**Dental Clinic System**

Abstract:

* My project “Dental Clinic System” is a web application from which the users can know the information about the services provided by the dentistry online.
* This project provides an interface for the users to get register themself and book the appointment for the services done in the dental clinic.
* Users can view all the services provided by the clinic in detail from the website within a minute at anywhere using the internet.
* The user can easily communicate with the Doctor for asking in detail information about the services provided by them. This project makes all the services provided by the clinic easier.

Acknowledgement:

Declaration:

Table Of Contents:

List Of Tables:

List Of Figures:

CHAPTER:1 INTRODUCTION.

* 1. Background:
* Presently the clinic functionalities are done manually. That is, if the patient wants to consult a Doctor he/she can visit there till his chance called.
* An appointments are distributed directly.
* The work in the clinic is done manually, paper works and registers are maintained for the patients.
* The main disadvantage is time consuming.
* Limitation of existing system is, if patient loses their receipt, difficulties to find out the patient token assign.
* To defect this limitations we do computer based web application.

1.2 Objectives:

* The aim of this system is to improve the management of the dental clinic and reduce the human efforts.
* The main objective of the project on Dental Clinic System is to manage the details of Dental Clinic, Doctor, Patients, Tests, Appointments.
* The objective of this system is to ensure the workflow in the clinic will be more systematic and organize.
* It tracks all the details about the patient, Tests and Appointments.

1.3 Purpose, Scope and Applicability:

1.3.1 Purpose:

* Our purpose at creating this application is that it will automate the process in Dental Clinic.
* Using this it is possible to keep the track of a database for clinic like information about the patient and appointment schedule, personal records, previous medical records, treatment records, prescription, case reports.
* This project is being created for collecting, managing, saving, and retrieval of medical information for the patients, and for creating reports for the patients.
  + 1. Scope:
* The scope of this project is patient and dentist/consultant management system of dental clinic and has to fulfill all their needs such as registering the patients and capturing their data.
* It may help in collecting perfect management details. In a very short period of time, the collection will be obvious, simple and sensible.
* It also helps in all current works related to Dental Clinic System.
* The level of accuracy in the proposed system will be higher. All the operation would be done correctly and it ensures that whatever information coming from the center is accurate.
* It will reduce the redundancy like no information is repeated in the storage. This would assure the economic use of storage space and consistency in the data stored.
* The retrieval of any type of information would be available whenever it is required.
* It will also reduce the cost of collecting the management and the collection procedure will go on smoothly.
  + 1. Applicability:
* Dental Clinic System is an automated system designed and developed precisely for the smooth and efficient conduct of the procedure followed in [**Dental Clinics**](https://en.wikipedia.org/wiki/Dental_care).
* It is provisioned with logins for user under profile as a patient for the Registration.
* The application can be made accessible to the outside world on the internet through the website.
* Appointment booking and payment gateway can be enabled directly to the patients with necessary notifications to the employees involved, along with the doctor.
* This system can used by the Doctors in their Dental Clinic.
  1. Achievements:
  2. Organization of Report:

CHAPTER:2 SURVEY TECHNOLOGIES

* 1. **Front End:**
* The Front End of a software program or website is with which the user interacts.
* The Front End is a Graphical Interface with which the user interact.
* It is the interface design and the programming that makes the interface function.
* The front end of a software or [website](https://techterms.com/definition/website) should be intuitive and easy to use.

The following programming languages can be used as a Front End Language to develop “Dental Clinic System”:

1] C++:

* C++ is an Object-Oriented Programming Language. This is the most important feature of the C++.
* C++ is a simple language that is it provides structured approach, rich set of library functions, data types etc.
* C++ can be the base language for many other programming languages that supports the feature of object-oriented programming.
* C++ program can support [unions](https://www.geeksforgeeks.org/union-c/) and [structures](https://www.geeksforgeeks.org/structures-c/) that are a mix of stand-alone and put-together files.
* It uses the standard C++ application mentioned as **“.cpp”**.
* C++ uses the reserved library word mentioned as “[goto](https://www.geeksforgeeks.org/goto-statement-in-c-cpp/)” that’s the same as Java’s continue, or break commands.
* C++ uses multi-paradigm programming. Paradigm is planning and programming.
* It supports Exception Handling. It uses “cin” and “cout” for printing input and output statements.

2] C#:

* C# is a general-purpose, object-oriented programming language developed by Microsoft.
* C# enables you to develop different types of application such as Windows Based Application, Console Based Application, Web Based Application.
* C# was especially developed for .NET Framework.
* Interoperability enables the C# programs to do almost anything that a native C++ application can do.
* C# is a scalable and updateable language.
* In C#, a very efficient system installed that collects and erases garbage automatically present on the system.
* C# language has a rich class of libraries that makes functions easy to be implemented.

3] Java:

* Java is an [object-oriented](https://www.javatpoint.com/java-oops-concepts) programming language. Everything in Java is an object.
* Java is platform independent because it is different from other languages like [C](https://www.javatpoint.com/c-programming-language-tutorial), [C++](https://www.javatpoint.com/cpp-tutorial), etc.
* Java syntax is based on C++.
* Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behaviour.
* Java is a simple programming language since it is easy to learn and easy to understand.
* Java is a robust programming language since it uses strong memory management.
* Java uses a multi-threaded environment in which a bigger task can be converted into various threads and run separately.
* Java is a secured programming language because it doesn't use explicit pointers.

4] VB.Net:

* VB.Net or Visual Basic .Net is an object oriented programming language.
* VB.Net is one of the top programming languages with simple structural syntax for programming the code blocks.
* It is a platform-independent programming hence it could be compiled on a various of computer platforms and can be run of different OS.
* VB.Net is a part of .NET Framework.
* VB.Net follows the concept of a component.
* VB.Net syntaxes are easy to learn.

I will be using Front End as Asp.Net with C#.

* 1. **Back End:**
* Back End refers to any part of a software program or website that users do not see.
* The backend is the "data access layer”.
* Back End development is all about making the apps render server-side.
* The back end of a website consists of an application, and a database.
* Back End stores and arranges data, and also makes sure everything on the client-side of the website works fine.
* The parts developed by backend designers are indirectly accessed by users through a front end applications.

The following programming languages can be used as a Back End Language to develop “Dental Clinic System”:

1] MySQL:

* MySQL is a Open-Source Database, which uses Structured Query Language.
* MySQL holds the ACID (Atomicity, Consistency, Isolation, Durability) property, and also allows distributed multi-version support.
* MySQL can run at high speed and also provide master and slave replication configuration and also provide cluster servers.
* MySQL is a Relational Database management System.
* MySQL supports powerful mechanisms to ensure the security of the data.
* MySQL uses a very fast thread-based memory allocation system.
* MySQL is very easy to install, it’s also an easy database to work with.

2] Oracle:

* Oracle Database is cross-platform. It can run on various hardware across operating systems including Windows Server, Unix and Linux.
* Oracle Database allows you to quickly and safely store and retrieve data.
* Oracle uses the logical data structure to store data.
* The memory caching architecture in oracle allows you to scale up a very large database that still can perform at a high speed.
* Oracle ensure the integrity of the data in case of system failure.

3] Firebase:

* Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents.
* Firebase allows us to deliver and receive messages in a more reliable way across platforms.
* Firebase auth has a built-in email/password authentication system.
* Firebase includes an easy-to-use hosting service for all of your static files.
* Firebase Storage provides a simple way to save binary files.

4] MongoDB:

* MongoDB is a document database, which means it stores data in JSON like documents.
* MongoDB is a document-oriented NoSQL database used for high volume data storage.
* MongoDB is a source-available cross-platform.

I will be using MySQL as Back End in my project.

* 1. **Survey:**

1. Have you ever visited dental clinic before?
2. Yes b. No
3. Have you faced any Dental Health Problem?

a. Yes b. No

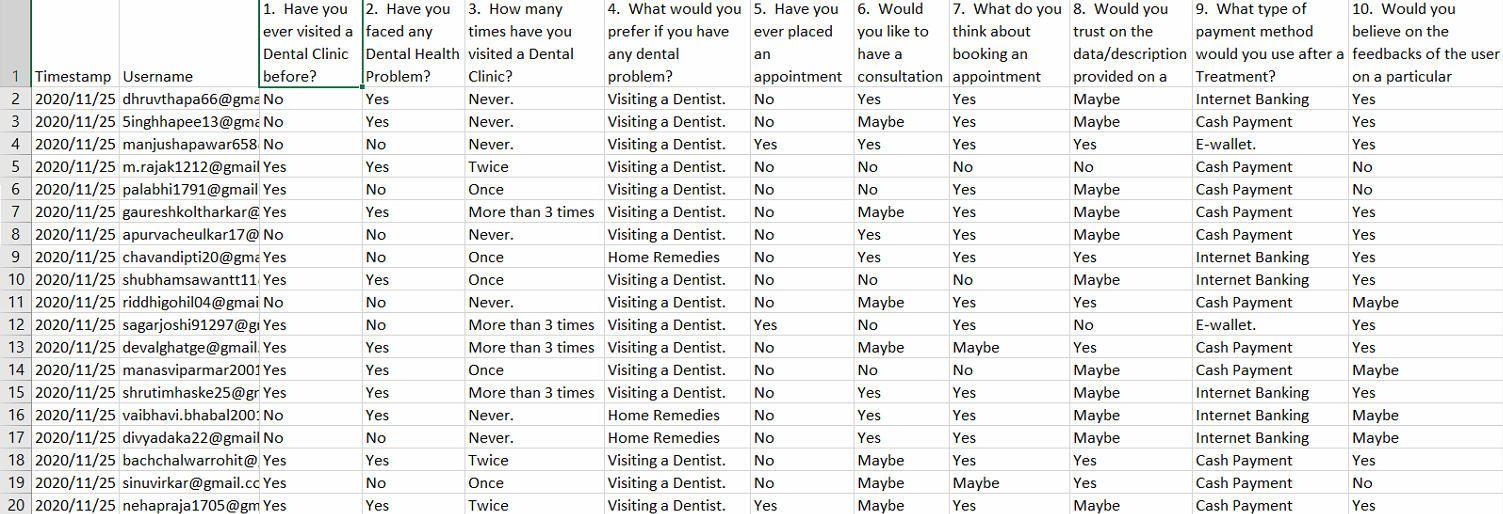
1. How many times have you visited a Dental Clinic?
2. Once b. Twice c. More than 3 times d. Never
3. What would you prefer if you have any dental problem?
4. Home remedies b. Visiting a Dentist
5. When you book an offline appointment, you have to wait at the dentist office for your turn. What do you feel?
6. It is Time Consuming b. It is not Time Consuming.
7. Have you ever placed an appointment related to dental issue online?
8. Yes b. No
9. Would you like to have a consultation online through chats/calls instead of meeting a doctor in a clinic?

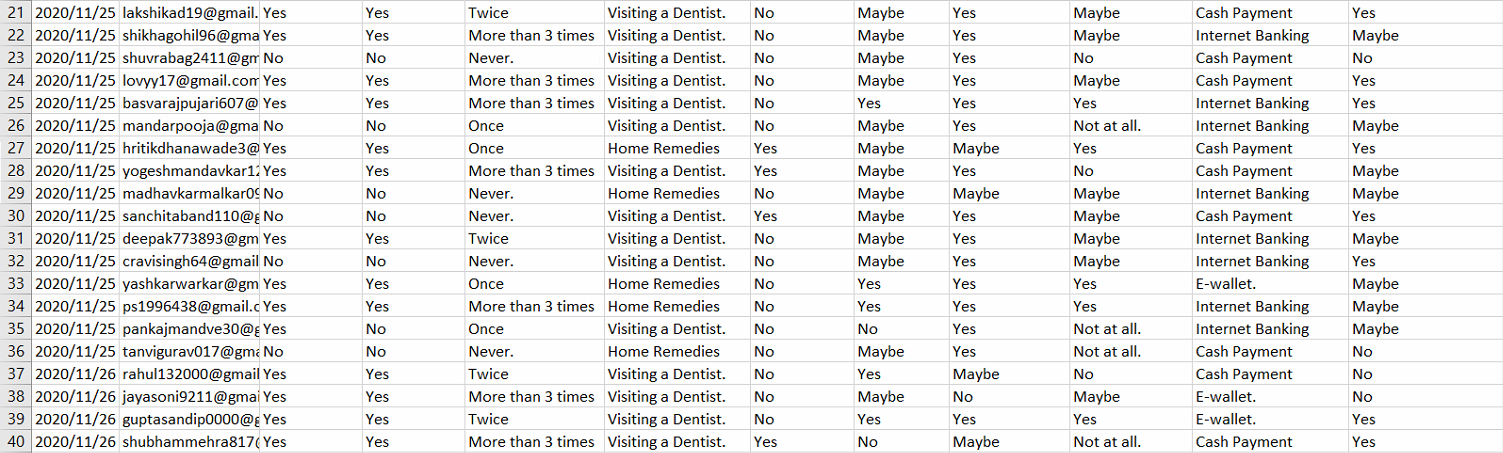
a. Yes b. Maybe c. No

1. What do you think about booking an appointment through a registration form online on a website is time saving?
2. Yes b. Maybe c. No
3. Would you trust on the data/description provided on a particular website?
4. Yes b. Maybe c. No d. Not at all
5. What type of payment method would you use after a Treatment?
6. Internet Banking b. Cash Payment c. E-Wallet

1. Would you believe on the feedbacks of the user on a particular website/system?
2. Yes b. Maybe c. No

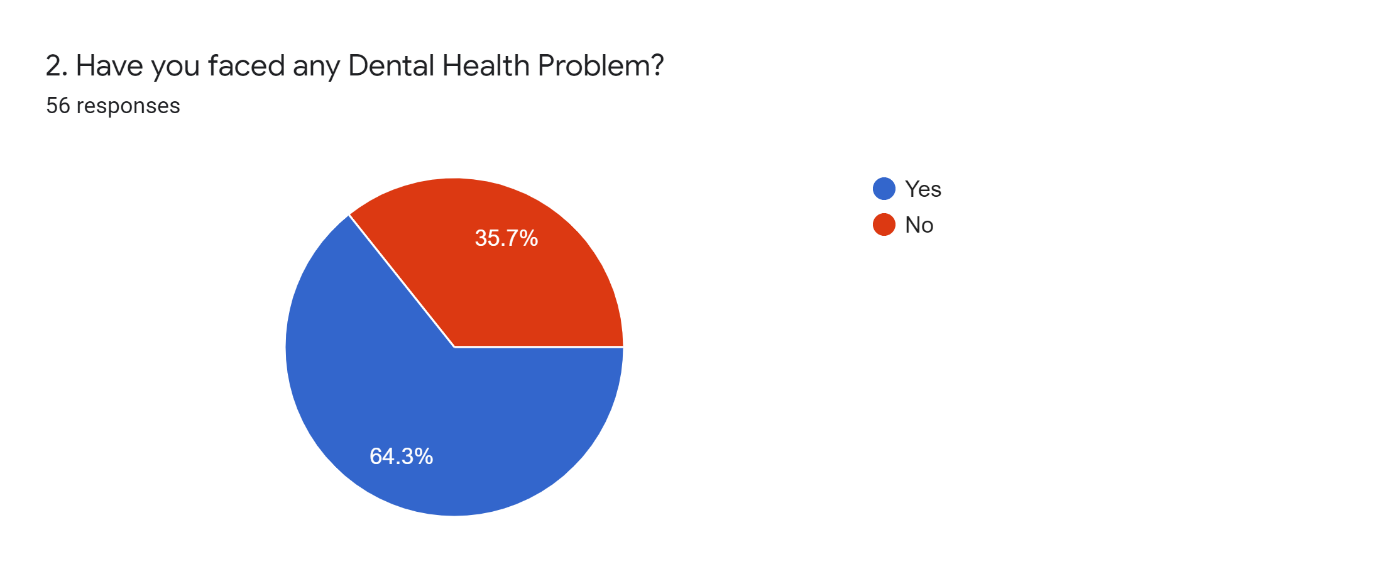
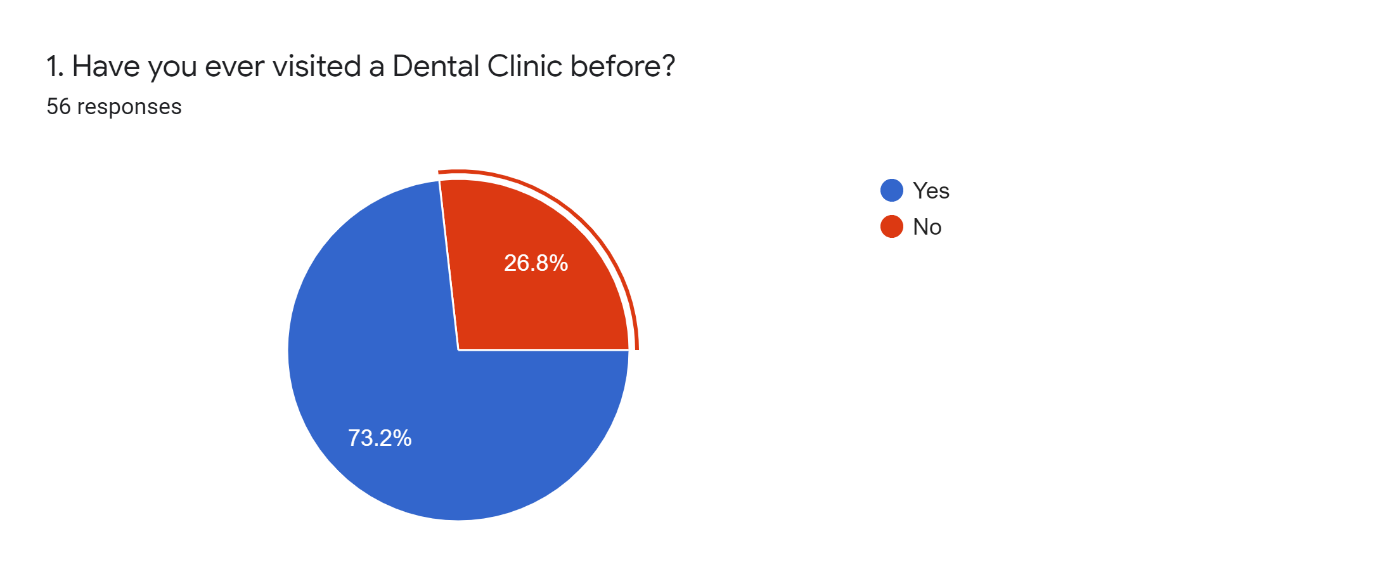
* Responses:

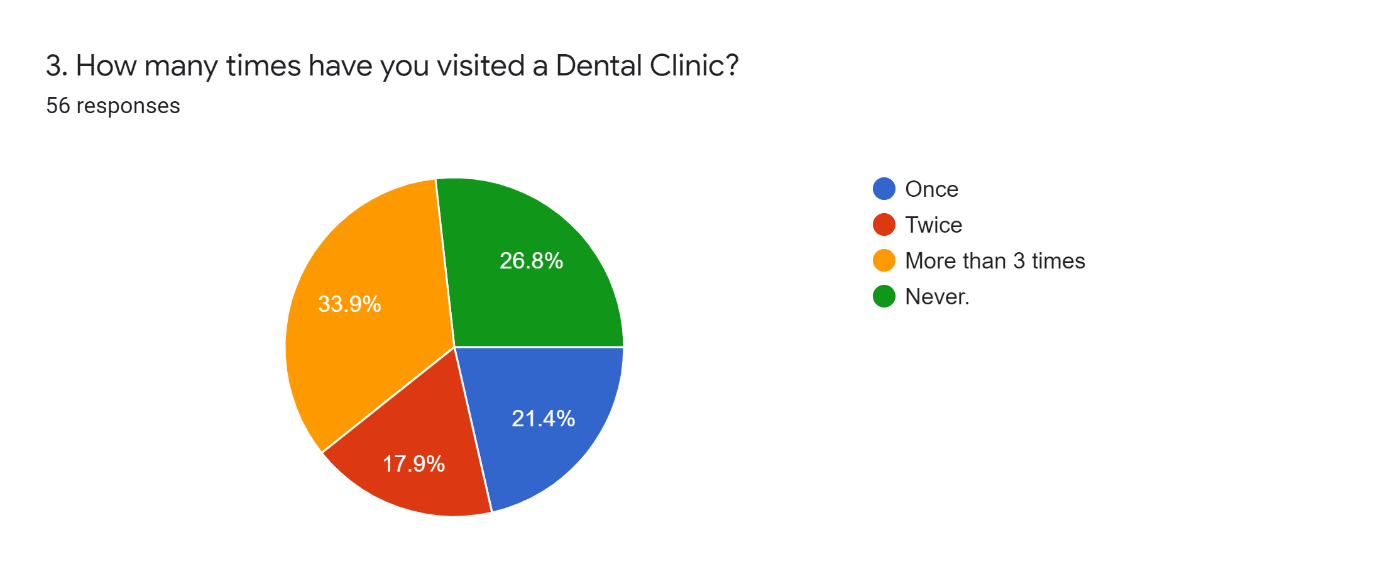


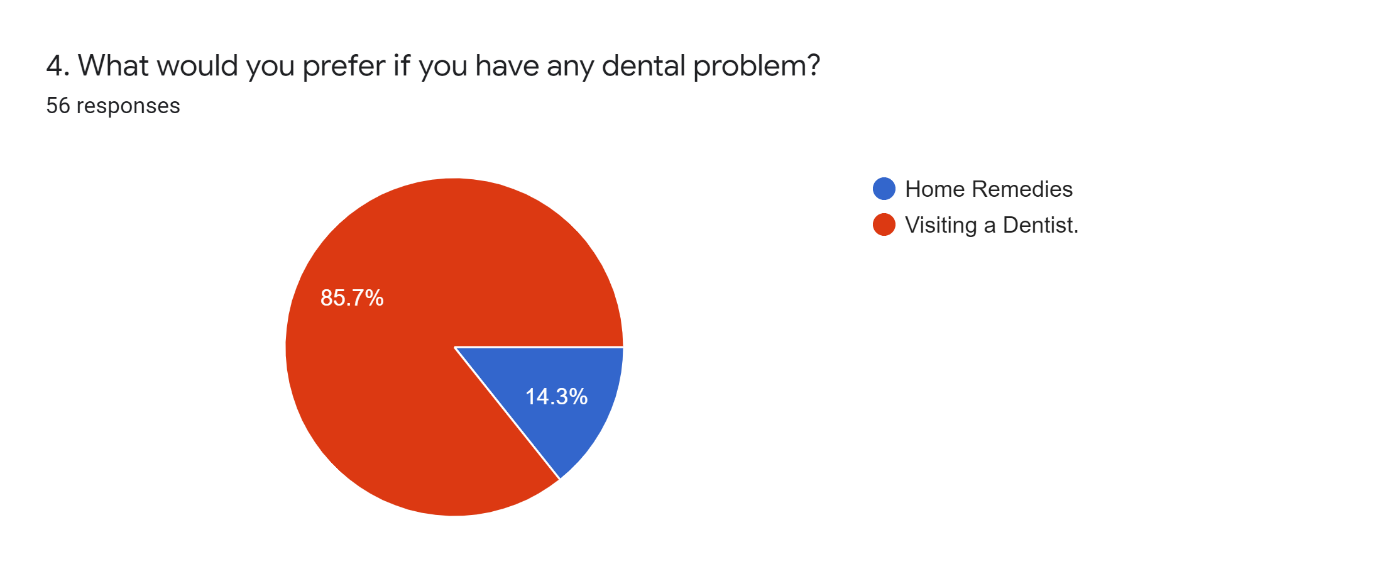


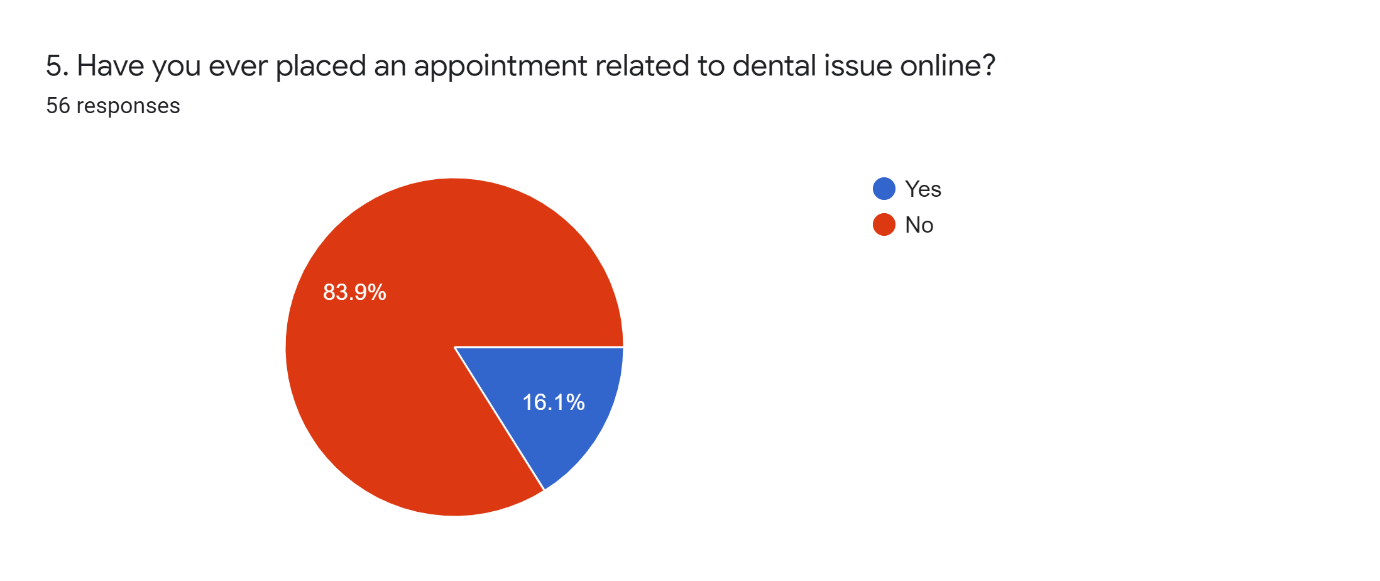


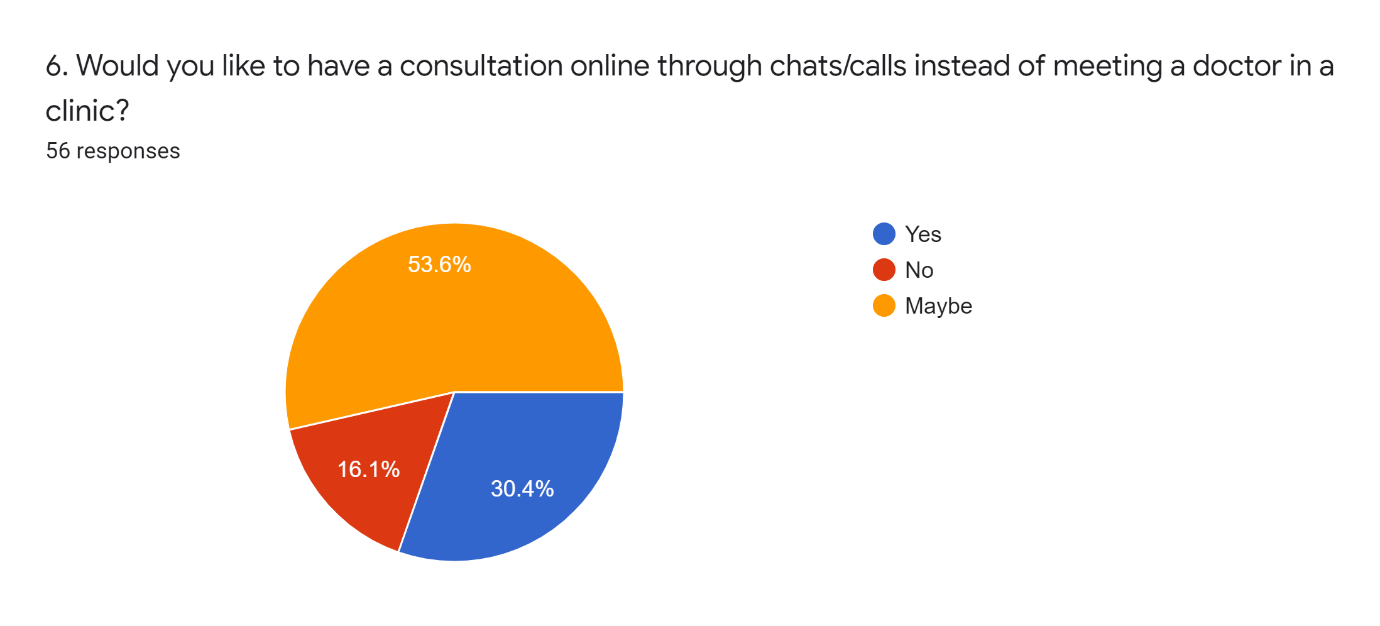
* Statistics:

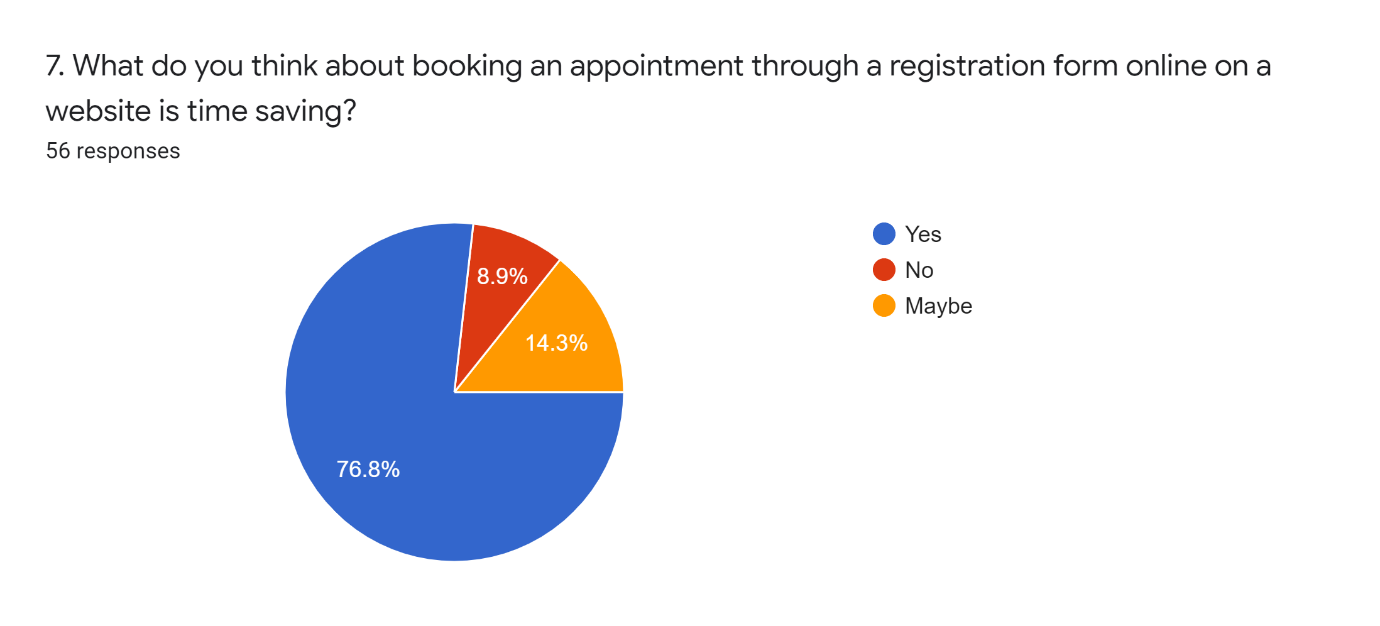


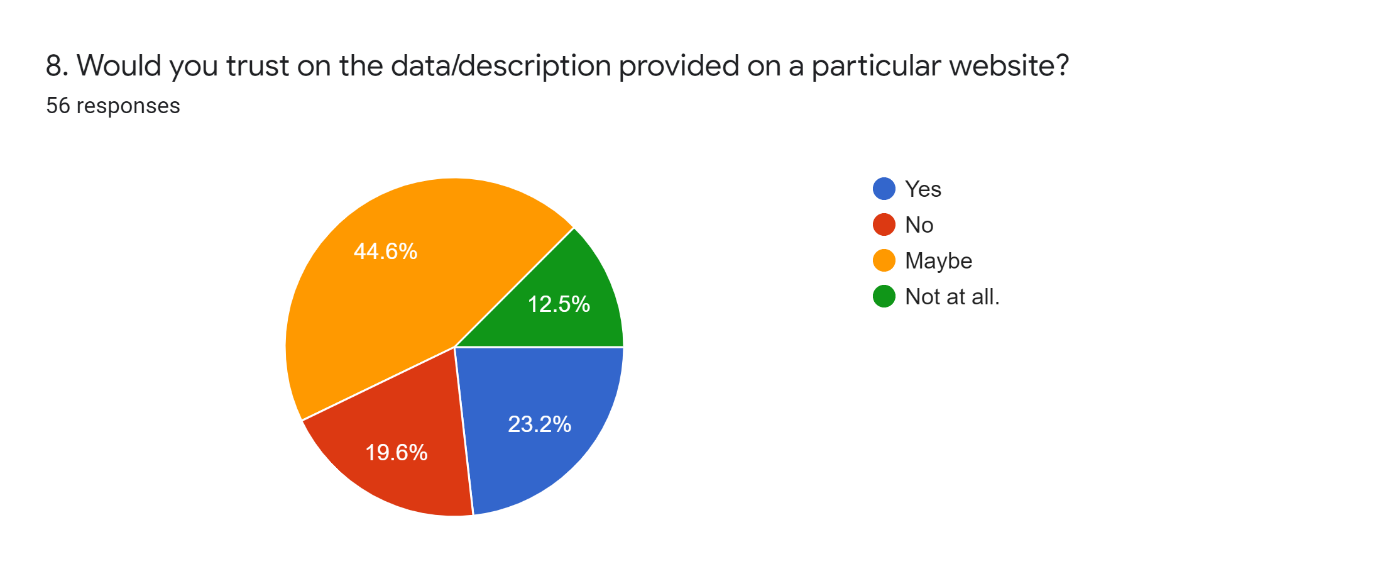
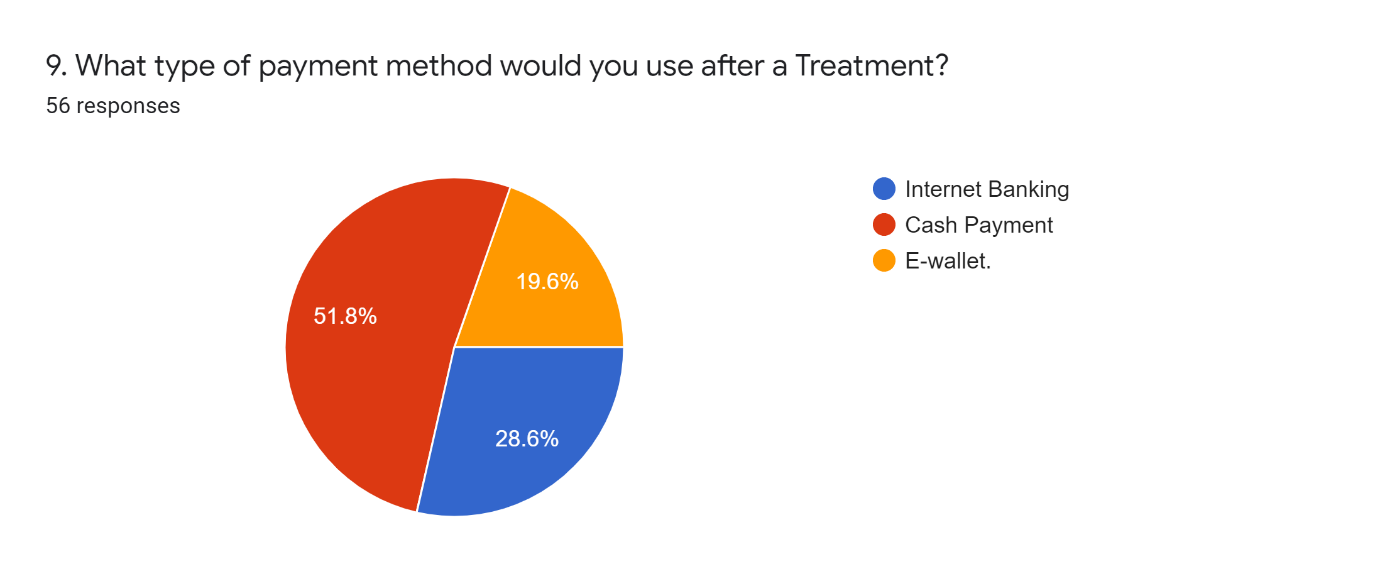


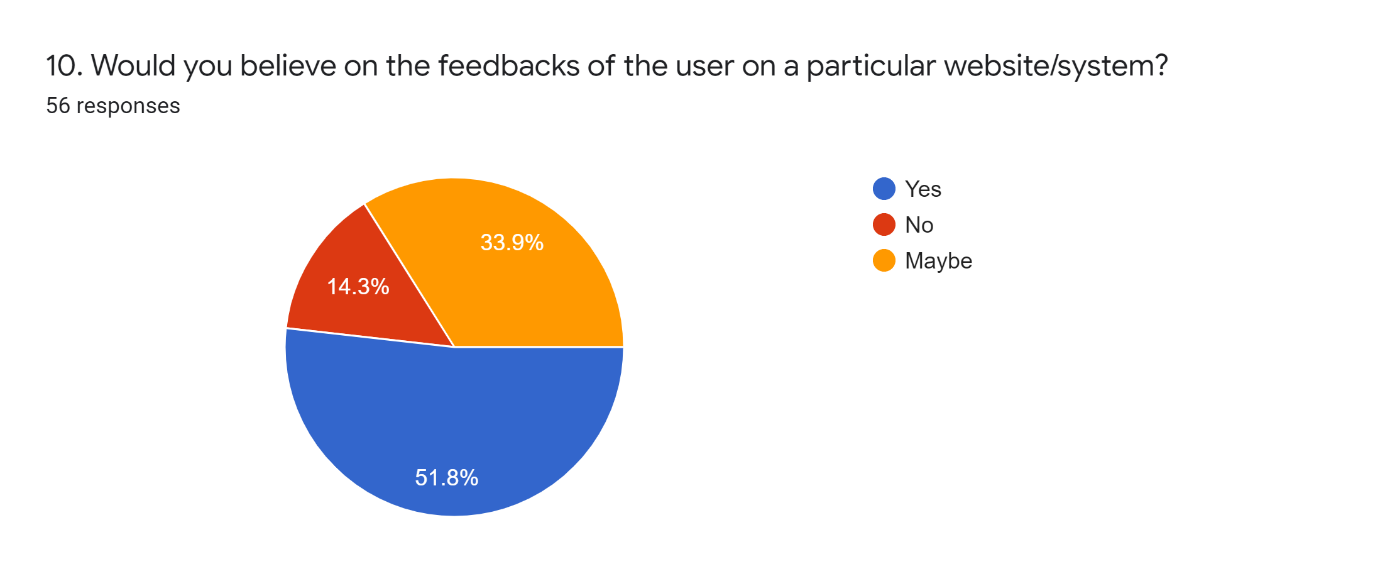










* Result:

1. In the above survey, 56 people have responded the survey in which 73.2% respondents have visited the dental care and 26.8% respondents have never visited any dental clinic before.
2. 64.3% of respondents have faced a dental health problem and around 35.7% respondents never faced any dental health problems.
3. From 100% respondents, 21.4% of respondents have visited dental clinic only once, 17.9% of respondents visited twice and 33.9% more than thrice.
4. In survey it is found that if anyone will suffer from dental problem, 85.7% respondents will visit dentist and 14.3% of respondents will use home remedies.
5. 83.9% of respondents have never placed any appointments online and 16.1 of respondents have placed their appointments online.
6. It is found that 30.4% would like to take consultation of a doctor on call/chats and 53.6% of respondents are not sure about taking a consultation through calls, while 16.3% of respondents would not like to take consultation through calls.
7. 76.8% of respondents thinks that booking an appointment through appointment form is time saving, while 14.3% of respondents are not sure that it is time saving and rest of the respondents thinks that it is not time saving.
8. It is found that 23.2 % of respondents will believe/trust on the data provided on particular website about their service and it is also found that 19.6% of responders won’t trust while 12.5% won’t trust at all and rest are not sure about the data.
9. After visiting the dental clinic and getting done treatment, 51.8% of respondents willing to pay their bill by Cash, and 28.6% are willing to pay their Internet Baking and the remaining respondents are willing pay by E-Wallet.
10. 51.8% of respondents will believe on the feedbacks on a particular website wile 14.3% will not believe and rest of the respondents are not sure to believe or not.

CHAPTER:3 REQUIREMENT AND ANALYSIS

3.1 Problem Definition:

3.2 Models:

3.2.1 Agile Methodology:

* Agile development methodologies are becoming popular due to their dynamic nature and easy adaptability to the situation.
* Agile methodology is a combination of iterative and incremental process models.
* In Agile methodology, the tasks are divided to time boxes to deliver specific features.
* The most popular Agile methods include Rational Unified Process, Scrum, Crystal Clear, Extreme Programming, etc.

3.2.2 Waterfall Model:

* The waterfall model is a classical model used in system development life cycle to create a system with a linear and sequential approach.
* The waterfall model is the oldest model and it was proposed by ‘Winston Royce’.
* This model is divided into different phases and the output of one phase is used as the input of the next phase.
* Every phase has to be completed before the next phase starts and there is no overlapping of the phases.
* The phases in waterfall model are: 1) Requirement Gathering, 2) System Analysis, 3) System Design, 4) Coding, 5) System Testing, 6) Coding, 7) System Testing, 8) System Implementation, 9) System Maintenance.

3.2.3 Spiral Model:

* The spiral model is an evolutionary and iterative software process model developed by Boehm.
* The spiral development model assumes that the customer requirements are obtained in the multiple iteration.
* Many big software are built by using Spiral Model.
* Spiral Model is represented in the form of cartesian diagram with 4 quadrants that represents 4 phases of software development.
* Spiral model finds all risk in the project and also finds future risks.
* Phases of Spiral model are: 1) Requirement Gathering, 2) Risk Analysis

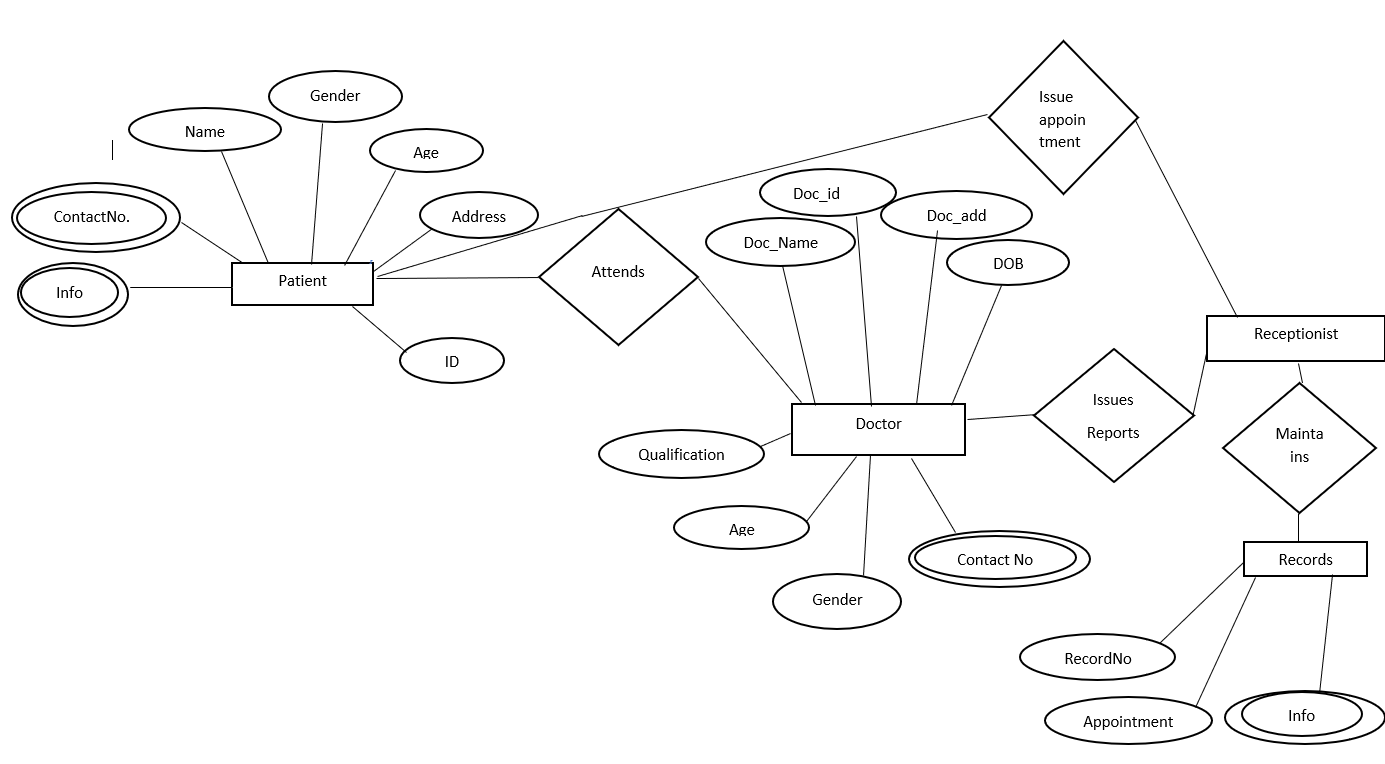
3) Engineering, 4) Customer Evaluation.

3.2.4 Incremental Model:

* Incremental Models are used in developing huge systems.
* The incremental model combines elements of waterfall model applied in Iterative fashion.
* Incremental model adapts all the phases of Iterative model.
* Incremental Model results in better testing, since testing each increment is easier than testing the entire system.
* Incremental Model provides the feedback to the client, which is useful for determining the final requirements of the system.

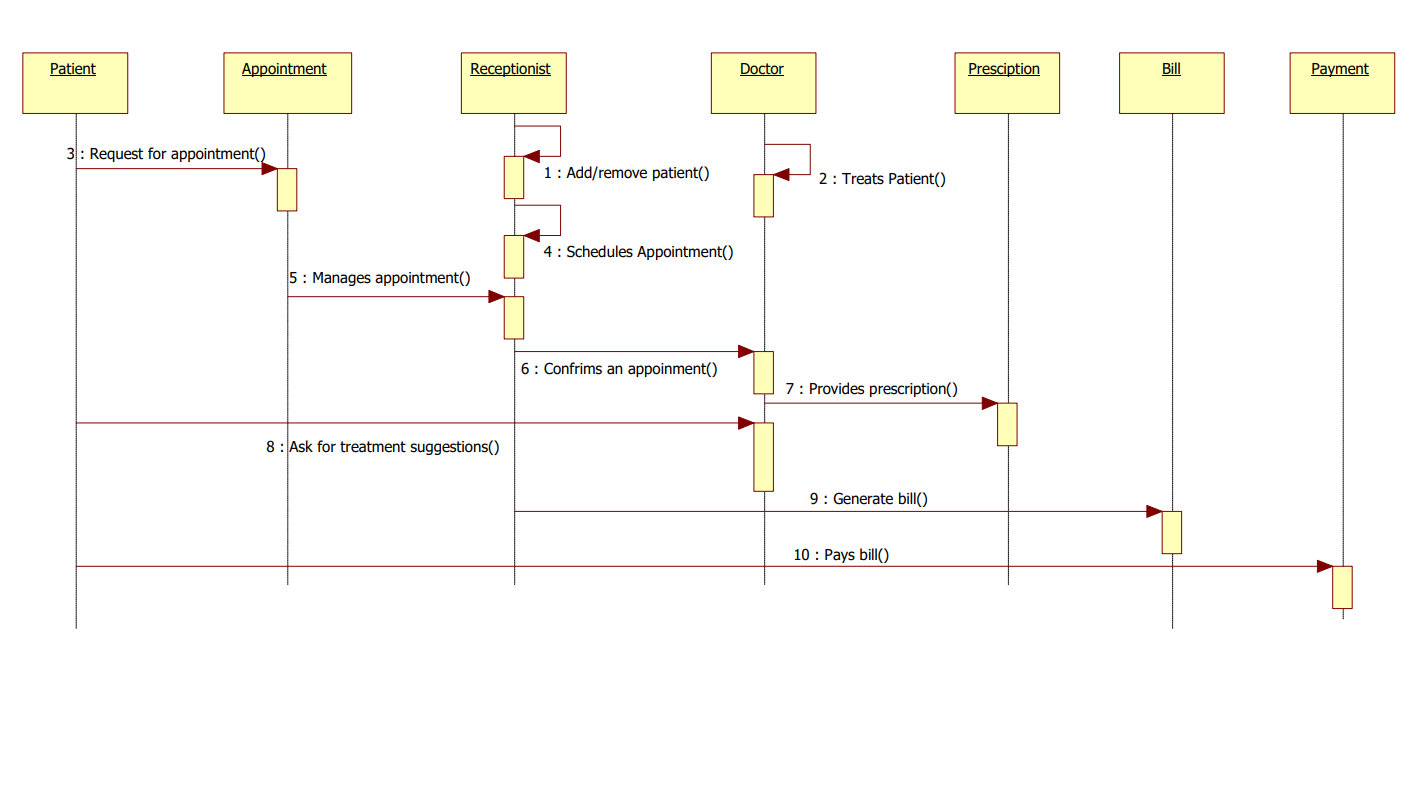
3.3 Conceptual Models

3.3.1 Entity Relationship Diagram:



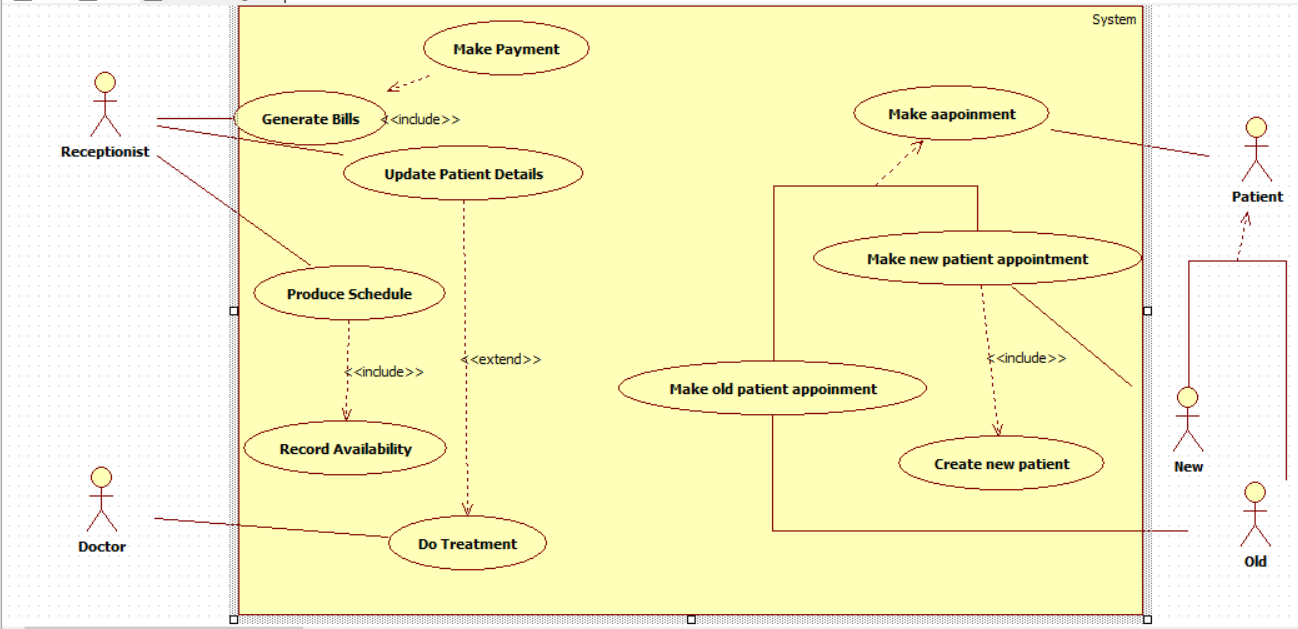
* An entity relationship model, also called as entity-relationship diagram, is a graphical representation of entities and their relationship to each other, typically used in computing in regard to the organization of data within databases or information systems.
* It shows the attributes in tables of the database.
* The entity-relationship data shows representation of database tables and the relations between patient, Doctor and the receptionist.

3.3.2 Sequence Diagram:



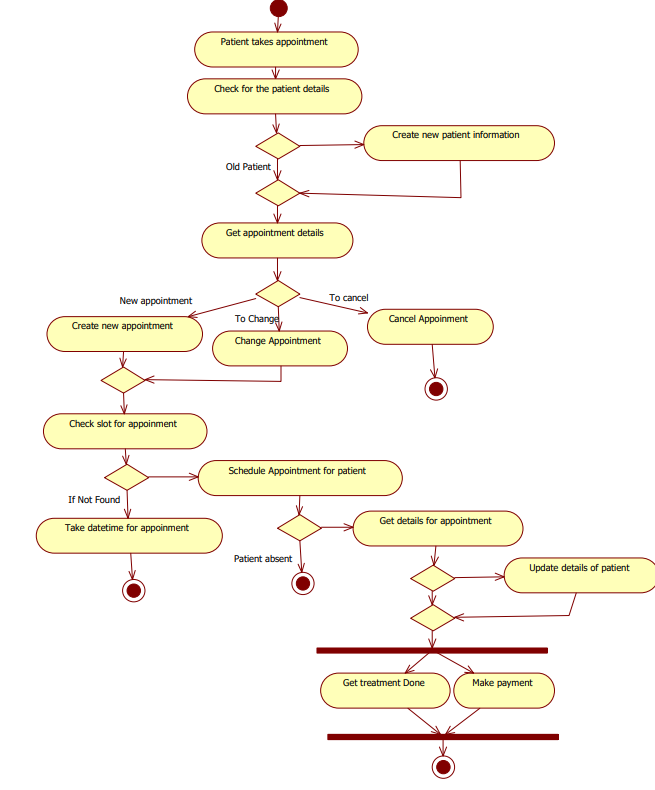
* The sequence diagram represents the flow of messages in the system and is also called as an event diagram.
* The sequence diagram shows the interaction login between the objects of clinic, appointments, patients, receptionists.

3.3.3 Use Case Diagram:



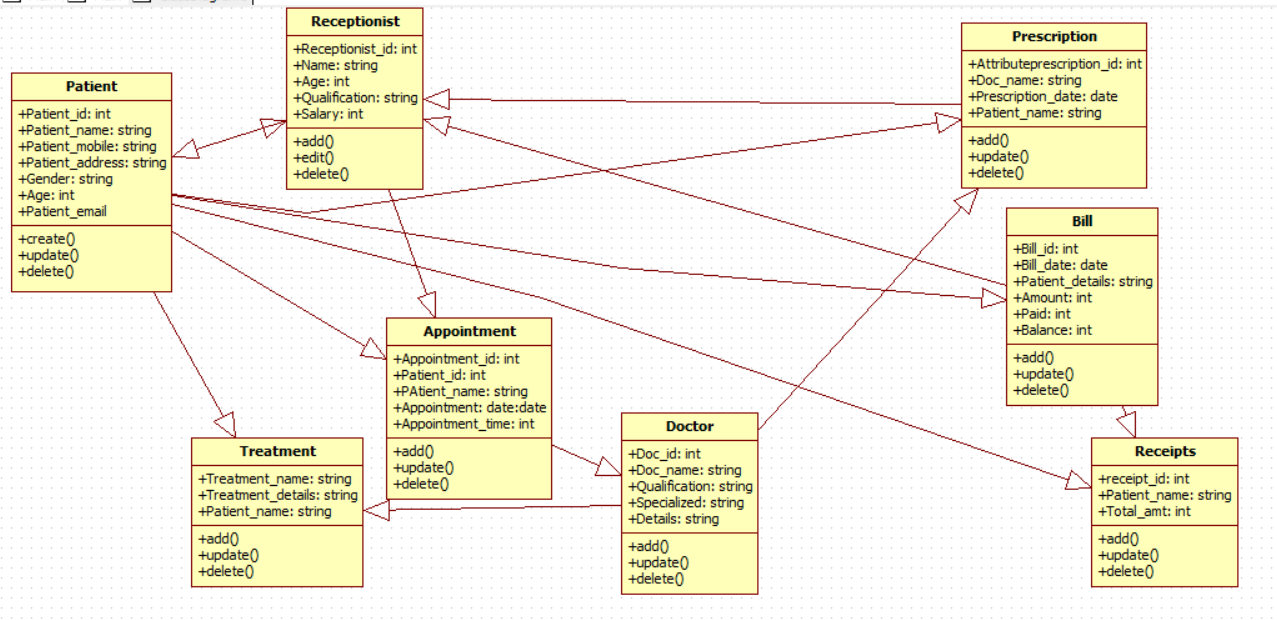
* A use case diagram is a graphic depiction of the interactions among the elements of a system.
* A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.
* The actors, usually individuals involved with the system defined according to their roles.
* The patient can perform, make an appointment and the receptionist can perform, can generate bills, produce schedule and update patients.

3.3.4 Activity Diagram:



* An activity diagram is an UML diagram which describes the aspects of the system.
* It is basically a flowchart diagram which describes the flow from one activity to other activity. It is also described as an operation of the system.
* The flow in the diagram can be sequential, branched and concurrent.
* It shows the flows between the activity of patient, receptionist and the doctor.
* The activities are as follows: Patient activity, Doctors activity, Receptionist activity.

3.3.5 Class Diagram:



* A class diagram in the Unified Modelling Language (UML) by showing

is a type of static structure diagram the describes the structure of a system

showing the system’s classes, their attributes, operations, and the

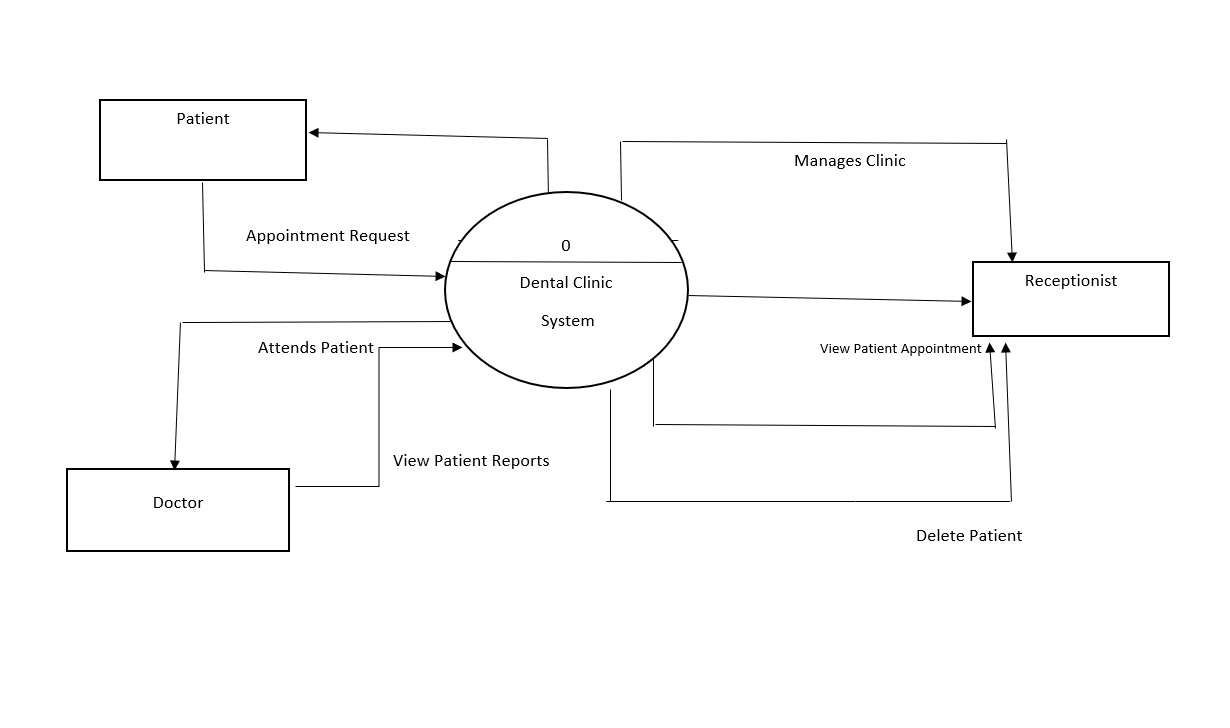
relationship among objects.

* The main classes are Patient, Receptionist, Doctor, Treatment and Treatment.

3.3.6 Data Flow Diagram:

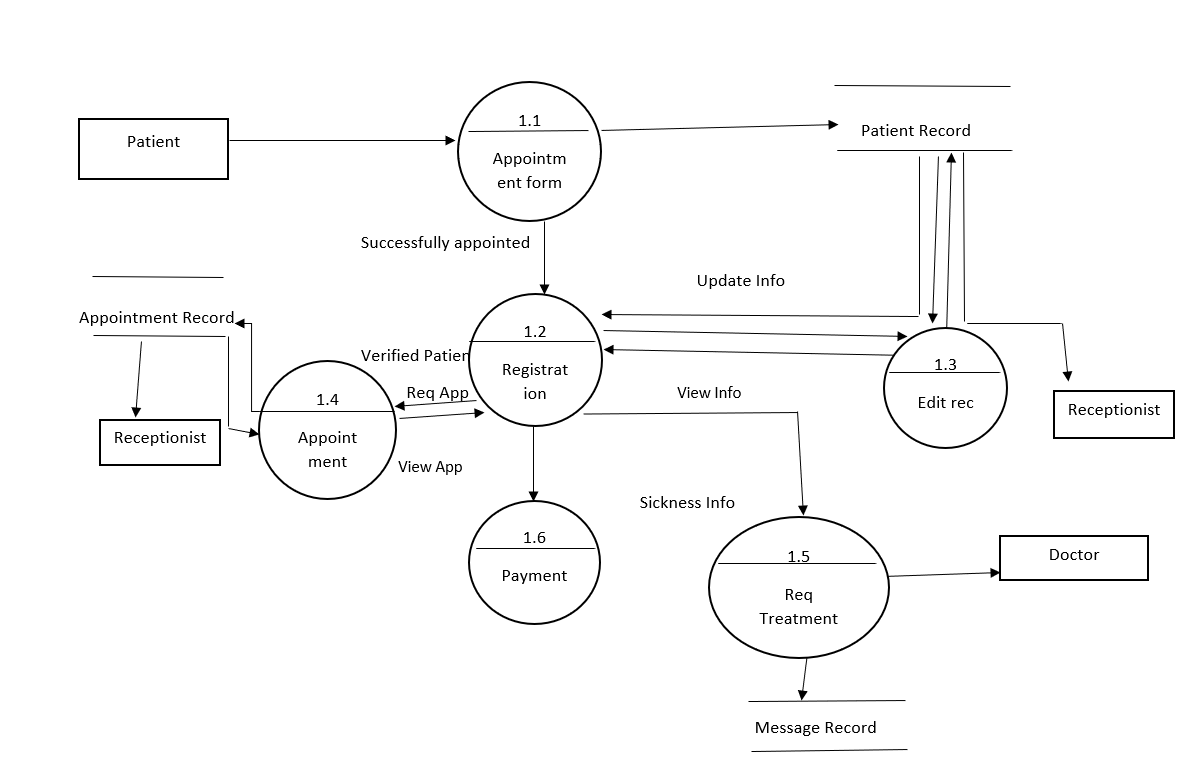
* A data flow diagram is a graphical representation of the flow of data through an information system.
* It shows how information is input to and output from the system, the sources and destinations of that information, and where that information is stored

LEVEL 0:



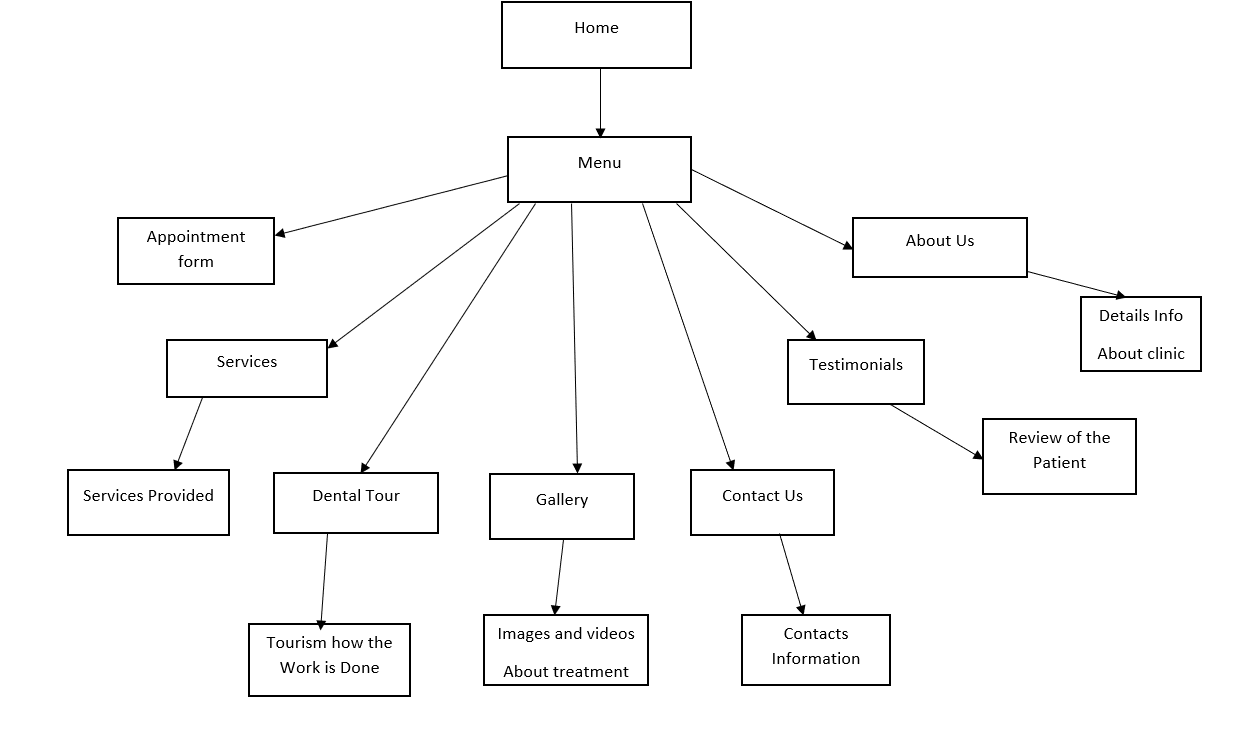
* Highest abstraction level DFD is known as level 0 DFD, which depicts the entire information system as one diagram concealing all the underlying details.
* It is the 0 level DFD of Dental Clinic System, where we have described about the high-level processes of the system.
* It represents the overview of a whole system that is being analysed or modelled.

LEVEL 1:



* The Level 0 DFD is broken down into more specific, Level 1 DFD. Level 1 DFD depicts basic modules in the system and the flow of data among various modules.
* Level 1 DFD also mentions the basic processes and sources of information.
* Level 1 DFD shows how the system is divided into subsystem, each of which deals with one or more of the data flows.
* It also shows the internal data stores of Appointment, Receptionist and the Doctor.

3.3.7 Flowchart:



* It represents the flow of the system as a whole and also shows how the system will work.
* It shows the flow from the start to end.

3.4 Planning and Scheduling:

3.4.1 Gantt Chart:

3.4.2 Pert Chart:

3.5 Software and Hardware:

3.5.1 Software:

* Windows 10.
* Visual Studio (2015).
* MySQL Server.

CHAPTER:4 SYSTEM DESIGNING

4.1 Pseudo Code:

Login:

Begin

if(user)

{

appointment page;

}

else if(admin)

{

dashboard page;

}

else

{

Invalid Phone no. and Password

}

end if

4.2 Software Testing:

* Software Testing is a method to check whether the actual software product matches expected requirements of the user and the software is free from defects.
* Testing is conducted at the phase level in software development life cycle or at module level in program code.
* The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.
* **Software Testing is important** because if there are any bugs or errors in the software, it can be identified early and can be solved before delivery of the software product.
* Software Testing involves execution of software or system using manual or automated tools.

4.3 Designing:

CHAPTER:5 IMPLEMENTATION AND TESTING

5.1 Model:

5.1.1 Agile Methodology:

* Agile development methodologies are becoming popular due to their dynamic nature and easy adaptability to the situation.
* Agile methodology is a combination of iterative and incremental process models.
* In Agile methodology, the tasks are divided to time boxes to deliver specific features.
* The most popular Agile methods include Rational Unified Process, Scrum, Crystal Clear, Extreme Programming, etc.

5.1.2 Waterfall Model:

* The waterfall model is a classical model used in system development life cycle to create a system with a linear and sequential approach.
* The waterfall model is the oldest model and it was proposed by ‘Winston Royce’.
* This model is divided into different phases and the output of one phase is used as the input of the next phase.
* Every phase has to be completed before the next phase starts and there is no overlapping of the phases.
* The phases in waterfall model are: 1) Requirement Gathering, 2) System Analysis, 3) System Design, 4) Coding, 5) System Testing, 6) Coding, 7) System Testing, 8) System Implementation, 9) System Maintenance.

5.1.3 Spiral Model:

* The spiral model is an evolutionary and iterative software process model developed by Boehm.
* The spiral development model assumes that the customer requirements are obtained in the multiple iteration.
* Many big software are built by using Spiral Model.
* Spiral Model is represented in the form of cartesian diagram with 4 quadrants that represents 4 phases of software development.
* Spiral model finds all risk in the project and also finds future risks.
* Phases of Spiral model are: 1) Requirement Gathering, 2) Risk Analysis

3) Engineering, 4) Customer Evaluation.

5.1.4 Incremental Model:

* Incremental Models are used in developing huge systems.
* The incremental model combines elements of waterfall model applied in Iterative fashion.
* Incremental model adapts all the phases of Iterative model.
* Incremental Model results in better testing, since testing each increment is easier than testing the entire system.
* Incremental Model provides the feedback to the client, which is useful for determining the final requirements of the system.

5.2 Coding:

**Login:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using MySql.Data.MySqlClient;

using MySql.Data;

using MySql.Data.Types;

namespace Integration.admin

{

public partial class login : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

UnobtrusiveValidationMode = UnobtrusiveValidationMode.None;

Label3.Visible = false;

}

protected void Button2\_Click(object sender, EventArgs e)

{

string connectionString;

object obj;

connectionString = "server=localhost;port=3307;user=root;database=dentalcare;password=Jyoti\_8451;pooling=false";

MySqlConnection conn = new MySqlConnection(connectionString);

conn.Open();

try

{

MySqlCommand cmd = new MySqlCommand("SELECT category FROM register WHERE phone=@phone AND password=@password", conn);

cmd.CommandType = System.Data.CommandType.Text;

cmd.Parameters.AddWithValue("@phone", TextBox1.Text);

cmd.Parameters.AddWithValue("@password", TextBox2.Text);

// int count = Convert.ToInt32(cmd.ExecuteNonQuery());

// String category = Convert.ToString(cmd.ExecuteReader());

obj = cmd.ExecuteScalar();

if (Convert.ToString(obj).Equals("user"))

{

Response.Redirect("~/appointment.aspx", false);

Context.ApplicationInstance.CompleteRequest();

}

else if (Convert.ToString(obj).Equals("Admin"))

{

Response.Redirect("~/dashboard.aspx", false);

Context.ApplicationInstance.CompleteRequest();

}

else

{

Label3.Visible = true;

}

}

catch (Exception ex)

{

Console.Write("Exception : " + ex.StackTrace);

}

finally

{

conn.Close();

}

}

}

}

**Registration:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using MySql.Data.MySqlClient;

using MySql.Data;

using MySql.Data.Types;

using System.Data;

namespace Integration

{

public partial class register : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

UnobtrusiveValidationMode = UnobtrusiveValidationMode.None;

}

protected void Button1\_Click(object sender, EventArgs e)

{

try

{

string connectionString;

//connectionString = "SERVER=localhost;" + "DATABASE=dentalcare; " +"UID=root;" + "PASSWORD=Jyoti\_8451;";

connectionString = "server=localhost;port=3307;user=root;database=dentalcare;password=Jyoti\_8451;pooling=false";

MySqlConnection conn = new MySqlConnection(connectionString);

// SqlConnection conn = new SqlConnection("Data Source=. ; Initial Catalog=dentalcare; Integrated Security=True");

conn.Open();

//SqlCommand cmd = new SqlCommand("insert into register" + "(firstname,lastname,email,phone,gender,password) values (@fname,@lname,@email,@phone,@gender,@password)", conn);

MySqlCommand cmd = new MySqlCommand("insert into register" + "(firstname,lastname,email,phone,gender,password,category) values (@fname,@lname,@email,@phone,@gender,@password,'user')", conn);

cmd.Parameters.AddWithValue("@fname", TextBox1.Text);

cmd.Parameters.AddWithValue("@lname", TextBox2.Text);

cmd.Parameters.AddWithValue("@email", TextBox3.Text);

cmd.Parameters.AddWithValue("@phone", TextBox4.Text);

cmd.Parameters.AddWithValue("@gender", DropDownList1.SelectedItem.Value);

cmd.Parameters.AddWithValue("@password", TextBox6.Text);

cmd.ExecuteNonQuery();

conn.Close();

Label8.Text = "Registered Successfully";

}

catch (Exception ex)

{

Console.Write("Exception : " + ex.StackTrace);

}

}

}

}

**Appointment:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using MySql.Data.MySqlClient;

using MySql.Data;

using MySql.Data.Types;

namespace Integration

{

public partial class appointment : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

UnobtrusiveValidationMode = UnobtrusiveValidationMode.None;

}

protected void Button2\_Click(object sender, EventArgs e)

{

try

{

string connectionString;

connectionString = "server=localhost;port=3307;user=root;database=dentalcare;password=Jyoti\_8451;pooling=false";

MySqlConnection conn = new MySqlConnection(connectionString);

conn.Open();

MySqlCommand cmd = new MySqlCommand("insert into appoint" + "(datetime,name) values (@datetime,@name)", conn);

cmd.Parameters.AddWithValue("@datetime", TextBox2.Text);

cmd.Parameters.AddWithValue("@name", TextBox1.Text);

cmd.ExecuteNonQuery();

Label7.Text = "Appointment Fixed";

conn.Close();

}

catch (Exception ex)

{

Console.Write("Exception : " + ex.StackTrace);

}

}

}

}

5.3 Testing Approaches:

5.3.1 Unit Testing:

* Unit is the smallest part of the software which is testable.
* The purpose is to validate that each unit of the software code performs as expected.
* Unit Testing is done during the development of an application by the developers.
* Unit Testing performed by software developers themselves or their peers.
* Unit testing increases confidence in changing or maintaining code.

5.3.2 Whitebox Testing:

* **White Box Testing** is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security.
* White Box Testing tests internal coding and infrastructure of a software focus on checking of predefined inputs against expected and desired outputs.
* The white box testing is also known as clear box or transparent box.
* The clear box or white box or transparent box name denote the ability to see through the software's outer shell into its inner workings.
* The developers perform the white box testing and then send the software to the testing team.

5.3.3 Error Handling Testing:

* Error handling testing is a type of software testing that is performed to check whether the system is able to handle the errors that may happen in future.
* This type of testing is basically performed with the help of both developers and the testers.
* Application is expected to help user through error messages, if anything unexpected happens with the system.
* Error handling has a direct relationship with usability of an application, and a distant relationship with the security of the system.
* Error handling testing not only focuses on the determination of error but also focuses on the exception handling.
* When user tries to enter the wrong data, the system identifies the wrong entry and prevents such entry in the system.
* In this web-based application the suggestive messages are used in registration and login page:

1. If the user enters a invalid email address than the messages arises as “Invalid Email Address”.
2. If the user enters a password range less than 7 characters than the message arises as “password must contain at least & characters ”.

5.3.4 Graphical User Interface Testing:

* Graphical User Interface is most important part of the application along with the functionality.
* Graphical User Interface is also known as ‘GUI’ or ‘UI’ Testing.
* Other than the system type of software, most of the application have the user interface from where a user interacts with the system.
* Graphical User Interface Testing includes the following:
* All colours used in background, control colours and font colours have major impact on the user.
* All words, fonts and alignment used on the screen which would be read when the user is interacting with the application.
* Error messages and the information given to the user must be usable to the user. Message must guide the user to perform the correct action.

5.3.5 System Testing:

* System Testing represents the final testing done on a system before it is delivered to the customer.
* System testing validates that the entire system meets its functional/non-functional requirement as defined by the customer in software requirement specification.
* The criteria for system testing may involve an entire domain or selected part depending upon the scope of the testing.
* System testing goes through the following stages that is, Functional testing and User Interface Testing.
* Functional Testing intends to find whether all the function as per the requirement definition working or not.
* User Interface Testing may involve colours, navigations, spellings and fonts.

5.4 Test Cases:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case | Priority | Test Case/Objective | Step/Procedure | Input Test Data | Expected Result |
| TC01 | P1 | Adding a numeric value to the first name | Enter digits | Any numeric value | First name should contain alphabets. |
| TC02 | P1 | Adding a numeric value to the last name | Enter digits | Any numeric value | Last name should contain alphabets |
| TC03 | P1 | Adding multiple dots to the email id. | Enter multiple dots. | Dot Character  value | Invalid Email Address. |
| TC04 | P1 | Not adding ‘@’ character to the email id | Do not enter ‘@’ character | ‘@’ character value | Invalid Email Address. |
| TC05 | P1 | Adding invalid mobile number | Enter number less than or more than 10 digits | Value less than or more than 10 digits | Must contain at least 10 digits. |
| TC06 | P1 | Entering password | Enter password less than 7 characters | Value less than 7 characters | Must contain at least 10 digits. |
| TC07 | P1 | To verify password | Enter different password from assigned password | Password different from assigned password | Password not matching |
| TC08 | P1 | Registering with all the fields blank | Leaving all the fields blank | Null | Fields cannot be blank |
| TC09 | P1 | Verifying the email field by entering a valid email address | Entering valid email address | Valid email address | Registered successfully |
| TC10 | P1 | To verify login page by entering valid Phone number and password | Entering a valid Phone number and password | Valid Phone number and password | Logged in successfully and redirected to appointment page |

5.5 Modification and Improvements:

CHAPTER:6 RESULTS AND DISCUSSION

6.1 Test Reports:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case | Priority | Test Case/Objective | Step/Procedure | Input Test Data | Expected Result |  |
| TC01 | P1 | Adding a numeric value to the first name | Enter digits | Any numeric value | First name should contain alphabets. |  |
| TC02 | P1 | Adding a numeric value to the last name | Enter digits | Any numeric value | Last name should contain alphabets |  |
| TC03 | P1 | Adding multiple dots to the email id. | Enter multiple dots. | Dot Character  value | Invalid Email Address. |  |
| TC04 | P1 | Not adding ‘@’ character to the email id | Do not enter ‘@’ character | ‘@’ character value | Invalid Email Address. |  |
| TC05 | P1 | Adding invalid mobile number | Enter number less than or more than 10 digits | Value less than or more than 10 digits | Must contain at least 10 digits. |  |
| TC06 | P1 | Entering password | Enter password less than 7 characters | Value less than 7 characters | Must contain at least 10 digits. |  |
| TC07 | P1 | To verify password | Enter different password from assigned password | Password different from assigned password | Password not matching |  |
| TC08 | P1 | Registering with all the fields blank | Leaving all the fields blank | Null | Fields cannot be blank |  |
| TC09 | P1 | Verifying the email field by entering a valid email address | Entering valid email address | Valid email address | Registered successfully |  |
| TC10 | P1 | To verify login page by entering valid Phone number and password | Entering a valid Phone number and password | Valid Phone number and password | Logged in successfully and redirected to appointment page |  |

6.2 User Documentation:

6.3 Database Design/Table:

CHAPTER:7 CONCLUSION

7.1 Conclusion:

7.2 Limitation:

7.3 Scope of Future:

References: