DOCTOR APPOINTMENT APP

(Front Page)

Project Report submitted to

St. Xavier's College – Autonomous Mumbai

For the partial fulfilment for the award of the degree of

Bachelor of Science (BSc) in

Information Technology

By

Valencia Dsouza and Linore Cardozo

(UID. : 205025 and 205028)

Under the Supervision of

Ms. Lydia Fernandes

INFORMATION TECHNOLOGY

ST. XAVIER'S COLLEGE (AUTONOMOUS),

MUMBAI-400001, INDIA

April 2023

PROJECT CERTIFICATE

This is to certify that the project entitled **Doctor Appointment app** undertaken at the Information Technology Department of St. Xavier's College – Autonomous Mumbai has been submitted by: **Valencia Dsouza (UID No: 205025) and Linore Cardozo (UID No: 205028)**, in partial fulfilment of Bachelor's in Information Technology degree (Semester VI) Examination. It is further certified that they have completed all required phases of the project.

Signature Signature Signature

(Internal Guide) (Internal Examiner) (External Examiner)

Signature College Seal

(HOD – Information Technology department)

Student Declaration

We**, Valencia Dsouza (UID No: 205025) and Linore Cardozo (UID No: 205028),** do hereby, certify that:

1. that the project report titled, "Santé -A doctor appointment app" which is being submitted in partial fulfilment of the requirements for the Degree of Bachelor of Science with a specialization in Information Technology is the result of the **original work** carried out by us under the guidance of the Ms, Lydia Fernandes, faculty of Information Technology Department, St. Xavier's College, Mumbai-01.
2. This project has not previously formed the basis for the award of any degree, diploma, or certificate of this college or of any other college or university.
3. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them by providing references of them in the project documentation.
4. From the plagiarism test, it is found that the similarity index of whole submission is within ___%. [optional]

Date:

Place: Mumbai

___________________	___________________
{sign} Valencia Dsouza	{sign} Linore Cardozo
205025	205028

ACKNOWLEDGEMENTS

We would like to express our profound gratitude to our mentor Ms. Lydia Fernandes for guiding us throughout the planning and development phase of the system. Ms. Lydia Fernandes' guidance and cooperating nature has provided us with knowledge and directions on how to prepare the final documentation.

We would also like to thank our principal and Head of Department Prof. Roy Thomas for their valuable support and guidance.

Our gratitude extends further to the rest of the course staff for their assistance and support. Their contributions have been valuable in helping us to understand the concepts and complete the project successfully.

In addition, we would like to thank our classmates for their support and collaboration during this project. Their insights and perspectives have helped to enrich my understanding of the material and have made this a truly collaborative and educational experience.

Finally, we also extend our heartiest thanks to our parents, friends, and well-wishers for being with us and extending encouragement throughout the project.

INDEX

1. PRELIMNARY INVESTIGATION

1.1 INTRODUCTION

Life is becoming too busy to get medical appointments in person and to maintain proper healthcare. If anybody is ill and wants to visit a doctor for checkup, he or she needs to visit the hospital and wait until the doctor is available. The patient also waits in a queue while getting an appointment. If the doctor cancels the appointment for some emergency reasons, then the patient is not able to know about the cancellation of the appointment unless or until he or she visits the hospital. As the mobile communication technology is developing rapidly, therefore, one can use the mobile's applications to overcome such problems and inconvenience for the patients.

1.2 DESCRIPTION OF THE SYSTEM

Santé has been developed to override the problems prevailing in the practice of manual appointment system. The main idea of this app is to provide ease and comfort to patients while taking appointments from doctors and it also resolves the problems that the patients must face while making an appointment.

The app is user friendly which leads to error free, secure, reliable, and fast management system. The application acts as a client whereas the database containing the doctor's details, patient's details and appointment details is maintained by a website that acts as a server.

Every organization, whether big or small, has challenges to overcome and managing the information of Appointment, Doctor, Booking, Doctor Fees, Doctor Schedule. This is designed to assist in strategic planning and will help you ensure that you are equipped with the right level of information and details. Also, for those busy, who are always on the go, our app comes with remote access features, which will allow you to manage/book your appointment anytime, at all times.

The benefits of apps to schedule doctor appointments are twofold. First, they provide a simple way for patients to arrange visits with the healthcare providers of their choice. And secondly, these systems store all appointments and data entries under the same roof. Your hospital staff will thank you.

Here is how it works in the real world:

- ❖ From a patient's perspective. With an app, patients no longer need to call your office to schedule appointments. Instead, in just a few taps or clicks, they can arrange visits with their doctors at a time that works for them. This saves both you and your patient's time.
- ❖ From a clinic's perspective. An app or a web-based platform is the backbone of a clinic booking system software. It automates the appointment booking processes your staff may have trouble handling manually (e.g., entering patients' personal details, setting up doctors' time slots, dealing with paperwork, etc.). An app is also particularly helpful for tracking revisits and improving clinic-patient communication. This can result in less hassle when confirming, canceling, or rescheduling appointments.

1.3 OBJECTIVE AND SCOPE

Helping people to search for doctors and get appointments is our main objective. Users can search for doctors which can make finding a specific doctor an easy task.

Another objective of the project is to manage the details of Doctor, Appointment, Patient, Booking, Doctor Schedule. The proposed project is a smart appointment booking system that provides patients or any user an easy way of booking a doctor's appointment online.

This application overcomes the issue of managing and booking appointments according to the user's choice or demands. The task sometimes becomes very tedious for the compounder or doctor himself in manually allocating appointments for the users as per their availability. Hence this project offers an effective solution where users can view various booking slots available and select the preferred date and time.

1.4 KEY FEATURES:

- Search Module
- Appointment Booking
- Category (Different Doctors)
- Appointment management
- Schedule a timing
- Ratings and Reviews

1.5 BASIC APP FEATURES:

Below are three of the essential features present in Santé.

Doctor profiles

An app is where patients connect with their potential doctors for the first time. That's why physicians' profiles should be clear and detailed. They should include photos, core competencies, experience, and all the necessary information to facilitate the patient's decision-making process. You can also consider including consultation fees and treatment costs.

Search option

If you have an extensive database, patients might find it difficult to search through all those profiles. That's where a search option can be helpful. It allows patients to filter the profiles by desired treatments, doctor ratings, locations, and more.

Booking form

Ideally, arranging medical visits should be as easy as going to an online doctor appointment website or opening an app. Once a patient chooses a physician, they should be able to easily navigate available appointment dates and time slots. The fewer steps it takes to book a visit, the better.

2 ADD-ON FEATURES THAT CAN IMPROVE THE APP

Here are five extra-value features that can make the app stand out from the competition.

Medical records

It is always a good idea to be transparent with your patients, allowing them to keep tabs on how their treatments progress and access their test results. You can even include an upload option so that they can share diagnostic scans with their physicians.

Electronic prescriptions

This e-prescription feature enables patients to request refills directly from your app. Why is this important? It saves time for both you and your patients. They can now avoid traveling to your clinic, and you can write prescriptions faster. Refills are then available at in-network pharmacies.

Payments

Allowing patients to pre-pay for their consultations and treatments is especially convenient. Incorporating this option into your app can reduce your no-show rate, as paid appointments hardly ever turn into missed ones. Read also an article about customized medical billing software development.

Telemedicine

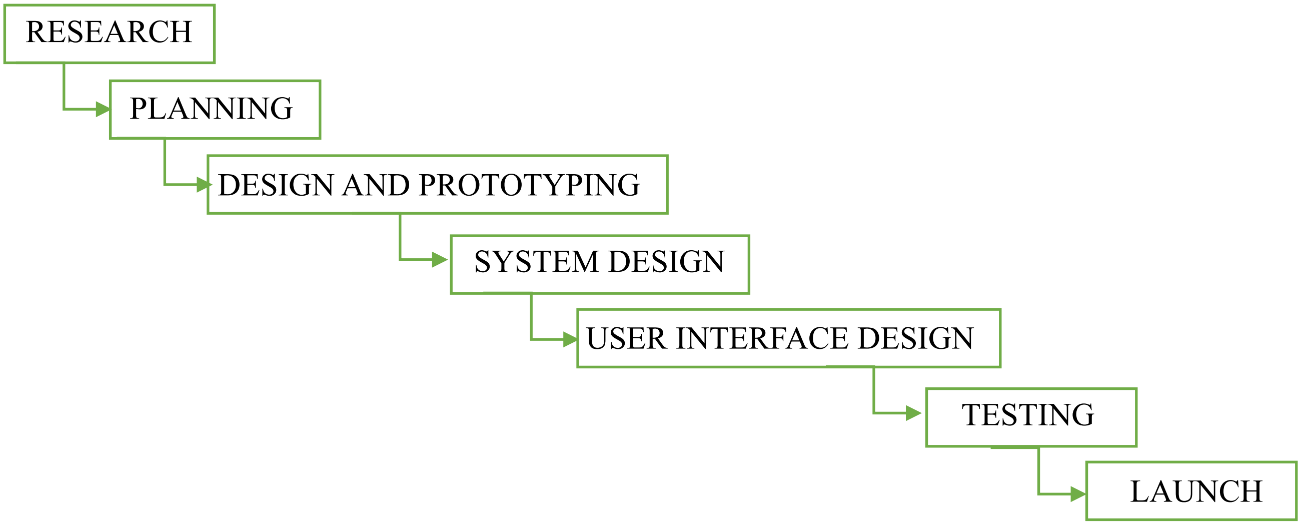
Even though including a telemedicine feature will likely drive up the cost of on-demand doctor appointment application development, it’s definitely worth it. By offering online consultations, you can reach those patients who are unable to attend face-to-face meetings.

Directions

Directions are essential if you have many locations. When directions are enabled in your app or web platform, patients can easily find the nearest clinic and how to get there. Coupled with business hours and contact details, directions take the guesswork out of medical visits.

3 PROCESS OF CREATING THE APP

It is one thing to just roll out an app; it’s another to make it scalable, fail-proof, and ready for thousands or even millions of users.



- **Research**

There is nothing wrong with learning from the mistakes the competitors have already made. We can even turn them into our assets.

In the research phase, we:

❖ Took a closer look at the competitors’ apps

❖ Analysed their pros and cons

❖ Spot missed opportunities

❖ Combined all of this with our patients’ needs

We then used our research findings to move onto the next step of our app development journey.

- **Planning**

This is where we decided on the features to implant into our app as well as the technology to rely on. We methodically planned the process down to the last detail, all while setting milestones for our roadmap.

At this stage, we also checked with our local regulations. Since doctor appointment solutions use patients' personal information, we need to make sure it is adequately protected. This involves thinking through encryption technologies and other technical safeguards for our software

.

- **Design and prototyping**

Now our research-backed plan moves into a more tangible phase. During the design and prototyping step, we:

❖ Decided on colors, patterns, and other visuals in our app

❖ Built a user interface

❖ Created an easy-to-navigate booking system

❖ Designed a menu and all clickable features

❖ Fine-tuned it for iOS and Android

- **System Design**

In this phase, a logical system is built which fulfils the given requirements. Design phase of software development deals with transforming the client's requirements into a logically working system. Normally, design is performed in the following two steps:

1. **Primary Design Phase:** In this phase, the system is designed at block level. The blocks are created based on analysis done in the problem identification phase. Different blocks are created for different functions, emphasis is put on minimising the information flow between blocks. Thus, all activities which require more interaction are kept in one block.
2. **Secondary Design Phase:** In the secondary phase the detailed design of every block is performed.

- **User Interface Design**

User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventually presentation of desired inputs and outputs. The overall flow of screens and messages is called a dialogue.

- **Testing**

We cannot build a doctor appointment app without letting patients (and physicians!) give it a test run first. This is where we check how intuitive our app is from both sides and what tweaks need to be made. It pays to ask patients and healthcare providers for some feedback to improve their experience down the road.

Usability aside, testing is also about how stable and secure our app is. It involves simulating extreme load conditions to monitor whether it can survive in real-life scenarios.

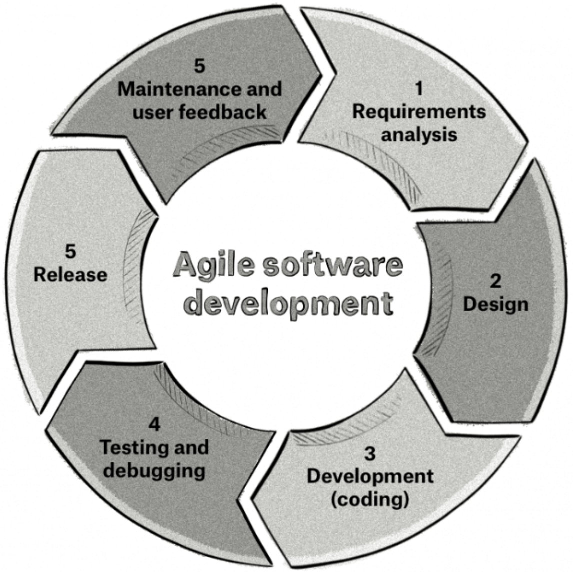
- **Launch**

Once we've made the final feedback-based improvements, our app is ready for its debut. But it doesn't simply end with uploading your app to the App Store.

Maintaining our app is also critically important. We don't want our patients' experience to be marred by technical glitches, nor give up adding new features later on. Regular updates are key to a killer app, especially in the healthcare industry.

4 Software development lifecycle model (SDLC Model)

We used the agile model for the development of our application . Agile methodology is a practice which promotes continue interaction of development and testing during the SDLC process of any project. In the Agile method, the entire project is divided into small incremental builds. All of these builds are provided in iterations, and each iteration lasts from one to three weeks.



We divided our project into several small parts, such as sign in page, home page, sign up page, etc. After developing each module we tested each model using dynamic testing technique to test the output. This has helped us to rectify the errors and bugs in the program , before it was to late to fix the issue.

5 SOFTWARE REQUIREMENTS

For creating the app, we have used Flutter for the UI and Laravel for the database. To run flutter and Laravel we will require the VS Code IDE.

- **Flutter**

Flutter is an open-source UI software development kit created by Google. It is used to develop cross-platform applications for Android, iOS, Linux, macOS, Windows, Google Fuchsia, and the web from a single codebase. Flutter is chosen as Google's application-level framework for its next-generation operating system. Flutter is exceptional because it is dependent on the device's OEM widgets rather than consuming web views. Flutter uses a high-performance rendering engine to render each view component using its own. This provides a chance to build applications that are as high-performance as native applications can be. Hot reload feature in Flutter is called as Stateful hot reload and it is a major factor for boosting the development cycle. Flutter supports it during development. Stateful hot reload is implemented by sending the updated source code into the running Dart Virtual Machine (Dart VM) without changing the inner structure of the application, therefore the transitions and actions of the application will be well-preserved after hot reloading.

- **Dart – Programming language used by flutter**

In Flutter, every application is written with the help of Dart. Google has developed and maintained a programming language called Dart. It is extensively used inside Google and it has been verified to have the proficiency to develop enormous web applications, such as AdWords. Originally Dart was developed to replace and succeed JavaScript. Thus, it implements most of the important characteristics of JavaScript's next standard (ES7), such as the keywords "async" and "await". Nonetheless, to attract developers that are not acquainted with JavaScript, Dart has a Java-like syntax. Flutter application renews the view tree on every new frame even when few other systems use reactive views. This behaviour leads to a drawback that many objects, which might survive for a singular frame, will be created. As Dart is a modern programming language, it is optimized to handle this scenario in memory level with the help of "Generational Garbage Collection."

- **Laravel**

Laravel is a free and open-source PHP web framework, created by Taylor Otwell and intended for the development of web applications following the model–view–controller (MVC) architectural pattern and based on Symfony. Some of the features of Laravel are a modular packaging system with a dedicated dependency manager, different ways for accessing relational databases, utilities that aid in application deployment and maintenance, and its orientation toward syntactic sugar. The source code of Laravel is hosted on GitHub and licensed under the terms of MIT License.

- **VS Code**

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add functionality. In the Stack Overflow 2022 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool among 71,010 respondents, with 74.48% reporting that they use it. Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including C, C#, C++, Fortran, Go, Java, JavaScript, Node.js, Python, Rust, Flutter, Dart. Due to the open-source nature of the application, the telemetry code is accessible to the public, who can see exactly what is collected.

6 HARDWARE REQUIREMENTS

To install and run the software's, your development environment must meet these minimum requirements:

- ❖ Operating Systems: Windows 10 or later (64-bit), x86-64 based.
- ❖ Disk Space: 1.64 GB (plus 1 GB disk space for IDE/tools).
- ❖ Tools: Flutter depends on these tools being available in your environment.
    - o Windows PowerShell 5.0 or newer (this is pre-installed with Windows 10).
    - o Git for Windows 2.x, with the Use Git from the Windows Command Prompt option.

7 SYSTEM STUDY AND ANALYSIS

7.1 Software Requirement Specification

The Software Requirements Specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description, a detailed functional and behavioural description, an indication of performance requirements and design constraints, appropriate validation criteria, and other data pertinent to requirements.

7.2 Identification of need

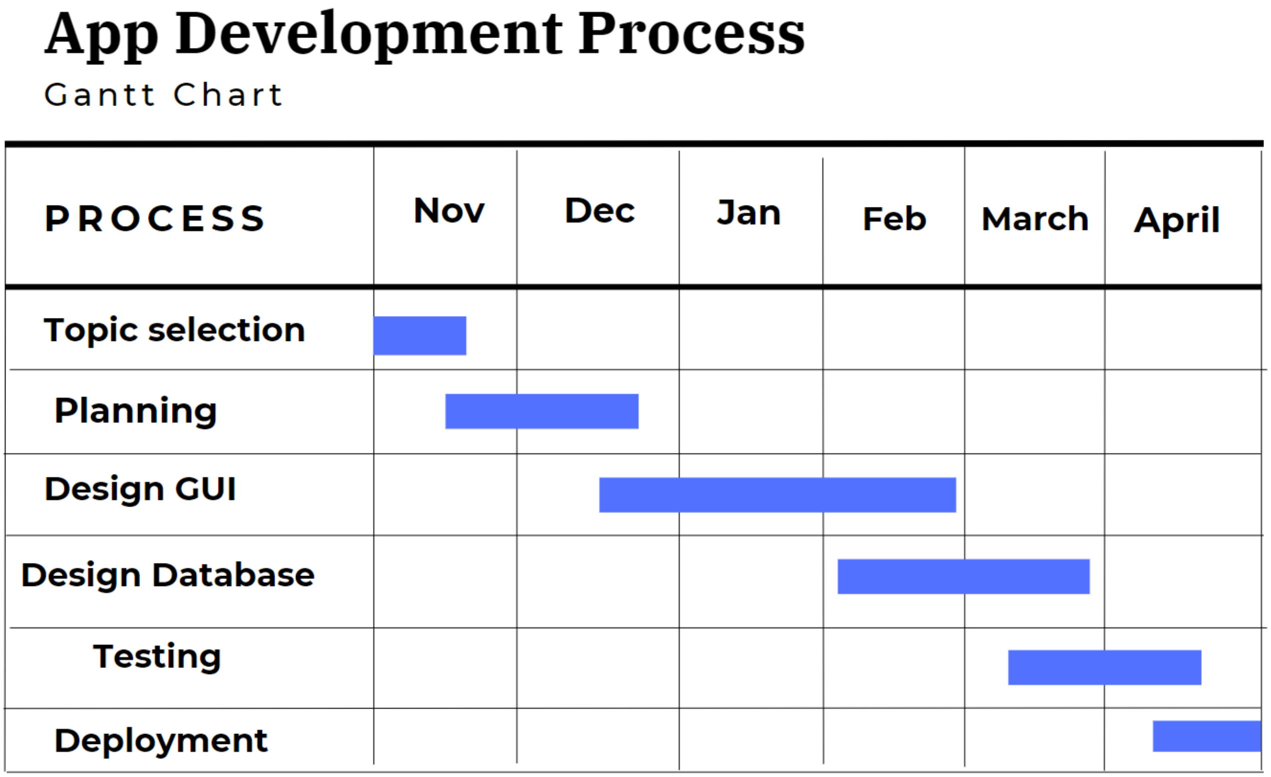
The old manual system was suffering from a series of drawbacks. Since whole of the system was to be maintained with hands the process of keeping, maintaining and retrieving the information was very tedious and lengthy. The records were never used to be in a systematic order. there used to be lots of difficulties in associating any particular transaction with a particular context. If any information was to be found it was required to go through the different registers, documents there would never exist anything like report generation. There would always be unnecessary consumption of time while entering records and retrieving records. One more problem was that it was very difficult to find errors while entering the records. Once the records were entered it was very difficult to update these records. The reason behind it is that there is lot of information to be maintained and have to be kept in mind while running the business. For this reason, we have provided features. This application overcomes the issue of managing and booking appointments according to the user's choice or demands. The task sometimes becomes very tedious for the compounder or doctor himself in manually allocating appointments for the users as per their availability. Hence this project offers an effective solution where users can view various booking slots available and select the preferred date and time.

7.3 Feasibility Study

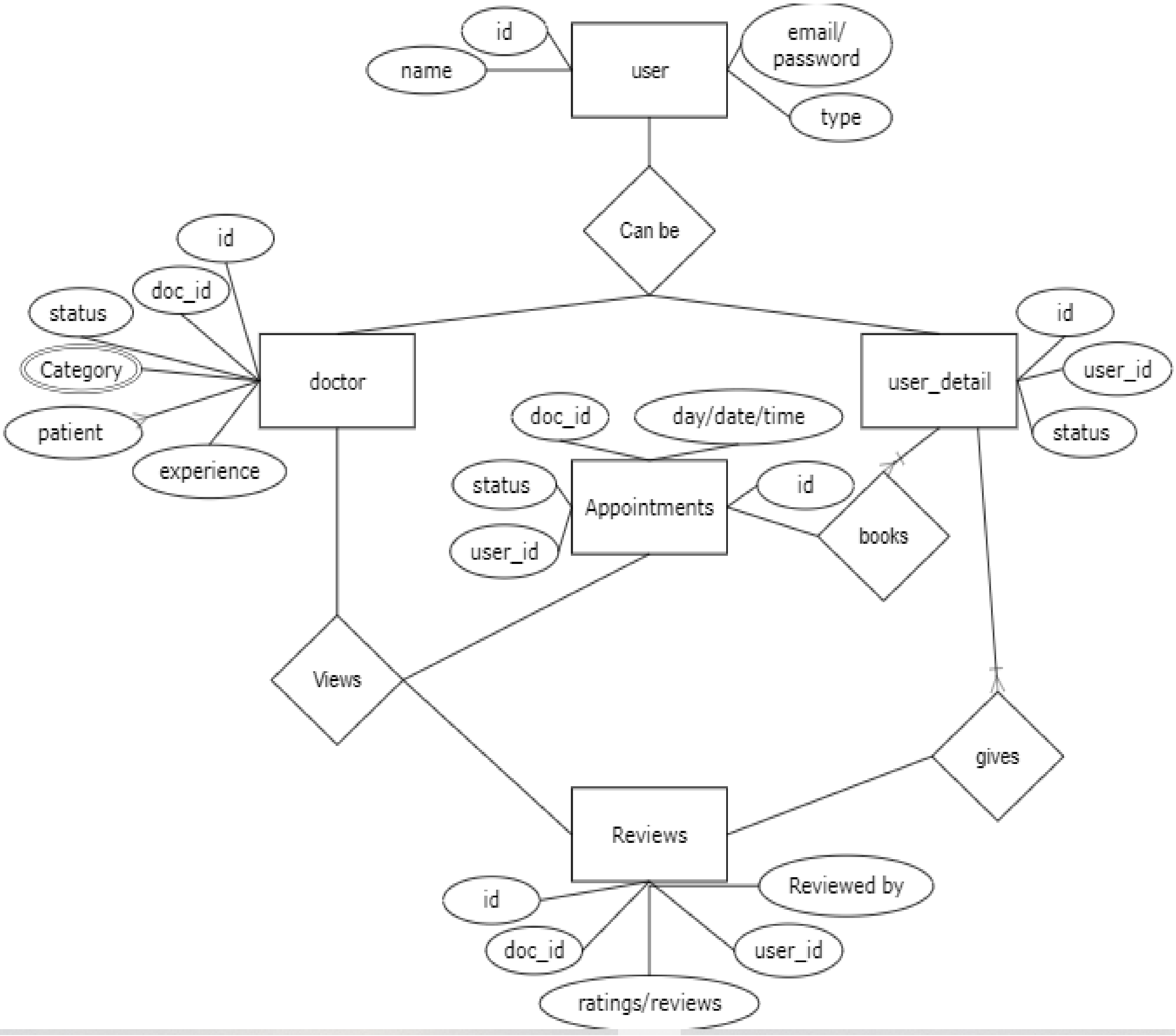
After doing the project Doctor Appointment System, study and analysing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible - given unlimited resources and infinite time. Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

8 SYSTEM DEVELOPMENT

8.1 Gantt Chart

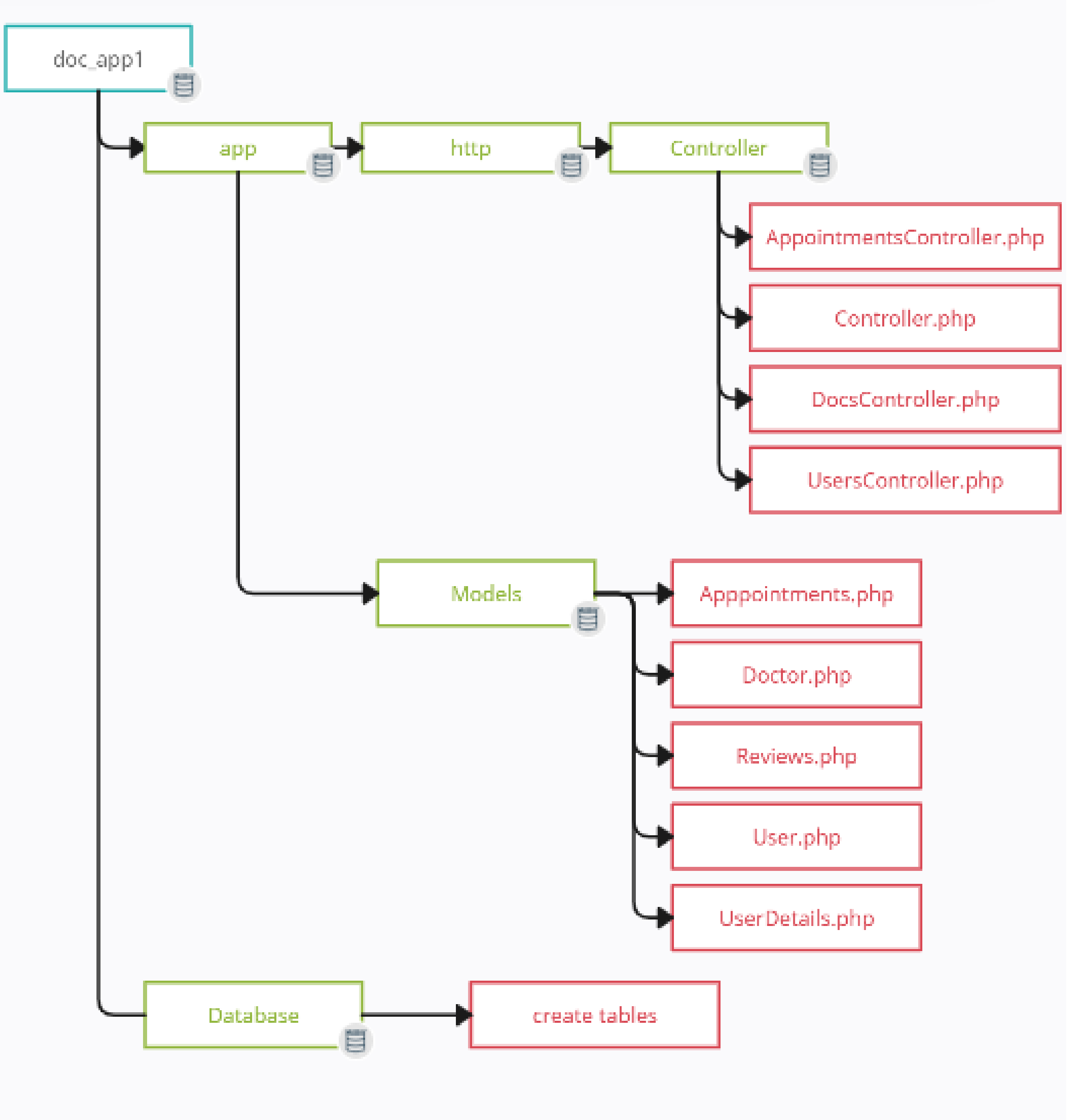


8.2 ER Diagram



9 SYSTEM CODING

File structure for Laravel and flutter backend



- Controllers can group related request handling logic into a single class. For example, a UserController class might handle all incoming requests related to users, including showing, creating, updating, and deleting users.
- A model is used as a way for questioning data to and from the table within the database.

Code snippets of Flutter UI files:

- **Main_layout.dart**

```
import 'package:doctor_app1/screens/appointment_page.dart';
import 'package:doctor_app1/screens/fav_page.dart';
import 'package:doctor_app1/screens/home_page.dart';
import 'package:doctor_app1/screens/profile_page.dart';
import 'package:flutter/material.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';

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

  @override
  State<MainLayout> createState() => _MainLayoutState();
}

class _MainLayoutState extends State<MainLayout> {
  //variable declaration
  int currentPage = 0;
  final PageController _page = PageController();


  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: PageView(
        controller: _page,
        onPageChanged: ((value) {
          setState(() {
            //update page index when tab pressed / swich page
            currentPage = value;
          });
        }),
        children: const <Widget>[
          HomePage(),
          FavPage(),
          AppointmentPage(),
          ProfilePage(),
        ],
      ),
      bottomNavigationBar: BottomNavigationBar(
        currentIndex: currentPage,
        onTap: (page) {
          setState(() {
            currentPage = page;
            _page.animateToPage(
              page,
              duration: const Duration(milliseconds: 500),
              curve: Curves.easeInOut,
            );
          });
        },
        items: const <BottomNavigationBarItem>[
          BottomNavigationBarItem(
            icon: FaIcon(FontAwesomeIcons.houseChimneyMedical),
            label: 'Home',
          ),
          BottomNavigationBarItem(
            icon: FaIcon(FontAwesomeIcons.solidHeart),
            label: 'Favourite',
          ),
          BottomNavigationBarItem(
            icon: FaIcon(FontAwesomeIcons.solidCalendarCheck),
            label: 'Appointments',
          ),
          BottomNavigationBarItem(
            icon: FaIcon(FontAwesomeIcons.solidUser),
            label: 'Profile',
          ),
        ],
            ),
          );
        }
      }
```

- **Appointment _page**

```
import 'dart:convert';
import 'package:doctor_app1/providers/dio_provider.dart';
import 'package:doctor_app1/utils/config.dart';
import 'package:flutter/material.dart';
import 'package:shared_preferences/shared_preferences.dart';

class AppointmentPage extends StatefulWidget {
  const AppointmentPage({Key? key}) : super(key: key);
  @override
  State<AppointmentPage> createState() => _AppointmentPageState();
}

enum FilterStatus { upcoming, complete, cancel }

class _AppointmentPageState extends State<AppointmentPage> {
  FilterStatus status = FilterStatus.upcoming; //Initial status
  Alignment alignment = Alignment.centerLeft;
  List<dynamic> schedules = [];

  //for getting the appointment details
  Future<void> getAppointments() async {
    final SharedPreferences prefs = await SharedPreferences.getInstance();
    final token = prefs.getString('token') ?? '';
    final appointment = await DioProvider().getAppointments(token);
    if (appointment != 'Error') {
      setState(() {
        schedules = json.decode(appointment);
        print(schedules);
      });
    }
  }
```

```
@override
void initState() {
  getAppointments();
  super.initState();
}

@override
Widget build(BuildContext context) {
  //In this appointment page there are three states: upcoming, complete and cancel.
  //this returns filtered appointment.
  List<dynamic> filteredSchedules = schedules.where((var schedule) {
    switch (schedule['status']) {
      case 'upcoming':
        schedule["status"] = FilterStatus.upcoming;
        break;
      case 'complete':
        schedule["status"] = FilterStatus.complete;
        break;
      case 'cancel':
        schedule["status"] = FilterStatus.cancel;
        break;
    }
    return schedule['status'] == status;
  }).toList();


  return SafeArea(
    child: Padding(
      padding: const EdgeInsets.only(left: 20, top: 20, right: 20),
      child: Column(
          crossAxisAlignment: CrossAxisAlignment.stretch,
          children: <Widget>[
            const Text(
              'Appointment Schedule',
              textAlign: TextAlign.center,
              style: TextStyle(
                fontSize: 18,
                fontWeight: FontWeight.bold,
              ),
            ),
Config.spaceSmall,
Stack(
  children: [
    Container(
      width: double.infinity,
      height: 40,
      decoration: BoxDecoration(
          color: Colors.white,
          borderRadius: BorderRadius.circular(20)),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          //filter Tabs
          for (FilterStatus filterStatus in FilterStatus.values)
            Expanded(
              child: GestureDetector(
                onTap: () {
                  setState(() {
                    if (filterStatus == FilterStatus.upcoming) {
                      status = FilterStatus.upcoming;
                      alignment = Alignment.centerLeft;
                    } else if (filterStatus ==
                        FilterStatus.complete) {
                      status = FilterStatus.complete;
                      alignment = Alignment.center;
                    } else if (filterStatus ==
                        FilterStatus.cancel) {
                      status = FilterStatus.cancel;
                      alignment = Alignment.centerRight;
                    }
                  });
                },
                child: Center(child: Text(filterStatus.name)),
              ),
            ),
        ],
      ),
    ),
    AnimatedAlign(
      alignment: alignment,
      duration: const Duration(milliseconds: 200),
      child: Container(
        width: 100,
        height: 40,
        decoration: BoxDecoration(
          color: Config.primaryColor,
          borderRadius: BorderRadius.circular(20),
        ),
        child: Center(
          child: Text(
            status.name,
            style: const TextStyle(
              color: Colors.white,
              fontWeight: FontWeight.bold,
            ),
          ),
        ),
      ),
    )
  ],
),
Config.spaceSmall,
Expanded(
  child: ListView.builder(
    itemCount: filteredSchedules.length,
    itemBuilder: ((context, index) {
      var schedule = filteredSchedules[index];
      bool isLastElement = filteredSchedules.length + 1 == index;
      return Card(
        shape: RoundedRectangleBorder(
          side: const BorderSide(
            color: Colors.grey,
          ),
          borderRadius: BorderRadius.circular(20),
        ),
        margin: !isLastElement
            ? const EdgeInsets.only(bottom: 20)
            : EdgeInsets.zero,
        child: Padding(
          padding: const EdgeInsets.all(15),
child: Column(
  crossAxisAlignment: CrossAxisAlignment.stretch,
  children: [
    Row(
      children: [
        CircleAvatar(
          backgroundImage: NetworkImage(
              "http://192.168.0.195:8000${schedule['doctor_profile']}"),
        ),
        const SizedBox(
          width: 10,
        ),
        Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text(
              schedule['doctor_name'],
              style: const TextStyle(
                color: Colors.black,
                fontWeight: FontWeight.w700,
              ),
            ),
            const SizedBox(
              height: 5,
            ),
            Text(
              schedule['category'],
              style: const TextStyle(
                color: Colors.grey,
                fontSize: 12,
                fontWeight: FontWeight.w600,
              ),
            ),
          ],
        ),
      ],
    ),
    const SizedBox(
      height: 15,
    ),
  //Schedule Card
  ScheduleCard(
    date: schedule['date'],
    day: schedule['day'],
    time: schedule['time'],
  ),
  const SizedBox(
    height: 15,
  ),
  Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      Expanded(
        child: OutlinedButton(
          onPressed: () {},
          child: const Text(
            'Cancel',
            style:
                TextStyle(color: Config.primaryColor),
          ),
        ),
      ),
      const SizedBox(
        width: 20,
      ),
      Expanded(
        child: OutlinedButton(
          style: OutlinedButton.styleFrom(
            backgroundColor: Config.primaryColor,
          ),
          onPressed: () {},
          child: const Text(
            'Reschedule',
            style: TextStyle(color: Colors.white),
          ),
        ),
      ),
    ],
  ),
],
                    ),
                  ),
                );
              }),
            ),
          )
        ]),
      ),
    );
  }}
class ScheduleCard extends StatelessWidget {
  const ScheduleCard(
      {Key? key, required this.date, required this.day, required this.time})
      : super(key: key);

  final String date;
  final String day;
  final String time;

  @override
  Widget build(BuildContext context) {
    return Container(
        decoration: BoxDecoration(
          color: Colors.grey.shade200,
          borderRadius: BorderRadius.circular(10),
        ),
        width: double.infinity,
        padding: const EdgeInsets.all(20),
        child: Row(
          crossAxisAlignment: CrossAxisAlignment.center,
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: <Widget>[
            const Icon(
              Icons.calendar_today,
              color: Config.primaryColor,
              size: 15,
            ),
            const SizedBox(
              width: 5,
            ),
            Text(
              '$day, $date',
              style: const TextStyle(color: Config.primaryColor),
            ),
            const SizedBox(
              width: 20,
            ),
            const Icon(
              Icons.access_alarm,
              color: Config.primaryColor,
              size: 17,
            ),
            const SizedBox(
              width: 5,
            ),
            Flexible(
                child: Text(
              time,
              style: const TextStyle(color: Config.primaryColor),
            )),
          ],
        ));
  }
}
```

- **Auth page**

```
import 'package:doctor_app1/components/login_form.dart';
import 'package:doctor_app1/components/sign_up_form.dart';
import 'package:doctor_app1/components/social_button.dart';
import 'package:doctor_app1/utils/config.dart';
import 'package:doctor_app1/utils/text.dart';
import 'package:flutter/material.dart';

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

  @override
  State<AuthPage> createState() => _AuthPageState();
}

class _AuthPageState extends State<AuthPage> {
  bool isSignIn = true;
  @override
  Widget build(BuildContext context) {
    Config().init(context);
    //Login text field
    return Scaffold(
        body: SingleChildScrollView(
      padding: const EdgeInsets.symmetric(
        horizontal: 15,
        vertical: 15,
      ),
      child: SafeArea(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.start,
crossAxisAlignment: CrossAxisAlignment.start,
children: <Widget>[
 Text(
  AppText.enText['welcome_text']!,
  style: const TextStyle(
   fontSize: 30,
   fontWeight: FontWeight.bold,
  ),
 ),
 Config.spaceSmall,
 Text(
  isSignIn
     ? AppText.enText['signIn_text']!
     : AppText.enText['register_text']!,
  style: const TextStyle(
   fontSize: 16,
   fontWeight: FontWeight.bold,
  ),
 ),
 Config.spaceSmall,
 //Login Components
 isSignIn ? LoginForm() : SignUpForm(),
 Config.spaceSmall,
 isSignIn
   ? Center(
      child: TextButton(
        onPressed: () {},
        child: Text(
         AppText.enText['forgot-password']!,
         style: const TextStyle(
           fontSize: 17,
                  fontWeight: FontWeight.bold,
                  color: Colors.black),
            )),
      )
    : Container(),
//Social Account sign-in
//const Spacer(),
Center(
  child: Text(
    AppText.enText['social-login']!,
    style: TextStyle(
      fontSize: 16,
      fontWeight: FontWeight.normal,
      color: Colors.grey.shade500,
    ),
  ),
),
Config.spaceSmall,
Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: const <Widget>[
    SocialButton(social: 'google'),
    SocialButton(social: 'facebook'),
  ],
),
Config.spaceSmall,
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: <Widget>[
    Text(
      isSignIn
```

```
                      ? AppText.enText['signUp_text']!
                      : AppText.enText['registered_text']!,
                  style: TextStyle(
                    fontSize: 16,
                    fontWeight: FontWeight.normal,
                    color: Colors.grey.shade500,
                  ),
                ),
                TextButton(
                  onPressed: () {
                    setState(() {
                      isSignIn = !isSignIn;
                    });
                  },
                  child: Text(
                    isSignIn ? 'Sign Up' : 'Sign In',
                    style: const TextStyle(
                      fontSize: 16,
                      fontWeight: FontWeight.bold,
                      color: Colors.black,
                    ),
                  ),
                ),
              ],
            )
          ],
        ),
      ),
    ));
  }
}
```

- **BOOKING PAGE**

```
import 'package:doctor_app1/components/button.dart';
import 'package:doctor_app1/components/custom_appbar.dart';
import 'package:doctor_app1/utils/config.dart';
import 'package:doctor_app1/main.dart';
import 'package:doctor_app1/providers/dio_provider.dart';
import 'package:flutter/material.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:table_calendar/table_calendar.dart';
import 'package:doctor_app1/models/booking_datetime_converted.dart';

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

  @override
  State<BookingPage> createState() => _BookingPageState();
}

class _BookingPageState extends State<BookingPage> {
  CalendarFormat _format = CalendarFormat.month;
  DateTime _focusDay = DateTime.now();
  DateTime _currentDay = DateTime.now();
  int? _currentIndex;
  bool _isWeekend = false;
bool _dateSelected = false;
bool _timeSelected = false;
String? token; //get token for inserting booking date n time into database


Future<void> getToken() async {
  final SharedPreferences prefs = await SharedPreferences.getInstance();
  token = prefs.getString('token') ?? '';
}


@override
void initState() {
  getToken();
  super.initState();
}


@override
Widget build(BuildContext context) {
  Config().init(context);
  final doctor = ModalRoute.of(context)!.settings.arguments as Map;
  return Scaffold(
    appBar: CustomAppBar(
      appTitle: 'Appointment',
      icon: const FaIcon(Icons.arrow_back_ios),
    ),
body: CustomScrollView(
  slivers: <Widget>[
    SliverToBoxAdapter(
      child: Column(
        children: <Widget>[
          //table calendar
          _tableCalendar(),
          const Padding(
            padding: EdgeInsets.symmetric(horizontal: 10, vertical: 25),
            child: Center(
              child: Text(
                'Select Consultaion Time',
                style: TextStyle(
                  fontWeight: FontWeight.bold,
                  fontSize: 20,
                ),
              ),
            ),
          ),
        ],
      ),
    ),
    _isWeekend
        ? SliverToBoxAdapter(
            child: Container(
              padding: const EdgeInsets.symmetric(
                  horizontal: 10, vertical: 30),
      alignment: Alignment.center,
      child: const Text(
        'Weekend is not availabel. Select another date.',
        style: TextStyle(
          fontSize: 18,
          fontWeight: FontWeight.bold,
          color: Colors.grey,
        ),
      ),
    ),
  )
: SliverGrid(
    delegate: SliverChildBuilderDelegate(
      (context, index) {
        return InkWell(
          splashColor: Colors.transparent,
          onTap: () {
            setState(() {
              //when selected update index and set time selected to true
              _currentIndex = index;
              _timeSelected = true;
            });
          },
          child: Container(
            margin: const EdgeInsets.all(5),
            decoration: BoxDecoration(
              border: Border.all(
          color: _currentIndex == index
            ? Colors.white
            : Colors.black,
         ),
         borderRadius: BorderRadius.circular(15),
         color: _currentIndex == index
           ? Config.primaryColor
           : null,
        ),
        alignment: Alignment.center,
        child: Text(
         '${index + 9}:00 ${index + 9 > 11 ? "PM" : "AM"}',
         style: TextStyle(
          fontWeight: FontWeight.bold,
          color:
            _currentIndex == index ? Colors.white : null,
         ),
        ),
       ),
      );
     },
     childCount: 8,
    ),
    gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
      crossAxisCount: 4, childAspectRatio: 1.5),
   ),
SliverToBoxAdapter(
        child: Container(
          padding: const EdgeInsets.symmetric(horizontal: 10, vertical: 80),
          child: Button(
            width: double.infinity,
            title: 'Make Appointment',
            onPressed: () async {
              //press button here to store booking details like date and time
              final getDate = DateConverted.getDate(_currentDay);
              final getDay = DateConverted.getDay(_currentDay.weekday);
              final getTime = DateConverted.getTime(_currentIndex!);

              final booking = await DioProvider().bookAppointment(
                  getDate, getDay, getTime, doctor['doctor_id'], token!);

              //if booked return 200, then redirect to success booking page
              if (booking == 200) {
                MyApp.navigatorKey.currentState!
                    .pushNamed('success_booking');
              }
            },
            disable: _timeSelected && _dateSelected ? false : true,
          ),
        ),
      )
    ],
  ),
);
```

```
  }



//table calendar
  Widget _tableCalendar() {
    return TableCalendar(
      focusedDay: _focusDay,
      firstDay: DateTime.now(),
      lastDay: DateTime(2023, 12, 31),
      calendarFormat: _format,
      currentDay: _currentDay,
      rowHeight: 48,
      calendarStyle: const CalendarStyle(
        todayDecoration:
            BoxDecoration(color: Config.primaryColor, shape: BoxShape.circle),
      ),
      availableCalendarFormats: const {
        CalendarFormat.month: 'Month',
      },
      onFormatChanged: (format) {
        setState(() {
          _format = format;
        });
      },
      onDaySelected: ((selectedDay, focusedDay) {
        setState(() {
          _currentDay = selectedDay;
          _focusDay = focusedDay;
          _dateSelected = true;

          //checks if weekend is selected
          if (selectedDay.weekday == 6 || selectedDay.weekday == 7) {
            _isWeekend = true;
            _timeSelected = false;
            _currentIndex = null;
          } else {
            _isWeekend = false;
          }
        });
      }),
    );
  }
}
```

• DOCTOR DETAILS PAGE

```
import 'package:doctor_app1/components/button.dart';
import 'package:doctor_app1/utils/config.dart';
import 'package:flutter/material.dart';
import 'package:flutter/rendering.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import '../components//custom_appbar.dart';


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

 @override
 State<DoctorDetails> createState() => _DoctorDetailsState();
}

class _DoctorDetailsState extends State<DoctorDetails> {
 //for favourite button
 bool isFav = false;
 @override
 Widget build(BuildContext context) {
  //get arguments passed from doctor card
  final doctor = ModalRoute.of(context)!.settings.arguments as Map;
  return Scaffold(
   appBar: CustomAppBar(
  appTitle: 'Doctor Details',
  icon: const FaIcon(Icons.arrow_back_ios),
  actions: [
    IconButton(
      onPressed: () {
        setState(() {
          isFav = !isFav;
        });
      },
      icon: FaIcon(
        isFav ? Icons.favorite_rounded : Icons.favorite_outline,
        color: Colors.red,
      ),
    ),
  ],
),
body: SafeArea(
  child: Column(
    children: <Widget>[
      //Doctor  Information
      AboutDoctor(
        doctor: doctor,
      ),
      //details of doctor
      DetailBody(
        doctor: doctor,
      ),
            const Spacer(),
            Padding(
              padding: const EdgeInsets.all(10),
              child: Button(
                width: double.infinity,
                title: 'Book Appointment',
                onPressed: () {
                  //navigates to booking page
                  Navigator.of(context).pushNamed('booking_page',
                      arguments: {"doctor_id": doctor['doc_id']});
                },
                disable: false,
              ),
            )
          ],
        ),
      ),
    );
  }
}

  class AboutDoctor extends StatelessWidget {
  const AboutDoctor({super.key, required this.doctor});

  final Map<dynamic, dynamic> doctor;

  @override
```

```
Widget build(BuildContext context) {
 Config().init(context);
 return Container(
  width: double.infinity,
  child: Column(
   children: <Widget>[
    CircleAvatar(
     radius: 30.0,
     backgroundImage: NetworkImage(
      "http://192.168.0.195:8000${doctor['doctor_profile']}",
     ),
     //backgroundImage: AssetImage('assets/default.jpg'),
     backgroundColor: Colors.white,
    ),
    Config.spaceMedium,
    Text(
     'Dr ${doctor['doctor_name']}',
     style: const TextStyle(
      color: Colors.black,
      fontSize: 19.0,
      fontWeight: FontWeight.bold,
     ),
    ),
    Config.spaceSmall,
    SizedBox(
     width: Config.widthSize * 0.95,
     child: const Text(
      //BDS - Bachelor in Dental Surgery
      'BDS (Nair Hospital Dental College, Mumbai), Dental Medicine (Florida International University, Miami)',
      style: TextStyle(
        color: Colors.grey,
        fontSize: 13,
      ),
      softWrap: true,
      textAlign: TextAlign.center,
    ),
  ),
  Config.spaceSmall,
  SizedBox(
    width: Config.widthSize * 0.95,
    child: const Text(
      //BDS - Bachelor in Dental Surgery
      'Jaslok Hospital and Research Center',
      style: TextStyle(
        color: Colors.black,
        fontWeight: FontWeight.bold,
        fontSize: 15,
      ),
      softWrap: true,
      textAlign: TextAlign.center,
    ),
  ),
],
      ),
    );
  }
}



class DetailBody extends StatelessWidget {
  const DetailBody({super.key, required this.doctor});

  final Map<dynamic, dynamic> doctor;


  @override
  Widget build(BuildContext context) {
    Config().init(context);
    return Container(
      padding: const EdgeInsets.all(20),
      margin: const EdgeInsets.only(bottom: 30),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.stretch,
        children: <Widget>[
          Config.spaceSmall,
          //doctor information
          DoctorInfo(
            patients: doctor['patients'],
            exp: doctor['experience'],
          ),
          Config.spaceMedium,
          const Text(
            'About Doctor',
            style: TextStyle(fontWeight: FontWeight.w600, fontSize: 18),
          ),
          Config.spaceSmall,
          Text(
            'Dr. ${doctor['doctor_name']} is a dedicated ${doctor['category']} Speciaist based in Jaslok, Mumbai. Graduated since 2008, and has been in practice ever since.',
            style: const TextStyle(
              fontWeight: FontWeight.w500,
              height: 1.5,
            ),
            softWrap: true,
            textAlign: TextAlign.justify,
          ),
        ],
      ),
    );
  }
}
 class DoctorInfo extends StatelessWidget {
 const DoctorInfo({super.key, required this.patients, required this.exp});

 final int patients;
 final int exp;
```

```
  @override
  Widget build(BuildContext context) {
    return Row(
      children: <Widget>[
        InfoCard(label: 'Patients', value: '$patients'),
        const SizedBox(
          width: 15,
        ),
        InfoCard(label: 'Experience', value: '$exp years'),
        const SizedBox(
          width: 15,
        ),
        const InfoCard(label: 'Rating', value: '4.0'),
      ],
    );
  }
}

class InfoCard extends StatelessWidget {
  const InfoCard({super.key, required this.label, required this.value});

  final String label;
  final String value;

  @override
  Widget build(BuildContext context) {
```

```
return Expanded(
  child: Container(
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(15),
      color: Config.primaryColor,
    ),
    padding: const EdgeInsets.symmetric(
      vertical: 15,
      horizontal: 7.5,
    ),
    child: Column(
      children: <Widget>[
        Text(
          label,
          style: const TextStyle(
            color: Colors.black,
            fontSize: 12,
            fontWeight: FontWeight.w600,
          ),
        ),
        const SizedBox(
          height: 5,
        ),
        Text(
          value,
          style: const TextStyle(
            color: Colors.white,
        fontSize: 15,
        fontWeight: FontWeight.w800,
       ),
      ),
     ],
    ),
   ),
  );
 }
}
```

- **Homepage.dart**

```
import 'dart:convert';


import 'package:dio/dio.dart';
import 'package:doctor_app1/components/appointment_card.dart';
import 'package:doctor_app1/components/doctor_card.dart';
import 'package:doctor_app1/providers/dio_provider.dart';
import 'package:doctor_app1/screens/appointment_page.dart';
import 'package:doctor_app1/utils/config.dart';
import 'package:flutter/material.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:shared_preferences/shared_preferences.dart';



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



  @override
  State<HomePage> createState() => _HomePageState();
}



class _HomePageState extends State<HomePage> {
  Map<String, dynamic> user = {};
```

```
 Map<String, dynamic> doctor = {};
 List<Map<String, dynamic>> medCat = [
   {
     "icon": FontAwesomeIcons.userDoctor,
     "category": "General", //general doctors
   },
   {
     "icon": FontAwesomeIcons.heartPulse,
     "category":
         "Cardiology", //doctors who specialize in the treatment of heart disease
   },
   {
     "icon": FontAwesomeIcons.teeth,
     "category": "Dentistry", //dentists
   },
   {
     "icon": FontAwesomeIcons.earListen,
     "category":
         "Otolaryngology / ENT", //handle concerns relating to the ear, nose, throat, and related head and neck areas.
   },
   {
     "icon": FontAwesomeIcons.personPregnant,
     "category": "Obstetrics & gynaecology",
   },
   /*{
    "icon":FontAwesomeIcons.person,
    "category": "Neurologists",
   },*/
   {
     "icon": FontAwesomeIcons.hand,
     "category": "Dermatology", //treat skin related disorders
   },
   {
     "icon": FontAwesomeIcons.children,
     "category":
         "Paediatrics", //deals with the medical treatment of newborns, children, and adolescents.
   },
   {
     "icon": FontAwesomeIcons.brain,
     "category":
         "Psychiatry", // doctors who specialize in mental health, which includes substance abuse disorders
   },
   {
     "icon": FontAwesomeIcons.xRay,
     "category": "Radiology", //carry out X-ray , MRI etc
   },
   {
     "icon": FontAwesomeIcons.lungs,
     "category":
         "Pulmonology", //specialize in the diagnosis and treatment of respiratory illnesses in both men and women.
   },
   {
     "icon": FontAwesomeIcons.bone,
     "category":
         "Orthopaedic surgery", //treat disorders that affect the bones, joints, ligaments muscles etc
   },
 ];


 Future<void> getData() async {
   //get token from shared pref.
   final SharedPreferences prefs = await SharedPreferences.getInstance();
   final token = prefs.getString('token') ?? ' ';


   if (token.isNotEmpty && token != ' ') {
     //get user data
     final response = await DioProvider().getUser(token);
     if (response != null) {
       setState(() {
         //json encode
         print(response.toString());
         user = json.decode(response.toString());


         //check if any appointment today
         for (var doctorData in user['doctor']) {
           if (doctorData['appointments'] != null) {
             doctor = doctorData;
           }
         }
       });
     }
   }
 }


 @override
 void initState() {
   getData(); //do again
   super.initState();
 }


 @override
 Widget build(BuildContext context) {
   Config().init(context);


   return Scaffold(
     //if user is empty, then return progress indicator
     body: user.isEmpty
         ? const Center(
             child: CircularProgressIndicator(),
           )
         : Padding(
 padding: const EdgeInsets.symmetric(
   horizontal: 15,
   vertical: 15,
 ),
 child: SafeArea(
   child: SingleChildScrollView(
     child: Column(
       mainAxisAlignment: MainAxisAlignment.start,
       crossAxisAlignment: CrossAxisAlignment.start,
       children: <Widget>[
         Row(
           mainAxisAlignment: MainAxisAlignment.spaceBetween,
           children: <Widget>[
             Text(
               user['name'],
               style: const TextStyle(
                 fontSize: 24,
                 fontWeight: FontWeight.bold,
               ),
             ),
             const SizedBox(
               child: CircleAvatar(
                 radius: 30,
                 backgroundImage: AssetImage(
                     'assets/profilepic2.jpeg'), // insert the user image here
               ),
             )
   ],
 ),


 Config.spaceMedium,
 //category listing
 const Text(
   'category',
   style: TextStyle(
     fontSize: 16,
     fontWeight: FontWeight.bold,
   ),
 ),


 Config.spaceSmall,
 //refer to list from line 13
 SizedBox(
   height: Config.heightSize * 0.05,
   child: ListView(
     scrollDirection: Axis.horizontal,
     children:
         List<Widget>.generate(medCat.length, (index) {
       return Card(
         margin: const EdgeInsets.only(right: 20),
         color: Config.primaryColor,
         child: Padding(
             padding: const EdgeInsets.symmetric(
               horizontal: 15, vertical: 9),
             child: Row(
               mainAxisAlignment:
                 MainAxisAlignment.spaceAround,
               children: <Widget>[
                FaIcon(
                  medCat[index][
                    'icon'], //for displaying the icons based on med category eg : .lungs
                  color: Colors.white,
                ),
                const SizedBox(
                  width: 20,
                ),
                Text(
                  medCat[index][
                    'category'], //for displaying the name of the  med category in the icons
                  style: const TextStyle(
                    fontSize: 16,
                    color: Colors.white,
                  ),
                ),
               ]),
            ),
           );
         }),
        ),
 ),


 Config.spaceSmall,
 const Text(
   'Appointment Today',
   style: TextStyle(
     fontSize: 16,
     fontWeight: FontWeight.bold,
   ),
 ),


 Config.spaceSmall,
 //display appointment card here. Creating appintment card widget
 doctor.isNotEmpty
     ? AppointmentCard(
         doctor: doctor,
         color: Config.primaryColor,
       )
     : Container(
         width: double.infinity,
         decoration: BoxDecoration(
             color: Colors.grey.shade300,
             borderRadius: BorderRadius.circular(10)),
         child: const Center(
           child: Padding(
             padding: EdgeInsets.all(20),
             child: Text(
               'No Appointment Today',
               style: TextStyle(
                   fontSize: 16,
                   fontWeight: FontWeight.w600),
             ),
           ),
         ),
       ),


 Config.spaceSmall,
 const Text(
   'Top Doctors',
   style: TextStyle(
     fontSize: 16,
     fontWeight: FontWeight.bold,
   ),
 ),


 //list of top doctors
 //building doctors card here
 Config.spaceSmall,
 Column(
   children: List.generate(user['doctor'].length, (index) {
```

```
             return DoctorCard(
              route: 'doc_details',
              doctor: user['doctor'][index],
             );
            }),
           ),
          ],
         ),
        ),
       ),
      ),
  );
 }
}
```

- **Main.dart**

```
import 'package:doctor_app1/models/auth_model.dart';
import 'package:doctor_app1/main_layout.dart';
import 'package:doctor_app1/screens/auth_pg.dart';
import 'package:doctor_app1/screens/booking_page.dart';
import 'package:doctor_app1/screens/doctor_details.dart';
import 'package:doctor_app1/screens/success_booked.dart';
import 'package:doctor_app1/utils/config.dart';
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';


void main() {
  runApp(const MyApp());
}


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


  //for push navigator
  static final navigatorKey = GlobalKey<NavigatorState>();


  @override
  Widget build(BuildContext context) {
    //Defining Theme data here
    return ChangeNotifierProvider<AuthModel>(
      create: (context) => AuthModel(),
      child: MaterialApp(
        navigatorKey: navigatorKey,
        title: 'Santé',
        theme: ThemeData(
            //pre-define input decoration
            inputDecorationTheme: const InputDecorationTheme(
              focusColor: Config.primaryColor,
              border: Config.outlinedBorder,
              focusedBorder: Config.focusBorder,
          errorBorder: Config.errorBorder,
          enabledBorder: Config.outlinedBorder,
          floatingLabelStyle: TextStyle(color: Config.primaryColor),
          prefixIconColor: Colors.black38,
        ),
        scaffoldBackgroundColor: Colors.white,
        bottomNavigationBarTheme: BottomNavigationBarThemeData(
          backgroundColor: Config.primaryColor,
          selectedItemColor: Colors.white,
          showSelectedLabels: true,
          showUnselectedLabels: false,
          unselectedItemColor: Colors.grey.shade700,
          elevation: 10,
          type: BottomNavigationBarType.fixed,
        )),
      initialRoute: '/',
      routes: {
        //Initial route of the app i.e. the auth pg
        '/': (context) => const AuthPage(),
        //for main layout after login
        'main': (context) => const MainLayout(),
        //for doctor details after the doctor card
        'doc_details': (context) => const DoctorDetails(),
        'booking_page': (context) => BookingPage(),
        'success_booking': (context) => const AppointmentBooked(),
      },
    ),
  );
 }
}
```

- **Success of booking**

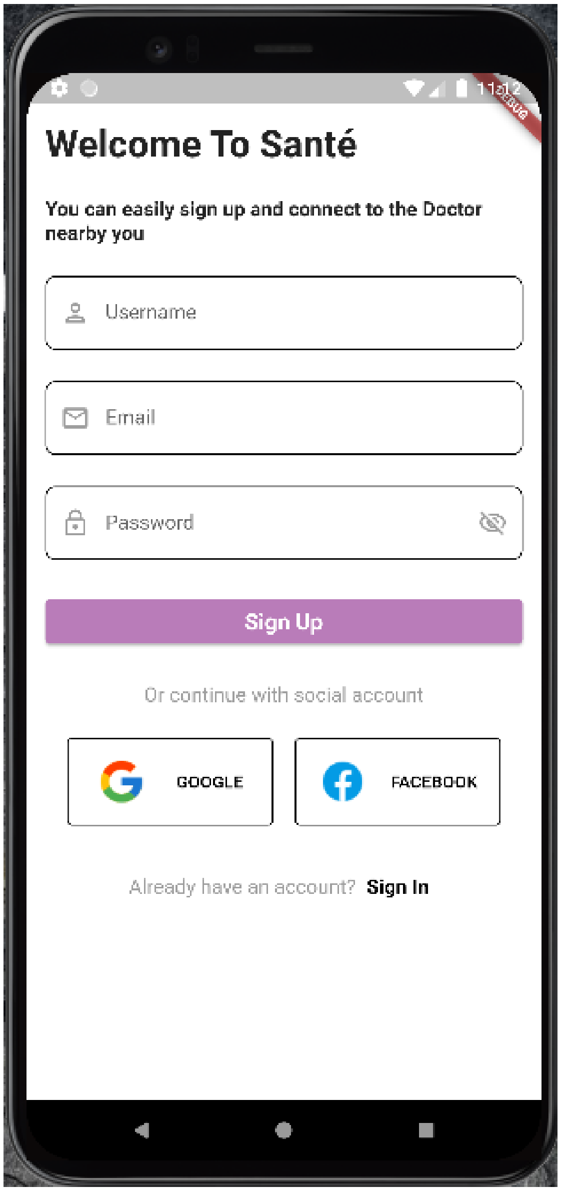
```
import 'package:flutter/material.dart';
import 'package:lottie/lottie.dart';
import '../components/button.dart';

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SafeArea(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Expanded(
              flex: 3,
              child: Lottie.asset('assets/success.json'),
            ),
            Container(
              width: double.infinity,
              alignment: Alignment.center,
              child: const Text(
                'Booking Successful',
                style: TextStyle(
                  fontSize: 20,
                  fontWeight: FontWeight.bold,
                ),
              ),
            ),
            const Spacer(),
            //To go back to home page
            Padding(
              padding: const EdgeInsets.symmetric(horizontal: 10, vertical: 15),
              child: Button(
                width: double.infinity,
                title: 'Back to Home Page',
                onPressed: () => Navigator.of(context).pushNamed('main'),
                disable: false,
              ),
            )
          ],
        ),
      ),
    );
  }
}
```

10 SCREENSHOTS

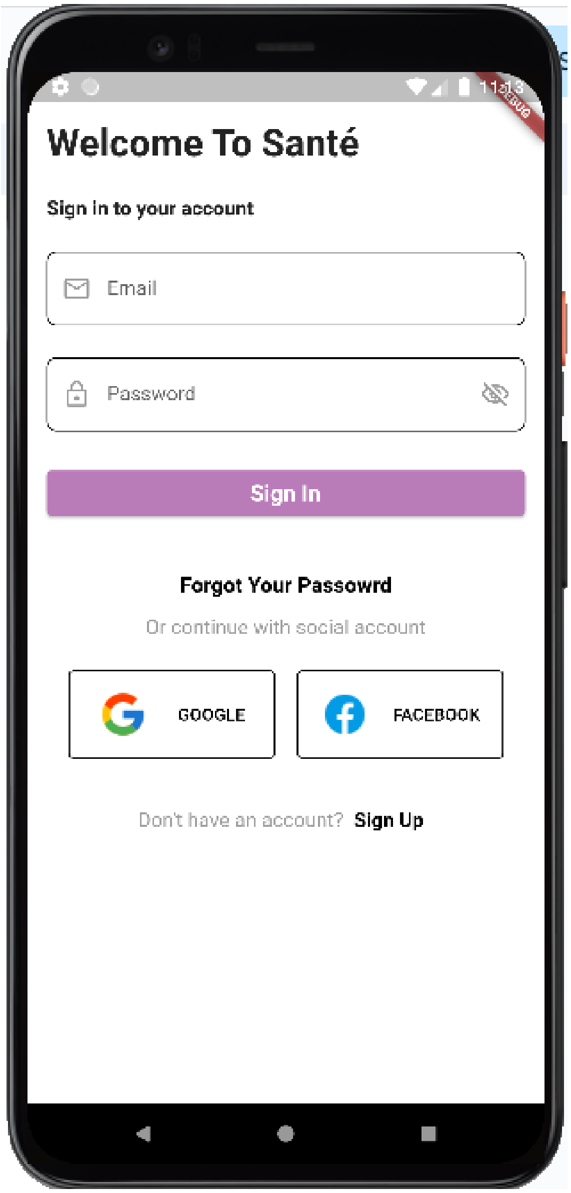
- **SIGN IN/ SIGN UP PAGE**



Welcome To Santé

You can easily sign up and connect to the Doctor nearby you

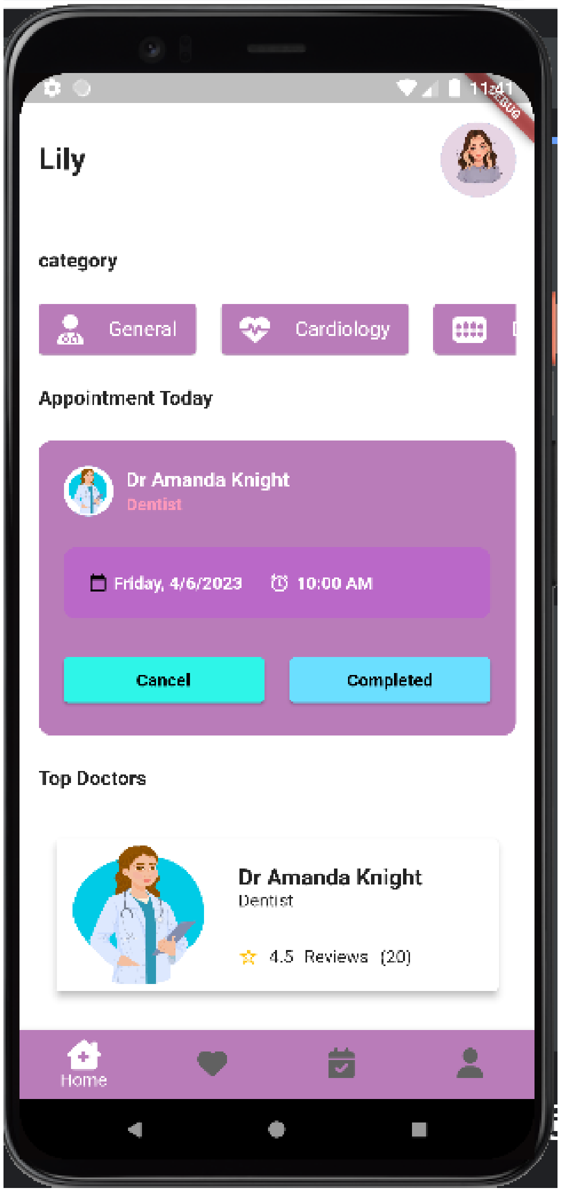
11:12
DEBUG
Username
Email
Password
Sign Up
Or continue with social account
GOOGLE
FACEBOOK
Already have an account? Sign In



Welcome To Santé

Sign in to your account

- **HOME PAGE**



Lily

category

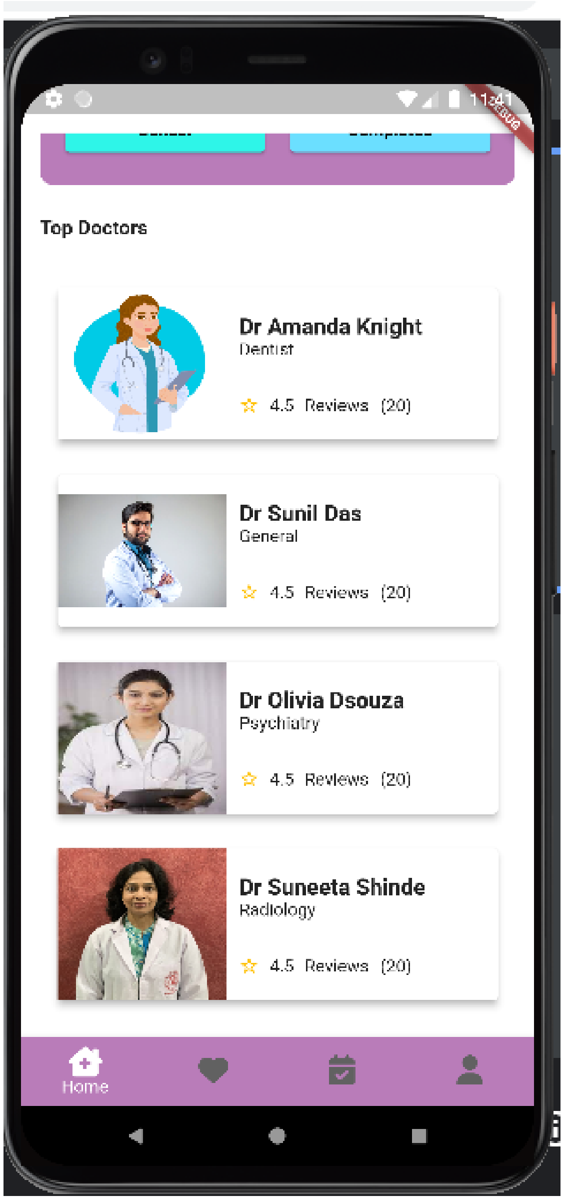
Appointment Today

Dr Amanda Knight
Dentist

Friday, 4/6/2023 10:00 AM

Top Doctors

Dr Amanda Knight
Dentist
☆ 4.5 Reviews (20)



Top Doctors

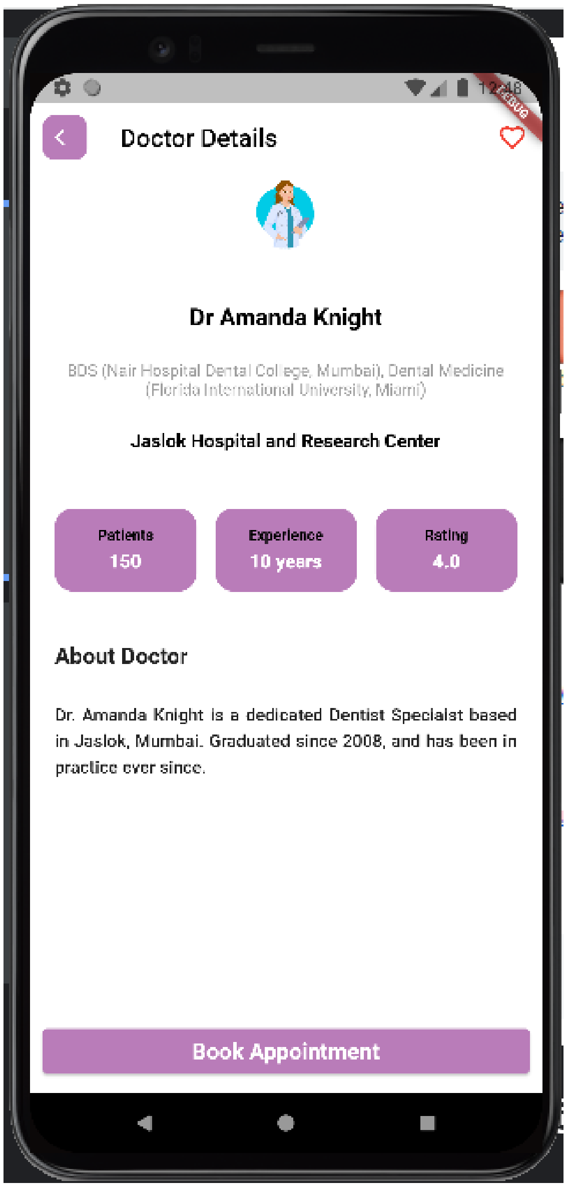
Dr Amanda Knight
Dentist
☆ 4.5 Reviews (20)

Dr Sunil Das
General
☆ 4.5 Reviews (20)

Dr Olivia Dsouza
Psychiatry
☆ 4.5 Reviews (20)

Dr Suneeta Shinde
Radiology
☆ 4.5 Reviews (20)

- **Doctor details page**



Dr Amanda Knight

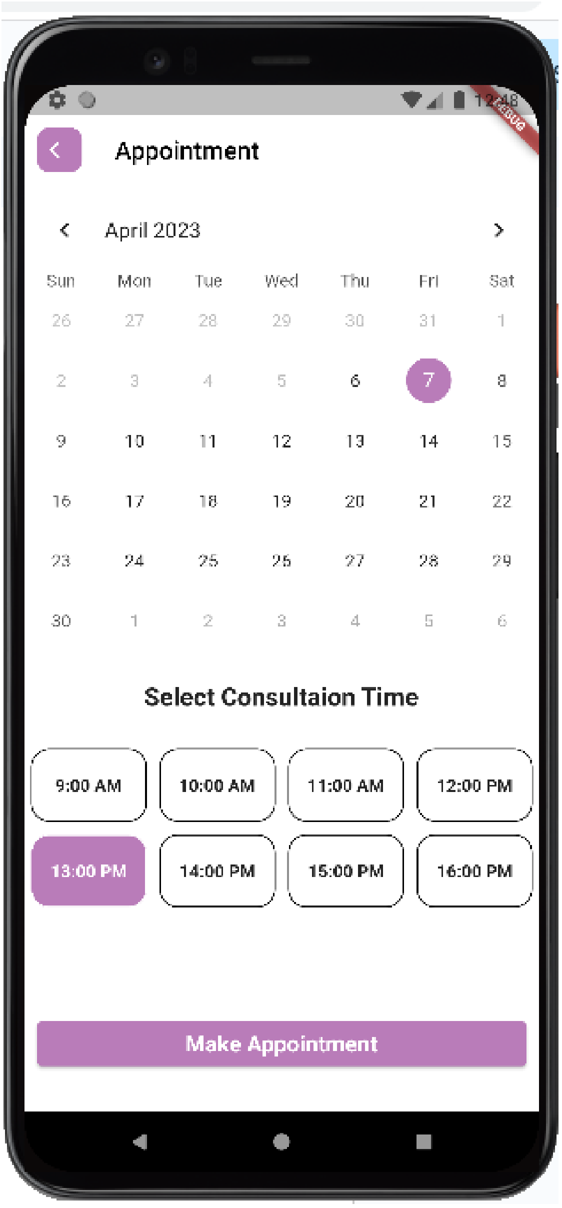
BDS (Nair Hospital Dental College, Mumbai), Dental Medicine (Florida International University, Miami)

Jaslok Hospital and Research Center

Patients	Experience	Rating
150	10 years	4.0

About Doctor

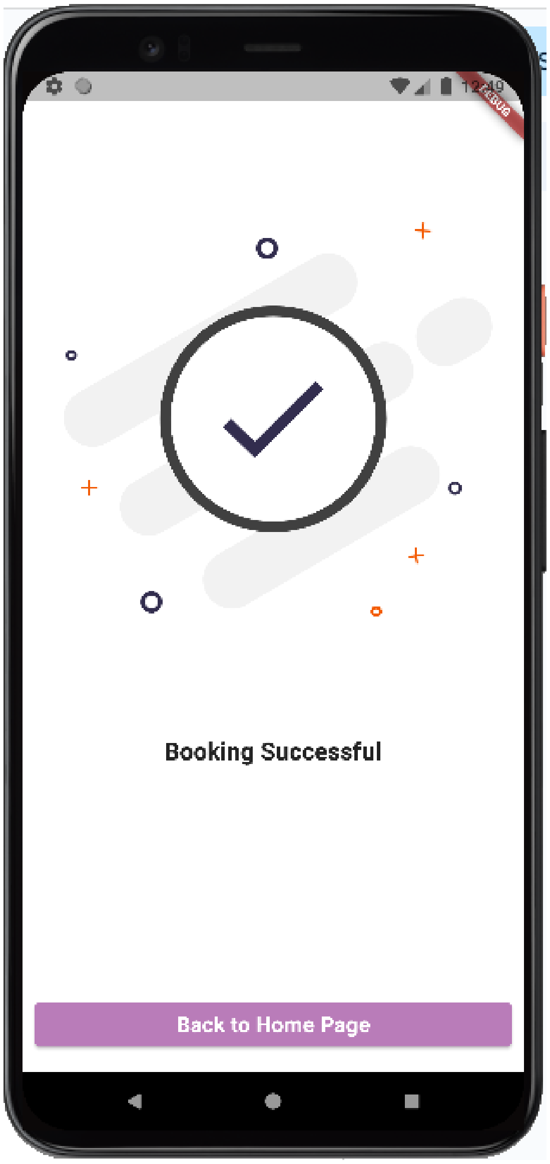
Dr. Amanda Knight is a dedicated Dentist Specialst based in Jaslok, Mumbai. Graduated since 2008, and has been in practice ever since.



April 2023

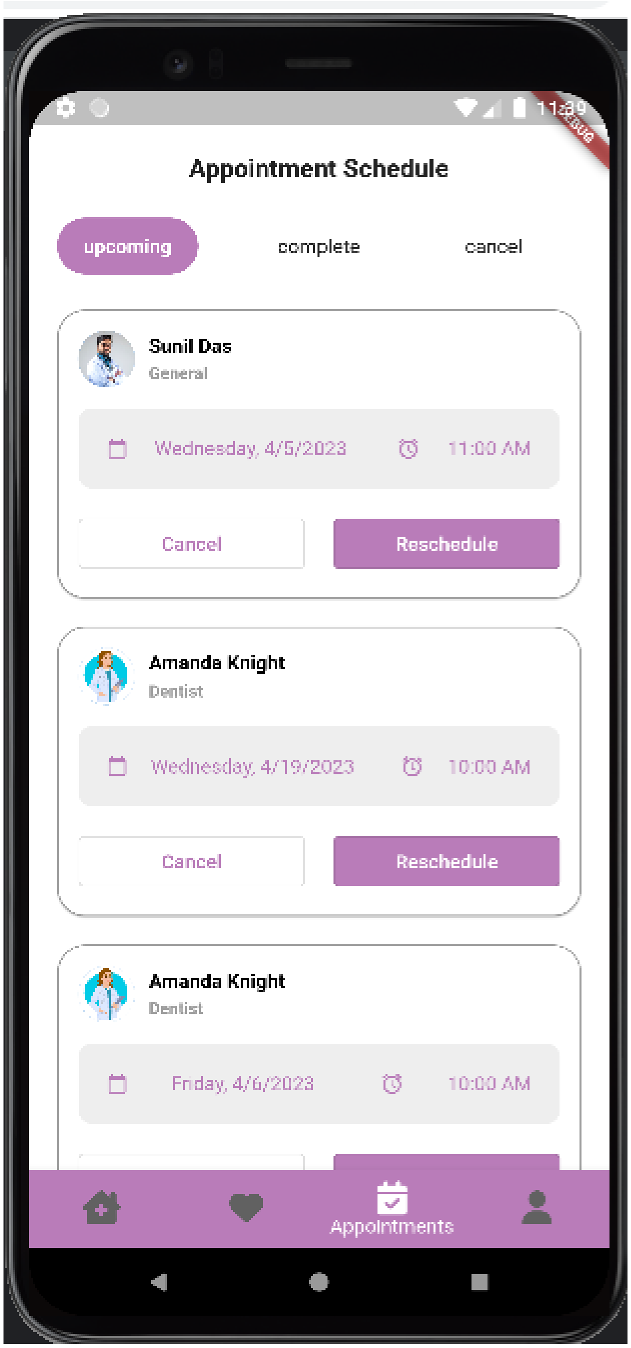
Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

Select Consultaion Time



Booking Successful

- **Appointments page**



Appointment Schedule

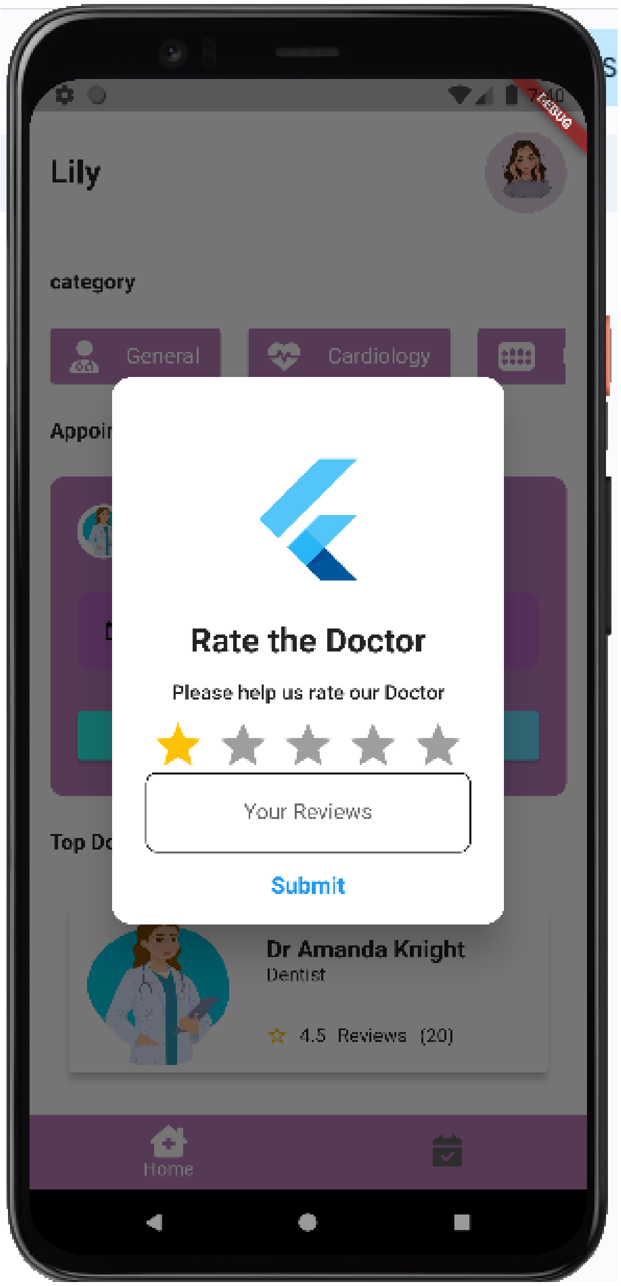
upcoming complete cancel

Sunil Das
General
Wednesday, 4/5/2023 11:00 AM
Cancel Reschedule

Amanda Knight
Dentist
Wednesday, 4/19/2023 10:00 AM
Cancel Reschedule

Amanda Knight
Dentist
Friday, 4/6/2023 10:00 AM

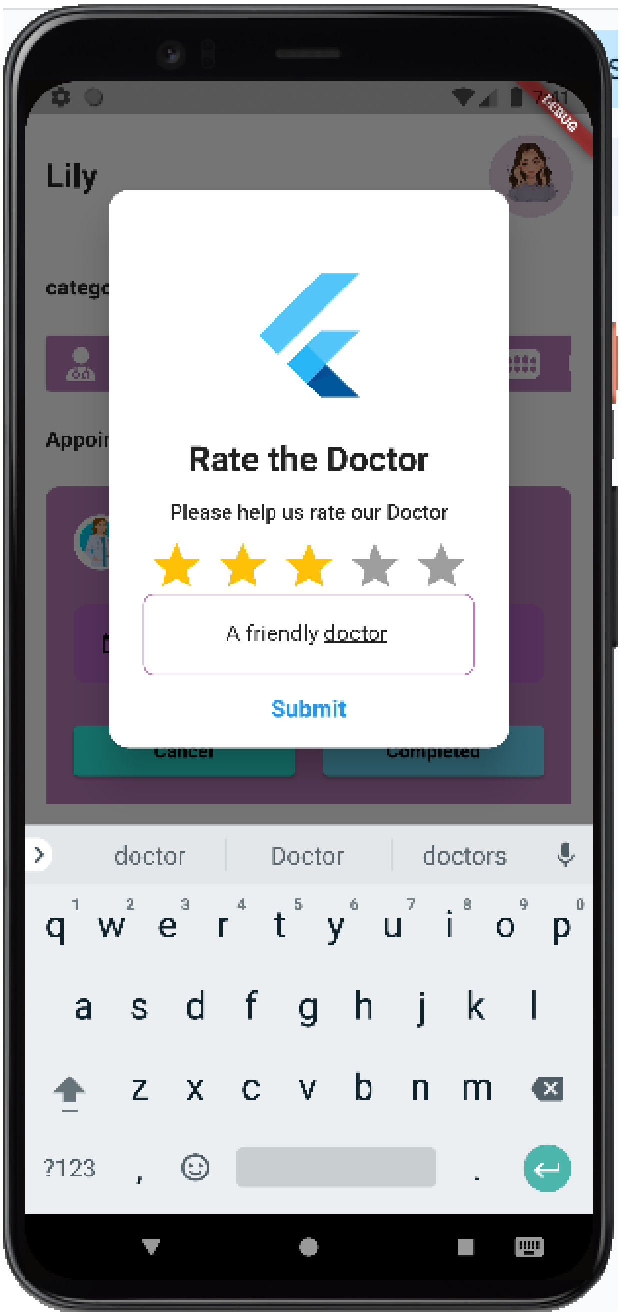
- **RATINGS/REVIEWS**



Rate the Doctor

Please help us rate our Doctor

Your Reviews

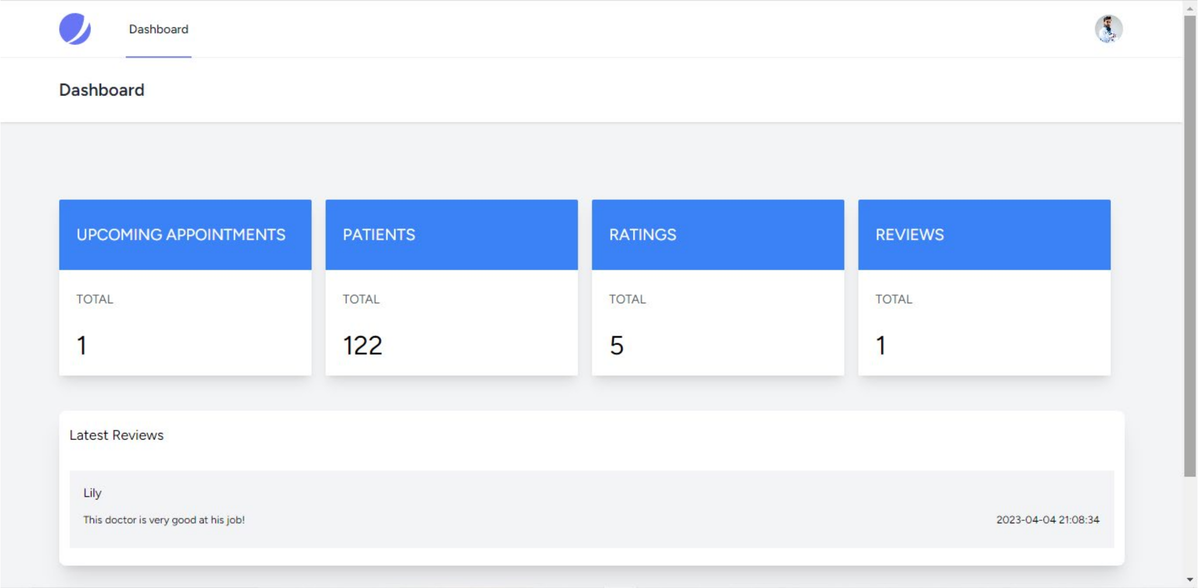


Rate the Doctor

Please help us rate our Doctor

A friendly doctor

- **DOCTOR DASHBOARD**



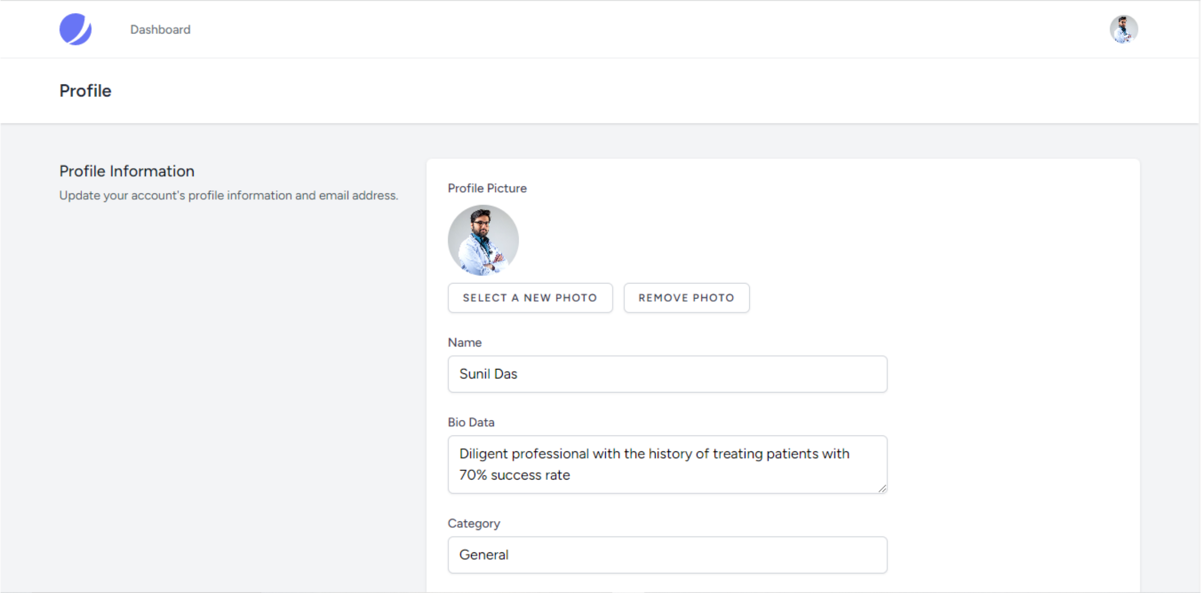
Dashboard

UPCOMING APPOINTMENTS	PATIENTS	RATINGS	REVIEWS
TOTAL	TOTAL	TOTAL	TOTAL
1	122	5	1

Latest Reviews

Lily

This doctor is very good at his job! 2023-04-04 21:08:34



Profile

Profile Information

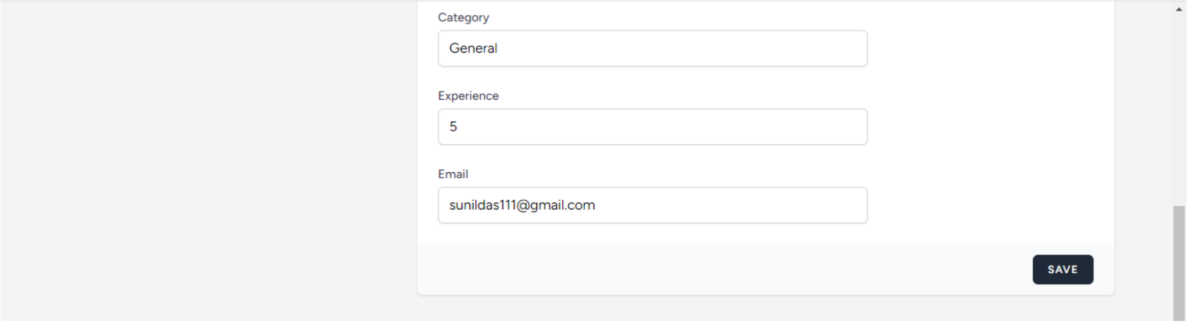
Update your account's profile information and email address.

Profile Picture

Name: Sunil Das

Bio Data: Diligent professional with the history of treating patients with 70% success rate

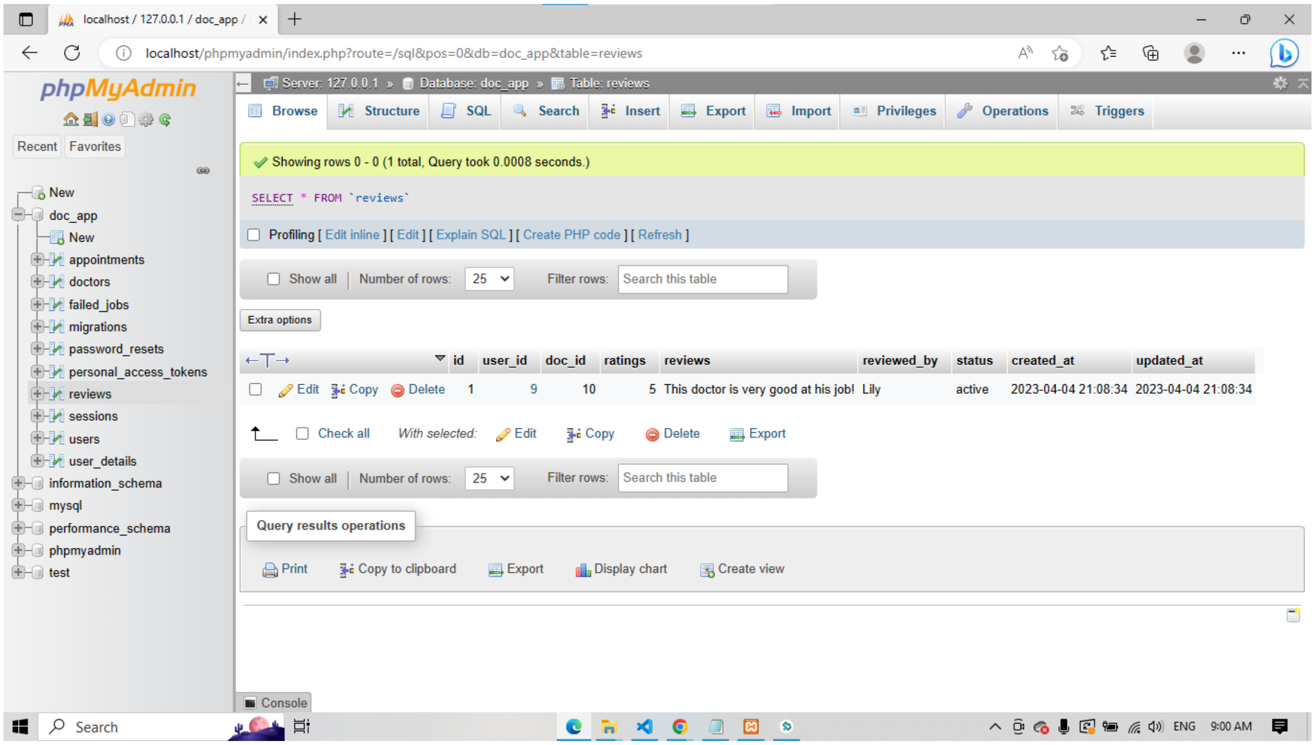
Category: General


Category: General

Experience: 5

Email: sunildas111@gmail.com

- **DATABASE**



localhost/phpmyadmin/index.php?route=/sql&pos=0&db=doc_app&table=reviews

✓ Showing rows 0 - 0 (1 total, Query took 0.0008 seconds.)

```
SELECT * FROM `reviews`
```

id	user_id	doc_id	ratings	reviews	reviewed_by	status	created_at	updated_at
1	9	10	5	This doctor is very good at his job!	Lily	active	2023-04-04 21:08:34	2023-04-04 21:08:34

11 TESTING

SR. No	Description	Input	Actual O/P	Expected O/P	Status Pass/fail
1	When user click on the app the sign in / sign up page is displayed		Sign in /sign up page	As expected	Pass
2	When the email and password are entered and sign up in clicked	User email and password	Home page	As expected	Pass
3	When the sign up button is clicked .		Sign in page	As expected	Pass
4	When the details for sign in are entered	Name, email, password	Home page	As expected	Pass
5	When the user clicks on one of the doctors displayed in the list.	Click on a doctor	Doctor details page	As expected	Pass
6	When the use clicks on book appointment	Click on book appointment button	Calendar with the available time slots	As expected	Pass
7	When the user selects make appointment	Click on make appointment	Success page shows up	As expected	Pass

8	-When the user clicks on completed in the home page - When userclick on submit	Click on completed button Click on submit button	Rate the doctor and give the reviews window shows up Ratings should be stored in database	As expected Ratings do not get stored	Pass Fail

12 Advantages and Disadvantages of the system

Booking the doctors via online platforms comprised both advantages and disadvantages in parallel. Here, I can list the advantages and disadvantages clearly of these booking apps.

- **ADVANTAGES:**

Save the Hospital staff Time- The demand via phone booking systems require a dedicated staff to handle the bookings and entries. The arrival of online platforms reduces the time to spend on entries and makes them focus on further activities.

Book Anytime & Anywhere- Telephone booking systems mainly suffered by the issues in call management where the staff are unavailable to pick up the call during peak hours. Online platforms allow the patients to book the appointment anytime and anywhere via simple clicks. **Service Quality High**- With the customized booking options in the application, the number of patients who participated is more. Due to the instant on-boarding, the treatment quality is high. Results that are quick and immediate. There are a lot of options to pick from. The seekers can examine online certification and authentication for themselves. Effortless interaction with a simple UI. It is cost-effective and efficient. Services available 24 hours a day, 7 days a week.

- **DISADVANTAGES:**

Monetary cost High-The implementation cost of such platforms and the cost spent for the bug issues handling and support are high.

Consistency Internet Access- Online platforms are efficient only if the services can be widened across more professionals to provide high-quality treatment.

Data Theft- One of the main disadvantages is that data theft due to the insecure database for patients as well as patient and doctor details. There is no genuine word of mouth. It is entirely dependent on your Internet service.

Even though the booking of doctors via online platforms has both advantages and disadvantages, the usage is greatly observed in recent conditions.

13 BIBLIOGRAPHY AND REFERENCES

- A. *Doctor appointment app development: Benefits, features & cost*, *Demigos*. Available at: https://demigos.com/blog-post/doctor-appointment-app-development/ (Accessed: April 2, 2023).
- *PHP tutorial*. Available at: https://www.w3schools.com/php/ (Accessed: April 4, 2023).
- *Laravel - Overview Tutorials Point*. Available at: https://www.tutorialspoint.com/laravel/laravel_overview.htm (Accessed: March 16, 2023).
- *Tutorials Dart*. Available at: https://dart.dev/tutorials (Accessed: February 15, 2023).
- *Tutorials Flutter*. Available at: https://docs.flutter.dev/reference/tutorials (Accessed: March 2, 2023).
- Creately (2021) *Folder structure diagram tool to organize folders*, *Creately*. Available at: https://creately.com/usage/folder-structure-diagram-tool/ (Accessed: April 3, 2023).
- *Project PPT - report of doctor appointment system introduction of the project doctor appointment Studocu*. Available at: https://www.studocu.com/in/document/gopal-narayan-singh-university/management/project-ppt/25056050 (Accessed: January 20, 2023).
- *What is Agile Methodology in project management?* (no date) *Versatile & Robust Project Management Software*. Available at: https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/ (Accessed: April 6, 2023).

- *Biessek, A. (2019) Flutter for beginners: An introductory guide to building cross-platform mobile applications with flutter and Dart 2. Birmingham, UK: Packt Publishing.*
- *Sinha, S. (2017) Beginning laravel a beginner's guide to application development with Laravel 5.3. Berkeley, CA: Apress.*
- *Tutorials (no date) Flutter. Available at: https://docs.flutter.dev/reference/tutorials (Accessed: April 6, 2023).*