**EDOC: DOCTOR APPOINTMENT SYSTEM**

**A PROJECT REPORT**

# Submitted by

DIWAKAR MISHRA

ARYANSH GUPTA

*in partial fulfillment for the award of the degree of*

# BACHELOR OF TECHNOLOGY

*in*

## COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)



**SCHOOL OF COMPUTING SCIENCE ENGINEERING AND ARTIFICIAL INTELLIGENCE**

# VIT BHOPAL UNIVERSITY KOTHRIKALAN, SEHORE MADHYA PRADESH - 466114

DECEMBER 2024

## ABSTRACT

The eDoc Doctor Appointment System is a modern, web-based platform designed to streamline appointment scheduling and enhance healthcare accessibility. Developed using **HTML**, **CSS**, and **Node.js**, the system offers a secure login mechanism powered by **Google Cloud OAuth** for seamless authentication and user safety.

The platform features an intuitive navigation interface and an efficient doctorpatient management system that ensures users are connected to the appropriate healthcare professionals based on their medical needs. A built-in **real-time chatbot** provides instant assistance, guiding users through the platform and addressing their queries. The system integrates a robust database to store and manage user data, appointments, and doctor details securely.

A key component of the system is the **appointment box**, where users can view, book, or manage their appointments. By categorizing doctors based on specialties, the platform ensures targeted care, linking patients to specialists suitable for their conditions. This innovative approach makes the eDoc Doctor Appointment System an essential tool for improving healthcare delivery, offering patients an efficient, accessible, and user-friendly solution to manage their healthcare needs.

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Chapter no.** | **Title** | | **Page no.** |
| **1.** | **CHAPTER-1:**  **PROJECT DESCRIPTION AND OUTLINE**  1.1 Introduction  1.2 Motivation for the work  1.3 Problem Statement  1.4 Objective of the work | | **1** |
| **2.** | **DESIGN**  2.1  2.2  2.3 | **CHAPTER-2:**  **METHODOLOGY AND ITS NOVELTY**  Methodology  Goal  Software Architectural designs | 9 |
| **3.** | 3.1  3.2  3.3  3.4 | **CHAPTER-3:**  **USER INTERFACE DESIGN**  Flowchart  Login page  Appointment registration  Chatbot | 12 |
| 4. | **CHAPTER-4:**  **PROJECT OUTCOME AND APPLICABILITY**  4.1 Outline  4.2 Significant project outcomes  4.3 Project applicability on Real-world applications | | 16 |

|  |  |  |
| --- | --- | --- |
| 5. | **CHAPTER-5:**  **CONCLUSIONS AND RECOMMENDATION**  5.1 Outline  5.2 Limitation/Constraints of the System  5.3 Future Enhancements | 18 |
|  | *Reference* |  |

**CHAPTER-1:**

# PROJECT DESCRIPTION AND OUTLINE

**1.1 Introduction**

The Edoc Doctor Appointment System is a modern, web-based platform designed to simplify appointment scheduling and enhance healthcare accessibility. Developed using HTML, CSS, and Node.js, this system provides secure login through Google Cloud's OAuth initiative, ensuring privacy and protection. With a user-friendly interface, the platform enables efficient doctorpatient management. Key features include a real-time chatbot for assistance, a database to store patient and appointment data, and a scheduling feature that allows users to view and book appointments based on their illness. The system categorizes doctors according to their specialties, making it easy for patients to select the right medical professional for their needs.

**1.2 Motivation for the Work**

The healthcare sector has seen rapid technological advancements, yet appointment scheduling and management can still be cumbersome, especially in regions where access to healthcare is limited. This project is motivated by the need to streamline the appointment booking process, improve accessibility to doctors, and reduce the complexity faced by patients and healthcare providers. By creating an easy-to-use, secure, and efficient platform, the Edoc system aims to enhance the healthcare experience for all users**.**

**1.3 Problem Statement**

Currently, scheduling medical appointments often involves long waiting times, complicated systems, and a lack of clear communication between patients and

doctors. This is especially problematic for patients with urgent needs or those living in underserved areas. Furthermore, managing patient information securely and efficiently remains a challenge. The absence of a user-friendly interface for appointment booking can lead to confusion and frustration for both patients and healthcare providers. There is a clear need for a digital solution that addresses these challenges**.**

**1.4 Objective of the Work**

The primary objective of the Edoc Doctor Appointment System is to provide a secure, user-friendly platform for booking doctor appointments. Specific goals include:

1. Streamlining Appointment Scheduling: Allow users to easily book, reschedule, or cancel appointments with doctors based on their illness and preference.
2. Improving Healthcare Accessibility: By offering a web-based platform, the system ensures that patients from various geographic locations can access healthcare services without significant barriers.
3. Ensuring Secure Communication: Utilize Google Cloud's OAuth for secure login and data protection.
4. Enhancing Doctor-Patient Management: Provide features that allow doctors and patients to manage their appointments efficiently, including a real-time chatbot for assistance.
5. Simplifying Patient and Doctor Information Management: Implement a database to store relevant patient and doctor information securely, ensuring efficient management and retrieval of data.

**CHAPTER-2:**

**DESIGN METHODOLOGY AND ITS NOVELTY**

. **2.1 Methodology**

The Edoc Doctor Appointment System follows an **Agile Software Development** methodology to ensure iterative development and flexibility for continuous improvement. The development process can be broken down into several stages:

1. **Requirement Gathering**: Identifying the needs of users (doctors and patients) and understanding the healthcare environment.
2. **Design**: Creating wireframes and flowcharts to design an intuitive user interface for both doctors and patients. Ensuring the system integrates securely with Google Cloud for login authentication.
3. **Development**: Using modern web technologies such as HTML, CSS, and Node.js to build a secure, responsive, and user-friendly platform.
4. **Testing**: Conducting unit and integration tests to ensure the system works seamlessly and securely.
5. **Deployment**: Deploying the system to a secure cloud environment, ensuring high availability and performance.
6. **Maintenance & Updates**: Continuously monitoring the system for bugs, security threats, and adding new features based on user feedback.

**2.2 Goal**

The main goal of the **Edoc Doctor Appointment System** is to:

* **Simplify the Appointment Scheduling Process**: Make it easy for patients to find and book appointments with doctors based on illness type.
* **Improve Healthcare Accessibility**: Provide patients with easy access to healthcare providers and reduce wait times through efficient scheduling.
* **Enhance Doctor-Patient Management**: Allow doctors to manage their schedules and patient information in a streamlined manner.
* **Enable Real-Time Assistance**: Provide a chatbot to assist patients in booking appointments, answering basic healthcare questions, and navigating the platform.
* **Ensure Data Security**: Use secure login via Google Cloud OAuth and encrypt patient data to maintain privacy and confidentiality.

**2.3 Software Architectural Design**

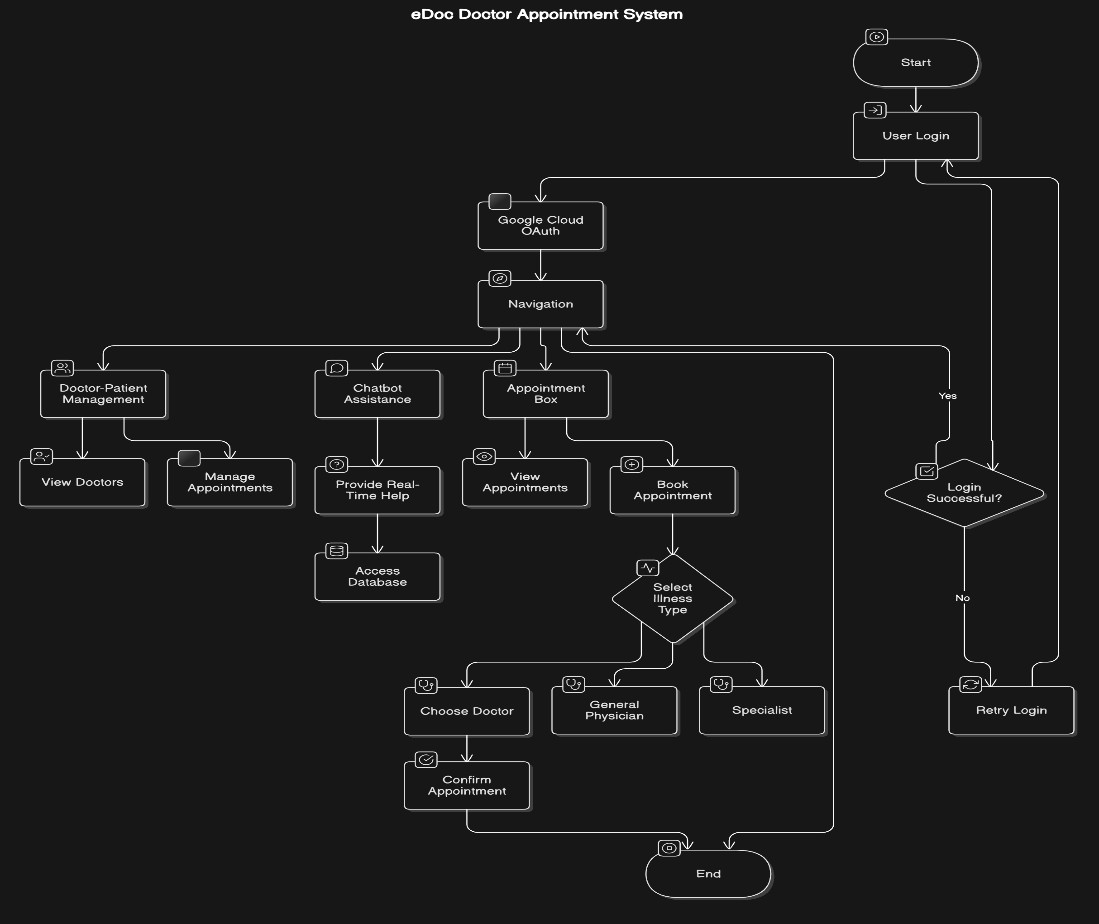
The system is built with a **multi-tier architecture**, ensuring a clear separation of concerns and scalability:

1. **Presentation Layer (Frontend)**:
   * Built using **HTML** and **CSS** to provide a user-friendly interface.
   * **ReactJS** or **Vue.js** could be added for a dynamic, responsive experience. oDisplays the appointment calendar, booking options, and the chatbot interface.
2. **Application Layer (Backend)**:
   * Built using **Node.js** with Express for handling HTTP requests and managing sessions. oRESTful API design to interact between the frontend and the database.
   * Integration with **Google Cloud OAuth** for secure login and authentication.
3. **Database Layer**:
   * A **Relational Database** (e.g., MySQL, PostgreSQL) stores patient and doctor data, appointment schedules, and medical histories.
   * The system should use **SQL queries** to fetch, insert, update, or delete data based on user actions.
   * **MongoDB** could be considered for more flexibility with unstructured data.
4. **Real-Time Layer**:
   * A **chatbot** powered by AI/ML models that can help patients with appointment bookings, FAQs, and other queries.
   * **Socket.io** or similar technology can be used to facilitate realtime communication.
5. **Security Layer**:
   * + User authentication via **Google Cloud OAuth** for secure login. o**SSL/TLS encryption** to protect data during transmission.
     + **Role-based access control (RBAC)** to ensure that only authorized users (doctors, admin, patients) can access sensitive data.
6. **Deployment**:
   * + Hosted on a **cloud platform** (e.g., AWS, Google Cloud) to ensure scalability and availability.
     + CI/CD pipelines for continuous deployment and updates.

**CHAPTER 3 :**

# USER INTERFACE DESIGN

## 3.1 Flowchart



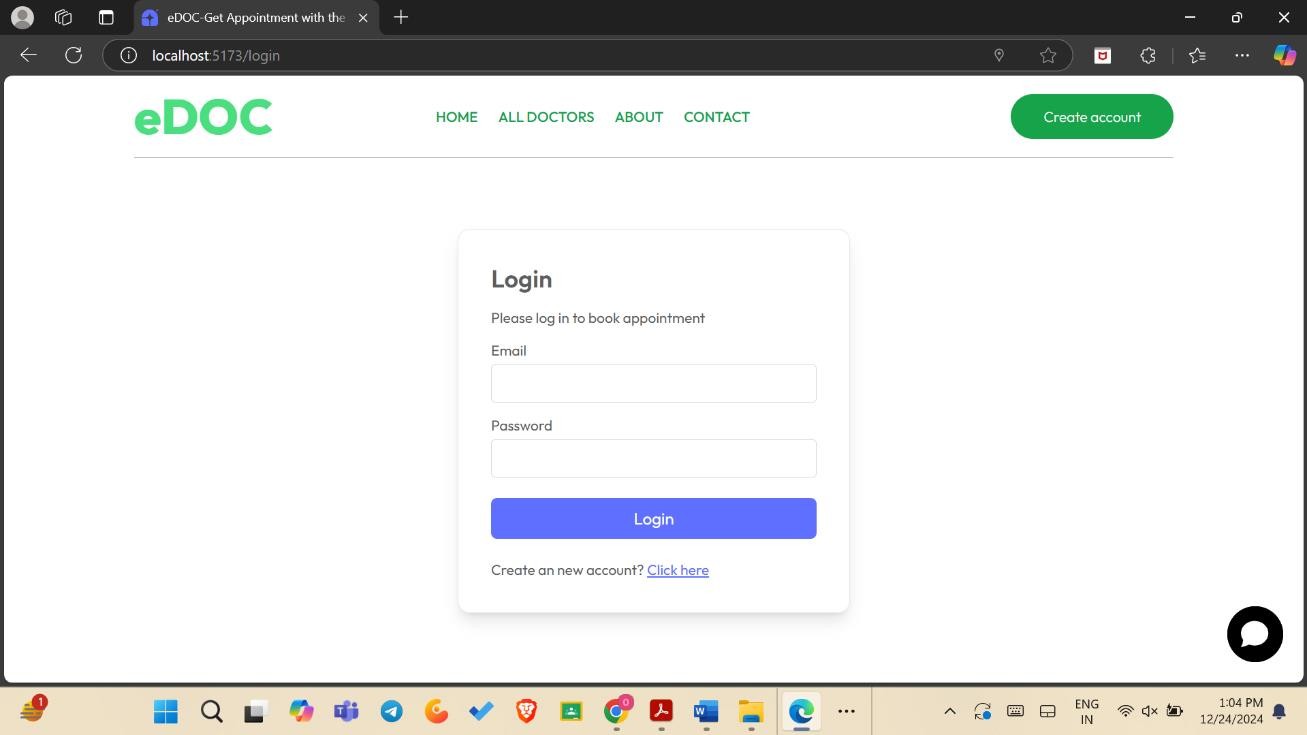
A flowchart could represent the process of interacting with the system, showing the steps involved in booking an appointment, using the chatbot, and navigating through the login process. Here’s a rough outline for your flowchart:

1. User Opens System o Options: Login or Register o Option: Use Chatbot for assistance
2. Login Process o Enter Credentials

o Google Cloud OAuth Authentication o Successful Login → Main Dashboard

1. Dashboard o View Available Doctors o Book Appointment
2. Appointment Booking o Select Doctor based on Illness Type o Choose Appointment Time o Confirmation/Cancelation Option
3. Chatbot Assistance o Provides FAQs or Guidance o Directs to Appointment Booking if needed
4. Database o Stores Appointment and User Data o Tracks Appointments & Communication

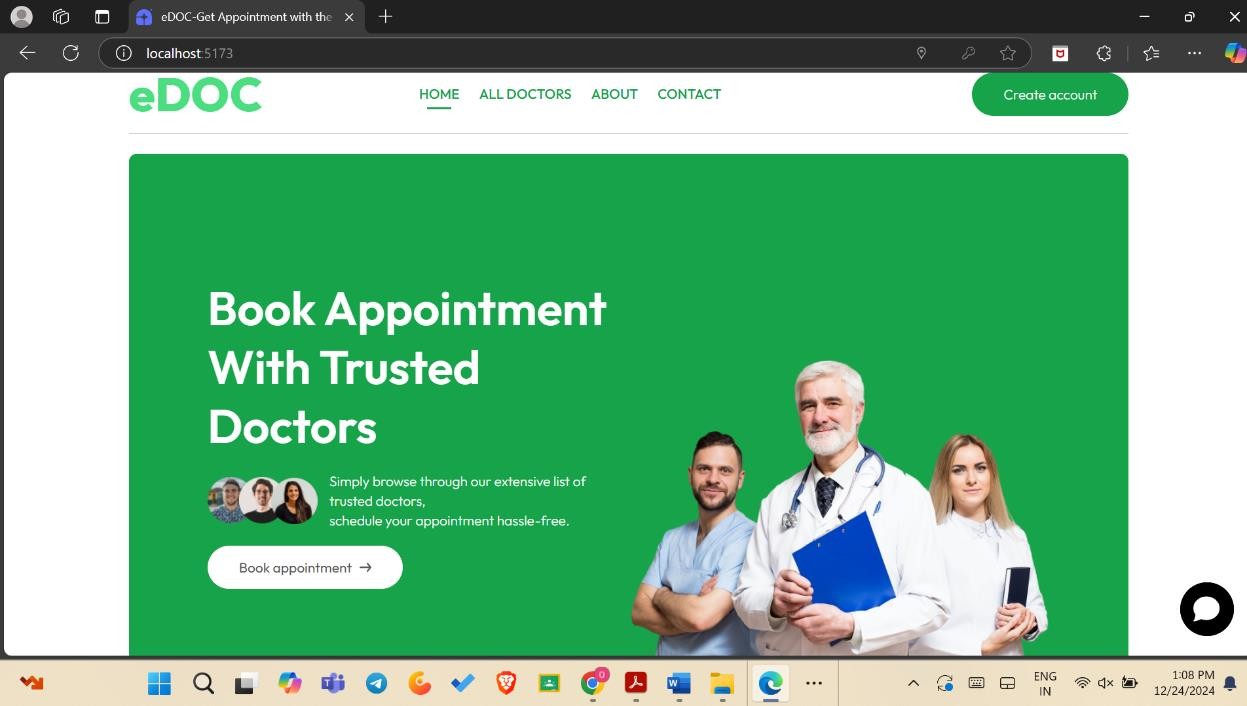
## 3.2 Login Page



he login page should be straightforward and secure, integrating Google Cloud OAuth for authentication. It could look like this:

* Fields:
  + Email/Username o Password
  + Login via Google (OAuth button)
* Actions:
  + Forgot Password o Register a New Account

## 3.3 Appointment Registration



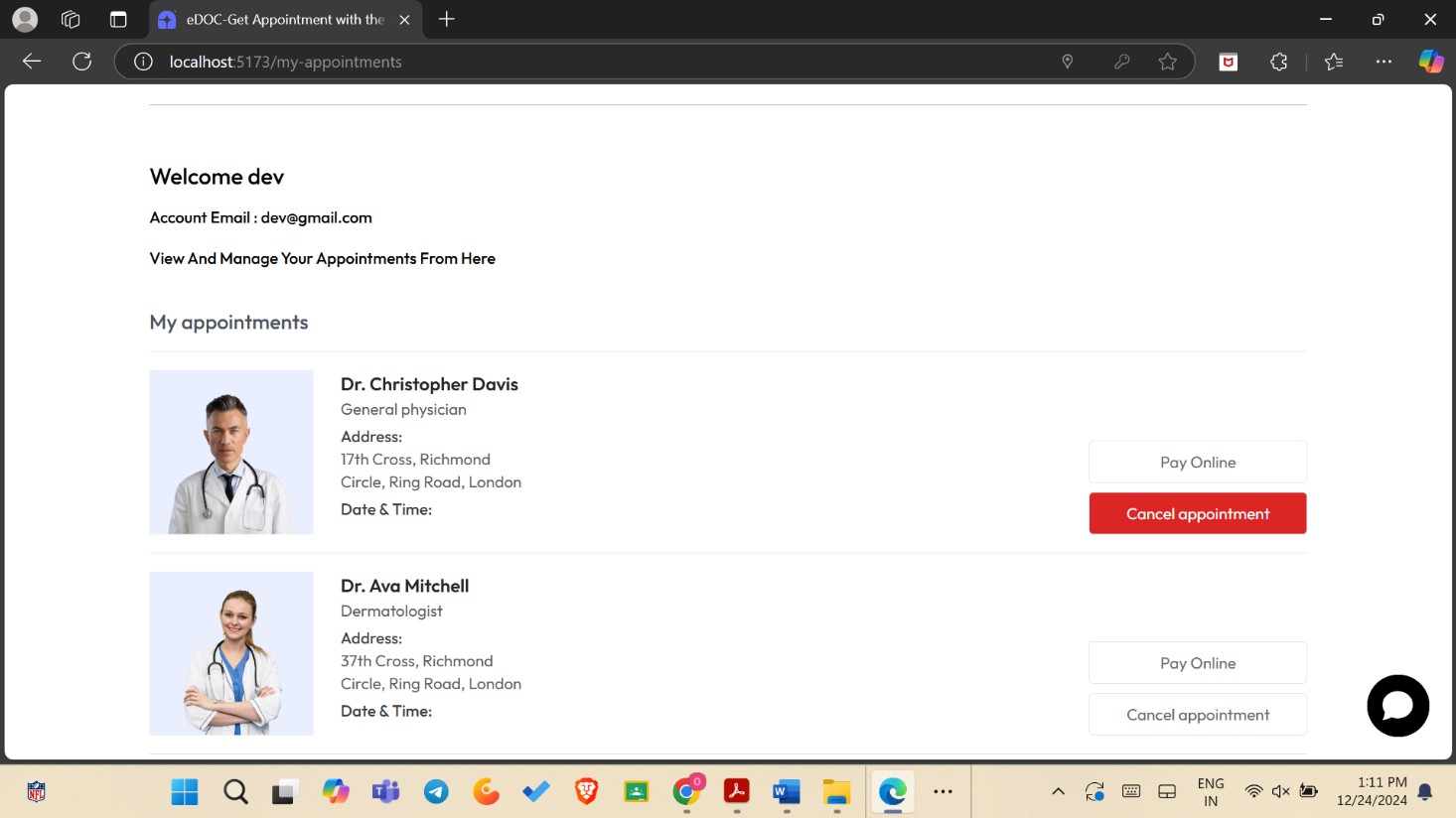
Users should be able to book and manage their appointments easily. The registration page could include:

* **Step 1: Select Illness Type** o A dropdown or search box with common medical conditions (e.g., fever, headache, etc.) o Based on illness type, the system recommends doctors.
* **Step 2: Select Doctor** o List of doctors who specialize in the selected illness.
* **Step 3: Choose Appointment Time**

o Available time slots for the selected doctor.

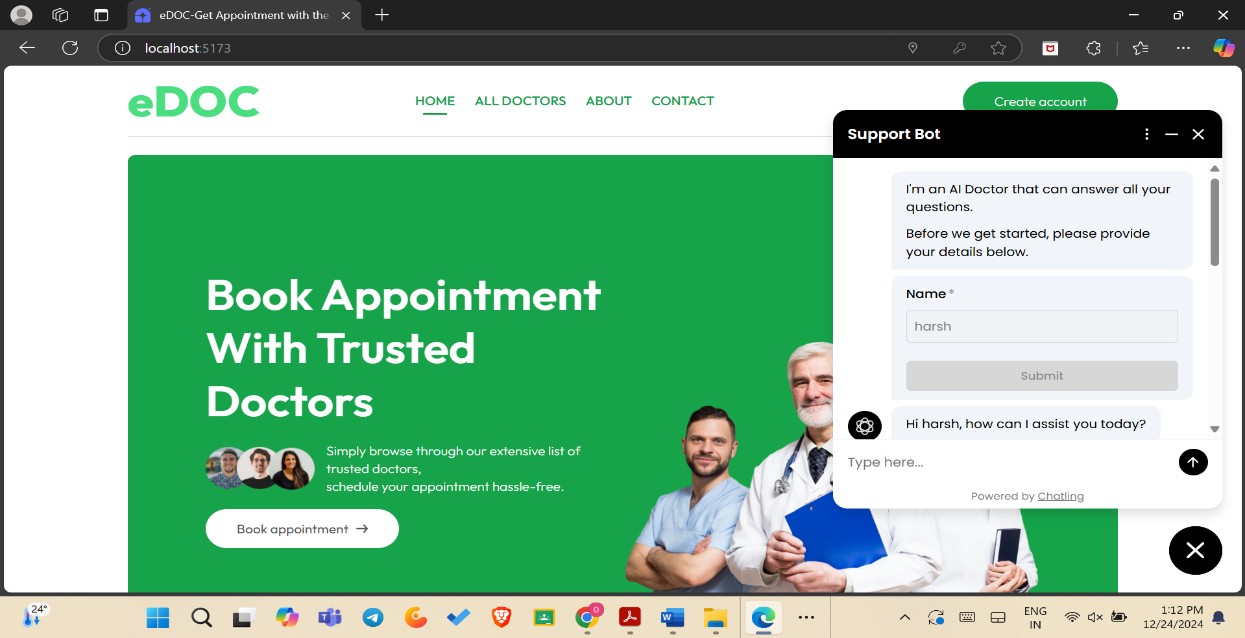
* **Step 4: Confirmation** o Review details and confirm the appointment.

.



.

## 3.4 Chatbot



The chatbot could be designed to assist with common tasks such as:

* **Booking Help:** Assist with choosing doctors or scheduling appointments.
* **FAQs:** Answer questions about the system, doctors, or medical procedures.
* **Appointment Reminders:** Send reminders about upcoming appointments.
* **General Assistance:** Answer queries like business hours, services, etc.

Would you like help fleshing out any of these sections or adding more details to the system?

**CHAPTER-4:**

# PROJECT OUTCOME AND APPLICABILITY

**4.1 Outline**

This section will typically provide a structured overview of the project, including the key features, design considerations, and technologies used. Here's a suggested outline based on your description:

**Introduction**

Overview of the system (modern web-based platform for appointment scheduling and healthcare accessibility)

Key objectives (simplifying appointment scheduling, improving patient-doctor communication, etc.)

**Technology Stack**

**Frontend**: HTML, CSS (Design & Layout)

**Backend**: Node.js (for server-side functionality)

**Authentication**: Google Cloud OAuth for secure login

**Database**: For storing appointment and patient data

**Real-time Assistance**: Chatbot integrated into the platform

**Appointment Management**: Appointment scheduling, status updates, and doctor assignment based on illness type

**Key Features**

Secure login via Google Cloud OAuth

Efficient navigation and user interface

Doctor-patient management (appointment booking, display, cancellation)

Chatbot for real-time assistance and FAQ handling

Database for storing patient, doctor, and appointment data

Appointment box to display booked appointments

**Challenges and Solutions**

Issues faced during the project and how they were solved (e.g., security, scalability, real-time functionality, etc.)

**Conclusion**

Summary of the project

Potential for future enhancements or updates

**4.2 Significant Project Outcomes**

In this section, you would summarize the key results or achievements from completing the project. Possible outcomes might include:

**Enhanced Patient Experience**: The system offers an intuitive, accessible interface for patients to manage appointments, reducing wait times and improving overall satisfaction.

**Efficient Doctor-Patient Management**: Doctors can easily manage their schedules, helping them stay organized and efficient.

**Increased Healthcare Access**: Patients from various locations can access healthcare providers without the need for in-person visits, improving healthcare accessibility. **Secure and Scalable**: Google Cloud OAuth ensures secure login and protects patient information, while Node.js offers a scalable solution for handling increasing users and data.

**4.3 Project Applicability in Real-World Applications**

This section would describe how the system can be applied to real-world healthcare scenarios. Some points could include:

**Telemedicine**: The platform could be used by telemedicine providers to schedule virtual consultations with doctors, enabling remote healthcare access.

**Hospital and Clinic Use**: Hospitals or private clinics could use this platform to manage patient appointments, reducing administrative burden and improving patient care.

**Patient Monitoring**: The system could integrate with health monitoring tools, allowing doctors to track patient health over time and adjust treatment plans accordingly.

**Health Insurance Integration**: The system could be linked with health insurance providers for automatic claim processing related to appointments or treatments.

**CHAPTER 5:**

# CONCLUSION AND RECOMMENDATION

**5.1 Outline of the System Overview:**

* The system is designed to simplify appointment scheduling for patients and improve healthcare accessibility.
* Built using **HTML**, **CSS**, and **Node.js**, the platform provides:
  + **Secure login** via Google Cloud OAuth, ensuring user privacy and data security.
  + **Efficient navigation** for both patients and doctors. o**Doctor-patient management** tools, allowing for the efficient handling of patient information, appointment scheduling, and medical records.
  + A **chatbot** feature that provides **real-time assistance** and can answer patient queries or help navigate the system. oA **database** for storing patient data, doctor details, and appointment schedules. o**Appointment box** that displays upcoming appointments, where patients can book new appointments based on their medical conditions.

**Main Features:**

1. **Secure Authentication:** Users can log in using their Google account.
2. **Doctor Listing and Specializations:** Patients can browse available doctors by illness or medical specialty.
3. **Appointment Booking:** Patients can view available slots and book appointments.
4. **Real-time Assistance:** A chatbot provides instant support and guidance.
5. **Appointment Box:** Displays the current and future appointments booked by the patient.

**5.2 Limitations/Constraints of the System** 1. **Internet Dependency:**

oAs a web-based system, it requires a stable internet connection to function properly, making it inaccessible in areas with unreliable or no internet.

1. **Limited Doctor Availability:**

oThe system may face issues when there is a limited number of doctors available in the platform for certain medical specialties, causing delays in appointments.

1. **Security Concerns:**

oWhile Google Cloud OAuth ensures secure login, the system may still be vulnerable to cyber threats or breaches, particularly if not regularly updated and maintained.

1. **Platform Compatibility:**

oThe system might not be fully compatible with older browsers or devices, potentially limiting access for some users.

1. **Data Privacy:**

oAs it stores sensitive patient data, there are risks related to privacy breaches or unauthorized access, even though it uses a secure login.

1. **Limited AI Capabilities:**

oThe chatbot feature might not fully understand or handle complex queries and medical issues, making it less effective in certain situations.

**5.3 Future Enhancements**

1. **Mobile App Development:**
   * Expand the system to include a mobile application, improving accessibility for users who prefer using smartphones over web platforms.
2. **Integration with Health Monitoring Devices:**
   * Allow integration with health tracking devices (e.g., smartwatches, blood pressure monitors) to collect real-time patient health data for doctors to review during consultations.
3. **Enhanced AI Chatbot:**
   * Improve the chatbot’s capabilities using advanced natural language processing (NLP) to offer better responses, including medical advice or symptom checking, under appropriate circumstances.
4. **Appointment Reminders:**
   * Add automatic SMS or email reminders for upcoming appointments to reduce no-shows and missed visits.
5. **Multi-language Support:**
   * Implement multi-language options to cater to a broader range of users, ensuring non-English speaking patients can also benefit from the platform.
6. **Telemedicine Features:**
   * Integrate video consultation features, enabling virtual appointments for patients who cannot visit doctors physically.
7. **Doctor and Patient Ratings/Reviews:**
   * Allow patients to leave ratings or reviews for their doctors, which could

help others make informed choices.

1. **Payment Gateway Integration:**
   * Add the ability for patients to pay for appointments or medical services directly through the system, integrating with popular payment methods like credit cards or digital wallets.

## RELATED WORK INVESTIGATION

The Edoc Doctor Appointment System draws inspiration from several existing platforms that aim to streamline healthcare access and scheduling. Below are some related works in the field:

1. **ZocDoc**: A well-known online medical care appointment booking platform, ZocDoc connects patients with healthcare providers. It allows patients to search for doctors based on specialty, insurance, and location, and book appointments online. Its feature set includes doctor reviews, appointment reminders, and secure communication between patients and doctors.
2. **Practo**: Practo offers similar services, with the addition of a feature for online consultations. Patients can search for healthcare providers, book appointments, and access health records. The platform also integrates telemedicine services and diagnostic testing bookings, making it a comprehensive healthcare platform.
3. **HealthTap**: This platform allows users to book appointments with doctors, access virtual health consultations, and get advice from professionals via a mobile app. It also uses artificial intelligence to provide symptom analysis and guidance, which aligns with the AI-powered chatbot feature in the Edoc system.
4. **Doctor on Demand**: This service offers virtual doctor visits, with the option to schedule consultations via mobile or web. It includes features for mental health support, general healthcare consultations, and the ability to prescribe medications through telemedicine.

**Key Features for Comparison:**

* **Appointment Scheduling**: All platforms, including Edoc, offer a booking system that simplifies appointment scheduling, providing convenience and reducing administrative burden.
* **Doctor-Patient Communication**: Integration of real-time communication (e.g., chat or video calls) is common across platforms, though Edoc emphasizes chatbot support for instant help.
* **Healthcare Customization**: Platforms like Edoc that tailor doctor recommendations based on illness or symptoms aim to offer a personalized experience, similar to the search filters seen in ZocDoc and Practo.
* **Security & Privacy**: OAuth login and cloud security practices adopted by Edoc are in line with industry standards for data protection, seen across major health platforms like Practo and ZocDoc.

.