('Something went wrong, Try Again');

</script>

{% endif %}

<br><br><br><br>

<form method="post">

{% csrf\_token %}

<div class="container register-form mt-2" style="width : 600px;">

<div class="form">

<div class="note">

<p>Admin Login</p>

</div>

<div class="form-content">

<div class="row mt-5">

<div class="col-md-12">

<div class="form-group">

<input type="text" name="uname" class="form-control py-3" placeholder="Enter User Name">

</div>

</div>

<div class="col-md-12">

<div class="form-group">

<input type="password" name="pwd" class="form-control py-3" placeholder="Enter Password"><br>

</div>

</div>

</div>

<button type="submit" class="btnSubmit">Login</button>

</div>

</div>

</div>

</form>

background: #0062cc;

color: #fff;

}

</style>

</head>

<body>

{% if error == "no" %}

<script>

alert('Logged In Successfully');

window.location=('{% url 'admin\_home' %}');

</script>

{% endif %}

{% if error == "yes" %}

<script>

alert('Something went wrong, Try Again');

</script>

{% endif %}

<br><br><br><br>

<form method="post">

{% csrf\_token %}

<div class="container register-form mt-2" style="width : 600px;">

<div class="form">

<div class="note">

<p>Admin Login</p>

</div>

<div class="form-content">

<div class="row mt-5">

<div class="col-md-12">

<div class="form-group">

<input type="text" name="uname" class="form-control py-3" placeholder="Enter User Name">

</div>

</div>

<div class="col-md-12">

<div class="form-group">

<input type="password" name="pwd" class="form-control py-3" placeholder="Enter Password"><br>

</div>

</div>

</div>

<button type="submit" class="btnSubmit">Login</button>

</div>

</div>

</div>

</form>

{% endblock%}

## contact.html

{% extends 'navigation.html' %}

{% load static %}

{% block body %}

{% if error == "no" %}

<script>

alert('Your Message has been Send Successfully');

window.location=('{% url 'contact' %}');

</script>

{% endif %}

{% if error == "yes" %}

<script>

alert('Something went wrong, Try Again');

</script>

{% endif %}

<section style="background-color : #21446B; height : 650px;">

<div class="container py-5 mt-3">

<div class="card">

<div class="card-body">

<h1 class="font-weight-light text-center py-3">Book Appointment</h1>

<div class="row">

<div class="col-lg-6 col-md-12 col-sm-12">

<div class="row pt-3">

<div class="col-lg-1 offset-1 col-md-2 col-sm-2">

<span style="font-size: 30px; color: cadetblue"><i class="fas fa-map-marker-alt"></i></span>

</div>

<div class="col-lg-10 col-md-9 col-sm-9">

<h3 class="font-weight-light">Hospital</h3>

<p>Benz Circle<br>

Vijayawada<br>

Andhra Pradesh(A.P)

</p>

</div>

</div>

<div class="row pt-3">

<div class="col-lg-1 offset-1 col-md-2 col-sm-2">

<span style="font-size: 30px; color: coral"><i class="fas fa-phone-volume"></i></span>

</div>

<div class="col-lg-10 col-md-9 col-sm-9">

<h3 class="font-weight-light">Give Us a Ring</h3>

<p>Dr.Ranganath<br>

+91 8639204898<br>

Andhra Pradesh(A.P)

</p>

</div>

</div>

</div>

<div class="col-lg-6 col-md-12 col-sm-12">

<form method="post">

{% csrf\_token %}

<div class="form-row">

<div class="form-group col-lg-6 col-md-12 col-sm-12">

<label>Full Name</label>

<input type="text" name="name" class="form-control" placeholder="Full Name" required>

</div>

<div class="form-group col-lg-6 col-md-12 col-sm-12">

<label>Contact Number</label>

<input type="text" name="contact" class="form-control" placeholder="Contact Number" required>

</div>

<div class="form-group col-lg-6 col-md-12 col-sm-12">

<label>Email Id</label>

<input type="email" name="email" class="form-control" placeholder="Email ID" required>

</div>

<div class="form-group col-lg-6 col-md-12 col-sm-12">

<label>Subject</label>

<input type="text" name="subject" class="form-control" placeholder="Enter Subject" required>

</div>

<div class="form-group col-lg-12 col-md-12 col-sm-12">

<label>Message</label>

<textarea name="message" class="form-control" placeholder="Describe your Problem"></textarea>

</div>

<div class="form-group col-lg-6 col-md-12 col-sm-12">

<input type="submit" value="Send" class="form-control btn btn-primary">

</div>

<div class="form-group col-lg-6 col-md-12 col-sm-12">

<input type="reset" value="Clear" class="form-control btn btn-primary">

</div>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

</section>

{% endblock%}

## View\_appointment.html

{% extends 'admin\_nav.html' %}

{% load static %}

{% block content %}

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.6.3/css/all.css" integrity="sha384-UHRtZLI+pbxtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ8lWUE00s/" crossorigin="anonymous">

<br><br><br><br>

<div class="container card shadow m-5">

<div class="card-body">

<h5 class="p-2" style="border-bottom: 2px solid orange">View Appointment</h5>

<div class="container mt-5">

<h1 class="text-success text-center"></h1>

<table id="example" class="table table-striped table-bordered table-sm" style="width:80%">

<thead>

<tr>

<th>Doctor Name</th>

<th>Patient Name</th>

<th>Date</th>

<th>Time</th>

<th>Action</th>

</tr>

</thead>

<tbody>

{% for i in appointment %}

<tr>

<td>{{i.doctor}}</td>

<td>{{i.patient}}</td>

<td>{{i.date1}}</td>

<td>{{i.time1}}</td>

<td><a href="{% url 'delete\_appointment' i.id %}" onclick="return confirm('Are you ready to delete this record ?')" class="btn btn-danger">Delete</a></td>

</tr>

{% endfor %}

</tbody>

</table>

</div>

</div>

</div>

{% endblock content %}

## View\_doctor.html

{% extends 'admin\_nav.html' %}

{% load static %}

{% block content %}

<head>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.6.3/css/all.css" integrity="sha384-UHRtZLI+pbxtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ8lWUE00s/" crossorigin="anonymous">

</head>

<br><br><br><br>

<div class="container card shadow m-5">

<div class="card-body">

<h5 class="p-2" style="border-bottom: 2px solid orange">View Doctors</h5>

<div class="container mt-5">

<h1 class="text-success text-center"></h1>

<div class="table-responsive">

<table id="example" class="table table-striped table-bordered table-sm" style="width:80%">

<thead>

<tr>

<th>ID</th>

<th>Doctor Name</th>

<th>Contact</th>

<th>Specialization</th>

<th>Action</th>

</tr>

</thead>

<tbody>

{% for i in doc %}

<tr>

<td>{{i.id}}</td>

<td>{{i.name}}</td>

<td>{{i.mobile}}</td>

<td>{{i.special}}</td>

<td><a href="{% url 'edit\_doctor' i.id %}" class="btn btn-primary">Edit</a> |

<a href="{% url 'delete\_doctor' i.id %}" onclick="return confirm('Are you ready to delete this record ?')" class="btn btn-danger">Delete</a></td>

</tr>

{% endfor %}

</tbody>

</table>

</div>

</div>

</div>

</div>

{% endblock content %}

## View\_patient.html

## {% extends 'admin\_nav.html' %}

## {% load static %}

## {% block content %}

## <head>

## <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

## <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

## <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

## <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.6.3/css/all.css" integrity="sha384-UHRtZLI+pbxtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ8lWUE00s/" crossorigin="anonymous">

## </head>

## <br><br><br><br>

## <div class="container card shadow m-5">

## <div class="card-body">

## <h5 class="p-2" style="border-bottom: 2px solid orange">View Patient</h5>

## <div class="container mt-5">

## <h1 class="text-success text-center"></h1>

## <table id="example" class="table table-striped table-bordered" style="width:80%">

## <thead>

## <tr>

## <th>Patient ID</th>

## <th>Patient Name</th>

## <th>Gender</th>

## <th>Contact</th>

## <th>Address</th>

## <th>Action</th>

## </tr>

## </thead>

## <tbody>

## {% for i in pat %}

## <tr>

## <td>{{i.id}}</td>

## <td>{{i.name}}</td>

## <td>{{i.gender}}</td>

## <td><a href="{% url 'edit\_patient' i.id %}" class="btn btn-primary">Edit</a> |<a href="{% url 'delete\_patient' i.id %}" onclick="return confirm('Are you ready to delete this record ?')" class="btn btn-danger">Delete</a></td>

## </tr>

## {% endfor %}

## </tbody>

## </table>

## </div>

## </div></div>

## {% endblock content %}

## Django Main File

## admin.py

from django.contrib import admin

from .models import \*

admin.site.register(Doctor)

admin.site.register(Patient)

admin.site.register(Appointment)

**apps.py**

from django.apps import AppConfig

class HospitalsConfig(AppConfig):

default\_auto\_field = 'django.db.models.BigAutoField'

name = 'hospitals'

**models.py**

from django.db import models

class Doctor(models.Model):

name = models.CharField(max\_length=50)

mobile = models.IntegerField()

special = models.CharField(max\_length=50)

def \_\_str\_\_(self):

return self.name;

class Patient(models.Model):

name = models.CharField(max\_length=50)

gender = models.CharField(max\_length=10)

mobile = models.IntegerField(null=True)

address = models.CharField(max\_length=50)

def \_\_str\_\_(self):

return self.name;

class Appointment(models.Model):

doctor = models.ForeignKey(Doctor,on\_delete=models.CASCADE)

patient = models.ForeignKey(Patient, on\_delete=models.CASCADE)

date1 = models.DateField()

time1 = models.TimeField()

def \_\_str\_\_(self):

return self.doctorname+"--"+self.patient.name;

class Contact(models.Model):

name = models.CharField(max\_length=100, null=True)

contact = models.CharField(max\_length=15, null=True)

email = models.CharField(max\_length=50, null=True)

subject = models.CharField(max\_length=100, null=True)

message = models.CharField(max\_length=300, null=True)

msgdate = models.DateField(null=True)

isread = models.CharField(max\_length=10,null=True)

def \_\_str\_\_(self):

return self.id

from django.shortcuts import render,redirect

from django.contrib.auth.models import User

from django.contrib.auth import authenticate,logout,login

from .models import \*

from datetime import date

def About(request):

return render(request,'about.html')

def Index(request):

return render(request,'index.html')

def contact(request):

error = ""

if request.method == 'POST':

n = request.POST['name']

c = request.POST['contact']

e = request.POST['email']

s = request.POST['subject']

m = request.POST['message']

try:

Contact.objects.create(name=n, contact=c, email=e, subject=s, message=m, msgdate=date.today(), isread="no")

error = "no"

except:

error = "yes"

return render(request, 'contact.html', locals())

def adminlogin(request):

error = ""

if request.method == 'POST':

u = request.POST['uname']

p = request.POST['pwd']

user = authenticate(username=u, password=p)

try:

if user.is\_staff:

login(request, user)

error = "no"

else:

error = "yes"

except:

error = "yes"

return render(request,'login.html', locals())

def admin\_home(request):

if not request.user.is\_staff:

return redirect('login\_admin')

dc = Doctor.objects.all().count()

pc = Patient.objects.all().count()

ac = Appointment.objects.all().count()

d = {'dc': dc, 'pc': pc, 'ac': ac}

return render(request,'admin\_home.html', d)

def Logout(request):

logout(request)

return redirect('index')

def add\_doctor(request):

error=""

if not request.user.is\_staff:

return redirect('login')

if request.method=='POST':

n = request.POST['name']

m = request.POST['mobile']

sp = request.POST['special']

try:

Doctor.objects.create(name=n,mobile=m,special=sp)

error="no"

except:

error="yes"

return render(request,'add\_doctor.html', locals())

def view\_doctor(request):

if not request.user.is\_staff:

return redirect('login')

doc = Doctor.objects.all()

d = {'doc':doc}

return render(request,'view\_doctor.html', d)

def Delete\_Doctor(request,pid):

if not request.user.is\_staff:

return redirect('login')

doctor = Doctor.objects.get(id=pid)

doctor.delete()

return redirect('view\_doctor.html')

def edit\_doctor(request,pid):

error = ""

if not request.user.is\_authenticated:

return redirect('login')

user = request.user

doctor = Doctor.objects.get(id=pid)

if request.method == "POST":

n1 = request.POST['name']

m1 = request.POST['mobile']

s1 = request.POST['special']

doctor.name = n1

doctor.mobile = m1

doctor.special = s1

try:

doctor.save()

error = "no"

except:

error = "yes"

return render(request, 'edit\_doctor.html', locals())

def add\_patient(request):

error = ""

if not request.user.is\_staff:

return redirect('login')

if request.method == 'POST':

n = request.POST['name']

g = request.POST['gender']

m = request.POST['mobile']

a = request.POST['address']

try:

Patient.objects.create(name=n, gender=g, mobile=m, address=a)

error = "no"

except:

error = "yes"

return render(request,'add\_patient.html', locals())

def view\_patient(request):

if not request.user.is\_staff:

return redirect('login')

pat = Patient.objects.all()

d = {'pat':pat}

return render(request,'view\_patient.html', d)

def Delete\_Patient(request,pid):

if not request.user.is\_staff:

return redirect('login')

patient = Patient.objects.get(id=pid)

patient.delete()

return redirect('view\_patient.html')

def edit\_patient(request,pid):

error = ""

if not request.user.is\_authenticated:

return redirect('login')

user = request.user

patient = Patient.objects.get(id=pid)

if request.method == "POST":

n1 = request.POST['name']

m1 = request.POST['mobile']

g1 = request.POST['gender']

a1 = request.POST['address']

patient.name = n1

patient.mobile = m1

patient.gender = g1

patient.address = a1

try:

patient.save()

error = "no"

except:

error = "yes"

return render(request, 'edit\_patient.html', locals())

def add\_appointment(request):

error=""

if not request.user.is\_staff:

return redirect('login')

doctor1 = Doctor.objects.all()

patient1 = Patient.objects.all()

if request.method=='POST':

d = request.POST['doctor']

p = request.POST['patient']

d1 = request.POST['date']

t = request.POST['time']

doctor = Doctor.objects.filter(name=d).first()

patient = Patient.objects.filter(name=p).first()

try:

Appointment.objects.create(doctor=doctor, patient=patient, date1=d1, time1=t)

error="no"

except:

error="yes"

d = {'doctor':doctor1,'patient':patient1,'error':error}

return render(request,'add\_appointment.html', d)

def view\_appointment(request):

if not request.user.is\_staff:

return redirect('login')

appointment = Appointment.objects.all()

d = {'appointment':appointment}

return render(request,'view\_appointment.html', d)

def Delete\_Appointment(request,pid):

if not request.user.is\_staff:

return redirect('login')

appointment1 = Appointment.objects.get(id=pid)

appointment1.delete()

return redirect('view\_appointment.html')

def unread\_queries(request):

if not request.user.is\_authenticated:

return redirect('login')

contact = Contact.objects.filter(isread="no")

return render(request,'unread\_queries.html', locals())

def read\_queries(request):

if not request.user.is\_authenticated:

return redirect('login')

contact = Contact.objects.filter(isread="yes")

return render(request,'read\_queries.html', locals())

def view\_queries(request,pid):

if not request.user.is\_authenticated:

return redirect('login')

contact = Contact.objects.get(id=pid)

contact.isread = "yes"

contact.save()

return render(request,'view\_queries.html', locals())

**manage.py**

import os

import sys

def main():

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'HospitalManagementSystem.settings')

try:

from django.core.management import execute\_from\_command\_line

except ImportError as exc:

raise ImportError(

) from exc

execute\_from\_command\_line(sys.argv)

if \_\_name\_\_ == '\_\_main\_\_':

main()

**IMPLEMENTATION USING**

**FLASK**

**Flask Implementation**

## manage.py

import os

import sys

def main():

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'HospitalManagementSystem.settings')

try:

from django.core.management import execute\_from\_command\_line

except ImportError as exc:

raise ImportError(

) from exc

execute\_from\_command\_line(sys.argv)

if \_\_name\_\_ == '\_\_main\_\_':

main()

**apps.py**

from flask import Flask,send\_from\_directory,render\_template

from flask import send\_from\_directory

from flask\_restful import Resource, Api

from package.patient import Patients, Patient

from package.doctor import Doctors, Doctor

from package.appointment import Appointments, Appointment

from package.common import Common

from package.medication import Medication, Medications

from package.department import Departments, Department

from package.nurse import Nurse, Nurses

from package.room import Room, Rooms

from package.procedure import Procedure, Procedures

from package.prescribes import Prescribes, Prescribe

from package.undergoes import Undergoess, Undergoes

import json

import os

with open('config.json') as data\_file:

config = json.load(data\_file)

app = Flask(\_\_name\_\_, static\_url\_path='')

api = Api(app)

api.add\_resource(Patients, '/patient')

api.add\_resource(Patient, '/patient/<int:id>')

api.add\_resource(Doctors, '/doctor')

api.add\_resource(Doctor, '/doctor/<int:id>')

api.add\_resource(Appointments, '/appointment')

api.add\_resource(Appointment, '/appointment/<int:id>')

api.add\_resource(Common, '/common')

api.add\_resource(Medications, '/medication')

api.add\_resource(Medication, '/medication/<int:code>')

api.add\_resource(Departments, '/department')

api.add\_resource(Department, '/department/<int:department\_id>')

api.add\_resource(Nurses, '/nurse')

api.add\_resource(Nurse, '/nurse/<int:id>')

api.add\_resource(Rooms, '/room')

api.add\_resource(Room, '/room/<int:room\_no>')

api.add\_resource(Procedures, '/procedure')

api.add\_resource(Procedure, '/procedure/<int:code>')

api.add\_resource(Prescribes, '/prescribes')

api.add\_resource(Undergoess, '/undergoes')

# Routes

@app.route('/favicon.ico')

def favicon():

return send\_from\_directory(os.path.join(app.root\_path, 'static'),

'favicon.ico',mimetype='image/vnd.microsoft.icon')

@app.route('/')

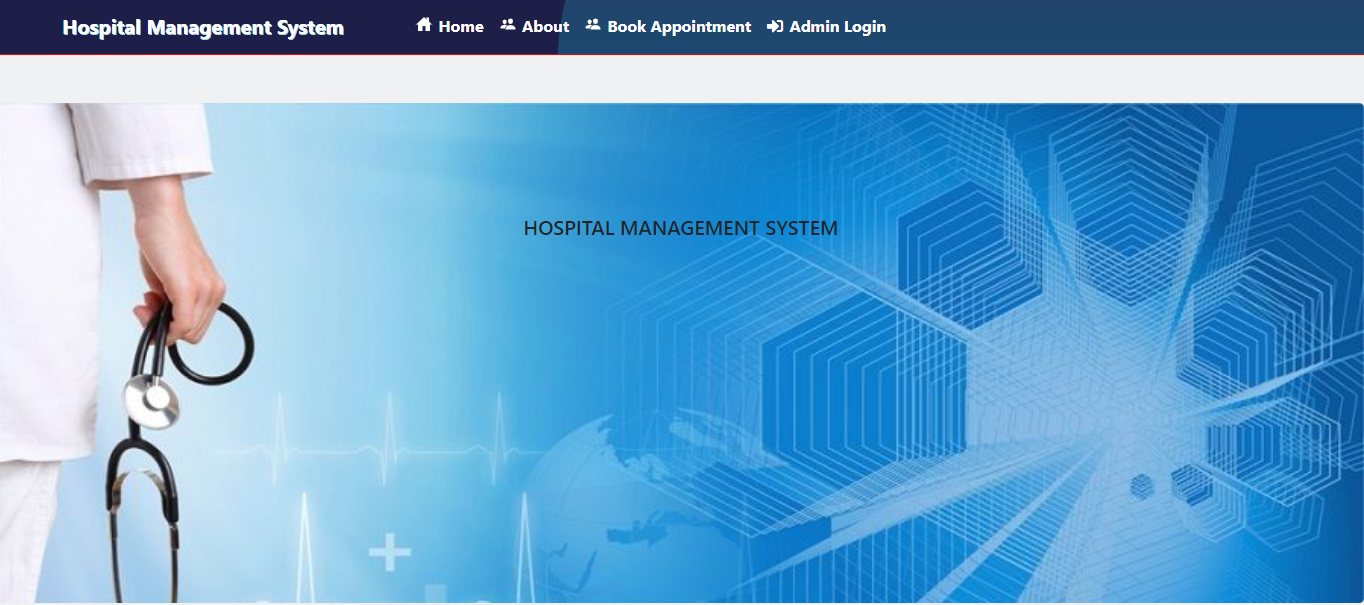
def index():

return app.send\_static\_file('index.html')

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True,host=config['host'],port=config['port'])

**OUTPUT SCREENSHOTS**

****

## Fig 7.1 Home Page

**Graphical user interface, application

Description automatically generated**

**Fig 7.2 Book Appointment Page**

**Graphical user interface, text, application, website

Description automatically generated**

## Fig 7.3 About Page

**Graphical user interface, application

Description automatically generated**

**Fig 7.4 Admin Login Page**

**Graphical user interface, website

Description automatically generated**

**Fig 7.5 Admin Dashboard**

**Graphical user interface, text, application, email

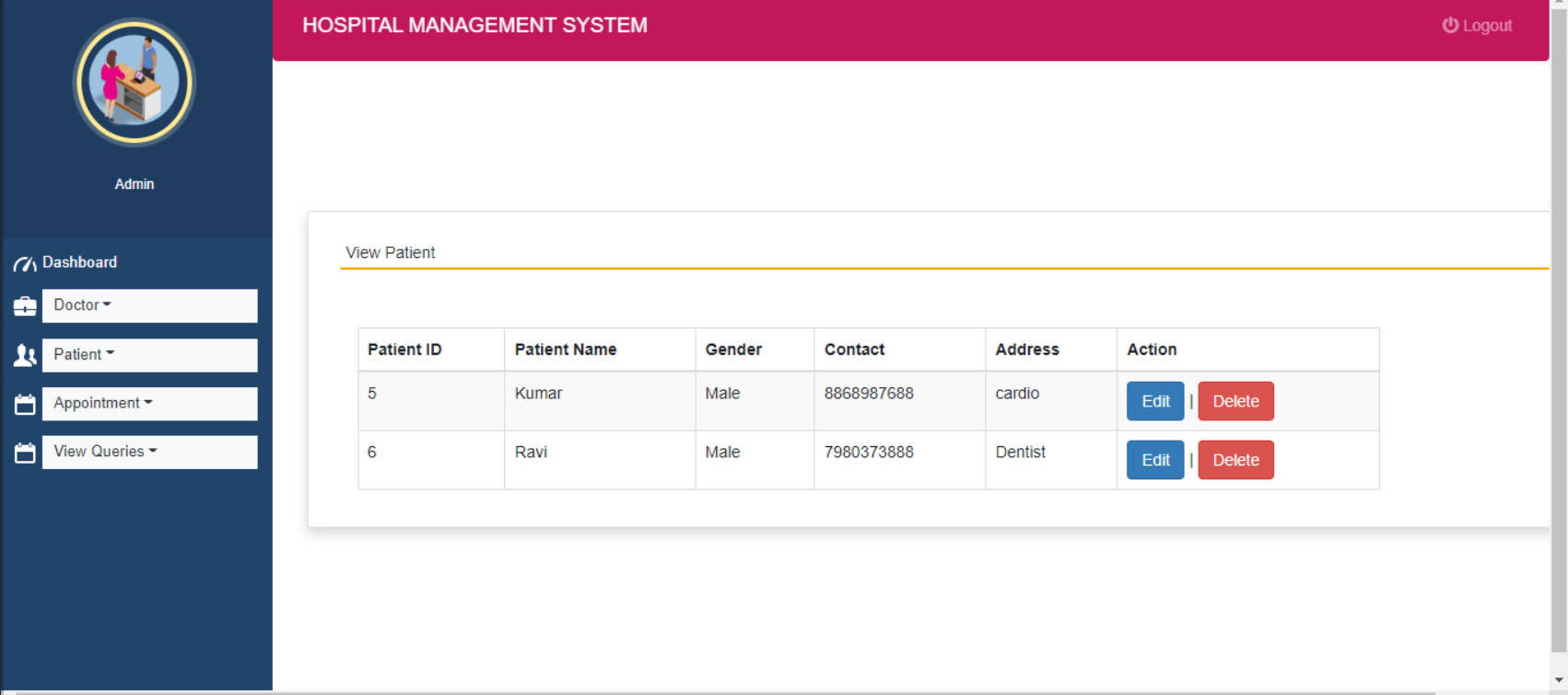
Description automatically generated**

**Fig 7.6 Adding Doctors Page**

**Graphical user interface, text, application, email

Description automatically generated**

**Fig 7.7 Adding Patient Details Page**

****

**Fig 7.8 Viewing Patient Page**

**Graphical user interface, application

Description automatically generated**

**Fig 7.9 Add Appointment Page**

**Graphical user interface, text, application, email

Description automatically generated**

**Fig 7.10 View Appointment Page**

**Table

Description automatically generated**

**Fig 7.11 Read Queries Page**

**CONCLUSION**

The goal of this research was to create and deploy a Django-based hospital management

system. The responders and intended end users were shown the produced project for

evaluation and assessment. The study's findings demonstrated that the created system

satisfies the demands and specifications of the respondents and end-users. Regarding acceptability, efficacy, quality, and productivity in automating hospital administration,

the respondents gave the system satisfactory ratings. The researchers concluded that the

system created is effective in automating hospital management. By delivering high-quality healthcare and other benefits, the system will do away with all human mistakes, efficiently manage hospital operations, and increase patient happiness. The system will enhance the operation of the hospital.