DENTCARE

Dental Clinic Management System

Mini Project Report

Submitted by

Riya Robin

Reg. No.: AJC19MCA-I047

In Partial fulfillment for the Award of the Degree of

INTEGRATED MASTER OF COMPUTER APPLICATIONS

(INMCA)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with ‘A’ grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2023-2024

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, “**DENTCARE DENTAL CLINIC”** is the bona fide work of **RIYA ROBIN (Regno: AJC19MCA-I047)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2023-24.

Ms. Gloriya Mathew

Internal Guide

Ms. Meera Rose Mathew

Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose

Head of the Department

DECLARATION

I hereby declare that the project report **“DENTCARE”** dental clinic system is a bona fide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Integrated Master of Computer Applications (INMCA) from APJ Abdul Kalam Technological University, during the academic year 2023-2024.

Date: **RIYA ROBIN**

KANJIRAPPALLY **Reg: AJC19MCA-I047**

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Ms. Gloriya Mathew** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

RIYA ROBIN

ABSTRACT

.

The goal of this project is to design and develop a website dedicated to manage the database of the dental clinic which they currently perform in a paper-based system. Other than a traditional system we are planning to enhance the system with some additional features as well. The "DENTCARE DENTAL CLINIC" dental clinic management system has been developed to override the problems prevailing in the practicing manual system. The objective is to allow the user to store the patient details and consults about their dental related problems. The users will spend less time on paper work through this automated system. This project is mainly divided into four modules: Admin, Patient, Doctors, Staff. The system includes a range of functionalities, including patient management, appointment scheduling, electronic health records (EHR), billing and invoicing. Moreover, this system is designed for the particular need of the management to carry out operations in a smooth and effective manner. The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this all it proves it is user-friendly. The System, as described above, can lead to error free, secure, reliable and fast management system.

CONTENT

List of Abbreviation

IDE - Integrated Development Environment

HTML - Hyper Text Markup Language

CSS - Cascading Style Sheet

SQL - Structured Query Language

PHP – PHP: - Hypertext Preprocessor

JS – JavaScript

VS - Visual Studio

BDD – Behavior Driven Development

RDBMS – Relational Database Management System

UML - Unified Modeling Language

EHR - Electronic Health Records

GPL – General Public License

BDS - Bachelor of Dental Surgery

MDS - Master of Dental Surgery

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

To overcome the issues, present in the practicing manual system, the "DENTCARE" dental clinic management system has been created. The difficulties this current system faces are supported by this software in an effort to eliminate and, in some cases, reduce them. Moreover, this system is created to address a specific need of the business to conduct operations in a smooth and efficient way.

The application is kept as simple as possible to reduce data entry errors. It sends an error message as well when entering invalid data. No formal education is required for to use this system by the user. This alone demonstrates how user-friendly it is. DentCare appointment system can result in error-free, secure, dependable, and quick administration system. It can help the user focus on their other activities instead of focus on maintaining the records. Consequently, it will aid organizations in making better use of resources.

1.2 PROJECT SPECIFICATION

The purpose of the system is to automate the existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with. The aim is to automate its existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same.

This is a website in which user can make appointments as per their need. The system consists of 4 modules. They are:

Admin Dashboard:

- This module provides administrators with a centralized dashboard to manage patient accounts, doctor accounts and staff accounts.

- Admins can add doctors and staff and also view the staff details.

Patient Dashboard:

- This module allows patients to create accounts, log in securely, and manage their profile information.
- They can make their online appointments, manage account, view medical history, make payments and bill generation.

Doctor Dashboard:

- This module provides doctors with the functionality to set their schedules for the next days.
- They can manage their profile and view if payment done or not.
- They can add the new dental procedure, dental fee and can view the patients.

Staff Dashboard:

- This module provides staffs the ability to view the patient details and manage their appointments.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

The "DENTCARE " dental clinic management system has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner. The purpose of the system is to automate the existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

2.2 EXISTING SYSTEM

Traditionally dental appointments are taken through phone calls really it is wastage of time, The proposed system provide online booking, And the system helps to avoid the documentation of paper files and patient’s queue in the clinic.

2.2.1 NATURAL SYSTEM STUDIED

Travelling to the clinic to make appointments can be tiring and time-consuming. It may take hours to make appointments when there is a large queue of patients. However, with the advancements in technology, the world is changing rapidly, and businesses are adapting to the new technological aspects every day. As humans, we prefer to do things without losing our comfort, and this is where the online aspect of business comes into play. Online appointment booking has numerous benefits.

2.2.2 DESIGNED SYSTEM STUDIED

To overcome the disadvantages of booking appointments in offline mode, this system allows the users to make appointments based on their availability from their own comfort zone itself in online mode. In this project the user can select the doctor based on their dental concern. The system also allows patients to pay appointment fee online. Once an appointment is booked, a confirmation report with selected date and time is generated and sent to the patient.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Lack of security of data.
- Time Consuming.
- More manpower.
- Consumes large volume of work
- Long wait times for appointments
- Dependence on clinic hours and availability

2.4 PROPOSED SYSTEM

The aim of the proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work. The user interface is designed to be user-friendly, allowing for easy navigation and hassle-free transactions.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- Ensure Data Accuracy
- Better Service
- Greater Efficiency
- User Friendly
- Minimum Time Required
- Convenience

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

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. This examination is a basic organize in figuring out if a project will accomplish the objectives of the association in connection to the assets, work, and time put into it. It helps the designer in assessing the project's conceivable focal points and conceivable outcomes for long run. To discover in case a proposed framework is doable and beneficial of advance examination, a possibility consider must be conducted.

3.1.1 Economical Feasibility

The economic feasibility analysis is a crucial process in determining the worth of a new project in terms of cost and time investment. It involves a thorough analysis of all factors that can influence the success of the initiative. This is a very important aspect to be considered while developing a project. We decided the technology based on minimum possible cost factor. All hardware and software cost has to be borne by the organization. Overall, we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later on running cost for system. The proposed system, DentCare Dental Clinic System, has undergone cost-benefit analysis and is found to be feasible and economical within the pre-assumed cost of this project.

3.1.2 Technical Feasibility

Technical feasibility refers to the assessment process that determines whether it is possible to develop and implement a product or service using available technology and resources. The technical feasibility analysis evaluates the proposed plan's effectiveness by examining its tools, materials, labor, logistics, and technology. It is a critical step in identifying and addressing potential project issues before starting the work. The DentCare Dental Clinic System is user-friendly and requires minimal training as it is self-explanatory. The application is easy to use, even for novice users. The system is readily available, making it economical for customers, with the only time lost being the surfing time. The system has interactive features that empower users to navigate through the environment, making them the masters of the system.

3.1.3 Behavioral Feasibility

No doubt the proposed system is fully GUI based that is very user friendly and all inputs to be taken all self-explanatory even to a layman. Besides, a proper training has been conducted to let know the essence of the system to the users so that they feel comfortable with new system. As far our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing. Additionally, all behavioral factors were taken into account during the feasibility study to ensure that the project is behaviorally feasible. Overall, the proposed system is expected to achieve its objectives with a high level of success.

3.1.4 Feasibility Study Questionnaire

1. **Project Overview?**

   The "DENTCARE " dental clinic system has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by the existing system. The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. It will serve as a centralized hub to store, access, and manage patient records, appointments, billing, and other essential clinic activities. It offers online invoicing and integrates with payment gateways for secure and convenient payment processing. Thus, by this all it proves it is user-friendly. Dental Clinic Appointment System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resource.

2. **To what extend the system is proposed for?**

   The proposed Online Dental Clinic System is designed to offer a comprehensive and efficient solution for operating a virtual dental clinic. It caters specifically to the needs of dental practices that provide dental services online through web-based platforms. The system aims to streamline clinic operations by facilitating online appointment scheduling, providing secure telemedicine capabilities for remote consultations, and managing patient records, treatment plans, and billing information in a centralized and secure manner. It enables online invoicing and integrates with payment gateways for seamless and secure payment processing.

3. **Specify the Viewers/Public which is to be involved in the System?**

   Dental patients, Doctors, Staff

4. **List the Modules included in your System?**

   Admin, Patient, Doctor, Staff

5. **Identify the users in your project?**

   Doctor, Staff, Patients

6. **Who owns the system?**

   Dr. Tony Michael Edakkara

7. **System is related to which firm/industry/organization?**

   Dental clinic management organization

8. **Details of person that you have contacted for data collection?**

   Dr. Tony Michael Edakkara

9. **Questionnaire to collect details about the project? (min 10 questions, include descriptive answers, attach additional docs (e.g., Bill receipts, certificate models), if any?)**

   1. **How you store the patient's details?**

   Clinic records are stored in paper files. But patient details are lately stored into computer

   2. **What dental services are provided through a dental clinic?**

      ▪ Dental consultations

      ▪ Teeth Cleanings

      ▪ Dental Fillings

      ▪ Dentures

      ▪ Oral health advice and education

      ▪ Orthodontic treatments

   3. **What are the Surgery tests?**

      ▪ Cone Beam Computed Tomography (CBCT)

      ▪ Dental X-rays

      ▪ Periodontal surgery

      ▪ Root Canal

      ▪ Dental Implants

      ▪ Wisdom Tooth Extraction

      ▪ Reconstructive dental surgery

      ▪ Cosmetic dental surgery

   4. **What about the leave management at hospital?**

   Manually fill the leave forum and submitting it to the HOD/Manager

   5. **How the payment is collected from the user?**

By cash or internet payment facility

6. **How often should a patient have a dental exam and cleaning?**

   While most people need a dental checkup every 6 months, everyone is different, so it's good to ask what's best in your specific case.

7. **What are the qualifications and experience of the dentists and dental staff working at the clinic?**

   Dentists hold either a Bachelor of Dental Surgery (BDS) or Master of Dental Surgery (MDS) degree. Dental specialists have completed additional years of education and training in specific areas of dentistry. Dental Assistants may have formal training from dental assistant programs or on-the-job training.

8. **How do you decide on the pricing for the treatments?**

   The pricing for the treatments is decided by the Dental Clinics Organization.

9. **What marketing strategies do you use to promote dental clinic?**

We use local advertising, word-of-mouth, and occasional events to reach potential customers.

10. **What feedback do you receive from customers, and how do you use it to improve your services?**

Feedbacks are collected from patients requiring more than one sitting through feedback forms. Those patients will provide genuine feedback rather than those with single sitting treatments. Customer feedback helps us understand their preferences and make adjustments to our service offerings

3.1 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	- Intel core i3
RAM	- 4 GB
Hard disk	- 1 TB

3.2.2 Software Specification

Front End	-	HTML, CSS
Back End	-	PHP, MYSQL

Database	-	MYSQL
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, AJAX, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 Eg.PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. The 1995 PHP reference execution, created by Rasmus Ledorf, was created by the PHP bunch. Instep than standing for "individual domestic page" because it did initially, the PHP truncation presently stands for "PHP: Hypertext Preprocessor". PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP. PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge. The PHP processor on the internet server may comprehend the PHP code and deliver the ultimate web page. It isn't fundamental to call an outside record to handle the information since PHP code may be specifically put into the crude HTML substance. A command has moreover been included to PHP.

3.3.2Eg. MySQL

MySQL may be a broadly used open-source social database administration framework (RDBMS) that's claimed and upheld by Prophet Organization. It is profoundly prevalent among web designers and is known for its unwavering quality, versatility, and ease of utilize.

• MySQL is a Database Management System

A database is an organized collection of data that can vary from a simple list of items to a sophisticated system that stores vast amounts of information. A database management system (DBMS) such as MySQL Server is necessary to manage, retrieve, and manipulate data stored in a computer database

• MySQL software is Open Source

Within the setting of program, "Open Source" refers to program that's free to utilize and alter by anybody. MySQL is an Open-Source database administration framework, which implies that anybody can download and utilize the computer program without any taken a toll.

• **MySQL Database Server is very fast, reliable, scalable, and easy to use**

MySQL Server may be a solid and productive database administration framework appropriate for utilize on desktop computers, tablets or other gadgets without the required for broad support. It is profoundly consistent with other applications and web servers and permits customization of settings to optimize execution based on accessible assets such as Slam, CPU control and I/O capacity. MySQL Server bolsters both client/server and inserted frameworks

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Any designed system or product's development starts with the design phase. An efficient system depends on well-executed design, which is a creative process. The definition of "design" is "the act of utilizing diverse ideas and methodologies for the specifying a process or system with enough depth to allow for its physical realization”. It entails utilizing a variety of approaches and concepts to define a process or system in enough depth to allow for its actual execution. Regardless of the development model chosen, the design phase is critical in software engineering. It strives to produce the architectural detail needed to build a system or product and serves as the technical backbone of the software engineering process.

This program has through a thorough design phase that optimizes every aspect of effectiveness, performance, and accuracy. A user-oriented document is converted into a document for programmers or database employees during the design process.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML stands for Unified Modeling Language. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general-purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. UML has a direct relation with object-oriented analysis and design. UML may be utilized for non-software frameworks, such as fabricating forms, indeed though it is generally utilized to speak to program frameworks.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

4.2.1 USE CASE DIAGRAM

Use case diagrams are graphical representations used to capture the functional requirements of a system. The Unified Modelling Language (UML), a standard language for modelling actual things and systems, is frequently used to construct use case diagrams.

The system boundaries, actors, use cases, and their connections together make up a use case diagram. The system boundary establishes the system's boundaries in reference to its surroundings. Actors are often defined depending on the roles they play and reflect the people or systems that interact with the system. The precise activities or behaviors that actors carry out within or close to the system are known as use cases. Finally, the graphic shows the connections between actors and use cases as well as the use cases themselves.

Use Case diagrams are used to analyze the system’s high-level requirements. These requirements are expressed through different use cases. We notice three main components of this UML diagram:

- Functional requirements – represented as use cases; a verb describing an action
- Actors – they interact with the system; an actor can be a human being, an organization or an internal or external application
- Relationships between actors and use cases – represented using straight arrows.

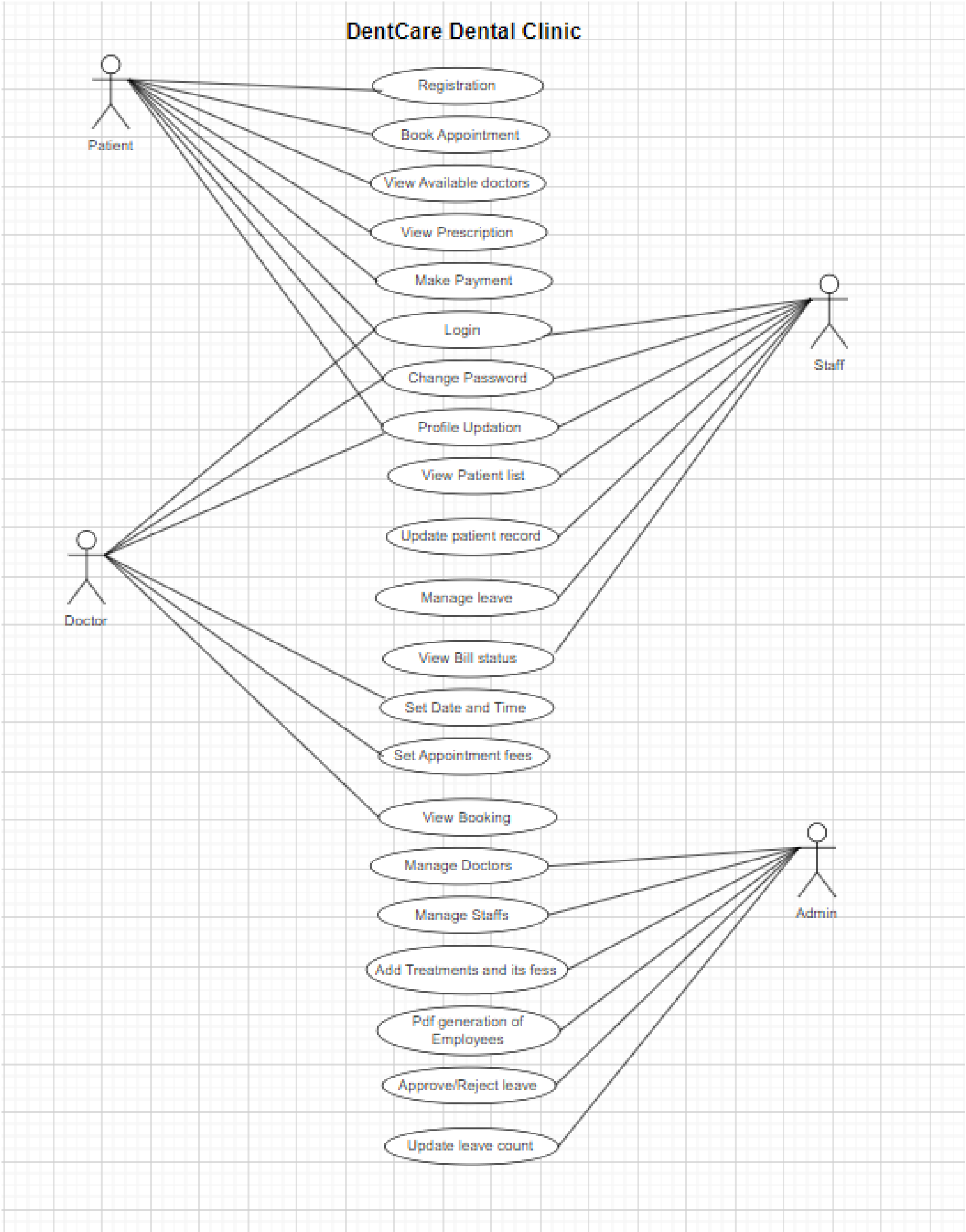


Figure 1. Use Case Diagram of DentCare

4.2.2 SEQUENCE DIAGRAM

Sequence diagrams are probably the most important UML diagrams among not only the computer science community but also as design-level models for business application development. Lately, they have become popular in depicting business processes, because of their visually self-explanatory nature. As the name suggests, sequence diagrams describe the sequence of messages and interactions that happen between actors and objects. Actors or objects can be active only when needed or when another object wants to communicate with them. All communication is represented in a chronological manner. To get a better idea, check the example of a UML sequence diagram below. As the name suggests, structural diagrams are used to depict the structure of a system. More specifically, it is used in software development to represent the architecture of the system and how the different components are interconnected (not how they behave or communicate, simply where they stand).

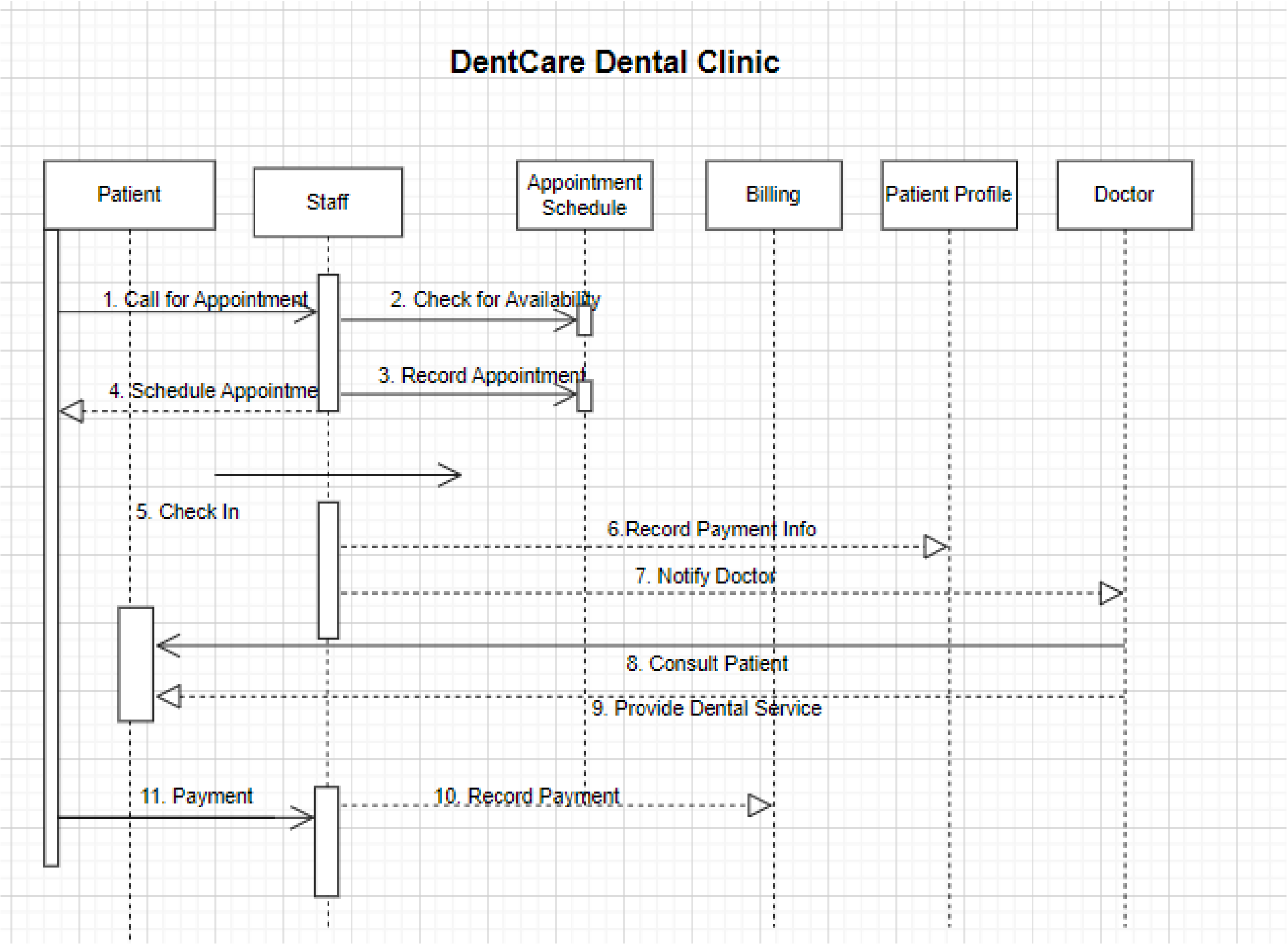


Figure 2. Sequence Diagram of DentCare

4.2.3 STATE CHART DIAGRAM

A state diagram is a visual representation, often created using the Unified Modeling Language (UML), that shows the different states that an object can exist in and how it can transition between those states. The State Chart Diagram is a behavioral diagram in UML that describes the behavior of a system or object over time. State machine UML diagrams, also referred to as State chart diagrams, are used to describe the different states of a component within a system. It takes the name state machine because the diagram is essentially a machine that describes the several states of an object and how it changes based on internal and external events. A very simple state machine diagram would be that of a chess game. A typical chess game consists of moves made by White and moves made by Black. White gets to have the first move and thus initiates the game. The conclusion of the game can occur regardless of whether it is the White's turn or the Black's. The game can end with a checkmate, resignation or in a draw (different states of the machine).

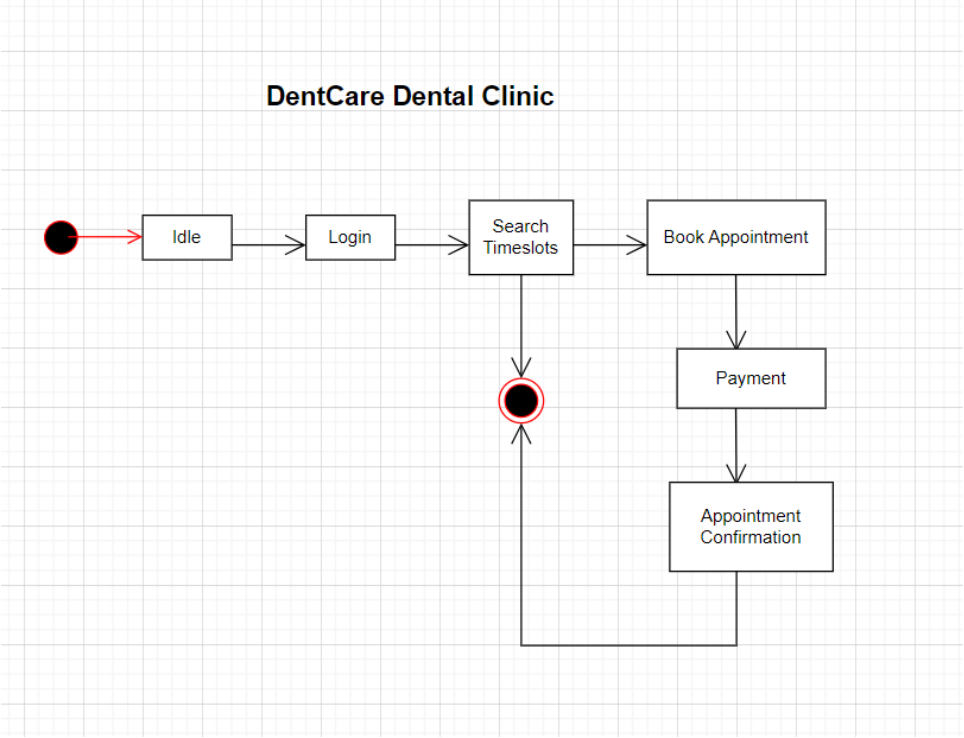


Figure 3. State Chart Diagram of DentCare

4.2.4 ACTIVITY DIAGRAM

An activity diagram is a visual representation of a workflow that shows how one activity leads to another. An activity is referred to as a system operation, and one operation leads to another in the control flow. A flow can be parallel, concurrent, or branched, and activity diagrams use various functions such as branching, joining, etc., to manage all types of flow control. They describe the objects used, consumed or produced by an activity and the relationship between the different activities. Activity diagrams are a type of behavior diagram that shows the behavior of a system. They show the flow of control from the start point to the end point and show the different decision paths that exist during the execution of the activity

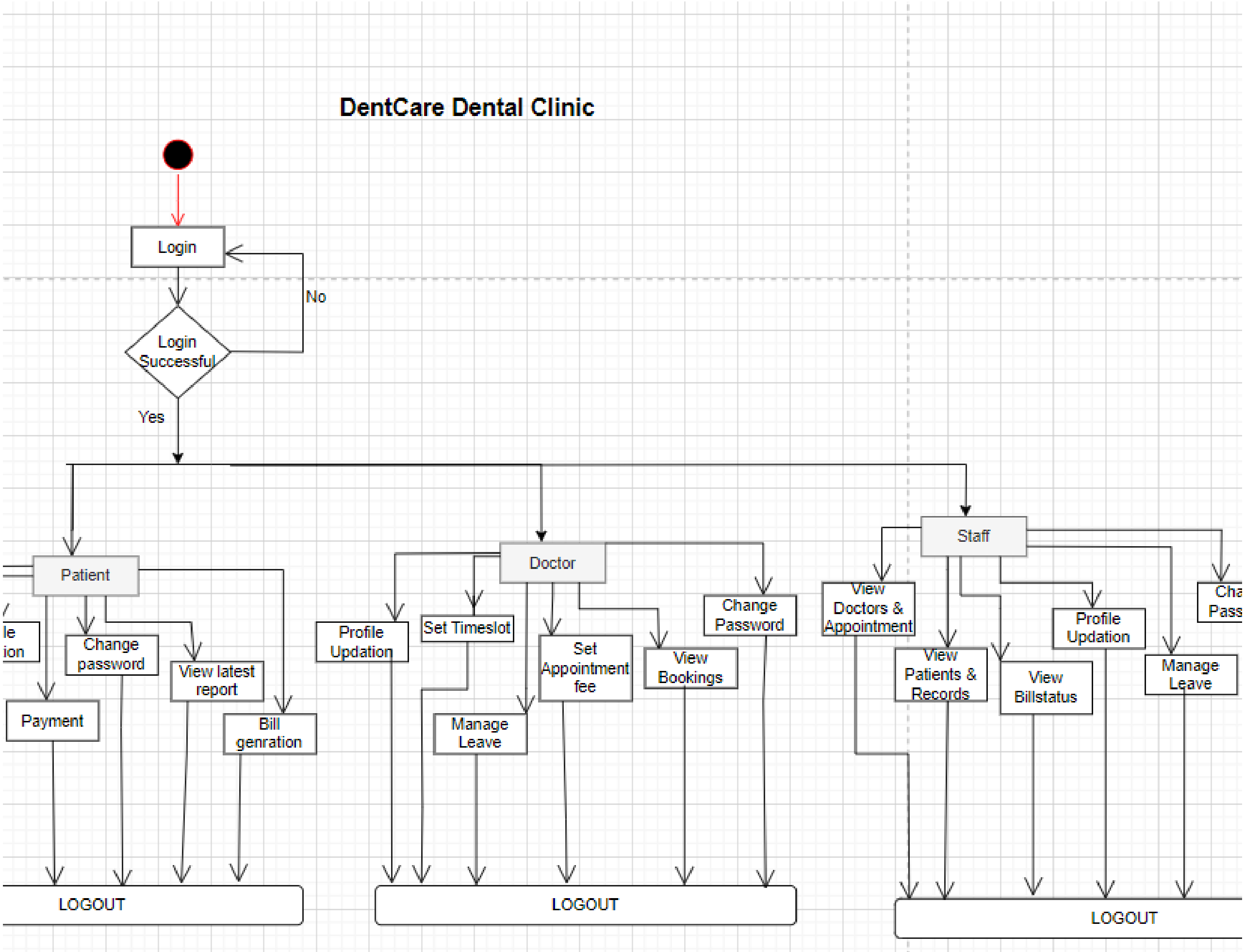


Figure 4. Activity Diagram of DentCare

4.2.5 CLASS DIAGRAM

Class diagrams are the main building block of any object-oriented solution. It shows the classes in a system, attributes, and operations of each class and the relationship between each class. In most modeling tools, a class has three parts - Name at the top, attributes in the middle and operations or methods at the bottom. In a large system with many related classes, classes are grouped together to create class diagrams. Different relationships between classes are shown by different types of arrows. The class diagram is a fundamental component of object-oriented modeling and serves as the primary means of conceptual modeling for the structure of an application. Additionally, class diagrams can be used for detailed modeling that can be translated into programming code. Class diagrams are a crucial component of UML used to represent classes, objects, interfaces, and their relationships and attributes in a system.

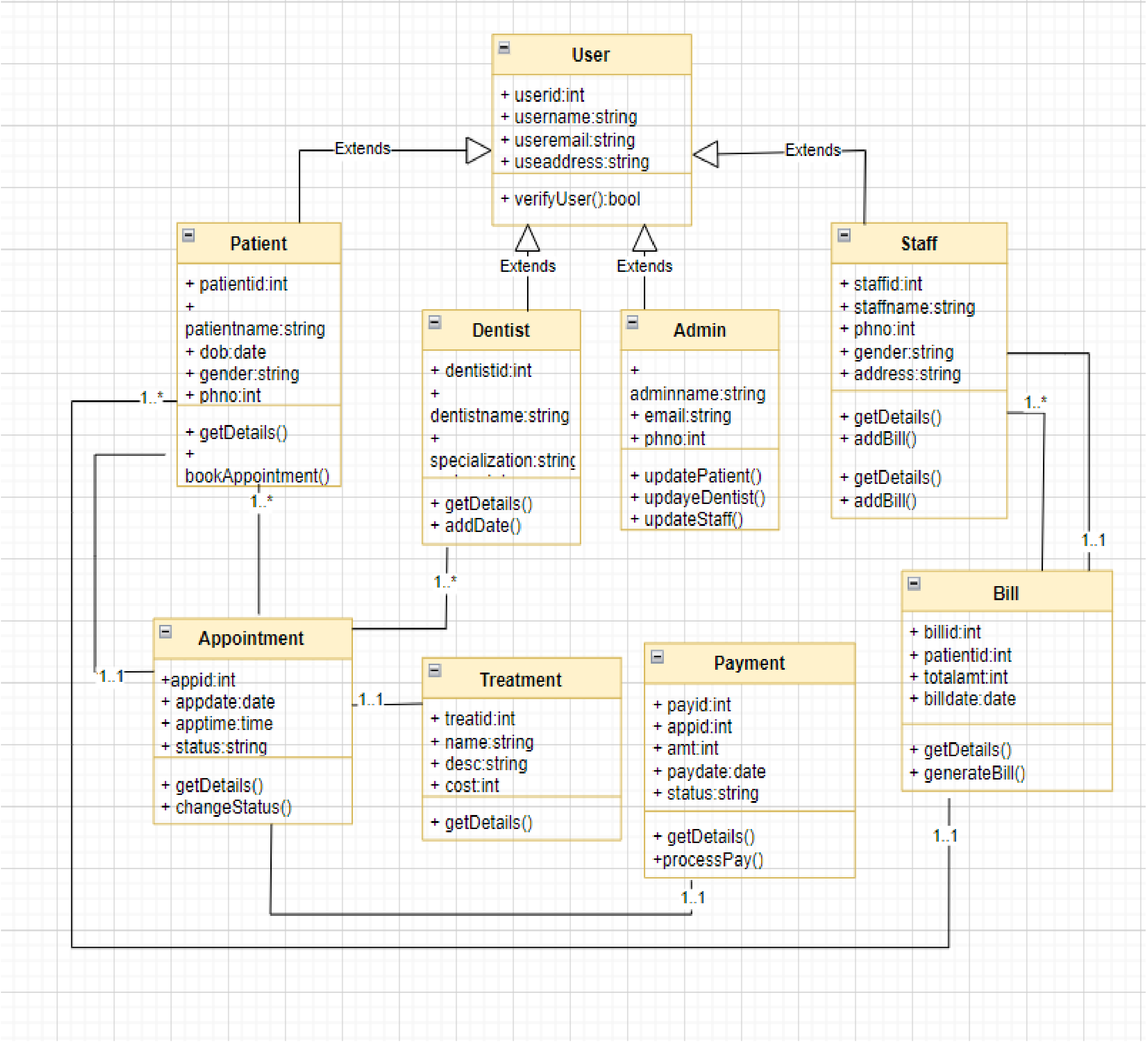


Figure 5. Class Diagram of DentCare

4.2.6 OBJECT DIAGRAM

Class diagrams and object diagrams are closely related in object-oriented modeling. Object diagrams are instances of class diagrams, which represent a snapshot of the system at a given moment in time. Both types of diagrams use the same concepts and notation to represent the structure of a system. While class diagrams are used to model the structure of the system, including its classes, attributes, and methods, object diagrams represent a group of objects and their connections at a specific point in time. An object diagram is a type of structural diagram in UML that shows instances of classes and their relationships.

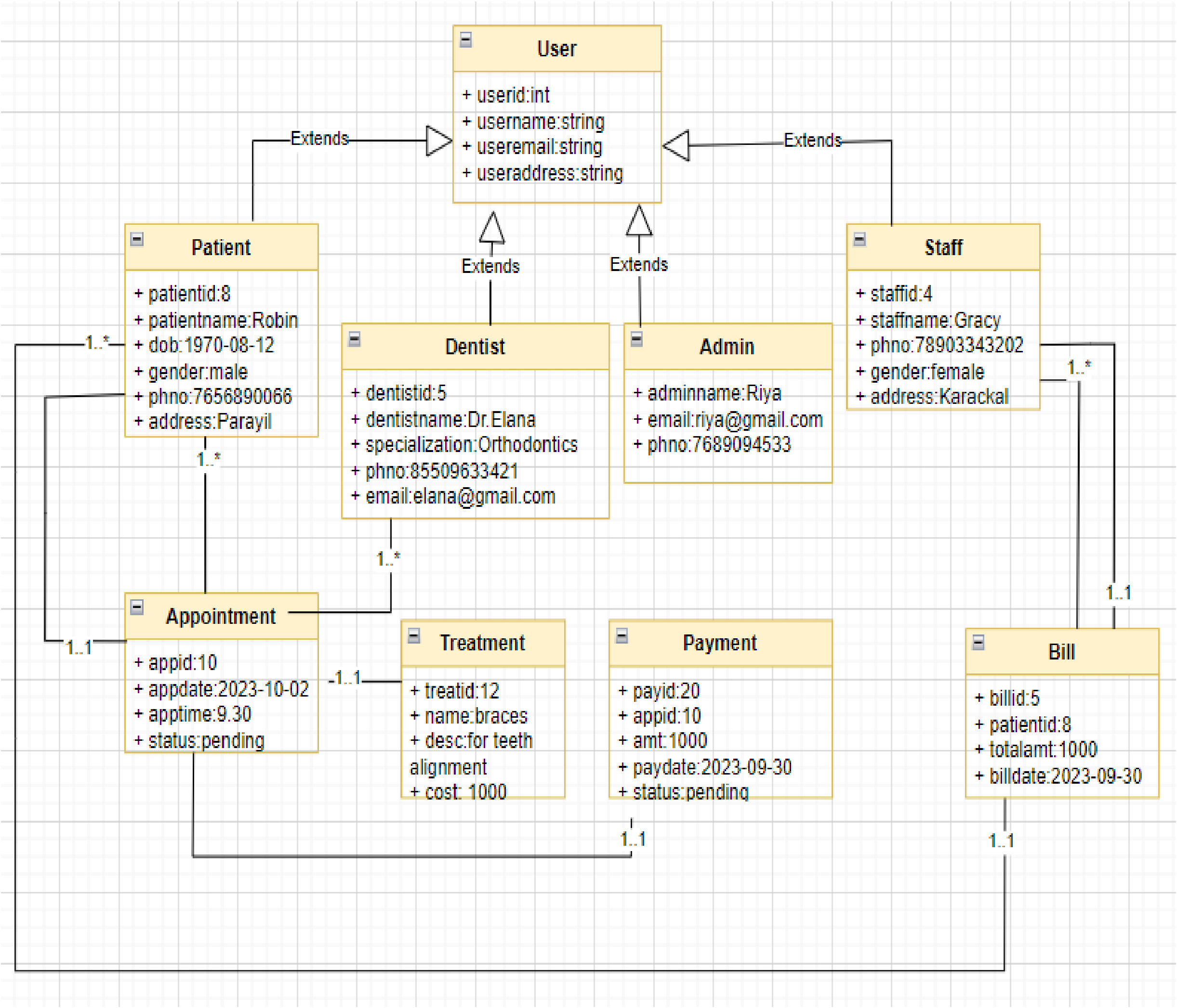


Figure 6. Object Diagram of DentCare

4.2.7 COMPONENT DIAGRAM

A component diagram in UML illustrates how various components are interconnected to create larger components or software systems. It is an effective tool for representing the structure of complex systems with multiple components. By using component diagrams, developers can easily visualize the internal structure of a software system and understand how different components work together to accomplish a specific task. Component diagrams are useful for modeling the architecture of a software system, and can help identify potential issues and improvements in the design. They can also be used to communicate the structure and behavior of a system to stakeholders, such as developers and project managers.

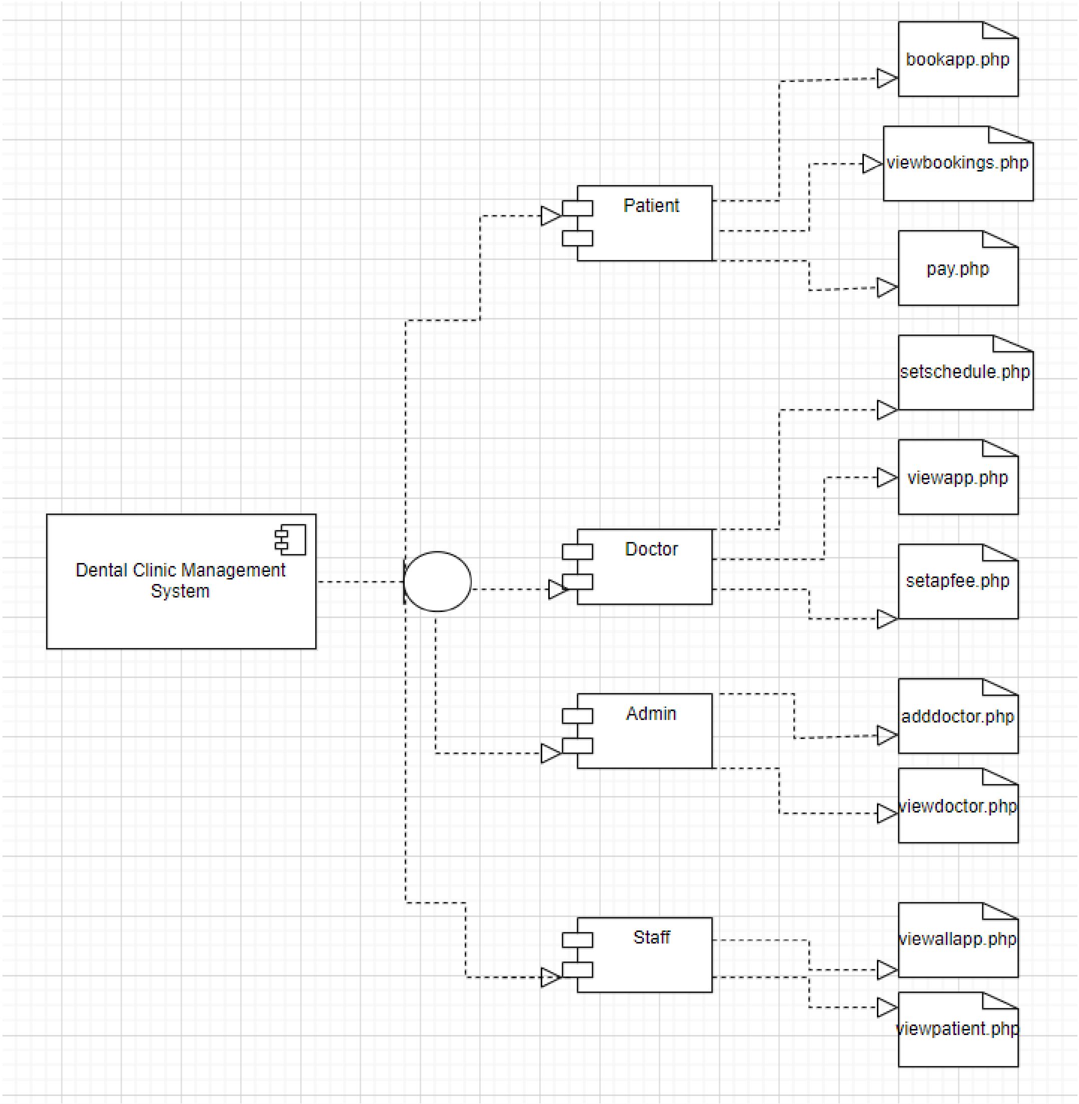


Figure 7. Component Diagram of DentCare

4.2.8 DEPLOYMENT DIAGRAM

Deployment diagrams are used to visualize the relation between software and hardware. To be more specific, with deployment diagrams we can construct a physical model of how software components (artifacts) are deployed on hardware components, known as nodes. It provides a static view of a system's deployment and involves nodes and their relationships. The deployment diagram maps the software architecture to the physical system architecture, showing how the software will be executed on nodes. Communication paths are used to illustrate the relationships between the nodes. Unlike other UML diagram types, which focus on the logical components of a system, the deployment diagram emphasizes the hardware topology.

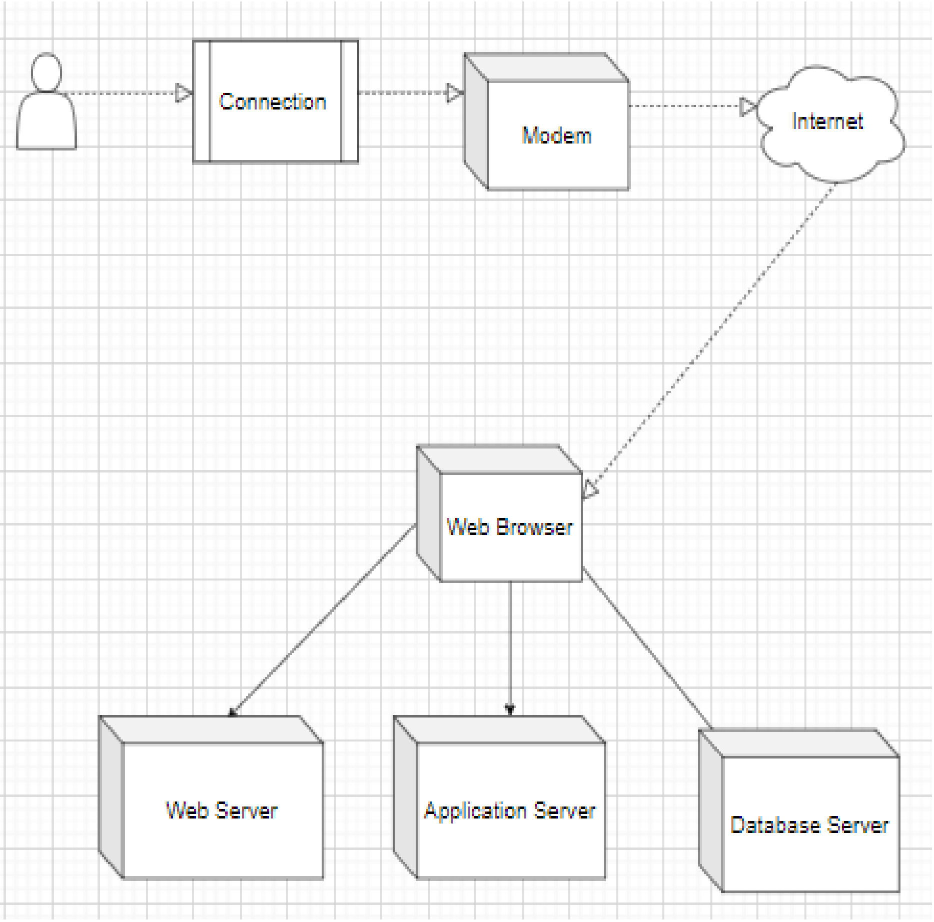


Figure 8. Deployment Diagram of DentCare

4.2.9 COLLABORATION DIAGRAM

A collaboration diagram is a diagram that is used to represent the relationships between objects in a system. It is similar to a sequence diagram in that it represents the same information, but it does so in a different way. Instead of showing the flow of messages between objects, it depicts the structure of the objects in the system. This is because collaboration diagrams are based on object-oriented programming, where objects have various attributes and are connected to each other. Thus, collaboration diagrams are a visual representation of the object architecture in a system.

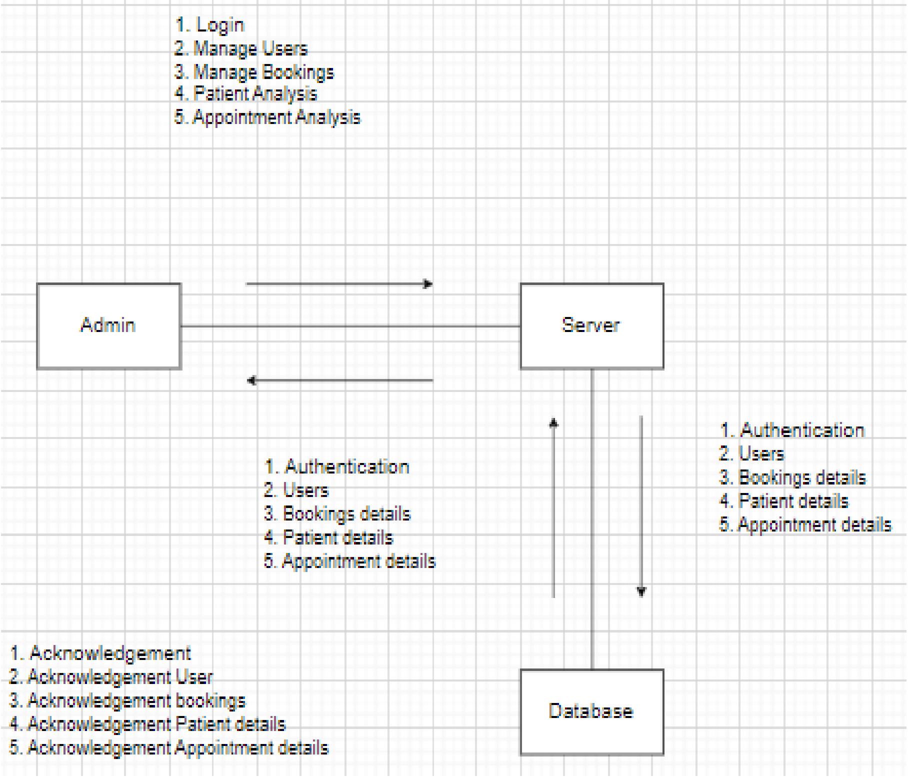
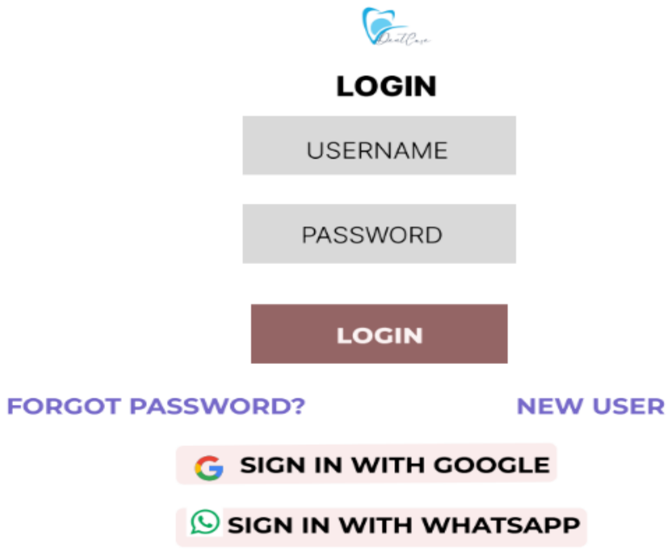


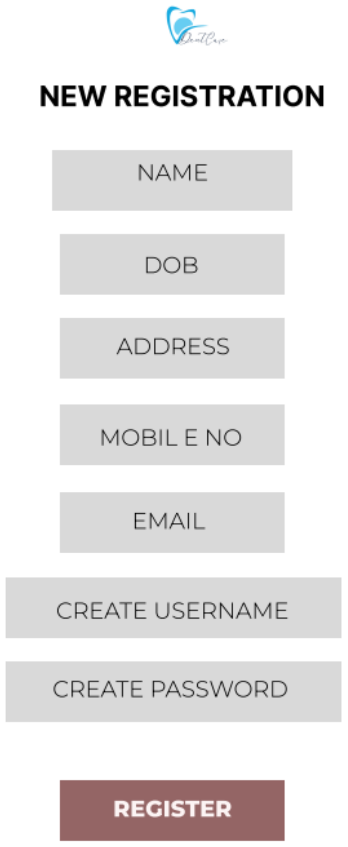
Figure 9. Use Case Diagram of DentCare

4.3 USER INTERFACE DESIGN USING FIGMA

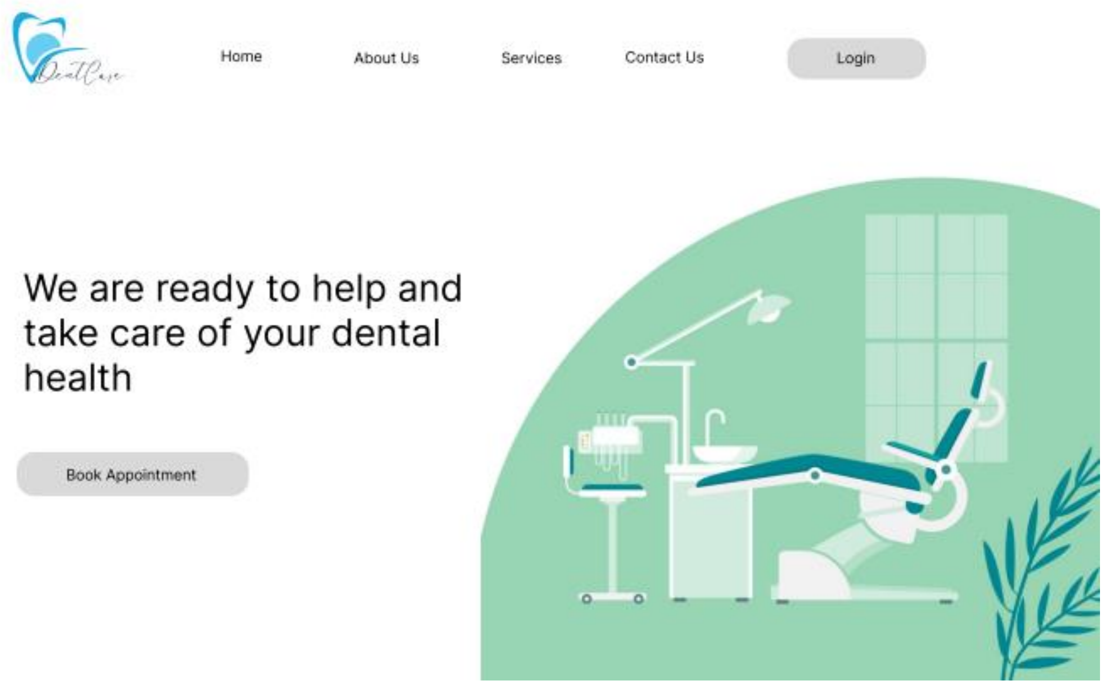
Form Name: Login page



Form Name: Registration page

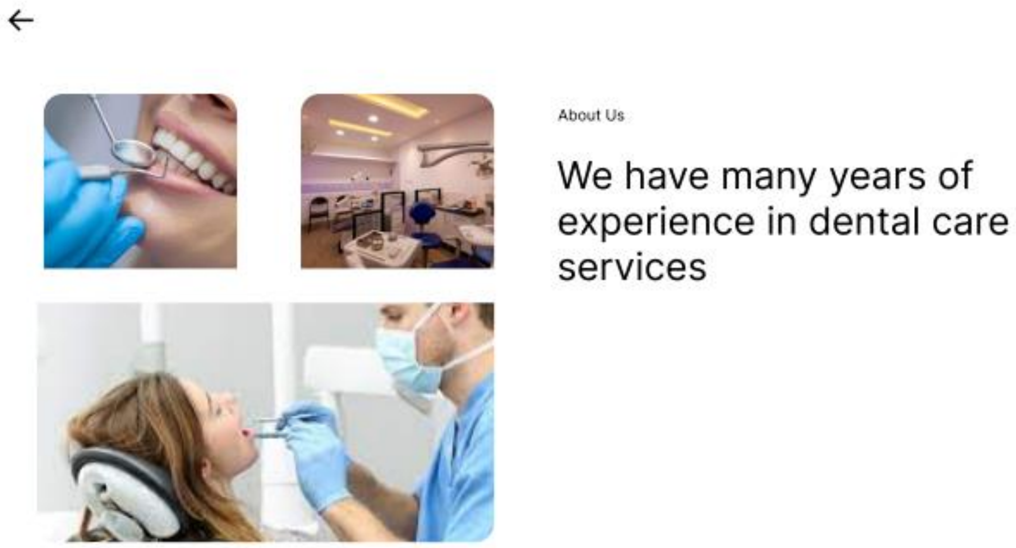


Form Name: Home page



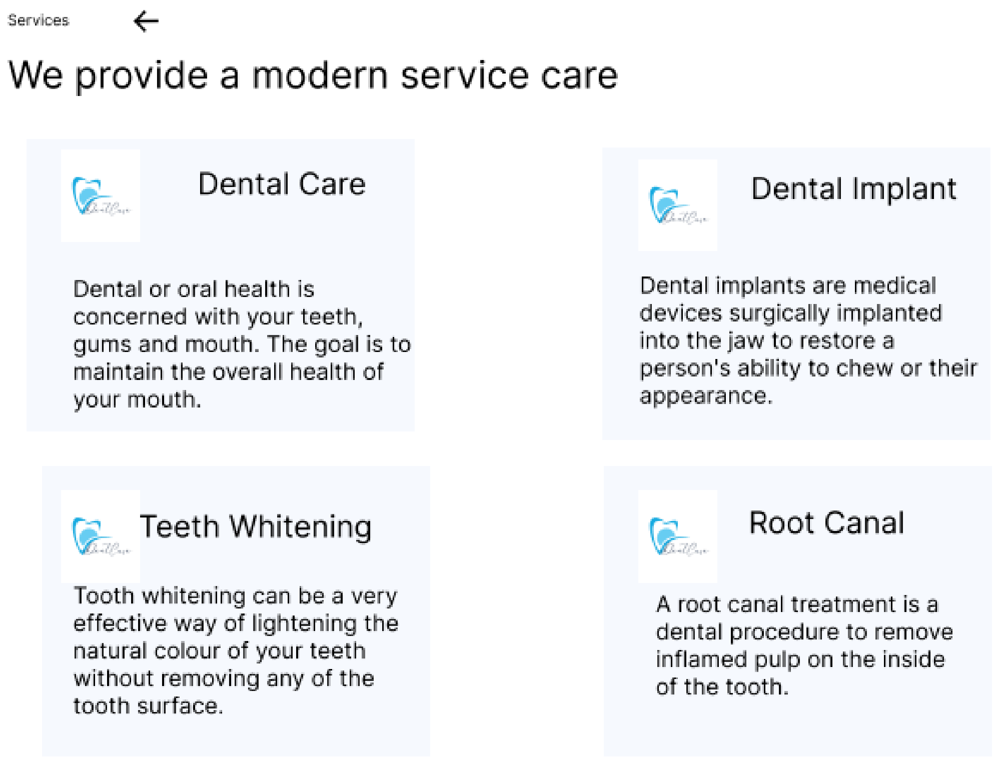
We are ready to help and take care of your dental health

Form Name: About page



We have many years of experience in dental care services

Form Name: Services page



Services

We provide a modern service care

Dental Care

Dental or oral health is concerned with your teeth, gums and mouth. The goal is to maintain the overall health of your mouth.

Dental Implant

Dental implants are medical devices surgically implanted into the jaw to restore a person's ability to chew or their appearance.

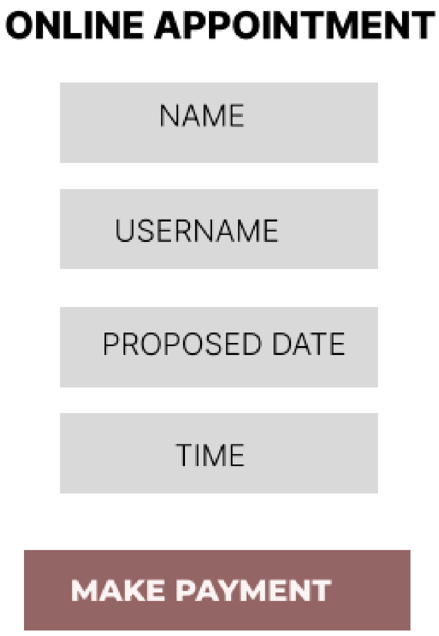
Teeth Whitening

Tooth whitening can be a very effective way of lightening the natural colour of your teeth without removing any of the tooth surface.

Root Canal

A root canal treatment is a dental procedure to remove inflamed pulp on the inside of the tooth.

Form Name: Book appointment page



ONLINE APPOINTMENT

NAME

USERNAME

PROPOSED DATE

TIME

4.4 DATABASE DESIGN

A database is an organized collection of information that's organized to enable easy accessibility, administration, and overhauls. The security of information could be an essential objective of any database. The database design process comprises of two stages. In the first stage, user requirements are gathered to create a database that meets those requirements as clearly as possible. This is known as information-level design and is carried out independently of any DBMS. In the second stage, the design is converted from an information-level design to a specific DBMS design that will be used to construct the system. This stage is known as physical-level design, where the characteristics of the specific DBMS are considered. Alongside system design, there is also database design, which aims to achieve two main goals: data integrity and data independence

4.4.1 Relational Database Management System (RDBMS)

RDBMS stands for Relational Database Management System. All modern database management systems like SQL, MS SQL Server, IBM DB2, ORACLE, My-SQL, and Microsoft Access are based on RDBMS. It is called Relational Database Management System (RDBMS) because it is based on the relational model introduced by E.F. Codd. Data is represented in terms of tuples (rows) in RDBMS.A relational database is the most commonly used database. It contains several tables, and each table has its primary key. Due to a collection of an organized set of tables, data can be accessed easily in RDBMS.

4.4.2 Normalization

Normalization is the process to eliminate data redundancy and enhance data integrity in the table. Normalization also helps to organize the data in the database. It is a multi-step process that sets the data into tabular form and removes the duplicated data from the relational tables. Normalization organizes the columns and tables of a database to ensure that database integrity constraints properly execute their dependencies. It is a systematic technique of decomposing tables to eliminate data redundancy (repetition) and undesirable characteristics like Insertion, Update, and Deletion anomalies

4.4.3 Sanitization

Data sanitization is the process of removing any illegal characters or values from data. In web applications, sanitizing user input is a common task to prevent security vulnerabilities. PHP provides a built-in filter extension that can be used to sanitize and validate various types of external input such as email addresses, URLs, IP addresses, and more. These filters are designed to make data sanitization easier and faster. For example, the PHP filter extension has a function that can remove all characters except letters, digits, and certain special characters (! #$%&'*+- =? _` {|} ~@. []), as specified by a flag. Web applications often receive external input from various sources, including user input from forms, cookies, web services data, server variables, and database query results. It is important to sanitize all external input to ensure that it is safe and does not contain any malicious code or value

4.4.4 Indexing

Indexing is a way to optimize the performance of a database by minimizing the number of disk accesses required when a query is processed. It is a data structure technique which is used to quickly locate and access the data in a database. Indexes are created using a few database columns. The first column is the Search key that contains a copy of the primary key or candidate key of the table. These values are stored in sorted order so that the corresponding data can be accessed quickly. The data may or may not be stored in sorted order. The second column is the Data Reference or Pointer which contains a set of pointers holding the address of the disk block where that particular key value can be found.

4.5 TABLE DESIGN

1.tbl_login

Primary key: LoginId

Foreign key: LoginId references table tbl_patient, tbl_docotor, tbl_staff

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	LoginId	int(10)	Primary key	Login id
2	Email	varchar(20)	NULL	Email
3	Password	varchar(20)	NULL	Password
4	Role	varchar(20)	NOT NULL	Role of users
5	Status	int(10)	NOT NULL	Status - Active/Inactive

2.tbl_patient

Primary key: Id

Foreign key: LoginId references table tbl_login

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Id	int(10)	Primary key	Patient id
2	LoginId	int(10)	Foreign key	Login id
3	Email	varchar(20)	NOT NULL	Email
4	Password	varchar(20)	NOT NULL	Password
5	FullName	varchar(20)	NOT NULL	Full Name
6	Age	int(10)	NOT NULL	Age
7	Address	varchar(20)	NOT NULL	Address
8	MobileNo	varchar(20)	NOT NULL	Mobile number

3.tbl_admin

Primary key: Id

Foreign key: LoginId references table tbl_login

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Id	int(10)	Primary key	Admin id
2	LoginId	int(10)	Foreign key	Login id
3	Email	varchar(20)	NOT NULL	Email
4	Password	varchar(20)	NOT NULL	Password
5	FullName	varchar(20)	NOT NULL	Full Name

4.tbl_doctor

Primary key: Id

Foreign key: LoginId references table tbl_login

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Id	int(10)	Primary key	Doctor id
2	LoginId	int(10)	Foreign key	Login id
3	Email	varchar(20)	NOT NULL	Email
4	Password	varchar(20)	NOT NULL	Password
5	FullName	varchar(20)	NOT NULL	Full Name
6	Specialization	varchar(20)	NOT NULL	Specialization
7	Address	varchar(20)	NOT NULL	Address
8	Pincode	varchar(20)	NOT NULL	Pincode
9	apfee	varchar(20)	NOT NULL	Appointment fee
10	apfeeupdatetime	varchar(20)	NOT NULL	Appointment fee edited time
11	feestatus	int(11)	NOT NULL	Fees status
12	Comments	varchar(100)	NOT NULL	Comments on leave
13	orgfee	varchar(20)	NOT NULL	Organization’s fee
14	lvbal	varchar(20)	NOT NULL	Leave Balance
15	Status	varchar(20)	NOT NULL	Status-Active/Inactive
16	Approval_status	varchar(20)	NOT NULL	Approved/Rejected
17	MobileNo	varchar(20)	NOT NULL	Mobile number

5.tbl_staff

Primary key: Id

Foreign key: LoginId references table tbl_login

No:	Field name	Datatype (Size)	Key Constraints	Description of the field

1	Id	int(10)	Primary key	Staff id
2	LoginId	int(10)	Foreign key	Login id
3	Email	varchar(20)	NOT NULL	Email
4	Password	varchar(20)	NOT NULL	Password
5	FullName	varchar(20)	NOT NULL	Full Name
6	MobileNo	varchar(20)	NOT NULL	Mobile number
7	Address	varchar(20)	NOT NULL	Address
8	Status	varchar(20)	NOT NULL	Status-Active/Inactive
9	Approval_status	varchar(20)	NOT NULL	Approved/Rejected

6.tbl_specialization

Primary key: Id

Foreign key: LoginId references table tbl_login

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Id	int(10)	Primary key	Admin id
2	LoginId	int(10)	Foreign key	Login id
3	Specialization	varchar(20)	NOT NULL	Doctor specialization
4	CreationDate	timestamp	NOT NULL	Specialization updated date

7. tbl_schedule

Primary key: Id

Foreign key: LoginId references table tbl_login

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Id	int(10)	Primary key	Doctor slot id
2	LoginId	int(10)	Foreign key	Login id
3	availdate	date	NOT NULL	Available date

4	slot1	int(10)	NOT NULL	Slot 1 time
5	slot2	int(10)	NOT NULL	Slot 2 time
6	slot3	int(10)	NOT NULL	Slot 3 time
7	slot4	int(10)	NOT NULL	Slot 4 time
8	slot5	int(10)	NOT NULL	Slot 5 time
9	slot6	int(10)	NOT NULL	Slot 6 time
10	slot7	int(10)	NOT NULL	Slot 7 time
11	available_slots	int(10)	NOT NULL	Number of Available slots
12	booked_slots	int(10)	NOT NULL	Number of Booked slots

8. tbl_booking

Primary key: Id

Foreign key: LoginId references table tbl_login, did references table tbl_doctor

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	Id	int(10)	Primary key	Booking id
2	did	int(10)	Foreign key	Doctor id
3	LoginId	int(10)	NOT NULL	Login id
4	bkdate	date	NOT NULL	Booking Date
5	bktimeslot	varchar(20)	NOT NULL	Booking time slot
6	status	int(10)	NOT NULL	Booking status
7	paymentStatus	varchar(20)	NOT NULL	Payment status
8	paymentDate	varchar(20)	NOT NULL	Payment date
9	amount	varchar(20)	NOT NULL	Payment amount
10	txnid	varchar(20)	NOT NULL	Transaction id
11	rfnid	varchar(20)	NOT NULL	Reference id

9. tbl_experience

Primary key: Id

Foreign key: LoginId references table tbl_login, did references table tbl_doctor

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	ug	varchar(20)	NOT NULL	Under graduation degree
2	ugyear	year(4)	NOT NULL	UG pass out year
3	pg	varchar(20)	NOT NULL	Post graduation
4	pgyear	year(4)	NOT NULL	PG pass out year
5	exp	varchar(20)	NOT NULL	Experience certificate
6	expduration	varchar(20)	NOT NULL	Experience duration
7	Login id	int(10)	NOT NULL	Login id

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Software testing involves executing a software program in a controlled manner to determine if it behaves as intended, often using verification and validation methods. Validation involves evaluating a product to ensure it complies with specifications, while verification can involve reviews, analyses, inspections, and walkthroughs. Static analysis examines the software's source code to identify issues, while dynamic analysis examines its behavior during runtime to gather information like execution traces, timing profiles, and test coverage details.

Testing involves a series of planned and systematic activities that start with individual modules and progress to the integration of the entire computer-based system. The objectives of testing include identifying errors and bugs in the software, ensuring that the software functions according to its specifications, and verifying that it meets performance requirements. Testing can be performed to assess correctness, implementation efficiency, and computational complexity.

A successful test is one that detects an undiscovered error, and a good test case has a high probability of uncovering such errors. Testing is crucial to achieving system testing objectives and can involve various techniques such as functional testing, performance testing, and security testing.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan. The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Before any other testing can take place, it is essential to test data flow over a module interface. If data cannot enter and exit the system properly, all other tests are irrelevant. Another crucial duty during unit testing is the selective examination of execution pathways to anticipate potential errors and ensure that error handling paths are set up to reroute or halt work when an error occurs. Finally, boundary testing is conducted to ensure that the software operates correctly at its limits.

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover, differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

The final stage of the testing process involves testing the entire software system as a whole, including all forms, code, modules, and class modules. This is commonly referred to as system testing or black box testing. The focus of black box testing is on testing the functional requirements of the software. A software engineer can use this approach to create input conditions that will fully test each program requirement. The main types of errors targeted by black box testing include incorrect or missing functions, interface errors, errors in data structure or external data access, performance errors, initialization errors, and termination errors.

5.2.4 Output Testing or User Acceptance Testing

User acceptance testing is performed to ensure that the system meets the business requirements and user needs. It is important to involve the end users during the development process to ensure that the software aligns with their needs and expectations. During user acceptance testing, the input and output screen designs are tested with different types of test data. The preparation of test data is critical to ensure comprehensive testing of the system. Any errors identified during testing are addressed and corrected, and the corrections are noted for future reference.

5.2.5 Automation Testing

Automation Testing is a software testing technique that performs using special automated testing software tools to execute a test case suite. On the contrary, Manual Testing is performed by a human sitting in front of a computer carefully executing the test steps. The automation testing software can also enter test data into the System Under Test, compare expected and actual results and generate detailed test reports. Software Test Automation demands considerable investments of money and resources. Successive development cycles will require execution of same test suite repeatedly. Using a test automation tool, it’s possible to record this test suite and re-play it as required. Once the test suite is automated, no human intervention is required. This improved ROI of Test Automation. The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing altogether.

5.2.6 Selenium Testing

Selenium is an open-source automated testing framework used to verify web applications across different browsers and platforms. Selenium allows for the creation of test scripts in various programming languages such as Java, C#, and Python.

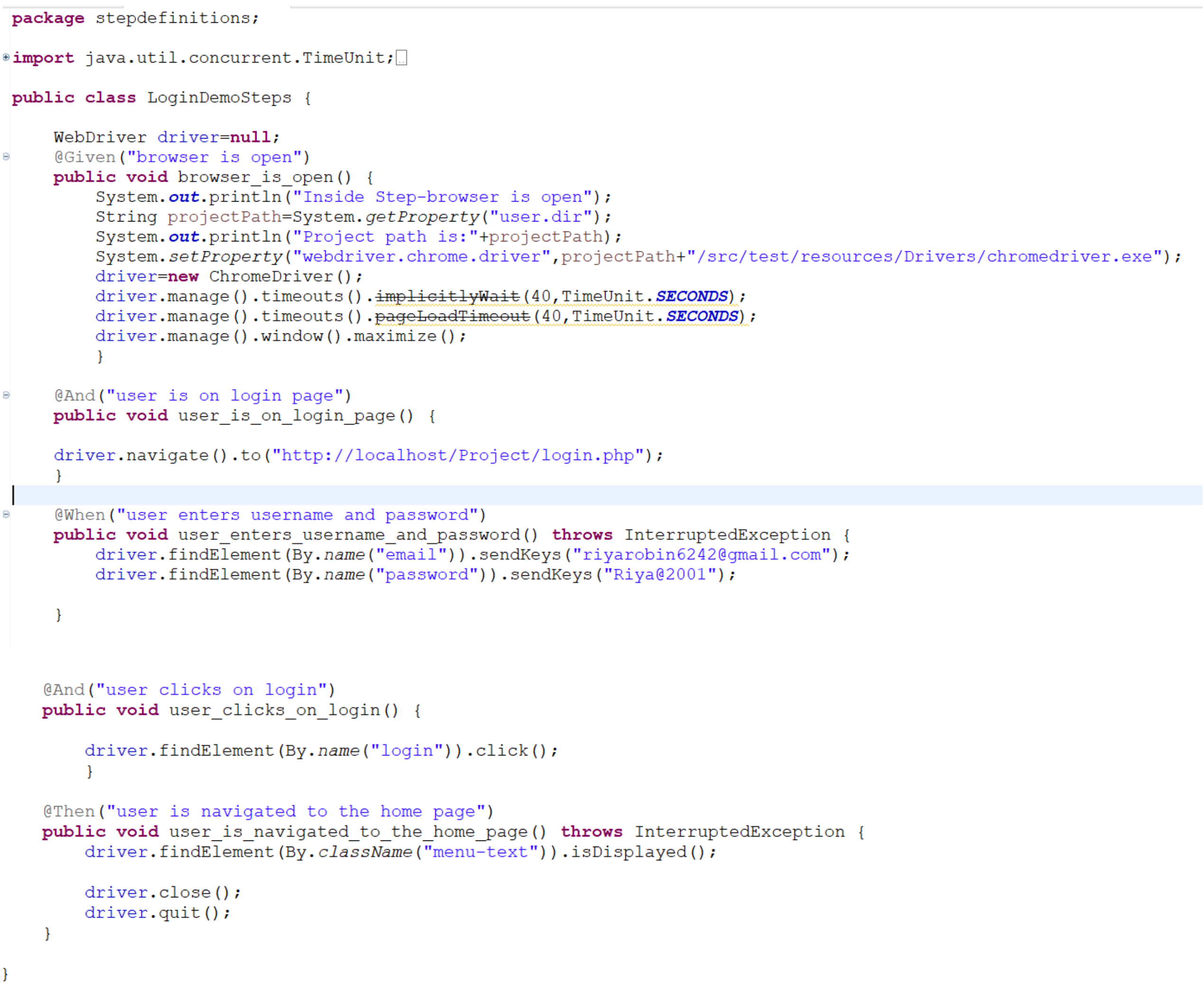
In addition to Selenium, another popular tool used for automated testing is Cucumber. Cucumber is an open-source software testing framework that supports behavior-driven development (BDD). It allows for the creation of executable specifications in a human-readable format called Gherkin. Cucumber facilitates effective communication and collaboration during the testing process. It promotes a shared understanding of the requirements and helps ensure that the developed software meets the intended business goals.

Cucumber can be integrated with Selenium to combine the benefits of both tools. Selenium is used for interacting with web browsers and automating browser actions, while Cucumber provides a structured framework for organizing and executing tests.

This combination allows for the creation of end-to-end tests that verify the behavior of web applications across different browsers and platforms, using a business-readable and maintainable format.

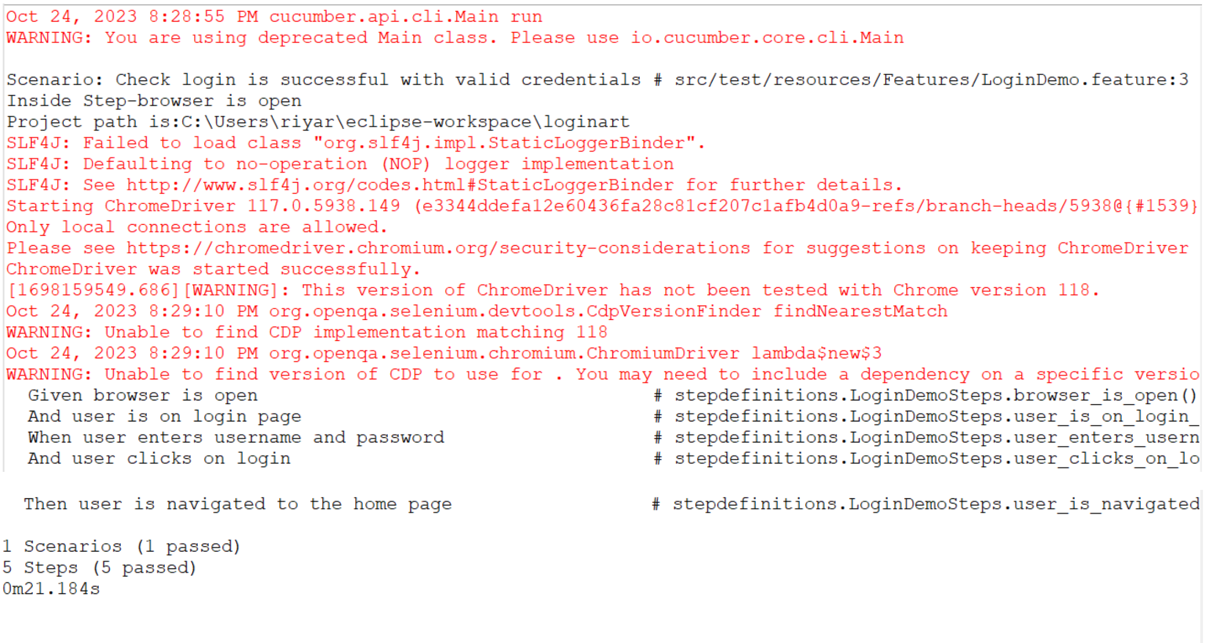
Test Case 1 – Login Test

Code:



```
package stepdefinitions;

import java.util.concurrent.TimeUnit;

public class LoginDemoSteps {

    WebDriver driver=null;
    @Given("browser is open")
    public void browser_is_open() {
        System.out.println("Inside Step-browser is open");
        String projectPath=System.getProperty("user.dir");
        System.out.println("Project path is:"+projectPath);
        System.setProperty("webdriver.chrome.driver",projectPath+"/src/test/resources/Drivers/chromedriver.exe");
        driver=new ChromeDriver();
        driver.manage().timeouts().~~implicitlyWait~~(40,TimeUnit.SECONDS);
        driver.manage().timeouts().~~pageLoadTimeout~~(40,TimeUnit.SECONDS);
        driver.manage().window().maximize();
        }

    @And("user is on login page")
    public void user_is_on_login_page() {

    driver.navigate().to("http://localhost/Project/login.php");
    }

    @When("user enters username and password")
    public void user_enters_username_and_password() throws InterruptedException {
        driver.findElement(By.name("email")).sendKeys("riyarobin6242@gmail.com");
        driver.findElement(By.name("password")).sendKeys("Riya@2001");

    }


   @And("user clicks on login")
   public void user_clicks_on_login() {

       driver.findElement(By.name("login")).click();
       }

   @Then("user is navigated to the home page")
   public void user_is_navigated_to_the_home_page() throws InterruptedException {
       driver.findElement(By.className("menu-text")).isDisplayed();

       driver.close();
       driver.quit();
   }

}
```

Screenshot:



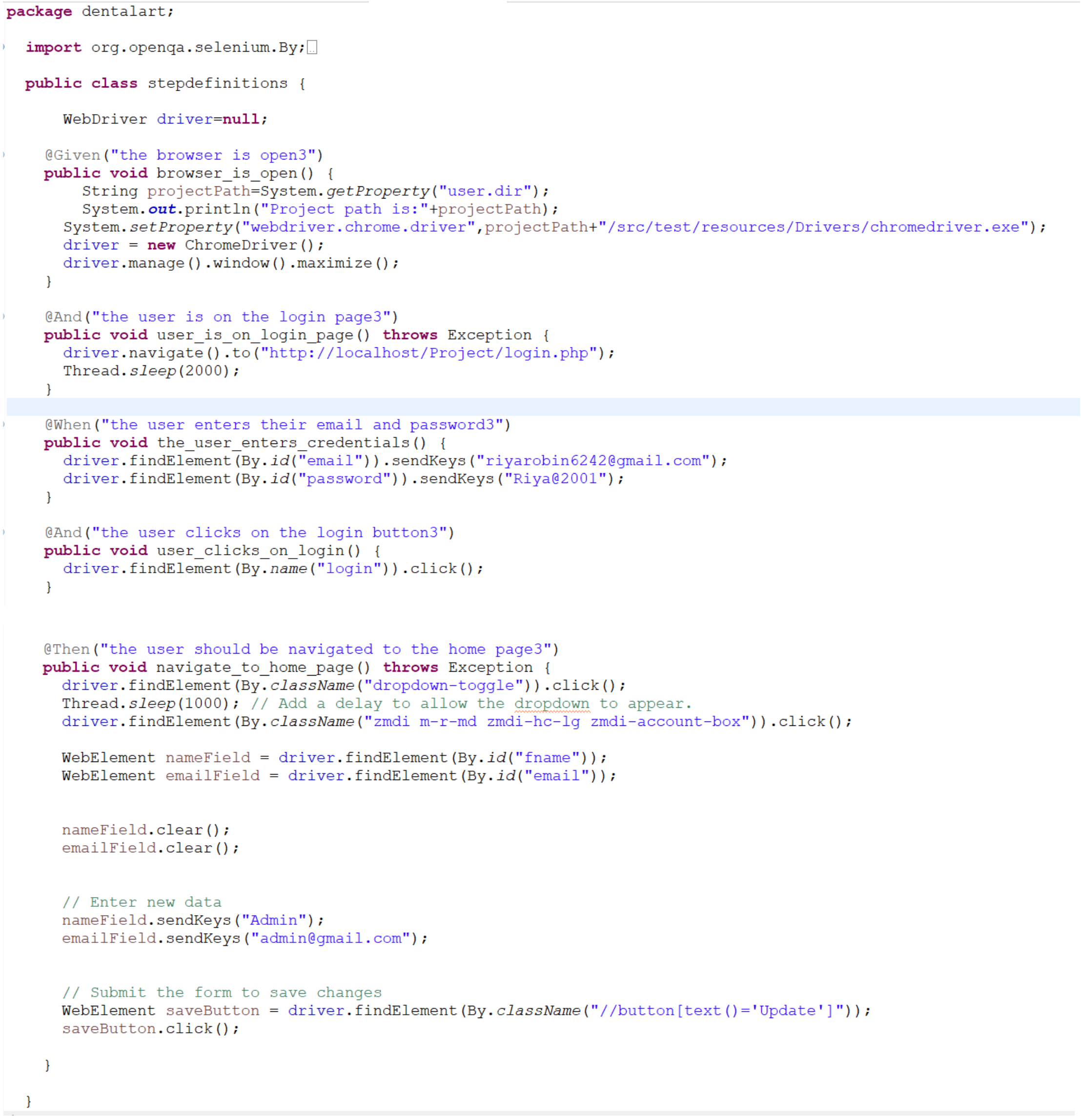
```
Oct 24, 2023 8:28:55 PM cucumber.api.cli.Main run
WARNING: You are using deprecated Main class. Please use io.cucumber.core.cli.Main

Scenario: Check login is successful with valid credentials # src/test/resources/Features/LoginDemo.feature:3
Inside Step-browser is open
Project path is:C:\Users\riyar\eclipse-workspace\loginart
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 117.0.5938.149 (e3344ddefa12e60436fa28c81cf207c1afb4d0a9-refs/branch-heads/5938@{#1539}
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver
ChromeDriver was started successfully.
[1698159549.686][WARNING]: This version of ChromeDriver has not been tested with Chrome version 118.
Oct 24, 2023 8:29:10 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find CDP implementation matching 118
Oct 24, 2023 8:29:10 PM org.openqa.selenium.chromium.ChromiumDriver lambda$new$3
WARNING: Unable to find version of CDP to use for . You may need to include a dependency on a specific versio
  Given browser is open                                   # stepdefinitions.LoginDemoSteps.browser_is_open()
  And user is on login page                               # stepdefinitions.LoginDemoSteps.user_is_on_login_
  When user enters username and password                  # stepdefinitions.LoginDemoSteps.user_enters_usern
  And user clicks on login                                # stepdefinitions.LoginDemoSteps.user_clicks_on_lo

  Then user is navigated to the home page                 # stepdefinitions.LoginDemoSteps.user_is_navigated

1 Scenarios (1 passed)
5 Steps (5 passed)
0m21.184s
```

Test Report:

Test Case 1					
Project Name: DENTCARE DENTAL CLINIC					
Login Test Case					
Test Case ID: Test_1			**Test Designed By:** Riya Robin		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 24-10-2023		
Module Name: Login Screen			**Test Executed By:** Ms. Merin Manoj		
Test Title: Admin Login			**Test Execution Date:** 24-10-2023		
Description: Verify login with valid email and password					
Pre-Condition: Admin has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	Navigation to Login Page		Dashboard should be displayed	Login page displayed	Pass
2	Provide Valid Email	Email: riyarobin6242 @gmail.com	Admin should be able to Login	Admin Logged in and navigated to Admin Dashboard with records	Pass
3	Provide Valid Password	Password: Riya@2001			
4	Click on Login button				
Post-Condition: Admin is validated with database and successfully login into account. The Account session details are logged in database					

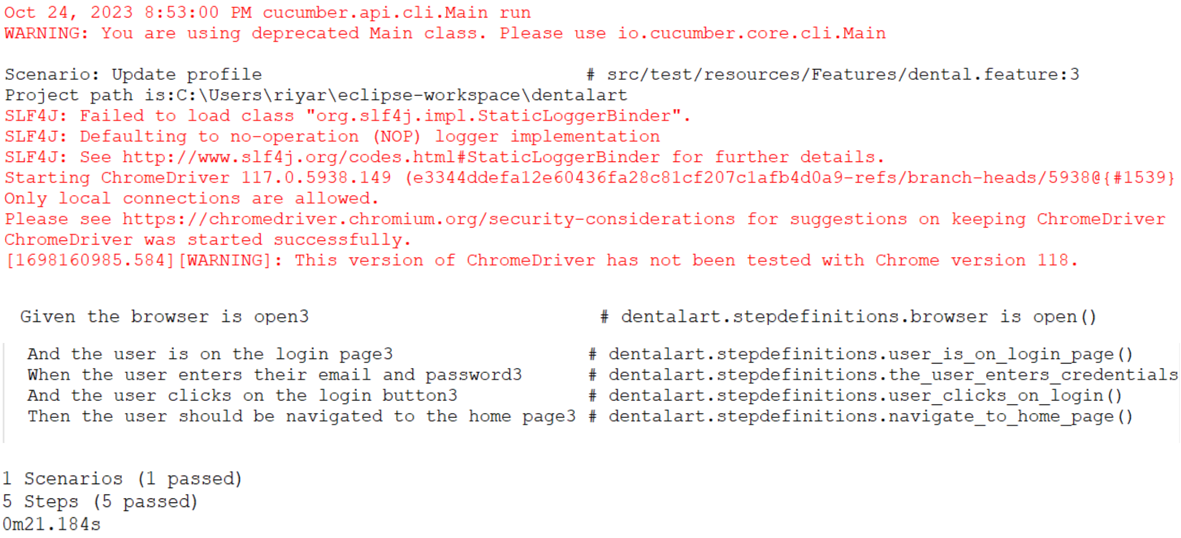
Test Case 2: Admin Profile Updation

Code:



```
package dentalart;

  import org.openqa.selenium.By;□

  public class stepdefinitions {

      WebDriver driver=null;

    @Given("the browser is open3")
    public void browser_is_open() {
        String projectPath=System.getProperty("user.dir");
        System.out.println("Project path is:"+projectPath);
      System.setProperty("webdriver.chrome.driver",projectPath+"/src/test/resources/Drivers/chromedriver.exe");
      driver = new ChromeDriver();
      driver.manage().window().maximize();
    }

    @And("the user is on the login page3")
    public void user_is_on_login_page() throws Exception {
      driver.navigate().to("http://localhost/Project/login.php");
      Thread.sleep(2000);
    }

    @When("the user enters their email and password3")
    public void the_user_enters_credentials() {
      driver.findElement(By.id("email")).sendKeys("riyarobin6242@gmail.com");
      driver.findElement(By.id("password")).sendKeys("Riya@2001");
    }

    @And("the user clicks on the login button3")
    public void user_clicks_on_login() {
      driver.findElement(By.name("login")).click();
    }


    @Then("the user should be navigated to the home page3")
    public void navigate_to_home_page() throws Exception {
      driver.findElement(By.className("dropdown-toggle")).click();
      Thread.sleep(1000); // Add a delay to allow the dropdown to appear.
      driver.findElement(By.className("zmdi m-r-md zmdi-hc-lg zmdi-account-box")).click();

      WebElement nameField = driver.findElement(By.id("fname"));
      WebElement emailField = driver.findElement(By.id("email"));


      nameField.clear();
      emailField.clear();


      // Enter new data
      nameField.sendKeys("Admin");
      emailField.sendKeys("admin@gmail.com");


      // Submit the form to save changes
      WebElement saveButton = driver.findElement(By.className("//button[text()='Update']"));
      saveButton.click();

    }

  }
```

Screenshot:



```
Oct 24, 2023 8:53:00 PM cucumber.api.cli.Main run
WARNING: You are using deprecated Main class. Please use io.cucumber.core.cli.Main

Scenario: Update profile                              # src/test/resources/Features/dental.feature:3
Project path is:C:\Users\riyar\eclipse-workspace\dentalart
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 117.0.5938.149 (e3344ddefa12e60436fa28c81cf207c1afb4d0a9-refs/branch-heads/5938@{#1539}
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver
ChromeDriver was started successfully.
[1698160985.584][WARNING]: This version of ChromeDriver has not been tested with Chrome version 118.

  Given the browser is open3                           # dentalart.stepdefinitions.browser is open()

  And the user is on the login page3                  # dentalart.stepdefinitions.user_is_on_login_page()
  When the user enters their email and password3      # dentalart.stepdefinitions.the_user_enters_credentials
  And the user clicks on the login button3            # dentalart.stepdefinitions.user_clicks_on_login()
  Then the user should be navigated to the home page3 # dentalart.stepdefinitions.navigate_to_home_page()

1 Scenarios (1 passed)
5 Steps (5 passed)
0m21.184s
```

Test report:

Test Case 2

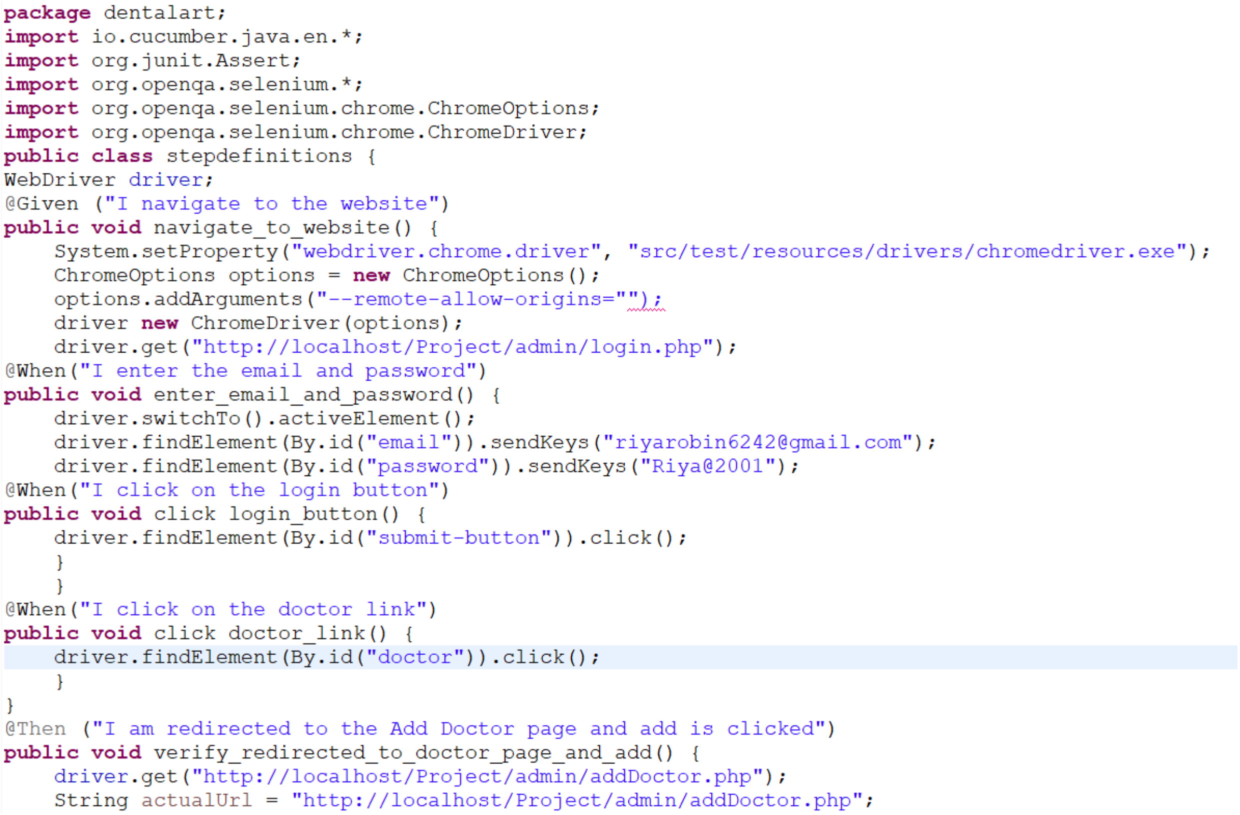
Project Name: DENTCARE DENTAL CLINIC	
Update Profile Test Case	
Test Case ID: Test_2	**Test Designed By:** Riya Robin
Test Priority (Low/Medium/High): High	**Test Designed Date:** 24-10-2023
Module Name: Update Profile Details	**Test Executed By:** Ms. Merin Manoj
Test Title: Admin Update Profile	**Test Execution Date:** 24-10-2023
Description: Update profile details by admin	
Pre-Condition: Admin has valid username and password	

Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	Navigation to Login Page		Dashboard should be displayed	Login page displayed	Pass
2	Provide Valid Email	Email: riyarobin6242@gmail.com	Admin should be able to Login	Admin Logged in and navigated to Admin Dashboard with records	Pass
3	Provide Valid Password	Password: Riya@2001			
4	Click on Login button				
5	Click on account link		Admin should be able to navigate to account page	Admin navigated to account page	Pass
6	Provide name and email id	Name: Admin Email id: admin@gmail.com			Pass
7	Click on Update button		Admin should be able to update the profile details	Admin updated the profile details	Pass

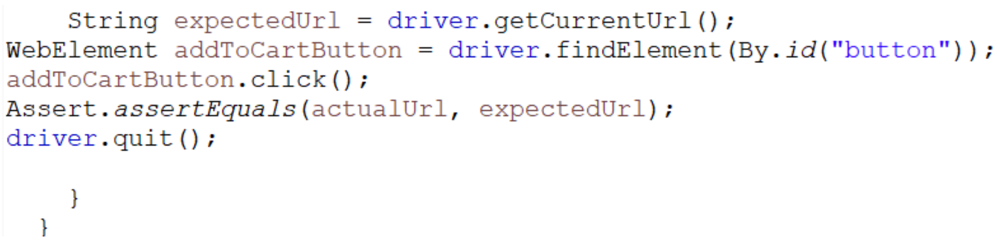
Post-Condition: Profile details are successfully updated by the admin

Test Case 3:

Code:

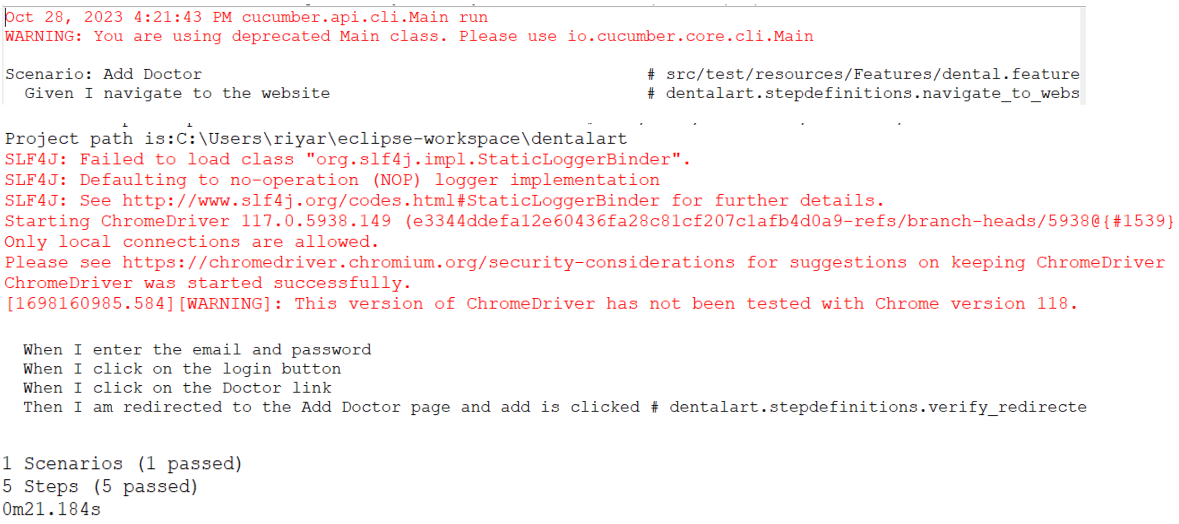


```
package dentalart;
import io.cucumber.java.en.*;
import org.junit.Assert;
import org.openqa.selenium.*;
import org.openqa.selenium.chrome.ChromeOptions;
import org.openqa.selenium.chrome.ChromeDriver;
public class stepdefinitions {
WebDriver driver;
@Given ("I navigate to the website")
public void navigate_to_website() {
    System.setProperty("webdriver.chrome.driver", "src/test/resources/drivers/chromedriver.exe");
    ChromeOptions options = new ChromeOptions();
    options.addArguments("--remote-allow-origins="");
    driver new ChromeDriver(options);
    driver.get("http://localhost/Project/admin/login.php");
@When("I enter the email and password")
public void enter_email_and_password() {
    driver.switchTo().activeElement();
    driver.findElement(By.id("email")).sendKeys("riyarobin6242@gmail.com");
    driver.findElement(By.id("password")).sendKeys("Riya@2001");
@When("I click on the login button")
public void click login_button() {
    driver.findElement(By.id("submit-button")).click();
    }
    }
@When("I click on the doctor link")
public void click doctor_link() {
    driver.findElement(By.id("doctor")).click();
    }
}
@Then ("I am redirected to the Add Doctor page and add is clicked")
public void verify_redirected_to_doctor_page_and_add() {
    driver.get("http://localhost/Project/admin/addDoctor.php");
    String actualUrl = "http://localhost/Project/admin/addDoctor.php";
```



```
     String expectedUrl = driver.getCurrentUrl();
WebElement addToCartButton = driver.findElement(By.id("button"));
addToCartButton.click();
Assert.assertEquals(actualUrl, expectedUrl);
driver.quit();

      }
   }
```

Screenshot:



```
Oct 28, 2023 4:21:43 PM cucumber.api.cli.Main run
WARNING: You are using deprecated Main class. Please use io.cucumber.core.cli.Main

Scenario: Add Doctor                                          # src/test/resources/Features/dental.feature
  Given I navigate to the website                             # dentalart.stepdefinitions.navigate_to_webs

Project path is:C:\Users\riyar\eclipse-workspace\dentalart
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 117.0.5938.149 (e3344ddefa12e60436fa28c81cf207c1afb4d0a9-refs/branch-heads/5938@{#1539}
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver
ChromeDriver was started successfully.
[1698160985.584][WARNING]: This version of ChromeDriver has not been tested with Chrome version 118.

  When I enter the email and password
  When I click on the login button
  When I click on the Doctor link
  Then I am redirected to the Add Doctor page and add is clicked # dentalart.stepdefinitions.verify_redirecte

1 Scenarios (1 passed)
5 Steps (5 passed)
0m21.184s
```

Test report:

Test Case 3

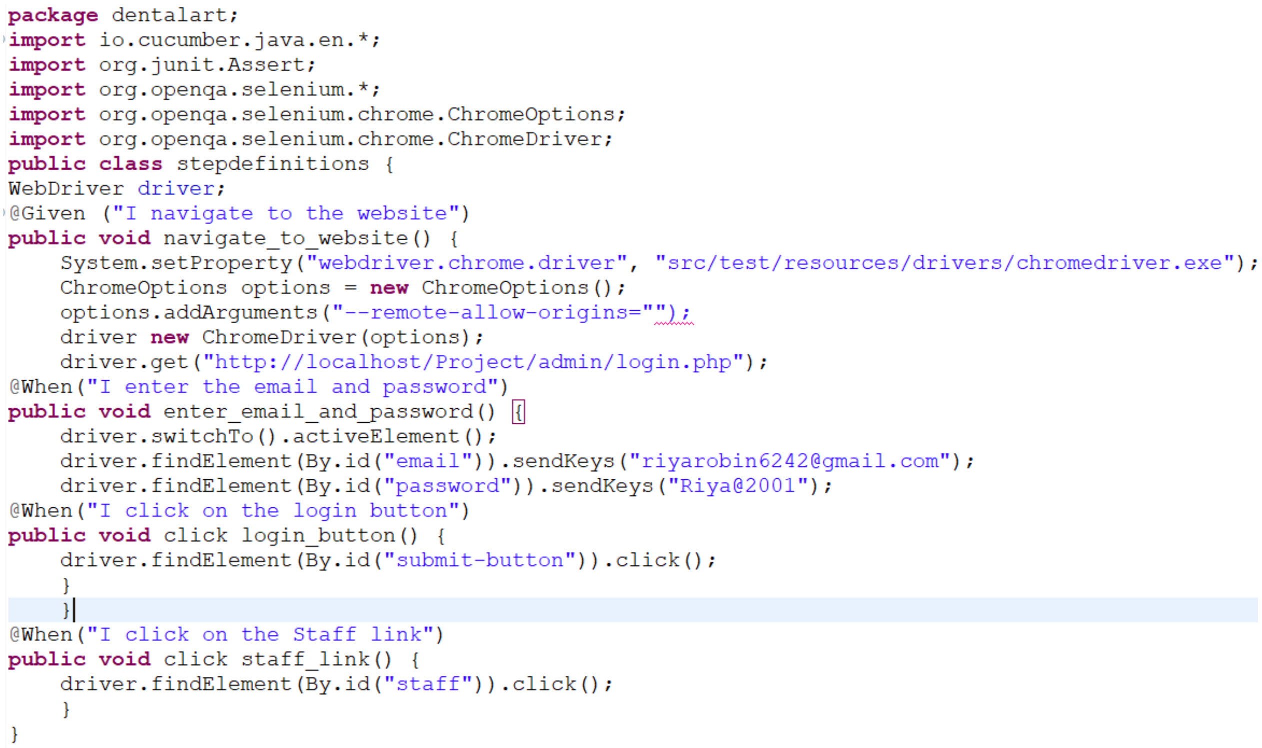
Project Name: DENTCARE DENTAL CLINIC	
Add Doctor Test Case	
Test Case ID: Test_3	**Test Designed By:** Riya Robin
Test Priority (Low/Medium/High): High	**Test Designed Date:** 28-10-2023
Module Name: Add Doctor	**Test Executed By:** Ms. Merin Manoj
Test Title: Add Doctor	**Test Execution Date:** 28-10-2023
Description: Doctor is added by the admin	
Pre-Condition: Admin has valid username and password	

Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/ Fail)**
1	Navigation to Login Page		Dashboard should be displayed	Login page displayed	Pass
2	Provide Valid Email	Email: riyarobin6242@gmail.com	Admin should be able to Login	Admin Logged in and navigated to Admin Dashboard with records	Pass
3	Provide Valid Password	Password: Riya@2001			
4	Click on Login button				
5	Admin navigates to the doctor page and clicks on add doctor button		Admin should redirect to doctor page and should be able to add doctor	Admin navigated to doctor page and added a doctor	Pass

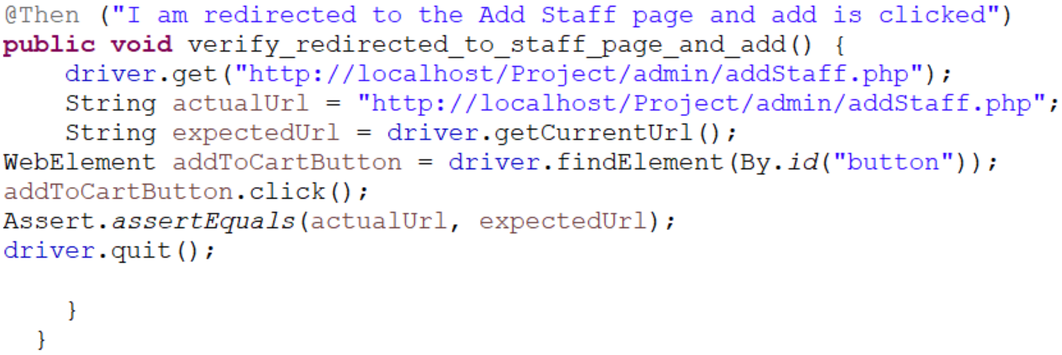
Post-Condition: Admin successfully added doctor.

Test Case 4:

Code:

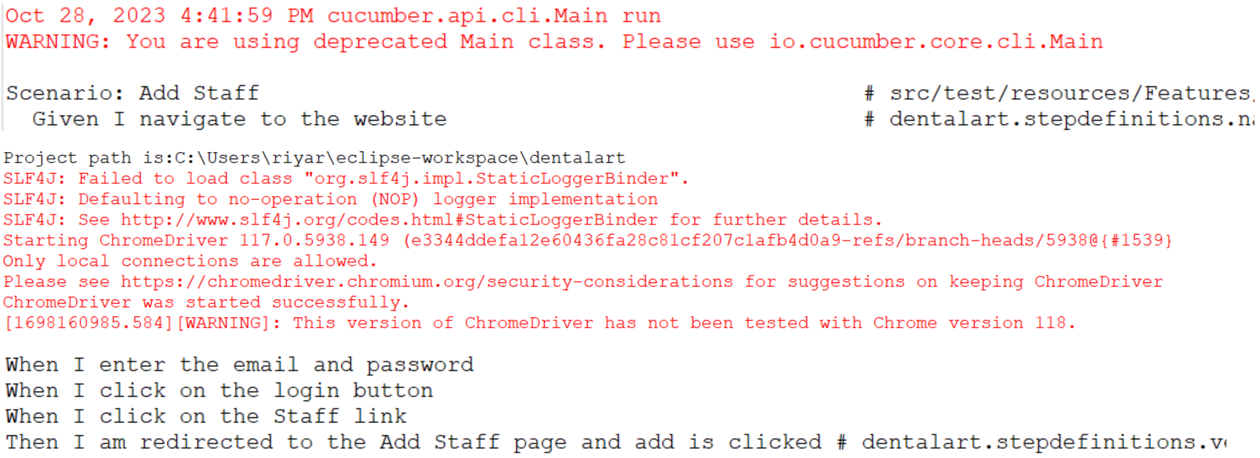


```
package dentalart;
import io.cucumber.java.en.*;
import org.junit.Assert;
import org.openqa.selenium.*;
import org.openqa.selenium.chrome.ChromeOptions;
import org.openqa.selenium.chrome.ChromeDriver;
public class stepdefinitions {
WebDriver driver;
@Given ("I navigate to the website")
public void navigate_to_website() {
    System.setProperty("webdriver.chrome.driver", "src/test/resources/drivers/chromedriver.exe");
    ChromeOptions options = new ChromeOptions();
    options.addArguments("--remote-allow-origins="");
    driver new ChromeDriver(options);
    driver.get("http://localhost/Project/admin/login.php");
@When("I enter the email and password")
public void enter_email_and_password() {
    driver.switchTo().activeElement();
    driver.findElement(By.id("email")).sendKeys("riyarobin6242@gmail.com");
    driver.findElement(By.id("password")).sendKeys("Riya@2001");
@When("I click on the login button")
public void click login_button() {
    driver.findElement(By.id("submit-button")).click();
    }
    }
@When("I click on the Staff link")
public void click staff_link() {
    driver.findElement(By.id("staff")).click();
    }
}
```

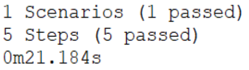


```
@Then ("I am redirected to the Add Staff page and add is clicked")
public void verify_redirected_to_staff_page_and_add() {
     driver.get("http://localhost/Project/admin/addStaff.php");
     String actualUrl = "http://localhost/Project/admin/addStaff.php";
     String expectedUrl = driver.getCurrentUrl();
WebElement addToCartButton = driver.findElement(By.id("button"));
addToCartButton.click();
Assert.assertEquals(actualUrl, expectedUrl);
driver.quit();

     }
   }
```

Screenshot:



```
Oct 28, 2023 4:41:59 PM cucumber.api.cli.Main run
WARNING: You are using deprecated Main class. Please use io.cucumber.core.cli.Main

Scenario: Add Staff                                            # src/test/resources/Features,
  Given I navigate to the website                              # dentalart.stepdefinitions.n

Project path is:C:\Users\riyar\eclipse-workspace\dentalart
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Starting ChromeDriver 117.0.5938.149 (e3344ddefa12e60436fa28c81cf207c1afb4d0a9-refs/branch-heads/5938@{#1539}
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver
ChromeDriver was started successfully.
[1698160985.584][WARNING]: This version of ChromeDriver has not been tested with Chrome version 118.

When I enter the email and password
When I click on the login button
When I click on the Staff link
Then I am redirected to the Add Staff page and add is clicked # dentalart.stepdefinitions.v
```



```
1 Scenarios (1 passed)
5 Steps (5 passed)
0m21.184s
```

Test report:

Test Case 4

Project Name: DENTCARE DENTAL CLINIC	
Add Staff Test Case	
Test Case ID: Test_4	**Test Designed By:** Riya Robin
Test Priority (Low/Medium/High): High	**Test Designed Date:** 28-10-2023
Module Name: Add Staff	**Test Executed By:** Ms. Merin Manoj
Test Title: Add Staff	**Test Execution Date:** 28-10-2023
Description: Staff is added by the admin	
Pre-Condition: Admin has valid username and password	

Step	Test Step	Test Data	Expected Result	Actual Result	Status (Pass/ Fail)
1	Navigation to Login Page		Dashboard should be displayed	Login page displayed	Pass
2	Provide Valid Email	Email: riyarobin6242 @gmail.com	Admin should be able to Login	Admin Logged in and navigated to Admin Dashboard with records	Pass
3	Provide Valid Password	Password: Riya@2001			
4	Click on Login button				
5	Admin navigates to the staff page and clicks on add staff button		Admin should redirect to staff page and should be able to add staff	Admin navigated to staff page and added a staff	Pass

Post-Condition: Admin successfully added staff.

CHAPTER 6

IMPLEMENTATION

6.1INTRODUCTION

The implementation phase of a project is where the design is transformed into a functional system. It is a crucial stage in ensuring the success of the new system, as it requires gaining user confidence that the system will work effectively and accurately. User training and documentation are key concerns during this phase. Conversion may occur concurrently with user training or at a later stage. Implementation involves the conversion of a newly revised system design into an operational system.

During this stage, the user department bears the primary workload, experiences the most significant upheaval, and has the most substantial impact on the existing system. Poorly planned or controlled implementation can cause confusion and chaos. Whether the new system is entirely new, replaces an existing manual or automated system, or modifies an existing system, proper implementation is essential to meet the organization's needs. System implementation involves all activities required to convert from the old to the new system. The system can only be implemented after thorough testing is done and found to be working according to specifications. System personnel evaluate the feasibility of the system. Implementation requires extensive effort in three main areas: education and training, system testing, and changeover. The implementation phase involves careful planning, investigating system and constraints, and designing methods to achieve changeover.

6.2 IMPLEMENTATION PROCEDURES

Software implementation is the process of installing the software in its actual environment and ensuring that it satisfies the intended use and operates as expected. In some organizations, the software development project may be commissioned by someone who will not be using the software themselves. During the initial stages, there may be doubts about the software, but it's important to ensure that resistance does not build up. This can be achieved by:

- Ensuring that active users are aware of the benefits of the new system, building their confidence in the software.
- Providing proper guidance to the users so that they are comfortable using the application.

Before viewing the system, users should know that the server program must be running on the server. Without the server object up and running, the intended process will not take place.

6.2.1 User Training

The purpose of user training is to get the user ready to test and modify the system. It is crucial for the participants to have faith in their roles in the new system in order to achieve the goal and benefits anticipated from a computer-based system. Training is more necessary as systems get more complicated. The user learns how to enter data, handle error warnings, query the database, call up routines that will generate reports, and execute other important tasks through user training.

6.2.2 Training on the Application Software

The user will need to receive the essential basic training on computer awareness after which the new application software will need to be taught to them. This will explain the fundamental principles of how to use the new system, including how the screens work, what kind of help is displayed on them, what kinds of errors are made while entering data, how each entry is validated, and how to change the date that was entered. Then, while imparting the program's training on the application, it should cover the information required by the particular user or group to operate the system or a certain component of the system. It's possible that this training will vary depending on the user group and the level of hierarchy.

6.2.3 System Maintenance

The mystery of system development is maintenance. When a software product is in the maintenance stage of its lifecycle, it is actively working. A system should be properly maintained after it has been effectively implemented. An essential part of the software development life cycle is system maintenance. In order for a system to be flexible to changes in the system environment, maintenance is required. Of course, software maintenance involves much more than just “Finding Mistakes”.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

My project is only a humble venture to satisfy the needs to manage the dental clinic management works. The proposed system provides all functionalities of the existing system along with new features. The patient user can select a slot and doctor as per the doctor’s availability and patient can view their previous records. The doctor can set their schedules for the next day and they can accept or reject the online appointments. The staff can view the available doctors and their approved appointments.

In conclusion, the concept of an online dental clinic has the potential to revolutionize the field of dental care. The digitalization of dental services offers numerous benefits for both patients and dental practitioners. It provides increased accessibility to quality dental care, convenience, and efficiency in appointment scheduling and consultations, and the opportunity to leverage advanced technologies for diagnosis and treatment planning.

7.2 FUTURE SCOPE

- Leave management of doctors.
- Leave management of staffs.
- Cancellation of online appointments.
- Telehealth and Telemedicine: Online dental clinics can leverage telehealth platforms to provide remote consultations, diagnosis, and treatment planning. This allows patients to receive dental advice and guidance from the comfort of their homes
- 3D Printing: Online dental clinics can use 3D printing for producing dental prosthetics, like crowns and dentures, quickly and with high precision, reducing the need for physical visits

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- www.scribd.com/document/379538989/Dental-Clinic-Management-System
- www.freeprojectz.com/paid-projects/php-mysql/dental-clinic-management-system
- www.freeprojectscodes.com/dental-clinic-management-system-with-full-source-code-for-free/
- Pankaj Jalote, “Software engineering: a precise approach”
- Gary B. Shelly, Harry J. Rosenblatt, “System Analysis and Design”, 2009

WEBSITES:

- www.w3schools.com
- www.bootstrap.com
- https://chat.openai.com/chat
- www.jquery.

CHAPTER 9
APPENDIX

9.1 Sample Code

login.php

```
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');

if (isset($_POST['login'])) {
    $email = $_POST['email'];
    $password = md5($_POST['password']);

    $sql = "SELECT Id, Status, Role FROM tblrole WHERE Email=:email AND
Password=:password";
    $query = $dbh->prepare($sql);
    $query->bindParam(':email', $email, PDO::PARAM_STR);
    $query->bindParam(':password', $password, PDO::PARAM_STR);
    $query->execute();
    $row = $query->fetch(PDO::FETCH_ASSOC);

    if ($row) {
        $userId = $row['Id'];
        $userRole = $row['Role'];
        $userStatus = $row['Status'];

        if ($userStatus == 1) {
            if ($userRole == 0) {
                $sql_admin = "SELECT ID, Email FROM tbladmin WHERE Email=:email";
                $query_admin = $dbh->prepare($sql_admin);
                $query_admin->bindParam(':email', $email, PDO::PARAM_STR);
                $query_admin->execute();
                $result_admin = $query_admin->fetch(PDO::FETCH_ASSOC);

                if ($result_admin) {
                    $_SESSION['damsid'] = $result_admin['ID'];
                    $_SESSION['damsemailid'] = $result_admin['Email'];
                    $_SESSION['login'] = $_POST['email'];
                    header("Location: admin/profile.php");
                }
            } else if ($userRole == 1) {
                $sql_patient = "SELECT ID, Email FROM tblpatient WHERE
Email=:email";
                $query_patient = $dbh->prepare($sql_patient);
                $query_patient->bindParam(':email', $email, PDO::PARAM_STR);
                $query_patient->execute();
                $result_patient = $query_patient->fetch(PDO::FETCH_ASSOC);

                if ($result_patient) {
                    $_SESSION['damsid'] = $result_patient['ID'];
                    $_SESSION['damsemailid'] = $result_patient['Email'];
                    header("Location: patient/profile.php");
                }
            } else if ($userRole == 2) {

                $sql_doctor = "SELECT ID, Email FROM tbldoctor WHERE Email=:email
AND Status='Active' AND Approval_status='Approved'";
                $query_doctor = $dbh->prepare($sql_doctor);
                $query_doctor->bindParam(':email', $email, PDO::PARAM_STR);
                $query_doctor->execute();
                $result_doctor = $query_doctor->fetch(PDO::FETCH_ASSOC);

                if ($result_doctor) {
                    $_SESSION['damsid'] = $result_doctor['ID'];
                    $_SESSION['damsemailid'] = $result_doctor['Email'];
                    header("Location: doctor/profile.php");
                }
            } else if ($userRole == 3) {
                $sql_staff = "SELECT ID, Email FROM tblstaff WHERE Email=:email AND
Status='Active' AND Approval_status='Approved'";
                $query_staff = $dbh->prepare($sql_staff);
                $query_staff->bindParam(':email', $email, PDO::PARAM_STR);
                $query_staff->execute();
                $result_staff = $query_staff->fetch(PDO::FETCH_ASSOC);

                if ($result_staff) {
                    $_SESSION['damsid'] = $result_staff['ID'];
                    $_SESSION['damsemailid'] = $result_staff['Email'];
                    header("Location: staff/profile.php");
                }
            } else {
                // Handle other roles if needed.
            }
        } else if ($userStatus == 2) {
            echo "<SCRIPT type='text/javascript'>alert('Permission Denied.....!!');
            window.location.replace(\"index.html\");
            </SCRIPT>";
        } else {
            echo "<SCRIPT type='text/javascript'>alert('Approval Pending.....!!');
            window.location.replace(\"index.html\");
            </SCRIPT>";
        }
    } else {
        echo "<SCRIPT type='text/javascript'>alert('Invalid User.....!!');
        window.location.replace(\"index.html\");
        </SCRIPT>";
    }
}
```

```
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>DentCare - Login Page</title>
    <link rel="stylesheet" href="css/font-awesome.min.css">
    <link rel="stylesheet" href="libs/bower/material-design-iconic-
font/dist/css/material-design-iconic-font.min.css">
    <link rel="stylesheet" href="libs/bower/animate.css/animate.min.css">
    <link rel="stylesheet" href="assets/css/bootstrap.css">
    <link rel="stylesheet" href="assets/css/core.css">
    <link rel="stylesheet" href="assets/css/misc-pages.css">
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Raleway:400,500,600,700,800,900,300">
    <style>
        /* Add some CSS for the eye icon */
        .password-toggle {
            position: relative;
        }

        .password-toggle input[type="password"] {
            padding-right: 30px;
        }

        .password-toggle .toggle-eye {
            position: absolute;
            top: 50%;
            right: 10px;
            transform: translateY(-50%);
            cursor: pointer;
        }
    </style>
</head>
<body class="simple-page">
<div id="back-to-home">
    <a href="index.html" class="btn btn-outline btn-default"><i class="fa fa-home
animated zoomIn"></i></a>
</div>
<div class="simple-page-wrap">
    <div class="simple-page-logo animated swing">
        <span style="color: white"><i class="fa fa-gg"></i></span>
        <span style="color: white">DentCare</span>
    </div>
    <!-- logo -->
    <div class="simple-page-form animated flipInY" id="login-form">
        <h4 class="form-title m-b-xl text-center">Sign In With Your DentCare
Account</h4>
```

```
        <form method="post" name="login">
            <div class="form-group">
                <input type="text" class="form-control" placeholder="Enter
Registered Email ID" required="true"
                       name="email">
            </div>
            <div class="form-group password-toggle">
                <input type="password" class="form-control" placeholder="Password"
name="password" required="true">
                <span class="toggle-eye" id="togglePassword">
                    <!-- Font Awesome eye icon for password visibility toggle -->
                    <i class="fa fa-eye" id="eyeIcon" aria-hidden="true"></i>
                </span>
            </div>
            <input type="submit" class="btn btn-primary" name="login" value="Sign
In">

        </form>
        <hr/>
        <p style="text-align: center;">OR</p>
        <div class="form-group">
            <a href="google/index.php" class="btn btn-google">
                <img src="images/g-logo.png" alt="Google Logo" class="google-logo"
style="width: 15px; height: 15px;">
                Sign In with Google
            </a>
        </div>
        <a href="index.html">Signup/Registration</a>
    </div><!-- #login-form -->
    <div class="simple-page-footer">
        <p><a href="login-system-main/recover_psw.php">FORGOT YOUR PASSWORD
?</a></p>
    </div><!-- .simple-page-footer -->
</div><!-- .simple-page-wrap -->

<script>
    const passwordInput = document.querySelector("input[name='password']");
    const eyeIcon = document.getElementById("eyeIcon");

    // Toggle password visibility when the eye icon is clicked
    eyeIcon.addEventListener("click", function () {
        if (passwordInput.type === "password") {
            passwordInput.type = "text";
            eyeIcon.classList.remove("fa-eye");
            eyeIcon.classList.add("fa-eye-slash");
        } else {
            passwordInput.type = "password";
            eyeIcon.classList.remove("fa-eye-slash");
            eyeIcon.classList.add("fa-eye");
        }
    });
</script>
</body>
</html>
```

addDoctor.php

```
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');

    if (isset($_POST['submit'])) {
    $fname=$_POST['fname'];
    $mobno=$_POST['mobno'];
    $email=$_POST['email'];
    $sid=$_POST['specid'];
    $password = md5($_POST['password']);
        // TODO: Validate and sanitize user inputs here

        $sql = "Insert Into
tbldoctor(FullName,MobileNumber,Email,Specialization,Password,Status,Approval_statu
s)Values(:fname,:mobno,:email,:sid,:password,'Active','Pending')";
        $query = $dbh->prepare($sql);
        $query->bindParam(':fname',$fname,PDO::PARAM_STR);
        $query->bindParam(':email',$email,PDO::PARAM_STR);
        $query->bindParam(':mobno',$mobno,PDO::PARAM_INT);
        $query->bindParam(':sid',$sid,PDO::PARAM_STR);
        $query->bindParam(':password',$password,PDO::PARAM_STR);
        if ($query->execute()) {
            echo '<script>alert("Doctor has been added successfully.Pending for
approval")</script>';
        } else {
            // Handle the case where the query fails
            echo '<script>alert("Error adding doctor. Please try again
later.")</script>';
        }
    }
?>
<!DOCTYPE html>
<html lang="en">
<head>
<title>DentCare - Add Doctor Profile</title>

  <link rel="stylesheet" href="libs/bower/font-awesome/css/font-awesome.min.css">
  <link rel="stylesheet" href="libs/bower/material-design-iconic-
font/dist/css/material-design-iconic-font.css">
  <!-- build:css assets/css/app.min.css -->
  <link rel="stylesheet" href="libs/bower/animate.css/animate.min.css">
  <link rel="stylesheet" href="libs/bower/fullcalendar/dist/fullcalendar.min.css">
  <link rel="stylesheet" href="libs/bower/perfect-scrollbar/css/perfect-scrollbar.css">
  <link rel="stylesheet" href="assets/css/bootstrap.css">
  <link rel="stylesheet" href="assets/css/core.css">
  <link rel="stylesheet" href="assets/css/app.css">
  <!-- endbuild -->
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Raleway:400,500,600,700,800,900,300">
  <script src="libs/bower/breakpoints.js/dist/breakpoints.min.js"></script>
  <script>
    Breakpoints();
  </script>
</head>

<body class="menubar-left menubar-unfold menubar-light theme-primary">
<!--============== start main area -->

<?php include_once('includes/header.php');?>

<?php include_once('includes/sidebar.php');?>

<!-- APP MAIN ==========-->
<main id="app-main" class="app-main">
  <div class="wrap">
  <section class="app-content">
    <div class="row">

      <div class="col-md-12">
        <div class="widget">
          <header class="widget-header">
            <h3 class="widget-title">Add Doctor</h3>
          </header><!-- .widget-header -->
          <hr class="widget-separator">
          <div class="widget-body">

            <form class="form-horizontal" method="post"onsubmit="return checkAll()">
              <div class="form-group">
                <label for="exampleTextInput1" class="col-sm-3 control-label">Doctor Name:</label>
                <div class="col-sm-9">
                  <input id="fname" type="text" class="form-control" placeholder="Full Name" name="fname" required="true" onkeyup="firstName()">
                  <div id="nameError" class="error"></div>
                </div>
              </div>
              <div class="form-group">
                <label for="email2" class="col-sm-3 control-label">Email:</label>
                <div class="col-sm-9">
                  <input type="email" class="form-control"placeholder="Email"
id="email" name="email" required='true'onkeyup="emailUser()">
                  <div id="emailError" class="error"></div>
                </div>
              </div>
               <div class="form-group">
                <label for="email2" class="col-sm-3 control-label">Mobile
Number:</label>
                <div class="col-sm-9">
                <input type="text" class="form-control"placeholder="Mobile Number"
id="mobno" name="mobno" required="true"onkeyup="phoneUser()">
                <div id="mobileError" class="error"></div>
                </div>
              </div>
              <div class="form-group">
                <label for="email2" class="col-sm-3 control-
label"placeholder="Specialization"onkeyup="specId()">Specialization:</label>
                <div class="col-sm-9">
                <select class="form-control" name="specid" id="specid">
                <div id="specError" class="error"></div>
                <option value="">Choose Specialization</option>
    <?php
    $sql1="SELECT * from tblspecialization";
    $query1 = $dbh -> prepare($sql1);
    $query1->execute();
    $results1=$query1->fetchAll(PDO::FETCH_OBJ);


    if($query1->rowCount() > 0)
    {
    foreach($results1 as $row1)
    {               ?>
                    <option value="<?php  echo htmlentities($row1-
>Specialization);?>"><?php  echo htmlentities($row1-
>Specialization);?></option><?php }} ?>
                </select>
                </div>
              </div>
              <div class="form-group">
                <label for="email2" class="col-sm-3 control-
label">Password:</label>
                <div class="col-sm-9">
                <input type="text" class="form-control" id="password"
name="password"value="123"required>
                </div>
              </div>

              <div class="row">
```

```
                <div class="col-sm-9 col-sm-offset-3">
                  <button type="submit" class="btn btn-success" 
name="submit"onkeyup="checkAll()">Update</button>
                </div>
              </div>

            </form>
          </div><!-- .widget-body -->
        </div><!-- .widget -->
      </div><!-- END column -->

    </div><!-- .row -->
  </section><!-- #dash-content -->
</div><!-- .wrap -->
  <!-- APP FOOTER -->
  <?php include_once('includes/footer.php');?>
  <!-- /#app-footer -->
</main>
<!--========== END app main -->


  <!-- SIDE PANEL -->
  <script>
  function firstName() {
    const nameInput = document.getElementById("fname");
            const nameError = document.getElementById("nameError");
            nameError.style.color = "red";
            const name = nameInput.value.trim();
            const nameRegex = /^[A-Za-z]+$/;
            let hasConsecutiveSameChars = false;
            for (let i = 0; i < name.length - 1; i++) {
                if (name[i] === name[i + 1]) {
                    hasConsecutiveSameChars = true;
                    break;
                }
            }
            if (!nameRegex.test(name)) {
                nameError.textContent = "Name should only contain letters";
            } else if (hasConsecutiveSameChars) {
                nameError.textContent = "Name should not have consecutive same 
characters";
            } else {
                nameError.textContent = "";
            }
        }
function emailUser() {
    const email = document.getElementById("email").value.trim();
            const emailError = document.getElementById("emailError");
            const emailRegex = /^[^\s@]+@gmail\.com$/;
            if (!emailRegex.test(email)) {
                emailError.textContent = "Invalid email format";
                emailError.style.color = "red";
                return false;
            }

            emailError.textContent = "";
            return true;
        }

function specId(){
    const dropdown = document.getElementById("specid");
        const specError = document.getElementById("specError");
      if (dropdown.value === "") {
        specError.textContent = "Please select an option";
        specError.style.color = "red";
        return false;
      } else {
        specError.textContent = "";
        return true;
      }
    }

function phoneUser() {
    const mobile = document.getElementById("mobno").value.trim();
            const mobileError = document.getElementById("mobileError");
            const mobileRegex = /^[789]\d{9}$/;
            const sameDigitRegex = /^(\d)\1+$/;

            if (!mobileRegex.test(mobile) || sameDigitRegex.test(mobile)) {
                mobileError.textContent = "Invalid mobile number";
                mobileError.style.color = "red";
                return false;
            }

            mobileError.textContent = "";
            return true;
        }

function checkAll() {
if(firstName()&&specId()&&emailUser()&&phoneUser())
{
return true;
}
}
</script>

  <!-- build:js assets/js/core.min.js -->
  <script src="libs/bower/jquery/dist/jquery.js"></script>
  <script src="libs/bower/jquery-ui/jquery-ui.min.js"></script>
  <script src="libs/bower/jQuery-Storage-API/jquery.storageapi.min.js"></script>
  <script src="libs/bower/bootstrap-sass/assets/javascripts/bootstrap.js"></script>
  <script src="libs/bower/jquery-slimscroll/jquery.slimscroll.js"></script>
  <script src="libs/bower/perfect-scrollbar/js/perfect-scrollbar.jquery.js"></script>
  <script src="libs/bower/PACE/pace.min.js"></script>
  <!-- endbuild -->

  <!-- build:js assets/js/app.min.js -->
  <script src="assets/js/library.js"></script>
  <script src="assets/js/plugins.js"></script>
  <script src="assets/js/app.js"></script>
  <!-- endbuild -->
  <script src="libs/bower/moment/moment.js"></script>
  <script src="libs/bower/fullcalendar/dist/fullcalendar.min.js"></script>
  <script src="assets/js/fullcalendar.js"></script>
</body>
</html>
```

setSchedule.php

```
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');


?>

<!DOCTYPE html>
<html lang="en">
<head>
<title>DentCare - Add Doctor Schedule</title>

  <link rel="stylesheet" href="libs/bower/font-awesome/css/font-awesome.min.css">
  <link rel="stylesheet" href="libs/bower/material-design-iconic-font/dist/css/material-design-iconic-font.css">
  <!-- build:css assets/css/app.min.css -->
  <link rel="stylesheet" href="libs/bower/animate.css/animate.min.css">
  <link rel="stylesheet" href="libs/bower/fullcalendar/dist/fullcalendar.min.css">
  <link rel="stylesheet" href="libs/bower/perfect-scrollbar/css/perfect-scrollbar.css">
  <link rel="stylesheet" href="assets/css/bootstrap.css">
  <link rel="stylesheet" href="assets/css/core.css">
  <link rel="stylesheet" href="assets/css/app.css">
  <!-- endbuild -->
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Raleway:400,500,600,700,800,900,300">
  <script src="libs/bower/breakpoints.js/dist/breakpoints.min.js"></script>
  <script>
    Breakpoints();
  </script>
    <style>
        body {
            font-family: Arial, sans-serif;
        }
        .container {
            max-width: 400px;
            margin: 0 auto;
            padding: 20px;
            border: 1px solid #ccc;
            border-radius: 5px;
            background-color: #f9f9f9;
        }
        h2 {
            text-align: center;
        }
        .form-group {
            margin-bottom: 10px;
        }
        label {
            display: block;
            font-weight: bold;
        }
        input[type="date"] {
            width: 100%;
            padding: 8px;
            border: 1px solid #ccc;
            border-radius: 3px;
        }
        .time-slots label {
            display: inline-block;
            margin-right: 10px;
        }
        .btn-submit {
            display: block;
            width: 100%;
            padding: 10px;
            background-color: #007bff;
            color: #fff;
            border: none;
            border-radius: 3px;
            cursor: pointer;
        }
    </style>
</head>
```

```
<body class="menubar-left menubar-unfold menubar-light theme-primary">
<!--============== start main area -->

<?php include_once('includes/header.php');?>

<?php include_once('includes/sidebar.php');?>
<main id="app-main" class="app-main">
  <div class="wrap">
  <section class="app-content">
    <div class="row">

      <div class="col-md-12">
        <div class="widget">
          <header class="widget-header">
            <h3 class="widget-title">Set Your Schedule</h3>
          </header><!-- .widget-header -->
          <hr class="widget-separator">
          <div class="widget-body">

      <form class="form-horizontal" method="post" id="scheduleForm">
            <div class="form-group">
                <label for="availdate">Select Date:</label>
                <input type="date" id="availdate" name="availdate" required>
            </div>
            <div class="form-group">
                <label>Select Time Slots:</label><br>
                <div class="time-slots">
                    <label><input type="checkbox" name="timeslot[]" value="9am-
9.30am"> 9am-9.30am</label>
                    <label><input type="checkbox" name="timeslot[]" value="9.30am-
10am"> 9.30am-10am</label>
                    <label><input type="checkbox" name="timeslot[]" value="10am-
10.30am"> 10am-10.30am</label>
          <label><input type="checkbox" name="timeslot[]" value="10.30am-11.00am">
10.30am-11.00am</label>
          <label><input type="checkbox" name="timeslot[]" value="11am-11.30am">
11am-11.30am</label>
          <label><input type="checkbox" name="timeslot[]" value="11.30am-12.00am">
11.30am-12.00am</label>
                    <!-- Add more time slots as needed -->
                </div>
            </div>
            <button type="submit" class="btn-submit">Submit</button>
      </form>
          </div><!-- .widget-body -->
        </div><!-- .widget -->
      </div><!-- END column -->

    </div><!-- .row -->
```

```
 </section><!-- #dash-content -->
</div><!-- .wrap -->
 <!-- APP FOOTER -->
 <?php include_once('includes/footer.php');?>
 <!-- /#app-footer -->
</main>

  <script>
    // JavaScript to handle form submission
    document.getElementById("scheduleForm").addEventListener("submit", function
(event) {
      event.preventDefault(); // Prevent the form from submitting normally

      // Get selected date and time slots
      var selectedDate = document.getElementById("availdate").value;
      var selectedTimeSlots = [];
      var checkboxes =
document.querySelectorAll('input[name="timeslot[]"]:checked');
      checkboxes.forEach(function(checkbox) {
        selectedTimeSlots.push(checkbox.value);
      });

      // You can now send the selectedDate and selectedTimeSlots to the
server via AJAX or perform any other actions.

      // For example, displaying them in an alert for testing:
      alert("Selected Date: " + selectedDate + "\nSelected Time Slots: " +
selectedTimeSlots.join(", "));
    });
  </script>
 <script src="libs/bower/jquery/dist/jquery.js"></script>
 <script src="libs/bower/jquery-ui/jquery-ui.min.js"></script>
 <script src="libs/bower/jQuery-Storage-API/jquery.storageapi.min.js"></script>
 <script src="libs/bower/bootstrap-sass/assets/javascripts/bootstrap.js"></script>
 <script src="libs/bower/jquery-slimscroll/jquery.slimscroll.js"></script>
 <script src="libs/bower/perfect-scrollbar/js/perfect-
scrollbar.jquery.js"></script>
 <script src="libs/bower/PACE/pace.min.js"></script>
 <!-- endbuild -->

 <!-- build:js assets/js/app.min.js -->
 <script src="assets/js/library.js"></script>
 <script src="assets/js/plugins.js"></script>
 <script src="assets/js/app.js"></script>
 <!-- endbuild -->
 <script src="libs/bower/moment/moment.js"></script>
 <script src="libs/bower/fullcalendar/dist/fullcalendar.min.js"></script>
 <script src="assets/js/fullcalendar.js"></script>
</body>
```

```
</html>
```

bookAppointment.php

```
<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');
include_once('includes/header.php');
include_once('includes/sidebar.php');

$ptid = $_SESSION['damsid'];
$ap_date = $_POST['ap_date'];
$did = $_POST['did'];
$tsslot = $_POST['tsslot'];
$status = 0;

// Check if the selected time slot is available (not booked by 3 people)
$sql_check_slot = "SELECT COUNT(*) as booked_count FROM tblbooking WHERE did = :did
AND bkdate = :ap_date AND bktimeslot = :tsslot";
$check_stmt = $dbh->prepare($sql_check_slot);
$check_stmt->bindParam(':did', $did, PDO::PARAM_INT);
$check_stmt->bindParam(':ap_date', $ap_date, PDO::PARAM_STR);
$check_stmt->bindParam(':tsslot', $tsslot, PDO::PARAM_STR);
$check_stmt->execute();
$slot_info = $check_stmt->fetch(PDO::FETCH_ASSOC);

if ($slot_info['booked_count'] < 3) {
    $sql_insert_booking = "INSERT INTO tblbooking(did, bkdate, bktimeslot, status,
LoginId) VALUES (:did, :ap_date, :tsslot, :status, :ptid)";
    $insert_stmt = $dbh->prepare($sql_insert_booking);
    $insert_stmt->bindParam(':did', $did, PDO::PARAM_INT);
    $insert_stmt->bindParam(':ap_date', $ap_date, PDO::PARAM_STR);
    $insert_stmt->bindParam(':tsslot', $tsslot, PDO::PARAM_STR);
    $insert_stmt->bindParam(':status', $status, PDO::PARAM_INT);
    $insert_stmt->bindParam(':ptid', $ptid, PDO::PARAM_INT);

    if ($insert_stmt->execute()) {
        echo "<SCRIPT type='text/javascript'>alert('Appointment Booked
Successfully');
             window.location.replace(\"viewdrbookings.php\");
             </SCRIPT>";
    } else {
        echo "<SCRIPT type='text/javascript'>alert('Appointment Booking Failed');
             window.location.replace(\"drbooking.php\");
             </SCRIPT>";
    }
} else {
    echo "<SCRIPT type='text/javascript'>alert('This time slot is fully booked.
Please choose another time slot.');
```

```
         window.location.replace(\"drbooking.php\");
         </SCRIPT>";
}
?>

<!DOCTYPE html>
<html lang="en">
<head>

  <title>DentCare - Appointment</title>

  <link rel="stylesheet" href="libs/bower/font-awesome/css/font-awesome.min.css">
  <link rel="stylesheet" href="libs/bower/material-design-iconic-
font/dist/css/material-design-iconic-font.css">
  <!-- build:css assets/css/app.min.css -->
  <link rel="stylesheet" href="libs/bower/animate.css/animate.min.css">
  <link rel="stylesheet" href="libs/bower/fullcalendar/dist/fullcalendar.min.css">
  <link rel="stylesheet" href="libs/bower/perfect-scrollbar/css/perfect-
scrollbar.css">
  <link rel="stylesheet" href="assets/css/bootstrap.css">
  <link rel="stylesheet" href="assets/css/core.css">
  <link rel="stylesheet" href="assets/css/app.css">
  <!-- endbuild -->
  <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Raleway:400,500,600,700,800,900,300">
  <script src="libs/bower/breakpoints.js/dist/breakpoints.min.js"></script>
  <script>
    Breakpoints();
  </script>
  <script type="text/javascript">
function checkpass()
{
if(document.changepassword.newpassword.value!=document.changepassword.confirmpasswo
rd.value)
{
alert('New Password and Confirm Password field does not match');
document.changepassword.confirmpassword.focus();
return false;
}
return true;
}

</script>
</head>

<body class="menubar-left menubar-unfold menubar-light theme-primary">
<!--============== start main area -->

<?php include_once('includes/header.php');?>
```

```
<?php include_once('includes/sidebar.php');?>

<!-- APP MAIN ==========-->
<main id="app-main" class="app-main">
  <div class="wrap">
  <section class="app-content">
    <div class="row">

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

          </header><!-- .widget-header -->
          <hr class="widget-separator">
          <div class="widget-body">


          <section class="content-header">

      </section>

       <!-- Main content -->
      <section class="content">
        <div class="container-fluid">
          <!-- Small boxes (Stat box) -->
          <div class="row">

              <div class="col-lg-8 col-12">

          </div>
          </div>
          <!-- /.row -->
          <!-- Main row -->
          <div class="row">
            <!-- Left col -->
            </div>

  </div>

<!-- right col -->
</div>
<!-- /.row (main row) -->
</div>
<?php
include('footer.php');
?>
<!-- /.container-fluid -->
</section>
```

```
<!-- /.content -->
</div>
<!-- /.content-wrapper -->




</div>
<!-- ./wrapper -->


</div><!-- .widget-body -->
</div><!-- .widget -->
</div><!-- END column -->

</div><!-- .row -->
</section><!-- #dash-content -->
</div><!-- .wrap -->
<!-- APP FOOTER -->
<?php include_once('includes/footer.php');?>
<!-- /#app-footer -->
</main>
<!--========== END app main -->
<!-- SIDE PANEL -->


<!-- build:js assets/js/core.min.js -->
<script src="libs/bower/jquery/dist/jquery.js"></script>
<script src="libs/bower/jquery-ui/jquery-ui.min.js"></script>
<script src="libs/bower/jQuery-Storage-API/jquery.storageapi.min.js"></script>
<script src="libs/bower/bootstrap-sass/assets/javascripts/bootstrap.js"></script>
<script src="libs/bower/jquery-slimscroll/jquery.slimscroll.js"></script>
<script src="libs/bower/perfect-scrollbar/js/perfect-scrollbar.jquery.js"></script>
<script src="libs/bower/PACE/pace.min.js"></script>
<!-- endbuild -->

<!-- build:js assets/js/app.min.js -->
<script src="assets/js/library.js"></script>
<script src="assets/js/plugins.js"></script>
<script src="assets/js/app.js"></script>
<!-- endbuild -->
<script src="libs/bower/moment/moment.js"></script>
<script src="libs/bower/fullcalendar/dist/fullcalendar.min.js"></script>
<script src="assets/js/fullcalendar.js"></script>
</body>
</html>
```

viewAppointment.php

```
<?php
```

```
     session_start();
     include('includes/dbconnection.php');
     include_once('includes/header.php');
 include_once('includes/sidebar.php');

    $did = $_GET['t'];
      $sql="select * from tblpatient inner join tblbooking on
tblpatient.ID=tblbooking.LoginId where did='".$did."' order by tblbooking.bkdate
desc";
      $ex2 = $dbh->prepare($sql);
      $ex2->execute();
      $flag=0;
       while ($row = $ex2->fetch(PDO::FETCH_ASSOC))
      {
        $flag=1;
      }
      $sql="select * from tblpatient inner join tblbooking on
tblpatient.ID=tblbooking.LoginId where did='".$did."' order by tblbooking.bkdate
desc";
      $ex2 = $dbh->prepare($sql);
      $ex2->execute();

?>

  <!DOCTYPE html>
<html lang="en">
<head>

  <title>DentCare - Staff Profile</title>

  <link rel="stylesheet" href="libs/bower/font-awesome/css/font-awesome.min.css">
  <link rel="stylesheet" href="libs/bower/material-design-iconic-
font/dist/css/material-design-iconic-font.css">
  <!-- build:css assets/css/app.min.css -->
  <link rel="stylesheet" href="libs/bower/animate.css/animate.min.css">
  <link rel="stylesheet" href="libs/bower/fullcalendar/dist/fullcalendar.min.css">
  <link rel="stylesheet" href="libs/bower/perfect-scrollbar/css/perfect-
scrollbar.css">
  <link rel="stylesheet" href="assets/css/bootstrap.css">
  <link rel="stylesheet" href="assets/css/core.css">
  <link rel="stylesheet" href="assets/css/app.css">
  <!-- endbuild -->
  <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Raleway:400,500,600,700,800,900,300">
  <script src="libs/bower/breakpoints.js/dist/breakpoints.min.js"></script>
  <script>
    Breakpoints();
  </script>
</head>
```

```
<body class="menubar-left menubar-unfold menubar-light theme-primary">
<!--============== start main area -->

<?php include_once('includes/header.php');?>

<?php include_once('includes/sidebar.php');?>

<!-- APP MAIN ==========-->
<main id="app-main" class="app-main">
  <div class="wrap">
  <section class="app-content">
    <div class="row">

      <div class="col-md-12">
        <div class="widget">
          <header class="widget-header">
            <h3 class="widget-title">Appointment Booking Details</h3>
          </header><!-- .widget-header -->
          <hr class="widget-separator">
          <div class="widget-body">
               <div class="form-group">
                 <font color="orange" size="5px"><b><center>Booking Date : <?php
echo date('d-m-Y'); ?></center></b></font>
               </div>
<?php
  if($flag==1)
  {
?>

        <table class="table table-bordered" id="table"  data-toggle="table" data-
height="460"  data-pagination="true"   data-search="true">
        <thead>
    <tr style="text-align: center;">
      <th>#</th>
      <th>Patient Name</th>
      <th>Booking Date</th>
      <th>Time Slot</th>
    </tr>
  </thead>
  <tbody>
  <?php $c=1;
  while ($row = $ex2->fetch(PDO::FETCH_ASSOC))
  { ?>
    <tr style="text-align: center;">
      <td><?php echo $c; $c=$c+1; ?></td>
      <td><?php echo $row['FullName']; ?></td>
      <td><?php echo $row['bkdate']; ?></td>
      <td><?php
```

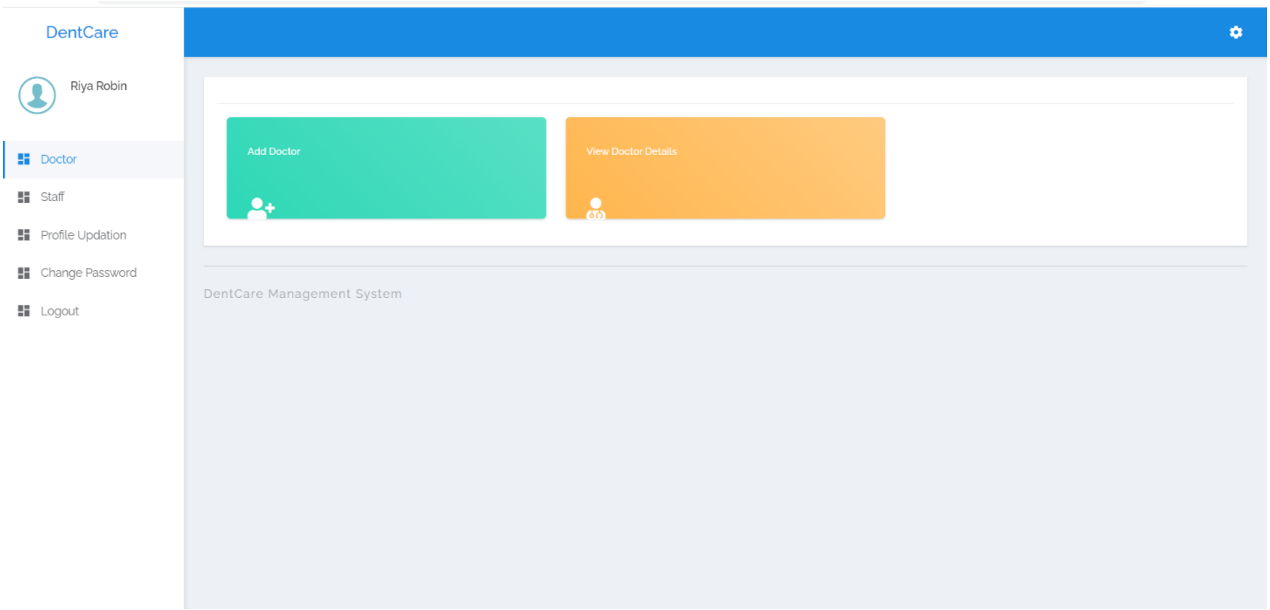
```
        $ts=$row['bktimeslot'];
        if($ts=='s1')
        {
          echo "9am-9.30am";
        }
        if($ts=='s2')
        {
          echo "9.30am-10am";
        }
        if($ts=='s3')
        {
          echo "10am-10.30am";
        }
        if($ts=='s4')
        {
          echo "11am-11.30am";
        }
        if($ts=='s5')
        {
          echo "12.30pm-1pm";
        }
        if($ts=='s6')
        {
          echo "2pm-2.30pm";
        }
        if($ts=='s7')
        {
          echo "3pm-3.30pm";
        }
?></td>

      <!-- <td><button class="btn btn-secondary" data-toggle="modal" data-
target="#example<?php //echo $row['ambkey']; ?>">Notify</button></td> -->

    </tr>
  <?php } ?>
  </tbody>
</table>
<div class="form-group"> <br>
          <a href="viewdrs.php"><button class="btn btn-outline-success btn-
block"><i class="fa fa-undo" aria-hidden="true"></i>&nbsp;&nbsp;Back To
Home</button></a>
      </div>
<?php
}
else
{ ?>
      <div class="form-group"> <br><br><br><br><br><br><br><br>
```

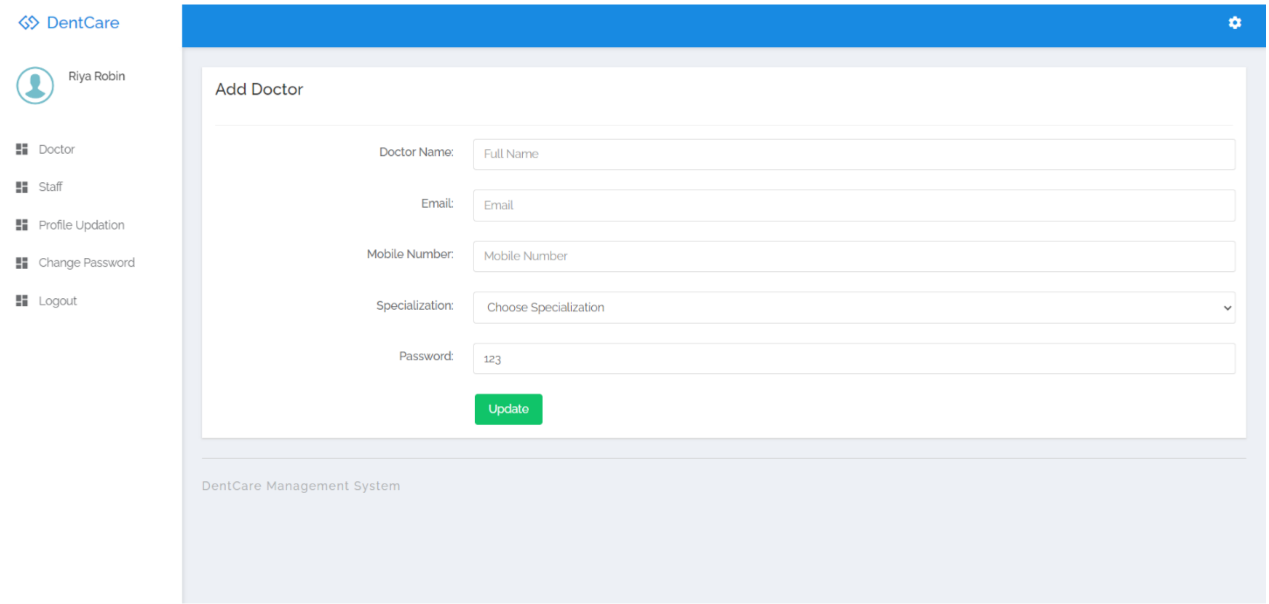
```
          <font color="red" size="4px"><b><center>No Booking Found For The Selected Date</center></b></font><br>
          <a href="viewdrs.php"><button class="btn btn-danger btn-block"><i class="fa fa-undo" aria-hidden="true"></i>&nbsp;&nbsp;Back To Home</button></a>
      </div>
<?php
}
?>


          </div><!-- .widget-body -->
        </div><!-- .widget -->
      </div><!-- END column -->

    </div><!-- .row -->
  </section><!-- #dash-content -->
</div><!-- .wrap -->
  <!-- APP FOOTER -->
  <?php include_once('includes/footer.php');?>
  <!-- /#app-footer -->
</main>
<!--========== END app main -->

  <!-- build:js assets/js/core.min.js -->
  <script src="libs/bower/jquery/dist/jquery.js"></script>
  <script src="libs/bower/jquery-ui/jquery-ui.min.js"></script>
  <script src="libs/bower/jQuery-Storage-API/jquery.storageapi.min.js"></script>
  <script src="libs/bower/bootstrap-sass/assets/javascripts/bootstrap.js"></script>
  <script src="libs/bower/jquery-slimscroll/jquery.slimscroll.js"></script>
  <script src="libs/bower/perfect-scrollbar/js/perfect-scrollbar.jquery.js"></script>
  <script src="libs/bower/PACE/pace.min.js"></script>
  <!-- endbuild -->

  <!-- build:js assets/js/app.min.js -->
  <script src="assets/js/library.js"></script>
  <script src="assets/js/plugins.js"></script>
  <script src="assets/js/app.js"></script>
  <!-- endbuild -->
  <script src="libs/bower/moment/moment.js"></script>
  <script src="libs/bower/fullcalendar/dist/fullcalendar.min.js"></script>
  <script src="assets/js/fullcalendar.js"></script>
</body>
</html>
```

9.2 Screen Shots

ADMIN:

Add Doctor:





Add Doctor

Doctor Name: Full Name

Email: Email

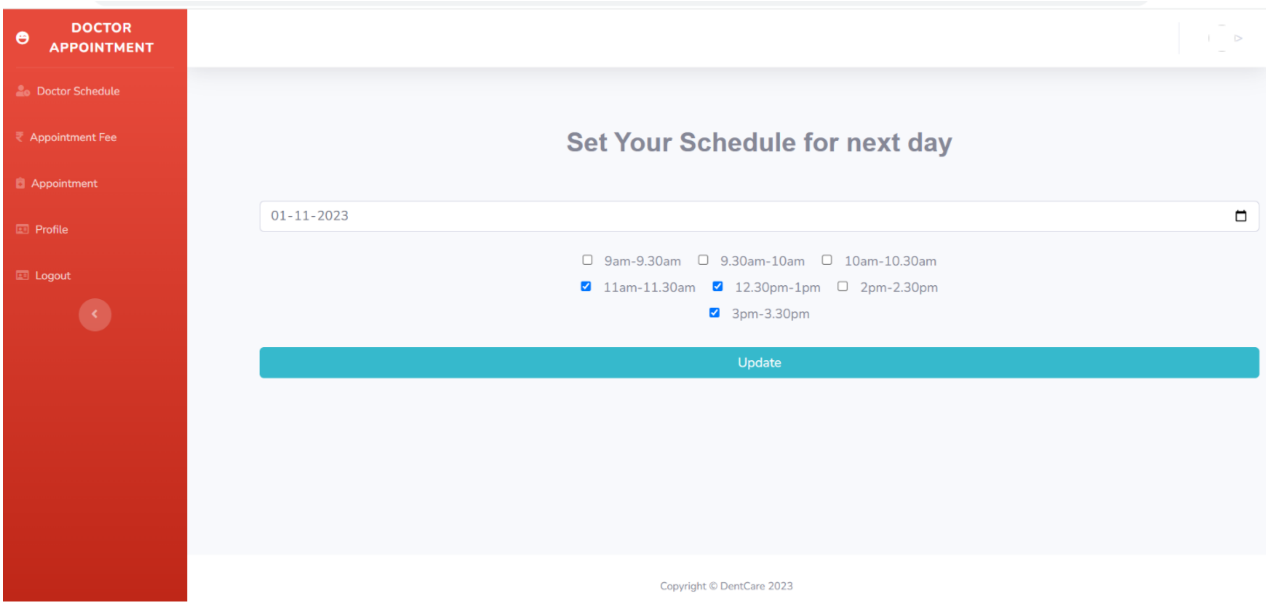
Mobile Number: Mobile Number

Specialization: Choose Specialization

Password: 123

DOCTOR:

Set Schedule:



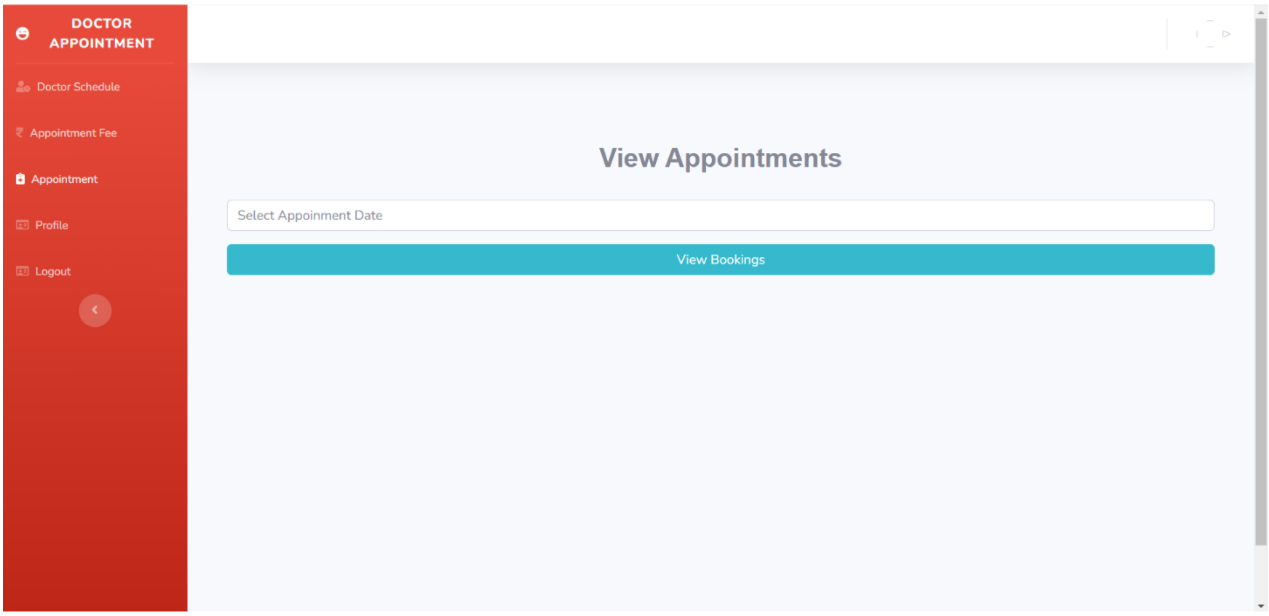
Set Your Schedule for next day

01-11-2023

□ 9am-9.30am □ 9.30am-10am □ 10am-10.30am
☑ 11am-11.30am ☑ 12.30pm-1pm □ 2pm-2.30pm
☑ 3pm-3.30pm

Copyright © DentCare 2023

View Appointments:

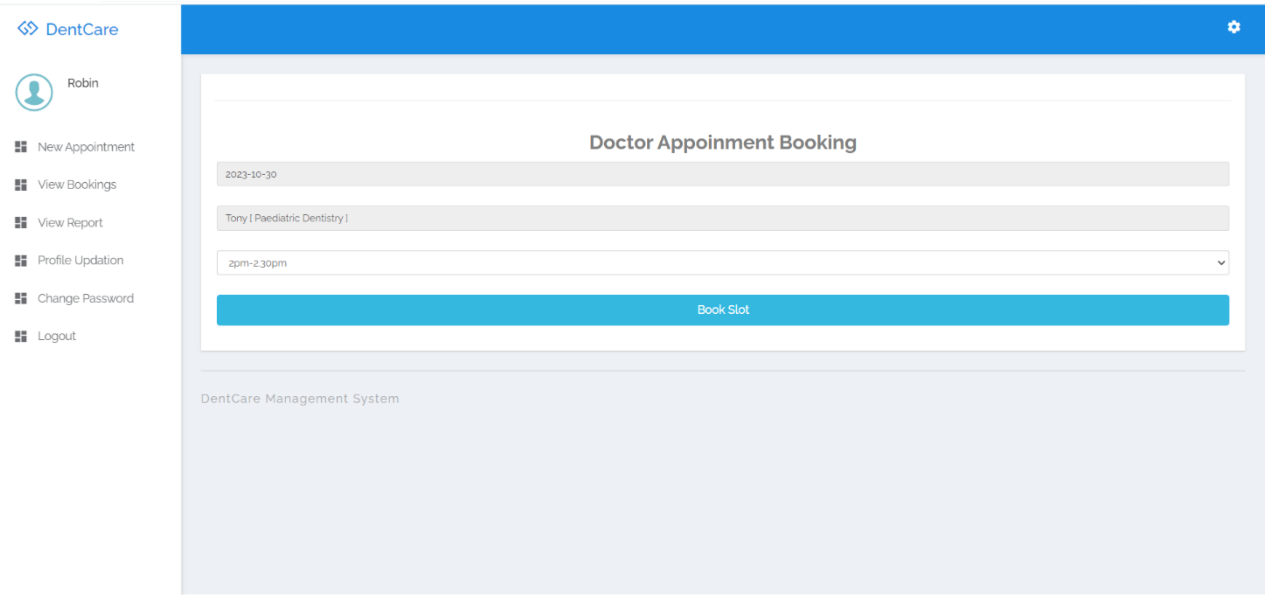


View Appointments

Select Appoinment Date

PATIENT:

Book Appointments:



Doctor Appoinment Booking

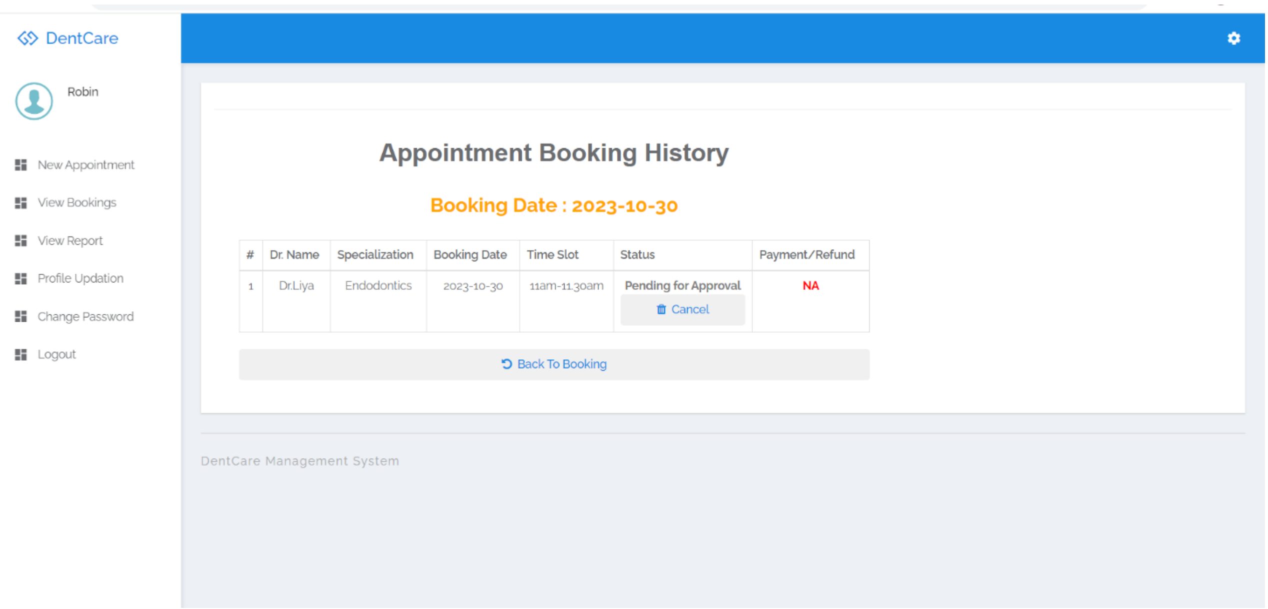
2023-10-30

Tony [Paediatric Dentistry]

2pm-2.30pm

DentCare Management System

View Bookings:



Appointment Booking History

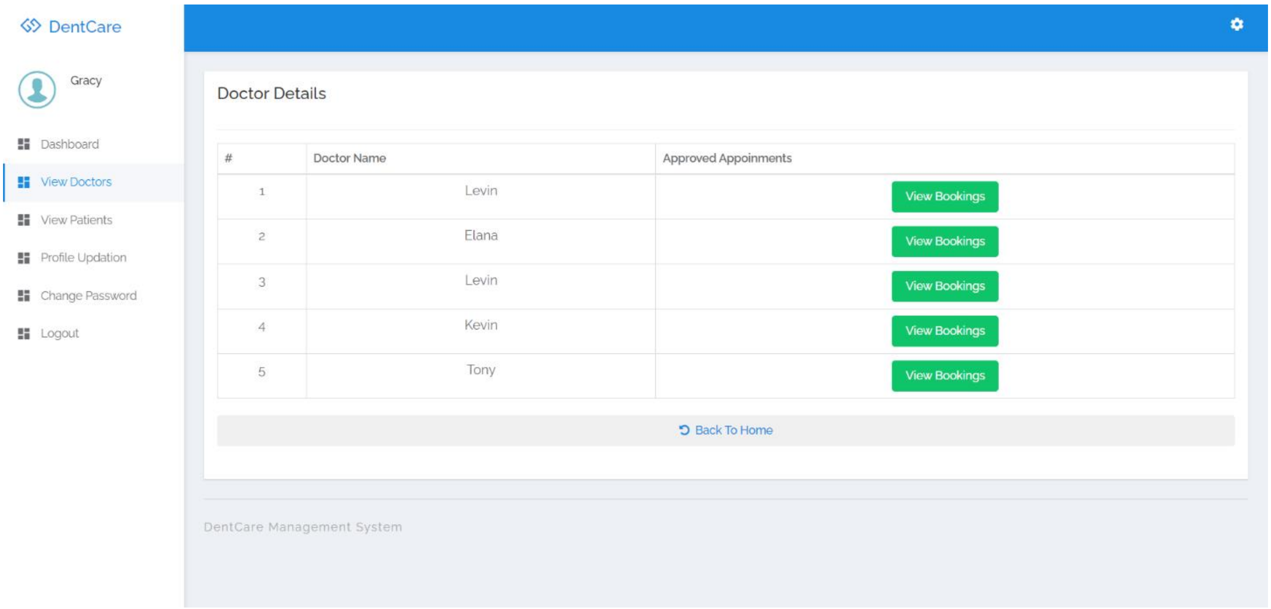
Booking Date : 2023-10-30

#	Dr. Name	Specialization	Booking Date	Time Slot	Status	Payment/Refund
1	Dr.Liya	Endodontics	2023-10-30	11am-11.30am	Pending for Approval Cancel	NA

DentCare Management System

STAFF:

View Doctors:



Doctor Details

#	Doctor Name	Approved Appoinments
1	Levin	View Bookings
2	Elana	View Bookings
3	Levin	View Bookings
4	Kevin	View Bookings
5	Tony	View Bookings

Back To Home

DentCare Management System