# MediCare

## LEVEL_4 [2025]

## CS Department

Supervised by :

Dr. Seham Elaw Amer

# Project Team Members

| Name | Role | Department |
|---|---|---|
| Abdulrahman Abdel Samee | Back-End | CS |
| Mohamed Ahmed Ali | Back-End | CS |
| Nehal Tarek | Back-End | CS |
| Nada Mohamed | Front-End | CS |
| Mayada Mohamed | Front-End | CS |
| Abdallah Adel | UI/UX | CS |
| Ahmed Mansour | Mobile Developer | CS |
| Ali Maher Mohamed | Mobile Developer | CS |

# Contents

# Figures

## List of Figures

# Tables

## List of Tables

# Abbreviations

| Symbol | Expression |
|---|---|
| HTML | Hypertext Markup Language |
| CSS | Cascading Style Sheets |
| ERD | Entity Relation Diagram |
| DFD | Data Flow Diagram |
| SDLC | Software Development Life cycle |

# Acknowledgment

First and foremost, we would express our deepest gratitude and appreciation to our Advisor Dr. Seham for her support, outstanding guidance, and encouragement throughout our graduation project.

As well as Engineer. Dr.Elnomery Zanaty for his continuous support and help as he was always there for us providing more than the needed time and effort from him since the very beginning of our project.

We would like to thank our families, especially our parents. We hope they are proud of us in our last year of education, we hope that we will start giving back to the community very soon! Thank you for supporting us and providing us with the needed time, eort, encouragement, patience, and assistance over the years.

Finally, our faculty for providing us with the courses that guided us in the right direction in our lives and the help of all the professors that left a great impact on our lives,

Thank you.

# Chapter 1: Introduction

In this chapter, we are going to discuss and go deeper into the overview of the project and know more about its scope and limitations and explain some terminologies we will find throughout the document.

# Introduction

**MediCare**, is an application and website designed to enhance the healthcare experience by integrating a variety of medical services into a single, comprehensive platform. The application allows users to easily book doctor appointments, view their specializations, and access a transparent rating system that helps them make well-informed decisions. The project prioritizes credibility, ensuring that all information about doctors is verified for a secure and reliable experience. By leveraging modern technologies and a streamlined design, MediCare provides an innovative solution that meets user needs and improves the quality of healthcare services.

# Chapter 2: Problem Definition

In this chapter, we will explore the core problem
that the project aims to address.
We will delve into the challenges and issues faced
in medical field that led to the development
of this application. .

# Problem Definition

Accessing reliable healthcare services remains a challenge for many individuals due to fragmented systems, lack of verified information, and limited accessibility to medical professionals. Patients often struggle to find accurate information about doctors, such as their specializations, availability, or credibility. Furthermore, there is no centralized platform that offers a seamless and trustworthy way to book appointments, check doctor ratings, and manage healthcare needs efficiently. These challenges can lead to delays in treatment, frustration for patients, and inefficiencies in the healthcare system.

# Issues and Objectives

| Issue | | Weight |
|---|---|---|
| 1. | Lack of a unified platform that consolidates essential healthcare features like booking and reviews. | 10 |
| 2. | Difficulty in accessing verified and accurate information about doctors | 9 |
| 3. | Limited tools for efficient and convenient healthcare management. | 8 |

| Objectives | |
|---|---|
| 1. | Ensure all information displayed about doctors is verified and credible to enhance user trust. |
| 2. | Implement a transparent and user-friendly rating system to help patients make informed decisions. |
| 3. | Provide an intuitive, technology-driven solution that streamlines healthcare access and improves user satisfaction. |

# Requirements and constrains

## Requirements

1. User Registration and Authentication:.
2. Doctor Information Management.
3. Appointment Booking System.
4. Search and Filter Functionality.
5. Administrative Features.

## Constraints

1. Ensure compatibility with major operating systems (Windows, macOS, Android, iOS) and browsers
2. Manual Verification: Doctor credential verification may require manual processes, leading to potential delays.
3. Internet Dependence: The system requires reliable internet connectivity..
4. Medical Expertise: Limited availability of medical professionals for consultation, feedback, and testing..

# Chapter 3:Project Analysis

In this section,

we will analyze the proposed solution in terms of its feasibility, objectives, and expected outcomes.

We will examine the requirements of the project

and evaluate the different approaches

# Project Analysis

## ➢ What Is Agile?

The Agile methodology is a project management approach that involves breaking the project into phases and emphasizes continuous collaboration and improvement. Teams follow a cycle of planning, executing, and evaluating.

## ➢ Why Choose Agile?

Using **Agile** in Our project, especially in software development, can offer several benefits that align with the dynamic nature of modern projects, especially in the context of our **MediCare** application. Here's why Agile would be a great fit:

### 1. Flexibility and Adaptability

- **Rapid Changes:** In healthcare, requirements can change quickly, whether due to regulatory changes, new technologies, or user feedback. Agile allows you to adapt to these changes without derailing the entire project.
- **Iterative Improvements:** You can continuously improve the features of the MediCare app as you receive feedback from users, stakeholders, or testing phases.

### 2. Customer-Centric Approach

- **Continuous Feedback:** Agile emphasizes frequent collaboration with stakeholders (such as doctors, patients, or healthcare professionals), ensuring that the app meets their real-time needs and expectations. This is crucial for a project like MediCare, where user experience and accuracy are key.
- **Faster Delivery of Features:** With Agile, you can prioritize the most important features (such as doctor ratings, appointment scheduling) and deliver them incrementally, ensuring early value to users.

### 3. Improved Quality

- **Testing and Feedback Loops:** Agile encourages continuous testing throughout the development cycle, helping identify issues early on and maintain high quality in your project. For an app in the medical field, where reliability and accuracy are paramount, this is especially beneficial.
- **Refinement of Features:** Each iteration or sprint provides an opportunity to refine features, ensuring that they meet both user and technical requirements.

### 4. Enhanced Collaboration

- **Cross-functional Teamwork:** Agile promotes teamwork among developers, designers, product managers, and other stakeholders. This improves communication and ensures that all aspects of the MediCare project (technical, design, usability) are aligned.
- **Transparency:** Agile methodologies encourage regular meetings (like sprint reviews and stand-ups), making the development process transparent to all team members and stakeholders, which is vital in a collaborative project like MediCare.

### 5. Risk Management

- **Early Identification of Risks:** Agile's iterative approach helps to identify potential risks early in the project, allowing you to address them before they become significant issues.
- **Frequent Releases:** Each sprint produces a working version of the product, so you can catch potential problems early in the development cycle and make necessary adjustments.

### 6. Faster Time to Market

- **Quick Iterations:** Agile enables you to deliver a basic version of the MediCare app quickly (e.g., basic appointment scheduling or doctor information), allowing you to get valuable user feedback sooner and make improvements in the next cycle.
- **Customer Value First:** By breaking down the project into manageable pieces and releasing them incrementally, Agile ensures that users get the most critical features first.

### 7. Continuous Improvement

- **Post-Release Enhancements:** Once the basic functionalities of the app are delivered, Agile allows for continuous improvement through new sprints, ensuring that the MediCare app evolves based on user needs and market demands.

Conclusion:

For **MediCare** project, Agile helps ensure that the development process remains flexible, user-focused, and efficient. It allows for rapid adjustments, minimizes risks, and delivers valuable features early, which is especially beneficial in the fast-paced and constantly evolving healthcare domain.

## ➢ Stages of Agile

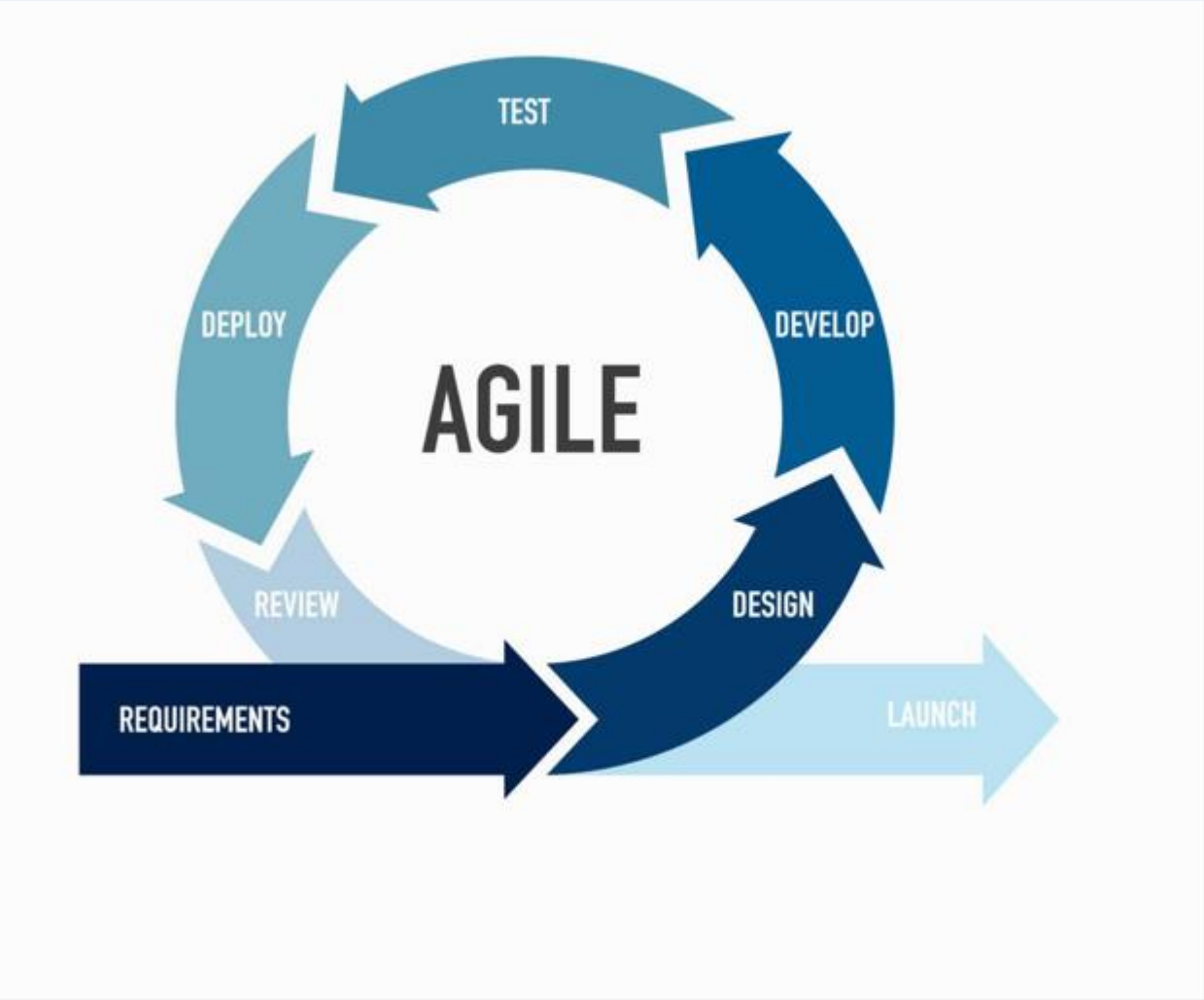



**Figure 3.1 Agile Development**

**Requirement** → In Agile, you begin by identifying the key features and functionalities, much like the requirement phase. This aligns with the Sprint Planning phase, where you plan what to deliver in each sprint, breaking down the tasks into user stories.

**Design** → The design phase can overlap with development in Agile, as design decisions are continuously made during the development process. During sprints, you focus on designing and iterating the user interface and architecture, and developers work on coding the features.

**Development** → The development phase is where the main work happens. In Agile, this is the Sprint Execution phase, where developers implement features and continuously test them.

**Testing** → In Agile, testing happens continuously throughout the sprints. You can think of your testing phase as part of Sprint Review and ongoing testing to ensure the application works as expected.

**Deployment** → Deployment aligns with the Release phase in Agile, where features and updates are deployed in smaller increments. After testing and approvals, the application is released or deployed to users.

**Review** → Finally, the Review phase is part of the Sprint Review and Retrospective. This is where you evaluate what went well, what can be improved, and plan the next steps for future sprints.

# Scenario

## ➢ Use Case Scenario

| | |
|---|---|
| Use case name: MediCare System. | Unique ID: UC-BD-01 |
| Area: : MediCare System - End-to-End Functionality. | |
| Actor(s): Patients, Doctors, Admins. | |
| Stakeholders: Patients, Doctors, Admins. | |
| Description : MediCare system provides features for patients to search for doctors, hospitals, pharmacies, and labs; book appointments; and interact with the system via various functionalities. Doctors manage appointments, blogs, and offers, while admins oversee the entire system. | |
| Triggering Event: When a user logs into the system to perform actions like searching, booking, or managing resources. | |
| Trigger type: External , Temporal. | |
| **Patient Use Cases** | |
| **Steps performed (Main Path)** | **Information for Steps** |
| 1. The patient registers on the system. | Fill out the registration form with name, contact, email, password, and more. |
| 2. The patient logs into the system. | Enter email and password on the login page. |
| 3. The patient searches for doctors, hospitals, pharmacies, or labs. | Uses filters (specialty, price, location) to refine results. |
| 4. The patient views search results. | Details include names, locations, services, reviews, prices, and ratings. |
| 5. The patient selects a doctor and books an appointment. | Select time slots and confirms the appointment. Payment is made online or at the clinic. |
| 6. The patient views hospital or care center information. | Information includes services, departments, consultation fees, and operating hours. |
| 7. The patient rates a pharmacy or lab after service. | Rates from 1 to 5 stars. The average rating is updated. |
| **Doctor use cases** | |
| 1 . The doctor registers on the system. | Fill out the registration form with name, contact details, specialty, email, and password. |

| | |
|---|---|
| 2. The doctor logs into the system. | Enter email and password on the login page. |
| 3. The doctor views their dashboard. | Views upcoming appointments, patient details, and schedule summary. |
| 4. The doctor manages appointments. | Updates appointment statuses (e.g., completed, rescheduled, or canceled). |
| 5. The doctor creates blogs. | Adds title, content, and optional links, then publishes the blog for public viewing. |
| 6. The doctor manages offers. | Creates, edits, or deletes promotional offers such as discounts or packages with start/end dates. |
| 7. The doctor updates clinic information. | Edits or adds clinic details such as address, contact info, working hours, and consultation fees. |
| 8. The doctor reviews patient feedback. | Views and responds to ratings or feedback to improve services. |
| **Admin Use Cases Summary** | |
| 1. The admin logs into the system. | logs in with email and password to access the admin dashboard. |
| 2. The admin manages users. | views, edits, or deletes user accounts (patients and doctors). |
| 3. The admin manages doctors and clinics. | adds or updates doctor profiles and clinic information. |
| 4. The admin manages hospitals, pharmacies, and labs. | adds or updates hospitals, pharmacies, and lab details. |
| 5. The admin manages content. | approves, edits, or deletes blogs and promotional offers. |
| 6. The admin manages offers and discounts. | oversees doctor-created offers, ensuring accuracy and validity. |
| 7. The admin monitors appointments and bookings. | views and manages appointments, resolving conflicts or cancellations. |
| 8. The admin manages feedback and ratings. | monitors and resolves issues with reviews and ratings. |
| 9. The admin generates system reports. | generates reports on system usage, activities, and payments. |
| 10. The admin manages media. | Admin uploads and organizes media content related to doctors, clinics, and offers. |

Postconditions

• Patients successfully book appointments, search for doctors, and access detailed healthcare information.

• Doctors maintain accurate schedules and manage offers effectively.

• Admins ensure system reliability, secure transactions, and up-to-date records for smooth operation

**Assumptions**

- All users have stable internet access for seamless interaction with the system.
- The system incorporates a secure and efficient payment gateway.
- Healthcare providers (doctors, hospitals, and labs) regularly update their schedules and availability

**Questions**

1. Should the system send automated appointment reminders via email and/or SMS for better user engagement?
2. Should patients have the ability to cancel appointments independently without requiring admin approval?

# Diagrams

## Analysis Diagrams

- ✓ Use Case Diagram
- ✓ Context Diagram
- ✓ Data Flow Diagram
- ✓ Entity-Relationship Diagram
- ✓ Mapping

## ➢ Use Case Diagram

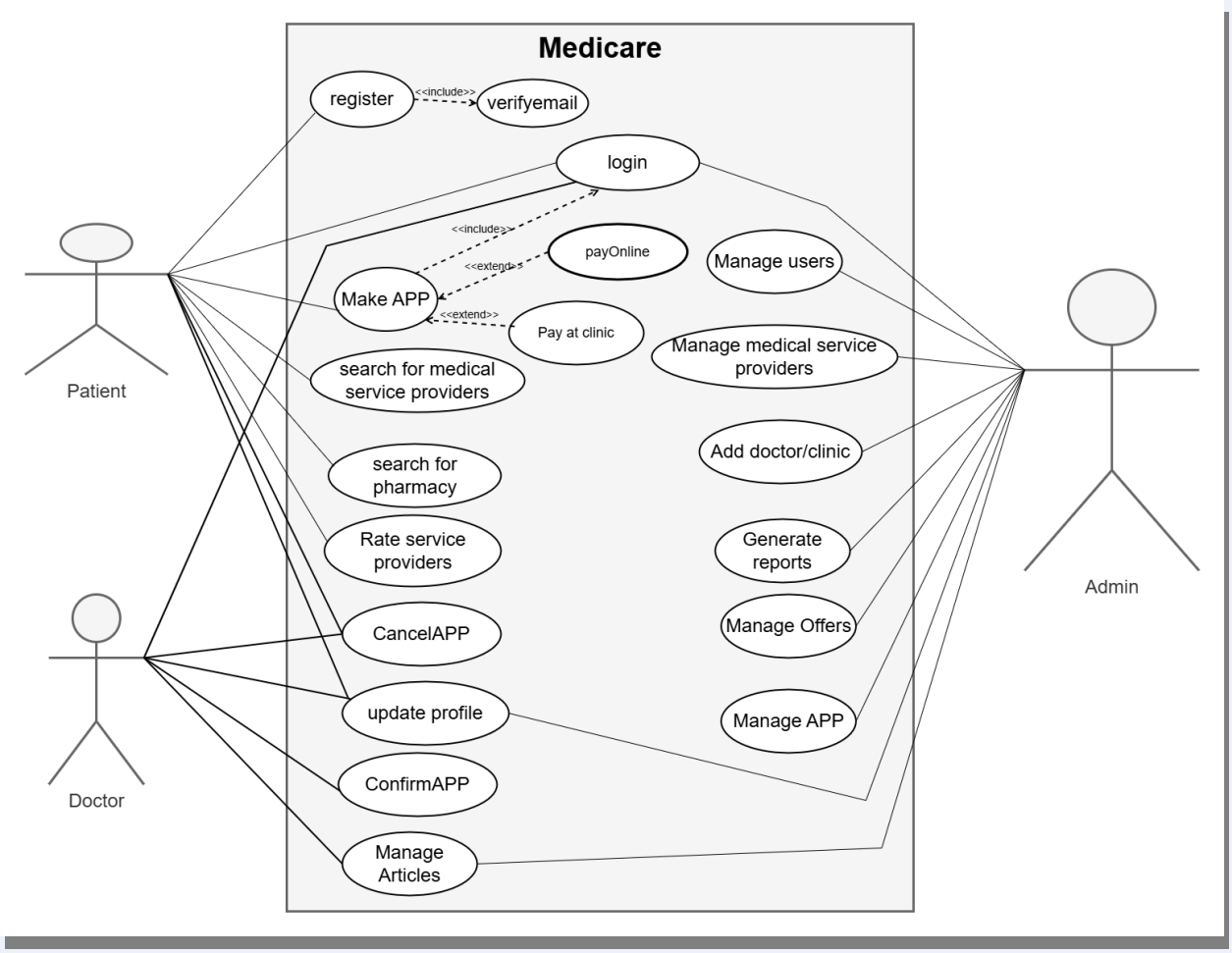



**Figure 3.1 Use Case Diagram**

## ➢ Context Diagram

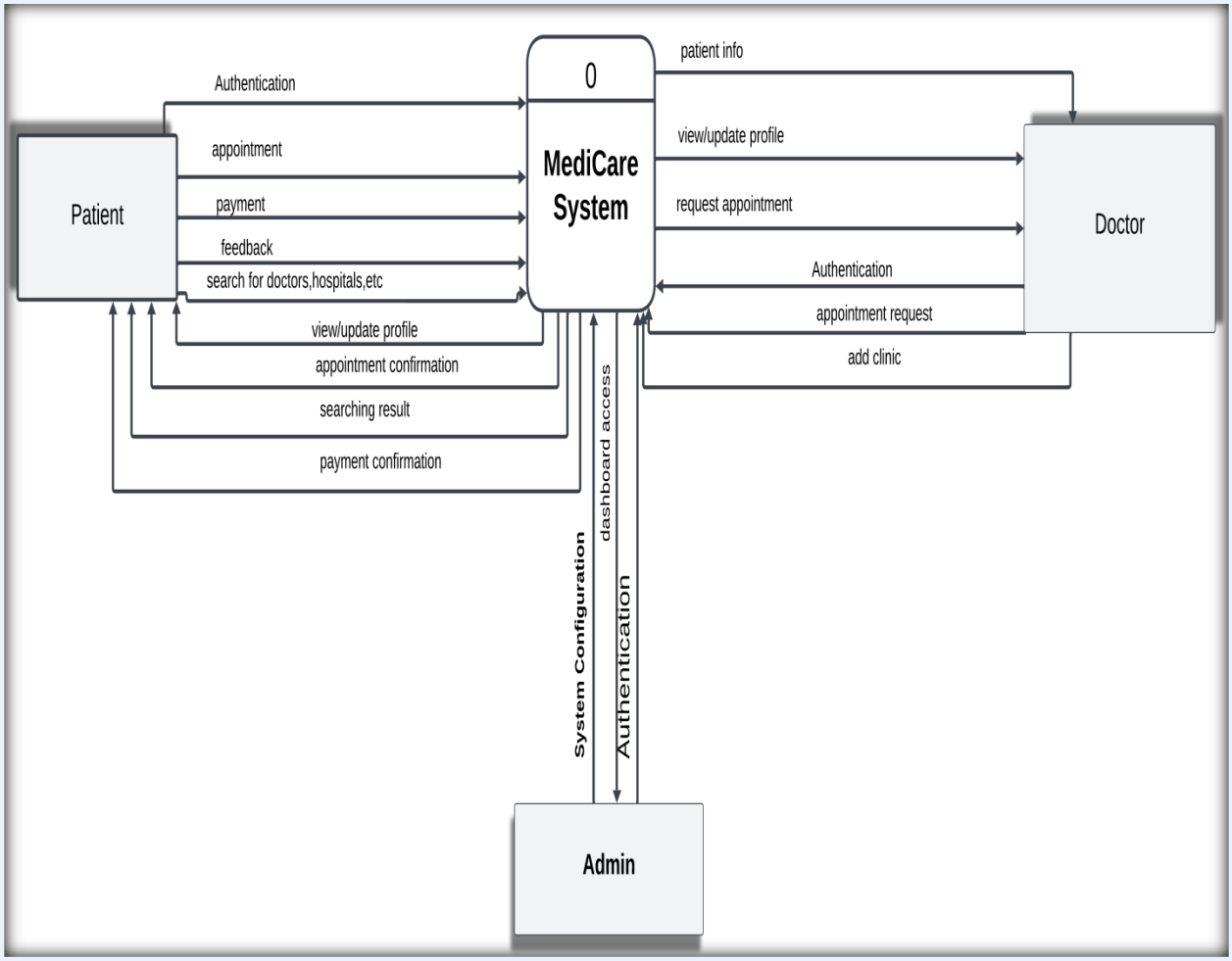



**Figure 3.2 Context Diagram**

## ➢ Data Flow Diagram

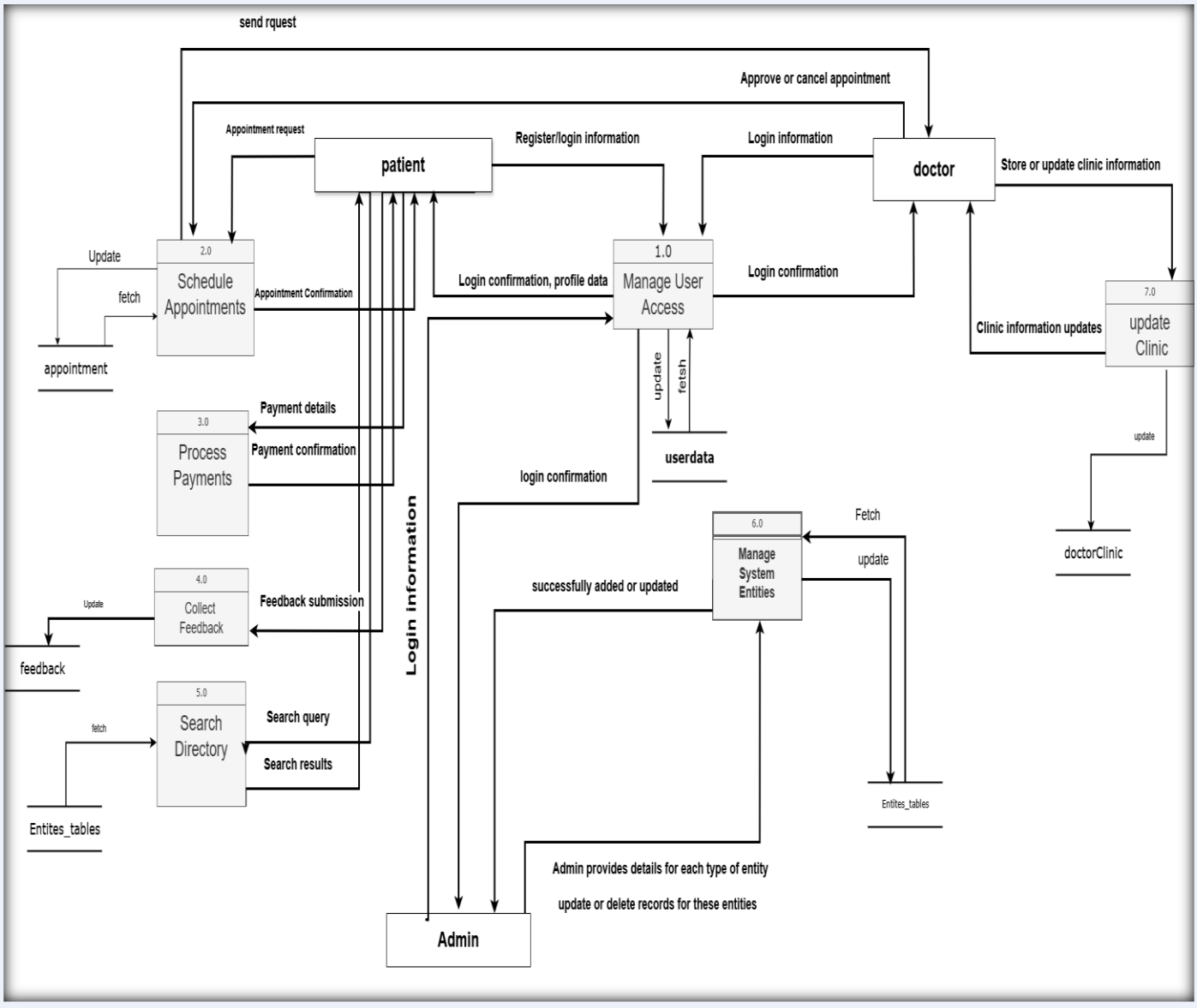



**Figure 3.3 Data flow Diagram**

## ➢ Entity Relations Diagram

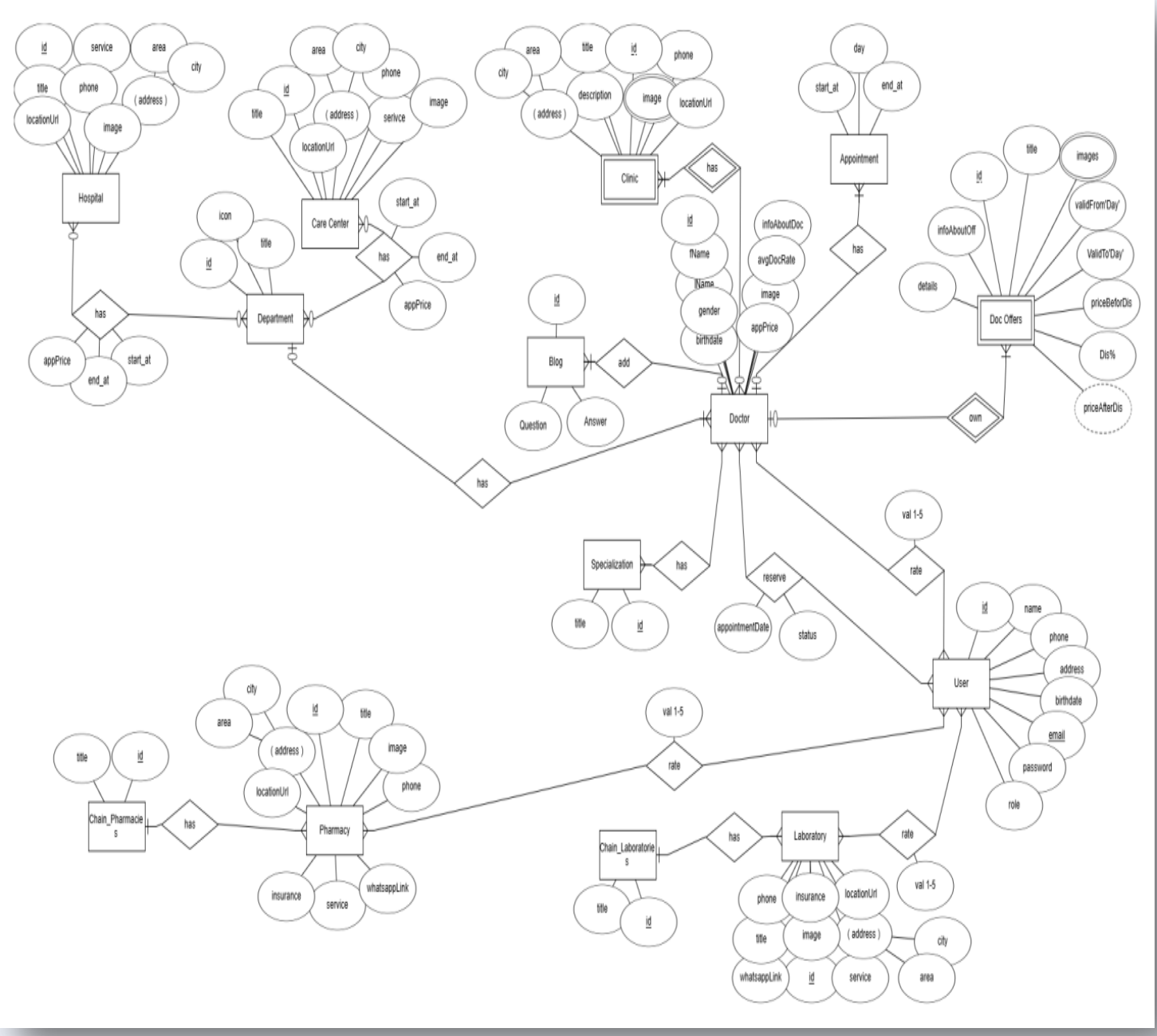
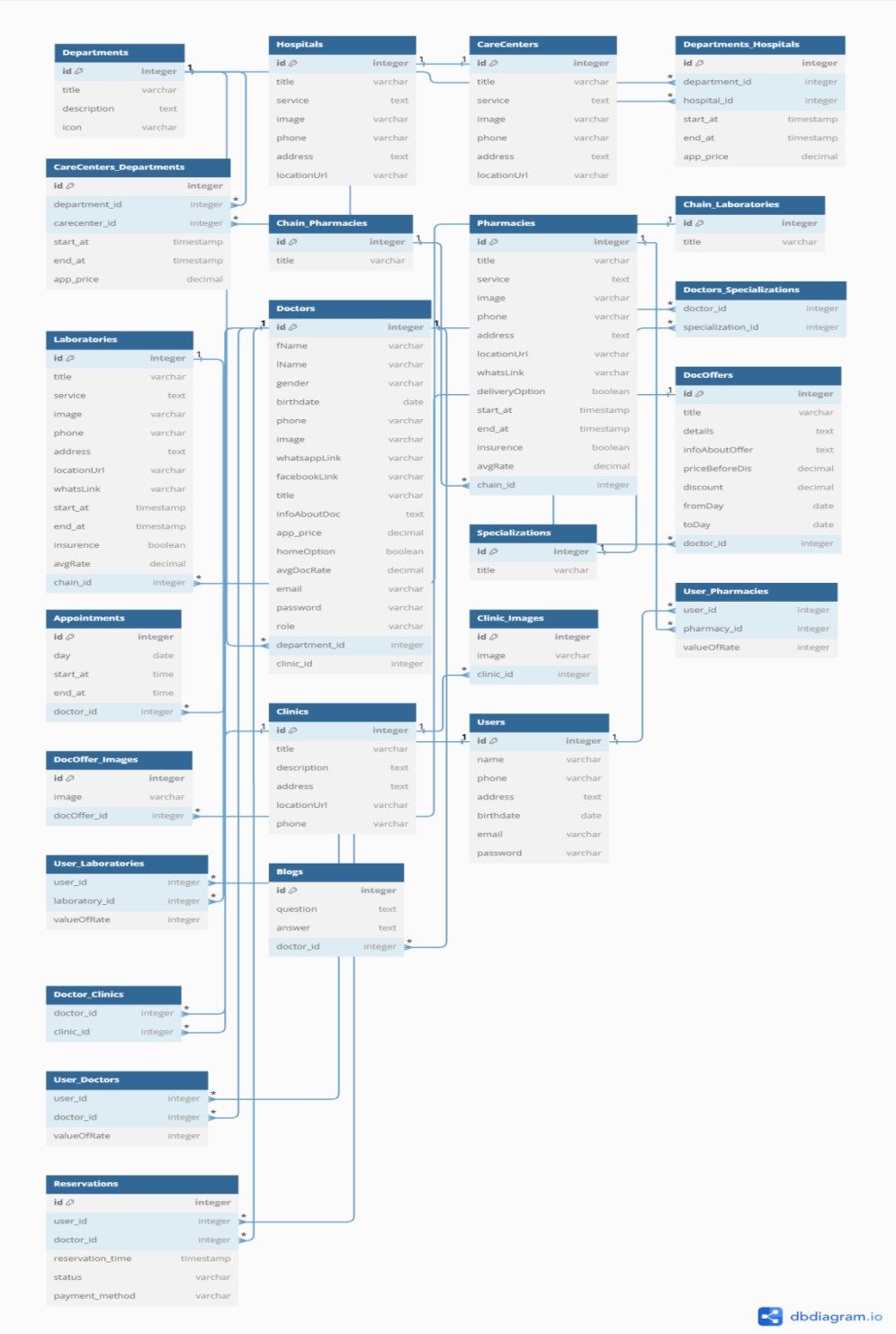



**Figure 3.5 Entity Relations Diagram**

## ➢ Mapping


Departments
id integer
title varchar
description text
icon varchar
Hospitals
id integer
title varchar
service text
image varchar
phone varchar
address text
locationUrl varchar
CareCenters
id integer
title varchar
service text
image varchar
phone varchar
address text
locationUrl varchar
Departments_Hospitals
id integer
department_id integer
hospital_id integer
start_at timestamp
end_at timestamp
app_price decimal
CareCenters_Departments
id integer
department_id integer
carecenter_id integer
start_at timestamp
end_at timestamp
app_price decimal
Chain_Pharmacies
id integer
title varchar
Pharmacies
id integer
title varchar
service text
image varchar
phone varchar
address text
locationUrl varchar
whatsLink varchar
deliveryOption boolean
start_at timestamp
end_at timestamp
insurence boolean
avgRate decimal
chain_id integer
Chain_Laboratories
id integer
title varchar
Doctors_Specializations
doctor_id integer
specialization_id integer
Doctors
id integer
fName varchar
lName varchar
gender varchar
birthdate date
phone varchar
image varchar
whatsappLink varchar
facebookLink varchar
title varchar
infoAboutDoc text
app_price decimal
homeOption boolean
avgDocRate decimal
email varchar
password varchar
role varchar
department_id integer
clinic_id integer
Laboratories
id integer
title varchar
service text
image varchar
phone varchar
address text
locationUrl varchar
whatsLink varchar
start_at timestamp
end_at timestamp
insurence boolean
avgRate decimal
chain_id integer
DocOffers
id integer
title varchar
details text
infoAboutOffer text
priceBeforeDis decimal
discount decimal
fromDay date
toDay date
doctor_id integer
Specializations
id integer
title varchar
User_Pharmacies
user_id integer
pharmacy_id integer
valueOfRate integer
Appointments
id integer
day date
start_at time
end_at time
doctor_id integer
Clinic_Images
id integer
image varchar
clinic_id integer
Clinics
id integer
title varchar
description text
address text
locationUrl varchar
phone varchar
Users
id integer
name varchar
phone varchar
address text
birthdate date
email varchar
password varchar
DocOffer_Images
id integer
image varchar
docOffer_id integer
User_Laboratories
user_id integer
laboratory_id integer
valueOfRate integer
Blogs
id integer
question text
answer text
doctor_id integer
Doctor_Clinics
doctor_id integer
clinic_id integer
User_Doctors
user_id integer
doctor_id integer
valueOfRate integer
Reservations
id integer
user_id integer
doctor_id integer
reservation_time timestamp
status varchar
payment_method varchar
dbdiagram.io

# Chapter 4: Project Tools

In this section we will talk about the project tools.

# Tools

➢ **Tools which we will use in Back-end:**

- PHP
- MySQL
- Laravel

➢ **Tools which we will use in Mobile Application:**

- Flutter
- Dart
- FireBase

➢ **Tools which we will use in front-end:**

- HTML
- CSS
- Java Script
- Bootstrap
- Angular

➢ **Tools which we will use in UI/UX:**

- Figma
- Canva
- Photoshop

# Chapter 5

## System Design and Implementation

## (Design & Code)

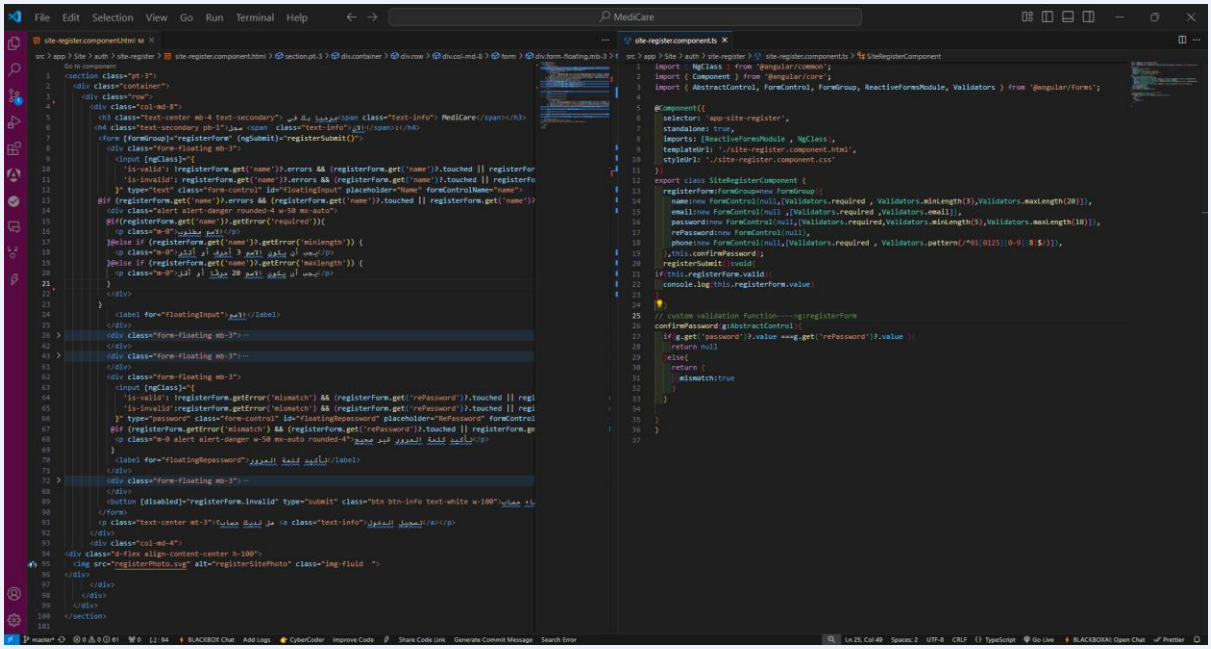
**In this section ,we will focus on detailing the design, and how the system is implemented.**

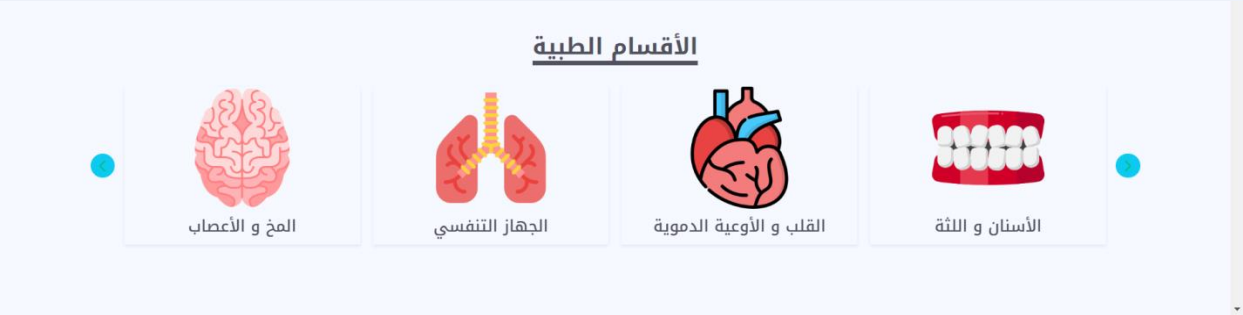
**This includes the design of the user interface (UI), and how the features of the system are coded.**

# Web Site Design

## 1-Register Page


## Used Code






## Used Code

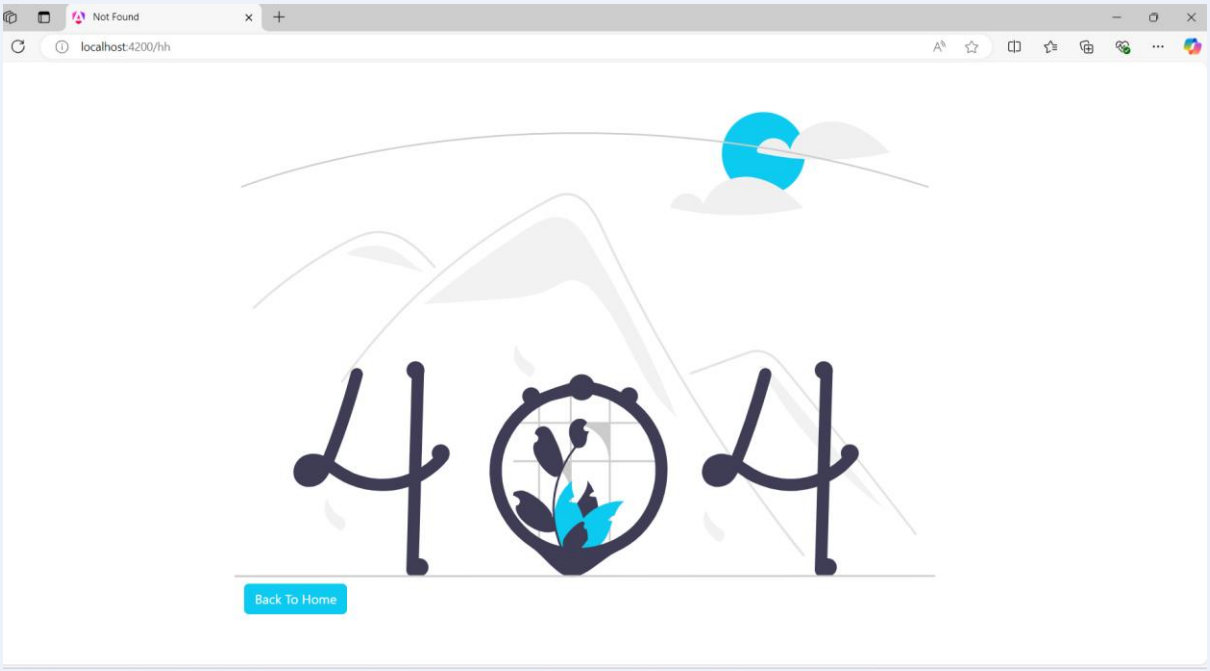
```
<section id="departments">
  <div class="container text-center">
    <h3 class="head-title">الأقسام الطبية</h3>
  </div>

  <div class="container mt-4">
    <p-carousel id="department-carousel" class="py-5" [value]="Departments" [numVisible]="4"
[circular]="true"
      [responsiveOptions]="responsiveOptions">
      <ng-template let-department pTemplate="item">
        <div class="border shadow-sm border-0 rounded-border m-2">
          <div class="relative mx-auto">
            <img [src]="department.imageUrl" class=" d-block mx-auto rounded-border" />
          </div>
          <h5 class="title">{{ department.title }}</h5>
        </div>
      </ng-template>
    </p-carousel>
  </div>
</section>
```


## Used Code

```html
<!-- Start Pharmacy Section -->
<section id="Pharmacy">
  <div class="container text-center">
    <h2 class="head-title">الصيدليات</h2>
  </div>
  <div class="container my-5">
    <div class="row">
      <div class="col-md-6">
        <img src="./pharmacy.jpg" alt="صورة صيدلية" class="img-fluid">
      </div>
      <div class="col-md-6 d-flex flex-column justify-content-center">
        <h3>
          مع <span>MediCare</span>
 بنقدملك حل مبتكر يجمع كل الصيدليات في محافظة سوهاج في منصة واحدة .
          دلوقتي تقدر توصل لأقرب صيدلية وتتعرف على خدماتها بسرعة وسهولة. لأن صحتك دايمًا أو
 لويتنا.
        </h3>
      </div>
    </div>
  </div>
</section>
<!-- End Pharmacy Section -->
```


## Used Code

```
<section id="DocOffers">
  <div class="container text-center">
    <h3 class="head-title">عروض الأطباء</h3>
  </div>

  <div class="container my-4">
    <p-carousel id="docOffers-carousel" class="py-5" [value]="DocOffers" [numVisible]="4"
[circular]="true"
      [responsiveOptions]="responsiveOptions">
      <ng-template let-docOffer pTemplate="item">
        <div class="docOffer-item shadow-sm p-6 border border-0 rounded-border m-2">
          <div class="info">
            <h6>25%</h6>
            <h6>OFF</h6>
          </div>
          <div class="relative mx-auto">
            <img [src]="docOffer.imageUrl" class="w-100 rounded-border" />
          </div>
          <h5 class="title">{{ docOffer.title }}</h5>
        </div>
      </ng-template>
    </p-carousel>
  </div>
</section>

```




## Used Code

```
<div>
  <div class=
"vh-75 d-flex justify-content-center align-items-center">
    <img src="not-found.svg"  class="w-100"  alt=
"not-found">
  </div>
  <div class="container">
    <div class="row">
      <div class="col-md-12 mt-2">
        <button routerLink="/home" class=
"btn btn-info text-white">Back To Home</button>
      </div>
    </div>
  </div>
</div>

```

# Mobile Design

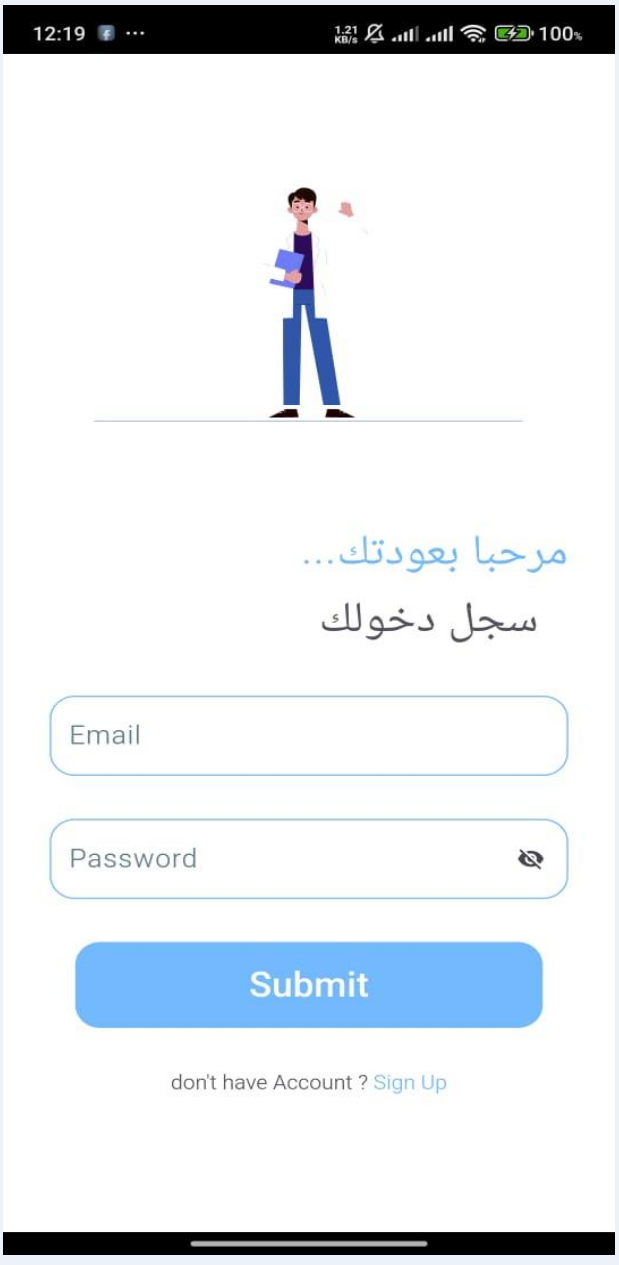
## Splash Screen

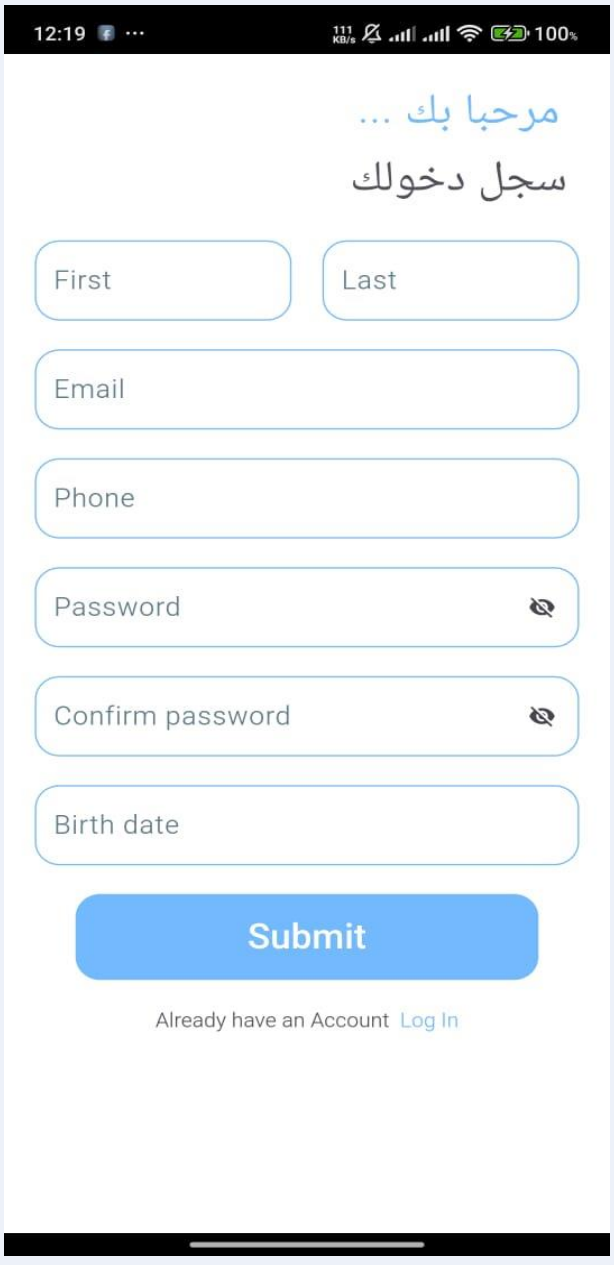
```dart
import 'package:animated_splash_screen/animated_splash_screen.dart';
import 'package:flutter/material.dart';
import 'package:lottie/lottie.dart';
import 'package:media_care/presentation/views/intro/introduction_page_view.dart';

class SplashViewBody extends StatelessWidget {
  const SplashViewBody({
    super.key,
  });

  @override
  Widget build(BuildContext context) {
    return AnimatedSplashScreen(
      splash: Column(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          Flexible(
            child: Lottie.asset(
              'assets/animation/doctorAnimation1.json',
            ),
          ),
          Flexible(
            child: Image.asset(
              'assets/animation/Medicare.png',
            ),
          )
        ],
      ),
      splashIconSize: 500,
      nextScreen: const IntroView(),
      // duration: 3500,
      // backgroundColor: Colors.white,
    );
  }
}
```

# OnBoarding Screen

```dart
import 'package:flutter/material.dart';
import 'package:media_care/presentation/views/intro/widgets/custom_intro_page.dart';

import '../../../../core/utlis/app_images.dart';
import 'custom_bottom_sheet.dart';
import 'custom_lets_go_button.dart';

class IntroViewBody extends StatefulWidget {
  const IntroViewBody({super.key});

  @override
  State<IntroViewBody> createState() => _IntroViewBodyState();
}

final controller = PageController();
bool isLastPage = false;

List<CustomIntroPage> intros = [
  CustomIntroPage(
      image: Assets.imagesDoctors,
      title: 'صحتك تبدأ هنا !',
      subTitle: 'استكشف مجموعة متكاملة من الخدمات الطبية بسهولة وسرعة .'),
  CustomIntroPage(
      image: Assets.imagesApointment,
      title: 'احجز موعدك بضغطة زر !',
      subTitle: 'ابحث عن الاطباء المتخصصين واحجز موعدك في ثوانٍ .'),
  CustomIntroPage(
      image: Assets.imagesArticals,
      title: 'معلومات طبية في أي وقت !',
      subTitle: 'احصل على استشارات طبية موثوقة مباشرة عبر التطبيق .'),
  CustomIntroPage(
      image: Assets.imagesLabs,
      title: 'الرعاية الصحية أصبحت أسهل !',
      subTitle: 'استمتع بتجربة طبية مريحة ومناسبة لجميع احتياجاتك .'),
];

class _IntroViewBodyState extends State<IntroViewBody> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        padding: const EdgeInsets.only(bottom: 50),
        child: PageView(
          onPageChanged: (index) => setState(() {
            isLastPage = index == intros.length - 1;
          }),
          controller: controller,
          children: intros,
        ),
      ),
      bottomSheet: isLastPage
          ? CustomLetsGoButton()
          : CustomButomSheet(controller: controller, intros: intros),
    );
  }
}
```

# Auth Screen


12:19
100%
مرحبا بعودتك...
سجل دخولك
Email
Password
Submit
don't have Account ? Sign Up

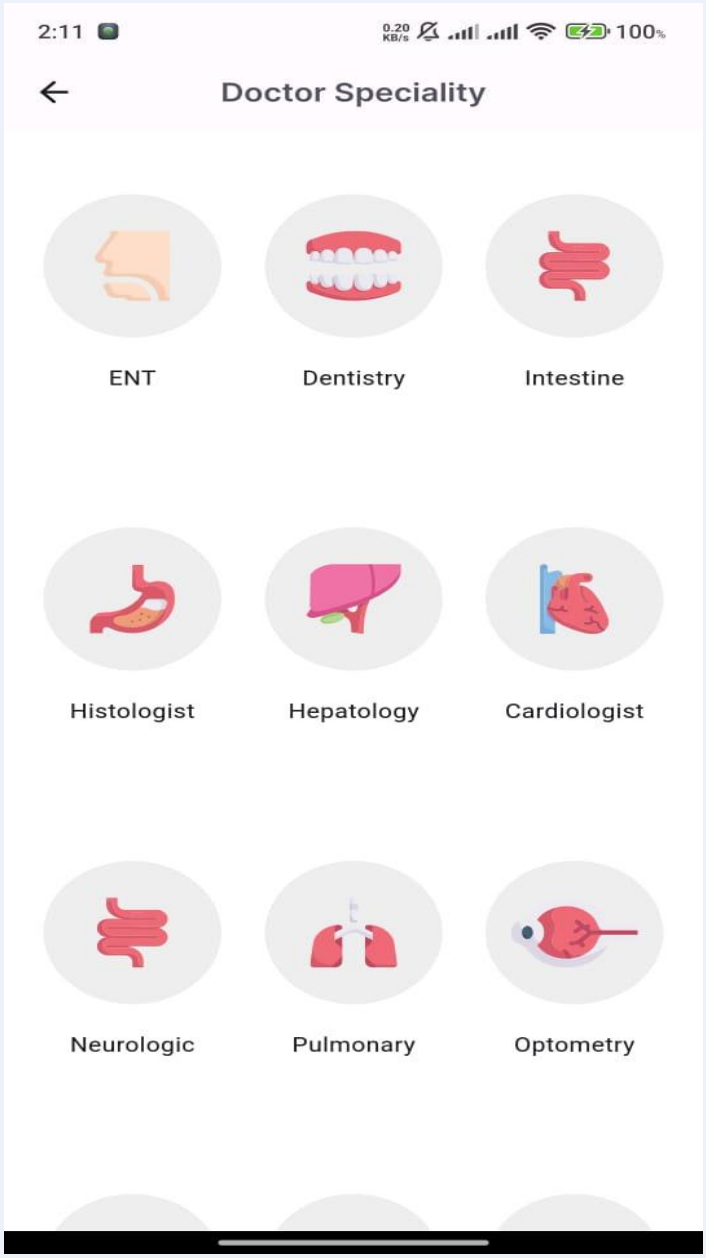

12:19
100%
مرحبا بك ...
سجل دخولك
First
Last
Email
Phone
Password
Confirm password
Birth date
Submit
Already have an Account Log In

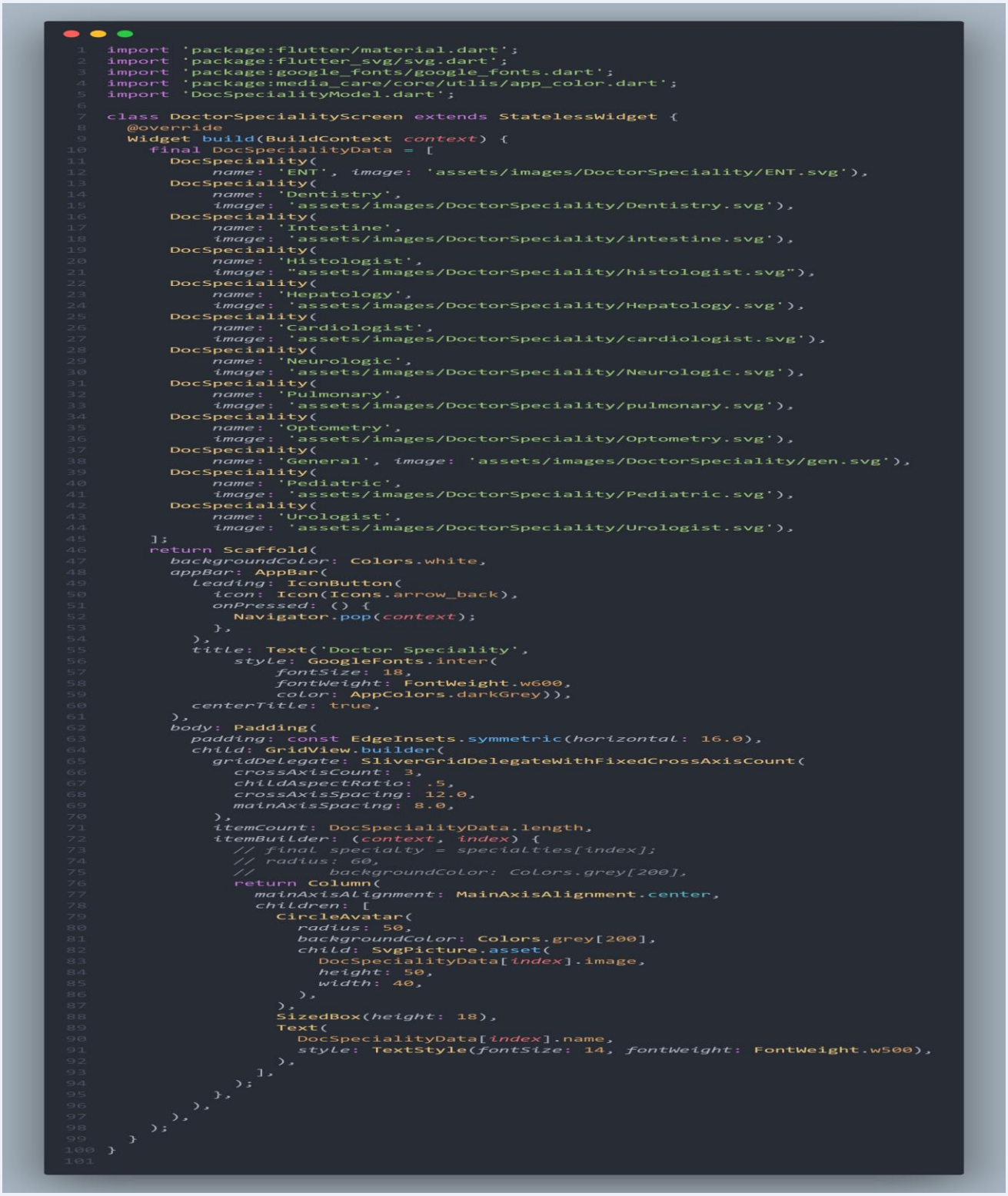
```dart
import 'package:flutter/material.dart';
import 'package:lottie/lottie.dart';
import 'package:media_care/presentation/views/Auth/register/regester_view.dart';

import '../../../../../core/utlis/app_color.dart';
import 'dont_have_email_password.dart';
import 'email_and_passowrd_form.dart';

class LoginViewBody extends StatelessWidget {
  const LoginViewBody({
    super.key,
  });

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        backgroundColor: Colors.white,
        body: SingleChildScrollView(
          child: Padding(
            padding: const EdgeInsets.symmetric(horizontal: 30, vertical: 24),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.center,
              children: [
                SizedBox(
                    height: 300,
                    width: 300,
                    child: Lottie.asset(
                      'assets/animation/doctorWelcomed.json',
                    )),
                Align(
                  alignment: AlignmentDirectional.centerEnd,
                  child: Column(
                    children: [
                      Text(
                        'مرحبا بعودتك...',
                        style: TextStyle(
                          fontSize: 30,
                          color: AppColors.primary,
                        ),
                      ),
                      Text(
                        'سجل دخولك ',
                        style: TextStyle(
                          fontSize: 30,
                          color: AppColors.darkGrey,
                        ),
                      ),
                    ],
                  ),
                ),
                SizedBox(
                  height: 30,
                ),
                EmailAndPasswordForm(),
                SizedBox(
                  height: 30,
                ),
                DontHaveAccountText(
                  router: RegisterView(),
                  text: "don't have Account ?",
                  boldText: " Sign Up",
                ),
              ],
            ),
          ),
        ),
      );
    }
}

```

```dart
import 'package:flutter/material.dart';
import 'package:media_care/core/utlis/app_regex.dart';
import 'package:media_care/presentation/views/Auth/login/login_view.dart';
import 'package:media_care/presentation/views/Auth/login/widgets/custom_login_button.dart';

import '../../login/widgets/custom_text_form_field.dart';
import 'first_last_names_form.dart';

class RegisterForm extends StatefulWidget {
  const RegisterForm({
    super.key,
  });

  @override
  State<RegisterForm> createState() => _RegisterFormState();
}

class _RegisterFormState extends State<RegisterForm> {
  GlobalKey<FormState> formKey = GlobalKey();
  bool isSecure = true;

  @override
  Widget build(BuildContext context) {
    return Form(
      key: formKey,
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.center,
        children: [
          FirstAndLastNameForm(),
          SizedBox(
            height: 20,
          ),
          CustomTextField(
            label: 'Email',
            validator: (value) {
              if (!AppRegex.isEmailValid(value!)) {
                return 'Enter a Valid Email';
              }
              return null;
            },
          ),
          SizedBox(
            height: 20,
          ),
          CustomTextField(
            inputType: TextInputType.phone,
            label: 'Phone',
            validator: (value) {
              if (value == null || value.isEmpty) {
                return 'field is required';
              } else if (!AppRegex.isPhoneNumberValid(value)) {
                return 'Enter correct form of password';
              }
              return null;
            },
          ),
          SizedBox(
            height: 20,
          ),
          CustomTextField(
            suffixIcon: GestureDetector(
                onTap: () {
                  setState(() {
                    isSecure = !isSecure;
                  });
                },
                child: isSecure
                    ? Icon(Icons.visibility_off)
                    : Icon(Icons.visibility)),
            label: 'Password',
            isObscureText: isSecure,
            validator: (value) {
              if (value == null || value.isEmpty) {
                return 'field is required';
              } else if (!AppRegex.isPasswordValid(value)) {
                return 'Enter correct form of password';
              }
              return null;
            },
          ),
```

```
    SizedBox(
                height: 20,
              ),
              CustomTextField(
                isObscureText: isSecure,
                suffixIcon: GestureDetector(
                    onTap: () {
                      setState(() {
                        isSecure = !isSecure;
                      });
                    },
                    child: isSecure
                        ? Icon(Icons.visibility_off)
                        : Icon(Icons.visibility)),
                label: 'Confirm password',
                validator: (value) {
                  if (value == null || value.isEmpty) {
                    return 'field is required';
                  } else if (!AppRegex.isPasswordValid(value)) {
                    return 'Enter correct form of password';
                  }
                  return null;
                },
              ),
              SizedBox(
                height: 20,
              ),
              CustomTextField(
                inputType: TextInputType.datetime,
                label: 'Birth date',
                validator: (value) {
                  if (value == null || value.isEmpty) {
                    return 'field is required';
                  } else if (!AppRegex.isDateOfBirthValid(value)) {
                    return 'Enter Invalid Date';
                  }
                  return null;
                },
              ),
              SizedBox(
                height: 20,
              ),
              CustomLoginButton(
                  text: 'Submit',
                  onPresed: () {
                    if (formKey.currentState!.validate()) {
                      Navigator.push(context, MaterialPageRoute(
                        builder: (context) {
                          return LoginView();
                        },
                      ));
                    }
                  }),
              SizedBox(
                height: 20,
              ),
            ],
          ),
        );
    }
}
```

# Department Screen





```dart
import 'package:flutter/material.dart';
import 'package:flutter_svg/svg.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:media_care/core/utlis/app_color.dart';
import 'DocSpecialityModel.dart';

class DoctorSpecialityScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final DocSpecialityData = [
      DocSpeciality(
          name: 'ENT', image: 'assets/images/DoctorSpeciality/ENT.svg'),
      DocSpeciality(
          name: 'Dentistry',
          image: 'assets/images/DoctorSpeciality/Dentistry.svg'),
      DocSpeciality(
          name: 'Intestine',
          image: 'assets/images/DoctorSpeciality/intestine.svg'),
      DocSpeciality(
          name: 'Histologist',
          image: "assets/images/DoctorSpeciality/histologist.svg"),
      DocSpeciality(
          name: 'Hepatology',
          image: 'assets/images/DoctorSpeciality/Hepatology.svg'),
      DocSpeciality(
          name: 'Cardiologist',
          image: 'assets/images/DoctorSpeciality/cardiologist.svg'),
      DocSpeciality(
          name: 'Neurologic',
          image: 'assets/images/DoctorSpeciality/Neurologic.svg'),
      DocSpeciality(
          name: 'Pulmonary',
          image: 'assets/images/DoctorSpeciality/pulmonary.svg'),
      DocSpeciality(
          name: 'Optometry',
          image: 'assets/images/DoctorSpeciality/Optometry.svg'),
      DocSpeciality(
          name: 'General', image: 'assets/images/DoctorSpeciality/gen.svg'),
      DocSpeciality(
          name: 'Pediatric',
          image: 'assets/images/DoctorSpeciality/Pediatric.svg'),
      DocSpeciality(
          name: 'Urologist',
          image: 'assets/images/DoctorSpeciality/Urologist.svg'),
    ];
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBar(
        leading: IconButton(
          icon: Icon(Icons.arrow_back),
          onPressed: () {
            Navigator.pop(context);
          },
        ),
        title: Text('Doctor Speciality',
            style: GoogleFonts.inter(
                fontSize: 18,
                fontWeight: FontWeight.w600,
                color: AppColors.darkGrey)),
        centerTitle: true,
      ),
      body: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 16.0),
        child: GridView.builder(
          gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
            crossAxisCount: 3,
            childAspectRatio: .5,
            crossAxisSpacing: 12.0,
            mainAxisSpacing: 8.0,
          ),
          itemCount: DocSpecialityData.length,
          itemBuilder: (context, index) {
            // final specialty = specialties[index];
            // radius: 60,
            //         backgroundColor: Colors.grey[200],
            return Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                CircleAvatar(
                  radius: 50,
                  backgroundColor: Colors.grey[200],
                  child: SvgPicture.asset(
                    DocSpecialityData[index].image,
                    height: 50,
                    width: 40,
                  ),
                ),
                SizedBox(height: 18),
                Text(
                  DocSpecialityData[index].name,
                  style: TextStyle(fontSize: 14, fontWeight: FontWeight.w500),
                ),
              ],
            );
          },
        ),
      ),
    );
  }
}
```

# Summary

MediCare is a comprehensive medical application and website designed to simplify healthcare access. It enables users to book doctor appointments, view doctor specializations, and read ratings for informed decision-making. The platform ensures the credibility and accuracy of doctor information, with each professional having a rating system to evaluate their performance. MediCare aims to provide a seamless and reliable healthcare experience for users.