جامعة ٦ أكتوبر
October 6 University

علوم الحاسب ونظم المعلومات
Info Sys & Comp Science

**Graduation Project**

# AI-Enhanced Dental Hospital Website Project


**Prepared by:**

| | |
|---|---|
| Ahmed Ashraf Ali Qasem | CS |
| Ibrahim Soliman Abdalaziz | CS |
| Ahmed Hassan Mahmoud Mostafa | CS |
| Moustafa Abdelgawad Elsayed | CS |
| Ahmed Abdelhamid Aboelmakarem | CS |
| Youssef Mourad Roshdi | CS |
| Mohamed Abdo Mohamed | CS |
| Abdelrahman Basiony Soliman | IS |

**Supervised by**

**Dr. Marwa Hussien**

**Co-Supervised by**

**Eng. Sohaila Sameeh**

**Eng. Abdelkareem Essam**


**2023-2024**

# Acknowledgment

# Acknowledgment

Both scientifically and practically, we have put a lot of effort into our project. All of this wouldn't have been possible without Allah, Doctors, Assistants, and team members.

We all would like to extend our gratitude to the dean **Prof. Dr. Nabila Mohamed Hassan**.

We all have special thanks to **Dr. Marwa Hussien** (The General supervisor) for his efforts and his guidance and constant supervision as well as for providing necessary information regarding the project and for his support in completing the project.

We all have special thanks to **Eng. Sohaila Sameeh** for her continuous support with us and giving us comments and her kind cooperation and encouragement, which helped us in completing this project.

We all have special thanks to **Eng. Abdelkareem Essam** for her continuous support with us and giving us comments and her kind cooperation and encouragement, which helped us in completing this project.

# Table of Contents

# Table of Contents


- Abstract ..... vi
- List of Abbreviation ..... vii:viii
- List of Figures ..... ix:xi
- Chapter One "Introduction" .....

1.1 Introduction of the project ..... 1

1.2 The Problem Definition ..... 2

1.3 Keys Features ..... 3

1.4 Project Objectives ..... 3

1.5 Stakeholders ..... 4

1.6 Problems & Solutions ..... 5

1.7 Tools ..... 6

1. 1.7.1 Front-End Development: ..... 6

1. HTML: ..... 6
2. CSS: ..... 6
3. JavaScript: ..... 6
4. React JS: ..... 6

2. 1.7.2 Back-End Development: ..... 6

1. PHP: ..... 6
2. Larverl: ..... 6

3. 1.7.3 Database: ..... 6

1. MYSQL: ..... 6

4. 1.7.4 UI/UX Design: ..... 6

1. Adobe XD: ..... 6

- Chapter Two"Literature Review" ..... 7

2.1 Introduction ..... 8

1. Vezeeta: ..... 8:10
2. Clinido: ..... 10:11
3. Ridentdental: ..... 12:13
4. Techno Clinics: ..... 14:15
5. Dental Clinic: ..... 15:17

2.3 System Components ..... 18:19

2.4 The scalability and performance ..... 20:21

- Chapter Three “Design and Methodology ” ..... **22**
  - 3.1 Requirement and Analysis ..... 23:25
  - 3.2 Functional Requirements ..... 26
  - 3.3 User Requirements ..... 27
  - 3.4 System Model ..... 28:45
- Chapter Four “Implementation ” ..... 46
  - 5. Front Design: ..... 47:61
  - 6. Code: ..... 62:75
  - 7. AI Implementation: ..... 76:89
  - 2.5 Conclusion ..... 90
  - References ..... 91:93

# Abstract

The aim of this graduation project is to develop a comprehensive web application for dental healthcare, including an artificial intelligence model to assist patient in Initial diagnosis.

The application will be designed with a user-friendly interface to ensure ease of use for both dentists and patients .

The artificial Intelligence model will be trained on a comprehensive dataset of dental images and patient records to identify potential dental Issues.

This will enable dentists to make more informed decisions during initial consultations, leading to improved patient care.

The application will also include features for managing patient records and scheduling appointments. This will help dentists streamline their practices and provide more efficient care .

# List of Abbreviation

# List of Abbreviations

- **AI: Artificial Intelligence**
- **UI: User Interface.**
- **UX: User Experience.**
- **JS: JavaScript.**
- **DFD: Data Flow Diagram.**
- **ERD: Entity Relationship Diagram.**
- **HTML: Hypertext Markup Language.**
- **CSS: Cascading Style Sheets.**
- **MySQL: My Structured Query Language.**
- **ER: Entity Relationship.**

# List of Figures

# List of Figures


- **Fig. 2.1** Vezeeta ……8
- **Fig. 2.2** Vezeeta ……9
- **Fig. 2.3** Vezeeta ……9
- **Fig. 2.4** Vezeeta ……9
- **Fig. 2.5** Clinido …… 10
- **Fig. 2.6** Clinido …… 11
- **Fig. 2.7** Clinido ……11
- **Fig. 2.8** Rident ……12
- **Fig. 2.9** Rident …… 12
- **Fig. 2.10** Rident …… 13
- **Fig. 2.11** Techno…… ……14
- **Fig. 2.12** Techno …… 14
- **Fig. 2.13** Ai-Enhanced Eental Hospital Website …… 15
- **Fig. 2.14** Ai-Enhanced Eental Hospital Website…… 16
- **Fig. 2.15** Ai-Enhanced Eental Hospital Website…… 16
- **Fig. 3.1** Data flow diagram (Context diagram) …… 31
- **Fig. 3.2** Data flow diagram (level 1) …… ……33
- **Fig. 3.3** Use-case diagram …… ……34
- **Fig. 3.4** Sequence Diagram ……37
- **Fig. 3.5** Class diagram (Structural Perspective)…… ……39
- **Fig. 3.6** Activity diagram (Behavioral Perspective) …… … …41
- **Fig. 3.7** Entity relationship diagram (ERD)…… …… ……43
- **Fig. 3.8** Scheme (System Design) …… 45
- **Fig. 4.1** Home Page Design …… 47
- **Fig. 4.2** Login Design ……47
- **Fig. 4.3** Register Patient Design …… 48
- **Fig. 4.4** Services Design ……49
- **Fig. 4.5** Doctors Design ……50
- **Fig. 4.6** Articles Design …… 50
- **Fig. 4.7** Patient Appointment Scheduling Design ……51

- **Fig. 4.8** Patient Profile Design …… 52
- **Fig. 4.9** Chat Design ……53
- **Fig. 4.10** AI Model Design …… 54
- **Fig. 4.11** Add Article Design …… 55
- **Fig. 4.12** Add Doctor Design ……56
- **Fig. 4.13** Add Services Design …….…… ……57
- **Fig. 4.14** Delete Doctor Design ……..……58
- **Fig. 4.15** Reservations Admin Design …… 59
- **Fig. 4.16** Reservations Doctor Design …… 60
- **Fig. 4.17** Search for patient files by national number Design……61

# Chapter One
# “Introduction”

## 1.1 Introduction of the project

The dental healthcare industry is continuously evolving, with advancements in technology and treatment methodologies shaping the future of patient care. In this context, the development of a comprehensive full-stack web application for dental healthcare, integrated with an AI- powered diagnostic system, represents a significant step towards enhancing the quality and efficiency of dental services. This graduation project aims to address the challenges faced by Dental professionals in the realm of patient management, record keeping, and initial diagnosis by providing a centralized and AI-driven solution. The proposed full-stack web application will serve as a unified platform for dental practitioners, enabling them to seamlessly manage patient information, access comprehensive dental records, and leverage AI- generated diagnostic insights to make informed treatment decisions. This innovative approach seeks to revolutionize the dental care landscape, fostering improved patient outcomes, streamlined workflows, and enhanced collaboration among dental professionals.

## 1.2 The problem Definition

The primary concern addressed by this graduation project is the inefficient use of time caused by the absence of advanced appointment reservations, leading to prolonged waiting periods. Moreover, there lacks a centralized database or system for the doctor to access a patient's historical data and previous diagnosis status. Furthermore, before the examination, the patient remains unaware of any preliminary diagnosis. Hence, the envisioned solution endeavors to resolve these issues through the creation of a secure, efficient, and user-friendly Web-based platform. This platform enables advance scheduling, provides precise appointment details for patients, and streamlines the doctor's access to patient information and diagnoses. A crucial remedy involves employing artificial intelligence to offer an initial diagnosis of the illness.

## 1.3 Keys Features

1. Contact information
2. Online Appointments.
3. AI diagnosis
4. Dentistry articles
5. Chat with doctor
6. Patient record
7. Save patient's time
8. Appointment scheduling

## 1.4 Project Objectives

The primary objective of this project is to design and develop a user-friendly, interactive for a dental website.

Enhancing communication with doctors : Helping patients communicate with the doctor via chat to answer questions and help them.

Improved Patient Experience : Enhance the overall experience of patients interacting with the dental website through user-friendly interfaces and personalized features.

Customize AI: Use AI for dental screening by uploading a photo of the condition and diagnosing it.

Delivery of educational content: Develop a platform to provide articles to patients, including information about oral health, treatment options, and preventive care.

Designing a system that provides most of the patients' data and medical history, allowing doctors to view the medical history and treatment reports at any time they want electronically.

## 1.5 Stakeholders

- **The stakeholders of project**

  1.System admin

  2.Doctor

  3.user

- **System admin**

  1. Create Account Doctors
  2. delete Account Doctors
  3. Check Bookings
  4. System Administration

- **doctor**

  1. View Patient Records.
  2. View Bookings.
  3. Upload diagnostic file.

- **user**

  1. Book appointment
  2. Read blogs
  3. Cancel appointment

## 1.6 Problems & Solutions

### What are the problems that the site was able to solve?

1- The doctor doesn't know the patient's diagnosis last time

2- Waiting problem

3- Difficulty examining images of the mouth

### Solve these problems:

1- It is available for the doctor to easily upload the case diagnosis file, so he can obtain it at any time because this file will be present in the patient's account.

2- The problem of waiting was solved by facilitating reservations for the patient and choosing the appropriate and easily available time for him. This helped save time and effort for both the doctor and the patient.

3- These images are carefully examined by artificial intelligence and the disease is identified with high quality.

## 1.7 Tools

### 1.6.1 Front-End Development:

**1- HTML (Hypertext Markup Language)**: Used for structuring web content and creating the basic layout of a webpage.

**2- CSS (Cascading Style Sheets)**: Responsible for styling and formatting the visual appearance of web elements.

**3- JavaScript (JS)**: A versatile scripting language that enables interactivity, animations, and dynamic behavior on web pages.

**4- ReactJS**: A popular JavaScript library for building interactive user interfaces (UIs) and single-page applications (SPAs).

### 1.6.2 Back-End Development:

1. **PHP**: A server-side scripting language used for processing data, handling forms, and managing server interactions.

2. **Laravel**: A PHP framework that simplifies back-end development by providing tools for routing, database access, and more.

### 1.6.3 Database:

1. **MySQL**: A widely used relational database management system (RDBMS) for storing and retrieving data efficiently.

### 1.6.4 UI/UX Design:

1. **Adobe XD**: A powerful design tool for creating wireframes, prototypes, and user interfaces with a focus on user experience (UX).

# Chapter 2

# “Literature Review”

# Chapter 2

## Literature Review

## 2.1 Introduction

the literature review is a crucial part of any research or project since it can offer the reader a different perspective on how to design and build the system in question so that it can function more effectively and completely.

## 2.2 Web-based service to workflow management system for the project

# 1.Vezeeeta:

- **Advantages**

1-The possibility of buying medicines online from the pharmacy on the website.


**Figure 2.1 Vezeeta**

2-The patient can talk and consult the doctor online.


**Figure 2.2 Vezeeta**

3-Real evaluations of patients about doctors


**Figure 2.3 Vezeeta**

4-Home visit to the patient


**Figure 2.4 Vezeeta**

- **Disadvantages:**

1-There is no artificial intelligence intervention.

2-There is no upload of patient status data.

3-There is no Articles about teeth

## 2. Clinido:

- **Advantages:**

1- Search in the website using voice

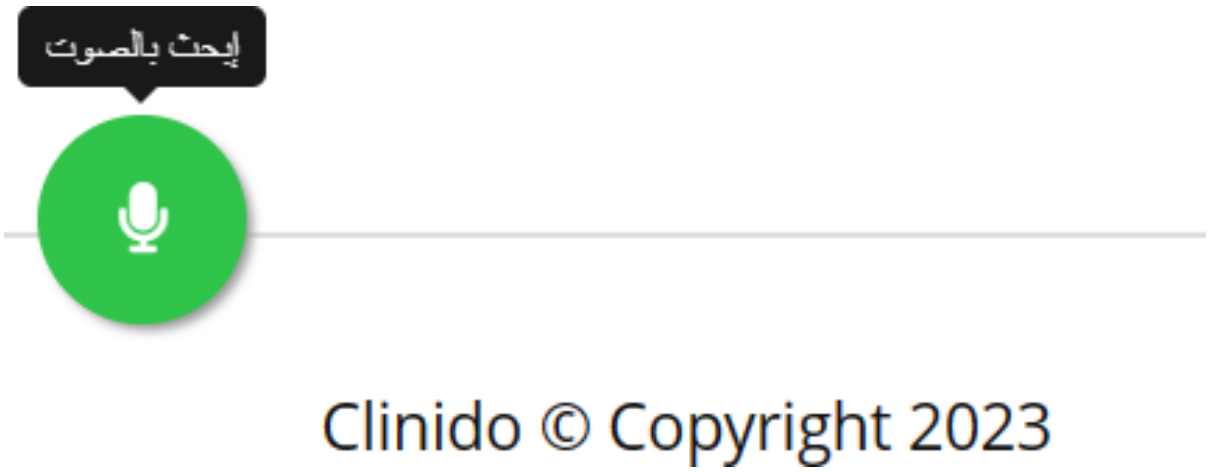



**Figure 2.5 Clinido**

2- There were a lot of questions and answers

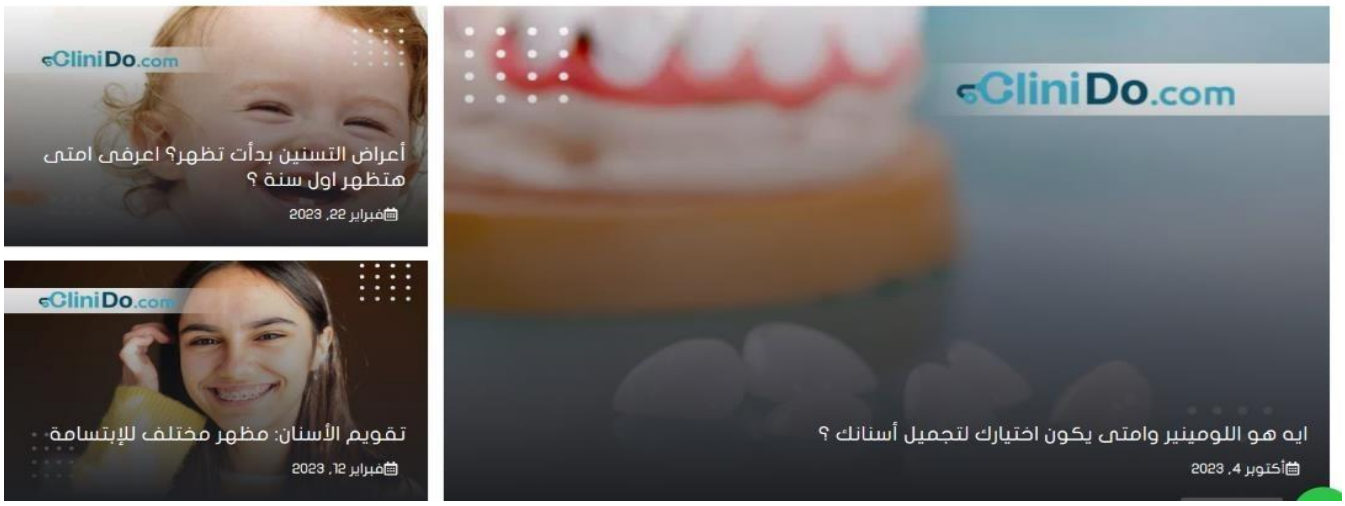



**Figure 2.6 Clinido**

3- Possibility of testing from home


**Figure 2.7 Clinido**

- **Disadvantages**

1-There is no artificial intelligence intervention

2-There is no pharmacy to buy medicines through the site

# 3. Rident dental:

- **Advantages**

  1. Show photos and videos of case results


**Figure 2.8 Rident**

2. There are more choices about your inquiries.


**Figure 2.9 Rident**

3-Ask questions via WhatsApp, which makes things easier forthe user

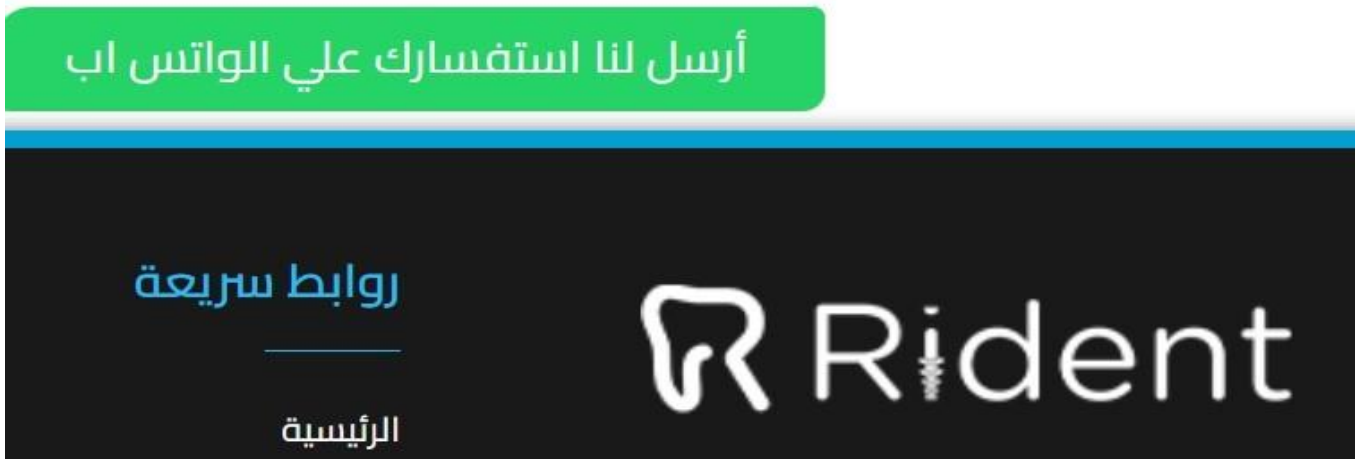



**Figure 2.10 Rident**

- **Disadvantages**

1-There is no search feature on the site

2-There is no user account

# 4. Techno clinics:

- **Advantages:**

1- Home visit to the patient


**Figure 2.11 Techno**

2- Provide doctor information such as a CV This sometimes relieves the user.

**Figure 2.12 Techno**

- **Disadvantages**

  1- you cannot contact with doctor

  2- There is no Articles about teeth

# 4. Dental Clinic:

- **Advantages**

  1- discovered oral diseases through images using artificial intelligence.

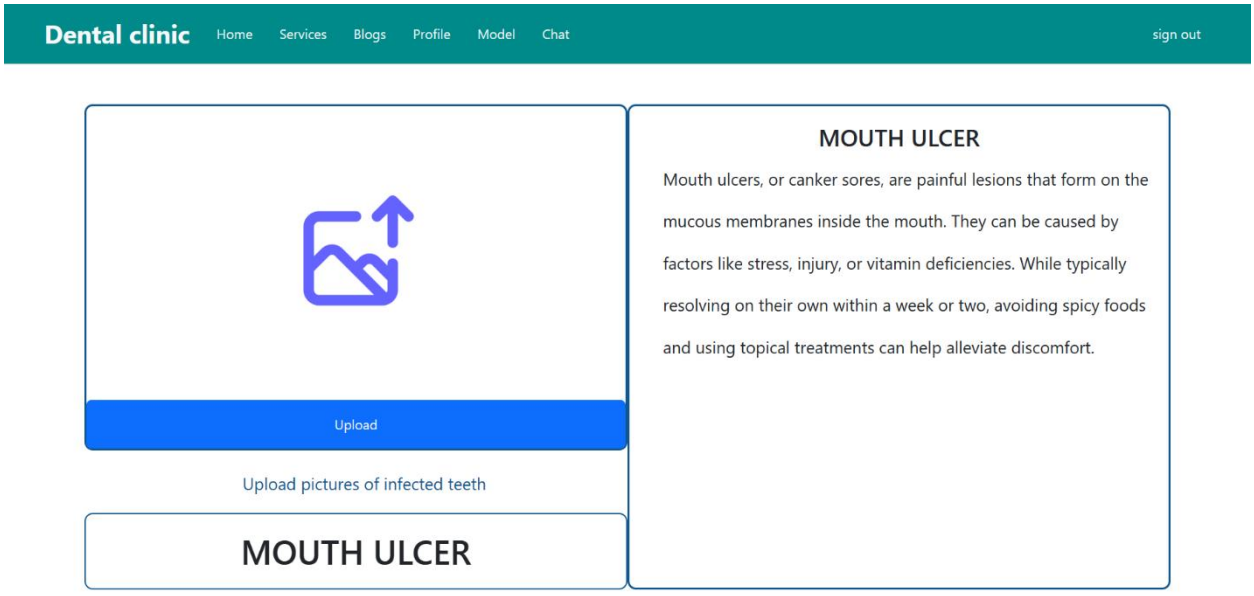



**Figure 2.13 Ai-Enhanced Dental Hospital Website**

2- Articles about dentistry

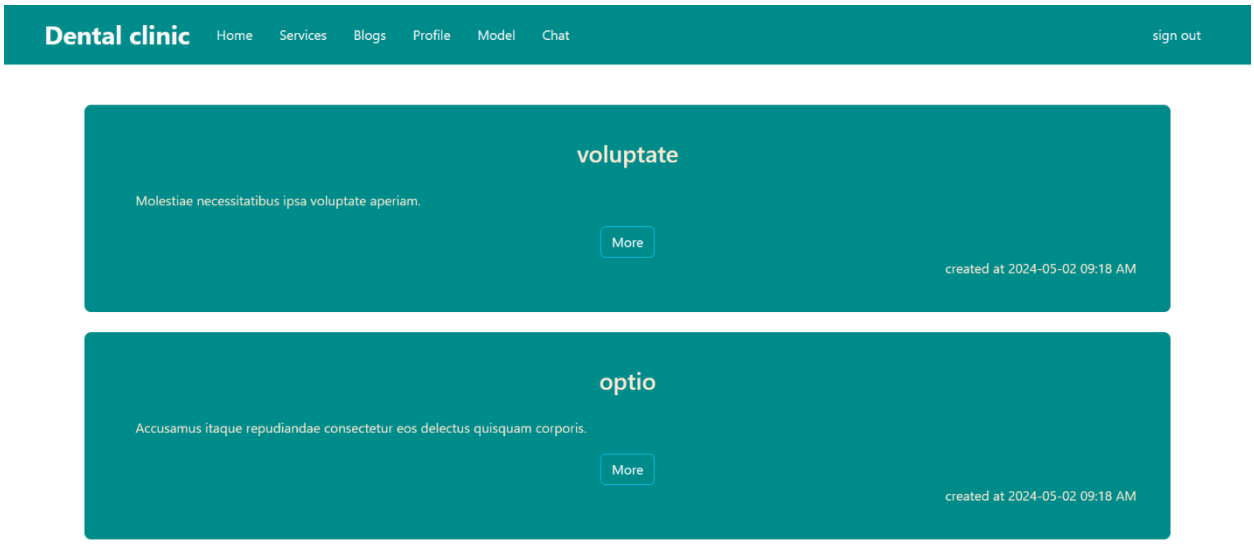



**Figure 2.14 Ai-Enhanced Dental Hospital Website**

3- Chat with doctor

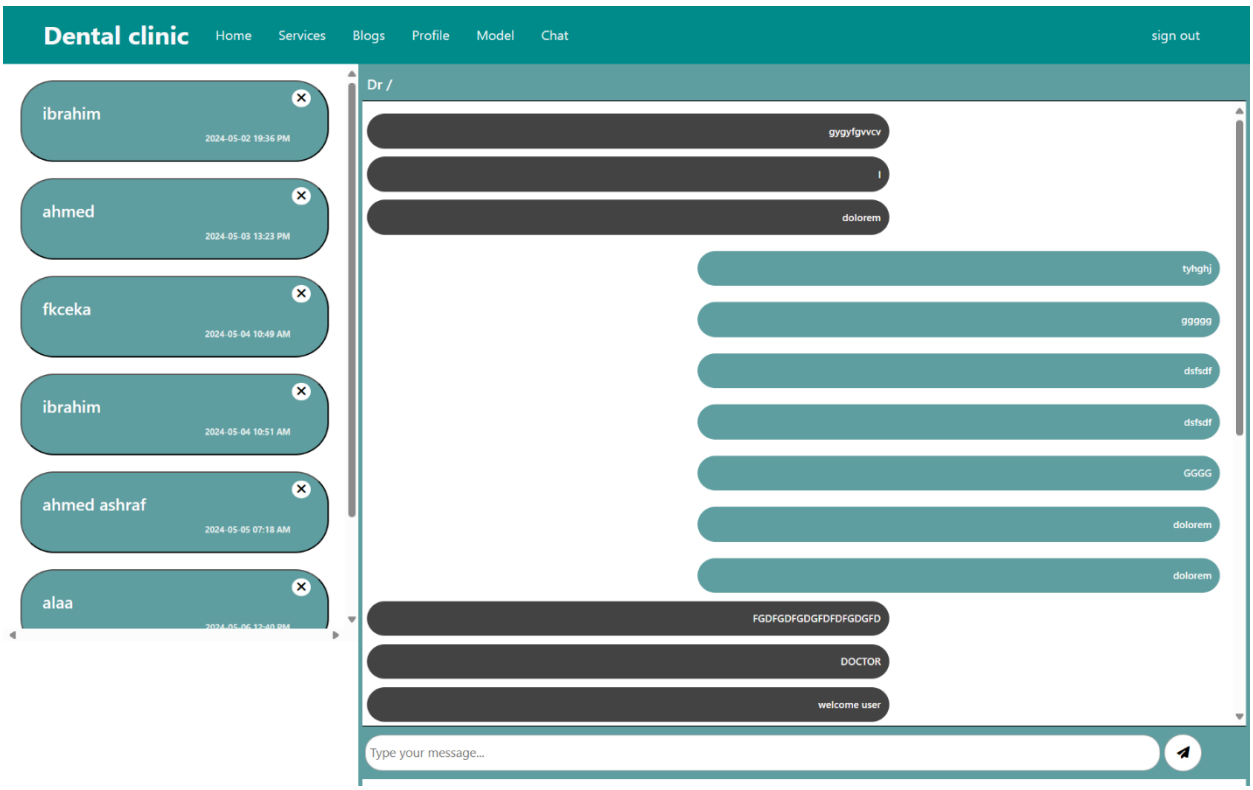



**Figure 2.15 Ai-Enhanced Dental Hospital Website**

4- The doctor can easily upload the case diagnosis file, so hecan obtain it at any time because this file exists to be in the patient's account.

- **Disadvantages**

1- Not being able to visit home

2- There are no patient reviews for Doctors

## 2.3 System Components

### 1. UI/UX components

We can say that our web application is characterized by a user interface design that is simple and easy to use and targets users from all classes and different cultures. It includes activity logs, dashboards, notifications, settings, statistics, etc. these components have nothing to do with the operation of a web application architecture. Instead, they are part of the interface layout plan of this kind of architecture. This helps improve and streamline your architecture.

### 2. Database Component

In our project, we pay close attention to databases, which are a device that stores the registration data required from the user, comments added, or reports uploaded to the site. It plays an important role in structuring web applications. Databases are characterized by many types, some of which have their own advantages and disadvantages, and some of which are the most expansive and prominent in most web applications. In our backend section, we used PHP and Laravel, which in turn PHP is used on approximately 80% of all websites because It is platform-independent, the best of which is MySQL are based on a relational framework and make use of tables, rows as well as columns for storing information. Data is organized in tables, each with a distinct schema or structure determined by the columns as well as the data type. The tables are linked to one another through keys that allow data to be read and shared across several tables.

## 3. Application Server Component

A server for applications is the backend of a Web Application Architecture. It handles business logic, data storage, and communications to other platforms. It is accountable for the processing and management of the information that is transmitted and received by the user interface as well as other systems. It also integrates the business logic of the Web Application Architecture, for example, validating input from users as well as performing calculations and applying the business rules. It is also able to connect with other services or systems including the search engine, mail servers as well as social media platforms, in order to enhance the functionality of the Web Application Architecture. it receives the request via the UI then processes it and then forwards the request to database systems or any other system to be processed further and then transmits the appropriate response to the requester.

## 4. Web Server Component

Web servers are software that processes the inbound HTTP requests and provides appropriate responses, for example, the blog's website pages. It is responsible for processing and handling requests made by the browser on the client and sending the appropriate responses. When a visitor requests an online page on a blog it is sent through the server which retrieves the required resources, including the HTML, CSS, and JavaScript files, and transmits the files back to the user's browser. The browser renders the website page and shows it to the user.

## 2.4 The scalability and performance

### 1. Performance Metrics:

**1.1. Response Time:**
The time taken for the system to respond to user requests, including appointment booking and AI-based tooth problem detection.

**1.2. Throughput:**
The number of requests the system can handle per unit of time, ensuring efficient usage during peak hours.

**1.3. Resource Utilization:**
Monitor CPU, memory, and disk usage to optimize resource allocation for improved performance.

### 2. Scalability:

**2.1. Horizontal Scalability:**

Design the system to scale horizontally by adding more servers to handle increased load. This can be achieved through containerization or cloud-based deployment on platforms like AWS or Google Cloud.

**2.2. Database Scalability:**

Choose a scalable database solution like NoSQL or cloud-based database services to accommodate growing data volumes from appointments, patient records, and AI model results.

## 3. Performance Optimization Strategies:

### 3.1. Caching Mechanisms:

Implement caching mechanisms for frequently accessed data like appointment types, dentist profiles, and common procedures to reduce database load and improve response times.

### 3.2. Load Balancing:

Distribute incoming requests evenly across multiple servers to avoid overloading a single server.

### 3.3. Asynchronous Processing:

Utilize asynchronous processing for non-blocking operations, enhancing the overall responsiveness of the system.

## 4. AI Model Optimization:

### 4.1. Model Compression:

Reduce the size of the AI model without significant loss of accuracy to improve inference speed.

### 4.2. Parallelization:

Implement parallel processing techniques to expedite the tooth problem detection process.

# Chapter Three

# “Design and Methodology”

## 3.1 Requirements and Analysis

### 3.1.1 Software Process Model:

**Agile Methodology Overview**

**Introduction**

The graduation project follows the Agile methodology, a dynamic and iterative approach to project management and software development. Agile is characterized by its flexibility, collaboration, and focus on delivering incremental, customer-centric results.

**Key Principles**

**1. Iterative Development:**

Agile breaks down the project into smaller, manageable increments, allowing for continuous improvement and adaptation. This approach facilitates the delivery of functional increments in short cycles.

**2. Cross-Functional Teams:**

Our project team is composed of diverse members with varied skills to ensure a holistic and collaborative approach. This promotes effective communication, shared responsibility, and a comprehensive skill set to address project needs.

**3. Adaptability to Change:**

Agile embraces change and is designed to accommodate evolving project requirements. This adaptability allows us to respond to shifting priorities, emerging insights, and unforeseen challenges in a timely manner.

**Project Management Frameworks**

Our project employs the Scrum framework, a widely adopted Agile methodology, to structure and guide our development process. Scrum includes key elements such as sprint planning, daily stand-ups, sprint reviews, and retrospectives.

**Methodology in Practice**

**1. Sprint Planning:**

Before each sprint, the team collaboratively plans and prioritizes the tasks to be undertaken, selecting items from the product backlog based on stakeholder priorities.

**2. Daily Stand-ups:**

Daily stand-up meetings keep the team informed about progress, challenges, and upcoming tasks. This promotes transparency and enables quick problem resolution.

**3. Sprint Reviews:**

At the end of each sprint, the team presents the deliverables to stakeholders for feedback. This continuous feedback loop ensures that the project stays aligned with stakeholder expectations.

**4. Retrospectives:**

Following each sprint review, the team conducts a retrospective to reflect on what went well, what could be improved, and how to enhance overall team performance in the next iteration.

**Advantages of Agile in Our Project**

**Flexibility and Adaptability:**

Agile's iterative nature allows us to adapt to evolving project requirements, ensuring that our solution remains relevant and effective.

**Stakeholder Involvement:**

Regular engagement with stakeholders ensures that the project aligns with their expectations, fostering a collaborative and customer-centric development process.

**Faster Delivery of Value:**

Incremental delivery of functional components ensures that stakeholders receive tangible results at the end of each sprint, promoting a sense of progress and achievement.

By adopting Agile methodology, our graduation project aims to foster collaboration, prioritize stakeholder needs, and deliver a solution that is not only technically robust but also aligned with the expectations of our end-user.

## 3.1.2 Functional Requirements

### 3.1.2.1 System Users:

#### Admin :

admin is the main administrator of the system. After logging into the system, it performs all the tasks related to the system such as administering the system, register doctor, add records, register equipment, check active visits, fill assessment sheets, and manage appointments.

#### Doctor:

The doctor logs into the system through the fingerprint attendance system, and through the system he can also look at the patients record.

#### Moderator:

The moderator logs in to the system and then can add records, register Equipment, fill assessment sheets, check active visit and manage appointment.

## 3.1.3 User Requirements

### 1- Patient Registration:

Patients should be able to register in the system, providing necessary personal and contact details

### 2- AI-Driven Initial Diagnosis:

Implement an AI model that analyzes patient-provided information and offers an initial diagnosis of potential dental issues.

### 3- Appointment Scheduling:

Patients should be able to schedule appointments with dentists

### 4- Real-time communication:

Enable real-time communication between patients and doctors for consultations

### 5- Medical History Tracking:

Maintain a comprehensive record of each patient's dental history, including diagnoses, treatments, and prescribed medications.

### 6- Loading and storage:

Allow patients to upload relevant information, such as images, to assist in the diagnostic process.

### 7-Educational Resources:

Provide educational resources on dental health, treatments, and preventive measures to empower patients with information.

## 3.2 System Models

### 1. Data Flow Diagram (DFD)

- DFD is a graphical representation of the flow of data through a system.

  It provides information about the input & output of each entity in the system and what the process itself contains.

- It has multiple levels: Level 0 (Context Diagram), Level 1.

- Context Diagram is a simple overview of the system that only shows high-level system processes.

- Level 1 DFD is a more detailed diagram, where some of the general processes in the context diagram are broken up to show smaller ones. This shows more detail about how data is handled in the system.

### 2. Use-Case Diagram

- A use-case diagram shows the interaction between the system & the external actors. It's a graphical representation of user requirements.

### 3. Sequence Diagram

- The sequence diagram shows process interactions over a timeline. It shows the sequence of messages exchanged between different entities in the system to carry out certain processes.

## 4. Class Diagram

- The UML Class diagram is a graphical notation used to construct and visualize object-oriented systems. A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system.
- The class diagram is the main building block of object-oriented modeling.

## 5. Activity Diagram

- The activity diagram is a graphical representation of step-by-step workflows from start to end. They show iteration, choice and concurrency of actions in the system.

## 6. Entity Relation Diagram (ERD)

- ERD displays the different entities in the system and shows the specific relationships between them. It's used to show the structure of the system database.

## 3.2.1 Data Flow Diagram (DFD)

### i. Context Diagram

This image depicts a flowchart that outlines the architecture and interaction flow of a healthcare website. It illustrates how different types of users (Admin, Doctor, and general User) interact with the website and with each other through various functions and features. Here's a breakdown:

**1. User:**

- Can book or delete an appointment via the website, and receives a confirmation in return.
- Can view appointments, which is a feature facilitated through the website.

2**. Website:**

- Acts as the central node facilitating all interactions.
- Users book/delete appointments through the website.
- Doctors can upload diagnostic files and view patient history.
- Admin can add or remove doctors, approve or reject appointments, and modify blog entries.
- The website sends confirmation messages to users and admins based on their actions (e.g., booking an appointment or modifying the blog).

**3. Doctor:**

- Has the capability to view a patient's history.
- Can upload diagnostic files.
- Can view appointments set by users.

**4. Admin:**

- Has the authority to add or remove doctors.
- Can approve or reject appointments booked by users.

- Can modify the content of the blog.

- Receives confirmations for actions taken (like adding or removing doctors).

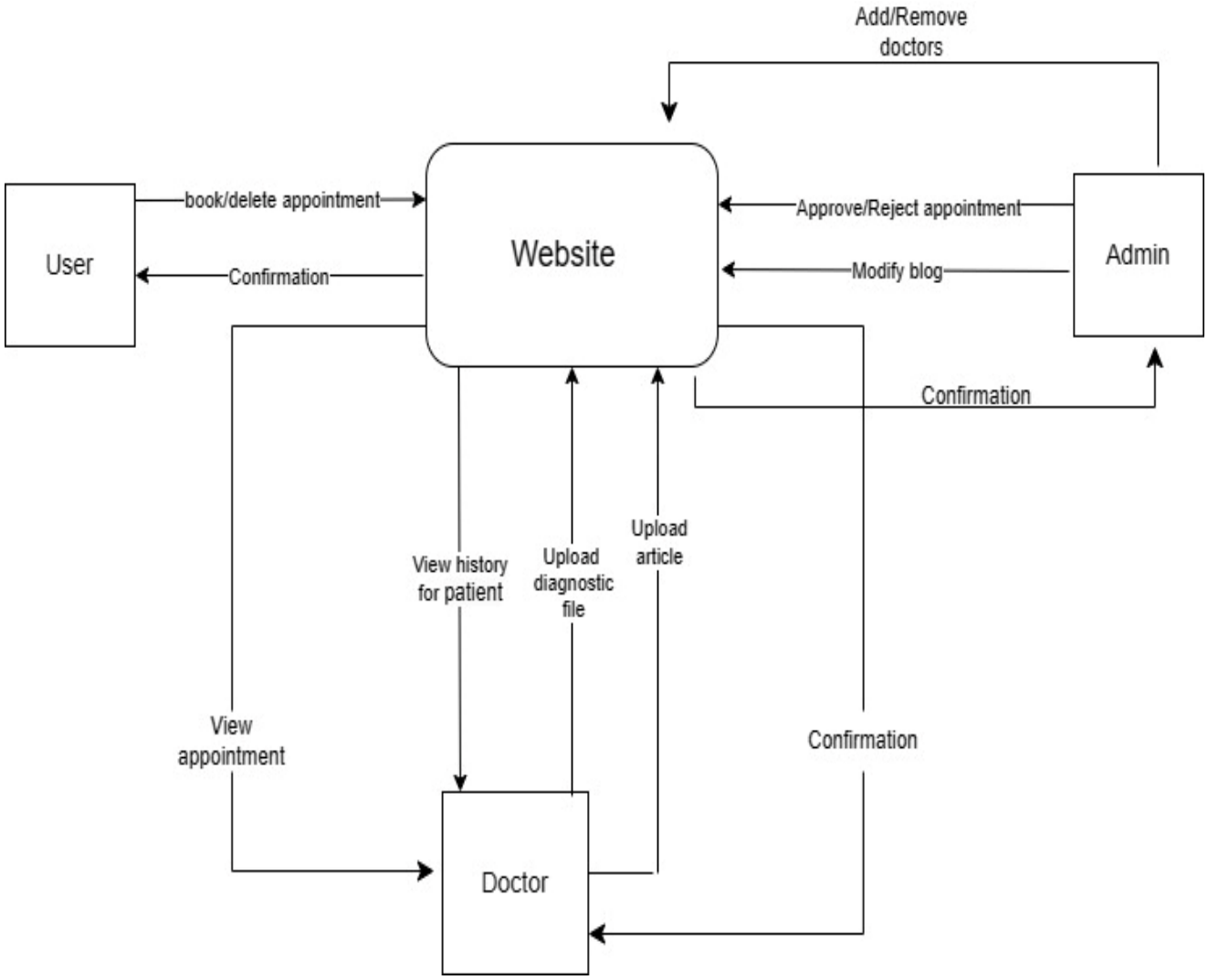



**Figure 3.1 Context Diagram**

## ii. DFD (Level 1)

This summary encapsulates the flow and management of data and processes within the system, including user, doctor, and admin interactions, as well as additional functionalities like AI analysis and content management.

**1. User Interactions:**

- User Authentication: Verifies user credentials for system access.
- Manage Appointments: Users can add, modify, or delete appointments, interacting with the appointments database.
- Account Management: Users can update their profile details stored in a user profiles database.

**2. Doctor Interactions:**

- Doctor Management: Admins can add or remove doctors, updating the doctors database.
- Diagnostic Management: Doctors can upload diagnostic files to a designated storage.

**3. Admin Interactions:**

- System Management: Admins monitor and control system operations, storing logs for activities.

**4. Additional Features:**

- AI Image Analysis: Medical images are analyzed using AI, with inputs and results managed through respective databases.
- Content Management (Blog): Users and doctors can publish and manage blog content.
- Patient History Viewing: Medical histories can be accessed and viewed, involving interactions with a patient records database.

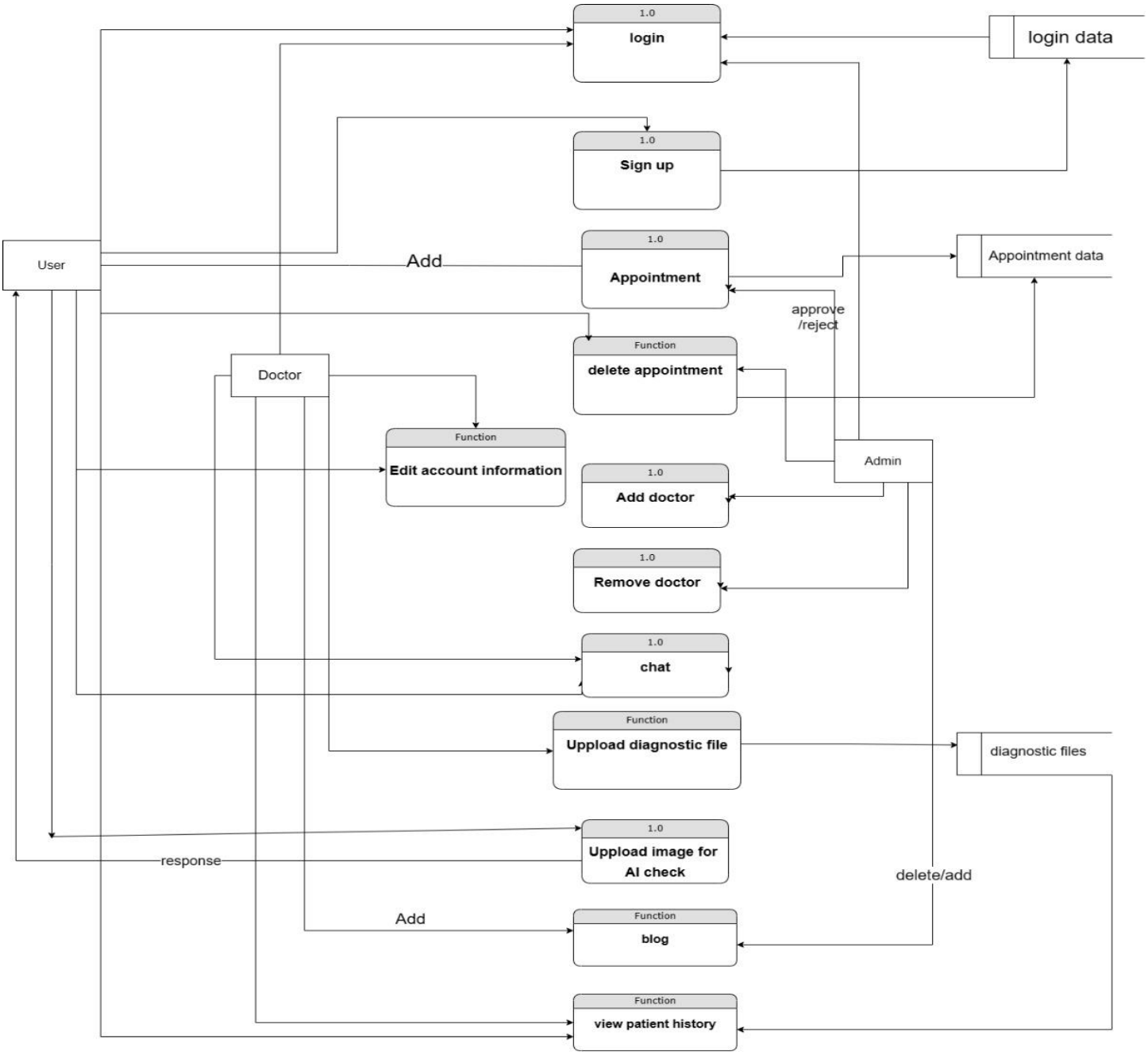



**Figure 3.2 DFD (Level 1)**

# 1. Use-Case Diagram

This use diagram likely depicts a dental clinic management system. It shows how different users interact with the system to manage appointments, patient records, and potentially clinic operations.

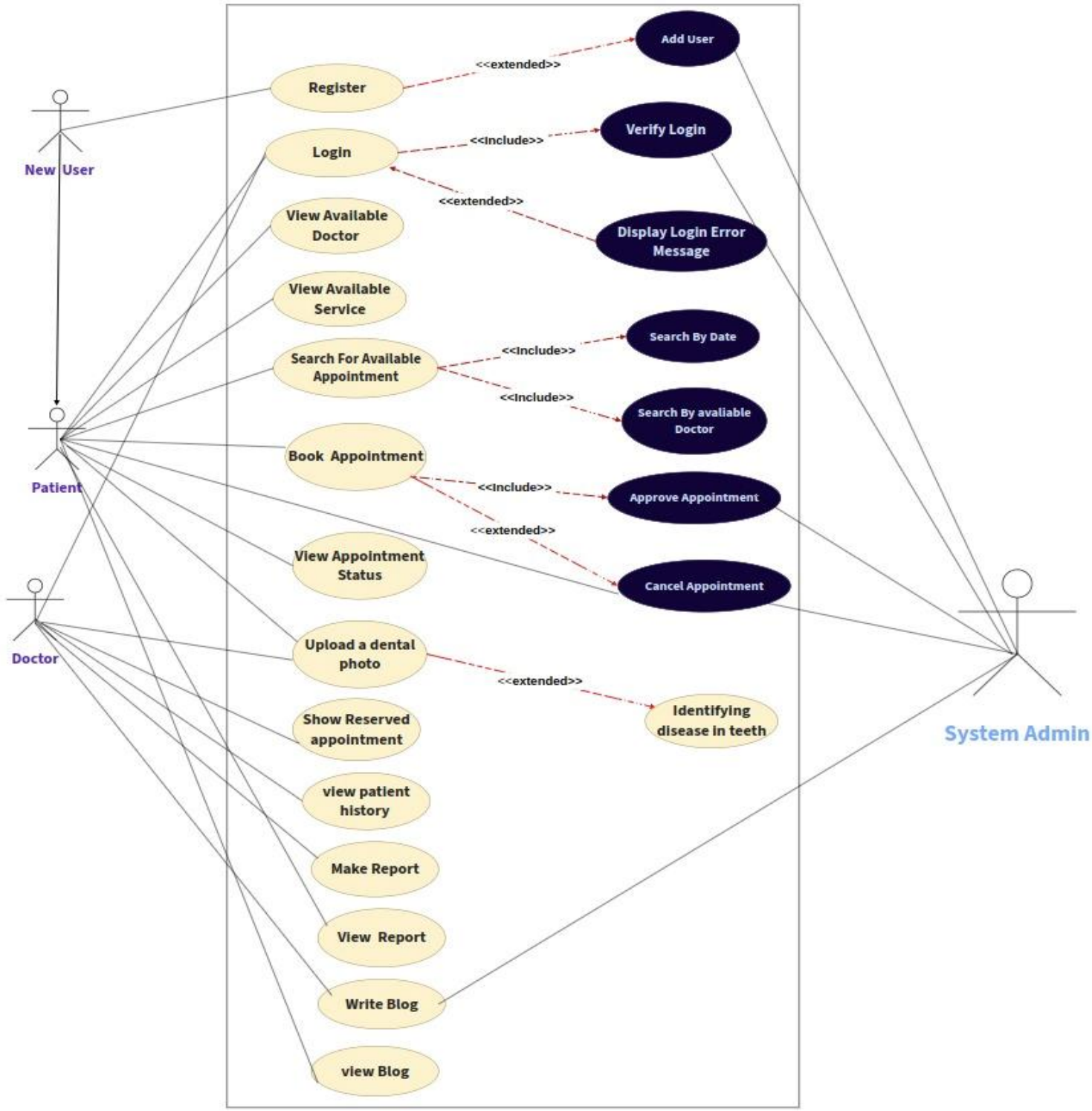



**Figure 3.3 Use-Case Diagram**

## • Sequence Diagram

This sequence diagram illustrates the interactions between different actors (Patient, Doctor, AI Model, Admin, Web Application) and the database for various operations in a web-based healthcare application. Here is a detailed description of the processes depicted:

**1. Creating an Account on the Web App:**

**Patient/Doctor:** Initiates the account creation process.

**Web Application:** Receives the request and stores the patient/doctor data in the server's database.

**Database:** Saves the data successfully and confirms account creation.

**2. Logging In:**

**Login with Valid Credentials:**

**Patient:** Sends login credentials.

**Web Application:** Verifies login credentials by checking data from the database.

**Database:** Confirms the validity of the credentials.

**Web Application:** Grants access upon successful validation.

**Login with Invalid Credentials:**

**Patient:** Sends login credentials.

**Web Application:** Verifies login credentials by checking data from the database.

**Database:** Identifies the credentials as invalid.

**Web Application:** Denies access and reports login failure.

**3. Booking an Appointment:**

**Patient:** Books an appointment for examination.

**Web Application:** Checks availability and confirms the booking.

**4. Uploading and Processing Dental Images Using AI:**

**Patient:** Uploads an image for the AI Model.

**AI Model:** Analyzes the image to identify the current dental condition and sends the pre-diagnostic assessment to the database.

**Web Application:** Processes the image and stores the patient's status in the database.

**Database:** Confirms the data is stored successfully.

This diagram ensures that the web application effectively manages user authentication, appointment bookings, and pre-diagnostic assessments using AI, while securely interacting with the database for data storage and retrieval.

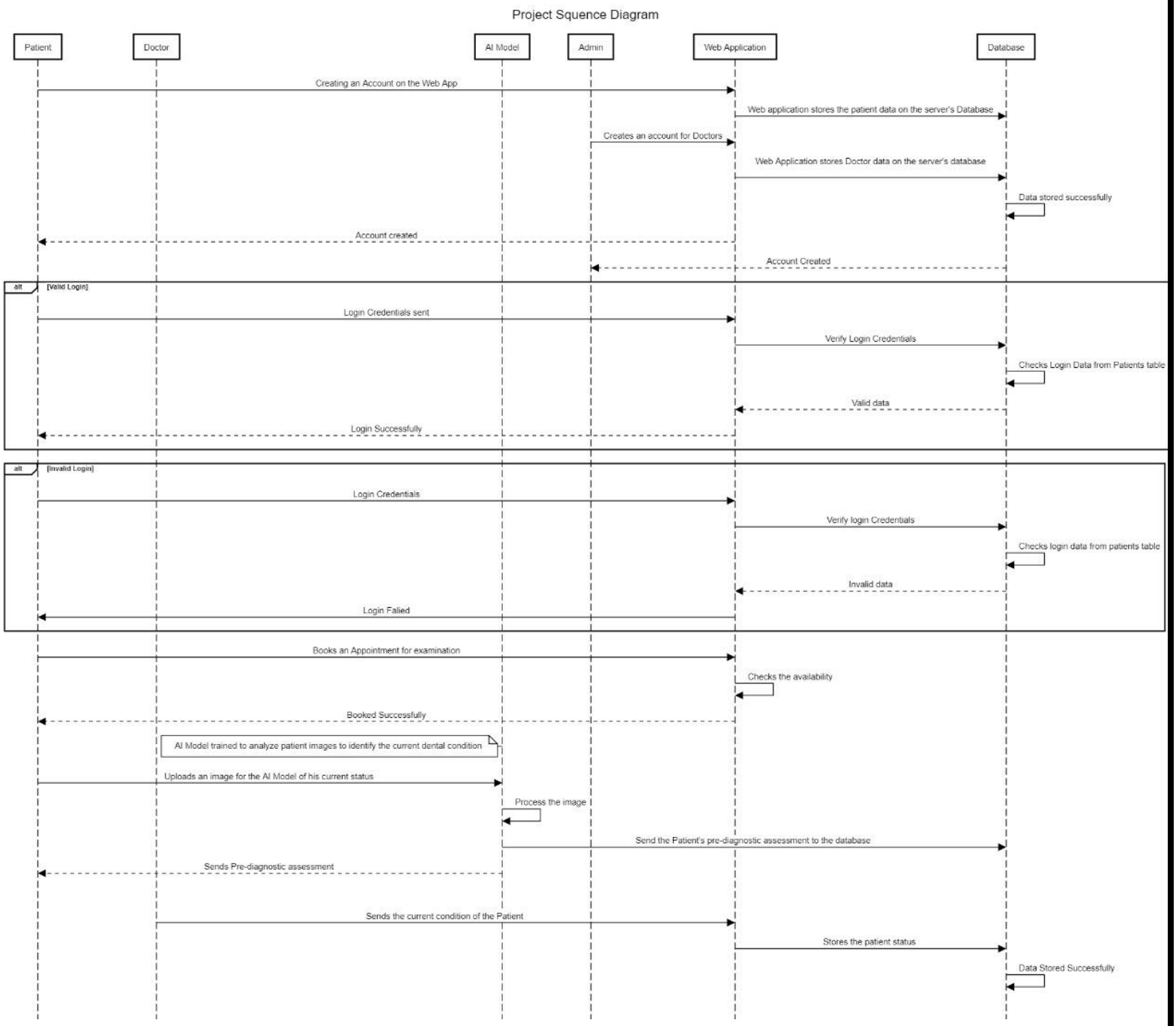



**Figure 3.4 Sequence Diagram**

## 3.2.3 Structural Perspective

### • Class Diagram

as a user of this system, you log in to view health-related blog posts, check out doctors, and book appointments. The admin manages the content and oversees the doctors' information, while the AI assistant helps with analyzing medical photos. The entire system is designed to make managing your health easier and more efficient.

**Admin:**

Manages the system, can add, update, and delete health posts, and check doctors' information.

**Blogs:**

Contains health information written by the admin, which users can read for health tips.

**Users (Patients):**

Have their own accounts to log in, view posts, check doctor information, and book appointments.

**Doctors:**

Have profiles they can edit, and users can view these profiles to choose the right doctor.

**Booking (Appointments):**

Allows users to book appointments with doctors at specific times.

**AI Assistance:**

Helps analyze medical photos and provides results to assist with diagnosis.

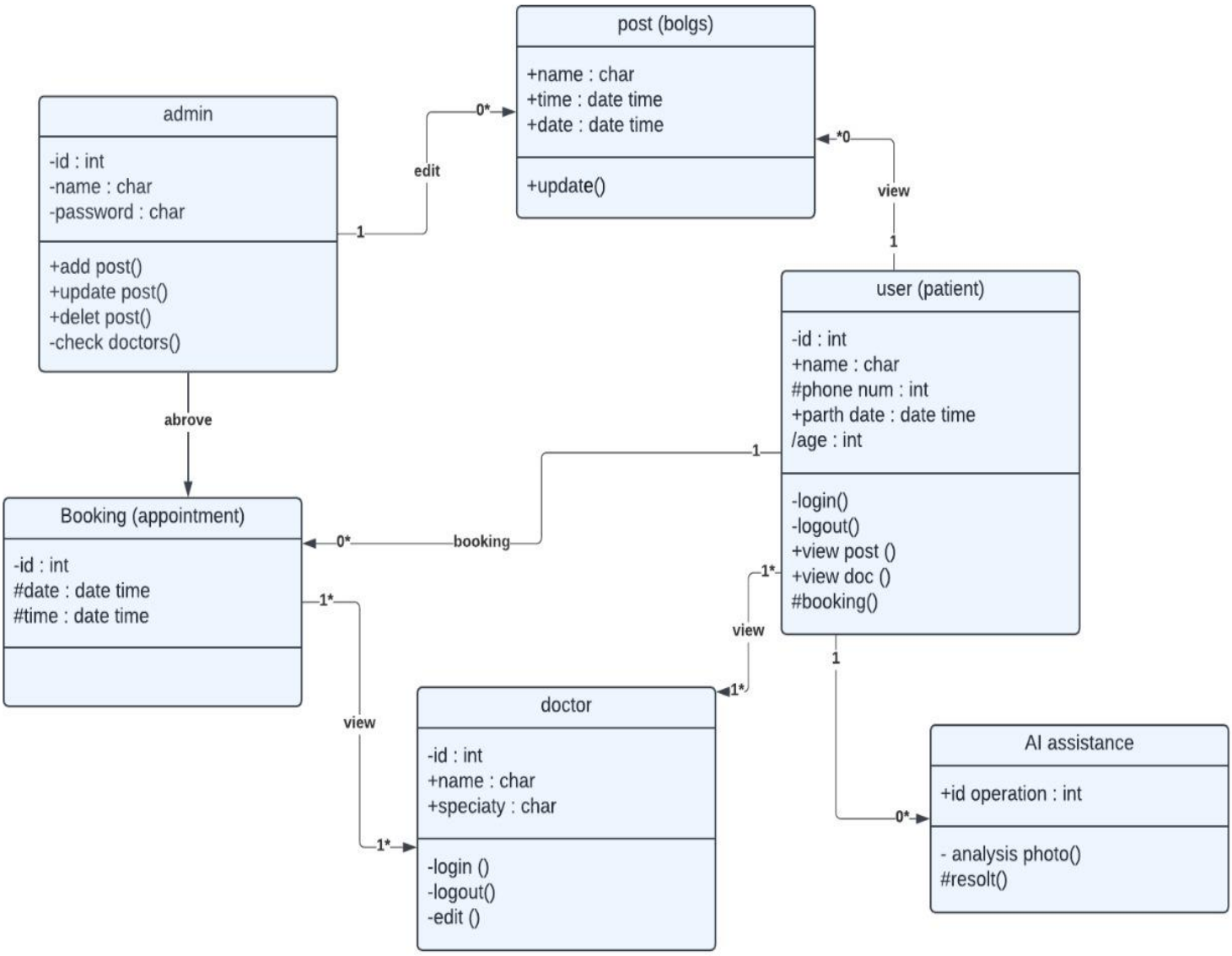



**Figure 3.5 Class Diagram**

## 3.2.4 Behavioral Perspective

### • Activity Diagram

The image is a flowchart diagram representing the user interaction and administrative processes of a web or mobile application, possibly for a healthcare or clinic management system. The flowchart is divided into several sections indicating different user roles and their respective functionalities:

Top Section: This area handles the initial user interaction with the application, starting from the registration or login process. If the user logs in, they are prompted to enter the correct data. If the data entered is right, the user proceeds; if not, they likely stay at the login phase.

Doctor Section: This includes functionalities specific to doctors, such as chatting, uploading diagnostic files, viewing bookings, and adding articles.

Admin Section: Administrators have the ability to create or delete articles and doctors, approve or reject reservations, and add services to the application.

User Section: Regular users can access their profiles, read blogs, chat, access services, and use an AI model (potentially for diagnostic purposes, such as uploading a picture of teeth here). They can select services, and based on their selections, decide whether to book a doctor. If they choose to book, they proceed to make a reservation.

The flowchart uses standard symbols: rectangles for processes, diamonds for decision points, and arrows to show the flow of the processes, leading to a holistic view of how different user roles interact within the system. The final connection points suggest that users, doctors, and admins converge in their functionalities to a final outcome or operation closure.

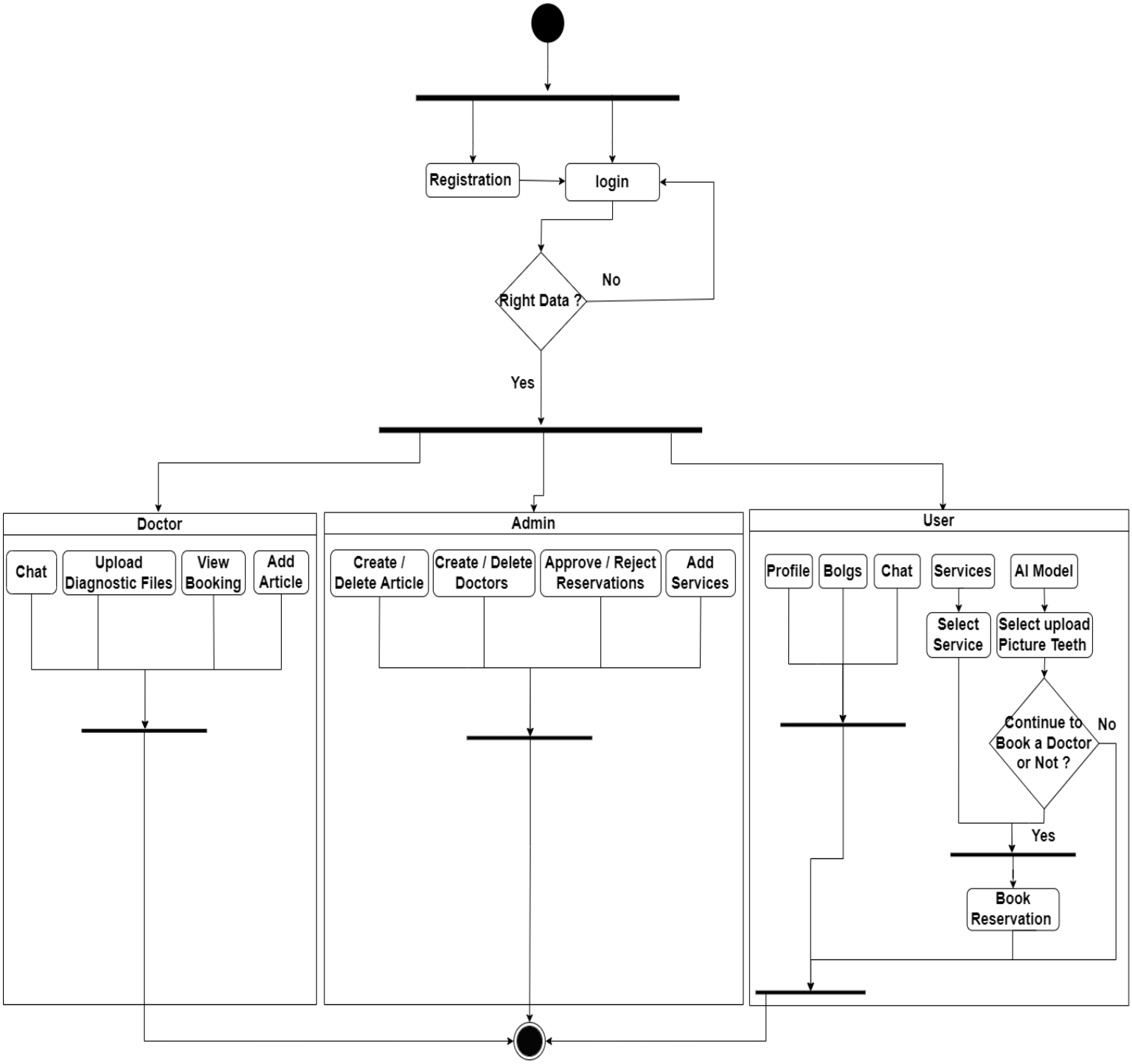



**Figure 3.6 Activity Diagram**

## 3.2.5 Entity Relationship Diagram (ERD)

ER Diagram outlines a medical appointment system featuring entities like Appointment, Patient, Doctor, Admin, and Prescription/Medical Records, detailing their attributes and relationships. Patients and doctors are linked through appointments, while doctors manage prescriptions/medical records. Admins handle entity management and approval processes, ensuring efficient healthcare operations.

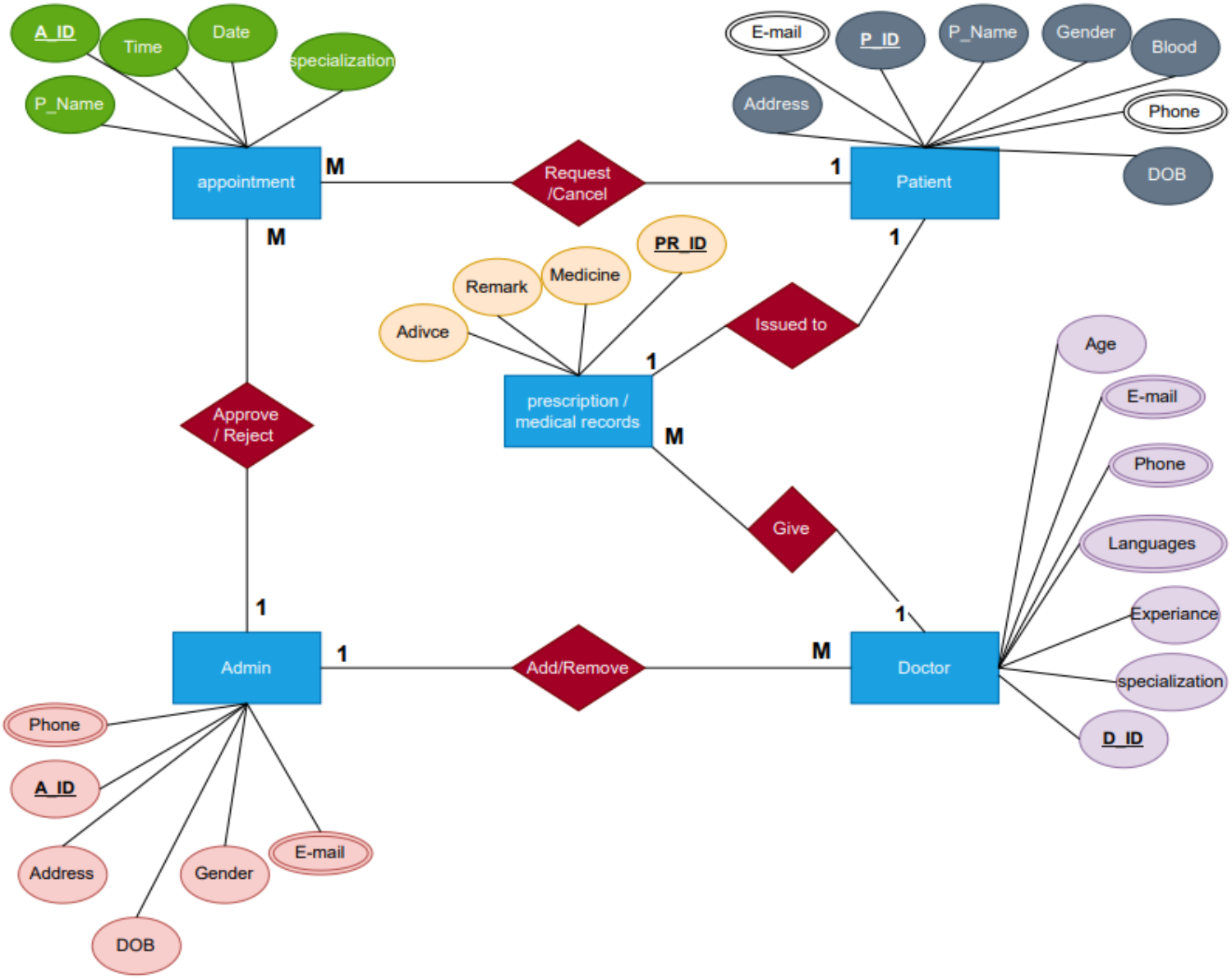



**Figure 3.7 Entity Relationship ( ERD )**

## 3.2.6 System Design

- **Schema**

The Oral care RDM comprises entities for patients, doctors, and appointments, capturing essential personal and medical details. Patients can have multiple email addresses and phone numbers, while prescriptions connect patients with their respective doctors. Appointments detail interactions between patients and administrative staff. Doctors' information includes their contact details, languages spoken, and specializations.

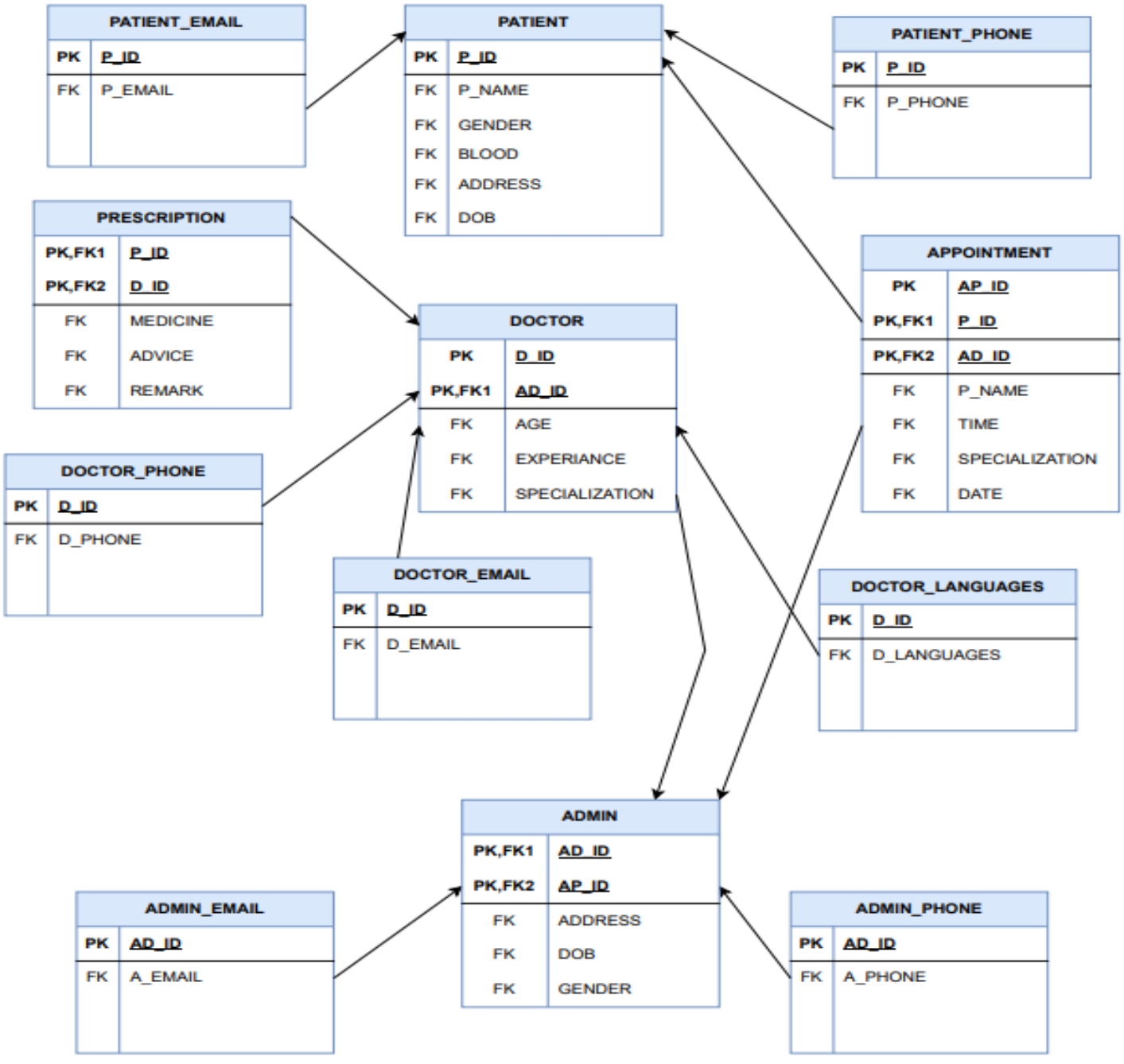



**Figure 3.8 Schema**

# Chapter Four

# “Implementation”

# Front Design

## 4.1 Home Page Design

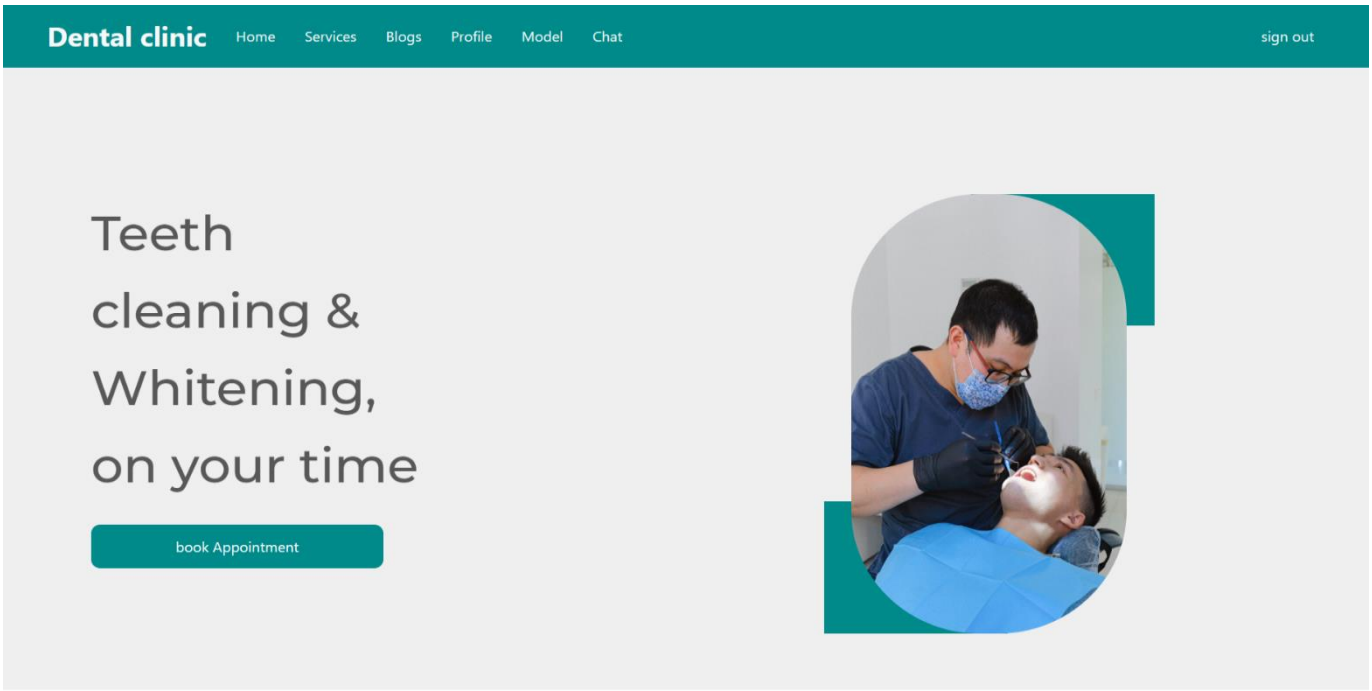



**Figure 4.1 Home Page Design**

## Login Design

Admin or Patient or Doctor access the system by adding their email and password, and it differs for credentials of admin

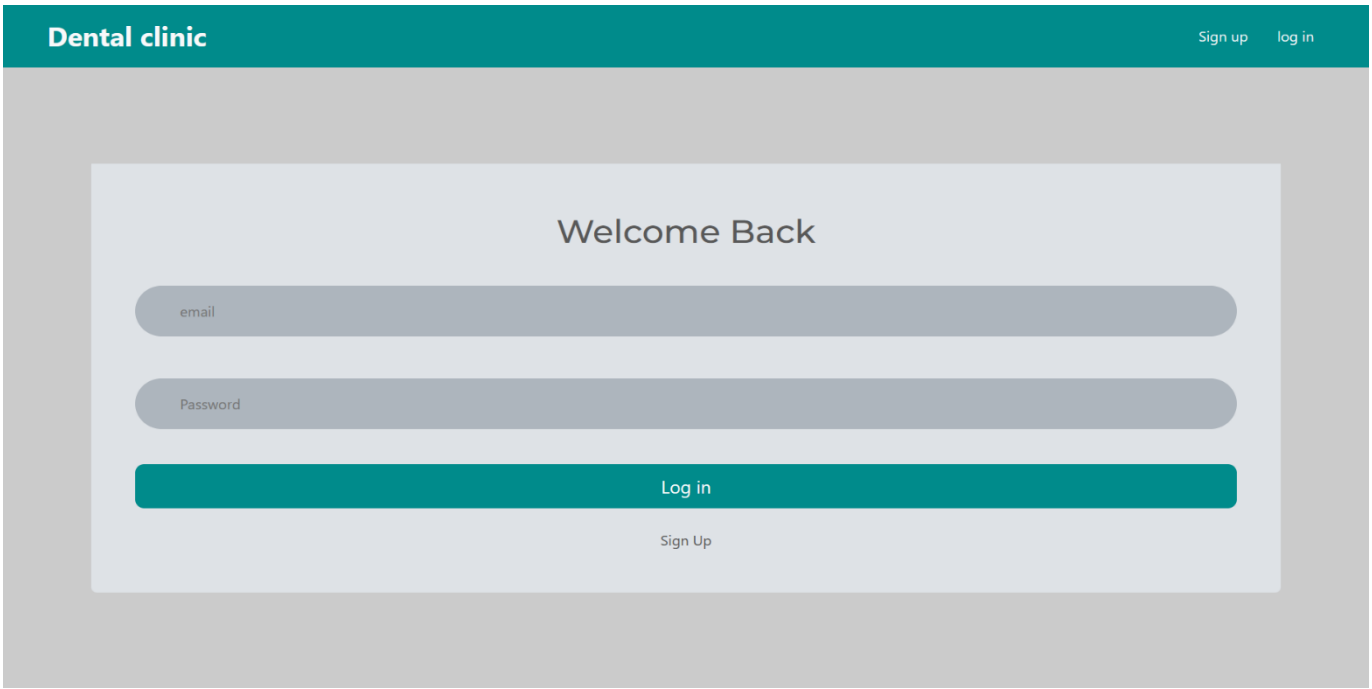



**Figure 4.2 Login Design**

# Register Patient Design

Registering patient requires Name, Email, Phone Number, National ID, Address, Password

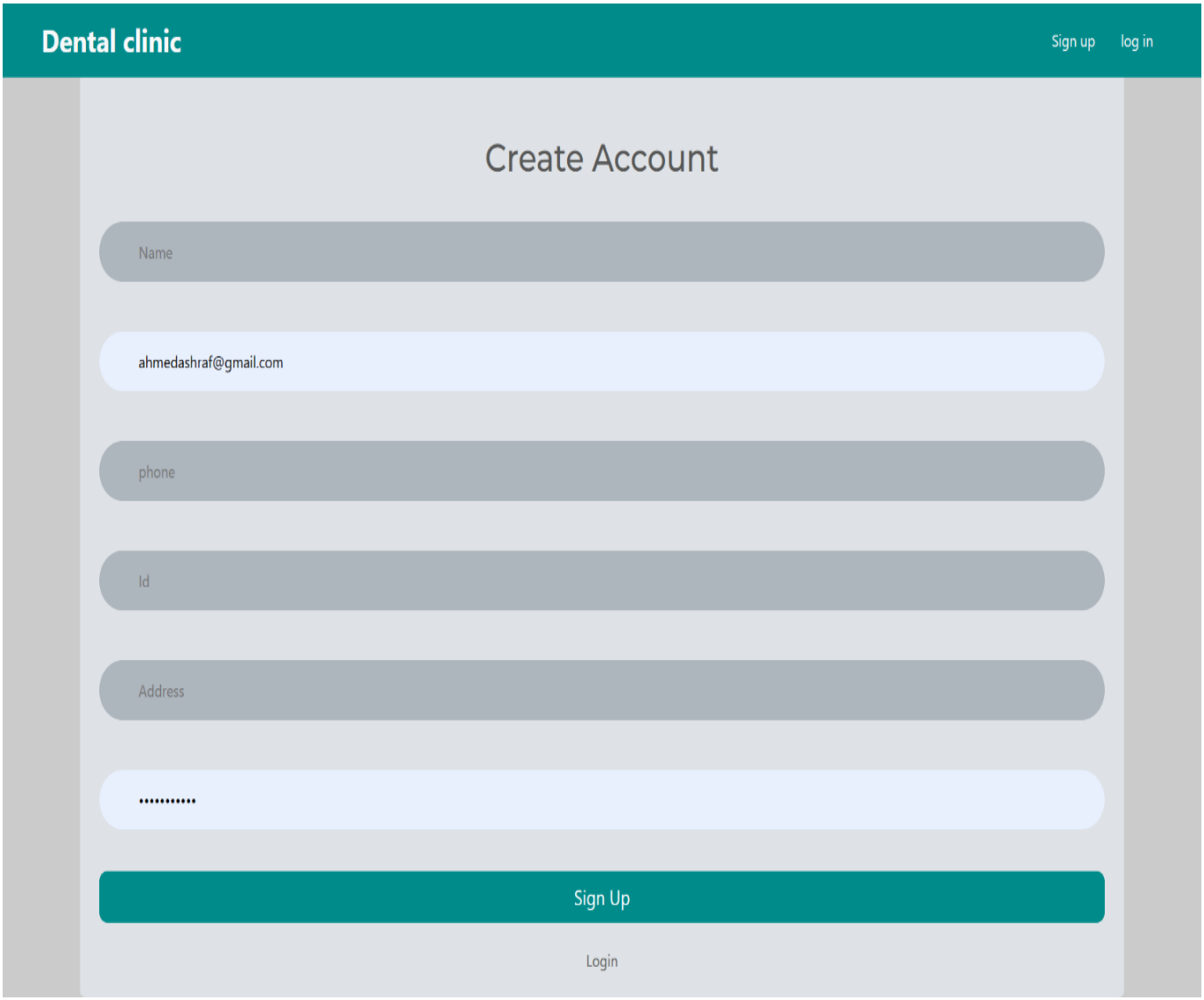



**Figure 4.3 Register Patient Design**

# Services Design

In this part, the departments appear from which the patient chooses a specific department to help him

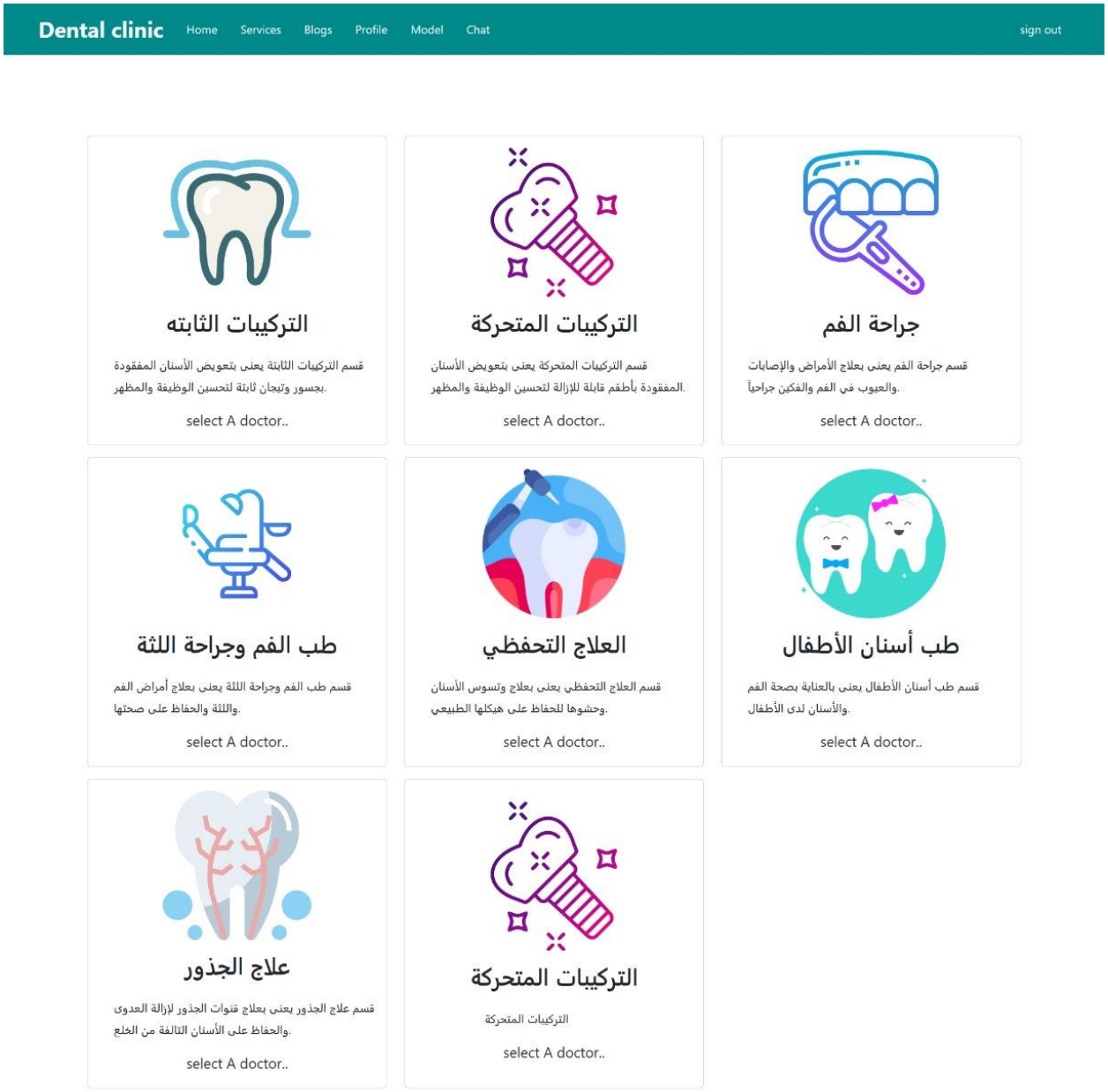



**Figure 4.4 Services Design**

# Doctors Design

In this part, the doctors appear in this section

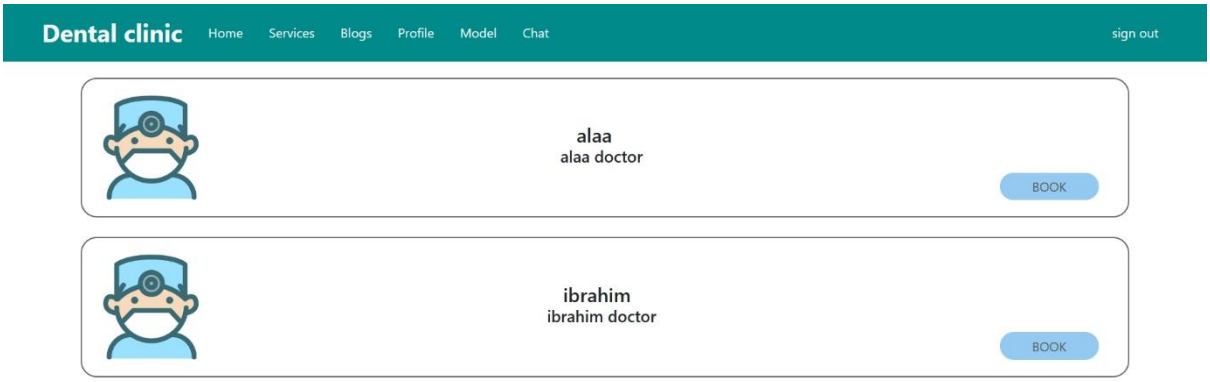



**Figure 4.5 Doctors Design**

# Articles Design

This page shows the articles that appear to the patient to educate and help him

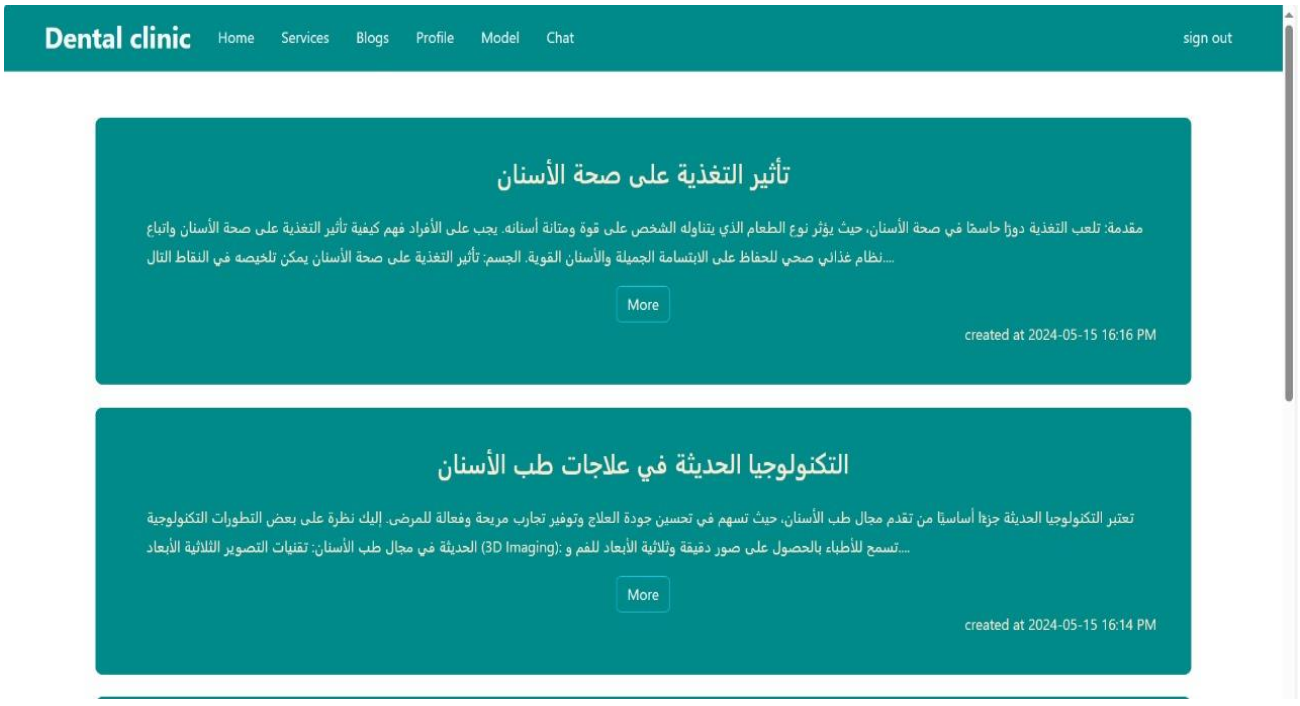



**Figure 4.6 Articles Design**

# Patient Appointment Scheduling Design

Scheduling a meeting with doctor by adding Phone Number, age, Date of Session, Time

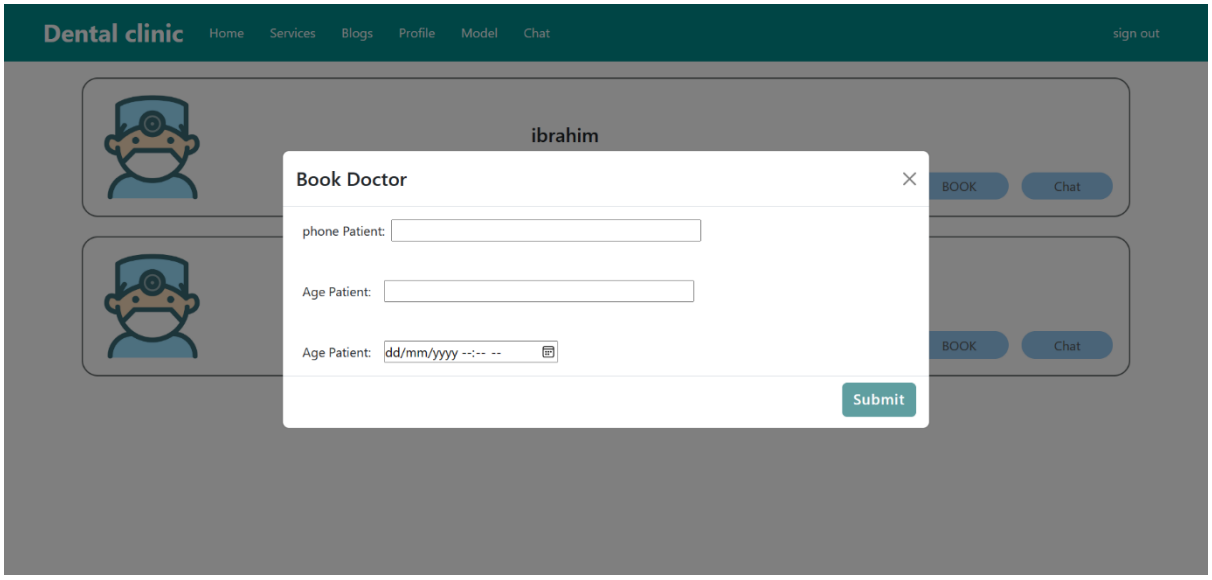



**Figure 4.7 Patient Appointment Scheduling Design**

# Patient Profile Design

On this page, the patient's information and history appear, which are files uploaded by the doctor about the patient and the reservations that the patient made with the doctor.

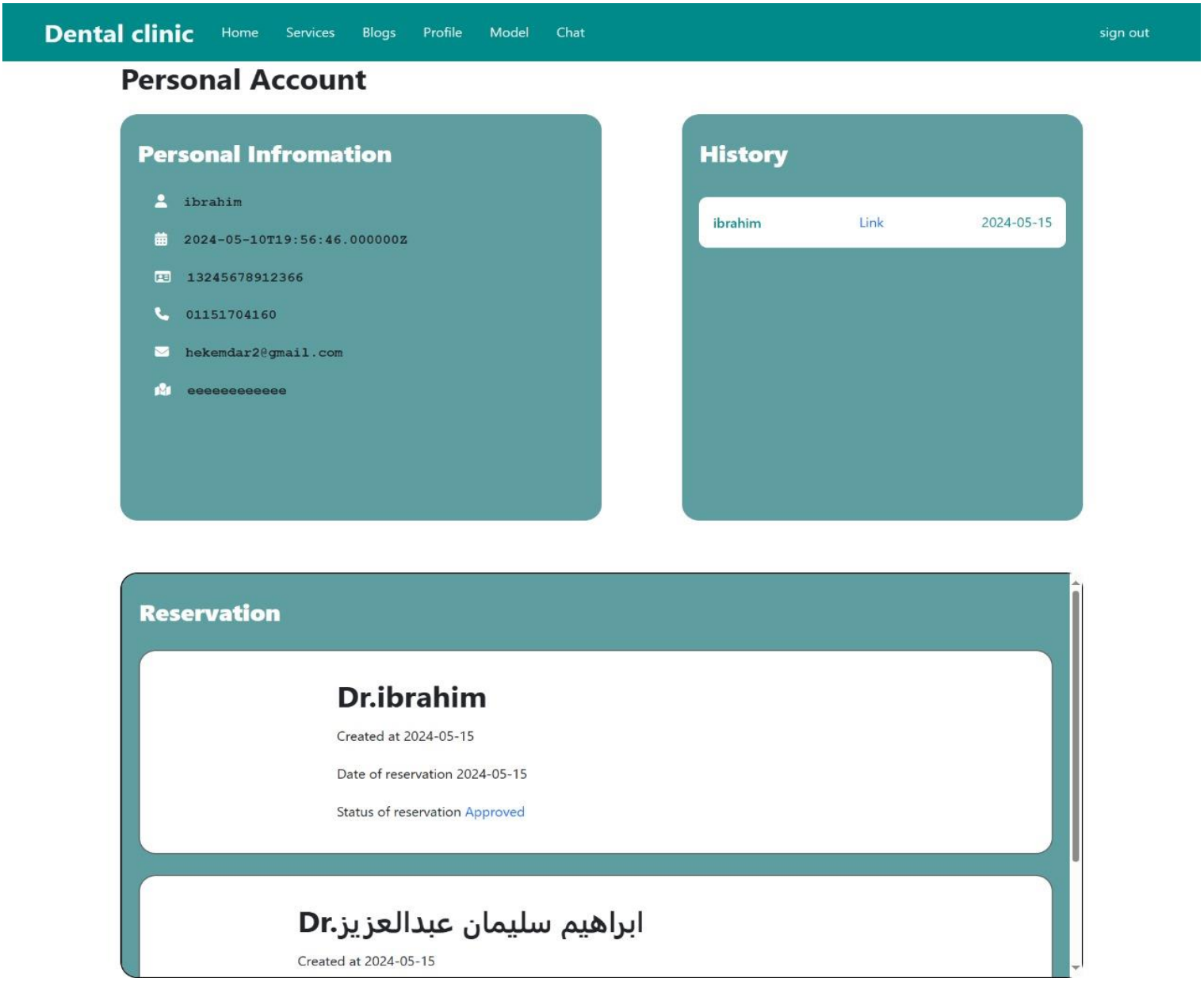



**Figure 4.8 Patient Profile Design**

# Chat Design

On this page, a chat between the doctor and the patient appears, where the patient can talk to the doctor in a chat and the doctor as well

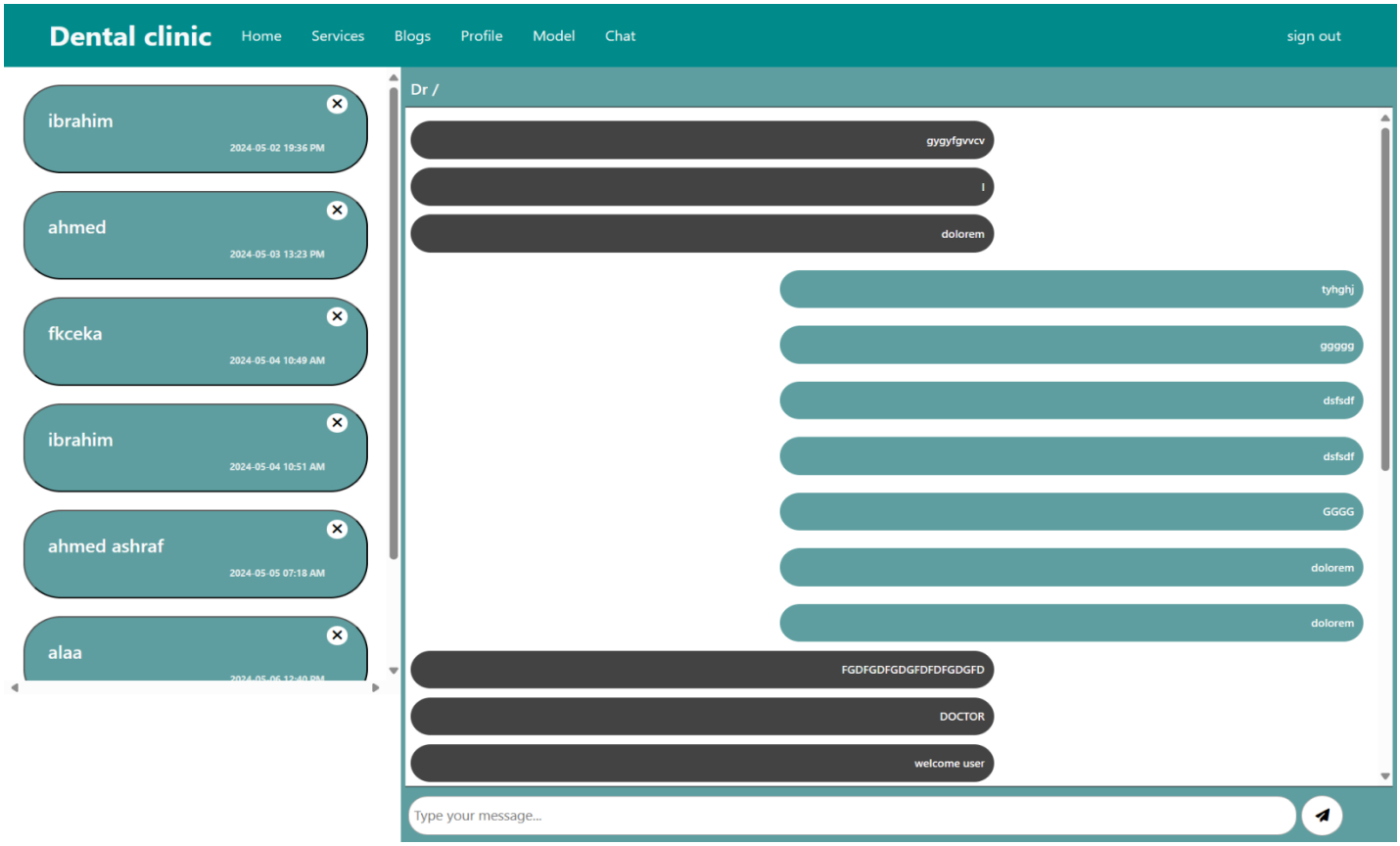



**Figure 4.9 Chat Design**

# AI Model Design

On this page, Moodle AI appears, where a picture of the patient's teeth or mouth is uploaded, and the result of any disease he has is given to him, and information about that disease is shown.

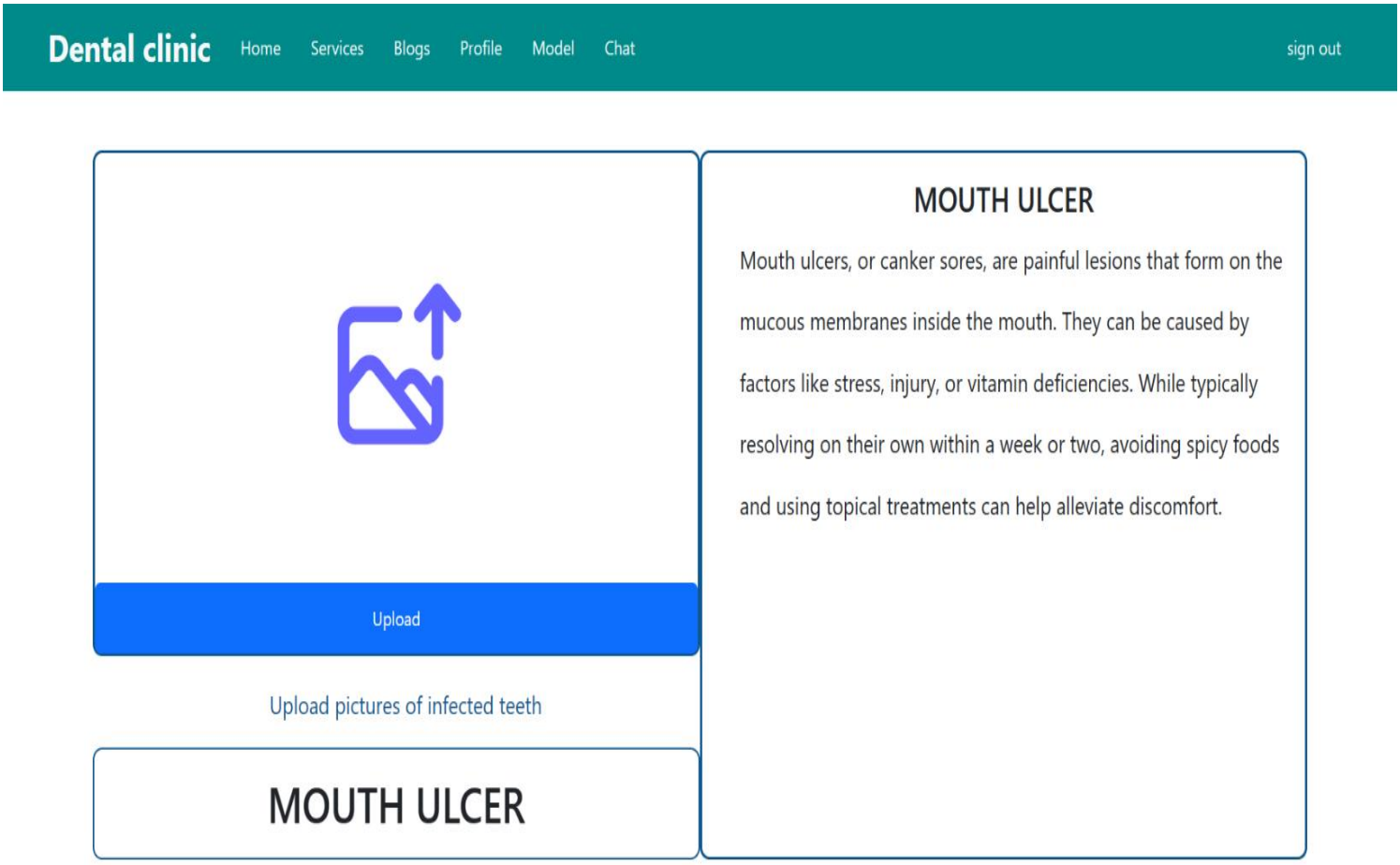



**Figure 4.10 AI Model**

# Add Article Design

On this page, the doctor or administrator can add an article about how to maintain teeth, etc.

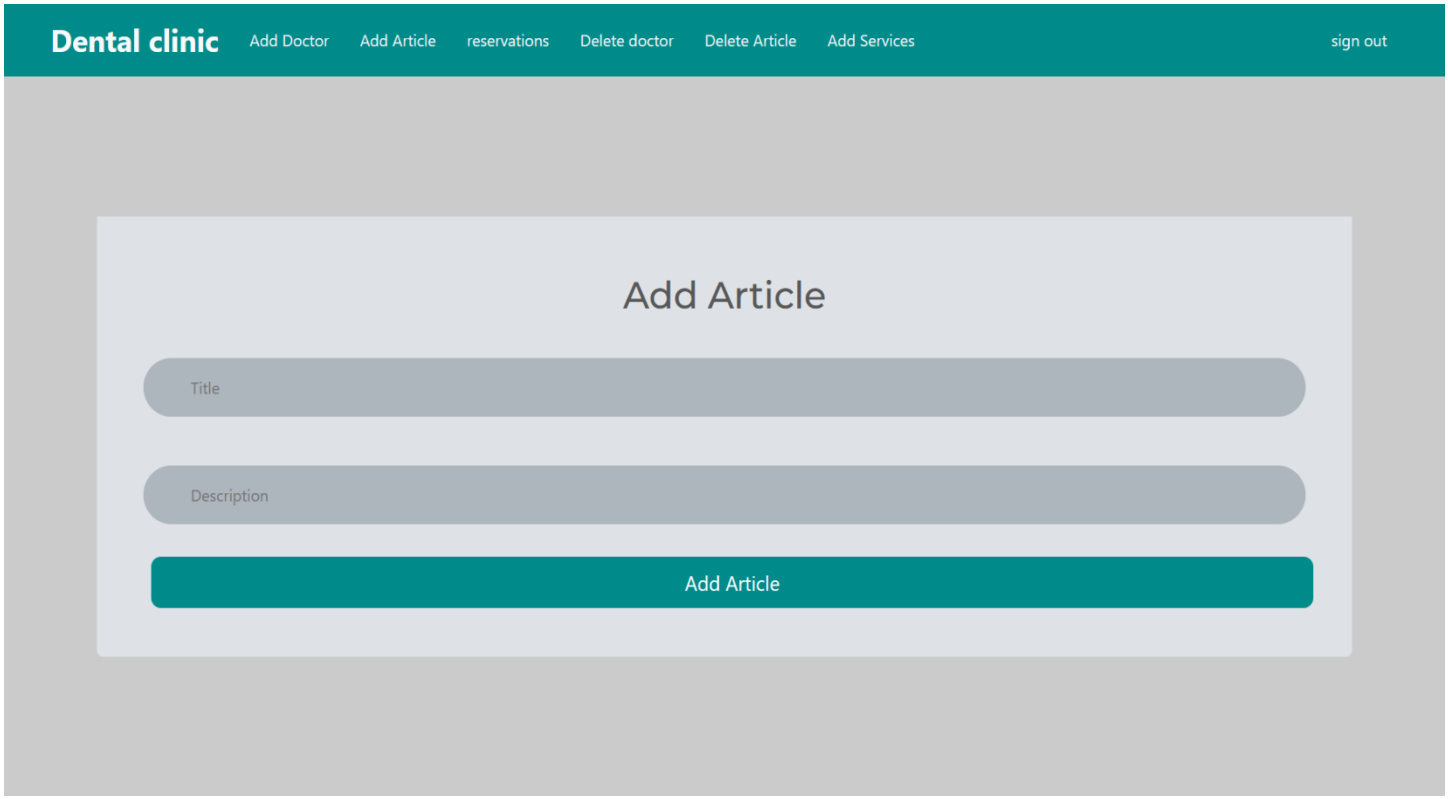



**Figure 4.11 Add Article Design**

# Add Doctor Design

On this page, the admin adds a new doctor to the site by registering the name, email, phone, ID, address, password, category, description, and specialization.


**Figure 4.12 Add Doctor Design**

# Add Services Design

On this page, the admin adds a new service (new section) by registering the name, description, slug, and a picture of the section's appearance.

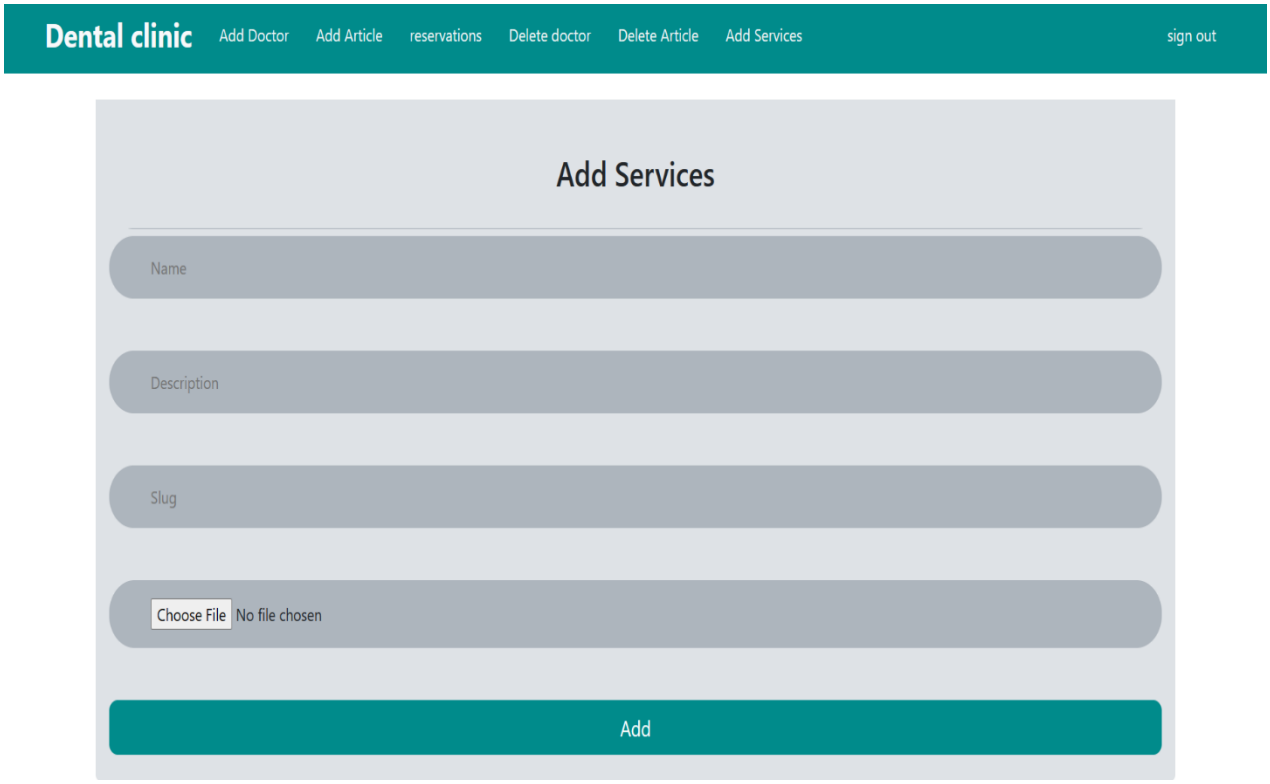



**Figure 4.13 Add Services Design**

# Delete Doctor Design

On this page, the admin can delete a specific doctor

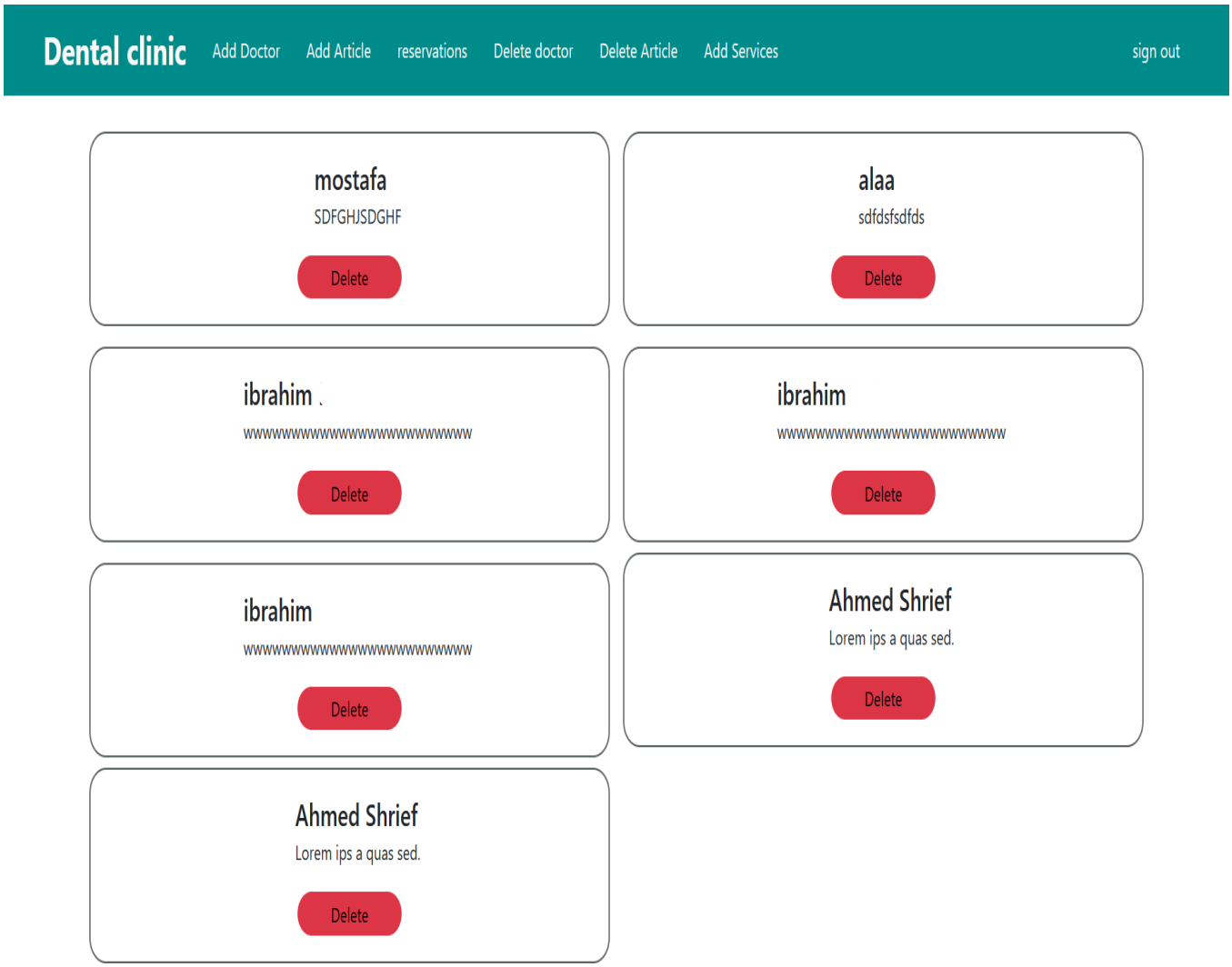



**Figure 4.14 Delete Doctor Design**

# Reservations Admin Design

On this page, the admin can see information about each doctor about the patient, such as his name, gender, age, etc. He can also delete a specific patient from a specific doctor.

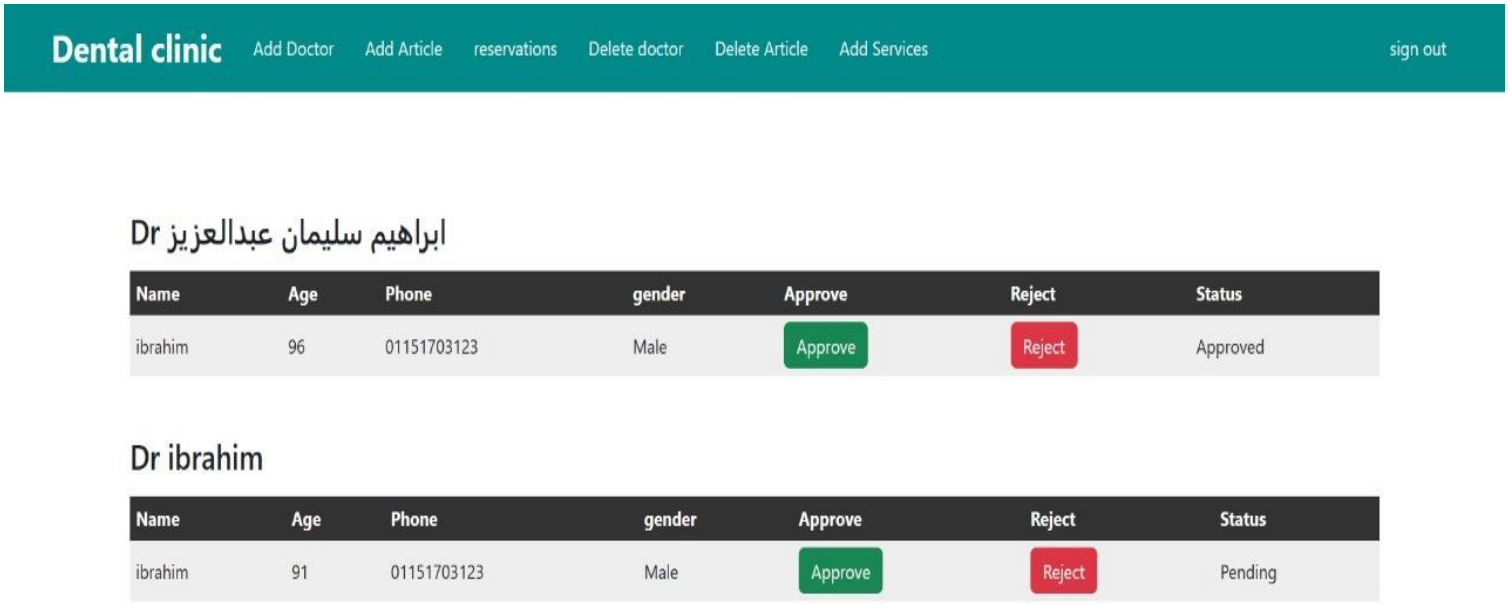
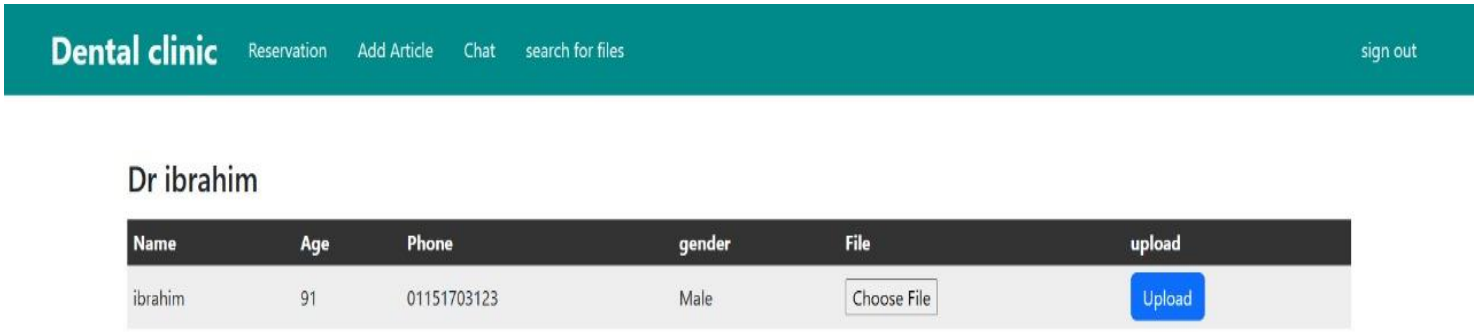



**Figure 4.15 Reservations Admin Design**

# Reservations Doctor Design

On this page, the doctor shows all his reservations about the patient, such as his name, gender, age, etc. He can also delete a specific patient.




Dr ibrahim

| Name | Age | Phone | gender | File | upload |
|---|---|---|---|---|---|
| ibrahim | 91 | 01151703123 | Male | Choose File | Upload |

**Figure 4.16 Reservations Doctor Design**

# Search for patient files by national number Design

In this part, the doctor can search for the patient's files using his national number

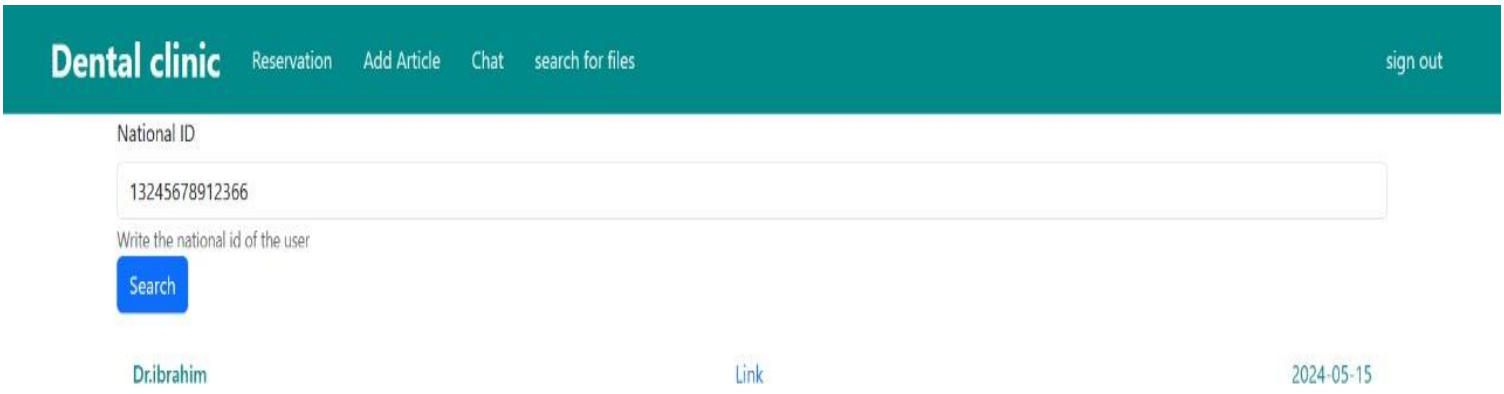



**Figure 4.17 Search for patient files by national number Design**

# Code

## Home Page

```
import React from 'react'
import "./home.css"
import { useNavigate } from "react-router-dom";
import { Col, Container, Row } from 'react-bootstrap'
import homeImage from"../../Assets/pexels-anna-shvets-3845736.jpg"
import { useEffect } from 'react'
export default function Home() {
   const navigate=useNavigate()
   useEffect(()=>{
      if(!localStorage.getItem("userData"))
      {
// navigate("login")
      }

   })
  return (
   <div className='home py-5 vh-100'>
      <Container className='py-5'>
```

```
        <Row className='py-5' >
          <Col md={6}  className='order-sm-0'>


          <div className="home-content">
            <h2 >Teeth <br /> cleaning & <br /> Whitening,<br /> on your time</h2>
            <button className='mt-3 '>book Appointment</button>
          </div>


          </Col>
          <Col md={6} className=' d-md-flex justify-content-center  seconcol'>
          <div className="home-image">
            <img className='w-100' src={homeImage} alt="" />
          </div>
          </Col>
        </Row>
      </Container>
      </div>
  )
}
```

# Login

```
import React from 'react'
import "./login.css"
import { useNavigate } from "react-router-dom";
import { useState } from 'react'
export default function Login() {
  const [email,setEmail]=useState("")
  const [password,setPassword]=useState("")
 const navigate=useNavigate()
  function HandleLogin(e){
    e.preventDefault()
    let
userData=JSON.stringify({email:email,password:password})

window.localStorage.setItem("userData",userData)
console.log(userData);
navigate("/")
  }
  return (

    <div className="login vh-100  ">
```

```
      <div className="container d-flex   justify-content-center
align-items-center h-100">
          <div className="block text-center py-4 px-4">
          <h2>Welcome Back</h2>
          <form action="" className='d-flex flex-column'
onSubmit={HandleLogin}>
          <input onChange={(e)=>setEmail(e.target.value)}
type="email" className= '  d-block  my-4 px-5 py-3 inp'
name="email" id="email" placeholder='Email' />
          <input onChange={(e)=>setPassword(e.target.value)}
type="password" className=' d-block  my-4 px-5 py-3 inp'
name="password" id="password" placeholder='Password' />
          <button type='submit'  className='my-3'>Log
in</button>
          </form>
          {/* <a href='/' className='px-3 mt-2 mx-2 w-25  '>sign
up</a>
          <a href='/' className='px-3 mt-2 mx-2  w-50  '>forgot
password</a> */}
          </div>
      </div>
   </div>
  )
}
```

# Register

```
import React from 'react'
// import "./login.css"
import {  Link, useNavigate } from "react-router-dom";
import { useState } from 'react'
import Joi from 'joi';
import { Alert } from 'react-bootstrap';
export default function Register() {
  const [info,setInfo]=useState({
    name: "",
    email:"",
    phone:"",
    password:"",
    id_number:"",
    address:"",
    gender:1,
    role:"user"
  })
  const [error,setError]=useState("")

  const [errorList,setErrorList]=useState([])
```

```
 const [loading,setLoading]=useState(false)
 const navigate=useNavigate()
 function handelValidation(e){
  const schema=Joi.object({

email:Joi.string().email({minDomainSegments:2,tlds:{allow:['com','net']}}).min(14),
  password:Joi.string().min(8),
  name:Joi.string().min(2),
  phone:Joi.string().min(11),
  id_number:Joi.string().max(14),
  address:Joi.string().max(14),
  gender:Joi.number().max(1),
  role:Joi.string(),



  })
  return schema.validate(info,{abortEarly:false})
   }
 const handelChange=(e)=>{
  let copy={...info}

    copy[e.target.name]=e.target.value
```

```
  setInfo(copy)
  // console.log(e.target.name);
}
 async function  HandleLogin(e){
  setErrorList([])
    e.preventDefault()
    // console.log(info);
    setLoading(true)
    let validate=handelValidation()
    if(!validate.error)
    {

      const headers = {
        "Content-Type": "application/json",
        "Accept": "application/json",
      };
      try{
        setLoading(true)
        let login=await fetch('http://aidoctortest.visooft-
code.com/api/auth/register',{
          method: "POST",
          headers,
```

```
      body: JSON.stringify(info),
    }).then(response => response.json())
    console.log("cdcd",login);
    if(login.errors){

      setError(Object.values(login.errors));
    }
    setLoading(false)
   }
   catch(err){
    console.log('Error: '+ err);

    // setError(response.message)
   }
  }
  else {
   setErrorList(validate.error.details )
  }
setLoading(false)
 }

 return (
```

```
   <div className="login    d-flex justify-content-center   ">
      <div className=" container     row   justify-content-center
align-items-center w-100 h-100   ml-1  " >
         <div className="block text-center    rounded-bottom
py-4 px-4 col-12">


         <h2 className='pt-4'>Create Account</h2>


      {error?error.map(err=><Alert key={err[0]}
variant='danger'>{err[0]}</Alert>


     ):null}
      {errorList?errorList.map(err=><Alert key={err.message}
variant='danger'>{err.message}</Alert>):null}
         <form  className='d-flex flex-column'
onSubmit={HandleLogin}>


         <input onChange={handelChange}  type="text"
className= '  d-block  my-4 px-5 py-3 inp' name="name"
id="name" placeholder='Name' />
         <input onChange={handelChange}  type="email"
className= '  d-block  my-4 px-5 py-3 inp' name="email"
id="email" placeholder='Email' />
```

```
        <input onChange={handelChange}  type="text"
className= '  d-block  my-4 px-5 py-3 inp' name="phone"
id="phone" placeholder='phone' />

        <input onChange={handelChange}  type="text"
className= '  d-block  my-4 px-5 py-3 inp' name="id_number"
id="id_number" placeholder='Id' />

        <input onChange={handelChange}  type="text"
className= '  d-block  my-4 px-5 py-3 inp' name="address"
id="address" placeholder='Address' />

        <input onChange={handelChange}  type="password"
className=' d-block  my-4 px-5 py-3 inp' name="password"
id="password" placeholder='Password' />

        <button type='submit'    className='my-3'>{loading?<i
className="fa-solid fa-rotate-right fa-spin"></i>:'Sign
Up'}</button>

        </form>

         <Link to='/login' className='px-3 mt-2 mx-2 w-25  '
>Login</Link>

        <a href='/' className='px-3 mt-2 mx-2  w-50  '>Forgot
password</a>

        </div>

      </div>

   </div>

  )

}
```

# Add Services

```
import React, { useEffect, useState } from 'react'
export default function AddServices() {
  const [loading,setLoading]=useState(false)
  const [services, setServices] = useState({
    name:"",
    description:"",
    slug:'',
    image:null
  })
  const handelChange=(e)=>{
    let copy={...services}
    if(e.target.name==="image"){
      copy[e.target.name]=  e.target.files[0]
    }
    else{
      copy[e.target.name]=e.target.value

    }
    setServices(copy)
    // console.log(e.target.name);
```

```
  }
  const handelSubmit = async (e) => {
    e.preventDefault();
    setLoading(true)
    console.log(services);
    const body = new FormData();
    body.append('name', services.name);
    body.append('description', services.description);
    body.append('slug', services.slug);
    body.append('image', services.image);

    const headers = {

      "Accept": "application/json",
  };
  console.log("ewe",body);
    try {

      const send = await fetch('https://aidoctortest.visooft-
code.com/api/categories',{
        method: "POST",
        headers,
        body,
```

```
  }).then(response => response.json());


    console.log(send);
  } catch (error) {
    console.error('Error:', error);
  }
  setLoading(false)
 };

 return (
   <>

   <div className='admin    '>
    <div className="container pt-4">
    <div className="  d-flex justify-content-center   ">
     <div className=" container     row   justify-content-center
align-items-center w-100 h-100   ml-1  " >
        <div className="block text-center    rounded-bottom
pt-5 px-3 col-12">

        <h2 className=' '>Add Services</h2>
        <form  className='d-flex flex-column'
onSubmit={handelSubmit}   >
```

```
        <input onChange={handelChange}    type="text"
className=' d-block  my-4 px-5 py-3 inp' name="name"
id="name" placeholder='Name' />

        <input onChange={handelChange}   type="text"
className=' d-block  my-4 px-5 py-3 inp' name="description"
id="description" placeholder='Description' />

        <input onChange={handelChange}   type="text"
classNam= '  d-block  my-4 px-5 py-3 inp' name="slug"
id="slug" placeholder='Slug' />

        <input onChange={handelChange}    type="file"
classNam=' d-block  my-4 w-100 ps-5 py-3 inp' name="image"
id="image" placeholder='image' />

        <button type='submit'  className='my-4 '>{loading?<i
classNam="fa-solid fa-rotate-right fa-
spin"></i>:'Add'}</button>

        </form>

        </div>

      </div>

   </div>

    </div>

   </div>

    </>

  )

}
```

# Chapter Four

# “Implementation”

# AI Model

## AI Models For Mouth and Oral Diseases with Dataset

Computational diagnostics receives helpful assistance from the artificial intelligence subfield of machine learning (ML). Machine learning employs historical data to train algorithms and allows them to "infer" (make predictions about) data they have never seen before. Deep learning, a machine learning technology, has found applications in numerous fields . The medical and agricultural industries, along with facial expression recognition, intelligent equipment, and language translation, are just a few examples. we propose the Conventional neural network model InceptionV3 for the classification of CAL, DC, HYPO, MU, OC, GING, and TD labels affecting the mouth and oral cavity. When compared to previous models, the suggested InceptionV3 achieves better results. CAL, DC, HYPO, MU, OC, GING, and TD are all part of the self-created Mouth and Oral Disease dataset used to train the InceptionV3's transfer learning architecture. The most important results of this study are: 1.Mouth and Oral Diseases Classification using the InceptionV3 method to recognizes the CAL, DC, HYPO, MU, OC, GING, and TD diseases. 2.The performance of the proposed method is enhanced in terms of accuracy than the existing models. 3.A new dataset of Mouth and Oral Diseases such as CAL, DC, HYPO, MU, OC, GING, and TD has been developed.

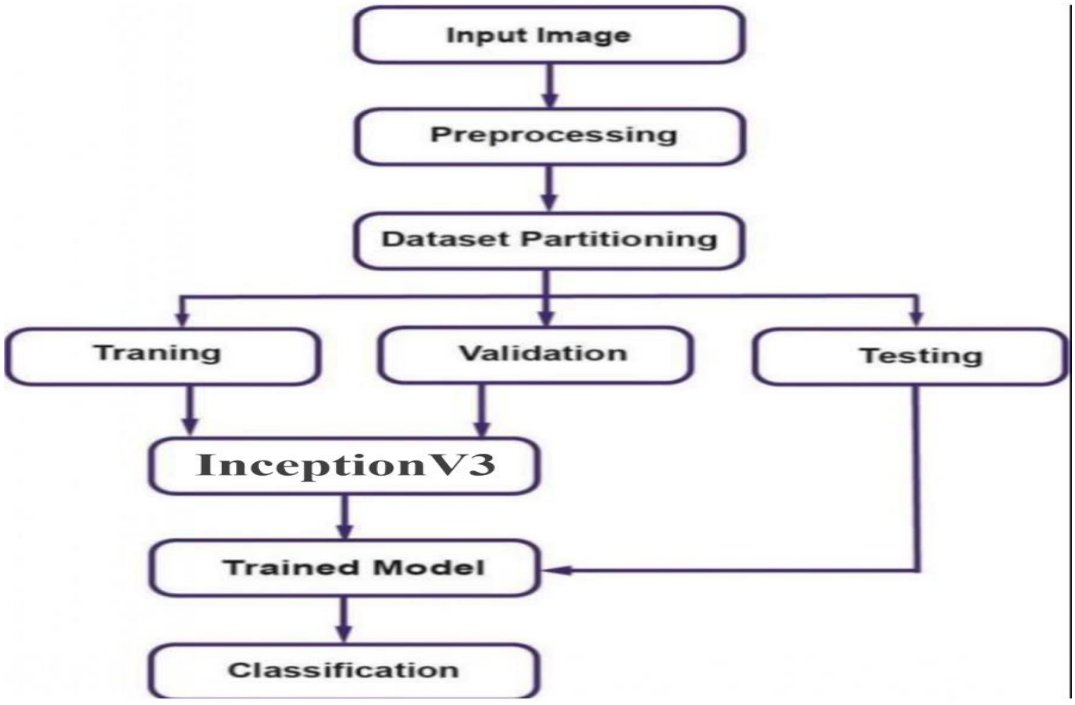
Gingivitis can be diagnosed through the use of periodontal probing depth (PPD) measurements, the presence of bleeding during probing, and a radiographic evaluation of alveolar bone loss . Because of the wide variety of probes, patients may experience discomfort or even agony during these operations . It requires considerable time and energy to carry out such examinations. Although 72% of patients looking for a new dentist rely on online reviews , many dentists still have the opportunity to improve their skillset. Of individuals between the ages of 20 and 64, 91% had at least one cavity, and 27% of adults had decay so severe that they lost a tooth . Mouth and Oral Diseases Classification Using Multiple Methods was established in the current study to identify diseases such as caries, calculus, gingivitis, tooth discoloration, ulcers, and hypodontia. The new collection, termed "Mouth and Oral Diseases" , comprises seven distinct categories of data. Compared to state-of-the-art approaches, the proposed model's 94.02% accuracy is significantly higher

## Goals of our AI Model

Early diagnosis of oral and dental diseases is an important and essential factor in developing methods to protect dental health The main goal of the model is to help the medical and dental staff reduce the time spent between diagnosis and treatment, so doctors and nurses can resume work and begin reporting to the patient quickly and accurately. The early detection and diagnosis of oral, dental, and mouth disorders is where the dental industry can profit from using machine learning techniques. Intraoral imaging is useful for the early diagnosis of periodontitis as well. While these studies provide useful information, they do have certain drawbacks, such as a bias towards people in the more advanced stages of diseases such as periodontitis .

## Iteration method of Model

Millions of images are utilized for training transfer learning algorithms. Our project primarily utilized models based on transfer learning. These models are learned through transfer learning. These models' adaptability to new information results from an extensive training on a large corpus. The features gained from a larger dataset can be highly helpful when applied to an issue with a smaller dataset. Training a model from scratch is avoided more easily.



## Our Oral Diseases Dataset

The Dental Condition Dataset is a comprehensive collection of images specifically curated for dental research and analysis. This dataset encompasses a wide range of dental conditions, including caries, calculus, gingivitis, tooth discoloration, ulcers, and hypodontia. It serves as a valuable resource for dental professionals, researchers, and machine learning enthusiasts interested in developing and training models for dental condition detection and classification.

## Type of our dataset

Image Dataset: It includes a dataset consisting of images. This is mostly used to differentiate the types of Oral diseases, and so on.

## Features of a Dataset :

The features of a dataset may allude to the columns available in the dataset. The features of a dataset are the most critical aspect of the dataset, as based on the features of each available data point, will there be any possibility of deploying models to find the output to predict the features of any new data point that may be added to the dataset.

It is only possible to determine the standard features from some datasets since their functionalities and data would be completely different when compared to other datasets. We can say that our dataset features similar to :

## Dataset Details :

**Image Categories:** The dataset consists of carefully annotated images classified into various dental conditions:

**Caries:** Images showing tooth decay, cavities, or carious lesions.

**Calculus:** Images depicting dental calculus or tartar buildup on teeth.

**Gingivitis:** Images displaying inflamed or infected gums.

**Tooth Discoloration:** Images showcasing tooth discoloration or staining.

**Ulcers:** Images exhibiting oral ulcers or canker sores.

**Hypodontia:** Images representing the condition of missing one or more teeth.

**Image Sources:** The dataset is a compilation of images sourced from multiple hospitals and reputable dental websites. These sources ensure the diversity and authenticity of the dental conditions depicted in the dataset.

**Number of Samples :** The total number of samples and a total number of labels for each category are also listed in the table

| Class labels | Images |
|---|---|
| Calculus | 1296 |
| Data caries | 2601 |
| Gingivitis | 2349 |
| hypodontia | 1251 |
| Mouth Ulcer | 2806 |
| Tooth Discoloration | 2017 |

## Examples of our Datasets :

| 1) Calculus | 2) Data Caries |
|---|---|
|  |  |
| 3) Gingivitis | 4) Hypodontia |
|  |  |
| 5) Mouth Ulcer | 6) Tooth Discoloration |
|  |  |

## Image Resizing

The dataset is expanded in Python to a size of 1 by 255. This will drastically reduce the processing time at the expense of the model's accuracy. this standardize the pixel values, making the optimization process more stable.

## Data preprocessing and augmentation

To prevent over-fitting and expand the training set's variety, we used the picture data generator function from the Keras package in Python. The computer's performance would improve if the variation in pixel values were reduced. By default, the parameter value (1. /255) restricts pixel values to the range [0,1]. The photos were flipped to a target orientation of 40 degrees. These parameters allow randomly shifting the width and height of the images by a maximum fraction of 20%. It applies shear transformations to the images with a maximum shear intensity of 20%. This parameter randomly flips the images horizontally. Horizontal flipping can help invariance to left-right orientation, which is often useful in object recognition tasks. It randomly zooms into the images by a maximum factor of 20%. This augmentation can simulate the effect of objects appearing closer or farther away in the image.

## Annotation and Augmentation

Each image in the dataset is meticulously annotated with bounding boxes, accurately delineating the specific dental condition present. Furthermore, data augmentation techniques have been applied to enhance the dataset's diversity and generalization capabilities, including rotation, flipping, scaling, and noise addition.

## Used data Augmentation techniques

| Transformations | setting |
|---|---|
| **Scaling** | Ranged from 0 to 1 |
| **Rotation** | 40o |
| **Width shift** | 0.2 |
| **Height shift** | 0.2 |
| **Shear** | 20o |
| **Horizontal flipping** | True |
| **Zooming** | 0.2 |

## Example from our model code:

The data preprocessing in this code snippet involves rescaling the pixel values of the images. This is achieved by dividing each pixel value by 255, which effectively scales them to the range between 0 and 1. This preprocessing step is crucial for deep learning tasks, particularly with image data, as it aids in faster convergence during training and helps in normalizing the input data, making it easier for the model to learn meaningful patterns from the images.

```
#Instancing from 'ImageDataGenerator' object.
generator = ImageDataGenerator(rescale=1/255,
                               rotation_range=40,
                               width_shift_range=0.2,
                               height_shift_range=0.2,
                               shear_range=0.2,
                               horizontal_flip=True,
                               zoom_range=0.2)
```

## Dataset splitting

The entire dataset was used to generate three distinct data sets: training, validation, and testing. InceptionV3 was first trained on the training dataset, and then its accuracy and efficiency were determined using the validation and test datasets. We used a 80:10:10 split for our data, allocating 79.97% to training and 10% each to validation and testing. As can known 11653 images were trained, validated, and tested using the dataset. CAL, DC, HYPO, MU, OC, GING, and TD labels were used on 80% of the photos in this study's model introduction. Another 20% of new images were used for testing and validation on the  dataset, with each stage receiving 20% of the total. The suggested InceptionV3 method for classifying images of the mouth and oral disorders produced highly accurate predictions for all of the dataset's labels.

## Chosen model

The Inception V3 is a deep learning model based on Convolutional Neural Networks, which is used for image classification. The inception V3 is a superior version of the basic model Inception V1 which was introduced as GoogLeNet in 2014. As the name suggests it was developed by a team at Google.

The inception v3 model was released in the year 2015, it has a total of 42 layers and a lower error rate than its predecessors. Let's look at what are the different optimizations that make the inception V3 model better.

## Model features :

### Architecture

Inception V3 builds upon the original Inception architecture introduced in the Inception V1 (GoogLeNet) model. It employs a deep convolutional neural network with multiple layers, including convolutional layers, pooling layers, and fully connected layers.

### Computational Efficiency

Inception V3 focuses on improving computational efficiency while maintaining high accuracy. It achieves this through the use of various techniques such as factorization, aggressive regularization, and auxiliary classifiers.

### Factorization

Inception V3 utilizes factorized convolutions, including 1x1, 3x3, and 5x5 convolutions, to capture features at different spatial scales efficiently.

### Auxiliary Classifiers

The model incorporates auxiliary classifiers at intermediate layers during training. These auxiliary classifiers help combat the vanishing gradient problem and provide additional supervision signals for the earlier layers.

### Pre-trained Models

Google provides pre-trained Inception V3 models trained on large-scale image datasets such as ImageNet. These pre-trained models can be fine-tuned on specific tasks or used as feature extractors for transfer learning.

### Applications

Inception V3 is widely used in various computer vision tasks, including image classification, object detection, and image segmentation. Its efficient architecture makes it suitable for deployment in resource-constrained environments such as mobile devices and embedded systems.

### Performance

Inception V3 achieves state-of-the-art performance on benchmark image classification datasets such as ImageNet, surpassing its predecessor Inception V1 and competing models.

## Major modifications done on the Inception V3 model

1. Factorization into Smaller Convolutions
2. Spatial Factorization into Asymmetric Convolutions
3. Utility of Auxiliary Classifiers
4. Efficient Grid Size Reduction

## Model inside the code

### 1. Loading the Base Model:

- It loads the InceptionV3 model without its classification layer (include_top=False), indicating that we will add our own classification layers on top of it.

- It specifies the input shape as (155, 155, 3), suitable for processing images with a resolution of 155x155 pixels and 3 color channels (RGB).

```
def Create_model(optm,Loss,acc)->None:
    '''function docstring'''
    pre = InceptionV3(include_top=False,input_shape=(155,155,3))
```

### 2. Freezing Layers:

- It freezes all the layers in the loaded InceptionV3 model by setting their (trainable) attribute to (False). This prevents their weights from being updated during training, effectively using them as feature extractors.

```
    for layer in pre.layers :
        layer.trainable = False
```

### 3. Adding Custom Layers:

- It adds custom dense layers on top of the output of a specific layer ('mixed7') from the InceptionV3 model.

- These dense layers gradually reduce the dimensionality of the features extracted by the InceptionV3 model and add non-linearity to the network.

- The final dense layer has a softmax activation function, suitable for multi-class classification tasks with 6 output classes.

```
    last_l = pre.get_layer('mixed7')
    out = last_l.output
```

### 4. Compiling the Model:

- It compiles the model using the Adam optimizer ('adam'), the specified loss function (Loss), and the specified metrics (acc).

- The optimizer, loss function, and metrics are passed as arguments to the function.

```
model = Model(pre.input,x)
model.compile(optimizer='adam',
              loss=Loss,
              metrics=[acc])
```

### 5. Function Call:

- The function is called with specific arguments to create a model instance (alg).

- It uses the RMSprop optimizer with a learning rate of 0.0001, categorical cross-entropy loss function, and accuracy metric.

```
alg = Create_model(RMSprop(learning_rate=0.0001),'categorical_crossentropy','accuracy')
```

## Results and discussion

To train and test the suggested models, we used a kaggle notebook for this purpose, we employed a transfer deep learning model. The Categorical Cross entropy loss function was employed during model creation, and the Adam optimizer was used in all trials with the proposed technique at a learning rate of 0.001. The proposed InceptionV3 used early stopping criteria with the lowest val_loss, and the best val_accuray was used as a saving option; batch size with 20 and 20 epochs was chosen to conduct all the experiments. The proposed method's performance was evaluated on the following:

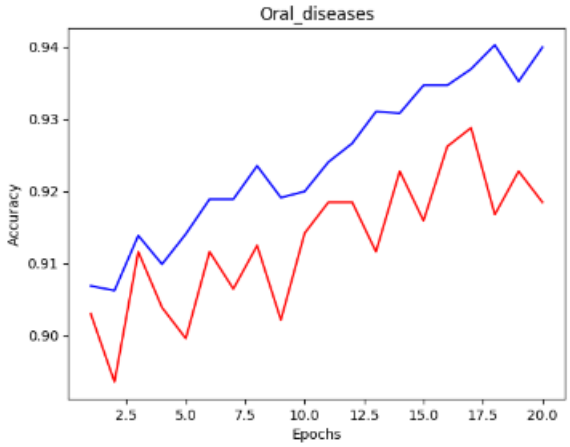
- Mouth and Oral Disease Classification with a Deep Learning Model (InceptionV3) was observed using our dataset to assess its efficacy.
- The experiments were conducted with and without data augmentation to compare the performances for the ablation study of the model.
- The comparison analysis was performed of the proposed model with existing studies.

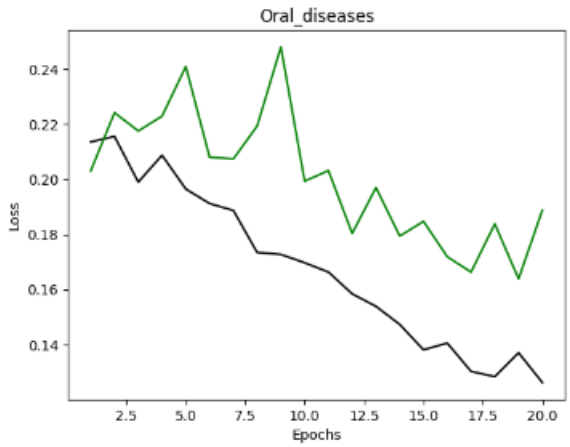
**The performance analysis of the proposed model Oral disease classification using InspectionV3 method**

The proposed Oral Disease Classification utilizing the InceptionV3 method performance was evaluated. , we can see the loss and accuracy of the InceptionV3 model throughout training and validation. Over time, the proposed model's training accuracy went from 91.40 to 94.02%.

```
107/107 [==============================] - 11s 87ms/step - loss: 0.1316 - accuracy: 0.9401 - recall: 0.9358 - precision: 0.94
55
```

InceptionV3's model achieved 90.69% validation accuracy after the first epoch and 94.02% accuracy after the last epoch. The loss in training went down from 0.21 to 0.13 after the last epoch, while the loss in validation went down from 0.20 to 0.18.



**Recall , f1 score , and Accuracy of the proposed InceptionV3 model**

The proposed InceptionV3 model's accuracy, precision, recall, and F1 score across a variety of classes are detailed in Table. Calculus (CAL), Data Caries (DC), Gingivitis (GING), Mouth Ulcer (MU), Tooth Discoloration (TD), and Hypodontia (HYPO) are the categories.

| class | precision | Recall | F1 score | Accuracy |
|---|---|---|---|---|
| CAL | 82% | 82% | 82% | - |
| DC | 99% | 98% | 98% | - |
| GING | 89% | 89% | 89% | - |
| MU | 99% | 97% | 98% | - |
| TD | 93% | 99% | 96% | - |
| HYPO | 98% | 98% | 98% | - |
| Average Accuracy | | | | 94.02% |

In [26]:
```
for i in alg.predict(img[None]):
        print(dic[i.argmax(-1)])
```

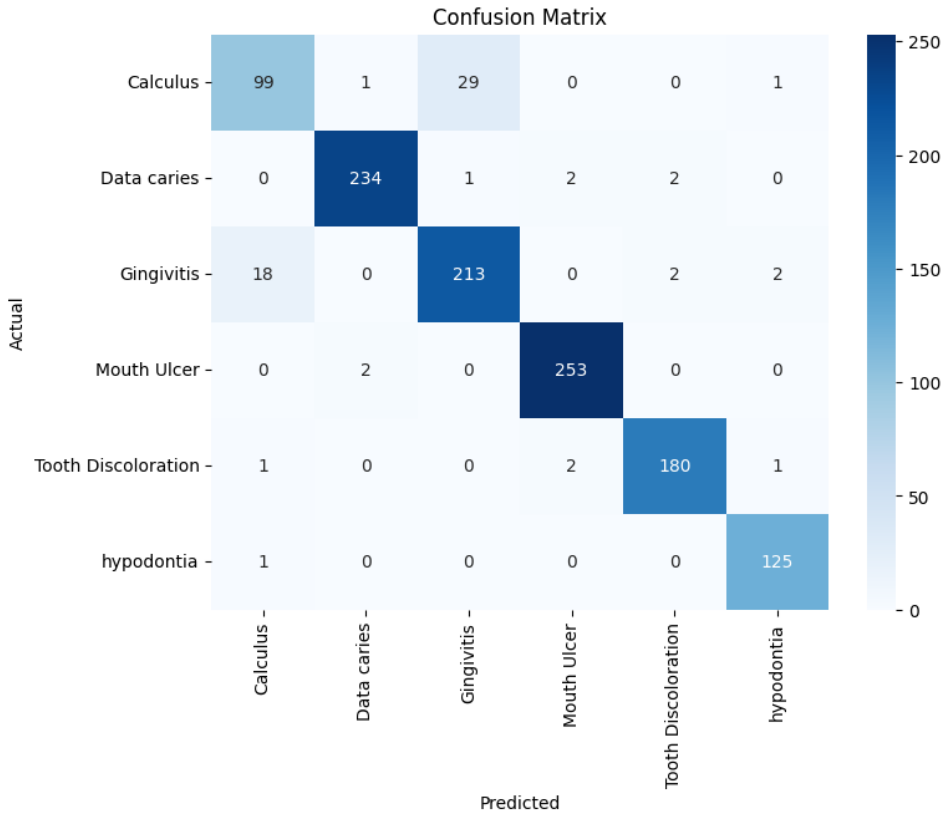
```
1/1 [==============================] - 1s 1s/step
Mouth Ulcer
```

In [27]:
```
for i in alg.predict(img[None]):
   for o in list(reversed(i.argsort( ))):
        print(dic[o])
```

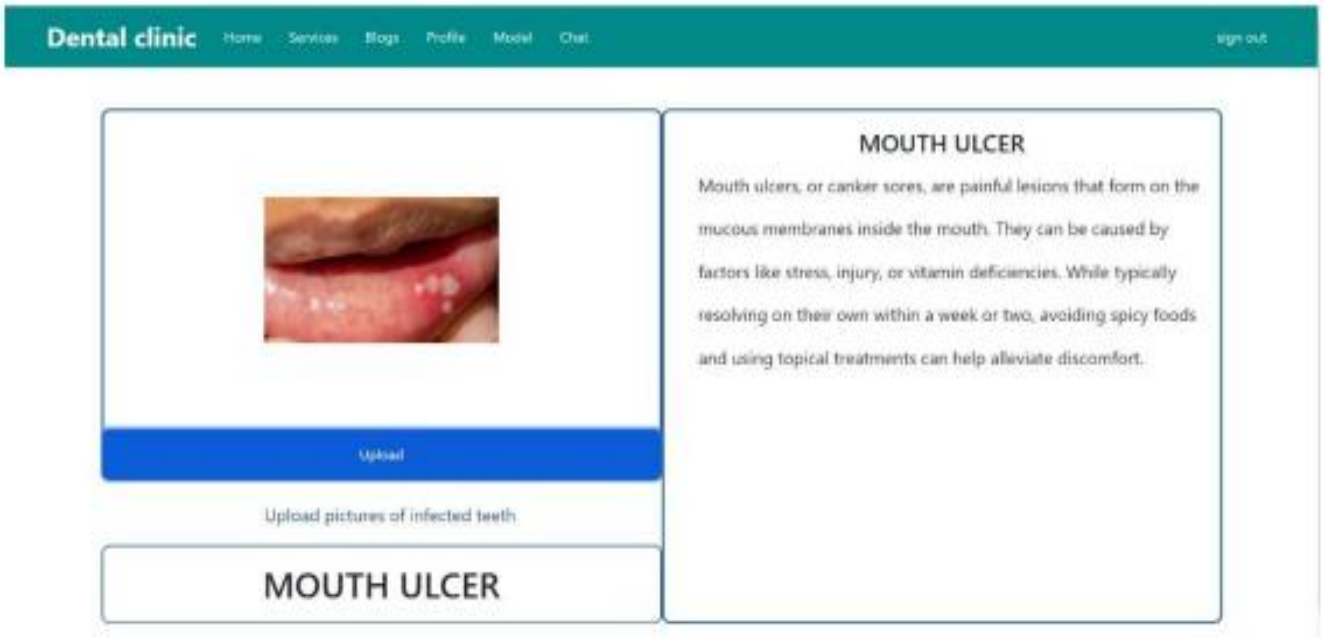
```
1/1 [==============================] - 0s 24ms/step
Mouth Ulcer
Calculus
Tooth Discoloration
Data caries
Gingivitis
hypodontia
```

## The confusion matrix of the proposed InceptionV3 model

The Following figure displays the confusion matrix of the proposed InceptionV3 model on the test set, showcasing the results of applying data augmentation techniques. The matrix illustrates the predicted and actual labels for each class in the classification task. The classes represented in the confusion matrix are Calculus (CAL), Data Caries (DC), Gingivitis (GING), Mouth Ulcer (MU), Tooth Discoloration (TD), and Hypodontia (HYPO) . The values in the confusion matrix indicate the number of instances that were predicted to belong to a particular class, with the actual class labels represented by the rows and the predicted class labels represented by the columns.



## Results on website




# Conclusion and future work

The primary objective of this project is to design and develop a user-friendly, interactive for a dental website.

Enhancing communication with doctors : Helping patients communicate with the doctor via chat to answer questions and help them.

Improved Patient Experience : Enhance the overall experience of patients interacting with the dental website through user-friendly interfaces and personalized features.

Customize AI: Use AI for dental screening by uploading a photo of the condition and diagnosing it.

Delivery of educational content: Develop a platform to provide articles to patients, including information about oral health, treatment options, and preventive care.

Designing a system that provides most of the patients' data and medical history, allowing doctors to view the medical history and treatment reports at any time they want electronically.

it is heartening to see that deep learning models successfully detect calculus (CAL), data caries (DC), gingivitis (GING), mouth ulcer (MU), tooth discoloration (TD), and hypodontia (HYPO). With such a limited sample size for training and testing, it's no surprise that some models perform poorly in instability. The Oral Disorders Our dataset was developed to address this issue and is split into six subcategories. Compared to other models, the 94.02% accuracy of our novel InceptionV3 technique for diagnosing mouth and oral disorders is significantly higher. A wide variety of data-augmentation techniques were employed to further improve accuracy. Suppose we want to improve our ability to detect and diagnose oral and dental disorders early on. In that case, future research should broaden the scope to cover additional diseases on a wider and more generalizable dataset.

# References

**1. Larissa Hirsch M (2018) KidsHealth / Parents / Mouth and TeethMouth and Teeth**

**2. Gezgin O, Botsali M (2018) Evaluation of teeth development in unilateral cleft lip and palate patients in mixed dentition by using medical image control systems. Niger J Clin Pract 21:156–162**

**3. Aberin STA, de Goma JC (2018) Detecting periodontal disease using convolutional neural networks. 2018 IEEE 10th international conference on humanoid, nanotechnology, information technology, communication and control, environment and management (HNICEM). IEEE, pp 1–6**

**4. Rana A, Yauney G, Wong LC, Gupta O, Muftu A, Shah P (2017) Automated segmentation of gingival diseases from oral images. 2017 IEEE Healthcare Innovations and Point of Care Technologies (HI-POCT). IEEE, pp 144–147**

**5. Watanabe H, Ariji Y, Fukuda M, Kuwada C, Kise Y, Nozawa M, Sugita Y, Ariji E (2021) Deep learning object detection of maxillary cyst-like lesions on panoramic radiographs: preliminary study. Oral Radiol 37:487–493**

**6. Abdalla-Aslan R, Yeshua T, Kabla D, Leichter I, Nadler C (2020) An artificial intelligence system using machine-learning for automatic detection and classification of dental restorations in panoramic radiography. Oral Surg Oral Med Oral Pathol Oral Radiol 130:593–602**

**7. system component of web application architecture**

**https://www.brisklogic.co/the-components-of-web-application-architecture/**

**8. Entity relationship diagram**

**https://app.diagrams.net/**

**9. American Dental Association (ADA). (2022). [American Dental Association]**
**https://www.ada.org/**

**10. World Health Organization (WHO). (2022). [Oral health]**

**https://www.who.int/health-topics/oral-health**

**11. Centers for Disease Control and Prevention (CDC). (2022). [Division of Oral Health]**

**https://www.cdc.gov/oralhealth/index.html**

**12. American Association of Dental Schools (AADS). (2022). [American Association of Dental Schools]**

**https://www.adea.org/**

**13. Accreditation Association for Ambulatory Health Care (AAAHC). (2022). [Accreditation Association for Ambulatory Health Care]**

**https://www.aaahc.org/**